DE HAAGSE HOGESCHOOL

LEVEL LEVEL
Full Service Digital Agency

Het realiseren van een security information & event management-systeem voor Level Level

Afstudeerverslag

Wouter Honselaar
De Haagse Hogeschool
Ten behoeve van Level Level B.V.
4 juni 2021
Versie 1.0.0

Referaat

Er vinden regelmatig hackaanvallen plaats op internet. Organisaties weten pas dat ze gehackt zijn door meldingen die ze krijgen. Dit kunnen meldingen zijn van gebruikers van applicaties die melden dat er vreemd gedrag optreedt (erg traag, etc.). Waar hacken vroeger nog alleen het doel had om een grap met iemand uit te halen, is dit in de jaren ‘90 veranderd naar informatiediefstal, verhindering van bedrijfsprocessen, chantage of financiële doelen.

Daarnaast vallen succesvolle hackaanvallen tegenwoordig minder op: een aanzienlijk deel van alle op internet aangesloten apparaten maakt onderdeel uit van een botnet. In 2016 is de Algemene Verordening Gegevensbescherming van kracht geworden (sinds 2018 van toepassing, er was sprake van een overgangsperiode), een wet die geldt in elk EU-land (Europese Unie, 2016). Deze wet beschrijft hoe er omgegaan moet worden met persoonsgegevens en andere gevoelige data. Goede logging kan hier enorm bij helpen.

Een belangrijk onderdeel van informatiebeveiliging is het op de hoogte gehouden worden van gebeurtenissen. In dit onderzoek staat beschreven aan welke eisen een SIEM voor een WordPress webhosting-organisatie precies moet voldoen. Een SIEM houdt relevante gebeurtenissen bij die betrekking hebben op informatiebeveiliging. Indien er sprake is van verdacht gedrag of een vermoedelijke hackaanval, wordt dit onmiddellijk gemeld aan beheerders die hier vervolgens op kunnen reageren. Daarnaast kunnen sommige SIEM-systemen ook automatisch reageren op verdacht gedrag, denk aan het blokkeren van al het verkeer vanaf een bepaald IP-adres.

Inhoudsopgave

Samenvatting

Dit verslag is het resultaat van het afstudeeronderzoek van een cyber security student. Middels gefaseerd onderzoek is antwoord gegeven op deelvragen en uiteindelijk ook op de hoofdvraag. De hoofdvraag luidt ‘Hoe kan een SIEM zodanig geïntegreerd worden in een webhostingarchitectuur dat deze zoveel mogelijk valide aanvallen opmerkt en daarop reageert?’.

Met behulp van een SIEM kunnen digitale beveiligingsgebeurtenissen op grote schaal verzameld en verwerkt worden. Daarnaast zijn SIEM’s tot op zekere hoogte in staat deze gebeurtenissen zelfstandig te classificeren. Indien er een urgente reeks aan gebeurtenissen optreedt, denk hierbij aan een poging om een systeem binnen te dringen door kwaadwillenden, dan wordt er een melding gegeven aan medewerkers die daar vervolgens actie op kunnen ondernemen. Doel van de opdracht is, na onderzoek, specificeren, ontwerpen en zo mogelijk realiseren van een SIEM als onderdeel van het nieuwe WordPress-platform van Level Level. Level Level ontwerpt, beheert, maakt en host WordPress websites. Een SIEM kan een goede toevoeging zijn om de actieve beveiliging te verbeteren. Eventuele incidenten worden eerder opgemerkt, doordat de beheerders een melding van de SIEM krijgen bij verdacht verkeer. Dat is eerder dan zonder SIEM, dan wordt het namelijk pas opgemerkt door verdachte gebeurtenissen.

Deze opdracht is gefaseerd aangepakt met gebruik van Kanban als methodiek. Het antwoord op de hoofdvraag bestaat uit een lijst met aanbevelingen voor Level Level teneinde het implementeren van een SIEM-oplossing die optimaal past bij de omgeving.

Voorwoord

Voor u ligt het afstudeerverslag over de implementatie van een Security Information & Event Management-systeem voor Level Level, ten behoeve van een betere beveiliging van de hosting opstelling. Dit document is samengesteld door een vierdejaarsstudent van de opleiding HBO-ICT Network & Systems Engineering.

De opdracht heb ik als uitdagend en leerzaam ervaren.

Graag wil ik de organisatie, Level Level, bedanken, waaronder mijn bedrijfsmentor Bernard Zijlstra, alle collega's en mijn stagebegeleider van de opleiding, Hans van der Burg.

Daarnaast bedank ik alle docenten HBO-ICT die mij les gaven. Ik heb erg veel geleerd tijdens de opleiding waar ik de rest van mijn leven nog veel aan heb.

Ik heb de organisatie en het team als leerzaam en zeer prettig ervaren. Ondanks de grote invloed van het coronavirus heb ik toch genoten van mijn periode als afstudeerder.

Veel leesplezier,

Wouter Honselaar

's-Gravenzande, 4 juni 2021

1. Inleiding

In dit verslag wordt het volledige afstudeertraject van een student Network & Systems Engineering aan De Haagse Hogeschool en afstuderend bij Level Level gerapporteerd.

De opdracht heeft als doel om een SIEM-oplossing te specificeren, ontwerpen en zo mogelijk realiseren als onderdeel van het nieuw op te zetten WordPress-hostingplatform van Level Level. Ik ga onderzoeken aan welke eisen de SIEM-oplossing moet voldoen en hoe deze geoptimaliseerd kan worden. Een SIEM-oplossing dient voor het opsporen van en reageren op bedreigingen en kwetsbaarheden.

Daarom is er, volgens een gekozen methodiek en fasering, geanalyseerd, gespecificeerd en ontworpen hoe een mogelijke SIEM-oplossing zodanig geconfigureerd kan worden dat het optimaal aansluit bij de omgeving van Level Level. Een globale beschrijving hiervan is te vinden in de samenvatting op pagina 5.

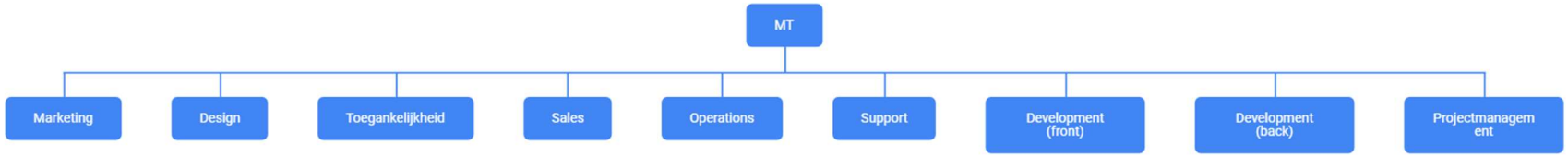
In hoofdstuk 2 bevindt zich de beschrijving van de organisatie en informatie over mijn opdracht. Daarin bevindt zich ook informatie over de aanpak van de opdracht. In hoofdstuk 3 bevindt zich de beschrijving van de oriëntatiefase. Daaropvolgend is de beschrijving van de analysefase, in hoofdstuk 4. Vervolgens is de ontwerpfase in hoofdstuk 5 beschreven. In hoofdstuk 6 is de implementatiefase beschreven. Hoofdstuk 3 t/m 6 beschrijven dus de doorlopen fasen.

Daarna komt een overzicht met de hoofdvraag en de daaruit afgeleide deelvragen met antwoorden. In hoofdstuk 8 staat de conclusie, in hoofdstuk 9 de aanbevelingen aan de organisatie en in hoofdstuk 10 mijn reflectie op de beroepstaken van de opdracht. Hoofdstuk 11 beschrijft de evaluatie van de aanpak en product, 12 de geraadpleegde literatuur, 13 de afkortingen, en ten slotte de bijlagen.
De bijlagen luiden: Routerconfiguraties in lokale testomgeving, Testrapport, Plan van Aanpak en Afstudeerplan.

2. Level Level

In dit hoofdstuk volgt de omschrijving van de organisatie waar de opdracht wordt uitgevoerd en wat de opdracht daaraan gaat bijdragen.

Level Level is een commercieel full-service WordPress-bureau. Ze maken maatwerk websites die werken met het contentmanagementsysteem WordPress. Een contentmanagementsysteem (CMS) is een softwarepakket om de inhoud van websites te beheren. Het bedrijf verzorgt zowel de hosting van de websites, het beheer van de WordPress-installaties als de manier waarop de content aangeboden wordt. Er werken ruim dertig mensen. Het kantoor is gevestigd in Rotterdam. Het gros van de klanten zijn Nederlandse organisaties, maar Level Level opereert wereldwijd. Voorbeelden van klanten zijn Cordaid en ABN AMRO. Hieronder staat het organigram:



Figuur 2.0a - organigram Level Level

Zoals zichtbaar in het organigram, is Level Level een lijnorganisatie met een managementteam (MT), en de afdelingen Marketing, Design, Toegankelijkheid, Sales, Hosting/Operations, Support, Front-end development, Back-end development en Projectmanagement. Ik studeer af op de afdeling Hosting/Operations. Deze afdeling houdt zich bezig met het veilig en beschikbaar houden van de infrastructuur voor de websites van klanten en voor Level Level zelf. Het managementteam bestaat uit drie mensen, waaronder mijn opdrachtgever en begeleider die ook werkzaam is op de afdeling Hosting/Operations.

Figuur 2.0b - organigram managementteam

De afdeling Hosting/Operations bestaat uit twee personen, exclusief mezelf. Hosting/Operations heeft regelmatig contact met de afdeling Support, voor problemen met betrekking tot infrastructuur. Hier volgt het organigram van het managementteam.

De BIV-factoren (**b**eschikbaarheid, **i**ntegriteit en **v**ertrouwelijkheid) spelen een belangrijke rol bij de veiligheid van de websites. Omdat de organisatie naast de software ook de hosting verzorgt, is cyber security een belangrijk aspect. Level Level heeft geen servers in eigendom of colocatie, maar ze maken gebruik van VM's bij diverse cloudhosting leveranciers. Servers in colocatie zijn servers in eigendom (of gehuurde servers) ondergebracht in een datacenter, waarbij het datacenter door meerdere organisaties gebruikt wordt. Omdat ze geen colocatie-servers maar VM's hebben is het niet verantwoordelijk voor de hardware. VPS staat voor Virtual Private Server. Het is een gevirtualiseerde server die particulieren en bedrijven kunnen huren. Meerdere klanten maken dan gebruik van dezelfde fysieke server. Hierbij is sprake van volledige isolatie tussen virtuele servers van klanten en mag elke klant een gedeelte van de beschikbare capaciteit gebruiken.
Beheer van de hardware wordt gedaan door de partij die de VPS'en verhuurt, beheer van software wordt gedaan door de hurende partij. Level Level heeft momenteel een nieuwe hostingarchitectuur in ontwikkeling. De organisatie heeft zich als doel gesteld dat deze nieuwe architectuur veel beter moet zijn dan de vorige generatie, vooral op het gebied van schaalbaarheid en actieve beveiliging.

2.1 Aanleiding

Level Level heeft nog geen SOC, waardoor een SIEM mogelijk meer geautomatiseerd moet kunnen werken dan in een organisatie met een SOC. SOC staat voor **s**ecurity **o**perations **c**enter en is een afdeling die alle technische cyber security-gerelateerde zaken constant monitort, en zo nodig ingrijpt bij incidenten zoals succesvolle aanvallen. Denk bijvoorbeeld aan het tijdelijk blokkeren van verkeer vanaf een bepaald IP-adres dat een hackpoging doet. Daarnaast fungeert het als een meldkamer voor medewerkers of buitenstaanders die security-gerelateerde zaken willen melden.

Het detecteren van beveiligingsincidenten speelt een belangrijke rol bij webhosting, omdat er schade toegebracht kan worden aan (klanten van) het bedrijf als een eventuele aanval te laat ontdekt wordt. Nu worden beveiligingsincidenten voornamelijk door één persoon opgemerkt en afgehandeld. Dit is een single-point-of-failure: als de betreffende persoon niet in staat is snel een incident af te handelen, kan dit bijvoorbeeld leiden tot websites die niet beschikbaar zijn of gevoelige data die lekt met alle gevolgen van dien.
De organisatie is een nieuwe hostingarchitectuur aan het ontwikkelen omdat er behoefte is aan meer mogelijkheden met betrekking tot schaalbaarheid. Hiermee blijven websites die binnen korte tijd veel meer bezocht worden snel doordat er meer servercapaciteit ingezet kan worden (opschalen), en omgekeerd (afschalen). Hiermee wordt de beschikbare servercapaciteit efficiënter ingezet. Daarnaast wordt er gebruik gemaakt van meerdere datacenters, wat de beschikbaarheid ten goede komt. De websites van klanten blijven werken als er een storing optreedt binnen een datacenter, doordat ze dan vanuit een ander datacenter gehost worden.

SIEM staat voor Security Information & Event Management en is de naam voor een systeem waarin verdachte activiteit real-time gesignaleerd wordt, en waarbij al dan niet geautomatiseerd actie kan worden ondernomen. Daarnaast is er één persoon die regelmatig scans en tests ten behoeve van de security uitvoert op het netwerk. Er wordt nu gebruikgemaakt van een in eigen beheer ontwikkelde hosting setup. Er wordt voornamelijk gewerkt met cloud hosters (DigitalOcean, AWS, Tilaa, etc.) voor het verzorgen van Virtual Private Servers.

2.2 Probleemstelling

De probleemstelling is dat de veiligheid van de huidige hostingopstelling niet meer conform de standaard van Level Level is. De wens is om verhoudingsgewijs meer proactief bezig te zijn dan reactief op het gebied van security. SIEM is een middel waarmee deze standaard waarschijnlijk wel behaald kan worden, namelijk door de BIV-factoren zodanig te bewaken dat ze beter zijn dan in de huidige situatie. Uitgangspunt is dat de veiligheid in brede zin beter gewaarborgd kan worden als er een SIEM wordt gebruikt. Een SIEM wordt vaak gebruikt in organisaties met een SOC.

Vrijwel alleen grote organisaties hebben een SOC. Kleinere organisaties met de behoefte voor een SOC kunnen gebruik maken van een ‘managed SOC’: een externe partij die de SOC-werkzaamheden uitvoert die anders in een eigen SOC uitgevoerd worden. Een SIEM draagt bij aan de beveiliging door alle verdachte activiteit op een overzichtelijke manier te presenteren aan de SOC-medewerkers.

2.3 Doelstelling

De doelstelling van de opdracht is het specificeren, ontwerpen en zo mogelijk realiseren van een SIEM als onderdeel van het nieuwe WordPress-platform van Level Level voor het opsporen van bedreigingen en kwetsbaarheden. Om de beschikbaarheid en integriteit van WordPress-websites van de klanten beter dan in de huidige situatie te garanderen, is het belangrijk om eventuele kwetsbaarheden en aanvalspogingen real-time in kaart te kunnen brengen, zodat hier, al dan niet geautomatiseerd, op geacteerd kan worden. Na afloop van de opdracht zal er een onderzoek opgeleverd worden dat een ontwerp en belangrijke metrics bevat die van toepassing zijn op de situatie van Level Level. Eventueel wordt een geoptimaliseerde Proof of Concept-opstelling gerealiseerd als eindproduct. Deze kan door de organisatie gebruikt worden om een uiteindelijke SIEM te implementeren. Dit kan een bestaand systeem zijn waarbij de motivatie van de keuze is beschreven in het eindverslag, of een geheel of gedeeltelijk nieuw systeem dat op maat ontwikkeld wordt.

3. Oriëntatiefase

Tijdens de oriëntatiefase heb ik gebruikgemaakt van deskresearch op het wereldwijde web met de centrale vraag ‘Welke mogelijkheden bieden verschillende SIEM-systemen?’. Daarnaast stel ik een Plan van Aanpak en Programma van Eisen op, op basis van de door Level Level gebruikte hostingomgeving met bijbehorende wensen. Gedurende uitvoering van de fasen worden deelvragen opgesteld.

3.1 Aanpak

Zoals in bovenstaande alinea staat, wordt deze fase uitgevoerd middels deskresearch op het wereldwijde web en binnen het bedrijf om het Programma van Eisen op te stellen. Daarnaast voer ik een interview met de opdrachtgever om de wensenlijst op te stellen. Aan de hand van de opdracht en het PvE is een Plan van Aanpak opgesteld. De randvoorwaarden zijn dat het product kosteloos en open source moet zijn. Het eindproduct is niet een geïmplementeerde SIEM-oplossing, maar een lijst met aanbevelingen hoe de gekozen SIEM geoptimaliseerd kan worden voor gebruik in de omgeving van Level Level. Gedurende het onderzoek wordt gebruik gemaakt van open source SIEM’s in verband met de aanpasbaarheid en eenvoudige en kosteloze beschikbaarheid hiervan.

3.2 Concrete werkzaamheden

In de oriëntatiefase ben ik begonnen met het inhoudelijk invulling van mijn afstudeeropdracht. Tevens heb ik gewerkt aan het Plan van Aanpak en het afstudeerverslag heeft een indeling gekregen. In de eerste week heb ik mij verdiept in de werkwijze en systemen die Level Level gebruikt, en ingelezen over SIEM-oplossingen en wat deze systemen als toegevoegde waarde kunnen hebben voor cloud-gebaseerde hostingomgevingen. De antwoorden hierop staan onder het kopje ‘Deelresultaten’. Daarnaast heb ik ook geëxperimenteerd met AlienVault OSSIM (OSSIM: The Open Source SIEM | AlienVault, z.d.), enerzijds om de mogelijkheden van een bekende SIEM-oplossing te verkennen, anderzijds om na te gaan of de tot nu toe opgestelde wensen reëel zijn zonder meteen de route van maatwerksoftware te kiezen. De set wensen is zowel belangrijke informatie voor het verder opstellen van het Plan van Aanpak (de opdracht wordt ernstig bemoeilijkt indien er onrealistische/onmogelijke wensen zijn) als voor de analysefase.

3.3 Deelresultaten

Het voornaamste deelresultaat van deze fase is het Plan van Aanpak, alsmede het Programma van Eisen wat hier onderdeel van is. Het Plan van Aanpak is te vinden in bijlage C. Het Plan van Aanpak beschrijft de aanleiding, doelstelling, probleemstelling, wensen van de opdracht, de aanpak en fasering. Dat staat ook geschreven in de vorige twee hoofdstukken van dit verslag.

De wensen van de opdrachtgever luiden:

3.3.1 Functionele wensen

1. Informatie over incidenten is te raadplegen via een centrale GUI. De belangrijkste metrics zijn zichtbaar in de GUI. *Welke metrics dit zijn wordt verder gespecificeerd in de analysefase.*
2. Bij elke aanval wordt een escalatiematrix gebruikt die de beheerafdeling op de hoogte stelt. Hiermee worden beheerders buiten het dashboard om op de hoogte gesteld van een eventuele aanval. *De escalatiematrix werkt als volgt: zowel gedurende kantooruren als daarbuiten wordt er een bericht geplaatst in een speciaal Slack-kanaal, waarmee de beheerders een melding krijgen op hun computer.*
3. Indien er sprake is van een duidelijke hackpoging wordt deze gelijk geblokkeerd. *Wat onder 'duidelijke hackpoging' valt, wordt in de analysefase verder gespecificeerd, en kan aan de hand van bevindingen in verdere fasen verder gespecificeerd worden.*
4. Data kan opgehaald worden bij specifieke clients met of zonder agents, en ook direct op de netwerkinterface van de SIEM. *Om welke data dit gaat, wordt in de analysefase uitgezocht. Gedurende de volgende fasen wordt dit verder gespecificeerd.*
5. Er kunnen automatische routine-tests uitgevoerd worden in de vorm van vulnerability scans. *Dit is handig omdat deze momenteel handmatig uitgevoerd worden, wat tijd kost, terwijl ze goed te automatiseren zijn. Routine tests zijn exact gedefinieerd, gestandaardiseerd en makkelijk in code uit te voeren. Deze worden met dezelfde tijdsintervallen uitgevoerd.*

3.3.2 Niet-functionele wensen

6. Logs kunnen niet aangepast worden door gebruikers.
7. Voldoet aan geldende wet- en regelgeving.
8. De performance is zoals verwacht mag worden op basis van de handelingen die verricht worden. *Hiermee wordt bedoeld dat de load van een server gemiddeld gezien niet boven 50% van maximale capaciteit uitkomt.*
9. Heeft een uptime van 99,8%.
   99,8% uptime is vastgelegd in de Service Level Agreements met klanten voor hun websites. De SIEM moet dezelfde uptime hebben.
10. Vertraagt het netwerk of de diensten die aan klanten geleverd worden niet. *Dit is meetbaar door te controleren of de CPU-belasting en netwerkgebruik (bandbreedtegebruik) niet significant oplopen na het inschakelen van de SIEM-agent.*
11. Gebruiksvriendelijk genoeg voor technische beheerders. *Hiermee wordt verstaan dat er gebruik gemaakt wordt van een GUI (graphical user interface). De technische beheerders zijn gewend om met CLI's (command-line interfaces) te werken. Algemeen kan gesteld worden dat een gebruiker die een CLI kan bedienen ook een GUI kan bedienen, omdat de moeilijkheidsgraad van een CLI hoger is dan van een GUI.*
12. De scope bestaat uit alle servers die direct verbinding met internet hebben, waaronder servers die persoonsgegevens verwerken.

Ik heb deze wensen vergeleken met AlienVault OSSIM:

Wens	**Voldoet?**
1	Ja, OSSIM werkt met een centrale GUI.
2	Dit is weliswaar functioneel, maar puur organisatorisch. OSSIM kan hier niet op zichzelf aan voldoen. OSSIM is wel in staat actie te ondernemen (beheerders waarschuwen) na een hackpoging. Het maakt het dus wel mogelijk.
3	OSSIM kan niet zelfstandig hackpogingen blokkeren.
4	OSSIM werkt met agents op de clients die data ophalen en naar OSSIM sturen. Daarnaast kan OSSIM ook zonder agents op clients werken door middel van SSH. De agent is namelijk in staat via SSH commando's uit te voeren en deze uitvoer vervolgens te analyseren.
5	OSSIM kan op basis van vulnerability scans die gepland en geautomatiseerd uitgevoerd worden eventueel gevonden kwetsbaarheden en onnodige open poorten als security event behandelen.
6	Zolang er mensen zijn die root-toegang hebben of fysieke toegang tot de opslagschijf/schijven van de servers is aanpassen van logs door gebruikers niet helemaal te voorkomen. Mitigatie hiervan kan door security events zo snel mogelijk te verwerken: als er een melding is wordt een beheerder gewaarschuwd. Deze is dan al gewaarschuwd op het moment dat de betreffende log entry door een kwaadwillende verwijderd zou worden.
7	Zolang er gehandeld wordt volgens geldende wet- en regelgeving voldoet OSSIM hieraan. OSSIM zelf stuurt geen data met betrekking tot clients of security events naar servers toe.
8	Zolang de CPU-load niet gemiddeld boven 50% uitkomt na een kwartier nadat de SIEM ingeschakeld is, wordt er aan deze eis voldaan.
9	Zolang dezelfde zorg gedragen wordt voor

	de OSSIM-server als voor de klantservers die een uptime van 99,8% hebben kan er vanuit gegaan worden dat ook deze server dezelfde uptime heeft.
10	Zodra de netwerkactiviteit (bytes per seconde) niet met 20% verhoogd is als de SIEM ingeschakeld wordt, is aan deze eis voldaan.
11	Als er gebruik gemaakt wordt van een GUI is hieraan voldaan.
12	Dit is een randvoorwaarde en daarom niet testbaar.

Tabel 3.3.2a - wensen vergeleken met mogelijkheden AlienVault OSSIM

3.4 Aanpak opdracht

De opdracht zal middels een methodiek worden aangepakt, namelijk Kanban met enkele aspecten uit de watervalmethode. Daarom pak ik de opdracht gefaseerd en per fase in iteraties aan. Ik heb gekozen voor Kanban in combinatie met een fasering en iteraties. Bij netwerkprojecten is de watervalmethode gebruikelijk, omdat de eisen doorgaans niet veranderen tijdens uitvoering van het project.

Deze opdracht heeft niet snel veranderende wensen, maar de eisen kunnen wel ieder moment aangepast worden omdat zowel de opdrachtgever als ik geen eerdere ervaring heeft met SIEM-systemen. De watervalmethode kan niet terug naar een vorige stap. Indien er toch een veranderende eis is, moet het project vanaf de eerste fase opnieuw worden uitgevoerd. Omdat dit een groot risico met zich meebrengt (de opdracht moet immers binnen een bepaalde tijd af zijn) heb ik niet voor een watervalmethode gekozen. Ik heb voor Kanban gekozen omdat het flexibeler is dan Scrum. Scrum werkt met iteraties van vaste lengtes waarbij er aan het begin van elke iteratie een vaste set aan taken wordt toegewezen aan de programmeur(s). Omdat de eisen en daarmee de taken nog niet vastgesteld waren, zou het zomaar kunnen gebeuren dat tijdens iteraties taken toegevoegd of weggehaald moesten worden. Het zou dan niet efficiënt zijn om Scrum te gebruiken.

Ik heb daarom voor Kanban gekozen, want dan heb ik meer vrijheid ten opzichte van Scrum en de watervalmethode, maar wel in combinatie met een fasering en iteraties. Kanban is zonder fasering te gebruiken, maar enige structuur in de aanpak van dit project middels fasering is gewenst omdat ik eerst uit wil zoeken wat precies de uitdaging is, aan welke eisen/wensen een oplossing moet voldoen om daar vervolgens taken mee te maken.

Daarnaast heb ik gekozen voor iteraties, waarbij ik aan het begin van elke iteratie bepaal welke taken ik in deze iteratie wil doen. Dit is gelimiteerd door de WIP-limiet van Kanban. Elke iteratie duurt een week. Mocht er tijd over zijn of te weinig tijd zijn of een andere reden waardoor ik meer/minder taken doe gedurende een iteratie, dan kan ik een taak die nog moet gebeuren toevoegen of een taak verplaatsen naar de volgende iteratie.

Het grootste voordeel van iteraties is dat inzichtelijk is of ik voor- of achter op schema loop: immers, als ik binnen de ontwerpfase bijvoorbeeld nog bezig ben met veel taken uit de analysefase, loop ik op dat gebied achter. Door goed te anticiperen op de rest van de huidige iteratie en komende iteraties kan ik bepalen hoe ik een eventuele achterstand kan oplossen.

Het Plan van Aanpak bevindt zich in bijlage C.

De opdracht zal gefaseerd worden aangepakt. Per fase wordt de beste werkwijze bepaald. Dit zijn de fasen: de oriëntatiefase, analysefase, ontwerpfase en implementatiefase. De oriëntatiefase heeft als doel om inzicht te krijgen in de mogelijkheden en eisen/wensen van het bedrijf. De analysefase heeft als doel om inzicht te krijgen in de exacte wensen en behoeften in de nieuwe hostingarchitectuur, en waaruit deze architectuur (technisch) bestaat. De ontwerpfase heeft als doel een ontwerp te maken, met alle gedetailleerde functionele en niet-functionele eisen, en hoe dit geïmplementeerd zal gaan worden.
De implementatiefase heeft als doel het systeem in de vorm van een simulatie te implementeren. Tijdens de analyse-, ontwerp- en implementatiefase zullen Proof of Concept-opstellingen gemaakt worden. Deze zullen gebruikt gaan worden voor verder onderzoek om er achter te komen welke systemen het best geschikt zijn voor Level Level, in hoeverre hier maatwerk- of bestaande systemen voor nodig zijn, welke metrics van belang zijn, en hoe er het best omgegaan kan worden met false positives. Daarnaast wordt deze ook gebruikt om te achterhalen welke data precies door de SIEM verwerkt zal moeten worden, en hoe er het beste omgegaan wordt met false positives.

4. Analysefase

De eisen waaraan de SIEM-oplossing moet voldoen zullen verder gespecificeerd worden in deze fase. Daarnaast wordt onderzocht welke SIEM-oplossing het beste geschikt is als proof of concept voor verder onderzoek. Hierbij wordt (met marktonderzoek) ook gekeken of een bestaande SIEM-oplossing (zoals AlienVault OSSIM) het beste binnen de organisatie past, of dat een maatwerkoplossing beter is. Naast deskresearch wordt ook gebruikgemaakt van experimenteel onderzoek. Experimenteel onderzoek heeft het doel om de Proof of Concept-opstelling te testen en zo nodig bij te stellen zodat deze geschikt wordt gemaakt voor een productieomgeving. Om dit allemaal te kunnen doen, worden eerst de eisen die in de oriëntatiefase opgesteld zijn verder gespecificeerd zodat ze SMART zijn en gebruikt kunnen worden in de zoektocht naar geschikte SIEM-oplossingen, en in de overweging voor het al dan niet gebruiken van maatwerk.

4.1 Aanpak

Er is, naar aanleiding van de deelresultaten van de oriëntatiefase (‘Welke mogelijkheden bieden verschillende SIEM-systemen?’), een lijst opgesteld met mogelijk te gebruiken bestaande SIEM-oplossingen. Dit is gedaan omdat het verder onderzoeken van de mogelijkheden van SIEM-systemen per systeem kunnen verschillen, en het onderzoeken van de mogelijkheden van één systeem nietszeggend is over de mogelijkheden van andere SIEM-systemen. Deze lijst staat in de tabel op de volgende pagina.

SIEM	Kenmerken t.o.v. eisen
AlienVault OSSIM	Gratis en open source, source niet te compileren want instructies ontbreken
Elastic SIEM / ELK Stack	Gratis, Elastic wordt al gebruikt
Wazuh	Gratis en open source, laag bovenop Elastic
Security Onion	Gratis en open source, aparte distributie
Splunk	Gratis met veel beperkingen, closed source
SolarWinds Security Event Manager	Vrijwel dezelfde features als Wazuh, closed source en kost geld
OSSEC (zelfstandig)	Gratis en open source, op zichzelf krachtig maar beperkt met betrekking tot user interfaces
IBM QRadar	Closed source en kost geld
ArcSight	Closed source en kost geld
McAfee ESM	Closed source en kost geld
Suricata (zelfstandig)	Tegenhanger van OSSEC, IDS[1], gratis en open source, krachtig maar beperkte interface, vooral NIDS in plaats van HIDS
Apache Metron	Bestaat niet meer sinds december 2020
Mozilla MozDef	Gratis en open source
Prelude OSS	Gratis en open source, geeft een lage performance bij de niet-betaalde versie en kan niet zelfstandig reageren op urgente beveiligingsgebeurtenissen
Sagan	Gratis en open source IDS, kan werken met Snort
Siemonster	Gratis (beperkt) en open source, heeft zeer veel integraties en is als het ware laag boven Wazuh en andere SIEM's, werkt ook met Elasticsearch, gratis versie ondersteund generatie van twee rapporten

Tabel 4.1a - te overwegen SIEM-systemen

[1] IDS = intrusion detection system, systeem dat vermoedelijke hackpogingen detecteert, HIDS is IDS op hostniveau (applicaties die op een host draaien), NIDS is IDS op netwerkniveau (zoals verdacht TCP verkeer)

Deze keuzes zijn te motiveren omdat ze veelgebruikte open source SIEM-oplossingen zijn (Berman, 2019). Voordat deze lijst opgesteld is, is een architectuurontwerp gemaakt van de huidige architectuur. In de ontwerpfase is deze gevisualiseerd. Aan de hand van dit ontwerp is het PvE uitgebreidt en verder gespecificeerd. De in de analysefase opgestelde eisen zijn specifieker dan in de oriëntatiefase, omdat er is onderzocht welke data precies door de SIEM moet worden kunnen gedetecteerd en hoe moet worden gereageerd op incidenten. Hiervoor is een deelvraag opgesteld: ‘Aan welke precieze eisen moet de SIEM voldoen?’

In de analysefase zal, op basis van het PvE en bovenstaande lijst, een afweging gemaakt worden van geschikte en ongeschikte SIEM-oplossingen, en in welke mate sprake zal zijn van een maatwerkoplossing.

4.2 Concrete werkzaamheden

Er is gewerkt aan het onderzoeken van antwoorden op de in deze fase opgestelde deelvragen. Dit is gedaan middels experimenteel onderzoek aan één of meerdere PoC’s. Deze zijn allemaal getest binnen dezelfde testomgeving, bestaande uit een netwerk van een client (Ubuntu 20.04 LTS Server, verse installatie) en de SIEM. De eerste SIEM die getest is, is AlienVault OSSIM. OSSIM is een open source SIEM. De tweede SIEM die getest is, is ElasticSearch SIEM. Daarnaast zijn eisen opgesteld aan de hand van de wensen. Ten slotte is Wazuh ook getest. Ik heb gekozen om deze drie SIEM’s te testen omdat ze allemaal open source zijn.

4.3 Deelvragen

Gedurende de analysefase zijn de volgende deelvragen opgesteld:

1. Hoe ziet de architectuur van de nieuwe hosting setup van Level Level eruit?
2. Welke SIEM-oplossingen worden in overweging gebracht?
3. Aan welke functionele eisen moet de SIEM voldoen en waarom?

4.4 Deelresultaten

Middels een gesprek met de opdrachtgever ben ik te weten gekomen hoe de architectuur van de nieuwe hosting setup eruit gaat zien.

4.4.1 Huidige situatie

Elke website van elke klant (die door Level Level gehost wordt) werkt met de webserver nginx, PHP 7 en een MariaDB database server. Sites van klanten kunnen bereikt worden middels een reverse proxyserver. Deze wordt voor veel klanten gebruikt. Daarnaast gebruikt een aantal klanten een load balancer. De webserver draait binnen een LXD container op een Ubuntu 20.04 LTS VM. Veel klanten hebben eigen VM’s voor de webserver(s), reverse proxy en database. Als reverse proxy wordt nginx gebruikt. Op een tweede deel van het platform worden de VM’s gedeeld met andere klanten, al vindt er wel isolatie plaats middels de LXD-containers (elke klant heeft een eigen container).

Dit hangt af van de grootte van de klant en de hoeveelheid verkeer naar de site. Dit platform is niet redundant of schaalbaar uitgevoerd, en staat binnen één datacenter locatie. Zie figuur 4.4a.

Daarnaast is aan de hand van de wensen van de opdrachtgever (zie 3.4) een lijst met eisen opgesteld:

4.4.2 Functionele eisen

1. Er is een grafische gebruikersinterface waarin real-time informatie over security events te zien is. Het moet in één oogopslag duidelijk zijn of er een ernstige aanval gaande is of was. Middels diagrammen wordt getoond wat voor security events er hebben plaatsgevonden en wanneer, zodat abnormale grote hoeveelheden van bepaalde soorten events snel gezien kunnen worden. Details over security events moeten geraadpleegd kunnen worden, waarbij metadata (minimaal source/dest IP en poort, beschrijving, timestamp, soort gebeurtenis, classificatieniveau) zichtbaar moet zijn per gebeurtenis.
2. Tussen 8:00 uur en 17:00 uur op werkdagen worden vermoedelijke aanvallen geautomatiseerd via het daarvoor bestemde Slack-kanaal gemeld aan de beheerders.
3. Indien er sprake is van een aanval, dient deze zo snel mogelijk geblokkeerd te worden door de SIEM. Het betreffende IPv4-adres wordt voor één uur alle toegang tot de betreffende aangevallen server ontzegd door een blokkade.
4. Alle data die nodig is voor het succesvol voldoen aan eis 1 (het tonen van alle relevante security events in een GUI) wordt middels agents opgehaald bij servers. De communicatie tussen de agent en de SIEM is versleuteld.
5. Er worden automatische vulnerability scans uitgevoerd. Telkens als er een verdachte poort of kwetsbaarheid is gedetecteerd wordt dit behandeld als security event en als zodanig geclassificeerd dat het gemeld wordt aan beheerders.

4.4.3 Niet-functionele eisen

6. Logs kunnen niet aangepast worden door gebruikers.
7. Voldoet aan geldende wet- en regelgeving, zoals de Algemene Verordening Gegevensbescherming, de aanbevelingen in de BIO en de ISO 27001-norm.
8. De performance is zoals verwacht mag worden op basis van de handelingen die verricht worden.
   Hiermee wordt bedoeld dat de load van een server gemiddeld gezien niet boven 50% van maximale capaciteit uitkomt.
9. Heeft een uptime van 99,8%.
   99,8% uptime is vastgelegd in de Service Level Agreements met klanten voor hun websites. De SIEM moet dezelfde uptime hebben en altijd online zijn als de klantenwebsites ook online zijn.
10. Vertraagt het netwerk of de diensten die aan klanten geleverd worden niet.
    Dit is meetbaar door te controleren of de CPU-belasting en netwerkgebruik (bandbreedtegebruik) niet significant oplopen na het inschakelen van de Wazuh-agent.

11. Gebruiksvriendelijk genoeg voor technische beheerders.
    Hiermee wordt verstaan dat er gebruik gemaakt wordt van een GUI (graphical user interface). De technische beheerders zijn gewend om met CLI's (command-line interfaces) te werken. Algemeen kan gesteld worden dat een gebruiker die een CLI kan bedienen ook een GUI kan bedienen, omdat de moeilijkheidsgraad van een CLI hoger is dan van een GUI.
12. De scope bestaat uit alle servers die verbinding met internet hebben, waaronder servers die persoonsgegevens verwerken.

4.4.4 Toekomstige situatie

De gewenste situatie bestaat uit een private cloud omgeving met pfSense firewalls, loadbalancers en LXD containers per klant op een reeks VM's. Ten behoeve van een hogere beschikbaarheid dan de huidige situatie wordt gebruik gemaakt van meerdere datacenterlocaties zodat de klantenwebsites online blijven als de opstelling in één datacenter om wat voor reden dan ook niet meer beschikbaar is. De private cloud moet in Nederland staan en lid zijn van NaWas (NBIP, z.d.). Zie figuur 4.4b.

NaWas is een samenwerkingsverband van Nederlandse internetproviders en dient voor het filteren van ongewenst verkeer, zodat DDoS-verkeer wordt geneutraliseerd. Al het verkeer wordt door de 'NaWas-straat' geleid indien er een aanval gaande is. Indien er geen aanval gaande is, wordt het verkeer niet door de 'NaWas-straat' geleid. Nadat het verkeer door de firewall is gegaan, wordt het via de loadbalancers naar de LXD containers van de klant gestuurd. De LXD containers draaien op VM's met het Ubuntu 20.04 LTS besturingssysteem.

Het platform wordt gemanaged met OpenNebula (OpenNebula – Open Source Cloud & Edge Computing Platform, z.d.). OpenNebula is een applicatie die dient voor het beheer van een cloud infrastructuur. Vanuit de interface van OpenNebula kunnen VM's en containers beheerd worden. Binnen de LXD containers draait de nginx webserver met PHP 7. Voor kort lopende services kunnen Docker images worden gebruikt onder de paraplu van OpenNebula. En ook volledige op KVM[2] gebaseerde VM's kunnen worden ingezet.
Deze opstelling zal in een tweede fase uitgebreidt worden over meerdere datacentra locaties, zodat de klantenwebsites blijven werken tijdens een storing of onderhoud in één van de datacentra.
Middels OpenNebula is het migreren en load balancen van containers tussen verschillende servers te managen, waardoor bij een geplande en ongeplande verstoring geen onderbrekingen meer optreden. Met de huidige opstelling kan dit niet.

4.4.5 Huidige versus gewenste situatie

Het doel van de gewenste nieuwe situatie is om de BIV-factoren beter te waarborgen. Dit kan bereikt worden door gebruik van een SIEM en schaling van containers. Logging van gebeurtenissen zoals PHP foutmeldingen gebeurt al in de huidige situatie, maar logging van beveiligingsgebeurtenissen niet.

[2] KVM (niet te verwarren met KVM-switch) is een open source softwarepakket voor virtualisatie voor Linux, vergelijkbaar met VMWare en Oracle VirtualBox.

Naast gebruik van de SIEM zijn betere load balancing en het gebruik van een tweede datacenter ook belangrijke verbeteringen ten opzichte van de huidige situatie.

4.4.6 Jumphost

In zowel de huidige als gewenste nieuwe opstelling wordt gebruik gemaakt van een jumphost. Een jumphost is bedoeld voor technische beheerders (middels SSH) om verbinding mee te maken, waarna die clients via de jumphost kunnen verbinden naar de server waarmee ze willen verbinden. Om een verbinding op te zetten met een server van een klantenwebsite, moet er dus eerst verbinding worden gemaakt met de jumphost en daarvandaan met de server van de klant.

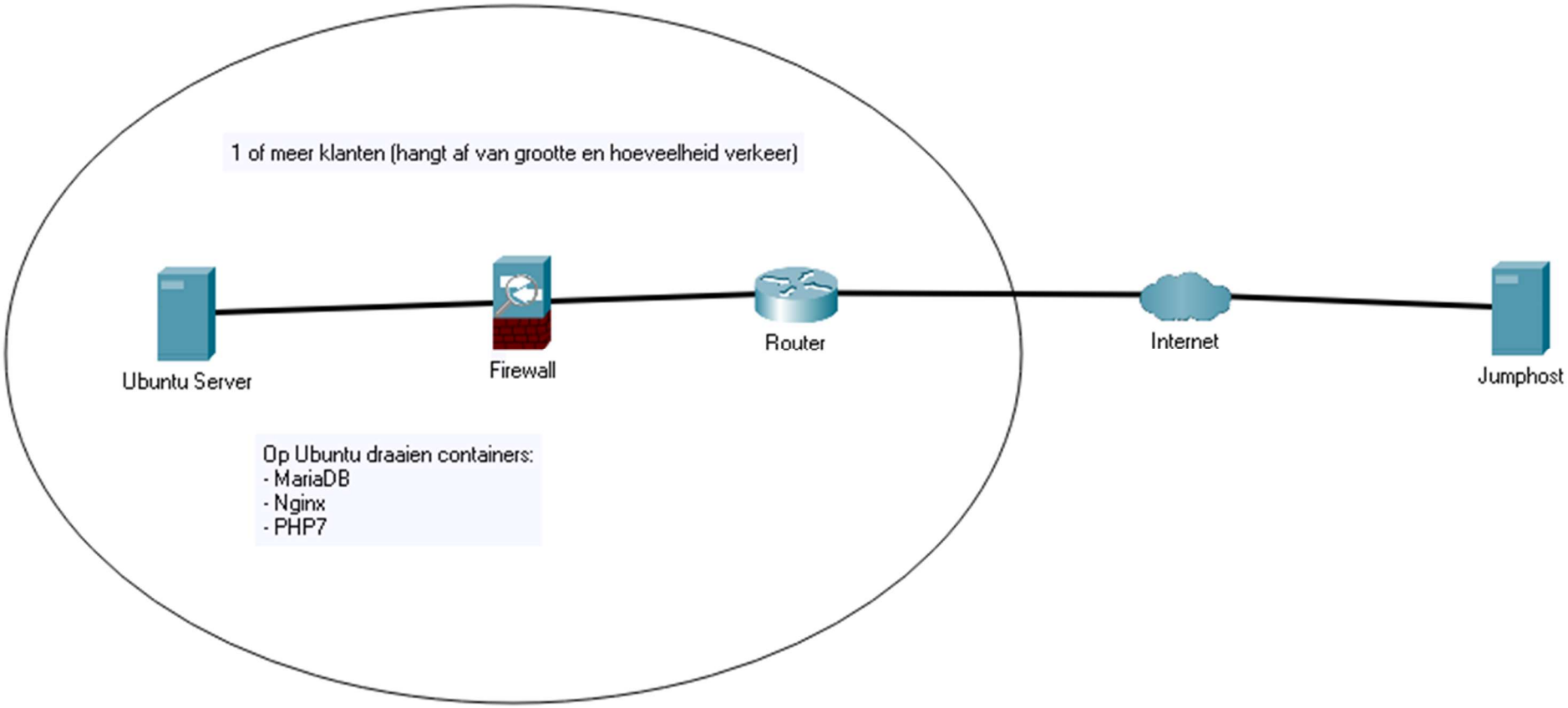
4.4.7 Monitoring

Aan de nieuwe setup zal naast traditionele monitoring ook een SIEM worden toegevoegd. Er wordt dan gebruikgemaakt van één centrale SIEM voor monitoring van activiteiten op servers van alle klanten. Op alle servers wordt een OSSEC agent geïnstalleerd die data verzameld op de clients. Het is belangrijk dat deze SIEM, naast het melden van security events aan beheerders, ook in staat is om zelfstandig tot actie over te gaan. Denk hierbij aan het blokkeren van verkeer van een bepaald IP-adres indien er binnen korte tijd meerdere aanvalspogingen kwamen van dit adres.

4.4.8 SIEM

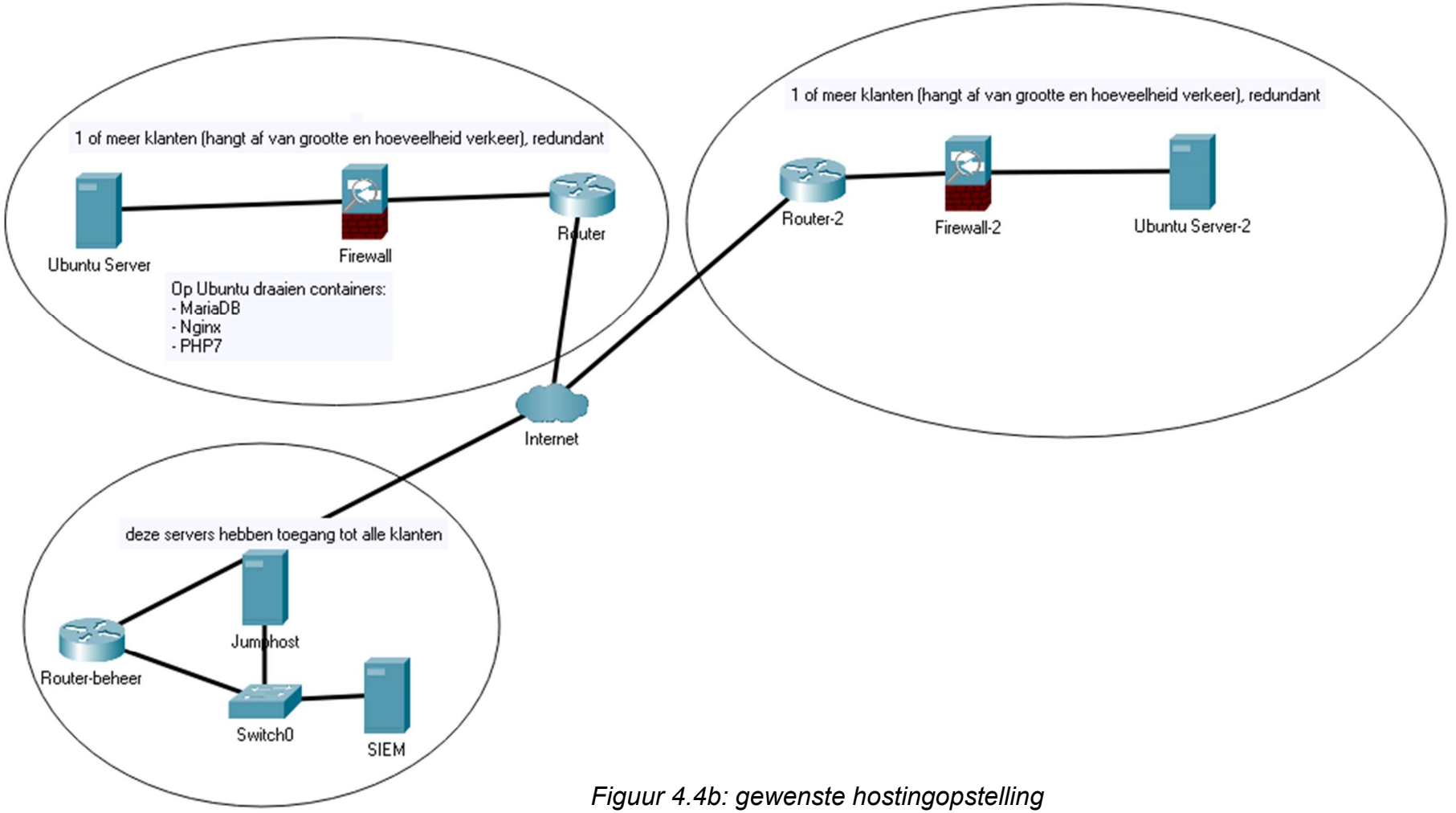
De SIEM’s die in overweging genomen zijn alle open source oplossingen die staan in bovenstaande tabel 4.1a van de analysefase. Alle te overwegen systemen zijn open source systemen zonder kostprijs. De organisatie maakt al gebruik van Elasticsearch voor onder andere de zoekfunctie op websites van klanten. Zowel Elasticsearch SIEM als Wazuh werken met Elasticsearch. Gezien de ervaring met Elasticsearch binnen de organisatie is dit een voordeel. Uiteindelijk is in samenspraak met het bedrijf gekozen voor Wazuh, omdat Wazuh meer mogelijkheden biedt met betrekking tot het reageren op incidenten dan Elasticsearch SIEM en overige SIEM’s. Hiervoor is gekozen in overleg met de opdrachtgever, omdat er al gebruik gemaakt wordt van Elasticsearch en daarom al enige ervaring met de bijbehorende interface is. Daarnaast is het een open source oplossing waarmee het voldoet aan de betreffende randvoorwaarde.

Vanwege de featureset van Wazuh en modulariteit van Elasticsearch is gekozen voor gebruik van een SIEM gebaseerd op Elasticsearch, bij voorkeur een bestaand systeem in verband met ondersteuning.De huidige architectuur van de hostingopstelling ziet er als volgt uit:



Figuur 4.4a: huidige hostingopstelling

De nieuwe architectuur van de hostingopstelling ziet er als volgt uit:



Figuur 4.4b: gewenste hostingopstelling

Middels GNS3, een netwerksimulator, is de opstelling ontworpen. Ik heb ernaar gestreefd deze zo realistisch mogelijk te maken met de kennis die ik bezit over computernetwerken. Zo gebruik ik Cisco-routers in dit netwerk, omdat ik hiermee al enige ervaring heb.

Het netwerk bestaat uit vier providers: ISP1 is verbonden met het datacenter waarin zich een webserver van Level Level bevindt. ISP2 is verbonden met het beheerdersgedeelte van Level Level, waarin de jumphost en SIEM zich bevinden. Hiervoor is gekozen omdat er in de huidige en nieuwe situatie gebruik gemaakt wordt van een jump (bastion) host. Alleen die host heeft SSH-toegang tot alle hosting servers van Level Level-klanten. Dit draagt bij aan de veiligheid want er is maar één host die toegang heeft tot de servers, in plaats van alle computers van medewerkers. De jumphost kan bereikt worden door bepaalde gewhiteliste IPv4-adressen, waarbij elke gebruiker eigen authenticatiegegevens heeft. ISP3 is verbonden met het kantoor van Level Level. Alleen vanaf dat kantoor kan middels SSH verbinding worden gemaakt met de jumphost, en dan via de jumphost met de webserver. ISP4 is verbonden met het lokale netwerk van een bezoeker. Dit netwerk (van de bezoeker) heeft geen speciale toegangsrechten tot de jumphost of webserver, waardoor deze alleen middels HTTP(S) verbinding kan maken met de webserver. In de simulatie wordt dit netwerk gebruikt om hackpogingen te doen op de webserver.

Elke ISP wordt gerepresenteerd door één router. Middels BGP worden routes tussen de ISP's gedeeld. Elke ISP is namelijk met elkaar verbonden: ze zijn allemaal direct verbonden met twee andere ISP's. Daarnaast is ISP1 ook aangesloten op AMS-IX en ISP4 op NL-IX. Hiervoor gebruik ik de NAT-verbindingsfunctionaliteit van GNS3. AMS-IX en NL-IX zijn grote internetknooppunten in Nederland. In deze simulatie zijn ze uiteraard fictief en aangesloten op mijn netwerkverbinding.

Dit brengt daadwerkelijke internetverbinding binnen de simulatie. Naast deze vier ISP-routers worden er ook routers gebruikt tussen de ISP's en hosts, net zoals in echte netwerken. Deze werken middels overloaded NAT (PAT). BGP is een bekend protocol om IP-routes tussen Autonomous Systems door te geven. Een autonomous system (AS) is een combinatie van netwerken die een bepaald doel dienen. Denk hierbij aan het netwerk van een ISP. Elke ISP heeft minimaal één autonomous system. Het doorgeven van routes tussen routers binnen AS'en, en tussen AS'en onderling, gebeurt middels BGP. Het handmatig invoeren van routes is ook mogelijk, alleen is dit problematisch omdat er honderdduizenden IP-routes routes gebruikt worden op internet (APNIC, 2021). Elke route moet bekend zijn bij elk AS om alle adressen voor elke internetgebruiker ter wereld bereikbaar te houden. Dit staat overigens los van het doel van de opdracht, dus hier wordt niet verder op ingegaan.

Ik heb er bewust voor gekozen om een zo realistisch mogelijke simulatie te gebruiken, zodat ik de werking van de SIEM kan testen in reële scenario's en ik een afgeschermde omgeving heb waarbinnen ik aanvallen kan uitvoeren op de webserver om er achter te komen welke data in de SIEM wordt verzameld en hoe deze bijgesteld kan worden.

5. Ontwerpfase

Er zal onderzoek worden gedaan naar de beste manier om een SIEM-oplossing op te nemen binnen de nieuwe hostingarchitectuur van Level Level. Daarnaast wordt onderzocht wat op het gebied van de BIV-factoren verandert is ten opzichte van de huidige hostingarchitectuur. Middels motiveringen wordt uitgelegd waarom bepaalde keuzes gemaakt zijn. Middels een reële PoC-opstelling wordt achterhaald welke data verzameld zal moeten worden door de SIEM. Door de PoC-opstelling steeds bij te stellen worden antwoorden op de deelvragen verkregen.

5.1 Aanpak

Er is, naar aanleiding van de deelresultaten van de analysefase, een virtuele opstelling gemaakt van de huidige en gewenste hosting setup. Dit is in het vorige hoofdstuk beschreven.

Deze opstelling is gebaseerd op de nieuwe hostingarchitectuur. Er is gekozen om met een zo realistisch mogelijke simulatie te werken, zodat de verkregen resultaten zo realistisch mogelijk zijn. Daarnaast kan in een simulatie zonder schade aan te brengen aan netwerken buiten de simulatie uitgebreidt getest worden of en hoe de SIEM-oplossing reageert op aanvallen. In de ontwerpfase is besloten dat er in eerste instantie zoveel mogelijk gebruik gemaakt wordt van de bestaande SIEM-oplossing Wazuh. In de ontwerpfase wordt in eerste instantie een simulatie gemaakt ten behoeve van het hebben van een overzichtelijke, kloppende situatieschets van de nieuwe netwerktopologie, en het werken met een werkende simulatie.

5.2 Concrete werkzaamheden

In eerste instantie is er gekozen voor Cisco Packet Tracer als programma om een simulatie mee te maken, aangezien het een veelgebruikt programma is waar ik veel ervaring mee heb vanuit de opleiding. Op een gegeven moment heb ik er voor gekozen om over te stappen op Graphical Network Simulator 3. Ik liep namelijk tegen de beperkingen van Packet Tracer aan; het was niet mogelijk om een VM, container of een fysiek netwerk aan de topologie toe te voegen. Voor een zo realistisch mogelijke test, waarbij er gebruik gemaakt wordt van verschillende hosts (waaronder een jumphost), de SIEM zelf, een webserver, en een daadwerkelijke internetverbinding, was het niet meer mogelijk om dit in Packet Tracer te doen. Daarom is er voor gekozen om over te stappen op GNS3. Naast GNS3 had ik ook kunnen kiezen voor andere software, zoals eve-ng. Vanwege de limitaties van eve-ng heb ik gekozen voor GNS3. De kosteloos te gebruiken versie van eve-ng biedt namelijk geen ondersteuning voor het verbinden van de te simuleren netwerkapparatuur aan een echt netwerk. Dit bemoeilijkt het installeren van software op virtuele machines binnen de virtuele netwerkomgeving.

Daarnaast heb ik de simulatie, zoals beschreven in de analysefase, tot uitvoering gebracht. Dit heeft veel tijd gekost, want alle hosts en routers moesten geconfigureerd worden. De precieze configuratie van de gesimuleerde netwerkapparatuur staat in bijlage A.

Ik heb ook een PoC-opstelling opgezet in DigitalOcean. Deze dient voor het verder uitvoeren van het onderzoek. Deze opstelling bestaat uit Wazuh op een Ubuntu 20.04 LTS VM. Daarnaast wordt er een agent geïnstalleerd op een webserver met WordPress, omdat de SIEM uiteindelijk de OSSEC-agent zal gaan gebruiken die op de webserver geïnstalleerd staat.

5.3 Deelvragen

Gedurende de ontwerpfase zijn de volgende deelvragen opgesteld:

1. Hoe kan een reële onderzoeksopstelling worden opgezet?
2. Welke data wordt verzameld door de PoC-opstelling?
3. Welke bevindingen zie ik binnen de PoC-opstelling in GNS3, en de PoC-opstelling op DigitalOcean?
4. Hoe ziet het uiteindelijke ontwerp van een mogelijk best passende SIEM-oplossing eruit?

5.4 Deelresultaten

Middels een GNS3-simulatie heb ik een reële onderzoeksopstelling opgezet. Uitleg hierover is te vinden enkele paragrafen hierboven (‘Deelresultaten’ in ‘Analysefase’). Er wordt gebruik gemaakt van Wazuh, want dit is in een deelresultaat geconcludeerde SIEM in de analysefase.

De PoC-opstelling op DigitalOcean draait 24/7. Dit betreft een Wazuh-installatie op Ubuntu Server 20.04. Er draait een webserver op poort 80 en 443 en een SSH-server op poort 22. Doel hiervan is om data te verkrijgen over echte aanvallen die uitgevoerd worden. In eerste instantie wordt gebruik gemaakt van de default configuratie van Wazuh.

In de Wazuh-webinterface op DigitalOcean zie ik tot nu toe alleen maar een grote hoeveelheid aan mislukte SSH-inlogpogingen. Alleen poort 22 staat open voor alle IP-adressen.

Binnen de GNS3-omgeving zie ik zoals verwacht geen meldingen binnen Wazuh als ik een TCP SYN flood-aanval simuleer middels Kali Linux. OSSEC is namelijk een host intrusion detection system (HIDS) en geen network intrusion detection system (NIDS). Een HIDS verzamelt data die niet netwerkgerelateerd is (zoals TCP Layer 4-aanvallen), terwijl een NIDS data verzamelt die wel netwerkgerelateerd is.
Ik heb nog niet geprobeerd wat er gebeurt als ik aanvallen doe om in te breken in het systeem, bijvoorbeeld privilege escalation of alle soorten brute force attacks.

6. Implementatiefase

6.1 Aanpak

Er is, naar aanleiding van de deelresultaten van de ontwerpfase, verder geëxperimenteerd met de SIEM-omgeving. Dit is in het vorige hoofdstuk beschreven.

In de implementatiefase zal de implementatie van de in de ontwerpfase opgestelde omgeving beschreven worden. In eerste instantie worden twee WordPress-VM's toegevoegd aan de GNS3- en DigitalOcean-simulaties. Daarna worden, onder andere middels Kali, testen uitgevoerd voor de finetuning van de opstellingen. De aanvallen en of ze wel/niet gedetecteerd worden door Wazuh, dragen bij aan het zo passend mogelijk maken van de omgeving.

6.2 Concrete werkzaamheden

In de implementatiefase heb ik gewerkt aan de implementatie van de SIEM binnen de productieomgeving van Level Level. Dit is nadrukkelijk niet bedoeld als eindproduct, maar nog steeds te onderzoeken hoe de SIEM zodanig aangepast kan worden dat het voldoet aan de eisen. Dit vindt plaats in de huidige topologie van Level Level, maar moet ook zonder aanpassingen werken in de nieuwe architectuur. De meest realistische omgeving om de SIEM te kunnen testen en verder te kunnen optimaliseren is in de productieomgeving. Dit hangt wel af van of er gebruik gemaakt wordt van de huidige of gewenste hostingarchitectuur. Omdat de nieuwe architectuur nog niet in gebruik is, doe ik dit in de huidige architectuur. In samenspraak met de opdrachtgever mag ik de SIEM in de productieomgeving testen, in eerste instantie met de webserver van Level Levels eigen site en de website van een klant.

Ik heb binnen DigitalOcean een VM ingericht met Wazuh. Op twee servers (een klant en de Level Level-webserver) is de OSSEC-agent geïnstalleerd en geconfigureerd met het IPv4-adres van de VM van Wazuh. Communicatie tussen de webservers en de Wazuh-manager lukte gelijk zonder problemen. Vervolgens heb ik gedurende enige tijd geobserveerd welke meldingen binnenkwamen. Het viel mij op dat dit allemaal SSH- of integriteitsgerelateerde meldingen waren, geen meldingen uit webserver logs. Dit kwam doordat Level Level niet standaard /var/log/nginx/error.log en /var/log/nginx/access.log gebruikt, maar /var/log/nginx/klantnaam.error.log en /var/log/nginx/klantnaam.access.log. Een collega wekte de suggestie dat dit misschien mogelijk was met het gebruik van een wildcard in de OSSEC-configuratie. Dit werkt inderdaad. Sindsdien krijgt de SIEM wel meer meldingen binnen, denk hierbij aan HTTP-foutmeldingen maar ook PHP fouten en waarschuwingen. Omdat naast integriteit en vertrouwelijkheid ook beschikbaarheid een belangrijk aspect is, heb ik het OSSEC level van PHP errors verhoogd van 5 naar 11. In de vorige fase heb ik namelijk besloten om alle meldingen van level 10 en hoger in Slack te laten zien, omdat dat in de praktijk grotendeels gaat om serieuze security-meldingen, en wat false positives die wel verdacht zijn. Omdat PHP errors de beschikbaarheid van webpagina's negatief beïnvloeden, heb ik er dus voor gekozen om het niveau naar 11 te verhogen zodat deze worden gemeld in het Slack-kanaal van Wazuh.

Daarnaast heb ik in de lokale testomgeving een active response-functie van Wazuh ingeschakeld, namelijk host-deny.sh. Wazuh bevat een aantal scripts die automatisch uitgevoerd kunnen worden als er verdacht verkeer vanaf een bepaald IPv4-adres naar een Wazuh-agent gaat. Eén van de standaard aanwezige scripts is host-deny.sh.
Deze voegt het bron IPv4- of IPv6-adres van het verdachte verkeer toe aan hosts.deny, waardoor er geen TCP-verkeer meer kan plaatsvinden (Venema, z.d.) vanaf het betreffende IP-adres voor een bepaalde hoeveelheid seconden.

Ik heb deelvragen opgesteld met als doel de werking van de SIEM te optimaliseren en aan te passen op de omgeving van Level Level, zoals het testen met bekende aanvallen, het doorgeven van meldingen aan beheerders.

6.3 Deelvragen

In de implementatiefase zijn deze deelvragen bedacht:

1. Hoe reageert de SIEM op bekende (CVE) aanvallen?
2. Hoe kan de SIEM urgente meldingen communiceren aan de beheerders?
3. In hoeverre moet de SIEM zelf nog geoptimaliseerd worden?
4. Hoe moet de SIEM zelf beveiligd worden om gebruikt te worden in een productieomgeving?
5. Hoe moet er op een veilige manier omgegaan worden met de SIEM?

Ik heb gekozen voor specifiek deze vragen omdat ze betrekking hebben op de situatie bij Level Level. Het is immers belangrijk dat de SIEM reageert op bekende aanvallen, omdat (klanten van) Level Level hier slachtoffer van kunnen worden. Daarnaast is het optimaliseren van de SIEM en het communiceren met beheerders ook erg belangrijk binnen de productieomgeving, zodat de SIEM optimaal ingezet kan worden en echt bijdraagt aan de veiligheid van de klantenwebsites.

6.4 Deelresultaten

De SIEM reageert standaard op bekende (CVE) aanvallen. Ik heb dit getest met een Shellstock attack[3], hier reageert hij op. Middels Kali heb ik namelijk een dergelijke aanval uitgevoerd. Dit triggert gelijk een Level 15 Alert in Wazuh. Dit heb ik zowel getest in de DO omgeving als in GNS3. Middels het nalopen van logfiles registreert OSSEC een eventuele aanval bij bepaalde (rules/decoders) verdacht gedrag.

[3] Bekende aanval, zie https://blog.cloudflare.com/inside-shellshock/ voor informatie

Wazuh heeft een verdeling waarin de security events worden ondergebracht. Dit gaat door middel van niveaus. Er wordt gewerkt met vijftien niveau's: 0 t/m 15 (1 bestaat niet). Niveau 0 is de laagste, niveau 15 is de hoogste. De classificatie van niveaus luidt:
0 - Genegeerd, niks te maken met veiligheid;
2 - Lage prio systeembericht, niks te maken met veiligheid;
3 - Gelukte of toegestane gebeurtenis;
4 - Lage prio systeemfout, komt door misconfiguratie of door ongebruikte software;
5 - Fout door gebruiker, denk aan verkeerd ingevulde sudo-wachtwoorden;
6 - Fouten die vaak voorkomen en redelijk onschuldige aanvallen;
7 - Logentries die woorden zoals 'bad' of 'error' bevatten, deze indiceren een vermoedelijke fout en kunnen zo goed wel als geen security relatie hebben;
8 - Gebeurtenissen die voor het eerst plaatsvinden, zoals een nieuwe gebruiker die de eerste keer inlogt. Het starten van monitoring events (zoals een sniffer) valt hier ook onder;
9 - Fouten m.b.t. root-account of inlogpogingen met onjuiste gebruikers of vanaf onjuist bronnen;
10 - Meerdere fouten van gebruikers, zoals meerdere mislukte inlogpogingen in korte tijd. Dit kan zowel een aanval zijn als een gebruiker die zijn/haar wachtwoord is vergeten;
11 - Veranderende integriteit, binaries die verandert zijn of gevonden rootkits. Er is een grote kans dat dit ontstaat door een succesvolle aanval, of door software updates;
12 - Gebeurtenis met hoge prioriteit, een fout/waarschuwing in het systeem, de kernel of specifieke applicaties. Dit betreft een waarschijnlijk succesvolle aanval;
13 - Ongewone fout, dit is een reeks van gebeurtenissen die in een bepaald patroon plaatsvinden die overeenkomst met het patroon dat door bekende aanvallen gebruikt wordt. Er is vrijwel zeker sprake van een aanval;
14 - Hoge prioriteit beveiliging gebeurtenis, dit is een gebeurtenis die door middel van correlatie van meerdere gebeurtenissen leidt tot een zeer vermoedelijke aanval;
15 - Zware aanval, dit is een zonder twijfel een aanval. Actie moet ondernomen worden.

Om met succes aanvallen te detecteren, wordt gebruik gemaakt van rulesets. OSSEC, de agents op de client(s), leest logfiles, commando-output, netwerksockets en systeeminformatie. Dit wordt doorgegeven aan OSSEC op de Wazuh-server (manager).

Elke gebeurtenis wordt vergeleken met rules. Rules bepalen aan welk niveau iets voldoet. Middels decoders wordt relevante informatie uit logbestanden gehaald. Daarna wordt middels rules bepaald aan welke niveau iets voldoet, en welke beschrijving de betreffende event krijgt. De rules zijn geschreven in XML en staan in een reeks bestanden. Dit werkt door middel van reguliere expressies en handmatige classificatie op niveaus. Hiermee is Wazuh helemaal naar de hand te zetten. Mocht er iets gedetecteerd worden, dan wordt het dus geclassificeerd aan de hand van rules. Er wordt bepaald of iets gevaarlijk is. Zo ja, dan wordt dat in Wazuh gezet. Middels triggers en alerts is te bepalen wanneer beheerders op de hoogte gesteld moeten worden.

De gewenste wijze om beheerders op de hoogte te stellen binnen de organisatie is via Slack. Slack wordt gebruikt voor communicatie binnen de organisatie, zowel voor directe berichten tussen personen en voor berichten in kanalen waaraan een deel of iedereen van de medewerkers deelneemt. Voor verschillende diensten wordt al gebruik gemaakt van Slack om meldingen door te geven, denk aan meldingen van klanten over problemen met hun site.

Middels webhooks worden deze berichten door de betreffende diensten automatisch in het corresponderende Slack-kanaal gezet. Een webhook is een webadres waarmee data vanaf de client naar de webserver gestuurd kan worden, en vervolgens verwerkt worden door de webserver. Wazuh ondersteund het plaatsen van berichten in Slack middels een webhook.

De organisatie wil graag dat er zo min mogelijk false positives, en uiteraard zoveel mogelijk true positives, gemeld worden aan de beheerders. De ervaring uit het verleden met vorige SIEM-systemen die binnen Level Level gebruikt zijn is dat er bij het optimaliseren van het systeem of veel false positives waren, of veel false negatives. Daarom is gekozen om alle meldingen van level 10 (en hoger) door te geven aan het betreffende Slack-kanaal. Hierdoor worden waarschijnlijke (mislukte en gelukte) aanvallen doorgegeven aan de beheerders. De betreffende beheerders zijn namelijk onderdeel van het Slack-kanaal waarnaar deze meldingen door Wazuh worden gedaan. De melding bevat onder andere het tijdstip van plaatsvinden van de gebeurtenis, de gebeurtenis, de corresponderende logregel of (in het geval van combinaties van regels of correlatie) de gebeurtenissen waardoor deze melding getriggerd wordt.

Een melding die in de vorige fase regelmatig naar voren kwam maar in deze fase niet, is een SSH brute force aanval. Omdat de SSH-poort, poort 22, niet openstaat voor alle IPv4-adressen is het niet mogelijk om vanaf de meeste IPv4-adressen een brute force aanval uit te voeren middels SSH. Zolang een brute force aanval mislukt is er geen reden tot zorg, zolang men zich ervan bewust is dat de SSH-poort open staat voor alle IPv4-adressen als bron. Dit levert dus een hoop false positives op. Tenminste, het gaat daadwerkelijk om SSH-inlogpogingen, maar deze mislukken en vormen daarmee geen bedreiging.

Doordat de organisatie gebruik maakt van een jumphost voor SSH en alleen die host met SSH kan inloggen op de webservers (ander verkeer wordt geblokkeerd door de firewall, alleen het bron IPv4-adres van jumphost mag middels SSH verbinden met de webservers) vindt er dus geen brute force aanval plaats. Indien dit wel gebeurt, gebeurt dit via de jumphost en dat is wel reden tot zorg. Dat impliceert namelijk dat een kwaadwillende toegang heeft tot de jumphost.

Wazuh draagt op deze manier bij aan security by design en hardening: hoe meer gehardend het systeem is, hoe meer het principle of less privilege gevolgd wordt, des te minder false positives er gedetecteerd worden. Het uitdagende proces van verkleinen van het aantal false positives en vergroten van het aantal true positives heeft als bijeffect dat het dus bijdraagt aan de gehele security. Immers, als het niet mogelijk is om vanaf elk IPv4-adres met SSH een verbinding te maken, kunnen er ook geen mislukte inlogpogingen ontstaan via SSH vanaf een degelijk IPv4-adres dat een melding triggert. Hoe kleiner de attack surface is, des te veiliger de omgeving is.

Wazuh kent zonder modificaties al veel rules, waaronder een aantal die met WordPress te maken hebben. Zodra iemand inlogt in WordPress, wordt er standaard bijvoorbeeld een melding getriggerd. Middels een WordPress-plugin (WP-fail2ban) kunnen mislukte WordPress-inlogpogingen en verwante gebeurtenissen gelogd worden in een systeemlog (/var/auth/log). Dit kan ook door Wazuh gebruikt worden. Sterker nog, door de al aanwezig standaard rules en decoders, wordt /var/auth/log al gecheckt en ook een melding gemaakt als de betreffende plug-in iets toevoegt aan de logfile.

Meldingen van andere WordPress-gerelateerde meldingen kunnen zodanig worden gebruikt dat ze meldingen genereren in Wazuh.

Uiteindelijk moet de oplossing geschikt zijn om één SIEM (Wazuh manager/server) te hebben waarbij elke webserver een OSSEC-agent heeft draaien. Alle verwerkingen vinden plaats op de server. Het goed afstemmen van welke data wel/niet verwerkt/opgeslagen wordt en hoe lang moet nog onderzocht worden. Dit is puur maatwerk: elke klant kan weer andere eisen/wensen hebben, en alles moet altijd aan de geldende wet/regelgeving voldoen. Door de manier waarop OSSEC en Wazuh werken (Wazuh classificeert meldingen, OSSEC filtert eventueel belangrijke meldingen) is het probleem waar SIEM's bekend om staan (veel false positives, veel resource usage) geen reële issue meer. Wel is het belangrijk om ruim voldoende opslagruimte te hebben, zodat niet tijdens langdurige aanvallen informatie gaat ontbreken waardoor er vervolgens geen goed onderzoek naar de toedracht van de reeks gebeurtenissen gedaan kan worden. Daarnaast is het belangrijk dat alleen mensen toegang hebben tot de SIEM-interface die dat ook echt nodig hebben (least amount of privilege).

In hoeverre de SIEM nog verder geoptimaliseerd moet worden om in een productieomgeving te gebruiken volgt in de volgende alinea's.

De SIEM is in de productieomgeving gekoppeld aan een klant en aan de website van Level Level zelf. Op DigitalOcean is een VM gemaakt met Ubuntu 20.04 LTS. Hier heb ik Wazuh op geïnstalleerd, het standaardwachtwoord van Wazuh veranderd ten behoeve van de veiligheid en TCP poorten 1514 en 1515 zijn opengezet in de firewall zodat de webservers verbinding kunnen maken hiermee.

De eerste kwetsbaarheid is gedetecteerd doordat een melding kwam over een mislukte reeks inlogpogingen die gedetecteerd werden als een SSH brute force attacks met meerdere mislukte inlogpogingen met wachtwoorden. Dit waren PAM-entries in de log die die meldden. Binnen de Level Level-omgeving wordt PAM[4] gebruikt in combinatie met SSH voor de authenticatie (PKI). Dat er een wachtwoordinlogpoging kwam is vreemd, omdat er uitsluitend gebruik gemaakt wordt van een PKI-infrastructuur om in te loggen. Mits goed ingesteld zou PAM helemaal geen password inlogpogingen moeten registreren, omdat password authenticatie voor geen enkel account nodig is. Dit duidde op een onveilige configuratie (of in ieder geval een niet optimaal beveiligde configuratie). Door de SSH-configuratie aan te passen is dit opgelost.

[4] PAM: pluggable authentication modules, een centraal systeem binnen Linux dat door verschillende applicaties ter authenticatie gebruikt kan worden.

De volgende configuratie is gebruikt om de SIEM te optimaliseren:
in /var/ossec/rulesets/rules/0265-php_rules.xml heb ik de volgende code aangepast:

```
<rule id="31420" level="6">
    <if_sid>31402, 31405</if_sid>
    <description>PHP Fatal error.</description>
  </rule>
```

naar

```
<rule id="31420" level="11">
    <if_sid>31402, 31405</if_sid>
    <description>PHP Fatal error.</description>
  </rule>
```

. Hierdoor worden PHP foutmeldingen voortaan als een level 11-event bestempeld in plaats van een level 6-event. Daarnaast heb ik de code van de Slack-integratie (/var/ossec/integrations/slack.py) aangepast: van

```
def main(args):
    debug("# Starting")
    # Read args
    alert_file_location = args[1]
    webhook = args[3]

    debug("# Webhook")
    debug(webhook)

    debug("# File location")
    debug(alert_file_location)

    # Load alert. Parse JSON object.
    with open(alert_file_location) as alert_file:
        json_alert = json.load(alert_file)
    debug("# Processing alert")
    debug(json_alert)

    debug("# Generating message")
    msg = generate_msg(json_alert)
    debug(msg)

    debug("# Sending message")
    send_msg(msg, webhook)
```

naar (zie volgende pagina):

```
def main(args):
    debug("# Starting")
    # Read args
    alert_file_location = args[1]
    webhook = args[3]
    with open('/var/ossec/etc/webhooks.txt','r') as f:
      webhookFile = f.read().splitlines()[0]
    #split webhook in strings in array of maps webhooks, consisting of
agent ID and webhook

    webhooks = dict(x.split("=") for x in webhookFile.split(";"))
    debug("# Webhook")
    debug(webhook)

    debug("# File location")
    debug(alert_file_location)

    # Load alert. Parse JSON object.
    with open(alert_file_location) as alert_file:
        json_alert = json.load(alert_file)
    print(json_alert)
    debug("# Processing alert")
    debug(json_alert)

    debug("# Generating message")
    msg = generate_msg(json_alert)
    debug(msg)

    debug("# Sending message")
    #webhook = corresponding webhook with agent ID
    for k, v in webhooks.items():
      if (v == json_alert['agent']['id']):
        send_msg(msg, k)
```

Om de slack integratie toe te voegen zodat het voldoet aan de wensen en eisen, dient het volgende gedaan te worden: de code van de Slack-integratie (/var/ossec/integrations/slack.py) kan worden aangepast van (zie volgende pagina)

```
def main(args):
    debug("# Starting")
    # Read args
    alert_file_location = args[1]
    webhook = args[3]

    debug("# Webhook")
    debug(webhook)

    debug("# File location")
    debug(alert_file_location)

    # Load alert. Parse JSON object.
    with open(alert_file_location) as alert_file:
        json_alert = json.load(alert_file)
    debug("# Processing alert")
    debug(json_alert)

    debug("# Generating message")
    msg = generate_msg(json_alert)
    debug(msg)

    debug("# Sending message")
    send_msg(msg, webhook)
```

naar (zie volgende pagina)

```
def main(args):
    debug("# Starting")
    # Read args
    alert_file_location = args[1]
    webhook = args[3]
    with open('/var/ossec/etc/webhooks.txt','r') as f:
      webhookFile = f.read().splitlines()[0]
    #split webhook in strings in array of maps webhooks, consisting of
agent ID and webhook

    webhooks = dict(x.split("=") for x in webhookFile.split(";"))
    debug("# Webhook")
    debug(webhook)

    debug("# File location")
    debug(alert_file_location)

    # Load alert. Parse JSON object.
    with open(alert_file_location) as alert_file:
        json_alert = json.load(alert_file)
    print(json_alert)
    debug("# Processing alert")
    debug(json_alert)

    debug("# Generating message")
    msg = generate_msg(json_alert)
    debug(msg)

    debug("# Sending message")
    #webhook = corresponding webhook with agent ID
    for k, v in webhooks.items():
      if (v == json_alert['agent']['id']):
        send_msg(msg, k)
```

Daarnaast moeten in het bestand /var/ossec/etc/webhooks.txt de webhooks gedefinieerd worden volgens het formaat ‘url=id;url=id’, waarbij url elke keer vervangen moet worden door de URL van de Slack-webhook van het betreffende kanaal die correspondeert met de agent, en id door de betreffende agent id. Hiermee kunnen alle events in een apart Slack-kanaal per agent geplaatst worden.

Voor de active response heb ik de volgende code toegevoegd in /var/ossec/etc/ossec.conf (zie volgende pagina):

```
<active-response>
  <command>host-deny</command>
  <location>local</location>
  <level>10</level>
  <timeout>600</timeout>
</active-response>
```

Zoals beschreven in 6.2 heb ik ook de OSSEC-configuratie aangepast omdat Level Level andere bestandsnamen gebruikt voor de nginx-logbestanden dan gebruikelijk. Deze aanpassing dient te gebeuren omdat er anders geen verdachte activiteit op de website van de klant gedetecteerd kan worden. Ik heb daartoe de volgende code aangepast in /var/ossec/etc/ossec.conf op de agents (webservers):

```
<localfile>
    <log_format>syslog</log_format>
    <location>/var/log/nginx/access.log</location>
  </localfile>
<localfile>
    <log_format>syslog</log_format>
    <location>/var/log/nginx/error.log</location>
  </localfile>
```

naar

```
<localfile>
    <log_format>syslog</log_format>
    <location>/var/log/nginx/*access.log</location>
  </localfile>
<localfile>
    <log_format>syslog</log_format>
    <location>/var/log/nginx/*error.log</location>
  </localfile>
```

Hiermee worden in plaats van alleen access.log en error.log ook alle andere bestanden die eindigen op access.log en error.log in de map /var/log/nginx, waardoor ook alle nginx-logbestanden van de klanten verwerkt worden (formaat: ‘klantnaam.access.log’ en ‘klantnaam.error.log’).

De SIEM is zonder verdere instellingen veilig te gebruiken, met uitzondering van het aanpassen van het wachtwoord. Daarnaast heb ik een lijst met organisatorische en procesmatige aanbevelingen opgesteld met betrekking tot een veilig gebruik van de SIEM. Deze lijst is te vinden in hoofdstuk 9. Daarnaast staan de technische aanbevelingen ook in dat hoofdstuk.

7. Deelvragen & -resultaten

In dit hoofdstuk staan de deelvragen die in de fasen opgesteld zijn:

- Hoe ziet de architectuur van de nieuwe hosting setup van Level Level eruit?
- Welke SIEM-oplossingen worden in overweging gebracht?
- Aan welke functionele eisen moet de SIEM voldoen en waarom?
- Hoe kan een reële onderzoeksopstelling worden opgezet?
- Welke data wordt verzameld door de PoC-opstelling?
- Welke bevindingen zie ik binnen de PoC-opstelling in GNS3, en de PoC-opstelling op DigitalOcean?
- Hoe ziet het uiteindelijke ontwerp van een mogelijk best passende SIEM-oplossing eruit?
- Hoe reageert de SIEM op bekende (CVE) aanvallen?
- Hoe kan de SIEM urgente meldingen communiceren aan de beheerders?
- In hoeverre moet de SIEM nog geoptimaliseerd worden?
- Hoe moet de SIEM zelf beveiligd worden om gebruikt te worden in een productieomgeving?
- Hoe moet er op een veilige manier omgegaan worden met de SIEM?

De hoofdvraag luidt ‘Hoe kan een SIEM zodanig geïntegreerd worden in een webhostingbedrijf dat deze zoveel mogelijk aanvallen opmerkt en daarop reageert?’. Gedeurende de onderzoeksfasen is deze vraag beantwoord middels deelvragen. Voor elke deelvraag wordt een beknopt antwoord gegeven, en aan de hand daarvan wordt de conclusie bepaald. Deze zal zich gaan bevinden in het volgende hoofdstuk.

- Hoe ziet de architectuur van de nieuwe hosting setup van Level Level eruit?
    - De nieuwe hostingarchitectuur bestaat uit een jumphost en SIEM voor alle klanten, en een Ubuntu 20.04 LTS LEMP-server en load balancer/firewall (reverse proxy) per klant.
- Welke SIEM-oplossingen worden in overweging gebracht?
    - Er zijn drie SIEM-oplossing in overweging gebracht: AlienVault OSSIM, Elasticsearch SIEM en Wazuh.
- Aan welke functionele eisen moet de SIEM voldoen en waarom?
    - Zie paragraaf 4.4.2.
- Hoe kan een reële onderzoeksopstelling worden opgezet?
    - Middels een lokale simulatie in GNS3 en een online simulatie op DigitalOcean kunnen reële onderzoeksopstelling worden opgezet.
- Welke data wordt verzameld door de PoC-opstelling?
    - De PoC-opstelling verzamelt data met betrekking tot beveiligingsgebeurtenissen. Dit gaat in de DigitalOcean-opstelling om reële data, omdat deze regelmatig aangevallen wordt door (hoogstwaarschijnlijk geautomatiseerde) kwaadwillenden.

- Welke bevindingen zie ik binnen de PoC-opstelling in GNS3, en de PoC-opstelling op DigitalOcean?
    - Zoals verwacht zie ik binnnen de PoC-opstelling op DigitalOcean een reeks aan uitgevoerde (mislukte) aanvallen. Binnen de GNS3-PoC-opstelling zie ik uitsluitend data van aanvallen die ik zelf middels Kali Linux binnen de omgeving uitvoer.
- Hoe ziet het uiteindelijke ontwerp van een mogelijk best passende SIEM-oplossing eruit?
    - Zie paragraaf 5.4.
- Hoe reageert de SIEM op bekende (CVE) aanvallen?
    - Indien er een CVE-kwetsbaarheid gevonden is die van toepassing is op de gebruikte omgeving, wordt een security event getriggerd. Zodra er misbruik gemaakt tracht te worden van een CVE-aanval, ook al is de omgeving gepatcht, dan wordt een een security event getriggerd.
- Hoe kan de SIEM urgente meldingen communiceren aan de beheerders?
    - Urgente meldingen worden middels Slack doorgegeven aan beheerders zodat deze actie kunnen ondernemen.
- In hoeverre moet de SIEM zelf nog geoptimaliseerd worden?
    - Optimalisatie hoeft nauwelijks uitgevoerd te worden. Er wordt immers gebruik gemaakt van veelgebruikte bestaande programmatuur zoals nginx en PHP waarvoor standaard rules en decoders bestaan en standaard gebruikt worden. De enige aanpassing die plaats moet vinden aan de uiteindelijke SIEM, is de wildcard toevoegen voor nginx en de configuratie verandering voor de Slack integratie. Daarnaast wordt het niveau van PHP-foutmeldingen verhoogd naar 11, omdat deze de beschikbaarheid negatief beïnvloeden. De broncode van de Wazuh-installatie heb ik zodanig aangepast dat security meldingen per klant in een verschillend Slack-kanaal geplaatst worden. Ten slotte zijn active responses ingeschakeld (hosts-deny.sh) zodat IPv4- en IPv6-adressen gedurende een uur geblokkeerd worden na een level 12 event (of hoger).
- Hoe moet de SIEM zelf beveiligd worden om gebruikt te worden in een productieomgeving?
    - De standaardwachtwoorden van de SIEM moeten aangepast worden. Organisatorisch en procesmatig, maar ook technisch, moeten alle aanbevelingen in hoofdstuk 9 gevolgd worden. Het is belangrijk dit op te volgen.
- Hoe moet er op een veilige manier omgegaan worden met de SIEM?
    - Het is belangrijk dat er alleen mensen toegang hebben tot de SIEM die dat daadwerkelijk nodig hebben. Daarnaast dienen alle aanbevelingen in hoofdstuk 9 opgevolgd te worden.

In het volgende hoofdstuk bevindt zich de conclusie, gevormd door de antwoorden op deze deelvragen.

8. Conclusie

Ik kan concluderen dat dit onderzoek met succes is afgerond. Het doel van de opdracht is immers het analyseren van de eisen waaraan de SIEM moet voldoen en vervolgens het verder onderzoeken van hoe een SIEM-oplossing optimaal voor Level Level kan worden ingezet. Dit doel is behaald. De hoofdvraag luidt: ‘Hoe kan een SIEM zodanig geïntegreerd worden in een webhostingplatform dat deze zoveel mogelijk valide aanvallen opmerkt en daarop reageert?’. Middels het doorlopen van de fasen zoals beschreven in dit verslag is deze vraag beantwoord. Het antwoord op de hoofdvraag luidt:

De gekozen SIEM-oplossing is een bestaande SIEM, namelijk Wazuh. Met de standaardconfiguratie werkt Wazuh al goed voor Level Level, want er is een grote hoeveelheid rules en decoders standaard aanwezig die overeenkomen met de webserverapplicaties (nginx, MariaDB, WordPress) die Level Level gebruikt. Wat wel veranderd is ten opzichte van de standaardconfiguratie, is dat PHP errors een hoger level moeten krijgen om de beschikbaarheid van websites te garanderen. Doch PHP foutmeldingen geen beveiligingsincident lijken, zijn ze het wel, omdat ze de beschikbaarheid van klantenwebsites negatief beïnvloeden. Daarnaast moet Wazuh zodanig geconfigureerd worden dat events in het daarvoor bestemde Slack-kanaal geplaatst worden. Ten slotte moet op de OSSEC-agents een wildcard gebruikt worden voor de nginx log, omdat niet alleen de standaard access.log en error.log gebruikt wordt maar ook klantnaam.nginx.log.

De beveiligingsgebeurtenissen die door de SIEM gedetecteerd worden kunnen gebruikt worden om nog meer security by design toe te passen. Denk hierbij aan het uitschakelen of verwijderen van ongebruikte applicaties waardoor ze ook niet misbruikt kunnen worden. Ik had verwacht ook regels te moeten toevoegen om aan te sluiten op de omgeving van Level Level, maar het blijkt dat Wazuh standaard al veel regels aan boord heeft met betrekking tot de bij Level Level gebruikte software. Daarom was het niet nodig om extra regels toe te voegen.

9. Aanbevelingen

In verband met de wens van de organisatie om meer logging met betrekking tot security te gebruiken wil Level Level een SIEM gaan gebruiken. Daar heb ik onderzoek naar gedaan. De SIEM verzamelt potentieel gevoelige informatie. Daarom is het erg belangrijk dat er op een wijze manier wordt omgegaan met de SIEM. Ik heb daarom de volgende lijst opgesteld:

1. Verwijder data die langer dan een aantal jaar is gelogd indien het persoonsgegevens bevat, bewaren van persoonsgegevens voor logging is gedurende x jaar toegestaan.
2. Verander de standaard credentials van Elasticsearch en geef alleen toegang aan medewerkers voor wie dat strikt noodzakelijk is.
3. Zorg ervoor dat alleen de webinterface van buitenaf toegankelijk is. De Elasticsearch API en Wazuh API zelf hoeven niet te communiceren met de buitenwereld. Alleen Kibana (webinterface) moet kunnen communiceren met de buitenwereld.
4. Zorg ervoor dat Wazuh alleen via HTTPS bereikbaar is, en dat er gebruik gemaakt wordt van een geldig TLS certificaat. Gebruik bij voorkeur TLS 1.3. Dit is momenteel de nieuwste en veiligste versie van TLS (Rescorla, 2018).
5. Maak gebruik van een betrouwbare hostingpartij met een SLA en ISO-gecertificeerd (ISO 27001).
6. Zorg ervoor dat de SSH-interface van de Wazuh-server alleen toegankelijk is middels een Public Key Infrastructure.
7. Zet Wazuh in om te monitoren of software binnen de organisatie up-to-date is.
8. Kijk hoe de systemen van Cloudflare en WordFence te integreren zijn in Wazuh (zowel Cloudflare als WordFence worden gebruikt binnen de organisatie).

De configuratie die gebruikt kan worden, op basis van de in DigitalOcean-gebruike PoC-opstelling in de implementatiefase, is beschreven in dit verslag. Hier volgt de uiteindelijke configuratie ten opzichte van een ‘kale’ Ubuntu 20.04 LTS installatie. De stappen die doorlopen zijn luiden:

- Installatie Wazuh
- Configuratiebestanden Wazuh vervangen door config in dit verslag
- Standaardcredentials veranderen
- Agents installeren

Voor de installatie van Wazuh is gebruik gemaakt van de unattended all-in-one installatie van Wazuh (Wazuh Inc., z.d.). Daarnaast zijn de volgende veranderingen teweeggebracht in de configuratiebestanden:

- Slack integratie toevoegen (middels een aangepaste integratie).
- Niveau van fatale PHP fout verhogen van 6 naar 11.
- Active responses aanzetten.
- Ubuntu en NVD[5] vulnerability aanzetten.

[5] National Vulnerability Database: database die beveiligingskwetsbaarheden bevat, eigendom van de Amerikaanse overheid en wordt internationaal gebruikt.

Om de slack integratie toe te voegen zodat het voldoet aan de wensen en eisen, dient het volgende gedaan te worden: de code van de Slack-integratie (/var/ossec/integrations/slack.py) kan worden aangepast van

```
def main(args):
    debug("# Starting")
    # Read args
    alert_file_location = args[1]
    webhook = args[3]

    debug("# Webhook")
    debug(webhook)

    debug("# File location")
    debug(alert_file_location)

    # Load alert. Parse JSON object.
    with open(alert_file_location) as alert_file:
        json_alert = json.load(alert_file)
    debug("# Processing alert")
    debug(json_alert)

    debug("# Generating message")
    msg = generate_msg(json_alert)
    debug(msg)

    debug("# Sending message")
    send_msg(msg, webhook)
```

naar (zie volgende pagina):

```
def main(args):
    debug("# Starting")
    # Read args
    alert_file_location = args[1]
    webhook = args[3]
    with open('/var/ossec/etc/webhooks.txt','r') as f:
      webhookFile = f.read().splitlines()[0]
    #split webhook in strings in array of maps webhooks, consisting of
agent ID and webhook

    webhooks = dict(x.split("=") for x in webhookFile.split(";"))
    debug("# Webhook")
    debug(webhook)

    debug("# File location")
    debug(alert_file_location)

    # Load alert. Parse JSON object.
    with open(alert_file_location) as alert_file:
        json_alert = json.load(alert_file)
    print(json_alert)
    debug("# Processing alert")
    debug(json_alert)

    debug("# Generating message")
    msg = generate_msg(json_alert)
    debug(msg)

    debug("# Sending message")
    #webhook = corresponding webhook with agent ID
    for k, v in webhooks.items():
      if (v == json_alert['agent']['id']):
        send_msg(msg, k)
```

Daarnaast moeten in het bestand /var/ossec/etc/webhooks.txt de webhooks gedefinieerd worden volgens het formaat ‘url=id;url=id’, waarbij url elke keer vervangen moet worden door de URL van de Slack-webhook van het betreffende kanaal die correspondeert met de agent, en id door de betreffende agent id. Hiermee kunnen alle events in een apart Slack-kanaal per agent geplaatst worden.

Het niveau van de ‘PHP Fatal error’ kan verhoogd worden van 6 naar 11 op de volgende wijze: in /var/ossec/rulesets/rules/0265-php_rules.xml dient de volgende code aangepast te worden van (zie volgende pagina):

```
<rule id="31420" level="6">
    <if_sid>31402, 31405</if_sid>
    <description>PHP Fatal error.</description>
  </rule>
```

naar:

```
<rule id="31420" level="11">
    <if_sid>31402, 31405</if_sid>
    <description>PHP Fatal error.</description>
  </rule>
```

De active response ‘host-deny.sh’, die SSH-toegang weigert voor IP-adressen (zowel IPv4 als IPv6) die een aantal keer kort achter elkaar een mislukte inlogpoging via SSH hebben gedaan, kan op de volgende manier ingeschakeld worden. Deze code dient toegevoegd te worden aan /var/ossec/etc/ossec.conf:

```
<active-response>
  <command>host-deny</command>
  <location>local</location>
  <level>10</level>
  <timeout>600</timeout>
</active-response>
```

De vulnerability detectie dient op de volgende wijze geactiveerd te worden (aanpassing dient te worden gedaan in /var/ossec/etc/ossec.conf):

```
<vulnerability-detector>
    <enabled>no</enabled>
    <interval>5m</interval>
    <ignore_time>6h</ignore_time>
    <run_on_start>yes</run_on_start>

    <!-- Ubuntu OS vulnerabilities -->
    <provider name="canonical">
      <enabled>no</enabled>
      <os>trusty</os>
      <os>xenial</os>
      <os>bionic</os>
      <os>focal</os>
      <update_interval>1h</update_interval>
    </provider>
```

naar (zie volgende pagina)

```
<vulnerability-detector>
    <enabled>yes</enabled>
    <interval>5m</interval>
    <ignore_time>6h</ignore_time>
    <run_on_start>yes</run_on_start>

    <!-- Ubuntu OS vulnerabilities -->
    <provider name="canonical">
      <enabled>yes</enabled>
      <os>trusty</os>
      <os>xenial</os>
      <os>bionic</os>
      <os>focal</os>
      <update_interval>1h</update_interval>
    </provider>
```

De correcte verwerking van de logbestanden van de nginx-webserver die Level Level gebruikt kan worden bereikt door de volgende aanpassing. Verander in /var/ossec/etc/ossec.conf (op alle agents) de volgende code van

```
<localfile>
    <log_format>syslog</log_format>
    <location>/var/log/nginx/access.log</location>
  </localfile>
<localfile>
    <log_format>syslog</log_format>
    <location>/var/log/nginx/error.log</location>
  </localfile>
```

naar

```
<localfile>
    <log_format>syslog</log_format>
    <location>/var/log/nginx/*access.log</location>
  </localfile>
<localfile>
    <log_format>syslog</log_format>
    <location>/var/log/nginx/*error.log</location>
  </localfile>
```

Naast het eenmalig nalopen en implementeren van deze lijst is het belangrijk eventuele datalekken te melden indien dat moet[6]. Daarnaast is het regelmatig en systematisch opnieuw doorlopen van deze lijst ook aanbevolen, zodat eventuele veranderingen er niet voor langere tijd voor zorgen dat er niet aan deze lijst meer wordt voldaan, met een hoger risico op datalekken.

[6] actuele regelgeving met betrekking tot het melden van datalekken staat op https://autoriteitpersoonsgegevens.nl/nl/onderwerpen/beveiliging/meldplicht-datalekken

10. Reflectie

De gekozen beroepstaken zijn:

- A1 - Analyseren probleemdomein & opstellen probleemstelling
    - Het Plan van Aanpak en het uiteindelijke systeem moeten gericht zijn op de probleemstelling welke voortkomt uit analyse van het probleemdomein.
        - Deze beroepstaak heb ik behaald, ik heb namelijk een probleemstelling opgesteld en in alle fasen het probleemdomein geanalyseerd.
- B2 - Adviseren over inrichting van ICT-gerelateerde oplossingen en processen
    - Het doel van de opdracht is het opleveren van een SIEM-oplossing. Advies geven over de inrichting en gebruik hiervan is onderdeel van de opdracht.
        - Deze beroepstaak heb ik behaald, ik heb namelijk een onderzoeksresultaat opgeleverd over in hoeverre een SIEM-oplossing verder geoptimaliseerd moet worden om binnen de productieomgeving van Level Level te gebruiken.
- C4 - Ontwerpen technische infrastructuur
    - Het doel van de opdracht is het opleveren van een SIEM-oplossing. Dit is een technisch product dat ontworpen en geïmplementeerd wordt.
        - In de ontwerpfase heb ik de technische infrastructuur ontworpen en in de implementatiefase ten behoeve van de laatste onderzoeksfase geïmplementeerd. Deze beroepstaak heb ik dus behaald.
- C8 - Ontwerpen stelsel van securitymaatregelen
    - Om de veiligheid van de websites te waarborgen en de SIEM optimaal te laten werken moeten richtlijnen worden opgesteld om te bepalen hoe de hosting van websites en de werking van de SIEM op een veilige manier kan plaatsvinden.
        - Ik lever, naast het onderzoek beschreven in dit verslag, een lijst met aanbevelingen op welke ik ten strengste aanraadt aan Level Level om ze te volgen, zodat ze voldoen aan de wet- en regelgeving, security best practices en mijn mening wat betreft het omgaan met data die de SIEM verzamelt.
- Gc - Kritisch, onderzoekend en methodisch werken
    - Omdat het eindproduct zo foutloos mogelijk moet werken, moet kritisch gewerkt worden. Daarnaast moet onderzoeken en methodisch gewerkt worden, omdat er onderzoek gepleegd zal worden naar welke bestaande of niet-bestaande SIEM-oplossing het beste gebruikt kan worden.
        - Gedurende de iteraties heb ik mij vaak afgevraagd wat de beste manier is om iets aan te pakken, zowel technisch als niet-technisch. Ik heb gedurende het hele project, zowel inhoudelijk als procesmatig, gewerkt met Kanban als methodiek. Alle taken heb ik methodisch aangepakt, zoals mij geleerd is in projecten gedurende de opleiding.

11. Evaluatie

11.1 Aanpak

Aan het begin van het project twijfelde ik welke aanpak ik zou kiezen: ‘Welke methodiek past het beste bij de opdracht?’, ‘Pak ik het gefaseerd aan of niet?’ en ‘Hoe ziet de opdracht er überhaupt uit?’. Ik heb uiteindelijk gekozen voor Kanban in combinatie met een fasering. De methoden die ik in overweging heb genomen, zijn Agile-methodieken (zoals Scrum), Kanban en watervalmethoden en verwanten (V-model). Ten grondslag staat de vraag wat voor opdracht het is: veranderen de eisen continu? Is het erg als de eisen op een later moment veranderen? Is onderzoek naar de eisen onderdeel van de opdracht? Wat lever ik op?

Aan de hand van antwoorden op deze vragen (eisen veranderen niet per se, zou wel kunnen door inzichten die de opdrachtgever of ik naarmate vordering van opdracht krijgen doordat we beiden weinig kennis hadden van SIEM-oplossingen en de (niet-)functionele en technische werking ervan, opdracht kan grote vertraging krijgen als eisen veranderen op later moment en alles opnieuw doorlopen moet worden, wensen en eisen en omgeving onderzoek ik ook) heb ik geconcludeerd dat een watervalmethode te veel risico’s met zich meebrengt. Een Agile-methodiek zoals Scrum kan wel goed werken, en Kanban ook. Ik heb uiteindelijk gekozen voor Kanban omdat Scrum werkt met Sprints met vaste taken per iteratie. Daarvan kan niet afgeweken worden. Gezien de aard van dit onderzoek kan het zo zijn dat bepaalde taken op het moment dat ik eraan werk veranderd worden omdat ik tot inzichten kom dat het niet handig is om dan al aan die taak te werken. Deze flexibiliteit kent Scrum wel voor en na sprints, maar niet tijdens een sprint. Met Kanban kan dit wel, omdat er geen vaste iteraties zijn. Om toch enige structuur in de planning aan te brengen, maak ik wel gebruik van iteraties, maar taken kunnen elk moment toegevoegd worden aan de huidige iteratie.

11.1.1 Methode

In feite gebruik ik dus Kanban met een fasering en per fasering iteraties. Elke iteratie duurt een week. Ik ben blij dat ik hiervoor in samenspraak met mijn opdrachtgever en begeleider van school voor heb gekozen, want dit lijkt mij de methode met de minste risico’s voor dit specifieke project. Op elk moment zouden namelijk aanpassingen kunnen plaatsvinden die verandering in een vorige fase eisen. Dit probeer ik zoveel mogelijk te voorkomen omdat het de structuur van de gebruikte fasering onderuit haalt. Mocht het toch gebeuren dat er iets vanuit een vorige fase moet gebeuren, dan is dat geen probleem. Dit is gedurende deze stage ook gebeurd.

11.1.2 Plan van Aanpak

Het Plan van Aanpak was later af dan ik beoogd had. Dit kwam vooral doordat de eisen verder gespecificeerd moesten worden na inlevering van twee conceptversies. Desondanks ben ik voordat het PvA helemaal goedgekeurd was, wel alvast aan de slag gegaan met de oriëntatiefase, ondanks dat ik nog niet zeker wist dat zowel de opdrachtgever als de opleiding het goed zouden vinden als ik de betreffende fasering gebruik.
Dit was dus een risico, al had ik toen wel al positieve feedback gehad op de fasering die ik wilde gebruiken, maar formeel was het nog niet goedgekeurd.

11.1.3 Uitlooptijd

Inhoudelijk kostte het inrichten van een GNS3-omgeving en het opzetten van een simulatie veel tijd. Dit is niet ten koste gegaan van het inhoudelijke deel van de opdracht, maar toen ik daar mee bezig was heb ik gedurende die weken relatief weinig gewerkt aan het verslag. Om, vooral op het gebied van het verslag, puntjes op de i te zetten voor de inleverdeadline heb ik twee uitloopweken ingepland. Ik heb mijn best gedaan om dit niet in te hoeven zetten. Er kunnen namelijk altijd onverhoopte gebeurtenissen plaatsvinden waardoor het, om wat voor reden dan ook, toch nodig is om deze uitlooptijd te benutten voor de opdracht.

11.1.4 Achterstand

Ik begon later dan gepland aan de ontwerpfase omdat het opzetten van de GNS3 omgeving meer tijd dan gepland in beslag nam. Uiteindelijk heb ik deze achterstand helemaal ingehaald: het implementeren van de SIEM (met als doel informatie vergaren over hoe hij geoptimaliseerd kan worden voor Level Level) kostte aanzienlijk minder tijd dan ik van te voren verwachtte. Daardoor kon ik meer tijd besteden aan het verslag. Daarnaast wist ik van te voren al dat het opzetten van de omgeving in GNS3 veel haken en ogen zou kunnen hebben en dat het veel tijd zou kosten, maar dat het wel een zeer reële testomgeving opleverde waarmee ik in volgende fasen veel tijd kon gaan besparen.

11.1.5 Bedrijfsbezoek

Op 19 maart had ik het bedrijfsbezoek van mijn begeleider. Hier bespraken mijn opdrachtgever en ik de stand van zaken van mijn opdracht met mijn begeleider van school. Ik kreeg veel feedback op mijn aanpak van de begeleider op school. Ik heb hier veel aan gehad. Ik heb hier ook een demo gegeven. Op die demo kreeg ik nuttige feedback welke ik gebruikt heb om de uiteindelijke demo te maken voor het eindassessment. Tijdens dit bezoek heb ik ook kritiek gekregen op de wijze waarop de aanvliegroute van het project in het plan van aanpak in beschreven en hoe ik het daadwerkelijk doe. Volgens het Plan van Aanpak ligt de aanpak vooral op het implementeren van een SIEM, en minder op het analyseren. Dit terwijl ik wel veel bezig was met analyseren, en dat belangrijker is dan implementeren. De aanpak staat dus anders beschreven in dit verslag zodat de nadruk veel meer op analyse en onderzoek ligt en nauwelijks meer op implementatie. Deze feedback heb ik dus verwerkt.

11.1.6 Tussentijds assessment

Op 21 april had ik het tussentijds assessment. Ik kreeg hier veel feedback met betrekking tot mijn verslag waar ik niet nader op in ga, omdat het grotendeels gaat om kleine verbeterpuntjes zoals het gebruiken van een ik-vorm in plaats van derde persoon en het gebruik van hoofdstuknummering. Het belangrijkste kritiekpunt betreft het gebrek aan specificatie, meetbaarheid en testbaarheid aan de eisen die ik heb opgesteld aan de hand van de wensen. In de weken na dit assessment heb ik alle feedback verwerkt.

11.1.7 Invloed coronavirus COVID-19

Ondanks de coronamaatregelen - waardoor ik maar tweemaal op kantoor geweest ben en de begeleidende examinator nooit in het echt heb kunnen ontmoeten - heb ik toch veel geleerd van deze stage. Ten opzichte van mijn derdejaarsstage merk ik wel dat ik minder heb meegekregen van het dagelijkse gang van zaken.

Dit heeft mij niet beïnvloed in het maken van de opdracht, mocht ik informatie missen dan vroeg ik het altijd aan iemand. Er vonden regelmatig team standups plaats, waarbij iedereen wordt bijgepraat over nieuws binnen het bedrijf. Daarnaast werden ook regelmatig borrels georganiseerd. Al met al heeft Level Level zich naar mijn idee ruim voldoende ingezet om mijn stage noodgedwongen op afstand toch goed te faciliteren. Hetzelfde geldt voor de opleiding. Hierdoor heb ik toch genoten van mijn tijd als afstudeerder.

11.2 Productevaluatie

Gedurende dit project heb ik de volgende producten opgeleverd:

1. Afstudeerplan.
2. Plan van Aanpak.
3. Eventuele Proof of Concept-opstelling.
4. Afstudeerverslag (bestaande uit onder andere aanbevelingen en testrapport).

Ik kies er voor al deze producten te evalueren, omdat er geen sprake is van een technische opstelling als eindproduct. Alle in deze lijst genoemde producten zijn te evalueren.

11.2.1 Afstudeerplan

Het eerste, weliswaar alleen ten behoeve van de opleiding, opgeleverde product is het afstudeerplan. Deze bestaat uit een aantal belangrijke zaken, zoals beroepstaken, persoonlijke gegevens over de student, gegevens over het bedrijf, gegevens over de examinatoren van school, en gegevens over de begeleider van het bedrijf. Daarnaast bevinden zich de aanleiding van de opdracht, doelstelling en probleemstelling zich ook in dit document.

11.2.2 Plan van Aanpak

Het Plan van Aanpak bestaat uit een beschrijving van de gekozen aanpak van de opdracht, en het Pakket van Eisen. De planning maakt hier ook onderdeel van uit.

11.2.3 Eventuele Proof of Concept-opstellingen

Gedurende het project heb ik in de analyse-, ontwerp-, en implementatiefase onderzoek gedaan middels PoC-opstellingen ter ondersteuning van beantwoording van de deelvragen. De uiteindelijk Proof of Conceptopstelling voldoet zodanig aan de wensen en eisen dat deze zonder aanpassingen in de productieomgeving van Level Level geïmplementeerd kan worden. Alle hiervoor benodigde informatie staat in dit afstudeerverslag. Er zijn twee PoC-opstellingen gebruikt: één binnen een lokale simulatie en één binnen de productieomgeving. De laatstgenoemde gebruikt echte data van servers van klanten van Level Level.

11.2.4 Afstudeerverslag (het onderzoek)

Het belangrijkste eindproduct van deze stage is het afstudeerverslag, omdat de aanbevelingen aan Level Level (inclusief configuratie van de SIEM zodat deze in de productieomgeving gebruikt kan worden) hier deel van uitmaken. Ik heb alle eisen behaald.

Als ik naar de initiële wensen kijk heb ik deze wensen niet behaald, maar dat kwam omdat deze gedurende de fasen verder gespecificeerd worden waardoor ze niet meer op meerdere manieren geïnterpreteerd kunnen worden. Ik heb alle fasen doorlopen en de deelvragen beantwoord, waarmee ik voldoende informatie heb vergaard om de aanbevelingen te schrijven en daarmee het verslag af te ronden. Aan de hand van een hoofdvraag, en antwoorden op de deelvragen, heb ik de conclusie opgesteld.

12. Geraadpleegde literatuur

- NBIP. (z.d.). NaWas – de Nationale Wasstraat tegen DDos-aanvallen. Geraadpleegd op 9 april 2021, van https://www.nbip.nl/nawas/
  OpenNebula – Open Source Cloud & Edge Computing Platform. (z.d.). OpenNebula. Geraadpleegd op 9 april 2021, van https://opennebula.io/
  OSSIM: The Open Source SIEM | AlienVault. (z.d.). AlienVault. Geraadpleegd op 9 april 2021, van https://cybersecurity.att.com/products/ossim
- Europese Unie. (2016, 27 april). VERORDENING (EU) 2016/679 VAN HET EUROPEES PARLEMENT EN DE RAAD. Autoriteit Persoonsgegevens. https://autoriteitpersoonsgegevens.nl/sites/default/files/atoms/files/verordening_2016_-_679_definitief.pdf
- APNIC. (2021, 5 januari). IPv4 routing table since 1994, as seen by Route Views peers [Grafiek]. BGP in 2020 – The BGP Table | APNIC Blog. https://blog.apnic.net/wp-content/uploads/2021/01/Figure-1-%E2%80%93-IPv4-routing-table-since-1994-as-seen-by-Route-Views-peers.png
- Venema, W. (z.d.). hosts.deny(5) - Linux man page. Die.Net. Geraadpleegd op 14 mei 2021, van https://linux.die.net/man/5/hosts.deny
- Van Houten, P., Spruit, M., & Wolters, K. (2015). Informatiebeveiliging onder controle (3de editie). Amsterdam, Nederland: Pearson Benelux.
- Johansen, G. (2020). Digital Forensics and Incident Response – Second Edition. Packt Publishing.
- Bejtlich, R. (2013). The Practice of Network Security Monitoring. No Starch Press.
- Wazuh Inc. (z.d.). Unattended installation - All-in-one deployment. Wazuh - The Open Source Security Platform. Geraadpleegd op 25 mei 2021, van https://documentation.wazuh.com/current/installation-guide/open-distro/all-in-one-deployment/unattended-installation.html
- Berman, D. (2019, 7 november). 11 Open Source SIEM Tools. Logz.Io. https://logz.io/blog/open-source-siem-tools/
- Rescorla, E. (2018, augustus). RFC 8446. Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/rfc8446

13. Afkortingen

Afkorting	Betekenis
BGP	Border Gateway Protocol: protocol om IP-routes automatisch tussen routers te delen
ISP	Internet Service Provider: bedrijf dat internetverbindingen aanbiedt
SIEM	Security Information & Event Management: systeem dat gebruikt wordt om informatie over beveiligingsgebeurtenissen te verzamelen en verwerken
IP	Internet Protocol
DHCP	Dynamic Host Configuration Protocol: protocol om geautomatiseerd IPv4- en IPv6-adressen aan apparaten toe te wijzen
LXC	Linux Containers: software om middels containerisatie applicaties op Linux te draaien
LXD	Managementprogramma voor LXC, LXC-containers die gebruikmaken van dit programma worden LXD-containers genoemd
GNS3	Graphical Network Simulator 3: softwarepakket om computernetwerken mee te simuleren
TCP	Transmission Control Protocol: protocol om, op basis van IPv4- en IPv6-adressen, verbindingen op te zetten tussen netwerkapparatuur en eindsystemen
SYN	Synchronise: eerste pakketje wat vanaf TCP-client naar TCP-server wordt gestuurd om een TCP-verbinding op te zetten
SSH	Secure Shell: protocol om versleutelde terminalverbindingen op te zetten
HTTP	HyperText Transfer Protocol: protocol om met webservers te communiceren

NaWas	Nationale Wasstraat: afweersysteem tegen DDoS-aanvallen, het betreffende DDoS-verkeer wordt weggefilterd gedurende een aanval door het via een reeks netwerkapparaten te leiden die filtering toepassen
SOC	Security Operations Center: afdeling die alle technische cyber security-gerelateerde zaken constant monitort, en zo nodig ingrijpt bij incidenten zoals succesvolle aanvallen
PAM	Pluggable Authentication Modules: een centraal systeem binnen Linux dat door verschillende applicaties ter authenticatie gebruikt kan worden
GUI	Graphical User Interface: grafische gebruikersinface
CLI	Command-Line Interface: gebruikersinterface met opdrachtregel

Bijlagen

A. Routerconfiguraties in lokale simulatie

In deze bijlage bevinden zich de routerconfiguraties die gebruikt zijn op de gevirtulaliseerde Cisco-routers in de lokale GNS3-simulatie.

ISP1

```
interface GigabitEthernet0/0
 ip address 1.1.2.1 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/1
 ip address 1.1.3.1 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/2
 ip address dhcp
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/3
 ip address 1.0.0.1 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/4
 no ip address
 shutdown
 duplex auto
 speed auto
 media-type rj45
!
router bgp 1
 bgp log-neighbor-changes
 network 0.0.0.0
 redistribute connected
 neighbor 1.1.2.2 remote-as 2
 neighbor 1.1.3.2 remote-as 3
 neighbor 1.1.4.2 remote-as 4
```

```
 default-information originate
```

ISP2

```
interface GigabitEthernet0/0
 ip address 1.1.2.2 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/1
 ip address 1.2.4.1 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/2
 ip address 1.2.3.1 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/3
 ip address 2.0.0.1 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
!
router bgp 2
 bgp log-neighbor-changes
 redistribute connected
 neighbor 1.1.2.1 remote-as 1
 neighbor 1.2.3.2 remote-as 3
 neighbor 1.2.4.2 remote-as 4
```

ISP3

```
interface GigabitEthernet0/0
 ip address 1.1.3.2 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/1
 ip address 1.3.4.1 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
```

```
!
interface GigabitEthernet0/2
 ip address 1.2.3.2 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/3
 ip address 3.0.0.1 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
!
router bgp 3
 bgp log-neighbor-changes
 network 3.0.0.0 mask 255.255.255.0
 redistribute connected
 neighbor 1.1.3.1 remote-as 1
 neighbor 1.2.3.1 remote-as 2
 neighbor 1.3.4.2 remote-as 4
```

ISP4

```
interface GigabitEthernet0/0
 ip address 1.2.4.2 255.255.255.252
 ip nat inside
 ip virtual-reassembly in
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/1
 ip address 1.3.4.2 255.255.255.252
 ip nat inside
 ip virtual-reassembly in
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/2
 ip address dhcp
 ip nat outside
 ip virtual-reassembly in
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/3
```

```
 ip address 4.0.0.1 255.255.255.0
 ip nat inside
 ip virtual-reassembly in
 duplex auto
 speed auto
 media-type rj45
!
router bgp 4
 bgp log-neighbor-changes
 network 0.0.0.0
 redistribute connected
 neighbor 1.1.4.1 remote-as 1
 neighbor 1.2.4.1 remote-as 2
 neighbor 1.3.4.1 remote-as 3
 default-information originate
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
ip nat inside source list 10 interface GigabitEthernet0/2 overload
!
!
!
access-list 10 permit any
```

R-datacenter

```
interface GigabitEthernet0/0
 ip address 1.0.0.2 255.255.255.252
 ip nat outside
 ip virtual-reassembly in
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/1
 ip address 10.0.0.1 255.255.255.0
 ip nat inside
 ip virtual-reassembly in
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/2
 no ip address
```

```
 shutdown
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/3
 no ip address
 shutdown
 duplex auto
 speed auto
 media-type rj45
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
ip nat pool NAT 1.0.0.2 1.0.0.2 netmask 255.255.255.0
ip nat inside source list 10 pool NAT
ip nat inside source static tcp 10.0.0.10 22 1.0.0.2 22 extendable
ip nat inside source static tcp 10.0.0.10 80 1.0.0.2 80 extendable
ip route 0.0.0.0 0.0.0.0 1.0.0.1
!
!
!
access-list 10 permit 10.0.0.0 0.255.255.255
```

R-beheer

```
interface GigabitEthernet0/0
 ip address 2.0.0.2 255.255.255.252
 ip nat outside
 ip virtual-reassembly in
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/1
 ip address 10.0.0.1 255.255.255.0
 ip nat inside
 ip virtual-reassembly in
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/2
 no ip address
```

```
 shutdown
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/3
 no ip address
 shutdown
 duplex auto
 speed auto
 media-type rj45
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
ip nat pool NAT 2.0.0.2 2.0.0.2 netmask 255.255.255.0
ip nat inside source list 10 pool NAT overload
ip nat inside source static tcp 10.0.0.10 22 2.0.0.2 22 extendable
ip nat inside source static tcp 10.0.0.10 80 2.0.0.2 80 extendable
ip nat inside source static tcp 10.0.0.11 1514 2.0.0.2 1514 extendable
ip route 0.0.0.0 0.0.0.0 2.0.0.1
!
!
!
access-list 10 permit 10.0.0.0 0.255.255.255
```

R-kantoor_beheer

```
interface GigabitEthernet0/0
 ip address 3.0.0.2 255.255.255.252
 ip nat outside
 ip virtual-reassembly in
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/1
 ip address 10.0.0.1 255.255.255.0
 ip nat inside
 ip virtual-reassembly in
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/2
 ip address 5.0.0.1 255.255.255.0
```

```
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/3
 no ip address
 shutdown
 duplex auto
 speed auto
 media-type rj45
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
ip nat pool LAN 3.0.0.2 3.0.0.2 netmask 255.255.255.0
ip nat inside source list 10 pool LAN overload
ip route 0.0.0.0 0.0.0.0 3.0.0.1
!
!
!
access-list 10 permit 10.0.0.0 0.255.255.255
```

R-bezoeker

```
interface GigabitEthernet0/0
 ip address 4.0.0.2 255.255.255.0
 ip nat outside
 ip virtual-reassembly in
 duplex auto
 speed auto
 media-type rj45
 no cdp enable
!
interface GigabitEthernet0/1
 ip address 10.0.0.1 255.255.255.0
 ip nat inside
 ip virtual-reassembly in
 duplex auto
 speed auto
 media-type rj45
 no cdp enable
!
interface GigabitEthernet0/2
 no ip address
 shutdown
 duplex auto
```

```
 speed auto
 media-type rj45
 no cdp enable
!
interface GigabitEthernet0/3
 no ip address
 shutdown
 duplex auto
 speed auto
 media-type rj45
 no cdp enable
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
ip nat pool NAT 4.0.0.2 4.0.0.2 netmask 255.255.255.0
ip nat inside source list 10 pool NAT overload
ip route 0.0.0.0 0.0.0.0 4.0.0.1
!
!
!
access-list 10 permit 10.0.0.0 0.255.255.255
```

B. Testrapport

Om aan te tonen dat de eisen SMART zijn heb ik een testrapport gemaakt. De precondities van alle SIEM's zijn hetzelfde. Ze bestaan namelijk uit de in dit verslag beschreven lokale GNS3-testomgeving en de DigitalOcean-omgeving.

Eis 1: 1. Navigeer naar de SIEM URL middels een webbrowser.
2. Type de credentials 'admin' (gebruikersnaam) en 'admin' (wachtwoord) en druk op 'Login'.
3. Indien dit de eerste keer is dat een gebruiker op de GUI inlogt sinds de laatste start van de server, moet de gebruiker enkele minuten wachten.
4. Klik met de muiscursor op 'Security events'.
5. Er verschijnt een weergave van grafieken en absolute aantallen van metric met betrekking tot security events.

Eis 2: 1. Open PuTTY op een werkdag. Een werkdag is gedefiniëerd als een dag die niet op zaterdag of zondag valt (bij Level Level), begint om 8:00 u en eindigt om 17:30 u, en niet een feestdag betreft[7].
2. Klik op het veld om een hostname/IPv4-adres in te voeren.
3. Typ hier het publieke IPv4-adres van de Wazuh-server in.
4. Klik op 'Open'.
5. Voer de gebruikersnaam 'test' in en wachtwoord 'test'.
6. Voer stap 5 3 keer uit.
7. Nadat de sessie beëindigd wordt, voer stap 1 t/m 6 éénmaal opnieuw uit.
8. In het daarvoor bestemde Slack-kanaal staat nu een melding met betrekking tot de reeks SSH-inlogpogingen.

Eis 3: 1. Voer testplan 'Eis 2' uit.
2. Tussen stap 4 en 5 verschijnt er een foutmelding: 'Network error: Software caused connection abort'.

Eis 4: 1. Verbindt als root met SSH naar de server waar de Wazuh-agent op draait (niet de manager.
2. Type het volgende in: 'iptables -A OUTPUT -p tcp --dport 1514 -j DROP' en druk op enter.
3. Wacht één minuut.
3. Surf met een webbrowser naar het publieke IPv4-adres van de Wazuh-manager en log in met 'admin' als gebruikersnaam en 'admin' als wachtwoord.
4. Op de pagina die nu verschijnt is onder andere het aantal agents weergegeven, het aantal verbonden agents, en het aantal niet-verbonden ('disconnected') agents. Eén agent zal 'disconnected' zijn.

Eis 5. 1. Open een webbrowser en surf naar de Wazuh-webinterface.
2. Druk op 'Vulnerability Detection'.
3. Er verschijnt een pagina met gevonden kwetsbaarheden o.b.v. CVE's. Deze pagina kan leeg of gevuld zijn.

[7] Zie https://www.rijksoverheid.nl/onderwerpen/arbeidsovereenkomst-en-cao/vraag-en-antwoord/officiele-feestdagen voor data van feestdagen

De niet-functionele eisen zijn niet middels een testplan te testen. Toch is implementatie van deze eisen belangrijk. Voor eis 6 wordt gebruik gemaakt van toegangsbeveiliging zodat alleen mensen waarvoor het strikt noodzakelijk is de logbestanden kunnen benaderen. Aanbevelingen die daartoe gevolgd kunnen worden staan in hoofdstuk 10.

C. Plan van Aanpak

DE HAAGSE HOGESCHOOL

LEVEL LEVEL
Full Service Digital Agency

Het realiseren van een security incident & event management-systeem voor Level Level

Plan van Aanpak

Wouter Honselaar
De Haagse Hogeschool
Ten behoeve van Level Level B.V.
22 maart 2021
Versie 1.0

Inhoudsopgave

1. Inleiding

In dit hoofdstuk zal besproken worden wat de aanleiding is voor deze opdracht. Verder zal de accordering en bijstelling ook besproken worden.

Aanleiding

De aanleiding is dat de organisatie bezig is met het ontwikkelen en opzetten van een nieuwe generatie van haar WordPress-platform die op het gebied van security een flinke stap beter moet zijn dan de vorige generatie om kwalitatief hoogwaardige diensten te kunnen blijven leveren. Security-incidenten worden nu door één persoon afgehandeld, en er is geen overzicht van security-incidenten, want de organisatie gebruikt nu geen SIEM. SIEM staat voor Security Information & Event Management en is de naam voor een systeem waarin verdachte activiteit real-time bijgehouden wordt, en waarbij al dan niet geautomatiseerd actie kan worden ondernomen. Dit is een versimpelde definitie. Daarnaast is er één persoon die regelmatig scans en tests ten behoeve van de security uitvoert op het netwerk. Er wordt nu gebruikgemaakt van een eigen hosting setup. Deze hebben ze zelf ontwikkeld en wordt ook zelf onderhouden. Er wordt voornamelijk gewerkt met cloud hosters voor het verzorgen van Virtual Private Servers. Momenteel is de organisatie bezig met het ontwikkelen van een geheel nieuwe architectuur om de websites te hosten. De uitgangspunten hierbij zijn dat er betere schaalbaarheid en betere veiligheid mogelijk wordt.

Accordering en bijstelling

Het Plan van Aanpak zal worden bestudeerd en goedgekeurd door de afstudeerbegeleider van Level Level en de afstudeerbegeleider van de opleiding. Na aanpassing van dit plan wordt het opnieuw ter goedkeuring aangeboden aan hen.

Toelichting op de opbouw van het plan

In hoofdstuk 2 zal in detail beschreven worden hoe de opdracht uitgevoerd zal worden, en wat de opdracht precies inhoudt.

Na bespreking van de opdracht zullen in hoofdstuk 3 de eisen en randvoorwaarden van de opdracht besproken worden.

In hoofdstuk 4 zal worden toegelicht welke onderzoeksmethoden gebruikt zijn gedurende deze opdracht, met daarbij een fasering die in deze opdracht gehanteerd gaat worden.

Hoofdstuk 5 beschrijft de beheersing tijdens deze opdracht. Hierin wordt besproken welke producten opgeleverd worden, op welke manier er gecommuniceerd wordt tussen de student en de opdrachtgever, en wat de kwaliteit van de deliverables dient te zijn.

Het laatste hoofdstuk beschrijft de planning. Dit zal bestaan uit deadlines van deliverables, start- en einddata van fasen en andere belangrijke data.

2. Opdracht

In dit hoofdstuk wordt beschreven wat de context van de opdracht is en wordt de probleem- en doelstelling gegeven. Ook worden de op te leveren producten beschreven.

Context

Level Level is een commercieel full-service WordPress bureau. Ze maken websites die werken met het contentmanagementsysteem WordPress. Een contentmanagementsysteem (CMS) is een softwarepakket om de inhoud van websites te beheren. Level Level verzorgt zowel de hosting van de websites, het beheer van de WordPress-installaties als de manier waarop de content aangeboden wordt. Er werken ruim dertig mensen. Het kantoor is gevestigd in Rotterdam. Het gros van de klanten zijn Nederlandse organisaties, maar Level Level opereert wereldwijd. Voorbeelden van klanten zijn Cordaid en ABN AMRO.

De BIV-factoren (beschikbaarheid, integriteit en vertrouwelijkheid) spelen een belangrijke rol bij de veiligheid van de websites. Omdat Level Level ook de hosting verzorgt, is cyber security een belangrijk aspect. Level Level heeft geen servers in eigendom of colocatie, maar ze maken gebruik van VPS'en. Servers in colocatie zijn servers in eigendom (of gehuurd) die staan in een datacenter, waarbij het datacenter door meerdere organisaties gebruikt wordt. Omdat Level Level geen colocatie-servers maar VPS'en heeft is het niet verantwoordelijk voor de hardware. VPS staat voor Virtual Private Server. Het is een gevirtualiseerde server die particulieren en bedrijven kunnen huren. Meerdere klanten maken dan gebruik van dezelfde fysieke server. Hierbij is sprake van volledige isolatie tussen virtuele servers van klanten en mag elke klant een gedeelte van de beschikbare capaciteit gebruiken. Beheer van de hardware wordt gedaan door de partij die de VPS'en verhuurt, beheer van software wordt gedaan door de partij die de VPS'en huurt. Level Level heeft momenteel een nieuwe hosting setup in ontwikkeling. De organisatie heeft als doel gesteld dat deze nieuwe setup beter moet zijn dan de vorige generatie, vooral op het gebied van schaalbaarheid.

Level Level heeft nog geen SOC, waardoor een SIEM mogelijk meer geautomatiseerd moet kunnen werken dan in een organisatie met een SOC. SOC staat voor security operations center en is een afdeling die alle cyber security-gerelateerde zaken constant monitort, en zo nodig ingrijpt bij incidenten zoals succesvolle aanvallen. Denk bijvoorbeeld aan het tijdelijk blokkeren van verkeer vanaf een bepaald IP-adres dat een hackpoging doet.

Het detecteren van beveiligingsincidenten speelt een belangrijke rol bij webhosting, omdat er schade toegebracht kan worden aan (klanten van) Level Level als een eventuele hack laat ontdekt wordt. Nu worden beveiligingsincidenten door één persoon opgemerkt en afgehandeld. Dit is een single-point-of-failure: als de betreffende persoon niet in staat is snel een incident af te handelen, kan dit bijvoorbeeld leiden tot websites die niet beschikbaar zijn of data die lekt. Level Level is een nieuwe hosting setup aan het ontwikkelen waarbij hoofdzakelijk behoefte is aan meer mogelijkheden met betrekking tot schaalbaarheid. Hiermee blijven websites die binnen korte tijd veel meer bezocht worden snel doordat er meer servercapaciteit ingezet kunnen worden (opschalen), en omgekeerd (afschalen). Hiermee wordt de beschikbare servercapaciteit efficiënter ingezet. Daarnaast wordt er

gebruik gemaakt van meerdere datacenters, wat de beschikbaarheid ten goede komt. De websites van klanten blijven werken als er een storing optreedt binnen een datacenter, doordat ze dan vanuit een ander datacenter gehost worden.

Probleemstelling

De probleemstelling is dat de veiligheid van de huidige hosting setup niet conform de standaards van Level Level is, de wens is om verhoudingsgewijs meer proactief bezig te zijn dan reactief. De gewenste standaard van Level Level is sinds kort beschreven: de ISO27001 wordt zoveel mogelijk opgevolgd, alsmede de BIO (Baseline Informatiebeveiliging Overheid) en de OWASP-richtlijnen. Nu worden deze richtlijnen wel gevolgd, maar er is nog een ISO27001-certificering. Een SIEM is een middel waarmee deze standaards wel behaald kunnen worden door de BIV-factoren zodanig te bewaken dat ze beter zijn dan in de huidige situatie. De veiligheid in brede zin kan daardoor beter gewaarborgd worden als er een SIEM wordt gebruikt. Een SIEM wordt vaak gebruikt in organisaties met een SOC. Vrijwel alleen grote organisaties hebben een SOC. Er bestaan ook kleinere organisaties met SOC, die maken vaak gebruik van diensten van een bedrijf dat een SOC als dienst aanbiedt. Level Level heeft geen SOC. Een SIEM draagt hieraan bij door alle verdachte activiteit op een overzichtelijke manier te presenteren aan de SOC-medewerkers.

Doelstelling

De doelstelling van de opdracht is het specificeren, ontwerpen en realiseren van een SIEM (met mogelijkheid tot het geven van meldingen aan beheerders) als onderdeel van het nieuwe WordPress-platform van Level Level voor het opsporen van en reageren op bedreigingen en kwetsbaarheden. Om de beschikbaarheid en integriteit van WordPress-websites van de klanten beter dan in de huidige situatie te garanderen, is het belangrijk om eventuele kwetsbaarheden en aanvalspogingen real-time in kaart te kunnen brengen, zodat hier, al dan niet geautomatiseerd, op geacteerd kan worden. Na afloop van de opdracht zal er een programma opgeleverd worden dat beheerders waarschuwt bij een gevonden kwetsbaarheid of aanval. Dit kan een bestaand systeem zijn waarbij de motivatie van de keuze is beschreven in het eindverslag, of een geheel of gedeeltelijk nieuw systeem dat op maat ontwikkeld wordt.

Onderzoeksvraag

Tijdens de oriëntatiefase wordt gebruikgemaakt van deskresearch op het world wide web met de centrale vraag ‘Welke SIEM-oplossing kan het beste gebruikt worden voor optimale beveiliging van Level Level?’. Binnen de organisatie wordt voornamelijk met open source software gewerkt, maar het is geen eis dat ook de SIEM-oplossing open source is.

Producten

Als eindproduct zal een eindverslag worden opgeleverd bestaande uit een motivatie van het gekozen SIEM-systeem en hoe deze optimaal gebruikt kan worden, gebaseerd op de onderzoeksbevindingen.

Daarnaast zal een SIEM-systeem opgeleverd worden, welke is geïmplementeerd bij Level Level, minimaal in de vorm van een Proof of Concept-opstelling. Gedurende de fasen van de opdracht deze PoC-opstelling gebruikt ter ondersteuning van het onderzoek.

3. Programma van Eisen

In dit hoofdstuk zijn alle eisen van de op te leveren producten, wensen van de opdrachtgever en randvoorwaarden beschreven waar de op te leveren producten van deze opdracht aan moeten voldoen. Door de wensen en afbakeningen aan het begin van de opdracht te bepalen zal er een efficiënter werkproces gecreëerd worden.

Randvoorwaarden

- Aan het einde van de opdracht zal ten minste een werkende Proof of Concept-opstelling in een productieomgeving opgeleverd worden.
- Het eindproduct zal gebruikt gaan worden door de hosting-afdeling, niet door klanten.
- Het eindproduct mag een combinatie van maatwerk en bestaande producten zijn, mag volledig maatwerk zijn of volledig een bestaand product.

Eisen

De onderstaande eisen zijn in de oriëntatiefase opgesteld aan de hand van deskresearch, communicatie met de opdrachtgever en . Ze worden gespecificeerd en uitgebreidt in de analysefase.

Functionele eisen

1. Informatie over incidenten is te raadplegen via een centrale GUI. De belangrijkste metrics zijn zichtbaar in de GUI. *Welke metrics dit zijn wordt verder gespecificeerd in de analysefase.*
2. Bij elke aanval wordt een escalatiematrix gebruikt die de beheerafdeling op de hoogte stelt. Hiermee worden beheerders buiten het dashboard om op de hoogte gesteld van een eventuele aanval. *De escalatiematrix werkt als volgt: gedurende kantooruren wordt er een bericht geplaatst in een speciaal Slack-kanaal, waarmee de beheerders een melding krijgen op hun computer. Buiten kantooruren gaat de melding naar telefoon en Slack van beheerders. Wordt hier niet op gereageerd, dan wordt er een melding gestuurd naar de telefoons van het managementteam.*
3. Indien er sprake is van een duidelijke hackpoging wordt het gelijk geblokkeerd. *Wat onder ‘duidelijke hackpoging’ valt, wordt in de analysefase verder gespecificeerd, en kan aan de hand van bevindingen in verdere fasen verder gespecificeerd worden.* ***In eerste instantie is dit geen eis.***
4. Data kan opgehaald worden bij specifieke clients met en zonder agents, en ook direct op de netwerkinterface(s) van de SIEM. *Om welke data dit gaat, wordt in de analysefase uitgezocht. Gedurende de volgende fasen wordt dit verder gespecificeerd.*
5. Er kunnen automatische routine-tests uitgevoerd worden, zoals portscans en vulnerability scans. *Dit is handig omdat deze momenteel handmatig uitgevoerd worden, wat tijd kost, terwijl ze goed te automatiseren zijn. Routine tests zijn exact gedefinieerd, gestandaardiseerd en makkelijk in code uit te voeren. Deze worden met dezelfde tijdsintervallen uitgevoerd.*

Niet-functionele eisen

1. Logs kunnen niet aangepast worden door gebruikers.
2. Voldoet aan geldende wet- en regelgeving.
3. De performance is zoals verwacht mag worden op basis van de handelingen die verricht worden.
4. Heeft een uptime van 99,8%.
5. Vertraagt het netwerk of de diensten die aan klanten geleverd worden niet.
6. Gebruiksvriendelijk genoeg voor technische beheerders.
7. De scope bestaat uit alle servers die direct verbinding met internet hebben, waaronder servers die persoonsgegevens verwerken.

4. Aanpak & fasering

In dit hoofdstuk bevinden zich de te gebruiken aanpak en methode.

Fasering

De opdracht zal gefaseerd worden aangepakt. Per fase wordt de beste werkwijze bepaald. Dit zijn de fasen: de oriëntatiefase, analysefase, ontwerpfase en implementatiefase. Gedurende fasen kan er altijd iets uit een vorige fase verbeterd worden, mocht dit nodig zijn. Elke fase bestaat uit iteraties die een week duren. Aan het begin van elke week selecteer ik een aantal taken die nog niet uitgevoerd zijn, en ga ik daaraan werken. Dit zijn taken die behoren tot de huidige fase van de opdracht, of eventueel van een vorige fase/iteratie indien de betreffende taken nog niet af zijn.

Tijdens de oriëntatiefase wordt gebruikgemaakt van deskresearch op het world wide web met de centrale vraag ‘Welke SIEM-oplossing kan het beste gebruikt worden voor optimale beveiliging van webhosters?’. Daarnaast stel ik een Plan van Aanpak en Programma van Eisen op, op basis van de door Level Level gebruikte omgeving met bijbehorende wensen. De huidige technische situatie speelt hier ook uiteraard een rol. Gedurende uitvoering van de fasen worden deelvragen opgesteld.

Er zal onderzoek worden gedaan naar welke SIEM-oplossing het beste geschikt is voor Level Level, en hoe deze optimaal ingezet kan worden. Hierbij wordt (met marktonderzoek) ook gekeken of een bestaande SIEM-oplossing (zoals AlienVault OSSIM) binnen de organisatie past, of dat een maatwerkoplossing beter is. Naast deskresearch wordt ook gebruikgemaakt van veldonderzoek en laboratoriumonderzoek. Veldonderzoek heeft het doel om de Proof of Concept-opstelling te testen en zo nodig bij te stellen zodat deze geschikt wordt gemaakt voor een productieomgeving. Laboratoriumonderzoek heeft het doel om de PoC-opstelling uitvoerig te testen aan de hand van bekende aanvallen en kwetsbaarheden binnen een testomgeving. Dit is onderdeel van de analyse- en ontwerpfase. Gedurende deze fasen wordt gekozen of een bestaand SIEM-systeem gebruikt zal gaan worden, of dat een maatwerkoplossing beter is. Naar aanleiding van de bevindingen wordt een ontwerp opgesteld.

Hierna wordt een testomgeving ingericht op DigitalOcean. Deze zal worden gebruikt voor de PoC-opstelling, beschreven in de volgende alinea.
Na het uitvoeren van deskresearch wordt de PoC-opstelling gemaakt. Daarna wordt de PoC-opstelling door middel van laboratoriumonderzoek geoptimaliseerd en uitvoerig getest binnen een kunstmatige omgeving. In deze omgeving zullen aanvallen uitgevoerd worden waarop de SIEM zou moeten reageren. Denk hierbij aan aanvallen zoals denial of service-attacks en pogingen tot privilege escalation op VPS-servers waarop de WordPress-websites worden gehost.

Ten slotte wordt de PoC-opstelling binnen de productieomgeving geplaatst en zo nodig verder geoptimaliseerd. Gezien de beperkte tijd van de opdracht is het implementeren van een volledig en optimaal werkende SIEM geen doel. Het inzichtelijk maken middels analyse van welke data door de SIEM verwerkt zal moeten worden, is wel een doel. De

PoC-opstellingen gedurende de onderzoeksfasen worden gebruikt als middel voor het doel. Bevindingen van deze drie onderzoeksmethoden worden beschreven in het eindverslag. Dit is onderdeel van de implementatiefase.

Methode

Er is gekozen om gebruik te maken van Kanban als methode. Kanban is verkozen boven andere methoden die als iteratieve- of watervalmethode werken, omdat met Kanban taken incrementeel toegevoegd kunnen worden en het daardoor vrijheid creëert. Scrum werkt met vaste cycli en een watervalmethode ervoor zorgt dat er niet teruggegaan kan worden naar vorige fases. Kanban kent deze genoemde beperkingen niet. Er is daarom voor Kanban gekozen in combinatie met iteraties die een week duren, zodat goed kan worden bijgehouden of ik op schema loop.

Om Kanban te gebruiken wordt er gewerkt met een digitaal Kanban-bord op Asana. Alle taken met betrekking tot deze opdracht, zowel inhoudelijk als organisatorisch, worden op dit Kanban-bord geplaatst zodat alle taken op een overzichtelijke wijze inzichtelijk zijn. Kanban is een takenbord die verdeeld wordt in meerdere lijsten, over het algemeen: 'Taken', 'Bezig', ‘Feedback’ en 'Klaar', maar het is ook mogelijk om meerdere lijsten te gebruiken om een verdeling te maken die beter bij het doel past.. Er wordt ook gebruik gemaakt van een WIP (Work In Progress)-limiet, wat inhoudt dat er een maximum aantal taken is welke tegelijkertijd in de 'Bezig'-lijst(en) mag staan, om te voorkomen dat men aan te veel taken tegelijk wil werken waardoor de productiviteit negatief beïnvloed kan worden.

Er is bewust gekozen voor Kanban in combinatie met deze fasering en iteraties. Een watervalmethode kent namelijk beperkingen op het opnieuw uitvoeren van taken in een eerdere fase. Als er een eis bij komt, moet vrijwel elke fase opnieuw doorlopen worden. Kanban kent geen fasering en dus ook deze beperkingen niet. Om toch gestructureerd te kunnen werken, met de vrijheid van Kanban (alle taken kunnen door elkaar lopen, mits ze niet afhankelijk van elkaar zijn), maar zonder de beperkingen van de watervalmethode, is gekozen om gebruik te maken van een fasering. Per fase zullen taken opgesteld worden voor op het Kanban-bord. Gedurende volgende fasen kan altijd verder gewerkt worden aan een taak uit een voorgaande fase.

Concreet betekent dit dat de opdracht verdeeld is in fasen. Deze fasen zijn verdeeld in iteraties welke één week duren. Deze iteraties bestaan uit een set Kanban-taken. Op een Kanban-bord worden alle taken bijgehouden. Gedurende een volgende iteratie kan altijd gewerkt worden aan een taak uit een eerdere iteratie, mocht dan nodig zijn. Door het op te delen in iteraties per fase, is het duidelijk of er sneller gewerkt moet worden om zoveel mogelijk taken tegen het einde van de iteratie af te hebben.

5. Beheersing

In dit hoofdstuk wordt beschreven hoe de opdracht georganiseerd wordt. Ook het aantal te besteden uren wordt beschreven. Daarnaast wordt ook beschreven hoe een goede kwaliteit van de deliverables gegarandeerd kan worden.

Organisatie

Contactgegevens van de bij deze opdracht betrokken personen staan in onderstaande tabel.

Naam	**Rol**	**E-mailadres**
Hans van der Burg	Begeleidend examinator opleiding	hvdburgthu@gmail.com
Bernard Zijlstra	Afstudeerbegeleider en opdrachtgever bedrijf	bernard@level-level.com

Tabel 1: contactgegevens betrokkenen

De contactgegevens van de student (uitvoerder van opdracht) staan in onderstaande tabel.

Naam	**Rol**	**E-mailadres**
Wouter Honselaar	Student (uitvoerder van opdracht)	W.D.Honselaar@student.hhs.nl

Tabel 2: contactgegevens student

Tijdsbesteding

Er zal 36 uur per week besteed worden aan de opdracht.

Kwaliteit van de producten

Alle op te leveren documenten zullen grondig door de student en mogelijk ook door de begeleidend examinator en opdrachtgever gecontroleerd worden. Gedurende het werken aan documentatie, zullen geschreven stukken regelmatig nagelezen en verbeterd worden. Alle stappen gedurende de technische fasen (analysefase, ontwerpfase en implementatiefase) zullen nauwkeurig gecontroleerd worden om te voorkomen dat er fouten gemaakt worden.

Communicatie

Communicatie tussen de student en het bedrijf gebeurt middels Slack. Communicatie tussen de student en de opleiding vindt plaats via e-mail. Minstens eenmaal per week stuurt de student een update over de stand van zaken met betrekking tot de opdracht naar de afstudeerbegeleider van het bedrijf.

6. Planning

In dit hoofdstuk wordt de planning met betrekking tot deadlines gegeven en hoe er te werk gegaan wordt voor het halen van deliverables.

Deadlines

Door de opleiding zijn deadlines opgesteld. Deze staan in onderstaande tabel.

Datum	**Taak**
4 juni 2021	Inleveren afstudeerverslag
rond 1-12 maart 2021	Bedrijfsbezoek
Uiterlijk 7 mei 2021	Tussentijds assessment

Tabel 3: Deadlines

Roadmap

In onderstaande tabel is een gedetailleerde planning te zien, met de taken die per week uitgevoerd worden. Er wordt gestreefd om zoveel mogelijk deze roadmap te volgen en er zo min mogelijk van af te wijken, om te voorkomen dat de kwaliteit van de eindproducten laag is vanwege tijdgebrek.

Week	**Taak**
Week 1 (oriëntatiefase)	● Werken aan Plan van Aanpak. ● Zoeken naar mogelijke geschikte oplossingen op het www en hiermee experimenteren.
Week 2 (oriëntatiefase)	● Conceptversie Plan van Aanpak afronden. ● Definitieve versie Plan van Aanpak afronden. ● Verslag structuur geven.

Week 3 - 5 (analysefase)	• Is een bestaande oplossing of maatwerk beter? (door middel van marktonderzoek en analyse PvE, is deskresearch) • Naar aanleiding van bevindingen conclusie trekken (zowel functionele, niet-functionele eisen en randvoorwaarden spelen hierbij een rol). • Experimenteren met PoC's (zowel bestaand als maatwerk). • Architectuur nieuwe hosting setup inzichtelijk maken. • Bevindingen oriëntatie- en analysefase in verslag schrijven.
Week 6 - 9 (ontwerpfase)	• Middels bevindingen uit de eerste twee fasen een ontwerp maken voor een te gebruiken systeem. Hierbij worden afwegingen gemaakt en alle keuzes worden onderbouwd. De uiteindelijke architectuur zoals hij geïmplementeerd gaat worden wordt hier ontworpen. Hier wordt uiteraard rekening gehouden met alle eisen. • Ontwerp aan verslag toevoegen.
Week 10 - 14 (implementatiefase)	• Aan de hand van de vorige fase wordt alle infrastructuur geregeld om implementatie mogelijk te maken. Hierna worden alles ingericht zodat er een PoC-opstelling ontstaat welke hetzelfde is als een productieomgeving volgens de nieuwe hosting setup. • Ook verslag schrijven.
Week 15 - 17 (uitloop)	• Verdergaan met implementeren als er nog problemen zijn. • Verslag helemaal afmaken.

Tabel 3: Roadmap

D. Afstudeerplan

Afstudeerblok	2021-1.1
Startdatum uitvoering afstudeeropdracht	8 februari 2021
Inleverdatum afstudeerdossier volgens jaarrooster	4 juni 2021

Studentnummer	17053994
Achternaam	Honselaar
Voorletters	W.D.
Roepnaam	Wouter
Adres	Marnixlaan 36
Postcode	2692 DS
Woonplaats	’s-Gravenzande
Telefoonnummer	0174-421404
Mobiel nummer	06-21837216
Privé emailadres	wouterhonselaar@outlook.com

Differentiatie	NSE
Afstudeerprogramma	CST
Domein*	CST
Locatie	Delft
Variant	Voltijd

**geef hier aan binnen welk domein jouw opdracht valt, m.a.w. wat voor type opdracht het is*

Naam studieloopbaanbegeleider	Adri Pronk
Naam begeleidend examinator	Hans van der Burg
Naam expert examinator	Marinus Maris

Bedrijf

Naam	Level Level
Afdeling	n.v.t.
Bezoekadres	Willem Buytewechstraat 42
Postcode	3024 BN
Plaats	Rotterdam
Telefoonnummer	010-8429259
URL	https://level-level.com

Opdrachtgever

Achternaam	Zijlstra
Voorletters / Voornaam	Bernard
Titel / Opleiding	BSc / IT-opleiding aan Hogeschool Utrecht
Functie	Chief Technical Officer Lead operations/hosting
Afdeling	Operations/hosting
Telefoonnummer (werk)	010-8429259
Emailadres (werk)	bernard@level-level.com
is de opdrachtgever ook de bedrijfsmentor?	Ja

Titel afstudeeropdracht

Het realiseren van een security incident & event management-systeem voor Level Level

Opdrachtomschrijving

1. Bedrijf & Probleemdomein

Level Level is een commercieel full-service WordPress bureau. Ze maken websites die werken met het contentmanagementsysteem WordPress. Een contentmanagementsysteem (CMS) is een softwarepakket om de inhoud van websites te beheren. Level Level verzorgt zowel de hosting van de websites, het beheer van de WordPress-installaties als de manier waarop de content aangeboden wordt. Er werken ruim dertig mensen, aldus de website. Het kantoor is gevestigd in Rotterdam. Het gros van de klanten zijn Nederlandse organisaties, maar Level Level opereert wereldwijd. Voorbeelden van klanten zijn Cordaid en ABN AMRO.

De BIV-factoren (beschikbaarheid, integriteit en vertrouwelijkheid) spelen een belangrijke rol bij de veiligheid van de websites. Omdat Level Level ook de hosting verzorgt, is cyber security een belangrijk aspect. Level Level heeft geen servers in eigendom of colocatie, maar ze maken gebruik van VPS'en. Colocatie houdt in dat eigen servers staan in een datacenter die gebruikt wordt door meerdere organisaties. VPS staat voor Virtual Private Server. Het is een virtuele server die particulieren en bedrijven kunnen huren. Beheer van de hardware wordt gedaan door de partij die de VPS'en verhuurt, beheer van software wordt gedaan door de partij die de VPS'en huurt. Level Level heeft momenteel een nieuwe hosting setup in ontwikkeling. De organisatie heeft als doel gesteld dat deze nieuwe setup beter moet zijn dan de vorige generatie, vooral op het gebied van schaalbaarheid.

Level Level heeft geen SOC, waardoor een SIEM mogelijk meer geautomatiseerd moet kunnen werken dan in een organisatie met SOC. SOC staat voor security operations center en is een afdeling die alle cyber security-gerelateerde zaken constant monitort, en zo nodig ingrijpt bij incidenten zoals succesvolle aanvallen. Denk bijvoorbeeld aan het tijdelijk blokkeren van verkeer vanaf een bepaald IP-adres dat een hackpoging doet.

Het detecteren van beveiligingsincidenten speelt een belangrijke rol bij webhosting, omdat er schade toegebracht kan worden aan (klanten van) Level Level als een eventuele hack laat ontdekt wordt. Nu worden beveiligingsincidenten door één persoon opgemerkt en afgehandeld. Dit is een single-point-of-failure: als de betreffende persoon niet in staat is snel een incident af te handelen, kan dit bijvoorbeeld leiden tot websites die niet beschikbaar zijn of data die lekt. Level Level is een nieuwe hosting setup aan het ontwikkelen waarbij hoofdzakelijk behoefte is aan meer mogelijkheden met betrekking tot schaalbaarheid. Hiermee blijven websites die binnen korte tijd veel meer bezocht worden snel doordat er meer servers ingezet kunnen worden (opschalen), en omgekeerd (afschalen). Hiermee wordt de beschikbare servercapaciteit efficiënter ingezet. Daarnaast wordt er gebruik gemaakt van meerdere datacenters, wat de beschikbaarheid ten goed komt. De websites van klanten blijven werken als er een storing optreedt binnen een datacenter, doordat ze dan vanuit een ander datacenter gehost worden.

2. Aanleiding & probleemstelling

De aanleiding is dat de organisatie bezig is met het ontwikkelen en opzetten van een nieuwe generatie van zijn WordPress-platform welke op het gebied van security beter moet zijn dan de vorige generatie om kwalitatief hoogwaardige diensten te kunnen blijven leveren. Security-incidenten worden nu door één persoon afgehandeld, en er is geen overzicht van security-incidenten, want de organisatie gebruikt nu geen SIEM. SIEM staat voor Security Incident & Event Management en is de naam voor een systeem waarin verdachte activiteit real-time bijgehouden wordt, en waarbij al dan niet geautomatiseerd actie kan worden ondernomen. Daarnaast is er één persoon die regelmatig scans en tests ten behoeve van de security uitvoert op het netwerk. Er wordt nu gebruikgemaakt van een eigen hosting setup. Deze hebben ze zelf ontwikkeld en wordt ook zelf onderhouden. Er wordt voornamelijk gewerkt met cloud hosters voor het verzorgen van Virtual Private Servers. Momenteel is de organisatie bezig met het ontwikkelen van een geheel nieuwe architectuur om de websites te hosten. Uitgangspunten hierbij zijn dat er mogelijkheden tot schaalbaarheid en betere veiligheid komen.

Voor deze nieuwe setup heeft de organisatie behoefte aan een systeem om op eventuele aanvallen en bekende kwetsbaarheden te scannen. De hostingsetup bestaat uit clusters van servers om de beschikbaarheid zo goed mogelijk te kunnen garanderen. Dit gebeurt middels containers. Containerisatie is een relatief nieuwe vorm van virtualisatie, waarbij in tegenstelling tot conventionele virtualisatie geen besturingssysteem gevirtualiseerd wordt. Alles wat binnen een container uitgevoerd wordt draait in een afgeschermde omgeving, maar wel direct op het besturingssysteem van de server. Bij conventionele virtualisatie wordt er een besturingssysteem gevirtualiseerd, bij containerisatie niet. Containers kunnen geschaald worden op meerdere servers. Daardoor is dit een geschikte manier om webservers (en daarmee websites) goed te kunnen schalen naar gebruik van een website. Omdat het plan voor deze nieuwe opstelling nog in ontwikkeling is en er nog geen aandacht is besteed aan de security ervan, is dit een uitdaging.

De probleemstelling is dat de veiligheid van de huidige hosting setup niet conform de standaards van Level Level is. Een SIEM is een middel waarmee deze standaards wel behaald kunnen worden door de BIV-factoren zodanig te bewaken dat ze beter zijn dan in de huidige situatie. De veiligheid kan daardoor beter gewaarborgd worden als er een SIEM wordt gebruikt. Een SIEM wordt vaak gebruikt in organisaties met een SOC. Vrijwel alleen grote organisaties hebben een SOC. Een SIEM draagt hieraan bij door alle verdachte activiteit op een overzichtelijke manier te presenteren aan de SOC-medewerkers.

3. Doelstelling

De doelstelling van de opdracht is het specificeren, ontwerpen en realiseren van een SIEM als onderdeel van het nieuwe WordPress-platform van Level Level voor het opsporen van bedreigingen en kwetsbaarheden. Om de beschikbaarheid en integriteit van WordPress-websites van de klanten beter dan in de huidige situatie te garanderen, is het belangrijk om eventuele kwetsbaarheden en aanvalspogingen real-time in kaart te kunnen brengen, zodat hier, al dan niet geautomatiseerd, op geacteerd kan worden. Na afloop van de opdracht zal er een programma opgeleverd worden dat beheerders waarschuwt bij een gevonden kwetsbaarheid of aanval. Dit kan een bestaand systeem zijn waarbij de motivatie van de keuze is beschreven in het eindverslag, of een geheel of gedeeltelijk nieuw systeem dat op maat ontwikkeld wordt.

4. Concrete werkzaamheden en producten.

Er zal een Plan van Aanpak worden opgesteld voor dit onderzoek, bestaande uit fasering, methoden en planning. Ik ben van plan de opdracht gefaseerd aan te pakken. Per fase bepaal ik de beste werkwijze. Dit zijn de fasen: de oriëntatiefase, analysefase, ontwerpfase en implementatiefase.

Tijdens de oriëntatiefase wordt gebruikgemaakt van deskresearch op het world wide web met de centrale vraag ‘Welke SIEM-oplossing kan het beste gebruikt worden voor optimale beveiliging van webhosters?’. Daarnaast stel ik een plan van aanpak en programma van eisen op, op basis van de door Level Level gebruikte omgeving met bijbehorende wensen. Gedurende de stage worden deelvragen opgesteld.

Er zal onderzoek worden gedaan naar welke SIEM-oplossing het beste geschikt is voor Level Level, en hoe deze optimaal ingezet kan worden. Hierbij wordt (met marktonderzoek) ook gekeken of een bestaande SIEM-oplossing (zoals Splunk) binnen de organisatie past, of dat een maatwerkoplossing beter is. Naast deskresearch wordt ook gebruikgemaakt van veldonderzoek en laboratoriumonderzoek. Veldonderzoek heeft het doel om de Proof of Concept-opstelling te testen en zo nodig bij te stellen zodat deze geschikt wordt gemaakt voor een productieomgeving. Laboratoriumonderzoek heeft het doel om de PoC-opstelling uitvoerig te testen aan de hand van bekende aanvallen en kwetsbaarheden binnen een testomgeving. Dit is onderdeel van de analyse- en ontwerpfase. Gedurende deze fasen wordt gekozen of een bestaand SIEM-systeem gebruikt zal gaan worden, of dat een maatwerkoplossing beter is. Naar aanleiding van de bevindingen wordt een ontwerp opgesteld.

Hierna wordt een testomgeving ingericht op DigitalOcean. Deze zal worden gebruikt voor de PoC-opstelling, beschreven in de volgende alinea.

Na het uitvoeren van deskresearch wordt de PoC-opstelling gemaakt. Daarna wordt de PoC-opstelling door middel van laboratoriumonderzoek geoptimaliseerd en uitvoerig getest binnen een kunstmatige omgeving. In deze omgeving zullen aanvallen uitgevoerd worden waarop de SIEM zou moeten reageren. Denk hierbij aan aanvallen zoals denial of service-attacks en pogingen tot privilege escalation op VPS-servers waarop de WordPress-websites worden gehost.

Ten slotte wordt de PoC-opstelling binnen de productieomgeving geplaatst en zo nodig verder geoptimaliseerd. Gezien de beperkte tijd van de opdracht is het implementeren van een volledig en optimaal werkende SIEM geen doel. Het opleveren van een werkende PoC-opstelling die met weinig of geen aanpassingen geïmplementeerd zou kunnen worden, is wel een doel. Bevindingen van deze drie onderzoeksmethoden worden beschreven in het eindverslag. Dit is onderdeel van de implementatiefase.

Als eindproduct zal een eindverslag worden opgeleverd bestaande uit een motivatie van het gekozen SIEM-systeem en hoe deze optimaal gebruikt kan worden, gebaseerd op de onderzoeksbevindingen. Daarnaast zal een SIEM-systeem opgeleverd worden, welke is geïmplementeerd bij Level Level.

De globale planning van de fasen is te zien in onderstaande afbeelding. De groene vakken duiden de fasen aan per week, aangeduid met het weeknummer van het afstuderen (1 = eerste week afstuderen, 17 = laatste week afstuderen):

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Oriëntatiefase (o.a. Plan van Aanpak maken)																	
Analysefase																	
Ontwerpfase																	
Implementatiefase																	
Uitloop en werken aan verslag																	

5. Beroepstaken

De beroepstaken waar de opdracht betrekking op heeft zijn:

- A1 – Analyseren probleemdomein & opstellen probleemstelling
  - Het Plan van Aanpak en het uiteindelijke systeem moeten gericht zijn op de probleemstelling welke voortkomt uit analyse van het probleemdomein.
- B2 – Adviseren over inrichting van ICT-gerelateerde oplossingen en processen
  - Het doel van de opdracht is het opleveren van een SIEM-oplossing. Advies geven over inrichting en gebruik hiervan is onderdeel van de opdracht.
- C4 – Ontwerpen technische infrastructuur
  - Het doel van de opdracht is het opleveren van een SIEM-oplossing. Dit is een technisch product dat ontworpen en geïmplementeerd wordt.
- C8 – Ontwerpen stelsel van securitymaatregelen
  - Om de veiligheid van de websites te waarborgen en de SIEM optimaal te laten werken, moeten richtlijnen worden opgesteld om te bepalen hoe de hosting van websites en de werking van de SIEM op een veilige manier kan plaatsvinden.
- Gc – Kritisch, onderzoekend en methodisch werken
  - Omdat het eindproduct zo foutloos mogelijk moet werken, moet kritisch gewerkt worden. Daarnaast moet onderzoekend en methodisch gewerkt worden, omdat er onderzoek gepleegd zal worden naar welke bestaande of niet-bestaande SIEM-oplossing het beste gebruikt kan worden.

6. Concept Bibliografie

Van Houten, P., Spruit, M., & Wolters, K. (2015). *Informatiebeveiliging onder controle (3de editie)*. Amsterdam, Nederland: Pearson Benelux.
Johansen, G. (2020). *Digital Forensics and Incident Response – Second Edition.* Packt Publishing.
Bejtlich, R. (2013). *The Practice of Network Security Monitoring*. No Starch Press.