Onderzoeksvoorstel

Ontwerp van een geautomatiseerd systeem voor security monitoring en incident response in SaaS-omgevingen binnen een DevOps-infrastructuur.

Bachelorproef, 2025-2026

Sem Demoen

E-mail: sem.demoen@student.hogent.be

Samenvatting

Deze bachelorproef onderzoekt hoe een geautomatiseerd systeem voor security monitoring en incident response kan worden ontworpen om securitydata binnen SaaS-omgevingen effectief te centraliseren, analyseren en rapporteren in een DevOps-infrastructuur. Het onderzoek vertrekt vanuit een concrete casus bij Devinity.eu, waarbij huidige uitdagingen zoals verspreide tools, vertraagde incidentdetectie en inconsistente rapportering dagelijks voorkomen. De centrale onderzoeksvraag luidt: "Hoe kan een geautomatiseerd systeem voor security monitoring en incident response worden ontworpen om securitydata binnen SaaS-omgevingen effectief te centraliseren, analyseren en rapporteren in een DevOps-infrastructuur?" Het doel is een proof-of-concept te ontwikkelen dat logs, events en alerts van SaaS-platforms (zoals Microsoft 365, Google Workspace, Okta) en DevOps-omgevingen verzamelt via API's, webhooks en logforwarders, deze gegevens normaliseert en analyseert, om vervolgens automatische rapportages te genereren voor de technische teams en management van het bedrijf. De methodologie omvat een literatuurstudie, analyse van bestaande security monitoring tools, ontwerp en implementatie van een prototype en evaluatie aan de hand van indicatoren zoals detectiesnelheid, foutpositieven en responstijd. Het verwachte resultaat is een werkend proof-of-concept dat operationele efficiëntie verhoogt, sneller incidenten detecteert en consistente, bruikbare rapportages oplevert. De meerwaarde voor de doelgroep bestaat uit een framework voor geautomatiseerde security monitoring binnen SaaS en DevOps, dat organisaties helpt risico's beter te beheersen en hun incident response te optimaliseren. GitHub: https://github.com/SemDemoen/bachproef

Keuzerichting: System & Network Administrator

Inhoudsopgave

5. Hoe beïnvloedt de huidige aanpak van monitoring en rapportering de operationele efficiëntie en het risicomanagement van SaaS-bedrijven?

Oplossingsgericht:

1. Welke architectuur en technologieën zijn het meest geschikt om een geautomatiseerd securitymonitoringsysteem te ontwikkelen dat past binnen een DevOps-omgeving (bijv. Elastic Stack, Wazuh, OpenSearch)?
2. Hoe kunnen correlatie- en analysemethoden (zoals machine learning of rule-based detection) bijdragen aan snellere en accuratere incidentdetectie?
3. Welke data-integratieprocessen (APIs, agents, pipelines) zijn het meest efficiënt om logs en securitydata uit verschillende bronnen te verzamelen en te normaliseren?
4. Hoe kan automatische rapportgeneratie worden geïmplementeerd zodat rapporten zowel technische details als managementinzichten bevatten?
5. Welke prestatie-indicatoren (KPIs) kunnen gebruikt worden om het succes van het systeem te meten, zoals detectiesnelheid, foutpositieven of responstijd?
6. Hoe kan de beveiliging en betrouwbaarheid van het geautomatiseerde systeem zelf worden gegarandeerd binnen de bestaande DevOps-pipeline?

Doelstelling: Het ontwikkelen van een proof-of-concept van een gecentraliseerd systeem dat:

- Securitydata uit SaaS-applicaties en DevOps-omgevingen verzamelt via APIs, webhooks en logforwarders;
- Data normaliseert en analyseert om incidenten sneller en accurater te detecteren;
- Automatische rapportages genereert voor technische teams en management;
- KPIs levert om de effectiviteit van monitoring en incident response te meten.

Het concrete eindresultaat is een werkend prototype dat de voordelen van centralisatie, snelle detectie en consistente rapportage aantoont.

2. Literatuurstudie

De toenemende nood van bedrijven voor complexe netwerkinfrastructuren en cloudgebaseerde diensten heeft geleid tot een sterke groei in het aantal beveiligingstools en monitoringoplossingen. Ondanks deze evolutie blijkt het voor veel organisaties moeilijk om een doordachte en effectieve architectuur voor security monitoring te ontwerpen en te implementeren. Volgens **Obregon_2015<emp** resulteert een minder goed ontworpen architectuur vaak in verlies van zichtbaarheid op netwerkverkeer en beveiligingsevents en verhoogde kosten voor bijkomende hardware en technologie. Dit probleem wordt samengevat door de stelling dat preventie ideaal is, maar detectie noodzakelijk blijft: zonder voldoende detectiemechanismen blijven incidenten onopgemerkt, ongeacht de ingezette preventieve maatregelen (**Obregon_2015**).

Een fundamenteel punt binnen security monitoring is zichtbaarheid. Zoals vaak gesteld wordt: men kan niet beschermen wat men niet kan zien. Om deze zichtbaarheid te vergroten, is netwerksegmentatie heel erg belangerijk binnen een informatiebeveiligingsstrategie. Door het netwerk op te delen in duidelijk afgebakende zones wordt niet alleen de kans verkleind dat een succesvolle aanval zich kan verspreiden, maar wordt ook de analyse van netwerkverkeer vereenvoudigd en gerichter (**Obregon_2015**). Netwerksegmentatie vormt daarmee de basis voor een veilige netwerkarchitectuur en ondersteunt zowel detectie als respons.

Een tweede cruciaal punt van security monitoring is logbeheer. Het **NIST_CSF_2018<empty citation>** benadrukt dat logging en monitoring onmisbaar zijn binnen de functies *Detect* en *Respond*. Een log kan worden gedefinieerd als een registratie van gebeurtenissen binnen een computersysteem of netwerk, waarbij deze gebeurtenissen lokaal worden opgeslagen of doorgestuurd naar een gecentraliseerde logmanagementinfrastructuur voor verdere analyse (**NIST_CSF_2018**). Logs bieden inzicht in wat er zich binnen een organisatie afspeelt en vormen een belangerijke informatiebron voor probleemoplossing, onderzoek en het ondersteunen van doelstellingen. Logmanagement omvat het volledige proces van het genereren, verzamelen, transporteren, opslaan, analyseren en uiteindelijk verwijderen van loggegevens afkomstig uit diverse bronnen (**NIST_CSF_2018**). Door deze processen te structureren en te centraliseren kunnen organisaties beter omgaan met grote hoeveelheden data.

Hoewel logging en monitoring onmisbaar zijn, blijft incidentdetectie één van de meest uitdagende aspecten van incident response. Incident

handlers worden geconfronteerd met onvolledige, tegenstrijdige en verwarrende signalen die geanalyseerd moeten worden om vast te stellen of er daadwerkelijk sprake is van een beveiligingsincident (**NIST_SP800_61**). Dit wordt nog meer ingewikkeld door het feit dat detectiemechanismen zoals intrusion detection systems vaak false positives genereren. Daarnaast zijn gebruikersmeldingen, zoals klachten over onbereikbare servers, niet altijd correct of volledig. Hierdoor is het noodzakelijk om elk signaal kritisch te bekijken voor je begint conclusies te trekken.

De schaal van dit probleem is heel erg groot. Organisaties ontvangen een heel groot aantal indicatoren per dag, afkomstig van detectiesystemen, antivirussoftware en loganalyseplatformen (**NIST_SP800_61**). Het onderscheiden van echte security-incidenten van onschuldige gebeurtenissen is niet gemakkelijk. Zelfs wanneer een indicator correct is, betekent dit niet automatisch dat er sprake is van een incident, aangezien veel symptomen ook veroorzaakt kunnen worden door menselijke fouten of technische storingen. Binnen incident response wordt een onderscheid gemaakt tussen *precursors* en *indicators*. Precursors zijn signalen die wijzen op een mogelijke toekomstige aanval, zoals het detecteren van een kwetsbaarheidsscan in webserverlogs of een publieke aankondiging van een nieuwe exploit (**NIST_SP800_61**). Hoewel precursors relatief zeldzaam zijn, kunnen ze organisaties in staat stellen hun beveiligingshouding proactief aan te passen. Indicators daarentegen zijn signalen dat een incident mogelijk al heeft plaatsgevonden of gaande is, zoals malwaredetecties, mislukte inlogpogingen of afwijkende netwerkstromen. Deze indicatoren komen veel vaker voor en vormen de kern van operationele security monitoring.

Tot slot definieert **NIST_SP800_61<empty citation>** een computer security incident als een daadwerkelijke of dreigende schending van beveiligingsbeleid of gangbare beveiligingspraktijken. Voorbeelden hiervan zijn datalekken, malware-infecties, denial-of-service-aanvallen en ongeautoriseerde toegang tot gevoelige gegevens. De noodzaak van een gestructureerde incident response-capaciteit komt voort uit de frequente en impactvolle aard van dergelijke incidenten. Een systematische aanpak van incident response helpt organisaties om schade te beperken, dienstverlening te herstellen en lessen te trekken die bijdragen aan een betere voorbereiding op toekomstige incidenten.

3. Methodologie

Het onderzoek wordt uitgevoerd in vier fasen:

1. **Probleemanalyse:** Samenwerking met Devinity.eu om huidige monitoringtools, workflows en tekortkomingen in kaart te brengen. Analyse van bestaande logs, incidenten en KPI's.
2. **Architectuurontwerp:** Selectie van technologieën zoals Elastic Stack, Wazuh en OpenSearch voor het verzamelen, normaliseren en analyseren van data. Definiëren van datastromen via API's, webhooks en logforwarders.
3. **Proof-of-concept ontwikkeling:** Implementatie van centrale logging, correlatie- en analysemethoden (rule-based en eenvoudige machine learning), en automatische rapportage. Testen binnen een DevOps-pipeline.
4. **Evaluatie en KPI-meting:** Meten van detectiesnelheid, aantal false positives, responstijd en percentage automatisch afgehandelde incidenten. Analyse van resultaten en aanbevelingen voor optimalisatie.

Tijdsplanning:

- Fase 1: 3 weken - requirements-analyse en dataverzameling.
- Fase 2: 2 weken - architectuurontwerp en selectie van tools.
- Fase 3: 6 weken - implementatie van proof-of-concept.
- Fase 4: 2 weken - evaluatie, metingen en rapportering.

De deliverables per fase zijn:

- Fase 1: Analyseverslag van huidige situatie en probleemstelling.
- Fase 2: Ontwerpspecificatie van de architectuur en datastromen.
- Fase 3: Werkend proof-of-concept systeem.
- Fase 4: Evaluatierapport met KPIs, conclusies en aanbevelingen.

4. Verwacht resultaat, conclusie

Het verwachte resultaat is een proof-of-concept van een gecentraliseerd securitymonitoringsysteem dat:

- Securitydata uit meerdere SaaS- en DevOps-bronnen centraliseert en normaliseert;
- Incidenten sneller en betrouwbaarder detecteert via correlatie en analyse;
- Automatische rapportages genereert voor technische teams en management;
- KPI's levert zoals MTTD, MTTR, aantal false positives en responstijd.

De meerwaarde voor de doelgroep (DevOps- en securityteams bij SaaS-bedrijven) ligt in hogere operationele efficiëntie, betere risico-inschatting en snellere incidentafhandeling. Het onderzoek levert zowel praktische inzichten als een concrete demonstratie van een geautomatiseerd monitoring- en incident response-systeem.