Opinnäytetyö

Tradenomi, Tietojenkäsittelyn koulutus

2019

Juho Laine

JA3-TUNNISTEIDEN KÄYTTÖ OSANA IDS-JÄRJESTELMÄÄ

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Bachelor's degree in Business, Information technology

2019 | 32 pages

Juho Laine

JA3-HASHES IN INTRUSION DETECTION SYSTEMS

This thesis focused on developing detecting anomalies from encrypted traffic. The research is based on modern Intrusion Detection System's the need to consistently and reliably detect threats that use encrypted traffic in communication. Traditionally detecting threats form encrypted traffic is difficult and it generates a lot of false alarms.

The thesis is based on researching JA3-technology and implementing it into an existing open source IDS. JA3-hashes, or fingerprints, make it possible to determine the application that makes the encrypted connection without weakening the encryption or decrypting it. This makes it possible to detect anomalies reliably while maintaining high-performance. Both insider threats and attackers from outside can be caught regardless of the tools they use.

JA3-technology has been implemented into some open source Intrusion Detection Systems. The hashes are easy to form with the tools provided by the developers and in together with other available information they provide valuable information that can be used to detect evasion techniques used by the most skilled attackers. An important attribute that the hashes have is that they are easily shared within the information security community. When used in signatures with other data that is already available, it creates reliable detection with minimal false-positives.

KEYWORDS:

Information security, IDS, Intrusion Detection, JA3, Cyber Security

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tradenomi, Tietojenkäsittely

Kevät 2019 | 32 sivua

Juho Laine

JA3-TUNNISTEIDEN KÄYTTÖ OSANA IDS-JÄRJESTELMÄÄ

Opinnäytetyön tavoitteena oli kehittää uhantunnistusta salatusta liikenteestä. Tutkimus perustuu nykyaikaisen uhantunnistusjärjestelmän tarpeisiin havaita uhat luotettavasti ja tarkasti myös salatusta liikenteestä. Tällä hetkellä uhantunnistus salatusta verkkoliikenteestä on vaikeaa ja se aiheuttaa suuren määrän vääriä hälytyksiä.

Työ perustuu JA3-teknologian tutkimiseen ja sen implementoimiseen osaksi IDS-järjestelmää. JA3-tunnisteet mahdolistavat liikenteen lähettävän ohjelman havaitsemisen kuitenkaan purkamatta tai heikentämättä salausta. Näin voidaan rakentaa järjestelmä, joka havaitsee uhat luotettavasti, mutta jonka suorituskyky ei laske merkittävästi. Tunnisteiden käyttö helpottaa merkittävästi niin sisäisten kuin ulkoistenkin hyökkääjien havaitsemista riippumatta heidän käyttämistään työkaluista.

JA3-tunnisteet on otettu käyttöön joissain avoimen lähdekoodin IDS-järjestelmissä. Ne on helppo muodostaa mm. niiden kehittäjien tarjoamilla työkaluilla ja kun ne yhdistetään muihin tietueisiin voidaan tehdä tarkkoja johtopäätöksiä tietoliikenteen tarkoitusperästä ja näin havaita mahdolliset uhkatekijät. Eräs tärkeä ominaisuus niille on helppo jaettavuus, joka tekee niiden jakamisesta tietoturvayhteisön sisällä sujuvaa. Yhdistämällä useampia tietueita voidaan luoda tarkkoja signatuureja jotka havaitsevat taitavammatkin hyökkääjät. Opinnäytetyössä rakennettiin tällainen signatuuri ja se havaitsi annetun hyökkäysskenaarion onnistuneesti.

ASIASANAT:

Tietoturva, IDS, JA3, tietotekniikka, kyberturvallisuus

SISÄLTÖ

KUVAT

LYHENTEET JA SYMBOOLIT

APT	Advanced Persistent Threat, kehittynyt kohdennettu hyökkäys. Hyökkäys, jossa on tietty kohde ja jossa hyökkääjä yrittää niin kauan, että hyökkäys onnistuu.
C&C, C2	Command and Control. Haittaohjelmien ns. kotiin soittaminen, eli ne ottavat yhteyttä hallintapalvelimeensa.
Hyökkäysvektori	Reitti, jota pitkin hyökkääjä pyrkii pääsemään sisälle yrityksen verkkoon, kuten haavoittuvuus tai huijaussähköposti
IDS	Intrusion Detection System, tunkeutujan havaitsemisjärjestelmä
JA3	Kolmen tietoturvatutkijan nimikirjaimista muodostuva termi tavalle muodostaa tunnisteita salatusta liikenteestä.
OSINT	Open Source Intelligence, avoimista lähteistä, kuten julkisista sosiaalisen median tileistä saatava tieto
Phishing	Tietojen kalastelu
TLS ja SSL	Transport Layer Security ja Secure Sockets Layer ovat tietoliikenteen salaamiseen käytettyjä protokollia. TLS on uudempi versio mutta molemmista puhutaan yleensä nimellä SSL
MD5	Tiivistealgoritmi, jota käytetään esimerkiksi kryptografiassa. Algoritmi muuttaa sille annetun syötteen MD5-tiivisteeksi

1 JOHDANTO

Yritykset ovat jatkuvasti erilaisten kyberhyökkäysten kohteena. Hyökkäyksiä voidaan pyrkiä havaitsemaan ja ehkäisemään erinäisin keinoin, mutta kaikkea haitallista liikennettä ei koskaan voida suodattaa pois. Viime vuosina yhä suurempi osa internetistä on siirtynyt käyttämään salattua yhteyttä. Salattu yhteys (useimmin nettisivujen HTTPS-yhteys) on hyvä asia yksilön tietosuojalle ja yksityisyydelle, mutta vaikeuttaa merkittävästi hyökkäysten ja muun haitallisen liikenteen havaitsemista.

Koska hyökkääjät pyrkivät peittämään jälkensä, he käyttävät usein kohteen laitteella jo olemassa olevia työkaluja ladatakseen ja suorittaakseen haitallisia ohjelmia. Koska liikenne on salattua, ei ole mahdollista lukea liikennettä ja määrittää siten liikenteen tarkoitusperää.

JA3 on tunniste, joka voidaan muodostaa helposti salatusta liikenteestä. Nämä tunnisteet ovat helposti jaettavissa yhteisön kesken. Ne voidaan myös implementoida osaksi NIDS-järjestelmää. Tunniste muodostetaan salauksen neuvottelussa kerätyistä tietueista. Jokaisella ohjelmalla on oma tunnisteensa, jota kutsutaan JA3-sormenjäljeksi. Sen avulla on mahdollista tunnistaa liikenteen tarkoitusperä kuitenkaan heikentämättä salausta tai vaarantamatta työntekijöiden tietosuojaa tai yksityisyyttä.

Opinnäytetyössä tarkastellaan tunnisteen luomista, ja pohditaan sen soveltuvuutta osaksi NIDS-järjestelmää. Tavoitteena on luoda toimivia signatuureja, joilla voidaan havaita viime vuosien suurissa tietomurroissa käytettyjä tekniikoita. Näistä tekniikoista merkittävin on Windows-käyttöjärjestelmään sisäänrakennettujen työkalujen käyttö haittaohjelmien lataamiseen. Lisäksi implementoidaan yksi esimerkkitapaus osaksi signatuuripohjaista NIDS-järjestelmää. Opinnäytetyön toimeksiantaja on turkulainen tietoturvayritys Fiarone Oy. Fiarone erikoistuu tietoteknisten järjestelmien ympärivuorokautiseen valvontaan Suomessa ja maailmalla.

Myös palvelimista voidaan luoda vastaavia tunnisteita. Nämä yhdistettäessä ohjelman tuottamaan tunnisteeseen voidaan luotettavasti havaita tunnettujen haittaohjelmien toiminta. Jotta saadaan luotettava lista C2-palvelinten tunnisteista, tarvitaan valtava määrä esimerkkidataa, joten se jää tämän opinnäytetyön ulkopuolelle. Tunnisteita voidaan hyödyntää myös tekoälyn avulla, mutta sitä ei käsitellä tässä opinnäytetyössä.

2 YRITYKSEN TIETOTURVA

Tietoturva on laaja käsite, jolla viitataan yleisesti tietokoneiden, älylaitteiden, sähköisten järjestelmien, tietoliikenneverkkojen ja niiden sisältämän datan suojaamiseen (Kaspersky 2019). Maailmassa, josta jatkuvasti suurempi osa toimii tietokoneiden varassa, on vastaavasti jatkuvasti tärkeämpää huomioida niiden turvallisuus. Tietoturva voidaan jakaa karkeasti kolmeen osaan käyttäen CIA-mallia: tietojen luottamuksellisuuteen (Confidentiality), eheyteen (Integrity) ja saatavuuteen (Availability). (Techtarget 2019)

Tietojen luottamuksellisuus tarkoittaa, että ne ovat ainoastaan niiden henkilöiden saatavilla, joilla on oikeus niihin. Tietojen luottamuksellisuutta voidaan verrata yksityisyyteen. Tietojärjestelmissä tietojen luottamuksellisuus varmistetaan usein salaamalla liikenne ja vaatimalla käyttäjää tunnistautumaan nähdäkseen tietoja.

Tietojen eheys tarkoittaa tietojen luotettavuutta ja tietojen muuttumattomuutta. Tämä tarkoittaa käytännössä, että tahot, joilla ei ole oikeutta muokata tietoja eivät pysty muokkaamaan niitä. Tietoliikenteen salaaminen, versionhallinta ja tarkistussumma pyrkivät varmistamaan, että tieto ei muutu matkalla.

Saatavuus viittaa siihen, että tiedot ja järjestelmät ovat aina niiden henkilöiden saatavissa, joilla on oikeus niihin. Haitalliset tahot pyrkivät usein häiritsemään yrityksien toimintaa palvelunestohyökkäyksillä, jolloin tavalliset käyttäjät eivät pysty käyttämään järjestelmää tai palvelua. Tietojen saatavuus varmistetaan jatkuvalla tietojärjestelmien päivityksillä ja varmuuskopioilla, joita ylläpidetään jatkuvasti sen varalta, että järjestelmästä tulee käyttökelvoton.

2.1 Uhat yrityksen tietoturvalle

Yritykset ovat jatkuvasti hyökkääjien kohteena. Tietoturva on helppo mieltää virustorjuntaohjelmaksi, joka pyörii laitteella valvoen ja estäen haitallisten ohjelmien suorittamisen. Tietoturva on kuitenkin hyvin laaja alue, eikä hyökkäysvektori useimmiten ole haavoittuva tietojärjestelmä vaan sen käyttäjä. Yksi käyttäjä voi altistaa koko yrityksen haittaohjelmille, jotka ovat usein vasta toinen vaihe hyökkäyksessä.

2.1.1 Tietojen kalastelu

Macy Bayer (2018) luettelee artikkelissaan viisi suurinta uhkaa tietoturvalle vuonna 2018. Artikkelin lähdeaineistona on käytetty kyselyä, johon vastasi 900 tietoturvan ammattilaista. Yli puolet, 55%, vastasi tietojen kalastelun (Phishing) olevan suurin uhka yritysten tietoturvalle. Vain 45% vastanneista piti kiristyshaittaohjelmia suurempana uhkana kuin kalastelua. Tämä on loogista, sillä haittaohjelmat päätyvät yrityksen järjestelmiin tietojen kalastelun seurauksena.

Symantecin vuoden 2018 ISTR:n (Symantec 2018) mukaan jopa 92,8 % kaikista haittaohjelmista toimitetaan sähköpostin välityksellä. Hyökkääjä pyrkii saamaan haltuunsa ensin yrityksen työntekijän sähköpostin, jolta voidaan lähettää haittaohjelmia esimerkiksi liitetiedostoissa tai latauslinkkien avulla. Muut työntekijät painavat näitä linkkejä ja avaavat tiedostoja todennäköisemmin, koska ne ovat luotettavasta lähteestä.

Symantecin raportin mukaan 55% kaikesta sähköpostiliikenteestä on haitallista. Alert Logicin artikkelin (Alert Logic 2018) mukaan keskivertotyöntekijä saa kuukaudessa 16 kalastelusähköpostia. Yritys voi pyrkiä vähentämään tietojenkalastelun uhkaa esimerkiksi käyttämällä monivaiheista kirjautumista, mutta tekniset toimet eivät yksin riitä kalastelulta suojautumiseen. Henkilöstön koulutus on suuressa osassa yrityksen tietoturvassa.

2.1.2 Haittaohjelmat

Haittaohjelmat ovat perinteisin uhka tietoturvalle. Haittaohjelma tarkoittaa ohjelmaa, jonka tarkoitus on haitata tai estää tartunnan saaneen järjestelmän toimintaa, vakoilla käyttäjää tai hyväksikäyttää järjestelmän resursseja. Symantecin raportin mukaan kiristyshaittaohjelmat aiheuttavat eniten vahinkoja yrityksille. Kiristyshaittaohjelman tarkoitus on salata saastunut järjestelmä ja tietojen katoamisen uhalla kiristää uhria maksamaan lunnaat. Vuonna 2017 julkisuudessa olleet WannaCry ja Petya ovat esimerkkejä tällaisista ohjelmista.

Suosittuja haittaohjelmia ovat myös Emotetin tapaiset pankkitapahtumiin kohdistetut troijalaiset. Näiden tarkoitus on ohjata pankkeihin tarkoitettu liikenne ja tapahtumat haitallisille sivustoille. Näihin troijalaisiin on usein rakennettu sisään myös ns. takaovi, joka antaa hyökkääjän ottaa järjestelmän haltuun etänä. Troijalaiset ovat monikäyttöisiä

ja voivat edellä mainittujen lisäksi mm. tallentaa käyttäjän näppäimien painalluksia, varastaa evästeitä ja asentaa muita haitallisia ohjelmistoja (Investopedia 2019).

2.1.3 Kryptovaluutan louhiminen

Vuonna 2016 kryptovaluutat alkoivat saada suosiota ja niiden arvo nousi räjähdysmäisesti. Tämän seurauksena myös kyberrikolliset siirtyivät osittain näitä valuuttoja louhivien haittaohjelmien käyttämiseen. Kryptovaluutan louhiminen tarkoittaa sitä, että louhijaohjelma käyttää suuren määrän prosessointitehoa ratkaistakseen monimutkaisia matemaattisia ongelmia ja ne ratkaistuaan palkkioksi saadaan louhittua valuuttaa. Kryptovaluuttojen louhiminen vie suuren määrän järjestelmän resursseja ja kuluttaa valtavan määrän sähköä. Tämän takia rehellinen louhiminen ei ole usein kannattavaa yksittäisille toimijoille (Trendmicro 2017).

Kyberrikolliset ovat siirtyneet suurissa määrin kryptovaluutan louhimiseen uhrien järjestelmissä. Kryptovaluutan louhiminen on varmempi tulonlähde kuin esimerkiksi kiristyshaittaohjelmat. Kiristyshaittaohjelmien ongelmana on, että yrityksellä ei ole takeita siitä, että he saavat tietonsa ja järjestelmänsä takaisin. Tämän vuoksi on usein suositeltavaa, että yritys ei maksa lunnaita. Lisäksi jos järjestelmästä ylläpidetään varmuuskopioita, on se helppo palauttaa, jolloin lunnaita ei edes tarvitse maksaa.

2.1.4 Kehittyneet hyökkäykset

APT-hyökkäykset ovat kohdennettuja, moniosaisia ja pitkäkestoisia hyökkäyksiä. Ne hyödyntävät suurta määrää eri haavoittuvuuksia ja tekniikoita tietojen kalastelusta (Phishing) nollapäivähaavoituvuuksiin. Yksittäiset vaiheet eivät ole erityisen kehittyneitä, mutta hyökkääjillä on tarvittaessa taidot ja resurssit nostaa hyökkäyksen kehittyneisyyden tasoa. Kehittyneiden hyökkäyksien takana ovat usein vieraiden valtioiden hakkerit tai muut suuret alan toimijat. APT-hyökkäykset eroavat perinteisistä hyökkäyksistä siten, että ne on kohdistettu tarkasti tiettyyn kohteeseen, ja hyökkääjä käyttää useaa hyökkäysvektoria. Sen sijaan perinteisissä hyökkäyksissä kohteita on useita, ja mikäli kokeiltu tekniikka ei toimi, pyritään etsimään kohde, joka on haavoittuva toisen tekniikan kokeilemisen sijaan.

APT-hyökkäykset voidaan jakaa kuuteen vaiheeseen: Tiedustelu ja aseistaminen, toimitus, alustava tunkeutuminen, C&C, poikittainen liikkuminen ja tietojen tuominen kohteesta (Chen, Desmet and Huygens 2014).

Tiedustelulla ja aseistamisella tarkoitetaan tiedon keräämistä, joka suoritetaan ennen varsinaista hyökkäystä. Hyökkääjä kerää mahdollisimman paljon teknistä tietoa kohdeympäristöstä ja -organisaatiosta sekä sen merkittävistä työntekijöistä ja yhteistyökumppaneista. Tiedonhakuvaiheessa hyödynnetään usein myös sosiaalista manipulointia ja OSINT-tiedonhakua. Sosiaalista manipulointia voidaan käyttää arkaluontoisen tiedon saamiseen, tai siten voidaan saada uhri suorittamaan haittaohjelma. OSINT tarkoittaa tiedon keräämistä avoimista lähteistä. Esimerkiksi kohteen web-sovellusten hyödyntämät ohjelmistoversiot ja niiden mahdolliset haavoittuvuudet voidaan joskus kerätä netistä. Kerätyn tiedon pohjalta hyökkääjät luovat monivektorisen hyökkäyssuunnitelman.

Toimitusvaiheessa (Delivery) hyökkääjä toimittaa haittaohjelman kohteeseen esimerkiksi huijaussähköposteilla. Näissä sähköposteissa käytetään kerättyä tietoa ja pyritään tekemään jokaisesta viestistä mahdollisimman henkilökohtainen. Hyökkääjien on myös todettu teeskentelevä olevansa esim. hyväntekeväisyysjärjestöjä. Kohde yritetään saada avaamaan liitetiedosto tai painamaan linkkiä, joka johtaa haittaohjelman lataamiseen ja suorittamiseen. Haittaohjelma voidaan toimittaa myös epäsuorasti kolmannen osapuolen kautta. Tällöin hyökkääjä pyrkii saamaan haltuunsa sivuston, jolla kohde vierailee usein ja toimittaa haittaohjelman tämän sivun kautta. Tätä kutsutaan watering hole (juomapaikka) -hyökkäykseksi, ja esimerkiksi OceanLotus (APT32, APT-C-00) ryhmä on tunnetusti käyttänyt niitä (Faou 2018).

Alustava tunkeutuminen tapahtuu, kun hyökkääjä pääsee sisään kohteen tietokoneelle tai verkkoon. Tunkeutuminen voidaan tehdä joko tieduteluvaiheessa kerätyn tiedon perusteella (tunnus ja salasana) tai toimitusvaiheessa toimitetun haittaohjelman tai haavoittuvuuden suorittamisen kautta. Haavoittuvuuksien kohteina ovat usein Adoben PDF- ja Flash-ohjelmistot ja muut yleiset toimisto-ohjelmat, kuten Microsoft Office. Onnistuessaan tämä vaihe johtaa yleensä ns. takaoven asentamiseen kohdelaitteelle, joka antaa hyökkääjälle pääsyn sille. Takaovi luo tässä vaiheessa liikennettä C2-palvelimelle, joten hyökkäys voidaan havaita.

Command and Control (C&C, C2) -vaiheessa ns. takaovi on asennettu, ja hyökkääjä ottaa laitteen haltuunsa. Yhteys laitteeseen otetaan usein legitiimien ja julkisten

palveluiden välityksellä, jotta voidaan välttää havaituksi tuleminen. Koska C2-liikenne voi kulkea legitiimien palveluiden kautta, jotka usein hyödyntävät salattua liikennettä, on sen havaitseminen vaikeaa.

Poikittainen liikkuminen (Lateral movement) vie suurimman osan hyökkäykseen kuluvasta ajasta. Hyökkääjä pyrkii levittäytymään muihin verkon laitteisiin keräten samalla tietoa, jolla nostaa käyttöoikeuksien tasoa käyttäjästä jäjestelmänvalvojaksi. Tässä vaiheessa pyritään myös kartoittamaan, missä kohteena olevat tiedostot ovat. Viime vuosina ns. ”living off the land” -tekniikat, joissa käytetään haittaohjelmien sijaan valmiiksi olemassa olevia, legitiimejä ohjelmistoja, kuten Windowsin PowerShellia, ovat yleistyneet. Nämä työkalut ovat usein järjestelmän ylläpitäjien käytössä, joten on vaikeaa erottaa hyökkääjää legitiimistä toiminnasta. Hyökkääjät toimivat hitaasti ja matalalla profiililla välttääkseen paljastumista.

Tietojen tuominen (Data Exfiltration) on viimeinen vaihe hyökkäystä. APT-hyökkäyksien tarkoitus on usein kerätä arkaluontoista tai salaista tietoa, jolla voidaan saavuttaa joko kilpailullisia tai strategisia etuja. Kun nämä tiedot on paikannettu ja niihin on saatu oikeudet, hyökkääjät tuovat ne salattua yhteyttä pitkin omalle tai muuten hallussaan olevalle palvelimelle.

2.2 Uhilta suojautuminen

Tässä kappaleessa kuvaillaan erilaisia tapoja, joilla yritys ovi suojautua edellä mainituilta uhilta. Lisäksi vertaillaan eri suojautumiskeinojen hyviä ja huonoja puolia. On kuitenkin tärkeää muistaa, että nämä järjestelmät toimivat usein toistensa tukena eikä niistä mikään ole täydellinen. Usean puolustuksen käyttäminen eri kerroksina antaa parhaan suojan uhkia vastaan.

2.2.1 Virustorjuntaohjelmat

Virustorjuntaohjelmisto on palomuurin ohella perinteisin tapa puolustautua haittaohjelmia vastaan. Ohjelmat pyörivät yrityksen työasemilla ja estävät haitallisten ohjelmien suorittamisen. Virustorjuntaohjelmat vertaavat suoritettavien ohjelmien lähdekoodin muodostamaa signatuuria tai MD5-tiivistettä tunnettujen virusten tietokantaan. Mikäli ohjelman tunniste vastaa tunnetun haittaohjelman tunnistetta ohjelma luo hälytyksen ja sen suorittaminen estetään. Koska vain tunnetut virukset ovat tietokannassa ei nollapäiväviruksia havaita virustorjuntaohjelmistoilla (How Does an Antivirus Program Work to Protect Your Computer? 2019).

Virustorjuntaohjelmia on sekä maksullisia että maksuttomia mutta yleisesti yrityksen käyttöön suositellaan maksullista ohjelmistoa. Maksulliset ohjelmistot havaitsevat uudet haittaohjelmat tehokkaammin ja niihin voi kuulua myös muita ominaisuuksia kuten pankkitapahtumien suojaus (Quain 2016).

2.2.2 Palomuurit ja IPS-järjestelmät

Palomuuri ja IPS ovat samankaltaisia mutta hieman erilaisia järjestelmiä. Palomuuri on perinteisempi ratkaisu. Se havaitsee mahdollisia uhkia muun muassa lähde- ja kohdeosoitteiden, käytettyjen porttien ja protokollan perusteella. Mikäli liikenne ei ole sallittua, palomuuri tiputtaa sen. IPS on kehittyneempi järjestelmä, joka sijaitsee palomuurin ja sisäverkon välissä. Näin liikenteen pitää ensin läpäistä palomuurin säännöt, jotka suodattavat pois suuren osan ei-halutusta liikenteestä. IPS tutkii paketin sisältöä ja voi analyysin perustella tiputtaa haitalliseksi epäillyn liikenteen. Järjestelmä päättää, onko liikenne haitallista signatuurien ja käytöksen analyysin avulla. Palomuuri

ja IPS voidaan myös yhdistää niin sanotuksi Next-Generation Firewalliksi (Chandan 2017).

2.2.3 IDS-järjestelmät

IDS toimii usein palomuurin ja IPS:n tukena. IDS analysoi koko paketin tai lokirivin ja vertaa sitä signatuureihin. Jos paketti täsmää johonkin signatuuriin, järjestelmä luo hälytyksen. Merkittävä ero IPS:ään on se, että IDS ei estä liikennettä. IDS-järjestelmät jaetaan usein kahteen ryhmään: HIDS eli Host-based IDS, joka perustuu kohteen lokitietojen läpikäymiseen, ja NIDS eli Network-based IDS, joka kuuntelee IPS:n tavoin verkkoliikennettä. Suurimmat ongelmat IDS-järjestelmissä on, että ne tuottavat paljon vääriä hälytyksiä, ja että ne vaativat jatkuvaa ylläpitoa. Signatuurit ovat usein liian herkkiä ja voivat laueta aivan muista tapahtumista, kuin mihin ne on tarkoitettu. Tämän vuoksi IDS tuottaa usein paljon ylimääräistä työtä, ja useat yritykset ovat ulkoistaneet palvelun. Näistä palveluista käytetään nimitystä Security Operations Center tai SOC.

2.2.4 Honeypotit

Honeypotit ovat verkkoon asetettuja ansoja. Ne simuloivat yrityksen verkon toimintaa ns. hiekkalaatikossa, jolloin hyökkääjän toimia voidaan havaita ja tutkia turvallisessa ja eristetyssä ympäristössä. Honeypoteilla voidaan tutkia ja havaita hyökkääjien käyttämiä nollapäivätekniikoita, tai ne voidaan asentaa osaksi yrityksen verkkoa.

Esimerkki yrityksen verkossa olevasta honeypotista on virtuaaliseen ympäristöön asennettu ylimääräinen tietokantapalvelin. Tällä palvelimella on lähes identtinen ohjelmisto kuin oikealla palvelimella, mutta sen sisältämä tietokanta on väärä. Lisäksi sinne voi kirjautua millä tahansa yrityksen tunnuksella, kun taas oikealle palvelimelle tarvitaan järjestelmänvalvojan tunnukset. Tavallisilla käyttäjillä ei ole tietoa tästä palvelimesta, joten he eivät yritä kirjautua sille vahingossa. Hyökkääjä puolestaan voi nähdä palvelimen suorittaessaan tiedustelua yrityksen verkossa. Löydettyään palvelimen hän yrittää kirjautua sisään, jolloin järjestelmä luo hälytyksen. Tämän jälkeen on mahdollista havaita, miltä laitteelta hyökkäys on peräisin, eristää se verkosta ja poistaa uhka. Honeypotit eivät ole täysin varma tapa havaita hyökkääjiä, ja ne toimivat perinteisempien tietoturvaratkaisujen tukena. Huomattava etu niissä on niiden mahdollisuus havaita nollapäiväuhkia.

2.2.5 Henkilöstön koulutus

Henkilöstön kouluttamisesta on viime vuosina tullut yhä tärkeämpi osa tietoturvapolitiikkaa. Hyökkääjät pyrkivät yhä useammin saamaan haltuunsa oikeiden käyttäjien tietoja kalasteluviesteillä. Mitä taitavampi hyökkääjä on, sitä uskottavampi huijausviesti. Hyökkääjät etsivät kohdeyrityksen työntekijöistä OSINT-tietoa ja luovat sen perusteella personalisoituja huijausviestejä.

Varsinkin vanhemman väestön on vaikeaa tunnistaa huijausviestejä. Ennen on opetettu, että huijausviestit on kirjoitettu huonolla suomella tai että ne tulevat varmentamattomasta sähköpostista. Viestit ovat kehittyneet viime vuosina paljon, eikä näihin tekijöihin voi enää luottaa.

Usein myös vedotaan siihen, että ”ei minun sähköpostillani kukaan mitään tee”, vaikka tosiasiassa yhden ihmisen sähköpostin saaminen mahdollistaa tietojenkalastelun kaikilta niiltä tahoilta, jotka luottavat tähän henkilöön. Tavallisen työntekijän sähköpostista voidaan esimerkiksi lähettää verokortiksi naamioitu haittaohjelma, joka leviää taloushallintoon ja sieltä kaikille yrityksen työntekijöille, asiakkaille ja yhteistyökumppaneille. Huijausviestit ovat helppo tie suoraan kohteen verkkoon, ja niitä on vaikea estää ja havaita käyttämällä teknisiä ratkaisuja. Tämä tekee niistä erittäin tehokkaan tavan saada jalansija yrityksen verkkoon.

3 JA3-TUNNISTEET OSANA IDS-JÄRJESTELMÄÄ

JA3 on kolmen tietoturvatutkijan, John B. Althousen, Jeff Atkinsonin ja Josh Atkinsin kehittämä tapa luoda tunnisteita salatusta yhteydestä. Näitä tunnisteita kutsutaan JA3-sormenjäljiksi. JA3-somenjäljet ovat helposti jaettavissa eri tiedonjakoalustoilla, joka tekee niistä hyvän tavan havaita tunnettujen haittaohjelmien muodostama liikenne. Koska lähes jokaisella ohjelmalla on erilainen, tai useampi, sormenjälki, niitä voidaan hyödyntää tapausten tutkinnassa.

JA3-tunnisteisiin pohjautuva analyysi tukee perinteisempää analyysiä, joka perustuu pitkälti IP-osoitteisiin, palvelinten nimiin ja paketin sisältöön. Varsinkin viimeinen tuottaa ongelmia NIDS-järjestelmille, sillä ne eivät voi lukea salatun paketin sisältöä. Vaikka yhteyden salaaminen on tärkeää yksilön tietosuojan kannalta, ja sillä voidaan ehkäistä Man-in-the-middle hyökkäyksiä. Salatusta liikenteestä saadaan joitain tietueita, kuten kohdepalvelimen nimi TLS-yhteyden SNI-tunnisteen avulla. Koska sisältöä ei voida lukea, on mahdotonta tietää mikä sen tarkoitus on. IP-osoite puolestaan luo suuren määrän vääriä hälytyksiä koska yhdellä palvelimella voidaan isännöidä satoja nettisivuja, jotka ovat harmittomassa käytössä.

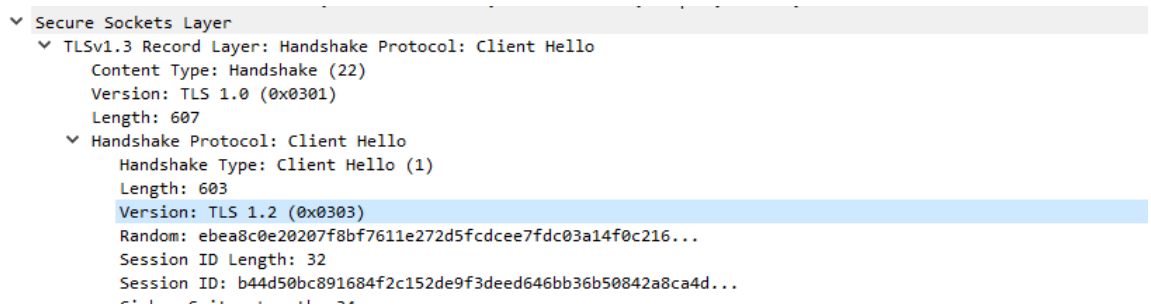
Tässä kappaleessa tarkastellaan, miten ja mistä tietueista tunnisteet muodostetaan ja pohditaan niiden soveltuvuutta ajoittain hektiseen SOC-ympäristöön.

3.1 Tunnisteiden muodostaminen

JA3-sormenjäljet muodostetaan sovelluksen käyttämästä SSL-versiosta, sen hyödyntämistä cipher suiteista, TLS-jatkeista, elliptisestä kaaresta ja sen pistemuodosta. Tämä tietuelista ovi olla melko pitkä, joka tekee sen jakamisesta ja käyttämisestä epäkäytännöllistä. Nämä tiedot asetetaan pilkulla erotettuina listaksi ja niistä muodostetaan MD5-tiiviste. Tätä tarkistetta kutsutaan JA3-sormenjäljeksi. Tarkistetta on helppoa jakaa ja sitä voidaan käyttää hyödyksi signatuuripohjaisessa IDS-järjestelmässä (Althouse 2019).

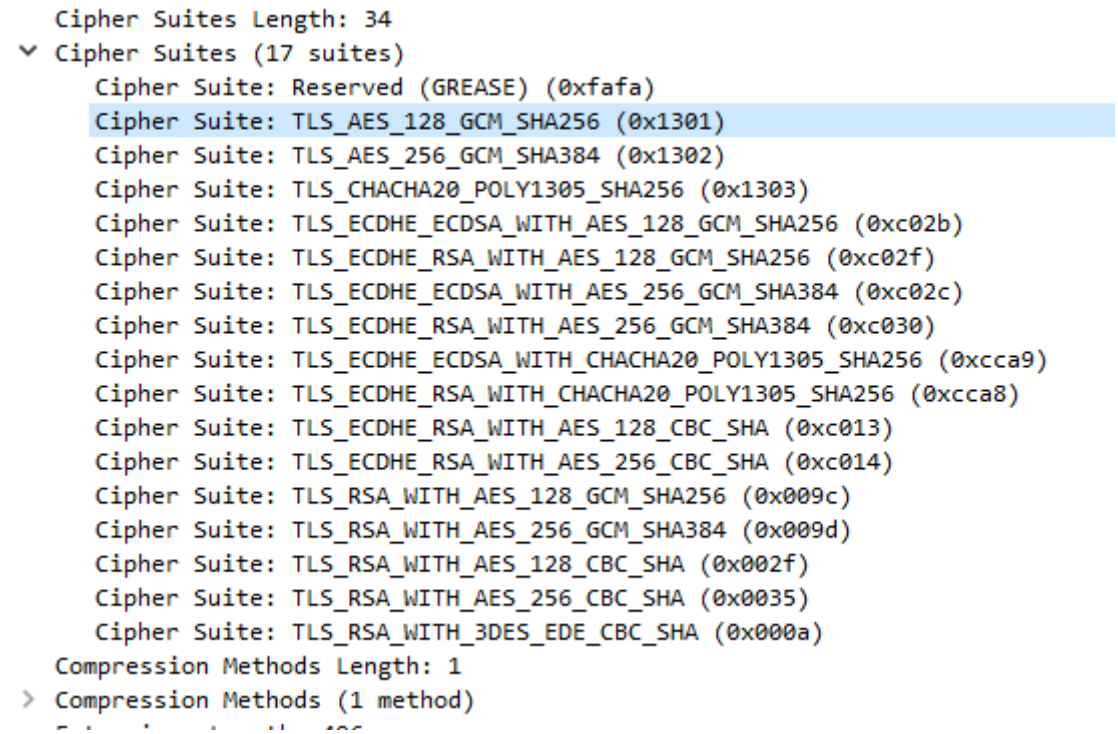
Tunniste muodostetaan SSL-yhteyden alussa lähetetyistä yhteydenneuvottelupaketeista (Client Hello). Näitä voidaan tarkastella käyttämällä verkkoliikenteen tutkimiseen tarkoitettua Wireshark-ohjelmistoa Windows-ympäristössä

tai TCPDump-ohjelmaa Linux-ympäristössä. Esimerkkiä varten otetaan salattu yhteys osoitteeseen https://ylilauta.org käyttäen Google Chromen 64-bittistä versiota 72.0.3626.109 Windows 10-laitteella.



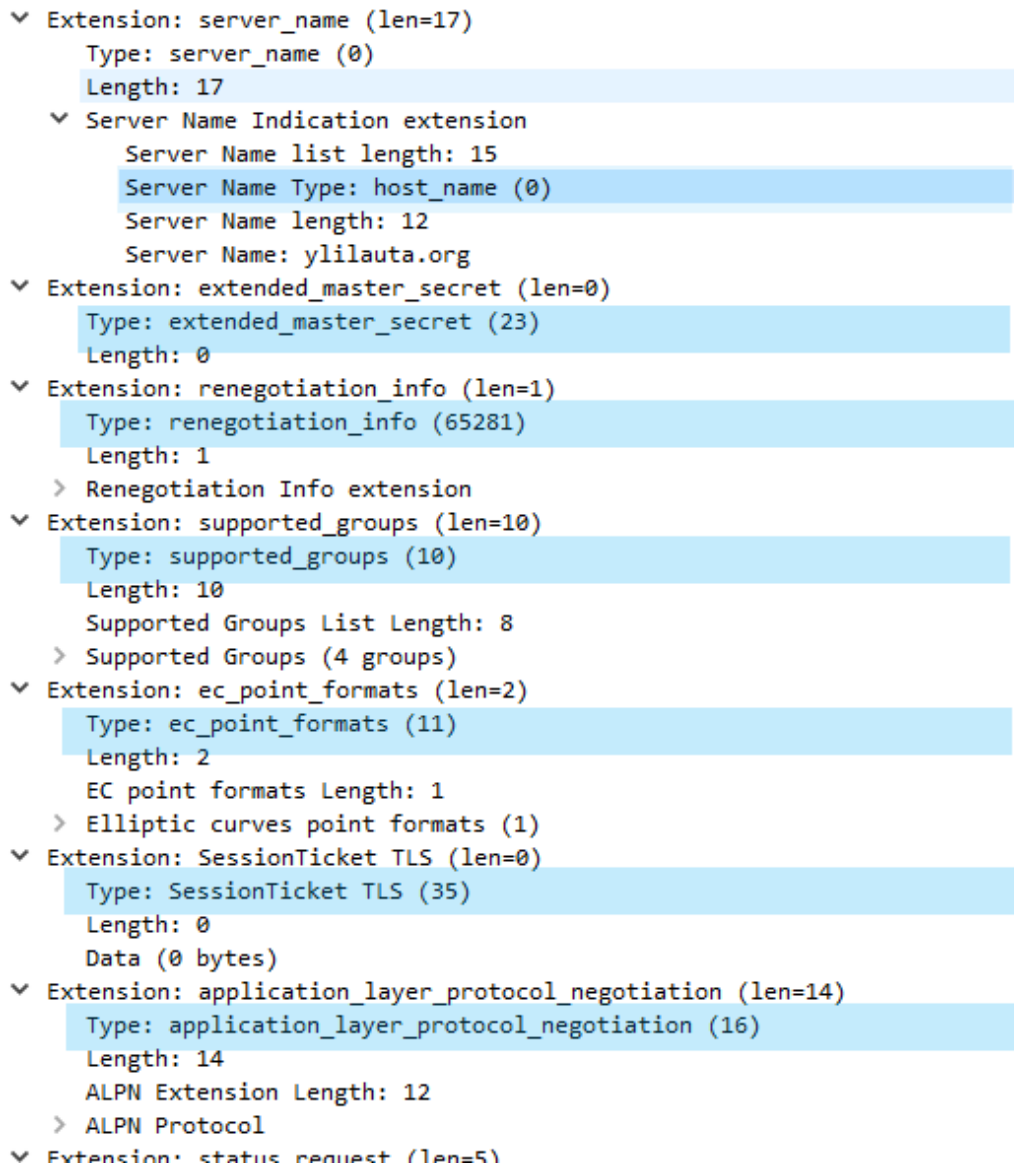
Kuva 1. TLS-versio unicode-heksadesimaalina.

Cipher suitet (Kuva 2) muunnetaan heksadesimaaleista desimaaliluvuiksi ja asetetaan listaksi. Koska niitä on useita, ne erotellaan väliviivalla. Näin saadaan seuraava merkkijono: 4865-4866-4867-49195-49199-49196-49200-52393-52392-49171-49172-156-157-47-53-10, jonka alkuun lisätään kuvan 1 osoittama TLS-versio (0x0303 = 771) pilkulla erotettuna.



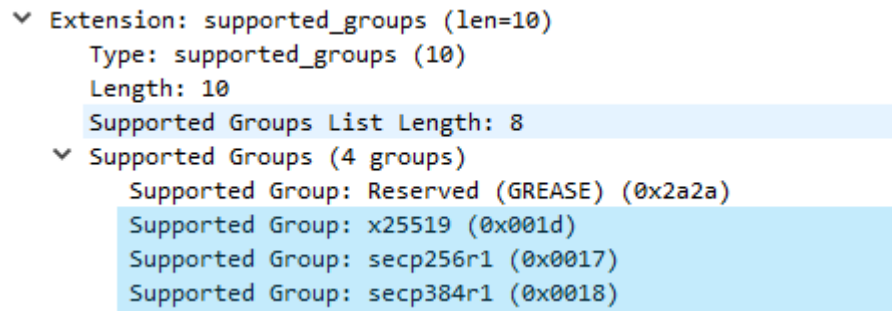
Kuva 2. Ohjelman käyttämät cipher suitet unicode-heksadesimaaleina.

Osa jatkeiden tunnisteluvuista jää kuvan 3 ulkopuolelle. Niistä saadaan merkkijono: 0-23-65281-10-11-35-16-5-13-18-51-45-43-27-41, joka jälleen lisätään pilkulla erotettuna osaksi aiemmin saatua listaa.



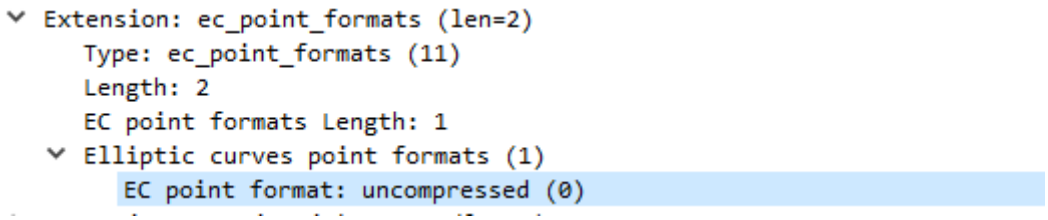
Kuva 3. TLS-jatkeiden desimaalimuotoiset tunnisteet.

Kuvan 4 luvut desimaaleiksi muunnettuna: 29-23-24.



Kuva 4. Elliptisen kaaren tunnisteet heksadesimaaleina.

Kuvasta 5 saadaan viimeinen tarvittava tietue eli elliptisen kaaren pistemuoto. Tässä tapauksessa sovellus tukee ainoastaan pakkaamatonta dataa, joten arvo on 0. Asettamalla kaikki saadut merkkijonot pilkulla eriteltyinä yhteen saadaan JA3-merkkijono: 771,4865-4866-4867-49195-49199-49196-49200-52393-52392-49171-49172-156-157-47-53-10,0-23-65281-10-11-35-16-5-13-18-51-45-43-27-41,29-23-24,0. Kun tästä merkkijonosta otetaan MD5-tiiviste, saadaan ohjelman JA3-sormenjälki: *554719594ba90b02ae410c297c6e50ad* (Althouse 2019).



Kuva 5. Elliptisen kaaren pistemuoto

JA3-tunnisteden muodostaminen käsin on varsin työlästä ja siihen on kehitetty useita työkaluja. JA3:n kehittäjien Pythonilla kirjoitettu työkalu tunnisteiden luomiseen on saatavilla heidän GitHub-repositoriostaan. Tämän opinnäytetyön käytännön osassa käytetään hyväksi avoimen lähdekoodin Suricata IDS-järjestelmää, jossa on versiosta 4.1 eteenpäin sisäänrakennettu tuki JA3-tunnisteiden muodostamiselle (Atkinson, Althouse and Atkins 2019). Käytännön osassa tutkitaan signatuurin rakentamista käyttäen JA3-tunnistetta.

3.2 Tunnisteiden käytettävyys

JA3-tunnisteiden käytettävyys osana luotettavaa uhantunninstusta ei ole yksinkertaista. Moni samalla tavalla toimiva ohjelma antaa usein saman tunnisteen. Esimerkiksi monet Javalla kirjoitetut ohjelmat antavat saman tai yhden tietyistä sormenjäljistä. Tiettyjen ohjelmien toimintaa voidaan kuitenkin havaita. Yhdistämällä tunnisteet esimerkiksi IP-osoitteisiin on mahdollista havaita, jos vaikka Windowsin järjestelmänvalvojille tarkoitettu PowerShell-ohjelma ottaa yhteyttä haitalliselle palvelimelle. PowerShell on skriptaustyökalu, joka on oletuksena asennettu kaikille Windows-laitteille. Tämä tekee siitä erittäin arvokkaan hyökkääjille, sillä heidän ei tarvitse asentaa ulkoista ohjelmaa kohteen laitteelle ottaakseen sen haltuun.

PowerShellilla on muutamia eri moduuleita, joita voidaan käyttää C2 -palvelimen kanssa kommunikointiin. Yleisimmät lienevät *Invoke-WebRequest* ja *BITSAdmin,* joilla voidaan hakea tiedostoja verkkopalvelimilta. Näin saatuaan jalansijaa yrityksen verkossa hyökkääjä voi ladata lisää työkaluja kohteen hallitsemiseksi. Koska nämä työkalut jättävät aina saman sormenjäljen, voidaan luoda signatuureja, joissa yhdistetään sormenjälki IP-osoitteisiin, palvelinten nimiin tai muihin jo olemassa oleviin tunnisteisiin. Nämä voidaan jakaa kolmeen kategoriaan:

Tunnetut hyvät. Palvelimet tai osoitteet, jotka ovat todistettavasti luotettavia. Esimerkiksi kun BITS ottaa yhteyden Microsoftin palvelimeen, on yhteys todennäköisesti Windowsin päivitys. Näistä ei luoda tapahtumia SIEM-järjestelmään eikä niitä tallenneta muuhunkaan tietokantaan.

Tuntemattomat. Tähän kategoriaan jää suurin osa palvelimista ja osoitteista. Niitä ei ole todettu luotettaviksi eikä haitallisiksi. Näistä luodaan tapahtuma tietokantaan jatkoanalyysia varten. Tapahtumaa ei luoda SIEM:iin.

Tunnetut pahat. Tunnetut palvelimet, joilla isännöidään haitallisia ohjelmia tai jotka ovat tunnetusti epäilyttävän tahon käytössä. Internetissä on useita palveluita, jotka ylläpitävät listoja haittaohjelmien C2-palvelimista, kuten Abuse.ch ja Bambenek Consulting.

Yhdistämällä tunnisteita keskenään saadaan rajattua pois suuri määrä vääriä hälytyksiä kuitenkaan vähentämättä havaintojen luotettavuutta. Tunnisteita voidaan hyödyntää myös tekoälyn avulla. Tekoäly voidaan asettaa oppimaan, millaiset sormenjäljet ottavat yhteyttä mihinkin palvelimiin ja ilmoittamaan poikkeamista.

3.3 Palvelinten JA-tunnisteet

Kuten ohjelmilla, myös palvelimilla on tietynlainen sormenjälki. Tämä muodostetaan lähes samalla tavalla kuin ohjelmien sormenjälki. Näin saatua tunnistetta kutsutaan JA3S-sormenjäljeksi. Tutkijat havaitsivat, että tietyn haittaohjelman ottaessa yhteyttä tiettyyn hallintapalvelimeen on niiden muodostaman yhteyden sormenjälki aina sama riippumatta siitä, miltä laitteelta tai mistä verkosta yhteys muodostetaan. Yhdistämällä JA3 ja JA3S -sormenjäljet voidaan luoda tunniste, joka kertoo lähes takuuvarmasti haittaohjelman toiminnasta. Tällaisten tunnisteiden luominen vaatii suuren määrän näytteitä ja tutkimusta. (Althouse 2019)

4 INTEGROINTI OSAKSI NIDS-JÄRJESTELMÄÄ

NIDS on järjestelmä, jonka tarkoitus on analysoida verkkoliikennettä ja luoda siitä signatuuri- ja käytöspohjaisen analyysin avulla hälytyksiä. Monet haittaohjelmat ovat siirtyneet käyttämään SSL-salattua yhteyttä ottaessaan yhteyttä hallintapalvelimeensa. Tämä auttaa niitä välttämään paketin sisältöön perustuvaa analyysia. IP-osoitteeseen perusteella hälytysten luominen puolestaan aiheuttaa kohtuuttoman määrän vääriä hälytyksiä. Tavallinen netin selaaminen laukaisee näitä hälytyksiä, sillä selain joutuu hakemaan sisältöä monesta paikasta, ja varsinkin mainokset on usein isännöity mahdollisimman halvalla palvelimella – jotka ovat myös haittaohjelmien suosiossa.

Tämän opinnäytetyön osan toteutukseen valittiin avoimen lähdekoodin NIDS-ohjelma Suricata. Suricata on helppokäyttöinen, ilmainen ja siinä on versiosta 4.1 eteenpäin sisäänrakennettu tuki JA3-tunnisteille. Suricata voidaan asettaa kuuntelemaan verkkoliikennettä, joka ohjataan sille reitittimeltä tai lukemaan pcap-tiedostoja. Pcap-tiedostot sisältävät tallennettua tietoliikennettä, joita voidaan luoda esimerkiksi Wireshark-ohjelmalla. Hälytykset tallennetaan Elastic-tietokantaan, josta niitä voidaan tarkastella esimerkiksi Kibana-alustaa käyttäen.

4.1 Suricata-signatuurin rakenne

Suricata käyttää samaa signatuuriformaattia kuin Snort. Esimerkissä käytetään ProofPointin Emerging Threats -palvelusta saatua signatuuria. Emerging Threatsin signatuurit ovat varsin kattavia ja pääasiassa saatavilla ilmaiseksi. Palvelusta on myös maksullinen versio, jonka sisältämät signatuurit päivittyvät useammin ja ovat kattavampia. Signatuurin osat on värikoodattu ja koodi on selitetty alle.

```
alert ip $HOME_NET any -> [109.196.130.50,151.13.184.200] any (msg:"ET
CNC Shadowserver Reported CnC Server IP group 1";
reference:url,doc.emergingthreats.net/bin/view/Main/BotCC;
reference:url,www.shadowserver.org; threshold: type limit, track
by_src, seconds 3600, count 1; flowbits:set,ET.Evil;
flowbits:set,ET.BotccIP; classtype:trojan-activity; sid:2404000;
rev:5307; metadata:affected_product Any, attack_target Any, deployment
Perimeter, tag Shadowserver, signature_severity Major, created_at
2012_05_04, updated_at 2019_03_15;)
```

(Emerging Threats 2019)

`alert` Kertoo mitä tapahtumalle tehdään. IDS:n säännöissä yleensä *alert*, IPS:n säännöissä *drop*

`ip` Protokolla. *Ip* arvoa käytetään, kun ei haluta määrittää protokollaa. Arvoksi voidaan asettaa eri kerrosten protokollia, kuten tcp, http ja https.

```
$HOME_NET any -> [109.196.130.50,151.13.184.200] any
```

Liikenteen lähde ($HOME_NET) ja kohde [109.196. ... .200] IP osoitteet. $HOME_NET muuttuja sisältää oletuksena yksityiset IP-avaruudet. Lähde- ja kohde portit (any), yleensä ovat määritetyn protokollan käyttämät portit. Tässä tapauksessa ei ole määritetty tiettyjä portteja. Nuoli merkkaa liikenteen suunnan ja se voi olla joko lähde-kohde (`->`) tai molempisuuntainen (<>).

`msg:` Määrittää tapahtumalle kuvaavan nimen, jota käytetään tunnistamaan tapahtumat SIEM:ssä.

`reference:` Antaa lisätietoa siitä, miksi hälytys luotiin ja mikä siinä on olennaista.

`threshold:` Määrä tapahtumia, joka pitää ylittyä ennen kuin hälytys muodostuu (type: threshold) tai rajoittaa luotujen hälytysten määrää (type: limit). Tyypiksi voidaan asettaa myös *both,* jolloin hälytys tulee kerran asetetun määrän ylittyessä. Thresholdit ovat IP-kohtaisia mutta signatuurissa voidaan määrittää ovatko ne lähde- vai kohde IP-kohtaisia (*track by_src* tai *track by_dst*)

```
flowbits:set,ET.BotccIP
```

Flowbitit ovat muuttujia, joita voidaan käyttää yhdistämään useita signatuureja ja näin voidaan luoda kehittyneempiä signatuureja. Tässä tapauksessa asetetaan *ET.BotccIP*-niminen muuttuja. Tämä mahdollistaa käytöspohjaisen tapahtuma-analyysin ikään kuin if-lauseen avulla.

`sid:` Signatuurin uniikki tunniste.

```
classtype: , rev:,  metadata:
```

Metadataa, kuten luontipäivämäärä, joka kertoo lisää signatuurin tarkoituksesta ja tulkinnasta.

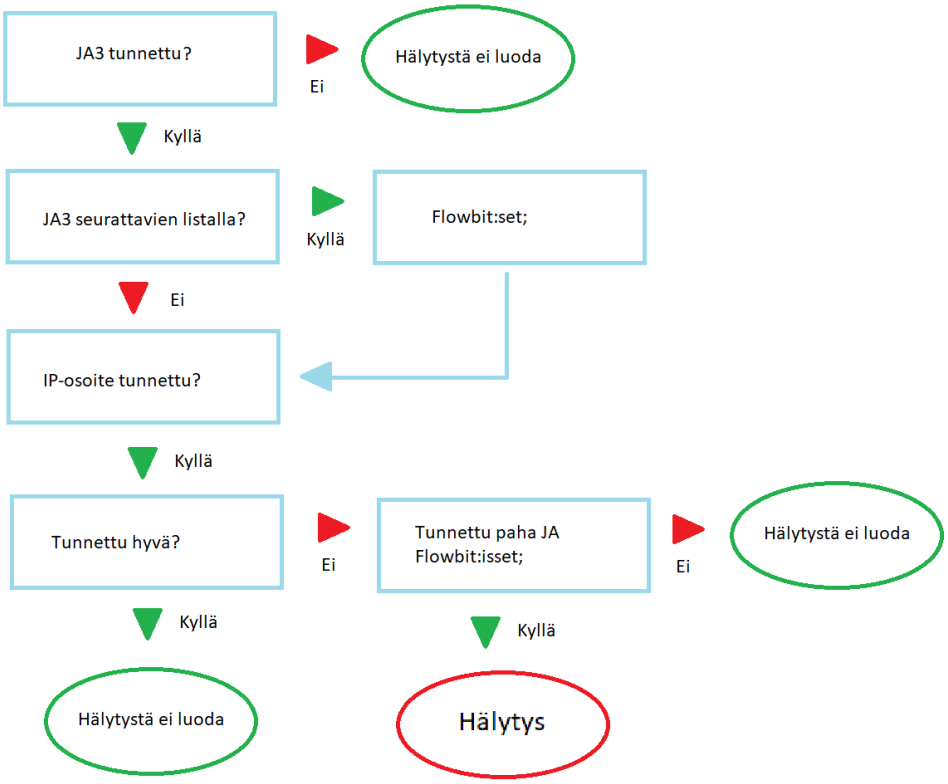
Näiden kenttien lisäksi signatuureille voidaan antaa protokollapohjaisia parametreja, kuten http:n tapauksessa http.host_name, jolloin voidaan viitata http paketin

kohdepalvelimen nimeen. Lista kaikista mahdollisista parametreista löytyy Suricatan dokumentaatiosta https://suricata.readthedocs.io/.

4.2 Signatuurin rakentaminen

Tarkoitus on luoda mahdollisimman luotettava signatuuri, joka ei kuitenkaan laukea turhaan. Tässä osassa muodostetaan ensin esimerkkisignatuuri perustuen aiempaan esimerkkiin ja sen jälkeen signatuuri, jolla voidaan havaita oikean haittaohjelman toimintaa. Tätä signatuuria tullaan myös kokeilemalla imitoimalla haittaohjelman käytöstä virtuaaliympäristössä nauhoittaen samalla liikenne.

Signatuureja rakentaessa on aina muistettava, että on hyvä, että säännöt laukeavat aina kun niihin on aihetta. Mutta jos sääntö laukeaa liian usein, se tukkii SIEM-järjestelmän. Lisäsi jatkuvat väärät hälytykset luovat tilanteen, jossa niitä ei enää oteta tosissaan. Ratkaisu on tehdä kaksi peräkkäistä säännöstöä, jotka käyttävät hyväkseen Suricatan *flowbit*-ominaisuutta. Ensin tarkistetaan, onko liikenteen JA3-tunniste tunnettujen joukossa. Sitten tarkistetaan tunnisteen laatu ja annetaan sen perusteella sen flowbitille arvo. Tämän jälkeen tarkistetaan, onko IP-osoite tai muu tunniste tunnettu, ja mikä sen laatu on. Jos JA3 todetaan epäilyttäväksi, ja IP-osoite on tunnettu haitallinen osoite, luodaan hälytys SIEM:iin. Jos vain toinen ehto täyttyy, hälytystä ei luoda, mutta tieto liikenteestä tallennetaan erilliseen tietokantaan manuaalista analyysia varten.



Kuva 6. Prosessikaavio hälytyksen muodostamisesta

Luodaan runko signatuurille. Sen halutaan havaitsevan *tcp*-liikennettä sisäverkosta ulkoverkkoon.

```
alert tcp $HOME_NET any -> !$HOME_NET any (msg:”Example signature”;)
```

Lisätään JA3-tunniste aiemmasta esimerkistä. JA3_hash on tähän käytettävä parametri.

```
alert tcp $HOME_NET any -> !$HOME_NET any (msg:”Example signature”;
ja3_hash; content:”554719594ba90b02ae410c297c6e50ad”;)
```

Lisätään signatuurille mielivaltainen tunniste, asetetaan flowbit-muuttuja ja muut tarvittavat metatiedot.

```
alert tcp $HOME_NET any -> !$HOME_NET any (msg:”Example signature”;
flowbits: set, chromeJA3; ja3_hash; content:”554719594ba90b02ae
410c297c6e50ad”; classtype:bad-unknown; sid:74760001; rev:1;)
```

Signatuuri lisätään testiympäristöön, jolle sitten annetaan pcap -tiedosto analysoitavaksi. Suricata luo onnistuneesti hälytyksen (kuva 7), josta nähdään signatuurin toimivan.

Kuva 7. Tapahtuma josta hälytys luotiin Kibanassa JSON-formaatissa

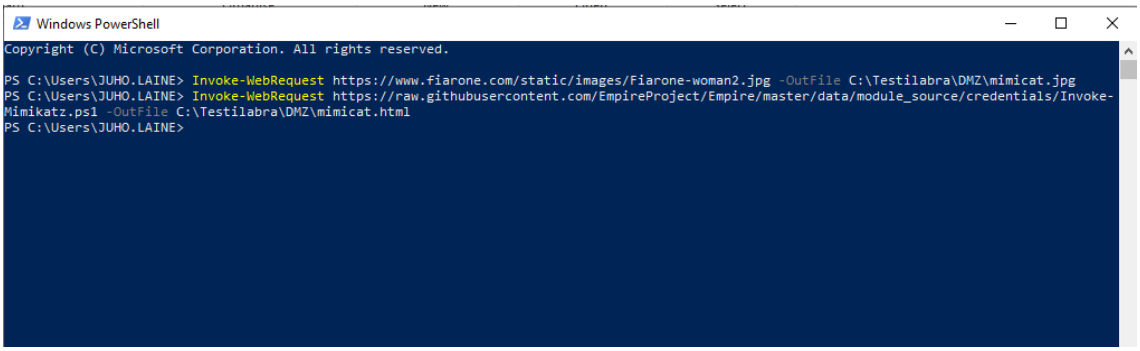
4.3 PowerShell Invoke-WebRequest -yhteydenotto

Seuraavaksi pyritään havaitsemaan hyökkääjien suosiossa olevan työkalun toimintaa. Tähän valitaan Windowsiin oletuksena asennettu skriptaustyökalu PowerShell. Tämä työkalu valitaan, koska se on sekä hyökkääjien että järjestelmänvalvojien suosiossa, ja sen toiminta ei siksi välttämättä poikkea hyökkäyksen yhteydessä. Yleisesti lokitietoihin pohjautuva järjestelmä havaitsee poikkeavan PowerShell-toiminnan huomattavasti tehokkaammin kuin verkkoliikenteeseen pohjautuva. Suurien yritysten tapauksessa ei kuitenkaan ole välttämättä mahdollista kerätä lokitietoja jokaiselta laitteelta. On varsin yleistä, että lokitietoja kerätään ainoastaan Domain Controllereilta ja muilta yrityksen toiminnalle keskeisiltä palvelimilta. Tämä jättää yrityksen työntekijöiden ja muiden yrityksen verkkoon yhteydessä olevien laitteiden lokitiedot järjestelmän ulkopuolelle.

Verkkoliikenteeseen pohjautuva järjestelmä asetetaan usein kuuntelemaan koko verkkoa, tai kaikkea tietyn reitittimen läpi kulkevaa liikennettä. Tämä johtuu siitä, että lokitiedot säilytetään aina kokonaisina, kun taas verkkoliikenteestä tallennetaan ainoastaan ns. kiinnostavat asiat. Tämä vähentää huomattavasti tarvittavaa tallennustilaa. On hyvin epätodennäköistä, että tunkeutuja pääsisi ensimmäiseksi yrityksen palvelimelle. Kuten osassa yksi todettiin, suurin osa haittaohjelmista ja kehittyneistä hyökkäyksistä saavat alkunsa sähköpostin välityksellä. Haitallinen sähköpostin liitetiedosto suoritetaan, jolloin varsinainen haittaohjelma ladataan käyttämällä jotain olemassa olevaa työkalua, esimerkiksi PowerShellia. Tällöin se

voidaan havaita verkkoliikenteestä ja haittaohjelma ehditään poistamaan ennen kuin se aiheuttaa vakavia vahinkoja.

Aloitetaan kopioimalla aiemmin luotu signatuuri ja vaihtamalla JA3-tunniste sellaiseen, jota PowerShellin Invoke-WebRequest käyttää. Tämä saadaan selville tekemällä kaksi kutsua käyttäen kyseistä moduulia ja kuuntelemalla niiden aiheuttamaa liikennettä (Kuva 8). Ensimmäinen kutsu lataa hyökkäystyökalun GitHubista ja toinen harmittoman kuvatiedoston erään turkulaisen yrityksen verkkosivuilta. Testiympäristöön asennettu Suricata muodostaa näistä automaattisesti tunnisteet, joita voidaan käyttää signatuurin luomiseen (Kuva 9).



Kuva 8. Kaksi pyyntöä käyttäen Invoke-WebRequest moduulia

@timestamp per 30 seconds

Time	http.hostname	tls.sni	tls.ja3.hash
March 17th 2019, 05:56:47.808	-	www.fiarone.com	3b5074b1b5d032e5620f69f9f700ff0e
March 17th 2019, 05:48:46.054	-	raw.githubusercontent.com	3b5074b1b5d032e5620f69f9f700ff0e

Kuva 9. Liikenteestä saadut JA3-tunnisteet

Näin saadaan ensimmäinen signatuuri. Tähän lisätään *Flowbit:noalert;* parametri, jolloin se asettaa muuttujan toista signatuuria varten, mutta ei luo hälytystä yksinään.

```
alert tcp $HOME_NET any -> !$HOME_NET any (msg:”Invoke-WebRequest over HTTPS”; flowbits: set, InvokeWebRequest; flowbits:noalert; ja3_hash; content:” 3b5074b1b5d032e5620f69f9f700ff0e”; classtype:bad-unknown; sid:74760002; rev:1;)
```

Lisäksi halutaan havaita mahdollisesti haitallinen yhteys. GitHub ei ole haitallinen sivusto, mutta siellä on jaossa suuri määrä hyökkääjien käyttämiä työkaluja. Voidaan olettaa, että yrityksen työntekijät lataavat työkaluja GitHubista pääasiassa nettiselaimella. Näin muodostetaan signatuuri, joka havaitsee palvelimen nimen perusteella yhteyden GitHubiin ja luo hälytyksen, jos flowbit muuttuja on asetettu.

```
alert tcp $HOME_NET any -> !$HOME_NET any (msg:”Connection to GitHub using Invoke-WebRequest”; flowbits: isset, InvokeWebRequest; tls_sni; content:”githubcontent.com”; nocase; isdataat:!1,relative; classtype:bad-unknown; sid:74760003; rev:1;)
```

Nämä kaksi signatuuria voidaan kirjoittaa myös yhdeksi. Suricatan signatuureissa ei ole TAI-operaattoria, joten skaalautuvuuden kannalta on parempi tehdä kaksi sääntöä. Tämä helpottaa signatuurien ylläpitämistä, muokkaamista ja ymmärtämistä. On helppo lisätä myös toinen sivusto, joka laukaisee oman signatuurinsa.

Toistetaan aiempi koe, jolloin nähdään, laukeaako hälytys halutusta syystä (Kuva 10, Kuva 11).

@timestamp per 30 seconds

Time	tls.ja3.hash	tls.sni
March 18th 2019, 02:19:37.716	3b5074b1b5d032e5620f69f9f700ff0e	raw.githubusercontent.com
March 18th 2019, 02:18:37.633	3b5074b1b5d032e5620f69f9f700ff0e	www.fiarone.com

Kuva 10. Samalla työkalulla saadaan samat tunnisteet

Kuva 11. Ainoastaan yhteys GitHubiin laukaisee hälytyksen

Koe väärien hälytysten löytämiseksi jatkuu ottamalla yhteys samaan githubusercontent.com -osoitteeseen selaimella. Tämä ei aiheuta hälytystä. Kokeillaan ottaa ensin yhteys Invoke-WebRequestilla toiselle sivustolle ja sen jälkeen ladata selaimella tiedosto GitHubista. Tämäkään ei aiheuta hälytystä, joten signatuurin voidaan todeta toimivan halutulla tavalla.

5 POHDINTA JA JATKOKEHITYS

Yrityksen verkko on vilkas ja monimutkainen rakenne, jossa on jatkuvasti käytössä suuri määrä erilaisia ohjelmistoja ja laitteita. Nämä ohjelmat ja laitteet luovat valtavan määrän verkkoliikennettä, josta jatkuvasti suurempi osa on salattua. Liikenteen salaaminen on tärkeä osa tietoturvaa ja yrityksen työntekijöiden tietosuojaa. Se kuitenkin vaikeuttaa myös haitallisen toiminnan havaitsemista. Lisäksi viime vuosina hyökkääjät ovat siirtyneet käyttämään kohteensa laitteella jo olemassa olevia työkaluja, mikä entisestään hankaloittaa uhkien havaitsemista ja torjuntaa.

JA3 on tapa tunnistaa, mikä liikenteen aiheuttaa purkamatta sitä. Sen lisäksi että haittaohjelmien kommunikointi niiden hallintapalvelimille voidaan havaita siitä huolimatta, mihin se on yhteydessä, on myös mahdollista erottaa tavallisten työkalujen epätavallinen toiminta. Käytännön osan esimerkissä luotiin signatuuri, jolla voidaan havaita ulkoisten työkalujen lataaminen saastuneelle laitteelle. Abuse.ch:n kaltaiset palvelut tarjoavat listoja tunnettujen haittaohjelmien JA3-tunnisteista. Nämä voidaan käytännössä ottaa suoraan osaksi Suricata NIDS-järjestelmää, tosin ne voivat aiheuttaa suurenkin määrän vääriä hälytyksiä. Käytännön osan tulosta käyttäen voidaan luoda signatuureja, jotka havaitsevat juuri harmittomien ohjelmien poikkeavaa tai haitallista käytöstä.

Tekoäly ja koneoppiminen ovat aiheita, jotka on viime vuosina saanut paljon huomiota ja kokeneet huomattavaa kehitystä. Nämä tulevat todennäköisesti mullistamaan myös IDS-järjestelmät. Koneoppiminen edistää valtavasti käytökseen perustuvaa analyysia. JA3-tunnisteita on pienenkin yrityksen verkossa satoja. Lisäksi ohjelmistopäivitykset saattavat tehdä muutoksen ohjelman sormenjälkeen. Tekoäly voidaan asettaa oppimaan, millaiset sormenjäljet ovat yhteydessä millaisiinkin osoitteisiin.

Aion jatkaa JA3-sormenjälkiin pohjautuvan analyysin ja signatuurien kehittämistä. Tavoitteenani on luoda signatuurit yleisimmille PowerShell-moduuleille, ja kunnianhimoisempana projektina myös mm. Burp Suite -auditointityökalulle. Tämä tulee vaatimaan suuren määrän sekä omaa tutkimusta että tietoturvayhteisön apua, mutta uskon, että JA3:a hyödyntämällä voidaan tulevaisuudessa liikenteen salauksesta huolimatta havaita uhat luotettavasti ja tarkasti.

LÄHTEET

Alert Logic. 2018. *Blog.* 22. elokuu. https://blog.alertlogic.com/must-know-phishing-statistics-2018.

Althouse, Jeff. 2019. *TLS Fingerprinting with JA3 and JA3S.* helmikuu. https://engineering.salesforce.com/tls-fingerprinting-with-ja3-and-ja3s-247362855967.

Atkinson, Jeff, John B. Althouse, ja Josh Atkins. 2019. *JA3 - A method for profiling SSL/TLS Clients.* tammikuu. https://github.com/salesforce/ja3.

Bayer, Macy. 2018. ”Cybersecurity rundown: The 5 most critical threats to businesses in 2018.” 17. heinäkuu: https://www.techrepublic.com/article/cybersecurity-rundown-the-5-most-critical-threats-to-businesses-in-2018/.

Chandan, Mandal. 2017. *Difference between IDS, IPS, and Firewall? is it possible to configure a firewall as an IDS?* 14. tammikuu. https://www.linkedin.com/pulse/difference-between-ids-ips-firewall-possible-mondal-ceh-jncia-.

Chen, Ping, Lieven Desmet, ja Christophe Huygens. 2014. ”A Study On Advanced Persistent Threats.” *IFIP International Conference on Communications and Multimedia Security.* Berlin: Springer. 63-72.

Emerging Threats. 2019. 13. Maliskuu. http://rules.emergingthreats.net/blockrules/emerging-botcc.suricata.rules.

Faou, Matthieu. 2018. *OceanLotus: New watering hole attack in Southeast Asia.* 22. helmikuu. https://www.welivesecurity.com/2018/11/20/oceanlotus-new-watering-hole-attack-southeast-asia/.

2019. *How Does an Antivirus Program Work to Protect Your Computer?* helmikuu. https://combofix.org/how-does-an-antivirus-program-work.php.

2019. *Investopedia.* helmikuu. https://www.investopedia.com/terms/b/banker-trojan.asp.

2019. *Kaspersky.* helmikuu. https://www.kaspersky.com/resource-center/definitions/what-is-cyber-security.

Quain, John R. 2016. *Do You Really Need to Pay for Antivirus Software?* 30. marraskuu. https://www.tomsguide.com/us/antivirus-software-pay-or-free,news-18570.html.

Symantec. 2018. *Internet Security Threat Report volume 23.* Symantec corp. https://www.symantec.com/content/dam/symantec/docs/reports/istr-23-2018-en.pdf.

2019. *Techtarget.* helmikuu. https://whatis.techtarget.com/definition/Confidentiality-integrity-and-availability-CIA.

Trendmicro. 2017. *Security 101: The Impact of Cryptocurrency-Mining Malware.* 5. Heinäkuu. Haettu 13. Huhtikuu 2019. https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/security-101-the-impact-of-cryptocurrency-mining-malware.