Osmo Mittilä

# SIEM-HÄLYTYSSTÄÄNTÖJEN YLLÄ-PITO SOAR-JÄRJESTELMÄN AVULLA

Opinnäytetyö

Liiketalouden ammattikorkeakoulututkinto

Tietojenkäsittelyn koulutus

2025

Kaakkois-Suomen ammattikorkeakoulu

Kaakkois-Suomen ammattikorkeakoulu


| | |
|---|---|
| Tutkintonimike | Tradenomi (AMK) |
| Tekijä/Tekijät | Osmo Mittilä |
| Työn nimi | SIEM-hälytyssääntöjen ylläpito SOAR-järjestelmän avulla |
| Toimeksiantaja | Istekki Oy |
| Vuosi | 2025 |
| Sivut | 31 sivua |
| Työn ohjaaja | Jukka Selin |


## TIIVISTELMÄ


Opinnäytetyön tarkoituksena oli tutkia FortiSIEM- ja XSOAR-järjestelmien välistä integraatiota ja toteuttaa mahdollisia SIEM-hälytyssääntöjen ylläpitoon liittyviä toimintoja XSOAR-pelikirjan avulla, sekä tutkia SOAR-järjestelmän muita mahdollisia toteutustapoja automaatiolle. Työn toimeksiantajana on Istekki Oy, joka halusi selvittää, kuinka uusia tietoturvatuotteita voidaan käyttää yhdessä.

Opinnäytetyön teoriaosuudessa selvitetään, miksi SIEM- ja SOAR-järjestelmiä on kehitetty. Lisäksi selvitetään, mistä SIEM-järjestelmä koostuu ja minkälaisia ominaisuuksia voi olla osana järjestelmää sekä SOAR:n toimintaa. Opinnäytetyö on rajattu käsittelemään järjestelmien teknisiä toiminnallisuuksia.

Opinnäytetyön käytännön osuudessa tutustutaan Fortinetin FortiSIEM -järjestelmään ja Palo Alton Cortex XSOAR -järjestelmään. FortiSIEM -järjestelmän osalta perehdytään hälytyssäännön toteutukseen, jotta ymmärretään miten hälytykset nousevat ja päätyvät XSOAR-tapahtumiksi. XSOAR:lla toteutetaan yksinkertainen pelikirja, jolla voidaan lisätä tietoa FortiSIEM-tarkkailulistalle. Vaihtoehtona pelikirjalle toteutetaan myös sama toiminnallisuus koodilla, joka voidaan liittää esimerkiksi painonappiin tai ajaa erillisenä komentona.

Hälytyssääntöjen prosessin avulla voidaan jatkossa kehittää omia sääntöjä FortiSIEM:ssä. Pelikirjojen logiikan ja automaation selvityksen perusteella voidaan jatkossa kehittää automaatiota vastamaan tietoturvakeskuksen tarpeita ja vähentämään toistuvaa manuaalista työtä.

Lopputuloksena valmistui pelikirja ja koodi, joilla voidaan lisätä tietoa FortiSIEM-tarkkailulistalle. Pelikirja ja koodi tarjoavat pohjan, jota voidaan käyttää eri käyttötarkoituksiin tehtäviin pelikirjoihin ja koodeihin. Työn tuloksessa nähdään, kuinka automaation avulla voidaan selkeä prosessi automatisoida.



**Asiasanat:** SIEM, SOAR, tarkkailulista, hälytyssääntö

South-Eastern Finland University of Applied Sciences

| | |
|---|---|
| Degree title | Bachelor of Business Administration |
| Author | Osmo Mittilä |
| Thesis title | Maintaining SIEM alert rules using SOAR |
| Commissioned by | Istekki Oy (ICMT service provider) |
| Time | 2025 |
| Pages | 31 pages |
| Supervisor | Jukka Selin |

## ABSTRACT

The purpose of the thesis was to study the integration between FortiSIEM and XSOAR and implement possible functions related to SIEM alert rule maintenance using the XSOAR playbook and examine other possible implementations of the SOAR system for automation. This objective was based on the commissioner's desire to study how new information security products could be used together.

The theoretical part of the thesis explains the reasons for the development of SIEM and SOAR systems, the components of SIEM, functionalities of SOAR and the features of the system. The scope of the thesis was limited to contain only the technical functionalities of the systems.

In the discussion part, Fortinet's FortiSIEM system and Palo Alto's Cortex XSOAR system are intoduced. As for FortiSIEM, the implementation of an alert rule is examined in order to understand how alerts are raised and transferred to XSOAR as events. With reference to XSOAR, a simple playbook was implemented that can be used to add information to the FortiSIEM watchlist. As an alternative to the playbook, the same functionality was implemented as a code that can be attached to a button or run as a separate command.

The result was a playbook and code that can be used to add information to the FortiSIEM watchlist. The playbook and code provide a foundation that can be used for different purposes. The alert rule process can be used to develop own rules in FortiSIEM. Based on the analysis of the logic and automation of the playbooks, automation can be developed in the future to meet the needs of the security operation center and reduce repetitive manual work.

**Keywords:** SIEM, SOAR, watchlist, alert rule

**SISÄLLYS**

# 1 JOHDANTO

Maailman digitalisoitumisen myötä on digilaitteiden käytöstä tullut arkipäivää. Verkon välityksellä voidaan hoitaa pankkiasioita, ohjata etänä erilaisia koneita, käydä kauppaa sekä hoitaa lukemattomia muita tehtäviä. Toimintojen siirtyessä verkkoon lisääntyy myös ihmisten käyttämä aika verkossa. Verkkorikollisuus on kasvanut myös digitalisaation myötä sekä kehittynyt kiihtyvään tahtiin teknologian mukana. Verkkorikollisuuden valvontaan ja reagointiin kehitetään jatkuvasti tietoturvatuotteita, joiden avulla on mahdollista estää, havaita ja reagoida mahdollisiin tietoturvatapahtumiin.

Istekki Oy on kotimainen voittoa tavoittelematon inhouse-yhtiö, joka tarjoaa kattavasti Information, Communications and Medical Technology (ICMT) - palveluita sen omistaja-asiakkailleen. Palveluita tarjotaan noin 1400 työntekijän voimin. Digiturvapalvelut ovat yksi Istekin palveluista. Niiden tarkoituksena on tarjota asiakkaille kokonaisvaltaisia palveluja tietoturvan ja -suojan parissa. Digiturvapalvelun alla toimii myös tietoturvavalvomo, joka tuottaa ympärivuorokautista valvontaa asiakkaiden ympäristöihin. Jokainen asiakasympäristö on erilainen ja tietoturvavalvomon resurssit ovat myös rajalliset.

Opinnäytetyön aiheen valinta perustuu siihen, että toimeksiantaja haluaa selvittää uusien tietoturvatuotteiden ominaisuuksia käytännössä. Security Information and Event Management (SIEM) -järjestelmällä hyödynnetään keskitettyjä lokeja verkon tapahtumien valvontaan ja Security Orchestration and Automation (SOAR) -järjestelmällä voidaan reagoida SIEM-järjestelmän tuottamiin hälytyksiin. Toimeksiannon tavoitteena on automatiikan avulla tehostaa SIEM-hälytyssääntöjen ylläpitoa. Hyötynä saadaan selkeä prosessi, automaattinen dokumentointi ja välitön vaikutus hälytyssääntöihin sekä säästetään paljon henkilötyötunteja.

Tässä opinnäytetyössä keskitytään kahden tietoturvatyökalun keskeisiin käsitteisiin ja syvennytään niiden teknisiin ominaisuuksiin. Toisessa pääluvussa käydään lävitse SIEM-tuotteisiin liittyviä aiheita kuten tiedon kulku, lokien normalisointi ja hälytykset sekä nykyaikaisia ominaisuuksia, joita toimittajat ovat

lisänneet tuotteisiinsa. Kolmannessa pääluvussa käsitellään keskeisiä asioita SOAR-teknologiasta. Neljännessä pääluvussa toteutetaan Fortinetin FortiSIEM- ja Palo Alton Cortex XSOAR -järjestelmien avulla hälytyssääntöjen ylläpitoa käytännössä ja samalla käydään läpi kyseisten tuotteiden opinnäytetyöhön liittyviä teknisiä ominaisuuksia.

# 2 SIEM

Kyberturvallisuuden riskit ovat kasvaneet valtavasti viime vuosina. Riskien nousun syynä ovat pääsääntöisesti olleet kyberrikolliset ja valtiolliset toimijat. Hyökkäykset ovat hienostuneet, ja sen myötä niiden havaitsemisesta mahdollisimman nopeasti on tullut haastavaa. NIST (National Institute of Standards and Technology) on tehnyt raportin, jonka mukaan ICT-järjestelmien pitäisi pystyä tarjoamaan reaaliaikaista anomalioiden havaitsemista, nopeampaa tapahtumien hallintaa ja visualisointia verkosta sekä kaikista verkkoon liittyvistä laitteista. (González-Granidillo ym. 2021.)

## 2.1 Mikä SIEM on?

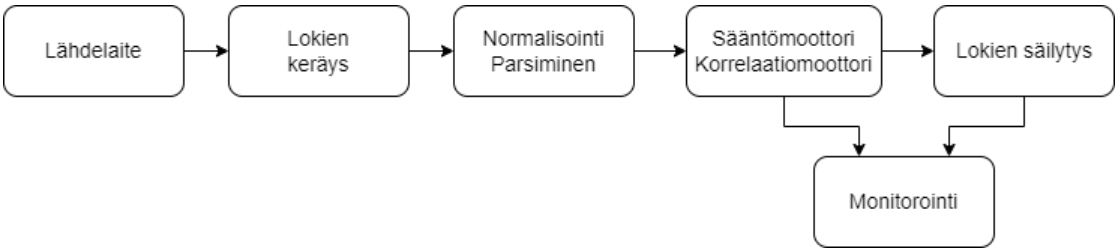
SIEM-järjestelmät on kehitetty vastaamaan nykyaikaisia vaatimuksia. Alkunsa SIEM on saanut 2000-luvun alkaessa. Kidd (2023) kertoo julkaisussaan, että SIEM on kehittynyt Security Information Management (SIM)- ja Security Event Management (SEM) –prosessien yhdistelmästä ja vuonna 2005 Gartner ® loi SIEM-termin virallisesti. Samassa julkaisussa Kidd arvioi SIEM-teknologiaratkaisujen markkinoiden jatkavan vahvaa kasvua Kasvun ajureina toimivat kyberrikollisuuden nopeasti kasvavat tapahtumat, laajamittaisiin datavirtoihin perustuvien IT-palveluiden käyttöönotot ja monimutkaiset IT- ja tieto-alustat, joilla hallinnoidaan tietoa ja ohjelmistoja pilvessä.

SIEM-järjestelmä antaa yleisnäkymän valvottavasta ympäristöstä tuomalla keskitetysti tietoturvaan liittyvät tiedot yhteen. Tärkeitä elementtejä SIEM:ssä ovat lokienhallinta, tapahtumien korrelointi, hälyttäminen, tapahtumien hallinta, sekä raportointi ja analytiikka. Reaaliaikainen monitorointi, automaattinen tapahtumiin reagointi, uhkatiedon yhdistäminen ja edistynyt analytiikka mahdollistavat SIEM:n käytön useissa käyttötapauksissa. (SentinelOne 2024.)

Käyttötapauksia voivat olla tietoturva poikkeaminen havaitseminen, kuten käyttäjän tunnusten vaarantumisen tai epätyypillisen toiminnan havaitseminen korkeiden oikeuksien tunnuksilla. Järjestelmiin liittyen voidaan seurata järjestelmien muutoksia tai kuormitusta. SIEM:ä voidaan käyttää lokienhallintaan, mikä mahdollistaa tietoturvauhkien jäljittämisen. Asetusten kuten GDPR:n, HIPAA:n tai PCI:n vaatimusten täyttyminen pystytään toteuttamaan SIEM:in avulla. SIEM:ssä voidaan automatisoida uhkien havainnointia, mikä toimii perustana automaattiselle tapahtumien käsittelylle. SIEM:stä voidaan lähettää tieto havaitusta uhkasta ulkoiseen järjestelmään kuten SOAR, joka hoitaa automaattisen tapahtuman käsittelyn. SOAR voi olla myös integroituna SIEM-järjestelmään. (Logpoint s.a.)

## 2.2 SIEM-järjestelmän rakenne

SIEM-järjestelmä koostuu useasta yksittäisestä osasta, jotka voivat toimia yksinään. SIEM-järjestelmä ei voi toimia kunnolla, jos yksikin osa on epäkunnossa. Järjestelmän toiminnan kannalta olennaisia osia ovat lähdelaite, lokin keräys, lokien normalisointi, sääntö- ja korrelaatiomoottori, lokien säilytys ja monitorointi (kuva 1). (Miller ym. 2010, 112.)




Kuva 1. SIEM-komponentit

Työnkulku SIEM-prosessissa alkaa kun lähdelaitteessa tuotetaan lokia ja se siirretään SIEM-järjestelmään. Loki normalisoidaan, jotta sitä voidaan käyttää sääntö- ja korrelaatiomoottorilla tehokkaasti. Monitorointi voidaan toteuttaa lähes reaaliaikaisiin lokeihin tai järjestelmään tallennettuihin lokeihin. Seuraavaksi perehdytään tarkemmin jokaisen komponentin toimintaan.

*Lähdelaite*

Reititin, kytkin, palomuuri, virustorjunta sekä muut laitteet ja ohjelmistot tuottavat lokia, jota voidaan viedä SIEM:iin prosessoitavaksi ja säilytettäväksi. Näitä

laitteita kutsutaan lähdelaitteiksi. Ne eivät ole varsinaisesti osa ostettavia SIEM-tuotteita, mutta ne ovat tärkeä osa SIEM-prosessia. Ilman lähdelaitteista tuotavaa lokia SIEM-järjestelmä on kuin tyhjä kuori. (Miller ym. 2010, 113–114.)

SIEM:llä valvottavan ympäristön lähdelaitteiden tunteminen on tärkeää arkkitehtuuria suunniteltaessa. Lokien keräämistä mietittäessä on myös tärkeä miettiä, mitä lokia kerätään. Lokilähteet voivat tuottaa valtavasti erilaista lokitietoa, josta osa on suunnitellun toiminnan kannalta oleellisempaa. SIEM:in käytössä on yleensä rajalliset resurssit, joten on mietittävä, kuinka paljon valittu ratkaisu pystyy rajallisilla resursseillaan käsittelemään lokia. (Miller ym. 2010, 114.)

*Lokien keräys*

Lokien keräämiseen on kahdenlaista tapaa. Lokeja voidaan lähettää SIEM:iin tai SIEM voi noutaa lokeja lokilähteestä. Lokien lähettämistä varten täytyy pystyttää vastaanotin tuleville lokeille. Lähdelaitteet konfiguroidaan lähettämään lokit vastaanottimeen, joka kerää ne SIEM-järjestelmään. Noudettaessa lokeja SIEM käynnistää yhteyden lähdelaitteeseen ja noutaa lokit. Lokien keräämismetodeja vertailtaessa on otettava huomioon, että lähetysmenetelmällä lokitus on lähes reaaliaikaista, kun taas noutamalla lokituksessa on viivettä riippuen konfiguraatiosta. (Miller ym. 2010, 117.)

Ympäristöissä, joista kerätään lokeja SIEM-järjestelmään, on yleensä useita erilaisia laitteita ja sovelluksia. SIEM:in toimittajasta riippuen on niihin yleensä valmiiksi rakennettu erilaisia metodeja lokien keräykseen, kuten valmiita autentikointimetodeja ja lokien noutometodeja. SIEM:ssä ei välttämättä ole kaikille halutuille lokeille standardisoitua lokitustapaa, jolloin täytyy tehdä mukautettu lokien keräys. (Miller ym. 2010, 118–119.)

*Normalisointi*

Lokien saapuessa SIEM:iin ovat ne alkuperäisessä formaatissaan. Jotta lokeista saataisiin suunniteltu hyöty, pitää lokit muuttaa yhtenevään muotoon. Tätä operaatiota kutsutaan normalisoinniksi. Jokaisella SIEM:llä on oma ta-

pansa normalisoida sisään tulevat lokit, mutta lopputuloksena lokit ovat yhtenevässä formaatissa, joka helpottaa lokien lukemista ja mahdollistaa yhtenevän tavan luoda hälytyssääntöjä. (Miller ym. 2010, 119–120.)

*Sääntö- ja korrelaatiomoottori*

Säännöt perustuvat pitkälti Boolen logiikkaan, jonka avulla määritellään, täyttyvätkö ehdot. Sääntömoottori tutkii normalisoidusta datasta täyttyvätkö säännön ehdot ja laukaisee hälytyksen ehtojen täyttyessä. Säännöt voivat yksinkertaisimmillaan olla yksittäisiä tapahtumia, kuten yksittäisten kirjautumisten seuraamista. Kompleksisuutta voidaan lisätä sääntöön esimerkiksi lisäämällä ehtoja siitä, minkälainen kirjautuminen kyseessä, onko kyseessä ylläpitäjä, mihin aikaan kirjautuminen on tapahtunut jne. (Miller ym. 2010, 121–122.)

Korrelaatiomoottori toimii sääntömoottorin osana. Korrelaatiomoottorin tarkoituksena on yhdistää tapahtumia eri laitteista yhdeksi korreloiduksi tapahtumaksi (Miller ym. 2010, 122–123). Käytännössä ilman korrelaatiomoottorin toimintaa viidestä epäonnistuneesta kirjautumisesta yhteen laitteeseen nousisi viisi eri hälytystä. Korreloinnin jälkeen nousisi yksi hälytys, joka kertoisi viidestä epäonnistuneesta kirjautumisesta laitteeseen.

*Lokien säilytys*

Jotta SIEM:iin tulevia lokeja voitaisiin käyttää säilytystarkoitukseen tai historiallisten hakujen tekemiseen, täytyy niille olla jonkinlainen tallennustapa. Tietokannat ovat yleisin tapa toteuttaa lokien säilytys. SIEM-järjestelmää asentaessa valitaan, mitä tietokantaa käytetään. SIEM-järjestelmästä riippuen toimittaja saattaa tarjota tukea rajoitetulle määrälle tietokantoja, joten tietokannan ylläpidon vastuut voivat olla täysin käyttävän organisaation vastuulla. Tietokannan tulee olla optimoitu toimimaan SIEM:in kanssa. (Miller ym. 2010, 124.)

Lokien säilytystä suunniteltaessa on mietittävä, paljonko lokia järjestelmään tuodaan. Säilytysaikaa voidaan määritellä erilaisilla lokia koskevilla vaatimusmäärittelyillä. Lokeilla on elinkaari, jonka loppuvaiheessa ne voidaan arkistoida tilankäytöllisistä syistä. Arkistoituja lokeja voidaan tarpeen vaatiessa käyttää. Arkistoimalla voidaan vapauttaa tilaa tärkeämmälle lokille. Kun lokille

ei enää ole käyttöä eikä säilyttäminen ole tarpeen, tulisi lokitiedolle olla määriteltynä poistamiseen menettely. (Traficom 2023.)

*Monitorointi*

Käyttöliittymän kautta käyttäjä pystyy monitoroimaan SIEM-järjestelmää. Käyttöliittymä mahdollistaa myös vuorovaikutuksen tallennetun tiedon kanssa. Monitorointi mahdollistaa tapahtumien tutkimisen, hakujen tekemisen tietokantaan, sekä sisällön ja sääntöjen kehittämiseen. Käyttöliittymä on pääasiallinen työkalu käyttää SIEM-järjestelmään tallennettuja tietoja. (Miller ym. 2010, 126.)

## 2.3 SIEM-ominaisuuksia

*Tekoäly ja koneoppiminen*

Koneoppiminen on tekoälyn osajoukko. Koneoppiminen perustuu algoritmeihin, joilla voidaan esimerkiksi korreloida tapahtumia tai ennustaa mahdollisia uhkia. Algoritmit on suunniteltu vastaamaan kasvavaan tiedon määrään. Tekoäly on kehitetty matkimaan ihmisen ajattelua, jonka apuna se käyttää koneoppimisen tekniikoita. (Paloalto s.a b.)

*UEBA*

UEBA tulee sanoista User and Entity Behavior Analytics eli käyttäjien ja kokonaisuuksien käyttäytymisen analytiikka. Se on ratkaisu, joka algoritmien ja koneoppimisen avulla tunnistaa anomalioita verkossa. Jotta UEBA voi toimia tehokkaasti, täytyy sen olla yhteydessä kaikkiin työntekijöihin ja laitteisiin, joita käytetään työntekoon. Opetusvaiheessa UEBA kerää tietoa laitteiden ja verkon käytöstä, jonka perusteella se määrittelee, mikä on normaalia toimintaa. (Fortinet s.a.)

*Generatiivinen AI*

Finger (2023) käsittelee blogissaan, miksi generatiivinen tekoäly on tullut osaksi SIEM-järjestelmiä. Pääsyiksi Finger mainitsee hälytysten määrän, ja se vaatii paljon aikaa käsittelyyn. Lisäksi hän mainitsee kehittyneiden hyökkäysten kasvaneen määrän. Kehittyneiden hyökkäysten myötä tietoturva-asiantuntijoiden osaamisvaatimukset ovat kasvaneet. Generatiivisen tekoälyn avulla

pyritään parantamaan tietoturva-asiantuntijoiden tehokkuutta turvallisuusoperaatioissa.

## 3 SOAR

SOAR ja SIEM ovat hyvin samankaltaisia järjestelmiä, joita usein verrataan toisiinsa. SOAR:lla on mahdollista kerätä samoja tietoja kuin SIEM:llä, mutta lisäksi voidaan kerätä tietoa useista muistakin lähteistä, kuten päätelaitesuojaus-ohjelmista, uhkatietosyötteistä ja kolmannen osapuolen lähteistä. SOAR mahdollistaa myös hälytysten nostamisen syötteistä, mutta vie käsittelyn pidemmälle esimerkiksi ennalta määriteltyjen pelikirjojen avulla. Niiden avulla voidaan ratkaista hälytyksestä muodostunut tapahtuma tai hoitaa automaattisesti tapahtuman hallintaa lähettämällä automaattiviestejä asiaan liittyville tahoille. (Sosa 2022.)

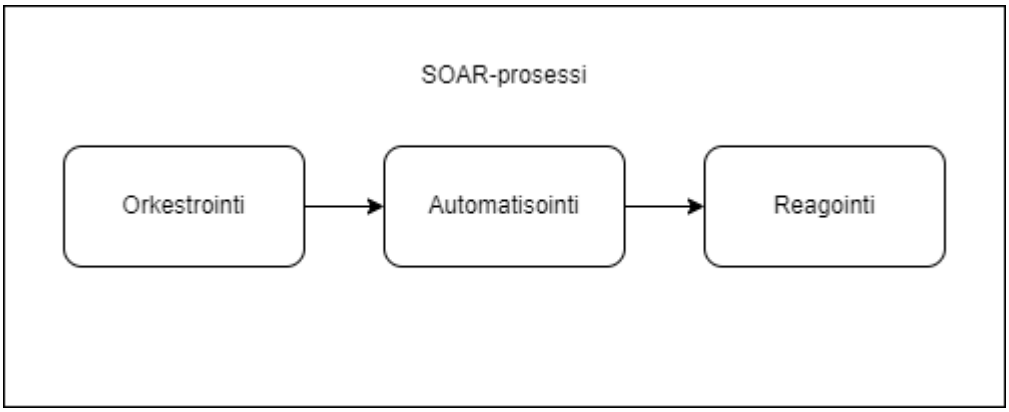
*Mikä SOAR on?*

Security orchestration, automation and response (SOAR) on tietoturvatyökalu, jonka avulla yritykset voivat reagoida nopeasti kyberhyökkäyksiin. Gartnerin määritelmän mukaan kattava SOAR-tuote on suunniteltu toimimaan uhkien ja haavoittuvuuksien hallintaan, tietoturvatapahtumiin reagointiin, sekä toimintojen automatisointiin. Orkestroinnilla tarkoitetaan teknologioita, jotka auttavat yhdistämään kyberuhkia. Automatisoinnilla viitataan teknologioihin, jotka mahdollistavat toimintojen automatisoinnin ja organisoinnin (Paloalto s.a a.)

Kasvavan digitalisoitumisen myötä organisaatioiden kohtaamat haasteet kyberturvallisuuteen liittyen ovat kasvaneet. SOAR mahdollistaa erilaisten tietoturvatuotteiden integroimisen yhteen keskitettyyn paikkaan, mikä vähentää eri työkalujen käyttöön vaadittavaa aikaa. Tietoturvatapahtumien reagointiajat lyhenevät, kun käytetään SOAR:in tarjoamia automatisointimahdollisuuksia. Tietojen kerääminen useasta lähteestä parantaa näkyvyyttä laajempaan kokonaisuuteen. Yhdessä paikassa tapahtuvat toiminnot tietoturvaan liittyen mahdollistavat kokonaisvaltaisen raportoinnin valvottavassa ympäristössä tapahtuvista asioista. Lisäksi SOAR:in avulla voidaan parantaa analyytikkojen päätöksentekoa pelikirjojen avulla. (Paloalto s.a a)

*SOAR-rakenne/-toiminta*

Sosa (2022) pureutuu blogissaan SOAR-elementteihin. Nimensä mukaan SOAR koostuu orkestroinnista, automaatiosta ja tapahtumiin reagoinnista (kuva 2). Kukin elementti pitää sisällään useita ominaisuuksia, joista se koostuu. Orkestrointiin sisältyy erilaisten tietojen keräämistä eri lähteistä. Automaatio sisältää ennalta määriteltyjä pelikirjoja ja automaattisesti suoritettavia tehtäviä. Tapahtumiin reagointiin kuuluu tapausten hallintaa, raportointia ja uhkatiedon jakamista.




Kuva 2. SOAR-prosessi

SOAR-työnkulku hämärtää elementtien rajoja. Islam (2020) syventyy artikkelissaan SOAR työnkulkuun. Perimmäinen tarkoitus on tehostaa automaatiota orkestroinnin avulla. Ensimmäinen tehtävä SOAR-alustalla on integroida tietoturvatyökalut alustaan. Toisena tehtävänä on orkestroida integraatioista saatu tieto yhtenevään muotoon, mikä mahdollistaa organisaation tietoturvaprosessien tai tapahtumiin reagointiprosessin toteutuksen. Kolmantena tehtävänä on automatisointi tai tapahtumiin reagointi.

Islam (2022. 42–45) on jakanut orkestroinnin yhdistämisyksikköön ja orkestrointiyksikköön. Yhdistämisyksikön tehtävä on yhdistää SOAR tietoturvatyökaluihin API-rajapintojen avulla. Rajapintojen avulla voidaan kerätä uhkatietoa, tietoturvatyökalujen tapahtumatietoja ja hälytyksiä tai muuta tietoturvatuotteiden tuottamaa tietoa. API-rajapinnan avulla SOAR-järjestelmän voi suorittaa kolmannen osapuolen tietoturvatuotteissa rajapinnan mahdollistamia toimintoja. Esikäsittelyn avulla saadaan raakadata eri lähteistä yhdistettyä yhteneväiseksi. SOAR-tuotteet tarjoavat näkymiä, joiden avulla voidaan nähdä kokonaisvaltainen tila valvottavan ympäristön tietoturvan tilasta.

Kun tutkitaan tapahtuma, voidaan havaita toistuvia työtehtäviä. Automaation tarkoituksena on pyrkiä keventämään analyytikkojen työkuormaa tekemällä yksinkertaisia ja toistuvia tehtäviä. Esimerkiksi jos tapahtuma sisältää yhteyksiä tuntemattomista IP-osoitteista, voidaan niistä hakea automaation avulla lisätietoa analyytikolle ennen kuin tapahtuman tutkintaa on aloitettu. (Kovacevic 2023, 32.)

Automaatioita varten olisi hyvä olla oma politiikkansa siitä, mitä voidaan automatisoida, mitä ei voida automatisoida ja minkälaisissa tapauksissa tarvitaan automaatiolle luvituspyyntö. Pelikirjojen avulla voidaan automatisoida yksinkertaisia ja selkeitä tutkimuksia. Päätöksiä vaativissa tehtävissä on hyvä ottaa analyytikko tekemään päätökset, joiden pohjalta automatiikka voi edetä pelikirjan mukaan. (Kovacevic 2023, 33–34.)

Hyvin organisoitu tapahtumienhallinta mahdollistaa tietoturva-analyytikon tehokkaan työskentelyn. SOAR:in yksi keskeisistä tehtävistä on tarjota keskitettyä tapahtumienhallintaa. Usean työntekijän tietoturvavalvomossa pystytään helposti seuraamaan, mitkä tiketit ovat työn alla ja kuka niitä hoitaa. Tikettien priorisointi on helppoa vakavuusluokittelun avulla. (Kovacevic 2023, 15.)

Tapahtumien hallinnan päätarkoituksena on mahdollistaa tapahtumien havaitseminen, tutkiminen, rajoittaminen ja toipuminen sekä dokumentointi (Kovacevic 2023, 14). Kaikkien vaiheiden tehokas dokumentointi auttaa tunnistamaan mahdolliset kehitystarpeet. Esimerkiksi jos tietyn tyyppisten tapahtumien ratkaisemiseen tarvittava aika poikkeaa huomattavasti muista, voidaan tutkia mistä asia johtuu ja kehittää ratkaisu ratkaisuajan tehostamiseksi.

Tapahtumien tutkintaan on tarjolla erilaisia raameja kuten NIST ja SANS. Organisaatiot voivat käyttää näitä valmiita puitteita tai kehittää omansa. Tutkintaa suoritettaessa on valmistautuminen ensimmäinen askel. Valmistautumisvaiheessa tehdään suunnitelma tapahtuman käsittelystä, mitkä ovat oleelliset tiedot selvitystyötä varten; keneen otetaan tarvittaessa yhteyttä, miten asiat

dokumentoidaan jne. Seuraavassa vaiheessa toteutetaan analysointi ja pyritään tunnistamaan, onko tapahtuma todella haitallinen ja luokittelemaan tapahtuma sen mukaan. (Kovacevic 2023, 22.)

Haitallisen tapahtuman toteamisen jälkeen eristetään ne asiat, joissa on havaittu haitallista toimintaa. Lisäksi hävitetään havaitut haitalliset asiat, kuten haittaohjelmat. Kun on varmistuttu siitä, että kaikesta haitallisesta on päästy eroon, aloitetaan toipuminen normaalitilaan. Tapahtuman jälkeen dokumentoidaan asiat, jotka liittyivät tutkintaan. Dokumentointia voidaan hyödyntää myöhemmissä vastaavanlaisissa tapahtumissa löytämään puutteita tutkintaprosessissa tai koulutusmateriaalina. (Kovacevic 2023, 23–24.)

# 4 FORTISIEM INTEGRAATION HYÖDYNTÄMINEN KÄYTÄNNÖSSÄ

Opinnäytetyö toteutettiin osana Palo Alton Cortex XSOAR -järjestelmän käyttöönottoa. Tarkoituksena oli kartoittaa mahdollisuuksia FortiSIEM-hälytyssääntöjen käsittelyyn XSOAR:ssa olevan integraation avulla ja kokeilla käytännössä niiden toimivuutta. Samalla kerättiin kokemuksia uuden SOAR-järjestelmän mahdollisuuksista.

Aloitushetkellä SIEM oli yksi tietoturvakeskuksen työkaluista asiakkaiden ympäristöjen valvontaan. Tietoturvakeskuksen saadessa hälytyksen SIEM-järjestelmän kautta tarkoitti se lähes poikkeuksetta manuaalista tutkintaa analytiikan avulla järjestelmästä löytyvistä lokeista ja tietojen etsimistä muista tietoturvakeskuksen käyttämistä työkaluista. SOAR:in avulla pystytään keräämään eri järjestelmien tuottamaa tietoa yhteen järjestelmään, jonka avulla pystytään korreloimaan tapahtumia keskenään, sekä tuottamaan toistuvien tehtävien automaatiota. Automaation avulla vapautetaan tietoturva-asiantuntijan työaikaa tärkeämpiin tehtäviin.

Opinnäytetyö jakaantuu kolmeen osaan, joissa perehdytään FortiSIEM-hälytyssääntöihin, XSOAR-toiminnallisuuksiin ja kyseisten ohjelmistojen väliseen integraatioon. Integraation perustana toimii FortiSIEM API. API-dokumentaa-

tion avulla pystyttiin kartoittamaan sen tarjoamia mahdollisuuksia hälytyssääntöjen ylläpitoon. Dokumentaation perusteella tarkkailulistojen (watchlist) käsittely oli mahdollista. Tarkkailulistoja voidaan käyttää useissa käyttötapauksissa, joten API:n tarjoamista ominaisuuksista rajattiin tarkempaan käsittelyyn tarkkailulistojen käsittely API:n avulla.

### 4.1 FortiSIEM

Fortinet osti vuonna 2016 AccelOps-nimisen yhtiön, jonka myötä se sai SIEM-ratkaisun. Fortinet nimesi SIEM-ratkaisun uudelleen FortiSIEM:ksi ja integroi sen osaksi tietoturvatuotteista koostuvaa verkostoaan (CRN 2016). FortiSIEM:in kehitys on jatkunut, ja sen myötä se on saanut nykyaikaisia ominaisuuksia kuten UEBA, generatiivinen tekoäly ja uhkatietokannat.

FortiSIEM:in voi käyttöönottaa palveluntarjoaja (service provider) - tai yritys (enterprise)- versioina. Palveluntarjoaja-versio koostuu yhdestä tai useammasta tenantista, joita voidaan hallita super-tenantin kautta. Yritysversio on yksittäinen tenantti, johon kerääntyy kaikki SIEM:iin tuleva loki. Työtä tehtäessä käytössä oli palveluntarjoajan-versio, joka koostui useista tenanteista. Jokainen tenantti on eriytetty toisistaan. Super-tenantin kautta on mahdollista tehdä kaikkia tenantteja koskevia muutoksia, mutta jokaiselle tenantille on myös mahdollista tehdä tenanttikohtaisia muutoksia.

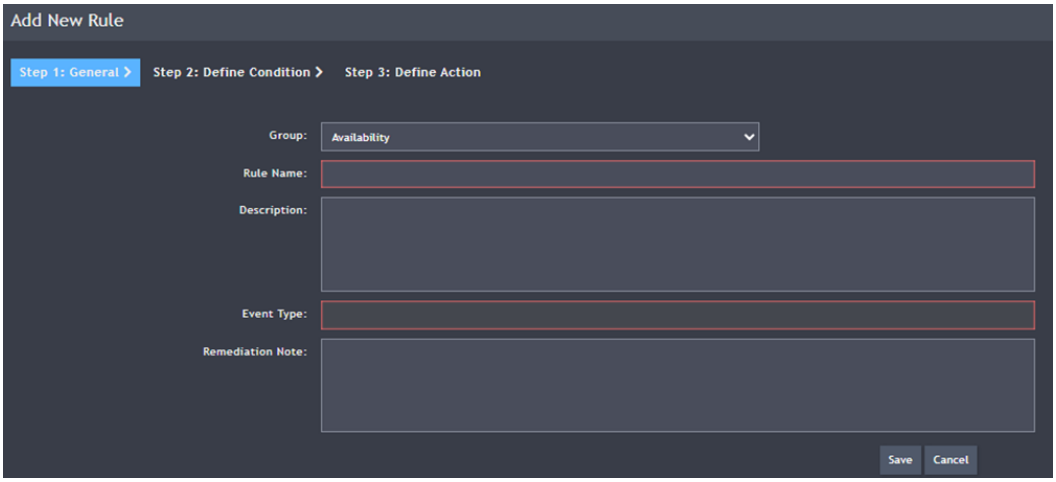
Lokien lisäksi FortiSIEM:iin voidaan kerätä tietoa CMDB:hen (Configuration Management Database) lokilähteinä olevista laitteista, sekä erilaisia resursseja lista muodossa, kuten uhkatietokantoja tai tarkkailulistoja. Lähes kaikkea kerättyä tietoa voidaan tutkia analytiikan avulla, jonka myötä hälytyssäännöissä voidaan käyttää kattavasti kaikkea tarjolla olevaa tietoa.

*Hälytyssäännön toteutus*

Hälytyssääntöjä suunniteltaessa on suositeltavaa miettiä käyttötapaus, jonka mukaan hälytyssääntöä aletaan rakentamaan ja määrittelemään. Jokainen valvottava ympäristönsä on omanlaisensa, joten harvoin on mahdollista käyttää valmiiksi määriteltyä käyttötapausta, joka sopisi omaan ympäristöön.

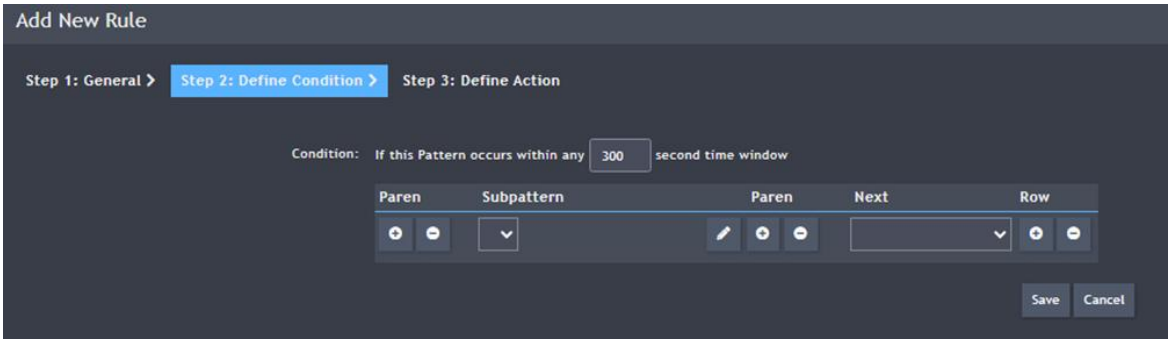
Valmiin käyttötapauksen voi kuitenkin muokata omaan ympäristöönsä sopivaksi. Esimerkiksi käyttötapauksessa, jossa halutaan valvoa kulkevia datamääriä, olisi hyvä määritellä minkälaiset datamäärät ovat normaaleja ja kulkevatko suuret määrät dataa aina samoja reittejä pitkin. Näin pystytään räätälöimään hälytyssääntö omaan ympäristöön sopivaksi, mikä vähentää huomattavasti ns. vääriä hälytyksiä.

Hälytyssääntöjen tekemiseen FortiSIEM tarjoaa kolmivaiheisen määrittelyn. Ensimmäisessä vaiheessa (kuva 3) määritellään yleiset tiedot säännölle. Pakollisina tietoina on säännön nimi ja tapahtuman tyyppi. Mikäli tapahtuman tyyppiä ei määritellä, muodostetaan se automaattisesti säännön nimestä. Analytiikalla sääntötapahtumia haettaessa pystyy itse tehdyt säännöt erottamaan FortiSIEM:in oletussäännöistä sillä, että oletus sääntöjen tyypit alkavat PH_RULE.



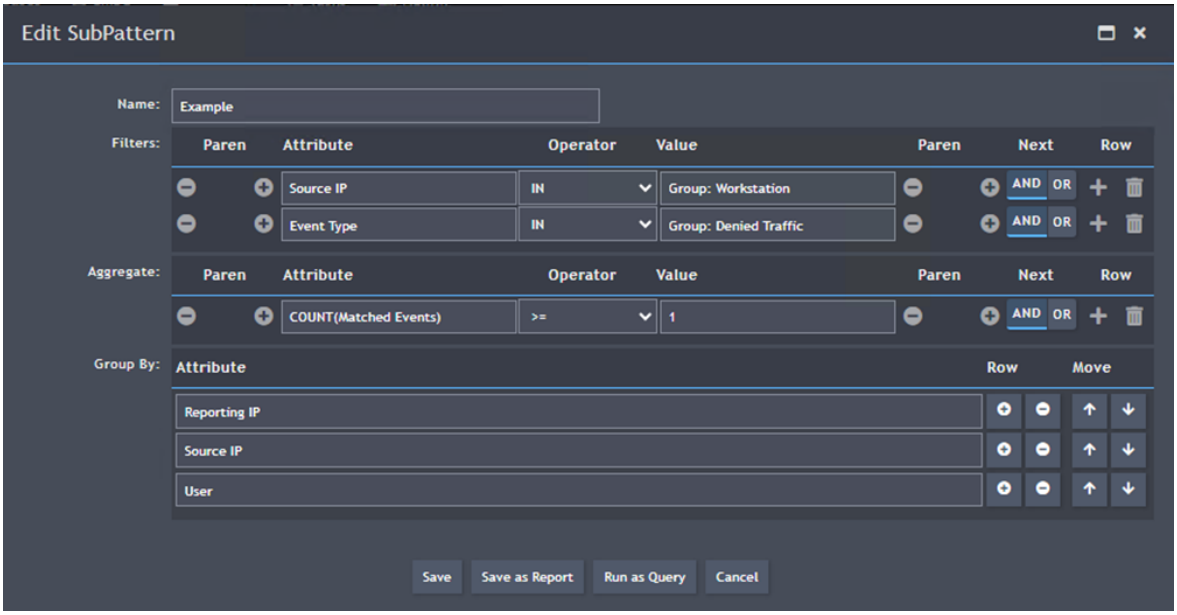

Kuva 3. Hälytyssäännön määrittelyn ensimmäinen vaihe

Kuvassa 4 määritellään ehdot säännölle subpatternien muodossa, sekä aika jonka puitteissa ehdon täytyy täyttyä, jotta hälytys laukeaa. Subpatterneja voi olla useampia ja niiden välille voidaan luoda suhteita ja rajoituksia. Esimerkiksi ensimmäisessä subpatternissa annetaan ilmoitus, jos käyttäjällä on ollut enemmän kuin viisi epäonnistunutta kirjautumista. Toisella subpatternilla määritellään, että hälytys laukeaa, jos epäonnistuneita kirjautumisia seuraa onnistunut kirjautuminen.




Kuva 4. Hälytyssäännön määrittelyn toinen vaihe

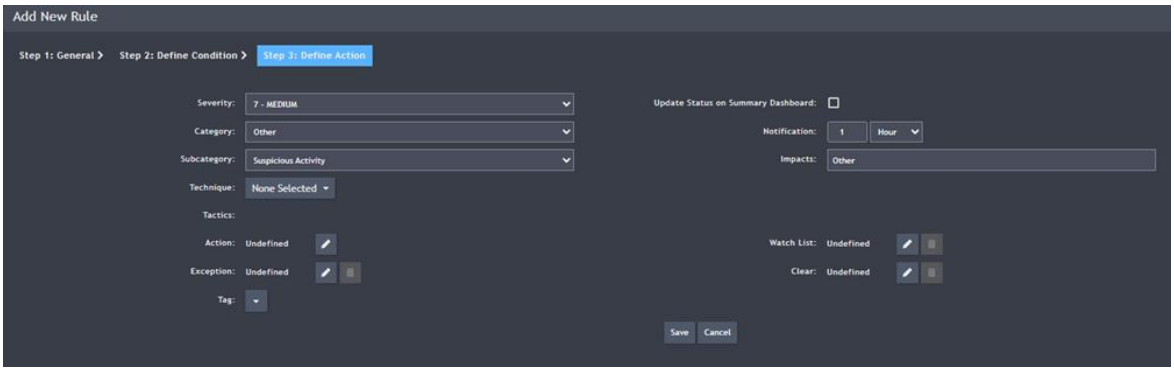
Subpatterniin (kuva 5) itseensä määritellään ehdot säännölle. Suositeltavaa on käyttää mahdollisimman paljon dynaamisia listoja ehtojen määrittelyssä. Dynaamiset listat helpottavat sääntöjen ylläpitoa ja pitävät määrittelyt siistinä. Esimerkiksi jos halutaan valvoa työasemia, voidaan kentän arvoksi määritellä työasema ryhmä. Kun uusi työasema liitetään ryhmään, tulee se samalla automaattisesti hälytyssäännön valvonnan alaiseksi.




Kuva 5. Subpatternin määritykset

Kolmannessa vaiheessa (kuva 6) määritellään tapahtumaan liittyviä tietoja ja toimintoja. Toiminnan (Action) alta pystyy muokkaamaan hälytykselle nousevia kenttiä, sekä itse tapahtuman nimeä. Poikkeuksien (Exception) avulla voidaan lisätä poikkeuksia sääntöön yleisesti tai tenanttikohtaisesti. Yleiset poikkeukset täytyy lisätä super-tenantissa ja tenanttikohtaiset poikkeukset halutussa tenantissa. Tarkkailulistojen (Watch List) avulla voidaan hälytykseen liittyvä entiteetti lisätä tarkkailulistalle tapahtuman sattuessa. Selvityksen (Clear)

avulla voidaan määritellä automaattinen ratkaisu tapahtumalle, jos määritellyt ehdot täyttyvät tapahtuman luonnin jälkeen.




Kuva 6. Hälytyssäännön määrittelyn kolmas vaihe

Määrittelyiden valmistuttua hälytyssääntö on heti käytettävissä ja alkaa tuottamaan tapahtumia, mikäli säännön ehdot täyttyvät ja sääntö on asetettu aktiiviseen tilaan. Itse tehtyjä hälytyssääntöjä voi käydä vapaasti muokkaamassa. Jos on tarvetta muokata oletussääntöä, täytyy se kloonata, jotta muutoksia voi tallentaa.

## 4.2 XSOAR

Palo Alto Networks osti vuona 2019 SOAR-yhtiö Demiston, ja sen myötä se liitti uuden SOAR-järjestelmän tuoteportfolioonsa nimellä XSOAR. XSOAR on osa Cortex-alustaa, joka koostuu tietoturvaan liittyvistä työkaluista. XSOAR toimii alustana, jonka avulla voidaan orkestroida ja automatisoida toimintoja tapahtumiin liittyen. Alustaan voidaan liittää kauppapaikasta (marketplace) sisältöpaketteja, joissa on valmiita näkymiä, pelikirjoja, automaatioita ja integraatioita.

### 4.2.1 XSOAR-järjestelmällä työskentely

Jos valmiista sisältöpaketeista ei löydä haluamaansa, voi kaiken myös tehdä itse. JavaScript, Python ja PowerShell ovat tuettuja ohjelmointikieliä, joiden avulla pystytään tekemään omia kustomoituja automaatioita. Omien integraatioiden tekeminen vaatii osaamista koodaamisesta, sekä integroitavan kohteen API-dokumentaation. Lisäksi olisi hyvä tietää, mihin integroitava järjestelmä on

suunniteltu, jotta ymmärretään mitä ollaan tekemässä. Pelikirjojen rakentaminen muistuttaa visuaalista ohjelmointia, jossa lisätään loogisessa järjestyksessä tehtäviä. YAML- ja JSON-tiedostoja käytetään konfiguroinnissa ja tiedonsiirroissa.

XSOAR tarjoaa hyvin vapaat kädet käyttää alustaa haluamallaan tavalla. Käyttötapausten määrittely on keskeisessä osassa alustan ja automaatioiden kehityksessä. Perusajatuksena on, että tietoa tuodaan sisään tapahtumien muodossa ja ne käsitellään XSOAR:ssa siihen muotoon, että ne tarjoavat käyttäjälle selkeän kuvan tapahtumasta ja mahdollistavat automaation tekemisen.

Työn käytännön osuuden tavoitteena oli tutkia XSOAR:in avulla, kuinka voidaan toteuttaa FortiSIEM-hälytysten ylläpitoa automaation avulla. Lisäksi tavoitteena oli kokeilla FortiSIEM API -rajapinnan tarjoamia mahdollisuuksia hälytyssääntöihin ja muihin FortiSIEM:iin liittyviin osioihin, kuten CMDB ja tarkkailulistat.

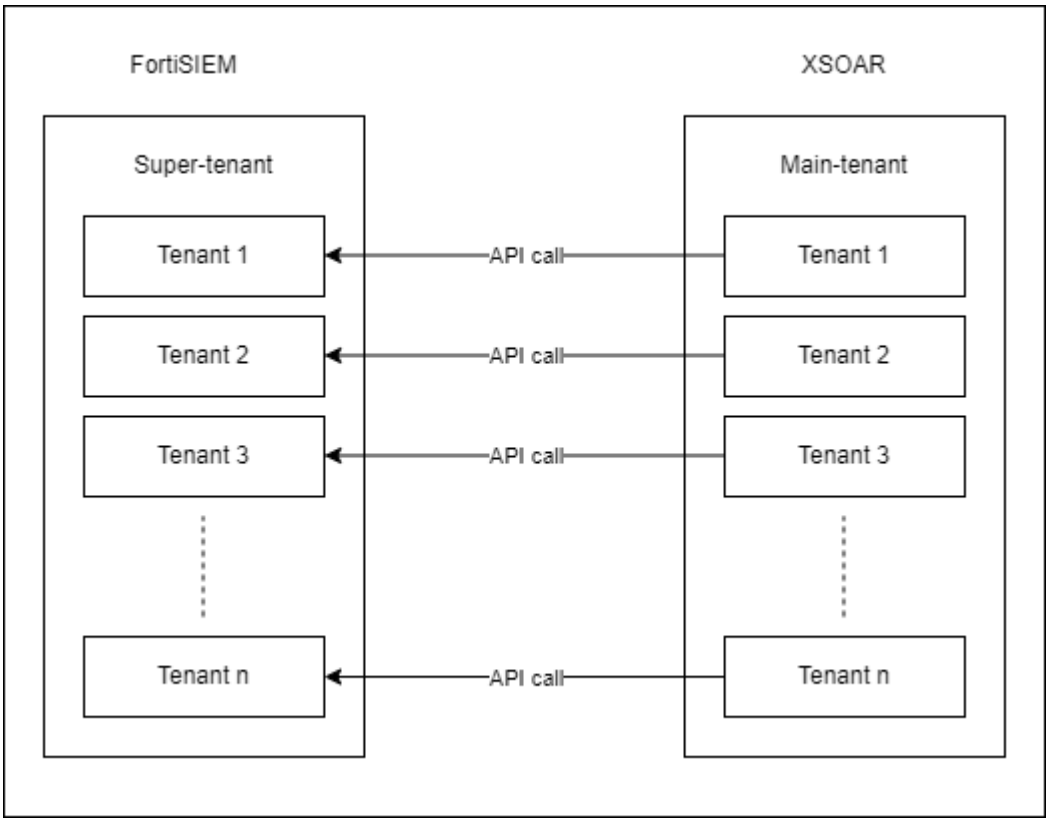
### 4.2.2 XSOAR-Integraatio FortiSIEM:iin

Valmista sisältöä voi XSOAR:iin tuoda kauppapaikasta. Kauppapaikka on keskitetty portaali, jonka kautta pystytään hallinnoimaan XSOAR:ssa olevaa sisältöä. Kauppapaikan sisältö koostuu sisältöpaketeista, joihin on koottu kokonaisuuksia, jotka sisältävät XSOAR:ssa käytettäviä objekteja kuten integraatioita, skriptejä, pelikirjoja jne.

FortiSIEM-integraatiota varten XSOAR-kauppapaikasta löytyy FortiSIEM-sisältöpaketti, jonka mukana tulee integraatio FortiSIEM:iin. Integraation lisäksi sisältöpakettiin kuuluu luokittelija FortiSIEM-tapahtumille, tapahtumakenttiä (Incident Fields), Tapahtumatyyppi (Incident Type), layout FortiSIEM-tyyppisille tapahtumille ja pelikirja FortiSIEM-tapahtumien hakemiseen. Integraatio itsessään sisältää komentoja, joilla voidaan käyttää FortiSIEM API-rajapintaa.

XSOAR:sta ja FortiSIEM:stä on käytössä multi-tenant- eli palveluntarjoajaversio. Tämä tarkoittaa sitä, että jokaiselle asiakkaalle on luotu oma tenantti

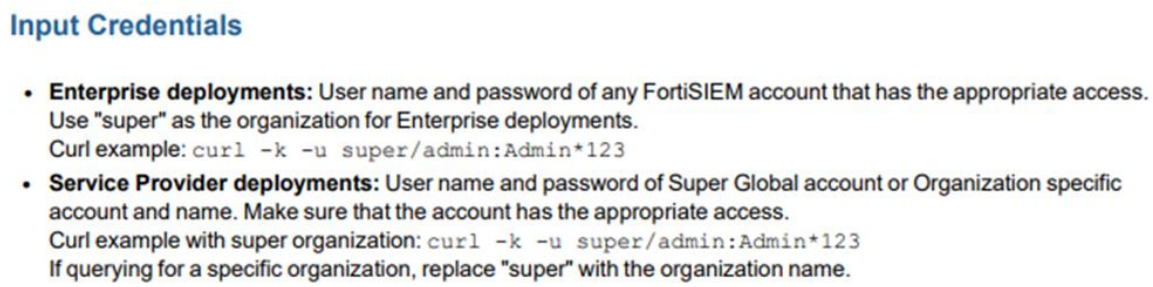
molempiin järjestelmiin, joka on eriytetty muista tenanteista. Integraatio on tehtävä tenanttien välille, jotta niissä olevat tiedot pysyvät vain asiakkaan ympäristöissä. Kuvassa 7 on esitetty arkkitehtuuri XSOAR:in ja FortiSIEM:in välisen integraation osalta. XSOAR käy hakemassa API-rajapinnan kautta tietoja. FortiSIEM:in ei tarvitse erikseen lähettää tietoa mihinkään.




Kuva 7. FortiSIEM:in ja XSOAR:in välinen integraatio

Integraation komentoja käyttääkseen täytyy integraatio konfiguroida. Konfigurointia varten pakollisia tietoja ovat palvelimen osoite, josta tietoa haetaan, sekä käyttäjätunnus. Multi-tenant FortiSIEM:iä käytettäessä tunnukset ovat tenanttikohtaisia. Tunnuksille tarvitsee antaa tarvittavat oikeudet käyttötarkoituksen mukaan. Jos on tarkoitus hakea vain tietoa, riittää lukuoikeudet. Tiedon muokkaamiseen ja lisäämiseen tarvitaan kirjoitusoikeudet.

Integraatiota konfiguroitaessa muodostui ongelma FortiSIEM:iin autentikoitumisen suhteen. FortiSIEM API -ohjeista löytyi ohjeet autentikoimiseen (kuva 8). Käyttäjätunnuksen tulee olla muodossa <organisaatio>/<käyttäjätunnus>.

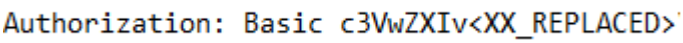


**Input Credentials**

- **Enterprise deployments:** User name and password of any FortiSIEM account that has the appropriate access. Use "super" as the organization for Enterprise deployments.
  Curl example: `curl -k -u super/admin:Admin*123`
- **Service Provider deployments:** User name and password of Super Global account or Organization specific account and name. Make sure that the account has the appropriate access.
  Curl example with super organization: `curl -k -u super/admin:Admin*123`
  If querying for a specific organization, replace "super" with the organization name.

Kuva 8. FortiSIEM API Guide 6.7.4 Autentikointi FortiSIEM:iin

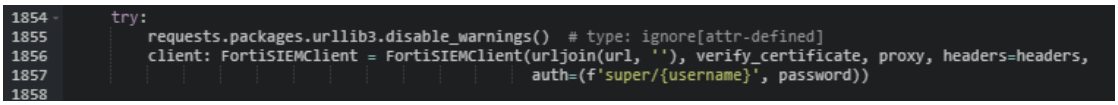
Konfiguroinnin yhteydessä pystytään testaamaan yhteyttä, joka antaa onnistuneesta yhteydestä ilmoituksen ja virheellisestä yhteydestä syyn epäonnistumiselle. Yhteyttä testattaessa voidaan ajaa debug-testi, josta saadaan tarkempaa tietoa virheestä. Debug-testin kautta selvisi, että integraatio lisäsi automaattisesti tunnukseen tekstiä, joka oli base64 koodattua (kuva 9).



```
Authorization: Basic c3VwZXIv<XX_REPLACED>
```

Kuva 9. Kovakoodattu organisaatio autorisoinnissa

Tekstin purku osoitti sen tarkoittavan ”super/”, eli integraatio ohjasi kaikki suoraan super-tenanttiin, joten seuraava vaihe oli käydä integraatiokoodi läpi ja tarkistaa, missä vaiheessa koodia kovakoodaus tapahtuu. Integraatiokoodista löytyi autentikointiin liittyvä koodirivi (Kuva 10), johon tunnistetiedot tulivat.



```
1854    try:
1855        requests.packages.urllib3.disable_warnings()  # type: ignore[attr-defined]
1856        client: FortiSIEMClient = FortiSIEMClient(urljoin(url, ''), verify_certificate, proxy, headers=headers,
1857                                                  auth=(f'super/{username}', password))
1858
```

Kuva 10. Kovakoodattu organisaatio FortiSIEM-integraatiossa

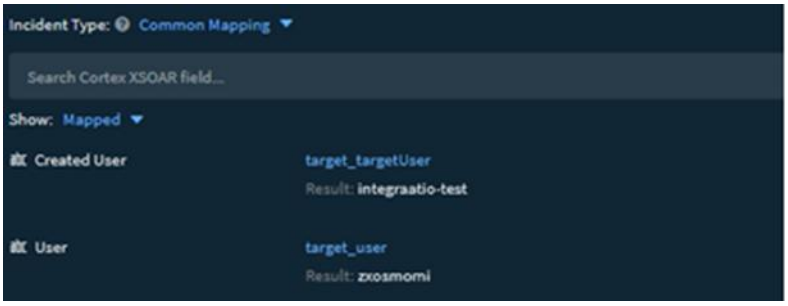
Koodiin oli kovakoodattuna autentikointi FortiSIEM super-tenantille. Palveluntarjoajan versiossa tunnistetiedot pitää antaa muodossa <organisaatio>/<tunnus>, joten integraatiokoodi kloonattiin ja ylimääräiset osat poistettiin, minkä jälkeen integraatio saatiin toimimaan halutulla tavalla.

### 4.2.3 XSOAR-kartoitus, luokittelu ja layout

XSOAR ei osaa automaattisesti käsitellä tietoja, joita se hakee integraatioiden avulla. Integraatiolle voidaan määritellä Incoming Mapper, joka on sisään tulevan tiedon kartoittaja. Kartoittajaan määritellään, mitä sisään tulevasta tiedosta halutaan käyttää. Kuvassa 11 on esitelty, millaiselta integraatioilla haettu data näyttää ennen kartoittamista.


Kuva 11. Integraatiolla haettua dataa

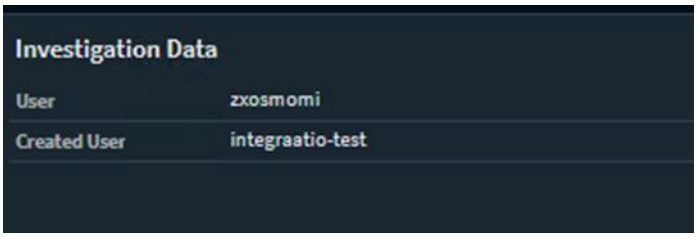
Kartoittajan avulla kartoitetaan tulevasta datasta halutut tiedot tapahtumakenttiin (incident field). XSOAR:ssa on tarjolla useita valmiita tapahtumakenttiä, sekä niitä pystyy luomaan tarpeeseen. Kentät voivat olla tyypiltään erilaisia kuten numero, lyhyt teksti, päivämäärä tai liite. Kartoitusta tehdessä on hyvä tapa kartoittaa tapahtumille yleiset tiedot yleiseen kartoitukseen (Common Mapping) ja tapahtumille ominaiset tiedot tyyppikohtaisesti. Kuvassa 12 on esitelty, kuinka kartoittajaan on tapahtumakentälle osoitettu polku tulevasta datasta, jonka avulla se osaa hakea oikean tiedon.




Kuva 12. Raakadatan kartoitus XSOAR kenttiin

Tapahtumien luokittelu tapahtuu XSOAR:ssa luokittelijan (Classifier) avulla. Luokittelijan avulla voidaan tapahtumalle antaa tapahtumatyyppi (Incident Type). Luokittelun periaate on lähes sama kuin kartoituksessa. Luokittelijalle annetaan tapahtumakenttä, jonka arvoille voidaan määritellä tyyppi. Esimerkiksi FortiSIEM:iin voidaan luoda paljon erilaisia hälytyksiä. Hälytykset tulevat XSOAR:iin yhden integraation kautta. Osa hälytyksistä voi liittyä Brute Forceen, kun taas osa on Lateral Movement -tyyppisiä. Oletuksena voidaan asettaa kaikki integraation kautta tulevat hälytykset FortiSIEM-tyyppisiksi luokittelijalla. Mahdollista on myös hälytyksen nimen perusteella luokitella Brute Force -hälytykset Brute Force -tyyppisiksi ja Lateral Movement -hälytykset Lateral Movement -tyyppisiksi.

Data esitellään käyttäjille layoutien avulla. Layoutilla tuodaan tapahtumatyypille oleelliset tiedot selkeästi esille. Layout voi koostua useista välilehdistä, joilla on omat funktionsa. Välilehdille määritellään osioita, joihin voidaan määritellä esiteltävää tietoa tai lisätä painikkeita erilaisille toiminnallisuuksille, kuten tiedon päivittämiselle, sähköpostin lähettämiselle ja tapahtuman ottamiselle omiin nimiin. Kuvassa 13 on kartoitettu data esiteltynä Investigation Data -osiossa.

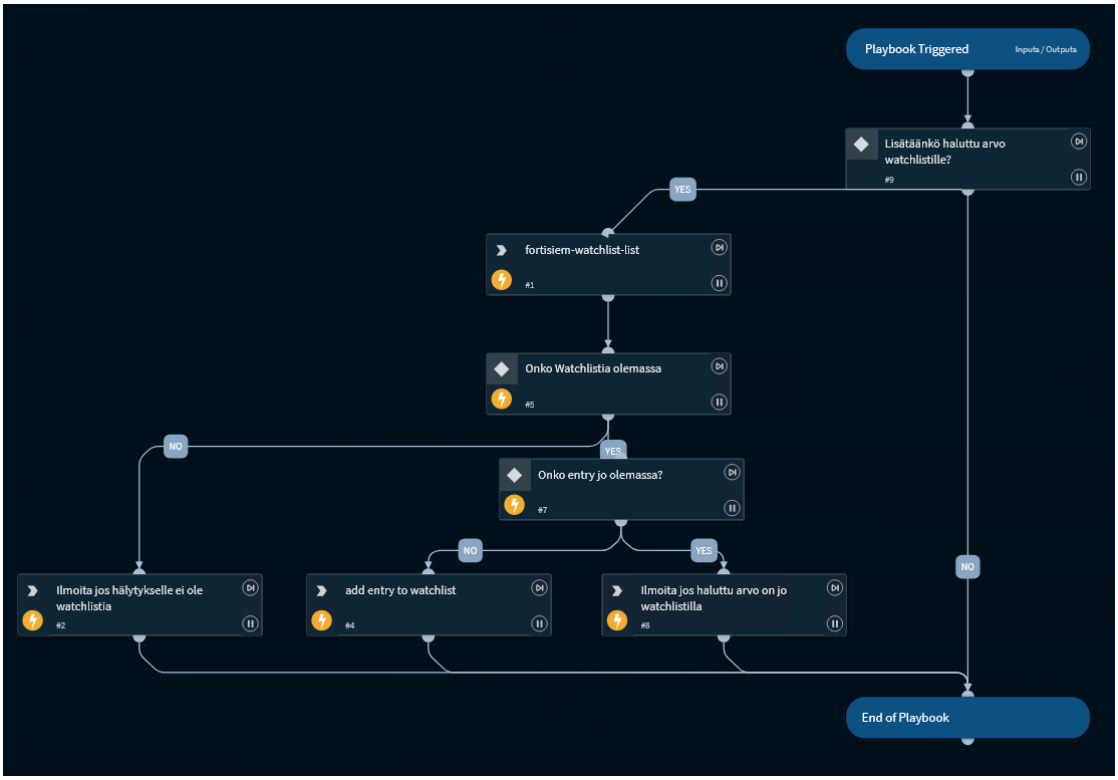



Kuva 13. Kartoitettu data esiteltynä Investigation Data -osiossa

Kartoittaja, luokittelija ja layout ovat yksittäisiä komponentteja XSOAR:ssa, jotka eivät yksinään tee juuri mitään. Kartoittajan ja luokittelijan voi liittää osaksi integraatiota, minkä jälkeen integraatiolla haettu tieto kartoittuu ja luokittuu halutulla tavalla. Layout liitetään osaksi tapahtumatyyppiä, jonka ansiosta voidaan tyypille ominaiset asiat esittää ilman ylimääräistä tietoa.

### 4.2.4 XSOAR-Automaatio

XSOAR automaatio perustuu skripteihin ja ohjelmakoodeihin. Koodia voi kirjoittaa JavaScriptillä, Pythonilla tai PowerShellillä. Skriptit ovat pieniä koodinpätkiä, joilla toteutetaan tarkkaan määritelty toiminto. Ohjelmakoodi on skriptiä laajempi kokonaisuus, jolla voidaan toteuttaa monimutkaisempia toimintoja. Yleisin tapa toteuttaa automaatiota ovat pelikirjat. Ne muodostuvat tehtävistä, joissa voidaan käyttää skriptejä tehtävien suorittamiseksi. Pelikirja muodostaa visuaalisen kaavion, josta selviää tehtävien järjestys ja logiikka (kuva 14).




Kuva 14. Pelikirjan visuaalinen kuvaus

Pelikirja muodostuu erilaisista tehtävistä. Standarditehtävän tarkoitus on ajaa automaatio eli skripti. Automaatiolle voidaan määritellä syötteet, jotka se hakee tapahtuman kontekstista. Suoritettuaan toiminnon automaatio voi lisätä tietoa tapahtuman kontekstiin tutkintaa ja muita automaatioita varten. Ehdollisissa tehtävissä määritellään output-arvo, jolle annetaan ehto, jonka täytyttyä edetään pelikirjassa määriteltyyn suuntaan. Ehdollisissa tehtävissä on myös mahdollista tehdä kysely tietoturva-asiantuntijalle tai lähettää sähköpostilla kysely. Automaatio skriptejä voidaan myös käyttää ehdollisissa tehtävissä. Lisätiedon keräämistä varten on tiedonkeräys (Data Collection) -tehtävä, jonka avulla voidaan kysyä tietoturva-asiantuntijalta lisätietoa tai lähettää sähköpostilla kysely lisätiedoista.

Job on tehtävä, joka suorittaa pelikirjan annettujen parametrien täyttyessä. Laukaisevina tekijöinä voivat olla syötteissä tapahtuvat muutokset tai ajastettu tehtävä. Jobin suorittaminen luo job-kategorian tehtävän XSOAR:iin. Pelikirjoilla on mahdollista myös luoda tapahtumakategorian tehtäviä, jolloin jobin suorittamisesta jää järjestelmään kaksi tapahtumaa.

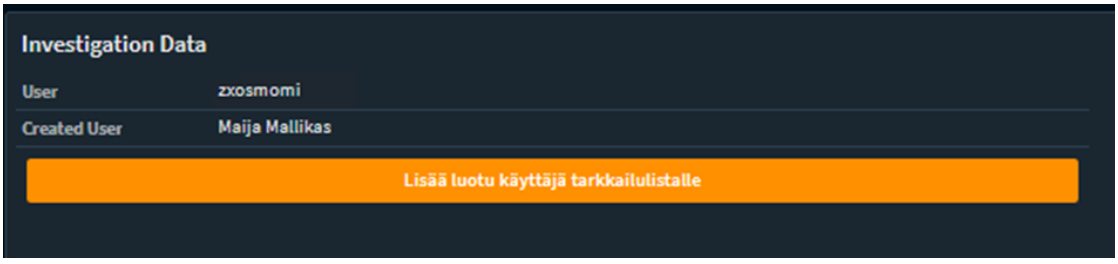
Automaation monipuolisuus tulee esille koodia tehtäessä. Skripteillä voidaan tehdä pieniä komponentteja, joita liitetään suorittamaan erilaisia automaatioon liittyviä tehtäviä, kuten suorittamaan toimintoja tapahtumakentän muutoksen yhteydessä, dynaamisten alueiden täyttämiseen layouteilla tai ajaamaan XSOAR-komentoriviltä ylläpidollisia skriptejä. Koodilla voidaan toteuttaa suurempiakin kokonaisuuksia. Kuvassa 15 on esitetty kuvassa 14 oleva pelikirja koodina.

```
def add_entry_to_watchlist():

    # Kerätään tarvittavat muuttujat
    name = demisto.incident().get("name")
    value = demisto.incident()["CustomFields"]["createduser"]
    message = ""
    the_watchlist = ""

    watchlists = demisto.executeCommand("fortisiem-watchlist-list", {})[0]["Contents"]["response"]

    # Tarkastetaan onko halutun nimistä watchlistia olemassa
    for i in watchlists:
        if i["displayName"] == name:
            the_watchlist = i


    if the_watchlist == "":
        message = "Tarvittavaa watchlistiä ei löydy"

    # Tarkastetaan onko value jo tarkkailulistalla, jos löytyy päivitetään se
    for x in the_watchlist["entries"]:
        if x["entryValue"] == value:
            message = f"{value} arvo löytyy listalta, päivitetään"
            demisto.executeCommand("fortisiem-watchlist-entry-update", {"entry_id":x["id"],"value": value,"description":"XSOAR investigation exclude" })

    # Lisätään value tarkkailulistalle, jos ei ole tullut muita viestejä
    if message == "":
        message = f"Lisätään {value} FortiSIEM listalle {name}"
        demisto.executeCommand("fortisiem-watchlist-entry-add", {"watchlist_id":the_watchlist["id"],"value": value, "description":"XSOAR investigation exclude"})

    return message


# Ajetaan funktio ja printataan sen palauttama viesti war roomiin
print(add_entry_to_watchlist())
```

Kuva 15. Playbook toteutettuna skriptillä

Pelikirja kertoo visuaalisesti, miten tehtävät etenevät, sekä näyttää virheen tapahtuessa sen tehtävän, jonka aikana virhe on tapahtunut. Pelikirjat ovat yleensä liitettynä osaksi tapahtumia, joten tapahtumaa käsittelevä taho näkee pelikirjan. Virheestä raportointi ja virheen paikantaminen helpottuu pelikirjaa käytettäessä. Jotta ohjelmakoodia voitaisiin käyttää pelikirjan tavoin, on siitä tehtävä pelikirja. Tällöin pelikirja muodostuu yhdestä tehtävästä, johon on liitetty haluttu ohjelmakoodi. Virheen tapahtuessa voi olla haastava selvittää se ohjelmakoodin komponentti, joka aiheuttaa virheen.

Layouteille voidaan lisätä painikkeita, joiden avulla skriptejä voidaan suorittaa (kuva 16). Painikkeeseen konfiguroidaan haluttu skripti, jonka seurauksena tapahtuman käsittelijä voi suorittaa ennalta määrätyn toiminnon kyseisen painikkeen painalluksella. Painikkeelle tulisi antaa mahdollisimman hyvin toimintoa kuvaava nimi.




Kuva 16. Layoutilla oleva painike skriptin suorittamiseen

Automaatioita ja skriptejä voidaan suorittaa pelikirjoilla, painikkeilla, kenttien muutoksilla, XSOAR-komentoriviltä, sekä useissa muissa paikoissa. XSOAR:ssa on vapaat kädet koodin suorittamille erilaisille tehtäville. Skripteille, playbookeille ja integraatio komennoille on mahdollista määritellä, kuka niitä voi suorittaa. Onkin suositeltavaa ottaa käytännöksi määritellä, ketkä saavat suorittaa kyseisiä automaatioita. Esimerkiksi administrator-käyttäjät voivat suorittaa kaikkia toimintoja, mutta analyst-käyttäjien suoritukset on estetty. Hekin voivat kuitenkin suorittaa tapahtumiin liittyviä pelikirjoja ja painikkeisiin konfiguroituja skriptejä.

### 4.3 Työn tulokset

Työn tuloksena saatiin tietoa siitä, miten FortiSIEM-hälytyssäännöt rakentuvat ja miten niitä voidaan toteuttaa. Esimerkiksi jos halutaan valvoa tiettyjä laitteita tai IP-osoitteita, kannattaa ne lisätä hälytyssäännölle tarkkailulistan kautta, eikä kirjoittaa itse hälytyssääntöön. Näin listan hallinnointi helpottuu ja hälytyssääntö pysyy selkeämpänä.

XSOAR:in puolelta saatiin tietoa, miten pelikirjat toimivat, mistä ne muodostuvat ja miten niitä voidaan käyttää. Samalla logiikalla kuin tarkkailulistojen päivitys, voidaan toteuttaa tarkastelua muihinkin järjestelmiin. Vaihtoehtoisesti voidaan tarkastella, onko tarkkailulistalla olemassa jo jokin tieto, mikä liittyy toiseen tapahtumaan. Skriptien ja koodien kirjoittamisesta saatiin tietoa esimerkiksi siitä, miten XSOAR demisto-kirjasto toimii ja mitä ominaisuuksia se tarjoaa.

## 5 PÄÄTÄNTÖ

Opinnäytetyön aiheena oli tutkia, onko mahdollista ylläpitää SOAR-järjestelmällä SIEM-hälytyssääntöjä. SIEM-tuotteena oli Fortinetin FortiSIEM ja SOAR-tuotteena Palo Alton Cortex XSOAR. FortiSIEM tarjoaa API-rajapinnan, jonka kautta XSOAR:ssa pystytään ajamaan komentoja. API-dokumentaatio on FortiSIEM:in uudempien versioiden osalta siirretty Fortinetin Develpoer Networkiin, johon voi hakea tunnukset Fortinetiltä. XSOAR tarjoaa valmiin FortiSIEM-integraation, jonka ominaisuuksien hyödyntämiseen työ keskittyi.

Työssä tutustuttiin myös yleisesti SIEM- ja SOAR- järjestelmien ominaisuuksiin ja siihen, kuinka niitä voi hyödyntää käytännössä. Teoriaosuutta kirjoitettaessa kävi ilmi, että molemmista järjestelmillä on omat peruskomponentit ja toiminnot, jotka löytyvät tavalla tai toisella jokaisesta järjestelmästä. Kyberrikollisuuden kasvu on edistänyt molempien järjestelmien kehitystä, minkä myötä järjestelmiin on tuotu nykyaikaisia ominaisuuksia, kuten koneoppimista ja tekoälyä, sekä on alettu hyödyntämään uhkatietoa tehokkaammin. SIEM- ja SOAR-järjestelmiä on kuitenkin olemassa useita ja jokaisessa on toiminnot toteutettu omalla tavallaan. Tämän vuoksi voi olla haastavaa löytää yleispätevää tietoa jokaista järjestelmää koskien.

Työ toteutettiin testiympäristöillä, joiden ansiosta etenkin XSOAR-ominaisuuksien kokeileminen oli huoletonta. Valmiin integraation ominaisuuksista valittiin tarkkailulistat tarkempaan tarkasteluun. Tarkkailulistojen avulla pystyttiin lisäämään suodatuksia hälytyssääntöihin ja luomaan hälytyssääntöjä tarkkailulistoihin perustuen.

Aiemman SIEM kokemuksen perusteella FortiSIEM:ssä oli hyvää selkeästi koottu CMDB ja resurssit, joita pystyi käyttämään vaivattomasti analytiikassa. Käyttökokemuksen perusteella huonoa oli kuitenkin analytiikan kankeus. FortiSIEM:ssä ei ole datalle minkäänlaista selkeää kyselykieltä, kuten yleisesti tunnetut SQL tai KQL. Tottuneelle hakujen kirjoittajalle FortiSIEM-analytiikan toteutus voi tuntua kankealta. Muiden kuin perusfunktioiden haastava käyttö tekee käyttökokemuksesta surkean. Hälytyssäännöt perustuvat analytiikkaan, jonka takia yksinkertaisten hälytyssääntöjen tekeminen on helppoa, mutta monimutkaisemmat toteutukset vaativat syvää osaamista ja luovaa ongelmanratkaisua.

SOAR-järjestelmistä ei ollut aikaisempaa kokemusta, johon olisi voinut verrata. XSOAR:in perustoiminnot koostuivat selkeistä kokonaisuuksista, joita oli helppo käyttää, mutta ne tarjosivat myös paljon erilaisia mahdollisuuksia. Koodia kirjoitettaessa käytettiin python-ohjelmointikieltä. Oman koodin kirjoittaminen mahdollistaa ominaisuuksien toteuttamisen monella eri tavalla. Rajoittavaksi tekijäksi osoittautui tekijän koodaustaidot, eikä toistaiseksi löytynyt muita rajoitteita XSOAR:sta toteutusten suhteen.

Suunniteltu työ saatiin toteutettua mallikkaasti ilman suurempia vaikeuksia. Virhetilanteissa löydettiin ratkaisut hyvinkin nopeasti. XSOAR:n osalta jäi mietityttämään, missä tilanteissa asiat kannattaa toteuttaa pelikirjalla ja missä tilanteissa puhtaasti koodilla. Pelikirjasta näkee visuaalisesti logiikan ja toiminnot, mutta se ei välttämättä ole niin joustava kuin koodi, jolla voi toteuttaa mitä tahansa.

FortiSIEM:in ja XSOAR:in välisen integraation jatkokehityksenä voisi kokeilla ja keksiä käyttötapauksia muillekin ominaisuuksille. Yhtenä valmiin integraation ominaisuutena on event-search, eli analytiikkahakujen tekeminen. Ominaisuutta kokeiltiin, mutta ei saatu toimimaan edes FortiSIEM API-ohjeiden avulla, eikä asiaan panostettu toistaiseksi sen enempää. Jos on pääsy FortiSIEM API-ohjeisiin on mahdollista tehdä myös omia API-kutsuja ja laajentaa valmista komentopatteria.

SIEM ja SOAR täydentävät hyvin toisiaan. SIEM:ssä toteutetaan havainnot suurista lokimassoista ja SOAR:in avulla saadaan automatisoitua ja keskitettyä nousseiden tapahtumien käsittelyä. Useat toimittajat ovatkin valmiiksi integroineet omat SIEM- ja SOAR-järjestelmänsä yhteen, jolloin niitä voidaan käyttää samasta käyttöliittymästä.

**LÄHTEET**

CRN. 2016. Fortinet Dives Into SIEM Market With $28M Acquisition Of AccelOps. WWW-dokumentti. Päivitetty 7.6.2016. Saatavissa: https://www.crn.com/news/security/300080956/fortinet-dives-into-siem-market-with-28m-acquisition-of-accelops [viitattu 27.10.2024].

Finger, D. 2023. Fortinet Advisor Applies the Power of GenAI to SecOps. Blogi. Päivitetty 11.12.2023. Saatavissa: https://www.fortinet.com/blog/business-and-technology/fortinet-advisor-applies-power-of-genai-to-secops [viitattu 27.10.2024].

Fortinet s.a. What is UEBA. Fortinet. WWW-dokumentti. Saatavissa: https://www.fortinet.com/resources/cyberglossary/what-is-ueba [viitattu 27.10.2024].

González-Granidillo, G. González-Zarzosa, S. Diaz, R. 2021. Security Information and Event Management (SIEM): Analysis, Trends, and Usage in Critical Infrastructures. WWW-dokumentti. Saatavissa: https://doi.org/10.3390/s21144759 [viitattu 1.12.2024].

Islam, C. 2020a. Architecture-centric Support for Integrating Security Tools in a Security Orchestration Platform. https://www.researchgate.net/publication/344260727_Architecture-centric_Support_for_Integrating_Security_Tools_in_a_Security_Orchestration_Platform [viitattu 31.12.2024].

Islam, C. 2020b. Architecture-centric support for security orchestration and automation. Adelaiden yliopisto. Väitöskirja. PDF-dokumentti. Saatavissa: http://hdl.handle.net/2440/129206 [viitattu 31.12.2024].

Kidd, C. 2023. SIEM: Security Information & Event Management Explained. Blogi. Päivitetty 12.10.2023. Saatavissa: https://www.splunk.com/en_us/blog/learn/siem-security-information-event-management.html [viitattu 25.3.2024].

Kovacevic, B. 2023. Security Orchestration, Automation, and Response for Security Analysts. Birmingham: Packt Publishing Ltd. E-kirja. Saatavissa: https://www.packtpub.com/product/security-orchestration-automation-and-response-for-security-analysts [viitattu 15.4.2024].

Logpoint. s.a. Top 10 SIEM use cases to implement. WWW-dokumentti. saatavissa: https://www.logpoint.com/en/top-10-use-cases-implement/ [viitattu 5.12.2024].

Miller, D., Harris, S., Harper, A., VanDyke, S & Blask, C. 2010. Security Information and Event Management (SIEM) Implementation. New York: McGraw-Hill. E-kirja. Saatavissa: https://www.mhprofessional.com/security-information-and-event-management-siem-implementation-9780071701099-usa [viitattu 25.3.2024].

Paloalto s.a a. What is Soar. Paloalto Networks. WWW-dokumentti. Saatavissa: https://www.paloaltonetworks.com/cyberpedia/what-is-soar [viitattu 15.4.2024].

Paloalto s.a b. What Is the Role of AI and ML in Modern SIEM Solutions?. Paloalto Networks. WWW-dokumentti. Saatavissa: https://www.paloaltonetworks.com/cyberpedia/role-of-artificial-intelligence-ai-and-machine-learning-ml-in-siem [viitattu 27.10.2024].

SentinelOne. 2024. SIEM Use Cases: Top 10 Use Cases. WWW-dokumentti. Päivitetty 28.8.2024. Saatavissa: https://www.sentinelone.com/cybersecurity-101/data-and-ai/siem-use-cases/ [viitattu 3.12.2024].

Sosa, E. 2022. Is SOAR the tool of the future?, Blogi. Päivitetty 9.3.2022. Saatavissa: https://www.brierandthorn.com/post/soar-the-tool-of-the-future [viitattu 4.1.2025].

Traficom. 2023. Näin keräät ja käytät lokitietoja. WWW-dokumentti. Päivitetty 6.3.2023. Saatavissa: https://www.kyberturvallisuuskeskus.fi/fi/ajankohtaista/ohjeet-ja-oppaat/nain-keraat-ja-kaytat-lokitietoja [viitattu 25.3.2024].