Opinnäytetyö (AMK)

Tieto- ja viestintätekniikka

2023

Teppo Salonen

Tietoturvavalvomoarkkitehtuurin rakentaminen Docker-konttiteknologialla

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

Opinnäytetyö (AMK) | Tiivistelmä

Turun ammattikorkeakoulu

Tieto- ja viestintätekniikka

2023 | 66 sivua

Teppo Salonen

Tietoturvavalvomoarkkitehtuurin rakentaminen Docker-konttiteknologialla

Opinnäytetyössä rakennettiin tietoturvavalvomon (Security Operations Center, SOC) arkkitehtuuri konttiteknologialla. Työn aluksi suunniteltiin SOCin arkkitehtuuri ja esiteltiin yleiskatsaus SOCiin, jonka jälkeen syvennyttiin konttiteknologian perusteisiin ja työkaluihin. Konttipohjaisen SOCin toteutus eteni, siten että esiteltiin tarkemmin Proxmoxia ja Ubuntu palvelinta hyödyntävän virtualisoidun ympäristön asennusta, joka tarjosi vakaan ja turvallisen alustan Docker-konttien ajamiseen. Tämän jälkeen asennettiin Docker ja Portainer, jotka ovat helppokäyttöiset konttien hallintatyökalut, konttien orkestrointia ja hallintaa varten. SOC-komponenttien Docker Compose tiedostojen asennus ja määritys, joihin Elasticsearch, Kibana, Logstash, Filebeat, MISP ja Suricata on konfiguroitu toimimaan yhdessä. Tutkittiin konttipohjaisen tietoturvavalvomon käyttöönottoon liittyvää prosessia ja esiteltiin konttipohjaisen tekniikan tuomia etuuksia ja haasteita SOC-toiminnassa sekä vertailtiin kontitettua SOCia perinteiseen SOCiin.

Asiasanat:

Docker, Konttiteknologia, Tietoturva

Bachelor’s Thesis | Abstract

Turku University of Applied Sciences

Information and Communications Technology

2023 | 66 pages

Teppo Salonen

Building a Security Operations Center Architecture using Docker-Container Technology

In the thesis, the architecture of the security operations center (SOC) was built using container technology. At the beginning of the work, the architecture of the SOC was planned, and an overview of the SOC was presented. After that, we delved into the basics and tools of container technology. The implementation of the container-based SOC progressed, so that the installation of a virtualized environment utilizing Proxmox and the Ubuntu server virtual machine, which provided a stable and secure platform for running Docker containers, was presented in more detail. Installation of Docker and Portainer, easy-to-use container management tools for container orchestration and management. SOC components installation and configuration of log management stack, threat intelligence sharing stack, and intrusion detection and prevention stack, where Elasticsearch, Kibana, Logstash, Filebeat, MISP, and Suricata are configured to work together. The process related to the implementation of a container-based data security controller was studied and the advantages and challenges brought by container-based technology in SOC operations were presented, and a containerized SOC was compared to a traditional SOC.

Keywords:

Docker, Container technology, cyber security

Sisältö

Kuvat

Lyhenteet

DPI Deep Packet Inspection Menetelmä, jolla tutkitaan datapakettien sisältöä niiden kulkiessa verkon tarkistuspisteen kautta. (Larsson 2022)

GUI Graphical User Interface Käyttöliittymä, jonka avulla käyttäjä on vuorovaikutuksessa elektronisten laitteiden, kuten tietokoneiden kanssa kuvakkeiden, valikoiden ja muiden visuaalisten indikaattoreiden tai esitysten avulla. (Computer hope 2021)

IaC Infrastructure as Code Infrastruktuurin hallinta ja käyttöönotto koodin avulla manuaalisten prosessien sijaan. (IBM n.d.b)

IDPS Intrusion detection and prevention system Järjestelmä, joka valvoo verkkoa ja skannaa sen mahdollisten uhkien havaitsemiseksi ja varoittaa järjestelmänvalvojaa ja torjuu mahdollisia hyökkäyksiä. (Mohanakrishnan 2022)

IDS Intrusion detection system Laite tai ohjelmisto, joka valvoo verkkoa tai järjestelmiä haitallisen toiminnan tai käytäntöjen rikkomisen varalta. (Fortinet 2023b)

IoC Indicator of Compromise Näyttö siitä, että joku on saattanut murtautua organisaation verkkoon tai päätelaitteeseen. (Logsign 2019)

IPS Intrusion prevention system Verkkoturvallisuustyökalu, joka valvoo jatkuvasti verkkoa haitallisen toiminnan varalta ja toimii sen estämiseksi. (Fortinet 2023c)

KVM Kernel-based Virtual Machine Linux-kernelin Virtualisointimoduuli, jonka avulla kerneli voi toimia virtualisointialustana. (Proxmox 2023)

LDAP Lightweight directory access protocol protokolla, jonka avulla sovellukset voivat kysyä käyttäjätietoja nopeasti. (Portainer n.d.b)

LXC Linux Containers Käyttöjärjestelmätason virtualisointimenetelmä useiden eristettyjen Linux-järjestelmien käyttämiseksi. (Proxmox 2023)

NAS Network Attached Storage Verkkoon liitetty tallennuslaite, joka mahdollistaa tietojen tallentamisen ja hakemisen keskitetystä paikasta auktorisoitujen verkon käyttäjien ja erilaisten asiakkaiden käyttöön. (Seagate 2023)

Proxmox VE Proxmox Virtual Environment Palvelimien hallintaalusta virtualisointia varten. (Proxmox 2023)

RBAC Role-based access control Menetelmä tietokoneen tai verkon resurssien käytön hallitsemiseksi käyttäjien roolien mukaisesti. (Portainer n.d.b)

SAN Storage area network Dedikoitu nopea verkko, joka mahdollistaa tallennuslaitteiden käytettävyyden palvelimille liittämällä tallennustilan suoraan käyttöjärjestelmään. (Techtarget n.d.b)

SIEM Security information and event management Ratkaisukokonaisuus, joka auttaa organisaatioita havaitsemaan, analysoimaan ja reagoimaan tietoturvauhkiin ennen kuin ne aiheuttavat vahinkoa operatiiviselle toiminnalle. (Salinas 2023)

SOC Tietoturvavalvomo keskitetty yksikkö tai ryhmä, joka vastaa organisaation kyberturvallisuuden parantamisesta sekä uhkien ehkäisemisestä, havaitsemisesta ja niihin vastaamisesta. (Salinas 2023)

VLAN virtual local area network Looginen kerrostettu verkko, joka ryhmittää yhteen osajoukon laitteita, jotka jakavat fyysisen lähiverkon, ja eristää kunkin ryhmän liikenteen. (Techtarget n.d.c)

VM Virtual Machine Tietokoneresurssi, joka käyttää fyysisen tietokoneen sijasta ohjelmistojärjestelmää ohjelmien suorittamiseen ja sovellusten käyttöönottoon. (VMware 2023)

1 Johdanto

Nykyaikaisessa digitaalisessa ympäristössä organisaatiot joutuvat kohtaamaan yhä useampia kyberturvallisuusuhkia, jotka aiheuttavat merkittäviä riskejä niiden tiedoille, toiminnoille ja maineelle. Kyberhyökkäykset ovat kehittyneet yhä pidemmälle, ja uhkaajat käyttävät kehittyneitä tekniikoita hyödyntääkseen verkkojen, järjestelmien ja sovellusten haavoittuvuuksia. Onnistuneiden hyökkäysten seuraukset voivat olla tuhoisia ja johtaa taloudellisiin tappioihin, oikeudellisiin vastuisiin sekä vakavaan vahinkoon organisaation brändille ja asiakkaiden luottamukselle. (Teceze 2021.)

SOCit ovat nousseet keskeisiksi osiksi organisaation kyberturvallisuusstrategiaa. Kyberturvallisuusuhkien nopeasti kehittyvässä ympäristössä organisaatiot kohtaavat yhä kehittyneempiä hyökkäyksiä ja tietomurtoja. Luotettavan ja tehokkaan SOCin tarve on tullut ensiarvoisen tärkeäksi, jotta tietoturvapoikkeamat voidaan havaita, analysoida ja niihin voidaan reagoida ajoissa. Perinteiset SOC-arkkitehtuurit edellyttävät usein merkittäviä laitteistoinvestointeja ja monimutkaisia kokoonpanoja, mikä tekee niiden skaalaamisesta ja ylläpidosta haastavaa. (Salinas 2023.)

Näiden haasteiden ratkaisemiseksi tässä opinnäytetyössä tutkittiin SOC-arkkitehtuurin toteuttamista Docker-konttiteknologialla, jossa kontteina käytettiin Portaineria konttien hallintaan, Elasticsearchia haku- ja analyysimoottorina, Kibanaa datan visualisoinnissa, Logstashia lokien käsittelyssä, Filebeatia lokien välittäjänä, MISPiä käytettiin uhkatiedustelutiedon jakamiseen ja Suricataa tunkeutumisen havaitsemisessa ja estämisessä.

1.1 Perinteisten SOC-arkkitehtuurien haasteet

Perinteiset SOC-arkkitehtuurit ovat tyypillisesti perustuneet tietoturvatyökalujen ja -palveluiden käyttöönottoon erillisillä fyysisillä tai virtuaalisilla palvelimilla

(Salinas 2023). Vaikka nämä arkkitehtuurit ovat toimivia, niihin liittyy useita haasteita. Perinteisen tietoturvavalvomoninfrastruktuurin laajentaminen kasvavan tietoturvatietomäärän mukaan voi olla monimutkaista ja kallista, ja se edellyttää usein merkittäviä laitteistoinvestointeja ja laajoja konfiguraatioita (Salinas 2023). Perinteiset SOC-asetelmat voivat vaatia huomattavia resursseja, mikä johtaa korkeisiin käyttökustannuksiin (Fortinet n.d). Uusien tietoturvatyökalujen käyttöönotto tai nykyisten päivittäminen perinteisissä SOC-järjestelmissä voi olla aikaa vievää (Iiot-world 2022). Useiden tietoturvatyökalujen ylläpito ja päivittäminen eri palvelimilla voi olla hankalaa (Iiot-world 2022). Perinteiset SOC-arkkitehtuurit eivät välttämättä pysty mukautumaan nopeasti muuttuviin turvallisuusvaatimuksiin ja kehittyviin uhkakuviin (Iiot-world 2022).

1.2 Tavoitteet

Tämän opinnäytetyön ensisijaiset tavoitteet olivat suunnitella SOC-arkkitehtuuri Docker- konttiteknologiaa käyttäen, toteuttaa sekä konfiguroida SOC-arkkitehtuurissa olennaiset konttikohtaiset tietoturvatyökalut. Työssä tutkittiin konttipohjaisten työkalujen integrointia ja yhteen toimivuutta tiedonkulun virtaviivaistamiseksi ja uhkien havaitsemis- ja reagointivalmiuksien parantamiseksi ja esitettiin käytännön käyttötapaustutkimus, joka osoittaa konttipohjaisen tietoturvavalvomon tehokkuuden vaaratilanteiden havaitsemisessa, uhkatiedon jakamisessa ja lokianalyysissä.

1.3 Tutkimuksen laajuus ja rajoitukset

Tässä opinnäytetyössä keskityttiin Docker-konttiteknologiaa käyttävän SOC arkkitehtuurin suunnitteluun, toteutukseen ja arviointiin. Vaikka Docker on tähän tutkimukseen valittu konttialusta, käsitellyt käsitteet ja periaatteet voidaan

toteuttaa muissakin konttiteknologiossa. SOC-arkkitehtuuriin sisällytetään tiettyjä konttikohtaisia työkaluja tutkimustavoitteiden saavuttamiseksi.

Tässä opinnäytetyössä ei kuitenkaan pyritty kattamaan kaikkia mahdollisia markkinoilla saatavilla olevia tietoturvatyökaluja tai tarjoamaan tyhjentävää vertailua eri konttialustojen välillä. Siinä tiedostettiin myös, että mikään tietoturvajärjestelmä ei voi olla täysin immuuni kaikille uhkille, ja pääpaino onkin yleisen tietoturvatilanteen ja häiriötilanteisiin reagointikyvyn parantamisessa.

1.4 Opinnäytetyön rakenne

SOC-arkkitehtuurin suunnittelussa käsitellään Docker- SOC-arkkitehtuurin rakentamiseen liittyviä vaatimuksia, korkean tason suunnittelua ja turvallisuusnäkökohtia. Seuraavassa osassa luodaan yleiskatsaus tietoturvavalvomoihin, Docker- konttitekniikkaan sekä katsaus SOC-arkkitehtuurissa käytettyihin konttityökaluihin. Tietoturvatyökalujen kontitus osiossa kuvataan kunkin keskeisen tietoturvatyökalun ja niiden kokoonpanojen kontitusprosessi sekä käsitellään konttipohjaisten työkalujen integrointia, jonka jälkeen tehdään käyttötapaustutkimus, jossa käydään läpi työkalujen asennus ja konfigurointi. Lopuksi käsitellään Docker-pohjaisen SOC-arkkitehtuurin etuja, haasteita sekä lopuksi yhteenveto tutkimuksen tuloksista, panoksesta ja käytännön vaikutuksista.

1.5 Menetelmät

Tässä osiossa perehdytään opinnäytetyön menetelmiin, joita käytettiin työn rakentamiseen, kehittämiseen ja arviointiin SOC-arkkitehtuurissa käyttäen Docker-konttiteknologiaa ja integroituja työkaluja. Nämä määrittävät eri vaiheet, kuten järjestelmäarkkitehtuurin rakentamisen, konttien integroinnin ja

konfiguroinnin, tiedonkeruun ja toteutukseen liittyvät näkökohdat, käyttöönoton ja arvioinnin. Arkkitehtuurissa määritellään tiedonkulku, mukaan lukien keräys-, käsittely-, tallennus- ja visualisointivaiheet. Siinä määritellään myös kunkin työkalun rooli SOC-työnkulussa ja niiden käyttöliittymät.

Kontitus strategiaan kuuluu kunkin työkalun pakkaaminen Docker-kontteihin. Tässä korostetaan työkalujen eristämistä ristiriitojen välttämiseksi ja niiden skaalautuvuuden ja siirrettävyyden varmistamiseksi. Docker Composea käytetään paikalliseen kehitykseen ja testaukseen. Keskitytään kunkin työkalun konttien konfigurointiin, jotta kommunikaatio ja tiedonvaihto on mahdollista. Integrointiin kuuluu yhteyksien luominen, datamuotojen määrittäminen ja tarvittaessa todennusmekanismien asettaminen.

Asianmukaisella konfiguroinnilla varmistetaan koko SOC-arkkitehtuurin yhteensopivuus ja johdonmukaisuus. Toteutusta koskevissa huomioissa käsitellään konttipohjaisten SOC-työkalujen käyttöönottoon liittyviä haasteita ja huomioita. Käyttöönotto käsittää koko konttipohjaisen SOC-arkkitehtuurin käyttöönoton. Tähän sisältyy tarvittavan infrastruktuurin perustaminen, verkkoliikenteen konfigurointi ja konttien asianmukaisen suorittamisen varmistaminen.

2 SOC-arkkitehtuurin suunnittelu

SOC-arkkitehtuurin suunnittelussa luodaan yleiskatsaus integroituun SOC-ekosysteemiin ja korostetaan konttipohjaisten työkalujen rooleja ja vuorovaikutusta. Suunnittelussa hahmotellaan SOC-arkkitehtuurin keskeiset ominaisuudet ja toiminnot ja osoitetaan, miten se parantaa uhkien havaitsemista, lokien hallintaa ja uhkatiedon jakamista kyberturvallisuustoimintojen vahvistamiseksi.

Integroitu SOC-ekosysteemi koostuu toisiinsa kytketyistä konttipohjaisista työkaluista, jotka toimivat yhdessä SOCin tavoitteiden saavuttamiseksi. Portainer on konttien hallintatyökalu, joka yksinkertaistaa muiden konttien käyttöönottoa ja hallintaa (Portainer 2023a). Elasticsearch toimii hajautettuna haku- ja analyysimoottorina, joka indeksoi Filebeatin keräämät ja Logstashin käsittelemät lokitiedot (Elasticsearch 2023).

Kibana tarjoaa web-käyttöliittymän indeksoitujen lokitietojen tutkimiseen ja visualisointiin reaaliajassa (Kibana 2023). Logstash toimii datankäsittelyputkistona, joka kerää lokitiedot Filebeatista, rikastaa ne lisätiedoilla ja välittää ne Elasticsearchiin (Logstash 2023), Filebeat kerää lokitiedot palvelimilta ja lähettää ne Logstashiin käsiteltäväksi (Filebeat 2023a). MISP helpottaa uhkatiedustelutietojen jakamista ja vaihtamista luotettavien kumppaneiden ja vertaisten kanssa (MISP n.d). Tunkeutumisen havaitsemis- ja torjuntajärjestelmäksi (IDPS) (Mohanakrishnan. 2022) valittu Suricata valvoo verkkoliikennettä reaaliaikaisesti, havaitsee mahdolliset tietoturvauhat ja lieventää niitä tehostaakseen SOCin ennakoivaa puolustusta (Suricata 2023d)

Filebeat kerää lokitiedot palvelimilta ja välittää ne Docker-verkon kautta Logstashiin. Logstash käsittelee ja rikastaa lokitiedot ja valmistelee ne Elasticsearchin indeksointia varten. Elasticsearch indeksoi lokitiedot, ja Kibana tarjoaa visualisointeja ja dashboardeja reaaliaikaiseen lokianalyysiin. MISP jakaa strukturoitua uhkatiedustelutietoa Suricatan ja muiden SOC-työkalujen kanssa ja rikastuttaa lokitietoja asiaankuuluvalla uhkakontekstilla. Suricata

valvoo verkkoliikennettä ja havaitsee mahdolliset uhat tunnettujen signatuuri- ja poikkeavuuksien tunnistamismenetelmien perusteella.

käyttöönottostrategia, jota käytetään SOC-arkkitehtuurissa Docker-pohjaisen konttiteknologian työkalujen, käyttöön ja hallintaan. Strategiassa käsitellään konttien käyttöönottoon liittyviä asioita, kuten näköistiedostojen valintaa, resurssien jakamista ja konttien verkottamista, joilla varmistetaan luotettava SOC-ekosysteemi.

Kontti näköistiedoston valintaprosessissa valitaan sopivat Docker-näköistiedostot SOC-komponenteille. Konttityökaluja varten suositaan virallisia Docker-näköistiedostoja hyvämaineisista lähteistä. Viralliset näköistiedostot ovat hyvin ylläpidettyjä, säännöllisesti päivitettyjä ja niissä on asianmukainen dokumentaatio. Näköistiedostot luotettavista Docker-arkistoista tai vastaavien konttityökalujen virallisista arkistoista valitaan turvallisuuden ja luotettavuuden varmistamiseksi. Joissakin tapauksissa voidaan luoda tai muokata mukautettuja Docker näköistiedostoja SOC:n erityisvaatimusten täyttämiseksi, esimerkiksi lisäämällä mukautettuja kokoonpanoja tai lisäosia. (Docker 2023.)

Kunkin konttityökalun suorittimen ja muistin vaatimukset määritetään niiden käsittelytarpeiden ja resurssien käytön perusteella. Riittävä määrä suorittimen ytimiä ja muistia jaetaan, jotta työmäärä voidaan käsitellä sujuvasti. Kontit määritetään joustaviksi, mikä mahdollistaa dynaamisen skaalauksen kysynnän mukaan. Tämän joustavuuden ansiosta SOC-ekosysteemi pystyy käsittelemään vaihtelevia työmääriä ilman resurssipulaa. Kullekin kontille asetetaan resurssirajat, joilla estetään resurssien ylikäyttö ja varmistetaan tasapuolinen jako konttien kesken. (Docker Docs 2023a.)

Konttien yhdistämiseksi luodaan mukautetut Docker-verkot, joilla varmistetaan turvallinen ja eristetty viestintä konttityökalujen välillä. Vain välttämättömät portit altistetaan ulkoisesti SOC-isäntäympäristölle turvallisen ulkoisen käytön varmistamiseksi, kun taas muihin sisäisiin portteihin pääsee käsiksi vain mukautettujen Docker-verkkojen sisällä. Konttityökalut, jotka vaativat pääsyä tiettyihin isäntäkoneen resursseihin, kuten verkkoliitäntöihin tai

järjestelmälokeihin, määritetään asianmukaisilla isäntäkoneen yhteysasetuksilla. (Docker Docs 2023b.)

3 Kirjallisuuskatsaus

Kirjallisuuskatsauksessa tarkastellaan SOC- ja Docker-pohjaiseen konttiteknologiaan liittyviä aiheita kyberturvallisuuden yhteydessä. Tämän katsauksen tarkoituksena on luoda kattava käsitys SOC-keskusten roolista nykyaikaisissa kyberturvallisuusoperaatioissa, konttipohjaisuuden hyödyistä ohjelmistokehityksessä ja -käyttöönotossa sekä konttipohjaisten työkalujen käytön merkityksestä SOC-keskusten valmiuksien parantamisessa.

3.1 Yleiskatsaus SOCiin

Nykyisessä nopeasti kehittyvässä digitaalisessa ympäristössä organisaatiot kohtaavat yhä enemmän kehittyneitä kyberuhkia. Tämän seurauksena SOCin tarve on tullut ensiarvoisen tärkeäksi. SOC on erityinen laitos tai tiimi, joka vastaa organisaation verkon ja tietojärjestelmien tietoturvaloukkausten valvonnasta, havaitsemisesta, analysoinnista ja niihin vastaamisesta. SOC toimii 24/7 varmistaakseen jatkuvan seurannan ja reagoinnin tietoturvatapahtumiin ja vaaratilanteisiin. Sen henkilökuntaan kuuluu ammattitaitoisia turvallisuusanalyytikkoja, vaaratilanteisiin vastaavia henkilöitä, uhkien metsästäjiä, rikosteknisiä tutkijoita ja SOC-managereita, jotka työskentelevät yhteistyössä organisaation resurssien suojaamiseksi ja nopean reagoinnin mahdollistamiseksi uusiin uhkiin. (Techtarget n.d.)

SOC koostuu tyypillisesti useista keskeisistä komponenteista ja suorittaa erilaisia toimintoja tehokkaiden turvatoimien varmistamiseksi. Nämä komponentit ovat SIEM (Security Information and Event Management) (Salinas 2023) -järjestelmät, mitkä ovat SOC-järjestelmissä käytetty ydinteknologia, jolla kerätään, korreloidaan ja analysoidaan eri lähteistä peräisin olevia tietoturvatapahtumalokeja ja -tietoja. Ne auttavat tunnistamaan mahdolliset tietoturvaloukkaukset ja tarjoavat reaaliaikaisen näkyvyyden verkkotoimintaan. (Salinas 2023.)

Poikkeamienhallintaryhmä, joka vastaa tietoturvaloukkausten tutkimisesta ja niihin vastaamisesta. He työskentelevät tiiviissä yhteistyössä SOC-analyytikoiden kanssa uhkien hillitsemiseksi ja korjaamiseksi, riskien lieventämiseksi ja tietoturvaloukkausten vaikutusten minimoimiseksi. Tukeutuu uhkatiedusteluun, jotta he voi ennakoivasti tunnistaa uudet uhat, haavoittuvuudet ja hyökkäysmallit. Nämä tiedot auttavat SOC-analyytikkoja pysymään mahdollisten uhkien edellä ja vahvistamaan organisaation tietoturvaa. He valvovat jatkuvasti verkkoliikennettä, järjestelmälokeja ja tietoturvatapahtumia epäilyttävien toimintojen, poikkeamien ja vaaratekijöiden tunnistamiseksi. Tämä ennakoiva seuranta mahdollistaa mahdollisten tietoturvaloukkausten oikea-aikaisen havaitsemisen ja niihin reagoimisen. (IBM n.d.)

SOC Manageri valvoo SOC:n kokonaistoimintaa, asettaa strategiset linjaukset ja varmistaa, että se on linjassa tavoitteiden kanssa. Hän hallinnoi resursseja ja budjetointia ja määrittelee keskeiset suorituskykyindikaattorit, joilla mitataan SOC:n tehokkuutta. (Salinas 2023.)

Uhkatiedusteluanalyytikot, jotka tutkivat ja analysoivat uhkatiedon syötteitä, avoimen lähdekoodin tietoja ja sisäisiä turvallisuustietoja mahdollisten uhkien, uusien suuntausten ja vaarantumisindikaattoreiden tunnistamiseksi. He tuottavat käyttökelpoisia tiedustelutietoja, joilla parannetaan vaaratilanteiden havaitsemis- ja reagointivalmiuksia. (Salinas 2023.)

SOC-analyytikot, jotka tutkivat tietoturvaloukkauksia analysoimalla lokitietoja, tekemällä rikosteknisiä tutkimuksia ja hyödyntämällä uhkatietoja. He arvioivat vaaratilanteiden vakavuuden, vaikutukset ja perimmäiset syyt määrittääkseen asianmukaiset vastatoimet. Koordinoi vaaratilanteisiin vastaamista ja tekee tiivistä yhteistyötä IT-tiimien kanssa tietoturvaloukkauksien rajoittamiseksi, lieventämiseksi ja korjaamiseksi. Tähän kuuluu järjestelmän eristäminen, korjausten ja päivitysten käyttöönotto sekä vastatoimien toteuttaminen tulevien vaaratilanteiden estämiseksi. (Salinas 2023.)

SOCin ensisijainen tarkoitus on suojata organisaation digitaalista omaisuutta, tietoja ja infrastruktuuria mahdollisilta uhkilta ja hyökkäyksiltä. SOCin päätavoitteita ovat verkko- ja järjestelmätoimintojen jatkuva seuranta epätavallisen tai epäilyttävän käyttäytymisen tunnistamiseksi, joka voi viitata tietoturvaloukkaukseen. SOCin tietoturvatyökalujen ja uhkatiedustelun käyttäminen mahdollisten tietoturvauhkien, kuten haittaohjelmatartuntojen, tietomurtojen ja luvattomien pääsy yritysten havaitsemiseksi sekä SOCin nopea reagointi tietoturvaloukkauksiin ja ryhtymällä asianmukaisiin toimenpiteisiin vaikutusten lieventämiseksi ja lisävahinkojen estämiseksi. SOCin tehtäviin kuuluu myös tietoturvatietojen ja lokien analysointia, jotta saadaan tietoa uhkien luonteesta, hyökkäysmalleista ja uusista riskeistä sekä yksityiskohtaisten tutkimusten tekeminen tietoturvaloukkauksista perimmäisten syiden, vahingon laajuuden ja mahdollisten korjaavien toimien ymmärtämiseksi ja etsiä ennakoivasti uhkia tai haavoittuvuuksia, jotka ovat saattaneet kiertää perinteiset turvatoimet. (Salinas 2023.)

SOCin toiminnot ovat monitahoisia, ja ne kattavat erilaisia toimintoja, jotka yhdessä varmistavat organisaation kyberturvallisuuden häiriönsietokyvyn. Valvoo jatkuvasti verkkoliikennettä, järjestelmälokeja ja tietoturvahälytyksiä mahdollisten tietoturvaloukkausten tai poikkeavan käyttäytymisen tunnistamiseksi. Kun epätavallinen tapahtuma havaitaan, SOC tutkii tarkemmin, onko kyseessä aito tietoturvaloukkaus, ja arvioi sen vakavuuden. Noudattaa ennalta määriteltyjä tapahtumiin reagointimenettelyjä, jotta tietoturvaloukkaukset voidaan tehokkaasti rajoittaa, poistaa ja palautua niistä ja analysoi uhkatiedustelutietoja ja -raportteja ymmärtääkseen kehittyviä uhkia ja mahdollisia hyökkäystapoja. Kehittyneen analytiikan ja koneoppimistekniikoiden käyttö tietoturvadatan mallien ja poikkeavuuksien tunnistamiseksi, mikä mahdollistaa uhkien tarkemman havaitsemisen sekä tekee yhteistyötä organisaation muiden tiimien, kuten IT-, verkko-operaatio- ja sovellustiimien, kanssa varmistaakseen koordinoidun reagoinnin tietoturvaloukkauksiin. Tarkistaa ja kehittää jatkuvasti prosessejaan, välineitään ja menettelyjään sopeutuakseen muuttuviin uhkiin ja parantaakseen vaaratilanteisiin vastaamisen tehokkuutta. (Salinas 2023.)

3.2 Docker-pohjainen konttiteknologia

Docker on avoimen lähdekoodin alusta, joka helpottaa konttien luomista, jakelua ja suorittamista. Kontit ovat kevyitä, siirrettäviä ja eristettyjä ympäristöjä, jotka paketoivat sovelluksia ja niiden tarvitsemia kirjastoja, riippuvuuksia ja kokoonpanoja, jolloin niitä voidaan käyttää luotettavasti eri ympäristöissä. (Docker Docs 2023a.)

Docker-näköistiedostot ovat konttien rakennuspalikoita. Ne ovat itsenäisiä paketteja, jotka sisältävät sovelluskoodin, ajoajan, järjestelmätyökalut, kirjastot ja konfiguraatiotiedostot. Näköistiedostot tallennetaan rekisteriin, ja ne voidaan ladata ja ajaa millä tahansa isäntäkoneella, johon Docker on asennettu. Kontit ovat Docker-näköistiedostojen ajettavia instansseja. Ne tarjoavat sovelluksille eristetyn suoritusympäristön ja varmistavat, että niillä on omat tiedostojärjestelmänsä, verkkonsa ja prosessipuunsa.

Kontit ovat kevyitä, käynnistyvät nopeasti ja jakavat isäntäkoneen käyttöjärjestelmän kernelin, mikä tekee niistä tehokkaita ja siirrettäviä. Dockerfile on tekstitiedosto, joka sisältää ohjeet Docker-näköistiedoston rakentamiseen. Siinä määritetään peruskuva, lisätään riippuvuudet, kopioidaan tiedostoja, konfiguroidaan ympäristö ja määritellään komennot, jotka suoritetaan, kun kontti käynnistyy. Dockerfilet mahdollistavat Näköistiedostojen automaattisen ja toistettavan luomisen. (Docker Docs 2023a.) Konttien orkestrointialustat, kuten Kubernetes ja Docker Swarm, hallitsevat konttien käyttöönottoa, skaalautumista ja verkottumista useiden isäntien välillä. Ne tarjoavat kehittyneitä ominaisuuksia kuorman tasapainottamiseen, palvelujen löytämiseen, korkeaan saatavuuteen ja vikasietoisuuteen. (Docker Docs 2023c).

Docker-kontit tarjoavat monipuolista skaalautuvuutta, minkä ansiosta SOC-komponentteja voidaan helposti replikoida ja skaalata kysynnän mukaan. Kontit tarjoavat joustavan ja kevyen lähestymistavan skaalaamiseen, mikä varmistaa resurssien tehokkaan käytön ja mahdollistaa lisäinstanssien nopean käyttöönoton työmäärän kasvaessa. Kontitus edistää SOC-komponenttien modulaarisuutta ja paketointia. Kukin kontti paketoi tietyn toiminnon, kuten lokien yhdistämisen, uhkatiedon jakamisen tai verkkomurtojen havaitsemisen. Modulaarisen lähestymistavan ansiosta organisaatiot voivat valita ja yhdistellä kontteja omien SOC-vaatimustensa perusteella, mikä edistää joustavuutta ja mukautuvuutta.

Docker-kontit mahdollistavat resurssien tehokkaan käytön ajamalla useita kontteja yhdellä isännällä ja jakamalla taustalla olevan käyttöjärjestelmän kernelin. Tämä poistaa tarpeettomien järjestelmäresurssien tarpeen, vähentää laitteistokustannuksia ja parantaa yleistä resurssitehokkuutta SOC-ympäristöissä. SOC-komponentit voidaan helposti ottaa käyttöön ja käynnistää. Kontit voidaan luoda, päivittää ja jakaa nopeasti, mikä takaa tehokkaan käyttöönoton ja siirrettävyyden eri ympäristöissä, kuten kehitys-, testaus- ja tuotantoympäristöissä. Kontit tarjoavat eritasoisen eristyksen SOC-komponenttien välille, mikä varmistaa, että kukin kontti toimii itsenäisesti häiritsemättä muita. Tämä eristäminen auttaa rajaamaan mahdolliset tietoturvaloukkaukset ja pienentää hyökkäyspintaa, mikä parantaa SOC-arkkitehtuurin yleistä turvallisuutta. (Docker Docs 2023a.)

SOC-arkkitehtuurissa useiden konttien hallinta ja orkestrointi voi olla monimutkaista. On valittava sopivat konttien orkestrointialustat, kuten Kubernetes tai Docker Swarm, jotta voidaan varmistaa tehokas käyttöönotto, skaalautuminen, kuorman tasaus ja vikasietoisuus. Turvallisuusnäkökohdat ovat ratkaisevan tärkeitä Docker-pohjaisissa SOC-arkkitehtuureissa. On otettava käyttöön parhaat käytännöt konttiturvallisuutta varten, kuten varmistettava turvalliset konttinäköistiedostot, valvottava konttien käyttöoikeuksia ja korjattava ja päivitettävä kontteja säännöllisesti haavoittuvuuksien korjaamiseksi. (Docker Docs 2023b). SOC-arkkitehtuurit

tuottavat ja käsittelevät suuria määriä turvallisuuteen liittyviä dataa. On harkittava tehokkaita tiedon tallennus- ja hallintastrategioita, mukaan lukien konttipohjaisten tietokantojen hyödyntäminen tai konttipohjaisten tallennusratkaisujen integrointi. (Docker Docs 2023d). SOC-komponenttien integrointi olemassa oleviin järjestelmiin, kuten SIEM-alustoihin tai häiriötilanteisiin vastaamisen työvaiheisiin, vaatii huolellista suunnittelua ja konfigurointia. Asianmukainen viestintä ja datan jakaminen konttien ja ulkoisten järjestelmien välillä ovat olennaisen tärkeitä saumattoman integroinnin ja yhteistyön kannalta. Eri konttien ja versioiden välisen yhteensopivuuden ja yhteen toimivuuden varmistaminen on ratkaisevan tärkeää konttipohjaisissa SOC-arkkitehtuureissa. On varmistettava konttipohjaisten SOC-komponenttien yhteensopivuus ja otettava huomioon versionhallinta, jotta vältetään ristiriidat ja ylläpidetään sujuvaa toimintaa.

3.3 Portainer Konttien hallinta ja orkestrointi

Portainer on suosittu avoimen lähdekoodin konttien hallinta- ja orkestrointityökalu, joka tarjoaa käyttäjäystävällisen käyttöliittymän Docker-konttien ja -klusterien hallintaan. Se yksinkertaistaa konttisovellusten hallintaa ja valvontaa, mikä tekee siitä olennaisen osan Docker-pohjaisissa SOC-arkkitehtuureissa. (Portainer 2023a.)

Portainerin avulla voidaan helposti luoda, käynnistää, pysäyttää ja poistaa kontteja intuitiivisen verkkopohjaisen käyttöliittymän kautta. Se tarjoaa keskitetyn käyttöliittymän yksittäisten konttien ja niihin liittyvien resurssien, kuten verkkojen ja volyymien, hallintaan. (Portainer 2023b). Portainer helpottaa Docker Swarm -klustereiden hallintaa, mikä mahdollistaa konttien orkestroinnin ja skaalautumisen useiden isäntien välillä. Se yksinkertaistaa Swarm-palveluiden, pinojen luomista ja hallintaa, mikä helpottaa konttipohjaisten SOC-komponenttien käyttöönottoa hajautetussa ympäristössä. (Portainer n.d.a).

Portainer tukee roolipohjaisen pääsynhallinta-mekanismeja (role-based access control, RBAC) (Portainer n.d.b), joiden avulla ylläpitäjät voivat määritellä käyttäjäroolit ja -oikeudet. Näin varmistetaan turvallinen pääsy konttien hallintatoiminnallisuuksiin käyttäjän oikeuksiin perustuen. Lisäksi Portainer integroituu ulkoisiin todennuspalveluntarjoajiin, kuten LDAP (Lightweight directory access protocol) (Portainer n.d.b) tai OAuthiin, turvallisuuden ja käyttäjähallintaominaisuuksien parantamiseksi. Portainer tarjoaa reaaliaikaista konttien resurssien käytön seurantaa, mukaan lukien suorittimen, muistin ja verkkotiedot. (Portainer 2023c). Se integroituu myös lokityökaluihin, kuten Elasticsearchiin ja Logstashiin, jolloin käyttäjät voivat tarkastella konttilokeja, suorittaa lokien suodatusta ja vianmääritystä SOC-ympäristössä. (Portainer 2023d).

Portainer tukee mallien ja näköistiedostorekisterien luomista ja hallintaa, mikä yksinkertaistaa konttisovellusten käyttöönottoa. Käyttäjät voivat määritellä malleja yleisimmille SOC-komponenteille ja käynnistää helposti instansseja näiden ennalta määritettyjen kokoonpanojen perusteella. Portainer poistaa komentorivillä tapahtuvan vuorovaikutuksen tarpeen tarjoamalla graafisen käyttöliittymän (Graphical User Interface, GUI) (Computer hope 2021) konttiasetusten konfigurointia ja sovellusten käyttöönottoa varten. Tämä tekee konttien hallinnasta helpommin lähestyttävää käyttäjille, joilla ei välttämättä ole laajaa komentorivikokemusta. (Portainer 2023a.)

Portainerin käyttäjäystävällinen käyttöliittymä yksinkertaistaa Docker-konttien hallintaa, mikä vähentää SOC-järjestelmän ylläpitäjien ja operaattoreiden oppimiskäyrää. Se tarjoaa visuaalisen esityksen konttiympäristöstä, mikä helpottaa SOC-komponenttien seurantaa, konfigurointia ja vianmääritystä. Portainerin avulla organisaatiot voivat helposti skaalata SOC-arkkitehtuuriaan hallinnoimalla Docker Swarm -klustereita. Se yksinkertaistaa uusien isäntien lisäämistä, konttipalveluiden käyttöönottoa ja kuorman tasapainottamisen hallintaa, jolloin SOC-komponentit skaalautuvat horisontaalisesti työmäärän kasvaessa. (Portainer n.d.b.)

Portainer tukee RBACia, jolloin useat käyttäjät voivat tehdä yhteistyötä konttien hallintatehtävissä ja samalla hallita käyttöoikeuksia. Tämä edistää tiimityötä ja tehostaa SOC-toimintoja, kun tiimin eri jäsenet voivat hallita tiettyjä kontteja tai klustereita roolinsa ja vastuualueidensa perusteella. (Portainer n.d.b). Portainerin malli- ja näköistiedostojenhallintaominaisuudet tehostavat konttipohjaisten SOC-komponenttien käyttöönottoa. Järjestelmänvalvojat voivat luoda malleja yleisesti käytetyille SOC-konteille, mikä varmistaa yhdenmukaiset kokoonpanot ja nopeuttaa käyttöönottoprosessia. Portainerin valvonta- ja lokiominaisuudet tarjoavat reaaliaikaista tietoa konttien resurssien käytöstä ja helpottavat vianmääritystä. SOC-ylläpitäjät voivat tunnistaa suorituskyvyn pullonkaulat, seurata resurssien käyttöä ja analysoida konttilokeja diagnosoidakseen ja ratkaistakseen ongelmat nopeasti. (Portainer 2023d).

Portainer tarjoaa pääsyn kriittisiin konttien hallintatoimintoihin, joten asianmukaisten turvatoimien varmistaminen on ratkaisevan tärkeää. Portainer-käyttöliittymä on ehdottomasti suojattava vahvoilla todennusmekanismeilla, RBAClla ja pääsynvalvonnalla luvattoman käytön ja mahdollisten hyökkäysten estämiseksi. (Portainer n.d.b). Portainer voidaan integroida ulkoisiin valvonta- ja lokityökaluihin, kuten Elasticsearchiin tai Logstashiin, SOC-valvonta- ja analyysitoimintojen parantamiseksi. Portainerin integroiminen sopiviin työkaluihin olisi harkittava SOC-vaatimusten ja olemassa olevan infrastruktuurin perusteella. (Portainer n.d.b).

Tehtäväkriittisissä SOC-ympäristöissä suositellaan Portainerin käyttöönottoa hyvin saatavilla olevassa ja vikasietoisessa kokoonpanossa. Tähän kuuluu Portainer-instanssien käyttäminen konttipohjaisessa, klusteroidussa ympäristössä redundanssin, kuorman tasauksen ja saumattoman vikasietoisuuden varmistamiseksi järjestelmävikojen yhteydessä. (Portainer n.d.b). SOCin ylläpitäjien ja operaattoreiden olisi saatava asianmukaista koulutusta ja perehdyttävä Portainerin toimintoihin ja ominaisuuksiin. Näin varmistetaan tehokas käyttö, maksimoidaan tuottavuus ja optimoidaan SOC-toiminnot.

3.4 Elasticsearch: haku- ja analyysimoottori

Elasticsearch on tehokas haku- ja analyysimoottori, jota käytetään laajalti erilaisissa sovelluksissa, kuten SOCissa. Se tarjoaa hajautettuja, reaaliaikaisia tallennus-, haku- ja analysointimahdollisuuksia suurille määrille strukturoitua ja strukturoimatonta dataa. Elasticsearch on suunniteltu käsittelemään suuria datan syöttönopeuksia, monimutkaisia kyselyjä ja lähes välittömiä hakuvastauksia, mikä tekee siitä olennaisen osan konttipohjaisissa SOC-arkkitehtuureissa. (Elasticsearch 2023)

Elasticsearch käyttää hajautettua arkkitehtuuria, joka mahdollistaa tietojen tallentamisen ja käsittelyn useissa noodeissa tai palvelimissa. Tämä hajautettu lähestymistapa mahdollistaa skaalautuvuuden, vikasietoisuuden ja korkean käytettävyyden, minkä ansiosta Elasticsearch soveltuu laajoihin SOC-ympäristöihin. Elasticsearch on tunnettu kokoteksti hakuominaisuuksistaan, jotka mahdollistavat nopeat ja tarkat hakutulokset suurissa tietomäärissä. Se käyttää kehittyneitä hakualgoritmeja, tokenisointia ja indeksointitekniikoita tarjotakseen erittäin suorituskykyisiä ja relevantteja hakutuloksia. (Elastic 2023.)

Elasticsearch mahdollistaa lähes reaaliaikaisen analytiikan syötetyistä datasta. Se tukee monimutkaisia datan aggregointeja, suodatusta ja analysointia, minkä ansiosta SOC-analyytikot voivat johtaa arvokkaita havaintoja ja malleja tietoturvaan liittyvistä tiedoista. Elasticsearch on suunniteltu skaalautumaan horisontaalisesti lisäämällä klusteriin lisää noodeja. Tämä skaalautuvuus varmistaa, että SOC-arkkitehtuurit pystyvät käsittelemään kasvavia tietomääriä, monimutkaisia kyselyjä ja kasvavia työtehtäviä. Elasticsearchin hajautettu luonne parantaa myös suorituskykyä rinnakkaistamalla haku- ja analyysitoimintoja koko klusterissa. (Elastic 2023.)

Elasticsearchissa on useita vaihtoehtoja datan keräämiseen, mukaan lukien RESTful APIt, indeksointi ja integrointi suosittuihin tiedonkeruutyökaluihin, kuten Logstashiin ja Beatsiin. Tämän joustavuuden ansiosta SOC-arkkitehtuurit voivat integroida saumattomasti dataa eri lähteistä, kuten lokitiedostoista, tapahtumista ja uhkatietosyötteistä. Elasticsearchissa on sisäänrakennettuja

ominaisuuksia korkeaan käytettävyyteen ja vikasietoisuuteen. Data kopioidaan automaattisesti useisiin noodeihin, mikä varmistaa datan kestävyyden ja minimoi datan katoamisen riskin. Elasticsearch tukee myös automaattista shard-jakoa ja noodin palautusta, mikä takaa jatkuvan toiminnan myös vikatilanteissa. (Elastic 2023.)

Elasticsearch toimii keskustietovarastona eri SOC-komponenttien lokitietojen tallentamisessa ja indeksoinnissa. Se mahdollistaa tehokkaan lokien yhdistämisen, indeksoinnin ja haun, minkä ansiosta SOC-analyytikot voivat hakea ja analysoida lokitietoja uhkien havaitsemista, vaaratilanteiden tutkintaa ja vaatimustenmukaisuutta varten. Lähes reaaliaikaiset analyysiominaisuudet tekevät siitä arvokkaan uhkien metsästykseen ja havaitsemiseen SOC-arkkitehtuurissa. SOC-analyytikot voivat hyödyntää Elasticsearchia etsiessään vaarantumisindikaattoreita (Indicator of Compromise, IoC) (Logsign 2019), suorittaessaan käyttäytymisanalyysejä ja tunnistaessaan mahdollisia tietoturvauhkia tai -malleja koko datasta. (Elastic 2023.)

Elasticsearch integroituu saumattomasti visualisointityökaluihin, kuten Kibanaan, jolloin SOC-analyytikot voivat luoda interaktiivisia dashboardeja ja visualisointeja tietoturvaan liittyvistä tiedoista. Tämä antaa analyytikoille mahdollisuuden saada käyttökelpoisia näkymiä, valvoa SOCin suorituskykyä ja esittää havainnot tehokkaasti eri tahoille. SOC-arkkitehtuurit voivat rikastuttaa dataansa ulkoisilla uhkatiedustelulähteillä. Tämä integrointi parantaa uhkien havaitsemisen tarkkuutta ja mahdollistaa oikea-aikaisen reagoinnin uusiin uhkiin. SOC-arkkitehtuurit voivat täyttää vaatimustenmukaisuusvaatimukset ja säilyttää aiemmat tiedot rikosteknistä analyysia, tapahtumiin reagoimista ja auditointia varten. (Elastic 2023.)

Elasticsearchin varastointivaatimukset riippuvat SOC-arkkitehtuurin datamäärästä ja säilytysajasta. On suunniteltava ja varattava riittävästi tallennusresursseja ennakoitua datan kasvua ja säilytystarpeita varten. (Elastic 2023.)

Konfigurointi ja säätäminen optimaalisen suorituskyvyn saavuttamiseksi on ratkaisevan tärkeää SOC-arkkitehtuurissa. Tähän sisältyy sellaisia näkökohtia kuin shard-jako, indeksointistrategiat, heap-koon jakaminen ja kyselyjen optimointi tehokkaiden haku- ja analyysitoimintojen varmistamiseksi. (Elastic 2023.)

Elasticsearch sisältää arkaluonteisia turvallisuuteen liittyviä tietoja, ja siksi on toteutettava asianmukaiset turvatoimet. Tähän kuuluu Elasticsearch-klustereiden suojaaminen todennusmekanismeilla, salauksella, pääsynvalvonnalla sekä auditointi- ja seurantaominaisuuksien toteuttaminen. (Elastic 2023.)

SOC-arkkitehtuurissa tulisi luoda Elasticsearch-klustereille luotettavat datan varmuuskopiointi- ja palautusmekanismit. Säännölliset varmuuskopiot, kopiointi ja klusterin varmuuskopiointi varmistavat tietojen häiriönsietokyvyn ja helpottavat nopeaa palautumista datan katoamisen tai järjestelmävian sattuessa. Datan määrän kasvaessa Elasticsearchin skaalautumisesta tulee välttämätöntä. Olisi seurattava klusterin tilaa, suunniteltava noodien lisäämistä ja harkittava shardien hallintastrategioita, jotta voidaan varmistaa saumaton skaalautuvuus ja optimaalinen suorituskyky. (Elastic 2023.)

3.5 Kibana: Datan visualisointi ja hahmottaminen

Kibana on tiedon visualisointi- ja hahmotustyökalu, jota käytetään yleisesti yhdessä Elasticsearchin kanssa SOCissa. Se tarjoaa käyttäjäystävällisen käyttöliittymän Elasticsearchiin tallennettujen tietojen tutkimiseen, analysointiin ja visualisointiin, mikä tekee siitä olennaisen osan konttipohjaisissa SOC-arkkitehtuureissa. (Kibana 2023.)

Kibanan avulla käyttäjät voivat tutkia ja hakea Elasticsearchiin tallennettuja tietoja käyttämällä erilaisia hakukyselyjä, suodattimia ja yhdistelmiä. Se tarjoaa tehokkaan hakukielen ja syntaksin, jonka avulla SOC-analyytikot voivat tehdä

monimutkaisia hakuja ja hakea tiettyjä tietojen osajoukkoja. Kibanassa on laajan valikoiman visualisointivaihtoehtoja, kuten kaavioita, graafeja, karttoja ja taulukoita, joiden avulla tiedot voidaan esittää visuaalisesti havainnollisella ja ymmärrettävällä tavalla. SOC-analyytikot voivat luoda visuaalisia esityksiä tietoturvaan liittyvistä datasta, mikä mahdollistaa paremman ymmärryksen, trendien tunnistamisen ja poikkeavuuksien havaitsemisen. Kibana mahdollistaa interaktiivisten dashboardien luomisen, jotka yhdistävät useita visualisointeja yhdelle näytölle. Dashboardit tarjoavat kokonaisvaltaisen näkymän SOC-dataan, jolloin analyytikot voivat seurata keskeisiä metriikoita, seurata suorituskykyindikaattoreita ja saada reaaliaikaista tilannekuvaa. Käyttäjät voivat asettaa hälytyksiä ja ilmoituksia ennalta määritettyjen ehtojen tai raja-arvojen perusteella. Tämä ominaisuus mahdollistaa kriittisten tietoturvatapahtumien ennakoivan seurannan ja hälytyksen, mikä auttaa SOC-analyytikkoja pysymään ajan tasalla ja reagoimaan nopeasti mahdollisiin uhkiin. (Elastic 2023.)

Kibana tukee aikasarja-analyysiä, jonka avulla SOC-analyytikot voivat tunnistaa malleja, trendejä ja korrelaatioita aikapohjaisesta datasta. Tämä ominaisuus on erityisen hyödyllinen, kun halutaan havaita ajallisia tietoturvatapahtumia tai ymmärtää uhkien kehittymistä ajan myötä. Kibana integroituu saumattomasti Elasticsearchiin ja hyödyntää sen tehokkaita haku- ja analyysiominaisuuksia. Se tarjoaa intuitiivisen ja helppokäyttöisen käyttöliittymän Elasticsearchin tietojen kanssa toimimiseen, mikä helpottaa datan tutkimista ja visualisointia. (Elastic 2023.)

Kibanan avulla SOC-analyytikot voivat visualisoida ja valvoa keskeisiä tietoturvametriikoita, kuten verkkoliikennettä, lokitapahtumia, käyttäjien käyttäytymistä tai uhkatietoindikaattoreita. Analyytikot voivat luoda visualisointeja, jotka tuovat esiin tietoturvatrendejä, poikkeamia tai mahdollisia uhkia, mikä mahdollistaa nopean tunnistamisen ja reagoimisen. Datan tutkimisominaisuudet ovat arvokkaita SOC-analyytikoille, jotka suorittavat vaaratilanteiden analysointia ja tutkintaa. Analyytikot voivat tehdä ad-hoc-hakuja, soveltaa suodattimia ja tarkentaa tiettyjä datan osajoukkoja

paljastaakseen uusia havaintoja, tunnistaakseen malleja ja ymmärtääkseen tietoturvaloukkausten kontekstin. (Elastic 2023.)

Kibana-dashboardit tarjoavat SOC-tiimeille reaaliaikaisen tilannetietoisuuden yhdistämällä olennaiset visualisoinnit ja metriikat yhdelle näytölle. Dashboardien avulla analyytikot voivat seurata SOC:n suorituskykyä, seurata keskeisiä indikaattoreita ja havaita tietoturvatapahtumia tai poikkeamia reaaliajassa. SOC-analyytikot voivat hyödyntää Kibanan visualisointiominaisuuksia mahdollisten uhkien tunnistamiseen ja uhkien metsästykseen. Visualisoimalla tietoturvaan liittyviä dataa analyytikot voivat havaita poikkeamia, malleja tai korrelaatioita, jotka voivat viitata haitalliseen toimintaan tai uusiin uhkiin. Kibanan avulla SOC-analyytikot voivat luoda visuaalisesti havainnollisia raportteja ja esityksiä, joiden avulla he voivat välittää havaintoja ja näkemyksiä eri osapuolille. Analyytikot voivat luoda räätälöityjä raportteja, viedä visualisointeja ja jakaa interaktiivisia dashboardeja, mikä parantaa yhteistyötä ja tukee päätöksentekoprosesseja. (Elastic 2023.)

Kibanan tehokkaan käytön kannalta ratkaisevaa on datan asianmukainen indeksointi ja kartoittaminen Elasticsearchissa. SOC-arkkitehtuureissa olisi varmistettava, että tiedot indeksoidaan ja kartoitetaan oikein, jotta tietojen tarkka ja tehokas tutkiminen ja visualisointi on mahdollista. SOC-analyytikkojen olisi noudatettava visualisoinnin suunnittelun parhaita käytäntöjä, jotta voidaan varmistaa visualisointien selkeys, tarkkuus ja tehokkuus. Suunnitteluun liittyviä näkökohtia voivat olla esimerkiksi sopivien kaaviotyyppien valinta, värikoodaus, merkinnät ja tarkoituksenmukaisten ja olennaisten datan esitystapojen valinta. Kibanaan ja sen dashboardeihin pääsyä olisi valvottava asianmukaisesti tietoturvan varmistamiseksi.

SOC-arkkitehtuureissa olisi otettava käyttöön todennusmekanismit, RBAC ja salaus arkaluonteisten tietoturvaan liittyvien tietojen suojaamiseksi. Kibanan suorituskyvyn optimointi SOC-arkkitehtuureissa edellyttää sellaisten tekijöiden huomioon ottamista kuin kyselyjen optimointi, datan aggregointistrategiat ja Elasticsearch-indeksien asianmukainen konfigurointi. Kibanan ja Elasticsearchin asetusten seurannalla ja hienosäätämisellä voidaan varmistaa

optimaaliset vasteajat ja tehokas tiedonhaku. SOC-analyytikkojen tulisi saada koulutusta ja perehtyä Kibanan toimintoihin ja ominaisuuksiin, jotta sen potentiaali voidaan maksimoida. Koulutusohjelmat voivat auttaa analyytikoita hallitsemaan tietojen tutkimisen, visualisoinnin suunnittelun, dashboardien luomisen ja hälytysten konfiguroinnin Kibanassa. (Elastic 2023.)

3.6 Logstash: Lokien aggregointi ja käsittely

Logstash on datan käsittelyputkisto, joka helpottaa lokitietojen keräämistä, yhdistämistä ja muuntamista eri lähteistä SOC-arkkitehtuurissa. Se toimii avainkomponenttina konttipohjaisissa SOC-ympäristöissä mahdollistamalla keskitetyn lokien hallinnan ja parantamalla lokianalyysivalmiuksia. (Logstash 2023.)

Logstash tukee lokitietojen keräämistä erilaisista lähteistä, kuten lokitiedostoista, syslogista, verkkojen lähdevirroista, viestijonoista ja tietokantajärjestelmistä. Se tarjoaa laajan valikoiman syöttöliitännäisiä, minkä ansiosta se on joustava ja mukautettavissa erilaisiin datalähteisiin SOC-arkkitehtuurissa. SOC-ylläpitäjät voivat jäsentää ja muuntaa lokitietoja erilaisten suodattimien avulla. Se tukee säännöllisiä lausekkeita, ehdollisia lausekkeita ja tietojen rikastamistekniikoita, jotka mahdollistavat asiaankuuluvien datan poimimisen, lokimuotojen muuttamisen ja lokien rikastamisen kontekstisidonnaisilla tiedoilla. Logstash aggregoi lokitiedot useista eri lähteistä keskitettyyn arkistoon, kuten Elasticsearchiin. Se kokoaa yhteen lokit eri SOC-komponenteista, kuten verkkolaitteista, tietoturvalaitteista, palvelimista ja sovelluksista, mikä helpottaa lokien hallintaa ja analysointia. (Logstash 2023.)

Logstash mahdollistaa tietojen muuntamisen suodatinliitännäisten avulla. SOC-ylläpitäjät voivat normalisoida lokitietoja, muuntaa aikaleimoja yhteiseen muotoon, poimia tiettyjä kenttiä, anonymisoida arkaluonteisia tietoja tai suorittaa mukautettuja datan käsittelyjä valmistellakseen datan lisäanalyysiä varten. Logstash tukee datan rikastamista integroimalla ulkoisia tietolähteitä tai

palveluja. SOC-ylläpitäjät voivat rikastuttaa lokitietoja uhkatietosyötteillä, maantieteellisillä sijaintitiedoilla, käyttäjätiedoilla tai muilla asiaankuuluvilla kontekstitiedoilla. (Logstash 2023.)

Tämä rikastaminen parantaa lokitietojen analysointia ja korrelointia SOC-arkkitehtuurissa. Logstash integroituu saumattomasti ELK-pinon (Elasticsearch, Logstash, Kibana) muihin komponentteihin, mikä mahdollistaa lokitietojen louhinnan, muuntamisen ja lataamisen Elasticsearchiin indeksointia ja analysointia varten. Se tarjoaa sillan lokilähteiden ja Kibanan visualisointiominaisuuksien välille. (Logstash 2023.)

Logstash mahdollistaa lokien keräämisen ja keskittämisen eri SOC-komponenteista, mikä tarjoaa yhtenäisen näkymän lokitietoihin. Se auttaa SOC-analyytikkoja pääsemään kaikkiin asiaankuuluviin lokitietoihin keskitetyssä paikassa, mikä helpottaa tehokasta lokianalyysiä, korrelaatiota ja häiriötilanteiden tutkintaa. Jäsennys- ja suodatusominaisuuksien avulla SOC-ylläpitäjät voivat poimia lokitiedoista olennaista dataa, hylätä tarpeettomat datat ja normalisoida lokimuotoja. Näin SOC-analyytikot voivat keskittyä olennaiseen lokitietoon ja vähentää lokianalyysiprosessien häiriötä ja sekavuutta. (Logstash 2023.)

Logstash tukee datan muuntamista ja rikastamista, minkä ansiosta SOC-ylläpitäjät voivat parantaa lokitietoja lisäkontekstilla. Lokien rikastaminen uhkatiedoilla, maantieteellisillä sijaintitiedoilla tai käyttäjätiedoilla mahdollistaa kattavamman lokianalyysin, korrelaation ja tietoturvaloukkausten havaitsemisen. Keskeinen rooli lokianalyysissä ja korrelaatiossa SOC-arkkitehtuurissa. Logstash auttaa SOC-analyytikoita tunnistamaan lokitiedoista kuvioita, poikkeamia ja mahdollisia tietoturvauhkia aggregoimalla lokitietoja eri lähteistä, jäsentämällä ja muuntamalla dataa sekä rikastamalla sitä asiaankuuluvilla tiedoilla. Logstash integroituu saumattomasti Elasticsearchin ja Kibanan kanssa muodostaen tehokkaan kolmikon, joka tunnetaan nimellä ELK-pino (Elasticsearch, Logstash, Kibana). Tämän integraation ansiosta SOC-

arkkitehtuurit voivat hyödyntää Logstashin lokinkäsittelyominaisuuksia datan syöttämiseksi Elasticsearchiin indeksointia ja analysointia varten, kun taas Kibana tarjoaa visualisointi- ja dashboard-käyttöliittymän. (Elastic 2023.)

Logstashin suorituskyky ja skaalautuvuus riippuvat sellaisista tekijöistä kuin lokien määrästä, käsittelyvaatimuksista ja käytettävissä olevista järjestelmäresursseista. SOC-arkkitehtuurien olisi mitoitettava ja konfiguroitava Logstash-instanssit oikein, jotta ne voivat käsitellä odotettua lokitietomäärää ja varmistaa tehokkaan lokien käsittelyn. Olisi suunniteltava huolellisesti lokien jäsennys- ja suodatusmääritykset Logstashissa. Monimutkaiset jäsentely- tai suodatussäännöt voivat vaikuttaa suorituskykyyn tai aiheuttaa käsittelyvirheitä. Parsinta- ja suodatuskonfiguraatioiden säännöllinen testaaminen ja validointi on ratkaisevan tärkeää, jotta lokien tarkka käsittely voidaan varmistaa. Logstash saattaa käsitellä arkaluonteisia lokitietoja, jotka edellyttävät asianmukaisia turvatoimia. SOC-arkkitehtuureissa olisi suositeltavaa ottaa käyttöön salausta, pääsynvalvontaa ja tietojen anonymisointitekniikoita tarpeen mukaan lokitietojen luottamuksellisuuden ja yksityisyyden suojaamiseksi.

Käyttöönotossa olisi oltava asianmukaiset valvonta- ja virheenkäsittelymekanismit. Olisi seurattava Logstashin suorituskykyä, käsittelyvirheitä ja otettava käyttöön asianmukaiset virheenkäsittelystrategiat, jotta voidaan varmistaa luotettava lokien käsittely ja estää datan häviäminen. SOC-arkkitehtuurien olisi suositeltavaa ottaa käyttöön ylläpito- ja päivitysprosessit, jotta Logstash-instanssit pysyvät ajan tasalla uusimpien versioiden ja tietoturvakorjausten kanssa. Säännölliset ylläpitotoimet, mukaan lukien varmuuskopiointi- ja palautussuunnitelmat, varmistavat lokinkäsittelytoimintojen jatkuvuuden. (Elastic 2023.)

3.7 Filebeat: Lokien lähettäjä ja välittäjä

Filebeat on kevyt lokien lähetys- ja välityspalvelu, jolla on tärkeä rooli lokitietojen turvallisessa keräämisessä, käsittelyssä ja välittämisessä SOC-

arkkitehtuurissa. Se mahdollistaa lokitiedostojen tehokkaan siirron eri lähteistä keskitettyihin tallennus- tai analyysijärjestelmiin, mikä tekee siitä olennaisen osan konttipohjaisissa SOC-ympäristöissä. (Filebeat 2023.)

Filebeat on suunniteltu keräämään lokitietoja eri lähteistä, kuten lokitiedostoista, järjestelmälokeista, verkkolokeista ja sovelluslokeista. Se lukee lokitiedostoja tehokkaasti, seuraa muutoksia ja lähettää lokitiedot turvallisesti haluttuun kohteeseen. Filebeat on kevyt ja sen resurssivaatimukset ovat minimaaliset, joten se soveltuu käytettäväksi resurssirajoitteisissa ympäristöissä. Sen tehokkuus varmistaa, että se kuluttaa mahdollisimman vähän suorittimen, muistin ja verkon resursseja lokien keruu- ja lähetystoimintojen aikana. Filebeat tuo joustavuutta syöttölähteiden määrittelyssä, jolloin SOC-ylläpitäjät voivat valvoa lokitiedostoja ja -hakemistoja tiettyjen mallien, sijainnin tai tiedostotyyppien perusteella. Se tukee monenlaisia lokimuotoja ja -protokollia, joten se on mukautettavissa erilaisiin lokilähteisiin SOC-arkkitehtuurissa. (Filebeat 2023a.) Filebeat varmistaa turvallisen lokikuljetuksen verkon yli käyttämällä erilaisia salausprotokollia, kuten TLS / SSL. Se salaa lokitiedot siirron aikana, estää lokitiedostojen luvattoman käytön tai sieppaamisen ja varmistaa lokitietojen luottamuksellisuuden ja eheyden siirron aikana. (Filebeat 2023b). Filebeat integroituu saumattomasti Logstashin ja Elasticsearchin kanssa muodostaen tehokkaan dataputken. Se voi syöttää lokitiedot suoraan Logstashiin jatkokäsittelyä varten tai välittää lokit Elasticsearchiin indeksointia ja analysointia varten SOC-arkkitehtuurissa (Filebeat 2023a).

Filebeatilla on ratkaiseva rooli lokitietojen keräämisessä eri SOC-komponenteista ja niiden keskittämisessä jatkoanalyysiä ja korrelaatiota varten. Se valvoo lokitiedostoja, lukee lokitiedot reaaliajassa ja siirtää lokit tehokkaasti keskitettyihin tallennus- tai analyysijärjestelmiin varmistaen lokitietojen oikea-aikaisen saatavuuden SOC-toimintoja varten. Filebeat mahdollistaa lähes reaaliaikaisen lokinkäsittelyn seuraamalla jatkuvasti lokitiedostoja ja lähettämällä lokitiedot heti, kun uusia tapahtumia ilmenee. Tämä helpottaa nopeaa havaitsemista ja reagointia tietoturvaloukkauksiin, jolloin SOC-analyytikot voivat saada välittömästi tietoa lokitiedoista. Filebeatin kevyt

rakenne ja skaalautuvuusominaisuudet tekevät siitä hyvin sopivan laajoihin SOC-ympäristöihin. Se voidaan ottaa käyttöön useiden isäntien tai konttien välillä, mikä mahdollistaa tehokkaan lokien keräämisen ja lähettämisen erilaisista lähteistä samalla kun resurssien tarve minimoidaan. Suodatus- ja rikastusominaisuuksien avulla SOC-ylläpitäjät voivat esikäsitellä lokitietoja ennen niiden välittämistä. Tähän sisältyy tiettyjen lokitapahtumien poissulkeminen tai sisällyttäminen, lokimuotojen muuttaminen tai lokitietojen rikastaminen ylimääräisillä metatiedoilla. Nämä ominaisuudet parantavat lokitietojen laatua ja merkitystä SOC-arkkitehtuurissa. Filebeat varmistaa lokitietojen turvallisen siirron verkon kautta ja suojaa arkaluonteisia lokitietoja luvattomalta käytöltä tai sieppaukselta. Tämä on ratkaisevan tärkeää lokitietojen luottamuksellisuuden ja eheyden säilyttämiseksi kuljetuksen aikana, erityisesti SOC-arkkitehtuurissa, jossa lokitiedot voivat sisältää arkaluonteisia tietoja. (Filebeat 2023a.)

Filebeat on määritettävä huolellisesti tarkkailemaan asianmukaisia lokilähteitä SOC-arkkitehtuurissa. Tähän sisältyy oikeiden lokitiedostopolkujen, lokimallien ja lokimuotojen määrittäminen tarkan ja kattavan lokikeräyksen varmistamiseksi. SOC-arkkitehtuurien tulisi ottaa huomioon resurssien optimointi Filebeatia käyttöönotettaessa. Vaikka Filebeat on kevyt, suuren määrän lokitiedostoja tai suuren määrän lokilähteitä valvominen voi vaatia asianmukaista resurssien jakamista ja valvontaa optimaalisen suorituskyvyn varmistamiseksi. Suojatut siirto-ominaisuudet on määritettävä asianmukaisesti lokitietojen salaamiseksi siirron aikana. SOC-arkkitehtuurien olisi otettava käyttöön TLS/SSL-salaus ja määritettävä varmenteet lokitiedostojen turvallisen siirron varmistamiseksi. Suorituskyvyn seuranta ja virheiden tai vikojen käsittely on olennaista SOC-arkkitehtuurissa. Olisi suositeltavaa ottaa käyttöön asianmukaiset seurantamekanismit, kuten lokien seuranta tai kuntotarkastukset, ja otettava käyttöön virheenkäsittelyprosessit lokien keräys- ja lähetystoimintojen luotettavuuden ja jatkuvuuden varmistamiseksi. Integraatio Logstashin tai Elasticsearchin kanssa on konfiguroitava asianmukaisesti, jotta varmistetaan saumaton tiedonkulku SOC-arkkitehtuurissa. Tulisi tarkistaa Filebeatin ja analyysijärjestelmien yhteensopivuus ja versioyhteensopivuus

yhteensopivuusongelmien tai tietojenkäsittelyvirheiden välttämiseksi. (Filebeat 2023b.)

3.8 MISP: Uhkatiedustelun jakamisalusta

MISP on avoimen lähdekoodin uhkatiedon jakamisalusta, joka helpottaa kyberturvallisuuden uhkatiedon keräämistä, jakamista ja analysointia SOC-arkkitehtuurissa. Sen avulla SOC-tiimit voivat tehdä yhteistyötä, vaihtaa ja analysoida uhkatiedustelutietoja, mikä parantaa niiden kykyä havaita tietoturvauhkia, reagoida niihin ja lieventää niitä. (MISP n.d.)

MISPin avulla SOC-tiimit voivat kerätä uhkatietoja eri lähteistä, kuten avoimen lähdekoodin syötteistä, luotetuista kumppaneista, toimialaryhmistä ja sisäisestä tutkimuksesta. Se tukee strukturoitujen uhkatiedustelutietojen, kuten vaarantumisindikaattoreiden, uhkatoimijatietojen ja tietoturvatapahtumien, keräämistä. Edistää tietojen jakamista ja yhteistyötä antamalla SOC-tiimien jakaa uhkatiedustelutietoja luotettavien kumppaneiden ja kollegojen kanssa. Se helpottaa uhkaindikaattoreiden, analyysiraporttien ja asiayhteystietojen vaihtoa ja mahdollistaa yhteisölähtöisen lähestymistavan uhkatiedon jakamiseen. (MISP n.d.)

MISP tarjoaa valmiudet uhkaindikaattoreiden hallintaan, tallentamiseen ja korrelointiin keskitetyssä arkistossa. SOC-tiimit voivat luoda ja hallita indikaattoriprofiileja, liittää niihin metatietoja ja suorittaa korrelaatioanalyysejä kuvioiden, suhteiden ja mahdollisten uhkakampanjoiden tunnistamiseksi. MISP tukee uhka-analyysiä ja rikastamista integroimalla se ulkoisiin tietolähteisiin ja analyysityökaluihin. SOC-tiimit voivat hyödyntää näitä integraatioita uhkatiedustelun rikastamiseksi asiayhteyteen liittyvillä tiedoilla, kuten uhkasyötteillä, mainetiedoilla, maantieteellisillä sijaintitiedoilla tai haavoittuvuustietokannoilla. MISP mahdollistaa taksonomioiden ja merkintäjärjestelmien mukauttamisen uhkatiedustelutietojen luokittelua ja kategorisointia varten. SOC-tiimit voivat määritellä oman taksonomiansa omien

tarpeidensa mukaan, mikä mahdollistaa uhkaindikaattoreiden johdonmukaisen merkitsemisen ja luokittelun parempia analyysi- ja hakuominaisuuksia varten. MISP tukee automaattista indikaattorisyötteiden syöttämistä, jolloin SOC-tiimit voivat automaattisesti hakea ja tuoda uhkatietosyötteitä luotettavista lähteistä. Tämä automaatio tehostaa reaaliaikaisen uhkatiedon keräämistä ja integroimista SOC-arkkitehtuuriin. (MISP n.d.)

MISP toimii keskeisenä yhteyspisteenä uhkatiedustelun jakamisessa ja vaihtamisessa SOC-arkkitehtuurissa. SOC-tiimit voivat jakaa IoC-tietoja, uhkakuvauksia ja analyysituloksia luotettavien kumppaneiden ja yhteisön kollegojen kanssa, mikä mahdollistaa kollektiivisen puolustusmenetelmän tietoturvauhkien havaitsemiseksi ja lieventämiseksi. Indikaattorien hallinta- ja korrelaatio-ominaisuudet auttavat SOC-tiimejä tunnistamaan kuvioita, suhteita ja mahdollisia uhkakampanjoita eri uhkatiedustelulähteistä. Korreloimalla indikaattoreita voidaan paljastaa piilossa olevia yhteyksiä ja saada paremman käsityksen uhkakuvasta. Integrointi ulkoisiin tietolähteisiin ja analyysityökaluihin antaa SOC-tiimeille mahdollisuuden rikastuttaa uhkatiedustelua kontekstuaalisilla lisätiedoilla. Tämä rikastaminen parantaa uhkatiedon tarkkuutta ja relevanssia, mikä mahdollistaa paremman päätöksenteon ja paremman reagoinnin tietoturvauhkiin. (MISP n.d.)

MISP:n yhteistyöominaisuudet helpottavat yhteisiä tutkimuksia ja tietojen jakamista SOC-ryhmien välillä. Analyytikot voivat tehdä yhteistyötä uhkatutkimuksissa, jakaa havaintoja ja osallistua SOC-yhteisön kollektiiviseen tiedonkeruuseen, mikä edistää ennakoivaa ja yhteistoiminnallista turvallisuuslähestymistapaa. Tuki automaattiselle indikaattorisyötteiden sisäänotolle antaa SOC-tiimeille mahdollisuuden pysyä ajan tasalla uusimmista uhkatiedoista. Tuomalla automaattisesti syötteitä luotettavista lähteistä SOC-arkkitehtuurit voivat parantaa uhkien havaitsemisvalmiuksiaan ja varmistaa, että ne ovat ajoissa tietoisia uusista uhkista. (MISP n.d.)

SOC-arkkitehtuurissa olisi luotava MISPin puitteissa asianmukaiset tietosuoja- ja tiedonjakokontrollit. Tähän sisältyy pääsynvalvonnan määrittely, jakosopimusten tekeminen ja sen varmistaminen, että arkaluonteiset tai

luottamukselliset tiedot suojataan asianmukaisesti ja jaetaan vain luotettavien tahojen kanssa. SOC-tiimien olisi varmistettava MISPin kautta jaettujen uhkatiedustelutietojen laatu ja luotettavuus. Tietojen validointitarkistusten toteuttaminen, tietolähteiden luotettavuuden varmistaminen ja tietojen eheyttä koskevien käytäntöjen edistäminen auttavat ylläpitämään jaetun uhkatiedustelun tarkkuutta ja tehokkuutta. MISPin integrointi muihin SOC-komponentteihin, kuten SIEM-alustoihin tai vaaratilanteisiin reagointijärjestelmiin, parantaa uhkatiedon jakamisen yleistä tehokkuutta. Uhkatiedustelusyötteiden vastaanoton automatisointi ja MISPin integrointi muihin tietoturvatyökaluihin virtaviivaistaa työnkulkua ja parantaa uhkatiedustelun ajantasaisuutta ja merkityksellisyyttä SOC-arkkitehtuurissa. SOC-analyytikkojen olisi saatava koulutusta ja perehdyttävä MISPin toimintoihin, työnkulkuihin ja uhkatiedustelun jakamisen parhaisiin käytäntöihin. Näin varmistetaan MISPin ominaisuuksien käytön tuloksellinen käyttö, edistetään yhteistyötä ja maksimoidaan jaetun uhkatiedustelun hyödyt SOC-ympäristössä. (MISP n.d.)

3.9 Suricata: Verkon tunkeutumisen havaitsemisjärjestelmä

Suricata on suorituskykyinen avoimen lähdekoodin verkon tunkeutumisen havaitsemisjärjestelmä (IDS) (Fortinet 2023b) ja tunkeutumisen estojärjestelmä (IPS) (Fortinet 2023c), jota käytetään laajalti SOCissa. Se tarjoaa reaaliaikaisen verkkoliikenteen analyysin, uhkien havaitsemisen ja reagointiominaisuudet, mikä tekee siitä olennaisen osan konttipohjaisissa SOC-arkkitehtuureissa. (Suricata 2023a.)

Suricata suorittaa verkkoliikenteen pakettien syvätarkastuksen (DPI) (Larsson 2022) analysoimalla paketteja reaaliaikaisesti mahdollisten tietoturvauhkien tunnistamiseksi. Se tukee useita protokollia ja tarjoaa laajan näkyvyyden verkkoviestintään, minkä ansiosta voidaan havaita epäilyttävää toimintaa ja mahdollisia hyökkäyksiä. Suricata käyttää signatuuripohjaisia

havaintomekanismeja tunnettujen haitallisten toimintamallien tai tunnettujen hyökkäyssignatuurien tunnistamiseen. (Suricata 2023b.) Se vertaa verkkoliikennettä signatuuritietokantaan ja laukaisee hälytyksiä tai ryhtyy ennaltaehkäiseviin toimiin, kun löydetään vastaavuus, mikä auttaa SOC-tiimejä reagoimaan tunnettuihin uhkiin. Suricata tukee myös poikkeavuuksien havaitsemista, jossa verkon käyttäytymistä seurataan normaalista poikkeavien ilmiöiden varalta. Se voi havaita tuntemattomia tai nollapäivähyökkäyksiä analysoimalla liikenteen ominaisuuksia, liikennemääriä tai protokollan poikkeamia, mikä tarjoaa SOC-tiimeille ennakoivia uhkien havaitsemisominaisuuksia. (Suricata 2023c.)

Suricata suorittaa perusteellisen protokolla-analyysin, jossa verkkoliikennettä analysoidaan eri protokollien ja sovelluskerroksen protokollien tietojen poimimiseksi. Se voi poimia verkkoliikenteestä tiedostoja lisäanalyysiä ja -tarkastusta varten, mikä helpottaa mahdollisten haittaohjelmien tai epäilyttävän sisällön tunnistamista. Integroituu uhkatietosyötteisiin ja tietokantoihin, minkä ansiosta SOC-arkkitehtuurit voivat parantaa havaitsemisvalmiuksiaan. Hyödyntämällä ulkoista uhkatiedustelua Suricata voi havaita IoCt tai tunnetut haitalliset IP-osoitteet, mikä tarjoaa lisäkerroksen puolustukseen kehittyneitä uhkia vastaan. Suricata mahdollistaa mukautettujen sääntöjen luomisen tiettyjen verkkopohjaisten uhkien havaitsemiseksi tai SOCin erityisympäristöön räätälöityjen sääntöjen luomisen. SOC-analyytikot voivat määritellä omia sääntöjään tiettyjen uhkien havaitsemiseksi ja niihin reagoimiseksi, mikä mahdollistaa IDSän ja IPSän -ominaisuuksien hienojakoisen hallinnan ja mukauttamisen. (Suricata 2023c.)

Suricatan reaaliaikaiset verkkoliikenteen analyysitoiminnot antavat SOC-tiimeille mahdollisuuden havaita uhkia ja reagoida niihin heti, kun niitä ilmenee. Valvomalla verkkoliikennettä ja analysoimalla paketteja reaaliaikaisesti Suricata tunnistaa haitallisia toimintoja, epäilyttävää käyttäytymistä tai tunnettuja hyökkäyssignaaleja, jolloin voidaan reagoida nopeasti. Toimii IDSnä ja IPSnä, jolloin SOC-arkkitehtuurit voivat havaita ja estää verkkopohjaisia hyökkäyksiä ennakoivasti. Se laukaisee hälytyksiä, estää haitallisen liikenteen tai toteuttaa

ennaltaehkäiseviä toimia määriteltyjen sääntöjen ja signatuurien perusteella, mikä vähentää hyökkäyspintaa ja parantaa tietoturvaa. (Suricata 2023d.)

Suricatan poikkeavuuksien havaitsemisominaisuudet mahdollistavat tuntemattomien tai nollapäivän uhkien tunnistamisen, jotka eivät vastaa tunnettuja hyökkäyssignaaleja. Valvomalla verkon käyttäytymistä ja tunnistamalla poikkeamat normaaleista malleista Suricata voi havaita epäilyttäviä toimintoja tai mahdollisia hyökkäyksiä, jotka saattavat ohittaa perinteiset signatuuripohjaiset havaitsemismenetelmät. Kyky poimia tiedostoja ja suorittaa syvällinen protokolla-analyysi helpottaa rikosteknistä analyysiä ja tapahtumiin reagointia SOC-arkkitehtuurissa. Suricatalla voi tutkia kaapattuja verkkopaketteja, poimia olennaisia tiedostoja ja analysoida verkkoviestintää saadakseen tietoa tietoturvaloukkauksista, helpottaakseen vaaratilanteiden tutkintaa ja tukeakseen rikosteknistä analyysia (Suricata 2023d). Suricata integroituu SIEM-järjestelmiin ja muihin SOC-työkaluihin, mikä mahdollistaa saumattoman tiedonvaihdon ja parantaa yleistä tietoturvatoimintaa. Integrointi SIEM-alustojen kanssa mahdollistaa verkkopohjaisten tapahtumien korreloinnin muiden tietoturvatapahtumien ja lokitietojen kanssa, mikä antaa kokonaisvaltaisen kuvan tietoturvaloukkauksista. (Suricata 2023d.)

Suricatan tuloksellinen käyttö edellyttää asianmukaista sääntöjen hallintaa ja mukautusta. SOC-tiimien tulisi päivittää ja hienosäätää havaintosääntöjä säännöllisesti, jotta ne vastaisivat kehittyviä uhkia, poistaisivat vääriä positiivisia tuloksia ja varmistaisivat optimaalisen havaintotarkkuuden kuormittamatta järjestelmää tarpeettomilla hälytyksillä (Suricata 2023c). SOC-tiimin on optimoitava Suricatan suorituskyky, jotta se pystyy käsittelemään suuria verkkoliikennemääriä tinkimättä havaintotarkkuudesta. Oikea laitteiston kokoonpano, pakettien kaappausmääritysten säätäminen ja monisäikeisyysominaisuuksien hyödyntäminen auttavat varmistamaan, että Suricata toimii tehokkaasti ja tuloksellisesti SOC-ympäristössä (Suricata 2023f).

Suricatan signatuuritietokantojen säännöllinen päivittäminen ja integrointi uhkatietosyötteisiin ovat ratkaisevan tärkeitä, jotta pysytään ajan tasalla uusimpien uhkatietojen kanssa (Suricata 2023b). SOC-tiimien olisi luotava

prosessit, joiden avulla ne päivittävät säännöllisesti signatuureja ja uhkatiedustelulähteitä havaitsemisominaisuuksien parantamiseksi. SOC-tiimin luotava asianmukaiset valvontamekanismit, joilla varmistetaan Suricatan toimivuus ja suorituskyky. Lisäksi olisi oltava käytössä tehokkaat hälytysten käsittelyprosessit, joilla voidaan priorisoida Suricatan käynnistämät hälytykset ja reagoida niihin, jotta häiriötilanteisiin voidaan reagoida ajoissa ja minimoida vasteajat (Suricata 2023c). SOC-analyytikkojen olisi saatava asianmukaista koulutusta, jotta he ymmärtävät Suricatan toiminnot, sääntöjen luomisen ja hälytysten tulkinnan. Suricatan tulosten, lokimuotojen ja muiden SOC-työkalujen integroinnin tuntemus mahdollistaa Suricatan ominaisuuksien tehokkaan hyödyntämisen SOC-ympäristössä.

4 Tietoturvatyökalujen kontitus

SOC-arkkitehtuurin muodostavien tietoturvatyökalujen kontitukseen valitut tietoturvatyökalut otetaan käyttöön Docker-kontteina, jotta voidaan muodostaa integroitu ja skaalautuva SOC-ekosysteemi. Hahmotellaan kunkin tietoturvatyökalun asteittainen konttiprosessi, jolla varmistetaan niiden toimiva käyttöönotto ja vuorovaikutus konttipohjaisessa SOC-ympäristössä.

Virallinen Portainer Docker-näköistiedosto valitaan luotettavasta arkistosta luotettavuuden ja turvallisuuden varmistamiseksi. Portainer konfiguroidaan asianmukaisilla asetuksilla, kuten ylläpitäjän tunnistetiedoilla, pääsynvalvonnalla, sen verkkopohjaisen käyttöliittymän suojaamiseksi. Docker-volyymit kartoitetaan Portainerin tietojen pysyväksi, tällä varmistetaan, että konfigurointiasetukset ja -tiedot säilyvät konttien uudelleenkäynnistysten aikana. Portainer otetaan käyttöön konttina, ja sen verkkopohjainen käyttöliittymä tulee SOC-ylläpitäjien saataville.

Elasticsearch ja Kibana on kontitettu reaaliaikaisen lokitallennuksen, -analyysin ja -visualisoinnin mahdollistamiseksi. Viralliset Elasticsearch- ja Kibana Docker-näköistiedostot valitaan luotettavuuden ja yhteensopivuuden vuoksi. Elasticsearch ja Kibana konfiguroidaan asianmukaisilla asetuksilla klusterin muodostamista, indeksointia ja reaaliaikaista visualisointia varten. Kontit liitetään mukautettuun Docker-verkkoon sisäisen viestinnän helpottamiseksi. Isäntäkoneelle on kartoitettu tarvittavat portit, joiden kautta Kibanan web-käyttöliittymää voidaan käyttää.

Logstash ja Filebeat on kontitettu lokien keräämistä, käsittelyä ja välittämistä varten. Viralliset Logstash- ja Filebeat Docker-näköistiedostot valitaan hyvämaineisista arkistoista. Logstash ja Filebeat konfiguroidaan asianmukaisilla tulo-, suodatin- ja lähtöasetuksilla lokitietojen käsittelyä ja välittämistä varten. Kontit liitetään mukautettuun Docker-verkkoon saumatonta viestintää varten Elasticsearchin kanssa. Docker-volyymit kartoitetaan pysyviin Logstash- ja Filebeatin konfiguraatiotiedostoihin. Logstash ja Filebeat otetaan käyttöön erillisinä kontteina, jotka ovat valmiita käsittelemään ja lähettämään lokitietoja.

MISP uhkatiedustelun jakamisalusta, on kontitettu helpottamaan tietojen vaihtoa muiden SOC-työkalujen kanssa. Valitaan virallinen MISP:n Docker-näköistiedosto sen luotettavuuden ja yhteensopivuuden varmistamiseksi. MISP konfiguroidaan oikeilla asetuksilla uhkatiedustelutietojen jakamista ja turvallista viestintää varten. MISP:lle perustetaan tietokanta uhkatiedustelutietojen tallentamista ja hallintaa varten. MISP-kontti liitetään mukautettuun Docker-verkkoon, jotta se voi olla vuorovaikutuksessa muiden SOC-komponenttien kanssa.

IDPSnä toimiva Suricata on kontitettu verkkoliikenteen valvomista ja mahdollisten uhkien havaitsemista varten. Valitaan virallinen Suricata Docker näköistiedosto sen luotettavuuden ja yhteensopivuuden varmistamiseksi. Suricata konfiguroidaan asianmukaisilla asetuksilla verkon seurantaa, sääntöjen hallintaa ja hälytyksiä varten. Suricata-kontti liitetään mukautettuun Docker-verkkoon, jotta se voi olla vuorovaikutuksessa muiden SOC-komponenttien kanssa.

5 Integrointi ja automatisointi

Integroinnilla tarkoitetaan eri konttikohtaisten tietoturvatyökalujen yhteistyötä, jonka avulla ne voivat jakaa dataa ja työskennellä tehokkaasti yhdessä. Automaatio tarkoittaa automatisoitujen prosessien ja työnkulkujen käyttöönottoa SOC-toimintojen ja tietoturvaongelmiin reagoimisen tehostamiseksi. Datan integrointi on olennaisen tärkeää SOCille, jotta se voi korreloida eri lähteistä saatuja tietoja ja luoda kattavan käsityksen tietoturvatapahtumista. Konttipohjaisessa SOC-arkkitehtuurissa datan integrointi saavutetaan seuraavilla tavoilla: konttipohjaisten tietoturvatyökalujen, kuten Suricatan, Filebeatin ja MISPin, tuottamat lokitiedot keskitetään Elasticsearchiin.

Tämä helpottaa useista lähteistä peräisin olevien lokitietojen korrelointia ja analysointia. Logstash on konfiguroitu putkistoilla, joiden avulla voidaan käsitellä ja rikastaa eri konteista peräisin olevia lokitietoja. Nämä putkilinjat suodattavat, muuntavat ja aggregoivat tietoja ennen niiden lähettämistä Elasticsearchiin indeksointia varten. Jakaa MISPin tietokokonaisuuksia, kuten uhkaindikaattoreita ja haittaohjelmanäytteitä, muiden työkalujen, kuten Suricatan, kanssa uhkien havaitsemisvalmiuksien parantamiseksi.

Automaatiolla on ratkaiseva rooli häiriötilanteisiin vastaamisen nopeuttamisessa ja manuaalisten toimenpiteiden vähentämisessä. Suricata havaitsee mahdollisia tunkeutumisia, automaattiset hälytykset luodaan ja välitetään se välitöntä reagointia varten. MISP voidaan määrittää automatisoimaan asiaankuuluvien uhkakuvatietojen jakaminen luotettavien kumppaneiden ja vertaisten kanssa ennalta määritettyjen sääntöjen ja käytäntöjen perusteella.

6 Käyttötapaustutkimus

Käyttötapaustutkimuksessa tutkittiin SOCin arkkitehtuurin rakennusta ja käyttöönottoa Ubuntupalvelinvirtuaalikoneessa, joka toimii Proxmox-palvelimella. SOC on rakennettu käyttäen Docker-konttiteknologiaa, joka sisältää keskeisiä työkaluja. SOC-infrastruktuuri koostuu useista Docker-konttista, jotka on järjestetty kolmen eri Docker-Compose-tiedoston avulla.

6.1 Proxmoxin yleiskatsaus

Proxmoxvirtualisointiympäristö (Proxmox VE) (Proxmox 2023) on avoimen lähdekoodin virtualisointialusta, jossa yhdistyvät kaksi keskeistä virtualisointitekniikkaa. Virtualisointi käyttäen Kernel-pohjaista virtuaalikonetta (KVM) (Proxmox 2023) täydelliseen laitteistovirtualisointiin ja konttipohjainen virtualisointi käyttäen Linux kontteja (LXC) (Proxmox 2023). Proxmox tarjoaa kattavan ratkaisun virtuaalikoneiden (VM) (VMware 2023), konttien, tallennuksen, verkon ja korkean saatavuuden klusteroinnin hallintaan yhdellä integroidulla alustalla. Proxmoxia hallitaan käyttäjäystävällisen web-pohjaisen käyttöliittymän kautta, joten virtuaalikoneiden, konttien, tallennustilojen ja verkkojen määrittäminen ja valvonta sekä hallinta on helppoa. Proxmox tukee erilaisia tallennusvaihtoehtoja, mukaan lukien paikallinen tallennus, verkkoon liitetty tallennus (NAS) (Seagate 2023), tallennusalueverkot (SAN) (Techtarget n.d.b) ja ohjelmistokohtaiset tallennusratkaisut. Tämän joustavuuden ansiosta Proxmoxia voi räätälöidä tallennusinfrastruktuurin tarpeiden mukaan. Alusta tarjoaa työkaluja verkkoliitäntöjen ja siltojen sekä virtuaalilähiverkkojen (VLAN) (Techtarget n.d.c) hallintaan, minkä ansiosta voi määrittää ja hallita virtuaalikoneiden ja konttien verkkoyhteyksiä. (Proxmox 2023.)

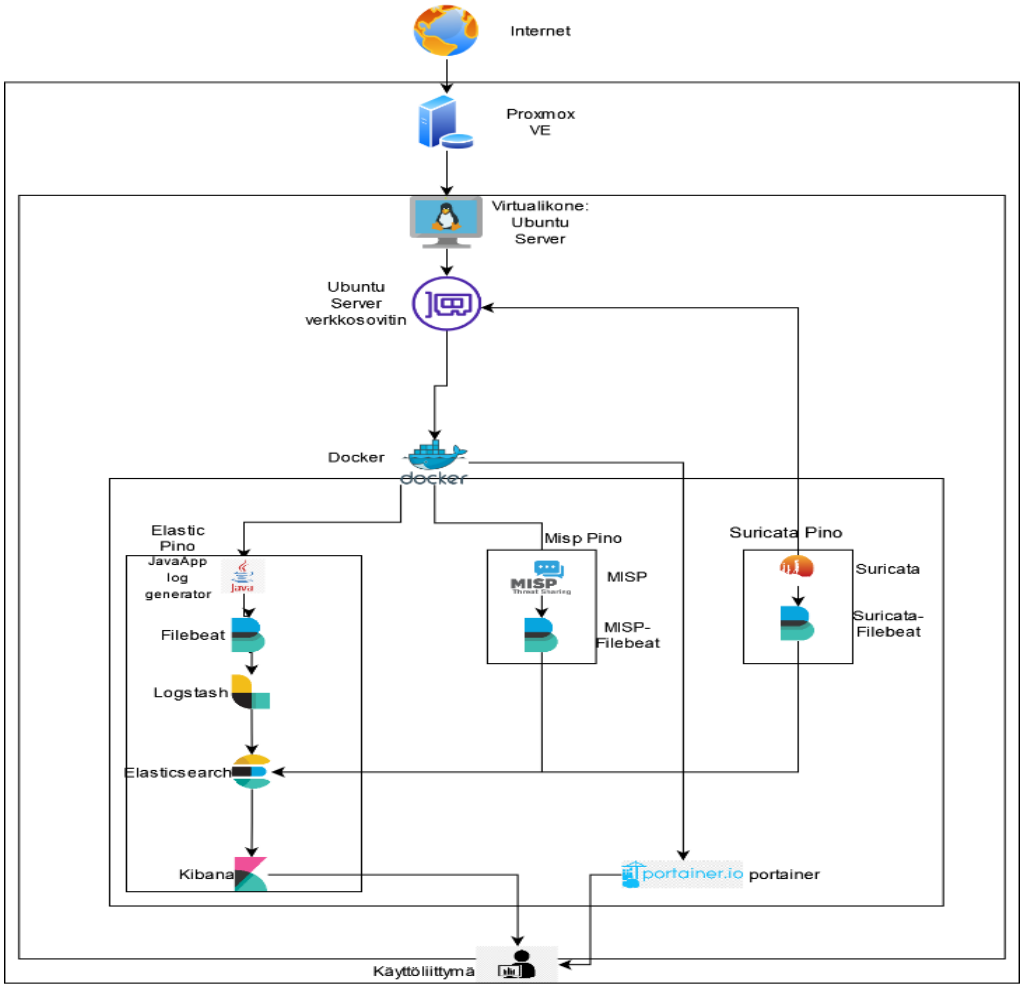
Proxmoxilla voi luoda useista noodeista koostuvan klusterin, joka tarjoaa virtuaalikoneille ja konteille korkean käytettävyyden sekä vikasietoisuusominaisuudet. Jos yksi noodi vikaantuu, työkuorma voidaan

siirtää automaattisesti terveeseen noodiin. Proxmoxissa on integroituja työkaluja virtuaalikoneiden ja konttien varmuuskopioiden luomiseen ja sillä voidaan ajoittaa varmuuskopiot ja palauttaa ne tarvittaessa. Proxmox on avoimen lähdekoodin ohjelmisto ja on saatavilla ilmaiseksi, mutta siitä on myös tilauspohjainen malli, joka tarjoaa lisäominaisuuksia, tukea ja päivityksiä. (Proxmox 2023.)

Proxmox tukee sekä perinteisiä virtuaalikoneita että kevyitä kontteja yhdessä integroidussa ympäristössä. Sitä käytetään usein erilaisiin tarkoituksiin, kuten kehitykseen ja testaukseen, palvelinten konsolidointiin, pilvipalveluihin ja muihin tarkoituksiin. (Proxmox 2023.)

6.2 Käyttötapaus SOC arkkitehtuurin yleiskatsaus

SOC-arkkitehtuuri koostuu Ubuntupalvelimesta, jota isännöidään Proxmox-palvelimella. Virtuaalikoneen sisällä käytetään Docker-kontteja SOC-komponenttien asennukseen. Käyttöönotossa käytetään kolmea Docker Compose -tiedostoa eri komponenttikokonaisuuksia varten. Portaineria käytetään erillisenä konttina hallinnoimaan ja orkestroimaan muita kontteja. (Kuva 1.)



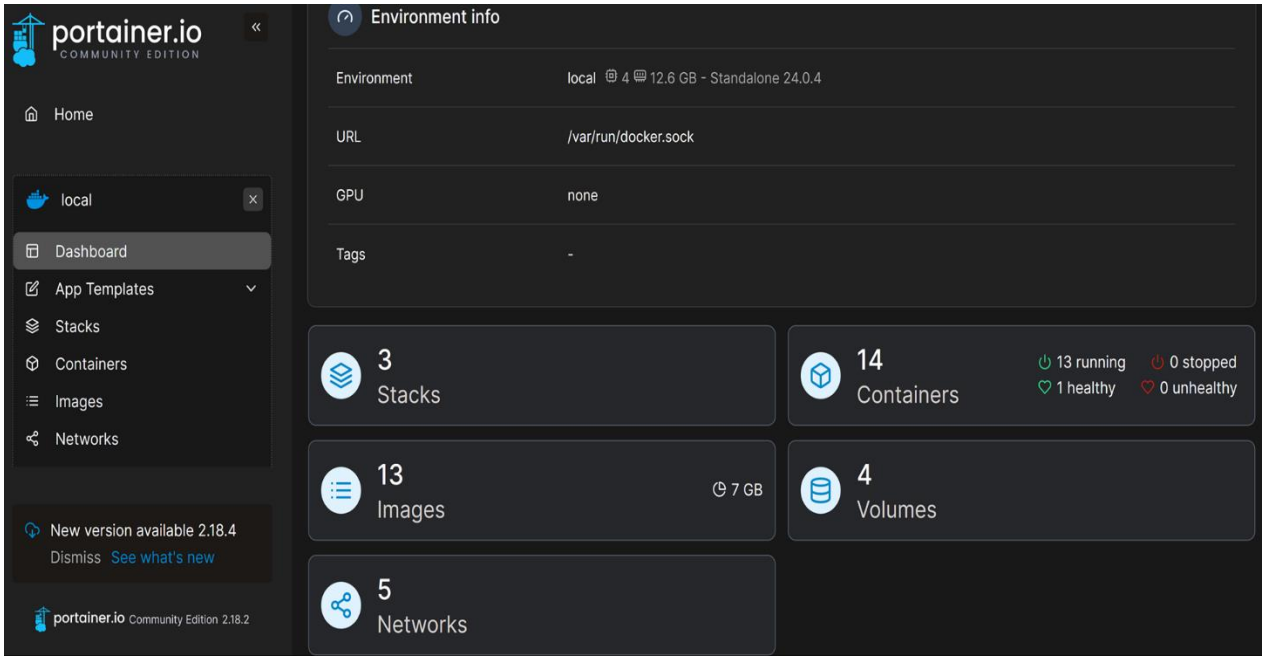
Kuva 1. Korkeatason arkkitehtuurikuva.

6.3 Docker ja Portainer

Yksi tärkeimmistä vaiheista konttipohjaisen SOCin asentamisessa oli Dockerin, ympäristön ytimen muodostavan kontitusalustan asentaminen. Docker valittiin sen suosion, hyvän käyttäjätuen ja konttien hallinnan helppouden vuoksi. Asennus sisälsi seuraavat vaiheet: ubuntu VM:n pakettivarastot päivitettiin, jotta varmistetaan pääsy uusimpiin Docker-ohjelman paketteihin. Docker asennettiin käyttämällä virallista Docker-tietovarastoa uusimman vakaan version varmistamiseksi. Tähän sisältyi komentojen suorittaminen arkiston lisäämiseksi,

paketti-indeksin päivittämiseksi ja Dockerin asentamiseksi. Asennuksen jälkeen käynnistettiin Docker-palvelu konttien luomista ja hallintaa varten ja varmistettiin, että Docker käynnistyy aina, kun ubuntu VM käynnistyy.

Konttien hallinnan ja seurannan yksinkertaistamiseksi konttipohjaiseen SOC-ympäristöön otettiin käyttöön Portainer. Portainerin verkkopohjainen käyttöliittymä tuo helppokäyttöisen alustan konttien hallintaan ja orkestrointiin. Portainerin asennus oli mutkaton prosessi, Portainer asennettiin itse Docker-kontiksi. Tämä tarkoitti Docker-komennon suorittamista Portainer-kontin luomiseksi ja tarvittavien asetusten, kuten volyymin kiinnityksen ja porttisidonnan, määrittämistä, kun Portainer-kontti oli toiminnassa, ylläpitäjä pääsee Portainer-dashboardiin (Kuva 2) verkkoselaimen kautta. Dashboard antoi näkymän kaikista konteista ja niiden tilasta ympäristössä.



Kuva 2. Portainer Dashboardi

6.4 Docker Compose -tiedostot

Docker Compose -tiedosto on konfiguraatiotiedosto, jota käytetään usean kontin Docker-sovellusten määrittelyyn ja hallintaan (Docker Docs 2023f).

6.4.1 Yleiskatsaus Docker Compose-tiedostoon

Docker Compose -tiedosto on YML/YAML konfigurointitiedosto, joka helpottaa usean kontin Docker-sovellusten määrittelyä, konfigurointia ja orkestrointia. Sen avulla voi määritellä sovelluksen eri komponentit, niiden väliset sidokset, konfiguraatiot ja muut käyttöönoton kannalta tarpeelliset seikat. Versio määrittää käytettävän Docker Compose -tiedostosyntaksin version. Tämä auttaa varmistamaan yhteensopivuuden Docker Compose -työkalun ja sen ominaisuuksien kanssa. Palvelut ovat sovelluksen keskeiset rakennuspalikat, jotka määritellään "services"-osiossa. Kukin palvelu vastaa konttia ja edustaa sovelluksen tiettyä komponenttia, kuten verkkopalvelinta, tietokantaa tai sovelluspalvelinta.

Näköistiedosto määrittää palvelussa käytettävän Docker-näköistiedoston. Se voi olla Docker Hubin virallinen näköistiedosto tai rakennettu ja mukautettu näköistiedosto. Portit määrittää isäntäkoneen ja kontin väliset porttikartoitukset, jotka mahdollistavat kommunikoinnin palvelun kanssa. Ympäristömuuttujat mahdollistavat ympäristömuuttujien asettamisen kontissa. Volyymit ovat hyödyllinen sovelluksen konfigurointivaihtoehtojen välittämiseen ja määrittävät kontin sisällä kiinnitettävät datavolyymit, mikä varmistaa datan pysyvyyden ja jakamisen isäntäkoneen ja kontin välillä. Verkot määrittävät palvelun yhteen tai useampaan Docker-verkkoon, mikä mahdollistaa saman verkon sisällä olevien palveluiden välisen viestinnän.

Riippuvuus määrittää palvelujen väliset riippuvuudet. Palvelu ei käynnisty ennen kuin sen riippuvaiset palvelut ovat käynnissä. Komento määrittää komennon, joka suoritetaan kontissa, kun palvelu käynnistyy. Verkot määrittävät

mukautetut Docker-verkot, joihin palvelut voidaan liittää. Verkot mahdollistavat konttien ja palveluiden välisen eristetyn viestinnän. Volyymit määrittävät nimetyt volyymit tai kiinnitykset, jotka mahdollistavat tietojen pysyvyyden ja jakamisen konttien ja isäntäjärjestelmän välillä. Konfiguraatiot ja salaukset ovat edistyneitä ominaisuuksia, joiden avulla voi hallita palvelujen konfiguraatiotiedostoja ja salauksia. (Docker Docs 2023f.)

Docker Compose yksinkertaistaa monimutkaisten sovellusten hallintaa useiden konttien avulla ja tehostaa kehityksen, testauksen ja käyttöönoton työnkulkua. Se kannustaa Infrastruktuuri koodina (Infrastructure as Code, IaC) (IBM n.d.b) -lähestymistapaan, sillä sen avulla voi määritellä sovelluksen arkkitehtuurin versioidussa tiedostossa, mikä parantaa kollaboraatiota ja toistettavuutta. (Docker Docs 2023f.)

6.4.2 Lokien hallintapinon compose tiedosto

Lokien hallintapino on konttipohjaisen SOCin keskeinen osa, jolla varmistetaan lokitietojen tehokas keruu, tallennus ja analysointi. Lokien hallintapinon konttien organisointiin ja hallintaan käytetään Docker Composea.

Elasticsearchin näköistiedostoksi on valittu versio 7.17.0. Kontti on nimetty es-node-01. Elasticsearch on konfiguroitu toimimaan porteissa 9200, 9300 ja yhdessä noodissa. Kibanan näköistiedostoksi on valittu versio 7.17.0. Kontti on nimetty kibanaksi ja konfiguroitu toimimaan portissa 5601. Ympäristömuuttujissa saapuvien palvelinkyselyjen maksimikokoa on lisätty 39321600 tavuun, http://es-node-01:9200 Elasticsearch-instanssin osoitteeksi, joita käytetään kaikkiin kyselyihin. JavaApp näköistiedostoksi on valittu febbweiss Java loki generaattori viimeisin versio. JavaApp on konfiguroitu käynnistymään vasta, kun Elasticsearch ja Kibana ovat käynnistyneet. Logstash näköistiedostoksi on valittu versio 7.17.0. Kontti on nimetty logstash-01 ja volyymeiksi on kiinnitetty polut tiedostoille logstash.conf ja logstash.yml sekä käynnistymään vasta, kun Kibana on käynnissä. Filebeat näköistiedostoksi on

valittu versio 7.17.0. Kontti on nimetty filebeat-elk. Käyttäjäksi on asetettu root. Volyymeiksi on kiinnitetty tiedosto filebeat.docker.yml ja luku oikeudet docker tiedostoihin Docker Unix liitäntään ja konttien dataan. Lokien hallintapinon kontit on konfiguroitu toimimaan Docker verkossa elastic. (Kuva 3.)

```
version: '3.8'
services:
    es-node-01:
        container_name: es-node-01
        ports:
            - '9200:9200'
            - '9300:9300'
        environment:
            - discovery.type=single-node
        image: 'docker.elastic.co/elasticsearch/elasticsearch:7.17.0'
        networks:
            - elastic
    kibana:
        container_name: kibana
        ports:
            - '5601:5601'
        environment:
            - 'ELASTICSEARCH_HOSTS=http://es-node-01:9200'
            - 'SERVER_MAXPAYLOADBYTES=39321600'
        image: 'docker.elastic.co/kibana/kibana:7.17.0'
        healthcheck:
            test: ["CMD", "curl", "-f", "kibana:5601"]
            interval: 50s
            timeout: 50s
            retries: 5
        depends_on:
            - es-node-01
        networks:
            - elastic

    javaApp:
        image: 'febbweiss/java-log-generator:latest'
        depends_on:
            - es-node-01
            - kibana
        networks:
            - elastic

    logstash:
        container_name: logstash-01
        volumes:
            - /path/to/docker-elk/logstash.conf:/usr/share/logstash/pipeline/logstash.conf
            - /path/to/docker-elk/logstash.yml:/usr/share/logstash/config/logstash.yml
        image: 'docker.elastic.co/logstash/logstash:7.17.0'
        depends_on:
             kibana:
               condition: service_healthy
        networks:
            - elastic


    filebeat:
        user: root
        container_name: filebeat-elk
        command: --strict.perms=false
        volumes:
            - /path/to/docker-elk/filebeat.docker.yml:/usr/share/filebeat/filebeat.yml
            - /var/lib/docker/containers:/var/lib/docker/containers:ro
            - /var/run/docker.sock:/var/run/docker.sock:ro
        image: 'docker.elastic.co/beats/filebeat:7.17.0'
        depends_on:
             kibana:
               condition: service_healthy
        networks:
            - elastic
networks:
  elastic:
    name: elastic
```

Kuva 3. Lokien hallintapinon compose tiedosto.

6.4.3 Uhkatiedustelutiedon jakamispinon compose tiedosto

Uhkatiedustelutiedon jakamispino helpottaa uhkatiedustelun yhteistoiminnallista jakamista ja parantaa SOCin puolustuskykyä. Uhkatiedustelutiedon jakamispinon konttien organisointiin ja hallintaan käytetään Docker Composea.

MISPin näköistiedostoksi on valittu coolacid misp.docker coren viimeisin versio. Konfiguroitu toimimaan porteissa 80 ja 443. Volyymeiksi on kiinnitetty MISPin konfiguraatiokansio, lokikansio, sovelluskansio ja sertifikaattikansio. Ympäristömuuttujissa MISPin osoitteeksi on konfiguroitu isäntäkoneen IP-osoite, Redis täysin määritellyksi verkkotunnukseksi redis, alustus MISPiin ja MISPin käyttäjän yksilöivä tunniste ajastettujen tehtävien ajamiseen. Kontti käynnistyy vasta, kun MySQL ja Redis on käynnissä. Sähköpostin lähettämiseksi käytetään näköistiedostoksi valittua Namshi SMTP konttia. Viestivälittäjänä käytetään näköistiedostoksi valittua Redis versio 5.0.6 konttia. Tietokantana käytetään näköistiedostoksi valittua MySQL versio 8.0.19. Ympäristömuuttujiksi on konfiguroitu MySQL käyttäjä, salasana, root salasana ja tietokannan nimi. Volyymiksi kiinnitetty MySQL data kansio. MISP moduulit näköistiedostoksi on valittu coolacid misp-docker moduulit viimeisin versio. Ympäristömuuttujaksi konfiguroitu Redis taustajärjestelmän nimeksi redis. Kontti käynnistyy vasta, kun Redis ja MySQL on käynnissä. Filebeat näköistiedostoksi on valittu versio 7.17.0. Käyttäjäksi asetettu root. Kontti on nimetty filebeat-misp. Volyymeiksi on kiinnitetty tiedostot filebeat.docker.yml, misp.yml ja lukuoikeudet docker tiedostoihin Dockerunixliitäntään ja konttien dataan. Uhkatiedustelutiedon jakamispinon kontit on konfiguroitu toimimaan

dockerverkossa misp. (Kuva 4.)

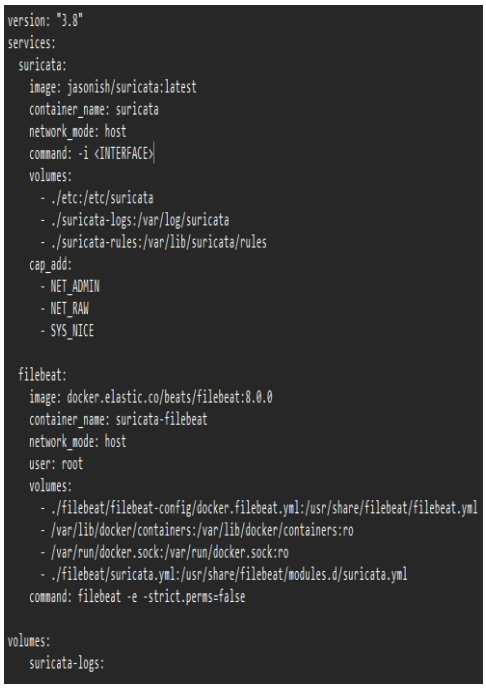
```
version: '3.8'
services:
  # This is capible to relay via gmail, Amazon SES, or generic relays
  # See: https://hub.docker.com/r/namshi/smtp
  mail:
    image: namshi/smtp
    networks:
      - misp
  redis:
    image: redis:5.0.6
    networks:
      - misp
  db:
    image: mysql:8.0.19
    command: --default-authentication-plugin=mysql_native_password
    restart: always
    environment:
      - "MYSQL_USER=misp"
      - "MYSQL_PASSWORD=example"
      - "MYSQL_ROOT_PASSWORD=password"
      - "MYSQL_DATABASE=misp"
    volumes:
      - mysql_data:/var/lib/mysql
    cap_add:
      - SYS_NICE  # CAP_SYS_NICE Prevent runaway mysql log
    networks:
      - misp
  misp:
    image: coolacid/misp-docker:core-latest
    depends_on:
      - redis
      - db
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - "./server-configs/:/var/www/MISP/app/Config/"
      - "./logs/:/var/www/MISP/app/tmp/logs/"
      - "./files/:/var/www/MISP/app/files"
      - "./ssl/:/etc/nginx/certs"
#     - "./examples/custom-entrypoint.sh:/custom-entrypoint.sh" # Use the example custom-entrypoint.sh
    environment:
      - "HOSTNAME=https://<IP>"
      - "REDIS_FQDN=redis"
      - "INIT=true"             # Initialize MISP, things includes, attempting to import SQL and the Files DIR
      - "CRON_USER_ID=1"        # The MISP user ID to run cron jobs as
#     - "SYNCSERVERS=1 2 3 4"  # The MISP Feed servers to sync in the cron job
      # Database Configuration (And their defaults)
#     - "MYSQL_HOST=db"
#     - "MYSQL_USER=misp"
#     - "MYSQL_PASSWORD=example" # NOTE: This should be AlphaNum with no Special Chars. Otherwise, edit config files after first run.
#     - "MYSQL_DATABASE=misp"
      # Optional Settings
#     - "NOREDIR=true" # Do not redirect port 80
#     - "DISIPV6=true" # Disable IPV6 in nginx
#     - "CERTAUTH=optional" # Can be set to optional or on - Step 2 of https://github.com/MISP/MISP/tree/2.4/app/Plugin/CertAuth is still required
#     - "SECURESSL=true" # Enable higher security SSL in nginx
#     - "MISP_MODULES_FQDN=http://misp-modules" # Set the MISP Modules FQDN, used for Enrichment_services_url/Import_services_url/Export_services_url
#     - "WORKERS=1" #If set to a value larger than 1 this will increase the number of parallel worker processes
    networks:
      - misp
  misp-modules:
    image: coolacid/misp-docker:modules-latest
    environment:
      - "REDIS_BACKEND=redis"
    depends_on:
      - redis
      - db
    networks:
      - misp

  filebeat:
    user: root
    container_name: filebeat-misp
    image: docker.elastic.co/beats/filebeat:7.17.0
    command: filebeat -e --strict.perms=false
    volumes:
      - ./filebeat/filebeat.docker.yml:/usr/share/filebeat/filebeat.yml
      - ./filebeat/misp.yml:/usr/share/filebeat/modules.d/threatintel.yml
      - /var/lib/docker/containers:/var/lib/docker/containers:ro
      - /var/run/docker.sock:/var/run/docker.sock:ro
    depends_on:
      - misp
    networks:
      - misp

volumes:
    mysql_data:

networks:
  misp:
    name: misp
```

Kuva 4. Uhkatiedustelutiedon jakamispinon compose tiedosto.

6.4.4 Tunkeutumisen havaitsemis- ja estojärjestelmä pinon compose tiedosto

Tunkeutumisen havaitsemis- ja estojärjestelmän pino, jonka taustalla on Suricata, joka toimii verkon tietoturvan valvojana.

Suricatan näköistiedostoksi on valittu jasonish suricata viimeisin versio ja kontti on nimetty suricataksi. Verkkotilaksi on asetettu isäntäverkko ja kontin komennoksi on asetettu tämän käyttämään isäntäkoneen verkkoliitäntää. Volyymeiksi kiinnitetty kansiot Suricatan konfiguraatioon, lokeihin ja sääntöihin. Kontin käyttösäännöiksi on asetettuna poistaa root oikeudet ja priorisoi kontin prosessit sekä oikeudet suorittaa kaikki verkkotoiminnot. Filebeat näköistiedostoksi on valittu versio 8.0.0 ja käyttäjäksi asetettu root. Kontti on nimetty suricata-filebeat. Volyymeiksi on kiinnitetty tiedostot docker.filebeat.yml (Kuva 10), suricata.yml (Kuva 11) ja lukuoikeudet docker tiedostoihin Docker Unix liitäntään ja konttien dataan. Tunkeutumisen havaitsemis- ja estojärjestelmän pinon kontit on konfiguroitu toimimaan isäntäkoneen verkossa. (Kuva 5.)



```
version: "3.8"
services:
  suricata:
    image: jasonish/suricata:latest
    container_name: suricata
    network_mode: host
    command: -i <INTERFACE>
    volumes:
      - ./etc:/etc/suricata
      - ./suricata-logs:/var/log/suricata
      - ./suricata-rules:/var/lib/suricata/rules
    cap_add:
      - NET_ADMIN
      - NET_RAW
      - SYS_NICE

  filebeat:
    image: docker.elastic.co/beats/filebeat:8.0.0
    container_name: suricata-filebeat
    network_mode: host
    user: root
    volumes:
      - ./filebeat/filebeat-config/docker.filebeat.yml:/usr/share/filebeat/filebeat.yml
      - /var/lib/docker/containers:/var/lib/docker/containers:ro
      - /var/run/docker.sock:/var/run/docker.sock:ro
      - ./filebeat/suricata.yml:/usr/share/filebeat/modules.d/suricata.yml
    command: filebeat -e -strict.perms=false

volumes:
    suricata-logs:
```

Kuva 5. Tunkeutumisen havaitsemis- ja estojärjestelmän pinon compose tiedosto.

6.5 SOC-komponenttien konfigurointi

Nämä konfiguraatiot ovat tärkeitä hyvin organisoidun ja tehokkaan konttipohjaisen SOCin luomisessa, jotta varmistetaan, että SOC voi tehokkaasti valvoa, havaita ja reagoida tietoturvauhkiin.

Filbeat konfiguraatiotiedostossa asetettu kontin debuggauslokit. Asetettu polku Filebeatmoduulille. Filebeatiin on konfiguroitu automaattisen haun komponentti keräämään JavaApp dataa Dockerista sekä julkaisemaan ja lähettämään datan Logstash osoitteeseen logstash-01:5400. (Kuva 6.)

```
logging.level: debug
logging.to_files: true
logging.files:
  path: /var/log/filebeat
  name: filebeat
  keepfiles: 7
  permissions: 0600

filebeat.config:
  modules:
    path: ${path.config}/modules.d/*.yml
    reload.enabled: false


filebeat.autodiscover:
  providers:
    - type: docker
      templates:
      - condition:
            equals:
              docker.container.image: febbweiss/java-log-generator:latest
        config:
          - type: container
            multiline.type: pattern
            multiline.pattern: '^[[:space:]]+(at|\.{3})[[:space:]]+\b|^java.lang.RuntimeException:'
            multiline.negate: false
            multiline.match: after
            paths:
              - /var/lib/docker/containers/${data.docker.container.id}/*.log
            encoding: utf-8


processors:

- add_locale:
      format: abbreviation
- add_host_metadata: ~


output.logstash:
   hosts: ["logstash-01:5400"]
```

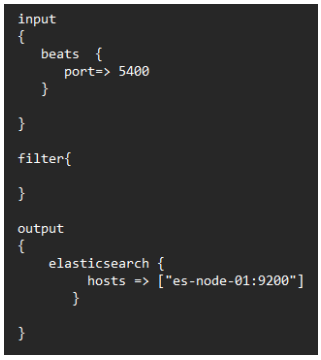
Kuva 6. filebeat.docker.yml.

Asetukset ovat asetettu Logstashin konfiguraatiotiedostossa alkuperäisasetuksille http.host joka on ohjelmointirajapinnan päätepisteen sidontaosoite on määritetty 0.0.0.0 ja xpack jolla hallitaan datan keräämistä noodeista on otettu pois käytöstä. (Kuva 7.)

```
http.host: "0.0.0.0"
xpack.monitoring.elasticsearch.hosts: [ "http://es-node-01:9200" ]

xpack.monitoring.enabled: false
```

Kuva 7. logstash.yml.

Logstash.conf tiedostossa määritellään, että sisääntulossa Filebeat syöttää lokitiedot porttiin 5400, suodatusta ei ole käytössä ja ulostuloksi valittu Elasticsearch ja osoitteeksi es-node-01 ja portiksi 9200. (Kuva 8.)



Kuva 8. logstash.conf.

Uhkatiedustelutiedon jakamispinon konfigurointitiedostot filebeat.docker.yml ja misp.yml. Tiedostossa filebeat.docker.yml määriteltiin Filebeat sisääntuloksi lokitietojen lähteet, joita Filebeatin tulee seurata, jotka ovat polut /docker-misp/logs ja /var/log/*.log. Määritetty Filebeatin MISPmoduulille polku Filebeat kontin moduuli listaan ja asetettu sieltä moduuli käyttöön. Shardeja on lisätty neljä jakamaan työkuormaa, parantamaan skaalautuvuutta ja vikasietoisuutta sekä tämä nopeuttaa MISPistä tuleva datan kulkua. Kibanan osoitteeksi on asetettu isäntäkoneen IP-osoite ja portiksi 5601 ja ulostuloksi määritetty Elasticsearchosoitteeksi isäntäkoneen IP-osoite ja portiksi 9200. (Kuva 9.)

```
filebeat.inputs:
    type: log
    enabled: true
    paths:
      - /path/to/docker-misp/logs
      - /var/log/*.log

filebeat.config:
  modules:
    path: ${path.config}/modules.d/*.yml
    reload.enabled: false
    enabled: true

setup.template.settings:
  index.number_of_shards: 4



setup:
  kibana.host: "http://<IP>:5601"

output.elasticsearch:
  hosts: "http://<IP>:9200"
  workers: 4


processors:
  - add_host_metadata:
      when.not.contains.tags: forwarded
  - add_docker_metadata: ~
```

Kuva 9. filebeat.docker.yml.

Tässä kohdassa määritellään threatintelmoduulin MISP asetukset, määritetään tiedosto muodoksi json ja päätepisteeseen yhdistävä osoite mistä tiedot siirtyvät MISPistä Filebeatiin, jotta tiedot siirtyvät Filebeatiin tarvitsee asettaa sovellusrajapinnan avain. Moduuliin asetetaan rajoitukset käsiteltäville tiedonsiirtopyynnöille. Filebeatin käynnistyttyä on määritetty kuinka vanhoja tapauksia hakea ja kuinka usein sovellusrajapinnasta haetaan päivitettyjä dataa. (Kuva 10.)

```
- module: threatintel
  misp:
    enabled: true
    var.input: httpjson
    var.url: https://<IP>/events/restSearch/
    var.api_token: <API KEY>
    var.http_request_body.limit: 1000
    var.ssl.verification_mode: none
    var.first_interval: 24h
    var.interval: 60
```

Kuva 10. misp.yml.

Tunkeutumisen havaitsemis- ja estojärjestelmäpinon konfigurointitiedostot docker.filebeat.yml ja suricata.yml. Tiedostossa docker.filebeat.yml määriteltiin Filebeat sisääntuloksi lokitietojen lähteet, joita Filebeatin tulee seurata, jotka ovat polut /var/log/suricata/*.log ja /*.json. Filebeatin Suricata moduulille on määritetty polku Filebeat kontin moduuli listaan. Kibanan host osoitteeksi on asetettu isäntäkoneen IP-osoite ja portiksi 5601 sekä ulostuloksi määritetty Elasticsearchosoitteeksi isäntäkoneen IP-osoite ja portiksi 9200. (Kuva 11.)

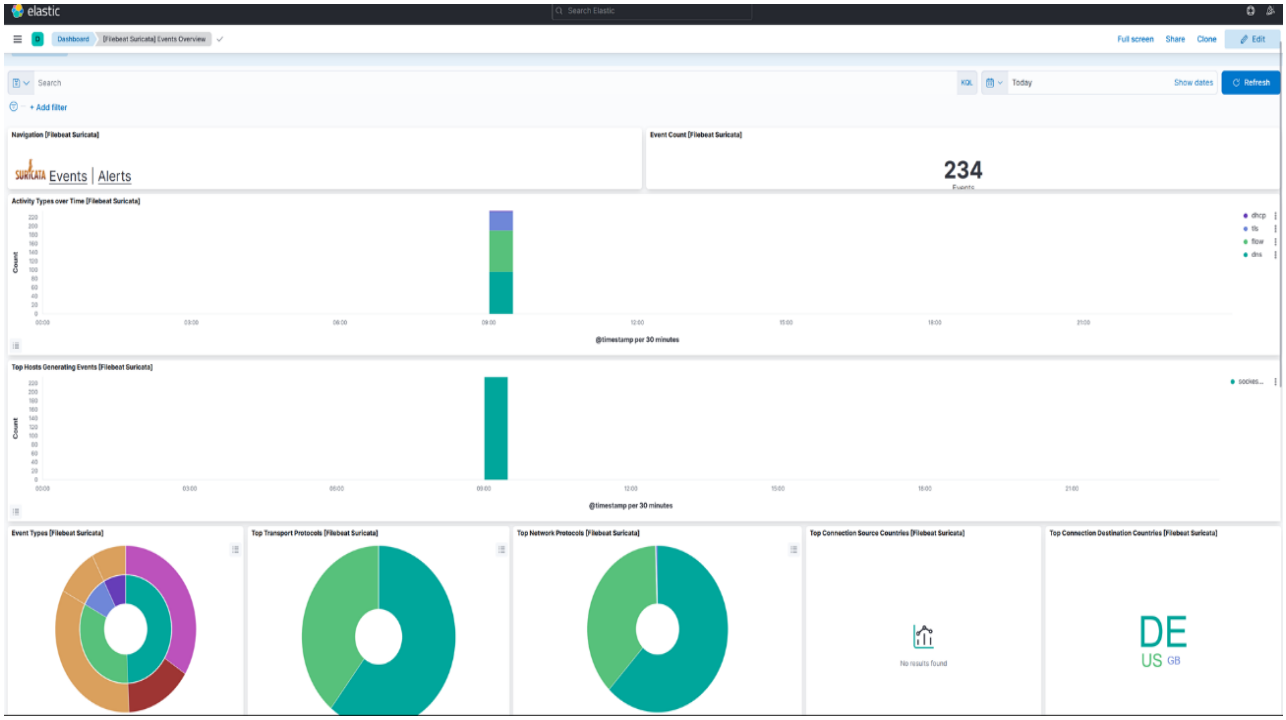
```
filebeat.inputs:
    type: log
    enabled: true
    paths:
      - /var/log/suricata/*.log
      - /var/log/suricata/*.json

filebeat.config.modules:
  path: /usr/share/filebeat/modules.d/*.yml
  reload.enabled: false

setup:
  kibana.host: "http://<IP>:5601"


output.elasticsearch:
  hosts: "http://<IP>:9200"
  #workers: 4
  ssl.verification_mode: none

processors:
  - add_host_metadata:
      when.not.contains.tags: forwarded
  - add_docker_metadata: ~
```

Kuva 11. docker.filebeat.yml.

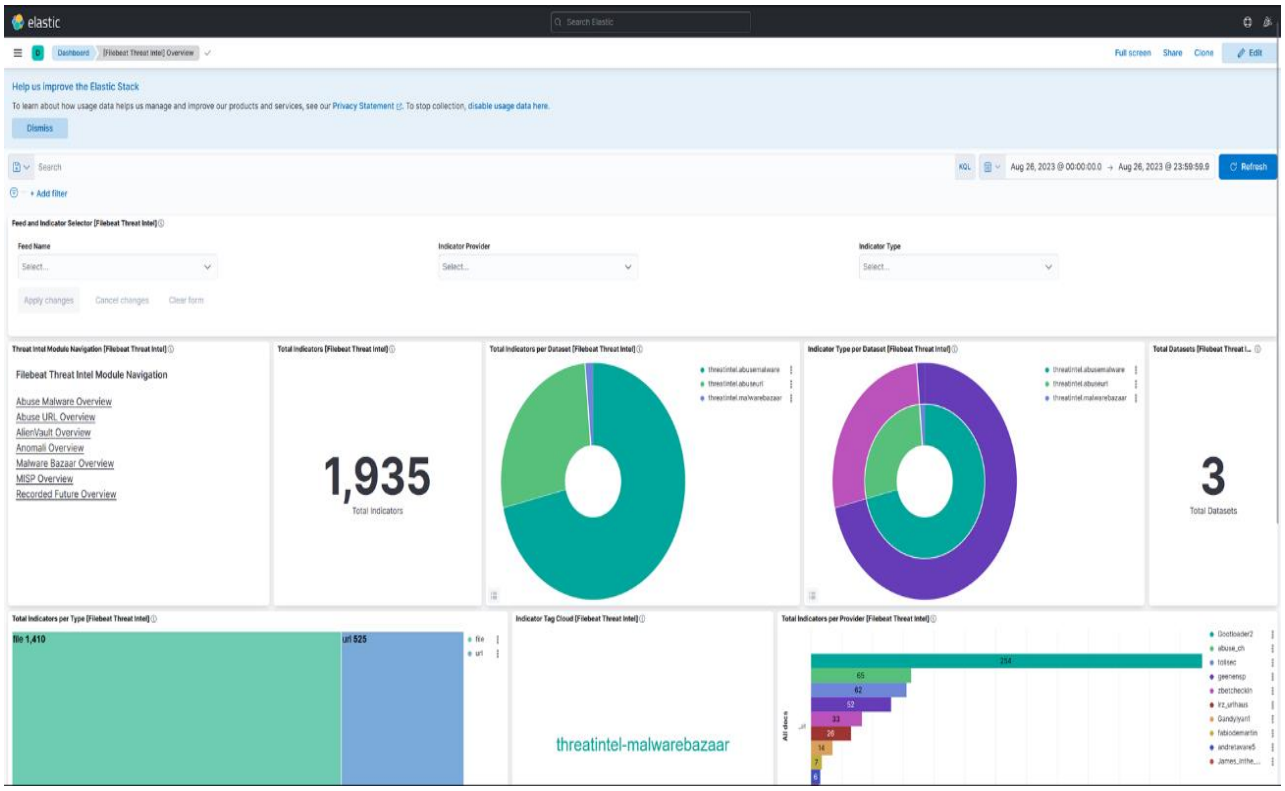
Asetetaan suricatamoduulin eve osa käyttöön ja polku mistä data haetaan var/log/suricata/eve.json. (Kuva 12.)

```
- module: suricata
  eve:
    enabled: true
    var.paths: ["/var/log/suricata/eve.json"]
```

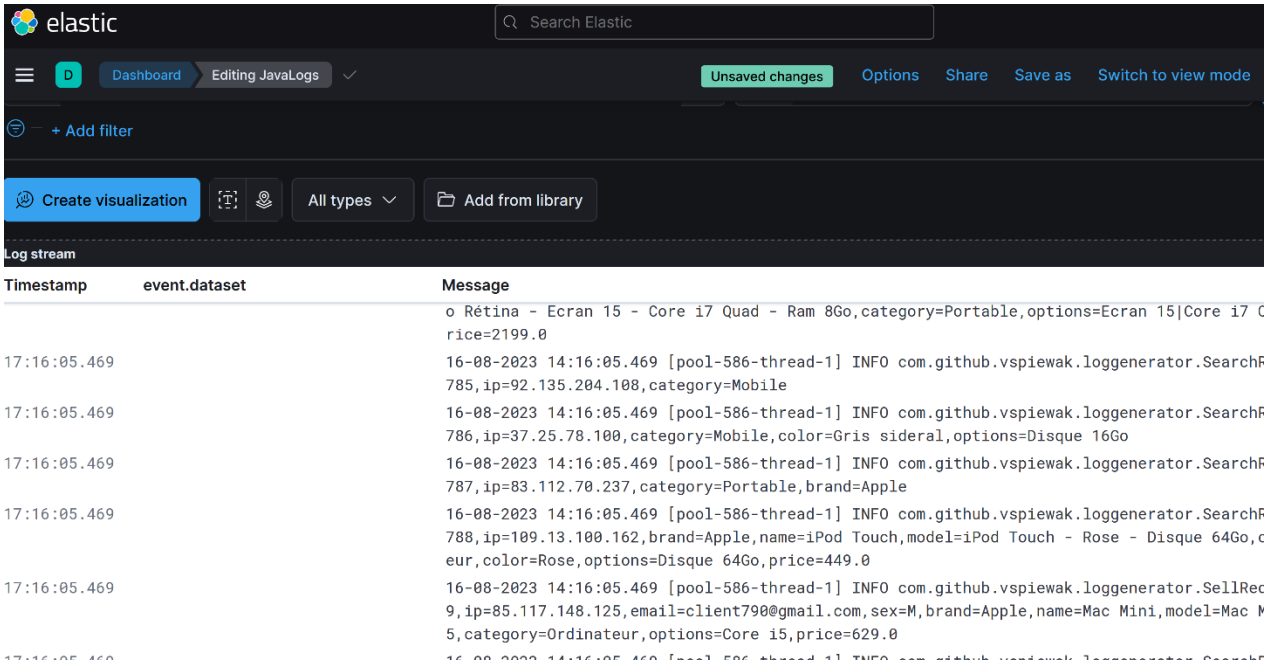
Kuva 12. suricata.yml.



Kuva 13. Suricata Events dashboardi.



Kuva 14. MISP yleiskuva dashboardi.



Kuva 15. JavaApp loki dashboardi

6.6 Vertailu Perinteiseen SOCiin

Kontit ovat resurssitehokkaita, ne jakavat isäntäkäyttöjärjestelmän kernelin, mikä minimoi kustannukset, tämä johtaa tehokkaaseen resurssien käyttöön ja kustannussäästöihin. Perinteiset järjestelmät eivät ole niin resurssitehokkaita, tämä voi johtaa resurssien riittämättömyyteen ja infrastruktuurikustannusten kasvuun. Kontitus mahdollistaa SOC-komponenttien nopean skaalautumisen, uusia kontteja voidaan lisätä tai vähentää tarpeen mukaan, mikä varmistaa, että SOC mukautuu muuttuviin työtehtäviin. Perinteisen SOCin skaalaus edellyttää usein lisälaitteiston tai VMien hankkimista ja konfigurointia, mikä on yleensä hitaampi ja vähemmän joustava prosessi. Kontit tarjoavat tehokkaan prosessieristyksen, mikä vähentää riskiä siitä, että yksi komponentti vaikuttaa muihin. Kukin kontti eristää riippuvuutensa, mikä parantaa turvallisuutta. Perinteisissä SOC-järjestelmissä samassa isäntäjärjestelmässä tai VMsä olevat

komponentit voivat jakaa riippuvuuksia, mikä voi aiheuttaa yhteensopivuus- ja tietoturvahaasteita.

Kontit voidaan ottaa käyttöön nopeasti, usein muutamassa sekunnissa, tämän nopean käyttöönoton ansiosta SOC-tiimit voivat reagoida nopeasti uusiin uhkiin. Uusien komponenttien käyttöönotto perinteisessä SOCissa voi vaatia pidempiä odotusaikoja laitteiston tai VMien käyttöönoton vuoksi. Konttien orkestrointityökalut, kuten Docker Compose ja Portainer, tarjoavat yksityiskohtaista hallintaa. Tämä joustavuus tehostaa SOC-toimintoja. Perinteiset SOC-ympäristöt saattavat olla riippuvaisia manuaalisista konfiguraatioista, eikä niissä ole samaa joustavuuden tasoa. Kontit ovat erittäin siirrettäviä, SOC-komponentteja voidaan siirtää helposti eri ympäristöjen, kuten kehitys-, testaus- ja tuotantoympäristöjen välillä. Perinteiset SOC-järjestelmät voivat kohdata haasteita, kun komponentteja siirretään eri ympäristöjen välillä riippuvuuksien ja kokoonpanojen vuoksi.

7 Päätelmä

Kontteihin koottuja SOC-komponentteja voidaan helposti skaalata ylös tai alaspäin tarpeiden mukaan. Kontit jakavat käyttöjärjestelmän kernelin, mikä optimoi resurssien käytön ja vähentää ylikuormitusta sekä Docker-konttien resurssitarve on pienempi kuin perinteisten virtuaalikoneiden. Kontteina toimivat SOC-komponentit kuluttavat vähemmän järjestelmäresursseja, mikä mahdollistaa laitteiston tehokkaamman käytön ja vähentää infrastruktuurikustannuksia. (Docker Docs 2023a.)

Kontit eristävät komponentit, estävät ristiriidat ja edistävät järjestelmän turvallisuutta. Kontit tarjoavat prosessitasoisen eristyksen, jolla varmistetaan, että SOC-komponentit toimivat turvallisessa ja valvotussa ympäristössä. Jokaisella kontilla on oma tiedostojärjestelmänsä, verkkopintansa ja prosessitilansa, mikä minimoi yksittäisten komponenttien tietoturvaloukkausten tai haavoittuvuuksien vaikutukset. (Docker Docs 2023a.)

Docker-kontit mahdollistavat SOC-komponenttien nopean käyttöönoton, mikä vähentää käyttöönottoon ja konfigurointiin kuluvaa aikaa ja vaivaa. Voidaan ottaa nopeasti käyttöön uusia kontti-instansseja, mikä mahdollistaa tietoturvatyökalujen nopeamman käyttöönoton ja lyhentää uusien SOC-ominaisuuksien käyttöönottoaikaa. Kontit varmistavat yhdenmukaisuuden kehitys-, testaus- ja tuotantoympäristöissä. Kontteihin pakattuja SOC-komponentteja voidaan helposti siirtää eri infrastruktuuriasetusten välillä, mikä takaa SOC-ympäristöjen yhdenmukaisuuden ja toistettavuuden. (Docker Docs 2023a.)

Useiden konttien orkestrointi vaatii huolellista konfigurointia ja synkronointia. Vaikka kontit mahdollistavat eristämisen, konttien turvallisuuden varmistaminen on ratkaisevan tärkeää. On otettava käyttöön hyviä turvallisuuskäytäntöjä, kuten luotettavien perusnäköistiedostojen käyttö, konttien näköistiedostojen säännöllinen päivittäminen ja asianmukaisten käyttöoikeuksien valvonta konttipohjaisiin SOC-komponentteihin. Resurssien asianmukainen jakaminen kullekin kontille on ratkaisevan tärkeää optimaalisen suorituskyvyn kannalta.

Konttien välinen verkottaminen ja konttipohjaisten SOC-komponenttien integroiminen olemassa olevaan verkkoinfrastruktuuriin voi olla monimutkaista. (Docker Docs 2023b.)

Konttipohjaisen SOCin käyttöönotto oli osoitus nykyaikaisten tietoturvakäytäntöjen sopeutumiskyvystä ja kestävyydestä. Tämä opinnäytetyö on kuvannut prosessia, jossa konttipakkausteknologia on otettu käyttöön SOC-ominaisuuksien toiminnan tehokkuuden parantamiseksi. Konttipohjaisuus on kevyen, skaalautuvan ja siirrettävän luonteensa vuoksi osoittautunut sopivaksi teknologiaksi SOCissa. Sen kyky eristää sovelluksia ja niiden riippuvuuksia antaa SOC-tiimeille mahdollisuuden reagoida ketterästi uusiin uhkiin ja optimoida samalla resurssien käyttöä. Opinnäytetyön infrastruktuuri oli rakennettu hyödyntäen Proxmox-virtualisointia, jolla isännöitiin ubuntu virtuaalikonetta, joka oli asennettu Docker-konttien ajamiseen. Tämä arkkitehtuuri tarjosi vakaan ja turvallisen alustan SOCin toiminnoille.

Dockerin asentaminen keskeiseksi konttialustaksi ja Portainerin asentaminen konttien hallintatyökaluksi yksinkertaisti konttien hallintaa. Keskeisten SOC-pinojen, lokien hallintapinon, uhkatiedon jakamispinon ja tunkeutumisen havaitsemis- ja estopinon, konfigurointi osoitti, että saumattomaan integrointiin huolellisuutta on noudatettava. Elasticsearch, Kibana, Logstash, Filebeat, MISP, Suricata ja muut komponentit sovitettiin onnistuneesti yhteen tehokkaan SOC-ekosysteemin luomiseksi. Nämä kokoonpanot helpottivat keskitettyä lokien hallintaa, reaaliaikaista analyysia, uhkatiedon yhteistoiminnallista jakamista ja tunkeutumisen havaitsemista ja estämistä, jotka kaikki edistivät joustavamman SOCin luomista.

Tämän konttipohjaisen SOC-toteutuksen hyödyt ovat moninaisia. SOCin käyttäjät voivat nyt käyttää keskitettyä lokien hallintaa ja reaaliaikaista tiedon visualisointia Kibana dashboardien (Kuva 13, Kuva 14, Kuva 15) avulla. MISPin mahdollistama uhkatiedon yhteistoiminnallinen jakaminen on parantanut ennakoivan puolustuksen valmiuksia. Suricatan syvä pakettitarkastus parantaa verkkoturvallisuuden valvontaa. Kontituksen ketteryys, skaalautuvuus ja resurssitehokkuus antavat SOCille mahdollisuuden mukautua vaihtuviin

vaatimuksiin ja minimoida samalla infrastruktuurikustannukset. Asianmukainen konfiguraationhallinta, tarkka resurssien jakaminen sekä versioiden hallinta ovat edelleen ensiarvoisen tärkeitä. Kattava lokitus on olennaista SOCin tehokkuuden varmistamisessa. Lopuksi opinnäytetyössä korostuu konttiteknologian mahdollisuudet tehdä uudistuksia SOC-toiminnoissa. Se korostaa jatkuvan parantamisen ja jatkuvan toiminnan kehittämisen tärkeyttä, sillä ne ovat nykyaikaisten kyberturvallisuuskäytäntöjen välttämättömiä osia. Ottamalla käyttöön konttiteknologian voidaan navigoida monimutkaisessa ja jatkuvasti muuttuvassa uhkamaisemassa entistä joustavammin ja tehokkaammin.

Lähteet

Computer hope 2021. GUI. Viitattu 10.6.2023.
https://www.computerhope.com/jargon/g/gui.htm

Docker 2023. Docker. Viitattu 17.5.2023. https://www.docker.com/

Docker Docs 2023a. Docker get started. Viitattu 19.5.2023.
https://docs.docker.com/get-started/overview/

Docker Docs 2023b. Security best practices. Viitattu 25.5.2023.
https://docs.docker.com/develop/security-best-practices/

Docker Docs 2023c. Orchestration. Viitattu 30.5.2023.
https://docs.docker.com/get-started/orchestration/

Docker Docs 2023d. Devs best practices. Viitattu 1.6.2023.
https://docs.docker.com/develop/dev-best-practices/

Docker Docs 2023f. Compose features. Viitattu 3.6.2023.
https://docs.docker.com/compose/features-uses/

Elasticsearch 2023. Elasticsearch. Viitattu 20.5.2023.
https://www.elastic.co/elasticsearch/

Elastic 2023. Elastic-Stack Features. Viitattu 21.5.2023.
https://www.elastic.co/elastic-stack/features

Filebeat 2023a. Filebeat. Viitattu 28.5.2023.
https://www.elastic.co/beats/filebeat

Filebeat 2023b. Securing communication elasticsearch. Viitattu 8.6.2023.
https://www.elastic.co/guide/en/beats/filebeat/current/securing-communication-elasticsearch.html

Fortinet 2023a. What is SOC. Viitattu 15.5.2023
https://www.fortinet.com/lat/resources/cyberglossary/what-is-soc

Fortinet 2023c. What is an IPS. Viitattu 25.6.2023.
https://www.fortinet.com/resources/cyberglossary/what-is-an-ips

Fortinet 2023b. Intrusion Detection System. Viitattu 26.6.2023. https://www.fortinet.com/resources/cyberglossary/intrusion-detection-system

IBM n.d.a. Security Operations center. Viitattu 16.6.2023. https://www.ibm.com/topics/security-operations-center

IBM n.d.b. Infrastructure as Code. Viitattu 3.7.2023. https://www.ibm.com/topics/infrastructure-as-code

Iiot-world 2022. Top challenges SOC are facing. Viitattu 22.5.2023. https://www.iiot-world.com/ics-security/cybersecurity/top-challenges-soc-are-facing/

Kibana 2023. Kibana. Viitattu 23.5.2023. https://www.elastic.co/kibana/#enterprise-search

Larsson, E. 2022. Boosting suricata with next gen deep packet inspection. Viitattu 14.7.2023. https://cybersecurity-magazine.com/boosting-suricata-with-next-gen-deep-packet-inspection/

Logstash 2023. Logstash. Viitattu 26.5.2023. https://www.elastic.co/logstash

Logsign 2019. What is IoC in cyber security. Viitattu 20.6.2023. https://www.logsign.com/blog/what-is-ioc-in-cyber-security/

MISP n.d. MISP. Viitattu 29.5.2023. https://www.misp-project.org/features/

Mohanakrishnan, R. 2022. What is IDPS. Viitattu 9.6.2023. https://www.spiceworks.com/it-security/vulnerability-management/articles/what-is-idps/

Portainer 2023a. Portainer. Viitattu 24.5.2023. https://docs.portainer.io/

Portainer 2023b. Dashboard. Viitattu 24.5.2023. https://docs.portainer.io/user/docker/dashboard

Portainer 2023c. Containers/view. Viitattu 26.5.2023. https://docs.portainer.io/user/docker/containers/view

Portainer 2023d. Containers/logs. Viitattu 29.5.2023. https://docs.portainer.io/user/docker/containers/logs

Portainer n.d.a. Docker-swarm container management. Viitattu 31.5.2023. https://www.portainer.io/docker-swarm-container-management-platform-gui

Portainer n.d.b. Features. Viitattu 1.6.2023. https://www.portainer.io/features

Proxmox 2023. Proxmox. Viitattu 1.8.2023. https://www.proxmox.com/en/proxmox-virtual-environment/features

Salinas, S. 2023. Security Operations Center a quick start guide. Viitattu 14.5.2023. https://www.exabeam.com/security-operations-center/security-operations-center-a-quick-start-guide/

Seagate 2023. What is NAS. Viitattu 3.8.2023. https://www.seagate.com/gb/en/blog/what-is-nas-master-ti/

Suricata 2023a. What is Suricata. Viitattu 9.6.2023. https://docs.suricata.io/en/latest/what-is-suricata.html

Suricata 2023b. Quickstart. Viitattu 10.6.2023 https://docs.suricata.io/en/latest/quickstart.html#

Suricata 2023c. rules. Viitattu 30.6.2023. https://docs.suricata.io/en/suricata-6.0.0/rules/intro.html

Suricata 2023d. Features. Viitattu 18.6.2023. https://suricata.io/features/

Suricata 2023f. Run modes. Viitattu 8.7.2023. https://docs.suricata.io/en/latest/performance/runmodes.html

Teceze 2021. Pros and Cons of SOC. Viitattu 18.5.2023. https://www.teceze.com/Pros-and-Cons-of-SOC

Techtarget n.d.a. SOC. Viitattu 27.5.2023. https://www.techtarget.com/searchsecurity/definition/Security-Operations-Center-SOC

Techtarget n.d.b. Storage Area Network. Viitattu 22.7.2023. https://www.techtarget.com/searchstorage/definition/storage-area-network-SAN

Techtarget n.d.c. Virtual LAN. Viitattu 25.7.2023. https://www.techtarget.com/searchnetworking/definition/virtual-LAN

VMware, 2023. Virtual Machine. Viitattu 29.7.2023.
https://www.vmware.com/topics/glossary/content/virtual-machine.html