Bacheloroppgave

NTNU
Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for datateknologi og informatikk

Andreas Finni Magnussen
Håkon Håbrekke Skaufel
Ådne Tøftum Svendsrud
Tord Valestrand

Effektivisering av hendelseshåndtering med automatisering

Bacheloroppgave i digital infrastruktur og cybersikkerhet
Veileder: Joakim Klemets
Mai 2023

NTNU
Kunnskap for en bedre verden

Andreas Finni Magnussen
Håkon Håbrekke Skaufel
Ådne Tøftum Svendsrud
Tord Valestrand

Effektivisering av hendelseshåndtering med automatisering

Bacheloroppgave i digital infrastruktur og cybersikkerhet
Veileder: Joakim Klemets
Mai 2023

Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for datateknologi og informatikk

NTNU
Kunnskap for en bedre verden

Abstract

The process of incident management aims to reduce the risk of harm to an organization's assets and services, while also restoring normal operations as quickly as possible after incidents. One key challenge of incident management is the issue of streamlining the process to address the complex threat landscape. We have explored practical solutions for streamlining incident management and recommended approaches for selecting tools and methods to improve efficiency.

The study has explored the issue of streamlining incident management through a qualitative research, consisting of several in-depth interviews with candidates experienced in incident management. Additionally, we have developed a proof of concept consisting of use-cases for enhancing the process. The study has demonstrated that automation can be used to streamline all phases of incident management. The results suggest that automation is not fully leveraged in all phases today and that there are efficiency gains to be made. We have identified a set of assessment criteria that should be evaluated prior to implementing automation tools. Among these, we have uncovered that the most important prerequisite for new technology is prioritizing the development of processes and enhancement of staff knowledge.

Sammendrag

Hendelseshåndtering skal redusere risiko for skade på organisasjoners verdier og tjenester, og gjenopprette normal drift så raskt som mulig ved hendelser. En sentral utfordring ved hendelseshåndtering er effektivisering av prosessen for å imøtekomme dagens trusselbilde. I denne oppgaven ser vi på muligheter for å effektivisere hendelseshåndtering med automatisering. Vi har utforsket praktiske løsninger for effektivisering av hendelseshåndtering, og anbefalt fremgangsmåter for valg av verktøy og metoder for effektivisering.

Rapporten har utforsket problemstillingen gjennom en kvalitativ undersøkelse i form av dybdeintervjuer, med kandidater som har erfaring og kompetanse innenfor hendelseshåndtering. I tillegg har vi utarbeidet et konseptbevis bestående av brukstester for effektivisering. Studien har vist at effektivisering med automatisering er mulig gjennom samtlige faser av hendelseshåndtering. Resultatene tyder på at automatisering ikke alltid utnyttes til det fulle idag, og at det er effektivitetsgevinst å hente i hver fase. Vi har kommet frem til et sett med vurderingsgrunnlag som bør utredes før innføring av automatiseringsverktøy. Blant disse har vi avdekket at den viktigste forutsetningen for innføring av ny teknologi, er at utviklingen av prosesser og menneskelig kompetanse bør komme i første rekke.

Innhold

Figurer

Akronymer

API Application Programming Interface. 19, 53, 66, 86, 91, 182

APT Advanced Persistent Threat. 12, 13

C2 Command and Control. 20, 58, 62, 65

CMDB Configuration Management Database. 78, 98, 103

CVE Common Vulnerabilities and Exposures. 95

DDoS Distributed Denial of Service. 9, 11

DoS Denial of Service. 11

EDR Endpoint Detection & Response. 30

HIDS Host Intrusion Detection Systems. 29

HIPS Host Intrusion Prevention Systems. 29

IaaS Infrastructure-as-a-Service. 47

IDPS Intrusion Detection and Prevention Systems. 29, 31, 47

IDS Intrusion Detection Systems. 29

IEEE Institute of Electrical and Electronics Engineers. 42

IM Incident Manager. 72

IOA Indicators of Attack. 36

IOC Indicators of Compromise. 36

IPS Intrusion Prevention Systems. 29

ITIL Information Technology Infrastructure Library. 42

KIT Konfidensialitet, integritet og tilgjengelighet. 7, 104, 110

KPI Key Performance Indicators. 24

NDR Network Detection & Response. 29

NIDS Network Intrusion Detection Systems. 29

NIPS Network Intrusion Prevention Systems. 29

NIST National Institute of Standards and Technology. 21

NSM Nasjonal Sikkerhetsmyndighet. 11

PaaS Platform-as-a-Service. 47

PPT People, Process & Technology. 10, 94, 102, 103, 110

SaaS Software-as-a-Service. 35, 47

SIEM Security Information and Event Management. 31, 32, 38, 48, 104

SOAR Security Orchestration, Automation and Response. 31, 33, 48, 78, 94, 99, 103, 104, 106, 110

SOC Security Operation Center. 9, 10, 35, 38, 42, 72

TTP Tactics, Techniques and Procedures. 36

XDR Extended Detection & Response. 30, 34, 53, 76, 96, 104

Ordliste

angrepsflate Angrepsflate (eng: *attack surface*) er de punkter på et system, miljø eller komponent som en angriper kan forsøke å utnytte seg av, for å forårsake skade på, eller hente ut data fra et system, miljø eller komponent [1]. Det er alle mulige punkter, eller angrepsvektorer, som en angriper kan utnytte seg av for å få utilsiktet tilgang til et system [2]. 13, 58, 96

angrepsvektor En angrepsvektor (eng: *attack vector*) er selve metoden, eller kombinasjonen av metoder som trusselaktører benytter for å kompromittere eller infiltrere målets nettverk [3]. Eksempler på angrepsvektorer er phishing, utnyttelse av offentlig tjeneste og kompromitterte legitimasjoner. 12, 19, 96

anomalitetsdeteksjon Anomalitetsdeteksjon er en deteksjonsteknikk som baserer seg på at et normalbilde av tjenesten, systemet eller nettverket dannes gjennom analyse av data over tid, og deteksjonen baserer seg på anomaliteter ved tjenesten som faller utenfor normalbilde [4]. 26, 37, 81

antatt datainnbrudd Antatt datainnbrudd (eng: *assume breach*) er et tankesett som antar at en trusselaktør allerede har kompromittert et nettverk eller et system. Ved å følge denne antagelsen kan fokus flyttes bort fra kun preventive tiltak, til å også inkludere tiltak som motvirker en trusselaktørs teknikker og handlinger etter kompromittering. 17

artefakt En artefakt innenfor informasjonssikkerhet er en gjenstand som inneholder bevis i form av tekst, rådata, logger, filer eller andre datatyper, som brukes under analyse og håndtering av en event eller en hendelse [5]. 52, 68, 85, 160, 170

avansert og vedvarende trussel En avansert og vedvarende trussel (eng: *advanced persistant threat*) er en nøye planlagt trussel som vanligvis retter seg mot sensitiv og økonomisk verdifull informasjon, ved å oppnå uautorisert tilgang til nettverk og systemer over lengre perioder [6]. Slike trusler stammer vanligvis fra en avansert og vedvarende trusselaktør. 16, 25, 30

bakdør «En bakdør innenfor IT en vei inn i et system som er åpnet av uvedkommende, og som ikke er kjent av systemets eier» [7]. Bakdører installeres ofte i sammenheng som nyttelast fra annen skadevare, eller installeres av

uvedkommende som allerede har tilgang til systemene. Hensikten med en bakdør er som oftest å gi trusselaktøren vedvarende tilgang til offerets systemer. 19, 31, 62, 65

botnett Et botnett (eng: *botnet*) er «en samling datamaskiner som uten eiernes viten deltar i et større nettverk» [8]. Et botnett kan bestå av alt fra hundrevis til millionvis av infiserte maskiner. En maskin blir som regel en del av botnettet via skadevare. 12, 19

brannmur En brannmur er «programvare eller maskinvare som skal avvise uønsket kommunikasjon til et program, en datamaskin eller et nettverk. Brannmuren hindrer at utenforstående skal kunne nå tjenester eller informasjon som ikke skal være tilgjengelig» [9]. 23, 35, 49

brute force *Et brute force*-angrep benytter rå datakraft for å avdekke gyldige brukernavn/passord til en tjeneste eller et system. Angrepet innebærer at en datamaskin forsøker pålogging ved bruk av en rekke forskjellige kombinasjoner av brukernavn og passord, med mål om å avdekke en gyldig kombinasjon og dermed oppnå tilgang til tjenesten eller systemet. 61

cloud-native Cloud-native teknologier er teknologier og applikasjoner som er utviklet for installasjon og bruk i skybaserte miljøer og plattformer, ofte med omfattende bruk av skalerbare og dynamiske løsninger og teknologier som containerteknologi [10]. 35

containerteknologi Containerteknologi er en teknologi som omfatter utrulling av applikasjoner som *containere*. En container er en virtuell pakke som inneholder alle de nødvendige komponentene og avhengighetene som en applikasjon trenger for å kjøre uavhengig av infrastruktur og annen programvare på en maskin [11]. 19

Cyber Kill Chain Cyber Kill Chain er en modell for levetiden til et cyberangrep fra trusselaktørers perspektiv, delt inn i de syv fasene rekognosering, bevæpning, leveranse, utnyttelse, installering, kommando og kontroll og handlinger på målet [6]. 16, 42, 55, 67, 97

datavirus «Datavirus er innen IT en type skadevare som kopierer seg inn i andre filer på en datamaskin og tilkoblede lagringsenheter» [12]. 11

deteksjonslogikk Deteksjonslogikk, eller deteksjonsregler, er logikk som definerer hva som regnes som legitim, illegitim, normal eller unormal data eller trafikk [4]. Innenfor hendelseshåndtering brukes begrepet hovedsaklig for å beskrive innebygde regler eller modeller i deteksjons- og analyseverktøy, som tolker data og trafikk for å generere eventer og alarmer basert på kjente signaturer eller anomaliteter. 17, 22, 26, 29, 56, 76, 96

domene Et domene er «et administrativt delområde i et datasystem eller datanettverk. På internett brukes domene eller domenenavn om en organisasjons unike navn på internett, for eksempel ibm.com eller telenor.no» [13]. 58, 68, 90, 184

eskalering av rettigheter Eskalering av rettigheter (eng: *privilege escalation*) er en måte for en trusselaktør å utnytte feil i et programs kildekode, operativsystem eller lignende for å oppnå flere rettigheter på systemet enn det en vanlig bruker ellers skal ha tilgang til. 58

event En event er en endring i programvare eller maskinvare som blir notert og logget av systemet [14]. 8, 27, 74, 76

falsk positiv En falsk positiv er en alarm om unormal eller ondsinnet aktivitet som genereres av et deteksjonssystem, når det i realiteten ikke har oppstått en unormal eller ondsinnet situasjon [4]. 23, 27, 37, 41, 81, 97, 101

forløper En forløper (eng: *precursor*) er et tegn på at en hendelse kan oppstå i fremtiden [15]. 22, 24

forsvar-i-dybden Forsvar-i-dybden (eng: *defense-in-depth*) er en informasjonssikkerhetsstrategi som integrerer mennesker, teknologi og operasjonsevner for å etablere forskjellige barrierer på tvers av lag og oppdrag i organisasjonen [16]. 40

frikildeetterretning Frikildeetterretning (eng: *open-source intelligence*) er innsamling og analyse av data fra åpne kilder, inkludert media, Internett, offentlig informasjon m.m. Innenfor informasjonsikkerhet blir dette blant annet brukt av hackere for å planlegge cyberangrep. 18

hash En hash er en kalkulert verdi som man får ved å fôre data gjennom en matematisk algoritme, kalt *hashing*, der utdataen representerer inndataen [16]. 26, 60, 68, 90, 174, 175, 184

indikator En indikator innenfor hendelseshåndtering er informasjon som beskriver eller antyder et datainnbrudd eller cyberangrep [6]. Mange indikatorer bygger opp et cyberangrep, som kan oppdages under hendelsesanalyse og benyttes for å detektere fremtidige cyberangrep. En form for indikator er atomiske indikatorer, som ikke kan brytes ned og som beholder sin verdi uavhengig av kontekst. Eksempler på dette er domener og IP-adresser. 17, 22, 26, 60, 68, 76, 81, 98, 136

IP-adresse «En IP-adresse er en serie med tall som identifiserer en node i et IP-nettverk som for eksempel internett. Hver node i et slikt nettverk har sin egen unike IP-adresse» [17]. 64, 68, 78, 90, 184

kjent trussel Kjente trusler (eng: *common threats*) er trusler som er gjeldende for et bredt spekter av organisasjoner, for eksempel fordi de retter seg mot sårbarheter i offentlig programvare brukt av mange organisasjoner, eller benytter seg av kjente angrepsvektorer. 76, 95

konseptbevis Et konseptbevis (eng: *proof of concept*) er en realisering av et konsept, eller idé, for å vise hvordan dette fungerer i praksis og om det er gjennomførbart i det hele tatt. Gjerne gjort for å budsjettere og finne mulige løsninger på problemer. Alternativt: Et konseptbevis viser en modell, eller en simulering av hvordan et konsept fungerer i praksis. 2, 45

kontekstuelle kilder Kontekstuelle kilder er eventer og andre rådatakilder som må analyseres og settes i kontekst før beslutninger kan gjennomføres på bakgrunn av dem [4]. 27

kunstig intelligens Kunstig intelligens (eng: *artificial intelligence*) handler om «datamaskiners evne til å utføre handlinger, fysisk eller digitalt, basert på tolkning og behandling av strukturerte eller ustrukturerte data, i den hensikt å oppnå et gitt mål. Enkelte systemer for kunstig intelligens kan også tilpasses gjennom analyser og vurderinger av hvordan tidligere handlinger har påvirket omgivelsene» [18]. 3, 15, 25, 37

lateral bevegelse Lateral bevegelse (eng: *lateral movement*) er en angrepsteknikk brukt av trusselaktører, som innebærer å bevege seg på tvers av systemer eller enheter innad i et nettverk for å unngå deteksjon eller oppnå endelige mål. 20, 61, 63

leverandørkjede Ifølge NSM omfatter en leverandørkjede «alle ledd i kjeden av leverandører og underleverandører som leverer eller produserer varer, tjenester eller andre innsatsfaktorer som inngår i en virksomhets leveranse av tjenester» [18]. Dette gjør leverandørkjeder sensitive for cyberangrep. 19

løsepengevirus Et løsepengevirus (eng: *ransomware*) kan defineres som både en type ondsinnet programvare og som en form for cyberangrep utført av ondsinnede aktører som krever løsepenger for å frigjøre kryptert eller stjålet data [19]. 11, 59

man-in-the-middle Et «man-in-the-middle»-angrep (MITM) er en type cyberangrep hvor trusselaktøren befinner seg mellom to kommuniserende parter hvor informasjonen som sendes mellom partene blir fanget opp/endret. Et MITM-angrep kan skje over forskjellige kommunikasjonskanaler, eksempelvis via e-post eller telefon. 62, 67

maskinlæring Maskinlæring (eng: *machine learning*) er en underart av kunstig intelligens som benytter datadrevne modeller for å skape systemer som fremstår som intelligente [20]. Maskinlæringsmodeller programmeres ikke

med forhåndsdefinerte regler som bestemmer hvordan modellen skal løse gitte problem, men lærer isteden kontinuerlig basert data og problemer som modellen blir trent opp på [21]. 15, 26, 37, 81

MITRE ATT&CK MITRE ATT&CK (*Adversarial Tactics, Techniques, and Common Knowledge*) er et rammeverk og handlingsbasert trusselmodell som omhandler taktikker og teknikker brukt av trusselaktører under et antatt datainnbrudd [22]. Rammeverket gir et detaljert overblikk over trusler og mulige angrepsvektorer, for å gjøre det lettere å identifisere og utbedre hull eller mangler i forsvarsmekanismer og infrastruktur. 17, 42, 55, 67, 97

nulldagssårbarhet En nulldagssårbarhet defineres av NSM som «en sårbarhet programvareutvikleren ikke har oppdaget eller laget sikkerhetsoppdatering for. Navnet kommer av at når en nulldagssårbarhet først er oppdaget, har programvareutvikleren hatt null dager på seg til å lukke sårbarheten før noen potensielt utnytter den» [18]. 19, 81

nøkkeltallsindikator Nøkkeltallsindikatorer (eng: *Key Performance Indicators [KPI]*) er nøkkeltall som brukes for å indikere om en organisasjon eller virksomhet er på rett vei til å nå sine overordnede mål. 93

on-prem *On-prem* (fra engelsk *on-premises*), beskriver infrastruktur eller tjenester som befinner seg – eller er installert på – organisasjonens egne fysiske IT-miljøer. Dette er i motsetning til skybasert infrastruktur og skybaserte tjenester, der det befinner seg i et datasenter og er tilgjengelig for sluttbrukere som en skytjeneste. 35

operasjonell teknologi Operasjonell teknologi omfavner et bredt spektrum av programmerbare systemer og enheter som kommuniserer og samhandler med fysiske systemer, slik som industrielle kontrollsystemer, transportsystemer og systemer for fysisk adgangskontroll [23]. Slik teknologi kan detektere eller forårsake endringer i de fysiske systemene gjennom overvåking og kontroll av de programmerbare systemene og enhetene. 32

orm Under informasjonssikkerhet er en orm, eller dataorm, en type skadevare som er i stand til å spre seg selv fra en datamaskin til en annen [24]. 11

phishing Phishing (norsk: *nettfisking*) er en betegnelse for svindel hvor trusselaktør forsøker å få bruker til å selv oppgi sensitive opplysninger som passord eller betalingskortinformasjon [25]. 13, 18, 19, 57, 62, 78

planlagte oppgaver Planlagte oppgaver (eng: *scheduled tasks*) er et begrep som brukes om oppgaver eller programmer som startes ved forhåndsdefinerte tidspunkter; etter gitte tidsintervaller eller ved gitte eventer på maskinen. Eksempler på dette er *Oppgaveplanlegging*-applikasjonen i Windows, eller Cron i Linux. 58

playbook En playbook, som beskrevet i denne rapporten, er en manual som beskriver prosedyrer og arbeidsflyt under håndteringen av en gitt hendelse eller event. 21, 22, 33, 52, 54, 74, 79, 106, 165, 167

prinsippet om minste privilegium Prinsippet om minste privilegium handler om at ethvert program og enhver bruker av et system skal ha så få rettigheter som mulig for å fullføre jobben sin. Prinsippet handler primært om at mulig skade som følge av en feil eller misbruk, det handler også om å minimere interaksjonen mellom privilegerte programmer, slik at utiltenkt eller uønsket bruk av rettigheter ikke skal forekomme [26]. 64

regulært uttrykk Regulære uttrykk (eng: *regular expression*) er et «programmeringsspråk for tekstsøk basert på mønstre. De kan enten benyttes av brukeren i applikasjoner som støtter regulære uttrykk, eller som en del av programkoden til en applikasjon» [27]. 174

rekognosering Under rekognosering (eng: *reconnaissance*) vil en trusselaktør identifisere og velge potensielle mål, og samle inn informasjon som vil bli brukt i senere faser av angrepet [6]. 18

sammensatte trusler Sammensatte trusler kombinerer diplomatiske, informasjonsmessige, militære, økonomiske, finansielle, etterretningsmessige og juridiske virkemidler for å nå strategiske målsettinger [18]. 12

script Et script er et mindre dataprogram som består av en serie instruksjoner i et scriptspråk. Scriptspråk kan variere fra å være komplekse programmeringsspråk til å være et sett med kontrollstrukturer for å sette sammen kommandoer fra underliggende systemer.[28] Script kjøres enten på klient-side eller på server-side. 58, 60, 95

signaturdeteksjon Signaturdeteksjon er en deteksjonsteknikk som baserer seg på at systemet allerede har kunnskap om indikatorer ved hendelsen eller sårbarheten som utnyttes, altså eksisterer det en signatur for eventen eller hendelsen [4]. 26, 37, 96

skadevare Skadevare (eng: *malware*) er en samlebetegnelse for skadelig programvare. Dette er programvare som utfører handlinger på brukerens systemer eller informasjon [29]. 40, 57

skadevareanalyse Skadevareanalyse (eng: *malware analysis*) er en teknikk for analyse av mistenkelige filer, der filer åpnes i isolerte «detonasjonskammer» for å undersøke hvilke handlinger som utføres når filen åpnes eller kjøres («detoneres»), og vurdere hvorvidt den mistenkelige filen er ondsinnet eller ikke [4]. 31, 175

sosial manipulering Sosial manipulering (eng: *social engineering*) innebærer det «å benytte psykologiske virkemidler for å gjøre angrep mot brukere av IT-systemer» [30]. 13, 19

spoofe Å spoofe betyr å forfalske avsender under kommunikasjon [31]. Begrepet benyttes vanligvis i digital kommunikasjon (e-post, SMS eller telefon). 81

sårbarhetsskanning Sårbarhetsskanning (eng: *vulnerability scanning*) er en metode for å undersøke organisasjonens verdier, inkludert servere og endepunkt for kjente sårbarheter, utdatert programvare og manglende sikkerhetskonfigurasjon [4, s. 15]. 36, 95

threat hunting Threat hunting er en proaktiv teknikk for å tildele ressurser for å se etter alle ustrukturerte indikatorer på hendelser i tillegg til rutinemessige deteksjoner og alarmer som behandles hver dag [4, s. 24]. Threat hunting gjennomføres ofte på bakgrunn av trusseletterretning. 32, 36, 40, 41, 68, 92, 98

tjenestenektangrep Et tjenestenektangrep (eng: *denial-of-service attack*) er et angrep som kun rammer tilgjengeligheten til de tjenester, systemer eller infrastrukturkompontener som blir angrepet [18]. 11, 29, 63

tjenestenivåavtale En tjenestenivåavtale (eng: *Service Level Agreement*) er en dokumentert avtale mellom en tjenesteleverandør og en kunde som identifiserer krav til tjenester og forventet servicenivå [32]. 23, 94, 99

trojaner En trojaner, også kalt en trojansk hest, er en form for skadevare skjult som et legitimt program eller fil. En trojaner kan enten være ondsinnet kode gjemt som en del av et misdannet legitimt program eller fil, eller det kan være en type fullstendig skadevare som kun deler navn eller andre kjennetegn med legitime program eller file [33]. 11, 18

trusseletterretning Trusseletterretning (eng: *threat intelligence*) er informasjon om – og indikatorer på – trusler som er samlet inn, analysert, tolket og beriket for å gi kontekst og støtte beslutninger om håndteringen av trusler. 36, 40, 68, 89, 95

tuning Tuning (eller *å tune*) benyttes i denne rapporten som et begrep for filtrering eller synliggjøring av større mengder data. 33, 77, 96

ukjent trussel Ukjente trusler (eng: *uncommon threats*) er trusler som er unike og særegne for en organisasjon, eller et fåtall av organisasjoner, fordi de for eksempel retter seg mot sårbarheter i programvare kun driftet eller benyttet av organisasjonen, eller mot andre unike egenskaper ved en organisasjons infrastruktur, systemer og tjenester. 76, 95

Kapittel 1

Introduksjon

1.1 Bakgrunn

I dag er alle organisasjoner direkte eller indirekte avhengig av digitale tjenester og en digital infrastruktur som underbygger deres kjernevirksomhet. Digitalt sikkerhetsarbeid er en nødvendig del av det daglige virket i alle ledd av organisasjonen, fra sluttbrukeren som benytter seg av den digitale samhandlingsplattformen, til de som har ansvaret for drift og oppsett av den digitale infrastrukturen. Uavhengig av om den digitale infrastrukturen er driftet selv, eller ansvaret er tjenesteutsatt til en leverandør av skytjenester, må enhver organisasjon ta visse sikkerhetshensyn for å beskytte seg mot uønskede hendelser.

Et av de mest sentrale aspektene av det digitale sikkerhetsarbeidet i en moderne organisasjon er hendelseshåndtering. Prosessen for hendelseshåndtering har dype røtter, og det samme har de metodene og teknikkene som har blitt brukt for å sikre organisasjonen mot uønskede hendelser og håndtering av de hendelsene som oppstår. En sentral utfordring er hvorvidt prosessen evner å henge med på utviklingen til trussel- og risikobilde, og hvorvidt metodene og teknikkene kan videreutvikles for å imøtekomme morgendagens utfordringer.

Realiteten til mange sikkerhetsanalytikere er at store deler av arbeidsdagen brukes på repeterende arbeidsoppgaver og reaktiv håndtering av alarmer og hendelser. Analytikere må håndtere store mengder informasjon fra en rekke forskjellige kilder, der kildene ofte ikke snakker sammen, som overlater store deler av tolkningsarbeidet til sikkerhetsanalytikeren. Blant informasjonen og alarmene som sikkerhetsanalytikere må tolke og håndtere, kan det være en stor mengde falske positiver, mindre kritiske alarmer, og rå maskindata og logger som er vanskelig for mennesker å tolke. Dette vil ofte virke tyngende og sinkende i sikkerhetarbeidet, ettersom en hendelse typisk består av flere symptomer som finnes på tvers av systemer og tjenester, hvor korrekt diagnostisering og behandling avhenger av at hele sykdomsbildet kan analyseres effektivt. Det er med andre ord en forutsetning at sikkerhetsanalytikere kan se det store bildet, som kan være svært utfordrende blant den store mengden informasjon som må tolkes.

Det digitale trusselbildet er i stadig endring, og det er generelt manglende

kompetanse innenfor cybersikkerhet i næringslivet [34]. Nye sårbarheter dukker opp og både mindre og større hendelser oppstår i økende antall [35, 36]. De siste årene har flere større hendelser påført millionregninger til norske organisasjoner, og nye sårbarheter kan dukke opp over natten. Grensene mellom informasjonssikkerhet og geopolitisk sikkerhet viskes ut, og det er i dag liten tvil om at digital krigføring kan føre til store konsekvenser for Norge, norske organisasjoner og grunnleggende nasjonale funksjoner[1] [18].

Digitaliseringen blant norske organisasjoner varierer og det gjenspeiler seg i ulike behov og hensyn. Den digitale utviklingen fortsetter i stor hastighet, imens flere organisasjoner fortsatt følger en klassisk modell med ansvar for store deler av egen IT-infrastruktur og drift, har andre flyttet hele infrastrukturen til skyen og overlatt driftsansvaret til en tredjepart. Organisasjoner har med andre ord forskjellige behov, ressurser og forutsetninger for å håndtere hendelser i sin infrastruktur og virksomhet. Det finnes dermed heller ikke én enkelt løsning, eller en beste praksis å følge, for å møte og løse problemet.

1.2 Problemstilling

Hvordan kan hendelseshåndtering effektiviseres med automatisering? Problemstillingen er valgt med bakgrunn i moderne organisasjoners utfordringer med å effektivisere hendelseshåndtering i takt med den raske digitale utviklingen og et trussellandskap i stadig endring. Denne oppgaven skal gjennomgå, forstå og foreslå løsninger for de forskjellige fasene som utgjør hendelseshåndtering. Dette skal underbygges med brukstester og konseptbevis av konkrete scenario og løsninger, som drøftes opp imot erfaringer og utfordringer fra sektoren. Dybdeintervju med fagpersonell skal illustrere hvordan prosessen gjennomføres i praksis, og legge til rette for drøfting av dagens utfordringer og morgendagens løsninger.

1.2.1 Forskningsspørsmål

Forskningsspørsmål 1: Hvilke faser og steg i hendelseshåndtering er egnet for effektivisering gjennom automatisering?

Forskningsspørsmål 2: Hvordan kan en organisasjon gå frem for å velge riktig verktøy og metoder for effektivisering av hendelseshåndtering, som er i tråd med organisasjonens modningsgrad, behov og hensyn?

1.2.2 Avgrensning og omfang

I denne oppgaven skal vi primært diskutere hendelseshåndtering i lys av sikkerhetshendelser. Oppgaven vil redegjøre for sentral teori relevant til oppgavens pro-

[1] Grunnleggende nasjonale funksjoner er tjenester, produksjon og andre former for virksomhet som er av en slik betydning at et helt eller delvis bortfall av funksjonen vil få konsekvenser for statens evne til å ivareta nasjonale sikkerhetsinteresser [18]

blemstilling og forskningsspørsmål, og fokus vil derfor ilegges teori direkte tilknyttet hendelseshåndtering. Flere relevante temaer tilknyttet informasjonssikkerhet og drift av IT-systemer diskuteres der det er nødvendig for å svare på oppgavens problemstilling. Gjennom oppgaven diskuteres effektivisering av hendelseshåndtering ved bruk av automatisering. I tillegg diskuteres noen potensielle bruksområder for kunstig intelligens. Samtidig skal vi ikke diskutere den grunnleggende teorien bak kunstig intelligens som fagfelt, og oppgaven vil heller ikke se på trening av maskinlæringsmodeller eller relaterte problemstillinger. Oppgaven tar for seg bruken av verktøy og tjenester for hendelseshåndtering som benytter automatisering og kunstig intelligens i større eller mindre grad, men vil ikke gå inn på detaljer om hvordan disse verktøyene og tjenestene benytter kunstig intelligens bak kulissene.

Oppgaven tar for seg hendelseshåndtering som et praktisk og teknisk fagfelt. Det vil si at vi setter søkelys på effektiviseringstiltak som innføres gjennom teknologi, verktøy, tjenester og IT-systemer. Oppgaven skal i liten grad ta opp administrative tema relatert til hendelseshåndtering, slik som hvordan et team for hendelseshåndtering kan settes opp eller organiseres. Vi vil heller ikke gå i dybden på temaer som økonomi, regulære forhold eller andre ikke-tekniske temaer, selv om disse vil kunne nevnes i mindre grad i deler av oppgaven. Oppgaven vil hovedsaklig fremme tiltak for effektivisering av de delene av hendelseshåndtering som er egnet for automatisering. Vi vil fremme forslag for effektivisering av de forskjellige fasene av hendelseshåndtering som prosess. Forslagene vil inkludere tiltak og fremme diskusjonspunkt rundt både mennesker, prosess og teknologi.

Gjennom oppgaven vil vi undersøke problemstillingen fra perspektivet til ansatte hos oppgavestiller som arbeider med hendelseshåndtering. Denne undersøkelsen gjennomføres i form av individuelle dybdeintervju med et utvalg ansatte. Resultatene fra undersøkelsen vil derfor ikke representere enhver organisasjons utfordringer, ønsker og behov, men generaliseres for å besvare problemstillingen på et generelt grunnlag. Oppgaven er gjennomført i samarbeid med oppgavestiller, men resultatene og diskusjonspunktene som fremmes i denne oppgaven skal ikke forstås som oppgavestillers meninger.

Konseptbevis av effektiviseringsløsningene konfigureres og settes opp gjennom et testmiljø, med mål om å demonstrere funksjonaliteten og egnetheten til verktøy og teknologi gjennom en serie med brukstester. Oppgaven vil ikke legge vekt på oppsettet av dette testmiljøet. Oppsettet vil derfor ha begrenset fokus på sikkerhet og robusthet slik det vil være forventet av et fullstendig produksjonsmiljø, foruten det som er nødvendig og forventet av testmiljøet, samt oppgavens gjennomføring.

Oppgaven vil utforske forskjellige verktøy og metoder for hendelseshåndtering, og hvordan disse kan benyttes og integreres som en del av prosessen for hendelseshåndtering til en organisasjon. Vi vil ikke foreslå eller utarbeide nye verktøy eller metoder og oppgaven vil heller ikke sammenligne lignende verktøy fra forskjellige aktører opp i mot hverandre. Vi skal ikke foreslå valg av leverandør for verktøy og tjenester. De verktøyene og tjenestene som vi bruker i denne

oppgaven vil kun være for demonstrasjonsformål og skal kunne tolkes som en generell fremvisning av funksjonalitet og muligheter for effektivisering. Vi vil ikke presentere all funksjonalitet i de verktøyene og tjenestene som vi benytter, men kun trekke frem et hensiktsmessig utvalg av funksjonaliteten for å gjennomføre oppgavens mulighetsstudie og for å svare på oppgavens problemstilling.

Oppgaven vil diskutere håndtering av cyberangrep og andre uønskede hendelser gjennom effektivisering av hendelseshåndtering. Tiltak som presenteres skal ikke forstås som en utfyllende eller dekkende liste over tiltak som kan og bør innføres for å beskytte en organisasjon mot forskjellige typer cyberangrep og andre uønskede hendelser. Tiltakene som fremmes er valgt i bakgrunn av de brukstester som demonstreres, med fokus på bruken av verktøy for effektiv hendelseshåndtering. Oppgavens brukstester vil i mange tilfeller anta og forutsette at trusselaktører eller andre parter forårsaker diverse typer hendelser, for eksempel ved å utnytte sikkerhetshull eller andre mangler i en organisasjons beskyttelse mot sikkerhetshendelser, og fokuset blir dermed på tiltak for mitigering og håndtering i lys av disse scenarioer.

1.3 Metodevalg og gjennomføring

I løpet av dette prosjektet skal vi gjennomføre kvalitative dybdeintervju med ansatte hos oppgavestiller. Dybdeintervjuene gjennomføres fysisk i form av semistrukturerte intervju som deretter analyseres før funnene legges frem og drøftes. I tillegg har vi gjennomført en mulighetsstudie som legger frem et konseptbevis, bestående av tre brukstester for å effektivisere hendelseshåndtering. Mulighetsstudien gjennomføres ved bruk av et testmiljø, før funnene presenteres og diskuteres sammen med resultatene fra dybdeintervju.

1.4 Om oppgavestiller

Denne bacheloroppgaven er gjennomført på oppdrag av Norsk helsenett SF. Norsk helsenett er et statlig foretak som legger til rette for trygg samhandling mellom de forskjellige aktørene i helsesektoren. Ved å legge til rette for trygge digitale tjenester, bidrar Norsk helsenett til en mer effektiv helsetjeneste og bedre pasientsikkerhet. Norsk helsenett har bidratt med veiledning under prosjektet og har som oppdragsgiver også motivert og påvirket oppgavens problemstilling og forskningsspørsmål, samt avgrensning og omfang.

1.5 Rapportens sammensetning

Kapittel 1: Introduksjon Introduksjonskapittelet omfatter bakgrunn for prosjektet, problemstilling og forskningsspørsmål, metodevalg og gjennomføring, samt en kort beskrivelse av oppgavestiller og rapportens sammensetning.

Kapittel 2: Teori Oppgavens teorikapittel omfatter en teoretisk utredning av relevant teori innenfor informasjonssikkerhet og hendelseshåndtering som fagfelt. Vi bruker også dette kapittelet til å utforske teoretiske kilder vi finner relevante for bruk til utarbeidelse av metodekapittelet. Teorikapittelet redegjør for definisjoner og sentrale konsepter, livssyklusen til et cyberangrep, metoder for hendelseshåndtering, samt utviklingen og utfordringer til et moderne SOC. Til slutt ser vi på relevant forskning som utforsker liknende tematikk som denne oppgaven.

Kapittel 3: Metode Metodekapittelet tar for seg de to vitenskapelige forskningsmetodene vi benytter gjennom denne oppgaven. Dette omhandler intervjuundersøkelsen, med struktur, utvalg, analyse og gjennomføring. I tillegg omhandler det mulighetsstudien, med hensikt, analyse, oppsett og gjennomgang av studiens forskjellige caser og hvordan brukstestene gjennomføres.

Kapittel 4: Resultat I resultatkapittelet presenteres funnene vi har samlet inn fra oppgavens metodedel. Dette inkluderer resultater fra intervjuundersøkelsen, i tillegg til resultater fra brukstestene vi utredet under mulighetsstudien.

Kapittel 5: Diskusjon I diskusjonskapittelet vil vi besvare oppgavens problemstilling og forskningsspørsmål. Diskusjonen vil basere seg på innsamlede erfaringer, refleksjoner og resultater samlet inn under oppgavens gjennomføring. Diskusjonen skal sammenfatte resultatene fra både dybdeintervju og mulighetsstudie, samt presentert teori, og sette søkelys på den helhetlige sammenhengen mellom disse.

Kapittel 6: Konklusjon I konklusjonen vil vi oppsummere oppgavens sentrale punkter og diskusjonsmomenter.

Kapittel 2

Teori

I dette kapittelet vil vi redegjøre for teori rundt hendelseshåndtering og informasjonssikkerhet. Teorien skal legge fundamentet for videre metodeforskning og diskusjon. I seksjon 2.1 vil vi forklare de sentrale konseptene oppgaven baserer seg på, samt definere viktige terminologier og begreper. I seksjon 2.2 ser vi nærmere på hva et cyberangrep er og hvordan det er bygd opp. Videre vil vi i seksjon 2.3 og 2.4 gå inn på hendelseshåndteringsprosessen i mer detalj, samt utforske metoder, teknikker og verktøy brukt i prosessen. Vi vil i tillegg redegjøre for noen av automatiserings- og maskinlæringsmetodene som brukes i hendelseshåndteringsprosessen i dag. I seksjon 2.5 vil vi skrive om sikkerhetsavdelingene som driver med hendelseshåndtering, deres ansvarsområder og utfordringer. Avslutningsvis vil vi i seksjon 2.6 redegjøre for tidligere relevant forskning om temaet og problemstillingen.

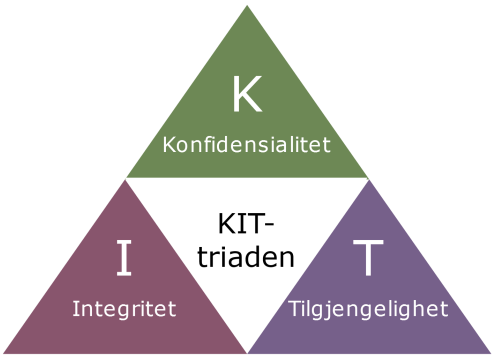
2.1 Definisjoner og sentrale konsepter

For å kunne diskutere effektivisering av hendelseshåndtering vil vi først redegjøre for sentrale konsepter innenfor informasjonsteknologi og hendelseshåndtering. Denne seksjonen trekker frem de viktigste konsepter og definisjoner som er nødvendig for å forstå oppgavens problemstilling og videre drøfting.

2.1.1 KIT - Konfidensialitet, integritet & tilgjengelighet

KIT-triaden regnes som grunnpilaren av informasjonssikkerhet, den er vist i figur 2.1. Det finnes utallige definisjoner på denne modellen. Et generelt fellestrekk for de ulike definisjonene er likevel at de tre delene modellen består av, konfidensialitet, integritet og tilgjengelighet, representerer de grunnleggende aspektene ved informasjonssikkerhet som må ivaretas for å beskytte sensitiv informasjon og systemer. Med informasjonssikkerhet i fokus, handler modellen i stor grad om å sikre systemer fra å bryte med KIT.

En mer overordnet hensikt med KIT-triaden handler om hvordan de tre delene av modellen fungerer sammen. Alle punktene må innfris for å oppnå tilstrekkelig



Figur 2.1: KIT-triaden

sikkerhet. Satt litt på spissen, kan man eksempelvis oppnå veldig god konfidensialitet ved å gjemme viktig informasjon langt inne i et hvelv med mange forskjellige passord og adgangsmekanismer, men tilgjengeligheten til denne informasjonen er langt fra optimal. En viktig del av informasjonssikkerhet handler om å finne kompromisser mellom de tre delene av KIT-triaden.

Konfidensialitet

Informasjon regnes som konfidensiell hvis den ikke er tilgjengelig eller utleveres til uautoriserte personer, enheter eller prosesser [37]. Med andre ord kan man si at informasjon er konfidensiell om den er tilgjengelig for autoriserte personer og utilgjengelig for uautoriserte.

Integritet

Integriteten til informasjon regnes ut i fra hvorvidt den er nøyaktig eller ikke, og hvor komplett den er [37]. Med andre ord betyr dette at informasjonen er korrekt og ikke er endret enten utilsiktet eller av uautoriserte personer.

Tilgjengelighet

Informasjon regnes som tilgjengelig når den kan aksesseres ved behov av autoriserte entiteter [37]. Informasjon har med andre ord ingen betydning med mindre den faktisk er tilgjengelig, noe man ikke må glemme når det kommer til informasjonssikkerhet.

2.1.2 Event og sikkerhetsevent

En event er en endring i en programvare eller maskinvare som blir notert og logget av systemet [14]. Dette gjør det mulig å korrelere loggede endringer med blant annet feil, ytelsesproblemer og sikkerhetsbrudd [38]. ISO/IEC 27035 standarden

definerer en sikkerhetsevent som en endring som indikerer et mulig brudd i informasjonssikkerhet eller svikt i sikkerhetskontroller. En sikkerhetsevent eskaleres ikke nødvendigvis til en sikkerhetsendelse [39].

2.1.3 Hendelse og sikkerhetshendelse

ITIL definerer en hendelse som «et ikke-planlagt avbrudd i en IT-tjeneste, eller en reduksjon i kvaliteten til en IT-tjeneste» [32]. Dette omfatter både driftsrelaterte svikt i konfigurasjonsenheter og forekomster som indikerer sikkerhetsbrudd i tjenesten eller systemet. Det er derfor hensiktsmessig å definere hva en sikkerhetshendelse er, for å tydeliggjøre forskjellen mellom hendelser relatert til sikkerhet, fra hendelser relatert til daglig drift [40].

ISO/IEC 27035 standarden definerer en sikkerhetshendelse som en eller flere sammenhengende sikkerhetseventer som kan skade organisasjonens verdier eller organisasjonens operasjoner [39]. Det bør likevel nevnes at enhver sikkerhetsevent ikke nødvendigvis medfører en sikkerhetshendelse. Det å skille mellom vanlige driftsrelaterte hendelser og sikkerhetshendelser er ikke alltid helt svart og hvitt. For eksempel: DDoS-angrep og rene ytelsesproblemer vil ofte føre til samme symptomer, selv om de bakenforliggende årsakene er vidt forskjellig. I slike situasjoner kan det være utfordrende å kategorisere hendelsen korrekt før ytterligere analyse er gjennomført.

2.1.4 Hendelseshåndtering innenfor IT

Hendelseshåndtering innenfor IT har som formål å «minimere de negative konsekvensene av hendelser ved å gjenopprette normal drift så tidlig som mulig», som definert av ITIL-rammeverket [41]. Hendelseshåndtering som begrep kan brukes om både en tjenesteprosess (eng. *Incident Management*), i tillegg til å beskrive spesifikke steg, faser og tiltak for å mitigere og forhindre uønskede hendelser (eng. *Incident Handling* eller *Incident Response*). I denne oppgaven vil vi bruke hendelseshåndtering som et samlebegrep for å beskrive begge disse aspektene. I seksjon 2.3 gjennomgås de forskjellige fasene av hendelseshåndteringsprosessen i større detalj.

2.1.5 Security Operations Center

I dagens samfunn har nesten enhver medium til stor organisasjon en form for sikkerhetsavdeling [4]. Det finnes flere ulike navnestandarder for sikkerhetsavdelinger med tilhørende ansvarsområder og tjenester. Allikevel handler disse tjenestene primært om å detektere, analysere og respondere på sikkerhetsbrudd; kjerneoppgaver som blir utført i et Security Operation Center (SOC). MITRE [4] definerer et SOC som «et team, primært bestående av spesialister innenfor cybersikkerhet, organisert for å forhindre, detektere, analysere, respondere og rapportere på sikkerhetshendelser.» Med bakgrunn i dette skal vi benytte termen «SOC» videre i rapporten for å referere til alle typer sikkerhetsavdelinger, uavhengig av størrelse.

2.1.6 PPT - People, Process & Technology

People, Process & Technology (PPT) er et kjent og mye brukt rammeverk i IT-bransjen. Rammeverket legger vekt på at for å kunne ha organisasjonell effektivitet må man ha en god balanse og relasjon mellom de tre dimensjonene menneske, prosess og teknologi [42]. På lik linje med KIT-triaden handler rammeverket om en sammenheng mellom forskjellige dimensjoner, der målet er å skape merverdi dersom man oppnår en god balanse mellom alle dimensjonene.

Rammeverket sammenliknes gjerne med en trebent krakk. Hvis et av benene forsvinner, så vil krakken falle sammen. På samme måte vil det innen informasjonssikkerhet ha liten betydning for virksomheter sin effektivitet å endre på bare én av dimensjonene uten å ta høyde for de to andre [43]. For å eksemplifisere dette vil det gi lite verdi å kjøpe inn den nyeste, mest avanserte teknologien om man ikke har fagpersonell som vet hvordan dette skal benyttes. Det samme kan sies om prosesser: hvis man gjør endringer på prosesser uten at dette speiles opp mot teknologien som blir brukt vil man ikke se mye resultat. Et mer konkret eksempel kommer fra MITRE, som sier at for å ha en effektiv SOC som opererer med et høyt tempo, så må man ha en sammenheng mellom mennesker, prosess og teknologi, slik at SOC-et kan detektere, forstå og agere på trusler raskt, både proaktivt og reaktivt [4].

2.1.7 Cyberangrep

NIST definerer et cyberangrep som «enhver form for ondsinnet aktivitet som forsøker å samle inn, forstyrre, nekte tilgang til, svekke eller ødelegge informasjonssystemressurser eller selve informasjonen» [16]. Vi vil videre bruke begrepet cyberangrep når vi snakker om enhver sikkerhetshendelse som oppstår, der en eller flere trusselaktører utfører handlinger for å oppnå en eller flere mål, basert på NIST sin definisjon.

2.1.8 Trusler og trusselaktører

ISO 27000 [44] definerer en trussel som en potensiell årsak til en uønsket hendelse, der hendelsen kan forårsake skade på et system eller en organisasjon. En trusselaktør kan defineres som en person, gruppe, organisasjon eller statsaktør som utfører eller planlegger å utføre ondsinnede handlinger [16]. Alle trusler trenger likevel ikke å stamme fra det vi oppfatter som en tradisjonell trusselaktør. Typiske trusler kan være fysiske trusler som brann, oversvømmelse, naturkatastrofer og tap av materiell, eller det kan være bevisste ondsinnede cyberangrep [45]. Blant disse truslene er det vanligvis kun naturlig å prate om en trusselaktør når det gjelder de bevisste ondsinnede handlingene, og ikke når hendelser forårsakes av naturkatastrofer, eller uaktsomme legitime brukere. Det er likevel fullt mulig at en trussel slik som brann stammer fra en bevisst handling forårsaket av en trusselaktør, men denne rapporten fokuserer på trusler mot informasjonssystemer som i hovedsak stammer fra bevisst ondsinnede trusselaktører.

Løsepengevirus og annen ondsinnet programvare

Løsepengevirus er en av de største og vanligste truslene moderne organisasjoner må forsvare seg mot under dagens trusselsituasjon. Ifølge IBM var installering av løsepengevirus den vanligste handlingen utført under cyberangrep i 2021, der det sto for hele 21 prosent av alle oppdagede handlinger [46]. I samme statistikk fra 2022 er tallet redusert til 17 prosent [47].

Løsepengevirus kan defineres både som en type ondsinnet programvare, og som en form for cyberangrep utført av ondsinnede aktører som krever løsepenger for å frigjøre kryptert eller stjålet data [19]. De fleste former for løsepengevirus vil spre seg på nettverket og kryptere offerets data før det krever en eller annen form for betaling for å frigjøre dataen tilbake til offeret. Videre handling kan være forskjellig fra løsepengevirus til løsepengevirus, og fra trusselaktør til trusselaktør, der noen aktører eksempelvis truer med å selge stjålet data til høystbydende på det mørke nettet, som er en underdel av internett som ikke er tilgjengelig fra ordinære nettlesere.

Løsepengevirus er kun én type skadevare som kan utgjøre en trussel mot organisasjoner. Skadevare kan på generelt grunnlag defineres som «et program som installeres på et system, vanligvis skjult, med mål om å kompromittere konfidensialiteten, integriteten eller tilgjengeligheten til offerets data, applikasjoner eller system, eller på annet vis forstyrre eller ødelegge for offeret» [16]. De tre vanligste typene skadevare er datavirus, ormer og trojanere. «Datavirus er innen IT en type skadevare som kopierer seg inn i andre filer på en datamaskin og tilkoblede lagringsenheter» [12]. En dataorm er en type skadevare som er i stand til å spre seg selv fra en datamaskin til en annen [24]. En trojaner, også kalt en trojansk hest, er en form for skadevare skjult som et legitimt program eller fil. For hver av disse finnes det en rekke varianter og underarter som har forårsaket, og fortsetter å forårsake, stor skade til organisasjoners digitale systemer og verdier hvert eneste år.

Tjenestenektangrep

NSM beskriver et tjenestenektangrep (eng: Denial of Service) som et angrep som kun rammer tilgjengeligheten til de tjenester, systemer eller infrastrukturkomponentener som blir angrepet. En trusselaktør gjennomfører et slikt angrep ved å «overbelaste en eller flere ressurser (nettverksbåndbredde, CPU, minne, disk, osv.) hos målet for angrepet, eller utnytte svakheter i nettverks- eller applikasjonsprotokoller som brukes av målet for angrepet» [18]. Tjenestenektangrep er blant de vanligste typene cyberangrep, i tillegg til å være svært synlig for både den berørte organisasjonen og ikke minst berørte sluttbrukere.

De fleste alvorlige tjenestenektangrep er såkalte distribuerte angrep (eng: *Distributed Denial of Service*). Det vil si at den overbelastende trafikken stammer fra en stor rekke forskjellige enheter. De fleste tjenester som blir utsatt for slike angrep er vanligvis rustet for å tåle større mengder legitim trafikk samtidig, så antall enheter som kreves for å forårsake en forstyrrelse av tilgjengelighet til en tjeneste

er normalt stor. Derfor vil de fleste trusselaktører utnytte såkalte botnett for denne typen cyberangrep, som er nettverk av kompromitterte systemer og enheter som kontrolleres av trusselaktøren som ønsker å forårsake hendelsen [48]. Botnett kan være svært komplekse og vanskelig å stoppe når de først har fått fotfeste og kontroll over et stort antall enheter. Mozi – et av de største botnettene i verden som i 2021 sto bak hele 74 prosent av nye kompromitterte enheter på tingenes internett[1] – er et eksempel på et mye benyttet botnet for tjenestenektangrep [46]. Dette botnettet benytter en nettverksstruktur som lar hver enkelt kompromittert enhet operere individuelt og kompromittere nye sårbare enheter, som gjør det svært vanskelig å stoppe spredningen av botnettet selvom de sentrale aktørene bak stoppes [50].

Avanserte og vedvarende trusselaktører

Avanserte og vedvarende trusselaktører (APT) skiller seg fra andre trusselaktører ved at de ofte er svært sofistikerte, ressurssterke og målrettede [6]. De har ofte sterke økonomiske eller politiske insentiv, og flere av de største aktørene har direkte eller indirekte bånd til statlige organisasjoner i forskjellige land [51]. En avansert og vedvarende trussel er nøye planlagt, retter seg mot sensitiv og økonomisk verdifull informasjon, og har som mål å forbli skjult og oppnå uautorisert tilgang til nettverk og systemer over lengre perioder. For å oppnå dette benytter trusselaktørene et bredt spekter av angrepsvektorer og en rekke avanserte teknikker og verktøy, i tillegg til å utnytte kritiske - ofte ukjente - sårbarheter (se 2.1.10). Bruken av avanserte teknikker og ukjente sårbarheter gjør det vanskelig å detektere slike angrep når de først har fått fotfeste innenfor et nettverk eller system [22].

Avanserte og vedvarende trusler står bak mange av verdens største og mest kostbare cyberangrep, ofte tett sammenknyttet geopolitiske kriser som krigen i Ukraina, og som en del av sammensatte trusler [18, 51]. Disse truslene har gjerne komplekse levesykluser som diskuteres i nærmere detalj i seksjon 2.2.

2.1.9 Angrepsvektorer

Trusselaktører utfører cyberangrep og benytter forskjellige angrepsvektor for å nå sine mål. En angrepsvektor er selve metoden, eller kombinasjonen av metoder som trusselaktører benytter for å kompromittere eller infiltrere målets nettverk [3]. Dette kan være med formål om å forstyrre, deaktivere, ødelegge eller ta kontroll over datasystemer eller -infrastruktur [16]. NSM påpeker i sin risikorapport for 2023 at «forsøk på å skaffe seg tilgang til virksomheters datasystemer i hovedsak blitt gjort ved velutprøvde metoder og varianter av sosial manipulasjon, for eksempel phishing-eposter, og automatiserte påloggingsforsøk» [18]. Med andre

[1] Tingenes internett (eng: *Internet of Things*) er fysiske enheter som kommuniserer med hverandre og internett, de er gjerne batteridrevne. Eksempler på slike enheter er kjøleskap, komfyrer, termostater, dører, kameraer m.m. [49]

ord er det flere vanlige og velprøvde angrepsvektorer som ofte går igjen. Følgende underseksjoner redegjør for de tre vanligste angrepsvektorene i 2022, som definert av IBM [47].

Phishing

En av de vanligste måtene å spre ondsinnet programvare er ved hjelp av phishing. Phishing er en teknikk hvor man med hensikt forsøker å lure individer til å oppgi sensitive personopplysninger via villedende og datamaskinbaserte midler [16]. Teknikken er en digital form av sosial manipulering, og benytter forfalsket innhold for å maskere den ondsinnede programvaren og fremstå som legitim. Det er mange angrepsflater for phishing, som vil si at det er mange områder i et system som er sårbare for phishing-angrep. Den vanligste angrepsflaten til phishing er e-post. I e-poster er ofte ondsinnet programvare maskert som lenker, som laster ned virus. Phishing kommer i flere former, eksempelvis målrettet phishing (eng: *spearphishing*). Dette er som regel e-poster som er målrettet utformet for å lure mottakeren. E-postens innhold og utforming fremstår som relevant og legitim for mottakeren [18].

Statistikk fra IBM viser at phishing-operasjoner var den vanligste kompromitteringsmetoden i 2021 [46]. Rapporten viser til at hele 41 prosent av hendelser benyttet phishing som angrepsvektor for å oppnå initiell aksess [46]. I tillegg viser statistikken fra 2021 at klikk-effektiviteten til målrettede phishingkampanjer var på 17,8 prosent, mens de kampanjene som la til telefonsamtaler (*voice phishing*) var opptil tre ganger så effektive. Disse kampanjene viste til hele 53,2 prosent klikk-effektivitet blant ofrene [46]. Også i 2022 tronet phishing på topp som den mest brukte og ledende aksess- og angrepsvektoren, hvor 41 prosent av alle hendelser ble initiert via phishing [47].

I tillegg til phishing, har man en rekke andre angrepsvektorer som er nært relatert eller som ofte brukes i sammenheng med phishing-angrep. For eksempel e-postvedlegg (eng: *email attachments*) som nevnt i avsnittet ovenfor regnes som en egen type angrepsvektor, men regnes også som en separat angrepsvektor. Andre vektorer som nettsider og eksterne lagringsmedium, er sammen med e-postvedlegg regnet som de tre mest utbredte metodene for leveranse av bevæpnet nyttelast blant APT-aktører i perioden 2004-2010 [6].

Utnyttelse av offentlig tilgjengelige tjenester og applikasjoner

Utnyttelse av offentlig tilgjengelige tjenester og applikasjoner er en teknikk brukt av trusselaktører for å oppnå initiell aksess til et system eller nettverk. Teknikken gjennomføres ved å utnytte sårbarheter i offentlige klienter eller systemer [52]. På IBM sin liste over vanligste angrepsvektorer finner vi teknikken på en sterk andreplass bak phishing. Denne angrepsvektoren sto for 26 prosent av hendelser forårsaket av trusselaktører i 2022 [47]. Angrepsvektoren er representert i mer enn en fjerdedel av alle hendelser, og har siden 2019 vært en foretrukket metode for trusselaktører siden 2019 [47]. Det finnes mange forskjellige typer sårbarheter

hvor denne angrepsvektoren benyttes. Noen av de vanligste er blant annet SQL injections og buffer overflow. På et overordnet nivå kan man si at utnyttede applikasjoner vanligvis er nettsider og servere, samt databaser og standardtjenester som SSH[2] [52].

Kompromitterte legitimasjoner

Kompromitterte legitimasjoner er en teknikk som trusselaktører kan benytte for å oppnå tilganger til systemer for ulike ressurser. Slike systemer kan blant annet være VPN-er, nettverksenheter eller tilkobling til eksternt skrivebord [53]. I tillegg kan kompromitterte legitimasjoner medføre tilganger til spesifikke deler av et nettverk, og kan samtidig gi økt rettigheter til forskjellige systemer [53]. Kompromitterte legitimasjoner er sammen med teknikkene ovenfor på topplisten av IBM sin statistikk. I 2022 var teknikken identifisert i 16 prosent av observerte hendelser [47]. I disse hendelsene brukte trusselaktørene legitimasjonene som et middel for å oppnå initiell aksess til systemer [47].

2.1.10 Sårbarhet

En sårbarhet er en kjent svakhet i et system, systemets sikkerhetsprosedyrer, interne kontroller eller implementering av systemet, som en trussel eller event kan utnytte bevisst eller ubevisst for å aksessere, modifisere eller avbryte normal drift av et system [54]. En sårbarhet som ikke er kjent kalles gjerne en nulldagssårbarhet. NSM definerer en nulldagssårbarhet som en «sårbarhet programvareutvikleren ikke har oppdaget eller laget sikkerhetsoppdatering for» [18]. Navnet nulldagssårbarhet kommer av antall dager utvikleren har hatt på seg til å få løst opp i sårbarheten etter at den har blitt kjent, enten ved at noen oppdager sårbarheten og offentliggjør informasjon om den (eng: *disclosed*) eller ved at den blir utnyttet av en trussel eller trusselaktør.

Antallet hendelser forårsaket av sårbarhetsutnyttelser har ifølge IBM økt de siste årene. I tillegg viser de til at det er en stor grad av nye sårbarheter, inkludert nulldagssårbarheter, som blir utnyttet mest. I 2021 var 4 av de 5 mest utnyttede sårbarhetene nye sårbarheter, inkludert nulldagssårbarheten Log4j, selv om denne ikke ble offentlig kjent før i desember 2021 [46].

2.1.11 Risikostyring

Risikostyring omhandler prosessen rundt å ivareta en virksomhets verdier [37]. Selve risikostyringsprosessen deles gjerne inn i tre deler: identifisere risiko, analysere risiko og evaluere risiko. Generelt sett vil risikostyringen til en virksomhet bygge rundt en definert risikoapetitt. Risikoapetitten til virksomheter er risikoterskelen, på bakgrunn av verdiene de ønsker å beskytte, som virksomheten er villig til å akseptere [37]. Denne risikoen blir altså ikke håndtert direkte, men vil heller

[2] `https://www.openssh.com/`

måtte monitoreres kontinuerlig. Risikobildet er i stadig endring, og trusselaktører benytter seg av stadig nye virkemidler [18]. Det er derfor særdeles viktig å merke seg at risikostyring vil være en kontinuerlig prosess, og en svært viktig en sådan.

2.1.12 Automatisering

«Automatisering er teknikken å få systemer til å fungere uten, eller med liten grad av menneskelig medvirkning» [55]. Et mye brukt synonym til begrepet automatisering er automasjon, men vi vil i denne rapporten konsekvent benytte automatisering. Viktigheten av automatiserte løsninger og diskusjon rundt hvorvidt det er mulig å automatisere forskjellige løsninger og arbeidsoppgaver vil være sentrale punkter senere i rapporten.

2.1.13 Kunstig intelligens

Kunstig intelligens er et mye omdiskutert og populært tema nå i 2023, og det vil være vanskelig å utforme en rapport om effektivisering og automatisering av datasystemer uten at begrepet diskuteres. NSM beskriver at kunstig intelligens handler om «datamaskiners evne til å utføre handlinger, fysisk eller digitalt, basert på tolkning og behandling av strukturerte eller ustrukturerte data, i den hensikt å oppnå et gitt mål. Enkelte systemer for kunstig intelligens kan også tilpasses gjennom analyser og vurderinger av hvordan tidligere handlinger har påvirket omgivelsene» [18]. I denne rapporten vil vi diskutere forskjellige metoder for hendelseshåndtering som benytter seg av kunstig intelligens i større eller mindre grad. Bruken av kunstig intelligens i slike metoder, verktøy og tjenester baserer seg enten på regelbaserte modeller (ekspertsystem) eller datadrevne modeller (maskinlæring) [21, 56]. Dette er underarter av kunstig intelligens, selv om begrepene maskinlæring og kunstig intelligens ofte brukes feilaktig om hverandre [20]. Kunstig intelligens er et generelt begrep for informasjonsteknologi som kan justere sin adferd for å fremstå som intelligent, der maskinlæring kun er en av modellene som benyttes for å oppnå dette.

Ekspertsystemer benytter forhåndsdefinerte regler som er programmert før modellen brukes [56]. Slike modeller kan benyttes innenfor verktøy og tjenester for hendelseshåndtering, for eksempel ved å avlaste arbeid som kan løses av en datamaskin basert på forhåndsdefinerte regler. Maskinlæringsmodeller vil på den andre siden ikke inkludere forhåndsdefinerte regler, men er programmert til å kontinuerlig lære basert på data modellen blir servert, og problemer modellen blir trent på [20, 21]. Maskinlæring innenfor hendelseshåndtering kan benyttes i en rekke verktøy og tjenester for å løse forskjellige problemer, både kjente problemer som modellen har blitt trent på tidligere, og ukjente problemer som modellen vil forsøke å løse basert på tidligere og lignende problemer. Videre i denne rapporten vil vi for ordens skyld kun bruke begrepet maskinlæring når vi snakker om modeller for kunstig intelligens som benyttes innenfor metoder, verktøy og tjenester for hendelseshåndtering. Vi poengterer at det likevel er mulig at regelbaserte model-

ler (ekspertsystem) er den underliggende modellen som er i bruk når vi diskuterer forskjellige bruksområder for kunstig intelligens innenfor hendelseshåndtering.

2.2 Livssyklusen til et cyberangrep

For å forstå hvordan sikkerhetshendelser oppstår, hvordan de utvikler seg og hvordan de får fotfeste i et system eller nettverk, er det nødvendig å bryte ned hendelsens livssyklus i sin helhet. Effektiv hendelseshåndtering handler ikke kun om å effektivisere prosessene som begynner i det en hendelse er oppdaget, men også om å effektivisere verktøyene og teknikkene for å oppdage og stanse cyberangrep i tidligere stadium av angrepets livssyklus, eller stanse angrep som tidligere har fått fotfeste ubemerket.

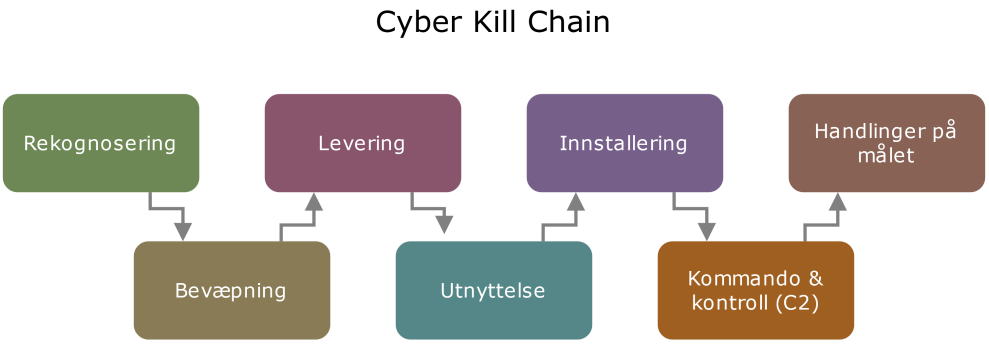
En sikkerhetshendelses livsløp starter sjeldent i det øyeblikk hendelsen oppstår, spesielt når det er snakk om hendelser forårsaket av avanserte og vedvarende trusler (se 2.1.8) [6]. Samtidig oppdages mange datainnbrudd først etter at en trusselaktør har hatt tilgang til nettverket over lengre tid, og andre vellykkede angrep har etter all sannsynlighet ikke blitt oppdaget den dag i dag. I slike tilfeller snakkes det om livssyklusen til et datainnbrudd (eng. *data breach lifecycle*). Dette er definert av IBM som tiden det tar fra første deteksjon av et datainnbrudd, til datainnbruddet er håndtert og normal drift gjenopprettes [57]. Tiden det tar å identifisere et datainnbrudd er tiden fra datainnbruddet oppstår til datainnbruddet først oppdages. Disse metrikkene kan brukes for å bedømme effektiviteten til hendelseshåndteringsprosessene til en organisasjon.

En rapport fra Deep Instinct [58] forteller at en bedrift i gjennomsnitt bruker 20,9 timer for å håndtere et pågående cyberangrep. Samtidig rapporterer IBM om at gjennomsnittlig tid for å identifisere og håndtere et vellykket datainnbrudd er på hele 277 dager. 207 av disse 277 dagene er gjennomsnittlig tid for å i det hele tatt detektere datainnbruddet, altså tiden fra datainnbruddet oppstår til det oppdages for første gang [57]. Merk at ikke alle cyberangrep leder til et velykket datainnbrudd. Statistikken fra Deep Instinct gjelder håndtering av detekterte og pågående cyberangrep, mens IBM diskuterer velykkede datainnbrudd. Dette illustrerer viktigheten av å detektere og håndtere hendelser så tidlig som mulig i innbruddets livssyklus for å begrense skade på organisasjonens verdier.

2.2.1 Cyber Kill Chain og MITRE ATT&CK

For å illustrere livssyklusen til et cyberangrep i form av enavansert og vedvarende trussel, tar denne rapporten utgangspunkt i en modell utviklet av forskere fra Lockheed Martin, kalt Cyber Kill Chain [6]. Modellen, vist i figur 2.2, strukturerer levetiden til et cyberangrep fra trusselaktørenes perspektiv, ikke sikkerhetsanalytikerens. Ved å betrakte et cyberangrep fra trusselaktørens perspektiv, kan tiltak iverksettes for å detektere og avverge angrep tidligere, og samtidig detektere angrep som tidligere har gått ubemerket. Modellen er bygget opp etter en antagelse om at ethvert cyberangrep går gjennom de samme syv fasene, slik at ethvert

angrep kan avlives - derav *kill chain* (norsk: drapskjede) - og avverges gjennom vellykkede tiltak rettet mot en gitt fase i livssyklusen.



Figur 2.2: Cyber Kill Chain [6]

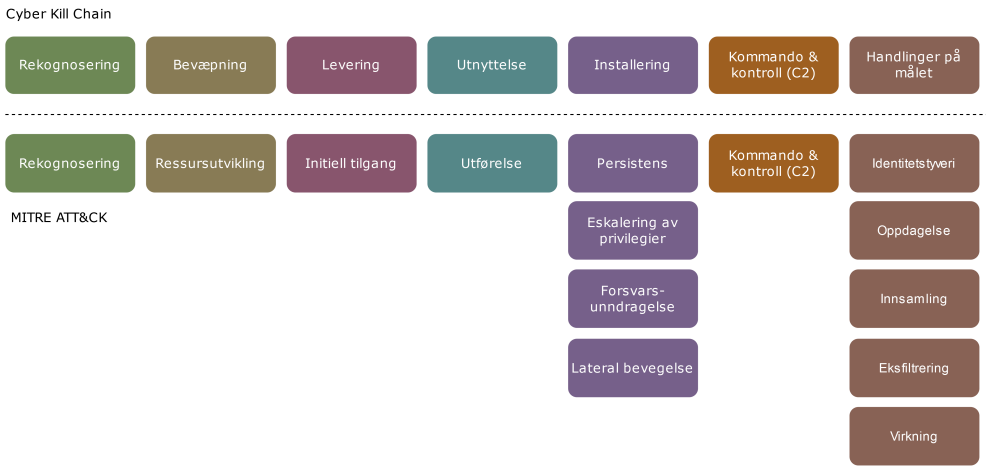
En av de største styrkene ved en slik forenkling under hendelsesanalyse er muligheten til å både rekonstruere fasene av et cyberangrep før deteksjon, og syntetisere (analysere seg frem til) potensielle faser etter deteksjon [6]. Dette gjør det mulig å finne nye indikatorer på potensielle cyberangrep, og dermed detektere fremtidige angrep tidligere, eller oppdage hitill ukjente kompromitteringer. I praksis vil de fleste alvorlige cyberangrep være unike i både fremgangsmåte og taktikk, og derfor blir ofte en rekke andre mer praktiske rettede modeller og rammeverk benyttet for å illustrere livssyklusen til et cyberangrep.

En av disse rammeverkene som ofte brukes i tilknytning med eller som alternativ til Cyber Kill Chain er rammeverket MITRE ATT&CK (*Adversarial Tactics, Techniques, and Common Knowledge*). I likhet med Cyber Kill Chain, omhandler ATT&CK taktikker og teknikker brukt av trusselaktører under et antatt datainnbrudd [22]. Dette er et tankesett som antar at en trusselaktør allerede har kompromittert et nettverk eller et system. Rammeverket kan benyttes for å identifisere og utbedre hull eller mangler ved forsvarmekanismer eller infrastruktur, samt utarbeide deteksjonslogikk (logikk som definerer hva som regnes som legitim, illegitim, normal eller unormal data eller trafikk [4]). Når ATT&CK-rammeverket først ble publisert i 2017 fokuserte det kun på aktiviteter en trusselaktør gjennomfører etter å ha kompromittert og fått fotfeste i et system eller nettverk, og det skilte seg dermed fra Cyber Kill Chain som tar for seg hele livssyklusen inkludert aktivitetene en trusselaktør gjennomfører før kompromittering. Nyere versjoner av ATT&CK har derimot lagt til flere taktikker og dekker nå også taktikker og teknikker før og under kompromittering[3] [59]. I tillegg redegjør ATT&CK i større detalj om de taktikker og teknikker trusselaktører benytter seg av, og hvordan disse kan motvirkes, der Cyber Kill Chain heller har fokus på etterretning og analyse av datainnbrudd.

Denne rapporten kombinerer både Cyber Kill Chain-modellen og ATT&CK-

[3]Nyere versjoner av MITRE ATT&CK publiseres kun gjennom ATT&CK sin offisielle nettside: `https://attack.mitre.org/resources/versions/`

rammeverket for å illustrere livssyklusen til et cyberangrep. I følgende underseksjoner defineres de syv fasene av et cyberangrep som strukturert i Cyber Kill Chain, og utfylles med tilhørende taktikker, teknikker og mottiltak som definert av ATT&CK. Figur 2.3 viser sammenhengen mellom fasene i Cyber Kill Chain og taktikkene i MITRE ATT&CK. Merk at disse definisjonene - i likhet med Cyber Kill Chain og ATT&CK - tar utgangspunkt i avanserte og vedvarende trusler der målet med cyberangrepet er å oppnå kontroll og tilstedeværelse på et system eller nettverk over lengre tid uten å bli oppdaget [51]. Likhetstrekk kan trekkes til andre trusler og former for cyberangrep, men utredes ikke i nærmere detalj.



Figur 2.3: Cyber Kill Chain vs. MITRE ATT&CK [6, 59]

Rekognosering

Under rekognosering vil en trusselaktør identifisere og velge potensielle mål, og samle inn informasjon som vil bli brukt i senere faser av angrepet [6]. Dette kalles også offentlig informasjonsinnsamling, eller frikildeetterretning. Dette gjelder informasjon om målet i seg selv - som kan være en person eller en organisasjon - samt informasjon relatert til målet, for eksempel informasjon om teknologier brukt i informasjonssystemene til en organisasjon.

Nyere versjoner av ATT&CK-rammeverket inkluderer rekognosering som en egen taktikk, med teknikker som inkluderer offentlig informasjonsinnsamling i tillegg til phishing for informasjon eller nettverksskanning av potensielle mål [59].

Bevæpning

Bevæpning omhandler utvikling av en form for ondsinnet programvare, vanligvis skjult under dekke av et legitimt program eller fil (slik som trojanere) [6]. Begrepet bevæpning stammer fra at den ondsinnede koden ofte tillegges eller skjules som legitime program eller filer, som dermed blir bevæpnet eller armert. Vanlige

eksempler er ondsinnet kode som en del av PDF-filer eller Microsoft Office dokumenter, altså filtyper og program med bred troverdighet blant brukere som også er lett å spre.

ATT&CK-rammeverket definerer taktikken ressursutvikling (eng: *resource development*), som i likhet med Cyber Kill Chain omhandler utvikling og bevæpning av ondsinnet programvare, i tillegg til mer avansert planlegging som oppsett av botnett og mindre cyberangrep som skal underbygge et større datainnbrudd [59].

Levering

Etter bevæpning blir ondsinnet programvare levert til målet gjennom en angrepsvektor (se 2.1.9) [6]. Typiske angrepsvektorer er phishing med ondsinnede vedlegg eller lenke til ondsinnede nettsteder, i tillegg til kompromitterte flyttbare medium slik som minnepenner. Slike angrepsvektorer utnytter typisk menneskelige svakheter gjennom sosial manipulering, for å levere ondsinnet programvare og kompromittere målet.

Taktikken initiell tilgang i ATT&CK inkluderer mange av de samme teknikkene, men går enda lenger i dybden på avanserte angrepsvektorer som blant annet kompromittering av leverandørkjeder, eller tilkobling til offerets nettverk med kompromitterte legitimasjoner fra tidligere faser og angrep [59].

Utnyttelse

Når en trusselaktør har oppnådd initiell tilgang et system, kan trusselaktøren utnytte en eller flere sårbarheter hos offeret for å oppnå sine mål [6]. Sårbarheten kan være menneskelig, som når brukere utsatt for phishing åpner og kjører ondsinnet programvare eller filer som inneholder ondsinnet kode, eller den kan være teknisk, for eksempel ved å utnytte en nulldagssårbarhet i et system eller en applikasjon.

ATT&CK-rammeverket inkluderer mange teknikker for utnyttelse og utførelse av kode, som i tillegg til eksemplene nevnt over inkluderer teknikker slik som utnyttelse av containerteknologi (teknologi som omfatter utrulling av applikasjoner som *containere*), skyressurser og Application Programming Interface (API), samt sette opp automatiserte jobber for gjentatt utnyttelse og utførelse av kode [59].

Installering

For å oppnå kontinuerlig og langvarig tilstedeværelse og tilgang til et kompromittert system, vil trusselaktører i mange tilfeller forsøke å installere en bakdør på systemet [6]. En slik bakdør har som mål å gi trusselaktøren skjult tilgang til systemet utover de normale, legitime tilgangsmetodene som eksisterer på systemet fra før.

I ATT&CK-rammeverket defineres dette som taktikken persistens (eng. *persistence*) [59]. Listen over teknikker for å oppnå persistens er lang, fra opprettelse av priviligerte brukere, til kapring og utnyttelse av legitim programvare eller endring

av sentrale prosesser på systemet. Felles for alle teknikkene er målet om å sikre tilgang selv om systemet restartes, påloggingsinformasjon endres, tilganger trekkes tilbake, eller sårbare programmer oppdateres og sikkerhetshull tettes. Dette er en av de viktigste fasene av et avansert og vedvarende datainnbrudd ettersom initiell tilgang og utnyttelse ofte er basert på angrepsvektorer og sårbarheter med naturlig begrenset levetid, som de fleste organisasjoner forsøker å motvirke kontinuerlig uavhengig av om de oppdager datainnbruddet eller ikke.

Kommando og kontroll

Når en trusselaktør har installert en bakdør på det kompromitterte systemet, kan aktøren kommunisere med systemet og utføre videre handlinger gjennom denne kommunikasjonskanalen. Dette refereres til som kommando og kontroll, ofte forkortet C2 [6].

Denne taktikken er helt sentral for å muliggjøre videre utnyttelse av det kompromitterte systemet, og en trusselaktør kan velge mellom en rekke teknikker for kommunikasjon [59]. Felles for alle teknikkene er et mål om å unngå at den ondsinnede trafikken mellom trusselaktør og offer oppdages. Dette kan blant annet oppnås ved å skjule instrukser som en del av «normal» internett- eller eposttrafikk. En annen teknikk innebærer å kryptere eller kode trafikken slik at den er uleselig eller uforståelig for alle utenom trusselaktøren eller dem som bevisst leter etter ondsinnet trafikk.

Handlinger på målet

Den siste fasen i livssyklusen til et vedvarende cyberangrep er utførelse av handlinger på det kompromitterte systemet, slik at trusselaktøren kan oppnå målene med angrepet [6]. Denne fasen påbegynnes normalt sett ikke før de andre fasene er gjennomført. Dette er fordi trusselaktøren først og fremst ønsker å sikre kontinuerlig kontroll for å ha tid til å gjennomføre komplekse handlinger, og fordi trusselaktøren kan ha langsiktige mål med angrepet. Et typisk langsiktig mål er uthenting av sensitiv eller verdifull data fra offerets systemer.

ATT&CK-rammeverket har flere taktikker som omhandler, og utvider, denne fasen av et cyberangrep [59]. Noen av disse taktikkene underbygger angrepet eller muliggjør mer komplekse angrep, gjennom eskalering av privilegier og tyveri av legitimiteter, samt utforsking og lateral bevegelse gjennom systemet eller nettverket. I tillegg vil en trusselaktør kontinuerlig bruke teknikker for å unngå deteksjon så lenge de er innenfor systemet. Andre taktikker baserer seg på trusselaktørens endelige mål. Dette inkluderer teknikker for innsamling og uthenting av sensitiv eller verdifull data, samt teknikker som skader integriteten eller påvirker tilgjengeligheten til data eller tjenester.

2.3 Hendelseshåndteringsprosessen

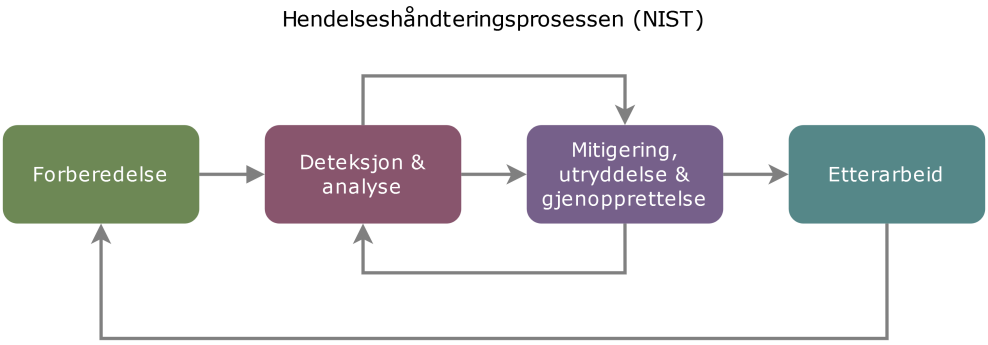
For å møte det digitale trusselbilde, motvirke potensielle og pågående hendelser og sikre verdier og tjenester, er enhver organisasjon direkte eller indirekte avhengig av en hendelseshåndteringsprosess. Dette er prosessen som koordinerer håndtering og oppfølging av hendelser, i tillegg til å forebygge potensielle fremtidige hendelser, og lede etterarbeid og analyse av tidligere hendelser for å forbedre prosessen og utarbeide tiltak. I kombinasjon med tekniske sikkerhetstiltak er prosessen det fremste verktøyet for å minimere negative konsekvenser av hendelser, uavhengig av om den er ledet av organisasjonen selv eller er tjenesteutsatt til en leverandør [41]. Effektiv håndtering når hendelser oppstår har en enorm påvirkning på hvordan sluttbrukere og kunder oppfatter sin tjenesteleverandør [41]. Antall dataangrep og digitale trusler fortsetter å øke fra år til år, noe som også gjør sikkerhetshåndtering viktigere enn noen gang [47].

Planer og prosess er likevel bare nyttig dersom det fungerer, og det er derfor også essensielt at prosessen testes og revideres regelmessig gjennom øvelser. IBM rapporterer om at organisasjoner som regelmessig tester sin hendelseshåndteringsprosess i gjennomsnitt sparer 58 prosent på kostnader relatert til datainnbrudd [57]. Øvelser kan blant annet innebære å utarbeide og teste playbooks, prosedyrer og arbeidsflyter som igangsettes ved gitte eventer og hendelser.

Hendelseshåndteringsprosessen har dype røtter, og det finnes en rekke rammeverk og standarder som definerer prosessen med tilhørende steg. Mange organisasjoner benytter ITIL-rammeverket for digital tjenesteforvaltning, som definerer hendelseshåndteringsprosessen gjennom 6 steg og knytter disse tett opp i mot ITIL sine andre praksiser og verdikjedesystem[4] [41]. I denne rapporten tar vi utgangspunkt i et rammeverk for hendelseshåndtering fra det amerikanske National Institute of Standards and Technology (NIST), samt ISO standard ISO/IEC 27035, som sammen danner en strukturert og fullstendig prosess for hendelseshåndtering [40].

NIST sin *Computer Security Incident Handling Guide* deler opp hendelseshåndteringsprosessen i de fire fasene (1) forberedelse (eng: *preparation*), (2) deteksjon og analyse (eng: *detection and analysis*), (3) mitigering, utryddelse og gjenoppretting (eng: *containment, eradication and recovery*) og (4) etterarbeid (eng: *post-incident activity*) [15]. Modellen er vist i figur 2.4. Det er disse fire fasene som benyttes som referanse videre i denne rapporten, og de defineres videre i seksjon 2.3.1 til 2.3.4. I tillegg vil vi integrere elementer fra ISO standard 27035 [39], som strukturerer hendelseshåndtering som en syklisk prosess der lærdom fra etterarbeid og analyse danner grunnlaget for forbedringer i planverk og rutiner.

[4] Verdikjedemodellen omfatter de aktiviteter og steg en organisasjon gjør for å skape verdi basert på behov/krav [41]. Modellen beskriver retningslinjer for hvordan en tjenesteleverander kan jobbe for å skape verdi ved å føge prinsipper og praksiser. Verdikjedesystemet består blant annet av de prosesser og aktiviteter en tjenesteleverandør har definert.



Figur 2.4: Hendelseshåndteringsprosessen (NIST) [15]

2.3.1 Forberedelse

Gode forberedelser er essensielt for å både motvirke fremtidige hendelser, og håndtere de hendelser som ikke kan unngås [15, s. 21–24]. Dette gjelder forberedelser som retter seg mot å sikre systemer, nettverk og tjenester, samt forberedelser i form av planverk, prosedyrer, rutiner og playbooks. Mer presist kan vi dele forberedelsesfasen inn i planlegging til å håndtere hendelser, og tiltak som skal forhindre at hendelser som oppstår. Håndteringsplanlegging involverer blant annet å sikre at kontaktinformasjon til alle relevante parter er på plass, at de nødvendige teknologiene, verktøyene og fasilitetene eksisterer og er klar for bruk, og at nødvendig dokumentasjon av systemer og rutiner for håndtering av forskjellige hendelser er utarbeidet og tilgjengelig når det trengs. Tiltak for å forhindre hendelser inkluderer gjennomføring av risikovurderinger for å avdekke verdier, sårbarheter og trusler, og for å bestemme sannsynligheten og konsekvensene ved forskjellige risikoer. I tillegg er tekniske tiltak for å sikre organisasjonen og dets verdier helt sentralt for å forhindre at hendelser oppstår i det hele tatt.

2.3.2 Deteksjon og analyse

Deteksjon og analyse omhandler tiltak og prosedyrer for å oppdage hendelser som oppstår så fort som mulig, i tillegg til å analysere dem effektivt og hente ut det nødvendige av informasjon for videre håndtering [15, s. 25–28]. For å detektere hendelser er det nødvendig å ha en god forståelse av sannsynlige angrepsvektorer og trusler, ha oversikt over nye og eksisterende sårbarheter, utarbeide deteksjonslogikk og kjenne til indikatorer for hendelser. Denne fasen av hendelseshåndtering inkluderer flere underdeler. Først og fremst er kartlegging av angrepsvektorer (se 2.1.9) viktig for å sørge for at deteksjonslogikk tilhørende alle sannsynlige angrepsvektorer er på plass. Deretter bør tegn på hendelser kartlegges. Dette inkluderer både tegn på at hendelser kan oppstå i fremtiden, kalt forløper (eng: *precursor*), og tegn på at en hendelse har oppstått (indikatorer).

Når en hendelse er detektert må den analyseres [15, s. 28–34]. Først og fremst handler dette om å kategorisere, klassifisere og prioritere hendelser, for å avdekke

hvorvidt det som er detektert faktisk er en hendelse (og ikke, for eksempel, bivirkninger av et planlagt vedlikeholdsarbeid eller lignende), hvorvidt hendelsen er reell (at hendelsen ikke er en falsk positiv) og hvor alvorlig hendelsen er [48]. Først når disse initielle vurderingene er gjennomført kan detaljert analyse av selve hendelsen begynne. Dette inkluderer analyse av hendelsens omfang for å avdekke hvilke systemer og brukere som er berørt av hendelsen, hvor og hvem hendelsen stammer fra, hvordan hendelsen har utviklet seg, hvilke angrepsvektorer som benyttes, hvilke sårbarheter som utnyttes, og hvilke konsekvenser hendelsen har for organisasjonen, brukere og partnere. I denne fasen av hendelseshåndtering er det også viktig å sørge for grundig dokumentasjon av hva som har skjedd, hvilken informasjon som samles inn og analyseres, og hva som vil bli gjort for å løse hendelsen. I tillegg er det viktig å opprettholde kontinuerlig kommunikasjon både internt og eksternt. Dette inkluderer varsling om hendelser, som skal informere berørte parter, kunder og andre interessenter om at en hendelse har oppstått, hva som er nåværende status, når hendelsen oppstod og når den er forventet å bli løst, hva som blir gjort for å håndtere hendelsen og hva som er berørt av hendelsen.

2.3.3 Mitigering, utryddelse og gjenopprettelse

Når en hendelse er detektert og analysert, må den håndteres i form av tiltak som mitigerer konsekvensene av hendelsen, utrydder hendelsen og gjenoppretter normal drift [15, s. 35–38]. Det første steget under denne fasen er valg av en håndteringsstrategi, som legger grunnlaget for resten av håndteringen [15, s. 35–36]. Kriterier for valg av håndteringsstrategi kan være for eksempel (1) Potensiell skade eller tap av ressurser (2) Behovet for bevisinnsamling (3) Tilgjengelighet av tjenester, gjerne bunnet i tjenestenivåavtale (4) Tid og ressurbruk på den planlagte strategien (5) Forventet tid løsningen er i bruk, for eksempel en hasteendring må bli forbedret eller fjernet i løpet av kort tid, men en permanent løsning vil ikke bli endret [15, s. 35]. Tidlig mitigering av en hendelse unngår overbelastning av ressurser, og de finansielle kostnadene og operative konsekvensene minimeres. I tillegg vil det frigjøre tid og ressurser til å utarbeide konkrete tiltak for å utrydde hendelsen og gjenopprette normaltilstand. Det er derfor hensiktsmessig å ha konkrete planer for hvordan forskjellige hendelser skal mitigeres. NIST trekker frem at dette stammer tilbake til forberedelsesfasen og det arbeidet som legges ned for å være forberedet når en hendelse først oppstår. Det er også nødvendig å være klar over konsekvensene av forskjellige mitigeringstiltak, for eksempel hva som blir berørt dersom systemer stenges ned, isoleres eller blokkeres i brannmuren («Brannmuren hindrer at utenforstående skal kunne nå tjenester eller informasjon som ikke skal være tilgjengelig» [9]). Slike vurderinger kan være tidkrevende og utfordrende under håndteringen av en hendelse, som igjen underbygger viktigheten av å ha forhåndsdefinerte håndteringsstrategier og -planer der det er mulig.

Videre steg under håndteringen av en hendelse inkluderer innsamling av bevis for videre analyse og for rapportering og dokumentasjonsformål [15, s. 36–37]. Alle steg som gjennomføres under håndtering av en hendelse bør dokumenteres.

Det er vanlig at bevis og informasjon dukker opp først under håndteringen av en hendelse, i tillegg til det som allerede finnes fra deteksjon og analyse, så det er viktig å kunne gjennomgå hva som har hendt. Som vist i figur 2.4 danner deteksjon og analyse en intern syklus med denne fasen, som gjentas frem til en hendelse er løst. Funn fra deteksjon og analyse legger grunnlaget for valg av passende responstiltak som forhåpentligvis skal løse hendelsen. Deretter kan resultatet av disse tiltakene og eventuelle nye funn skape behov for ytterligere deteksjon og analyse.

Når en hendelse er mitigert for å begrense skadeomfanget og frigjøre tid og ressurser, er neste steg utryddelse av selve hendelsen og gjenopprettelse av normaltilstand [15, s. 37–38]. Alvorlige hendelser kan kreve utryddelse av blant annet ondsinnet programvare og illegitime eller kompromitterte legitimasjoner, samt utryddelse av sårbarheter i form av tiltak som tetter sikkerhetshull for videre utnyttelse. Når hendelser er under kontroll må normaltilstand i system og nettverk gjenopprettes. Dette inkluderer selve gjenopprettelsen, verifisering av at alle systemer er tilbake i normal drift, samt strakstiltak for tetting av eventuelle sikkerhetshull og sårbarheter under gjenopprettelsen dersom det er aktuelt for hendelsen.

2.3.4 Etterarbeid

Den siste fasen av hendelseshåndteringsprosessen er etterarbeidsfasen [15, s. 38–42]. Selv med gode forberedelser og effektive prosedyrer og rutiner for deteksjon, analyse og håndtering av hendelser, vil enhver organisasjon bli utsatt for hendelser som krever at tiltak gjennomføres for å unngå at lignende hendelser oppstår igjen. Etterarbeidsfasen skal sørge for at det som er lært i løpet av håndteringen av en hendelse, fører til tiltak og forbedringer av hendelseshåndteringsprosessen i sin helhet. Dette skaper en syklisk modell der etterarbeid fra enhver hendelse underbygger forberedelsesfasen for fremtidige hendelser, som vist i figur 2.4. Etterarbeidsfasen består hovedsaklig av en gjennomgang av hendelsen for å finne ut hva som skjedde, hvorvidt rutiner ble fulgt, hva som eventuelt manglet i eksisterende rutiner, planverk eller verktøy, hva som fungerte og ikke fungerte under deteksjon, analyse og håndtering, hvilke forløpere og indikatorer til hendelsen som bør overvåkes i fremtiden, og hvilke tiltak som bør implementeres for å hindre at lignende hendelser oppstår [15, s. 38–39]. Dette kan gjennomføres over en eller flere møter med aktuelle parter og interessenter, før en skriftlig rapport utarbeides. En slik gjennomgang kalles ofte en hendelsesrevisjon, eller en hendelsesreview (fra engelsk: *incident revision/review*).

Et annet viktig aspekt av etterarbeidsfasen er innsamling av data for arkivering, dokumentasjon og generering av rapporter [15, s. 39–42]. Slik data kan blant annet brukes for å understreke behovet for ytterligere ressurser, teknologier eller verktøy, og for å fastsette metrikker (eng: *KPI-er*) som måler effektiviteten til hendelseshåndteringsarbeidet. I tillegg kan datagrunnlaget avdekke mulige fundamentale sikkerhetsmangler, feil eller skjulte forløpere og indikatorer for hen-

delser, som kan legge grunnlaget for ytterligere tiltak utover de som blir implementert som en del av hendelsesrevisjonen. Det kan også være nyttig i form av å kartlegge trender og mønstre blant hendelser, trusler og trusselaktører, som igjen kan benyttes for proaktivt forebyggende arbeid.

2.4 Metoder og teknologier for hendelseshåndtering

Hendelseshåndtering er en prosess som består av mange deler

2.4.1 Konvensjonelle og moderne metoder for hendelseshåndtering

Hendelseshåndtering er på ingen måter et nytt begrep, mennesker har håndtert hendelser i samfunn og industri i mange år. For eksempel ved å slukke branner, håndtere kriminalitet eller reparere verktøy og utstyr som slutter å fungere. Som en del av digitalisering har hendelser og håndteringen av dem også blitt digitale. Organisasjoner rundt om i verden har plassert store deler av sine verdier og kjernevirksomheter inn i det digitale domenet og med det følger også et skifte fra fysisk til digital hendelseshåndtering. Det har blitt viktigere enn noen gang at organisasjoner har planer og strategier for å slukke disse digitale brannene.

Konvensjonelle metoder for hendelseshåndtering

Hendelseshåndtering har blitt lagt frem i lys av ITIL-rammeverket, som omfatter administrasjon og drift av IT-tjenester. Et fokus på drift og sikring av IT-systemer, inkludert digital hendelseshåndtering, har vokst frem siden 1980-tallet og spiller i dag en minst like stor rolle som utvikling av informasjonsteknologisystemer [60, s. 5]. Rammeverket baserer seg hovedsaklig på manuelle prosesser, og brukes først og fremst for å systematisk utarbeide og bygge opp gode rutiner for hendelseshåndtering med mennesker og prosess i fokus [41]. ITIL og andre lignende rammeverk definerer hendelseshåndtering først og fremst som en tjenesteprosess, og ikke som et sett med tekniske og praktiske metoder for hendelseshåndtering. Viktigheten av å implementere rammeverk for IT-drift og hendelseshåndtering er godt dokumentert i flere tidligere forskningsartikler [60].

Moderne metoder for hendelseshåndtering

Nyere forskning og statistikk peker samtidig på at de konvensjonelle metodene for hendelseshåndtering ikke er tilstrekkelig for å møte dagens trusselbilde. Lockheed Martin påpeker at konvensjonelle metoder ikke er tilstrekkelig for mitigering mot avanserte og vedvarende trusler, fordi de i for stor grad baserer seg på to sviktende antakelser: at hendelseshåndtering skal skje etter kompromittering, og at kompromitteringer skjer som følge av feil som kan fikses [6, s. 1]. IBM rapporterer samtidig om at organisasjoner som benytter kunstig intelligens og automatisering som en del av sine sikkerhetssystemer, i gjennomsnitt har 65 prosent lavere

kostnader ved datainnbrudd [57]. Blant IBM sine analyser var bruken av kunstig intelligens og automatisering det tiltaket som førte til størst kostnadsbesparelse ved datainnbrudd, sammenlignet med organisasjoner som ikke benyttet det i sine systemer.

Denne utviklingen medfører at de konvensjonelle metodene for hendelseshåndtering i stadig større grad kombineres med løsninger som benytter kunstig intelligens og automatisering for økt effektivitet og reduserte kostnader. Videre i denne seksjonen vil vi redegjøre for moderne metoder og teknikker for hendelseshåndtering som blir viktigere og viktigere for å effektivisere hendelseshåndtering i moderne tid.

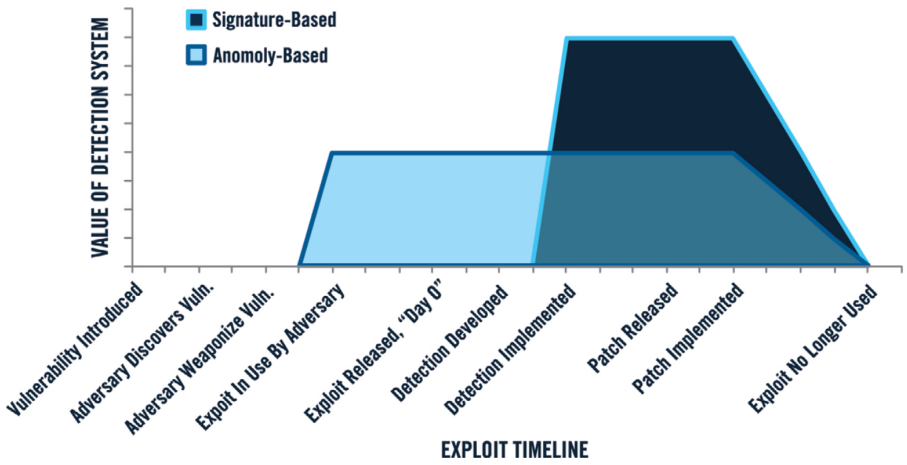
2.4.2 Signaturbasert og anomalitetsbasert deteksjon

Deteksjonsfasen under hendelseshåndtering er en sentral kandidat for implementering av automatisering og maskinlæring. To av de vanligste metodene for deteksjon er signaturdeteksjon og anomalitetsdeteksjon. Signaturdeteksjon kjennetegnes ved at systemet allerede har kunnskap om indikatorer ved hendelsen eller sårbarheten som utnyttes, altså eksisterer det en signatur for eventen eller hendelsen [4, s. 191]. En slik signatur er som regel kjente indikatorer som for eksempel hasher. En hash er en kalkulert verdi som man får ved å fôre data gjennom en matematisk algoritme [61].

Den andre typen deteksjon, altså anomalitetsdeteksjon fungerer ved at et normalbilde av tjenesten, systemet eller nettverket dannes gjennom analyse av data over tid. Deteksjonen baserer seg på anomaliteter ved tjenesten som faller utenfor normalbildet [4, s. 192]. Avvikene er typisk statistiske avvik der eventer faller utenfor den analyserte normalen, eller overskrider definerte terskeler for hva som er normalt [62].

Nytteverdien av deteksjon

Signaturdeteksjon har stor nytteverdi for sårbarheter der deteksjonslogikk er definert. Det skaper også verdi når en sårbarhet har vært kjent i lengre tid, men kan ikke benyttes før man har definerte signaturer. På den andre siden har anomalitetsdeteksjon nytteverdi så fort en sårbarhet er under aktiv utnyttelse av en trusselaktør, derimot er sannsynligheten for deteksjon mer usikker gjennom sårbarhetens levetid. De forskjellige bruksområdene og styrkene illustrerer behovet for implementering av både signatur- og anomalitetsdeteksjon hos organisasjoner. Figur 2.5 viser forskjeller i nytteverdi hos de to forskjellige deteksjonsmetodene. Figuren viser hovedsakelig hvordan anomalitetsdeteksjon dekker en større del av sårbarhetens livstid, siden det er mer usikkert om den klarer å oppdage sårbarhetene.



Figur 2.5: Signaturbasert vs. anomalitetsbasert deteksjon [4, s. 195]

2.4.3 Kontekstuelle kilder og handlingsdyktige alarmer

Både signaturbasert og anomalitetsbasert deteksjon skaper alarmer som sikkerhetsanalytikere kan agere på for håndtering av hendelser. Metodene er derimot lite verdt om de ikke settes i riktig kontekst. Med andre ord vil ikke enhver event eller alarm tilsi at en hendelse eller sikkerhetshendelse har oppstått [4]. En hendelse er bygget opp av kontekstuelle kilder og handlingsdyktige alarmer. Handlingsdyktige alarmer er informasjon som er satt i kontekst og som muliggjør konstruktive handlinger, inkludert alarmer generert av forskjellige deteksjonssystemer og verktøy for hendelseshåndtering [4]

Kontekstuelle kilder

Eventer og andre datakilder faller inn under kategorien kontekstuelle kilder, eksempler på datakilder er applikasjon- og systemloggfiler, nettverkstrafikk, metadata og lagringsdisker. Eventer og slike kilder representerer gjerne objektive sannheter, men de må analyseres og forstås i henhold til kontekst før eventuelle alarmer genereres eller handlinger kan besluttes [4, s. 17–22].

Handlingsdyktige alarmer

Kontekstuelle kilder underbygger handlingsdyktige alarmer, for eksempel alarmer som genereres av forskjellige deteksjonssystemer, eller utdata fra maskinlæringsmodeller [4, s. 17–22]. Hvis sikkerhetsanalytikere kan gjøre konstruktive handlinger basert på informasjonen – for eksempel sette inn preventive tiltak – og informasjonen kommer i et format som er mulig å tolke og som støtter en beslutning, regnes det som handlingsdyktig informasjon [4, s. 156]. Slike alarmer vil fortsatt kreve en viss grad av analyse og vurdering opp i mot kontekst, blant annet grunnet risikoen for falske positiver.

Falske positiver

Falske positiver er en svært vanlig utfordring ved deteksjonssystemer og er et sentralt problem for automatisering av hendelseshåndtering. En falsk positiv er en alarm om unormal eller ondsinnet aktivitet som genereres av et deteksjonssystem, når det i realiteten ikke har oppstått en unormal eller ondsinnet situasjon [4, s. 136]. Slike alarmer er med andre ord ikke handlingsdyktige alarmer, men dette er ikke nødvendigvis mulig å bekrefte før etter analyse av alarmen. I tillegg til falske positiver finnes også sanne positiver, falske negativer og sanne negativer. Disse er forklart i matrisen i figur 2.6.

Sanne og falske positiver og negativer

Aktivitet / Alarm	Reell unormal eller ondsinnet aktivitet	Ingen unormal eller ondsinnet aktivitet
Alarm utløst	**Sann positiv** Systemet detekterte reelle unormale eller ondsinnede aktiviteter	**Falsk positiv** Systemet utløser alarm, men aktiviteten var ikke unormal eller ondsinnet
Alarm ikke utløst	**Falsk negativ** Noe unormalt eller ondsinnet har skjedd, men systemet detekterte det ikke	**Sann negativ** Normale og legitime aktiviteter utløser ikke alarm

Figur 2.6: Sanne og falske positiver og negativer [4, s. 136]

Problemet med falske positiver er vanskelig å løse grunnet et naturlig kompromiss ved mange deteksjonssystemer: et deteksjonssystem som tilstreber å detektere flest mulig sanne positiver medfører i svært mange tilfeller et stort antall falske positiver også [4, s. 136]. Et stort antall falske positiver skaper unødvendig arbeid for sikkerhetsanalytikere, og kan i verste fall føre til at alarmer ignoreres i sin helhet grunnet begrensede ressurser eller antagelser om at alarmer er falske. I tillegg gjør falske positiver implementeringen av automatisk hendelseshåndtering betraktelig mer utfordrende – eller umulig – på grunn av graden av usikkerhet som følger med alarmene. For å redusere sannsynligheten for falske positiver og muliggjøre hendelseshåndtering med automatiserte systemer, er det derfor viktig at kvaliteten på de kontekstuelle kildene og handlingsdyktige alarmene er så gode som mulig.

2.4.4 Nettverksmonitorering

Nettverksmonitorering har dype røtter innenfor hendelseshåndtering for deteksjon av eventer og hendelser, og det var lenge den dominante formen for deteksjon [4, s. 205]. Nettverksmonitorering fungerer ved å sette opp systemer plassert

på nettverket som aggregerer og observerer nettverkstrafikk, og deretter sammenligner denne trafikken opp i mot deteksjonslogikk som definerer hva som regnes som legitim, illegitim, normal eller unormal trafikk [4, s. 191–192]. Logikken kan være enten signaturbasert eller anomalitetsbasert. Når systemet detekterer aktivitet som defineres som unormal eller illegitim, utløses eventer som sendes til et sentralt monitoreringssystem eller direkte til en sikkerhetsanalytiker. Slike systemer kalles *Intrusion Detection Systems* (IDS). Når slike system benyttes for nettverksmonitorering brukes ofte begrepet NIDS (*Network Intrusion Detection Systems*). Dette skiller det fra den nært relaterte teknologien HIDS (*Host Intrusion Detection Systems*), der deteksjon gjennomføres direkte på endepunkter og ikke på nettverket. Dette beskriver vi nærmere i seksjon 2.4.5.

Nettverksmonitorering med respons

I nyere tid har rene deteksjonssystem som IDS gradvis blitt faset ut av aktiv bruk til fordel for systemer som ikke bare observerer, men som også kan utføre handlinger på bakgrunn av hva den detekterer [4, s. 205–208]. Slike systemer, kalt *Intrusion Prevention Systems* (IPS) eller *Network Intrusion Prevention Systems* (NIPS), kan automatisk terminere eller blokkere trafikk og annen ondsinnet aktivitet for å stoppe mulige sikkerhetshendelser tidligere, og dermed redusere de potensielle konsekvensene og kostnadene. IPS-teknologi har likevel lenge vært preget av redusert effektivitet, og responskapabiliteten de tilbyr blir ofte utnyttet i liten eller ingen grad grunnet risikoen for falske positiver [4, s. 205–208]. Verdien av å håndtere reelle trusler automatisk vil i mange tilfeller ikke utveie konsekvensene av å blokkere legitime handlinger eller systemer og dermed skape et selvpåført tjenestenektangrep.

I dag er det veldig vanlig at deteksjon og respons er samlet innunder samme verktøy, kalt *Intrusion Detection and Prevention Systems* (IDPS). Dedikerte systemer for deteksjon og respons for nettverk markedsføres ofte under navnet *Network Detection and Response* (NDR) [4, s. 206]. Teknologien regnes som en videreutvikling av tradisjonelle NIDS- og NIPS-verktøy, og i tillegg til deteksjon og respons tilbyr mange NDR-løsninger funksjoner som for eksempel automatisk skadevareanalyse som vi skal gå inn på i senere seksjoner. NDR og andre tilknyttede teknologier har også som mål å redusere antall falske positiver gjennom blant annet sterkere deteksjonslogikk på tvers av nettverket [63].

2.4.5 Deteksjon og respons for endepunkter

De siste tiårene har ressurser for deteksjon, analyse og respons blitt flyttet nærmere dit mange sikkerhetshendelser oppstår, nemlig på endepunktene. Først ved HIDS, og deretter HIPS, i likhet med utviklingen observert for nettverksmonitorering. Dette skifte, fra nesten utelukkende nettverksbasert deteksjon til en stor grad av endepunktsbasert deteksjon, var blant annet motivert av behovet for mer detaljert informasjon utover det nettverkslogger og nettverkstrafikk kan tilby [4, s. 195–196]. Data fra endepunkter er betraktelig mer nyttig ved for eksempel

endepunktsanalyse etter datainnbrudd. Likevel er det ideelt med en kombinasjon av både endepunktsbasert og nettverksbasert deteksjon, ettersom nettverkstrafikk gir overblikk over større hendelser som opptrer på tvers av enheter, og støtter analyse av hendelser der kilden til problemet eller berørte enheter ikke nødvendigvis er kjent [4, s. 195–196].

I likhet med nettverksmonitorering har det skjedd en utvikling i nyere tid, der tradisjonelle HIDS- og HIPS-verktøy er samlet innunder samme løsning, kalt EDR (*Endpoint Detection and Response*). EDR-løsninger detekterer og analyserer trusler på endepunkter, og kan også respondere automatisk til en rekke typer trusler [64, s. 7]. Den første EDR-løsningen dukket opp i 2013, med mål om å gi sikkerhetsanalytikere mer detaljert endepunktsdata til bruk under analyse av kompromitterte enheter. Deteksjonslogikken som benyttes i disse løsningene er utvidet til å inkludere både signaturbasert og anomalitetsbaserte indikatorer og maskinlæring brukes i stadig større grad [4, s. 198–202]. Utover deteksjon og respons støtter EDR håndteringen av hendelser ved innsamling av kontekstuelle datakilder og generering av handlingsdyktige alarmer. I likhet med NDR-løsninger kan mange EDR-løsninger også utføre skadevareanalyse på endepunktet.

Utvidet deteksjon og respons

Den nyeste utviklingen innenfor deteksjons- og responsteknologier er XDR (*Extended Detection and Response*) [64, s. 15–16]. Denne videreutviklingen av EDR utvider deteksjons- og responskapabiliteten til hele spekteret av en organisasjons infrastruktur, inkludert endepunkter, nettverk, skytjenester og eksterne datakilder [63]. Ved å samle sammen kapabilitetene av tidligere separate teknologier som EDR, NDR og skybaserte responsteknologier, har XDR som mål å bryte ned informasjonssiloer (informasjon som ikke deles på tvers av organisasjoner) og synliggjøre alle endepunkter, tjenester og data gjennom samme plattform. Slike løsninger kan analysere og korrelere informasjon fra flere datakilder for å øke sannsynligheten for deteksjon av trusler og muliggjøre automatiserte prosesser og håndtering på tvers av systemer.

Denne utviklingen har vært viktig for effektiviseringen av hendelseshåndtering. IBM rapporterer om at organisasjoner som hadde implementert XDR-løsninger som en del av sine forsvarsmekanismer responderte betraktelig raskere til hendelser sammenlignet med organisasjoner uten [57]. I tillegg er levetiden til datainnbrudd i gjennomsnitt 29 dager kortere. Disse tallene underbygger viktigheten av endepunktsbasert deteksjon og respons. Slike sikkerhetsløsninger vil bli viktigere for enhver organisasjon i årene som kommer, da kostnadene for sikkerhetsbrudd øker for hvert år [57].

2.4.6 Skadevareanalyse

Deteksjon av mistenkelige filer er en sentral del av både nettverks- og endepunktsmonitorering. Kompromittering av systemer, spesielt ved avanserte og vedvarende trusler, innebærer i svært mange tilfeller at skadevare leveres til et offersystem

for å utnytte en sårbarhet, åpne en bakdør eller på annet vis skade systemet. Deteksjonsløsninger vil kunne oppdage slike mistenkelige filer på vei til eller ved ankomst til systemet, enten angrepsvektoren er et e-postvedlegg, en nedlastbar fil, en minnepenn eller en filoverføring. Fra det tidspunktet starter analysen av den mistenkelige filen, kalt skadevareanalyse (eng: *malware analysis*). Dette er en teknikk for analyse av mistenkelige filer, der filer åpnes i isolerte «detonasjonskammer» for å undersøke hvilke handlinger som utføres når filen åpnes eller kjøres (detoneres), og for å vurdere hvorvidt den mistenkelige filen er ondsinnet eller ikke [4, s. 213].

Skadevareanalyse kan utføres manuelt innad i detonasjonsmiljøet (også kalt en *sandbox*), men i nyere tider er det vanligere at analysen gjennomføres automatisk av et dedikert system [4, s. 213]. Skadevareanalyse implementert som en automatisk prosess kan redusere analyse- og responstid betraktelig ved deteksjon av mistenkelige filer på endepunkter eller blant nettverkstrafikk. Mistenkelige filer kan analyseres så fort de oppdages, og avhengig av resultatet kan andre automatiske prosesser utløses for å respondere til den mulige skadevaren. Denne prosessen kan gjennomføres uten at et menneske trenger å involveres.

En annen fordel ved skadevareanalyse i forhold til tradisjonelle løsninger som IDPS-verktøy, er styrken det har når det kommer til deteksjon av nulldagssårbarheter og annen ukjent skadevare [4, s. 213]. Ettersom skadevareanalyse ikke ser etter definerte signaturer for å bekrefte hvorvidt en fil er ondsinnet, altså signaturbasert deteksjon, men heller analyserer hvilke faktiske handlinger som utføres når en mistenkelig fil detoneres. Slik kan den oppdage typiske unormale handlinger og på det viset detektere hittill ukjent skadevare. Skadevareanalyse faller altså innunder anomalitetsbasert deteksjon.

2.4.7 Loggadministrasjon, SIEM og SOAR

vi har redegjort for flere teknologier og løsninger som detekterer unormal og ondsinnet aktivitet, samler inn data og logger, utfører analyse, genererer alarmer og støtter visse automatiske responstiltak. Samtidig vil mange organisasjoner ønske, eller ha et behov for, å samle alle disse datakildene i en eller flere sentrale verktøy som komplimenterer eller utvider kapabilitetene til de nevnte løsningene, og effektiviserer hendelseshåndteringsarbeidet i en organisasjon [4, s. 241]. I denne seksjonen vil vi redegjøre for tre av de mest utbredte teknologiene innenfor denne kategorien av verktøy: verktøy for loggadministrasjons, verktøy for administrasjon av sikkerhetsinformasjon og -hendelser (SIEM), og verktøy for sikkerhetsorkestrering, -automatisering og -respons (SOAR).

Loggadministrasjon

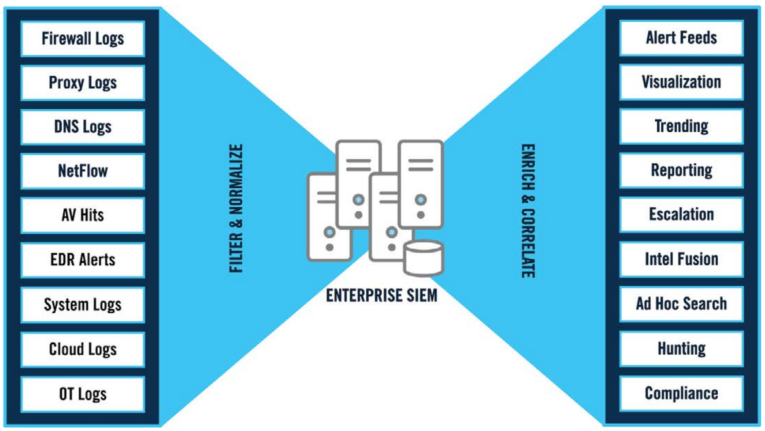
Et loggadministrasjonsverktøy er et verktøy for generell innsamling og aggregering av logger fra mange forskjellige enheter og systemer i en organisasjons infrastruktur [4, s. 252]. I tillegg støtter de fleste slike verktøy søk blant loggene, visualisering av data, generering av rapporter, oppsett av automatiske alarmer, og

annen funksjonalitet. Et loggadministrasjonsverktøy er vanligvis ikke spesialisert for håndtering av sikkerhetshendelser og mangler derfor ofte en del funksjonalitet rettet mot dette. På den andre siden er de vanligvis enklere å sette opp, egnet for en kombinasjon av både sikkerhetsrettet og ikke-sikkerhetsrettet bruk, og ofte billigere i drift enn store SIEM-verktøy (spesielt for organisasjonen som uansett har et loggadministrasjonsverktøy for generelt bruk). Det er derfor regnet som et godt sikkerhetsverktøy for mange organisasjoner som ønsker å bevege seg i retning av et SIEM [4, s. 252]. Arkitekturen til et loggadministrasjonsverktøy er ofte bygget opp av en eller flere sentrale noder, samt et større antall agenter spredt rundt på endepunkter og andre systemer [4, s. 246–247]. De sentrale nodene er der loggene samles og tilgjengeliggjøres for administrasjon i form av søk, visualisering, rapportering og oppsett av alarmer. Agentnodene vil ofte være enklere programvare installert og kjørende i bakgrunnen av enhetene der logger skal samles inn fra, som henter logger direkte fra råkilder og sender det tilbake til de sentraliserte nodene.

Administrasjon av sikkerhetsinformasjon og -hendelser (SIEM)

Security Information and Event Management (norsk: administrasjon av sikkerhetsinformasjon og -hendelser) er en teknologi for håndtering og behandling av sikkerhetsrelevant data i et system [65]. I likhet med typiske loggadministrasjonsverktøy vil et SIEM samle inn, aggregere, filtrere og lagre data fra utallige forskjellige kilder. Forskjellen mellom de to typene verktøy ligger i det faktum at SIEM-verktøy er rettet mot sikkerhetsarbeid. De har blant annet innebygget funksjonalitet for å analysere, kategorisere og korrelere store mengder sikkerhetsrelatert data for håndtering av hendelser og utførelse av sikkerhetsrelaterte oppgaver som for eksempel threat hunting [4, s. 243], som vi beskriver mer om i seksjon 2.4.9). SIEM-verktøy skal kunne aggregere eventer og annen data fra loggadministrasjonsverktøy, NDR, EDR og XDR-verktøy (se seksjon 2.4.4 og 2.4.5), eller fra direktekilder slik som brannmurlogger, systemlogger og logger fra operasjonell teknologi. Et overblikk over funksjonaliteten som er tilgjengelig i typiske SIEM-verktøy er vist i figur 2.7.

Et SIEM-verktøy skal først og fremst forenkle arbeidshverdagen til sikkerhetsanalytikere og andre som jobber med håndteringen av sikkerhetshendelser ved å fungere som et sentralt bibliotek av relevant sikkerhetsinformasjon og handlinger for hendelseshåndtering [65]. Verktøyet skal kunne analysere den store mengden data som aggregeres for å omgjøre kontekstuelle kilder til handlingsdyktige alarmer for sikkerhetsanalytikere og dermed muliggjøre mer effektiv hendelseshåndtering [4, s. 244]. Mer presist skal et SIEM understøtte hele arbeidsflyten under håndteringen av en hendelse: samle inn og prosessere store mengder data for å validere at en hendelse har skjedd; analysere data, finne kontekt, fjerne falske positiver og korrelere eventer til hendelser og rotårsak for å finne ut hva som har skjedd; generere saker (også kalt caser), rapportere, samt foreslå og gjennomføre tiltak for å finne ut hvordan hendelsen kan løses [4, s. 245]. Det de fleste SIEM



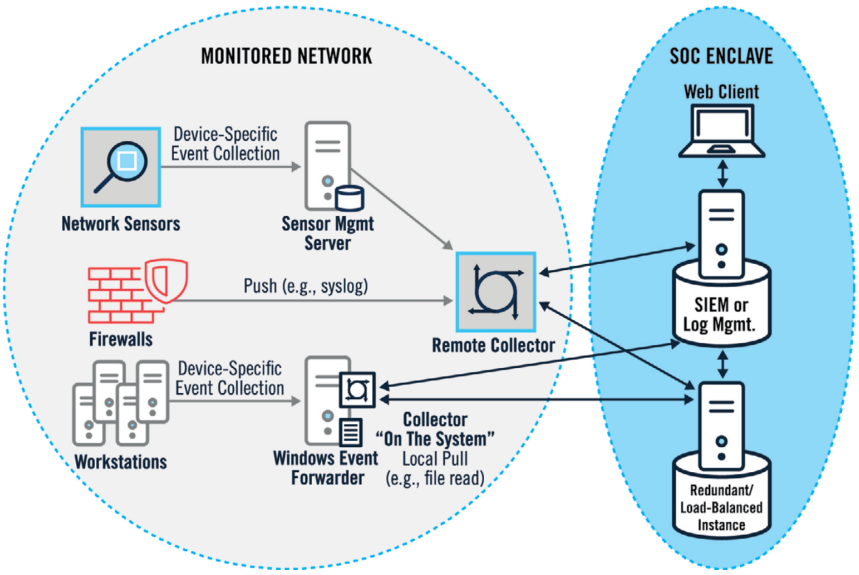
Figur 2.7: Overblikk over SIEM-funksjonalitet [4, s. 243]

verktøy *ikke* er spesialisert for, er helautomatisering og orkestrering av prosedyrene fra deteksjon til håndtering av hendelser. Slik teknologi er forholdsvis ny, og avhenger av en enda større grad av presis automatisering enn tidligere. Denne typen funksjonalitet er typisk forbeholdt SOAR-verktøy [65], som vi diskuterer i neste seksjon om disse verktøyene.

Arkitekturen bak et SIEM-verktøy er vanligvis lik arkitekturen bak et loggadministrasjonsverktøy [4, s. 246–251]. Det vil si sentraliserte noder der data samles inn og funksjonalitet tilgjengeliggjøres, og agentnoder spredt rundt direkte på enhetene (eller på dedikerte logginnsamlingsservere som står for innsamling av loggdata fra et subset av en organisasjons enheter). Figur 2.8 viser en forenklet modell av hvordan dette kan være satt opp, typisk for både SIEM- og loggadministrasjonsverktøy. Det er vanlig at agentnodene kan gjennomføre filtrering, *tuning* (synliggjøring av større mengder data), fjerning av duplikater og flytkontroll for å spare båndbredde, samt unngå å sende unødvendige eventer til de sentraliserte nodene. De sentrale nodene støtter vanligvis lagring av store mengder data spredt utover flere noder, har teknikker for å muliggjøre raske søk blant loggene som er samlet inn, og gir muligheten til å utføre avansert filtrering og visualisering av loggrunnlaget.

Sikkerhetsorkestrering, -automatisering og -respons (SOAR)

Security Orchestration, Automation and Response (norsk: sikkerhetsorkestrering, -automatisering og -respons) er en teknologi og serie av produkter som støtter opprettelsen av stegvise og sammenkoblede prosedyrer, kalt playbooks, for å orkestrere og automatisere handlinger under håndteringen av en hendelse [4, s. 270–273]. Nytteverdien til SOAR-verktøy ligger først og fremst i kombinasjonen av sikkerhetsorkestrering, -automatisering og -respons innenfor ett og samme verktøy [65]. Sikkerhetsorkestrering handler om å koordinere uavhengige prosedyrer og funksjonaliteten til flere forskjellige verktøy og tjenester, slik at disse kan in-



Figur 2.8: Arkitekturen til loggadministrasjon- og SIEM-verktøy [4, s. 247]

tegreres og jobbe sammen som et stort verktøy for å håndtere sikkerhetshendelser. Orkestrering handler også om muligheten til å styre alle disse delen fra en og samme plattform. Sikkerhetsautomatisering omhandler automatiseringen av disse prosedyrene og funksjonalitetene, med minimal eller ingen menneskelig involvering. Resultatet er derfor et verktøy og sett av funksjonaliteter som gir muligheten til å helautomatisere hele hendelseshåndteringsprosessen – fra deteksjon av unormale eller ondsinnede aktiviteter til hendelsen er under kontroll og normaltilstand gjenopprettet. Sikkerhetsrespons omhandler funksjonalitet for å respondere til hendelser med mitigerende tiltak og gjenopprettelse av normaltilstand. Samtidig vil forskjellige hendelser inneha ulike kompleksitetsnivå, og kreve forskjellige fremgangsmåter for å løse hendelsen. Derfor skal SOAR-verktøy muliggjøre både helautomatisering av enkle, repetitive prosedyrer ved kjente hendelser og feil, og en kombinasjon av automatiserte prosedyrer for informasjonsinnsamling, analyse og beslutningsstøtte ved feil som krever menneskelig involvering.

Det er mange likheter mellom SOAR og SIEM, samt SOAR og XDR [65]. Det er viktig å poengtere at hverken automatisert deteksjon, analyse eller håndtering av hendelser er ny teknologi kun forbeholdt SOAR-verktøy. XDR og forgjengerne EDR/NDR og IDPS (se 2.4.4 og 2.4.5) har lenge støttet både avansert deteksjonsteknologi, automatisk innhenting av data fra enheter, samt automatiske responstiltak ved eventer og hendelser. SIEM-verktøy, som tidligere diskutert, inneholder et bredt spekter av funksjonalitet for håndtering av sikkerhetshendelser med stor grad av automatisering [4, s. 270–273]. SOAR-verktøy vil også være avhengig av inndata i form av eventer og alarmer, samt datagrunnlag for innhenting av ytterligere informasjon og berikelse av saker, som det ofte får fra SIEM-, XDR- eller loggadministrasjonsverktøy. Hovedforskjellen ligger altså i SOAR-verktøyenes fo-

kus på orkestrering og tilgjengeliggjøring av alle disse funksjonalitetene i en og samme plattform og muligheten til å sømløst integrere de uavhengige prosedyrene og funksjonalitetene inn under samme automatiske prosedyrer.

2.4.8 Skybasert hendelseshåndtering

I nyere tid dukker stadig flere tjenester opp som skybaserte tjenester gjennom private og offentlige skytjenesteplattformer, og flere og flere organisasjoner flytter sine infrastrukturer inn i skyen. Vi kommer derfor ikke foruten at mange av sikkerhetsløsningene vi har diskutert hittil i dette underkapittelet også flytter inn i skyen. Mange av utfordringene med skybaserte løsninger er like de man har i on-prem-løsninger, som er infrastruktur som driftes av organisasjonen selv, men det er samtidig noen forskjeller. Et typisk Security Operation Center, vil ha like mye ansvar over organisasjonens skytjenester som de fysiske ressursene de eier [4, s. 225].

Skybaserte løsninger for hendelseshåndtering og informasjonssikkerhet har mange fellestrekk. Skybaserte løsninger for eksempelvis brannmur kommer som oftest med de samme utfordringene og egenskapene som deres on-prem motparter [226][4]. Skybaserte løsninger åpner opp for en rekke utfordringer. Typen ressurser og mengden i seg selv er et av de viktigste punktene med tanke på monitorering. Mengden ressurser for monitorering og deteksjon økes for skytjenester, gjerne raskere enn det et moderne SOC klarer å holde tritt med [4, s. 226–227]. Å finne en god balanse mellom skybaserte og on-prem-løsninger vil være et nøkkelpunkt for ethvert SOC i henhold til å sikre god hendelseshåndtering.

Mange av de vanligste kommersielle tjenestene for hendelseshåndtering kommer med versjoner for både on-prem og skybaserte løsninger. Eksempelvis finnes on-prem løsningen Splunk SOAR som vi har benyttet i vårt mulighetsstudie (se seksjon: 3.2) også som en skybasert løsning. Splunk SOAR for cloud gir på lik linje med on-prem løsningen muligheter for sikkerhetsorkestrering, automatisering og respons. Forskjellen er at løsningen blir levert som en Software-as-a-Service-løsning håndtert og vedlikeholdt av Splunk selv[66]. På samme måte finner man andre kommersielle løsninger fra andre organisasjoner. Microsoft har også en skybasert løsning for SIEM + SOAR kalt Azure Sentinel[5]. Azure Sentinel er en cloud-native løsning (teknologier og applikasjoner som er ut- viklet for installasjon og bruk i skybaserte miljøer og plattformer [10]) som bidrar med å levere sikkerhets- og trusselanalyse ved å gi en sammensatt plattform for alarmsdeteksjon, threat hunting og -respons.

2.4.9 Threat hunting og trusseletterretning

Hendelseshåndteringsprosessen kjennetegnes ofte som en reaktiv prosess i og med at man ofte *reagerer* på hendelser i form av alarmer som oppstår for å håndtere dem. Det finnes likevel flere metoder for å arbeide proaktivt, enten for å minske

[5] `https://learn.microsoft.com/en-us/azure/sentinel/overview`

sjansen for at hendelser skal oppstå, eller for å stille best mulig forberedt når de først inntreffer. Ved å betrakte cyberangrep gjennom alle faser – både før, under og etter kompromittering – har rammeverkene som mål om å avdekke og mitigere flere potensielle trusler og sårbarheter før hendelser oppstår.

Proaktive metoder for hendelseshåndtering er eksempelvis threat hunting, som har som hensikt å identifisere trusler før de kan skape hendelser og konsekvenser for en organisasjon [67]. Sikkerhetsanalytikere som aktivt søker, loggfører, monitorerer og nøytraliserer trusler før de kan skape seriøse problemer kalles noen ganger *cyber threat hunters*. Ifølge IBM kommer threat hunting i flere varianter: strukturert, ustrukturert og situasjonell hunting [67]. Strukturert hunting går etter indikatorer på angrep (IOA) og taktikker, teknikker og prosedyrer (TTP) brukt av en angriper, og alle hunting søk utføres strukturert på bakgrunn av disse. Ustrukturert hunting er som regel basert på en trigger som kommer fra en av flere IOC-er. Situasjonell threat hunting baseres på sikkerhetsanalytikers egen risikovurdering eller trender og sårbarheter som er gjeldende for systemene som benyttes.

Threat hunting vil ofte gjennomføres på bakgrunn av trusseletterretning. Begrepene er relativt like, men er viktig å skille fra hverandre. Trusseletterretning defineres av NIST som informasjon som er aggregert, transformert, analysert, tolket og beriket for å gi nødvendig kontekst for beslutningstakning [68]. Threat hunting kan også skape ny trusseletterretning.

2.4.10 Øvelser, penetrasjonstesting og sårbarhetsskanning

Tabletop-øvelser hvor man simulerer sikkerhetshendelser i kontrollerte miljø for å realistisk teste hendelseshåndteringsprosessen og identifisere forbedringsområder, er en annen metode for å jobbe proaktivt. En tabletop-øvelse er en strukturert diskusjon- og scenario-basert øvelse hvor personer med relevante roller og ansvarsområder møtes for å øve på og forbedre hendelseshåndteringsprosessen [4, s. 352–355].

Penetrasjonstesting beskrives av NIST som en metode for sikkerhetstesting hvor faktiske cyberangrep etterlignes [16]. En slik test gjennomføres vanligvis som et faktisk angrep på ekte systemer for at øvelsen skal fremgå så autentisk som mulig og for at organisasjonen skal stå best mulig forberedt ved faktiske angrep. Faktiske verktøy som benyttes av trusselaktører blir også benyttet.

Sårbarhetsskanning er en metode for å undersøke organisasjonens verdier, inkludert servere og endepunkter for kjente sårbarheter, utdatert programvare og manglende sikkerhetskonfigurasjon [4, s. 15]. Sårbarhetsskanning innen hendelseshåndtering er viktig for å begrense arbeidsflaten til organisasjonen, slik at sannsynligheten for vellykkede cyberangrep og potensielle konsekvenser reduseres.

2.4.11 Kunstig intelligens i metoder og verktøy for hendelseshåndtering

Som nevnt i seksjon 2.4.9, har hendelseshåndteringsprosessen utviklet seg til å bli en stadig mer proaktiv prosess i løpet av nyere tid. Deler av denne utviklingen kan kobles til at bruken av kunstig intelligens, heretter maskinlæring (se seksjon 2.1.13), innenfor hendelseshåndtering har økt. Verktøy og tjenester som benytter seg av maskinlæring vil i større grad enn før kunne avlaste mennesker for repeterende og enkle arbeidsoppgaver, i tillegg til at modellene kan behandle svært store mengder data innenfor kort tid, samt korrelere sammenhenger mellom store mengder eventer [69].

Forskjellige metoder og bruksområder krever forskjellige typer maskinlæring, hvor valget gjøres basert på hva slags maskinlæringsmodell det er snakk om, og hva disse modellene er gode på. Kort forklart skilles det mellom overvåket og uovervåket maskinlæring (eng: *supervised and unsupervised machine learning*). Overvåkede maskinlæringsmodeller krever menneskelig interaksjon ved at inndata og utdata markeres med etiketter (eng: *labeling*) [70]. Dette gjør at slike modeller blir mer presise, og modellene blir bedre på å løse oppgaver der både inndata og forventet resultat er kjent. Derfor er denne typen maskinlæringsmodeller egnet for bruk i for eksempel signaturdeteksjon. Uovervåket maskinlæring vil på den andre siden ta i bruk komplekse algoritmer for å analysere umerket inndata og forutsi ønsket resultat. Denne typen behøver ikke kjenne til inndatasettet på forhånd for å løse problemet, men det skaper samtidig større usikkerhet rundt presisjonen og riktigheten ved løsningen. Dette gjør uovervåket maskinlæring nyttig ved for eksempel anomalitetsdeteksjon. Mange modeller benytter en kombinasjon av både overvåket og uovervåket maskinlæring, kalt semi-overvåket maskinlæring, som trenes på datasett med både merket og umerket data.

I tillegg til signaturbasert og anomalitetsbasert deteksjon, som redegjort for i seksjon 2.4.2, finnes det flere andre bruksområder for maskinlæring innenfor hendelseshåndtering. Noen verktøy og tjenester har som mål å muliggjøre proaktiv hendelseshåndtering ved at maskinlæringsmodellene kan forutsi hendelser før de oppstår basert på anomaliteter i trafikk og eventer [69]. Andre tjenester retter seg mot å begrense den menneskelige arbeidsmengden, ved å aggregere og analysere store mengder eventer, og deretter håndtere oppgaver som er enkle og repeterende, og korrelere hvilke eventer som sannsynligvis hører sammen under samme hendelse. IBM rapporterer om at maskinlæring kan redusere mengden alarmer som må behandles av mennesker med 50 prosent [69]. De nevner også at maskinlæring kan redusere tiden brukt på å behandle falske positiver med 80 prosent.

Korrelering av mange eventer til felles hendelse og rotårsak kalles for alarmgruppering (eng: *alert grouping*) [64, s. 20]. Én enkelt hendelse kan utløse mange forskjellige eventer, ofte i flere forskjellige systemer og tjenester. Alarmgruppering fungerer ved at en maskinlæringsmodell utfører klyngeanalyse på et datasett av eventer, for å se sammenhenger mellom eventene og korrelere eventer som trolig

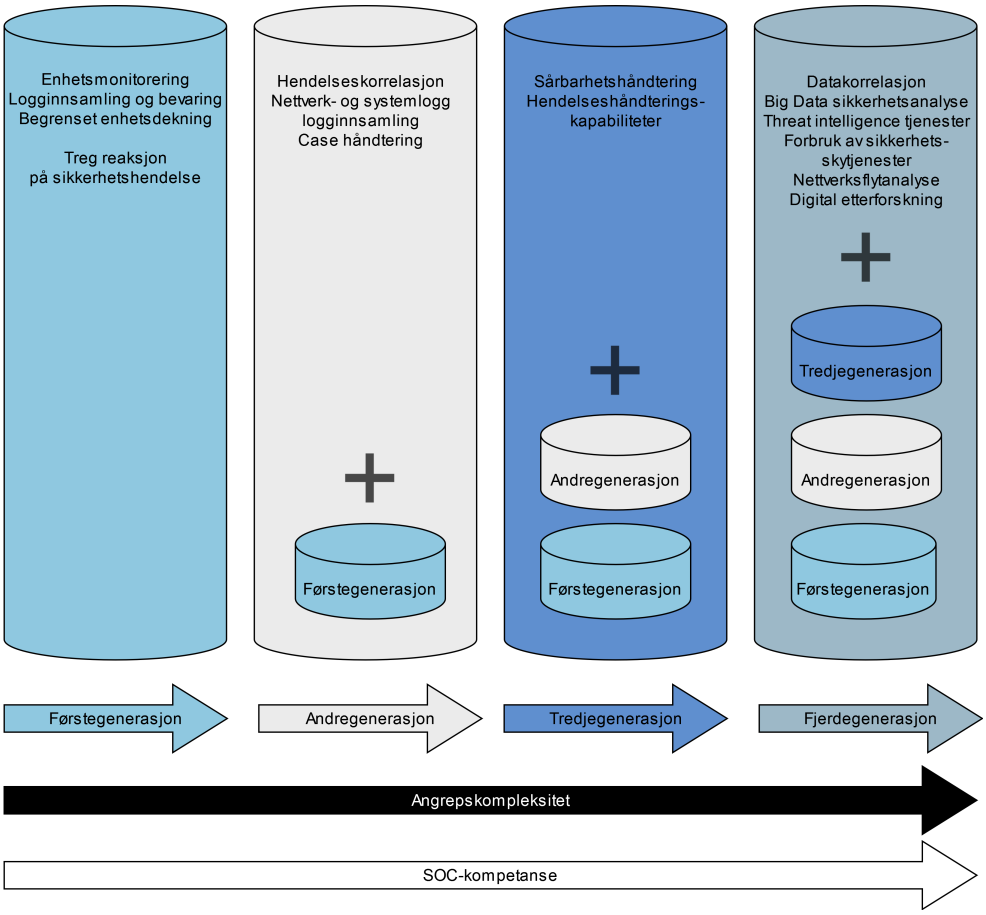
har samme rotårsak til én enkelt hendelse [71]. Dette reduserer det totale antallet eventer som må behandles av mennesker, i tillegg til å frigjøre tid for å prioritere unike eventer og hendelser som krever størst oppmerksomhet. I tillegg er det mulig å benytte klyngeanalyse for å oppdage alvorlige hendelser og potensielt nye sårbarheter og trusler raskere. Dette oppnås ved å benytte en lignende maskinlæringsmodell som igjen utfører klyngeanalyse på en stor mengde eventer, men istedenfor å kun gruppere eventer innunder felles hendelse og rotårsak, vil modellen lete etter og hente ut anomaliteter blant eventene. Det vil si eventer som oppstår i mindretall, eventer som oppstår sjeldent (eller som ikke har blitt detektert tidligere), og eventer som skiller seg fra normalsituasjonen. Slike eventer kan være indikatorer på hendelser som er i ferd med å oppstå, eller nye – potensielt alvorlige – hendelser som krever ekstra oppmerksomhet. Dette er hendelser som er svært vanskelig for et menneske å oppdage ved manuell gjennomgang av et store antall eventer.

2.5 SOC - Utvikling og utfordringer

Med bakgrunn i et stadig økende trussellandskap blir det vanskeligere å beskytte konfidensialiteten, integriteten og tilgjengeligheten til organisasjoners data og IT-systemer [72]. I forrige seksjon ble det redgjort for moderne metoder og teknologier som har som formål å effektivisere hendelseshåndtering. For å være i stand til å håndtere flere og mer avanserte sikkerhetshendelser er det derfor viktig at organisasjoners sikkerhetsavdelinger tilpasser seg det skiftende trussellandskapet.

Historisk sett, kan SOC-utviklingen de siste årene bli delt inn i flere generasjoner [73]. I begynnelsen var det hovedsaklig fokus på å detektere et angrep i rekognoseringsfasen, i tillegg til å detektere og respondere når organisasjonen var under direkte angrep. Etterhvert som trussellandskapet har utviklet seg, har også SOC-et hatt et behov for å utvikle seg. I moderne tid, bør et SOC ideelt sett være i stand til å kartlegge og predikere relevante trusselaktører, slik at angrepsvektorene kan mitigeres, eller stoppes før angrepet tar sted. I det minste, bør et SOC ideelt sett være i stand til å detektere en hendelse før skadeomfanget blir for stort [4, s. 23]. Figur 2.9 er hentet fra boken *Security Operations Center: Building, Operating, and Maintaining your SOC* [73, s. 21], utgitt av Cisco, og viser utviklingen av SOC frem til og med 2015 gjennom fire generasjoner. Selv om mye har skjedd siden 2015, er figuren fortsatt relevant for å et innblikk av hvordan fokuset til et SOC har skiftet seg historisk.

Som vist i figuren, hadde et første-generasjons SOC hovedfokus på monitoring. Dette var i hovedsak enhet- og nettverksmonitorering med formål om å opprettholde god tilgjengelighet. Videre ansvarsoppgaver inkluderte håndtering av antivirus-alarmer, samt innhenting av relevant loggdata. Det hadde på denne tiden ikke blitt etablert et formell SOC, noe som resulterte at ansvaret ofte var delt mellom flere ansatte som jobbet operasjonelt med IT. Med bakgrunn i dette, var håndteringen av hendelser ofte ineffektiv. Etterhvert som utviklingen beveget seg inn i den andre generasjonen, ble SIEM-løsninger implementert i ulike



Figur 2.9: SOC utvikling frem til 2015 gjennom fire generasjoner [73]

SOC. Sanntidsanalyse av loggdata effektiviserte hendelseshåndteringen og reduserte behovet for manuell analyse av store loggkilder. Dette gjorde det mulig for analytikere å fokusere på raskere deteksjon av ukjente hendelser. I tredje og fjerde generasjon begynte mer avanserte verktøy å nå markedet. Videre utvidet arbeidsoppgavene seg til å inkludere sårbarhetshåndtering i tredje generasjon, og økt trusseletterretning gjennom fjerde generasjon. I tillegg ble det rettet mer fokus mot å styrke organisasjoners motstandsdyktighet ved implementering av en forsvar-i-dybden-arkitektur [73, s. 22–24]. Forsvar-i-dybden-arkitektur beskrives av NIST som en informasjonssikkerhetsstrategi som integrerer mennesker, teknologi og operasjonsevner for å etablere forskjellige barrierer på tvers av lag og oppdrag i organisasjonen [16].

I moderne tid, har det stadig blitt behov for flere SOC-funksjoner, blant annet threat hunting, analyse av skadevare og digital etterforskning [4, s. 319]. Likevel, vil funksjonene som utføres i et SOC variere fra organisasjon til organisasjon. Funksjonene kan deles inn i forskjellige ansvarsområder og det er flere faktorer som påvirker hvilke ansvarsområder som er relevante for et SOC, inkludert organisasjonens størrelse, modenhet og SOC-ressurser [4, s. 13]. Uavhengig av behov, er det fremdeles noen kjerneoppgaver som bør være på plass i ethvert SOC, nemlig oppgaver som har som formål å identifisere og respondere på mulige cyberangrep [4, s. 17]. Vi skal ikke gå videre i detalj på de forskjellige SOC funksjonene i denne rapporten.

Avsnittene ovenfor har gjort rede for den historiske utviklingen av et SOC, der formålet hele tiden har vært å forbedre og effektivisere hendelseshåndtering. Likevel er det fremdeles mange SOC-utfordringer den dag i dag. Det er naturligvis ulike utfordringer for ulike SOC [74]. De resterende avsnittene skal dermed redgjøre for noen generelle utfordringer i lys av PPT-modellen.

2.5.1 Utfordringer relatert til mennesker (People)

En av utfordringene, relatert til den menneskelige involveringen i et SOC, er antall falske positiver som må håndteres. Enhver alarm trenger en initiell vurdering av en SOC-analytiker for å vurdere autentisiteten til alarmen. Å vurdere om en alarm er en falsk positiv, eller ikke, kan være lettere sagt en gjort, og er avhengig av hvilken type alarm det er. Tiden som må medregnes kan derfor variere fra noen minutter til noen timer [74]. Med andre ord er dette veldig repeterende arbeidsoppgaver, noe som flere andre studier også viser til [75, 76].

Videre kan et SOC sitt ansvar overfor organisasjonen være en utfordring for sikkerhetsanalytikere. Analytikerne må ofte jobbe under press og analysere store mengder data, der en feil kan medføre store konsekvenser. Det mentale, i kombinasjon med mye repeterende arbeid, kan medføre at en analytiker blir *låst* i det samme arbeidsmønstre over lengre tid. Dette kan resultere i utbrenthet for analytikeren, eller gjøre det vanskeligere for organisasjonen å holde på kvalifisert personell [74, 76].

En annen viktig utfordring relatert til den menneskelige involveringen i et

SOC, er utfordringen ved å finne riktig personell med nødvendige kompetanse [74]. *International Information System Security Certification Consortium* har gjennom sin «Cybersecurity Workforce Study» [77] identifisert et manglende gap innenfor cybersikkerhet på 3.4 millioner ansatte globalt.

2.5.2 Utfordringer relatert til prosess (Process)

Utfordringer relatert til prosesser i et SOC, er ofte knyttet til manglende overordnet forståelse av prosessen. Dette kan medføre feil fokus, da man ikke vet hvilke aspekter som bør fokuseres på [74]. Et eksempel på dette kan være å utvikle eller forbedre teknologi som man anseer som viktig, uten å vite at det finnes andre oppgaver hos SOC-et som haster mer.

2.5.3 Utfordringer relatert til teknologi (Technology)

Det blir stadig vanskeligere å identifisere og detektere nye trusler og hendelser, da den digitale infrastrukturen stadig blir mer komplekst [74]. Dette skaper utfordringer med å danne situasjonsforståelse for analytikerne som må håndtere alle alarmene. Det er mange forskjellige teknologier og verktøy på markedet, som har som formål å korrelere og forenkle håndteringen av alarmer. Likevel, behøver ikke problemene å forsvinne dersom nye verktøy implementeres. I noen tilfeller kan situasjonen forverres, dersom verktøyene feilkonfigureres [74]. Dette kan resultere i flere falske positiver, som vil være negativt med tanke på utfordringene påpekt tidligere. I tillegg kreves det mye ressurser for å vedlikeholde et verktøy [78]. Samtidig kan innføringen av nye verktøy føre til andre utfordringer, da noen verktøy kan lide av dårlig brukervennlighet og regelmessig funksjonsfeil [79].

En økning i antall loggkilder kan være en annen utfordring ved innføring av ny teknologi og verktøy. Dette medfører økt belastning for analytikerne som er nødt til å analysere disse eventene [74]. Kvaliteten til disse loggene er avhengig av teknologien og verktøyene som er brukt, som kan by på utfordringer relatert til analyse og forståelse. Mange arbeidsoppgaver til et SOC er preget av manuelt arbeid, slik som threat hunting, undersøkelse av alarmer og håndtering av hendelser. Økt automatisering kan bidra til å forenkle arbeidet til en sikkerhetsanalytiker, men selv om det finnes mange rapporter på temaet er det ikke gjort tilstrekkelig med brukertester til å si om det fungerer i praksis [74]. Det finnes flere forskjellige tilnærminger som er undersøkt, de som benytter maskinlæringsteknikker har en tendens til å produsere et høyt antall falske positiver som krever undersøkelse fra sikkerhetsanalytikere [74].

2.6 Relevant forskning

Det er mange tidligere forskningsartikler, studier og faglitteratur som har tatt opp tema automatisering innenfor hendelseshåndtering. Blant disse finner vi større og

mer kjente publikasjoner som *11 Strategies of a World-Class Cybersecurity Operations Center* [4] utgitt av *MITRE*, og mindre kjente – men anerkjente og innsiktsrike – artikler som *The Evolution of Security Operations and Strategies for Building an Effective SOC* utgitt av *CISA*. Disse utgivelsene går spesifikt inn på tematikk og strategier rundt hvordan et SOC kan bygges og driftes på best mulig måte, blant annet ved bruk av automatiseringsmetoder.

Problemstillingen er motivert av flere rapporter og andre utgivelser som diskuterer det raskt endrende digitale trusselbilde. Blant disse finner vi *NSM* årlige risikorapporter [18, 34, 80], *ENISA*s *Threat Landscape* [36] og CISCOs *The modern cybersecurity landscape-* og *Year in Review*-rapporter [35, 51].

Vi har i løpet av oppgaven brukt en rekke kvalitative og kvantitative undersøkelser og statistiske rapporter, for eksempel *IBM* sine *Threat Index-* og *Cost of a Data Breach*-rapporter [46, 47, 57], Deep Instinct sine *Voice of SecOps*-rapporter [58, 81] og *Information security incident management: Current practice as reported in the literature* [40]. Disse underbygger teorien presentert i oppgaven, og motiverer problemstilling og forskningsspørsmål. Disse artiklene inneholder statistiske data, trender og innsyn relevant for hendelseshåndtering og informasjonssikkerhet.

Relevant forskning om livssyklusen til et cyberangrep og trusselaktørers handlinger og taktikker før, under og etter gjennomføringen av et cyberangrep inkluderer eksempelvis *Cyber Kill Chain* [6] og *MITRE ATT&CK* [22]. Disse kildene referer i stor grad til moderne metoder for hendelseshåndtering. Hendelseshåndteringsprosessen i sin helhet er beskrevet i rapporten med utgangspunkt i *NIST* sin *Computer Security Incident Handling Guide* [15], ISO 27035 [39] og *ENISAs Good Practice Guide for Incident Management* [48]. Disse går i dybden på de forskjellige fasene av hendelseshåndteringsprosessen. I tillegg diskuterer vi hendelseshåndtering i lys av rammeverket *Information Technology Infrastructure Library (ITIL)* versjon 4 [41]. Dette rammeverket diskuterer hendesleshåndtering med fokus på prosess og styring. Sentrale konsepter rundt risikostyring benytter blant annet boken *Information Risk Management* av David Sutton [37].

Oppgaven tar for seg et utvalg av sentral teori, samt definisjoner og terminologi. Som informasjonskilder for dette benyttes blant annet rapporter, fagartikler og oppslagsverk fra organisasjoner som *NIST*, *NSM*, *IBM*, *CISCO*, *CrowdStrike* og *Institute of Electrical and Electronics Engineers (IEEE)*. I tillegg er flere av standardene i ISO 27000-serien tatt i bruk for å støtte opp under faglige termer og forklaringer. For metodedeler og utførelse av dette er dokumentasjonssider og relevante rapporter fra organisasjoner som *Splunk*, *Microsoft* og andre benyttet for å enklere gjennomføre oppsett og utførelse av det praktiske arbeidet. På et generelt nivå er boken *Bacheloroppgaven* [82] brukt som assistanse under disponering og oppsett av oppgavens helhetlige struktur og utforming, og som referanseverk under skriving av bacheloroppgaven.

Lignende rapporter, studier og forskningsartikler som tar opp lignende tema som de som diskuteres i denne oppgaven inkluderer bacheloroppgaven *SOAR Playbook Implementation - Incident Deduplication and Its Effects* [83], masteropp-

gaven *Correlating IDS alerts with system logs by means of a network-centric SIEM solution* [84], doktorgradsavhandlingen *Architecture-centric support for security orchestration and automation* [85] og artikkelene *SIEM integration with SOAR* [86] og *Artificial Intelligence based Security Orchestration, Automation and Response System* [87].

Kapittel 3

Metode

I metodedelen av rapporten vil vi redegjøre for hvordan vi gikk frem for å gjennomføre den praktiske delen av oppgaven. For å besvare oppgavens problemstilling har vi gjennomført dybdeintervju av kandidater med erfaring og kompetanse innenfor hendelseshåndtering. Vi har også gjennomført et mulighetsstudie for effektivisering av hendelseshåndtering med konseptbevis. I seksjon 3.1 beskriver vi dybdeintervjuene, herunder hensikt, struktur, utvalg, analyse og gjennomføring. I seksjon 3.2 redegjør vi for mulighetsstudie, herunder hensikt, analyse, gjennomføring, oppsett av testmiljø og de forskjellige casene som skal utredes.

3.1 Kvalitativ undersøkelse

3.1.1 Hensikt

Den første vitenskapelige forskningsmetoden som ble brukt under oppgaven er semistrukturerte samtaler i form av kvalitative dybdeintervju. Vi valgte å gjennomføre kvalitative intervju til fordel for kvantitative, da vi ikke ønsket enkle svar i form av statistikk, men heller et mer konsentrert utvalg av meninger fra fagfolk med bred forståelse, kompetanse og erfaringer. Vi gjennomførte individuelle semistrukturerte intervju. Gjennom slike intervju kunne vi stille oppfølgingsspørsmål og få hver respondent sine erfaringer i større detalj. Individuelle dybdeintervju gir større spillerom til hver respondent, fremfor fokusgrupper der enkeltindivider naturlig vil trekke seg tilbake i større grad.

3.1.2 Struktur

Den kvalitative undersøkelsen fulgte en semistrukturert form. Respondentene ble stilt de samme hovedspørsmålene, men de ble stilt forskjellige oppfølgingsspørsmål avhengig av undertemaene som ble trukket frem. Spørsmålene ble ikke nødvendigvis stilt i samme rekkefølge for hvert intervju. Intervjuene bestod av respondenten, en hovedintervjuer og en referent, mens de resterende deltakerene var observatører. Alle bidro med å stille oppfølgingsspørsmål der de anså dette

som hensiktsmessig. Spørsmålene ble utarbeidet med bakgrunn i oppgavens problemstilling. Spørsmålene tok opp tema relatert til effektivisering av hendelseshåndtering og prosesser som støtter oppunder dette. For å sikre åpenhet i spørsmålene og fjerne rom for tolkning, unngikk vi partiskhet og ledende spørsmål. Oppdragsgiver og veileder ble benyttet aktivt i prosessen for kvalitetssikring av både innhold og lengde, samt at spørsmålene ikke var partiske eller ledende. Det ble gjort opptak av hvert enkelt intervju, som ble transkribert før analyse.

Intervjuguide og samtykkeskjema kan sees i vedlegg A og B. Sikt sin informasjonsskrivmal for forskningsprosjekt[1] ble brukt for samtykkeskjema. På bakgrunn av anbefalinger fra Sikt og NTNU, forsikret vi deltakerne om at de kunne trekke sitt samtykket når som helst, både før, under og etter intervjuene.

3.1.3 Utvalg

Utvalget besto av fem kandidater fra Norsk helsenett. Samtlige kandidater er aktivt involvert i hendelseshåndtering, og det ble gjennomført et strategisk utvalg for å dekke alle faser av hendelseshåndteringsprosessen.

3.1.4 Gjennomføring

Samtlige intervju ble gjennomført fysisk. Intervjuene bestod av fire deler. Før intervjuene startet, ble en kort oppsummering av oppgavens formål og problemstilling presentert. Deretter fikk kandidatene spørsmål om stilling, bakgrunn og ansiennitet, før intervjuene gikk inn på oppgavens hovedspørsmål. Til slutt fikk samtlige deltakere mulighet til å presentere eventuelle tanker og andre spørsmål.

3.1.5 Analyse

Analysen av dybdeintervjuene startet med transkribering av opptakene gjort under gjennomførelsen. Deretter ble innholdet i transkriberingene kvalitetssikret, og eventuelle uklarheter ble fulgt opp med respondentene. Til slutt ble transkriberingene kodet og kategorisert etter tema. Kodene ble deretter gruppert i hovedtema som dannet grunnlag for den endelige inndelingen av resultatene. Resultatetene ble tilsendt respondentene, oppdragsgiver og oppgaveveileder for å sikre validitet.

3.2 Mulighetsstudie

3.2.1 Hensikt

Den andre forskningsmetoden som ble valgt for denne oppgaven var et mulighetsstudie. Gjennom mulighetsstudien har vi utarbeidet et konseptbevis med tre brukstester som utforsker potensielle løsninger som kan gi svar til problemstilling

[1] `https://sikt.no/informasjon-til-deltakarane-i-forskingsprosjekt`

og forskningsspørsmål. Vi valgte en mulighetsstudie fordi vi ønsket å undersøke flere muligheter for effektivisering, samt samle inn praktiske erfaringer, fremfor å teste en definert hypotese.

3.2.2 Analyse

Etter gjennomføring av brukstester ble resultatet dokumentert, strukturert og kvalitetssikret. Deretter analyserte vi utfallet av testene, og knyttet dette opp mot problemstillingen for å vurdere brukervennlighet og egnethet.

3.2.3 Verktøy og produkter brukt under mulighetsstudie

For å demonstrere effektivisering av hendelseshåndtering har vi i denne rapporten satt opp et testmiljø for gjennomføring av brukstester. Produkter og verktøy brukt i dette testmiljøet er valgt med bakgrunn i hva som var hensiktsmessig for oppgavens gjennomføring, men for hvert valgte produkt finnes det mange aktuelle alternativ for organisasjoner som ønsker å sette opp et lignende miljø. Følgende underseksjoner redegjør kort for de produktene og verktøyene vi brukte gjennom mulighetsstudien.

Microsoft Azure

Microsoft Azure er en komplett skyplattform som leverer applikasjoner, plattformer og infrastruktur som en tjeneste (SaaS, PaaS og IaaS), gjennom et bredt utvalg av tjenester fra både Microsoft og andre tredjeparter[2]. Microsoft Azure leverer tjenester gjennom datasentre plassert over hele verden, og er en av de største skyplattformene i verden med en markedsandel på over 20 prosent globalt [88]. Alternativer til Microsoft Azure inkluderer Amazon Web Services (AWS), Google Cloud Platform (GCP) og Openstack. For denne oppgaven ble Microsoft Azure valgt grunnet tilgjengelighet gjennom oppdragsgiver og at gruppen hadde kompetanse og erfaring med bruk av skyplattformen.

Sophos Firewall

Sophos Firewall er en brannmurløsning som tilbyr helhetlig beskyttelse ved hjelp av blant annet innbruddsdeteksjon og -beskyttelse (IDPS), trafikkaggregering, logging og nettverksregler. Sophos Firewall er utviklet av Sophos Group og spesialiserer seg på bruk i skyplattformer. Tjenesten er tilgjengelig for installasjon og oppsett direkte gjennom Microsoft Azure[3]. Andre alternative brannmurløsninger som kunne ha blitt benyttet som en del av vårt Microsoft Azure-miljø, inkluderer løsninger levert av Cisco, Fortinet, Palo Alto Networks og Checkpoint. Vi valgte

[2] `https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-azure/`

[3] `https://azuremarketplace.microsoft.com/nb-no/marketplace/apps/sophos.sophos-xg-firewall-solution?tab=Overview`

Sophos Firewall for vårt miljø ettersom det tilbydde tilstrekkelig funksjonalitet for oppgavens formål i forhold til kostnad og kompleksitet ved oppsett.

Splunk Enterprise

Splunk Enterprise er et loggadministrasjonsverktøy for dataaggregering, -søk, -analyse og -visualisering[4]. Splunk Enterprise kan samle data fra en rekke forskjellige kilder, blant annet gjennom tredjepartsapplikasjoner eller direkte tilkoblede rådatakilder. Deretter tilgjengeliggjøres dataen i en samlet plattform for behandling, samt automatisk generering av eventer og varsler.

Splunk Enterprise er i seg selv ikke regnet som et verktøy for administrasjon av sikkerhetsinformasjon og -hendelser (SIEM), men det inneholder funksjonalitet som muliggjør mye av den samme funksjonaliteten som et tradisjonelt SIEM-verktøy. Splunk har utviklet Splunk Enterprise Security[5] som en tilleggsapplikasjon til Splunk Enterprise-plattformen, som gir fullstendig SIEM-funksjonalitet. Splunk Enterprise Security og andre komplette SIEM-verktøy er utenfor omfanget av denne oppgaven. Denne oppgaven fokuserer derfor på bruken av loggadministrasjons- og SOAR-verktøy for orkestrering og automatisering av hendelseshåndtering (se neste underseksjon). For ytterligere beskrivelser av loggadministrasjon og SIEM-verktøy, se seksjon 2.4.7.

Splunk Enterprise ble valgt som loggadministrasjonsverktøy på grunn av eksisterende kompetanse blant gruppemedlemmer og oppdragsgiver. I tillegg dekket verktøyet alt som var nødvendig for gjennomføring av oppgaven, inkludert en plattform for innsamling og lagring av logger, og generering av søk og alarmer. Alternative verktøy for loggadministrering eller SIEM-funksjonalitet inkluderer Palo Alto Networks XSIAM, SolarWinds SIEM, Fortinet FortiSIEM, Rapid7 InsightIDR, IBM QRadar SIEM og Microsoft Azure Sentinel.

Splunk SOAR

Splunk SOAR er et verktøy for sikkerhetsorkestrering, automatisering og respons[6]. SOAR-verktøy mottar eventer og annen data fra en rekke forskjellige kilder, for eksempel Splunk Enterprise, Sophos Firewall eller direkte fra endepunkter. Splunk SOAR integrerer funksjonalitet og handlinger fra en rekke forskjellige tjenester og orkestrerer disse innunder samme plattform for å skape automatiske prosedyrer for å håndtere hendelser. Splunk SOAR er designet for å kunne benyttes sammen med et stort antall forskjellige produkter og tjenester fra mange forskjellige leverandører. For ytterligere beskrivelser av SOAR-verktøy, se seksjon 2.4.7.

Vi valgte å benytte Splunk SOAR i denne oppgaven, ettersom det var det eneste produktet vi fant som tilbydde en åpen prøveversjon med tilstrekkelig funksjonalitet for oppgavens gjennomføring. Alternative produkter innenfor SOAR-

[4] `https://docs.splunk.com/Documentation/Splunk/9.0.4/Overview/AboutSplunkEnterprise`

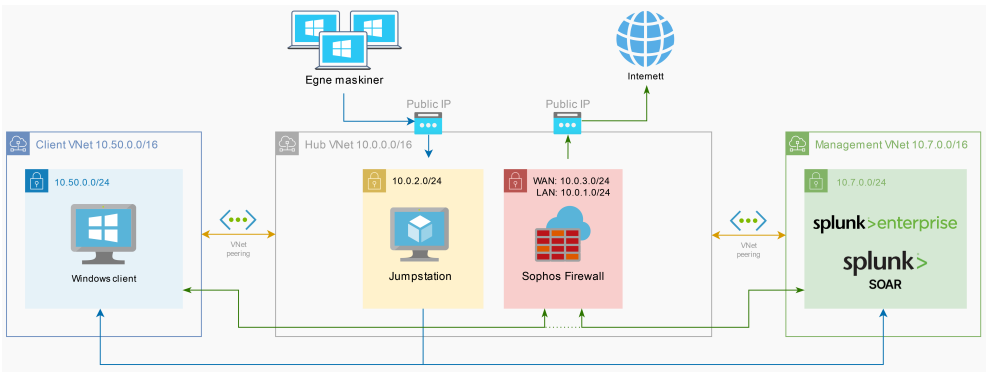
[5] `https://docs.splunk.com/Documentation/ES/7.1.0/User/Overview`

[6] `https://www.splunk.com/en_us/data-insider/what-is-soar.html`

segmentet er Palo Alto Networks XSOAR, IBM QRadar SOAR, Swimlane SOAR, Rapid7 Insight Connect, Google Chronicle, Fortinet FortiSOAR og Microsoft Azure Sentinel.

3.2.4 Oppsett av testmiljø

For å sette opp testmiljøet for konseptbeviset ble Microsoft Azure valgt som infrastrukturteknologi, dette ga fleksibilitet og skalerbarhet som forenklet oppsettet. Figur 3.1 viser en oversikt over topologien til testmiljøet. Figuren viser de tre virtuelle nettverkene, de virtuelle maskinene og tjenestene, og sammenkoblingen mellom disse. Denne rapporten fokuserer på oppsett, konfigurasjon og testing av verktøy for hendelseshåndtering, og beskriver ikke i detalj oppsettet av den underbyggende infrastrukturen i Microsoft Azure, slik som virtuelle nettverk og virtuelle maskiner. For å koble til testmiljøet i Microsoft Azure benyttet vi en *jumpstation*, en virtuell maskin som er tilgjengelig fra utenfor det interne testmiljøet som videre benyttes for tilkobling til de andre maskinene i testmiljøet. Oppsett av Jumpstation er diskutert i vedlegg C.1.



Figur 3.1: Topologi

Videre bestod testmiljøet av en brannmur, en Windows-klient og en Linux-server som kjørte Splunk Enterprise og Splunk SOAR. Brannmuren aggregerte og logget trafikk som ble sendt mellom klienten og Linux-serveren, samt trafikk mellom maskinene og internett. Brannmurloggene ble benyttet for å generere events og varsler som håndteres av enhetene. Oppsett og konfigurasjon av Sophos Firewall er diskutert i vedlegg C.2. På Linux-serveren installerte vi Splunk Enterprise for datainnsamling og generering av events og varsler, og Splunk SOAR som verktøy for orkestrering og automatisering av hendelseshåndtering. Oppsett og konfigurasjon av Linux-serveren, Splunk Enterprise og Splunk SOAR er henholdsvis diskutert i vedlegg C.3, C.4 og C.5. Windows-maskinen ble satt opp for å simulere en vanlig bruker og klientendepunkt som blir utsatt for eller forårsaker diverse eventer og hendelser. Oppsett av Windows-klient er diskutert i vedlegg C.6.

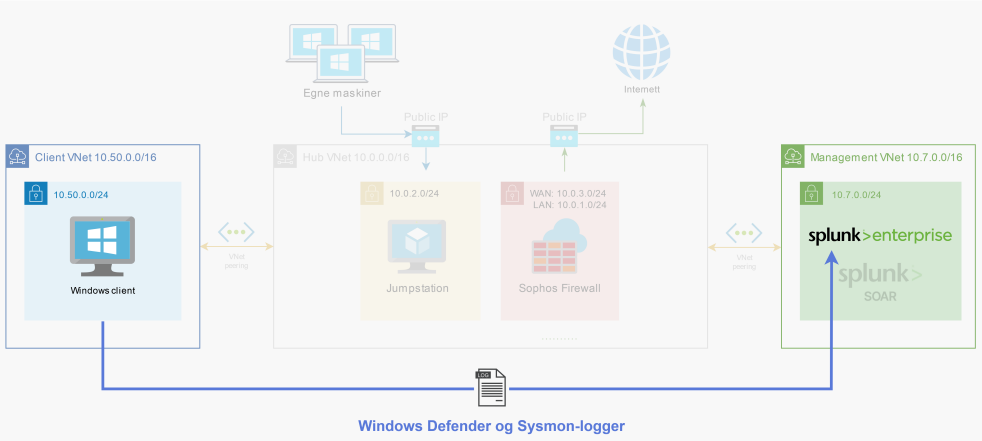
3.2.5 Dokumentasjon av konfigurasjon og bruk

Følgende underseksjoner redegjør for ytterligere konfigurasjon av verktøyene nevnt hittill. I tillegg dokumenterer vi hvordan teknologiene ble benyttet gjennom mulighetsstudiens brukstester. Disse seksjonene legger forutsetningene for å forstå og gjenskape de tekniske aspektene bak brukstestene.

Sende loggdata fra klient til Splunk Enterprise

Et av de fremste bruksområdene for Splunk Enterprise er innsamling av data fra klienter og endepunkter. Den vanligste måten å samle og videresende data til Splunk Enterprise er ved bruk av instanser kalt Splunk Forwarders[7], som installeres direkte på klienter og andre endepunkter. I denne oppgaven benyttet vi en type Splunk Forwarder installert på Windows-klienten, kalt Splunk Universal Forwarder, for å videresende loggdata generert av Microsoft Defender og Sysmon til Splunk Enterprise.

Splunk Universal Forwarder er en instans av Splunk som kjører som en bakgrunnsagent på endepunkter og klienter, og som muliggjør videresending av logger og annen data i sanntid[8]. Instansen inneholder kun den nødvendige funksjonaliteten for å videresende data til en indekser, som er en fullstendig Splunk Enterprise instans der det er mulig å søke, analysere og behandle dataen som samles inn[9]. I denne rapporten fokuserte vi på videresending av logger fra Microsoft Defender og Sysmon. Oppsett og konfigurasjon av Splunk Universal Forwarder for videresending av loggdata er dokumenter i vedlegg D.1.



Figur 3.2: Loggdata fra klient blir sendt til Splunk Enterprise

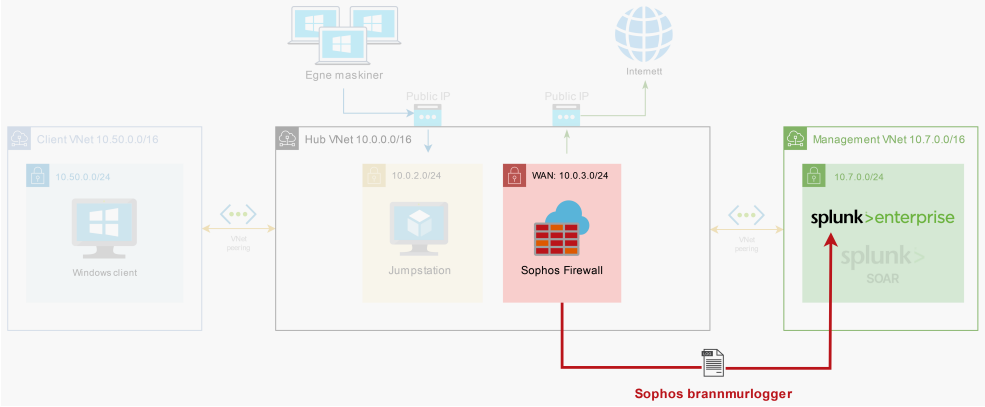
[7] https://docs.splunk.com/Splexicon:Forwarder

[8] https://docs.splunk.com/Documentation/Forwarder/9.0.4/Forwarder/Abouttheuniversalforwarder

[9] https://docs.splunk.com/Splexicon:Indexer

Sende loggdata fra brannmur til Splunk Enterprise

I tillegg til logger og data samlet inn på Windows-klienten, har vi i denne oppgaven samlet inn, analysert og håndtert logger fra brannmuren vi satt opp, beskrevet i vedlegg C.2. Brannmurlogger gir et unikt overblikk over nettverkstrafikken både internt i nettverket og ut mot internett. Analyse av disse loggene er viktig for å avdekke og håndtere potensielle hendelser. Loggdata fra brannmuren ble indeksert i Splunk Enterprise, i likhet med loggene fra klienten, men for brannmuren installerte vi ikke en Splunk Universal Forwarder. Istedenfor konfigurerte vi Splunk Enterprise for mottak av trafikk direkte fra brannmuren. Dette demonstrerer hvordan data kan indekseres i Splunk Enterprise ved bruk av en rekke forskjellige metoder, både ved bruk av Splunk-agenter eller ved bruk av løsninger som er uavhengig av Splunk på endepunktene. Konfigurasjon for videresending og indeksering av brannmurdata er dokumenter i vedlegg D.2.



Figur 3.3: Loggdata fra brannmur blir sendt til Splunk Enterprise

Søk og filtrering i Splunk Enterprise

Når data er samlet inn og indeksert i Splunk Enterprise, tilgjengeliggjøres det for søk og analyse gjennom Splunk-plattformens grensesnitt. Søk og filtrering blant den indekserte dataen danner grunnlaget for analyse og generering av eventer og alarmer. Disse eventene og alarmene benyttes under automatisk deteksjon og analyse av hendelser. Bruk av søkefunksjonaliteten i Splunk gjennomgås i større detalj i videre seksjoner, og under gjennomgang av casene senere i rapporten[10].

Generere alarmer i Splunk Enterprise

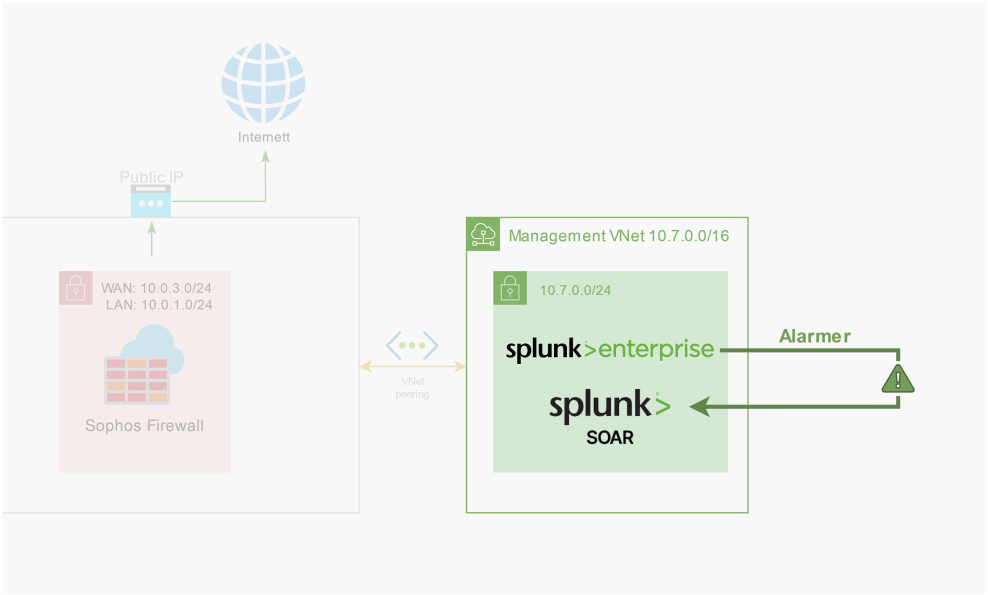
Etter at et søk er satt opp for å hente ut og filtrere ønsket data, kan alarmer genereres ut i fra disse. I noen tilfeller kan søk settes opp der ethvert resultat på

[10]Splunk opprettholder også en grundig dokumentasjon på deres søkespråk og hvordan søk og filtrering kan utformes `https://docs.splunk.com/Documentation/Splunk/9.0.4/SearchReference/WhatsInThisManual`

søket vil tilsi at en alarm skal genereres, for eksempel et søk som leter etter kjent skadevare på serverene til en organisasjon. I andre tilfeller vil et søk nesten alltid gi et eller flere resultater, men det er egenskaper ved selve resultatet som tilsier hvorvidt en alarm skal genereres eller ikke. Et eksempel er alarmer som kun genereres dersom det detekteres unormal eller ondsinnet aktivitet i brannmurlogger. Generering av alarmer i Splunk er dokumentert i vedlegg D.3.

Sende alarmer til Splunk SOAR

Alarmer som genereres i Splunk Enterprise kan videresendes til Splunk SOAR for å muliggjøre automatisk analyse og håndtering av hendelser. Når alarmer sendes til Splunk SOAR konfigureres blant annet sensitiviteten og kritikaliteten til alarmen, for å forenkle prioritering og klassifisering. Når alarmer mottas i Splunk SOAR opprettes det en sak (kalt event i SOAR-verktøyet) basert på alarmen, og alarmen fra Splunk Enterprise gjøres om til en artefakt (en bevisgjenstand) som gir kontekst og datagrunnlag til håndteringen. Deretter kan ytterligere bevis samles inn fra andre datakilder. Videresending av utløste alarmer fra Splunk Enterprise til Splunk SOAR er dokumentert i vedlegg D.4.

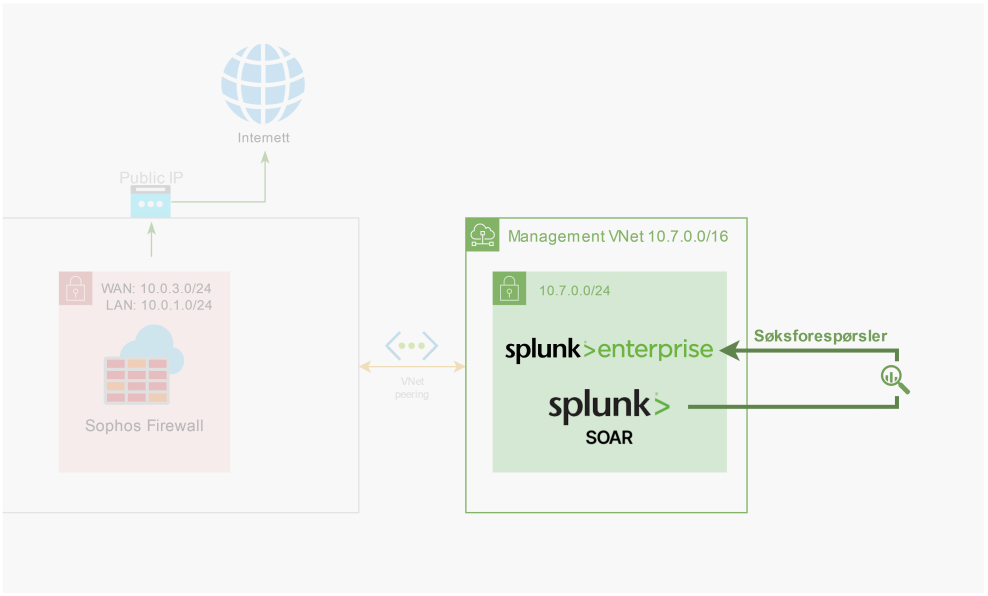


Figur 3.4: Alarmer trigget i Splunk Enterprise blir sendt til Splunk SOAR

Utføre søk i Splunk Enterprise fra Splunk SOAR

I mange tilfeller vil det være ønskelig å gjennomføre søk i Splunk Enterprise under håndtering av en hendelse i Splunk SOAR. Dette kan automatiseres slik at søk kan gjennomføres som en del av playbooks uten menneskelig interaksjon. Dermed kan hendelser berikes dersom det er nødvendig for å analysere hva som har skjedd

og finne ut hvordan hendelsen bør håndteres. Ved å koble SOAR-verktøyet opp mot datagrunnlaget og søkemotoren til Splunk Enterprise, kan ønsket informasjon innhentes ved behov når analyse og håndteringsprosessen krever det. Oppsett av denne integrasjonen mellom Splunk SOAR og Splunk Enterprise er dokumentert i vedlegg D.5.



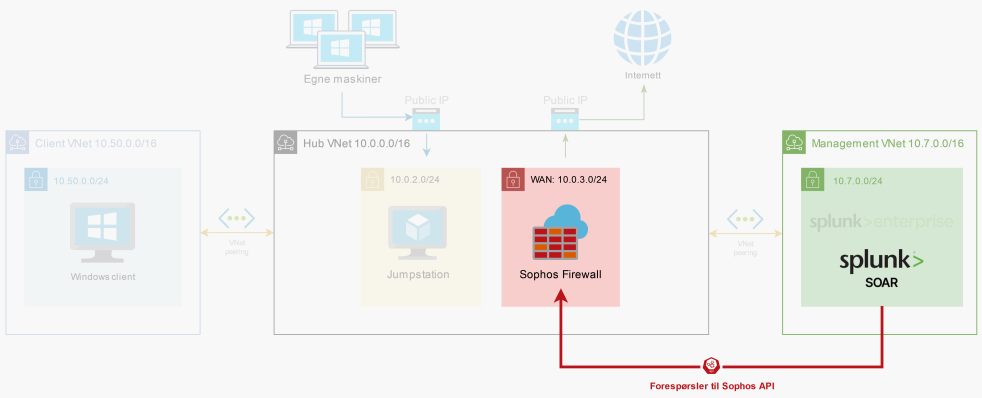
Figur 3.5: Søkeforespørsler sendes fra Splunk SOAR til Splunk Enterprise

Administrere brannmur fra Splunk SOAR

Automatisk respons er en sentral funksjonalitet i SOAR-verktøy. Vi har tidligere redegjort for innsamling av brannmurlogger fra Sophos Firewall til Splunk Enterprise. I tillegg ønsker vi også å administrere brannmuren fra Splunk SOAR, som kan konfigureres med et Application Programming Interface (API) som tillater forespørsler fra autentiserte noder. Deretter kan forespørsler mot Sophos Firewall sitt API inkluderes som handlinger i playbooks som utføres automatisk når hendelser krever det. Slike handlinger inkluderer oppsett av nye brannmurregler, samt modifikasjon eller deaktivering av eksisterende regler. Oppsett av brannmuradministrasjon er dokumentert i vedlegg D.6.

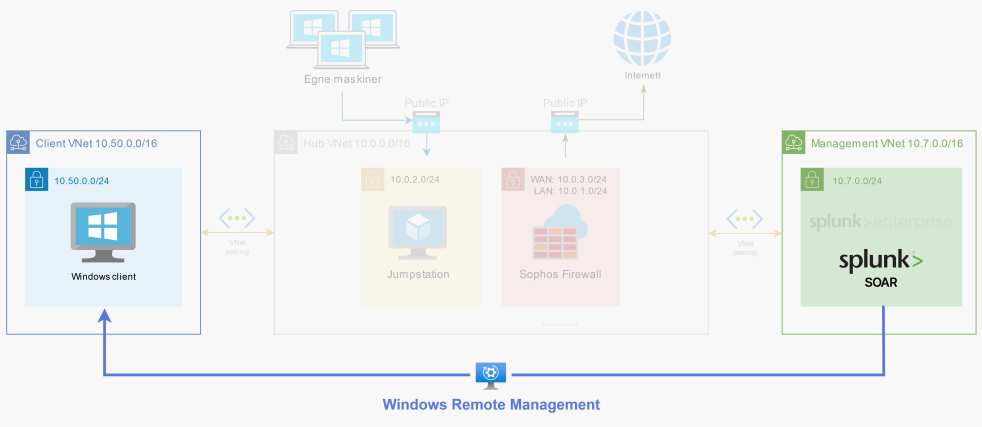
Administrere endepunkter fra Splunk SOAR

I tillegg til brannmuradministrasjon kan også vanlige klienter administreres direkte fra Splunk SOAR. Denne rapportens omfang inkluderer ikke oppsett av XDR eller lignende verktøy som vanligvis brukes for endepunktsadministrasjon. Vi vil derfor bruke alternativer for direkte kommunikasjon mellom SOAR og endepunk-



Figur 3.6: Administrere brannmur fra Splunk SOAR

ter. I denne oppgaven setter vi opp Windows Remote Management[11] (WinRM) på Windows-klienten, en enkel agent og protokoll for fjernadministrasjon av Windows-enheter og -servere. Over denne protokollen kan Splunk SOAR sende forespørsler og utføre handlinger direkte på Windows-klienten, som et ledd i håndtering av hendelser. Oppsett av Windows Remote Management og endepunktsadministrasjon fra SOAR er dokumentert i vedlegg D.7.



Figur 3.7: Administrere endepunkter fra Splunk SOAR

Lage playbooks i Splunk SOAR

Alarmer som kommer inn til Splunk SOAR håndteres ved bruk av playbooks. Playbooks er stegvise og sammenkoblede prosedyrer bestående av handlingsblokker som har forskjellige funksjoner. Blokkene kan blant annet kommunisere med eksterne tjenester, starte nye playbooks eller sende kommandoer til enheter i nettverket. Andre blokker kan legge igjen kommentarer i saken eller bestemme hvilket

[11] `https://learn.microsoft.com/en-us/windows/win32/winrm/portal`

steg i playbooken som skal utføres. Playbooks kan konfigureres slik at de utløses automatisk når visse alarmer sendes til Splunk SOAR (automatiseringsplaybook), eller de kan designes for bruk som en underdel av en overordnet playbook (input-playbook). I denne oppgaven vil vi vise bruk av begge typer, for å demonstrere forskjellige bruksområder. Playbooks lages hovedsaklig gjennom et grafisk grensesnitt der det er lett å visualisere playbooken og de sammenkoblede stegene, men funksjonaliteten kan være litt begrenset. I bakgrunnen ligger Python-kode, og det er også mulig å lage playbooks kun ved bruk av Python-kode. I denne oppgaven lager vi playbooks gjennom det grafiske grensesnittet, i tillegg til å legge til Python-kode der det grafiske grensesnittet ikke strekker til. Oppsett og design av playbooks er dokumentert i vedlegg D.8.

3.2.6 Casestruktur

I denne delen av rapporten presenterer vi tre caser for å demonstrere noen av de løsningene som er mulig ved bruk av de tekniske verktøyene og tjenestene vi har diskutert fra seksjon 3.2.3 til 3.2.5. Casene danner dermed grunnlaget for tre konseptbevis med praktiske demonstrasjoner av effektiviserte og automatiserte arbeidsflyter.

Et gjentagende tema gjennom alle underdelene til hver case er referanser til Cyber Kill Chain-modellen og MITRE ATT&CK-rammeverket [6, 22]. Disse brukes for å vise hvor hendelsen befinner seg i levetiden til et cyberangrep, hvilke faser trusselaktøren allerede har gjennomført, samt hvilke faser som kan nås dersom hendelsen ikke håndteres. Dette kobler casene direkte opp mot relevant teori for gjennomførelsen og håndteringen av cyberangrep.

MITRE ATT&CK brukes intensivt for å vise til spesifikke taktikker og teknikker trusselaktøren benyttet seg av både før og under angrepet, samt hvilke taktikker og teknikker som kan bli brukt i fremtiden. Der slike taktikker og teknikker nevnes, viser vi til seksjon 2.2 og fotnoter bak hver taktikk og teknikk – for ytterligere informasjon om hva de beskriver – hvilke underteknikker som finnes, samt potensielle mitigeringstiltak. For hver case har vi utarbeidet et varmekart basert på nyeste versjon av MITRE ATT&CK-rammeverket [59], som illustrerer taktikker og teknikker brukt før og under angrepet, taktikker og teknikker som kan bli brukt i fremtiden, og taktikker og teknikker som er mål for håndterings- og mitigeringstiltak. Casene vi beskriver i seksjon 3.2.7 til 3.2.9 følger samme struktur og inneholder de samme momentene, som vi beskriver i følgende underdeler.

Casebeskrivelse

Alle tre casene begynner med en kort beskrivelse, der vi bygger opp en bakgrunnshistorie for casen og setter konteksten for resten av innholdet og gjennomførelsen av denne. Casebeskrivelsen beskriver hvordan organisasjonen har havnet i den situasjonen de befinner seg i og hva slags hendelse det er snakk om. I tillegg redegjør vi for hva vi ønsker å oppnå og vise med casen.

Hver case inkluderer en figur over hendelsesflyten. Dette er en figur som viser hvordan hendelsen har oppstått. Figuren fokuserer på de handlingene trusselaktøren har gjort for å skape hendelsen, i tillegg til hva organisasjonen har gjort for å forårsake hendelsen der dette er relevant.

Forutsetninger

Denne underdelen utdyper casebeskrivelsen ved å redegjøre i større detalj for de forutsetninger som må være på plass for at hendelsen skal kunne utvikle seg. Forutsetninger inkluderer manglende sikkerhetsrutiner hos organisasjonen, handlinger utført av trusselaktøren, og andre antagelser som er nødvendig.

Konsekvens

I tillegg til forutsetninger setter vi for hver case søkelys på de mulige konsekvensene som kan oppstå – utover den skaden som allerede er gjort – dersom hendelsene ikke håndteres. Dette illustreres hovedsaklig gjennom handlinger trusselaktøren kan foreta seg basert på hendelsens nåværende status, med referanser til taktikker og teknikker i MITRE ATT&CK-rammeverket.

Tilstandsbilde

Vi klassifiserer hver case innunder en eller flere faser i Cyber Kill Chain-modellen, samt viser til en eller flere av taktikkene og teknikkene i MITRE ATT&CK-rammeverket. Her beskrives fasene og taktikkene som angrepet befinner seg i når casen finner sted og vil forklare mål for mitigering og håndtering av casene. Med andre ord beskriver vi hvordan angrepet ser ut nå og hvilke deler av fasene og taktikkene som casen skal vise mulig mitigering og håndtering av. I tillegg presenterer vi her et varmekart for hver case basert på MITRE ATT&CK-rammeverket, som viser taktikker og teknikker nevnt under casebeskrivelsen, varmekartet viser forutsetninger (rød farge), konsekvens (blå farge) og mål for håndtering i casen (gul farge).

Mitigering og håndtering

Under mitigerings- og håndteringsdelen av hver case beskriver vi hvilke tiltak vi foreslår for å detektere, analysere og håndtere hendelsen. Vi viser til oppsett av deteksjonslogikk, playbooks og andre tiltak som støtter opp under håndtering og mitigering av hendelsen.

Gjennomføring av case

Den siste delen av casestrukturen er en beskrivelse av hvordan casen gjennomføres, inkludert simulering av trusselaktøraktivitet og handlinger fra organisasjonens side som utløser gjennomføringen av casen. Deretter henviser vi til resultatdelen av oppgaven for beskrivelse av resultatet etter gjennomføring av hver case.

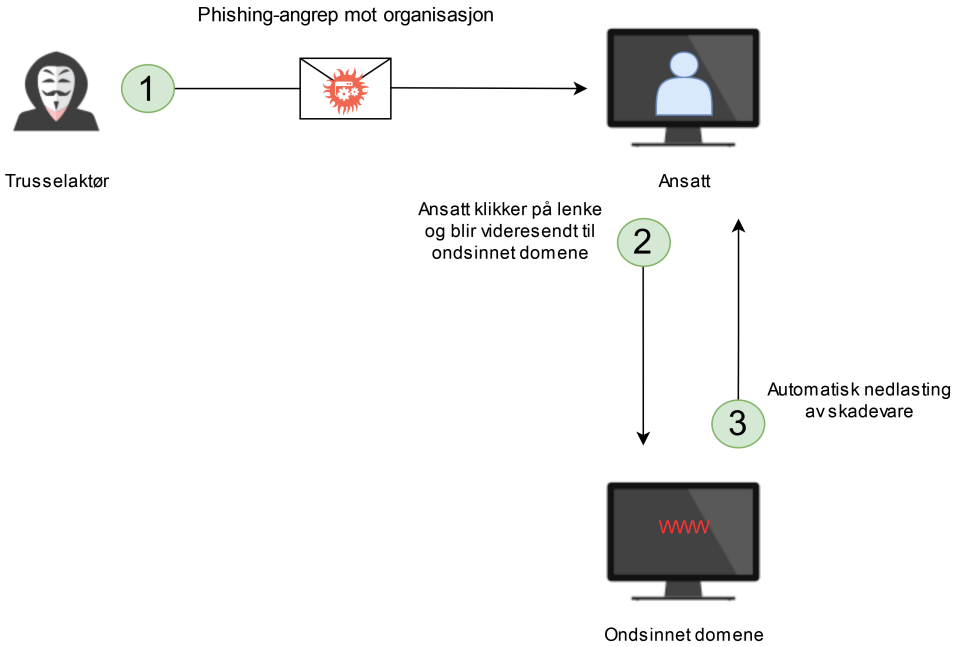
Kildekode

Kildekoder til alle playbooks diskutert i følgende seksjoner ligger i oppgavens kildekode-repository på GitHub[12].

3.2.7 Case 1 - Phishing med ondsinnet lenke

Casebeskrivelse

En ansatt har fått en phishing e-post, bestående av en ondsinnet lenke. Den ansatte trykket på lenken, som resulterte i nedlasting av skadevare på klientmaskinen. Etter at skadevaren ble lastet ned, skannet Microsoft Defender Antivirus[13] (heretter: Microsoft Defender) den nedlastede filen og identifiserte skadevaren som ondsinnet basert på kjent signatur. I denne casen skal vi undersøke hvorvidt skadevaren fortsatt er en aktiv trussel mot virksomhetens systemer, ved å blant annet sjekke om Microsoft Defender sine tiltak er utført vellykket. Figur 3.8 viser hendelsesflyten for case 1.



Figur 3.8: Case 1: Angrepets hendelsesflyt

[12] https://github.com/TordV/BachelorThesisNTNU2023

[13] https://learn.microsoft.com/en-us/microsoft-365/security/defender-endpoint/microsoft-defender-antivirus-windows?view=o365-worldwide

Forutsetninger

En trusselaktør har tilegnet seg navn og e-postadresse gjennom rekognosering av ansatte i virksomheten [89]. I bevæpningsfasen har trusselaktøren anskaffet skadevare som skal bli benyttet under angrepet [90]. I tillegg har trusselaktøren satt opp et ondsinnet domene («et administrativt delområde i et datasystem eller datanettverk» [13]), bestående av en nettside med et klientside-script som starter automatisk nedlasting av skadevare på målets maskin [91]. Teknikken linkmål (eng: *link target*) har også blitt benyttet ved å sette inn en hyperlenke i e-posten for å videresende brukeren til trusselaktøren sitt ondsinnede domene [92]. I leveringsfasen ble den ondsinnede e-posten sendt til en ansatt hos offerorganisasjonen, og filtreringen på e-postserveren klarte ikke å detektere og filtrere bort den ondsinnede e-posten [93].

Konsekvens

Dersom Microsoft Defender sine tiltak for å håndtere den ondsinnede nedlastingsfilen mislykkes, så vil dette resultere i at trusselaktøren kan få fotfeste i virksomhetens systemer. Dette vil åpne opp en ny intern angrepsflate som trusselaktøren kan utnytte for å bevege angrepet inn i nye faser. De ulike angrepsteknikkene som en trusselaktør vil benytte under de videre fasene kan være mange og vil avhenge av virksomhetens systemer. Følgende avsnitt kartlegger noen konkrete angrepsteknikker som vil være mulige konsekvenser for denne casen, dersom Microsoft Defender ikke klarer å håndtere skadevaren.

Ved vellykket eksekvering av skadevare vil trusselaktøren påbegynne arbeidet for å oppnå persistens på systemet [94, 95]. Teknikkene vil blant annet innebære brukeropprettelser og konfigurering av regelmessige planlagte oppgaver [96, 97]. Ved kompromittering av den ansattes brukerkonto og maskin, vil trusselaktøren ha mulighet til å benytte ekstern tilkobling for å få tilgang til interne ressurser som brukeren har rettigheter til. Teknikkene for å oppnå persistens vil også benyttes etterhvert som trusselaktøren beveger seg lateralt i systemet eller ved tilegnelse av nye rettigheter, for å sikre fremtidig tilgang [98, 99].

Eskalering av rettigheter (eng: *privilege escalation*) vil gi trusselaktøren tilgang til nye og mer virksomhetskritiske ressurser. Det er flere teknikker som trusselaktøren kan benytte i et forsøk på å tilegne seg høyere rettigheter, blant annet ved å omgå kontroller i Windows sin *User Account Control* (UAC) [100]. UAC er en tilgangskontroll i Windows som skal gjøre det mulig for prosesser å elevere rettighetene sine ved behov [101]. Videre kan trusselaktøren prøve å tilegne seg tilgangsnøkler med høyere rettigheter og starte nye prosesser med disse [102]. Dette kan la trusselaktøren omgå de vanlige tilgangskontrollene som eksisterer i systemet.

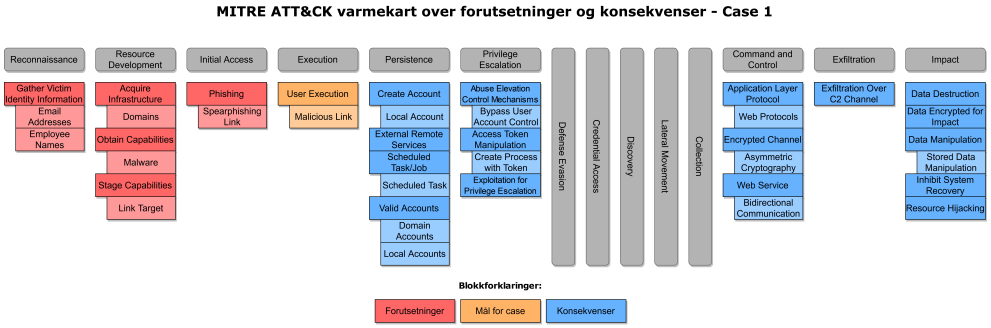
Etterhvert som angrepet beveger seg inn i de siste fasene kan trusselaktøren få et behov for å eksfiltrere data over C2-kanaler [103]. Kjente protokoller som opererer på applikasjonslaget, som for eksempel HTTP, kan benyttes for å overføre sensitiv data mellom virksomhetens systemer og servere kontrollert av trusselaktø-

ren. Etterhvert som trusselaktøren får eksfiltrert data til egne servere, vil det kunne være hensiktsmessig for trusselaktøren å manipulere virksomhetens data for å påvirke virksomhetsprosesser, virksomhetsforståelse og beslutningstaking [104].

Alternative konsekvenser i siste fase er at trusselaktøren krypterer eller destruerer data for å gjøre virksomhetens verdier utilgjengelige [105, 106]. Kryptering av filer har historisk vært den vanligste teknikken brukt i løsepengevirus-angrep, men *Cyderes Special Operations* og *Stairwell Threat Research* har oppdaget tegn på at digital utpressing kan bevege seg inn i en ny trend, der data destrueres i stedet for å krypteres. Destruering av data vil både effektivisere siste fase i Cyber Kill Chain for trusselaktøren, i tillegg til å øke sannsynligheten for et større skadeomfang [107].

Tilstandsbilde

Denne casen tar sted i fasen *utnyttelse* i Cyber Kill Chain. Figur 3.9 viser hvilke taktikker og teknikker, i henhold til MITRE ATT&CK rammeverket, som har blitt benyttet i dette angrepet. Teknikkene merket i rødt representerer forutsetningene i denne casen. Dette er derfor teknikker som har blitt vellykket utført fra trusselaktøren sitt ståsted. Videre representerer teknikkene merket i blått hvordan angrepet kan utvikle seg videre dersom håndteringen av skadevaren mislykkes. I og med at Microsoft Defender identifiserer den nedlastede filen som ondsinnet, skal casen undersøke om angrepet kan stoppes og håndteres under taktikken *eksekvering* og teknikken *ondsinnet lenke* i ATT&CK-rammeverket [108, 109].



Figur 3.9: Case 1: MITRE ATT&CK varmekart

Mitigering og håndtering

For å mitigere denne casen med en forhåndsdefinert playbook, er det en forutsetning at hendelsen blir detektert. På generelt grunnlag ville det vært hensiktsmessig å stoppe angrepet allerede i leveringsfasen, gjennom e-postfiltre og andre mitigerende tiltak slik at brukeren aldri får den ondsinnede e-posten i sin mailboks.

I denne casen antar vi likevel at den ondsinnede mailen kommer frem til offerets maskin der brukeren laster ned skadevare som Microsoft Defender detekterer

som ondsinnet. Med dette tatt i betrakning, ønsker vi å se på håndtering av hendelsen på bakgrunn av denne antivirus-alarmen. Det første tiltaket som settes inn for håndtering av en slik hendelse er videresending av logger fra Microsoft Defender til Splunk Enterprise, som vi har dokumentert i vedlegg D.1. Dermed er det mulig å agere på disse loggene så fort de oppstår.

I Splunk Enterprise har vi opprettet et sanntidssøk[14] som har som formål å se et etter nye antivirus alarmer. Et sanntidssøk er et søk som kontinuerlig kjøres, for å detektere spesifikke eventer i loggene med en gang de ankommer Splunk. Alternativt kan et søk settes opp som et planlagt søk[15] som kjøres ved et gitt intervall, for eksempel hvert minutt eller hvert 15. minutt. I denne casen er søket satt opp til å se etter antivirus alarmer, i tillegg til å hente ut informasjon om hva Microsoft Defender har gjort for å håndtere skadevaren. Ved deteksjon av nye antivirus alarmer, genererer søket en alarm som sendes til Splunk SOAR. For mer detaljert beskrivelse av søk og alarm, se vedlegg E.1.1

I Splunk SOAR har vi opprettet en playbook som automatisk startes når det ankommer en antivirus alarm fra Splunk Enterprise. En hardkodet etikk med navnet *antivirus* er koblet til både alarmen fra Splunk Enterprise, samt playbooken i Splunk SOAR. Dette vil sørge for at riktig playbook kjøres når alarmen ankommer Splunk SOAR. I denne casen, er playbooken satt opp til å analysere og håndtere en antivirus alarm. For å oppnå dette, vil playbooken analysere tiltakene gjort av Microsoft Defender, analysere skadevaren og gjennomføre en ny sikkerhetscan av maskinen for å verifisere at den ikke er kompromittert. Figur 3.10 viser den overordnede flyten i playbooken, og se vedlegg E.1.2 for en mer detaljert beskrivelse av playbooken.

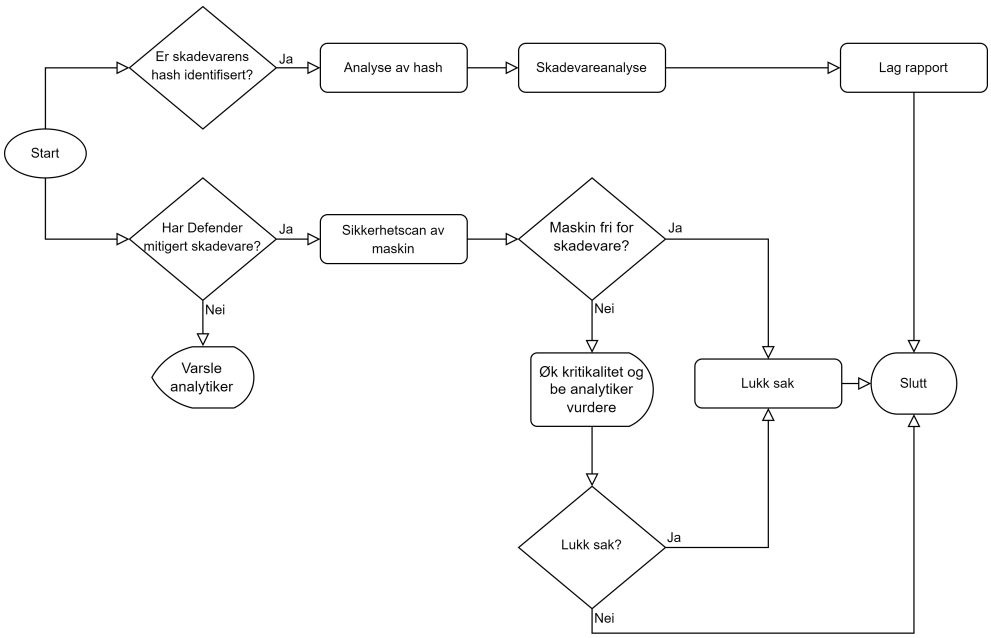
Gjennomføring av case

For å gjennomføre denne casen har vi simulert nedlasting av skadevare fra et ondsinnet domene, som brukeren besøker gjennom lenken i den ondsinnede e-posten. For å oppnå dette lager vi et script kalt `case1_trigger.ps1`, som oppretter en ny fil i nedlastingsmappen til klienten. Denne filen inneholder en EICAR-streng (også kalt EICAR-fil), som er en ufarlig tekststreng som benyttes i sammenheng med testing av deteksjonssystemer og antivirus-programmer[16]. En fil med denne tekststrengen skaper en hash som er lagt til som en indikator i de fleste databaser over kjente trusselindikatorer. Filen er altså ikke faktisk skadevare, men brukes for test-formål ettersom antivirus-program og andre deteksjonssystemer vil flagge den som en ondsinnet fil. Ved å lage filen kan vi derfor simulere nedlasting av skadevare på klienten som Microsoft Defender vil detektere. Gjennomføringen av casen initieres ved å kjøre dette scriptet. Resultatet etter gjennomføring av case 1 er beskrevet i seksjon 4.2.1.

[14] `https://docs.splunk.com/Splexicon:Realtimesearch`

[15] `https://docs.splunk.com/Splexicon:Scheduledsearch`

[16] Se følgende nettsted for mer informasjon: `https://www.eicar.org/download-anti-malware-testfile/`



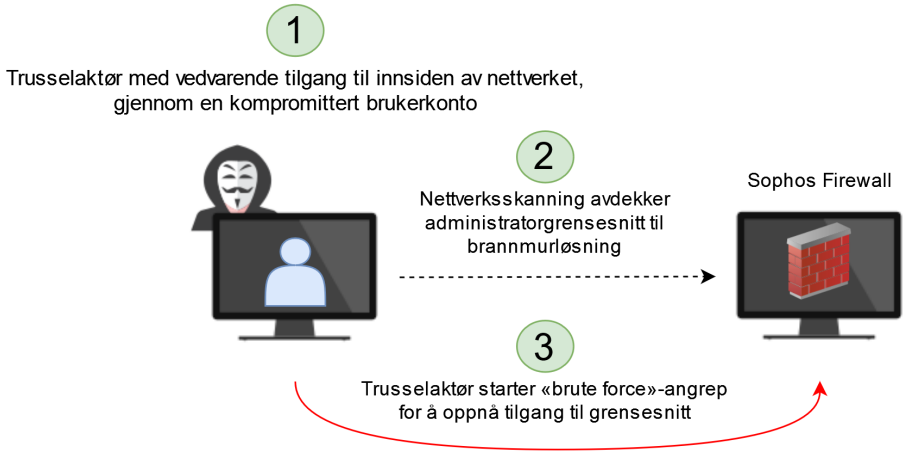
Figur 3.10: Case 1: Flytdiagram over playbook

3.2.8 Case 2 - Lateral bevegelse og brute force-angrep

Casebeskrivelse

En avansert trusselaktør har oppnådd vedvarende tilgang til en organisasjons interne nettverk, gjennom en brukerkonto med driftsrettigheter. Ved søk og skanning av nettverket og dets tilkoblede systemer så oppdager trusselaktøren forskjellige tjenester som kun er tilgjengelig fra det interne nettverket. Én av disse tjenestene er administratorgrensesnittet til den skybaserte brannmurløsningen organisasjonen benytter, Sophos Firewall. Grensesnittet er tilgangsstyrt og krever pålogging med riktig legitimasjon i form av et brukernavn og passord.

Gjennom rekognosering har trusselaktøren funnet ut at brannmurløsningen benytter `admin` som brukernavn, men passordet er ikke kjent. Ved lateral bevegelse kan trusselaktøren legge opp til videre angrep mot offeret, dette gjør de ved å få forsøke å få tilgang til brannmurløsningen. Derfor igangsetter trusselaktøren et *brute force*-angrep (angrep som benytter rå datakraft for å avdekke gyldige brukernavn/passord til en tjeneste eller et system) med mål om å avdekke riktig påloggingskombinasjon og dermed oppnå tilgang til grensesnittet. Denne casen undersøker hvorvidt brute force-angrepet kan oppdages og håndteres før trusselaktøren oppnår tilgang til grensesnittet. Figur 3.11 viser hendelsesflyten for case 2.



Figur 3.11: Case 2: Angrepets hendelsesflyt

Forutsetninger

I denne casen forutsettes det at en trusselaktør har oppnådd vedvarende tilgang til et system. I likhet med case 1 kan et phishing-angrep ha blitt benyttet for å installere en bakdør på det berørte systemet[93]. Det forutsettes videre at trusselaktøren over lengre tid har hatt vedvarende tilgang til denne legitime kontoen på systemet, som er forankret gjennom opprettelsen av C2-trafikk, eskalering av privelegier og regelmessige tiltak for å unngå deteksjon og sikre persistens[95, 98, 110, 111]. Trusselaktøren kan for eksempel ha deaktivert multifaktorautentisering på maskinen uten at dette har blitt flagget av organisasjonens sikkerhetssystem, i tillegg til å regelmessig fjerne *Windows Event*-logger for å skjule mistenkelig aktivitet [112, 113]. I løpet av denne tiden har trusselaktøren rekognosert det interne nettverket, og oppdaget forskjellige tjenester tilgjengelig fra den kompromitterte brukerkontoen [114]. Hittill har offeret ingen indikasjon på kompromittering i nettverket.

Konsekvens

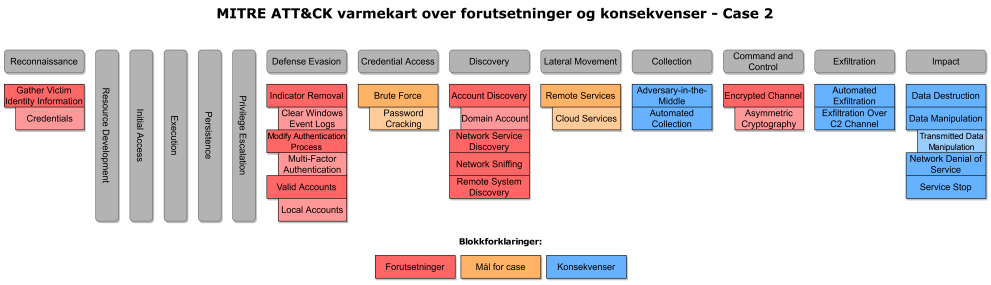
Dersom trusselaktøren lykkes med brute force-angrepet og oppnår tilgang til brannmurens administratorgrensesnitt, kan trusselaktøren utføre en rekke handlinger som kan skade organisasjonens verdier [115]. Trusselaktøren vil blant annet kunne samle inn og hente ut logger som er synlige gjennom grensesnittet, som vil være en betydelig form for datainnsamling som kan fortelle mye om organisasjonen og dens trafikk [116]. Det vil gi trusselaktøren en *man-in-the-middle* posisjon som kan utnyttes for å fange opp – og potensielt modifisere – trafikk imellom systemer og endepunkter hos organisasjonen [117]. Loggene som samles inn kan for eksempel eksfiltreres gjennom C2-kanalen som er opprettet [103]. Sikkerhetsselskapet Volexity oppdaget i 2022 et datainnbrudd der en trusselaktør hadde

oppnådd illegitim tilgang til Sophos Firewall, og deretter gjennomført et man-in-the-middle-angrep ved bruk av denne tilgangen [118].

Andre potensielle konsekvenser ved et slikt angrep inkluderer misbruk av tilgangen til grensesnittet for å endre brannmurregler, åpne opp for ondsinnet trafikk eller fjerne regler som beskytter organisasjonen. Trusselaktøren kan også utnytte det kompromitterte systemet som en base for videre angrep eller laterale bevegelser [99]. I tillegg kan trusselaktøren skade integriteten eller tilgjengeligheten til systemer og tjenester ved å stenge for legitim trafikk (dermed skape en form for tjenestenektangrep), slette eller korruptere data, eller stoppe selve brannmurtjenesten [119, 120].

Tilstandsbilde

Denne casen demonstrerer et angrep som tar sted under fasen *handlinger på målet* i Cyber Kill Chain [6]. Trusselaktøren har oppnådd vedvarende tilgang til offerets nettverk, og utfører nå målrettede angrep innenfor nettverket med ønske om å stjele data eller påføre skade til organisasjonen på annet vis. Under MITRE ATT&CK-rammeverket kan vi plassere angrepet mer spesifikt under teknikken brute force, som er en del av taktikken *identitetstyveri* (eng: *credential access*) [115, 121]. Trusselaktøren gjennomfører et angrep med mål om å oppdage legitimasjoner som gir tilgang til brannmurgrensesnittet. Samtidig er angrepet en del av taktikken *lateral bevegelse* (eng: *lateral movement*) mot *eksterne tjenester* (eng: *remote services*), ettersom målet med brute force-angrepet er tilgang og bevegelse inn i nye deler av organisasjonens systemer ved bruk av legitime brukerkontoer (eng: *valid accounts*) [53, 99, 122]. Vi kan visualisere angrepet, med forutsetninger og potensielle konsekvenser, som vist i varmekartet i figur 3.12.



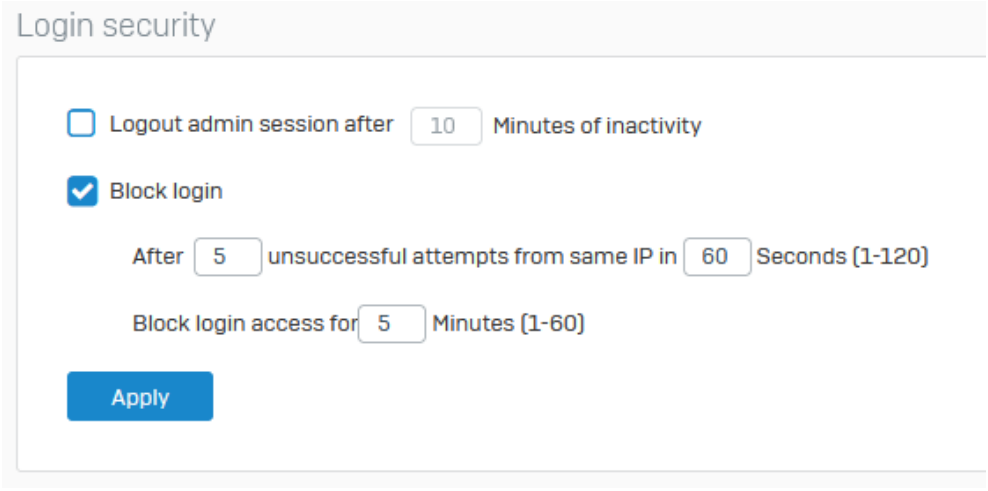
Figur 3.12: Case 2: MITRE ATT&CK varmekart

Mitigering og håndtering

For å mitigere og håndtere denne casen var det nødvendig å innføre teknikker for å oppdage indikatorer på brute force-angrep og automatisk generere alarmer når slike indikatorer oppdages. Alarmene skal utløse automatisk hendelseshåndtering gjennom en playbook og tilhørende tiltak for å stoppe brute force-angrepet så tidlig som mulig, samt varsle relevante parter i organisasjonen. Det er verdt å

nevne at streng tilgangsstyring til brannmurgrensesnittet – i henhold til prinsippet om minste privilegium – samt sterke passord er naturlige mitigeringstiltak som alltid bør være på plass for en slik tjeneste, men dette er utenfor omfanget av oppgaven.

Sophos Firewall har innebygget beskyttelse mot gjentatte påloggingsforsøk, et tiltak som er konfigurerbart gjennom administratorgrensesnittet til Sophos Firewall som vist i figur 3.13. Denne beskyttelsen vil blokkere videre påloggingsforsøk i 5 minutter etter 5 mislykkede forsøk fra samme IP-adresse innen 60 sekunder, og dermed forhindre de fleste typer brute force-angrep. Videre i denne casen antok vi likevel at denne typen beskyttelse er deaktivert i administratorgrensesnittet, for å demonstrere hvordan vi kunne oppnå lignende resultat gjennom andre metoder.



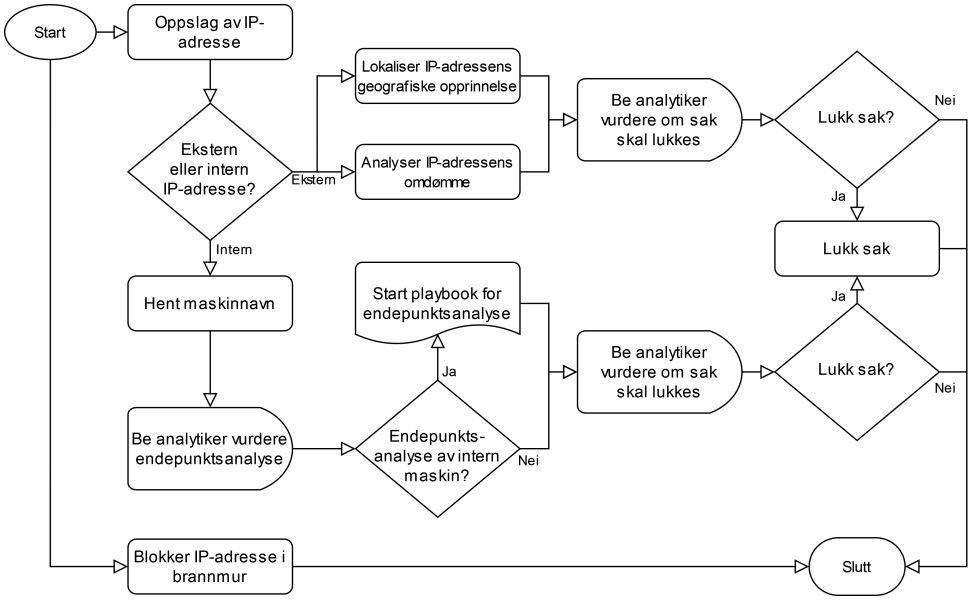
Figur 3.13: Case 2: Innebygget brute force-beskyttelse i Sophos Firewall

I likhet med case 1 har vi også i denne casen satt opp et sanntidssøk i Splunk Enterprise som leter etter mislykkede påloggingsforsøk blant brannmurens logger. Ved deteksjon av tre mislykkede forsøk fra samme IP-adresse mot samme brukerkonto innenfor et minutt, genererer dette søket en alarm som sendes til Splunk SOAR. For beskrivelse og oppsett av søk og alarm, se vedlegg E.2.1.

I Splunk SOAR har vi laget en playbook som vi koblet opp denne alarmen, slik at playbooken aktiveres automatisk når en slik alarm sendes til Splunk SOAR. Playbooken blokkerer trafikk fra IP-adressen som forårsaker et brute-angrep, i tillegg til å gjennomføre analyse av IP-adressen. Et overordnet flytdiagram av playbooken er vist i figur 3.14. For ytterligere beskrivelse av playbooken, se vedlegg E.2.2.

Gjennomføring av case

For å gjennomføre denne casen simulerte vi et brute force-angrep fra en Windows-klient på det interne nettverket. Vi brukte et terminalvindu og forsøkte pålogging over SSH til brannmurens administratorgrensesnitt på IP-addresse `10.0.3.4` for



Figur 3.14: Case 2: Flytdiagram over playbook

å simulere et angrep med kommandolinjeverktøy, som vist i figur 3.15. Under simuleringen benyttet vi oss ikke av kommandolinjeverktøy for brute force-angrep, men vi skrev bevisst inn feil passord flere ganger innenfor kort tid for å simulere angrepet. Resultatet etter gjennomføring av case 2 er beskrevet i seksjon 4.2.2.

```
PS C:\Users\admbachelor> ssh admin@10.0.3.4
admin@10.0.3.4's password:
Permission denied, please try again.
admin@10.0.3.4's password:
Permission denied, please try again.
admin@10.0.3.4's password:
admin@10.0.3.4: Permission denied (publickey,password,keyboard-interactive).
```

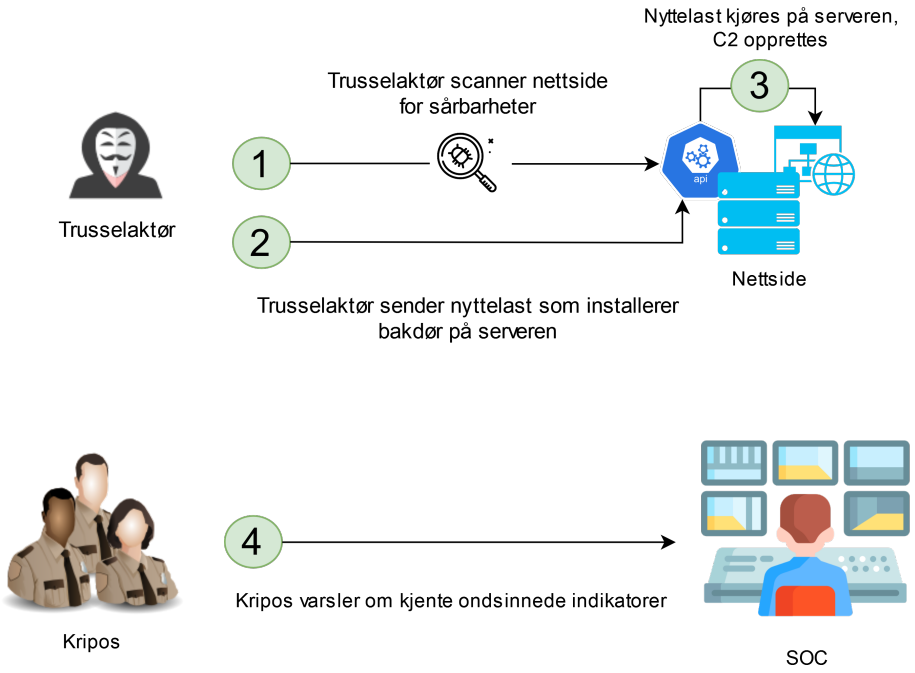
Figur 3.15: Case 2: Intern IP-adresse forsøker pålogging mot admin bruker på brannmur via terminal

3.2.9 Case 3 - Oppdagelse av mistenksom trafikk

Casebeskrivelse

En trusselaktør har oppnådd vedvarende tilgang til en organisasjons systemer gjennom å utnytte en av deres offentlige applikasjoner. Trusselaktøren har gjennomført sårbarhetsskanning og utviklet en utnyttelse for å angripe en spesifikk sårbarhet ved organisasjonens nettside. Ved utnyttelse av sårbarheten på den offentlige applikasjonen har trusselaktøren installert og opprettholdt en bakdør. Trusselaktøren har videre opprettet C2-trafikk mellom seg selv og nettserveren. Trusselaktøren har i løpet av sin tid inne i systemet, brukt den vedvarende tilgan-

gen til å hente ut sensitiv data fra serveren og tilkoblede databaser uten å bli oppdaget. På et senere tidspunkt blir organisasjonen varslet fra Kripos om indikatorer tilhørende kjente trusselaktører. Denne casen undersøker hvorvidt eksfiltrering av data over C2-trafikken kan oppdages på bakgrunn av denne trusseletterretningen, samt hvorvidt prosesser kan aktiveres for å håndtere og stanse datainnbruddet. Figur 3.16 viser hendelsesflyten for case 3.



Figur 3.16: Case 3: Angrepets hendelsesflyt

Forutsetninger

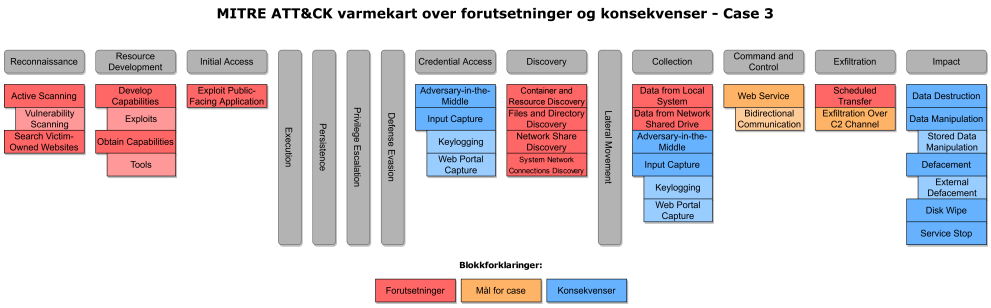
I denne casen forutsettes det at en trusselaktør ubemerket har kommet til *handlinger på målet*-fasen av Cyber Kill Chain, ved å utnytte en offentlig eksponert nettside [52]. Det forutsettes at trusselaktøren kontinuerlig samler inn sensitiv data fra organisasjonens server, og eksfiltrerer denne informasjonen gjennom C2-kanalen som er opprettet [103, 123, 124]. For at trusselaktøren skal ha oppnådd dette, forutsettes det som nevnt i beskrivelsen av casen, at organisasjonen har hatt en sårbarhet i sin offentlig tilgjengelige applikasjon (nettside eller API) som trusselaktøren har oppdaget gjennom sårbarhetsskanning, og videre utviklet en utnyttelse for [125, 126]. Deretter antar vi at angriperen har klart å utnytte sårbarheten uten å bli oppdaget, for å skape en bakdør på serveren og etablere persistens [95]. Det forutsettes også at indikatorer for sårbarhetsutnyttelse, C2-trafikk og eksfiltrering av sensitiv informasjon har gått ubemerket fram til nå [111].

Konsekvens

Dersom den aktive utnyttelsen ikke detekteres og stanses vil trusselaktøren fortsette å samle inn og eksfiltrere sensitiv informasjon fra organisasjonen, i tillegg til at trusselaktøren kan forårsake nye konsekvenser i fremtiden. Sensitiv data på avveie kan gjøre stor skade på organisasjonens verdier, og deres omdømme vil også kunne svekkes. Alvorlighetsgraden av disse konsekvensene avhenger av hvilken informasjon trusselaktøren klarer å samle inn, hvilken posisjon organisasjonen har i samfunnet og hvorvidt trusselaktøren velger å lekke informasjonen som stjeles. Andre mulige konsekvenser av hendelsen inkluderer utnyttelse av serveren for å skape en man-in-the-middle-posisjon, som kan fange opp informasjon og data på vei mellom brukere og organisasjonens systemer, for eksempel passord eller betalingsinformasjon [117, 127]. Systemfiler og kritisk programvare på serveren kan også modifiseres og utnyttes for å skape ytterligere sårbarheter eller bakdører [104]. Videre kan trusselaktøren bruke serveren for å spre skadevare, lenker eller annen ondsinnet informasjon til brukere av nettsiden [128]. Stenging av serveren eller tjenestene den leverer, eller ødeleggelse av data på serveren, er også aktuelle konsekvenser [106, 120, 129].

Tilstandsbilde

Casen demonstrerer et angrep som kan plasseres i fasene *kommando og kontroll* og *handlinger på målet* i Cyber Kill Chain-modellen [6]. Trusselaktøren har oppnådd vedvarende tilgang til organisasjonens systemer, og sensitiv informasjon sendes mellom serveren og trusselaktøren over en C2-kanal som er opprettet ved å benytte en nettjeneste for toveis-kommunikasjon. Denne casen kan derfor plasseres i MITRE ATT&CK-rammeverket under taktikken *kommando og kontroll* med teknikk *nettjenester* [110, 130]. I tillegg kan angripet plasseres i taktikken *eksfiltrering* med teknikk *eksfiltrering over C2-kanal* [103, 131]. En komplett oversikt over de gjeldende taktikker og teknikker for denne casen er vist i figur 3.17.



Figur 3.17: Case 3: MITRE ATT&CK varmekart

Mitigering og håndtering

Mitigering og håndtering av case 3 avhenger av prosesser og rutiner som utnytter trusseletterretning fra samarbeidspartnere for å gjennomføre *threat hunting* i organisasjonens nettverk og systemer, og håndtere eventuelle hendelser som oppdages på bakgrunn av denne informasjonen. I denne casen satt vi ikke opp sanntidssøk i Splunk Enterprise, i motsetning til både case 1 og 2. Dermed ble det ikke generert alarmer som automatisk utløste playbooks i Splunk SOAR for å håndtere hendelsen. Dette var på grunn av casens natur og det faktum at trusseletterretningen ikke var tidligere kjent for organisasjonen. Dermed var det ikke mulig å sette opp et søk som leter etter indikatorer fra trusseletterretningen.

I denne casen har vi likevel implementert flere tiltak som støtter håndteringen av eventuelle hendelser når organisasjonen gjøres kjent med trusseletterretningen. Disse tiltakene handlet både om å lage rutiner for hva som bør gjøres dersom organisasjonen blir kjent med trusseletterretning, og om å sette opp automatiserte prosesser som kan utnytte seg av trusseletterretningen for å drive threat hunting. Rutinen som ble utarbeidet gikk ut på at når organisasjonen mottar trusseletterretning, skal de opprette en manuell sak i Splunk SOAR der trusseletterretningen kan benyttes til threat hunting.

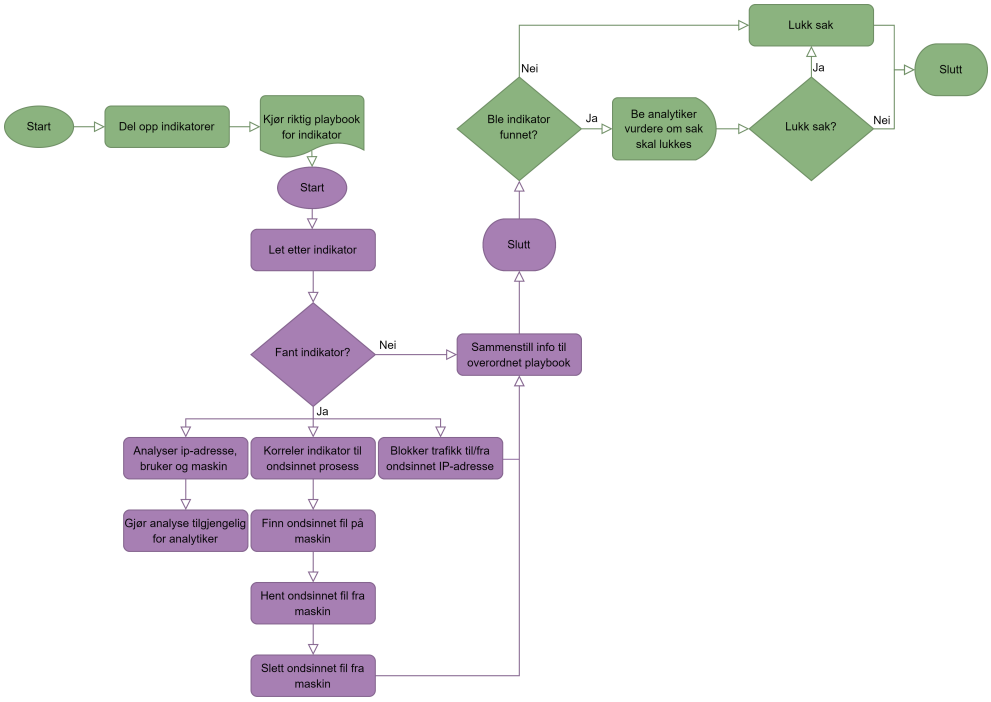
I Splunk SOAR lagde vi to playbooks som automatiserer threat hunting. Den ene playbooken ble konfigurert til å dele opp trusseletterretningen i individuelle indikatorer, og gjennomføre threat hunting for hver indikator. Den andre playbooken ble konfigurert til å gjennomføre selve threat hunting prosedyren, som inkluderer å lete etter indikatoren i organisasjonens systemer, og gjennomføre analyse og hendelseshåndtering dersom indikatoren blir oppdaget. Et overordnet flytdiagram av playbookene er vist i figur 3.18. Fargene representerer flyten separert i hver sin playbook. For ytterligere beskrivelse av playbookene, se vedlegg E.3.1.

Gjennomføring av case

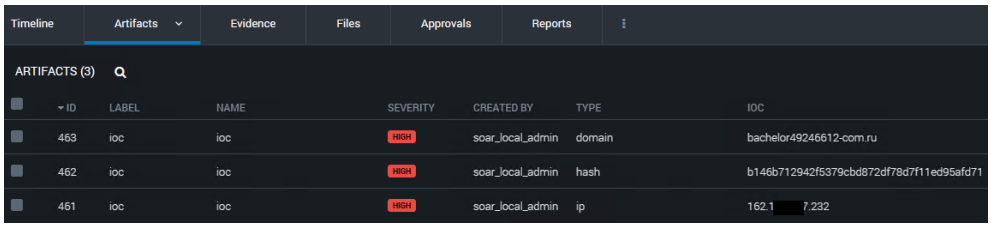
For å gjennomføre denne casen simulerte vi mottak av trusseletterretning fra Kripos ved å lage en CSV-fil som inneholdt tre indikatorer: en offentlig IP-adresse, en hash, og et domene. Merk at vi har sensurert IP-adressen ettersom dette er en offentlig IP-adresse til en tjeneste vi benyttet for å simulere casen.

```
162.1XX.XX7.232,ip
b146b712942f5379cbd872df78d7f11ed95afd71,hash
bachelor49246612-com.ru,domain
```

Deretter lagde vi en manuell sak i Splunk SOAR som vi kalte *Threat IOC Hunt*, hvor vi lastet opp CSV-filen for å generere bevis (kalt artefakter i Splunk SOAR) for hver indikator i filen. Figur 3.19 viser disse artefaktene, én for hver av de tre indikatorene vist over. Tilslutt utløste vi den første av de to playbookene, kalt *Hunt IOC*, for å gjennomføre threat hunting etter disse indikatorene. Resultatet etter gjennomføring av case 3 er beskrevet i seksjon 4.2.3.



Figur 3.18: Case 3: Flytdiagram over playbook



Figur 3.19: Case 3: Gjennomføring av case

Kapittel 4

Resultat

I denne delen av rapporten vil vi presentere resultatene for den praktiske delen som har blitt forklart i kapittel 3. Resultatene blir presentert separat og vil senere bli drøftet i kapittel 5. Funnene fra metodedelen har blitt analysert og kvalitetssikret, og seksjonene 4.1 og 4.2 redegjør for resultatene fra henholdsvis dybdeintervju og mulighetsstudie.

4.1 Dybdeintervju

Som en del av denne oppgaven har vi gjennomført dybdeintervju med ansatte fra Norsk helsenett som jobber med hendelseshåndtering og informasjonssikkerhet, med mål om å utforske relevante problemstillinger, avdekke mulige utfordringer og fallgruver, samt innhente erfaringer. I denne delen av rapporten presenterer vi funn fra dybdeintervju.

4.1.1 Hendelseshåndteringsprosessen

Et av de viktigste områdene i dybdeintervjuene var å avdekke viktigheten av hendelseshåndtering. Gjennom intervjuene undersøkte vi hva som danner grunnlaget for hendelseshåndtering som prosess og hvorfor det må prioriteres i en organisasjon. Intervjuene diskuterte også hvordan hendelseshåndtering kan organiseres i en moderne organisasjon.

Viktigheten av hendelseshåndtering

Det kom tydelig frem i intervjuene at det er enighet om viktigheten av hendelseshåndtering for Norsk helsenett. Norsk helsenett leverer kritiske tjenester til innbyggere og helsesektoren, og man vil ikke kunne ha tillit til Norsk helsenett som aktør dersom de ondsinnede hendelsene ikke avdekkes eller håndteres på en god måte. Hos Norsk helsenett er hendelseshåndtering etablert som prosess for å sikre stabil og sikker drift, med evne til å gjenvinne normaltilstand så raskt som mulig ved sikkerhetshendelser for å redusere skadeomfang- så fort som mulig under en

hendelse, for å minimere skaden. Den ene respondenten forklarte viktigheten av hendelseshåndtering med følgende analogi:

> Dersom man fjerner hendelseshåndtering, betyr ikke det at man ikke må håndtere hendelsen. Hva gjør man for eksempel hvis man har en kjele med olje som begynner å brenne? Veldig mange vil kaste vann på det, men da eksploderer oljen. I stedet kan man ha en prosess og følge denne. Der er første steg å ikke kaste vann på kjelen, men i stedet bruke et brannteppe. Poenget er å sette system på kaoset og det må være innøvd.

I tillegg ble det påpekt at hendelseshåndtering er viktig for revisjon av hendelser i ettertid. Dersom utfallet av en hendelse resulterer i brudd på konfidensialiteten, integriteten eller tilgjengeligheten til informasjon, kan organisasjonen vise til gode prosesser som forteller at organisasjonen har gjort alt i sin makt for å forhindre og håndtere hendelsen. Det er også viktig å revidere sine egne prosesser regelmessig, slik at man kan sørge for at prosessene forbedres og etterleves.

Gjennom intervjuene kom vi flere ganger inn på viktigheten av effektivisering av prosessene i hendelseshåndtering. Alle respondentene var enige om at effektiv hendelseshåndtering vil gi høy verdi tilbake til organisasjonen, da man ideelt sett kan opprettholde normaltilstand under selve sikkerhetshendelsen. I tillegg vil effektiv hendelseshåndtering bidra til å unngå videre kompromittering og tap av data, anseelse og omdømme. Den potensielle kostnaden relatert til en sikkerhetshendelse kan også bli ekstremt høy avhengig av hvor lang tid hendelseshåndteringen tar.

Organisering av hendelseshåndtering

Det kom også frem hvordan en organisasjon kan organisere hendelseshåndtering internt i organisasjonen. Representantene la i hovedsak frem egne erfaringer som blant annet innebar å etablere et operasjonssenter, et Security Operation Center (SOC) og dedikerte *incident managere* (IM). Dette ble understøttet ved at de poengterte viktigheten av å sentralisere hendelseshåndtering når organisasjonen er av en viss størrelse. Desto større en organisasjon er, desto mer kompleksitet og alarmering må tilføres for å oppnå tilstrekkelig deteksjon. Arbeidsoppgaver i et operasjonssenter inkluderer monitorering, alarmhåndtering, vaktutkalling og diverse driftsrutiner. I og med at sikkerhet har blitt et eget fagfelt vil etablering av et SOC bidra til å differensiere håndteringen av sikkerhetshendelser og vanlige driftshendelser. SOC vil operere som organisasjonens deteksjons- og responsmiljø ved sikkerhetshendelser, ettersom SOC har den sikkerhetsmessige fagekspertisen nødvendig for å detektere og respondere på slike hendelser. Dette bidrar til at operasjonssenteret skal kunne fokusere på vanlige driftshendelser.

Under hendelser som krever koordinering på tvers av organisasjonen, er en incident manager et godt tilskudd. Disse vil da være med på å koordinere håndteringen av hendelsen, samt varsle riktige parter innenfor gitte tidsrammer som

gjerne er stadfestet i tjenestenivåavtaler og kontinuitetsplaner. Videre har incident managere mandat til å innhente nødvendige ressurser, vurdere og iverksette strakstiltak i koordinasjon med fagressurser for å mitigere hendelsen, og ta beslutninger for å løse hendelser. De skal også vurdere konsekvensen av å sette inn tiltak for å løse hendelsen opp imot konsekvensen av å la være. Etter at hendelsen er håndtert skal de vurdere kvaliteten av håndteringen, og revidere prosessene slik at en tiltaksliste kan utarbeides og videresendes til relevante team. Dette fører til kontinuerlig forbedring av hendelseshåndteringsprosessen.

Selv om deteksjon og respons av sikkerhetshendelser er SOC sitt ansvarsområde, ble det også fremhevet at SOC ikke kan være sterkere enn resten av organisasjonen. Avhengig av hva som er berørt, kan SOC behøve å innhente fagkompetanse fra andre miljøer. SOC kan også struktureres slik at rollen blir todelt ved hendelseshåndtering. Det skal gis innspill for å bistå med løsningen av hendelsen. Samtidig skal den berørte tjenesten eller infrastrukturkomponenten analyseres for å danne et bilde av hva som har skjedd, hvordan det har skjedd og vurdere videre tiltak. Det blir derfor mye informasjon som må kartlegges, dokumenteres og overleveres. Norsk helsenetts SOC har hatt god suksess med å ha en eget SOC-koordinator under pågående hendelser. Dette gjør at koordinatoren kan ta seg av kommunikasjonen med de andre delene av organisasjonen, slik at de resterende sikkerhetsanalytikerne kan jobbe uforstyrret.

4.1.2 Effektivisering av hendelseshåndtering

I denne delen av intervjuene stilte vi respondentene spørsmål om opplevd effektivitet av hendelseshåndtering i Norsk helsenett. Arbeidsflyten til respondentene og de tilhørende seksjonene var også interessante underpunkter i denne delen. Dette resulterte i forskjellige synspunkt ut ifra hvilke erfaringer respondentene satt med.

Effektiviteten av hendelseshåndtering

Felles for respondentene var at hendelseshåndtering oppleves som en kontinuerlig prosess og at man alltids kan strekke seg lenger. Flere av respondentene fremhevet at flere deler av prosessen, og styringen av disse, kan effektiviseres. Samtidig er det slik at effektivisering kan medfølge ytterligere utfordringer. Blant annet ser man en økning i antall systemer som tas i bruk og som kobles opp mot hverandre og omverdenen, noe som igjen skaper et behov for mer overvåkning og deteksjon. En økning i overvåkning og deteksjon er i seg selv en bra ting, så lenge mengden alarmer ikke blir uhåndterlig. Hvis mengden blir uhåndterlig, kan ressursutfordringer oppstå og effektiviteten reduseres.

Det ble fremhevet at tiden det tar å rette opp feil ved hendelser er et viktig mål for effektiviteten til hendelseshåndtering. Mange organisasjoner prioriterer varsling til interessenter, men i noen tilfeller kan dette gå utover evnen til å håndtere selve hendelsen. Det er derfor viktig å finne riktig balansepunkt mellom disse

to områdene, for å prioritere ressursene mest mulig effektivt. Samtidig poengterer flere av respondentene at det kan være utfordrende å måle effektiviteten til hendelseshåndtering – spesielt når det stilles forskjellige krav fra forskjellige interessenter som kunder, samarbeidspartnere og andre deler av organisasjonen.

Evne til å samhandle ble lagt frem som et annet viktig område for å oppnå effektivisering av hendelseshåndtering. Med andre ord er bevisstgjøring av behovene til forskjellige ledd i organisasjonen viktig for å oppnå effektivitet. Respondentene trakk også frem at det er viktig å bygge etablerte rutiner og prosesser for hendelseshåndtering basert på erfaringer fra de ansatte i organisasjonen, men det kan oppstå utfordringer dersom omfanget og erfaringskravet blir for høyt. Det er viktig å unngå informasjonssiloer innad i organisasjonen, men heller oppfordre til samhandling og erfaringsdeling.

I løpet av Norsk helsenetts tid som statlig foretak, har det blitt lagt ned store ressurser for å forbedre hendelseshåndteringsprosessen. Likevel kommuniserer respondentene at det å videreutvikle egne prosesser og holde tritt med den raske teknologiske utviklingen oppleves som ressurskrevende og kostbart. Teknologien for å effektivisere er der, men flere av respondentene påpekte at det kan medfølge utfordringer og risiko dersom man investerer i teknologien uten tilstrekkelig behovsgrunnlag. Respondentene beskrev innføringen av et SOC i Norsk helsenett som en suksesshistorie. Innføringen har blant annet ført til økt sikkerhetskompetanse i organisasjonen og evne til å ta tak i sikkerhetshendelser raskere. I tillegg bidro et dedikert SOC til at mer ressurser ble tilegnet sikkerhetsarbeid og at sikkerhetskultur har fått en viktigere rolle i organisasjonen.

Arbeidsflyt under hendelseshåndtering

Ut fra resultatene i intervjuene kan vi sette sammen en felles beskrivelse av hvordan hendelseshåndteringsprosessen ser ut på generell basis. En hendelse oppstår av forskjellige årsaker; vanlige triggere er alarmer som varsler organisasjonens SOC via deteksjonssystemer. I tillegg kan hendelser trigges etter tips fra interne avdelinger, samarbeidsorganer eller kunder. Videre steg i arbeidet går på å følge forhåndsdefinerte playbooks eller andre vurderingskriterier for å danne et oversiktsbilde av situasjonen. Dette kalles gjerne for triage-delen av prosessen, hvor hendelser kategoriseres og vurderes basert på loggførte eventer og andre triggere. Eventuelle strakstiltak iverksettes her, før man går over til en analysefase der ytterligere informasjon samles inn og man bygger situasjonsforståelse. Videre løper prosessen inn i en håndteringsfase hvor tiltak for feilretting iverksettes og normaltilstand gjenopprettes. Her vil nødvendige tverrfaglige ressurser kobles på for å løse hendelsen så effektivt som mulig.

Ifølge respondentene er en god hendelseshåndteringsprosess en prosess der partene spiller hverandre gode og gir hverandre beslutningsstøtte. En stor del av håndteringsprosessen handler om de tekniske aspektene, men det handler også om å levere en situasjonsforståelse oppover og på tvers av organisasjonen. Jevnlige møter med kunder og interessenter under håndtering av hendelsen er viktig

for å opprettholde god samhandling og sikre at nødvendige parter holdes oppdatert. Når en hendelse blir løst, utarbeides det en sluttrapport og, om nødvendig, en hendelsesrevisjon for å gjennomgå hendelsen, utarbeide tiltak for å forhindre fremtidige hendelser, og lære av hva som kan gjøres bedre til neste gang.

Det ble fremhevet at forskjellige seksjoner og fagteam vil ha ulik tilnærming til hendelseshåndtering. Likevel er det en felles forståelse for at hovedfokuset ved hendelseshåndtering er å gjenopprette normaltilstand så fort som mulig. I tillegg er det stor enighet om at hendelseshåndteringsprosessen i større eller mindre grad kan og bør automatiseres. Dette vil kunne effektivisere prosessen og skape merverdi for organisasjonen ved at tiden blir brukt på de mest nødvendige og tidskritiske oppgavene.

Effektiv kategorisering av hendelser

Under deteksjons- og analysefasen av hendelseshåndtering er det viktig å kategorisere hendelser korrekt for mest mulig effektiv håndtering. Respondentene diskuterte noen av de utfordringene som kan oppstå under kategorisering av hendelser – og skille mellom rene driftshendelser og sikkerhetshendelser. Korrekt kategorisering legger grunnlaget for videre hendelseshåndtering og bestemmer blant annet hvilke seksjoner og ressurser som blir involvert under håndteringen. Respondentene poengterte også viktigheten av å starte håndtering så tidlig som mulig ved sikkerhetshendelser. Dette er på grunn av det korte vinduet som ofte eksisterer for å begrense alvorlige konsekvenser og gjenopprette normaltilstand før sikkerhetshendelser har muligheten til å eskalere. Gode rutiner, etablerte prosesser og samhandling mellom seksjoner i organisasjonen trekkes frem som viktige suksessfaktorer for å sikre rask og korrekt kategorisering av hendelser.

Under intervjuene ble distribuerte tjenestenektangrep ofte trukket frem som en typisk hendelse der skillet mellom driftshendelse og sikkerhetshendelse kan være vanskelig å trekke. Korrekt kategorisering av en slik hendelse kan være utfordrende. Mange rene driftsrelaterte hendelser vil ved første øyekast kunne føre til samme symptomer som et distribuert tjenestenektangrep, selv om bakenforliggende årsak og responstaktikk vil være vidt forskjellig. Respondentene påpekte at det i de fleste tilfeller ikke vil ta lang tid før det er mulig å fastslå at et distribuert tjenestenektangrep sannsynligvis står bak. Likevel kan dyrebar tid gå tapt dersom hendelsen ikke kategoriseres riktig så fort som mulig. Det nevnes blant annet at tidlig kategorisering kan føre til at mitigerende tiltak som trafikkfiltrering og -blokkering kan settes i gang før tjenester settes helt ut av spill. I tillegg er det alltids en risiko for at tjenestenektangrep benyttes som en avledning — eller er et forvarsel — for ytterligere sikkerhetshendelser.

Prioritering av ressurser

Som nevnt i forrige undertittel er hendelseshåndtering en prosess under kontinuerlig utvikling. En organisasjon har ikke ubegrenset med ressurser, derfor er det viktig å prioritere ressurser der det vil ha størst innvirkning på effektiviteten

til hendelseshåndtering. Basert på en organisasjons strategiske og langsiktige mål bør det investeres mer tid og ressurser på å jobbe mer proaktivt istedenfor reaktivt. Respondentene påpekte at proaktivt arbeid bidrar til å redusere sannsynligheten og konsekvensen av hendelser. Samtidig er det ikke alltid like lett å vurdere kost-nytte-verdien av en slik investering. Det er enklere å vurdere effektiviteten av en håndtert hendelse, enn å spekulere effektiviteten av tiltak mot hendelser som kan forekomme.

Under intervjuene ble det nevnt flere utfordringer knyttet opp mot prioritering av ressurser og midler for å effektivisere hendelseshåndtering. Investeringer i teknologi og verktøy må vurderes opp imot kost-nytte-verdi, ved å gi avkastning i form av enten redusert risiko, sparte kostnader eller økt verdiskapning. Evnen til å kommunisere både behov og nytteverdi av investeringer oppover i organisasjonen kan oppleves som utfordrende, men er like fullt viktig. I tillegg må kostnaden ved anskaffelse, implementering og vedlikehold av ny teknologi inkluderes i vurderingen. En kartlegging av nåværende behov og prosesser må være på plass før man beslutter å investere i ny teknologi. Et veikart ble nevnt som en god måte å planlegge veien videre og vurdere kost-nytte-verdi av mulige investeringer. I tillegg blir det lettere å sette realistiske forventninger og tidsfrister.

4.1.3 Automatisering av hendelseshåndtering

Gjennom våre dybdeintervju har vi avdekket et bredt spekter av arbeidsoppgaver og deler ved hendelseshåndtering og informasjonssikkerhet som er egnet for automatisering. Gjennom automatisering er det mulig å skape effektivisering i form av sparte ressurser og bedre prioriteringer. I denne seksjonen diskuteres temaer relatert til automatisering av de forskjellige fasene av hendelseshåndtering, samt begrensninger og fallgruver ved automatisering.

Automatisert deteksjon

Automatisering har stort potensiale innenfor deteksjon av eventer og hendelser, og det er her det har blitt – og blir – brukt i størst grad. Deteksjonssystemer bruker automatisering i stor grad for å oppdage unormal eller ondsinnet aktivitet som utløser generering av alarmer. Evnen til å detektere eventer og hendelser raskt og effektivt legger til rette for reduserte kostnader og bedre prioritering av ressurser.

Utrulling av leverandørtjenester, for eksempel verktøy for utvidet deteksjon og respons (XDR), gir en organisasjon stor verdi i form av forhåndsdefinert og utprøvd deteksjonslogikk. Respondentene påpekte at dette muliggjør deteksjon av en stor mengde indikatorer på eventer og hendelser, uten at ansatte må utvikle og teste all deteksjonslogikk. Samtidig er det begrenset hvor langt en organisasjon kommer ved hjelp av kun forhåndsdefinert deteksjonslogikk. Slik logikk er i stor grad basert på kjente trusler (eng: *common threats*) med deteksjon av kjente signaturer og indikatorer. Mange organisasjoner slik som Norsk helsenett har en stor grad av egenutviklede tjenester. For slike organisasjoner er det viktig å jobbe med deteksjon av ukjente trusler (eng: *uncommon threats*), som er spesifikke for

organisasjoner og de tjenestene de utvikler, drifter og leverer. Dette arbeidet krever utvikling av deteksjonslogikk, og bruk av automatisering for å rulle ut og ta i bruk logikken i organisasjonens deteksjonssystemer.

Et annet viktig område er hyppig utrulling av automatisering ved nye trusler og sårbarheter, slik som den velkjente Log4j-sårbarheten. I situasjoner slik som dette er det ofte nødvendig å arbeide raskt for å ikke havne på bakfoten. Her kan automatisering brukes effektivt. Respondentene nevnte blant annet bruk av automatiske søk i logger som detekterer trafikk som inneholder den ondartede Log4j-nyttelasten, og deretter varsler, analyserer og starter nødvendige tiltak.

Håndtering av store datamengder

For å muliggjøre deteksjon av eventer og hendelser, kreves først og fremst et godt datagrunnlag. Et gjengående tema under intervjuene var viktigheten av å synliggjøre data – det vil si å samle inn data og logger fra en rekke forskjellige kilder – før utvikling av deteksjonslogikk. Respondentene nevnte for eksempel viktigheten av å først implementere systemer for logging fra endepunkter, før man utvikler deteksjonslogikk for denne type data. Det er ingen verdi i å utvikle deteksjonslogikk som skal oppdage eventer som ikke først er synliggjort og tilgjengeliggjort.

Samtidig påpekte respondentene utfordringen som følger med stor synliggjøring av data i kombinasjon med god deteksjonslogikk: man kan få svært mange eventer som må håndteres. Uten en eller flere former for filtrering, gjerne kalt tuning, kan en organisasjon som Norsk helsenett forvente tusenvis av eventer og alarmer per dag. Av disse er det kun en liten brøkdel som normalt eskaleres til en hendelse, og ressurser bør i størst mulig grad fokuseres mot disse. Et viktig grep for effektiv hendelseshåndtering er derfor god tuning av eventer.

Innunder temaet tuning ble det under intervjuene diskutert risikoen for å tune bort for mange eventer, enten ved å filtrere de bort i sin helhet eller å etterlate dem fullstendig til automatiserte systemer. Respondentene vi snakket med anså risikoen som lav. Hvis mengden eventer er for stor vil uansett kvaliteten på analysearbeidet og håndteringen reduseres, og sannsynligheten øker for at eventer kvitteres uten tilstrekkelig behandling. Derfor poengterte heller respondentene viktigheten av å øke kvaliteten på tuning og de eventer som skal behandles, slik at ressursene og prioriteringene i størst mulig grad kan fokuseres dit organisasjonen ser størst nytteverdi. Et eksempel som kom frem var en tiltenkt tuning der 98 prosent av det som filtreres bort var unyttige eventer, men to prosent var eventer som det i utgangspunktet ville vært verdi i å plukke opp. Dette innebærer altså en risiko for at viktige eventer ikke blir fanget opp, men i realiteten vil det være vanskelig å detektere de to prosentene dersom man samtidig må grave gjennom de nittiåtte andre.

Respondentene foreslo tiltak for å mitigere konsekvensene av at viktige eventer unngår deteksjon. Tuning av data bør gjennomføres etter synliggjøring av data, men før deteksjonslogikk. På den måten vil dataen fortsatt være tilgjengelig dersom den behøves under håndteringen av en hendelse, uten at deteksjonslogikken

genererer en alarm. Sikkerhetsmekanismene til en organisasjon bør samtidig basere seg på en forsvar-i-dybden-arkitektur. Det betyr at selv om en event ikke fanges opp av ett deteksjonslag, bør en potensiell sikkerhetshendelse fortsatt detekteres eller stoppes av en eller flere andre lag i arkitekturen.

Automatisert analyse

Under intervjuene kom det frem at analysefasen av hendelseshåndtering kan og bør automatiseres i stor grad. Dette er for å lette arbeidsbelastningen på de ansatte og effektivisere repetitive og rutinepregede arbeidsoppgaver. Det er nødvendig å kunne fokusere analysearbeid mot de mest kritiske sakene. Samtidig er det viktig å prioritere unike og nye saker ved å effektivisere analysen av kjente hendelsesforløp der prosessene kan automatiseres basert på eksisterende prosesser. Som tidligere nevnt poengterte respondentene viktigheten av å gjøre tiltak mot store mengder eventer. Dette skaper en håndterlig arbeidsmengde og prioriterer ressurser der de er viktigst. Likevel er det begrenset hvor mye som kan eller bør tunes bort, og dette alene vil derfor ofte ikke være tilstrekkelig.

Respondentene trakk frem flere andre eksempler på bruk av automatisering for ytterligere effektivisering under analysefasen. Ved hjelp av konfigurasjonsstyring og en tjenestekatalog (eng: *CMDB*), kan avhengigheter og tilkoblede systemer og tjenester kartlegges. Dermed kan automatiserte systemer hente ut et mer komplett bilde av berørte eller utsatte tjenester og systemer under hendelser. Et annet eksempel som ble nevnt er bruk av verktøy for sikkerhetsorkestrering, -automatisering og -respons (eng: *SOAR*). Med slike verktøy kan tilkoblinger settes opp til tredjepartstjenester som spesialiserer seg på analyse av for eksempel indikatorer, mistenkelige filer eller detektert skadevare. Under Log4j-saken kunne dette ha blitt benyttet for automatisk analyse av IP-adresser som blir detektert i logger, som diskutert i forrige seksjon. På mange måter kan et slikt eller lignende verktøy fungere som et nav som sikkerhetsanalytikere kan jobbe opp mot, der automatiserte prosesser ligger samlet og er tilgjengeliggjort.

Automatisert håndtering

Et område av hendelseshåndtering som ikke alltid har vært like aktuell for automatisering, er fasen for mitigering, utryddelse og gjenopprettelse. Det er i denne fasen tiltak iverksettes for å mitigere hendelser og gjenopprette normaltilstand. Respondentene fremhevet flere områder av denne fasen hvor det er mulig å automatisere.

Mange hendelser oppstår ved regelmessige intervaller og kan forventes å være en trussel i fremtiden. Et eksempel respondentene fra Norsk helsenett trakk frem er phishing-angrep, der problemet kan forventes å vedvare, og håndtering følger lignende spor i mange av tilfellene. Dette er en ypperlig respondent for automatisering. Håndteringstiltakene som kan automatiseres er blant annet varsling om pågående phishing-kampanjer, og bestillinger for å ta ned domener eller blokkere

telefonnummer og e-postadresser involvert i phishingangrep. Generelt er automatisk varsling ved detektert og bekreftet degradert tjenestetilstand eller pågående hendelser en god respondent for automatisering, i de tilfeller der bakenforliggende årsak er kjent og repeterende. Det er stort potensiale for bruk av SOAR-verktøy for å automatisere denne fasen av hendelseshåndtering. Slike verktøy kan koble sammen deteksjons- og analysefasen, med verktøy og prosesser som mitigerer og håndterer hendelser. På den måten kan prosessene slås sammen i automatiserte prosedyrer og rutiner, kalt playbooks.

Anskaffelse og implementering av automatiseringsverktøy

Effektivisering av hendelseshåndtering med automatisering kan være kostbart dersom en organisasjon ikke er moden for å automatisere deler av sin hendelseshåndteringsprosess. Respondentene vi har snakket med foreslo flere vurderingsgrunnlag som må gjennomføres og tjenester som bør være på plass før effektiv automatisering kan finne sted. Blant annet bør sentrale forutsetninger slik som et godt saksbehandlingssystem, en velfungerende og oppdatert CMDB, og ikke minst tilstrekkelig datagrunnlag og deteksjonslogikk. Uten data å agere på, og tjenester å koble opp mot, er det lite nytteverdi i å ha verktøy for automatisk hendelseshåndtering. I tillegg, for å underbygge effektive prosesser og samhandling i organisasjonen, kom det frem at det er viktig å ha forankring og støtte av verktøyene som benyttes.

Et annet viktig poeng som ble fremmet er behovet for etablerte prosesser. Organisasjoner bør ikke gå til innkjøp av dyre automatiseringsverktøy før de har prosesser som kan og bør automatiseres. Når prosesser for hendelseshåndtering er på plass, har man samtidig et grunnlag for hva som kan automatiseres og hvor det ligger størst effektivitetsgevinst. Teknologi skal underbygge prosess, ikke motsatt. Teknologien skal løse de arbeidsoppgaver som eksisterer; ressurser bør ikke brukes på å finne arbeidsoppgaver til teknologien.

Når det gjelder produktvalg, for eksempel valg av et SOAR-verktøy, er det mulig å basere seg på en stor grad av åpen og gratis programvare. Likevel vil det for de fleste organisasjoner være nødvendig eller ønskelig å gå til anskaffelse av verktøy for hendelseshåndtering og automatisering fra kommersielle leverandører. Respondentene påpekte at disse leverandørene kan tilby skreddersydde løsninger, samt oppfølging i form av støtte og service av produktene sine. Dette kan være kostbare investeringer, og det er derfor viktig at nytteverdien til automatiseringen er større enn kostnaden ved anskaffelse, implementering og drift av produktene. Derfor bør organisasjoner først avdekke sine automatiserings- og effektivitetsbehov, og kartlegge områder som er egnet og mulig å automatisere, før de går til anskaffelse av disse produktene.

Et annet viktig aspekt som respondentene trakk frem, er den generelle modningsgraden til en organisasjon i forhold til teknologi for automatisering og bruken av dette. Det er viktig at organisasjonen har en plan for å iverksette automatisering. Det kreves mye arbeid for å automatisere, for eksempel design av detek-

sjonslogikk eller playbooks i et SOAR-verktøy. Det må derfor eksistere tilstrekkelig ressurser i form av tid og kunnskap for å komme i gang på en god måte. Respondentene påpekte også at det kan være lurt å begynne smått når en organisasjon skal rulle ut automatisering i større grad. Ved å automatisere enklere, repeterende tiltak og arbeidsoppgaver med liten grad av usikkerhet først, vil det kunne gi dobbel effekt. Først og fremst vil organisasjonen gradvis tilvenne seg en hverdag med mer automatisering og i tillegg vil det frigjøre ressurser til vanskeligere arbeidsoppgaver og problemstillinger.

Automatisering for proaktivt arbeid og redusert arbeidsbelastning

Gjennom intervjuene var ønsket om en mer proaktiv arbeidshverdag et gjentagende tema og motivasjon bak økt automatisering og effektivisering av hendelseshåndtering. Å bruke tid på repeterende arbeidsoppgaver og behandle en stor mengde like eventer og hendelser hver dag kan ha en negativ effekt på ansatte i en organisasjon og deres effektivitet. Ved å automatisere de repetitive arbeidsoppgavene, kan tid frigjøres og fokus flyttes fra reaktivt arbeid til en større grad av proaktivt arbeid. Dette er viktig for å holde følge med tempoet til mulige trusselaktører, og forhåpentligvis være bedre forberedt dersom alvorlige hendelser oppstår. Et annet tema som kommer frem, er utbrenthet blant personell som jobber med hendelseshåndtering og sikkerhetsanalyse. Ifølge respondentene var varierte og proaktive arbeidsoppgaver, og mindre tid brukt på repeterende arbeidsoppgaver viktige tiltak for å redusere utbrenthet. Selv om automatisering er arbeid som kan være utfordrende og tidkrevende, kan det oppleves det som givende og motiverende arbeidsoppgaver som i tillegg er verdifulle investeringer for en organisasjon.

Utfordringer og begrensninger ved automatisering

Under intervjuene har vi spurt respondentene om både muligheter og begrensninger ved bruk av automatisering for å effektivisere hendelseshåndtering. Det ble nevnt flere fallgruver og begrensninger ved automatisering. Norsk helsenett, som drifter systemer og tjenester som behandler og tilrettelegger for flyt av person- og helsedata, har informasjonssikkerhet og dataansvar som et av deres fremste fokusområder. I lys av dette er det ikke alle arbeidsflyter og oppgaver som bør automatiseres. Dette handler igjen om modningsgraden til en organisasjon, og om at det er viktig med en plan for automatisering dersom dette skal benyttes i større grad.

Automatisering for automatiseringens skyld bør også unngås. Her trakk respondentene frem risikoen for å automatisere for mye, i tillegg til å automatisere arbeidsoppgaver som ikke er egnet for automatisering. Dette vil kunne gå utover kvaliteten på arbeidsoppgavene. I tillegg er det viktig å ikke automatisere bort såpass mye at de ansatte i en organisasjon mister kunnskap om hvordan systemer og tjenester fungerer i bunn. Dette er viktig kunnskap for å kunne forstå hvorfor eventer og hendelser oppstår, og viktig kunnskap for å kunne analysere og

løse dem. Dersom for mye automatiseres bort, kan det ha en negativ effekt på en organisasjons evne til å løse hendelser på en effektiv måte.

Respondentene trakk også frem mulige sårbarheter ved bruk av automatisering. En trusselaktør som oppnår kunnskap om en organisasjons automatiserte prosesser kan utnytte dette for å skape ytterligere hendelser. En trusselaktør kan for eksempel misbruke automatisering som blokkerer adresser som sender gitte typer trafikk, ved å *spoofe* avsenderadressen slik at automatiseringen blokkerer legitime adresser eller tjenester og dermed skaper en hendelse. Andre eksempler er misbruk av aggressivt konfigurerte deteksjonssystemer som isolerer tjenester eller endepunkter når de detekterer visse indikatorer. Disse eksemplene, i tillegg til andre mulige sårbarheter, er en organisasjon nødt til å ta i betraktning ved utrulling av automatiserte løsninger.

Gjennom intervjuene var det bred enighet om at menneskelig involvering i hendelseshåndtering er, og kommer til å forbli, svært viktig. Mange vurderinger er vanskelig for en datamaskin å bedømme korrekt med stor nok sannsynlighet, ettersom de baserer seg på vurderingsgrunnlag som for mange hendelser kan være vanskelig å fastsette eller lage regler for. I slike situasjoner er det viktig at organisasjonens ansatte fortsatt har muligheten til å kontrollere og ta beslutninger. Andre eksempler er tiltak som har økonomiske konsekvenser, som å stenge ned tjenester og systemer. I slike situasjoner er en datamaskin avhengig av å ha konkret og sikker data, i tillegg til å kunne bedømme den økonomiske konsekvensen for organisasjonen. Det er også en risiko med falske positiver, som kan være betraktelig lettere for et menneske å oppdage og vurdere enn for en datamaskin.

Maskinlæring og automatisering

I tillegg til ren regelbasert automatisering, trakk flere av respondentene frem potensialet til maskinlæring som en del av hendelseshåndtering, og hvordan dette vil bli viktigere og viktigere i årene som kommer. Under dagens digitale trusselbilde må man forvente at trusselaktører benytter seg av en stor grad av både automatisering og maskinlæring, og forsvarsmekanismer er nødt til å tilpasse seg dette.

Respondentene diskuterte blant annet hvordan maskinlæring kan benyttes for å oppdage eventer og hendelser gjennom anomalitetsdeteksjon. På den måten kan en datamaskin gjennomgå store datamengder og plukke ut eventer som forårsaker avvik i normalbilde til et system eller tjeneste. Slike avvik kan være for subtile til at menneskelig analyse eller automatisert deteksjonslogikk vil plukke det opp. Dette kan også bidra til raskere deteksjon av nulldagssårbarheter eller indikatorer uten definerte signaturer, ettersom modellen ikke er avhengig av å ha sett den spesifikke signaturen tidligere. Et annet eksempel som ble diskutert er gruppering. Det innebærer at en maskinlæringsmodell utfører klyngeanalyse på store datamengder, og deretter avdekker avvik blant eventene som kan indikere kritiske hendelser. Dermed kan store mengder data fôres inn i slike modeller, og maskinen kan plukke ut det som er mest vesentlig og unikt uten at et menneske må gå gjennom hele datagrunnlaget.

I likhet med automatisering er det likevel noen begrensninger ved maskinlæring som må tas hensyn til. Respondentene nevnte blant annet utfordringer relatert til for lite kunnskap om maskinlæringsmodeller som tas i bruk, og hvordan de fungerer. De trakk også frem risikoer rundt falske positiver, samt det å tillegge for mye tillit til maskinlæring under beslutninger og vanskelige vurderingsspørsmål. På grunn av dette er det viktig å være varsom ved bruk av maskinlæring. Det kan være svært god beslutningsstøtte, men det er usikkerhet og risikoer knyttet til å la maskinlæringsmodeller gjøre endelige beslutninger.

4.2 Mulighetsstudie

Følgende underseksjoner redegjør for resultatet etter casene i mulighetsstudie. Formålet med disse brukstestene var å undersøke om verktøyene og teknologiene oppnådde ønsket resultat og håndterte hendelsene. I tillegg ønsket vi å samle inn erfaringer rundt brukervennligheten og verdiskapningen som verktøyene og teknologiene gir. Disse refleksjonene vil senere benyttes for å ta stilling til og drøfte oppgavens forskningsspørsmål, og hvorvidt denne formen for hendelseshåndtering kan erstatte – eller opptre som et tillegg til – manuell hendelseshåndtering.

4.2.1 Case 1

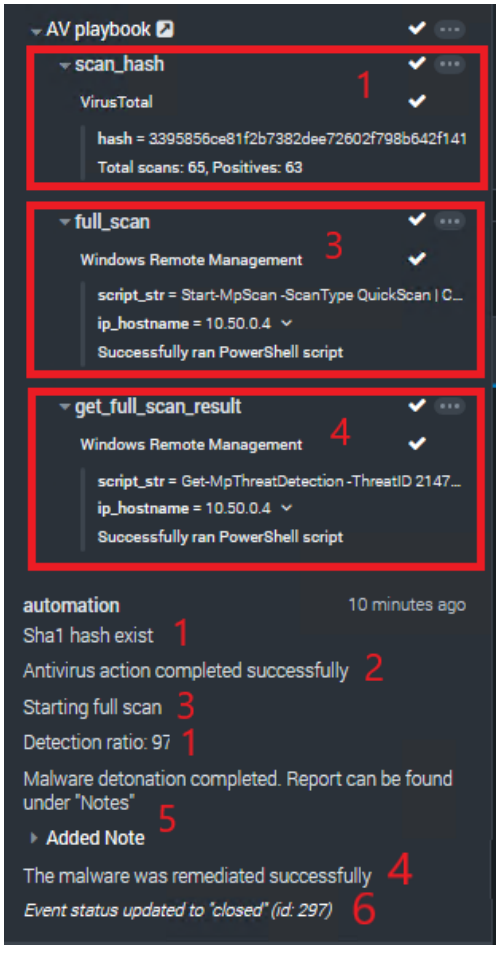
Vi utløste casen ved å simulere nedlasting av ondsinnet skadevare til en klientmaskin. Dette resulterte i et inngrep fra Microsoft Defender, da verktøyet kjente igjen skadevarens hash. Inngrepet genererte flere sikkerhetseventer i loggkildene, som er en forutsetning for å kunne håndtere hendelsen. Videre viser figur 4.1 at Splunk Enterprise detekterte sikkerhetseventene som ble generert av antivirus-verktøyet. Figuren viser resultatet av et sanntidssøk (se vedlegg E.1.1 for oppsett av søk), og her ser man at det har blitt fanget opp et logginnslag med *EventCode = 1116*. Dette betyr at Microsoft Defender har detektert skadevare på en maskin[1]. Gjennom søket ble flere eventer slått sammen til ett søkeresultat og dermed videresendt til Splunk SOAR, som diskutert i vedlegg E.1.1, før

Når søkeresultatet ble sendt til Splunk SOAR, medførte dette opprettelse av en ny event i verktøyet. Innholdet i søkeresultatet ble deretter lagt til som en artefakt til eventen. Å strukturere hendelsen på denne måten ble opplevd som verdiskapende til håndteringen. Årsaken til dette er at all informasjon relatert til hendelsen ble samlet på ett sted, noe som gjorde at vi kun trengte å forholde oss til ett grensesnitt gjennom hele håndteringen. Dette er spesielt verdiskapende når håndteringen av en hendelse delautomatiseres, da nødvendig beslutningsgrunnlag for videre håndtering ofte er tilgjengelig i verktøyet. For denne hendelsen, ble hele håndteringen automatisert. En forhåndsdefinert playbook ble automatisk startet etter at eventen ble opprettet.

[1] `https://learn.microsoft.com/en-us/microsoft-365/security/defender-endpoint/troubleshoot-microsoft-defender-antivirus?view=o365-worldwide`

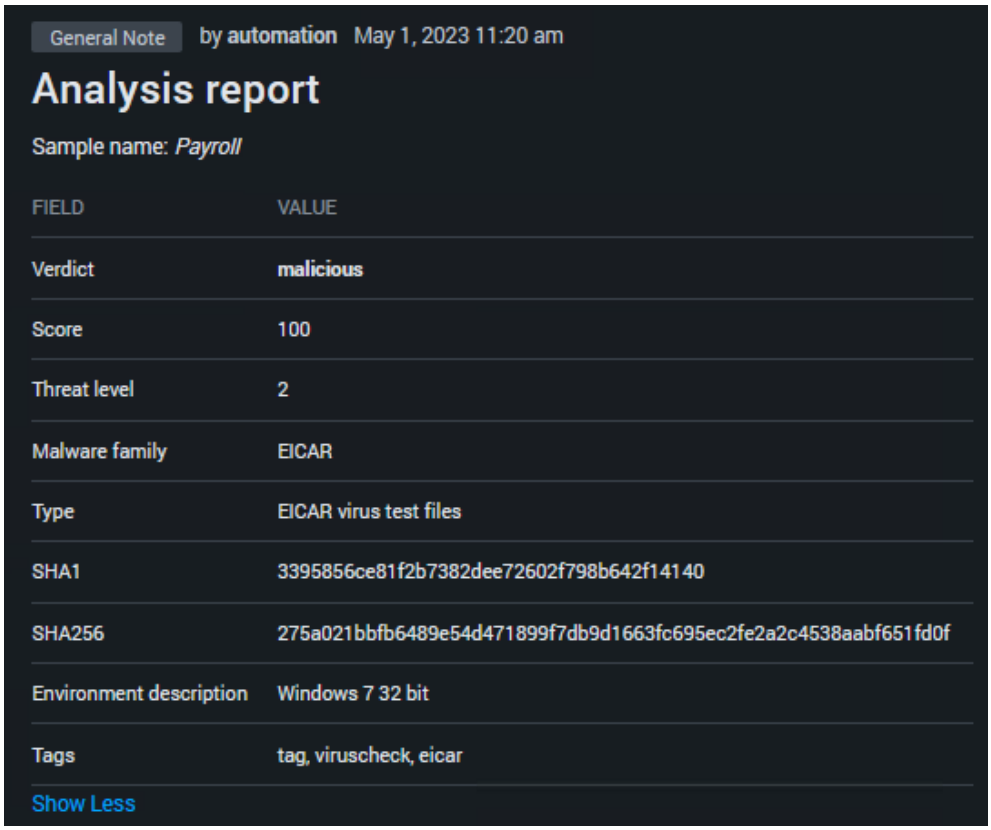
i	Time	Event
>	5/1/23 1:20:22.000 PM	05/01/2023 01:20:22 PM LogName=Microsoft-Windows-Windows Defender/Operational EventCode=1116 EventType=3 Show all 29 lines host = vm-windows-client

Figur 4.1: Case 1: Sikkerhetseventer fanget opp av Splunk Enterprise



Figur 4.2: Case 1: Playbook utløst for event

Som man kan se i figur 4.2, ble skadevarens hash analysert med VirusTotal. Hashen ble sjekket mot 65 ulike antivirus-skannere, og av disse ble hashen identifisert som ondsinnet av 63 scannere. Videre sjekket playbooken hvorvidt Microsoft Defender sitt inngrep var vellykket, når skadevaren ble identifisert for første gang. Det ble også iverksatt en full scan for å sjekke alle filer og programmer på maskinen for skadevare. I et parallelt løp til full scan, ble skadevaren analysert med Hybrid Analysis[2], som er en nettbasert detonasjonstjeneste der det er mulig å laste opp skadevare for automatisk analyse, samt gjøre oppslag i tidligere analyser. Som vist i figur 4.3, ble skadevareanalysen lagt til som et notat til eventen. På dette tidspunktet var hendelsen håndtert helautomatisk av Microsoft Defender og SOAR-verktøyet. Hendelsen ble derfor lukket automatisk, uten behov for menneskelig involvering.



Figur 4.3: Case 1: Rapport etter skadevareanalyse

Oppsummering av case 1

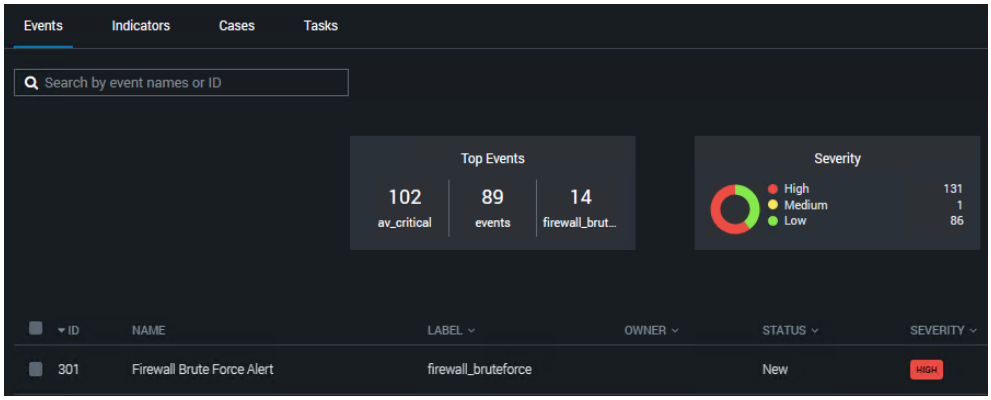
Håndteringen av denne casen var helautomatisert og utfallet var i samsvar med forventet resultat. Den største gevinsten i denne casen var hvor raskt data ble ana-

[2] `https://www.hybrid-analysis.com/`

lysert, samt hvor raskt ny data ble hentet inn til håndteringen. Implementering av playbooken tok en del tid, men til gjengjeld er dette noe som kan gjenbrukes videre. Erfaringsmessig opplevde vi at det kan være hensiktsmessig å dele opp logikken til en playbook i flere modulære playbooks med mindre omfang. Dette gjør det mulig å gjenbruke logikken når nye playbooks skal implementeres. I et virkelig scenario kan det være nødvendig å gjennomføre ytteligere tiltak enn det ble gjort i denne casen. Likevel illustrerer denne casen en hendelse der automatiserte verktøy potensielt kan erstatte manuell hendelseshåndtering i sin helhet. I denne casen vil det avhenge av skadevarens kritikalitet og hvorvidt Microsoft Defender håndterte skadevaren. Vi opplevde uansett automatiseringen av denne casen som verdiskapende, både som et tillegg eller en erstatning til manuell hendelseshåndtering.

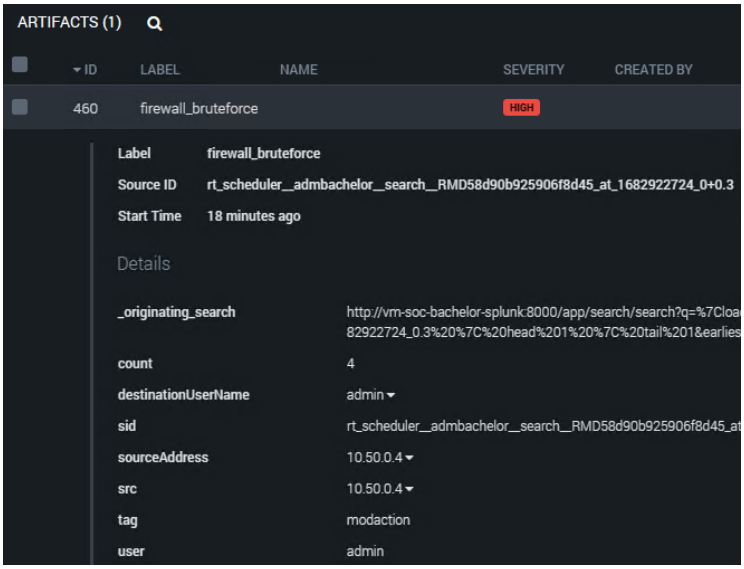
4.2.2 Case 2

Vi startet casen, som beskrevet i seksjon 3.2.8, med å simulere et *brute force*-angrep mot brannmurens administratorgrensesnitt. Dette ble gjennomført i form av fire mislykkede forsøk mot brukeren `admin` fra IP-adressen `10.50.0.4` i løpet av et minutt. Disse påloggingsforsøkene ble detektert umiddelbart av søket vi konfigurerte i Splunk Enterprise (se vedlegg E.2.1), og en alarm ble samtidig generert og videresendt til Splunk SOAR, som vist i figur 4.4. I saken som ble automatisk opprettet i Splunk SOAR ble søkeresultatet som inneholdt IP-adressen som sto bak hendelsen, brukeren som ble forsøkt pålogget, og antall mislykkede forsøk, lagt ved som en artefakt. Dette er vist i figur 4.5.



Figur 4.4: Case 2: Event mottatt i Splunk SOAR

Eventen som ble generert i Splunk SOAR er koblet til playbooken *Firewall Brute Force Attack*. Dette førte til at playbooken ble utløst automatisk, og det betyr at automatisk hendelseshåndtering ble startet så fort antall mislykkede forsøk gikk over terskelverdien på tre forsøk i løpet av et minutt. Figur 4.6 viser aktivitetshistorikken til playbooken som den ble dokumentert i Splunk SOAR. Tallene i figuren viser til rekkefølgen handlingene i playbooken ble utført. Først ble

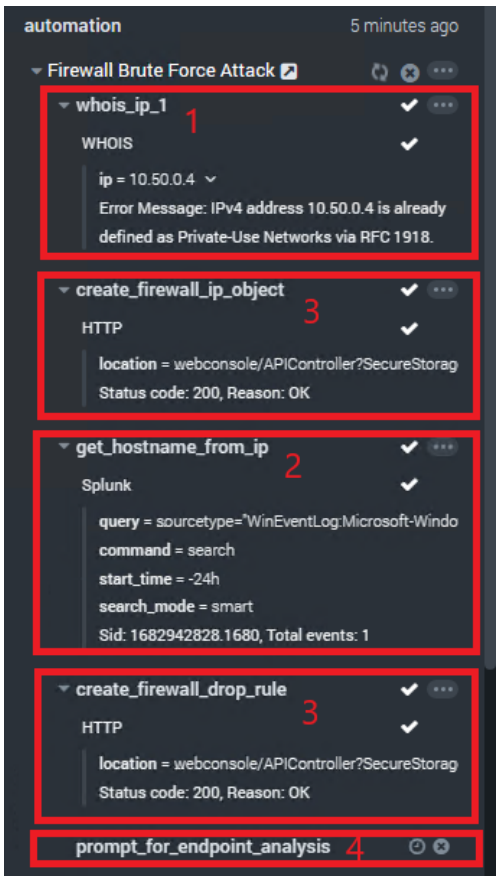


Figur 4.5: Case 2: Artefakt til event

et *WHOIS IP*-oppslag[3] gjort mot IP-adressen som forårsaket *brute force*-angrepet (`10.50.0.4`), med formål om å avdekke om angrepet stammet fra en intern eller ekstern IP-adresse. Dette er viktig informasjon for at playbooken skulle kunne håndtere hendelsen videre automatisk. Resultatet fra oppslaget returnerte med beskjed om at dette var en intern IP-adresse.

Dette sendte playbooken ned arbeidsflyten for håndtering av *brute force*-angrep fra interne IP-adresser, der neste steg var å gjøre et søk i Splunk Enterprise for å hente ut maskinnavnet til IP-adressen som forårsaket eventen. Dette returnerte med maskinnavnet `vm-windows-client`. Dette gjorde det mulig for playbooken å benytte maskinnavnet under dokumentering av hendelsen, og for videre handlinger. Samtidig ble en parallell flyt startet for å opprette en blokkeringsregel i Sophos Firewall som stanset all trafikk fra IP-adressen. Formålet med dette var å automatisk stoppe *brute force*-angrepet så fort som mulig, slik at en eventuell trusselaktør ikke får mulighet til å lykkes med angrepet under håndteringen av hendelsen. Ettersom dette ble gjennomført så fort hendelsen ble utløst, i motsetning til det som hadde vært tilfelle ved manuell håndtering av hendelsen, ble denne potensielle risikoen mitigert. Integrasjon mot brannmurens API for å opprette brannmurreglene krevde litt mer arbeid for å konfigurere og bruke riktig. Likevel var det enkelt å benytte API-et i de automatiske prosessene når konfigurasjonen var på plass. I ettertid kunne vi verifisere at regelen ble laget i brannmuren ved å undersøke listen over brannmurregler i administratorgrensesnittet, vist i figur 4.7. I tillegg bekreftet vi at den kompromitterte klienten ikke lenger kunne sende påloggingsforespørsler mot brannmuren, ved å simulere nye påloggingsforsøk som vist i figur 4.8.

[3] `https://www.whois.com/whois`



Figur 4.6: Case 2: Playbook utløst for event

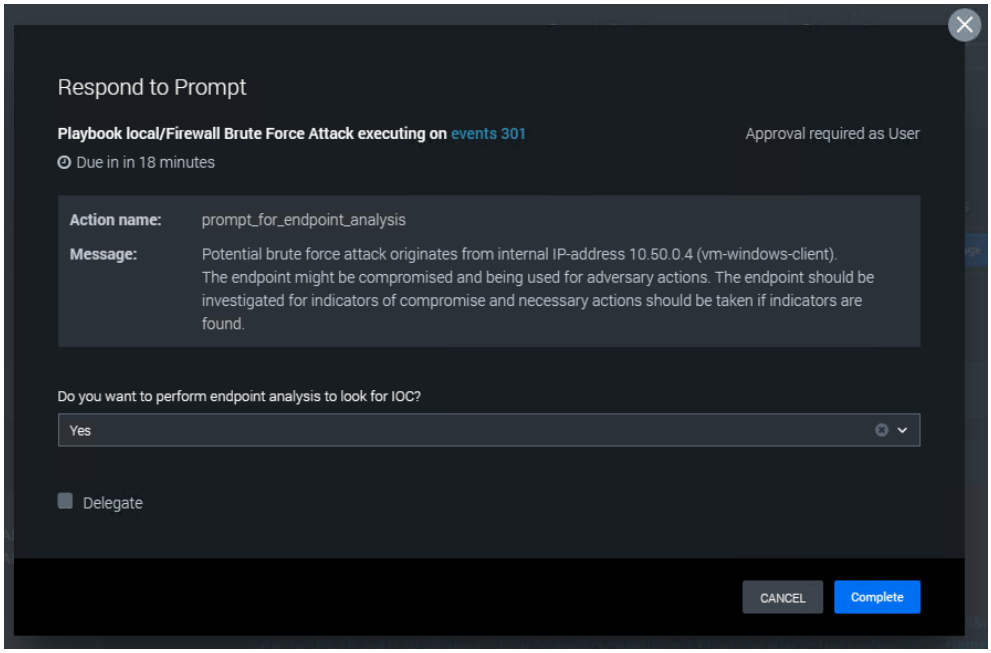
#		Name	Source	Destination	What	ID	Action
1		BlockIp10.50.0.4 in 0 B, out 45.96 KB	Any zone, 10.50.0.4	Any zone, Any host	Any service	#3	Drop
		VnetToVnetNetworkR... in 42.05 MB, out 1.85 GB					
		VnetToInternet in 95.57 MB, out 158.83 MB					
8		Drop all in 0 B, out 0 B	Any zone, Any host	Any zone, Any host	Any service	#0	Drop

Figur 4.7: Case 2: IP-adresse blokkert i brannmur

```
PS C:\Users\admbachelor> ssh admin@10.0.3.4
ssh: connect to host 10.0.3.4 port 22: Permission denied
PS C:\Users\admbachelor>
```

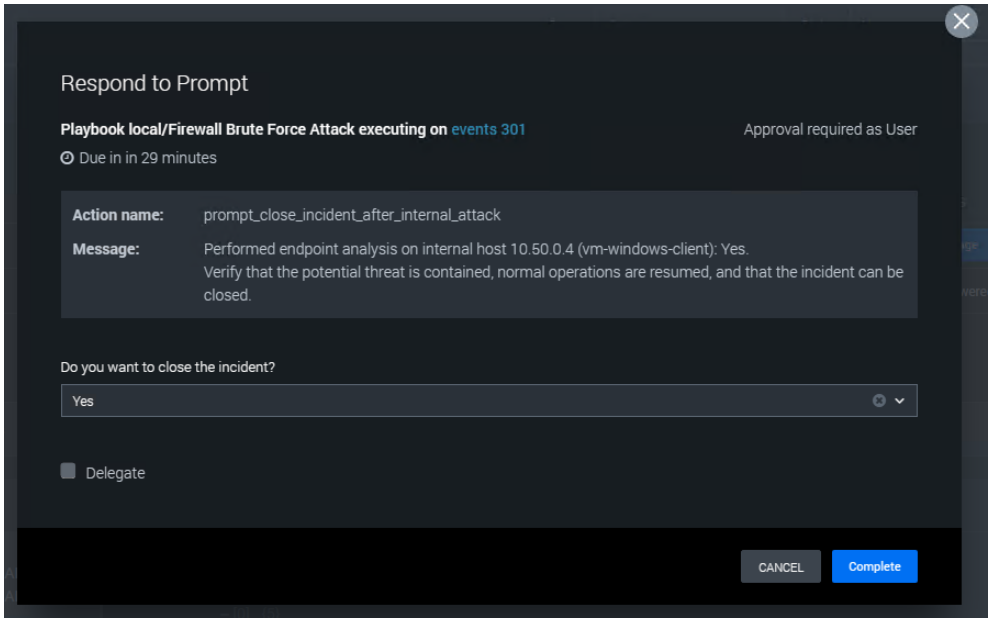
Figur 4.8: Case 2: Klient blokkert fra å fortsette *brute force*-angrep

Hittil har playbooken undersøkt IP-adressen, hentet maskinnavnet og blokkert adressen i brannmuren. Playbooken sendte på dette tidspunktet en forespørsel om menneskelig input, der en sikkerhetsanalytiker ble presentert den informasjonen som er samlet inn hittil, med valg om å starte analyse av endepunktet basert på informasjonen som blir presentert. Denne forespørselen er vist i figur 4.9. Playbooken ble på dette tidspunktet pauset i påvente av menneskelig vurdering. Samtidig er det viktigste mitigeringstiltaket i form av en blokkering av IP-adressen blitt gjennomført automatisk. Vi valgte å starte playbooken *Endpoint Analysis*[4] for å gjennomføre analyse av endepunktet. På grunn av oppgavens omfang designet vi ikke en playbook for fullstendig endepunktsanalyse, men vår oppfatning var at det ville vært mulig å designe en playbook som automatisk beriket saken med informasjon om endepunktet. Dermed hadde det vært mulig å utsette menneskelig involvering til etter denne analysen var gjennomført. Ved design av playbooken kunne vi også ha valgt at denne underordnede playbooken skal startes automatisk ved hendelser forårsaket av interne adresser. I denne casen valgte vi å vise hvordan en slik beslutning kan bli gjort av en sikkerhetsanalytiker. Tilslutt ble en ny forespørsel om menneskelig input sendt, der sikkerhetsanalytikeren ble spurt om å lukke hendelsen basert på det som hadde blitt gjort hittil i saken, vist i figur 4.10. Dette ble presentert på en oversiktlig måte slik at en sikkerhetsanalytiker kan vurdere handlingen uten behov for unødvendig mye egenanalyse. Vi valgte å lukke hendelsen, som markerte at hendelsen var løst og casen avsluttet.



Figur 4.9: Case 2: Forespørsel om endepunktsanalyse

[4]Som nevnt i seksjon 3.2.8 er dette en tom playbook for demonstrasjonsformål, og det eneste resultatet fra denne utførelsen er derfor en kommentar om at analyse har blitt gjennomført



Figur 4.10: Case 2: Forespørsel om å lukke hendelsen

Oppsummering av case 2

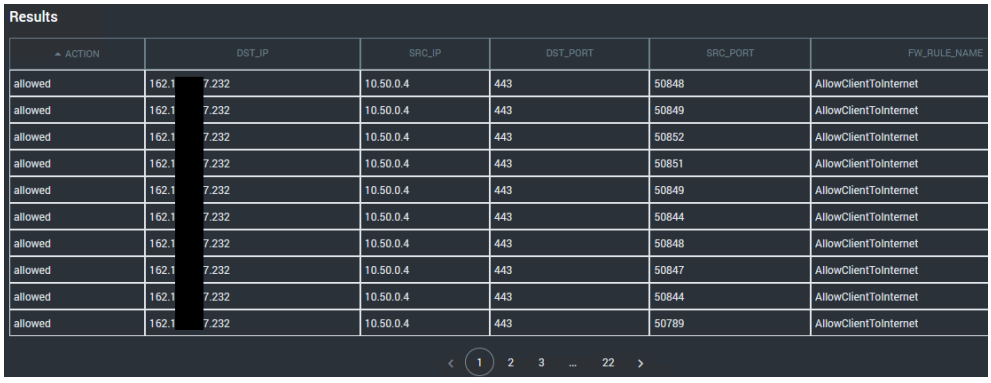
Brute force-angrepet ble håndtert og avverget automatisk av implementerte tiltak og de automatiske prosessene som ble tatt i bruk. Dermed oppnådde verktøyene og teknologiene det ønskede resultatet, uten behov for menneskelig involvering før etter hendelsen var mitigert. Playbooken klarte å automatisk blokkere trafikk fra IP-adressen som forårsaket *brute force*-angrepet. Dette regnet vi som det viktigste resultatet fra gjennomføringen av casen. Etter en slik hendelse vil det normalt være nødvendig med ytterligere tiltak og undersøkelser utover det som ble dekket i denne playbooken. Likevel opplever vi at effektiviteten ved å automatisk stanse *brute force*-angrepet, samt gjennomføre en initiell analyse av hendelsen, gir stor nytteverdi i forhold til manuell håndtering av hendelsen. Hendelsen ble mitigert mye tidligere enn den ville ha blitt ved manuell analyse og håndtering, og i et virkelig scenario kunne dette ha spart en organisasjon for store konsekvenser dersom *brute force*-angrepet var velykket i å oppdage passordet til grensesnittet. Derfor opplever vi at teknologien var et svært godt tillegg og støtteverktøy til manuell hendelseshåndtering.

4.2.3 Case 3

Gjennomføring av case 3 ble startet ved å simulere mottak av trusseletterretning fra Kripos, som beskrevet i seksjon 3.2.9. Vi benyttet manuelle funksjoner i SOAR-verktøyet for å opprette en ny sak, laste opp trusseletterretningen, og utløse playbooken *Hunt IOC*. Hittil i case 1 og 2 har playbookene blitt utløst automatisk når eventer ankommer Splunk SOAR. Etter gjennomføring av denne casen erfarte vi at

de manuelle funksjonalitetene i SOAR-verktøyet er brukervennlige og bidrar til effektiv håndteringsflyt. Etter at vi utløste playbooken manuelt, ble den gjennomført med resterende automatiske prosesser i likhet med tidligere caser. Playbooken itererte over artefaktene (indikatorene), og startet utførelse av riktig playbook (*Hunt IOCs IP*, *Hunt IOCs hash* eller *Hunt IOCs domain*). Hvilken playbook som ble utløst i dette steget avhengte av typen indikator (IP-adresse, hash eller domene) for hver artefakt.

I denne oppgaven har vi kun utarbeidet playbooken *Hunt IOCs IP*. Den første handlingen som ble utført av denne playbooken var et automatisk søk i Splunk Enterprise etter den potensielt ondsinnede IP-adressen. Dette søket returnerte med IP-adressen til den eller de interne enhetene som hadde vært i kontakt med den ondsinnede IP-adressen. Søket returnerte med IP-adressen `10.50.0.4`, som utløste ytterligere analyser. Videre ble et nytt søk gjennomført for å berike saken med en liste over all trafikk mellom den interne og den ondsinnede IP-adressen. Trafikkloggene, vist i figur 4.11, ga nyttig informasjon for videre håndtering av hendelsen. Resultatet viste blant annet hvilke portnummer og brannmurregel som ble benyttet under kommunikasjonen. Dette la grunnlaget for videre analyse. I tillegg ble et søk gjennomført for å hente ut maskinnavnet og brukerkontoene tilhørende den interne maskinen. Dette søket returnerte klientnavnet `vm-windows-client`, men ingen mistenkelige brukerkontoer. Dersom en ondsinnet brukerkonto hadde vært opprettet for å sende trafikk til og fra den ondsinnede IP-adressen, kunne dette ha blitt oppdaget gjennom søket.



Results

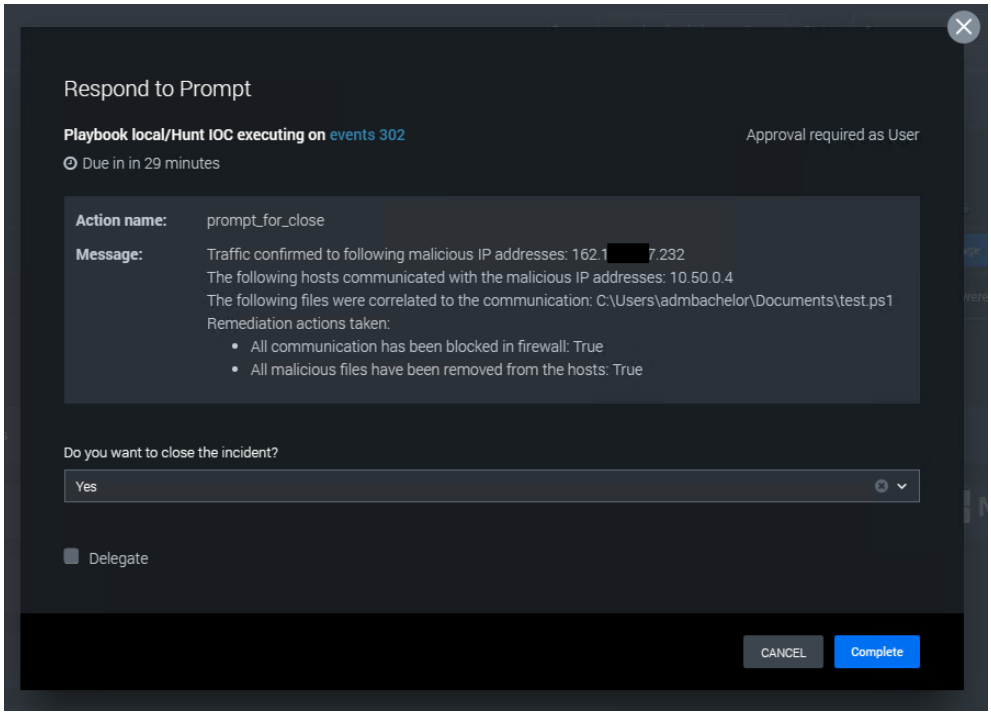
ACTION	DST_IP	SRC_IP	DST_PORT	SRC_PORT	FW_RULE_NAME
allowed	162.1[redacted]7.232	10.50.0.4	443	50848	AllowClientToInternet
allowed	162.1[redacted]7.232	10.50.0.4	443	50849	AllowClientToInternet
allowed	162.1[redacted]7.232	10.50.0.4	443	50852	AllowClientToInternet
allowed	162.1[redacted]7.232	10.50.0.4	443	50851	AllowClientToInternet
allowed	162.1[redacted]7.232	10.50.0.4	443	50849	AllowClientToInternet
allowed	162.1[redacted]7.232	10.50.0.4	443	50844	AllowClientToInternet
allowed	162.1[redacted]7.232	10.50.0.4	443	50848	AllowClientToInternet
allowed	162.1[redacted]7.232	10.50.0.4	443	50847	AllowClientToInternet
allowed	162.1[redacted]7.232	10.50.0.4	443	50844	AllowClientToInternet
allowed	162.1[redacted]7.232	10.50.0.4	443	50789	AllowClientToInternet

‹ 1 2 3 ... 22 ›

Figur 4.11: Case 3: Søk etter all trafikk mellom intern og ondsinnet IP-adresse

Basert på informasjonen avdekket gjennom trafikkloggene, kunne playbooken fortsette automatisk analyse av hendelsen. Dette ble gjort gjennom et nytt søk som korrelerte portnummer brukt under kommunikasjonen, til prosesser og tilhørende filer eller programmer på den interne maskinen hvor trafikken stammer fra. Søket returnerte en fil som viser kilden bak den ondsinnede trafikken. Denne automatiseringen krevde et mer avansert søk, men det var mulig kun ved hjelp av logger som var samlet inn kontinuerlig fra klienten. Deretter hentet playbooken en kopi av scriptet til Splunk SOAR som bevis, før playbooken slettet scriptet fra den interne maskinen for å stanse trafikken til den ondsinnede IP-adressen.

I tillegg til å utføre handlinger mot den interne enheten, sendte playbooken forespørsler til brannmurens API for å stanse all trafikk til eller fra den ondsinnede IP-adressen. Dette oppnådde vi ved å gjenbruke funksjonaliteten designet for case 2. Dette tiltaket førte til ytterligere mitigering av hendelsen, dersom det ikke var nok å slette scriptet fra den interne maskinen. Playbooken gjennomførte også oppslag etter den mistenksomme adressen i tjenestene *VirusTotal IP Reputation*[5], *MaxMind Geolocate*[6] og *WHOIS IP*, for å henholdsvis sjekke IP-adressens omdømme, geografisk opprinnelse og eierskap. Dette var nyttig informasjon som kunne blitt benyttet under videre analyser og som beslutningsgrunnlag for menneskelige vurderinger av saken. Det var først på dette tidspunktet at menneskelig vurdering var nødvendig for å avgjøre om hendelsen skal lukkes, vist i figur 4.12. Informasjonen oppsummerte hendelsen og redegjorde for de viktigste punktene som har blitt avdekket og analysert gjennom de automatiske prosessene.



Figur 4.12: Case 3: Forespørsel om å lukke hendelsen

Oppsummering av case 3

Case 3 har vist hvordan SOAR-verktøy kan benyttes manuelt for å håndtere hendelser. I tillegg har casen demonstrert en prosess bestående av flere sammenkoblede playbooks der både deteksjon, analyse og håndtering gjennomføres automatisk. Casen oppnådde det tiltenkte resultatet, ved at trusseletterretning raskt kan

[5] `https://developers.virustotal.com/reference/ip-info`
[6] `https://dev.maxmind.com/geoip`

benyttes under threat hunting for å avdekke og håndtere hendelser automatisk. I et reellt scenario ville ytterligere handlinger og permanente tiltak blitt iverksatt for å utrydde ondsinnet aktivitet og gjenopprette normaltilstand. Det tok tid å utvikle playbooks som vist i denne casen, men deretter kan de gjenbrukes på generelt grunnlag under threat hunting av trusseletterretning eller andre indikatorer. På grunn av mengden manuelle prosesser som er innebygget i playbooken, har vi fått et inntrykk av at slike playbooks vil effektivisere hendelseshåndtering over tid. Vi opplevde derfor denne casen som et godt eksempel på nytteverdien av automatisert hendelseshåndtering som et tillegg til manuell hendelseshåndtering.

Kapittel 5

Diskusjon

I denne rapporten har vi undersøkt hvordan hendelseshåndteringsprosessen kan effektiviseres med automatisering. I den følgende diskusjonsdelen vil vi besvare forskningsspørsmålene vi definerte i seksjon 1.2.1. Disse vil bli drøftet i lys av presentert teori og resultat fra dybdeintervju og mulighetsstudien. Forskningsspørsmålene vi ønsket å svare på gjennom denne oppgaven var:

Forskningsspørsmål 1: Hvilke faser og steg i hendelseshåndtering er egnet for effektivisering gjennom automatisering?

Forskningsspørsmål 2: Hvordan kan en organisasjon gå frem for å velge riktig verktøy og metoder for effektivisering av hendelseshåndtering, som er i tråd med organisasjonens modningsgrad, behov og hensyn?

5.1 Effektiv hendelseshåndtering

For å være i stand til å diskutere forskningsspørsmålene, er det nødvendig å ha en god forståelse for hva effektiv hendelseshåndtering innebærer. Hendelseshåndtering skal gjenopprette normal drift så raskt som mulig, og redusere negative konsekvenser. Med bakgrunn i dette skal vi drøfte hva vi legger i begrepet *effektiv hendelseshåndtering*.

Først og fremst har nøkkeltallsindikatorer ofte blitt brukt som et mål for effektiviteten av hendelseshåndtering. Hendelseshåndteringsprosessen er strukturert som en syklisk prosess. Dette legger til rette for kontinuerlig forbedring av prosessen, hvor nøkkeltallsindikatorer kan være til stor hjelp for å måle forbedring. Samtidig kan effektivitet defineres som «noe som virker på tilsiktet måte» [132]. Derfor vil vi definere effektiv hendelseshåndtering som en prosess som i størst mulig grad gjenoppretter normaltilstand uten unødvendige forsinkelser eller utilsiktede konsekvenser. Dette innebærer at effektivisering av hendelseshåndtering ikke skal gå på bekostning av forventet resultat.

Det er naturlig at forventningene til hendelseshåndtering er relatert til hvor *raskt* en hendelse blir håndtert. Som det ble påpekt gjennom dybdeintervjuene

kan den potensielle kostnaden relatert til en sikkerhetshendelse bli ekstremt høy avhengig av hvor lang tid hendelseshåndteringen tar. Likevel er det viktig å huske at hendelseshåndtering er tett knyttet opp mot organisasjonens verdikjedesystem. Prosessen skal bidra til å sikre organisasjonens verdier og tjenester, og dersom dette ikke tas i betraktning kan organisasjonen risikere å ta snarveier som kan bryte med krav stadfestet i ulike tjenestenivåavtaler. Det er derfor viktig å finne en balanse mellom effektivitet og kvalitet i hendelseshåndtering.

Et sentralt tema som kom frem gjennom dybdeintervjuene, var at teknologi skal underbygge prosess, ikke motsatt. Samtidig ble det nevnt at det er mennesker som skal benytte seg av teknologiene og gjennomføre prosessene. Ressurser bør derfor ikke brukes til å finne arbeidsoppgaver til teknologien, men i stedet bør teknologien være en ressurs som letter de nåværende arbeidsoppgavene. Ved å fokusere på tiltak som underbygger samtlige av de tre dimensjonene i modellen People, Process & Technology (PPT), mener vi at dette vil gi økt effektivitet tilbake til hendelseshåndteringen. Vi foreslår derfor at begrepet *effektiv hendelseshåndtering* bør sees i lys av PPT-modellen, istedenfor å kun benytte nøkkeltallsindikatorer.

5.2 Forskningsspørsmål 1

Hvilke faser og steg i hendelseshåndtering er egnet for effektivisering gjennom automatisering? Som redegjort i seksjon 2.3 består hendelseshåndteringsprosessen av fire primære faser med tilhørende understeg. Gjennom denne seksjonen vil vi drøfte egnethet til effektivisering av hver fase. Dette gjøres med bakgrunn i presentert teori og resultater fra dybdeintervju og mulighetsstudie.

5.2.1 Forberedelsesfase

Forberedelsesfasen av hendelseshåndtering er i stor grad preget av planlegging og manuelle prosesser, men er likevel en kandidat for effektivisering gjennom automatisering. Som nevnt i 2.3.1 består forberedelsesfasen av både planlegging for å håndtere hendelser, og tiltak som skal forhindre at hendelser oppstår.

Planlegging for å håndtere hendelser

Gjennom oppgavens dybdeintervju kom det frem at rutiner og prosedyrer er en nødvendig forutsetning for å effektivisere hendelseshåndtering. Utarbeidelse av prosedyrer er en prosess som ikke involverer automatisering i stor grad, og vil heller ikke være en prioritet for innføring av automatisering. Likevel vil implementering av SOAR-verktøy for å utvikle og sentralisere prosedyrer være et hjelpemiddel for automatisering av hendelseshåndteringsprosessen. Fra mulighetsstudie opplevde vi at SOAR gjorde prosedyrer for hendelseshåndtering lettere tilgjengelig ved at disse er sentralisert i én plattform. I tillegg var det lettere å gjøre endringer, samt lettere å følge etablerte prosedyrer ved håndteringen av en hendelse, sammenlignet med å slå opp prosedyrer manuelt i dokumentasjonsverktøy. Vi

opplevde utvikling av prosedyrer i SOAR-verktøy som tidvis krevende og erfarte at kompetansebygging innenfor verktøyet kan kreve mye ressurser. Dersom SOAR-verktøy skal legge grunnlaget for effektivisering av resten av prosessen, må denne kompetansebyggingen gjelde både de som skal håndtere hendelser og de som er med på å utarbeide prosedyrer og rutiner.

Tiltak for å forhindre hendelser

Kartlegging av verdier, sårbarheter og trusler er et potensielt bruksområde for automatisering. Automatisk kartlegging av verdier kan være utfordrende fordi enhver organisasjon vil være unik i form av hvilke tjenester de leverer og hvordan de har satt opp sine systemer. Enhver innføring av automatisering i dette arbeidet kan dermed kreve en del manuell validering, som fører til at manuell kartlegging av verdier kunne vært like, eller mer, effektivt.

MITREs *Common Vulnerabilities and Exposures (CVE)*[1] er et eksempel på en offentlig ressurs som kartlegger sårbarheter så fort de blir offentlig kjent. Informasjon fra slike tjenester kan integreres med verktøy for sårbarhetsskanning som automatisk kartlegger sårbarheter blant en organisasjons tjenester og systemer, slik at sikkerhetshull kan tettes før de utnyttes. Dette vil effektivisere hendelseshåndtering i form av at færre alvorlige hendelser oppstår, i tillegg til å spare tid ved automatisering av en tidkrevende arbeidsoppgave. Det vil være en nødvendig forutsetning at en organisasjon først har kartlagt sine verdier og tjenester, blant annet offentlig eksponerte tjenester og systemer. Selv om tiltaket kan gi effektiviseringsgevinst, er det likevel viktig å huske på at det kan være ressurskrevende i form av tid, penger og kompetansekrav.

I likhet med sårbarheter finnes det offentlige kilder som kartlegger aktuelle trusler og trusselaktører. Dette er viktig informasjon som kan benyttes for å forebygge hendelser. I likhet med kartlegging av sårbarheter, er innsamling av trusseletterretning tidkrevende arbeid som kan automatiseres. Slik informasjon kan samles inn ved bruk av script som samler inn informasjon fra offentlige kilder, eller gjennom kommersielle tjenester. Et viktig tema under oppgavens intervju var forskjellen mellom kjente trusler og ukjente trusler. Ettersom ukjente trusler er unike for enhver organisasjon, vil kartlegging av slike trusler kreve større ressurser i form av manuelle undersøkelser, enn kartlegging av kjente trusler fra offentlige kilder. På bakgrunn av dette vil vi argumentere for at ressurser bør investeres i å automatisere sårbarhetsskanning og innsamling av offentlig trusseletterretning, slik at ressurser heller kan prioriteres mot å avdekke ukjente trusler som kan påføre organisasjonen stor skade.

5.2.2 Deteksjons- og analysefase

Deteksjons- og analysefasen av hendelseshåndtering er sannsynligvis den fasen som i størst grad benytter automatisering i dag. De fleste deteksjonssystemer ba-

[1] `https://www.cve.org/`

serer seg på automatiserte regler og deteksjonslogikk for å detektere og varsle om unormal eller ondsinnet aktivitet. Likevel er det flere aspekter ved denne fasen som fortsatt preges av manuelle prosesser og tidkrevende manuelt arbeid. Gjennom denne seksjonen vil vi argumentere for tiltak som kan implementeres for effektivisering av deteksjons- og analysedelen av hendelseshåndtering.

Prioriteringer for deteksjon

Gjennom dybdeintervjuene har vi avdekket at effektiviteten til deteksjon avhenger av at hendelser detekteres der det er mest hensiktsmessig. Organisasjoners angrepsflate består av flere mulige angrepsvektorer. Derfor bør deteksjon av sikkerhetshendelser prioriteres mot de mest sannsynlige angrepsvektorene. Oppsett av deteksjonslogikk mot sjeldne eller usannsynlige angrepsvektorer vil gi lite nytteverdi i forhold til kostnaden ved implementering. Derfor foreslår vi på bakgrunn av dette at ressurser bør fokuseres mot de mest sannsynlige angrepsvektorene.

Deteksjonslogikk i en organisasjon vil avhenge av hvilke verdier de besitter og tjenester de leverer. Utarbeidelse av logikk krever mye tid, kunnskap og ikke minst erfaring, som ble poengtert under dybdeintervjuene. På bakgrunn av dette vil det være hensiktsmessig å lene seg på tjenester som eksempelvis XDR-løsninger fra kommersielle leverandører. Slike løsninger inkluderer deteksjonslogikk som er designet og testet basert på markedsbehov, trender og historiske datagrunnlag. Derimot kan de være kostbare å anskaffe, implementere og vedlikeholde. Nytteverdien avhenger derfor av organisasjonens evne til å utnytte løsningene på en god måte. Samtidig er det kostbart å designe, teste og implementere egenutviklet deteksjonslogikk. Disse løsningene kan ha stor dekningsgrad for kjente trusler, men ikke nødvendigvis for ukjente trusler. På bakgrunn av dette vil vi argumentere for at størst effektivitet oppnås ved å kombinere implementering av kommersielle løsninger for deteksjon av kjente trusler, med internutviklet deteksjonslogikk og ressursbruk rettet mot ukjente trusler.

Et annet tema som ble tatt opp under dybdeintervjuene var viktigheten og behovet for håndtering av store datamengder og tuning av alarmer. Manglende tuning av deteksjonslogikk, kan resultere i mer støy enn nytteverdi. Dette er et problem som potensielt kan løses med automatisk analyse og håndtering av alarmer. Dermed kan mest mulig data synliggjøres uten at det går utover effektiviteten til analysearbeidet. Baksiden ved slik tuning, som også ble diskutert under dybdeintervjuene, er risikoen for å automatisere bort *for mye*. På bakgrunn av dette kan effektivitetsgevinst oppnås ved å finne en god balanse mellom mengden data som synliggjøres, hvilke alarmer som håndteres automatisk og hvilke hendelser som håndteres av mennesker.

Automatisering og maskinlæring for deteksjon

Under dybdeintervjuene diskuterte respondentene potensialet til maskinlæring gjennom anomalitetsbasert deteksjon. En svakhet ved signaturdeteksjon, er deteksjon av ukjente trusler og sårbarheter. Respondentene trakk frem at anoma-

litetsbasert deteksjon bidrar til å se mønstre og avdekke hendelser blant store datagrunnlag, i tillegg til å redusere arbeidsmengden. Det gjør det også mulig å avdekke hendelser forårsaket av nulldagssårbarheter eller indikatorer uten definerte signaturer, hvor det ellers ikke foreligger etablert deteksjonslogikk. I tillegg trakk de frem bruk av maskinlæring for dynamisk generering av deteksjonslogikk. For eksempel ved å automatisk generere deteksjonslogikk basert på informasjon om en ny sårbarhet slik som Log4j. Avhengig av organisasjonens størrelse, angrepsflate, eksisterende kompetanse og ressurser, vil vi argumentere for at det vil gi effektivitetsgevinst å implementere teknologi som benytter maskinlæring i en eller annen form.

Trusselbasert perspektiv på deteksjon og analyse

Med utgangspunkt i Cyber Kill Chain-modellen og MITRE ATT&CK-rammeverket, bør deteksjon og analyse sees med et trusselbasert perspektiv. Med et slikt perspektiv kan funn fra kartlegging og analyse av hendelsesforløp omgjøres til indikatorer, som deretter automatisk integreres i deteksjonssystemer som nye deteksjonsregler. Dermed kan fremtidige hendelser oppdages på et tidligere stadium og gi effektivitetsgevinst ved å forhindre hendelser raskere. I tillegg bidrar Cyber Kill Chain og MITRE ATT&CK til lettere kartlegging av trusselaktører og deres fremgangsmåter. Imidlertid kan en overgang til et trusselbasert perspektiv på deteksjon og analyse være ressurskrevende. Basert på våre undersøkelser om behov og utfordringer med hendelseshåndtering, foreslår vi at organisasjoner i større grad enn tidligere sentrerer prosesser rundt dette perspektivet. Vi vil argumentere for at perspektivet gir et godt utgangspunkt for å implementere automatiserte prosesser og integrere disse bedre sammen for å håndtere hendelser gjennom alle faser av hendelsers livsløp.

Automatisering av analysearbeid

Under analyse av hendelser finnes det godt etablerte verktøy for automatisering. I mulighetsstudien benyttet vi blant annet funksjonalitet i Splunk Enterprise for å automatisk kategorisere alarmers sensitivitet og kritikalitet. I løpet av dybdeintervjuene ble oppgaver som å kategorisere gjengående hendelser, trukket frem som repetitivt arbeid som fjerner verdifull tid. Samtidig må hendelsers autentisiteten, det vil si hvorvidt de er reelle eller falske positiver, avklares. Det kan være enklere for en sikkerhetsanalytiker å vurdere hvorvidt en alarm er reell eller ikke, enn for en datamaskin. Dersom falske positiver ikke avdekkes og forhindres ved dette steget, kan det føre til lavere effektivitet i form av merarbeid for å håndtere falske alarmer og ukritiske hendelser. Implementering av automatiseringsløsninger i denne fasen må derfor vurderes opp i mot organisasjonens totale alarmmengde, ressursbelastning ved manuell håndtering, samt krav til kostnad, vedlikehold og kompetanse ved implementering av automatisering.

Automatisering er også godt egnet under den detaljerte analysen av hendelser. Utifra de erfaringer vi har gjort oss gjennom oppgavens mulighetsstudie, trekker

vi spesielt frem potensialet ved bruk av verktøy slik som SOAR. Vi benyttet blant annet SOAR for å integrere tredjepartstjenester for å gjennomføre analyse av indikatorer, mistenksomme filer og detektert skadevare. Videre kan automatiserte prosesser stå for bevisinnsamling, berikelse av saker, gjennomføring av threat hunting, samt dokumentering av funn og gjennomførte analyser. Respondentene trakk også frem et eksempel fra Log4j-saken, der SOAR-verktøy kunne blitt benyttet for automatisk analyse av IP-adresser som detekteres i sammenheng med den ondsinnede Log4j-nyttelasten. Manuell utførelse av slike analyseoppgaver krever en del ressurser i form av tid og personell, i tillegg til at de ofte innholder repetitive og programmerbare steg som en datamaskin er godt egnet for å gjennomføre. Derfor kan analysefasen av hendelseshåndtering automatiseres i større eller mindre grad ved hjelp av for eksempel SOAR-verktøy.

Kartlegging av hendelser

En viktig del av håndteringen under en hendelse er kartlegging av hendelsens omfang. Dette inkluderer kartlegging av berørte systemer, tjenester, brukere, kunder og tredjeparter, samt avhengigheter og tilkoblinger internt og eksternt. Under oppgavens dybdeintervju trakk flere av respondentene frem viktigheten av å konkretisere hva som har skjedd, samt hvem og hva som er berørt når en hendelse først oppstår. På bakgrunn av dette mener vi derfor at effektiv kartlegging avhenger av systemer for konfigurasjonsstyring og tjenestekatalog (CMDB). En tjenestekatalog vil være desto viktigere for automatiserte prosesser. Uten en oppdatert og tilgjengelig tjenestekatalog vil det være utfordrende for en datamaskin å vurdere omfanget av en hendelse og vurdere hvilke tjenester og systemer som er sammenkoblet. Slik kunnskap er nødvendig for å kunne danne situasjonsforståelse og for å gjøre sikre beslutninger ved innføring av mitigeringstiltak, som vi skal diskutere i neste seksjon. Derfor vil vi argumentere for at god konfigurasjonsstyring og tjenestekatalog er en helt sentral forutsetning for effektiv hendelseshåndtering støttet av automatiserte prosesser.

5.2.3 Mitigering, utryddelse og gjenopprettelsesfase

Under håndteringen av en hendelse vil det være en kontinuerlig syklus mellom fasen for deteksjon & analyse og fasen for mitigering, utryddelse & gjenopprettelse. Som vist i oppgavens mulighetsstudie er det vanlig å avdekke ytterligere informasjon og indikatorer under håndteringen av en hendelse som bør analyseres videre eller brukes for ny deteksjonslogikk. Informasjon og bevis som avdekkes og samles inn under håndteringen av en hendelse vil enkelt kunne kobles opp mot de samme automatiseringene som brukes i deteksjons- og analysefasen. Videre i denne seksjonen vil vi diskutere forslag for å effektivisere andre deler av håndteringsfasen.

Valg av håndteringsstrategi

En håndteringsstrategi og veldefinerte planer for hvordan en gitt hendelse skal mitigeres, er viktig for å iverksette tiltak så raskt som mulig. Målet er å velge en strategi som gjenoppretter normaltilstand uten unødvendige forsinkelser eller utilsiktede konsekvenser. Under dybdeintervjuene trakk respondentene frem viktigheten av å komme i gang med målrettede tiltak ved håndtering av alvorlige hendelser, før store konsekvenser eller skade påføres organisasjonen.

Automatisering kan være et hjelpemiddel under valg og gjennomføring av en håndteringsstrategi. Å bruke SOAR-verktøy som en sentralisert plattform vil forenkle strategivalget under håndtering av hendelser, eller gjennomføre valget automatisk basert på forhåndsdefinert logikk. Imidlertid vil valg av strategi for en gitt hendelse avhenge av forskjellige kriterier, og i mange tilfeller vil det fortsatt kreve menneskelig interaksjon for å velge riktig basert på hendelsens natur og kontekst. Mitigeringstiltak kan i verste fall ha store konsekvenser for tilgjengeligheten og integriteten til tjenester og systemer, i tillegg til at det kan føre til brudd på tjenestenivåavtaler og skade omdømmet til en organisasjon. Derfor er en god oversikt over organisasjonens tjenester og systemer, samt hvordan alle disse henger sammen, essensielt for å håndtere hendelser uten å risikere å skape nye hendelser.

Et gjennomgående tema gjennom både mulighetsstudien og dybdeintervju er tillit til at de automatiserte systemene velger riktig strategi. Hendelseshåndtering som fagfelt er tett knyttet opp mot risikostyring, og det vil alltid foreligge et behov for risikovurdering for å vurdere konsekvensene av mitigeringstiltak opp i mot konsekvensene ved å la være. Basert på våre erfaringer fra mulighetsstudien burde automatiseringer være enkle, modulære og forståelige. Med SOAR vil det innebære å lage flere mindre playbooks som hver for seg automatiserer deler av de fullstendige prosedyrene. Playbooks kan også utformes med inkludering av menneskelige vurderinger i form av enkle kontrollspørsmål og veivalg, som vist i flere av casene i mulighetsstudien. Samspillet mellom disse tiltakene reduserer risikoen for utilsiktede konsekvenser, ved å kombinere menneskelig vurdering med automatiserte prosedyrer. Derfor anbefaler vi å fokusere på delautomatisering, spesielt ved større og mer komplekse hendelser.

Utryddelse og gjenoppretting

Noen hendelser krever utryddelse og gjenopprettelse i form av for eksempel fjerning av skadevare, sletting av brukere, fjerning av tilganger eller tetting av sikkerhetshull. De fleste av disse tiltakene gjennomføres med en stor grad av automatisering, men iverksettes vanligvis av mennesker. For eksempel vil fjerning av skadevare ofte gjennomføres ved en automatisert reinstallering og gjenopprettelse av systemer, og sikkerhetsoppdateringer kan rulles ut og installeres automatisk på tvers av systemer i en organisasjon.

Automatisering kan benyttes for å verifisere at normaltilstand er gjenopprettet, for eksempel gjennom analyse av logger eller annen data som kan bekrefte

fravær av ondsinnet aktivitet. Slik automatisering kan føre til effektivitetsgevinst ved at dette kan gjennomgå store datagrunnlag mye raskere enn mennesker. Under oppgavens dybdeintervju ble det derimot trukket frem utfordringer ved å verifisere normaltilstand automatisk. Det er ofte usikkerhetsmomenter rundt *hva* ondsinnet aktivitet er og hvordan denne aktiviteten kan identifiseres. Det kan samtidig tenkes at maskinlæring i større grad vil kunne benyttes for slikt arbeid i fremtiden, som diskutert i intervjuundersøkelsen. Maskinlæringsmodeller brukes allerede i for eksempel anomalitetsbasert deteksjon, så det kan derfor tenkes at slike modeller kan benyttes for dataanalyse etter håndteringen av en hendelse for å bekrefte hvorvidt normaltilstand er gjenopprettet.

Utfordringer ved design av automatiske prosedyrer

Gjennom denne oppgavens mulighetsstudie har vi erfart at det kan være utfordrende å implementere vanskelige vurderingsgrunnlag i automatiserte playbooks og prosedyrer. Vi erfarerer at teknologien kan være både kraftig og effektiv, men at forutsetningen om korrekt datagrunnlag er vanskelig å komme foruten. Det er mulig å implementere avanserte analyser som en del av playbooks i SOAR som kan avdekke falske positiver og dobbeltsjekke at nødvendig informasjon og kontekst er på plass før tiltak implementeres. Derimot fører dette raskt til et kompleksitetsnivået i playbooks som ikke er skalerbart, og hvor mengden tid som må settes inn for å utvikle playbooks og teste for alle mulige scenarioer kan gå kraftig utover effektivitetsgevinsten. Vi vil derfor argumentere for at håndteringsfasen kan effektiviseres, samtidig som risiko og kostnader reduseres, ved å prioritere automatisering av enkle hendelser og delprosesser. Dermed kan menneskelige ressurser prioriteres mot hendelser som krever vurderinger og tilsyn.

Varsling av hendelser

Under oppgavens dybdeintervju ble det trukket frem utfordringer med å korrekt og effektivt varsle berørte parter under håndteringen av en hendelse. Blant annet kan det være utfordrende å varsle interessenter om en pågående hendelse før de selv opplever eller blir berørt av hendelsen. I tillegg kan det oppstå en interessekonflikt der ressurser benyttes for varsling, og ikke for håndtering. Automatiserte systemer for varsling kan sørge for raskere varsling til berørte parter, samtidig som at ressurser frigjøres til håndtering. Likevel er det nødvendig å påpeke at en del forutsetninger må være på plass for å kunne automatisere varsling, og det er samtidig potensielle fallgruver som bør unngås. Fallgruvene inkluderer varsling basert på falske positiver, varsling på feil tjeneste og varsling til feil mottakere. For det første må det eksistere tilstrekkelig teknologi for å avdekke *hva* som er berørt. Dette avhenger av deteksjons- og analysesystemer som kan kartlegge hendelser, avdekke avhengigheter og tilkoblinger, og verifisere feil. Det neste som må være på plass er et system som støtter automatisk varsling, for eksempel ved bruk av standardiserte maler og forhåndsdefinerte lister over hvem som skal varsles avhengig av tjeneste og system.

5.2.4 Etterarbeidsfase

Under ettarbeidsfasen for hendelseshåndtering er det potensiale for automatisering av innsamling og rapportering av data. Innsamlede data fra hendelser har stor nyttverdi, både for bruk under hendelsesrevisjoner og for generering av rapporter og statistikk. Gjennom oppgavens dybdeintervju har vi derimot avdekket utfordringer med å generere tilstrekkelige rapporter og statistikk som illustrerer effektiviteten av hendelseshåndtering, samt behovet for ressurser og teknologier. Rapporter danner beslutningsgrunnlag som er viktig for å demonstrere hvor det er størst effektivitetsgevinst å hente. Data fra deteksjonssystemer og verktøy for hendelseshåndtering er også helt nødvendig for å fastsette metrikker for effektiviteten til hendelseshåndtering. Slike metrikker er som nevnt vanskelig å fastsette, samtidig som det ofte ikke gir det komplette bilde av den faktiske effektiviteten. I tillegg er innsamling, analyse og strukturering av data tidkrevende arbeid, spesielt når det må gjennomføres fra flere systemer.

Automatisering fungerer veldig godt når det er snakk om repetitive arbeidsoppgaver som ikke krever utfordrende eller skiftende vurderingsgrunnlag. Innsamling av data fra systemer vil i de fleste tilfeller være repetitivt og bestå av de samme stegene gang etter gang. Gjennom relativt enkle automatiseringer tror vi de fleste organisasjoner kan gjennomføre dette på en god måte. Samtidig kan det medføre utfordringer dersom: data er samlet inn fra forskjellige systemer; data er på forskjellig format; eller data er samlet inn fra systemer som behandler ulik rådata med varierende kontekst. I tillegg kan datagrunnlaget inneholde falske positiver eller data som på annet vis er ufullstendig eller feil. Det er sannsynlig at mange organisasjoner på dette tidspunktet avhenger av menneskelig involvering for å strukturere, analysere og generere handlingsdyktige beslutningsgrunnlag. Dette er en arbeidsoppgave vi tror maskinlæring har stort potensial, spesielt ved å analysere store datasett for å trekke ut sammenhenger, trender og andre interessepunkt.

I tillegg har vi erfart gjennom mulighetsstudien at loggadministrasjons- og SOAR-verktøy kan forenkle prosessen med automatisk innsamling av data. Slike verktøy inkluderer innebygget funksjonalitet for filtrering, strukturering og generering av rapporter. Dette kan fjerne behovet for å utvikle egne automatiseringsverktøy eller scripts som må vedlikeholdes. Dersom store deler av datagrunnlaget for rapportering uansett er på plass gjennom slike verktøy, kan implementeringen av verktøyene dermed medføre ytterligere effektivisering i form av forenklet og automatisert rapportering. Samtidig må kostnader relatert til innføring, vedlikehold og drift, samt risikoen for å bli avhengig av spesifikke kommersielle verktøy, tas med i den helhetlige vurderingen.

5.3 Forskningsspørsmål 2

Hvordan kan en organisasjon gå frem for å velge riktig verktøy og metoder for effektivisering av hendelseshåndtering, som er i tråd med organisa-

sjonens modningsgrad, behov og hensyn? Under anskaffelse av verktøy, valg av metoder og implementering av løsninger kan organisasjoner følge en rekke forskjellige fremgangsmåter. Hvilken fremgangsmåte som er passende for en gitt organisasjon avhenger av flere faktorer som må avdekkes for å oppnå størst ressursgevinst. Gjennom denne seksjonen skal vi presentere de viktigste av disse vurderingspunktene, i lys av teori, metode og tidligere drøfting.

5.3.1 Modningsgrad, behov og hensyn

For å velge riktig fremgangsmåte for effektivisering må en organisasjon først definere egen modningsgrad, behov og hensyn. Når vi i denne rapporten diskuterer modningsgraden til en organisasjon, mener vi hvor utviklet den er i henhold til PPT-modellen. Det handler altså om hvor veldefinert og utviklet organisasjonens prosesser, roller og teknologiplattform er. Valg av fremgangsmåte må sees i lys av modningsgrad, slik at tiltak med størst effektivitetsgevinst kan implementeres. For eksempel vil organisasjonens eksisterende teknologier påvirke hvilke teknologiinvesteringer som gir størst gevinst. Behov og hensyn vil i mange tilfeller være knyttet opp i mot modningsgrad, ved at organisasjoner med lik modningsgrad vil dele mange behov og hensyn. Samtidig vil behov og hensyn også variere fra organisasjon til organisasjon avhengig av faktorer som sektor, organisasjonsstruktur, mål og risikoappetitt. For eksempel vil en organisasjon som behandler sensitive personopplysninger måtte ta større hensyn til personvern, og ha behov for sterkere beskyttelse av data. Derfor må valg av fremgangsmåte innenfor de forskjellige områdene som diskuteres i påfølgende seksjoner, alltid gjøres med grunnlag i organisasjoners modningsgrad, behov og hensyn.

5.3.2 Kartlegge mål, organisering og prosesser

Etter definering av sin modningsgrad, hensyn og behov, må en organisasjon kartlegge sine mål. Ved å ha en forståelse for hva organisasjonen ønsker å oppnå gjennom sin hendelseshåndteringsprosess, er det lettere å velge riktig fremgangsmåte. Dersom målet for eksempel er å bygge opp et fullstendig SOC, vil det legge føringer for hvilke investeringer som gjøres og hvordan organisasjonen struktureres. På den andre siden, hvis organisasjonen kun har et mål om å sikre et mindre sett med skybaserte tjenester, vil hendelseshåndteringsarbeidet struktureres annerledes. I tillegg er det nødvendig å vurdere hvilke tjenester organisasjonen leverer og hvilke verdier organisasjonen innehar, for å velge fremgangsmåte ut ifra disse. Dette vil legge føringer for hvilke effektiviseringstjenester som har nytteverdi. Behovsgrunnlaget bør ta utgangspunkt i at hendelseshåndtering er en balanse mellom proaktivt og reaktivt arbeid, slik at organisasjonen velger de verktøyene som passer deres behov.

Enhver organisasjon må definere klare roller og rutiner for hendelseshåndtering. Når en hendelse oppstår, er det nødvendig å vite hvem som har ansvar for hvilke deler av hendelseshåndteringsprosessen. Under dybdeintervjuene ble det

trukket frem viktigheten av definerte roller, bevisstgjøring av de forskjellige rollenes behov, og god samhandling mellom de forskjellige rollene. Det kom også frem at mennesker og prosess må komme før teknologi. Hvis ikke kompetansenivået til menneskene i organisasjonen er høy nok, og prosessene ikke er definert, vil det være lite nytteverdi i innføring av teknologi. Dette handler om viktigheten av å forankre fremgangsmåten i PPT-modellen. Teknologi og verktøy har som formål å forenkle arbeidsoppgaver, men prosessene må eksistere først slik at man vet hva som skal effektiviseres og hva som skal løses med verktøyene. Vi har erfart gjennom mulighetsstudien at det å benytte seg av verktøy slik som SOAR, krever en helhetlig forståelse for prosessene som skal automatiseres.

5.3.3 Grunnleggende forutsetninger for automatisering

Gjennom oppgavens intervjuundersøkelse, var det et gjennomgående tema at en rekke forutsetninger bør være på plass før automatiseringsløsninger kan implementeres. En av de forutsetningene som ble diskutert var opparbeidelsen av en felles forståelse og kultur for bruk av automatisering. Det innebærer blant annet å underbygge effektiviseringsbehov slik at gevinsten ved automatisering er tydelig for organisasjonen. Investeringer i automatiseringsverktøy kan være kostbart, og behovet som blir dekket må derfor presenteres på en slik måte at ledelsen i en organisasjon kan vurdere kost-nytte-verdien av investeringene. En måte å oppnå dette på, er ved tydeliggjøring av hvordan implementeringen av automatiseringsverktøy vil gjennomføres, og hvilke problemer dette vil løse over tid. Dette handler i stor grad om redegjørelse av behovsgrunnlag. Automatiseringer må gi avkastning i form av enten redusert risiko, sparte kostnader eller økt verdiskapning. Det vil være vanskelig å få gjennomslag for investering i automatiseringsteknologi som ikke kan kobles opp mot et tydelig behovsgrunnlag. I tillegg vil det være vanskelig å få nytteverdi ut av implementert automatiseringsteknologi som ikke dekker de grunnleggende behovene organisasjonen har.

Et annet viktig moment som ble nevnt var at grunnleggende tjenester bør være på plass før automatisering kan implementeres. Det ble eksempelvis påpekt at et godt etablert saksbehandlingssystem, konfigurasjonsstyring og tjenestekatalog (CMDB) og ikke minst et tilstrekkelig datagrunnlag, må være på plass for at investeringen av ny teknologi skal gi verdi. Uten synliggjøring og tilgjengeliggjøring av data, og integreringer mot tjenester og systemer, er det lite nytteverdi i verktøy for automatisk hendelseshåndtering. Vi foreslår derfor at organisasjoner først sørger for å ha på plass disse grunnleggende forutsetningene, før de går til investering av mer avanserte verktøy for hendelseshåndtering og automatiseringsløsninger.

5.3.4 Valg av teknologi for effektivisering

Når de grunnleggende forutsetningene er på plass, vil det være mulig å oppnå effektiviseringsgevinst ved investering i mer avanserte verktøy og teknologier. Investeringer i teknologi må samtidig vurderes opp i mot en rekke faktorer. Enhver

investering må underbygges med en grundig kost-nytte analyse for å sikre at investeringen gir avkastning over tid. Det kan innebære at investeringer som gir effektivitetsgevinst, likevel ikke bør implementeres for gitte organisasjoner, ettersom verdien de tilføyer ikke er større enn kostnaden ved anskaffelse, implementering og vedlikehold. For eksempel vil det for mange organisasjoner ikke være lønnsomt å investere i verktøy slik som SIEM og SOAR, fordi antall hendelser de håndterer, og kompetansen de innehar, vil føre til at investeringene ikke gir avkastning.

Et annet viktig moment for valg av teknologi og verktøy, vil være hvordan de underbygger organisasjonens krav til konfidensialitet, integritet og tilgjengelighet (KIT). Valg av verktøy og metoder for hendelseshåndtering burde derfor tilfredsstille organisasjonens krav til alle de tre dimensjonene. Dette vil avhenge av de behov og hensyn som er tidligere definert, ettersom forskjellige organisasjoner vil ha forskjellige krav til informasjonssikkerhet blant sine tjenester. For eksempel kan organisasjoner som har strenge krav til konfidensialiteten ved sine endepunkter, ha et ønske om å investere i avanserte verktøy for endepunktsdeteksjon og -respons, slik som XDR-verktøy. Derfor er det viktig å ha forståelse for organisasjonens verdier og tjenester, og velge verktøy for hendelseshåndtering som underbygger disse.

Når valg av verktøy og teknologier skal gjennomføres, vil det være en rekke alternativer å velge mellom. Ny teknologi bør kunne integreres med eksisterende prosesser og systemer som allerede brukes for å sikre en sømløs implementasjon. Dette innebærer at organisasjoners eksisterende verktøy og teknologier legger føring for hvilken fremgangsmåte de vil velge videre. For eksempel kan det tenkes at en organisasjon som allerede har implementert verktøy slik som SIEM og XDR, sannsynligvis har automatisert flere forskjellige analyse og responshandlinger i separerte prosesser og verktøy. For slike organisasjoner kan det naturlige neste steget være å samle slike automatiseringer innunder en sentralisert plattform slik som et SOAR-verktøy. Dermed kan de sammenkoble disse automatiseringene, i tillegg til å insentivisere til bruk av dem. På den andre siden vil det for organisasjoner som mangler synliggjøring av sine systemer og tjenester, være naturlig å først prioritere investeringer mot deteksjonssystemer.

5.3.5 Oppsummering av vurderingsgrunnlag

Før en organisasjon velger fremgangsmåte for effektivisering mener vi at man burde vurdere følgende:

(1) Hva er vår modningsgrad, hvilke behov har vi og hvilke hensyn må vi ta?

(2) Har vi kartlagt våre mål med hendelseshåndtering, tildelt roller og sørget for at organisasjonen kan samhandle effektivt?

(3) Har vi etablerte prosesser og kompetanse i organisasjonen, som kan støttes ved innføring av teknologi?

(4) Har vi kartlagt våre behovsgrunnlag og lagt en plan for hvordan verktøy for automatisering skal implementeres?

(5) Har vi de grunnleggende forutsetningene og tjenestene på plass for å få nytteverdi ut av automatiseringsløsninger?

(6) Har vi identifisert hvilke verktøy og teknologier som vil gi oss størst kostnytte-verdi, dekke våre krav til konfidensialitet, integritet og tilgjengelighet, og som kan integreres med våre eksisterende systemer?

5.4 Begrensninger ved studien

I denne diskusjonsdelen vil vi drøfte begrensninger ved studien. Vi vil drøfte begrensninger rundt de to forskjellige vitenskapelige metodene vi har benyttet – dybdeintervjuene og mulighetsstudie.

5.4.1 Begrensninger ved dybdeintervju

Validitet

Flere av respondentene har tatt del i utformingen av oppgavens problemstilling. Dette kan medføre at respondentene hadde dannet seg konkrete meninger og synspunkter rundt tematikken før intervjuene ble gjennomført. Den interne validiteten til intervjuundersøkelsen (til hvilken grad resultatene er gyldig for det utvalget og det fenomenet som er undersøkt) anser vi som god, fordi ansatte fra Norsk helsenett har deltatt i utarbeidelsen av oppgavens problemstilling. To av medlemmene i bachelorgruppen var under gjennomføring av denne oppgaven ansatt i Norsk helsenett. Vi anser ikke resultatet som påvirket av dette faktum. Derimot opplevde vi at vi har dratt fordel av bekjentskapet ved at respondentene har en relasjon til bachelorgruppen fra før.

Den eksterne validiteten til intervjuundersøkelsen (til hvilken grad resultatene kan overføres til andre utvalg og situasjoner) er til en viss grad redusert av det faktum at kun ansatte fra Norsk helsenett er intervjuet. Vi mener likevel at resultatene fra intervjuene kan ha en viss holdbarhet for andre IT-organisasjoner, siden flere av resultatene er generelle og basert på teorietiske drøftinger og kunnskap fra respondentene.

Begrensninger

Under intervjuundersøkelsen ble det valgt ut fem respondenter, fra team med forskjellige ansvarsområder innenfor hendelseshåndtering. Vi vurderer derfor spredningen blant respondentene som god. Likevel kunne resultatetene dratt fordel av å ha en enda bredere spredning blant respondentene, ved å inkludere flere team internt i Norsk helsenett, eller inkludere respondenter fra eksterne organisasjoner.

Sett i ettertid ville det også vært fordelaktig å gjennomført en kvantitativ spørreundersøkelse. Denne undersøkelsen kunne vi sendt ut til forskjellige miljøer i fagsektoren, og vi kunne med dette samlet inn en større mengde statistiske data. Dette ville kunne støttet vår argumentasjon i større grad. Valget om å kun gjennomføre dybdeintervju, ble gjort på bakgrunn av oppgavens tidsbegrensning og omfang.

En annen begrensende faktor ved dybdeintervjuene var vår manglende erfaring med kvalitative undersøkelser som vitenskapelig metode. Kvaliteten av undersøkelsen kunne vært av en høyere standard, dersom vi hadde sittet med denne erfaringen i forkant av studien. Dette kan ha ført til at tiden det tok å utarbeide den kvalitative undersøkelsen økte. Likevel anser vi ikke den manglende erfaringen som en utslagsgivende faktor for det endelige resultatet.

5.4.2 Begrensninger ved mulighetsstudien

En potensiell begrensning ved mulighetsstudien, var det faktum at brukstestene ble gjennomført etter at oppgavens dybdeintervju var fullført. Det kan ha ført til at resultatene fra den kvalitative undersøkelsen påvirket gjennomføringen av brukstestene. Likevel tror vi ikke dette hadde innvirkning på resultatet fra mulighetsstudien, ettersom brukstestene ble utarbeidet i paralell med den kvalitative undersøkelsen. Dermed hadde ikke dybdeintervjuenes resultat innvirkning på design og utvikling av brukstestene.

Vi valgte å designe og utarbeide alle playbooks selv. Hovedulempen med dette er at vi risikerte å unnlate viktige elementer som burde inkluderes i playbookene. For å unngå denne problemstillingen valgte vi på forhånd å studere og ta inspirasjon fra ferdige og relaterte playbooks for å sikre at sentrale punkter ikke ble utelatt. I tillegg drøftet vi tanker og ideer til enkelte playbooks med fagkyndige ansatte hos Norsk helsenett, for å forsikre oss om at vi var på rett spor. En annen utfordring med å designe playbooks selv var at dette var nytt territorium for samtlige medlemmer av bachelorgruppen. Dette medførte at vi måtte sette av tid til å bygge opp kompetanse om verktøyene som ble benyttet. Dessuten er SOAR-teknologi nokså nytt, så det fantes begrenset med kunnskap og dokumentasjon om bruken av SOAR-verktøy. Mye tid gikk derfor med på research og dokumentasjonslesning. Våre playbooks ble i tillegg noe begrenset i form av integrasjoner. Mange av integrasjonene krevde tilgang til kommersielle tredjepartsløsninger, eller lå bak betalingsmurer. På denne måten ble brukstestene begrenset i form av at vi ikke fikk utforsket et bredere spekter av integrasjoner og muligheter som tilbys av SOAR-verktøy.

5.5 Fremtidig arbeid

Oppgaven har utforsket spesifikke verktøy for automatisering, og dekker bare en liten del av mulighetene de innehar. Denne diskusjonsdelen tar for seg relevante tema som rapporten ikke dekker på bakgrunn av avgrensing og omfang, men som

kan danne grunnlag for videre arbeid.

Øke ekstern validitet Fremtidig arbeid kunne fokusert på å utvide den kvalitative undersøkelsen, slik at man med større sannsynlighet kan si at rapporten er relevant for et bredere publikum. Ved å gjennomføre kvantitative og kvalitative undersøkelser på tvers av organisasjoner og sektorer, vil det i større grad være mulig å avdekke problemstillinger, erfaringer og utfordringer ved hendelseshåndtering som enhver organisasjon har.

Teste bruken av SOAR i et større testmijlø Videre forskning kunne utforsket bruken av SOAR i større og mer komplekse testmiljø. Ved å knytte opp SOAR med verktøy slik som SIEM, XDR og CMDB, vil vi kunne vise til større og mer komplekse funksjoner med et SOAR. Automatisering av slike tjenester vil kunne vise til en betydelig større effektiviseringsgevinst enn det vi har demonstrert i vårt mulighetsstudie.

Undersøke hvordan kunstig intelligens i automatiseringsverktøy kan gi økt effektiviseringsgevinst Denne oppgaven har i mindre grad gått inn på kunstig intelligens og maskinlæring. Videre arbeid kan i større grad utforske bruken av kunstig intelligens for hendelseshåndtering.

Teste flere leverandørers versjoner av SIEM og SOAR-løsninger Videre arbeid kunne testet forskjellige leverandørers SIEM- og SOAR-løsninger. Dette vil gi oss bredere forståelse av hvilke muligheter og alternativer som finnes, samtidig som større forskjeller som kan være avgjørende i valg av leverandør kan avdekkes.

Kapittel 6

Konklusjon

I denne oppgaven har vi utforsket en problemstilling i form av to forskningsspørsmål: Hvilke faser og steg i hendelseshåndtering er egnet for effektivisering gjennom automatisering, og hvordan kan en organisasjon gå frem for å velge riktig verktøy og metoder for effektivisering av hendelseshåndtering, som er i tråd med organisasjonens modningsgrad, behov og hensyn?

Vi har gjennom teori beskrevet hendelseshåndtering i detalj, samt moderne sikkerhetsavdelingers ansvarsområder og utfordringer. Vi har i tillegg redegjort for metoder og teknologier for hendelseshåndtering, og hvordan automatisering og maskinlæring kan benyttes som en del av disse. Med utgangspunkt i teorien har vi gjennomført fem dybdeintervju av ansatte i Norsk helsenett for å undersøke og kartlegge et effektiviseringsbehov av hendelseshåndtering. Videre har vi lagt frem tre brukstester som demonstrerer muligheter for effektivisering av hendelseshåndtering.

En av de viktigste slutningene vi kan ta ut fra diskusjonen, er at alle faser av hendelseshåndtering til en viss grad er egnet for effektivisering med automatisering. Selv om dybdeintervjuene og mulighetsstudien var av et begrenset omfang, mener vi det er grunnlag for å si at resultatene svarer på generelle problemstillinger innenfor hendelseshåndtering og samsvarer med relevant teori.

Våre resultateter viste at effektivisering av forberedelsesfasen er mulig ved å automatisere innsamling av informasjon fra offentlige kilder for å gjennomføre sårbarhetsskanning og threat hunting. Videre har vi erfart at automatisering er godt etablert som en del av deteksjonsteknologi, og har stort potensiale for analysearbeid. På bakgrunn av aktuell teori og våre undersøkelser tror vi maskinlæring kan bidra til ytterligere effektivisering av både deteksjon og analyse. Gjennom mulighetsstudien har vi gjennomført automatisert håndtering av hendelser i fasen for mitigering, utryddelse og gjenopprettelse, og vist at effektivisering av denne fasen er mulig. Likevel kan ytterligere modenhet blant organisasjoner og teknologi være nødvendig for å automatisere denne fasen uten økt risiko for utilsiktede konsekvenser. Til slutt har vi erfart at etterarbeidsfasen kan effektiviseres i form av automatisert innsamling av data for generering av rapporter og kontinuerlig forbedring av hendelseshåndteringprosessesen.

I tråd med forskningsspørsmål 2 har vi kommet frem til forslag for hvordan en organisasjon kan gå frem for å velge riktig verktøy og metoder for effektivisering av hendelseshåndtering. En av de viktigste lærdommene fra våre resultater er at teknologi skal underbygge prosess, ikke motsatt. Det betyr at utvikling av prosesser og menneskelig kompetanse må komme før anskaffelse av avanserte teknologier og metoder, i henhold til modellen for mennesker, prosess og teknologi (PPT). Ved innføring av teknologi foreslår vi at dette forankres i organisasjonens behovsgrunnlag og krav til konfidensialitet, integritet og tilgjengelighet (KIT). For størst nytteverdi bør også teknologien samsvare med organisasjoners eksisterende systemer og tjenester.

Vi trekker spesielt frem potensialet til verktøy for sikkerhetsorkestrering, automatisering og respons (SOAR) for effektivisering av hendelseshåndtering. Likevel tyder våre resultater på at organisasjoner som ønsker å videreutvikle kapabiliteter for hendelseshåndtering, får større kost-nytte-verdi ved å først innføre sentrale forutsetninger før anskaffelse av SOAR eller lignende verktøy. Forutsetningene inkluderer innføring av teknologi for sentralisert innsamling av data, forbedring av endepunktsdeteksjon og -respons og systemer for konfigurasjonsstyring og tjenestekatalog.

Avslutningsvis kan vi konstatere at automatisering er sentralt for å effektivisere hendelseshåndteringsprosessen. Under dagens komplekse trusselbilde og geopolitiske situasjon der informasjonsteknologi er en grunnpillar bak grunnleggende nasjonale funksjoner, blir det stadig viktigere å videreutvikle og effektivisere hendelseshåndtering. Vår studie har vist at automatisering vil være en viktig bidragsyter for å oppnå denne effektiviseringen. Samtidig vil det være interessant å følge med på hvordan nyvinninger innenfor kunstig intelligens kan ytterligere påvirke denne utviklingen.

Bibliografi

[1] NIST, *attack surface*, ordliste, 2023. adresse: https://csrc.nist.gov/glossary/term/attack_surface.

[2] Fortinet, *What Is An Attack Surface?* nettside, 2023. adresse: https://www.fortinet.com/resources/cyberglossary/attack-surface.

[3] B. Lenaerts-Bergmans, *Attack Vectors: What They Are and How They Are Exploited*, nettside, 2023. adresse: https://www.crowdstrike.com/cybersecurity-101/threat-intelligence/attack-vector/.

[4] MITRE, *11 Strategies of a World-Class Cybersecurity Operations Center*. The MITRE Corporation, 2022. adresse: https://www.mitre.org/sites/default/files/2022-04/11-strategies-of-a-world-class-cybersecurity-operations-center.pdf.

[5] NIST, *artifact*, ordliste, 2023. adresse: https://csrc.nist.gov/glossary/term/artifact.

[6] E. M. Hutchins, M. J. Cloppert og R. M. Amin, «Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains,» Lockheed Martin Corporation, rapport, 2010. adresse: https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf.

[7] T. H. Nätt, *bakdør*, oppslagsverk, 2019. adresse: https://snl.no/bakd%5C%C3%5C%B8r.

[8] T. H. Nätt, *botnett*, oppslagsverk, 2019. adresse: https://snl.no/botnett.

[9] T. H. Nätt og E. Rossen, *brannmur*, oppslagsverk, 2019. adresse: https://snl.no/brannmur.

[10] Microsoft, *What is Cloud Native?* nettside, 2022. adresse: https://learn.microsoft.com/en-us/dotnet/architecture/cloud-native/definition.

[11] IBM, *What are containers?* nettside, 2022. adresse: https://www.ibm.com/topics/containers.

[12] T. H. Nätt, *datavirus*, oppslagsverk, 2022. adresse: https://snl.no/datavirus.

[13] T. H. Nätt og E. Rossen, *domene (IT)*, oppslagsverk, 2022. adresse: https://snl.no/domene_-_IT.

[14] A. Sharif, *What is an event log?* nettside, 2022. adresse: https://www.crowdstrike.com/cybersecurity-101/observability/event-log/.

[15] P. Cichonski, T. Millar, T. Grance og K. Scarfone, «Computer Security Incident Handling Guide,» NIST, rapport, 2012. adresse: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf.

[16] R. Kissel, *Glossary of Key Information Security Terms*, ordliste, 2013. adresse: https://nvlpubs.nist.gov/nistpubs/ir/2013/NIST.IR.7298r2.pdf.

[17] H. Øverby, *ip-adresse*, oppslagsverk, 2021. adresse: https://snl.no/IP-adresse.

[18] «Risiko 2023,» NSM, rapport, 2023. adresse: http://nsm.no/Risiko2023.

[19] «Definitive guide to ransomware,» IBM, rapport, 2022. adresse: https://www.ibm.com/downloads/cas/EV6NAQR4.

[20] A. Tidemann, *kunstig intelligens*, oppslagsverk, 2023. adresse: https://snl.no/kunstig_intelligens.

[21] A. Tidemann og A. C. Elster, *maskinlæring*, oppslagsverk, 2022. adresse: https://snl.no/maskinl%5C%C3%5C%A6ring.

[22] B. E. Strom, J. A. Battaglia, M. S. Kemmerer, W. Kupersanin, D. P. Miller, C. Wampler, S. M. Whitley og R. D. Wolf, «Finding Cyber Threats with ATT&CK™-Based Analytics,» The MITRE Corporation, rapport, 2017. adresse: https://www.mitre.org/sites/default/files/2021-11/16-3713-finding-cyber-threats-with-attack-based-analytics.pdf.

[23] NIST, *Operational Technology Security*, nettside, 2022. adresse: https://csrc.nist.gov/projects/operational-technology-security.

[24] T. H. Nätt, *orm (IT)*, oppslagsverk, 2020. adresse: https://snl.no/orm_-_IT.

[25] T. H. Nätt, *nettfiske*, oppslagsverk, 2021. adresse: https://snl.no/nettfiske.

[26] J. Saltzer og M. Schroeder, «The protection of information in computer systems,» tekn. rapp., 1975. DOI: 10.1109/PROC.1975.9939.

[27] T. H. Nätt, *regulære uttrykk*, oppslagsverk, 2023. adresse: https://snl.no/regul%5C%C3%5C%A6re_uttrykk.

[28] T. H. Nätt, *script*, oppslagsverk, 2023. adresse: https://snl.no/skript_-_IT.

[29] T. H. Nätt, *skadevare*, oppslagsverk, 2022. adresse: https://snl.no/skadevare.

[30] T. H. Nätt, *sosial manipulering*, oppslagsverk, 2020. adresse: https://snl.no/sosial_manipulering_-_datasikkerhet.

[31] T. H. Nätt, M. Bartnes og N. Kjerstad, *spoofing*, oppslagsverk, 2023. adresse: https://snl.no/spoofing.

[32] S. Ellingsen, S.-M. H. Bjerke, D. English og J. Iden, *ITIL-ordliste og forkortelser på norsk*, ordliste, 2011. adresse: https://www.uio.no/studier/emner/matnat/ifi/INF3510/v14/docs/is-glossary-dnb-2013.pdf.

[33] T. H. Nätt, *Trojaner (IT)*, oppslagsverk, 2019. adresse: https://snl.no/trojaner_-_IT.

[34] «Risiko 2022,» NSM, rapport, 2022. adresse: http://nsm.no/Risiko2022.

[35] «The modern cybersecurity landscape: Scaling for threats in motion,» Cisco, rapport, 2020. adresse: https://umbrella.cisco.com/info/technical-paper-modern-security-landscape-scaling-threats-motion.

[36] «ENISA Threat Landscape 2022,» ENISA, rapport, 2022. DOI: 10.2824/764318. adresse: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2022.

[37] D. Sutton, *Information risk management* (A practitioner's guide), Second. BCS Learning og Development Ltd., 2021, s. 219, ISBN: 978-1-78017-5744.

[38] R. Vaarandi, M. Kont og M. Pihelgas, «Event log analysis with the LogCluster tool,» 2016. adresse: https://ieeexplore.ieee.org/abstract/document/7795458.

[39] ISO, *ISO/IEC 27035-1:2023 Styring av informasjonssikkerhetshendelser*, standard, 2023. adresse: https://standard.no/no/Nettbutikk/produktkatalogen/Produktpresentasjon/?ProductID=1500620.

[40] I. A. Tøndel, M. B. Line og M. G. Jaatun, «Information security incident management: Current practice as reported in the literature,» rapport, 2014. adresse: https://www.sciencedirect.com/science/article/pii/S0167404814000819.

[41] *ITIL Foundation*. Stationary Office Books, 2019, ISBN: 9780113316076.

[42] B. Simon, *Everything You Need to Know about the People, Process, Technology Framework*, nettside, 2021. adresse: https://www.smartsheet.com/content/people-process-technology.

[43] P. Karlson, *Is The 60-Year-Old 'People Process Technology' Framework Still Useful?* nettside, 2022. adresse: https://www.forbes.com/sites/forbestechcouncil/2022/12/29/is-the-60-year-old-people-process-technology-framework-still-useful/.

[44] ISO, *ISO/IEC 27000:2020*, standard, 2020. adresse: https://www.standard.no/no/Nettbutikk/produktkatalogen/Produktpresentasjon/?ProductID=1128720.

[45] ISO, *ISO/IEC 27005:2022*, standard, 2022. adresse: https://standard.no/no/Nettbutikk/produktkatalogen/Produktpresentasjon/?ProductID=1446030.

[46] «X-Force Threat Intelligence Index 2022,» IBM, rapport, 2022. adresse: https://www.ibm.com/downloads/cas/ADLMYLAZ.

[47] «X-Force Threat Intelligence Index 2023,» IBM, rapport, 2023. adresse: https://www.ibm.com/downloads/cas/DB4GL8YM.

[48] «Good Practice Guide for Incident Management,» ENISA, rapport, 2010. adresse: https://www.enisa.europa.eu/publications/good-practice-guide-for-incident-management.

[49] H. Øverby, *tingenes internett*, oppslagsverk, 2021. adresse: https://snl.no/tingenes_internett.

[50] C. Osborne, *This is why the Mozi botnet will linger on*, nettside, 2021. adresse: https://www.zdnet.com/article/this-is-why-the-mozi-botnet-will-linger-on/.

[51] «2022 Year in Review,» Cisco Talos, rapport, 2022. adresse: https://blog.talosintelligence.com/talos-year-in-review-2022/.

[52] MITRE, *Exploit Public-Facing Application*, nettside, 2018. adresse: https://attack.mitre.org/versions/v13/techniques/T1190/.

[53] MITRE, *Valid Accounts*, nettside, 2017. adresse: https://attack.mitre.org/versions/v13/techniques/T1078/.

[54] NIST, *vulnerability*, ordliste, 2023. adresse: https://csrc.nist.gov/glossary/term/vulnerability.

[55] P. B. Andersen, *automatisering*, oppslagsverk, 2021. adresse: https://snl.no/automatisering.

[56] E. Rossen og A. Tidemann, *ekspertsystem*, oppslagsverk, 2021. adresse: https://snl.no/ekspertsystem.

[57] «Cost of a Data Breach Report 2022,» Ponemon Institute på oppdrag av IBM Corporation, rapport, 2022. adresse: https://www.ibm.com/reports/data-breach.

[58] «Voice of SecOps 2021,» Deep Instinct, rapport, 2021. adresse: https://www.deepinstinct.com/pdf/voice-of-secops-report-2nd-edition.

[59] MITRE, *MITRE ATT&CK v13*, nettside, 2023. adresse: https://attack.mitre.org/versions/v13/.

[60] M. Marrone og L. M. Kolbe, «Impact of IT Service Management Frameworks on the IT Organization,» *Business & Information Systems Engineering*, 2011, ISSN: 1867-0202. DOI: 10.1007/s12599-010-0141-5.

[61] K. Bratsbergsengen, *hashing*, oppslagsverk, 2021. adresse: https://snl.no/hashing.

[62] A. Tjora, *avvik*, oppslagsverk, 2021. adresse: https://snl.no/avvik.

[63] *EDR vs XDR*, nettside, 2023. adresse: https://www.paloaltonetworks.com/cyberpedia/edr-vs-xdr.

[64] L. Miller, *XDR For Dummies, Palo Alto Networks Special Edition*. John Wiley & Sons, Inc., 2022, ISBN: 978-1-119-85169-1. adresse: https://www.paloaltonetworks.com/content/dam/pan/en_US/assets/pdf/guides/xdr-for-dummies.pdf.

[65] Splunk, *What Is SOAR (Security Orchestration, Automation and Response)?* nettside, 2022. adresse: https://www.splunk.com/en_us/data-insider/what-is-soar.html.

[66] Splunk, *Splunk SOAR (Cloud) Service Description*, nettside, 2023. adresse: https://docs.splunk.com/Documentation/SOAR/current/ServiceDescription/SplunkSOARService.

[67] IBM, *What is threat hunting?* nettside, 2022. adresse: https://www.ibm.com/topics/threat-hunting.

[68] NIST, *threat Intelligence*, ordliste, 2023. adresse: https://csrc.nist.gov/glossary/term/threat_intelligence.

[69] M. Long, «The Impact of AI on Proactive Incident Management,» nettside, 2022. adresse: https://www.ibm.com/cloud/blog/announcements/the-impact-of-ai-on-proactive-incident-management.

[70] J. Delua, «Supervised vs. Unsupervised Learning: What's the Difference?» nettside, 2021. adresse: https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning.

[71] K. Marko, *How machine learning strengthens incident management*, nettside, 2021. adresse: https://www.techtarget.com/searchitoperations/tip/How-machine-learning-strengthens-incident-management.

[72] A. Ahmad, B. S. Maynard og S. Park, «Information security strategies: towards an organizational multi-strategy perspective,» rapport, 2012. adresse: https://link.springer.com/article/10.1007/s10845-012-0683-0.

[73] J. Muniz, *Security Operations Center: Building, Operating, and Maintaining your SOC*. Cisco Press, 2015, ISBN: 9780134052014.

[74] M. Vielberth, F. Böhm, I. Fichtinger og G. Pernul, *Security Operations Center: A Systematic Study and Open Challenges*, artikkel, 2020. adresse: https://ieeexplore.ieee.org/document/9296846.

[75] B. P. Hámornik og C. Krasznay, *A Team-Level Perspective of Human Factors in Cyber Security: Security Operations Centers*, artikkel, 2017. adresse: https://link.springer.com/chapter/10.1007/978-3-319-60585-2_21.

[76] D. Crémilleux, C. Bidan, F. Majorczyk og N. Prigent, *Enhancing Collaboration Between Security Analysts in Security Operations Centers*, artikkel, 2019. adresse: https://link.springer.com/chapter/10.1007/978-3-030-12143-3_12.

[77] «(ISC2) CYBERSECURITY WORKFORCE STUDY,» International Information System Security Certification Consortium, rapport, 2022. adresse: https://www.isc2.org//-/media/ISC2/Research/2022-WorkForce-Study/ISC2-Cybersecurity-Workforce-Study.ashx.

[78] A. Erola, I. Agrafiotis, J. Happa, M. Goldsmith, S. Creese og P. A. Legg, *RicherPicture: Semi-automated cyber defence using context-aware data analytics*, artikkel, 2017. adresse: https://ieeexplore.ieee.org/document/8073399.

[79] F. B. Kokulu, A. Soneji, T. Bao, Y. Shoshitaishvili, Z. Zhao, A. Doupé og G.-J. Ahn, *Matched and Mismatched SOCs: A Qualitative Study on Security Operations Center Issues*, artikkel, 2019. adresse: https://dl.acm.org/doi/abs/10.1145/3319535.3354239.

[80] «Nasjonalt digitalt risikobilde 2022,» NSM, rapport, 2022. adresse: http://nsm.no/NDIG2022.

[81] «Voice of SecOps 2022,» Deep Instinct, rapport, 2022. adresse: https://info.deepinstinct.com/voice-of-secops-v3-2022.

[82] A. Rognsaa, *Bacheloroppgaven*, 4. utg. Universitetsforlaget, 2015, ISBN: 978-82-15-02530-8.

[83] J. Purujoki, *SOAR Playbook Implementation - Incident Deduplication and Its Effects*, bacheloroppgave, 2020. adresse: https://www.theseus.fi/handle/10024/354155.

[84] A. Bråthen, *Correlating IDS alerts with system logs by means of a network-centric SIEM solution*, masteroppgave, 2011. adresse: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/143982.

[85] C. Islam, «Architecture-centric support for security orchestration and automation,» ph.d.-avh., 2020. adresse: https://hdl.handle.net/2440/129206.

[86] A. Sridharan og V. Kanchana, «SIEM integration with SOAR,» 2022. DOI: 10.1109/INCOFT55651.2022.10094537.

[87] R. Vast, S. Sawant, A. Thorbole og V. Badgujar, «Artificial Intelligence based Security Orchestration, Automation and Response System,» 2021. DOI: 10.1109/I2CT51068.2021.9418109.

[88] F. Richter, *Amazon, Microsoft & Google Dominate Cloud Market*, nettside, 2022. adresse: https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers/.

[89] MITRE, *Reconnaissance*, nettside, 2020. adresse: https://attack.mitre.org/versions/v13/tactics/TA0043/.

[90] MITRE, *Obtain Capabilities: Malware*, nettside, 2021. adresse: https://attack.mitre.org/versions/v13/techniques/T1588/001/.

[91] MITRE, *Acquire Infrastructure: Domains*, nettside, 2023. adresse: https://attack.mitre.org/versions/v13/techniques/T1583/001/.

[92] MITRE, *Stage Capabilities: Link Target*, nettside, 2023. adresse: https://attack.mitre.org/versions/v13/techniques/T1608/005/.

[93] MITRE, *Phishing*, nettside, 2023. adresse: https://attack.mitre.org/versions/v13/techniques/T1566/.

[94] MITRE, *User Execution*, nettside, 2022. adresse: https://attack.mitre.org/versions/v13/techniques/T1204/.

[95] MITRE, *Persistence*, nettside, 2019. adresse: https://attack.mitre.org/versions/v13/tactics/TA0003/.

[96] MITRE, *Create Account*, nettside, 2023. adresse: https://attack.mitre.org/versions/v13/techniques/T1136/.

[97] MITRE, *Scheduled Task/Job*, nettside, 2023. adresse: https://attack.mitre.org/versions/v13/techniques/T1053/.

[98] MITRE, *Privilege Escalation*, nettside, 2021. adresse: https://attack.mitre.org/versions/v13/tactics/TA0004/.

[99] MITRE, *Lateral Movement*, nettside, 2019. adresse: https://attack.mitre.org/versions/v13/tactics/TA0008/.

[100] MITRE, *Abuse Elevation Control Mechanism: Bypass User Account Control*, nettside, 2023. adresse: https://attack.mitre.org/versions/v13/techniques/T1548/002/.

[101] Microsoft, *How User Account Control Works*, nettside, 2023. adresse: https://learn.microsoft.com/en-us/windows/security/identity-protection/user-account-control/how-user-account-control-works.

[102] MITRE, *Access Token Manipulation*, nettside, 2023. adresse: https://attack.mitre.org/versions/v13/techniques/T1134/.

[103] MITRE, *Exfiltration Over C2 Channel*, nettside, 2023. adresse: https://attack.mitre.org/versions/v13/techniques/T1041/.

[104] MITRE, *Data Manipulation*, nettside, 2022. adresse: https://attack.mitre.org/versions/v13/techniques/T1565/.

[105] MITRE, *Data Encrypted for Impact*, nettside, 2022. adresse: https://attack.mitre.org/versions/v13/techniques/T1486/.

[106] MITRE, *Data Destruction*, nettside, 2021. adresse: https://attack.mitre.org/versions/v13/techniques/T1485/.

[107] «Exmatter Points to Potential Future of Data Extortion,» Cyderes, rapport. adresse: https://www.cyderes.com/blog/threat-advisory-exmatter-data-extortion/.

[108] MITRE, *Execution*, nettside, 2019. adresse: https://attack.mitre.org/versions/v13/tactics/TA0002/.

[109] MITRE, *User Execution: Malicious Link*, nettside, 2020. adresse: https://attack.mitre.org/versions/v13/techniques/T1204/001/.

[110] MITRE, *Command and Control*, nettside, 2019. adresse: https://attack.mitre.org/versions/v13/tactics/TA0011/.

[111] MITRE, *Defense Evasion*, nettside, 2019. adresse: https://attack.mitre.org/versions/v13/tactics/TA0005/.

[112] MITRE, *Modify Authentication Process: Multi-Factor Authentication*, nettside, 2023. adresse: https://attack.mitre.org/versions/v13/techniques/T1556/006/.

[113] MITRE, *Indicator Removal: Clear Windows Event Logs*, nettside, 2023. adresse: https://attack.mitre.org/versions/v13/techniques/T1070/001/.

[114] MITRE, *Network Service Discovery*, nettside, 2023. adresse: https://attack.mitre.org/versions/v13/techniques/T1046/.

[115] MITRE, *Brute Force*, nettside, 2023. adresse: https://attack.mitre.org/versions/v13/techniques/T1110/.

[116] MITRE, *Automated Collection*, nettside, 2022. adresse: https://attack.mitre.org/versions/v13/techniques/T1119/.

[117] MITRE, *Adversary-in-the-Middle*, nettside, 2023. adresse: https://attack.mitre.org/versions/v13/techniques/T1557/.

[118] S. Adair og T. Lancaster, *DriftingCloud: Zero-Day Sophos Firewall Exploitation and an Insidious Breach*, nettside, 2022. adresse: https://www.volexity.com/blog/2022/06/15/driftingcloud-zero-day-sophos-firewall-exploitation-and-an-insidious-breach/.

[119] MITRE, *Network Denial of Service*, nettside, 2022. adresse: https://attack.mitre.org/versions/v13/techniques/T1498/.

[120] MITRE, *Service Stop*, nettside, 2022. adresse: https://attack.mitre.org/versions/v13/techniques/T1489/.

[121] MITRE, *Credential Access*, nettside, 2019. adresse: https://attack.mitre.org/versions/v13/tactics/TA0006/.

[122] MITRE, *Remote Services*, nettside, 2023. adresse: https://attack.mitre.org/versions/v13/techniques/T1021/.

[123] MITRE, *Data from Local System*, nettside, 2023. adresse: https://attack.mitre.org/versions/v13/techniques/T1005/.

[124] MITRE, *Data from Network Shared Drive*, nettside, 2023. adresse: https://attack.mitre.org/versions/v13/techniques/T1039/.

[125] MITRE, *Active Scanning: Vulnerability Scanning*, nettside, 2023. adresse: https://attack.mitre.org/versions/v13/techniques/T1595/002/.

[126] MITRE, *Develop Capabilities: Exploits*, nettside, 2021. adresse: https://attack.mitre.org/versions/v13/techniques/T1587/004/.

[127] MITRE, *Input Capture*, nettside, 2023. adresse: https://attack.mitre.org/versions/v13/techniques/T1056/.

[128] MITRE, *Defacement*, nettside, 2022. adresse: https://attack.mitre.org/versions/v13/techniques/T1491/.

[129] MITRE, *Disk Wipe*, nettside, 2023. adresse: https://attack.mitre.org/versions/v13/techniques/T1561/.

[130] MITRE, *Web Service*, nettside, 2020. adresse: https://attack.mitre.org/versions/v13/techniques/T1102/.

[131] MITRE, *Exfiltration*, nettside, 2019. adresse: https://attack.mitre.org/versions/v13/tactics/TA0010/.

[132] C. Nilstun, *effektiv*, oppslagsverk, 2021. adresse: https://snl.no/effektiv.

[133] M. Warner, *5 Reasons Why We Love Sysmon*, nettside, 2022. adresse: https://www.blumira.com/sysmon-benefits/.

Vedlegg A

Intervjuguide

Intervjuguide – Dybdeintervju Bacheloroppgave

Formål

Formålet med denne undersøkelsen er å snakke med fagpersonell som jobber direkte eller indirekte med hendelseshåndtering hos Norsk helsenett for å få innsikt i hvordan prosessen gjennomføres i praksis, samt finne utfordringer og erfaringer som kan støtte oppunder problemstillingen og svare på våre forskningsspørsmål. Vi ønsker å få innsikt i deres daglige arbeidsoppgaver, og deres tanker og ønsker for fremtiden og hvordan prosessen kan effektiviseres. I tillegg vil vi sammenligne sentral teori rundt hendelseshåndtering opp mot kandidatenes erfaringer fra arbeidslivet.

Format

Dybdeintervjuene vil gjennomføres som individuelle intervju med hver enkelt kandidat.

Intervjuene gjennomføres som semistrukturerte intervju, der kandidatene stilles de samme spørsmålene som beskrevet i denne intervjuguiden, men får forskjellige oppfølgingsspørsmål.

Intervjuspørsmål

I	Hva er din arbeidsstilling?
II	Hvilken seksjon jobber du innenfor?
III	Hvor lenge har du jobbet i Norsk helsenett (NHN)?
IV	Hva er din seksjon sine arbeidsoppgaver for NHN?
V	Hva er dine arbeidsoppgaver i NHN og i din seksjon?

1	Hvorfor er hendelseshåndtering viktig, generelt og i NHN?
2	Hva er din seksjon sitt ansvarsområde under håndteringen av en hendelse?
3	Kan du forklare overordnet arbeidsflyten til deg og din seksjon under håndteringen av en hendelse?
4	Så vidt du vet: hvor mange alarmer/hendelser behandler din seksjon eller team i løpet av en normal uke eller måned? Du kan svare presist eller med et grovt anslag.
5	Til hvor stor grad opplever du hendelseshåndteringsprosessen i din seksjon som *effektiv*? Utdyp hvorfor du mener det.
6	I løpet av din tid innenfor Norsk helsenett, til hvor stor grad opplever du at hendelseshåndtering har blitt mer effektiv fra du startet til nå? Hva er dine erfaringer med denne utviklingen?
7	I løpet av din tid innenfor Norsk helsenett, har det eksistert konkrete planer og mål om å effektivisere hendelseshåndtering? Hva har blitt gjort? Til hvilken grad har disse planene og målene blitt gjennomført eller nådd?
8	I dine øyne, til hvor stor grad er det <u>behov</u> for økt effektivisering av hendelseshåndtering i din seksjon? Til hvor stor grad er det <u>mulig</u> (med tanke på kostnader eller andre begrensende faktorer) å effektivisere?
9	Har du noen erfaringer du kan dele om en hendelse som har blitt håndtert spesielt bra eller spesielt dårlig? Hvorfor gikk det bra/dårlig?
10	Hvilke erfaringer har du rundt automatisering av hendelseshåndteringsprosessen i din seksjon?
11	Hvor stor del av dine arbeidsoppgaver er automatisert? Hvordan er det automatisert? Hvordan synes du at denne automatiseringen fungerer?
12	Hvilke deler av dine arbeidsoppgaver er spesielt preget av manuelt arbeid? Er det noen av disse oppgavene du tenker kan eller bør automatiseres? Hvorfor disse oppgavene?
13	Er det noen arbeidsoppgaver du mener *ikke* burde bli automatisert? Hvorfor?
14	Hvordan vil du beskrive din daglige arbeidsmengde? Er det arbeidsoppgaver du skulle ønske du kunne brukt mer tid på?

15	Har dere i dag de verktøyene dere behøver for å automatisere eksisterende og fremtidige prosesser? Hva er det som eventuelt mangler for at dere skal kunne gjøre dette?
16	I dine øyne, er det noen sentrale begrensninger eller utfordringer ved bruk av automatisering som en del av hendelseshåndtering?

Vedlegg B

Samtykkeskjema

Vil du delta i forskningsprosjektet *Effektivisering av hendelseshåndtering med automatisering*?

Dette er et spørsmål til deg om å delta i et forskningsprosjekt hvor formålet er å få innsikt i Norsk helsenett SF sine rutiner og prosedyrer omkring hendelseshåndtering. I dette skrivet gir vi deg informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

Formål
Formålet med denne undersøkelsen er å snakke med fagpersonell som jobber direkte eller indirekte med hendelseshåndtering hos Norsk helsenett for å få innsikt i hvordan prosessen gjennomføres i praksis, samt finne utfordringer og erfaringer som kan støtte oppunder problemstillingen og svare på våre forskningsspørsmål. Vi ønsker å få innsikt i deres daglige arbeidsoppgaver, og deres tanker og ønsker for fremtiden og hvordan prosessen kan effektiviseres. I tillegg vil vi sammenligne sentral teori rundt hendelseshåndtering opp mot personellets erfaringer fra arbeidslivet. Informasjonen som innhentes i denne undersøkelsen skal kun benyttes i bachelorprosjektet med tittel: *Effektivisering av hendelseshåndtering med automatisering*. Denne oppgaven vil bli offentliggjort, men all data vil anonymiseres, og sensitiv informasjon vil fjernes før offentliggjøring.

Problemstilling og forskningsspørsmål
Hvordan kan hendelseshåndtering effektiviseres med automatisering? Problemstillingen er valgt med bakgrunn i moderne organisasjoners utfordringer med å effektivisere hendelseshåndtering i takt med den raske digitale utviklingen og et trussellandskap i stadig endring. Denne oppgaven skal gjennomgå, forstå og foreslå løsninger for de forskjellige stegene som utgjør hendelseshåndtering. Dette skal underbygges med brukstester og konseptbevis av konkrete scenario og løsninger, som drøftes opp imot erfaringer og utfordringer fra sektoren. Dybdeintervju med fagpersonell skal illustrere hvordan prosessen gjennomføres i praksis, og legge til rette for drøfting av dagens utfordringer og morgendagens løsninger.

Forskningsspørsmål #1: Hvilke faser og steg i hendelseshåndtering er egnet for effektivisering gjennom automatisering?

Forskningsspørsmål #2: Hvordan kan en organisasjon gå frem for å velge riktig verktøy og metoder for effektivisering av hendelseshåndtering, som er i tråd med organisasjonens modningsgrad, behov og hensyn?

Hvem er ansvarlig for forskningsprosjektet?
Institutt for datateknologi og informatikk ved NTNU er ansvarlig for prosjektet.

Oppgaven utføres i samarbeid med Norsk helsenett SF, som er oppgavestiller for prosjektet.

Hvorfor får du spørsmål om å delta?
Du er valgt ut til å bli intervjuet, da din stilling eller ditt ansvarsområde tilsier at du har informasjon som anses som relevant for prosjektet.

Hva innebærer det for deg å delta?
Hvis du velger å delta i prosjektet, innebærer det at du er med på et dybdeintervju. Du får en intervjuguide på forhånd med forhåndsdefinerte spørsmål (se vedlegg). Underveis i intervjuet vil det stilles oppfølgingsspørsmål for å få mer informasjon eller for å få ytterligere utdyping av tema. Intervjuet er forventet å ta ca. 60-90 minutter. Vi tar lydopptak og notater fra intervjuet.

Det er frivillig å delta
Det er frivillig å delta i prosjektet. Hvis du velger å delta, kan du når som helst trekke samtykket tilbake uten å oppgi noen grunn. Alle dine personopplysninger vil da bli slettet. Det vil ikke ha noen negative konsekvenser for deg hvis du ikke vil delta eller senere velger å trekke deg. Tilbaketrekk av samtykke kan skje enten muntlig eller skriftlig via e-post til aadnets@stud.ntnu.no.

Ditt personvern – hvordan vi oppbevarer og bruker dine opplysninger
Vi vil bare bruke opplysningene om deg til formålene vi har fortalt om i dette skrivet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket. De eneste som har tilgang til dine personopplysninger er prosjektgruppen og det lagres på tilgangsstyrte områder som er godkjent for lagring av fortrolig informasjon.

Du som kandidat vil ikke kunne gjenkjennes i publikasjonen av oppgaven, og ingen personopplysninger vil publiseres. Kun alminnelige personopplysninger blir innsamlet for bruk under behandling av dataen. Hvis det er ønskelig kan du når som helst be om å se hvordan dine personopplysninger brukes i prosjektet.

Hva skjer med personopplysningene dine når forskningsprosjektet avsluttes?
Prosjektet vil etter planen avsluttes 22.05.2023. En måned etter prosjektet avsluttes, vil dine personopplysninger bli slettet.

Hva gir oss rett til å behandle personopplysninger om deg?
Vi behandler opplysninger om deg basert på ditt samtykke. På oppdrag fra NTNU har Sikt – Kunnskapssektorens tjenesteleverandør vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

Dine rettigheter
Så lenge du kan identifiseres i datamaterialet, har du rett til:

- innsyn i hvilke opplysninger vi behandler om deg, og å få utlevert en kopi av opplysningene
- å få rettet opplysninger om deg som er feil eller misvisende
- å få slettet personopplysninger om deg
- å sende klage til Datatilsynet om behandlingen av dine personopplysninger

Hvis du har spørsmål til studien, eller ønsker å vite mer om eller benytte deg av dine rettigheter, ta kontakt med: Joakim Klemets, joakim.klemets@ntnu.no
Vårt personvernombud: thomas.helgesen@ntnu.no

Hvis du har spørsmål knyttet til vurderingen som er gjort av personverntjenestene fra Sikt, kan du ta kontakt via:

- E-post: personverntjenester@sikt.no eller telefon: 73 98 40 40.

Med vennlig hilsen
Joakim Klemets *Ådne Tøftum Svendsrud*
(Forsker/veileder)

Samtykkeerklæring

Jeg har mottatt og forstått informasjon om prosjektet *Effektivisering av hendelseshåndtering med automatisering*, og har fått anledning til å stille spørsmål. Jeg samtykker til:

- ☐ å delta i dybdeintervju
- ☐ å delta i eventuell fokusgruppe i ettertid

Jeg samtykker til at mine opplysninger behandles frem til prosjektet er avsluttet

---

(Signert av prosjektdeltaker, dato)

Vedlegg C

Oppsett av testmiljø

Dette vedlegget forklarer hvordan testmiljøet for gjennomføring av mulighetsstudie og konseptbevis ble satt opp.

C.1 Oppsett av Jumpstation-maskin

Jumpstation-maskinen følger et enkelt oppsett med to formål: muliggjøre tilgang til testmiljøet fra prosjektmedlemmenes maskiner over internett, og støtte videre tilkobling til de andre maskinene og tjenestene i testmiljøet. Maskinen ble satt opp med Windows 11 Enterprise multi-session lisens, som tillatter flere samtidige *RDP*-tilkoblinger (*Remote Desktop Protocol*) mot maskinen. Videre ble maskinen tilegnet en offentlig IP-adresse som, og brannmurregler i Azure ble konfigurert for å kun tillate RDP-trafikk fra medlemmenes godkjente IP-adresser.

Maskinen ble konfigurert med en passordbeskyttet administratorbruker, og gjennom denne ble det opprettet en egen brukerkonto for hvert prosjektmedlem, også disse passordbeskyttet. På hver brukerkonto ble det generert egne *SSH*-nøkler (*Seure Shell*), som benyttes for videre tilkobling til Linux-maskinen i testmiljøet.

C.2 Oppsett av Sophos Firewall

Sophos Firewall er tilgjengelig direkte fra Azure Marketplace[1]. Deretter fulgte vi Sophos sin egen installasjonsguide[2] for å installere og sette opp Sophos Firewall i testmiljøet.

Under oppsett av testmiljøets infrastruktur ble det satt opp tre virtuelle nettverk: et nettverk for klienter (*ClientVnet*), et nettverk for kjernenettet der brannmuren og jumpstation-maskinen er plassert i hvert sitt subnettverk (*HubVnet*), og

[1] `https://azuremarketplace.microsoft.com/nb-no/marketplace/apps/sophos.sophos-xg-firewall-solution?tab=Overview`

[2] `https://docs.sophos.com/nsg/sophos-firewall/18.5/Help/en-us/webhelp/onlinehelp/HighAvailablityStartupGuide/HAConfiguration/HAAzureConfiguration/index.html`

et nettverk for verktøy og administrasjon (*ManagementVnet*). Kjernenettet er tilkoblet både klientnettet og administrasjonsnettet ved hjelp av virtuelle koblinger i Microsoft Azure, som gjør at enheter på ett nettverk kan kommunisere med maskiner på de andre nettverkene. Det er ingen virtuell kobling mellom klientnettet og administrasjonsnettet, som betyr at enhetene i disse nettverkene i utgangspunktet ikke vet hvordan de skal nå enheter på det andre nettet. For at enhetene skal kunne nå hverandre er det derfor nødvendig å sette opp statiske rutingtabeller på både klientnettet og administrasjonsnettet, som ruter trafikk til kjernenettet for videre ruting til endelig destinasjon. Denne tabellen er vist i figur C.1. Rutingregelen har en tosidig effekt, ettersom det både muliggjør kommunikasjon mellom klientnettet og administrasjonsnettet som ellers ikke har en kobling mellom seg, i tillegg til at det ruter all trafikk gjennom brannmuren (som har IP-adresse `10.0.1.4`) på vei mellom de to nettene.

Name ↑↓	Address prefix ↑↓	Next hop type ↑↓	Next hop IP address ↑↓
RouteTrafficToFirewall	0.0.0.0/0	VirtualAppliance	10.0.1.4

Figur C.1: Ruting av trafikk gjennom brannmur

Det betyr at all trafikk aggregeres hos brannmuren på vei mellom de to nettene, og brannmuren kan dermed konfigureres for å kontrollere hva slags trafikk som er tillatt, i tillegg til å logge trafikken. Vi konfigurerer brannmurregler som tillater trafikk til internett fra enheteter i alle nettverkene, trafikk mellom klientnettet og administrasjonsnettet, samt trafikk fra kjernenettet til administrasjonsnettet. Enheter i klientnettet og kjernenettet må kunne kommunisere med enhetene i administrasjonsnettet for å sende loggdata og annen trafikk til verktøyene, og enhetene i administrasjonsnettet må kunne kommunisere mot enheter i klientnettet for å utføre diverse handlinger. Disse reglene er vist i figur C.2.

	#	Name	Source	Destination	What	ID	Action	Feature and service
		VnetToVnetNetworkR... in 77.93 KB, out 2.84 MB						
	1	AllowClientToMgmtC... in 77.93 KB, out 2.84 MB	Any zone, ClientVnet	Any zone, MgmtVnet	Any service	#6	Accept	IPS AV WEB APP QoS HB LinkedNAT PRX LOG
	2	AllowMgmtToClientC... in 0 B, out 0 B	Any zone, MgmtVnet	Any zone, ClientVnet	Any service	#1	Accept	IPS AV WEB APP QoS HB LinkedNAT PRX LOG
	3	AllowHubToMgmtCo... in 0 B, out 0 B	Any zone, HubVnet	Any zone, MgmtVnet	Any service	#2	Accept	IPS AV WEB APP QoS HB LinkedNAT PRX LOG
		VnetToInternet in 363.85 MB, out 64.71 MB						
	4	AllowHubToInternet in 0 B, out 0 B	Any zone, HubVnet	WAN, Any host	Any service	#10	Accept	IPS AV WEB APP QoS HB LinkedNAT PRX LOG
	5	AllowMgmtToInterne... in 337.37 MB, out 61.12 MB	Any zone, MgmtVnet	WAN, Any host	Any service	#9	Accept	IPS AV WEB APP QoS HB LinkedNAT PRX LOG
	6	AllowClientToInter... in 26.48 MB, out 3.59 MB	Any zone, ClientVnet	WAN, Any host	HTTP, HTTPS	#8	Accept	IPS AV WEB APP QoS HB LinkedNAT PRX LOG
	7	Drop all in 0 B, out 0 B	Any zone, Any host	Any zone, Any host	Any service	#0	Drop	IPS AV WEB APP QoS HB LOG

Figur C.2: Nettverksregler på brannmur

Vi gjør oppmerksom på at brannmur med konfigurasjonsregler kun er satt opp

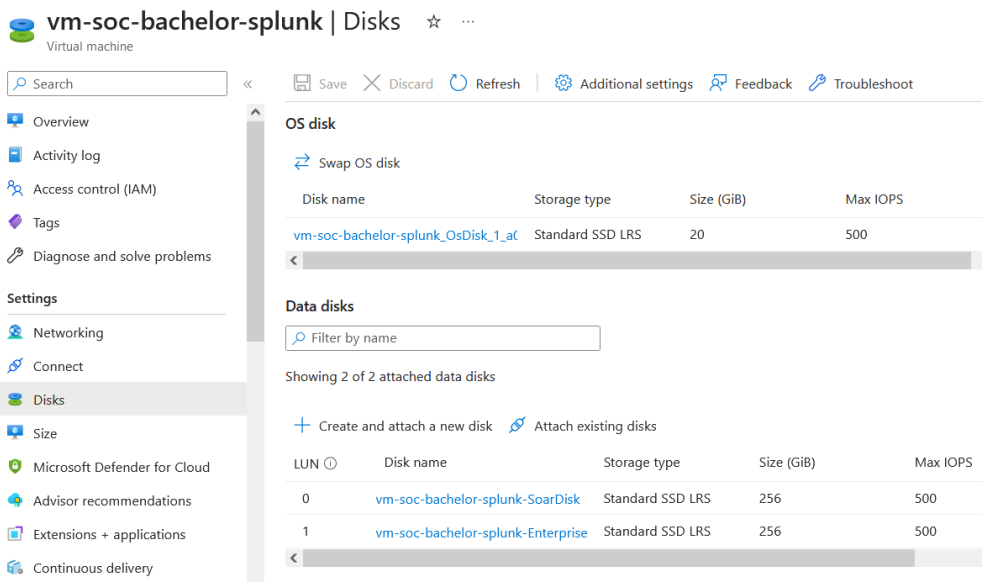
for å tilfredstille kravene til testmiljøet og mulighetsstudie. Fullstendig brannmurkonfigurasjon med alle sikkerhetstiltak som er nødvendig i et produksjonsmiljø er dermed ikke inkludert som en del av rapporten, og vår brannmurkonfigurasjon reflekterer ikke et slikt nivå av sikkerhet.

C.3 Oppsett av Splunk-maskin

Både Splunk Enterprise og Splunk SOAR, diskutert i seksjon C.4 og C.5, ble satt opp på samme Linux-maskin kjørende operativsystemet CentOS 7. Denne Linux-maskinen er kun tilgjengelig over SSH fra enheter med de SSH-nøklene som ble generert under oppsettet av Jumpstation-maskinen i seksjon C.1.

Ettersom begge tjenestene skal kjøre på samme maskin, og begge tjenestene krever en del plass til å lagre data, opprettes to disker på 256GB i Microsoft Azure, som vist i figur C.3. Disse diskene kobles til Linux-maskinens filsystem ved å først opprette to mapper i filsystemet, en for Splunk Enterprise og en for Splunk SOAR, og deretter «monteres» diskene (som foreløpig ligger tilkoblet som `sdc1` og `sdb1` på stien `/dev`) med `mount`-kommandoen:

```
sudo mkdir /opt/enterprise-disk/
sudo mkdir /opt/soar-disk
sudo mount /dev/sdc1 /opt/enterprise-disk/
sudo mount /dev/sdb1 /opt/soar-disk/
```



Figur C.3: Opprettelse av ekstra disker for Linux-maskin i Microsoft Azure

Deretter modifiserer vi filen `/etc/fstab` som vist i figur C.4, som er der Linux lagrer filsystemtabellen, for å sikre at denne monteringen skjer automatisk ved hver oppstart av maskinen.

```
#
# /etc/fstab
# Created by anaconda on Sun Jan 30 21:34:32 2022
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=43f9d994-9932-4adc-ac7d-49e7769a4fc5 / ext4 defaults 1 1
/dev/disk/cloud/azure_resource-part1 /mnt auto defaults,nofail,x-systemd.req
uires=cloud-init.service,comment=cloudconfig 0 2
UUID=f49206ac-1649-48c9-9043-f4eaba3474b8 /opt/soar-disk xfs defaults,nofail
 1 0
UUID=ef1dbf18-3a8e-4339-b5cb-9876ac7f610e /opt/enterprise-disk xfs defaults,
nofail 1 0
```

Figur C.4: Modifisering av `fstab` for å sikre automatisk montering av Azure-disker ved hver oppstart

C.4 Oppsett av Splunk Enterprise

For å installere og konfigurere Splunk Enterprise, følges Splunk sin egen guide[3]. Installasjonsfilen til Splunk Enterprise er tilgjengelig fra Splunk sine nettsider[4], og lastes ned til Linux-maskinen med `wget`-kommandoen:

```
wget https://download.splunk.com/products/splunk/releases/9.0.4/
    linux/splunk-9.0.4-de405f4a7979-linux-2.6-x86_64.rpm
```

Deretter installeres Splunk Enterprise fra `rpm`-filen, til mappen som ble satt opp for Splunk Enterprise i seksjon C.3:

```
sudo rpm -i --prefix=/opt/enterprise-disk splunk-9.0.4-de405f4a7979-
    linux-2.6-x86_64.rpm
```

Denne kommandoen vil installere Splunk Enterprise på stien `/opt/enterprise-disk/splunk/`. Etter installasjon kan tjenesten startes ved å kjøre `splunk`-programmet på stien `/opt/enterprise-disk/splunk/bin/`, som vist i figur C.5. Ved førstegangs oppstart blir man bedt om å opprette en administratorbruker og passord, som benyttes for pålogging til Splunk Enterprise sitt nettgrensesnitt.

```
admbachelor@vm-soc-bachelor-splunk: /opt/enterprise-disk/splunk/bin $  sudo ./splunk start --accept-license

This appears to be your first time running this version of Splunk.

Splunk software must create an administrator account during startup. Otherwise, you cannot log in.
Create credentials for the administrator account.
Characters do not appear on the screen when you type in credentials.

Please enter an administrator username:
```

Figur C.5: Førstegangsoppstart av Splunk Enterprise

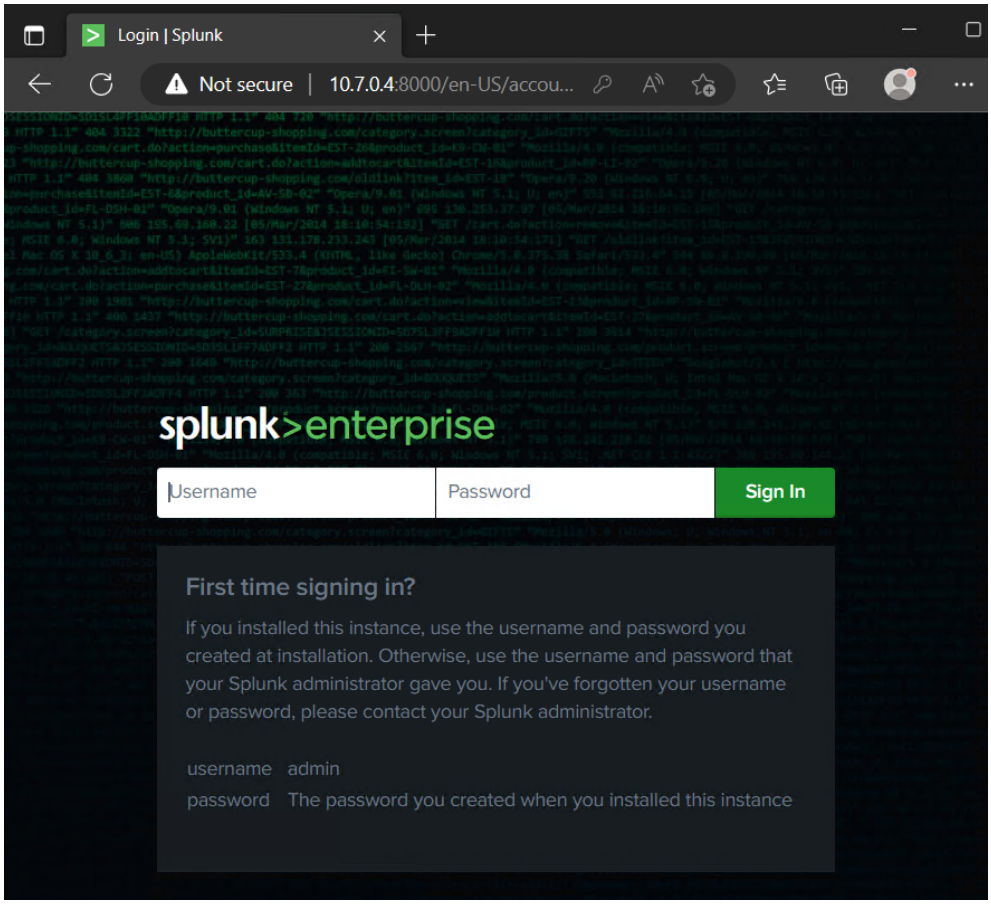
Etter at tjenesten er startet vil den være tilgjengelig gjennom nettleseren på

[3] `https://docs.splunk.com/Documentation/Splunk/9.0.4/SearchTutorial/InstallSplunk`
[4] `https://www.splunk.com/en_us/download/splunk-enterprise.html`

port 8000. En brannmuråpning i Linux-maskinens lokale brannmur er nødvendig for å tillate nettrafikk over denne porten:

```
sudo firewall-cmd --zone=public --add-port=8000/tcp --permanent
sudo firewall-cmd --reload
```

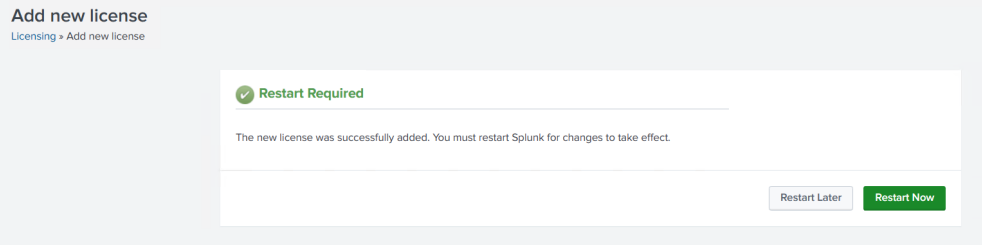
Deretter kan tjenesten nås som vist i figur C.6, der vi logger på med brukernavnet og passordet som ble konfigurert. Det første vi må gjøre etter pålogging er å registrere lisensen som benyttes. Når denne er vellykket registrert kan tjenesten restartes som vist i figur C.7.



Figur C.6: Pålogging til Splunk Enterprise sitt nettgrensesnitt

For å sikre at Splunk Enterprise automatisk startes ved hver oppstart av Linux-maskinen, kjøres følgende kommandoer for å stoppe tjenesten, aktivere automatisk oppstart, og starte tjenesten på nytt:

```
sudo /opt/enterprise-disk/splunk/bin/splunk stop
sudo /opt/enterprise-disk/splunk/bin/splunk enable boot-start
sudo /opt/enterprise-disk/splunk/bin/splunk start
```



Figur C.7: Registrering av Splunk Enterprise lisens

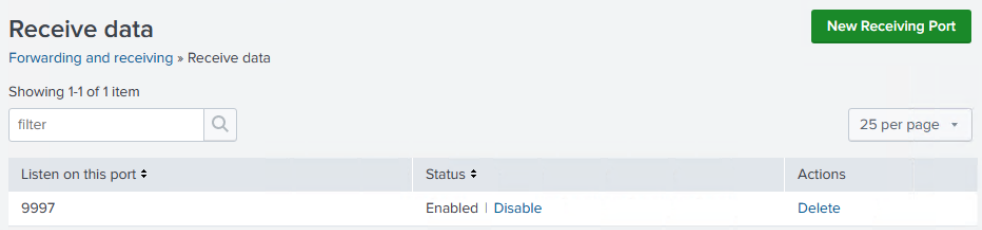
C.4.1 Oppsett av mottaker og indekser i Splunk Enterprise

Under oppsettet setter vi i tillegg opp en mottaker[5] i Splunk Enterprise. I vanlige Splunk-miljø er det normalt at mottakere installeres på indeksere[6], Splunk-instanser som er konfigurert for å omgjøre rådata til events, og lagrer dette i indekser i Splunk Enterprise. Et Splunk-miljø kan settes opp med mange indeksere, som mottar data fra mange forskjellige kilder. I vårt testmiljø kjører vi kun én instans av Splunk Enterprise, og denne ene instansen fungerer derfor som testmiljøets eneste indekser, og den må settes opp med en mottaker for å kunne ta i mot data.

Oppsett av mottaker kan enten gjøres i Splunk Enterprise sitt nettgrensesnitt eller gjennom kommandolinjen på serveren. Gjennom nettgrensesnittet gjøres dette ved å gå på `Settings > Forwarding and receiving > Configure receiving > New Receiving Port`, og deretter skrive inn ønsket portnummer og trykke `Save`. Vi gjorde dette direkte gjennom kommandolinjen ved å gå til Splunk Enterprise sin hjemmemappe på serveren, og deretter undermappen `etc/system/local`. Her opprettet vi en fil kalt `inputs.conf` der vi la inn følgende innhold for å sette opp en mottaker på port 9997:

```
1 [splunktcp://9997]
2 disabled = 0
```

Etter restart av Splunk Enterprise vil mottakeren være aktivert, som kan bekreftes gjennom nettgrensesnittet som vist i figur C.8:



Figur C.8: Mottaker aktivert i Splunk Enterprise

[5] `https://docs.splunk.com/Splexicon:Receiver`
[6] `https://docs.splunk.com/Splexicon:Indexer`

En ny brannmuråpning på Linux-serveren er nødvendig for at data skal kunne sendes til mottakeren gjennom denne portåpningen:

```
sudo firewall-cmd --zone=public --add-port=9997/tcp --permanent
sudo firewall-cmd --reload
```

På dette tidspunktet vil instansen av Splunk Enterprise kontinuerlig lytte på denne porten for å fange opp og indeksere all loggdata som sendes til Splunk Enterprise.

C.5 Oppsett av Splunk SOAR

I likhet med Splunk Enterprise er installasjonsfilen til Splunk SOAR tilgjengelig fra Splunk sine nettsider[7]. I denne oppgaven vil vi benytte Splunk SOAR «Community Edition», som er gratis med noe begrenset funksjonalitet. Vi følger Splunk sin guide for installasjon og konfigurasjon[8].

Splunk SOAR lastes ned med `wget`-kommandoen, og installasjonsfilen pakkes ut:

```
wget -O splunk_soar-unpriv-6.0.0.114895-cb859067-el7-x86_64.tgz "
    https://download.splunk.com/products/splunk_soar-unpriv/releases
    /6.0.0/linux/splunk_soar-unpriv-6.0.0.114895-cb859067-el7-x86_64
    .tgz"
tar -xzvf ./splunk_soar-unpriv-6.0.0.114895-cb859067-el7-x86_64.tgz
```

Dette vil pakke ut den nedlastede `tgz`-filen til mappen `splunk-soar`. Vi oppretter en hjemmemappe for Splunk SOAR på stien vi opprettet i seksjon C.3, og kjører deretter programmet `soar-prepare-system` fra den utpakkede installasjonsmappa for å forberede systemet for installasjon:

```
sudo mkdir /opt/soar-disk/splunk-soar/
sudo ~/splunk-soar/soar-prepare-system --splunk-soar-home /opt/soar-
    disk/splunk-soar/ --https-port 8443
```

Denne før-installasjonen vil blant annet installere nødvendige pakker, åpne opp nødvendige brannmuråpninger og porter, og opprette en egen bruker på Linux-maskinen med navn *phantom* (fra Splunk SOAR sitt tidligere navn, Splunk Phantom). Installasjonen av Splunk SOAR må fullføres fra denne nyopprettede brukeren, og dette oppnås ved å flytte den utpakkede installasjonsmappen fra administratorbrukerens hjemmemappe til *phantom*-brukerens hjemmemappe, endre rettigheter på mappen, logge inn på *phantom*-brukeren og fullføre installasjonen:

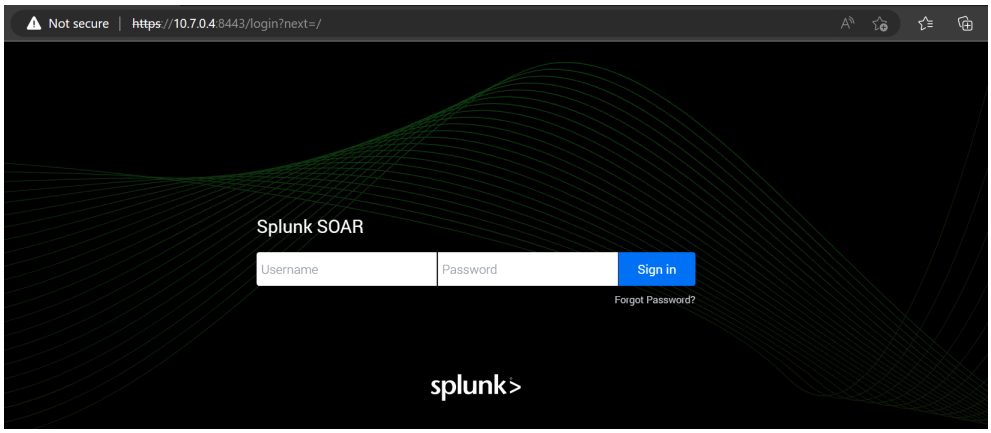
```
sudo su
cp -r /home/admbachelor/splunk-soar/ /home/phantom/
cd /home/phantom/
```

[7] `https://www.splunk.com/en_us/download/soar-free-trial.html`

[8] `https://docs.splunk.com/Documentation/SOARonprem/6.0.0/Install/InstallUnprivileged`

```
chmod 755 splunk-soar/
su -l phantom
./splunk-soar/soar-install --splunk-soar-home /opt/soar-disk/splunk-
    soar/ --https-port 8443 --ignore-warnings
```

Vi bruker `-ignore-warnings` for å ignorere et par advarsler som forteller oss at vår maskin ikke har den anbefalte mengden lagringsplass, eller støtter integrasjon mot Splunk SOAR mobilapplikasjon. Ettersom vi ikke trenger dette for vårt testoppsett så kan vi ignorere dette med `-ignore-warnings` for å fullføre installasjonen. Når installasjonen er ferdig kan vi logge på nettgrensesnittet til Splunk SOAR på port 8443, som vist i figur C.9. Som standard er brukernavnet satt til `soar_local_admin` og passordet satt til `password`. Passordet endres ved førstegangs pålogging.



Figur C.9: Pålogging til Splunk SOAR sitt nettgrensesnitt

C.6 Oppsett av Windows-klient

Windows-klienten i testmiljøet settes opp med Sysmon[9] for generering av loggdata. Sysmon er en loggtjeneste og driver som kjører i bakgrunnen på Windows-klienter for å logge konfigurerbare events til Windows Event Viewer[10]. Sysmon er en utvidelse av Windows Event Viewer som blant annet gir fordeler i form av utvidet og detaljert logging av blant annet prosesser og nettverkstrafikk, lettere deteksjon av indikatorer på hendelser, og lettere sporbarhet av eventer under analysen av en hendelse [133]. Sysmon lastes ned og installeres på Windows-klienten fra Microsoft sine sider, og kan deretter startes med følgende kommando:

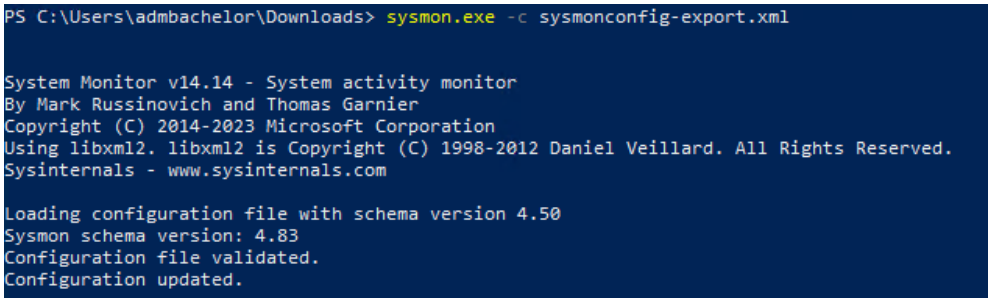
```
sysmon -accepteula -i
```

[9] `https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon`
[10] `https://learn.microsoft.com/en-us/shows/inside/event-viewer`

Logging av data begynner umiddelbart. Loggene generert av Sysmon vises i Windows Event Viewer under `Applications and Services Logs/Microsoft-/Windows/Sysmon/Operational`.

Splunk anbefaler bruk av egendefinerte Sysmon-konfigurasjoner, istedenfor å bruke den konfigurasjonsfilen som følger med Sysmon. Vi bruker en konfigurasjonsfil fra SwiftOnSecurity[11], som er anbefalt av Splunk for Sysmon-logging på Windows-klienter. Etter konfigurasjonsfilen er lastet ned, kjøres følgende kommando for å oppdatere Sysmon som vist i figur C.10:

```
sysmon.exe -c sysmonconfig-export.xml
```



Figur C.10: Oppdatering av Sysmon konfigurasjon

[11] `https://github.com/SwiftOnSecurity/sysmon-config`

Vedlegg D

Dokumentasjon

Dette vedlegget dokumenterer konfigurasjonen av verktøyene og systemene i testmiljøet, og hvordan de blir brukt under rapportens mulighetsstudie.

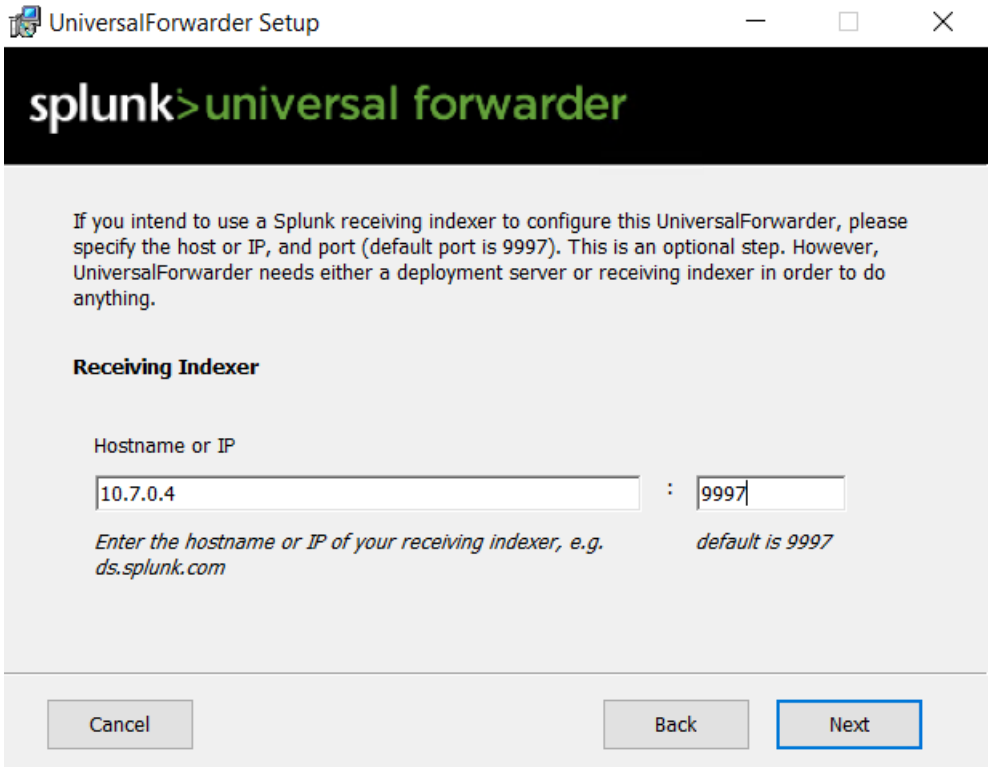
D.1 Sende loggdata fra klient til Splunk Enterprise

Splunk Universal Forwarder lastes ned og installeres på Windows-klienten fra Splunk sine sider[1]. Installasjonsprogrammet ber om hvilken type Splunk-oppsett som benyttes, og her velger vi `On-prem Splunk Solution` ettersom vi kjører en instanse av Splunk Enterprise på egen server. I tillegg må det konfigureres et brukernavn og passord for en administratorkonto benyttet av Universal Forwarder. Videre er det mulig å konfigurere en *Deployment server*[2] dersom det benyttes, men ettersom det ikke blir brukt i vårt testmiljø kan dette steget hoppes over. Tilslutt må det konfigureres en *Receiving indexer*, som bestemmer hvor Universal Forwarder skal sende Sysmon-loggene. Her skriver vi inn IP-adressen til serveren der Splunk Enterprise kjører (`10.7.0.4`), og porten som ble konfigurert som mottaker tidligere, `9997`. Dette er vist i figur D.1.

Deretter skal konfigurasjonen endres slik at Splunk Universal Forwarder sender ønsket loggdata til Splunk Enterprise. Dette kan gjøres gjennom kommandolinje eller ved å endre konfigurasjonsfilene direkte. Konfigurasjonsfilene til Splunk Universal Forwarder ligger på stien `C:\ProgramFiles\SplunkUniversalForwarder\etc\system\local`. Installasjonen av Universal Forwarder vil allerede ha satt opp filene `server.conf` og `outputs.conf`, som henholdsvis inkluderer informasjon om Windows-klienten og Splunk Indexeren der loggdataen skal sendes. Disse trenger ikke ytterligere oppsett for formålet med denne rapporten, men ytterligere konfigurasjon er mulig. Det som mangler er informasjon om hva Splunk Universal Forwarder skal samle inn av data på klienten og sende til indekseren. Dette bestemmes av konfigurasjonsfilen `inputs.conf`, som må opprettes manuelt

[1] `http://www.splunk.com/en_us/download/universal-forwarder.html`

[2] `https://docs.splunk.com/Documentation/Splunk/9.0.4/Updating/Aboutdeploymentserver`



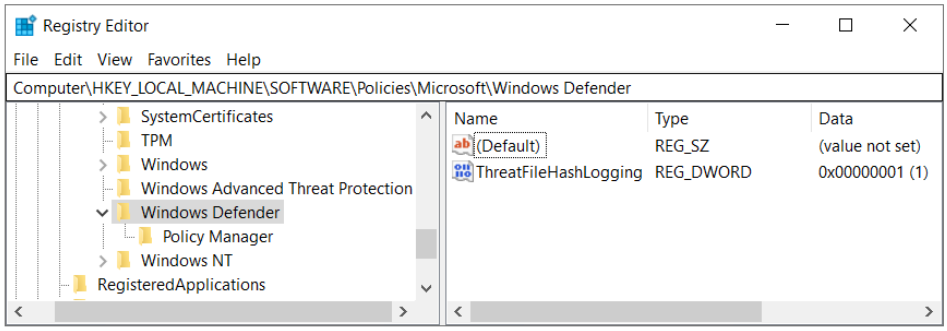
Figur D.1: Valg av mottakende indekser ved oppsett av Splunk Universal Forwarder

ved førstegangsoppsett. Vi fyller inn følgende innhold for å sette et hostnavn, og fortelle Splunk Universal Forwarder at den skal sende Sysmon-logger, Microsoft Defender-logger og flere forskjellige ytelseslogger:

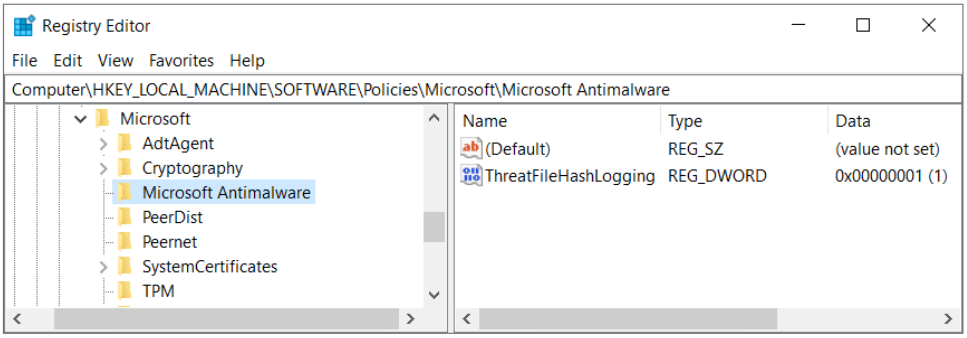
```
[default]
host = vm-windows-client

[WinEventLog://Microsoft-Windows-Sysmon/Operational]
disabled = 0

[WinEventLog://Microsoft-Windows-Windows Defender/Operational]
disabled = 0

[perfmon://CPU]
disabled = 0
object = Process*
counters = *
instances = *
interval = 10
useEnglishOnly=true

[perfmon://Memory]
disabled = 0
object = Memory
counters = *
instances = *
interval = 10
useEnglishOnly=true

[perfmon://Network]
disabled = 0
object = Network interface
counters = *
instances = *
interval = 10
useEnglishOnly=true

[perfmon://PhysicalDisk]
disabled = 0
object = PhysicalDisk
counters = *
instances = *
interval = 10
useEnglishOnly=true

```

```
[perfmon://LogicalDisk]
disabled = 0
object = LogicalDisk
counters = *
instances = *
interval = 10
useEnglishOnly=true
```

I tillegg gjør vi en regelendring på klienten som aktiverer en funksjon i Microsoft Defender som sjekker og logger hash-verdien til ondsinnet programvare oppdaget av Microsoft Defender. Dette er noe vi ønsker å aktivere ettersom hash-verdien til ondsinnet programvare er nyttig ved analyse og håndtering av hendelser som involverer ondsinnet programvare. For å oppnå dette opprettes regelen `ThreatFileHashLogging` under både `Computer/HKEY_LOCAL_MACHINE/SOFTWARE/Policies/Microsoft/Windows Defender` og `Computer/HKEY_LOCAL_MACHINE/SOFTWARE/Policies/Microsoft/Microsoft Antimalware` i *Registry Editor* på Windows-klienten:



Figur D.2: Legge til regler i *Registry Editor* [1/2]



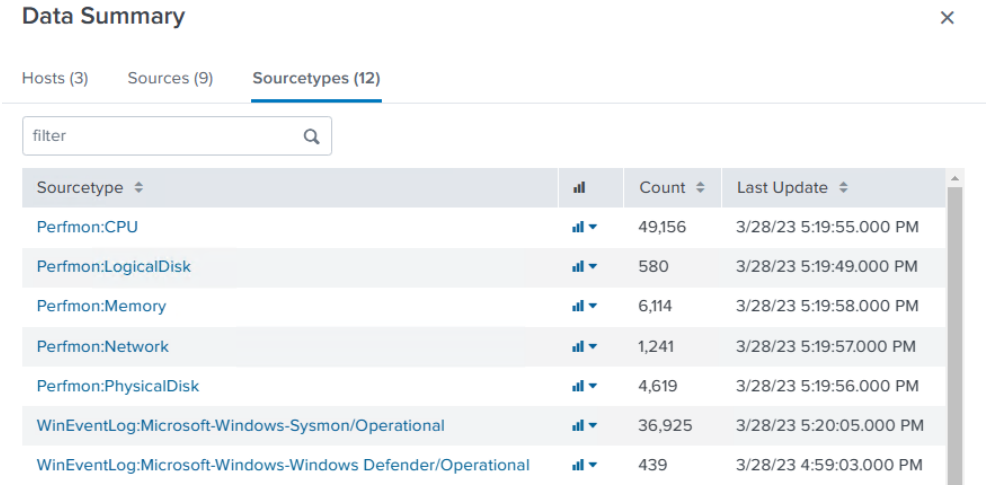
Figur D.3: Legge til regler i *Registry Editor* [2/2]

Deretter må Universal Forwarder restartes for at endringene skal ta effekt, dette gjøres ved å kjøre følgende Powershell-kommandoer:

```
Stop-Service SplunkForwarder
Start-Service SplunkForwarder
```

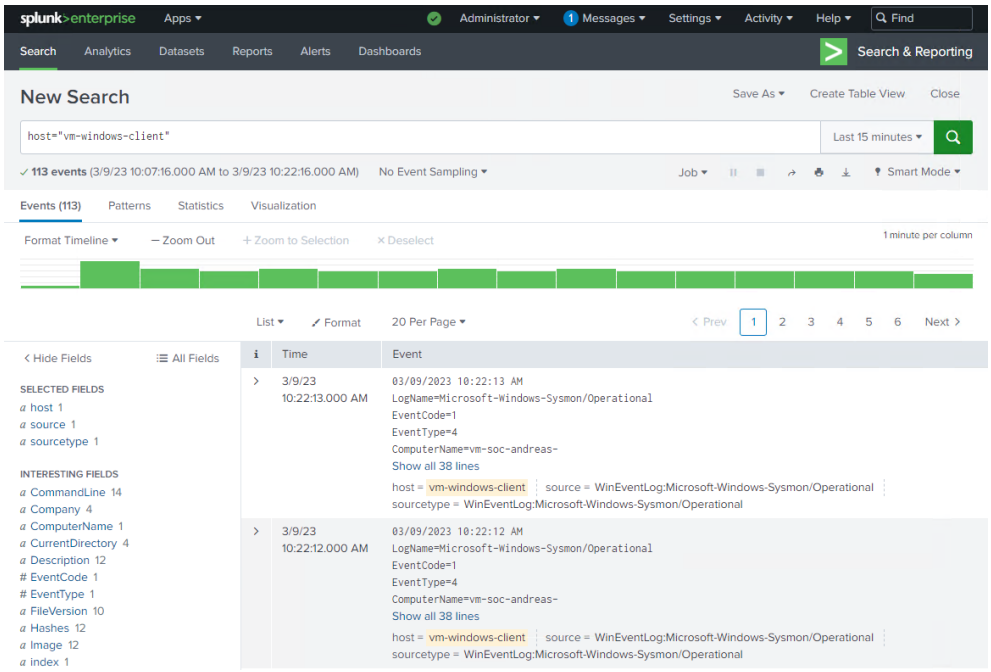
Når Universal Forwarder har startet opp på nytt skal den automatisk begynne å sende loggdata til mottakeren vi satt opp på Splunk-serveren. Dette kan vi verifisere i Splunk Enterprise ved å gå inn på appen `Search & Reporting`, gå til `Search` og velge `Data Summary`. Da vil det vises informasjon om hoster som har sendt data til Splunk, samt datakilder. Figur D.4 og D.5 viser vår ene Windows-klient, og datakildene Sysmon og Microsoft Defender. Denne dataen fra klienten er umiddelbart tilgjengelig for søk og behandling i Splunk-enterprise, som vist i figur D.6.

Figur D.4: Windows-klienten vises under *Host*

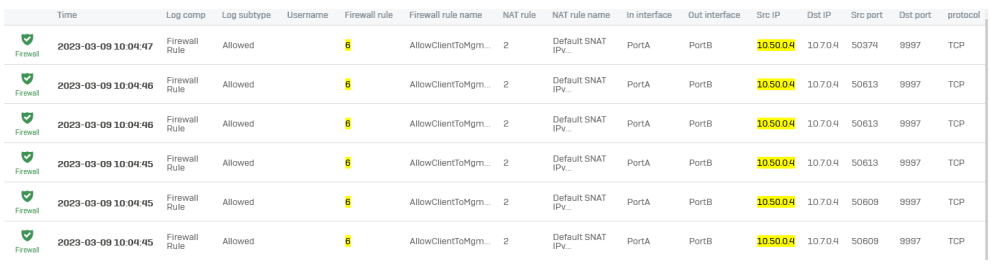


Figur D.5: Loggkildene vises under *Sources*

Trafikken mellom klienten og mottakeren vises også i loggene til Sophos Firewall, som fist i figur D.7.

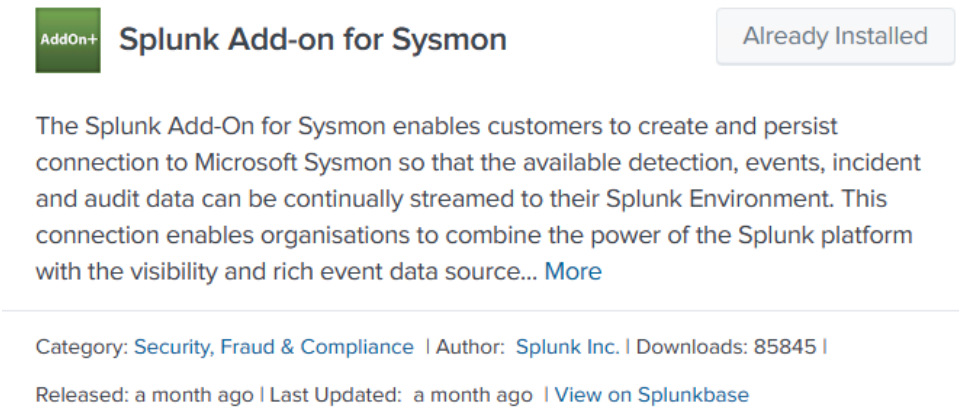


Figur D.6: Data fra klienten i Splunk



Figur D.7: Trafikk mellom klient og Splunk som logges i brannmuren

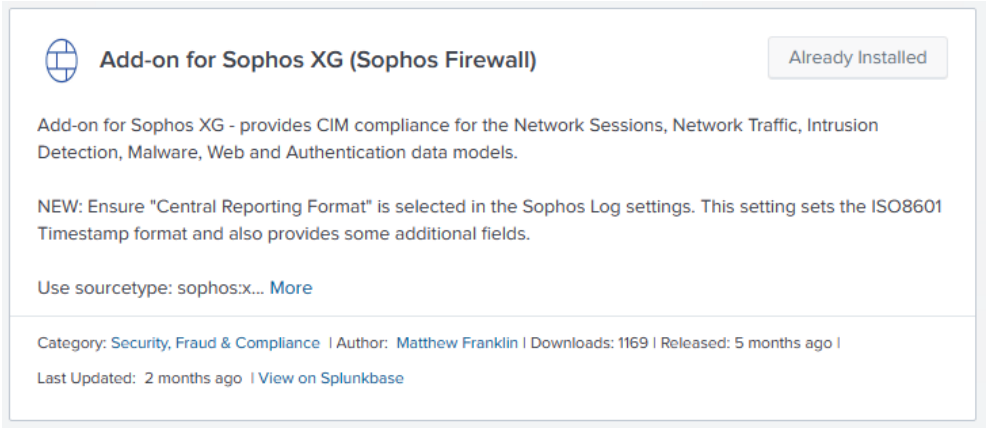
Splunk Enterprise har en add-on for Sysmon som er designet for å berike Sysmon-data i Splunk, for lettere søk og behandling av dataen. Denne installeres i Splunk Enterprise gjennom web-grensesnittet som vist i figur D.8.



Figur D.8: Installasjon av Sysmon add-on i Splunk Enterprise

D.2 Sende loggdata fra brannmur til Splunk Enterprise

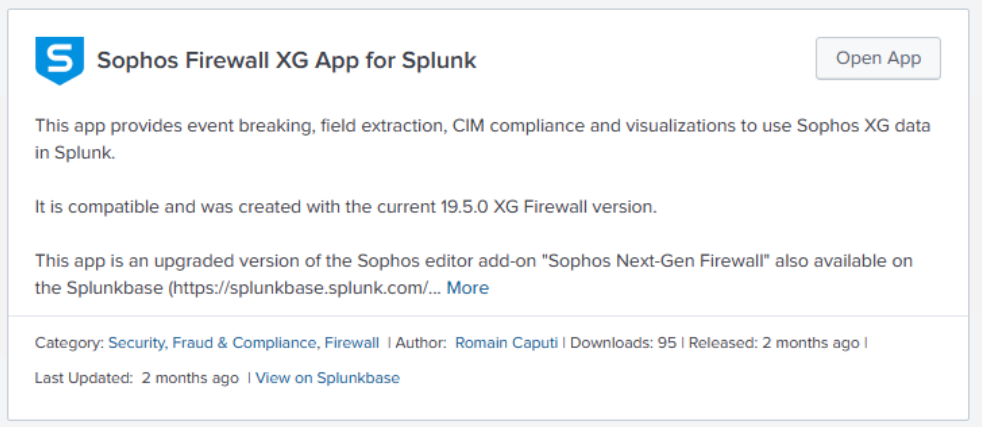
For å sende loggdata generert av Sophos Firewall til Splunk Enterprise installeres først to tillegg for Sophos Firewall i Splunk Enterprise, som vist i figur D.9 og D.10.



Figur D.9: Installasjon av Sophos-tillegg i Splunk Enterprise [1/2]

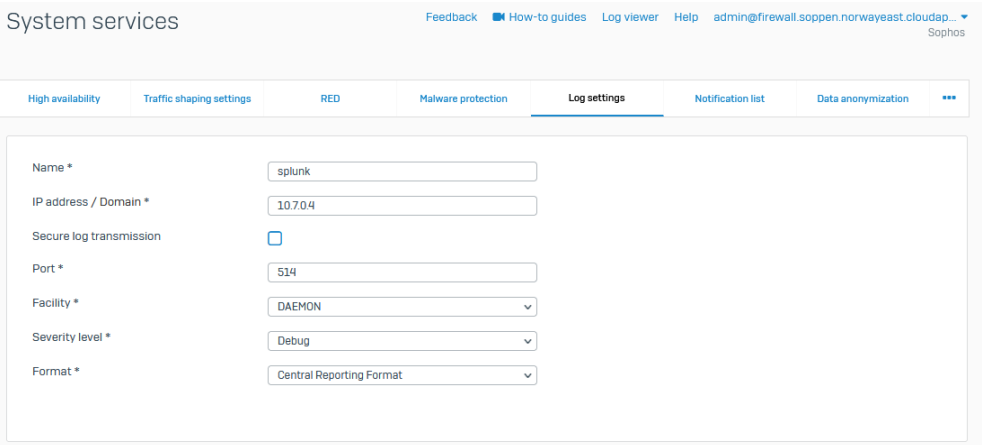
Deretter, fra administratorgrensesnittet til Sophos Firewall, settes det opp en syslog-tjeneste[3] som skal samle inn loggdata fra brannmuren og sende det til Splunk Enterprise som UDP-trafikk. Dette gjøres ved å gå til `Configure > System`

[3] `https://linux.die.net/man/8/syslogd`



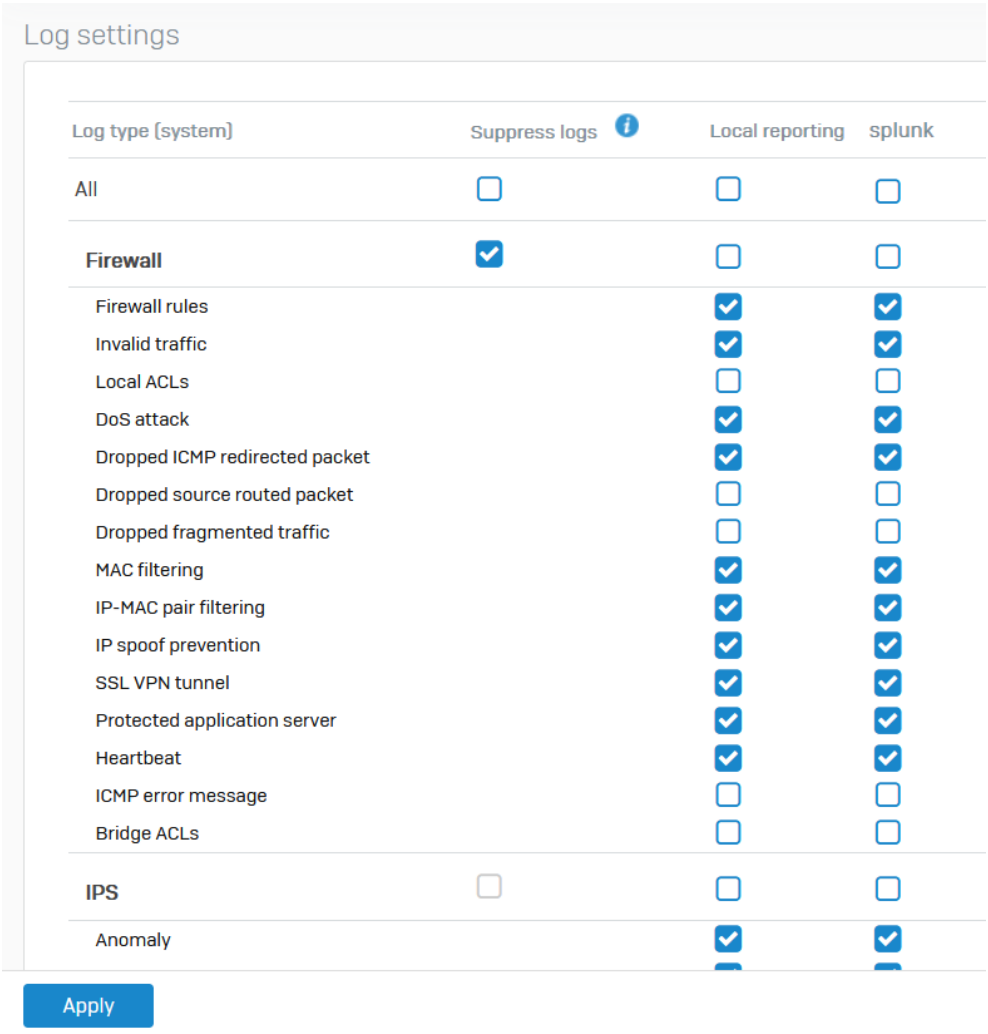
Figur D.10: Installasjon av Sophos-tillegg i Splunk Enterprise [2/2]

`services > Log settings`, og deretter trykke `Add`. I menyen som dukker opp konfigurerer vi et navn for syslog-tjenesten, `splunk`, samt IP-adressen og portnummeret som loggene skal sendes til, som i vårt tilfelle er Splunk Enterprise med IP-adresse `10.7.0.4` og portnummer `514`. Resten av innstillingene lar vi stå som standard. Dette er vist i figur D.11. Deretter velger vi hvilke loggkilder som skal sendes som en del av denne syslog-tjenesten, ved å velge de under kolonnen `splunk`, som vist i figur D.12. På figuren vises kun et utvalg av de mulige valgene.



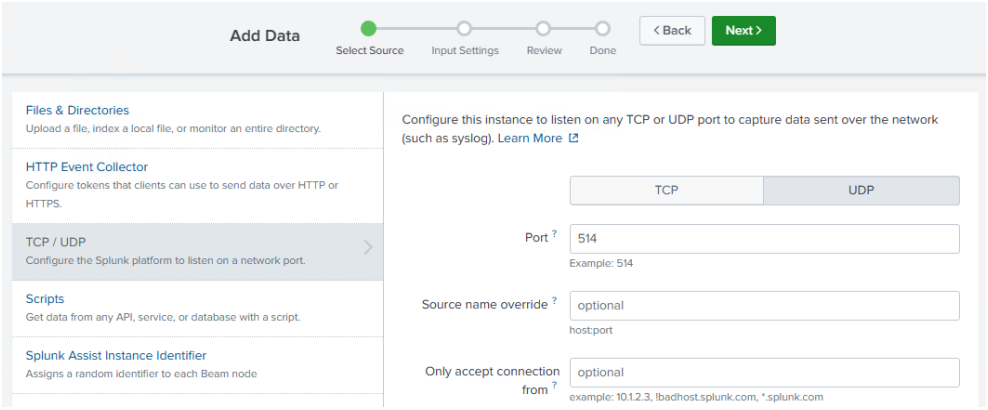
Figur D.11: Oppsett av Syslog i Sophos Firewall

I Splunk Enterprise konfigurerer vi en inndatakilde for UDP som skal ta i mot loggdataen som blir sendt fra Sophos Firewall. Dette gjøres fra nettgrensesnittet under `Settings > Data inputs > UDP`, og deretter `New Local UDP`. I menyen som vises i figur D.13 skriver vi inn portnummer `514`, det samme som vi valgte i Sophos Firewall. I påfølgende meny vist i figur D.14, velger vi `sophos:xg:logs` som *Source type* og `Add-on for Sophos XG` som *App context*. Dette er begge valg som stammer fra tilleggene for Sophos Firewall som vi installerte i Splunk En-

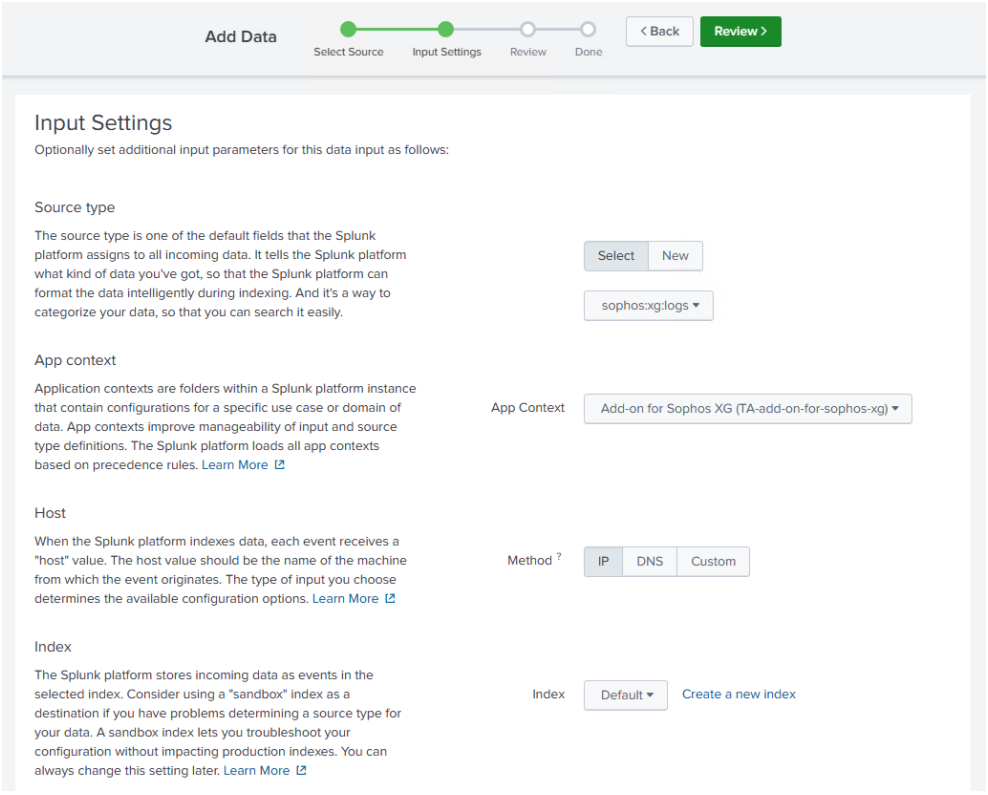


Figur D.12: Oppsett av Syslog i Sophos Firewall

terprise tidligere. Resten av innstillingene lar vi stå som standard, og oppsettet fullføres som vist i figur D.15.



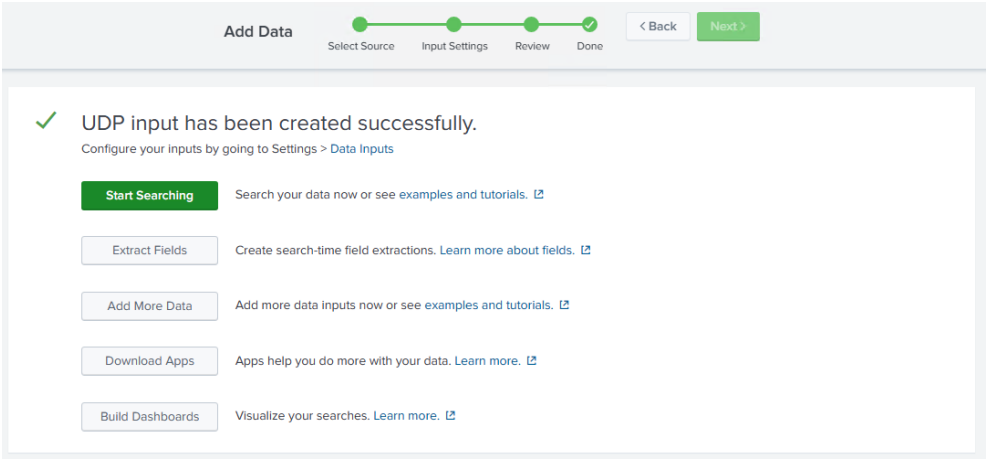
Figur D.13: Oppsett av UDP-lytter i Splunk Enterprise [1/3]



Figur D.14: Oppsett av UDP-lytter i Splunk Enterprise [2/3]

For å tillate UDP-trafikk til port 514 må vi i tillegg åpne opp for porten i Linux-maskinens lokale brannmur:

```
sudo firewall-cmd --zone=public --add-port=514/udp --permanent
```

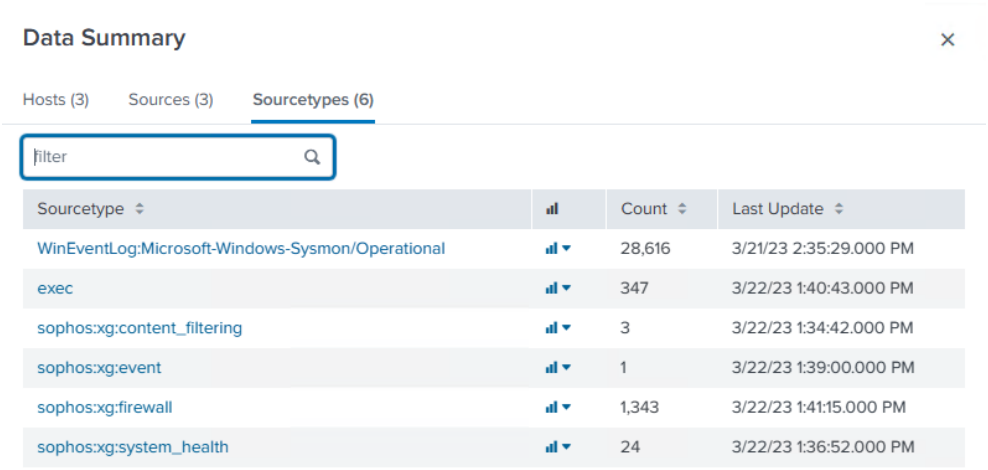


Figur D.15: Oppsett av UDP-lytter i Splunk Enterprise [3/3]

```
sudo firewall-cmd --reload
```

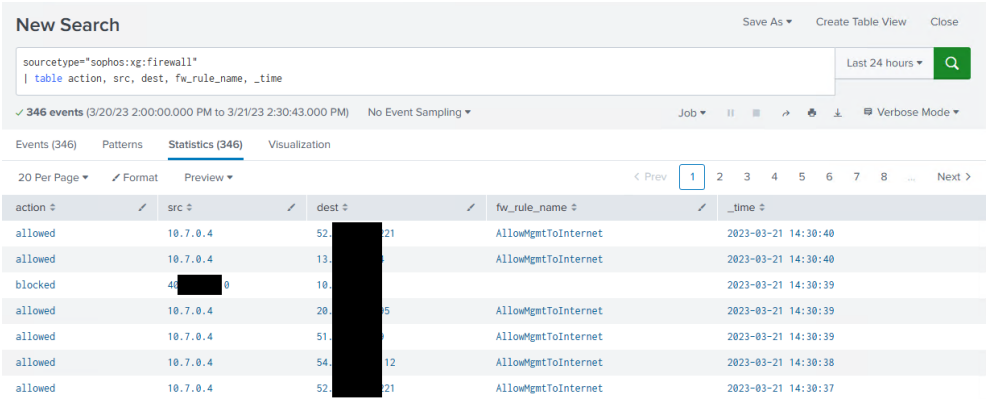
På dette tidspunktet vil valgt loggdata kontinuerlig sendes fra Sophos Firewall sin syslog-tjeneste og indekseres i Splunk Enterprise, som vist i figur D.16. Søket under viser deler av den informasjonen vi kan trekke ut fra brannmurloggene, som vist i figur D.17.

```
sourcetype="sophos:xg:firewall" | table action, src, dest,
    fw_rule_name, _time
```



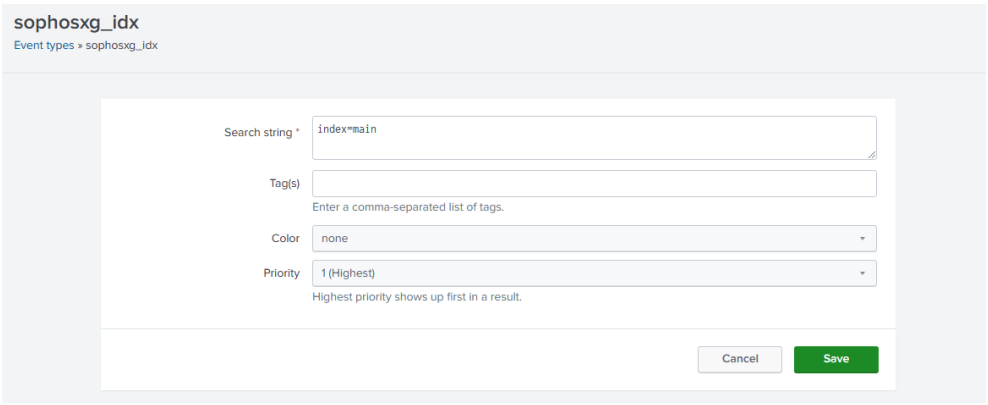
Figur D.16: Mottak av brannmurdata i Splunk

Sophos har i tillegg utviklet et dashbord for visualisering av brannmurlogger i Splunk Enterprise. For at dette dashbordet skal fungere må noen ytterligere innstillinger endres. Først må en event-type for Sophos Firewall konfigureres ved å gå til



Figur D.17: Søke gjennom brannmurdata i Splunk

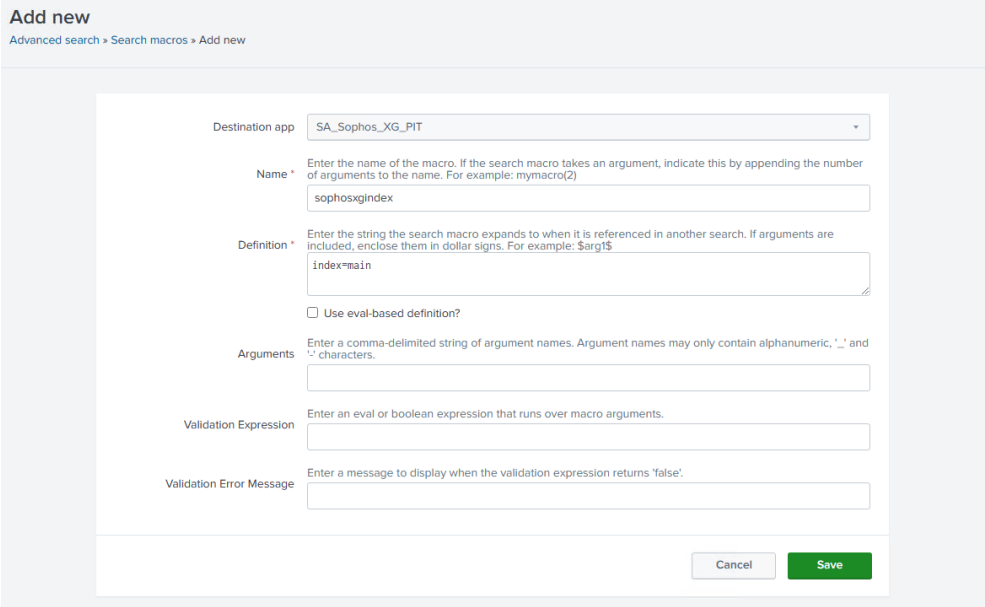
`Settings > Event types`, velge appen *Sophos XG App for Splunk*, og variabelen `sophosxg_idx`. I menyen som dukker opp legger vi til indeksen i Splunk Enterprise der brannmurloggene kommer inn, som i vårt tilfelle er indeksen *main*. Derfor skriver vi `index=main` i boksen *Search string*, som vist i figur D.18.



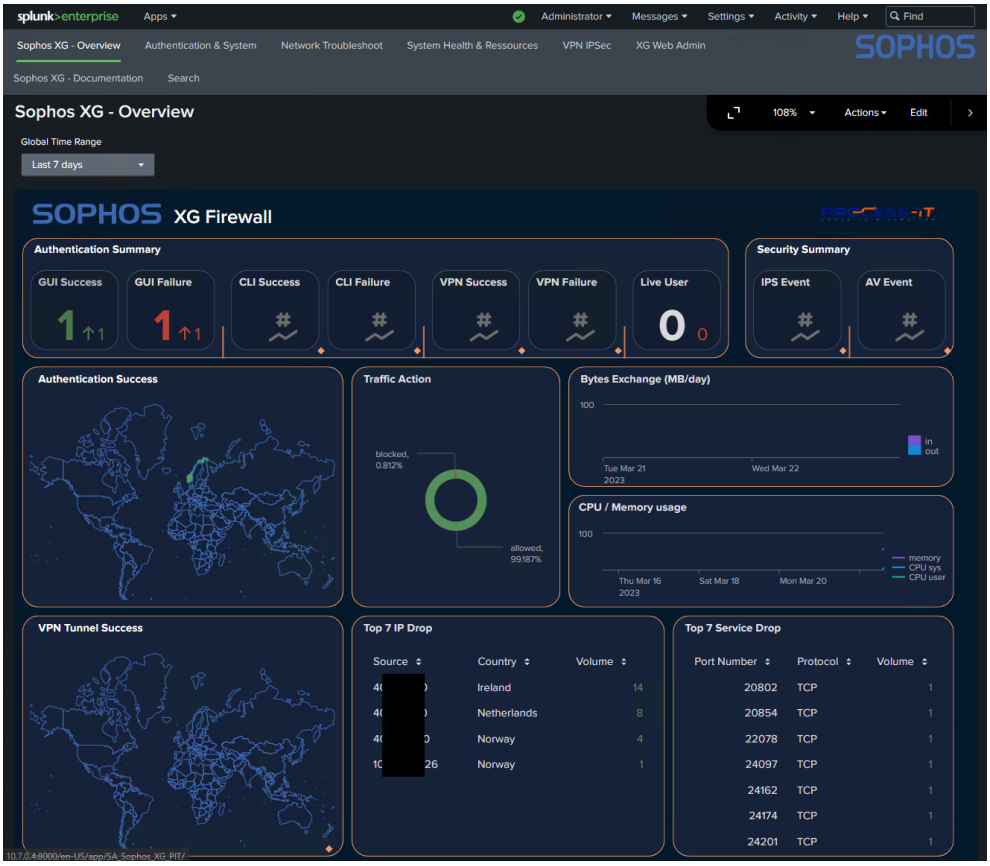
Figur D.18: Konfigurere event-type for Sophos Firewall i Splunk Enterprise

I tillegg må vi konfigurere en søkemakro ved å gå til `Settings > Advanced search > Search macros` og velge `Add new`. Her velger vi `SA_Sophos_XG_PIT` som *Destination app*, `sophosxgindex` som *Name*, og `index=main` som *Definition*, vist i figur D.19.

Når disse innstillingene er på plass kan dashbordet åpnes ved å gå til `Apps` og velge `Sophos XG App for Splunk`. Dashbordet inkluderer en oversikt, samt flere undersider som visualiserer forskjellige deler av brannmurloggene, som vist i figur D.20.



Figur D.19: Konfigurere søkemakro for Sophos Firewall i Splunk Enterprise

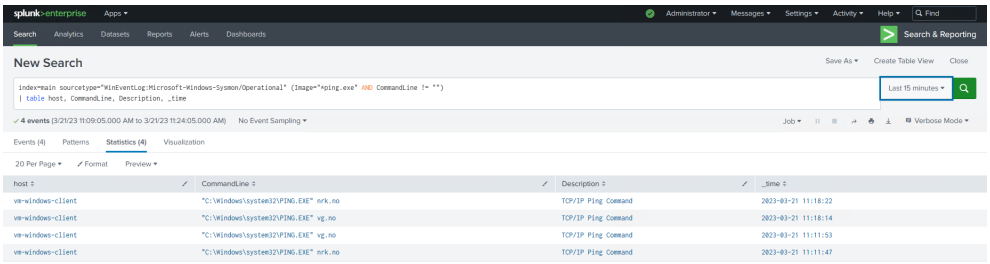


Figur D.20: Sophos Firewall dashboard i Splunk Enterprise

D.3 Generere alarmer i Splunk Enterprise

Enhver alarm i Splunk Enterprise begynner som et søk. Søk defineres for å plukke opp den eller de eventene (søkeresultatene) som indikerer at en alarm skal genereres. Et eksempel på et slikt søk er vist under, og resultatet er vist i figur D.21. Dette søket fanger opp alle *ping* som har blitt gjennomført av enheter som logger til Splunk Enterprise siste 15 minutter, og henter ut blant annet hvilken enhet som har kjørt kommandoen og hvilken IP-adresse eller domene det var rettet mot.

```
index=main sourcetype="WinEventLog:Microsoft-Windows-Sysmon/
    Operational" (Image="*ping.exe" AND CommandLine != "") | table
    host, CommandLine, Description, _time
```



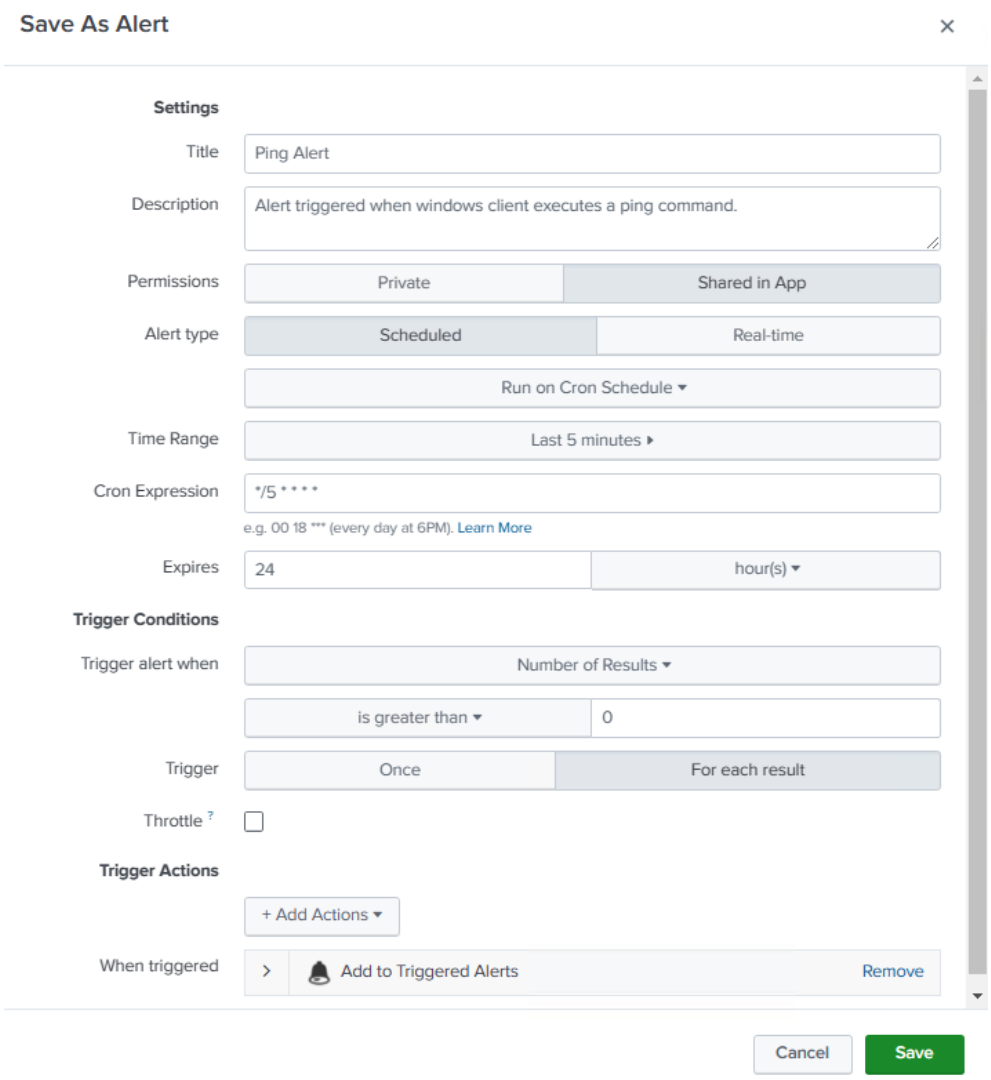
Figur D.21: Lage søk som skal bli til en alarm

For å sette opp en alarm basert på søket velges `Save As` fra resultatsiden, og i menyen som dukker opp i figur D.22, velges tittel og beskrivelse på alarmen og hvorvidt alarmen er privat eller delt med andre apper i Splunk Enterprise (vi velger sistnevnte for å muliggjøre videresending av alarmen til Splunk SOAR senere). Videre velger vi hvorvidt søket som underbygger alarmen skal kjøres i faste intervaller, eller om det skal søkes kontinuerlig. Sistnevnte vil innebære en større last på systemet, spesielt hvis man har mange søk og mange alarmer, men det vil samtidig muliggjøre så rask alarmering som mulig. I dette tilfellet velger vi at søket skal kjøres hvert 5. minutt.

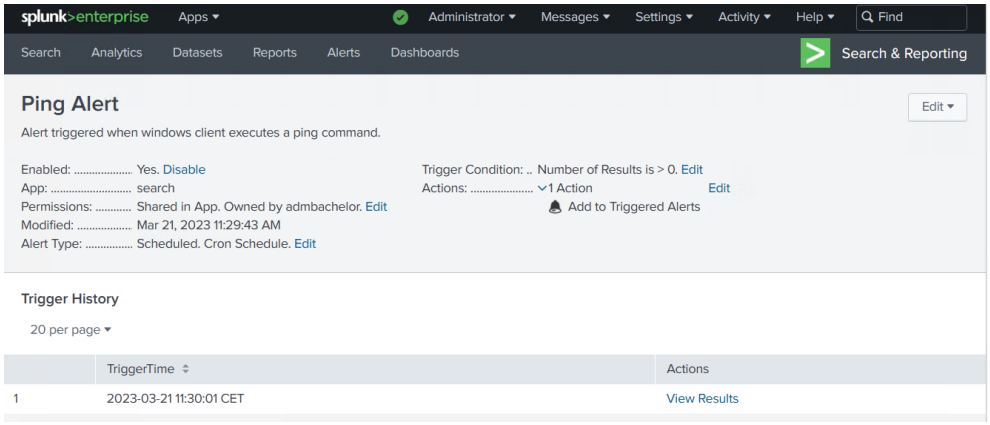
Til slutt konfigurerer vi når alarmen skal utløse, og hva som skal skje når alarmen utløses. I dette tilfellet velger vi at alarmen skal utløses så lenge antall resultater fra søket er større enn 0, altså så fort den får ett eller flere resultat, og at det skal genereres én alarm per resultat. Ved å aktivere *Throttle* kan man bestemme hvorvidt nye alarmer ikke skal utløses ved nye resultat i en gitt periode etter at en alarm er utløst, noe vi ikke ønsker å aktivere i dette tilfelle. Når denne alarmen utløses velger vi foreløpig at den kun skal loggføres som en utløst alarm i Splunk Enterprise uten videre handlinger.

En oversikt over alarmen er tilgjengelig ved å gå til appen `Search & Reporting`, velge `Alerts` i menyen på toppen, og deretter velge ønsket alarm. Når en alarm utløses vil den vises i oversikten her, som vist i figur D.23.

I tillegg til å sende alarmer til Splunk SOAR, som vi diskuterer i seksjon D.4, kan vi eksempelvis velge at resultater som utløser alarmen skal eksporteres til et

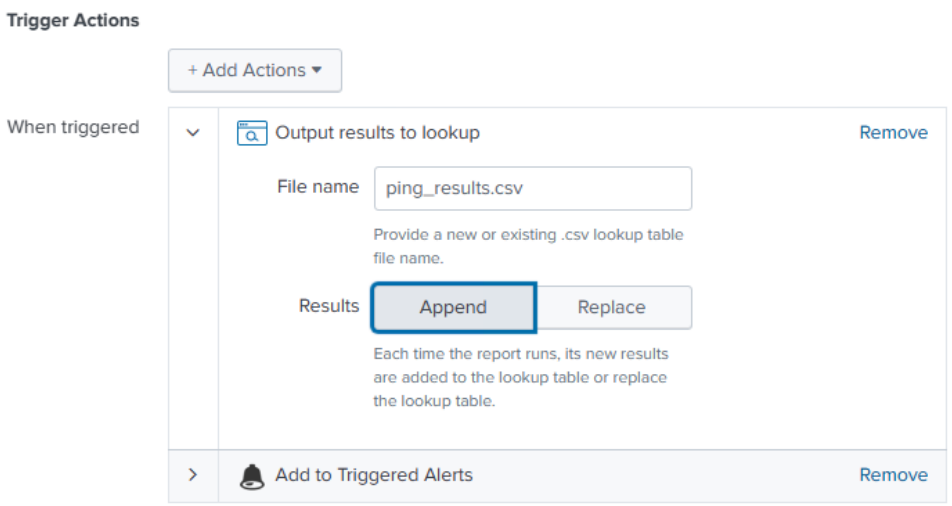


Figur D.22: Generering av alarm



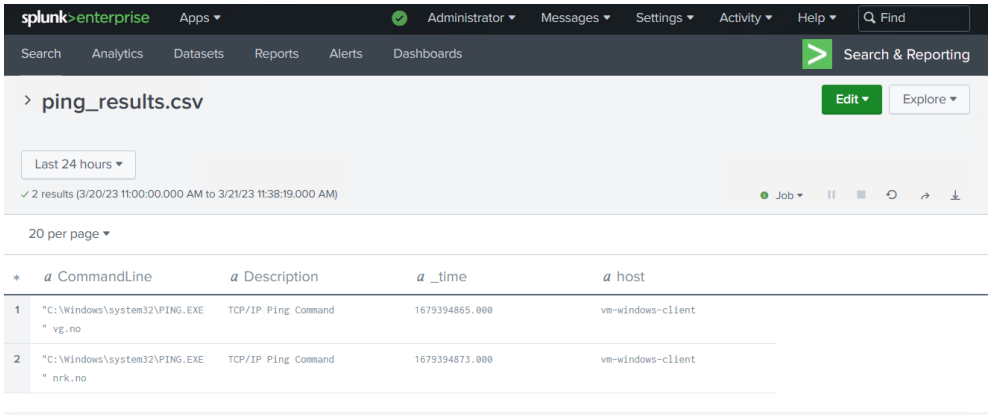
Figur D.23: Oversikt over generert alarm

datasett der vi kan ta vare på alle resultater for senere søk og rapportering. Ved å trykke `Edit` på alarmens `Actions` i figur D.23, kan vi velge `Add Actions` og `Output results to lookup` som vist i figur D.24. Her skriver vi inn navnet på en ny eller eksisterende `csv`-fil, i dette tilfelle ny, og velger hvorvidt resultatene skal legges til som nye linjer i filen, eller erstatte innholdet i filen. Vi velger at nye resultater skal legges til som nye linjer.



Figur D.24: Legge til CSV-eksportering til eksisterende alarm

Datasett er tilgjengelig i Splunk Enterprise fra appen `Search & Reporting` og menyvalget `Datasets`, der man kan søke opp og finne eksisterende datasett eller lage nye. Ved å åpne datasett kan man se alle resultatene lagret i dem, som vist i figur D.25, og man kan også gjøre andre handlinger som å dele eller eksportere datasett.

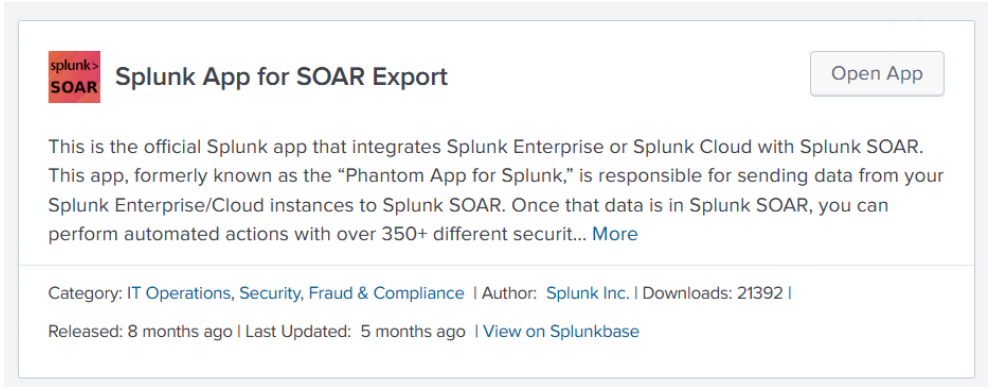


Figur D.25: Vise CSV fra Splunk datasett

D.4 Sende alarmer til Splunk SOAR

I seksjon D.3 viste vi hvordan alarmer kan genereres i Splunk Enterprise, samt noen eksempler på handlinger ved utløsing av alarm. I denne oppgaven vil vi fokusere på integrasjonen mellom alarmgenerering i Splunk og håndtering av slike alarmer og hendelser i SOAR-verktøy, der Splunk SOAR er valgt for denne oppgaven. Alarmer generert i Splunk Enterprise kan konfigureres slik at de blir sendt til SOAR-verktøy automatisk.

For å sende alarmer til Splunk SOAR må vi først installere en tilleggapplikasjon i Splunk Enterprise kalt *Splunk App for SOAR Export* som vist i figur D.26. Dette tillegget gjør det mulig å sende alarmer direkte mellom Splunk Enterprise og Splunk SOAR. Dette tillegget er utviklet for Splunk SOAR spesifikt, men det finnes også tillegg utviklet av andre tredjepartsleverandører av SOAR-verktøy som muliggjør bruk av disse verktøyene sammen med Splunk Enterprise.



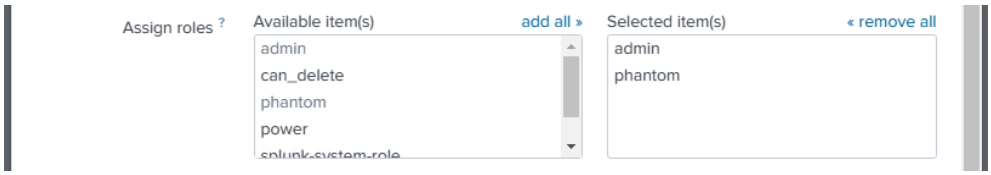
Figur D.26: Installasjon av tillegg for eksportering til SOAR i Splunk Enterprise

Deretter må vi gjøre en konfigurasjonsendring i filen `/opt/enterprise-disk/splunk/etc/system/local/authorize.conf` på Linux-maskinen, der vi legger til

phantom-rollen (som stammer fra `Splunk App for SOAR Export`) som en tildelbar rolle for administratorbrukere, som vist i figur D.27. Etter restart av Splunk Enterprise kan vi da gå til `Settings > Users`, velge administratorbrukeren vår, og legge til *phantom*-rollen som vist i figur D.28. Dette gir administratorbrukeren vår tilgangene som er nødvendig for å sette opp tilkobling fra Splunk Enterprise til Splunk SOAR.

```
admbachelor@vm-soc-bachelor-splunk: /opt/enterprise-disk/splunk/etc/system/local $  sudo vim authorize.conf
admbachelor@vm-soc-bachelor-splunk: /opt/enterprise-disk/splunk/etc/system/local $  sudo cat authorize.conf
[role_admin]
edit_log_alert_event = disabled
grantableRoles = admin;phantom
importRoles = can_delete;power;user
phantom_read = enabled
search_process_config_refresh = disabled
srchIndexesDefault = main
srchMaxTime = 8640000
admbachelor@vm-soc-bachelor-splunk: /opt/enterprise-disk/splunk/etc/system/local $
```

Figur D.27: Tilganger til Splunk SOAR i Splunk Enterprise [1/2]



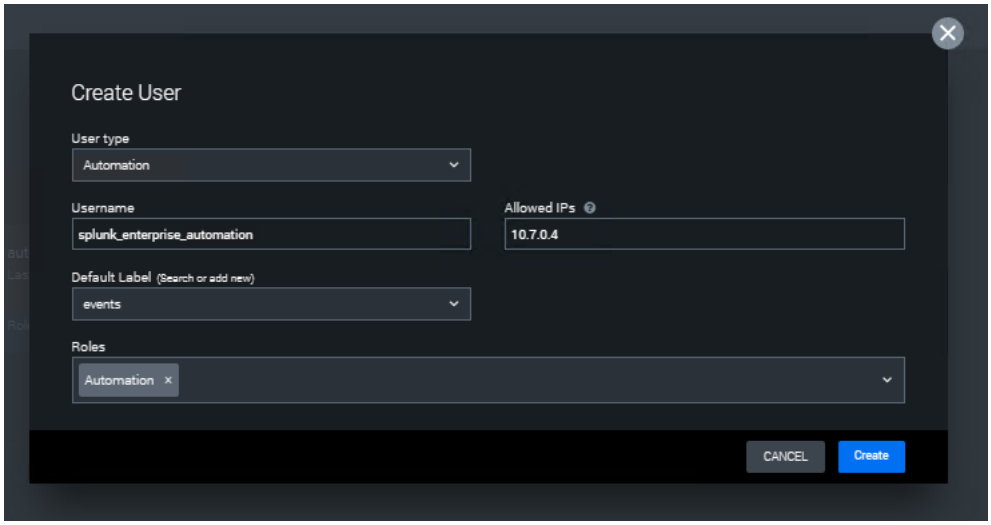
Figur D.28: Tilganger til Splunk SOAR i Splunk Enterprise [2/2]

I tillegg må SSL-sertifikatet til Splunk SOAR-tjenesten legges til som et godkjent sertifikat i Splunk Enterprise. Sertifikatet til SOAR-tjenesten hentes ut fra filen `/opt/soar-disk/splunk-soar/etc/ssl/certs/httpd_cert.crt`, og deretter opprettes filen `/opt/enterprise-disk/splunk/etc/apps/phantom/local/cert_bundle.pem` der sertifikatet limes inn. Da vil det være mulig å opprette en trygg kommunikasjonskanal der alarmer og annen data fra Splunk Enterprise kan sendes til Splunk SOAR.

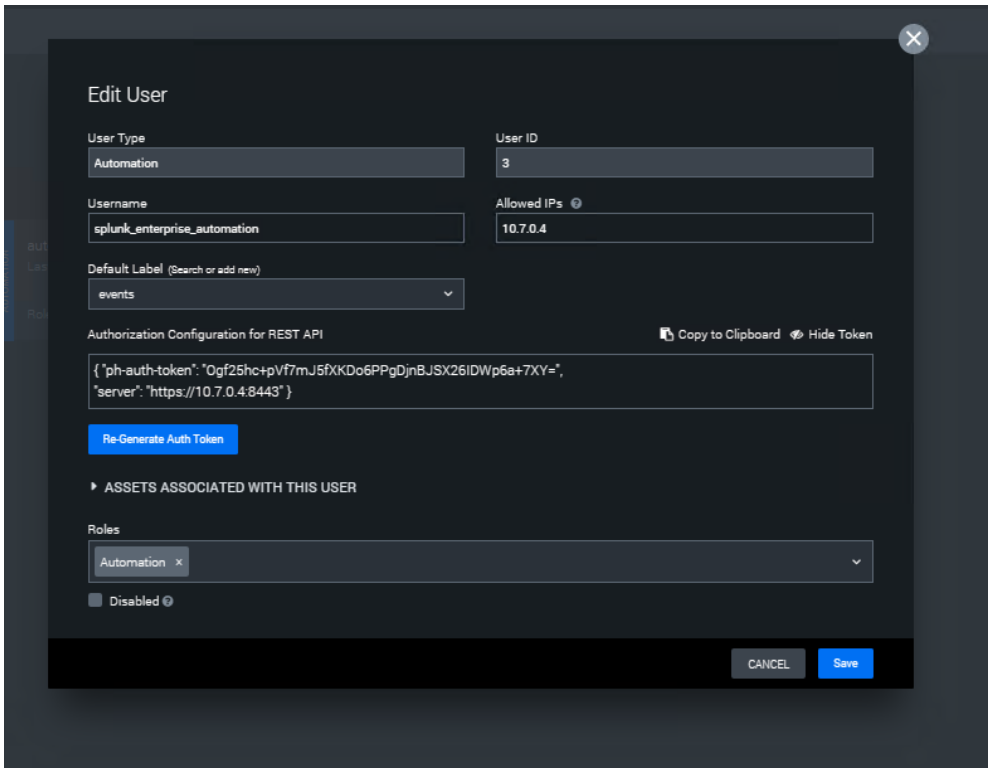
For å opprette denne tilkoblingen må vi først gå til nettgrensesnittet til Splunk SOAR og opprette det Splunk kaller en automatiseringsbruker. Dette gjør vi ved å gå til `Administration > User Management > Users` og deretter velge `+User`. I menyen som dukker opp velges brukertype `Automation`, et brukernavn, og hvilke IP-adresser som er tillatt, som vist i figur D.29. Deretter må vi velge brukeren og trykke `Edit`, som åpner menyen vist i figur D.30, der vi kan kopiere strengen under *Authorization Configuration for REST API*.

Deretter, i Splunk Enterprise, velger vi appen `Splunk App for SOAR Export` som vi installerte tidligere, går til `Configurations`, og trykker `Create Server`. I menyen som dukker opp i figur D.31 limer vi inn strengen vi kopierte, og gir serveren ett navn før vi lagrer. Gjennom grensesnittet er det også mulig å verifisere at tilkoblingen er aktiv og at kommunikasjon mellom verktøyene fungerer.

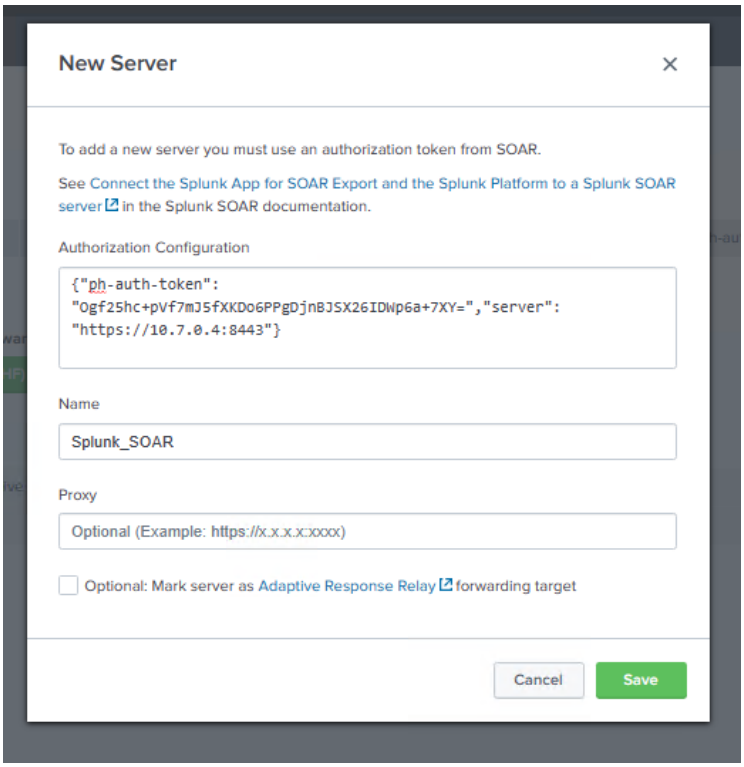
Når tilkoblingen mellom Splunk Enterprise og Splunk SOAR er satt opp, vil *Send to SOAR* bli tilgjengelig som et alternativ for hva som skal skje når en alarm



Figur D.29: Oppsett av automatiseringsbruker i Splunk SOAR [1/2]

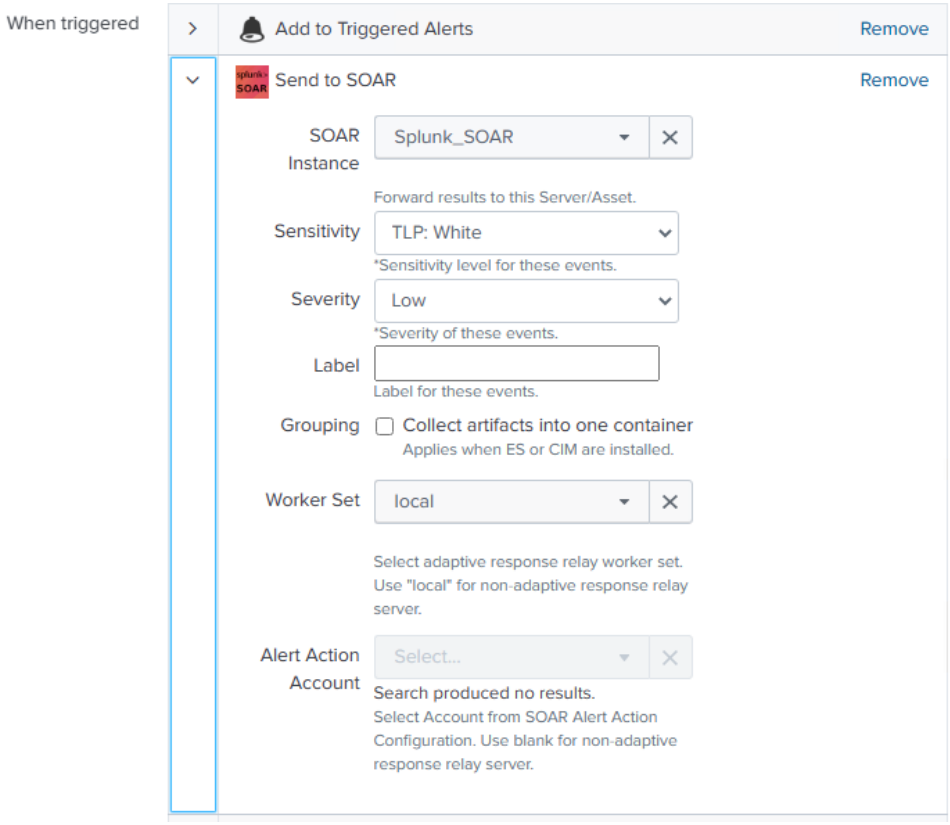


Figur D.30: Oppsett av automatiseringsbruker i Splunk SOAR [2/2]



Figur D.31: Sette opp tilkobling mellom Splunk Enterprise og Splunk SOAR

utløses. Vi redigerer vår eksisterende ping-alarm fra seksjon D.3 som vist i figur D.32, der vi velger SOAR-instansen vi satt opp i forrige steg, samt sensitiviteten og alvorlighetsgraden til alarmen. I noen Splunk miljø vil man kunne ønske å benytte noe som kalles *adaptive response relay worker set*[4], men i denne oppgaven brujer vi `local` som *Worker Set*. Neste gang alarmen utløses vil den nå sendes til Splunk SOAR.

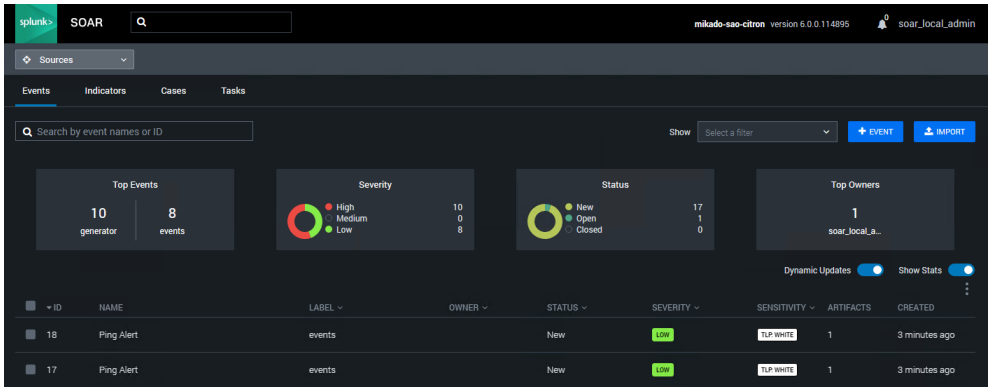


Figur D.32: Sette opp videresending av alarmer til Splunk SOAR

For å demonstrere denne funksjonaliteten starter vi to nye ping fra Windows-klienten i testmiljøet. Handlingene loggføres av Sysmon og sendes til Splunk Enterprise gjennom Splunk Universal Forwarder-agenten på maskinen. Ved neste søk fører dette til to resultat som generer hver sin alarm som sendes til Splunk SOAR. I figur D.33 ser vi våre to alarmer i nettgrensesnittet til Splunk SOAR, under `Sources` i nedtrekksmenyen.

I Splunk SOAR refereres det til disse alarmene som events, selvom det i dette tilfellet og gjennom denne rapporten er mer riktig å kalle disse for alarmer basert på events som allerede er detektert i Splunk Enterprise. Likevel vil vi videre referere til disse alarmene når mottatt i Splunk SOAR som events, da en

[4] `https://docs.splunk.com/Documentation/SOARExport/latest/UserGuide/Adaptiveresponserelay`



Figur D.33: Events mottatt i Splunk SOAR

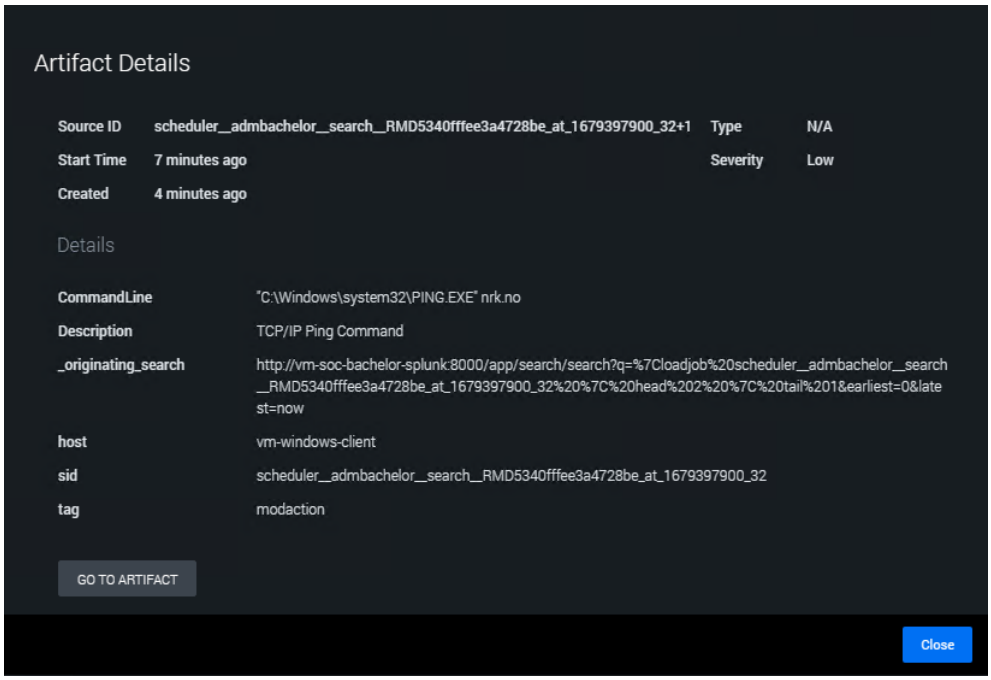
ny analyse og behandling av disse vil gjennomføres av SOAR-verktøyet, og flere alarmer i mange tilfeller vil kunne regnes som mindre events av varierende alvorlighetsgrad som sammen kan underbygge større hendelser. I et produksjonsmiljø vil det også være naturlig at mange forskjellige events, fra forskjellige datakilder og med varierende grad av tidligere behandling og analyse, vil mates inn til et slikt SOAR-verktøy. I sum er det derfor naturlig å kalle alle disse for forskjellige events fra SOAR-verkøyet sitt perspektiv.

Blant informasjonen vi har lett tilgjengelig i SOAR-grensesnittet er navnet på eventen, i dette tilfellet *Ping Alert*-alarmen vi satt opp, sensitiviteten og alvorlighetsgraden vi konfigurerte i forrige steg. Dette bidrar til bedre oversikt for en analytiker som skal behandle innkommende events, og som et hjelpemiddel for automatiserte prosesser. Vi kan også se status på eventen, samt hvor mange artefakter eventen består av, som i dette tilfelle er én.

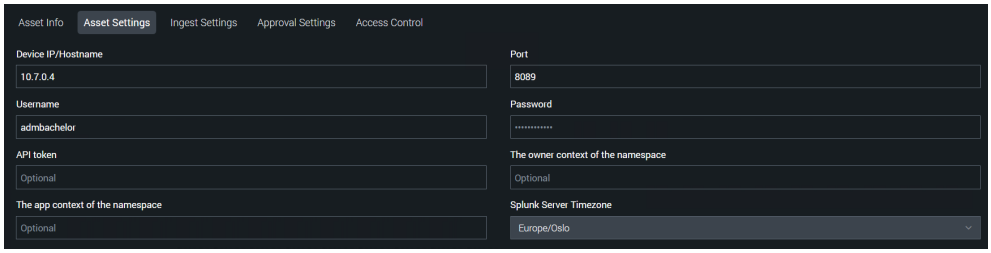
Hvis vi åpner eventen og ser nærmere på denne ene artefakten, ser vi i figur D.34 at det er alarmen som ble sendt fra Splunk Enterprise med det originale søkeresultatet. Fra SOAR-verktøyet sitt perspektiv er altså den originale alarmen fra Splunk Enterprise med søkeresultatet kun én del - én artefakt - av eventen, og det vil for eksempel være mulig å berike eventen med flere artefakter eller annen tilleggsinformasjon, eller slå sammen flere events til en større hendelse.

D.5 Utføre søk i Splunk Enterprise fra Splunk SOAR

For å gjennomføre søk i Splunk Enterprise direkte fra Splunk SOAR kreves tillegget *Splunk* installert gjennom nettgrensesnittet til Splunk SOAR. Deretter går vi til listen over *Unconfigured Apps*, lokaliserer tillegget og trykker *Configure New Asset*. Under *Asset Settings* i figur D.35 konfigurerer vi IP-adressen og administratorportnummeret til vår instanse av Splunk Enterprise, `10.7.0.4` og `8089`, samt brukernavnet og passordet til vår administratorbruker. Resten av innstillingene lar vi i vårt tilfelle stå som standard. Når innstillingene er lagret kan vi teste og verifisere at tilkoblingen er aktiv ved å trykke på *Test Connectivity* som vist i figur D.36.



Figur D.34: Eventinformasjon i Splunk SOAR

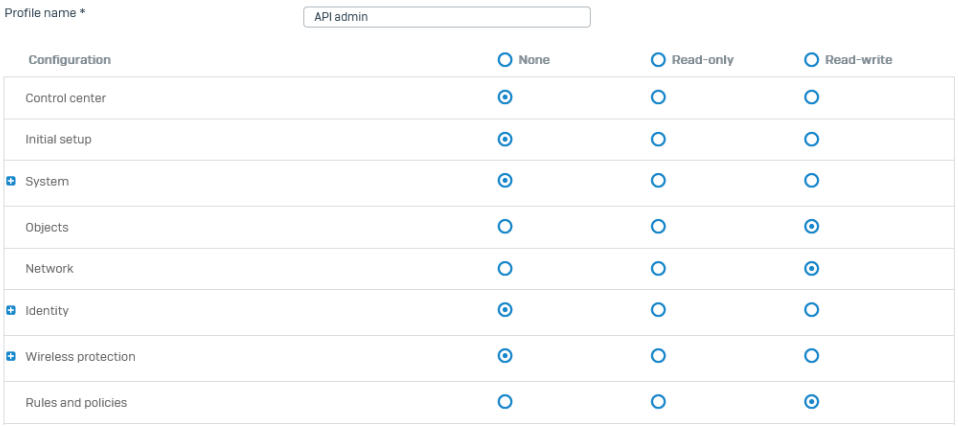


Figur D.35: Konfigurasjon av Splunk-tillegget i Splunk SOAR

Figur D.36: Test av tilkobling mellom Splunk SOAR og Splunk Enterprise

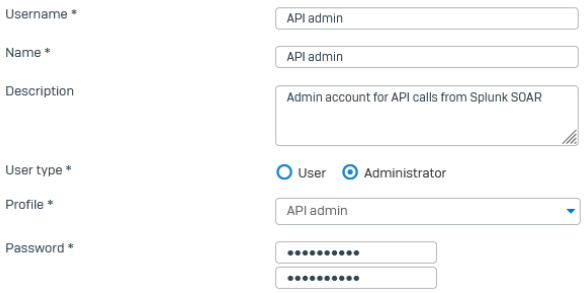
D.6 Administrere brannmur fra Splunk SOAR

For å kunne administrere Sophos Firewall fra Splunk SOAR må vi først konfigurere og aktivere API i Sophos Firewall. I administratorgrensesnittet til Sophos Firewall går vi til `System > Profiles > Device access`, og velger `Add` for å lage en ny profil. I menyen som dukker opp i figur D.37 skriver vi inn et profilnavn, *API admin*, og velger hvilken tilganger profilen skal ha. Ettersom dette er en administratorprofil som skal brukes over API, aktiverer vi lese- og skriverettigheter for objekter, nettverk, regler og retningslinjer. Profilen vil ikke ha tilgang til å lese eller endre på eksempelvis system- og identitetsinnstillinger for brannmuren.

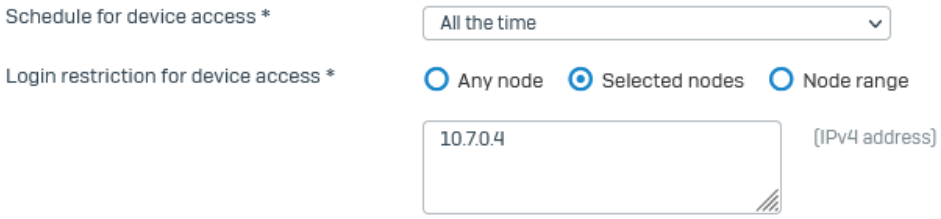


Figur D.37: Oppsett av administratorprofil for API i Sophos Firewall

Deretter kan vi opprette en brukerkonto i Sophos Firewall som skal få denne profilen (rollen). Vi går til `Configure > Authentication > Users` og velger `Add` for å opprette en ny brukerkonto. Som vist i figur D.38 gir vi brukeren et brukernavn og passord, setter brukertype til administrator og tilegner brukeren den profilen vi lagde i forrige steg. I tillegg bestemmer vi at brukeren kan benyttes hele døgnet, men at bruk begrenses til IP-adressen til vår maskin som kjører Splunk SOAR (`10.7.0.4`), vist i figur D.39.

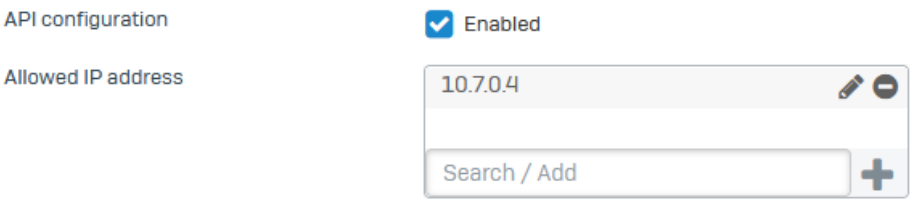


Figur D.38: Oppsett av administratorbruker for API i Sophos Firewall [1/2]



Figur D.39: Oppsett av administratorbruker for API i Sophos Firewall [2/2]

Til slutt må vi aktivere bruken av API i Sophos Firewall. Dette gjør vi fra `System > Backup & firmware > API`. Vi sjekker boksen for å aktivere API-konfigurasjon, og i tillegg legger vi igjen inn IP-adressen til Splunk SOAR som tillatt IP-adresse for bruk av API, vist i figur D.40. IP-adressen som skal benytte API-administrasjon må altså både tillates for API-bruk generelt, i tillegg til å tillates for en brukerkonto med administratorrettigheter.



Figur D.40: Aktivere API-konfigurasjon i Sophos Firewall

På dette tidspunktet er API-konfigurasjonen på Sophos Firewall ferdig, og maskinen som kjører Splunk SOAR skal være klar for å benytte API-et. Vi kobler oss til terminalvinduet til Linux-maskinen som kjører Splunk SOAR og tester tilkoblingen med følgende API kall over HTTPS til brannmurens grensesnitt (adresse *10.0.3.4:4444*), som vist i figur D.41.

```
curl "https://10.0.3.4:4444/webconsole/APIController?reqxml=<Request
    ><Login><Username>API admin</Username><Password>7tE4s8)TTV</
    Password></Login><Get><IPHost><Name></Name><IPFamily></IPFamily
    ><HostType></HostType><IPAddress></IPAddress></IPHost></Get></
    Request>" --insecure
```

```
admbachelor@vm-soc-bachelor-splunk: ~ $  curl "https://10.0.3.4:4444/webconsole/APIController?reqxml=%3CRequest%3E%3CLog
in%3E%3CUsername%3EAPI%20admin%3C/Username%3E%3CPassword%3E7tE4s8)TTV%3C/Password%3E%3C/Login%3E%3CGet%3E%3CIPHost%3E%3C
Name%3E%3C/Name%3E%3CIPFamily%3E%3C/IPFamily%3E%3CHostType%3E%3C/HostType%3E%3CIPAddress%3E%3C/IPAddress%3E%3C/IPHost%3E
%3C/Get%3E%3C/Request%3E" --insecure
<?xml version="1.0" encoding="UTF-8"?>
<Response APIVersion="1905.1" IPS_CAT_VER="0">
  <Login>
    <status>Authentication Successful</status>
  </Login>
```

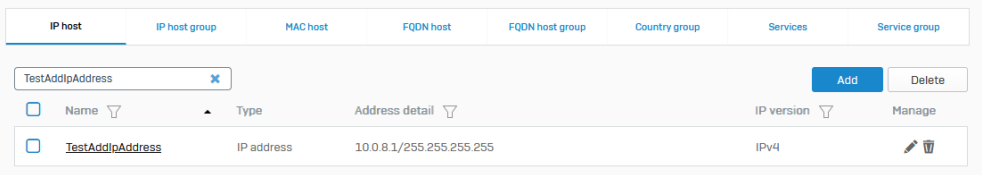
Figur D.41: Test av tilkobling mellom Splunk SOAR og Sophos Firewall over API

Dette er en enkel forespørsel som lister ut konfigurerte IP-hoster i brannmuren. Merk at i figuren er en del spesialtegn erstattet med URL-kodete tegn (tegn som starter med «%»), ettersom `curl`-programmet ikke fungerer med disse spesialtegnene i forespørselen. Som vist i figuren er autentiseringen vellykket, resten av resultatet er ikke vist.

Denne typen API-kall er den enkleste måten å sende forespørsler til API-et, men det er ikke den anbefalte måten for produksjonsmiljø, ettersom den blant annet krever at brukernavnet og passordet til administratorbrukeren sendes ved i klartekst som en del av URL-en. Istedenfor er det anbefalt å benytte kall med autorisasjons-tokens[5]. En annen ting som er verdt å merke er at flagget `--insecure` (samme som `-k`) må benyttes. Dette er fordi Sophos Firewall i vårt testmiljø benytter et selvsignert sertifikat som ikke er utstedt av en sertifikasjonsautoritet eller lagt inn som et godkjent sertifikat på Linux-maskinen. I slike tilfeller vil programvare slik som `curl` av sikkerhetsmessige årsaker blokkere kommunikasjon over HTTPS med mindre `--insecure` eller `-k` benyttes. I et produksjonsmiljø ville man ha lagt inn et sertifikat utstedt av en sertifikasjonsautoritet eller konfigurert det selv-signerte sertifikatet som godkjent på enhetene som skal kommunisere mot brannmuren.

Vi kan også legge til IP-adresser i brannmurens IP-register med følgende forespørsel, som legger til adressen `10.0.8.1` under navnet *TestAddIpAddress*. Figur D.42 viser at IP-adressen dukker opp i administratorgrensesnittet til brannmuren.

```
curl "https://10.0.3.4:4444/webconsole/APIController?
    SecureStorageMasterKey=Passw0rd@12345&reqxml=<Request><Login><
    Username>API admin</Username><Password>7tE4s8)TTV</Password></
    Login><Set><IPHost><Name>TestAddIpAddress</Name><IPFamily>IPv4</
    IPFamily><HostType>IP</HostType><IPAddress>10.0.8.1</IPAddress
    ></IPHost></Set></Request>" --insecure
```



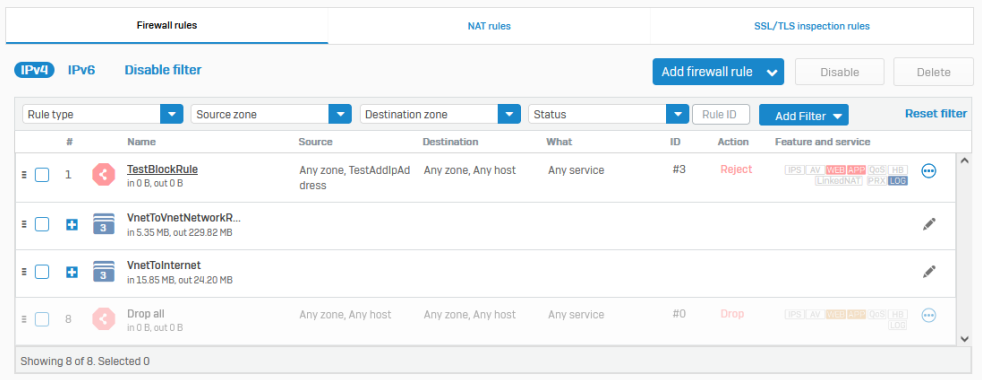
Figur D.42: Legge til IP-adresser i adresseregisteret til Sophos Firewall med API

Det er også mulig å lage nettverksregler i brannmuren over Sophos sitt API, som er svært nyttig for å utføre automatiske handlinger fra SOAR-verktøyet. Følgende forespørsel lager en regel med navn *TestBlockRule* som skal blokkere all trafikk fra IP-adresseobjektet vi lagde i forrige kall, *TestAddIpAddress* (`10.0.8.1`). Figur D.43 viser at regelen legges til i listen over nettverksregler i Sophos Firewall.

```
curl "https://10.0.3.4:4444/webconsole/APIController?
    SecureStorageMasterKey=Passw0rd@12345&reqxml=<Request><Login><
```

[5] `https://developer.sophos.com/firewall-management`

```
Username>API admin</Username><Password>7tE4s8)TTV</Password></
Login><Set operation='add'><FirewallRule transactionid=""><Name>
TestBlockRule</Name><Description>Test</Description><IPFamily>
IPv4</IPFamily><Status>Enable</Status><Position>Top</Position><
PolicyType>Network</PolicyType><NetworkPolicy><Action>Reject</
Action><LogTraffic>Enable</LogTraffic><SkipLocalDestined>Disable
</SkipLocalDestined><Schedule>All The Time</Schedule><
SourceNetworks><Network>TestAddIpAddress</Network></
SourceNetworks></NetworkPolicy></FirewallRule></Set></Request>"
--insecure
```

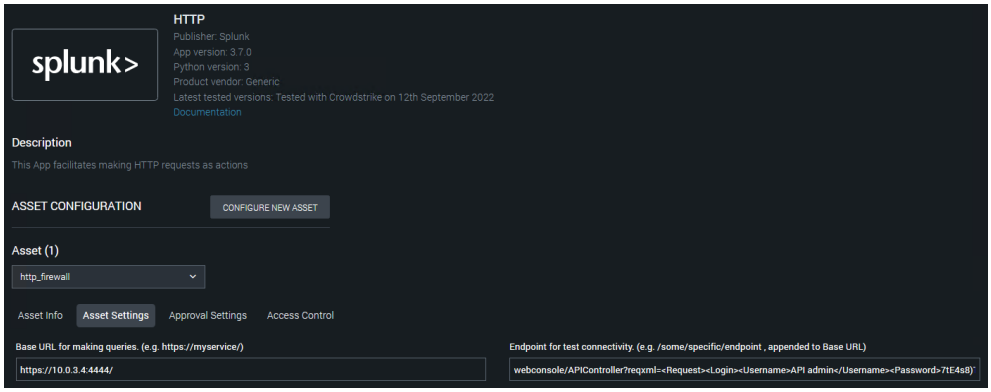


Figur D.43: Lage nettverksregler i Sophos Firewall med API

Vi ønsker å kunne gjennomføre slike API-kall mot brannmuren fra Splunk SOAR. For å oppnå dette installerer vi tillegget *HTTP* i Splunk SOAR, går til listen over *Unconfigured Apps*, lokaliserer tillegget og trykker *Configure New Asset*. Vi gir konfigurasjonen navnet *http_firewall* (et tillegg kan ha flere forskjellige konfigurasjoner, og denne konfigurasjonen er i vårt tilfelle spesifikk for kall mot brannmuren), og under *Asset Settings* konfigurerer vi URL-en som ligger i bunn for alle API-kall som *Base URL*, og vi legger inn ett enkelt kall som benyttes for å teste tilkoblingen, som vist i figur D.44. Nå er tillegget klart til å benyttes i playbooks i Splunk SOAR for å gjennomføre API-kall mot brannmuren.

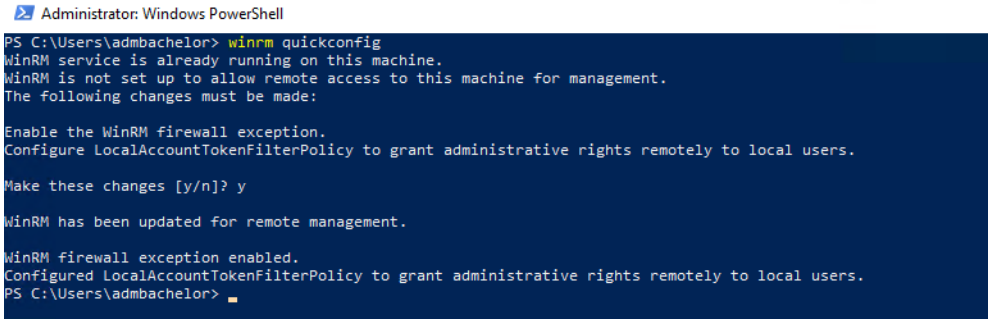
D.7 Administrere endepunkter fra Splunk SOAR

For å muliggjøre endepunktsadministrasjon installerer vi Windows Remote Management på klienten. Programvaren bak WinRM-tjenesten skal allerede være installert på alle nyere versjoner av Windows og Windows Server, og ingen ytterligere nedlasting er nødvendig. For å raskt aktivere WinRM-tjenesten for å tillate ekstern administrasjon kan følgende kommando brukes, som vist i figur D.45. Dette vil aktivere Windows Remote Management med lytting etter instrukser og kommandoer på port 5985 (med mindre annet spesifiseres).



Figur D.44: Konfigurere tillegg for API-kall mot brannmur fra Splunk SOAR

```
winrm quickconfig
```



Figur D.45: Oppsett av Windows Remote Management på klient [1/2]

Windows Remote Management støtter flere typer autentisering mellom administrasjonsserveren, i vårt tilfelle Splunk SOAR, og enhetene som skal administreres[6]. For brukstestene i denne oppgaven benytter vi oss av såkalt *Basic*-autentisering, der administrasjonsserveren autentiserer seg ved hjelp av brukernavn og passord til en administratorbruker hos enheten som skal administreres. Dette er den enkleste, men samtidig minst sikre formen for autentisering, og den er derfor ikke anbefalt i produksjonsmiljø. Ettersom sikkert oppsett og konfigurasjon av endepunktsadministrasjon er utenfor omfanget av denne oppgaven, holder vi oss til *Basic*-autentisering i brukstestene. Følgende kommandoer må gis på klienten for å tillate denne typen autentisering:

```
winrm set winrm/config/client/auth '@{Basic="true"}'
winrm set winrm/config/service/auth '@{Basic="true"}'
winrm set winrm/config/service @{AllowUnencrypted="true"}
```

For å verifisere at Windows Remote Management er aktivert og lytter på klienten, benyttes følgende kommando som vist i figur D.46:

[6] `https://learn.microsoft.com/en-us/windows/win32/winrm/authentication-for-remote-connections`

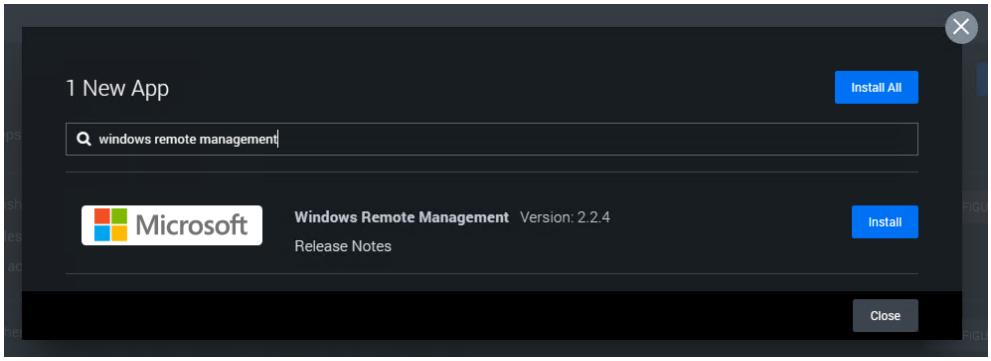
```
winrm enumerate winrm/config/listener
```

```
PS C:\Users\admbachelor> winrm enumerate winrm/config/listener
Listener
    Address = *
    Transport = HTTP
    Port = 5985
    Hostname
    Enabled = true
    URLPrefix = wsman
    CertificateThumbprint
    ListeningOn = 10.50.0.4, 127.0.0.1, ::1, fe80::f629:c8fa:e603:3f36%5
```

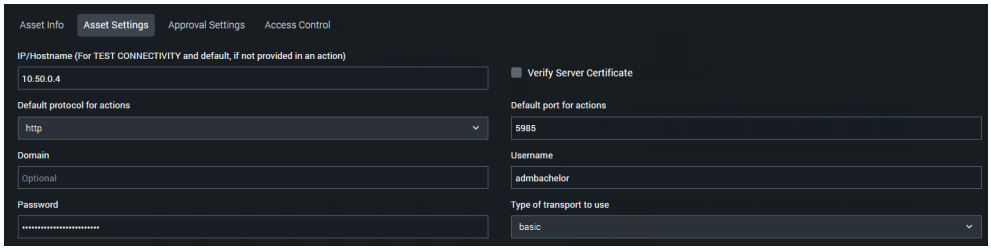
Figur D.46: Oppsett av Windows Remote Management på klient [2/2]

På dette tidspunktet er oppsettet på klienten ferdig, og resterende oppsett gjennomføres i Splunk SOAR. Først installerer vi tillegget *Windows Remote Management*, som vist i figur D.47. Deretter går vi til listen over *Unconfigured Apps*, lokaliserer tillegget og trykker *Configure New Asset*. Under *Asset Settings* i figur D.48 må vi konfigurere portnummer, som i vårt tilfelle er `5985`, brukernavn og passord til den lokale administratorbrukeren, samt velge transporttype (autentiseringstype) `basic`. I tillegg er det mulig, men ikke påkrevd, å konfigurere en IP-adresse i denne menyen, men dette er kun brukt for å teste tilkoblingen som vist i figur D.49.

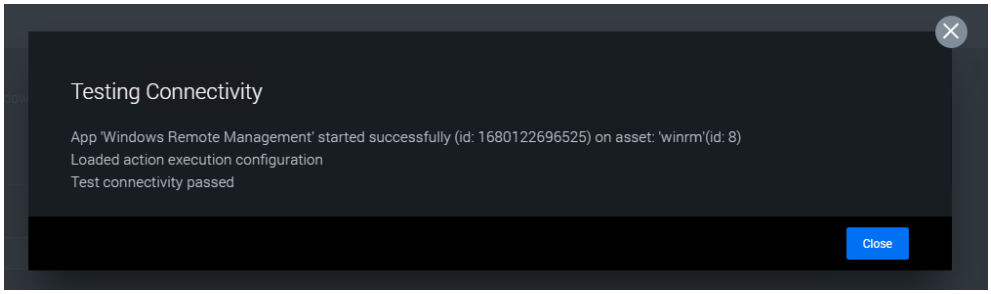
Som vi vil vise i seksjon D.8, legges IP-adressen til enheten som skal administreres direkte til i playbooks (typisk gjennom en variabel som henter ut IP-adressen til den spesifikke enheten som har utløst en event), slik at tillegget ikke er begrenset til å kun fungere med én enkelt enhet. Det er derimot verdt å merke seg at brukernavnet og passordet til den lokale administratorbrukeren må være det samme på tvers av alle enheter som administreres på denne måten, ettersom dette konfigureres direkte i tilleggets innstillinger. Dette er naturligvis en sentral begrensning ved denne typen autentisering, og ved endepunktsadministrasjon direkte fra Splunk SOAR, da det kreves en felles administratorbruker som eksisterer hos alle endepunkter.



Figur D.47: Installering av Windows Remote Management app i Splunk SOAR



Figur D.48: Konfigurasjon av Windows Remote Management app i Splunk SOAR [1/2]



Figur D.49: Konfigurasjon av Windows Remote Management app i Splunk SOAR [2/2]

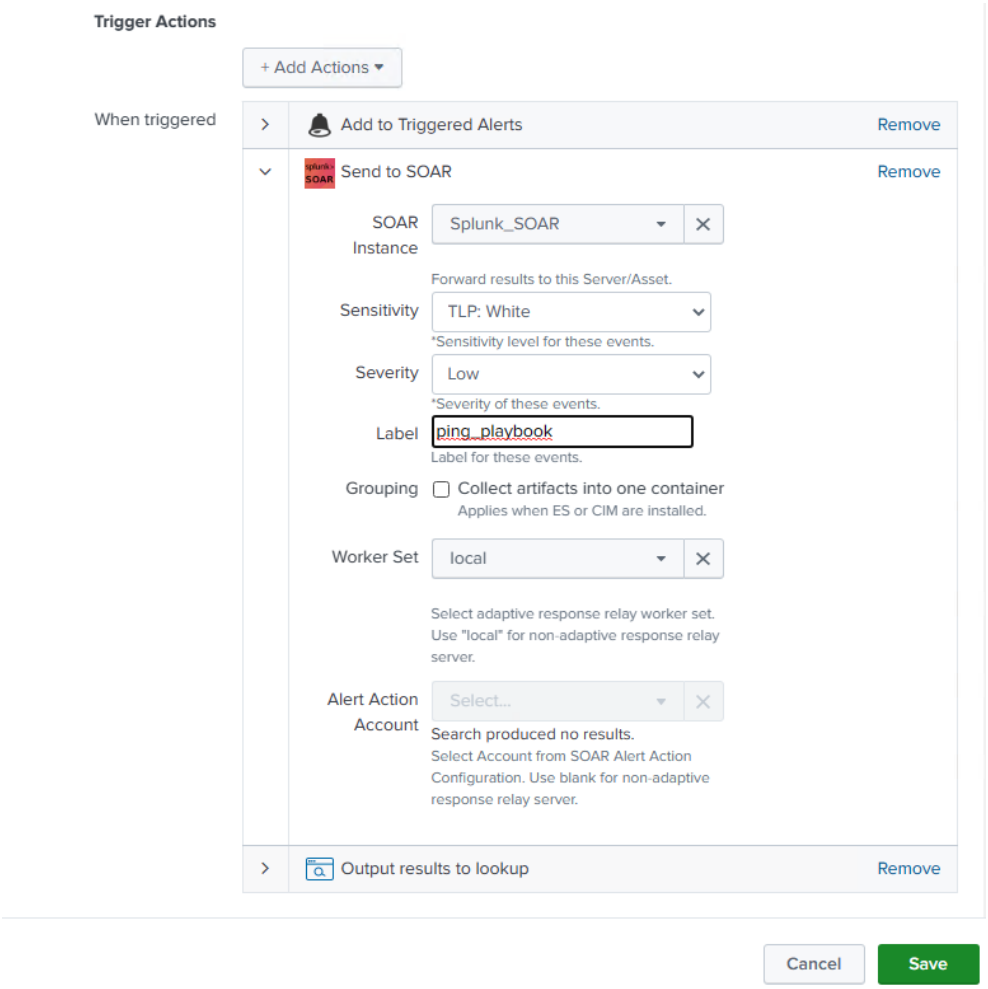
D.8 Lage playbooks i Splunk SOAR

Det første steget for å lage playbooks som utløses automatisk basert på kjente alarmer, er å definere etiketter som blir sendt ved alarmene som kommer fra Splunk Enterprise. Med utgangspunkt i samme alarm som vi lagde i seksjon D.3, og som vi satt opp videresending av til Splunk SOAR i seksjon D.4, kan vi redigere alarmen i Splunk Enterprise ved å legge til etiketten `ping_playbook` som vist i figur D.50.

Hittil er denne etiketten kun definert i alarmen fra Splunk Enterprise, og Splunk SOAR vet ikke hva den skal gjøre med events som har denne etiketten. Neste steg er å definere samme etikett i Splunk SOAR ved å gå til menyvalget `Administration > Label Settings`, og deretter trykke `+Label`. Da får vi opp en tekstboks der vi kan skrive inn navnet på etiketten og definere denne, som vist i figur D.51.

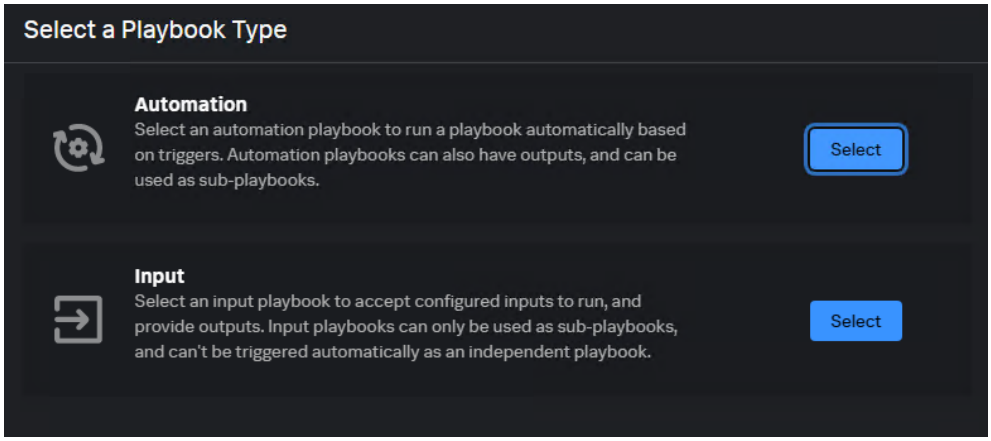
På dette tidspunktet kan vi lage en playbook som skal kobles til denne etiketten. Dette gjør vi ved å gå til menyvalget `Playbooks` og deretter trykke `+Playbook`. Dette gir oss to valg som vist i figur D.52: `Automation` eller `Input`. Som det står beskrevet i figuren så er det kun automatiseringsplaybooks som kan utløses automatisk og fungere uavhengig av andre playbooks. Input-playbooks kan kun legges til som en underdel av en overordnet automatiseringsplaybook.

Playbooks i Splunk SOAR består av et startfelt, et sluttfelt, og i mellom disse en rekke sammenkoblede blokker og prosesser som redegjør for hvordan en



Figur D.50: Legge til etikett på event i Splunk Enterprise for utløsing av playbook i Splunk SOAR

Figur D.51: Lage etikett i Splunk SOAR



Figur D.52: Lage Playbook i Splunk SOAR

hendelse skal håndteres. Det er tre hovedkategorier av blokker som kan legges til i en playbook: handlinger (eng: *execute actions*), prosessfiltre (eng: *process filters*) og menneskelig input (eng: *human input*). Innunder disse finner vi følgende underblokker:

D.8.1 Handlinger

Handling

En handlingsblokk[7] (eng: *action*) utfører en handling fra et installert og konfigurert tillegg i Splunk SOAR. Slike handlinger inkluderer søk mot Splunk Enterprise og administrasjon av endepunkter med Windows Remote Management (to tillegg vi allerede har diskutert tidligere), samt andre eksempler som oppslag av IP-adresser med WHOIS[8] eller MaxMind[9], eller analyse av mistenksomme filer eller nettadresser med VirusTotal[10]. Mye av styrken til Splunk SOAR og andre SOAR-verktøy ligger nettopp i den store mengden tillegg og tilhørende handlinger som er tilgjengelig under håndtering av en hendelse. Som inndata kan slike handlinger for eksempel benytte data fra eventens artefakter eller output fra tidligere blokker i playbooken. De fleste handlinger gir en form for output som blir til nye artefakter i eventen, der disse kan ha verdi i seg selv, eller benyttes videre i håndteringen av senere blokker.

[7] `https://docs.splunk.com/Documentation/SOARonprem/6.0.0/Playbook/ActionBlock`
[8] `https://datatracker.ietf.org/doc/html/rfc3912`
[9] `https://www.maxmind.com`
[10] `https://www.virustotal.com`

Playbook

Playbook-blokker[11] integrerer fullstendige underordnede playbooks i en overordnet playbook. Dette gir fleksibilitet under utvikling av playbooks og håndtering av hendelser, ved at etablerte og prøvde playbooks rett og slett kan inkluderes ved behov istedenfor at alle playbooks må definere alle prosesser som behøves på nytt. Ofte vil mange tiltak og prosesser være like under håndteringen av forskjellige hendelser.

Kode

Kodeblokker[12] kan inkluderes i playbooks for å muliggjøre mer avansert og/eller fleksibel behandling av inndata der det ikke finnes tilstrekkelige handlinger fra tillegg eller eksisterende playbooks som oppnår det ønskelige resultatet. Slike kodeblokker benytter Python, og tar inndata fra tidligere blokker og artefakter, og produserer artefakter som kan brukes videre i playbooken.

Verktøy

Et verktøy[13] (eng: *utility*) i Splunk SOAR ligner en del på handlinger, men de krever ikke installasjon av tillegg, og de utfører typisk enklere praktiske handlinger. Det finnes en lang rekke forhåndskonfigurerte verktøy i Splunk SOAR, inkludert opprettelse av forskjellige typer data, enkoding, dekoding, tekstmanipulasjon og mer. Slike verktøy er skrevet i Python, og det er mulig å kode nye verktøy. På mange måter ligner slike verktøy derfor på kodeblokker som beskrevet over, bare at de lagres og kan gjenbrukes i flere playbooks der ønskelig. Under verktøy finner vi også en del forhåndsdefinerte API-kall som samhandler direkte med SOAR-applikasjonen i seg selv, blant annet opprettelse av kommentarer og notater, eller endring av status, sensitivitet eller kritikalitet av eventer og hendelser. Dette er viktige verktøy for automatisering.

D.8.2 Prosessfiltre

Filter

En filtreringsblokk[14] brukes for å definere kriterier for filtrering, slik at kun artefakter som oppfyller gitte betingelser blir sendt videre gjennom filtreringsblokken og til senere deler av playbooken. Dette er spesielt nyttig hvis man har en event med flere artefakter, der ikke alle artefakter er nødvendig for håndteringen av hendelsen, eller der forskjellige artefakter krever forskjellig håndtering. Med filtreringsblokker er det mulig at artefakter samsvarer med flere betingelser og

[11] `https://docs.splunk.com/Documentation/SOARonprem/6.0.0/Playbook/PlaybookBlock`
[12] `https://docs.splunk.com/Documentation/SOARonprem/6.0.0/Playbook/CodeBlock`
[13] `https://docs.splunk.com/Documentation/SOARonprem/6.0.0/Playbook/UtilityBlock`
[14] `https://docs.splunk.com/Documentation/SOARonprem/6.0.0/Playbook/FilterBlock`

dermed sendes videre til flere forskjellige retninger av playbooken, i motsetning til beslutningsblokker.

Beslutning

Beslutningsblokker[15] ligner på filtreringsblokker ved at de også definerer kriterier som bestemmer den videre flyten i en playbook. En beslutningsblokk er bygget opp av *if*, *ifelse* og *else* betingelser, der artefakter måles opp mot betingelsene og sendes videre så fort en betingelse returnerer sann. De to store forskjellene mellom filtreringsblokker og beslutningsblokker er at artefakter kun samsvarer med én av betingelsene (den første som returnerer sann), og at artefakter må samsvare med minst én av betingelsene. Hvis ingen av betingelsene i en beslutningsblokk returnerer sann, vil playbooken feile.

Formatering

En formateringsblokk[16] kan benyttes for å generere egendefinerte tekststrenger (for eksempel kommentarer eller e-poster) basert på tidligere blokker og artefakter. Fremgangsmåten innebærer å først definere én eller flere variabler, og deretter utforme den egendefinerte tekststrengen der variablene inkluderes i teksten.

D.8.3 Menneskelig input

I mange playbooks vil fullstendig automatisering fra start til slutt enten være svært utfordrende, eller rett og slett ikke anbefalt, og derfor er det mulig å bruke blokker som innhenter menneskelig input[17]. Når en slik blokk aktiveres vil en melding sendes til en bruker eller en gruppe, og playbooken vil settes på pause inntil brukeren eller medlemmer av gruppen beslutter hva som skal skje videre. En slik beslutning vil vanligvis innebære å godkjenne eller avvise videre utførelse av en playbook, eller å fylle inn påkrevd informasjon som er nødvendig for playbookens videre flyt.

[15] `https://docs.splunk.com/Documentation/SOARonprem/6.0.0/Playbook/DecisionBlock`
[16] `https://docs.splunk.com/Documentation/SOARonprem/6.0.0/Playbook/FormatBlock`
[17] `https://docs.splunk.com/Documentation/SOARonprem/6.0.0/Playbook/PromptBlock`

Vedlegg E

Caser

Dette vedlegget inkluderer utdypende informasjon om oppsett av caser, søk, alarmer og playbooks.

E.1 Case 1

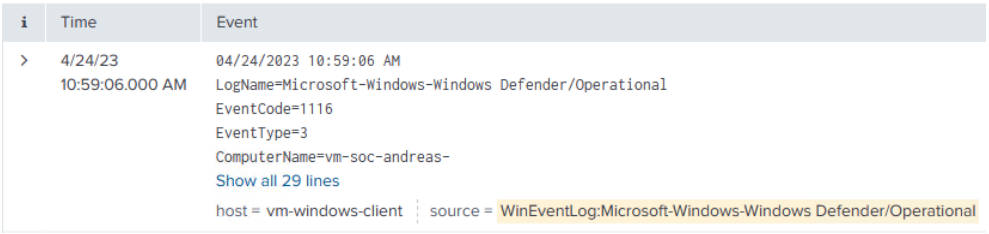
E.1.1 Case 1: Søk og alarm

Som en del av case 1 lagde vi et søk i Splunk Enterprise for å plukke opp logger fra Microsoft Defender som indikerer deteksjon av ondsinnet skadevare på Windows-klienter. Søket henter blant annet ut informasjon om loggenes *Event Code*[1]. Kodene forklarer hva Microsoft Defender har detektert og gjort, for eksempel at skadevaren er detektert, blitt satt i karantene eller slettet. I tillegg henter søket ut *Error Code* og *Error Description*, som forklarer hva som eventuelt har gått galt når Microsoft Defender har håndtert skadevaren. Søket i sin helhet vises under:

```
index=main source="WinEventLog:Microsoft-Windows-Windows Defender/
    Operational" EventCode = "1116"
|  join host [ search EventCode = 1120 source="WinEventLog:Microsoft
    -Windows-Windows Defender/Operational"
    |  fields host, Hashes, Message
    |  rex field=Hashes "(?<Hashes_sha1>\b[0-9a-f]{5,40}\b)"
    |  rename Message AS Hashes_Message]
|  join host [ search EventCode IN (1117,1118,1119) source="
    WinEventLog:Microsoft-Windows-Windows Defender/Operational"
    |  fields host, Action, Action_Status, EventCode, Error_Code,
        Error_description, Message
    |  rename Message AS Action_Message
    |  rename EventCode AS Action_EventCode]
```

[1] https://learn.microsoft.com/en-us/microsoft-365/security/defender-endpoint/troubleshoot-microsoft-defender-antivirus

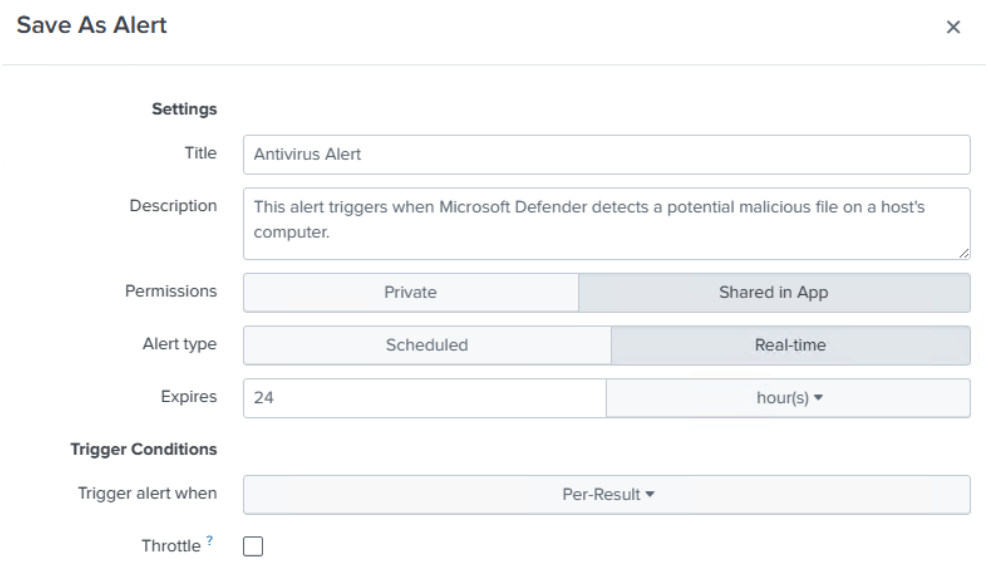
Sanntidssøket består hovedsakelig av tre deler som kobles sammen. Først og fremst finner søket alle logger fra Micrsoft Defender med eventkode `1116`. Dette er koden for «MALWAREPROTECTION_STATE_MALWARE_DETECTED», altså oppdaget skadevare. Deretter gjennomføres et delsøk som finner alle logger fra Microsoft Defender med eventkode `1120`. Dette er koden for «MALWAREPROTECTION_THREAT_HASH», altså skadevaren sin hash. Fra disse treffene henter søket ut kolonnene for maskinnavn, hash og loggbeskrivelse, før det benytter et regulært uttrykk for å omgjøre hash-verdien (fjerne et unødvendig prefix) slik at den kan senere benyttes under analyse. Til slutt gjennomføres enda et delsøk som finner alle logger fra Microsoft Defender med eventkode `1117`, `1118` eller `1119`. Dette er kodene for henholdsvis «MALWAREPROTECTION_STATE_MALWARE_ACTION_TAKEN», «MALWAREPROTECTION_STATE_MALWARE_ACTION_FAILED» og «MALWAREPROTECTION_STATE_MALWARE_ACTION_CRITICALLY_FAILED». Disse kodene gir svar på hvorvidt Microsoft Defender har klart å håndtere skadevaren eller ikke. Fra disse treffene henter søket ut kolonnene for maskinnavn, hvilken handling som er utført, status på handlingen, eventkode, feilkode, feilbeskrivelse og loggbeskrivelse. De tre søkene kobles sammen med kommandoen *join* på *host* (maskinnavn) som felles verdi. Ikke alle kolonner fra alle eventene er hentet ut i det søket slås sammen, for å begrense mengden data og fjerne informasjon som ikke er nødvendig for å håndtere hendelsen. Figur E.1 viser et eksempel på resultat fra dette søket.



Figur E.1: Case 1: Resultat av sammensatt søk som brukes for å generere alarmer

Videre opprettet vi en alarm med navn *Antivirus Alert* som genereres ved treff på dette søket, som ble konfigurert slik at treff videresendes til Splunk SOAR for håndtering. Konfigurasjonen til alarmen vises i figur E.2 og E.3. Her valgte vi at søket skal gjennomføres i sanntid og trigges per resultat. I tillegg konfigurerte vi at alarmen blir sendt videre til Splunk SOAR med etiketten *antivirus*, sensitivitet *TLP:Amber*[2] og kritikalitet *medium*.

[2]For mer informasjon om TLP-systemet for sensitivitetsklassifisering, se `https://www.first.org/tlp/`



Figur E.2: Case 1: Lage Splunk alarm [1/2]

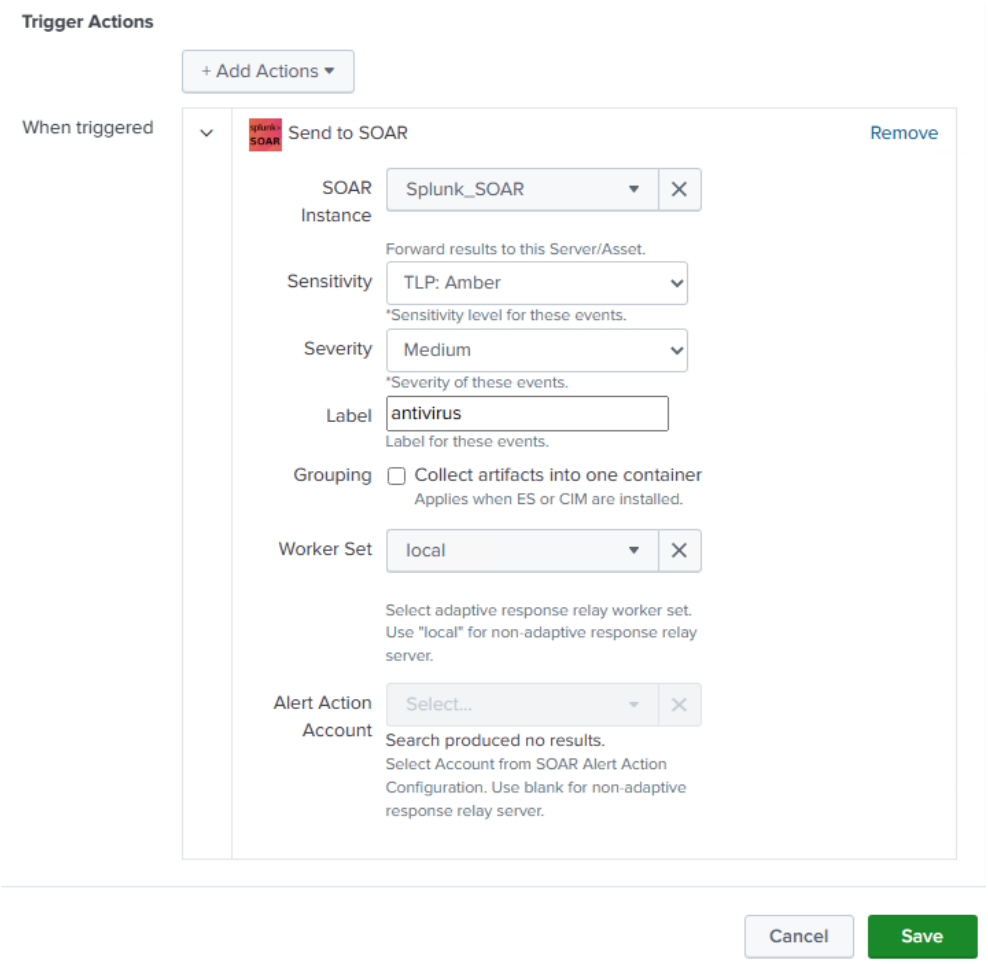
E.1.2 Case 1: Playbook

Figur E.4 viser hele playbooken slik den er konfigurert i Splunk SOAR. Kildekoden til denne playbooken er tilgjengelig gjennom oppgavens kode-repository på GitHub[3]. Playbooken starter med å gå ut i to parallelle avgreininger som kjøres likt. Den første grenen, vist til høyre i figuren, sjekker om Defender har lagt ved en hash til den detekterte skadevaren som en del av antivirus-alarmen. Dersom det finnes en hash fortsetter arbeidsflyten, hvor et oppslag av hashen gjøres i tjenesten *VirusTotal Scan Hash*[4]. Denne tjenesten tar hashen og sammenligner den opp mot en rekke forskjellige databaser og tilkoblede oppslagsverk over kjente skadevarer og andre ondsinnede indikatorer, for å gi et svar på hvor sannsynlig det er at hashen er ondsinnet. Dette gir en deteksjonsrate som regnes ut fra antall positive treff (hashen defineres som ondsinnet av en database eller oppslagsverk) sammenlignet med det totale antallet oppslag (positive + negative treff).

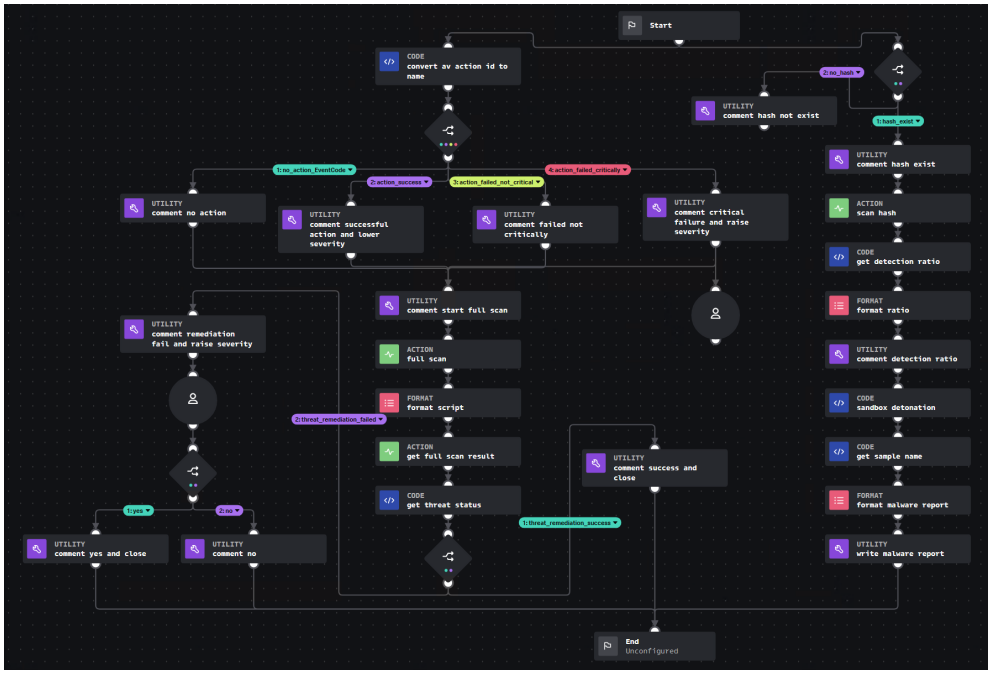
Videre detonerer vi den ondsinnede filen i et detonasjonsmiljø (se skadevareanalyse). For å oppnå dette benytter vi tjenesten *Hybrid Analysis*[5], som er en nettbasert detonasjonstjeneste der det er mulig å laste opp skadevare for automatisk skadevareanalyse, eller gjøre oppslag i tidligere analyser. For denne oppgaven benytter vi hashen vi har detektert for å gjøre oppslag i tidligere skadevareanalyser. Resultatet fra denne analysen hentes ned til Splunk SOAR der analyserapporten skrives ut. Hvis Defender ikke har lagt ved en hash til den detekterte skadevaren, kommenteres dette og grenen avsluttes.

Parallelt med gjennomgang av hashen så henter den andre grenen – det venst-

[3] `https://github.com/TordV/BachelorThesisNTNU2023`
[4] `https://developers.virustotal.com/reference/file-info`
[5] `https://www.hybrid-analysis.com/`



Figur E.3: Case 1: Lage Splunk alarm [2/2]



Figur E.4: Case 1: Playbook

re løpet i figuren – event-koder fra antivirus-alarmen for å avgjøre hva som har blitt detektert og hva Defender har gjennomført for å håndtere hendelsen så langt. Basert på hvilken av event-kodene som eksisterer i alarmen (1117, 1118 eller 1119), så kommenteres det ut forskjellige meldinger. Dersom Defender har håndtert skadevaren, altså event 1117 er registrert, så senkes kritikaliteten på saken og en melding om at handling var suksessfull blir printet ut. Dersom Defender ikke har klart å gjennomføre tiltak, så blir det kommentert at den feilet, dersom feilen var kritisk vil kritikaliteten økes og det vil bli forspurt menneskelig input for å undersøke saken nærmere. Dersom ingen av de tre eventene ble loggført, vil dette bli kommentert og grenen fortsetter.

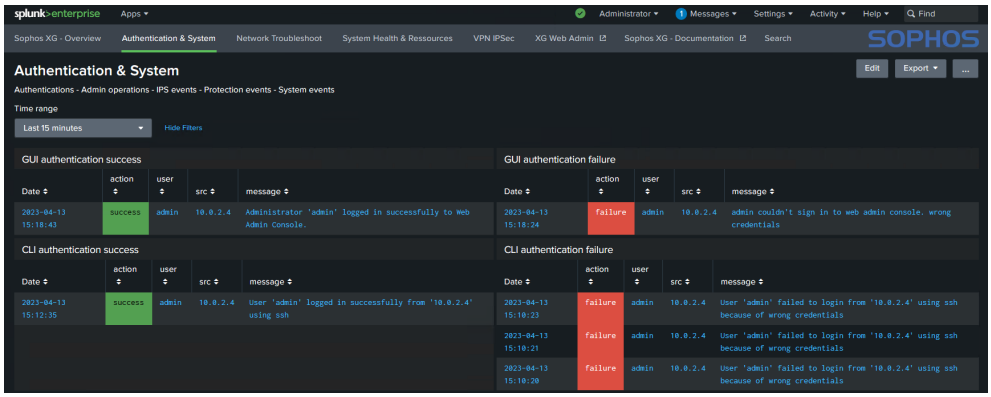
Deretter starter playbooken en fullstendig skanning av den berørte klienten ved hjelp av *Windows Remote Management*[6] (se vedlegg D.7 for beskrivelse og bruk av Windows Remote Management). Deretter gjøres det en ny automatisk beslutning basert på resultatet fra skanningen. Dersom resultatet tilsier at skadevaren har blitt håndtert i sin helhet av Defender, vil saken lukkes automatisk. Derimot, hvis resultatet tilsier at skadevaren fortsatt utgjør en trussel mot den berørte klienten, utløses en ny forespørsel om menneskelig input der en sikkerhetsanalytiker blir presentert informasjonen som er samlet inn så langt i saken, og må bestemme hvorvidt saken skal lukkes eller ikke. Når begge greinene er ferdig kjørt så avsluttes playbooken.

[6] `https://learn.microsoft.com/en-us/windows/win32/winrm/portal`

E.2 Case 2

E.2.1 Case 2: Søk og alarm

Som en del av Sophos Firewall sitt tillegg i Splunk Enterprise (se vedlegg D.2), har Sophos utviklet et dashbord med ferdig konfigurerte søk som visualiserer diverse hendelser og annen data gjennom brannmurloggene som sendes til Splunk Enterprise. Dette dashbordet kan utnyttes for å fange opp en del hendelser og automatisere håndteringen av dem. Blant søkene vi finner i dette dashbordet er søk etter alle vellykkede og feilede påloggingsforsøk, både gjennom nettleser (*GUI*) og kommandolinje (*SSH*), vist i figur E.5. Søkene gir informasjon om hvorvidt forsøket var vellykket eller feilet, hvilket brukernavn som ble brukt, og hvilken IP-adresse påloggingsforsøkene stammer fra. I figuren har vi demonstrert med administratorbrukeren `admin` fra den interne IP-adressen `10.0.2.4`.

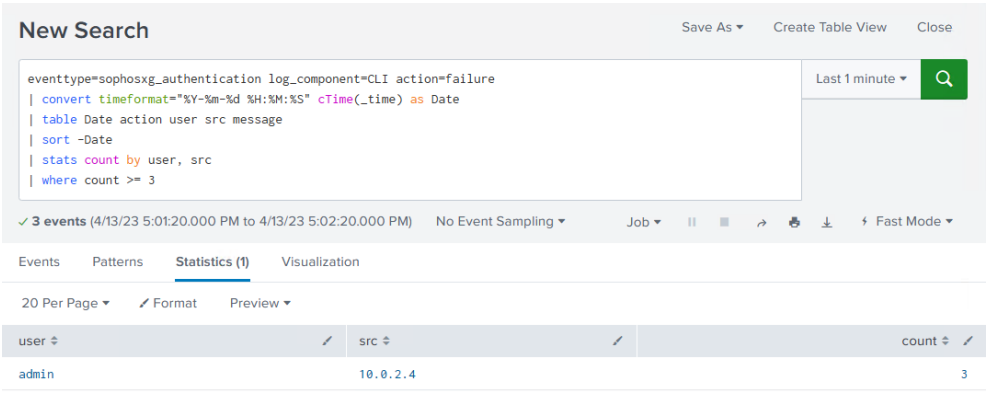


Figur E.5: Case 2: Oversikt over vellykkede og feilede påloggінger i Sophos Firewall sitt dashbord i Splunk Enterprise

For mitigering og håndtering av brute force-angrep fokuserte vi på feilede forsøk over kommandolinjen (*CLI*). Dette er fordi de fleste brute force-angrep gjennomføres med kommandolinjeverktøy som for eksempel *Hydra*[7] og andre lignende verktøy. Dette søket var optimalisert for visualisering i dashbord, men vi modifiserte det slik at det passer med tilstanden vi ønsker å generere alarm for. Vi beholdt søket slik som det var, men la til kommandoen `stats` for å gruppere resultatene på brukernavnet som blir forsøkt pålogget og IP-adressen som står bak de feilede forsøkene. Det betyr at tre feilede forsøk mot samme bruker fra samme IP-adresse blir gruppert på én linje med en ekstra kolonne kalt `count`, med verdi 3. Deretter la vi til kommandoen `where` og definerte at søket kun skal gi resultat dersom denne telleren blir større eller lik 3. Tilslutt bestemte vi at søket til enhver tid skulle se 1 minutt tilbake i tid, slik at det kun genereres resultat av søket dersom det er tre feilede forsøk mot samme bruker fra én enkelt IP-adresse innenfor 1 minutt. Søket vises under og i figur E.6.

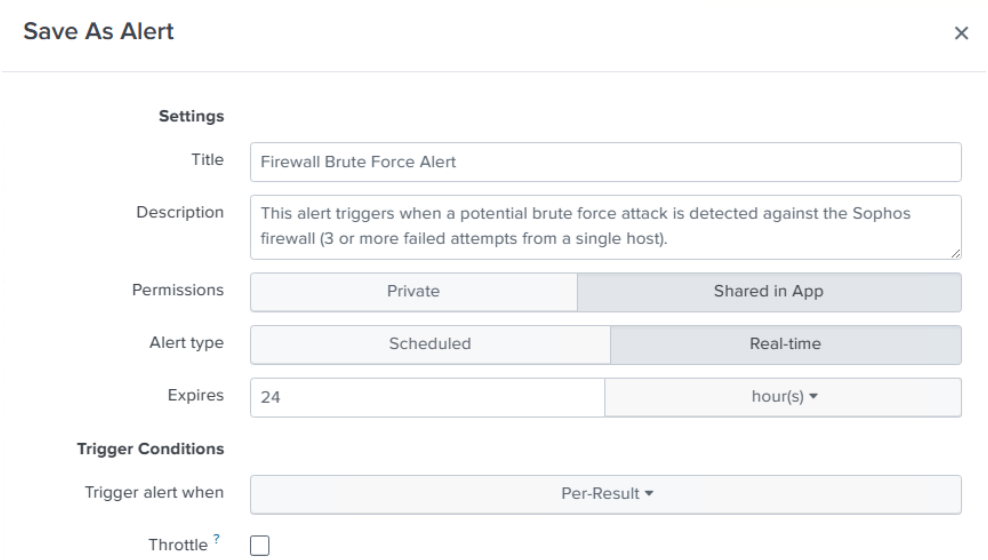
[7]Se følgende lenke for mer info om Hydra og hvordan det fungerer: `https://github.com/vanhauser-thc/thc-hydra`

```
eventtype=sophosxg_authentication log_component=CLI action=failure
| convert timeformat="%Y-%m-%d %H:%M:%S" cTime(_time) AS Date
| table Date action user src message
| sort -Date
| stats count BY user, src
| where count >= 3
```

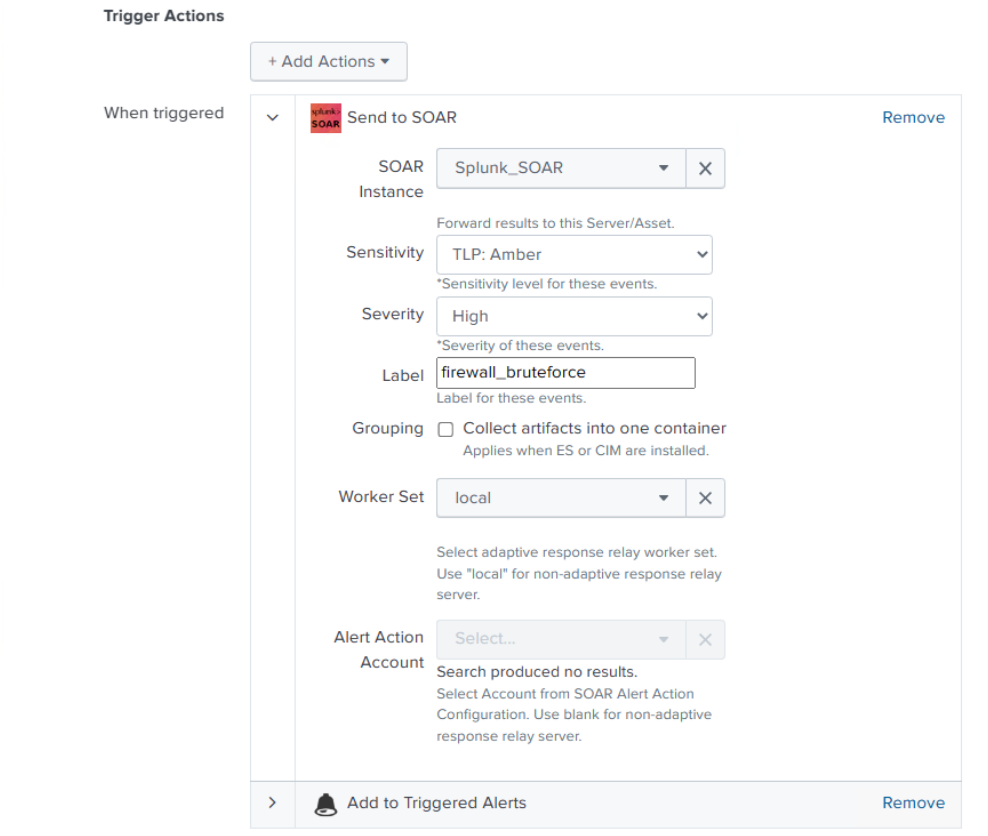


Figur E.6: Case 2: Modifisert søk som skal bli til Splunk-alarm

Dette søket lagret vi som en alarm med navn *Firewall Brute Force Alert*, vist i figur E.7. Vi satt opp alarmen som en sanntidsalarm, i likhet med case 1, og søket vil dermed plukke opp eventuelle hendelser så fort det er tre eller flere mislykkede forsøk innenfor et minutt. Alternativet hadde vært å sette opp søket slik at det utføres hvert minutt, men det åpner opp for at en potensiell trusselaktør kan gjennomføre brute force-angrep i opptil 60 sekunder før søket utføres på nytt og oppdager alle påloggingsforsøkene. Når alarmen utløses ønsket vi at den skal sendes til Splunk SOAR for håndtering, og dette konfigurerte vi som vist i figur E.8. Vi satt sensitivitet lik *TLP: Amber* og kritikalitet til *høy*, og ga alarmen etiketten *firewall_bruteforce*.

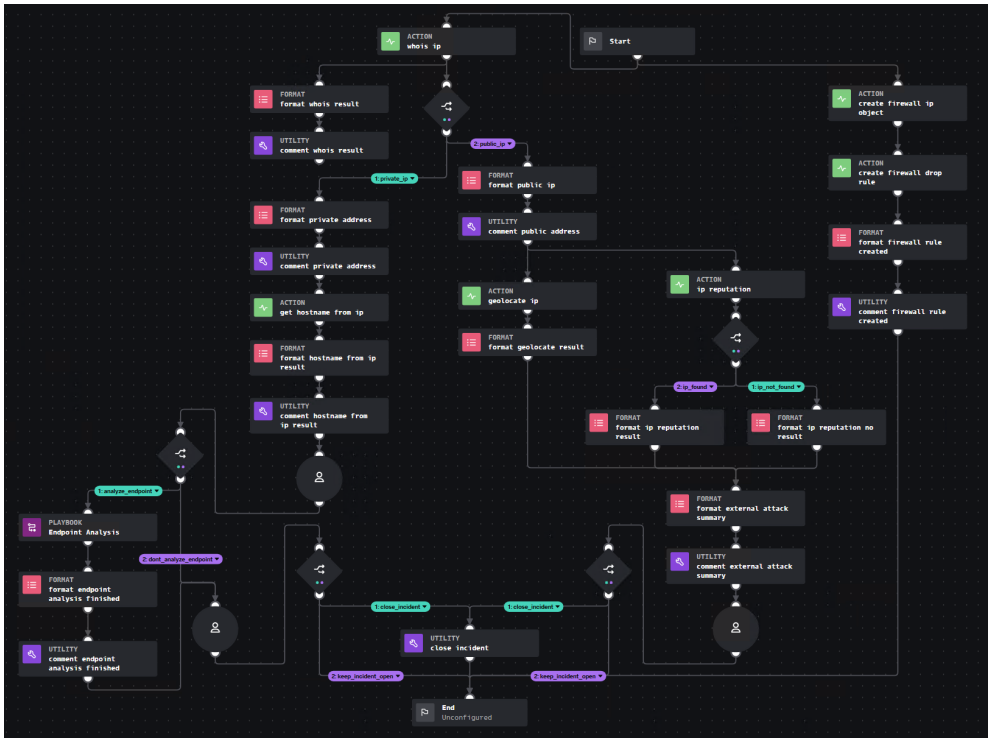


Figur E.7: Case 2: Lage Splunk alarm [1/2]



Figur E.8: Case 2: Lage Splunk alarm [2/2]

E.2.2 Case 2: Playbook



Figur E.9: Case 2: Playbook

Figur E.9 viser hele playbooken slik den er konfigurert i Splunk SOAR. Kildekoden til denne playbooken er tilgjengelig gjennom oppgavens kode-repository på GitHub[8]. Denne playbooken har to hovedgrener. Den første grenen, til venstre i figuren, begynner ved å gjennomføre et *WHOIS IP*[9]-oppslag på IP-adressen som forårsaket brute force-angrepet. Dette oppslaget gir svar på hvem som eier IP-adressen, og hvor den er registrert. Dette resultatet kommenteres i saken før playbooken fortsetter i én av to retninger avhengig av om resultatet tilsier at IP-adressen er offentlig (ekstern angriper) eller privat (intern angriper).

Dersom angripet stammer fra en ekstern angriper, gjennomføres ytterligere bevisinnsamling ved oppslag i tjenestene *MaxMind Geolocate*[10] og *VirusTotal IP Reputation*[11]. Disse tjenestene gir svar på den geografisk opprinnelsen til IP-adressen (MaxMind), og omdømmet til IP-adressen (VirusTotal). Omdømmet beregnes ut ifra antall rapporteringer om ondsinnet eller mistenkelig aktivitet knyttet til IP-adressen som er meldt inn til VirusTotal. Resultatet fra disse oppslagene kommenteres i saken. Grenen avsluttes med en forespørsel om menneskelig input hvor en

[8] `https://github.com/TordV/BachelorThesisNTNU2023`
[9] `https://www.whois.com/whois`
[10] `https://dev.maxmind.com/geoip`
[11] `https://developers.virustotal.com/reference/ip-info`

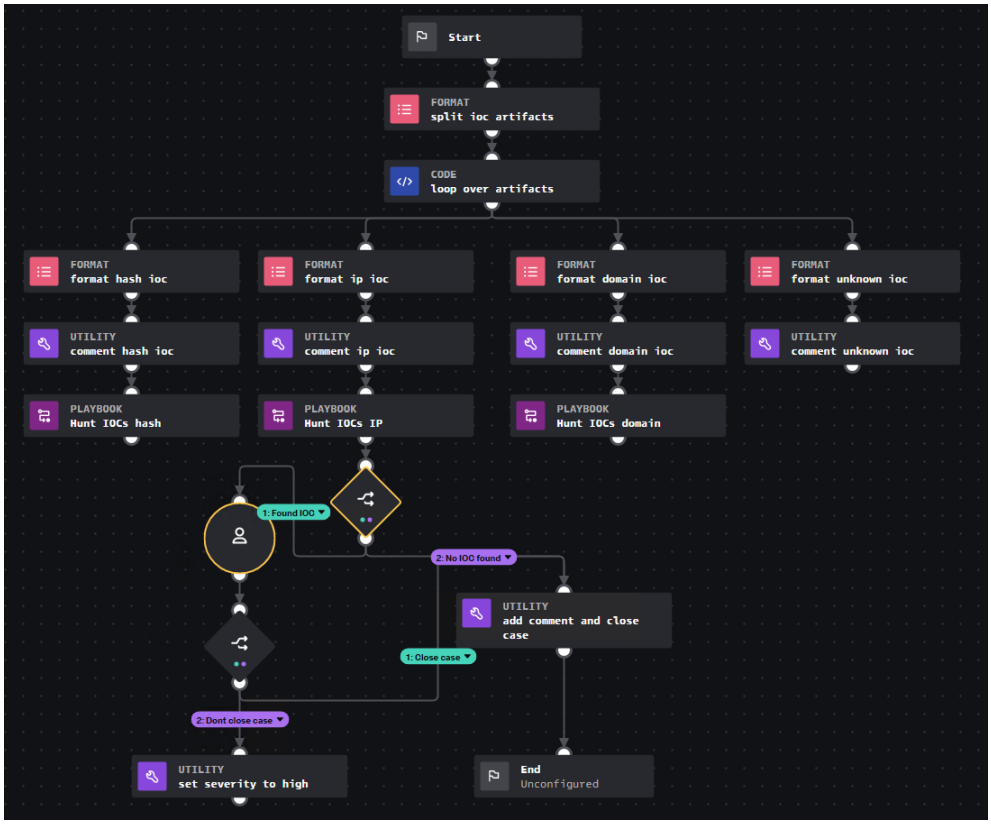
sikkerhetsanalytiker bestemmer om saken skal lukkes eller ikke ut ifra informasjonen som er samlet inn.

Ved interne angrep er arbeidsflyten annerledes. Her gjennomføres først et Splunk-søk som henter maskinnavnet til den interne IP-adressen (se vedlegg D.5 for dokumentasjon av Splunk søk fra Splunk SOAR). Dette kommenteres, før en forespørsel om menneskelig input utløses for å bestemme hvorvidt analyse av dette endepunktet skal gjennomføres eller ikke. Eventuell endepunktsanalyse gjennomføres i en tilkoblet playbook med navnet *Endpoint Analysis*, før resultatet av analysen kommenteres i saken. Utvikling av denne playbooken faller utenfor omfanget av denne oppgaven, og denne playbooken er derfor kun benyttet for å simulere en endepunktsanalyse, og det eneste resultatet den gir er en kommentar om at endepunktsanalyse har blitt gjennomført. I likhet med arbeidsflyten for eksterne angrep avsluttes denne grenen med en forespørsel om menneskelig input for å bestemme hvorvidt saken skal lukkes eller ikke.

Parallelt med alt dette gjennomføres en annen gren vist til høyre i figuren. I denne arbeidsflyten kaller playbooken på Sophos Firewall sitt API for å lage en brannmurregel som blokkerer all trafikk fra IP-adressen som går gjennom brannmuren (se vedlegg D.6 for dokumentasjon av hvordan dette API-et benyttes). Dette gjøres for å stoppe brute force-angrepet før det lykkes med å oppdage passordet til brannmurgrensesnittet.

E.3 Case 3

E.3.1 Case 3: Playbooks



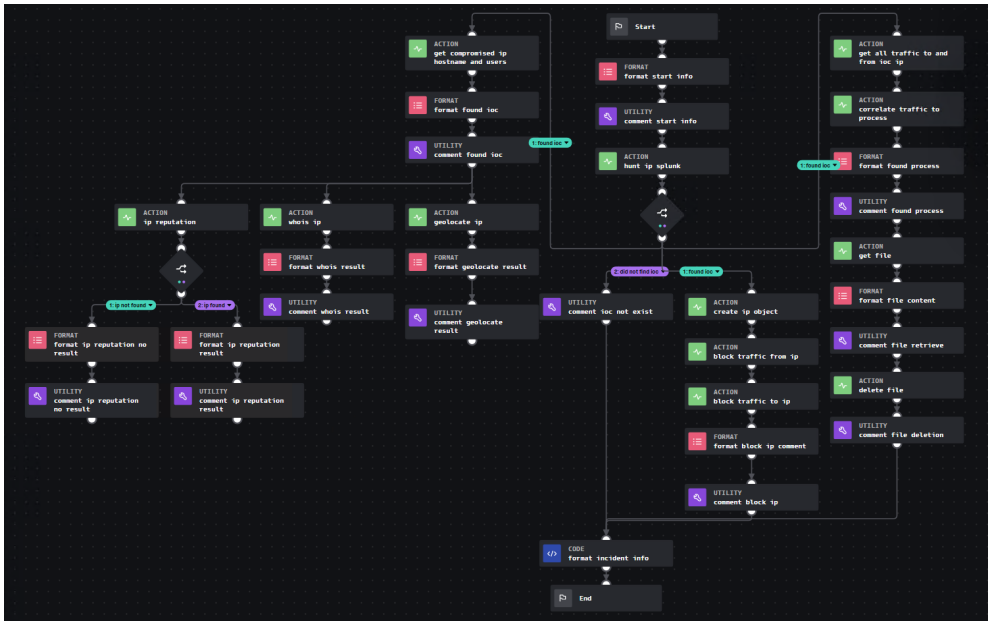
Figur E.10: Case 3: Playbook [1/2]

Oppsettet til case 3 besto av to forskjellige playbooks. Den viktige forskjellen fra case 1 og 2 er at playbookene i denne casen ble designet for å utløses manuelt og ved behov, på bakgrunn trusseletterretningen, og ikke automatisk ved deteksjon av en gitt forhåndsdefinert event. Kildekoden til begge disse playbookene er tilgjengelig gjennom oppgavens kode-repository på GitHub[12]. Den første playbooken vi lagde for gjennomføring av threat hunting kalte vi *Hunt IOC*, vist i figur E.10. Formålet med denne playbooken var å iterere over en eller flere indikatorer fra en fil med trusseletterretning, og starte videre analyse og threat hunting for hver enkelt av dem basert på type indikator. De indikatorene som støttes av denne playbooken er hasher, IP-adresser og domener.

Når playbooken utløses vil den lese inn artefaktene som ligger i den manuelt opprettede saken, og deretter iterere over dem for å undersøke hva slags type indikator det er snakk om. I denne playbooken har vi satt opp logikk for å håndtere

[12] `https://github.com/TordV/BachelorThesisNTNU2023`

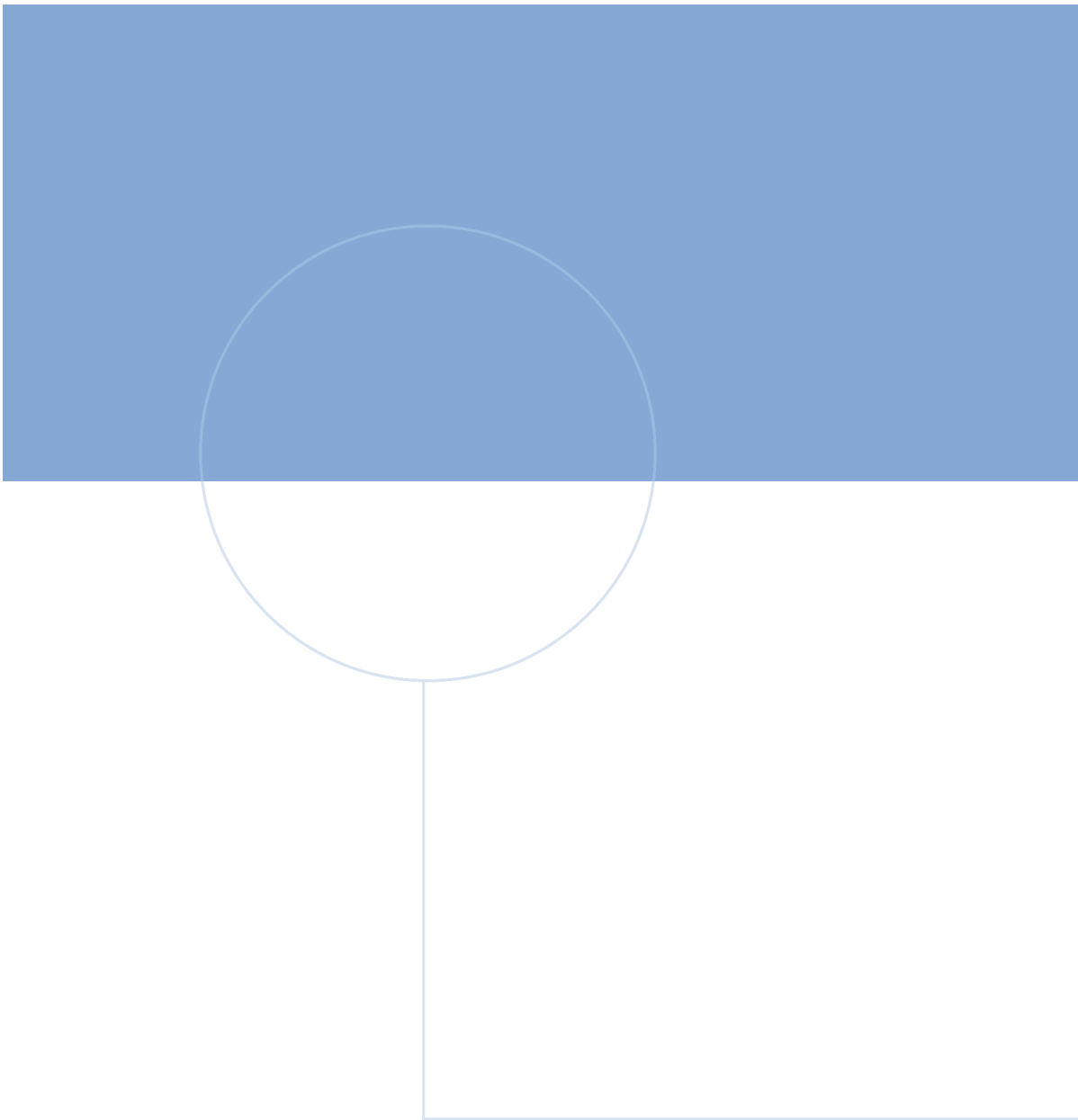
hasher, IP-adresser og domener. For hver artefakt legges det igjen en kommentar i saken om hva slags type indikator det er snakk om, samt verdien til indikatoren (hashen, IP-adressen eller domenenavnet). Deretter, for hver indikator, utløses en tilkoblet playbook for å fortsette threat hunting av den typen indikator. Når den underliggende playbooken er ferdig, vil den returnere med en variabel som forteller hvorvidt indikatoren ble funnet i organisasjonens nettverk eller systemer. Hvis indikatoren ble detektert, utløses en forespørsel om menneskelig input der en sikkerhetsanalytiker blir presentert den informasjonen som er samlet inn og analysert av den underliggende playbooken, med spørsmål om analytikeren ønsker å lukke saken eller ikke.



Figur E.11: Case 3: Playbook [2/2]

Av de tre tilkoblede playbookene, har vi i denne oppgaven begrenset omfanget til å kun inkludere design og implementering av én, playbooken for threat hunting av IP-adresser kalt *Hunt IOCs IP*. Dette er en betydelig større playbook, vist i figur E.11. Denne playbooken kommenterer i saken at threat hunting mot en IP-adresse er startet, før den gjennomfører et søk i brannmurloggene i Splunk Enterprise for å lete etter den ondsinnede IP-adressen. Hvis IP-adressen blir funnet i loggene så starter tre forskjellige arbeidsflyter. Den første avgreningen, vist til venstre i figuren, starter med å hente ut maskinnavnet og brukerkontoene til den eller de interne enhetene som har vært i kontakt med den ondsinnede IP-adressen. Dette kommenteres i saken, før ytterligere analyser av den ondsinnede IP-adressen gjennomføres gjennom tjenestene *VirusTotal IP Reputation*, *WHOIS IP* og *MaxMind Geolocate*. Disse benyttes, som nevnt under gjennomgang av case 2, for å finne ut IP-adressens omdømme, registreringsinformasjon og geolokasjon. Resultatet av disse analysene kommenteres i saken.

Den andre avgreningen, vist helt til høyre i figuren, gjennomfører først et nytt søk i Splunk Enterprise for å hente ut informasjon om all trafikk mellom den ondsinnede IP-adressen og den eller de interne IP-adressene den har vært i kontakt med, inkludert portnummer som er brukt under kommunikasjonen. Dette resultatet benyttes i et nytt Splunk-søk som korrelerer portnummer brukt av den eller de interne enhetene, til spesifikke prosesser på enhetene og de tilhørende programmene eller filene der trafikken stammer fra. Resultatet kommenteres i saken, før de programmene og filene der trafikken stammer fra blir hentet fra den eller de berørte enhetene og lagret som bevis i Splunk SOAR ved hjelp av *Windows Remote Management*-tjenesten. Avslutningsvis vil playbooken i denne grenen slette de programmer eller filer fra enhetene som er detektert og hentet inn som bevis, med mål om å stanse trafikken til den ondsinnede IP-adressen. Parallellt med de to nevnte grenene gjennomfører også playbooken blokkering av den ondsinnede IP-adressen i Sophos Firewall, i likhet med det som ble gjort i case 2.



NTNU

Kunnskap for en bedre verden