Jani Anttilainen

Tunkeutumisen laajentamisen havaitseminen koneoppimisella

Tietotekniikan kandidaatintutkielma

21. tammikuuta 2024

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Jani Anttilainen

Yhteystiedot: jani.jk.anttilainen@student.jyu.fi

Ohjaaja: Timo Tiihonen

Työn nimi: Tunkeutumisen laajentamisen havaitseminen koneoppimisella

Title in English: Detecting malicious lateral movement with machine learning

Työ: Kandidaatintutkielma

Sivumäärä: 24+0

Tiivistelmä: Tunkeutumisen laajentaminen on kyberhyökkäyksen vaihe, jossa hyökkääjä laajentaa saamaansa jalansijaansa kohdeympäristössä hankkimalla haltuunsa lisää käyttäjätunnuksia ja koneita. Tämä tutkielma määrittelee mitä tunkeutumisen laajentaminen on sekä esittelee koneoppimisen menetelmiä, joilla sitä voidaan havaita. Parhaimmillaan koneoppimisen luokittelijoilla pystytään tunnistamaan tunkeutumisen laajentaminen lähes aina.

Avainsanat: tunkeutumisen laajentaminen, koneoppiminen, APT

Abstract: Malicious lateral movement is a phase in a cyber attack where the attacker extends their foothold in the target environment by acquiring additional user credentials and machines. This thesis defines what malicious lateral movement is and explores methods for detecting it with machine learning techniques. Classifier models based on machine learning exhibit the potential to identify lateral movement consistently.

Keywords: lateral movement, machine learning, APT

Kuviot

Taulukot

Sisältö

1 Johdanto

Kyberhyökkäykset kehittyvät jatkuvasti nopeammiksi, edistyksellisemmiksi sekä ne aiheuttavat enemmän vahinkoa uhreiksi joutuvissa organisaatiossa ja yrityksissä. Valtioiden tukemat sekä osa yksityisistä hakkerointiryhmistä pystyvät kohdentamaan hyökkäyksensä kohteisiin, joiden tiedoilla voidaan saavuttaa merkittävää taloudellista sekä poliittista hyötyä. Kohdennettuja hyökkäyksiä (*Advanced Persistent Threat*) tekeviä ryhmiä kutsutaan myös APT-ryhmiksi.

Kohdennetuista hyökkäyksistä on tullut osa valtioiden kybervakoilua ja APT-ryhmien tekemät hienostuneet hyökkäykset muodostavat vakavan uhan valtioille, organisaatioille ja yrityksille. Onnistunut hyökkäys voi häiritä yhtiön toimintaa useita viikkoja tai lopettaa liiketoiminnan kokonaan. Hakkerointiryhmien parempien resurssien myötä pääsy kohdeympäristöön on entistä todennäköisempää, esimerkiksi kohdennetun tietojenkalastelun avulla. Koska ei voida olettaa hyökkäyksen jäävän kiinni ensimmäisessä pistessää, tietoturvavalvontaa tarvitaan myös sisäverkoissa tapahtuvan tunkeutumisen laajentamisen (*lateral movement*) havaitsemiseksi.

Tässä tutkielmassa keskitytään kohdennetuissa hyökkäyksissä useasti käytettyyn tunkeutumisen laajentamiseen. Lisäksi esitellään kuinka koneoppimista voidaan käyttää sen havaitsemisessa hyödyksi. Koneoppimisen toivotaan tuovan hyötyä valtavan tietomäärän käsittelyyn, jota nykyiset tietojärjestelmät tuottavat, sekä tapoja uusien ja hienostuneimpien hyökkäysketjujen havaitsemiseksi (Tyufu 2011).

Tutkielman luvussa 2 määritellään tunkeutumisen laajentaminen ja kuvataan yleisempiä tapoja, miten sitä voidaan hyödyntää osana kyberhyökkäystä. Luvussa 3 määritellään koneoppiminen ja kuinka sitä on mahdollista hyödyntää luokittelussa ja tietoaineistojen käsittelyssä. Lopuksi luku 4 esittelee kuinka aiemmissa tutkimuksissa koneoppimista on pystytty hyödyntämään tunkeutumisen laajentamisen havaitsemisessa.

2 Tunkeutumisen laajentaminen

Tunkeutumisen laajentaminen (*lateral movement*) tarkoittaa hyökkääjän liikkumista ja toimintaa kohteen sisäverkossa pyrkien kasvattamaan jalansijaa ympäristössä. Käsite sisältää useasti verkon kohteiden kartoittamisen (*discovery*), tunnusten haalimisen eri menetelmin sekä liikkumisen eri palvelimien ja työasemien välillä. Joskus mukaan luetaan myös arvokkaan tiedon kerääminen pelkän kartoittamisen lisäksi (Chen, Desmet ja Huygens 2014). MITRE ATT&CK® (2021) -viitekehys erottelee tiedon kartoittamisen ja tunnusten haalimisen omiksi taktiikoiksi ja sisällyttää tunkeutumisen laajentamiseen vain tekniikat, joiden avulla liikutaan kohdeympäristössä. Taulukko 1 esittelee MITRE:n erottamat tunkeutumisen laajentamisen tekniikat ja alitekniikat. Tässä kappaleessa määritellään tunkeutumisen laajentaminen käsitteenä ja mitä siihen liittyviä jälkiä on mahdollista käyttää sen tunnistamiseen.

Ilmiöstä on hyvin vähän suomenkielistä materiaalia saatavilla, mistä syystä sille ei ole vakiintunutta suomennosta. Käytetyin termi on tunkeutumisen laajentaminen (Kyberturvallisuuskeskus 2021), jonka lisäksi voidaan puhua lateraalisesta liikkumisesta. Jälkimmäisen termin voidaan kuitenkin tulkita sisältävän hyökkääjän liikkumisen lisäksi myös normaalit ylläpidolliset hallintayhteydet koneiden välillä.

2.1 Tunkeutumisen laajentamista hyödyntävä hyökkäys

Kohdennetuissa kyberhyökkäyksissä tunkeutumisen laajentaminen mahdollistaa, että hyökkääjä laajentaa jalansijaansa ympäristössä. Tämä mahdollistaa hyökkäyksen jatkumisen ja etenemisen, vaikka se havaittaisiin osittain sekä hyökkäyksen etenemisen. Tianin ym. (2019) mukaan tunkeutumisen laajentamista esiintyy useasti kyberhyökkäyksessä, varsinkin kohdeympäristön ollessa suurempi. Tunnusten lisäksi hyökkääjä voi päästä käsiksi yrityssalaisuuksiin sekä ympäristön käyttäjien henkilökohtaisiin tietoihin, joilla voi olla arvoa vakoilun tai kiristyksen kannalta. Yua ym. (2019) nostavat esille, että hyökkäysten tavoitteena on poliittinen tai taloudellinen hyöty.

2.2 Vaiheet

Chen, Desmet ja Huygens (2014) erottelevat tunkeutumisen laajentamisesta kolme tunnistettavaa vaihetta. Ensimmäisessä vaiheessa hyökättävään kohteeseen on jo päästy sisälle esimerkiksi tietojenkalastelulla ja aloitetaan sisäverkon kartoittaminen kuviossa 1 ensimmäinen haltuun satu kone on k_1. Tämän jälkeen tarkoituksena on löytää käyttäjätunnuksia laajemmilla oikeuksilla sekä verkkotason avauksia seuraaviin järjestelmiin. Tunnuksia voi löytyä esimerkiksi Active Directory -toimialueen ohjauskoneiden (*Domain Controller*) kautta. Peruskäyttäjää laajempia oikeuksia omaavia tunnuksia voidaan myös löytää, jos hallittuun koneeseen on otettu yhteyttä IT-tuen työntekijöiden toimesta. Tietoa ympäristöstä saadaan hallussa olevan koneen tiedoista sekä käyttöjärjestelmien omilla työkaluilla, kuten Windowsin Net- ja Get-ADGroup -komennoilla. Active Directory -toimialueen kartoittamiseen on saatavilla myös valmiita työkaluja. Ussath ym. (2016) mukaan meluisia ja helposti havaittavia keinoja kartoittamiseen, kuten porttiskannausta, vältetään.

Toisessa vaiheessa Chen, Desmet ja Huygens (2014) mukaan hyökkäyksessä pyritään saamaan haltuun lisää käyttäjätunnuksia sekä koneita. Kohteiksi valikoituu ensimmäisessä vaiheessa priorisoituneet käyttäjät ja järjestelmät, joilla on eniten oikeuksia, kuten ylläpitotunnukset. Näiden avulla hyökkääjän on helpompi pysyä ja toimia verkkoalueessa. Hyökkääjä voi käyttää esimerkiksi Mimikatz-työkalua (Delpy 2007), jolla on mahdollista varastaa Windows-koneen keskusmuistista salasanojen tiivistesummat sisältävät muistialueet. Tiivistesummista on mahdollista murtaa selkokielisiä salasanoja tai tiivistesummia voidaan itsessään käyttää autentikoinnissa. Haltuun saaduilla peruskäyttäjän tunnuksilla voidaan myös yrittää saada mahdollisilla hallintapalvelimilla enemmän oikeuksia käyttöoikeuksien eskaloinnilla (*privilege escalation*) hyödyntäen järjestelmien haavoittuvuuksia. Kun tunnukset on saatu haltuun, haitallisen käytön tunnistaminen vaikeutuu, koska hyökkääjät hyväksikäyttävät normaaleihin ylläpitotehtäviin käytettäviä työkaluja kuten Remote Desktop Protocol -etäkäyttöä. Näillä menetelmillä kuviossa 1 hyökkääjä pääsee siirtymään kohdeverkossa ensimmäiseltä työasemalta k_1 syvemmälle verkkotopologiassa järjestelmiin $k_2 - k_7$, jotka sisältävät itsessään mahdollista sensitiivisiä tietoja tai hyökkäystä edistäviä asioita. Poikkeamien havainnointi massasta vaati jatkuvaa sopeutumista ympäristön muutoksiin kuten uusiin ylläpitotunnuksiin, missä koneoppimismenetelmien mukautuvuus tuo etua perinteisempiin

raja-arvoihin tai yksittäisiin tapahtumiin perustuviin havaitsemismenetelmiin.

Kolmannessa vaiheessa pyritään tunnistamaan ja keräämään arvokasta tietoa, kuten kehityssuunnitelmia ja liikesalaisuuksia (Chen, Desmet ja Huygens 2014). Keräämisen jälkeen hyökkäyksen seuraavissa vaiheissa tiedot voidaan siirtää hyökkääjälle ja tiedoilla voidaan kiristää kohdetta. Saaduilla oikeuksilla tiedot ja varmuuskopiot voidaan myös salata osana kiristystä. Vaikka yritysten ja organisaatioiden tiedoilla ei olisi kovin suurta myyntiarvoa ulkopuolisille, on sillä todennäköisesti suuri rahallinen arvo itse tietojen omistajille palveluiden toimivuuden ja jatkuvuuden kannalta.

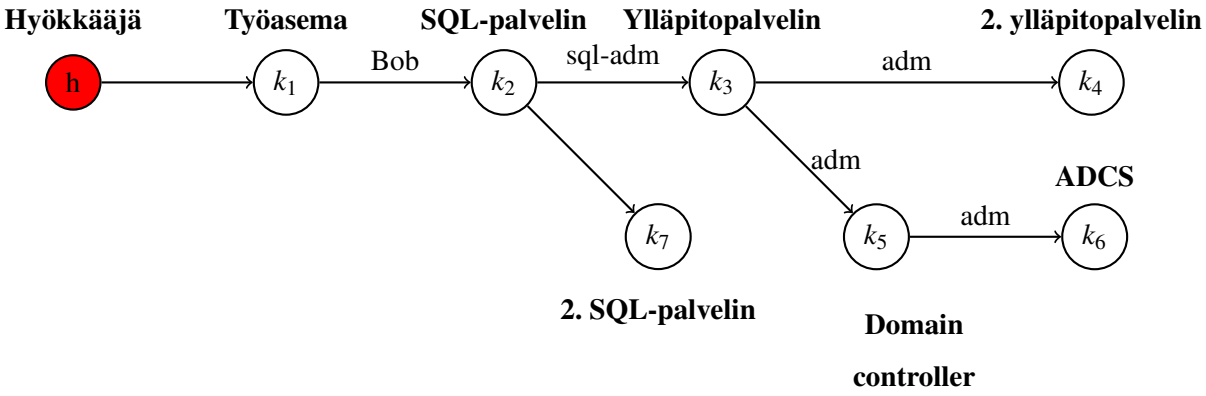
2.3 Tunkeutumisen laajentamisen jättämät jäljet

Edellä mainitut menetelmät ja vaiheet jättävät erilaisia jälkiä järjestelmien lokeihin. Työasemilla ja palvelimilla on mahdollista havaita asioita käyttöjärjestelmien ja sovellusten lokeista. Tunkeutumisen laajentamisessa käytetään yleisesti SSH-, RDP- tai WMI-protokollia eri palvelimille siirtyessä sekä komentojen ajamiseen etänä. Näitä voidaan tunnistaa kyseisiä protokollia käyttävien palveluiden tuottamista lokeista tai generoimasta verkkoliikenteestä (Bai ym. 2019). Windows-kirjautumislokien tutkiminen on keskeisessä asemassa hyökkäyksen havaitsemisessa Windowsin ollessa yksi keskeisimmistä käyttöjärjestelmistä yritysverkkojen tunnistamisessa ja todentamisessa.

Tunkeutumisen laajentamisessa tapahtuva liikenne eri koneiden välillä muistuttaa päivityksistä syntyvää liikkumista, mikä vaikeuttaa hyökkäyksien havaitsemista ja synnyttää helposti vääriä hälytyksiä havainnointijärjestelmissä (Bohara ym. 2017).

Tekniikka	Alitekniikat
Etäpalveluiden hyväksikäyttö	
Kohdennettu sisäinen tietojenkalastelu	
Työkalujen siirto	
Etäpalveluiden sessioiden kaappaus	SSH-session kaappaus
	RDP-session kaappaus
Etäpalvelut	Remote Desktop Protocol
	SMB/Windows Admin -verkkojaot
	Distributed Component Object Model
	SSH
	VNC
	Windows Remote Management
Leviäminen irroitettavalla medialla	
Ohjelmiston jakelusovellukset	
Jaetun sisällön muokkaaminen	
Vaihtoehtoinen autentikointimateriaali	Sovelluksen käyttöoikeustietue (*Access token*)
	Pass the Hash
	Pass the Ticket
	Web-sessiotunnisteet

Taulukko 1. MITRE (2021) -viitekehyksen tunkeutumisen laajentamisen tekniikat ja mahdolliset alitekniikat



Kuvio 1. Kuvaaja tunkeutumisen laajentamisesta (Bowman ym. 2020)

3 Koneoppimisen käyttäminen luokittelussa

Tässä luvussa esitellään koneoppimisen käsitteitä ja sen käyttämistä luokittelussa. Aluksi määritellään mitä koneoppiminen on sekä miten se voidaan jakaa ohjattuun ja ohjaamattomana oppimiseen. Samalla esitellään koneoppimisen hyödyntämistä luokittelussa ja datan esikäsittelyssä sekä kuinka luokittelumenetelmien toimivuutta voidaan mitata. Lisäksi luvussa käydään läpi luokitteluun liittyvät yleiset ongelmat eli ali- ja ylinäytteisyys. Tämä luku auttaa ymmärtämään, kuinka koneoppimista on mahdollista käyttää tunkeutumisen laajentumisen havainnoinnissa, jota käsitellään luvussa 4.

3.1 Koneoppiminen

Koneoppiminen on osa tekoälyn tieteenalaa ja se käsittää koneiden käyttämät oppimiseen kykenevät menetelmät. Koneoppimisella voidaan ratkaista ongelmia, joita ihmiset eivät pysty ratkaisemaan tai joita me pystymme ratkaisemaan, mutta joita emme pysty selittämään miten ja miksi. Lisäksi sillä voidaan ratkaista ongelmia, joissa ilmiöt muuttuvat jatkuvasti tai samantyylisiä ratkaisuja pitää pystyä mukauttamaan esimerkiksi käyttäjän mukaan. (Bhattacharyya 2013, s. 59) Koneoppimisen algoritmit pyrkivät oppimaan aineistosta uusia malleja, minkä avulla ne pystyvät tekemään päätöksiä esimerkiksi tulevaisuudesta tai uuden datapisteen luokasta (Murphy 2012)

3.2 Ohjattu koneoppiminen

Ohjattu koneoppiminen (*supervised learning*) kattaa Pariwatin (2017) mukaan seitsemänkymmentä prosenttia kaikesta koneoppimisesta ja näin ollen näin selvästi käytetympi koneoppimisen haara ohjaamattomaan koneoppimiseen verrattuna. Hänen mukaansa tekoälyn päätöksiä tekevä malli on opetettava merkatulla (*labeled*) datalla, josta tiedetään mitä luokkaa sen datapisteet esittävät tai mitä sen perusteella tekoälyn halutaan tekevän. On siis helppo jatkaa koneoppimismenetelmän opettamista, kunnes haluttu tarkkuus on saavutettu. Tämän ansioista valvottu koneoppiminen voi olla hyvinkin tehokasta, mutta sillä voidaan ennustaa ja tunnistaa lähinnä asioita, jotka toistavat samaa kaava ja näin historiallista dataa voidaan

käyttää uuden datan luokitteluun. Jos opetukseen käytetystä aineistosta puutuu merkkaukset, voidaan niitä luoda koneellisesti (Theiler ja Cai 2003). Bhattacharyya (2013, s. 125) nostaa esille ohjaamattomien luokittelijoiden merkittäväksi ongelmaksi niiden toimimisen, kun eri luokkien tapahtumia on suhteessa eri määrät. Ongelmaa voidaan koittaa ratkaista yli- ja alinäytteisyysmenetelmillä, joita käsitellään luvussa 3.6.

3.3 Ohjaamaton koneoppiminen

Ohjaamattomassa koneoppimisessa (*unsupervised learning*) kone oppii tunnistamaan datasta ryhmiä ja profiileja, joita alun perin mallidatassa ei kerrottu tai tiedetty olevan (Bhattacharyya 2013, s. 71). Tämä tulee hyödylliseksi, kun ei ole täysin selvää miltä poikkeamat näyttävät. Ohjaamatonta oppimista käytettäessä rajoja ei tarvitsee määrittää tiukasti, vaan koneoppimisella voidaan löytää joustavasti tarvittavat muuttujat. Tämän on verkkoliikenteen anomalioiden havaitsemisessa erityisen hyödyllistä, sillä esimerkiksi APT-hyökkäykset voivat olla niin hienovaraisia, että puhtaasti vaikka pakettikokoihin keskittyminen ei todennäköisesti auta niiden havaitsemisessa. Liikennettä on analysoitava laajemmassa kontekstissa, esimerkiksi tutkimalla onko paketti tietyssä ajassa tyypillisesti käyttäjän lähettämä vai onko syytä epäillä tunnuksen joutumista vääriin käsiin. Bhattacharyya (2013, s. 126) nostaa esille kuinka ohjaamattomat luokittelijat olettavat anomalioiden määrän olevan merkittävästi pienempi verrattuna normaaleihin tapahtumiin.

3.4 Luokittelumenetelmien suorituskyvyn mittaaminen

Luokittelumenetelmien tarkkuutta luokitella tapahtumia haitallisiksi ja haitattomiksi kuvataan yleensä oikeilla positiivisilla (*true positive, TP*) ja väärillä positiivisisilla (*false positive, FP*) sekä oikeilla negatiivisilla (*true negative, TN*) ja väärillä negatiivisilla (*false negative, FN*). Oikeiden positiivisten suhde kuvaa kuinka hyvin luokittelija pystyy luokittelemaan tapahtumia oikein haitallisiksi ja se lasketaan seuraavasti.

$$\text{Oikeiden positiivisten suhde} = \frac{\text{Oikeat positiiviset}}{\text{Oikeat positiiviset} + \text{Väärät negatiiviset}}$$

Väärien positiivisten suhde kuvaa kuinka paljon luokittelija luokittelee tapahtumia väärin

haitallisiksi. Väärien positiivisten ja väärien negatiivisten kanssa joudutaan monesti tasapainottelemaan. Jos väärien positiivisten määrä halutaan nollaan, on todennäköistä, että väärien negatiivisten määrä kasvaa (Fang ym. 2022). Väärien positiivisten suhde lasketaan seuraavasti.

$$\text{Väärien positiivisten suhde} = \frac{\text{Väärät positiiviset}}{\text{Väärät positiiviset}+\text{Oikeat negatiiviset}}$$

Huomionarvoista on, että monesti hyökkäyksestä lokeihin jäävät tapahtumat ovat marginaalinen osa kaikista tapahtumista. Jos algoritmi luokittelee kaikki tapahtumat normaaleiksi, on mahdollista päästä käytetystä aineistosta riippuen lähes sadan prosentin oikeiden positiivisten tarkkuuteen, kuten Bai ym. (2019) nostavat esille. Tämä korostaa tarvetta huolelliseen vertailuun ja varovaisuuteen tulosten tulkinnassa. Eri tutkimuksten tulokset eivät ole välttämättä suoraan verrattavissa toisiinsa valittujen mittarien ja käytettyjen aineistojen takia.

Lisäksi, yleisesti käytettyjä mittareita ovat ulkoinen tarkkuus (*accuracy*) ja sisäinen tarkkuus (*precision*). Ensimmäinen kuvaa kuinka hyvin luokittelija luokittelee tapahtumia oikein haitallisiksi ja sisäinen tarkkuus kuinka suuri osa haitallisiksi luokitelluista tapahtumista ovat oikeasti haitallisia (Bhattacharyya 2013, s. 237-241). Ne lasketaan seuraavasti.

$$\text{Ulkoinen tarkkuus} = \frac{\text{Oikeat positiiviset}+\text{Oikeat negatiiviset}}{\text{Oikeat positiiviset}+\text{Oikeat negatiiviset}+\text{Väärät positiiviset}+\text{Väärät positiiviset}}$$

$$\text{Sisäinen tarkkuus} = \frac{\text{Oikeat positiiviset}}{\text{Oikeat positiiviset}+\text{Väärät positiiviset}}$$

Vertailua voidaan myös tehdä tutkimalla luokittelualgoritmien suorituskykyvaatimuksia ajankäytön ja resurssien suhteen. Tämä voi olla tärkeää, jos aineistossa on miljoonia tapahtumia ja luokittelijan uudelleenopettamisessa menee minuuttien sijaan tunteja tai päiviä.

3.5 Muuttujien poiminta

Luokittelijoille annettavasta datasta valitaan usein vain osa muuttujista (*feature selection*) tai datan perusteella johdetaan uusia muuttujia (*feature extraction*). Näiden avulla käsiteltävän datan määrää voidaan saada jo vähennettyä merkittävästi, millä voi olla suuria vaikutuksia aikaan, joka menee luokittelijoiden opetukseen ja käyttöön. Tämän lisäksi ne voivat parantaa tarkkuutta ja auttaa ymmärtämään dataa paremmin. (Bhattacharyya 2013, s. 160-162)

Muuttujien määrää voidaan myös vähentää koneoppimisen avulla. Algoritmit kuten pääkomponenttianalyysi (*principal component analysis*, PCA) laskee eri muuttujien välisiä korrelaatioita ja luo niiden perusteella tietoa tiivistäviä muuttujia, joiden välillä ei ole korrelaatiota. Tämän avulla on mahdollista saada tarkempia tuloksia ja parantaa suorituskykyä. (Bohara ym. 2017)

3.6 Ali- ja ylinäytteisyys

Monet koneoppimisen luokittelualgoritmit menettävät tarkkuuttaan, jos käytetyssä datassa eri luokkien esiintyvyys vaihtelee suuresti. Bian ym. (2019) mukaan esiintyvyyden erot aiheuttavat enemmän esiintyvää luokan suosimista, mutta puolueellisuutta voidaan yrittää pienentää alinäytteisyyden (*down-sampling*) ja ylinäytteisyyden (*over-sampling*) menetelmillä. Alinäytteisyysalgoritmit poistavat eniten esiintyvän luokan alkioita, mutta samalla ne voivat hävittää oleellista tietoa, jota tarvitaan luokittelussa. Ylinäytteisyysalgoritmit lisäävät vähemmistöön kuuluvien alkioiden määrää, mikä taas puolestaan saa luokittelijat tulkitsemaan vääriä alkioita vähemmistöluokkaan (*over-fitting*). (Baldi 2011)

4 Koneoppimisen hyödyntäminen tunkeutumisen laajentamisen havaitsemisessa

Tunkeutumisen laajentamista on mahdollista havaita käyttäen apuna koneoppimisen menetelmiä. Luku 4.1 käsittelee mitä tietoaineistoja on käytetty aiemmissa tutkimuksissa tunkeutumisen laajenemisen havaitsemiseen ja luvussa 4.2 käydään läpi kuinka koneoppimisella on mahdollista esikäsitellä tietoja ennen luokittelua. Lopuksi esitellään ohjatun ja ohjaamattoman koneoppimisen havaitsemismenetelmiä, joita aiemmissa tutkimuksissa on hyödynnetty tunkeutumisen laajentamisen tunnistamisessa.

4.1 Käytetyt tietoaineistot

Ohjatun ja ohjaamattoman koneoppimisen metodit perustuvat luokittelijan opettamiseen tietoaineiston perusteella, jossa on hyökkäykseen ja normaaliin käyttöön liittyvää dataa. Yhdysvaltain Energialaitoksen alla toimivan Los Alamos National Laboratorion (LANL) avoimesti saatavilla oleva tietoaineistoa (Kent 2015) on käytetty selvästi eniten aiemmissa tutkimuksissa (Bowman ym. 2020; Bian ym. 2019; Kaiafas ym. 2017; Bohara ym. 2017; Bai ym. 2019). Sen 58 päivän lokit kattavat eroteltuna heidän ympäristössä tapahtuneet todentamiset, prosessien käynnistykset ja lopetukset, NetFlow-lokit verkkoliikenteestä, DNS-liikenteen sekä hyökkäystä simuloivan punaisen joukkueen (*red team*) todentamiset.

Fang ym. (2022) käytti myös pienempää *Insider Threat Test Dataset* (2016) -tietoaineistoa LANL-aineiston rinnalla. Koneellisesti generoitu aineisto kattaa 4000 käyttäjää ja 4400 konetta sekä viisi merkattua hyökkäystä.

Bai ym. (2019) nostavat esille, kuinka kaikkiin käyttötarkoituksiin ei ole saatavilla julkisesti täysin sopivaa aineistoa, ja hän joutuikin koneellisesti muokkaamaan aineiston hyökkäykseen liittyviä lokeja. Bhattacharyyan (2013, s. 143) mukaan julkisissa aineistoissa hyökkäyksien lokimäärät ovat yliedustettuina verrattuna oikeisiin hyökkäyksiin. Yrityksillä on mahdollisuus tuottaa omaa infraansa vastaavaa aineistoa hyökkäyssimulaatioiden avulla.

Bohara ym. (2017) nostavat esille, kuinka haitallisen tunkeutumisen laajentamisen havain-

noinnissa ongelmana on tietää kuinka oikeasti hyökkääjät liikkuvat tietoverkoissa. Tämä vaikeuttaa todenmukaisten aineistojen tuottamista ja rajaa tutkimusta entisestään muutamiin avoimesti saatavilla oleviin tietoaineistoihin. Lisäksi herää kysymys, eroavatko APT-ryhmien tavat jo liikaa, että esitetyt havainnointimallit toimisivat niiden tunnistamisessa.

4.2 Tutkittavat muuttujat

Valituilla muuttujilla on suuri merkitys, kuinka tarkasti koneoppimisella toteutettu luokittelu tulee toimimaan. Tietojärjestelmät tuottavat valtavan määrän lokidataa niiden tapahtumista, ja vaikka koneoppimisen algoritmeja on teoriassa triviaalia hyödyntää kaikkiin muuttujiin, saadaan hyödyllisiä tuloksia vain oikeiden muuttujien valinnalla sekä niiden mahdollisella jalostamisella. Bai ym. (2019) huomasivat, että käyttäjänimen sisällyttäminen luokittelussa käytettyihin kenttiin johti tilanteeseen, missä kerran hyökkäykseen käytetyn käyttäjänimen hyökkäyksen jälkeinen normaalikin liikenne tulkittiin haitalliseksi. Kyseisessä tutkimuksessa kirjautumisiin liittyvät kentät, kuten käyttäjänimi, lähdekone ja kohdekone pudotettiin käytettyjen muuttujien joukosta ja jäljelle jäi RDP-sessioihin liittyviä tilastollisia sekä aikaan liittyviä kenttiä, jotka on listattu taulukossa 2. Kaiafas ym. (2017) päätyivät käyttämään kirjautumisista laskettujen kenttien lisäksi kaikkia LANL-aineiston kenttiä.

Vääristymien ja aikavaatimusten takia tutkimuksissa on päädytty monesti karsimaan käytettyjä kenttiä tai luomaan useamman kentän varianssin sisällyttäviä uusia muuttujia. Valitut muuttujat voivat olla lokien kenttien arvoja tai niistä laskettuja tilastollisia määriä (Bai ym. 2019) tai voidaan käyttää lokien avulla muodostettuja arvoja kuten koneiden välisten yhteyksien määrää (Bohara ym. 2017). Koneoppimismenetelmät pystyvät löytämään tapahtumien muuttujien välillä valitsevaa korrelaatiota ja sisällyttämään sen yhteen uuteen muuttujaan. Bohara ym. (2017) pystyivät ohjaamattomalla koneoppimisen pääkomponenttianalyysillä säilyttämään 95 % alkuperäisen datan vaihtelevuudesta yhdistämällä muuttujat kaksiulotteiseen dimensioon. Laskenta-ajan säästämisen lisäksi tämä vähentää luokittelun puolueellisuutta sekä helpottaa tulosten kuvaamista kaksiulotteisella koordinaatistolla.

Suuri osa tutkimuksista keskittyy eri koneiden välille muodostettujen liikennöinti- ja kirjautumisgraafien tulkitsemiseen (Bohara ym. 2017; Chen ym. 2018; Bian ym. 2019; Kaiafas

ym. 2017; Bowman ym. 2020). Käyttäjätunnuksien kirjautumisten perusteella on mahdollista luoda tietueita, jotka sisältävät yksinkertaisimmillaan kuvion 1 kaltaista tietoa kirjautumisketjuihin käytetyistä tunnuksista, lähdekoneista ja kohdekoneista.

Bohara ym. (2017) ovat esitelleet tunkeutumisen laajentamisen havaitsemismallin, joka perustuu näiden ketjujen tarkkailuun. Jokaiselle koneelle lasketaan yhteysketjujen määrä, joissa kone on mukana sekä näille ketjuille pituuksien keskiarvo, vaihteluväli, mediaani ja interkvartiilialue. Arvojen muodostamiseen tarvittavat tiedot on mahdollista kerätä ympäristön verkkolokia tuottavista palomuureista tai reitittimistä ja koneiden lokeista. Yksi ketjun osa voi olla käyttäjän kirjautuminen omalle työasemalleen, josta hän kirjautuu tietokantapalvelimelle RDP-protokollaa käyttäen.

Kenttä	Selite
Session kesto	Kulunut aika sisään- ja uloskirjautumisen välissä
Käyttäjän kirjautumisten aikaero	Käyttäjän peräkkäisten kirjautumisten välinen aikaero
Lähdekoneen kirjautumisten aikaero	Lähdekoneen peräkkäisten kirjautumisten välinen aikaero
Kohdekoneen kirjautumisten aikaero	Kohdekoneen peräkkäisten kirjautumisten välinen aikaero
Käyttäjän sessioiden kesto	Tunnuksen RDP-sessioiden keston keskiarvo
Lähdekoneen sessioiden kesto	Tunnuksen RDP-sessioiden keston keskiarvo
Kohdekoneen sessioiden kesto	Tunnuksen RDP-sessioiden keston keskiarvo
Viikonpäivä	Viikonpäivä, jolloin sessio on tapahtunut
Kuluneet sekunnit	Monesko sekunti päivästä on menossa, kun sessio on alkanut

Taulukko 2. Bai ym. (2019) käyttämät kentät

4.3 Ohjaamattoman oppimisen käyttö

Tutkijoiden on vaikea saada varmaa tietoa kuinka oikeassa kyberhyökkäyksessä hyökkääjät hyödyntävät tunkeutumisen laajentamista. Tämän lisäksi APT-hyökkäyksissä metodeja sovitetaan kuhunkin ympäristöön sopivaksi. Näiden seikkojen takia voidaan ajatella, että ohjaamaton oppiminen sopisi paremmin tunkeutumisen laajentamisen havaitsemisessa ohjattua koneoppimista paremmin varsinkin, jos merkattua dataa ei ole olemassa. Kuten luvussa 3.3

mainittiin, monet ohjaamattoman koneoppimisen menetelmät olettavat anomalioiden olevan määrällisesti pieni osa koko massasta, mikä pitää monesti tunkeutumisen laajentamisen kanssa paikkaansa.

Boharan ym. (2017) malli perustuu koneiden välisien yhteyksien määriin ja niiden muodostamien ketjujen pituuksiin, koneiden liikennöintimääriin, uniikkeihin kohteisiin sekä verkkoliikenteeseen tehtyyn luokitteluun. He nostavat esille, kuinka hyökkäyksistä muodostuvat ketjut ovat selvästi ylläpidollisia ketjuja lyhyempiä, mutta moneen hyökkäykseen kuulumattomaan koneeseen ei liity yhtään ketjua. Heidän havainnointinsa hyödyntää extreme value - ja k:n keskiarvon klusterointimenetelmän yhdistelmää, joita ennen datalle tehdään pääkomponenttianalyysi. Näin pystyttiin tunnistamaan parhaimmillaan 92 % ulkoisella tarkkuudella oikeasti kompromisoiduista koneista, mutta väärin kompromisoiduksi tulkittujen koneiden prosentuaalinen määrä oli kohtalaisen suuri 14 %.

Fang ym. (2022) esittelemä LMTracker-malli muodostaa Windowsin lokitietojen perusteella heterogeenisen graafin. Sen linkit muodostuvat ajanhetkestä, lähde- ja kohdekoneesta, lähde- ja kohdekäyttäjästä, näiden välisten linkkien tyypistä ja tyyppiin liittyvistä muista attribuuteista. AutoEncoder-algoritmilla saavutettiin 91 % ulkoinen tarkkuus ja väärien positiivisten suhde oli parhaimmillaan 8 %, jota pidettiin tutkimuksessakin kohtalaisen suurena. Kun graafin muodostavat ketjut on koostettu kirjautumisia, koneiden välisistä yhteyksistä sekä prosessien ja tiedostojen luomisista, on odotettavissa satoja vääriä havaintoja päivässä.

4.4 Ohjatun oppimisen käyttö

Ohjatulla oppimisella päästään yleensä ohjaamatonta oppimista parempiin tuloksiin, mutta sen käyttäminen vaatii merkatun aineiston. Luvussa 3.2 esiteltiin ohjatun koneoppimisen luokittelijoiden luottavan monesti oletukseen, että kaikkia luokkia esiintyy datassa suurin piirtein samat määrät. Tunkeutumisen laajentumisesta generoima lokimäärä on useasti kuitenkin huomattavasti pienempi suhteessa normaaleihin tapahtumiin, minkä takia ennen opetus- ja luokitteluvaihetta dataa voi joutua käsittelemään luvussa 3.6 esitetyillä ali- ja ylinäytteisyysmenetelmillä. Bian ym. (2019) ja Bai ym. (2019) eivät kuitenkaan nähneet näytteisyysmenetelmien tuovan tutkimuksiensa mukaan merkittävää hyötyä luokittelutarkkuu-

teen suhteutettuna kasvavaan opetusaikaan.

Chen ym. (2018) pyrkivät parantamaan aiemmin esiteltyä Boharan ym. (2017) ohjaamattoman oppimisen luokittelijaa. Havainnointi perustuu koneiden välisistä yhteyksistä muodostuvaan graafiin, joka on koostettu kirjautumisista koneilta toisille ja verkkoliikenteestä koneiden välillä. K:n lähimmän naapurin luokittelumenetelmällä he saavuttivat Boharaa ym. (2017) paremman tarkkuuden. Ulkoiseksi tarkkuudeksi saatiin 99,9 % ja verrattava väärien positiivisten määrä oli vain 0,01 %. Menetelmän toimiminen vaatii kuitenkin, että hyökkääjän ja normaalien toimipiteiden aiheuttamat lokimäärät ovat tasapainossa ja LANL-dataa jouduttiin käsittelemään alinäytteisyysmenetelmällä. Tasamääräisyyden ei voida kuitenkaan olettaa pitävän paikkaansa oikeassa hyökkäyksessä ja ympäristössä. Alinäytteisyyden toteuttaminen vaatii dataa, jossa on hyökkäysketjuja mukana ja niitä pitää pystyä tunnistamaan. Ympäristössä, jossa ei ole havaittua tunkeutumisen laajentumista, tätä joudutaan generoimaan. Osana lähestymistapaa käytettiin myös ohjaamatonta koneoppimista datan dimensioiden vähentämisessä.

Bai ym. (2019) pääsivät useammalla ohjatun oppimisen luokittelijalla yli 99 % luokittelutarkkuuteen normaalin ja haitallisen RDP-liikenteen erottelussa ilman ali- tai ylinäytteisyysmenetelmiä. Tutkimuksessa kahdeksasta ohjatusta algoritmista parhaat tulokset saatiin LogitBoost-algoritmilla, jolla tehostettiin heikomman päätöspuuluokittelijan tuloksia. Luokittelukykyä ei merkittävästi pystytty parantamaan käyttämällä ohjaamatonta oppimista aineiston ryhmittelyssä ennen luokittelijaa. Menetelmän kykyä havaita haitallisia kirjautumisia simuloitiin lisäämällä hyökkäyksiin viittaavia lokitapahtumia, jotka havaittiin myös hyvin. Tällä pyritään osoittamaan, että menetelmä pystyy tunnistamaan myös hyökkäyksiä, joita ei ollut opetukseen käytetyssä datassa.

Bian ym. (2019) tutkimus on hyvin samanlainen Bai ym. (2019) tutkimuksen kanssa sen käyttäessä myös LANL-tietoaineiston kirjautumislokeja. Päätöspuuluokittelijoilla ja LogitBoostilla päästiin 77 - 84 % ulkoisiin tarkkuuksiin. Huonompi suorituskyky verrattuna Bain ym. menetelmään selittyy todennäköisesti eri muuttujien käyttämisellä.

Kaiafas ym. (2017) käyttivät LogitBoost-, Random Forest- ja Logistic Regression -algoritmien tuloksia yhdessä yhdistääkseen erityyppisten luokittelijoiden ominaisuuksia tarkempien tu-

loksien toivossa. Malli tunnisti LANL-datasta (2015) kaikki oikeat hyökkäykset, mutta väärin haitallisiksi tulkittujen tapahtumien määrä oli kuitenkin yli kolmekymmentätuhatta päivässä. Tällaista määrää ei ole mitenkään mahdollista käsitellä manuaalisesti.

5 Yhteenveto

Tässä tutkielmassa on avattu mitä kyberhyökkäyksen tunkeutumisen laajentaminen on ja kuinka sitä on mahdollista havaita koneoppimisella. Ohjatun oppimisen luokittelijoilla on eri tutkimuksissa pysytty saavuttamaan yli 99 % ulkoinen tarkkuus väärien positiivisten määrän ollessä erittäin pieni. Ohjaamattomat oppimisella on päästy parhaimmillaan 91 % ulkoiseen tarkkuuteen väärien positiivisten määrän ollessa varsin suurin 8 %.

Jatkotutkimus voisi keskittyä parhaimpien menetelmien toteuttamiseen tietoturvavalvontaan yleisesti käytetyssä SIEM (*Security Information and Event Management*) -järjestelmässä. Toteutuksessa olisi hyvä pitää mielessä sen toimiminen erilaisissa muuttuvissa ympäristöissä sekä taata analyytikoille tarpeeksi kontekstitietoa tutkintaan varten. Parhaimpia menetelmiä olisi mahdollista yhdistää sekä niiden toimintaa kuvata tarkemmin. Reaalimaailmaan suhteutettuna on suoraan vaikea arvioida pysyisikö väärien positiivisten määrä päivätasolla järkevissä määrissä vai pitäisikö havaintoja yhdistää muihin heikkoihin havaintoihin.

Lisäksi aiempien tutkimusten käyttäessä lähes poikkeuksetta LANL-aineistoa ei voida varmaksi sanoa kuinka luokittelijoita opetettaisiin uusiin ympäristöihin, joissa ei ole jälkiä tunkeutumisen laajentamisesta. Tutkimuksissa jätetään myös monesti huomioimatta kuinka meluisia luokittelijat ovat, jos ympäristössä ei ole käynnissä hyökkäystä.

Lähteet

Bai, Tim, Haibo Bian, Abbas Abou Daya, Mohammad A Salahuddin, Noura Limam ja Raouf Boutaba. 2019. “A machine learning approach for rdp-based lateral movement detection”. Teoksessa *2019 IEEE 44th Conference on Local Computer Networks (LCN),* 242–245. IEEE. doi:10.1109/LCN44214.2019.8990853.

Baldi, Pierre. 2011. “Autoencoders, Unsupervised Learning, and Deep Architectures”. Teoksessa *Proceed- ings of the International Conference on Unsupervised and Transfer Learning Workshop,* toimittanut Isabelle Guyon, Gideon Dror, Vincent Lemaire, Graham Taylor ja Daniel Silver, 37–49. Washington, USA: PMLR.

Bhattacharyya, Dhruba. 2013. *Network Anomaly Detection: A Machine Learning Perspective.* Tezpur, Assam, India: Chapman / Hall/CRC.

Bian, Haibo, Tim Bai, Mohammad Salahuddin, Noura Limam, Abbas Abou Daya ja Raouf Boutaba. 2019. “Host in Danger? Detecting Network Intrusions from Authentication Logs”. doi:10.23919/CNSM46954.2019.9012700.

Bohara, Atul, Mohammad Noureddine, Ahmed Fawaz ja William Sanders. 2017. “An Unsupervised Multi-Detector Approach for Identifying Malicious Lateral Movement”. doi:10.1109/SRDS.2017.31.

Bowman, Benjamin, Craig Laprade, Yuede Ji ja H Howie Huang. 2020. “Detecting lateral movement in enterprise computer networks with unsupervised graph AI”. Teoksessa *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020),* 257–268.

Chen, Mingyi, Yepeng Yao, Junrong Liu, Bo Jiang, Liya Su ja Zhigang Lu. 2018. “A Novel Approach for Identifying Lateral Movement Attacks Based on Network Embedding”. doi:10.1109/BDCloud.2018.00107.

Chen, Ping, Lieven Desmet ja Christophe Huygens. 2014. “A Study on Advanced Persistent Threats”. doi:https://doi.org/10.1007/978-3-662-44885-4_5.

Delpy, Benjamin. 2007. “Mimikatz”. Viitattu 9. helmikuuta 2022. https://github.com/gentilkiwi/mimikatz.

Fang, Yong, Congshuang Wang, Zhiyang Fang ja Cheng Huang. 2022. “LMTracker: Lateral movement path detection based on heterogeneous graph embedding”. *Neurocomputing* 474:37–47. doi:https://doi.org/10.1016/j.neucom.2021.12.026.

Insider Threat Test Dataset. 2016. Software Engineering Institute, Carnegie Mellon University. doi:10.1184/R1/12841247.v1.

Kaiafas, Georgios, Georgios Varisteas, Sofiane Lagraa, Radu State, Cu D Nguyen, Thorsten Ries ja Mohamed Ourdanes. 2017. “Detecting Malicious Authentication Events Trustfully”. doi:10.1109/NOMS.2018.8406295.

Kent, Alexander D. 2015. *Comprehensive, Multi-Source Cyber-Security Events.* Los Alamos National Laboratory. doi:10.17021/1179829.

Kyberturvallisuuskeskus. 2021. “Tunnetko tunkeutumisen laajentamisen (osa 1)”. Viitattu 11. helmikuuta 2022. https://www.kyberturvallisuuskeskus.fi/fi/ajankohtaista/tunnetko-tunkeutumisen-laajentamisen-osa-1.

MITRE. 2021. “MITRE ATT&CK® -tietopohja”. Viitattu 6. toukokuuta 2021. https://attack.mitre.org/.

Murphy, Kevin P. 2012. *Machine learning: a probabilistic perspective.* MIT press.

Pariwat, Ongsulee. 2017. “Artificial Intelligence, Machine Learning and Deep Learning”. Teoksessa *2017 Fifteenth International Conference on ICT and Knowledge Engineering.* Bangkok, Thailand. doi:10.1109/ICTKE.2017.8259629.

Theiler, James P, ja D Michael Cai. 2003. “Resampling approach for anomaly detection in multispectral images”. Teoksessa *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery IX,* 5093:230–240. SPIE.

Tian, Zhihong, Wei Shi, Yuhang Wang, Chunsheng Zhu, Xiaojiang Du, Shen Su, Yanbin Sun ja Nadra Guizani. 2019. “Real-time lateral movement detection based on evidence reasoning network for edge computing environment”, 15:4285–4294. 7. IEEE. doi:10.1109/TII.2019.2907754.

Tyufu, Enn. 2011. “Artificial Intelligence in Cyber Defense”. Teoksessa *3rd International Conference on Cyber Conflict*. Tallinn, Estonia. https://ieeexplore.ieee.org/document/5954703.

Ussath, Martin, David Jaeger, Feng Cheng ja Christoph Meinel. 2016. “Advanced Persistent Threats: Behind the Scenes”. doi:10.1109/CISS.2016.7460498.

Yua, Shi, Chang Xiaolina, Ricardo Rodríguezb, Zhang Zhenjiangc ja Kishor Trivedid. 2019. “Quantitative security analysis of a dynamic network system under lateral movement-based attacks”. doi:https://doi.org/10.1016/j.ress.2018.11.022.