Otto Kemppainen

Falco – Linux tietoturvavalvonta

Tietojenkäsittely

Tradenomi

Syksy 2021

KAMK • University of Applied Sciences

Tiivistelmä

Tekijä(t): Kemppainen Otto

Työn nimi: Falco – Linux tietoturvavalvonta

Tutkintonimike: Tradenomi, tietojenkäsittely

Asiasanat: Tietoturva, Linux, virtualisointi

Opinnäytetyön tavoitteena oli tutkia Sysdig-yrityksen kehittämän Falco-tietoturvatyökalun soveltuvuutta ulkoistetulle CSOC:lle. Opinnäytetyön toimeksiantajana toimi Loihde Trust Oy, joka tarjoaa kyberturva- ja yritysverkkopalveluita sekä fyysiseen turvallisuuteen kuuluvia valvontakamera- ja lukitusratkaisuja. Loihde Trust Oy on osa Loihde-konsernia. CSOC on organisaation tietoturvatiimi, joka valvoo ja tutkii tietoturvapoikkeamia. Falco on tietoturvatyökalu, jolla voidaan havaita uhkia Linux-palvelimilla. Falco on kehitetty pääasiassa konttiteknologialla rakennettujen ympäristöjen valvontaan. Uhkien havainnointi tapahtuu seuraamalla Linuxin järjestelmäkutsuja, joiden perusteella Falcoon voidaan rakentaa erilaisia sääntöjä.

Falco asennettiin Loihde Trustin testiympäristöön. Testiympäristö koostui Falcosta, Logstashista ja SIEM-järjestelmästä. Logstash toimii keskitettynä keräimenä Falcon hälytyksille ja välittää hälytykset eteenpäin SIEM-järjestelmään, jonka kautta Loihde Trustin SOC-analyytikot näkisivät hälytykset tuotantoympäristössä. Falcon säännöstöä testattiin tekemällä omia sääntöjä tunnettujen haavoittuvuuksien hyväksikäyttöyritysten havainnointiin sekä yleisesti epäilyttävän toiminnan huomaamiseen. Testisäännöt havaitsivat Redis-tietosäilön ja sudon tunnetut haavoittuvuudet. Epäilyttävää toimintaa havainnoitiin tekemällä sääntö, joka havaitsee listattuihin osoitteisiin liikennöinnin. Samalla testattiin oletussäännöstön toimintaa ja mahdollisten väärien hälytysten määrää.

Testauksen perusteella Falcon havaittiin tuovan lisäarvoa tietoturvavalvontaan. Järjestelmäkutsuihin perustuva säännöstö sallii todella tarkkojen sääntöjen luonnin verrattuna CSOC:n muihin työkaluihin. Falcoa ei voida siltikään käyttää ainoana tietoturvatyökaluna, koska Falco lähettää vain hälytyksen aiheuttaneen prosessin tiedot CSOC:lle. Tarkempaa tutkintaa varten tarvitaan lisätietoja, joita voidaan saada muiden CSOC:n muiden työkalujen avulla. Falcon laajemmalle käyttöönotolle havaittiin myös haasteita. Falcon säännöstön päivitystä ei voida tehdä etänä pelkästään Falcon avulla, vaan sitä varten tarvitaan erillinen työkalu automatisointia varten. Toinen haaste on Falcon yleinen ylläpito ja vianselvitys, koska CSOC:lla ei yleensä ole pääsyä asiakkaiden tuotantopalvelimille. Haasteista huolimatta, Falcoa voidaan harkita käytettäväksi erikoistapauksissa, varsinkin konttiteknologialla rakennetuissa ympäristöissä, joihin tarvitaan lisävalvontaa järjestelmäkutsuihin perustuvalla säännöstöllä.

KAMK • University of Applied Sciences

Abstract

Author(s): Kemppainen Otto

Title of the Publication: Falco – Linux Security Monitoring

Degree Title: Bachelor of Business Administration, Business Information Technology

Keywords: Cybersecurity, Linux, virtualization

The goal of the thesis was to investigate cybersecurity tool Falco from the viewpoint of a MSSP CSOC. The work in this thesis was made for Loihde Trust Oy, which offers various cyber- and physical security services. Loihde Trust Oy is part of Loihde Group. CSOC is part of the organization's security team, who monitors and investigates security incidents. Falco is a cybersecurity tool, which can be used to detect various threats on Linux servers. Falco is capable of monitoring environments made with containers. Falco detects threats by monitoring Linux system calls, which can be used to make rules in Falco.

Falco was installed on Loihde Trust's test environment. The test environment included Falco, Logstash and a SIEM cluster. In production environments Loihde Trust's SOC analytics would get the alerts via the SIEM system. Falco's rules were tested by making new rules based on known vulnerabilities and other potentially malicious activity. The test rules detected attempts to exploit vulnerabilities in Redis datastore and sudo. Potentially malicious activity was detected with a rule, which alerted on connections to listed IP addresses. Default rules were tested to see if they generated lots of falsepositive alerts.

Testing revealed that Falco does bring additional insight to SOC analysts. Rules based on Linux system calls allow CSOC to create very detailed rules, compared to other CSOC tools. However, Falco can't be used as the only cybersecurity tool, as Falco only sends details about the alert and nothing else. SOC analysts often need to do additional forensics to find out if the alert is result of something malicious, for example through SIEM or EDR. A few problems need to be solved before Falco can be considered for broad use in CSOC. Falco doesn't have a feature that allows rules to be updated remotely. Updating the rules would require some other tool or script to be used. The second problem is troubleshooting and updating Falco. CSOC usually doesn't have access to any customers production servers, so all actions would require help from customers IT-department. Even with these problems, use of Falco can be considered in some cases. Especially in environments that extensively use containers, which would require additional security monitoring that can be achieved with rules based in Linux system calls.

Sisällys

Symboliluettelo

C2 Command and Control -palvelimien avulla haittaohjelmien kehittäjät voivat komentaa haittaohjelmia etänä.

CSOC Cyber Security Operations Center valvoo organisaation digitaalista tietoturvaa.

EDR Endpoint Detection and Response kerää päätelaitteiden tapahtumat ja hälyttää havaitessaan haitallista toimintaa.

MSSP Managed Security Service Provider tarjoaa ulkoistettuja tietoturvapalveluita organisaatioille.

NDR Network Detection and Response -järjestelmä kuuntelee verkkoliikennettä ja hälyttää havaitessaan haitallista toimintaa.

RCE Remote Code Execution on haavoittuvuus, jonka avulla hyökkääjä voi ajaa omaa koodia kohdejärjestelmässä.

SIEM Security Information and Event Management. Järjestelmä, joka etsii haitallista toimintaa lokidatan avulla.

1 Johdanto

Opinnäytetyön aiheena on Linux-pohjaisten palvelinten tietoturvavalvonnan kehittäminen, käyttäen hyväksi Yhdysvaltalaisen Sysdig-yrityksen kehittämää Falco-tietoturvatyökalua. Opinnäytetyön toimeksiantajana toimii Loihde Trust Oy.

Loihde Trust tarjoaa mm. CSOC-palvelua, jossa CSOC-analyytikot käsittelevät eri tietoturvajärjestelmien huomaavia poikkeamia. Opinnäytetyön ideana on tutkia Falco-työkalun soveltuvuutta CSOC:n työkaluna. Linux-laitteiden valvontaan CSOC:lla on tällä hetkellä pääasiassa käytössä EDR sekä SIEM.

SIEM-järjestelmällä tehdyssä valvonnassa kaikki data täytyy lähettää SIEM-järjestelmään, jotta niistä voidaan huomata poikkeamat. Mitä tarkemmin palvelinta halutaan valvoa, sitä enemmän dataa täytyy lähettää SIEM-järjestelmään, mikä nostaa kustannuksia. EDR-järjestelmässä poikkeamia voidaan huomata käyttäen järjestelmän sisäänrakennettua heuristiikkaa ja tekoälyä sekä tekemällä sääntöjä EDR-järjestelmään lähetetyn datan perusteella. Falco toimii täysin itsenäisesti, eli sen käsittelemää dataa ei tarvitse lähettää toiseen järjestelmään analysointia varten. Itsenäinen toiminta sallii Falcon valvoa huomattavasti laajemmin laitteen toimintaa. Falco osaa käsitellä myös konttiteknologialla, esimerkiksi Dockerilla, rakennettua ympäristöä.

Opinnäytetyössä tullaan testaamaan Falcon toimintaa käytännössä. Falco asennetaan testipalvelimille, jossa testataan Falcon sisäänrakennettujen sääntöjen toimintaa. Falcon säännöstöä kehitetään tekemällä sääntöjä tunnistamaan haitallista toimintaa. Konttiympäristöjen valvontaa tullaan testaamaan ajamalla Dockerilla kontteja testipalvelimella. Tieto hälytyksistä lähetetään Loihteen SIEM-järjestelmään, jonka kautta CSOC-analyytikot näkevät hälytykset.

Työ tullaan tekemään Loihteen testiympäristössä. Testipalvelimet asennetaan eristettyyn virtualisointiympäristöön, jotta testit eivät vaikuta Loihteen tuotantopalveluihin.

2 Toimeksiantaja ja toimeksianto

Opinnäytetyön toimeksiantajana toimii Loihde Trust Oy. Loihde Trust on Loihde-konserniin kuuluva yritys, joka on yksi Suomen suurimpia yritysturvallisuuspalveluiden toimittajia. Loihde Trustin tuottamiin palveluihin kuuluu mm. kyberturva- ja yritysverkkopalveluita sekä fyysiseen turvallisuuteen kuuluvia valvontakamera- ja lukitusratkaisuja. [1.]

Loihde-konserni jakautuu kahteen eri osa-alueeseen: turvaratkaisuihin sekä digitaaliseen kehittämiseen. Turvaratkaisujen perusajatuksena on ”Yksi turvallisuus”. Turvaratkaisuihin kuuluvat mm. Loihde Trust Oy:n kyberturvallisuus sekä fyysisen turvallisuuden palvelut, Spellpoint Oy:n identiteetin- ja pääsynhallinnan palvelut sekä Tansec Oy:n ilmoituksensiirtojärjestelmät. Digitaalinen kehittäminen sisältää mm. Bitfactor Oy:n ohjelmistokehitys-, design- ja analytiikkapalveluita, Aureolis Oy:n data-analytiikkaan sekä Talent Base Oy:n dataan liittyvät konsultointipalvelut. [1.]

Vuonna 2020 Loihde-konsernin liikevaihto oli 106,8 miljoonaa euroa ja oikaistu käyttökate 6,2 miljoonaa euroa. Konserni työllisti 714 henkilöä vuonna 2020. [2.]

Toimeksiantona oli kehittää Linux-pohjaisten palvelinten ja konttiteknologialla rakennettujen palvelinten valvontaa. Tutkittavaksi työkaluksi valittiin yhdysvaltalaisen Sysdig-yrityksen kehittämä Falco.

Opinnäytetyössä testataan Falcon toimintaa käytännössä. Falco asennetaan testipalvelimille, jossa testataan Falcon sisäänrakennettujen sääntöjen toimintaa. Falcon säännöstöä kehitetään tekemällä sääntöjä tunnistamaan mahdollisesti haitallista toimintaa. Konttiympäristöjen valvontaa testataan ajamalla Dockerilla kontteja testipalvelimella. Tieto hälytyksistä lähetetään Loihteen SIEM-järjestelmään, jonka kautta CSOC-analyytikot näkevät hälytykset.

Lopputuloksena Loihde Trustille raportoidaan tuotteen soveltuvuudesta CSOC:n työkaluksi sekä mahdolliset ongelmakohdat, joita opinnäytetyön aikana löydetään.

3 Konttiympäristöjen tietoturvavalvonta

Tässä kappaleessa käydään läpi projektiin liittyvien käsitteiden teoriataustaa. Ensin avataan CSOC:n merkitystä organisaatioille. Seuraavaksi käsitellään virtualisoinnin ja konttiteknologian eroja. Viimeiseksi käydään läpi Falcon toimintaa.

3.1 Cyber Security Operations Center

CSOC vastaa organisaation tietoturvasta ja mahdollisten tietoturvapoikkeamien tutkinnasta. CSOC voi olla osa organisaatiota tai ostettu palveluna tietoturvapalveluntarjoajalta.

Gartnerin määrittelemää SOC-kolmiota käytetään yhä CSOC:n näkyvyyden kuvaamiseen. SOC-kolmio koostuu SIEM-järjestelmästä, NDR-järjestelmästä ja EDR-palvelusta. SIEM-järjestelmän kautta CSOC saa näkyvyyden mm. autentikointitapahtumiin, verkkoliikenteeseen ja audit-lokiin. SIEM-järjestelmän toimii pääasiassa analysoimalla lokiviestejä. NDR:ssä tallennetaan tieto verkkoyhteyksistä ja niiden sisällöstä. Verkkoliikennettä ei tarvitse tallentaa kokonaan, vaan näkyvyys voidaan toteuttaa käyttämällä esimerkiksi avoimen lähdekoodin Zeek-sovellusta, joka tallentaa verkkoyhteyksistä metatietoa. EDR analysoi päätelaitteilla ajettavia sovelluksia ja hälyttää huomatessaan mahdollisesti haitallisen sovelluksen. EDR:n keräämää tietoa voidaan tarvittaessa analysoida myös jälkikäteen. [3.]

Organisaatioihin kohdistuu jatkuvasti enemmän tietoturvauhkia. Euroopan unionin tietoturvaviraston julkaiseman 4.2020–7.2021 ajanjakson uhkakuvan mukaan kyberhyökkäysten määrä kasvaa vuosi vuodelta, vain uhkien tyyppi vaihtelee. Ajanjaksolla hyökkääjinä toimivat pääasiassa eri valtiot, ammattikyberrikolliset, maksua vastaan hakkeroivat ja aktivistihakkerit. Suurimpia uhkia ovat kiristyshaittaohjelmat, haittaohjelmat, luvaton kryptomainaus, sähköpostiin liittyvät uhat sekä organisaatioiden tietoon kohdistuvat uhat. [4.]

Kiristyshaittaohjelmat salaavat organisaation tietoja ja vaativat lunnaita tietojen palauttamista vastaan. Kiristyshaittaohjelmien yleisimmät leviämiskeinot ovat verkkoon avoimien Windows-etätyöpöytäyhteyksien sekä tunnusten kalastelun kautta. Haittaohjelma on yleinen termi ohjelmille, jotka toimivat käyttäjää kohtaan haitallisella tavalla. Haittaohjelmat voivat esimerkiksi

avata etäyhteyden uhrin päätelaitteelle, varastaa luottokorttietoja tai liittää päätelaitteen osaksi hyökkääjän bottiverkkoa, jonka kautta hyökkääjä voi komentaa jokaista bottiverkkoon liittynyttä laitetta etänä. Luvaton kryptomainaus tapahtuu yleensä haittaohjelman kautta, joka asentaa päätelaitteelle kryptovaluuttaa louhivan ohjelman. Kryptomainaus käyttää paljon resursseja päätelaitteella, kasvattaa sähkönkulutusta ja mahdollisesti vahingoittaa laitetta. Organisaation pilviympäristöön päästessään hyökkääjä voi käynnistää satoja virtuaalikoneita, aiheuttaa suuren kustannuksen organisaatiolle. Sähköpostiin liittyviin uhkiin kuuluu tunnusten kalastelu. Kalasteltuja tunnuksia voidaan käyttää sähköpostitilin kaappaamiseen, jolloin hyökkääjä voi jatkaa tunnusten kalastelua organisaatiossa tai levittää haittaohjelmia. Organisaation tietoon liittyviin uhkiin kuuluu usein kiristyshaittaohjelmien yhteydessä tapahtuva tiedon varastaminen ja julkaisu, ellei uhri suostu maksamaan lunnaita. [4.]

Pienille organisaatioille sisäisen CSOC:n perustaminen voi olla liian suuri kustannuserä. MSSP CSOC tarjoaa kustannustehokkaan tavan hankkia CSOC palveluna, jolloin organisaation ei tarvitse palkata omaa CSOC-tiimiä ja ostaa tarvittavia järjestelmiä [5]. Kansainvälisen tietoturva-ammattilaisten yhdistyksen, ISSA:n, mukaan organisaatioilla voi olla haasteita löytää tietoturvaosaajia vuosi vuodelta pahentuvan osaamispulan vuoksi [6].

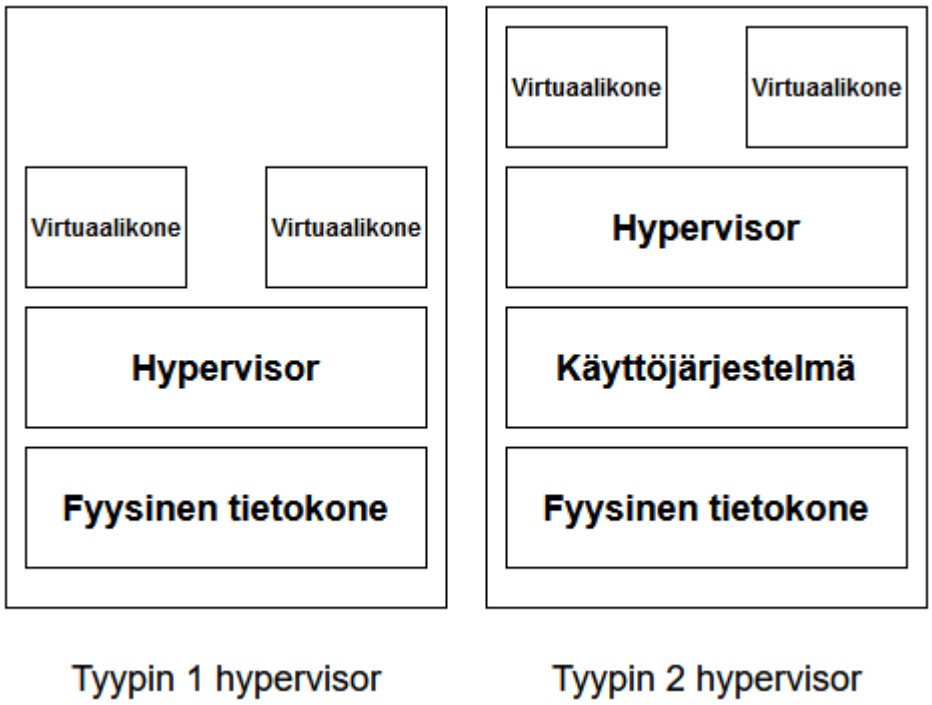
Trustwaven vuonna 2018 tekemän tutkimuksen mukaan organisaatioiden kokema paine tietoturvan suhteen kasvaa vuosi vuodelta. Vuonna 2017 54 % vastaajista koki enemmän tietoturvahaasteita kuin vuonna 2016. Isoimmat haasteet organisaatioiden mukaan olivat kohdistetut hyökkäykset, tietoturvabudjetin vähyys sekä tietoturvaosaamisen puutteet. Ulkoistamalla tietoturvan MSSP:lle organisaatiot voivat saada apua mainittuihin haasteisiin. [7.]

3.2 Virtualisointi

Virtualisoinnissa yhdellä tietokoneella ajetaan yhtä aikaa useita pienempiä virtuaalisia tietokoneita. Virtualisoinnin avulla voidaan jakaa palvelimen kapasiteetti useaan pienempään virtuaalikoneeseen, jolloin palvelinten käyttöastetta saadaan nostettua. Virtuaalikoneet saavat mm. virtuaaliset prosessorit, keskusmuistit ja kovalevyt. Jokainen virtuaalikone on eristetty toisistaan, mikä sallii eri käyttöjärjestelmien ajamisen virtuaalikoneilla. [8.]

3.2.1 Hypervisor

Hypervisor on ohjelmisto, jonka avulla voidaan luoda ja ajaa virtuaalikoneita. On olemassa kahdenlaisia hypervisoreita, tyypin 1 ja tyypin 2 (ks. Kuva 1).



Kuva 1. Tyypin 1 ja 2 hypervisorit

Tyypin 1 hypervisor on oma käyttöjärjestelmänsä ja sillä on suora pääsy isäntälaitteen resursseihin. Tyypin 1 hypervisor on turvallisin, koska isäntälaitteella ei ole muita käyttöjärjestelmiä, jotka voisivat laajentaa hyökkäyspinta-alaa. Esimerkki tyypin 1 hypervisorista on VMware ESXi. Tyypin 2 hypervisor asennetaan käyttöjärjestelmän, kuten Windowsin tai Linuxin, päälle. Microsoftin Hyper-V on tyypin 1 hypervisor, vaikka Hyper-V asennetaan Windowsiin. Asennuksen aikana Hyper-V virtualisoi asennettuna olleen Windowsin, jonka avulla Hyper-V saa suoran pääsyn laitteen resursseihin [9]. Tyypin 2 hypervisoreilla virtualisointi ei ole niin tehokasta, koska resurssien käyttö vaatii käyttöjärjestelmätason läpäisemisen, toisin kuin tyypin 1 hypervisorilla. [10.]

Hypervisorin avulla virtualisoidut virtuaalikoneet ovat eristettyjä toisistaan sekä hypervisorista. Virtuaalikoneet eivät tiedä toistensa läsnäolosta, eikä virtuaalikoneet tiedä pyörivänsä virtualisoi-

dussa ympäristössä. Virtuaalikoneet käyttävät yleensä prosessoria prosessorien tarjoaman virtualisointilisäosan kautta, mutta eivät voi ajaa korkeimman tason, ring 0, komentoja. Jokainen virtuaalikone saa oman virtuaalisen keskusmuistin, eikä muistia jaeta muiden virtuaalikoneiden kanssa. Virtuaalikoneet eivät käytä hyväksi hypervisorin kerneliä tai mitään muuta käyttöjärjestelmätason resurssia. [11.]

3.2.2 Konttiteknologia

Konttiteknologiaa voi ajatella kevyempänä virtualisointina. Toisin kuin virtualisoinnissa, kontteja ajetaan käyttöjärjestelmän päällä ja kontit jakavat isäntäkäyttöjärjestelmän osia, kuten kerneliä (ks. Kuva 2). Käyttämällä hyväksi isäntäkäyttöjärjestelmää, konteissa ei tarvitse ajaa erillistä käyttöjärjestelmää, mikä tekee konttien käytöstä huomattavasti kevyempää verrattuna virtualisointiin. [12.]

Kuva 2. Konttiteknologia

Konttien toimintatapa tekee niistä hyvän vaihtoehdon perinteisille virtuaalikoneille. Kontit voidaan rakentaa muuttumattomiksi, jolloin jokainen samasta konttikuvasta tehty kontti on samanlainen. Kontit vaativat huomattavasti vähemmän levytilaa verrattuna virtuaalikoneisiin, koska

kontit eivät tarvitse omaa käyttöjärjestelmää. Vikaantunut kontti voidaan poistaa käytöstä ja korvaava kontti saadaan asennettua sekunneissa, kun virtuaalikoneilla asennusaika lasketaan minuuteissa. [13.]

IBM teki vuonna 2020 kyselyn, jossa selvitettiin organisaatioiden suunnitelmia konttiteknologian hyödyntämisessä. Vastaajista 64 % suunnitteli vähintään joka toisen nykyisen sovelluksen muuttamista käyttämään konttiteknologiaa. Vastaajista 61 % käytti jo vähintään joka toisessa uudessa sovelluksessa konttiteknologiaa. Konttiteknologian käyttäjistä 61 % käyttää jotain konttien orkestrointityökalua ja 30 % aikoo siirtyä käyttämään lähiaikoina. [14.]

Konttiteknologia tuo mukanaan uusia haasteita tietoturvan kannalta. Kyberrikolliset ovat alkaneet kohdistaa hyökkäyksiään konttiteknologialla toteutettuihin ympäristöihin. Hyökkäykset tapahtuvat haitallisten konttilevykuvien sekä julkiverkossa olevien konttiorkestrointityökalujen kautta. Tavoitteena on saada ympäristö hyökkääjän haltuun. Ympäristöä käytetään yleensä hyväksi joko luvattomaan kryptomainaukseen tai uhrin tietojen keräämiseen. [4.]

3.2.3 Konttiorkestrointi

Konttien elinikä on usein lyhyt ja kontteja käynnistetään ja sammutetaan jatkuvasti, minkä vuoksi konttien hallintaan on kehitetty omia työkaluja. Yksi käytetyimmistä orkestrointityökaluista on Kubernetes. [15.]

Kubernetes perustuu Googlen kehittämään Borg-työkaluun. Google julkaisi Kubernetesin avoimen lähdekoodin projektina vuonna 2014. Kubernetes automatisoi useita konttityökuormiin liittyviä ylläpitotoimia. Kontit voidaan automaattisesti poistaa käytöstä, jos asetetut testit eivät mene läpi. Kontit voidaan asettaa sopivalle palvelimelle, riippuen konteille allokoidun muistin ja prosessoriytimien mukaan. Kubernetes voi ylläpitää tarvittavia arkaluonteisia tietoja, kuten salasanoja ja SSH-avaimia. [15.]

3.3 Falco / Linux

Falco on pilvinatiivi tietoturvatyökalu, jonka on alun perin kehittänyt yhdysvaltainen Sysdig Inc. Sysdig lahjoitti projektin Cloud Native Computing Foundation:lle tammikuussa 2020. Falco on ilmainen työkalu ja sen lähdekoodi on julkinen. Falco voidaan asentaa Linux-käyttöjärjestelmälle tai konttina konttiteknologialla toteutettuun ympäristöön. [16.]

Falcon toiminta perustuu Linuxin järjestelmäkutsujen ja Kubernetes-audit-lokien seurantaan. Kaikki käyttöjärjestelmässä ajettavat ohjelmat käyttävät hyväkseen järjestelmäkutsuja. Järjestelmäkutsut voidaan jakaa kuuteen eri ryhmään: prosessien hallintaan, tiedostojen hallintaan, laitteiden hallintaan, tietojen ylläpitoon, kommunikaatioon ja suojaukseen. Prosessien hallintaan kuuluvat esimerkiksi prosessien luominen, prosessien sammuttaminen ja muistinhallinta. Tiedostojen hallinta sisältää esimerkiksi tiedostojen poistamisen, luomisen, avaamisen ja sulkemisen. Laitteiden hallintaan kuuluu esimerkiksi laitteiden liittäminen ja poistaminen. Tietojen hallinta sisältää esimerkiksi prosessien, tiedostojen ja laitteiden attribuuttien hallinnoinnin. Kommunikointiin kuuluvia järjestelmäkutsuja on esimerkiksi etäyhteyksien poisto ja luonti. [17.]

Falco lukee järjestelmäkutsuja oletuksena käyttäen kernel-moduulia. Falco ei oletuksena seuraa kaikkia kernel-moduulin lukemia järjestelmäkutsuja, koska tiettyjen järjestelmäkutsujen seuraaminen nostaa Falcon kuormaa huomattavasti. Falco on mahdollista konfiguroida seuraamaan kaikkia järjestelmäkutsuja. [18.]

Kubernetes-audit-lokituksen kautta Falco voi monitoroida muutoksia Kubernetes-klusterissa. Audit-loki haetaan Kubernetesin API:n kautta. API:n kautta voidaan monitoroida esimerkiksi muutoksia klusterin konfiguraatioihin, salasanoihin, palveluihin ja oikeuksiin. [19.]

Falco vertaa analysoimiaan järjestelmäkutsuja ja Kubernetes-audit-lokeja säännöstöön. Sisäänrakennettu säännöstö sisältää sääntöjä mm. konttien muuttumattomuuden seurantaan sekä haavoittuvuuksien hyväksikäytön huomaamiseen. Uusia sääntöjä on mahdollista tehdä itse, sekä sisäänrakennettuja sääntöjä voidaan muokata. [20.]

Säännöstö koostuu viidestä pakollisesta osiosta. Esimerkki Falcon säännöstä on esitetty kuvassa 3. Rule-osiossa säännölle annetaan uniikki nimi. Desc-osiossa säännölle annetaan kuvaus. Condi-

tion-osiossa määritellään säännön logiikka. Esimerkkisäännössä järjestelmäkutsun tyyppi on execve, prosessin ajaminen. Kontin ID ei saa olla host, eli isäntäkone. Prosessinimeksi on määritetty Linuxin oletuskomentotulkki, bash. Sääntö aiheuttaa hälytyksen, jos bash-komentotulkki ajetaan kontin sisällä. Output-osiossa määritellään hälytyksen sisältö. Sisällössä voi käyttää muuttujia, kuten %container.name, joka tulostaa aina hälytykseen liittyvän kontin nimen. Priority-osiossa säännölle määritetään vakavuus. [20.]

```
- rule: shell_in_container
  desc: notice shell activity within a container
  condition: evt.type = execve and evt.dir=< and container.id != host and proc.name = bash
  output: shell in a container (user=%user.name container_id=%container.id
          container_name=%container.name shell=%proc.name parent=%proc.pname cmdline=%proc.cmdline)
  priority: WARNING
  tags: [container]
```

Kuva 3. Falcon esimerkkisääntö shell_in_container

Säännöissä voidaan käyttää listoja ja makroja, ettei samoja asioita tarvitse kirjoittaa useita kertoja eri sääntöihin. Lista voi olla esimerkiksi lista IP-osoitteista, joista yhteydet ovat sallittuja. Makrolla voidaan korvata osa condition-osiota. Makrojen ja listojen käyttö helpottaa säännöstön ylläpitoa, koska muutokset tulevat kerralla voimaan kaikkiin makroja ja listoja käyttäviin sääntöihin. [20.]

4 Toteutus

Projekti toteutettiin Loihde Trustin testiympäristössä, joka perustuu Hyper-V-virtualisointiin. Testiympäristö on eristetty tuotannosta, mutta sisältää vastaavat työkalut kuin tuotantoympäristö.

Falcon yksi tärkeimpiä ominaisuuksista on tuki konttiteknologialle. Konttien testaamista varten testipalvelimelle asennettiin Docker. Testikontteina käytetään Ubuntun virallista konttia, jota muokataan testitarkoitusta varten sopivaksi.

Testipalvelimella testattiin Falcon sisäänrakennettua säännöstöä pääasiassa, kuinka paljon vääriä hälytyksiä (eng. Falsepositive) säännöstö aiheuttaa. Sääntöjen tekemistä testattiin kehittämällä säännöstöä epäilyttävän toiminnan ja haavoittuvuuksien hyväksikäytön huomaamiseksi.

Sääntöjen aiheuttamat hälytykset ohjattiin Logstash-lokikeräimelle. Logstashin ideana on keskittää useiden Falco-asennusten hälytykset yhdelle palvelimelle, josta hälytykset voidaan ohjata SIEM-järjestelmään. Logstash sallii tarvittaessa myös hälytysten ohjaamisen muihin järjestelmiin, esimerkiksi Teams-kanaville tai Microsoftin pilvessä toimivaan SIEM-järjestelmään, Azure Sentineliin.

SIEM-järjestelmään tehtiin Falcon-hälytyslokille lokiparseri, jotta SIEM osaa rikastaa hälytyksen tiedot. Tietojen rikastusta tarvitaan mm. sääntöjen poikkeuksien tekemiseen SIEM-järjestelmän päässä. Joissain tilanteissa voidaan haluta tieto hälytyksestä SIEM-järjestelmään, mutta siitä ei haluta hälytystä CSOC-analyytikoille. Yksi esimerkki tällaisesta tilanteesta on kirjautuminen kontteihin ympäristössä, jossa kontteihin kirjautuminen on normaalia. Tieto kirjautumisista tulee silti SIEM-järjestelmään, jolloin tarvittaessa voidaan selvittää, mihin kontteihin on kirjauduttu ja milloin.

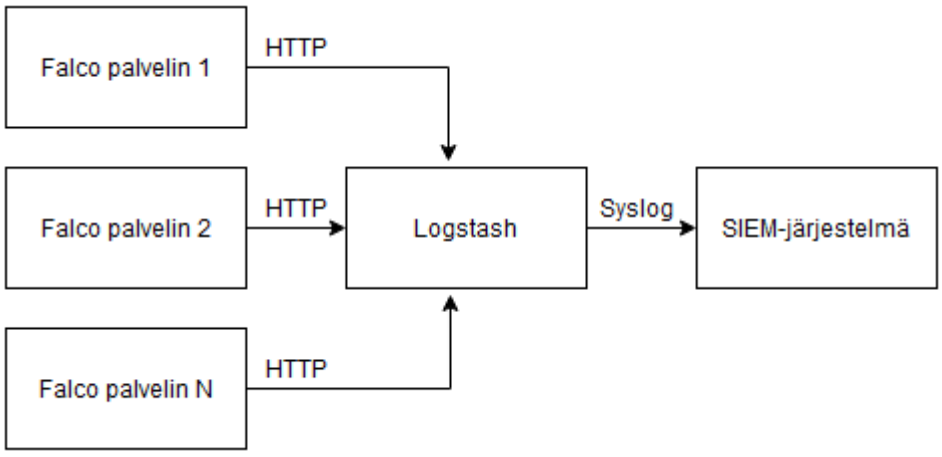
4.1 Testiympäristö

Projektissa käytettiin Hyper-V-virtualisointia käyttävää palvelinta, johon projektissa tarvittavat virtuaalipalvelimet asennettiin. Falcoa varten ympäristöön asennettiin kaksi virtuaalipalvelinta.

Falcoa on tarkoitus käyttää yrityskäytössä olevilla palvelimilla, joten käyttöjärjestelmiksi valittiin yrityskäytössä suosittu Linux-distribuutio Ubuntu 20.04. Ubuntu on Canonicalin ylläpitämä distribuutio, joka perustuu Debianiin. Canonical julkaisee kahden vuoden välein LTS (Long Term Support) -version distribuutiosta. Canonical tukee LTS-versiota 5 vuotta, mikä tekee siitä suositun yrityskäytössä. [21.]

Falcon luomat hälytykset lähetetään Logstash-palvelimelle. Logstash on Elasticin kehittämä työkalu, jolla voidaan kerätä lokia useista eri lähteistä, muokata eri muotoon ja lähettää eteenpäin eri kohteisiin. Logstash asennetaan CentOS 8 -distribuutiolle. CentOS perustuu maksulliseen Red Hat -distribuutioon. Red Hat on yksi suosituimmista distribuutioista yrityskäytössä, koska Red Hat kehittää ja tukee jokaista versiota jopa kymmenen vuoden ajan. CentOS on suosittu vaihtoehto Red Hatille, koska sen käyttö ei maksa mitään. CentOS on yhteensopiva Red Hatin kanssa, minkä vuoksi se valittiin projektiin. [22.]

Testiympäristössä on projektia varten testiasennus SIEM-järjestelmästä. Järjestelmä koostuu supervisor- ja keräinpalvelimista. SIEM-keräin kerää kaiken lokin ja käsittelee sen SIEM-supervisorin ymmärtämään muotoon. SIEM-supervisor korreloi lokia ja luo hälytyksiä sääntöjen mukaan. Logstashiltä loki lähetetään SIEM-keräimelle, josta se siirtyy SIEM-supervisorille tarkempaa analyysia varten. Liitosprosessi on esitetty kuvassa 4. Testiympäristö on arkkitehtuurisesti rakennettu vastaamaan tuotantoympäristöä. Jokainen konttiympäristön palvelin vaatii Falcon-asennuksen. Testiympäristö koostuu kahdesta palvelimesta, joille asennetaan Falco, mutta sama arkkitehtuuri toimii myös tuotantoympäristöille, joissa Falco asennuksia voi olla kymmeniä.



Kuva 4. Falcon liitosprosessi

4.2 Liitos

Falco asennettiin Loihde Trustin testiympäristöön Ubuntu 20.04 -distribuutiolle. Palvelimille annettiin 4 virtuaalista prosessoriydintä, 6 GB keskusmuistia sekä 50GB kovalevytilaa.

Palvelimilla on tarkoitus käyttää hyväksi kontteja, joten palvelimille asennettiin Docker. Asennuksessa käytettiin Dockerin virallista asennusskriptiä.

Falco voidaan asentaa konttina Kubernetesin tai isäntäkäyttöjärjestelmän päälle. Testiasennuksessa ei tulla käyttämään Kubernetesiä, joten Falco asennettiin isäntäkäyttöjärjestelmään. Asennus tehtiin lisäämällä Falcon pakettirepository ja asentamalla paketti nimeltä ”falco”.

Tärkeimpänä vaatimuksena projektissa on saada Falcon luomat hälytykset siirrettyä SIEM-järjestelmään. Falco tukee useita eri tapoja lähettää hälytysten tieto eteenpäin, kuten syslog, tiedostoon kirjoittaminen ja http. Projektissa päädyttiin lähettämään tieto http:n avulla, koska tarvittaessa tieto voidaan salata helposti käyttäen HTTPS-protokollaa.

Falcon falco.yaml -tiedostoon muokattiin kuvassa 5 esitetyt muutokset.

```
json_output: true
http_output:
  enabled: yes
  url: http://devlogstash:8080
```

Kuva 5. falco.yaml

Json_output -asetus muokkaa tiedon JSON-muotoon. http_output -osiossa määritellään Falco lähettämään hälytysten tieto HTTP-protokollalla osoitteeseen http://devlogstash:8080, jossa sijaitsee Logstash.

4.2.1 Logstash

Logstash asennettiin CentOS 8 -distribuutiolle. Palvelimelle annettiin 2 virtuaalista prosessoriydintä, 6GB keskusmuistia sekä 50GB kovalevytilaa. Logstash asennettiin käyttäen Logstashin virallista pakettirepositoryä.

Falcon lähettämät hälytykset kerätään Logstashiin. Logstash määriteltiin kuuntelemaan HTTP-liikennettä portissa 8080. Projektia varten tehtiin yksinkertainen konfiguraatio Logstashille, jossa kaikki vastaanotetut lokiviestit lähetetään eteenpäin syslog-protokollalla SIEM-järjestelmään. Logstash kirjoittaa viestit myös /opt/debug.log -tiedostoon, jotta testivaiheessa ongelmienselvitys on helpompaa. Tuotantokäytössä syslog-protokollan liikenne voidaan salata TLS-salauksella, mutta testivaiheessa liikenne lähetetään salaamattomana. Logstashin konfiguraatiomuutokset on esitetty kuvassa 6.

```
input {
  http {
    port => 8080
  }
}

output {
    syslog {
      appname => "Falco"
      host => "siemcollector"
      port => "514"
      protocol => "tcp"
      rfc => "rfc3164"
    },
    file {
      path => "/opt/debug.log"
      codec => line { format => "Message: %{message}"}
    }
}
```

Kuva 6. logstash.conf

4.2.2 SIEM

Projektissa käytetään hyväksi Loihde Trustin testiympäristössä olevaa SIEM-järjestelmää. Asennus koostuu SIEM-supervisorista ja SIEM-keräimestä.

Keräimelle lähetetystä lokista kerätään tietoja, kuten käyttäjä, kontin nimi, hälytyksen nimi ja lähdeprosessi. Tiedot merkataan attribuutteihin, jolloin SIEM-järjestelmän kautta voidaan hakea lokitapahtumia esimerkiksi käyttäjän tai lähdeprosessin mukaan. Attribuuttien avulla voidaan myös

tehdä poikkeuksia SIEM-hälytyksiin, jolloin säännöstöä ei välttämättä tarvitse muokata Falcossa, vaan tietyt muokkaukset sääntöihin voidaan tehdä myös SIEM-järjestelmän avulla.

5 Testaus

Falcon säännöstön toimivuutta päätettiin testata tekemällä testisääntöjä erilaisiin haavoittuvuuksiin ja epäilyttäviin toimiin, joita Falcon avulla voitaisiin havaita. Säännöstöt perustuvat oikeisiin haavoittuvuuksiin, jotka ovat jo korjattu. Testaamisen kannalta haavoittuvuuksien iällä ja korjauksilla ei ole väliä, koska tarkoituksena on testata Falcon säännöstön ominaisuuksia.

5.1 Redis RCE

Redis on avoimen lähdekoodin tietorakennesäilö, jota käytetään useissa sovelluksissa tietokantana ja välimuistina. Redis säilyttää tiedon pääasiassa palvelimen keskusmuistissa, mutta se voidaan myös konfiguroida kirjoittamaan tiedot levylle. [23.]

Redis-suunnittelussa ei aluksi otettu huomioon tietoturvaa, koska pääasiassa Redistä käytetään suojatuissa verkoissa. Useat käyttäjät kumminkin avasivat Redis-asennuksensa julkiverkkoon, joko vahingossa tai tarkoituksella. Vuonna 2015 Redisin kehittäjä Salvatore Sanfilippo teki blogipostauksen, jossa ensimmäistä kertaa julkisesti kerrottiin, kuinka Redisillä voidaan ajaa koodia etänä. Redis ei käytä oletuksena salasanaa, joten oletusasennuksella kuka tahansa pystyi ajamaan omaa koodiansa. [24.]

Trend Micron tutkijat havaitsivat vielä vuonna 2020 julkiverkossa olevan yli 8000 suojaamatonta Redis-asennusta [25]. Tutkijoiden honeypot-palvelimella havaittiin erilaisia hyväksikäyttötapoja. Yksi hyväksikäyttötapa on käyttää hyväksi Redisin tietojen tallentamista levylle. Käyttäjä voi itse määritellä, mihin tiedot tallennetaan. Hyökkääjät käyttivät ominaisuutta kahdella tavalla. Ensimmäisessä Redisin muistiin kirjoitettiin hyökkääjän SSH-avain, joka kirjoitettiin levylle redis-käyttäjän authorized_keys-tiedostoon, jonka avulla hyökkääjät pystyivät kirjautumaan palvelimelle SSH:n avulla [24]. Toinen tapa on kirjoittaa Redisin muistiin ajastettuja komentoja, jotka tallennettiin levylle /var/spool/cron/root- ja /var/spool/cron/crontabs/root -tiedostoihin. Käyttöjärjestelmä ajaa ajastetut komennot, joiden avulla hyökkääjä voi saada etähallinnan palvelimelle. [26.]

Haavoittuvuutta vastaan tehtiin Falcoon sääntö (ks. Kuva 7), joka havaitsee kirjoittamiset .ssh- ja /var/spool/-kansioihin redis-prosessin toimesta.

```
- list: protected_processes
  items: [redis-server]

- macro: forbidden_read
  condition: proc.name in (protected_processes) and
             (fd.name contains ".ssh" or fd.name startswith "/var/spool")

- rule: Redis potentially dangerous file write
  desc: Redis has written to a folder that can be used for RCE.
  condition: (open_write or open_read) and forbidden_read
  output: Redis opened potentially dangerous file (user=%user.name
          user_loginuid=%user.loginuid command=%proc.cmdline file=%fd.name)
  priority: CRITICAL
  tags: [file, mitre_persistence]
```

Kuva 7. Redis-sääntö

Säännössä käytettiin hyväksi Falcon listoja ja makroja. Forbidden_read -makro sisältää prosessin nimen tarkistuksen sekä kohdekansion .ssh tai /var/spool. Makro viittaa listaan protected_processes, joka sisältää redis-prosessin nimen redis-server. Säännön ulostulo sisältää muuttujina käyttäjän, käyttäjän ID:n, ajetun komennon, tiedostonimen sekä mahdollisen kontin tiedot.

Sääntöä testattiin luomalla kontti, johon asennettiin Redis. Redistä ajettiin epäturvallisessa tilassa ja etänä koitettiin kirjoittaa .ssh- ja /var/spool -kansioihin. Sääntö aiheutti hälytyksen ja aikaisemmin tehty SIEM-liitos toimi suunnitellusti.

5.2 Tunnettuihin haittaohjelmien osoitteisiin liikennöinti

Haittaohjelmat tekevät yleensä yhteyksiä ns. C2 (Command and Control) -palvelimille. C2-palvelinten kautta haittaohjelmat tietävät, mitä pitäisi tehdä, esimerkiksi lähettää tietyt tiedostot hyökkääjän palvelimelle, käynnistää ransomware-hyökkäyksen tai sammuttaa tietokoneen. Jotkut haittaohjelmat käyttävät myös julkisia palveluita, esimerkiksi Pastebiniä, C2-palvelimena. [27.]

C2-liikenteen havainnoinniksi tehtiin testisääntö (ks. Kuva 8), joka havaitsee yhteydet määritetyille IP-osoitteille ja domaineille.

```
- list: malicious_domains
  items: ["pastebin.com",
          "0bin.net",
          "zerobin.net"]

- list: malicious_ips
  items: ['"1.0.0.1"',
          '"8.8.4.4"']

- rule: Connection to known malicious host
  desc: Process has communicated with a known malicious host
  condition: evt.type = connect and fd.l4proto = tcp and ((fd.sip.name in
             (malicious_domains)) or (fd.sip in (malicious_ips)))
  output: Connection to known malicious host (command=%proc.cmdline conn=%fd.name
          user=%user.name user_loginuid=%user.loginuid
          container_id=%container.id image=%container.image.repository)
  priority: WARNING
  tags: [network]
```

Kuva 8. C2-sääntö

Sääntö koostuu kahdesta listasta, malicious_domains ja malicious_ips. Testiä varten malicious_domains sisältää kolme pastebinin kaltaista palvelua sekä malicious_ips sisältää kaksi julkista DNS-palvelinta. Tuotantokäytössä sääntöön muokattaisiin tarpeen mukaan oikeita haitallisia domaineja ja IP-osoitteita. Pastebin ja vastaavat sivustot voisivat kuulua tuotannossakin sääntöön, koska niihin on harvoin liikennettä palvelinverkoista ja palveluita käytetään usein C2-palvelimina. Säännön tulostulo kerää muuttujina tiedot ajetusta komennosta, yhteyden kohteesta, käyttäjästä sekä mahdollisesta kontista.

Sääntöä testattiin koittamalla yhdistää kontin sisältä osoitteisiin pastebin.com ja 1.0.0.1. Sääntö aiheutti hälytyksen ja integraatio SIEM-järjestelmään toimi suunnitellusti.

5.3 Sudo CVE-2019-14287

Sudo on Linuxin työkalu, jonka avulla käyttäjille voidaan antaa oikeus ajaa komentoja root-tason oikeuksilla. CVE-2019-14287 on haavoittuvuus, joka salli tietyissä tilanteissa käyttäjien ajaa root-tason komentoja, vaikka heiltä olisi sudon konfiguraatiossa estetty root-tason komennot. Haavoittuvuus toimii käyttämällä -u-vipua. Sudon -u-vivulla voidaan asettaa käyttäjä, jolla komento ajetaan. Vivussa voidaan käyttää myös käyttäjän UID, joka rootilla on 0. Käyttäessä ”-u#-1”- tai ”-u#4294967295” -vipua, sudo koittaa vaihtaa käyttäjän ID:n -1, mutta muutosta ei tapahdu, koska

sudoa ajetaan jo root-oikeuksilla. Logiikkavirheen vuoksi sudo antaa ajaa komennon root-oikeuksilla, vaikka konfiguraatio ei sitä normaalisti sallisi. [28.]

Haavoittuvuuden havainnointiin tehtiin sääntö (ks. Kuva 9), joka hälyttää sudo-komennon sisältäessä ”-u#-1”- tai ”-u#4294967295”. Normaalissa käytössä -u-vipua ei käytetä tällaisella tavalla, joten säännön ei pitäisi aiheuttaa vääriä hälytyksiä.

```
- rule: sudo_cve-2019-14287
  desc: Potential sudo CVE-2019-14287 abuse detected
  condition: evt.type = execve and evt.dir =< and proc.name = sudo and
             (proc.args contains "-u#4294967295" or proc.args contains "-u#-1")
  output: Sudo CVE-2019-14287 attempt (user=%user.name container_id=%container.id
          container_name=%container.name shell=%proc.name parent=%proc.pname cmdline=%proc.cmdline)
  priority: WARNING
```

Kuva 9. Sudo CVE-2019-14287 sääntö

Sääntöä testattiin ajamalla komento ”sudo -u#-1”. Sääntö aiheutti hälytyksen ja liitos SIEM-järjestelmään toimi odotetusti.

5.4 Oletussäännöstö

Falcon oletussäännöstö sisältää useita sääntöjä palvelimen ja konttien epäilyttävän toiminnan havaitsemiseen. Säännöstö voidaan jakaa kolmeen osioon, audit-tyyppisiin sääntöihin, huonojen käytäntöjen havainnointiin ja haitallisen toiminnan havaitsemiseen.

Audit-tyyppiset säännöt eivät suoraan havainnoi haitallista toimintaa, mutta voivat olla muiden hälytysten tutkinnan apuna. Säännöt voivat aiheuttaa hälytyksiä normaaleista ylläpitotoimista, kuten konfiguraatiotiedostojen muokkaamisesta /etc/-kansiossa tai korkeilla oikeuksilla käynnistetystä kontista. Osaan säännöistä voidaan tehdä poikkeuksia, joiden avulla vääriä hälytyksiä voidaan karsia vähemmäksi.

Huonoja käytäntöjä havainnoivat säännöt aiheuttavat hälytyksiä asioista, joita ei hyvin suunnitellussa ympäristössä ei pitäisi tapahtua, mutta eivät välttämättä ole merkki haitallisesta toiminnasta. Konttien tulisi olla muuttumattomia, eikä niihin pitäisi mennä SSH:n tai konttimoottorin

komentokehotteen kautta. Falco sisältää esimerkiksi säännön komentokehotteen käynnistymisestä ja uudesta ajettavasta tiedostosta kontin sisällä, joka hyvin suunnitellussa ympäristössä on epänormaalia. Sääntöjen hyödyllisyys laskee, jos huonoja käytäntöjä ei korjata.

Haitallisen toiminnan havaitsevat säännöt aiheuttavat hälytyksen vain tilanteissa, joissa jotain oikeasti haitallista on tapahtunut. Säännöstöön kuuluu useita haavoittuvuuksien hyväksikäyttöä havainnoivia sääntöjä, esimerkiksi sudossa olleen oikeuksien korotuksen haavoittuvuuteen (CVE-2021-3156) löytyy oma sääntönsä. Falcossa on myös sääntöjä kryptomainauksen havainnointiin kryptomainaukseen käytettyjen osoitteiden ja binäärien nimien perusteella.

Testiympäristössä oletussäännöstö aiheutti vääriä hälytyksiä pääasiassa isäntäkoneen /etc/-kansiossa sijaitsevien konfiguraatiotiedostojen muokkauksesta. Tuotantoympäristöissä konfiguraatiot ovat yleensä varsin staattisia, joten väärien hälytysten määrä kyseisen hälytyksen suhteen pitäisi olla vähäistä. Muiden sääntöjen osalta voi tulla yllätyksiä tuotantoympäristöissä, koska testiympäristö ei vastaa oikeaa tuotantoympäristöä.

5.5 Lopputulokset

Falcon käyttöönotto varsinkin konttiteknologiaa hyödyntävässä ympäristössä tuo huomattavaa hyötyä tietoturvavalvontaan. Falcolla ei silti voida korvata muita CSOC:n käytössä olevia työkaluja. Verratessa Gartnerin SOC-kolmioon, SIEM:ssä kaikki kerättävät lokitapahtumat tallennetaan SIEM-järjestelmään, jolloin tapahtumia voidaan tutkia tarvittaessa jälkikäteen. Falcossa tapahtumia ei tallenneta, eli Falcossa täytyy olla tietoturvapoikkeaman tapahtumahetkellä oikea säännöstö käytössä, jotta se voidaan havaita CSOC:n toimesta. Konttiympäristössä tarvittavien lokien kerääminen voi olla haasteellista, varsinkin jos ympäristöä rakentaessa ei ole otettu lokitusta huomioon, jolloin Falcoa voitaisiin käyttää paikkamaan näkyvyyttä kontteihin. Falco perustuu järjestelmäkutsujen seurantaan, minkä vuoksi sillä voidaan tehdä huomattavasti tarkempia sääntöjä verrattuna SIEM-järjestelmiin. Audit-lokitusta Falcolla ei voida korvata, mutta sitä voidaan tarvittaessa paikata tekemällä omia sääntöjä.

EDR havaitsee haitallisia tiedostoja sekä yleensä tallentaa päätelaitteen tapahtumia tutkintaa avustamaan. Falcon säännöstö perustuu epäilyttävien tapahtumien havainnointiin, jonka vuoksi

Falco ei tiedä, onko palvelimella ajettu jotain haitallista vai ei, eikä tallenna päätelaitteen tapahtumia. Falco voi tuoda lisäarvoa esimerkiksi haavoittuvuuksien hyväksikäytön havainnoinnissa. Hyvä esimerkki on komentokehotteen käynnistyminen kontissa. Riippuen käytetystä EDR-järjestelmästä, mariadb tai apache2 käynnistämä bash-komentokehto ei välttämättä aiheuta hälytystä, ellei komentokehotteen kautta ajeta haitallisiseksi havaittuja komentoja. Falco sisältää säännön komentokehotteen käynnistymisestä kontissa, jonka avulla CSOC saisi tietää, mikä prosessi on käynnistänyt komentokehotteen ja missä kontissa.

NDR havainnoi epäilyttäviä tapahtumia verkkoliikenteestä ja tallentaa tutkimusta auttamaan metadata-tietoa. Falcon säännöstöt ovat staattisia ja havaitsevat liikenteen jo valmiiksi haitalliseksi tunnettuihin osoitteisiin sekä yhteydet, joita ympäristössä ei pitäisi tapahtua, esimerkiksi liikennettä julkiverkosta tietokantapalvelimelle.

Falcon testauksessa havaittiin jatkotutkimusta vaativia ongelma, jotka täytyy selvittää ennen Falcon tuotantokäyttöönottoa. Falcon säännöstö on tallennettuna paikallisesti palvelimella, jossa Falcoa käytetään. Tuotantokäytössä säännöstöä täytyy päivittää CSOC:n toimesta, mutta manuaalisesti säännöstön päivitys ei ole kannattavaa. Manuaalinen päivitys vaatisi CSOC:lle root-tason oikeudet palvelimille, sekä päivitys olisi liian työlästä Falcon asennusmäärän kasvaessa. Ongelman voisi korjata muutamalla eri tavalla. Palvelimelle voitaisiin tehdä CSOC:lle käyttäjätunnus. Käyttäjätunnukselle sallittaisiin sudon konfiguraatiossa oikeus päivittää konfiguraatio ja käynnistää Falco uudestaan. SSH-yhteys voidaan tehdä esimerkiksi Logstash-palvelimen kautta ja päivitysprosessi voidaan automatisoida Ansiblella. Toinen tapa olisi muuten samanlainen kuin ensimmäinen, mutta konfiguraatiotiedosto lähetetään Ansiblella Logstash-palvelimelle ja Falco-palvelimet hakevat skriptin avulla päivitetyn konfiguraatiotiedoston. Konfiguraatiotiedoston version vaihtuessa skripti käynnistäisi Falcon uudestaan. Kolmannessa tavassa konfiguraatiotiedosto päivitetään yksityiseen Git-säilöön. Palvelimilla oleva skripti hakisi uusimman version Gitistä ja tarvittaessa päivittäisi konfiguraation.

Toinen jatkotutkimusta vaativa ongelma on Falcon yleinen ylläpito ja vianselvitys. CSOC:lla on harvoin suoraa pääsyä asiakkaiden palvelimille, minkä vuoksi Falcon mahdolliset vikatilanteet vaativat aina asiakkaan IT-osaston apua. CSOC:lle voi olla haasteellista tarkistaa Falcon toimivuus, koska Falco lähettää lokiviestejä vain hälytyksistä ja ympäristöstä riippuen hälytyksiä voi tulla hyvin vähän. Tuotantokäyttöön olisi suositeltavaa kehittää sääntö, joka ilmoittaa Falcon toimivan

säännöllisin väliajoin. SIEM:ssä voidaan silloin tehdä sääntö, joka hälyttää Falcon lokituksen pysähtyessä pidemmäksi aikaa.

Logstashin käyttö SIEM-keräimen ja Falco-palvelinten välillä ei ole pakollista, mutta tietyissä tilanteissa voi tuoda lisäarvoa. Logstash sallii SIEM-järjestelmän vaihtamisen toiseen ja mahdollistaa integraatiot muihin järjestelmiin, esimerkiksi tietyt hälytykset voidaan ohjata suoraan asiakkaan Teams-kanavalle. Logstash vaatii vähemmän resursseja kuin SIEM-keräin, minkä vuoksi se on hyvä vaihto pilviympäristöihin. Palvelimet voidaan konfiguroida lähettämään hälytysloki pilvessä ajettavalle Logstashille, joka ohjaa lokin haluttuun paikkaan.

Testeissä käytetty testiympäristö ei vastannut oikeaa tuotantoympäristöä, joten oletussäännöstön väärien hälytysten määrää on hankala arvioida. Falcoa käyttäessä on varauduttava sääntöjen vaativan tuunausta jokaisen asiakkaan tarpeiden ja ympäristöjen mukaan. Varsinkin ensimmäisessä käyttöönotossa tulee varautua suurempaan työkuormaan, koska säännöstöä ei ole vielä testattu oikeassa tuotantoympäristössä. Uusien sääntöjen luonti on pääasiassa yksinkertaista. Järjestelmäkutsuihin perustuvat säännöt sallivat myös todella tarkkojen sääntöjen luonnin, joka ei onnistu millään SOC-kolmion työkalulla.

6 Pohdinta

Ehdotin opinnäytetyön aihetta toimeksiantajalle, koska konttiteknologia on vuosi vuodelta tullut suositummaksi ja konttiteknologialla rakennettujen ympäristöjen tietoturvavalvonta vaatii erilaisia työkaluja ja ajattelutapaa kuin perinteisten ympäristöjen. Taustalla oli myös henkilökohtainen kiinnostus konttiteknologiaa kohtaan.

Opinnäytetyön kriteerinä oli selvittää, soveltuuko Falco MSSP CSOC:n käyttöön ja onko käyttöönotossa millaisia haasteita. Kysymykseen on hankala antaa kyllä- tai ei-vastausta. Projektin aikana selvisi, että Falcolla voidaan parantaa näkyvyyttä asiakkaiden ympäristöön. Falcon käyttöönoton hyöty riippuu paljon asiakkaalla jo käytössä olevista tietoturvatyökaluista. Ympäristössä, jossa on jo käytössä kaikki Gartnerin SOC-kolmion työkalut, hyöty voi jäädä vähäiseksi. Järjestelmäkutsuihin perustuva säännöstö sallii todella tarkkojen sääntöjen luonnin, verrattuna Gartnerin SOC-kolmion työkaluihin.

Projektin avulla saatiin selville Falcon sisältävän ainakin muutaman haasteen varsinkin MSSP CSOC:n käytössä. Falcon säännöstön päivittäminen tulee vaatimaan jatkotutkintaa ennen ensimmäistä käyttöönottoa, koska päivitystä ei voi tehdä etänä pelkästään Falcon avulla. Sääntöjen päivitykseen on muutamia vaihtoehtoja, mutta erillisten työkalujen tai skriptien avulla päivittäminen lisää mahdollisesti hajoavien asioiden määrää ympäristössä. Toinen jatkotutkinnan kohde on vikatilanteiden selvitys. Tuotannossa CSOC:lla todennäköisesti ei ole pääsyä Falcon palvelimelle, joten vikatilanteiden selvittäminen tulisi vaatimaan yhteistyötä asiakkaan kanssa.

Falco vaikuttaa sopivan parhaiten kriittisten palvelinten, ns. ”kruunun jalokivien”, valvontaa parantamaan, varsinkin jos käytössä on konttiteknologiaa. Laajempaa käyttöönottoa kannattaa miettiä siinä vaiheessa, kun esille tulleet haasteet on saatu selvitettyä.

Kokonaisuutena opinnäytetyön työstäminen on ollut todella opettavaista. Opin lisää konttiteknologiasta sekä konttien tietoturvavalvonnasta. Opinnäytetyön laajuus jäi alkuperäistä suunnitelmaa pienemmäksi. Kubernetes pudotettiin jo alkuvaiheessa pois ja projektissa esille tulleiden haasteiden tarkempi selvitys jäi tekemättä resurssien vähyyden vuoksi. Tarvittaessa nämä asiat voidaan tutkia tarkemmin opinnäytetyön ulkopuolella, jos toimeksiantaja katsoo jatkoselvitysten olevan tarpeellisia.

Lähteet

1 Loihde. Haettu 10.9.2021, sivustolta Loihde. Internetosoite: https://www.loihde.com/

2 Viria Oyj. 2021. Viria Oyj Tilinpäätöstiedote 2020. Haettu 10.9.2021, sivustolta Viria. Internetosoite: https://www.viria.fi/wp-content/uploads/2021/03/Viria-Oyj-Tilinpaatos-tiedote-2020.pdf

3 Anton Chuvakin. 2015. Your SOC Nuclear Triad. Haettu 31.5.2021, sivustolta Gartner. Internetosoite: https://blogs.gartner.com/anton-chuvakin/2015/08/04/your-soc-nuclear-triad/

4 ENISA. ENISA Threat Landspace 2021. Haettu 3.11.2021, sivustolta ENISA. Internetosoite: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2021

5 Gartner. Gartner Glossary. Haettu 31.5.2021, sivustolta Gartner. Internetosoite: https://www.gartner.com/en/information-technology/glossary/mssp-managed-security-service-provider

6 ISSA. 2021. The Life and Times of Cybersecurity Professionals 2021.

7 Trustwave. 2018. Security Pressures Report. Haettu 1.11.2021, sivustolta Trustwave. Internetosoite: https://www.trustwave.com/en-us/resources/library/documents/2018-security-pressures-report-spr/

8 VMware. 2006. Virtualization Overview Whitepaper. Haettu 11.10.2021, sivustolta VMware. Internetosoite: https://www.vmware.com/pdf/virtualization.pdf

9 Microsoft. Hyper-V Architecture. Haettu 3.11.2021, sivustolta Microsoft. Internetosoite: https://docs.microsoft.com/en-us/windows-server/administration/performance-tuning/role/hyper-v-server/architecture

10 VMware. What is a hypervisor?. Haettu 12.10.2021, sivustolta VMware. Internetosoite: https://www.vmware.com/topics/glossary/content/hypervisor

11 VMware. 2014. Security of the VMware vSphere Hypervisor. Haettu 12.10.2021, sivustolta VMware. Internetosoite: https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/whitepaper/techpaper/vmw-white-paper-secrty-vsphr-hyprvsr-uslet-101.pdf

12 Google. What are containers?. Haettu 11.10.2021, sivustolta Google Cloud. Internetosoite: https://cloud.google.com/learn/what-are-containers

13 Google. Best practices for operating containers. Haettu 31.5.2021, sivustolta Google Cloud. Internetosoite: https://cloud.google.com/architecture/best-practices-for-operating-containers

14 IBM. 2020. Containers in the enterprise. Haettu 1.11.2021, sivustolta IBM. Internetosoite: https://www.ibm.com/downloads/cas/VG8KRPRM

15 Kubernetes. What is Kubernetes. Haettu 1.11.2021, sivustolta Kubernetes. Internetosoite: https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/

16 Falco. The Falco Project. Haettu 31.5.2021, sivustolta Falco. Internetosoite: https://falco.org/docs/

17 Abraham Silberschatz. Operating System Concepts, 9th edition. 2013

18 Falco. Supported Syscall Events. Haettu 3.11.2021, sivustolta Falco. Internetosoite: https://falco.org/docs/rules/supported-events/

19 Falco. Kubernetes Audit Events. Haettu 2.11.2021, sivustolta Falco. Internetosoite: https://falco.org/docs/event-sources/kubernetes-audit/

20 Falco. Rules. Haettu 2.11.2021, sivustolta Falco. Internetosoite: https://falco.org/docs/rules/

21 Rhys Davies. 2020. What is an Ubuntu LTS release? Haettu 3.11.2021, sivustolta Ubuntu. Internetosoite: https://ubuntu.com/blog/what-is-an-ubuntu-lts-release

22 CentOS. CentOS Linux. Haettu 3.11.2021, sivustolta CentOS. Internetosoite: https://www.centos.org/about/

23 Redis. Introduction to Redis. Haettu 1.11.2021, sivustolta Redis. Internetosoite: https://redis.io/topics/introduction

24 Salvadore Sanfilippo. 2015. A few things about Redis security. Haettu 1.11.2021, sivustolta Antirez. Internetosoite: http://antirez.com/news/96

25 David Fiser. 2020. More Than 8,000 Unsecured Redis Instances Found in the Cloud. Haettu 1.11.2021, sivustolta TrendMicro. Internetosoite: https://www.trendmicro.com/en_us/research/20/d/more-than-8-000-unsecured-redis-instances-found-in-the-cloud.html

26 David Fiser, Jaromir Horejsi. 2020. Exposed Redis Instances Abused for Remote Code Execution, Cryptocurrency Mining. Haettu 1.11.2021, sivustolta TrendMicro. Internetosoite: https://www.trendmicro.com/en_fi/research/20/d/exposed-redis-instances-abused-for-remote-code-execution-cryptocurrency-mining.html

27 Varonis. What is C2? Command and Control Infrastructure Explained. Haettu 1.11.2021, sivustolta Varonis. Internetosoite: https://www.varonis.com/blog/what-is-c2/

28 Todd C. Miller. 2019. Sudo: CVE-2019-14287. Haettu 2.11.2021, sivustolta Openwall. Internetosoite: https://www.openwall.com/lists/oss-security/2019/10/14/1