Tampereen yliopisto

Erno Oljakka

KONEOPPIMISEN KÄYTTÖ HAITTAOHJELMAHYÖKKÄYSTEN EHKÄISYSSÄ

Informaatioteknologian ja viestinnän tiedekunta
Kandidaatintutkielma
Kesäkuu 2021

TIIVISTELMÄ

Erno Oljakka: Koneoppimisen käyttö haittaohjelmahyökkäysten ehkäisyssä
Kandidaattitutkielma
Tampereen yliopisto
Tietojenkäsittelytieteiden tutkinto-ohjelma
Kesäkuu 2021

Tutkimuksessa tiivistyy eri menetelmiä haittaohjelmien havaitsemiseen ja analysointiin käyttäen koneoppimismenetelmiä. Tutkimus toteutettiin, sillä haittaohjelman uhriksi voi nykyään joutua lähes kuka vaan ja niiden tuntemus on mielestäni hyödyllistä. Tutkielmassa selviää tärkeimmät piirteet, joita täytyy ottaa huomioon, kun käsitellään haittaohjelmia, käyttäen koneoppimismenetelmiä. Tutkielmassa käsitellään haittaohjelmien havaitsemista useissa eri ympäristöissä ja tilanteissa.

Tutkielmassa esittelen koneoppimismenetelmiä, joilla voidaan analysoida ja havaita haittaohjelmia ja siten ehkäistä haittaohjelmahyökkäyksiä. Tutkielma pohjustetaan alussa käsiteltävillä käsitteillä koneoppiminen, haittaohjelmat ja haittaohjelmahyökkäykset, joiden ymmärtäminen on tärkeää loppututkielmaa luettaessa. Listaan muun muassa yleisimpiä haittaohjelmia ja niiden ominaisuuksia.

Keskeisien käsitteiden esittelyn jälkeen aloitan tarkastelemaan valitsemiani koneoppimismenetelmiä, niistä tehtyjen tutkimusten perusteella. Tämän jälkeen vertailen näitä menetelmiä, sekä tutkin onko niissä samankaltaisuuksia. Koneoppimismenetelmille voidaan tehostaa ja parantaa tarkkuutta haittaohjelmien tunnistamisessa ja havaitsemisessa. Tulososiossa kiteytän tärkeimmät tulokset, jotka löysin tarkastelemalla tutkimuksia luvussa neljä.

Avainsanat: Koneoppiminen, haittaohjelma, haittaohjelmahyökkäys, koneoppimismenetelmät

Sisällysluettelo

1 Johdanto

Koneoppiminen on jo vuosikymmeniä kehittynyt eräs tekoälyn osa-alueista ja sen käyttö teknologia-aloilla lisääntyy joka vuosi. Ongelma, joka kasvaa suuremmaksi vuosi vuodelta on haittaohjelmahyökkäykset (Cook, 2021). Niinpä tämän kandidaatintutkielman aiheena on tutkia, kuinka koneoppimisella voidaan havaita haittaohjelmat ja niillä tehtävät hyökkäykset ajoissa ja näin pienentää hyökkäysten määrää.

Tutkielmassa tarkastellaan koneoppimismenetelmiä, joilla haittaohjelmien havaitseminen eri tilanteissa helpottuu sitä soveltaessa. Menetelmiä ei tarkastella kovin syvällisesti vaan tarkoituksena on käydä läpi vain niiden yleispiirteet.

Kuka tahansa teknologian käyttäjä on tänä päivänä hyötynyt koneoppimisesta, koska sitä käytetään muun muassa auttamaan käyttäjiä jakamaan kuvia kavereistaan sosiaalisen median alustoilla. Suoratoistopalvelut käyttävät *suosittelujärjestelmiä* (Recommendation system) suositellakseen käyttäjille sarjoja ja elokuvia, joista he aiemmin katsottujen sarjojen ja elokuvien perusteella pitäisivät. Näiden suositusjärjestelmien toiminta perustuu koneoppimisalgoritmeihin. (Tagliaferri, 2017)

Barut ja muut (2020) kertovat salatun verkkoliikenteen kasvaneen määrän synnyttäneen mahdollisuuden levittää haittaohjelmia salauksen avulla verkossa. Kryptovaluuttojen yleistyessä kyberrikolliset ovat alkaneet saastuttamaan koneita kryptovaluutan louhijoilla ja näin tekevät tuottoa laittomasti (Mansor et al., 2020). Esineiden internet -laitteet ovat myös haittaohjelmahyökkäysten tähtäimessä niiden laitteiston ja ohjelmistojen suuren monipuolisuuden takia (Tien et al., 2020). Chen ja muut (2017) kertovat, kuinka puhelimien kasvavan käytön takia niihin kohdistuvien haittaohjelmien määrä kasvaa koko ajan ja tämän takia niiden havaitseminen on entistä tärkeämpää. Kaikki edellä mainitut muutokset teknologiassamme ovat luoneet kohteita haittaohjelmahyökkäyksille, joita voidaan ehkäistä huomattavan paljon soveltamalla koneoppimista oikeilla tavoilla. Haittaohjelmahyökkäysten ehkäisyn tarve kasvaa jatkuvasti, koska yritykset ovat nykyään varsinkin kiristyshaittaohjelmien kohteena enemmän kuin ennen. Vuonna 2018 pelkästään Symantec, joka on tietoturvaohjelmia myyvä yritys huomasi 12 %:n kasvun kiristyshaittaohjelmahyökkäyksissä yrityksiä vastaan (Cook, 2021).

Lähestyn aihetta tutkimalla jo aiemmin tehtyjä tutkimuksia koskien haittaohjelmia tunnistusta koneoppimismenetelmien avulla. Näitä tutkimalla pystyn vertailemaan, onko menetelmissä samankaltaisuuksia sekä miten näiden menetelmien olemassaolo

edesauttaa haittaohjelmien havaitsemista ja ehkäisyä. Tältä kannalta asiaa tutkimalla pystyn kiteyttämään pääpiirteet koskien erilaisten haittaohjelmien avulla toteutettuja hyökkäyksiä.

Tutkielman alussa syvennyn tarkemmin käsitteisiin koneoppiminen, haittaohjelmat ja haittaohjelmahyökkäykset, niiden ymmärrys auttaa loppututkielman ymmärtämisessä. Käsiteosion jälkeen siirryn tarkastelemaan valitsemiani tutkimuksia, jotka käsittelevät haittaohjelmien analysointia ja havaitsemista. Luku, jossa käsittelen tutkimuksia, on jaettu viiteen osaan, jossa jokaisessa tarkastellaan erillistä tutkimusta. Seuraavaksi vertaillaan edellisessä osiossa läpikäytyjä tutkimuksia ja tämä luku sisältää myös omaa pohdintaa. Tutkielman lopussa on yhteenveto, jossa käsittelen tutkielmassa esille tulleet asiat ja tulokset.

2 Tutkimusmenetelmä

Tutkimusmenetelmänä tässä työssä on kirjallisuuskatsaus. Tutkimustyö suoritettiin hakemalla aiheeseen liittyviä tieteellisiä julkaisuja, joiden avulla aihetta käsiteltiin ja luotiin kokonaisuus, joka saatiin aikaan vertailemalla lähteiden sisältöjä ja tekemällä näistä johtopäätöksiä.

Tutkimusmateriaali on kerätty käyttäen tietokantoja ACM Digital Library, Andor, Computer Science Database, Google ja Google Scholar. Hakusanoja tarpeeksi rajaamalla löytyi hyvä määrä tuloksia, joista aineisto on kerätty. Hauissa käytin muun muassa hakusanoja kuten ”machine learning”, ”malware attacks” and ”malware analysis”. Tekstit, joita käsittelen valitsin suurimmaksi osaksi lukemalla niiden tiivistelmiä ja johdantoja saadakseni hyvän kuvan sopivatko ne työhöni lähteiksi.

3 Keskeiset käsitteet

Tässä luvussa käydään läpi koneoppimisen, haittaohjelmien ja haittaohjelmahyökkäysten peruskäsitteitä ja niiden peruspiirteitä.

3.1 Koneoppiminen

Koneoppiminen on yksi tekoälyn monista osa-alueista ja on tänä päivänä hyvin laajasti sovellettuna monilla eri aloilla. Termiä koneoppiminen käytti tunnetusti ensimmäistä kertaa Arthur Samuel vuonna 1952, joten ala on ollut olemassa ja kehittynyt jo kauan. Hänen kehittämä *Tammen pelaus ohjelma* (The Samuel Checkers-playing pprogram) oli ensimmäinen koneoppimismenetelmä, joka sai julkisesti huomiota. (Foote, 2019). Koneoppimisen tavoite on yleisesti ymmärtää datan rakennetta ja sijoittaa tämä data malleihin, joita ihminen ymmärtää ja pystyy hyödyntämään (Tagliaferri, 2017). Koneoppimisalgoritmien tarkoitus on oppia automaattisesti niille annettavista syötteistä. Algoritmien opettamisen tavoitteena on saada tietokone tekemään päätöksiä ja ennustuksia ilman, että sitä tarkalleen ohjelmoidaan tekemään sitä. .

Koneoppimismenetelmät voidaan jakaa kolmeen isoimpaan kategoriaan, jotka ovat ohjattu ja ohjaamaton oppiminen sekä vahvistusoppiminen. *Ohjatussa* ja

ohjaamattomassa oppimisessa (supervised learning, unsupervised learning) molemmissa tietokoneelle annetaan syötteitä. Menetelmien ero syntyy siitä, että ohjatussa oppimisessa halutut tulokset ovat tietokoneella tiedossa, toisin kuin ohjaamattomassa. Ohjatussa oppimisessa tarkoitus on, että algoritmi oppii vertaamalla varsinaisia tuloksia opetettuihin ja löytää näin virheitä ja muuntaa mallia paremmaksi Ohjaamattomassa oppimisessa tietokone yrittää löytää samankaltaisuuksia, sille annetusta datasta. (Tagliaferri, 2017)

3.2 Haittaohjelmatyyppejä

Sharp ja muut luokittelevat *Haittaohjelmat* (malware) useaan eri luokkaan riippuen siitä, miten ne tuodaan järjestelmään ja minkälainen hyökkäys niillä on tarkoitus aiheuttaa. Haittaohjelmat usein suunnitellaan niin, että ne tunnistavat, kun niitä ajetaan virtuaalisessa ympäristössä, jonka takia niitä voi olla vaikeampi havaita (McLaughlin et al., 2017). Tämä siis tarkoittaa, että haittaohjelmat on mahdollista luoda sellaisiksi, että ne tunnistavat, kun niitä testataan suljetussa ympäristössä ilman vaaraa testaajille.

Haittaohjelmat ovat usein piilotettuina toisiin ohjelmiin, jotta käyttäjä ei huomaisi haittaohjelman olemassaoloa. Ne voivat olla piilotettuina toisiin ohjelmiin, tekstitiedostoihin, sähköpostiliitteisiin, internet-mainoksiin ja verkkosivuille. Kuitenkin useimmat haittaohjelmista tulevat käyttäjän järjestelmään heidän ladatessaan saastuneen ohjelman epämääräiseltä sivulta tai painettuaan saastunutta linkkiä tai liitettä. (Johansen, 2019) Haittaohjelmat sisältävät useasti samankaltaista rakennetta ja koodia, joten koneoppimisalgoritmeja voidaan opettaa tunnistamaan tällaisia rakenteita.

Madot, virukset ja troijalaiset eroavat niiden leviämistavassa, mutta ovat muuten hyvin samanlaisia niiden vaikutuksilta tietokoneelle. Vaikutuksia voi muun muassa olla tiedostojen tuhoaminen tai muokkaaminen, kovalevyn uudelleenmuotoilu ja hakkereille *takaoven* (back door) luominen, jonka avulla he pääsevät käsiksi tietokoneeseen. Ne voivat myös vain olla järjestelmässä ja hidastaa sitä. (Comtact, 2019)

3.3.1 Madot

Tämä yleinen haittaohjelma yrittää tietokoneen saastuessa levittäytyä useampaan koneeseen lähettämällä itsestään kopioita toisille koneille nettiyhteyden kautta. Madot eivät myöskään tarvitse minkäänlaista toimintaa yhdeltäkään osapuolelta, jotta ne pystyvät leviämään. (Sharp, 2017)

3.3.2 Virukset

Tämä kyseinen haittaohjelma voi levitä todella nopeasti, sillä ne ovat usein hyvin edistyneitä. Viruksen leviämistapana on liittää kopioita itsestään eli viruskoodia muihin tietokoneen tiedostoihin, kun saastuneet tiedostot siirtyvät toiselle koneelle virus pääsee leviämään. (Sharp, 2017)

3.3.3 Troijalaiset

Troijalaiset (Trojan horse) piiloutuu näyttävästi hyödylliseen ohjelmaan. Ohjelman hyödyllinen osa on selvästi käyttäjälle näkyvissä ja tiedossa, mutta haittaohjelma on sen takana piilossa. (Sharp, 2017)

3.3.4 Kiristyshaittaohjelmat

Kiristyshaittaohjelma (Ransomware) on haittaohjelma, joka salaa tiedostoja ja lukitsee tietokoneen kokokaan, kunnes omistaja maksaa hakkereiden pyytämät lunnaat. Lunnaat pyydetään lähes aina bitcoineina, koska niiden siirtoa on vaikea jäljittää. (Sharp, 2017)

3.3.5 Vakoiluohjelmat

Vakoiluohjelma (Spyware) on haittaohjelma, joka lähettää yksityiskohtaisia tietoja salaa käyttäjältä heidän toiminnastansa tai heidän käyttöjärjestelmästänsä ja laitteistosta. (Sharp, 2017)

3.3.6 Botit

Botti on ohjelma, joka suorittaa automaattisesti sille annettuja tehtäviä. Niiden avulla voidaan luoda hyvin hyödyllisiä ohjelmia, mutta usein botteja nähdään myös käytettävän haittaohjelmana. Ne voidaan suunnitella tietokoneen haltuun ottamiseen, jotta hyökkääjä voi käyttää uhrin tietokonetta etänä tai yksityistietojen saamiseksi kuten salasanoja ja pankkitunnuksia. (Bell, 2018)

3.3.7 Cryptojacking

Cryptojacking tämä on uudenlainen haittaohjelma, joka on noussut pinnalle kryptovaluutan yleistymisen ja suosion kasvun takia. Hakkerit saastuttavat koneen, asentavat siihen kryptovaluuttalouhijoita ja käyttävät uhrien varastettuja tietoja luodakseen *kryptolompakoita* (cryptowallet), joilla he voivat suorittaa laittomia varojen siirtoja itselleen (Mansor et al., 2020).

3.3 Haittaohjelmahyökkäykset

Haittaohjelmahyökkäykseksi sanotaan tilannetta, kun kyberrikollinen luo haittaohjelman ja asentaa sen jonkun toisen koneelle tämän tietämättä. Useasti näin tehdään, jotta päästään käsiksi tietokoneeseen, sen verkkoon tai vaurioittamaan näistä toista tai molempia taloudellisen voiton toivossa. (Johansen, 2019)

Hyökkäyksiä kohdistuu niin yksityisiin henkilöihin kuin yrityksiinkin. Kyberrikollisella on monia tapoja yrittää asentaa haittaohjelma uhrin tietokoneelle. Johansen (2019) on muun muassa maininnut hyökkäystavat *haittaohjelma-alusta* (Exploit kit), saastuneet verkkosivut, *pahansuopa mainonta* (Malwertasing), *mies selaimessa -hyökkäys* (Man-in-the-browser, MitB), *väliintulohyökkäys* (Man-in-the-middle, MitM) ja *kalastelusähköpostit* (Phishing email).

Haittaohjelma-alusta on hyvin tehokas haittaohjelmahyökkäystapa, joka etsii haavoittuvuuksia järjestelmästä ja sellaisen löydettyään asentaa haittaohjelman tietokoneelle tätä kautta. Rikolliset voivat käyttää mainosalustoja laittamalla mainosten mukaan haittaohjelmakoodia, joka asentuu tietokoneelle, kun mainos avautuu. Mies selaimessa -hyökkäyksessä ja väliintulohyökkäyksessä hyökkääjä yrittää kaapata viestin uhrilta jollekin palvelimelle, esimerkiksi uhrin ja nettipankin välillä. Nämä kaksi hyökkäystapaa eroavat, koska väliintulohyökkäyksessä hyökkääjän täytyy olla fyysisesti lähellä kohdetta, koska hän kuuntelee viestiliikennettä hakkeroidun reitittimen kautta, kun taas mies selaimessa -hyökkäyksessä haittaohjelma on asennettu uhrin koneelle ja tieto viestiliikenteestä lähetetään etänä hyökkääjälle. Kalastelusähköposteissa hyökkääjä yleisesti yrittää huijata toista syöttämään tunnuksiaan valesivustolle, jonka kautta hyökkääjä saa tunnukset käyttöönsä. (Johansen, 2019)

Haittaohjelmahyökkäykset muuttavat muotoaan ja viime vuosina on näkynyt selvää muutosta. Artikkelissa esitellään dataa Googlelta, jonka mukaan haittaohjelmia

sisältävien verkkosivujen määrä vähenee joka vuosi. Tammikuussa 2021 Google havaitsi 600–800 haittaohjelman tartuttamaa sivustoa, kun taas tammikuun ja maaliskuun välillä vuonna 2019 havaittiin yli 3000 saastunutta sivustoa viikossa ja samalla aikavälillä vuonna 2018 5000–7000. Tästä voidaan päätellä, että teknologiat haittaohjelmien torjumiseksi ovat kehittyneet mutta myös sen, että rikolliset siirtyvät tekemään enemmän kohdennettuja hyökkäyksiä. Kalasteluhyökkäykset ovat tällä hetkellä hyvin suosittuja. Varsinkin nettipankit ja sosiaalisen median alustat ovat näiden hyökkäyksien kohteena (Cook, 2021).

4 Koneoppimismenetelmiä haittaohjelmien havaitsemiseksi

Luvussa käsitellään viittä tutkimusta koskien koneoppimismenetelmiä, joiden avulla voidaan havaita ja analysoida haittaohjelmia. Ucci ja muut (2019) löysivät kolme tärkeintä tavoitetta haittaohjelmien tutkimisessa. Ensimmäinen ja yleisin näistä on, tunnistaa varmasti, onko tarkasteluun annettu otos *haitallinen* (malicious). Toinen on löytää ja tunnistaa samankaltaisuuksia eri haittaohjelmien välillä ja kolmas on kategorisoida haittaohjelmia eri kategorioihin, joiden avulla tunnistaminen sujuu sulavammin tulevaisuudessa.

Luvun tarkoituksena on käydä tutkimuksissa käytettyjä koneoppimismenetelmiä pintapuolisesti läpi.

4.1 Haittaohjelmien havaitseminen salatussa verkkoliikenteessä

Kun verkkoon yhdistettyjen laitteiden määrä kasvaa, niin haitalliset verkkoaktiviteetit generoivat suunnattoman paljon dataa, joka vaatii turvallisuus ja yksityisyysanalysointia lähes reaaliajassa. Kasvavat verkkoyhteysnopeudet ja monimutkaisemmat haittaohjelmat ovat vain kaksi tekijää, jotka pahentavat tätä tilannetta. Kesäkuussa 2020 noin 95 % Googlen verkkoliikenteestä oli salattua, kun kuusi vuotta aiemmin luku oli vain 55 %. Salatun verkkoliikenteen kasvu on kasvattanut mahdollisuutta käyttää *TLS-salausta* (Transport Layer Security) haittaohjelmien levitykseen verkossa. (Barut et al., 2020). Tässä osiossa tarkastellaan, miten on mahdollista tunnistaa haittaohjelmia TLS-salatusta verkkoliikenteestä ja miten sitä voidaan tehostaa hyödyntämällä koneoppimista.

Barut ja muut (2020) keräävät kaksi tietoaineistoa, joiden avulla on tarkoitus kehittää koneoppimisalgoritmia. Tietoaineistot sisältävät salatusti siirrettyä tietoa kuten myös tunnettuja haittaohjelmia ja näiden haittaohjelmien avulla algoritmit oppivat tunnistamaan itsenäisesti samanlaisia ohjelmia niiden rakenteen ja ominaisuuksien perusteella. Tietoaineistojen keräämisen jälkeen on käytössä Cisco Joy-niminen ominaisuuksienpoimija, jota käytetään löytämään ominaispiirteitä, niin siirretylle datalle kuin haittaohjelmille (Barut et al., 2020). Barut kertoo, että TLS-salatut haittaohjelmat tunnistetaan nimenomaan löytämällä niiden oikeat ominaispiirteet, joilla ne voidaan tunnistaa.

Koneoppimisessa käytetään montaa eri luokittelumenetelmää, mutta Barut ja muut (2020) valitsivat tutkimuksen suorittamiseen neljä menetelmää, jotka ovat logistinen regressio, *satunnaismetsä* (random forest), tukivektorikone (support vector machine, SVM) ja *lähimmän naapurin hakua* (Nearest Neighbour). Testatessaan, mitkä näistä menetelmistä toivat parhaimmat tulokset, saatiin tulos, että SVM ja satunnaismetsä-menetelmät toimivat parhaimmalla tarkkuudella.

4.2 Ajonaikainen haittaohjelmien havaitseminen

Jatkuvasti lisääntyvät monimutkaiset modernit laskentajärjestelmät ovat johtaneet lisääntyneeseen turvallisuusriskien määrään, joka tekee järjestelmistä haavoittuvia hienostuneille kyberhyökkäyksille. Näitä hyökkäyksiä voidaan suorittaa piilottamalla haittaohjelmakoodi *hyvänlaatuisen ohjelman* (Benign application) koodiin ja näin haittaohjelmaa on vaikea havaita käyttämällä perinteisiä haittaohjelmien tunnistus-menetelmiä. Näin piilotettuja haittaohjelmia kutsutaan troijalaisiksi. (Sayadi et al., 2020).

Tietokoneen laitteistoavusteista haittaohjelma tunnistusta (HMD, Hardware-Assisted Malware Detection) voidaan käyttää tunnistamaan haittaohjelmia, jotka on piilotettu hyvänlaatuisiin ohjelmiin. Olemassa olevat HMD-menetelmät ovat limitetty tunnistamaan haittaohjelmamalleja, jotka ilmestyvät erilliseen säikeeseen ohjelman suorituksen aikana, näin ollen piilotettujen haittaohjelmien ajonaikainen havaitseminen on edelleen haaste (Sayadi ym., 2020).

Tässä osiossa tarkastellaan StealthMiner-nimistä koneoppimismenetelmää, jolla voidaan tunnistaa haittaohjelmia ajonaikaisesti. Piilevien haittaohjelmien päätarkoitus on pysyä huomaamattomana laitteen järjestelmässä mahdollisimman kauan, jotta

haittaohjelmalla on mahdollisimman kauan aikaa vaarantaa tietokone tai varastaa tietoja, ennen kuin sopiva havaintomenetelmä saadaan asennettua turvaamaan tieto-kone. Tämän kaltaisia haittaohjelmia kutsutaan myös *sulaututeiksi haittaohjelmiksi* (Embedded malware). Koska sulautetut haittaohjelmat eivät luo omaa säiettään ohjelman suorituksen aikana, on niiden havaitseminen hankalaa. (Sayadi et al., 2020)

StealthMiner-menetelmän toiminta perustuu ohjelman suorituksen aikana kymmenen millisekunnin intervallin tutkimiseen. Tätä intervallia tutkitaan neuroverkkoihin perustuvan luokittelun avulla, joka suodattaa tietoa lävitseen ja opetetut koneoppimisalgoritmit pystyvät tunnistamaan piilotetut haittaohjelmat. Tavoitteena on tunnistaa saastuneet intervallit ja hyödyntää niitä haittaohjelmien tunnistamiseen hyvänlaatuisen ohjelman sisässä. (Sayadi et al., 2020) Sayadin ja muiden (2020) tehdyssä tutkimuksessa heidän testauksillansa StealthMiner-menetelmällä he pystyivät tunnistamaan piilotetut haittaohjelmat 94 %:n tarkkuudella.

4.3 Android-laitteisiin kohdistuvat haittaohjelmahyökkäykset

Haittaohjelmien havaitsemisesta on syntynyt ongelma erityisesti mobiilialustoilla. Huomioiden mobiililaitteiden lisääntyvän määrän ja niihin liittyvät sovelluskaupat, uusien sovellusten volyymi on liian suuri niiden manuaaliseen tarkasteluun haitallisten piirteiden varalta. Havaitseminen on ennen perustunut koodin manuaaliseen tarkasteluun, mutta tämä ei toimi enää, sillä haittaohjelmat muuttuvat liian nopeasti. (McLaughlin et al., 2017)

Symantecin internetin turvallisuusuhka raportin mukaan vuonna 2017 joka viides Android-ohjelmista on haittaohjelma (Chen et al., 2017). Tämä aiheuttaa suuria uhkia puhelimien käyttäjille, kuten yksityistietojen varastamista, automaattisia soittoja maksullisiin numeroihin ja viestien lähettämiseen ilman lupaa. Seurauksena Android-haittaohjelmien havaitseminen on tärkeää niin haittaohjelmien vastaisille aloille kuten myös tutkijoille. (Chen et al., 2017). Android-laitteisiin kohdistuvat haittaohjelmat on tärkeää saada kitkettyä pois, sillä ne koskettavat monia ympäri maailmaa.

Chen ja muut (2017) kertovat Androidin käyttävän koulutettua luokittelumallia haittaohjelmien tunnistamiseen ja estääkseen niitä sekaantumasta käyttäjien puhelimien toimintaa. Chen ja muut (2017) tutkivat Android-laitteisiin kohdistuvia *muutos-hyökkäyksiä* (Adversial attack), joissa hyökkääjä käsittelee hyvänlaatuisen sovelluksen ominaisuuksia ja lisää sinne huomaamattomasti haitallisia ominaisuuksia. Esimerkiksi

hyökkääjät voivat lisätä manifest-tiedostoon oikeuksia ilman, että vaikuttavat sovelluksen toimintaan yhtään. Toinen mahdollisuus, miten hyökkääjät voivat muuttaa sovellusta on piilottamalla tai poistamalla ominaisuuksia ilman vaikutusta toimintaan, jonka he haluavat suorittaa haittaohjelmalla (Chen et al., 2017). Tutkimuksessa käytetään SecCLS-menetelmää, joka on ominaisuuksien valintamenetelmä ja sillä tässä tapauksessa rakennetaan luokittelua koneoppimisalgo-ritmia varten, jonka tarkoituksena on tasoittaa jokaisen ominaisuuden tärkeystaso ja rakentaa turvallisempi *luokitin* (Classifier). SecCLS-menetelmä vähentää mahdollisuutta valita koneoppimismallin rakentamiseen ominaisuuksia, joita hyökkääjät usein manipuloivat ja näin pakottavat hyökkääjät manipuloimaan useampia ominaisuuksia. Tehdessään näin hyökkäys on helpompi huomata, sillä useampaa ominaisuutta on muutettu. (Chen et al., 2017)

SecENS-menetelmää on käytetty parantamaan havaitsemistarkkuutta. Sen toiminta perustuu useiden SecCLS-menetelmällä rakennettujen luokittimien liittämistä yhteen. Yksittäisien luokittimien tulokset yhdistettäessä saadaan tuotettua tarkemmat päätökset, kuin yksittäisillä luokittimilla.

4.4 Esineiden internetiin kohdistuvien haittaohjelmien analysointi

Vuonna 2018 maailmassa oli noin 22 miljardia esineiden internet -laitetta (IoT) ja vuoteen 2025 mennessä niitä arvioidaan olevan 38.6 miljardia (Statista, 2021). IoT-haittaohjelmien uhka suurenee, sillä ne suurentuvalla tahdilla uhkaavat Internetin infrastruktuuria ja monia esineiden internetin sovelluksia riskiympäristöissä kuten terveys- ja turvallisuusaloilla. Monet IoT-laitteet käyttävät vanhentunutta käyttöjärjestelmää, joka ei tarjoa parasta viruksentorjuntaohjelmistoja verrattuna Windows-malliin, jotka jatkuvasti pakottavat päivityksiä tietokoneelle Osiossa käsitellään IoT-haittaohjelmien analysointia ELF- ja OpCode-toiminnoilla. ELF on tiedostomuoto ja OpCode on ohjeistuskoodi tietokoneelle, joka kertoo sille, mitä tehdä. (Tien et al., 2020).

Päätoimintona haittaohjelmien tunnistamiseen ja luokitteluun käytettiin OpCode-toimintoa, kirjoittavat Tien ja muut (2021), sillä sen tarkkuudesta on ollut jo näyttöä muissakin tutkimuksissa. Tutkimuksen tarkoituksena on tunnistaa uudesta näytteestä, onko se pahanlaatuinen vai ei ja luokitella se oikeaan haittaohjelmaperheeseen sen tunnusmerkkien perusteella (Tien et al., 2021).

Tien ja muut (2021) keräsivät 30000 IoT-haittaohjelma näytettä ja seuraavaksi näytteistä kerättiin 7 staattista ominaisuutta käyttäen tiettyjä preferenssejä ja lopuksi koneoppimismenetelmiä kehitettiin kahdellatoista OpCode-pääominaisuudella. He tar-

kastelivat, kuinka hyvä havaitsemis- ja luokittelutarkkuus saadaan ilman ELF- ja OpCode-ominaisuuksia ja niiden kanssa.

Tutkimuksessa tultiin tulokseen, että ELF- ja OpCode-ominaisuuksien avulla voidaan tehokkaasti havaita ja luokitella tuntemattomia IoT-haittaohjelmia. Tutkimuksessa käytettiin edellä mainittuja ominaisuuksia ja saatiin 97 % F-tulos ja 98 % havaitsemis- ja luokittelutarkkuus. (Tien et al., 2021)

4.5 Cryptojacking-haittaohjelmien analysointi

Cryptojacking-hyökkäyksessä kyberrikollinen saastuttaa tietokoneen ja asentaa siihen kryptolouhijoita ja käyttää varastettuja yksityistietoja siirtääkseen rahoja laittomasti. Varsinkin suurin osa esineiden internet -laitteista ovat vaarassa näille hyökkäyksille niiden heikkojen salasanojen takia. Suurin haittavaikutus Cryptojacking-haittaohjelmissa on se, että se vie suuren osan tietokoneen laskentatehosta. Cryptojacking-hyökkäykset nousivat räjähdysmäisesti vuonna 2017 kryptovaluuttojen suosion noustua. Crypto-jacking-haittaohjelmat ovat rikollisille erittäin tuottavia, sillä niitä on vaikea huomata, koska ne toimivat salassa ja niiden toimintaan ei vaadita käyttäjältä mitään toimintaa. (Mansor et al., 2020) Tämän seurauksena tietokoneelle ei pysty tekemään lähes mitään ennen kuin haittaohjelma on saatu poistettua koneelta. Tässä osiossa tarkastellaan tutkimusta, jossa on tutkittu, miten koneoppimisen avulla voidaan luokitella ja analysoida erilaisia Cryptojacking-haittaohjelmia.

Mansorin ja muiden (2020) mukaan Cryptojacking-haittaohjelmat takaavat varman tuoton ja vaativat minimaalista kontaktia uhrien kanssa toisin kuin muut haitta-ohjelmat kuten kiristyshaittaohjelma. Tämän he perustelevat siten, että Cryptojacking-hyökkäykseen vaaditan vain pätkä JavaScript-koodia, jotta se toimii.

Kuten yleensä koneoppimisalgoritmeilla, niin tässäkin tapauksessa Crypto-jacking-haittaohjelmien analysointi perustuu tuntemaan, niiden koodin ominaispiir-teitä. Haasteena on löytää oikeat ominaisuudet, joilla haittaohjelmien tunnistamisen tarkkuus on mahdollisimman korkea (Mansor et al., 2020).

Alun perin ominaisuuksia, joiden avulla algoritmeja kehitetään, oli 56 mutta käyttämällä *ylimääräiset puut luokittelua* (Extra Trees Classifier) saatiin poimittua 13 oleellisinta ominaisuutta. Ylimääräiset puut luokittelumenetelmä on opetusalgoritmi, jonka toiminta perustuu päätöspuihin ja se on lisäksi soveltuva *ominaisuusvalintaan* (feature selection). (Mansor et al., 2020)

Näitä kolmeatoista ominaisuutta hyödynnettiin kahden koneoppimisalgoritmin opettamiseen, jotka ovat satunnaismetsä ja Gradient boost -algoritmi. Satunnaismetsä tuotti paremmat tulokset haittaohjelmien tunnistamisessa.

5 Tulokset

Edellisessä luvussa käsiteltiin viittä eri tutkimusta, jotka kaikki liittyivät haittaohjelmien havaitsemiseen, analysointiin tai luokitteluun ja jokaisessa tutkimuksessa käytettyjä menetelmiä on tehostettu koneoppimismenetelmillä. Tutkimuksien tavoitteena oli löytää sopiva menetelmä tietynlaisten haittaohjelmien tunnistamiseen tietyssä ympäristössä, jonka avulla saadaan mahdollisimman korkea haittaohjelmien tunnistustarkkuus. Useissa tarkastelluissa tutkimuksissa käytetään samoja luokittelu- ja optimointimenetelmiä kuten satunnaismetsää ja tukivektorikonetta. Toinen samankaltaisuus, minkä huomaa tutkimuksista on se, että oikeiden piirteiden valinta, algoritmin kehittämiseksi on erittäin tärkeää, kun koneoppimisalgoritmeja opetetaan tunnistamaan haittaohjelmia itsenäisesti. Piirteillä tarkoitetaan haittaohjelmien ominaispiirteitä kuten koodin rakennetta, joiden avulla koneoppimisalgoritmit oppivat tunnistamaa niitä ja niiden tärkeyttä painotetaan jokaisessa tutkimuksessa. Oikeiden ominaisuuksien löydyttyä pystytään rakentamaan mahdollisimman hyvä ratkaisu, joka toimii hyvällä tarkkuudella kaikissa tapauksissa.

Koneoppimisen käytön avulla saadaan tehostettua prosesseja, joita tutkimuksissa tehdään. Hyvien koneoppimismallien käytön avulla pystytään tunnistamaan haittaohjelmat nopeasti ja toimimaan haittaohjelmatyypin mukaisesti ja estämään mahdollinen hyökkäys tai neutralisoimaan se mahdollisimman nopeasti. Tähän ei usein tarvita edes ihmisen läsnäoloa. Haittaohjelmien havaitsemisen tarve isommassa mittakaavassa tulee kasvamaan tulevaisuudessa, koska laitteiden määrä joihin haittaohjelmia piilotetaan, lisääntyy. Esimerkiksi IoT-laitteet ja Android-laitteet lisääntyvät päivittäin kuten Tien (2021) ja Chen (2017) kertovat tutkimuksissaan.

6 Yhteenveto

Tutkimus toteutettiin kirjallisuuskatsauksena ja siinä käsiteltiin tarkasti erilaisia haittaohjelmatyyppejä, haittaohjelmahyökkäyksiä ja koneoppimisen soveltamista haittaohjelmien tunnistuksen ja havaitsemisen parantamiseen. Kolmannessa luvussa käytiin läpi

tutkielman ymmärtämisen kannalta olennaisia käsitteitä. Neljännessä luvussa käsiteltiin tutkimuksia, joissa tutkittiin, kuinka koneoppimismenetelmillä pystytään tehostamaan haittaohjelmien analysointi, luokittelu- ja havaitsemismenetelmiä. Tarkasteltujen tutkimuksien rakenne on hyvin samanlainen. Jokaisessa tutkimuksessa painotetaan oikeiden haittaohjelmien ominaispiirteiden valintaa koneoppimisalgoritmin kehitykseen, kun oikeita piirteitä hyödynnetään, paranee haittaohjelmien havaitsemistarkkuus tutkimusten tarjoamilla menetelmillä.

Lähteet

Barut, O., Grohotolski M., DiLeo C., Luo Y., Li P. & Zhang T. (2020). Machine Learn ing Based Malware Detection on Encrypted Traffic: A Comprehensive perfor mance study. *7th International Conference on Networking, Systems and Secu rity*. 45-55. https://doi.org/10.1145/3428363.3428365

Bell, S. 2018. *Malware bots and the nasty things they can do*. Haettu 14.3.2021. https://www.bullguard.com/blog

Chen, L., Hou S. & Ye Y. (2017). SecureDroid: Enhancing Security of Machine Learn ing-based Detection against Adversarial Android Malware Attacks. *Proceed ings of the 2020 8th International Conference on Communications and Broad band Networking.* 362-372. https://doi.org/10.1145/3134600.3134636

Contact. (2019). *What are the different types of Malware?* Haettu 8.3.2021 https://comtact.co.uk/blog/

Cook, S. (2021). *Malware statistics and facts for 2021.* Comparitech. https://www.comparitech.com/antivirus/malware-statistics-facts/

Foote, K. (2019*). A Brief History of Machine Learning*. Dataversity. https://www.dataversity.net/a-brief-history-of-machine-learning/

Johansen, A. (2019). *Malware attacks: What you need to know.* NortonLifeLock. Haettu 15.3.2021. https://us.norton.com/internetsecurity

Mansor, W., Ahmad, A., Zainudin, W., Saudi, M. & Kama, M. (2020). Cryptojacking Classification based on Machine Learning Algorithm. *Proceedings of the 2020 8th International Conference on Communications and Broadband* Networking. 73–76. https://doi.org/10.1145/3390525.3390537

McLaughlin, N., del Rincon, J., Kang, B., Yerima, S., Miller, P., Sezer, S., Safaei, Y., Trickel, E., Zhao, Z., Doupé, A. & Ahn, G. (2017). Deep Android Malware De tec-tion. *Proceedings of the Seventh ACM on Conference on Data and Applica tion, Security and Privacy*. 301–308. https://doi.org/10.1145/3029806.3029823

Statista. (2021) Number of Internet of things (IoT) connected devices worldwide in 2018, 2025 and 2030. Statista. Haettu 18.3.2021. https://www.statista.com/statis tics/802690/worldwide-connected-devices-by-access-technology/

Sayadi H., Gao Y., Makrani H., Mohsenin T., Sasan A., Rafatirad S., Lin J. & Ho mayoun H. (2020). StealthMiner: Specialized Time Series Machine Learning for Run-Time Stealthy Malware Detection based on Microarchitectural Fea tures. *Proceedings of the 2020 on Great Lakes Symposium on VLSI.* 175-180. https://dl.acm.org/doi/10.1145/3386263.3407585

Sharp, R. (2017). *An Introduction to Malware.* https://orbit.dtu.dk/en/publications/an-introduction-to-malware-2

Tagliaferri, L. (2017). *An Introduction to Machine Learning.* DigitalOcean. Haettu 2.2.2021. https://www.digitalocean.com/community/tutorials/

Tien C., Chen S., Ban T. & Kuo S. (2020). *Machine Learning Framework to Analyze IoT Malware Using ELF and Opcode Features.* Digital Threats: Research and Practice. https://dl.acm.org/doi/10.1145/3378448

Ucci, D., Aniello, L. & Baldoni, R. (2019). Survey of machine learning techniques for malware analysis. *Computers & Security. 88.* 123–147. https://www.sciencedirect.com/science/article/pii/S0167404818303808