Hariet Tamvelius

Tekoälyn käyttö penetraatiotestauksessa

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikka

Insinöörityö

21.5.2024

Tiivistelmä

Tekijä:	Hariet Tamvelius
Otsikko:	Tekoälyn käyttö penetraatiotestauksessa
Sivumäärä:	39 sivua
Aika:	21.5.2024
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Tieto- ja viestintätekniikka
Ammatillinen pääaine:	Älykkäät IoT-järjestelmät
Ohjaajat:	Tutkijaopettaja Aarne Klemetti

Tämän insinöörityö on tutkielmatyyppinen raportti, jonka tarkoituksena on käsitellä tekoälyn käyttöä kyberturvallisuudessa, tarkemmin ottaen testauksen näkökannalta. Raportin aihe on saanut alkunsa omasta henkilökohtaisesta mielenkiinnosta kyberturvallisuutta ja hakkerointia kohtaan. Koska tekoälyä käytetään eri tavoin jo lähes kaikissa laitteissa, järjestelmissä ja palveluissa, on päätetty tutkia, miten sitä voisi hyödyntää myös penetraatiotestauksessa.

Aihealueen laajuuden vuoksi insinöörityö on jaettu neljään osuuteen. Ensimmäisessä osiossa perehdytään yleisellä tasolla kyberturvallisuuteen, suosittuihin hyökkäysmenetelmiin sekä kyberuhilta suojautumiseen valvonnan ja testauksen avulla. Toisessa osiossa määritellään tiivistetysti tekoäly ja joitakin siihen liittyviä käsitteitä sekä pohditaan tekoälyn käytön hyötyjä ja haasteita. Kolmannessa luvussa käsitellään kybertestauksen ja tekoälyn yhdistämistä kolmella eri tavalla. Pohditaan, miten tekoälyä voidaan hyödyntää muun muassa hunajapurkeissa, tietojenkalastelussa ja käyttäjän manipuloimisessa sekä ohjelmakoodin tuottamisessa ja analysoinnissa. Lopuksi luodaan ChatGPT:n avulla oma tekoälyä hyödyntävä penetraatiotestaustyökalu.

Päällimmäinen tarkoitus tämän tutkielman laatimisessa oli oppia lisää tekoälystä, sen eri osa-alueista ja menetelmistä, ja niiden kautta pohtia, miten tekoälyn käyttöä kyberturvallisuudessa voisi kehittää vielä entisestään ja millaisia mahdollisuuksia se on luonut tai tulee luomaan tulevaisuudessa.

Avainsanat: tekoäly, kyberturvallisuus, tietoturva, kyberhyökkäys, tietoturvahyökkäys, penetraatiotestaus

---

Abstract

Author:	Hariet Tamvelius
Title:	Use of Artificial Intelligence in Penetration Testing
Number of Pages:	39 pages
Date:	21 May 2024
Degree:	Bachelor of Engineering
Degree Programme:	Information Technology
Professional Major:	Smart IoT Systems
Supervisors:	Aarne Klemetti, Researching Lecturer

The purpose of this thesis is to investigate the use of artificial intelligence in cyber security, more precisely from the point of view of security testing. The topic of the study originated from the author's own personal interest in cyber security and hacking. Since artificial intelligence is already used in almost all devices, systems and services, it was decided to investigate how it could also be used in penetration testing.

Due to the scope of the subject, the thesis is divided into four parts. The purpose of the first part is to discuss cyber security at a general level and to pre-sent popular cyberattack methods and to show how to protect various systems from cyber threats through monitoring and testing. The second part briefly de-fines artificial intelligence and related concepts. In addition, the advantages and challenges of using artificial intelligence are analyzed. The third part focuses on combining cyber testing and artificial intelligence in three different ways. The thesis analyzes how artificial intelligence can be used in, for example, honeypots, phishing and social engineering, as well as code generation and analysis. Finally, the thesis describes how an AI penetration testing tool using ChatGPT was created.

The main purpose of the project was to learn more about artificial intelligence, its various methods, and through that to consider how the use of artificial intelligence in cyber security could be developed even further and what kind of opportunities it has created or will create in the future.

Keywords: artificial intelligence, cybersecurity, information security, cyberattack, information security attack, penetration testing

Sisällys

Lyhenteet

AI: *Artificial Intelligence.* Tekoäly on tietokonejärjestelmä tai -ohjelma, joka on algoritmien avulla ohjelmoitu ja koulutettu toimimaan perinteisesti ihmisälyn tavoin.

ChatGPT: *Chat Generative Pre-Trained Transformer.* Tekoälyn tukimuskeskus OpenAI:n luoma suuria kielimalleja hyödyntävä keskustelubotti.

DDoS: *Distributed Denial-of-Service Attack.* Hajautettu palvelunestohyökkäys on verkkohyökkäysmenetelmä, jonka tavoitteena on estää verkkosivun käyttö kohdistamalla siihen liiallista verkkoliikennettä kaapatuilla laiteverkostoilla.

DoS: *Denial-of-Service Attack.* Palvelunestohyökkäys on hyökkäysmenetelmä, jonka tavoite on sama kuin hajautetussa palvelunestohyökkäyksessä, mutta se on peräisin vain yhdestä internetiin liitetystä laitteesta.

GenAI: *Generative Artificial Intelligence.* Generatiivinen tekoäly on yksi tekoälyn osa-alueista, jonka tarkoituksena on ihmisen luovaa ajattelutapaa jäljitellen tuottaa mitä tahansa uutta laadukasta sisältöä.

IP: Internet Protocol. Protokolla, joka hallitsee Internetissä tai paikallisessa verkossa lähetettävien tietojen muotoa.

LLM: *Large language model*. Laaja kielimalli on tekoälyjärjestelmä, joka massiivisia tietokantoja apuna käyttäen kykenee tunnistamaan ja tuottamaan ihmiskieltä.

ML: *Machine Learning.* Koneoppiminen on tekoälyn osa-alue, joka kykenee oppimaan annetun datan perusteella ilman, että ihminen erikseen opettaa sitä.

RL: *Reinforcement Learning.* Vahvistusoppiminen on koneoppimistekniikka, jota ei erikseen ohjata vaan se oppii toiminnan perusteella saadusta palautteesta.

SSH: *Secure Shell.* Internetprotokolla, jonka avulla voidaan muodostaa salattu etäyhteys tietokoneeseen tai -palvelimeen.

SQL: *Structured Query Language.* Standardoitu kyselykieli on tietokonekieli, jota käytetään tietojen hakemiseen tietokannasta.

TCP: Transmission Control Protocol. Verkkoprotokolla, joka tekee tiedon siirron verkossa mahdolliseksi.

1 Johdanto

Nopean digitalisaation myötä lähes kaikki käytössä olevista laitteista ja järjestelmistä on jollain tapaa liitetty verkkoon. Oli kyseessä sitten kotona pyörivä pesukone tai kunnan jätevesihuolto, nykypäivänä suuri osa yhteiskunnallekin elintärkeistä palveluista on yhteydessä internetiin. Moni näistä laitteista, järjestelmistä ja palveluista käyttää lisäksi tekoälyä, jonka kehitys on kasvanut räjähdysmäisesti eritysesti muutaman viime vuoden aikana. Yleinen digitalisaatio ja siihen tekoälyn mukaan ottaminen on helpottanut ja nopeuttanut merkittävästi ihmisten arkea sekä työntekoa, mutta on samalla tuonut mukanaan vähintään yhtä paljon ongelmia. Pääsääntöisesti kaikki mikä on jollain tapaa liitetty verkkoon, on samalla tehty saavutettavaksi ja hyväksikäytettäväksi myös verkkorikollisille. Laitteen tai järjestelmän ei edes tarvitse olla suoraan kytkettynä internettiin vaan riittää, että se on kytketty toiseen internettiä käyttävään järjestelmään.

Koska tekoäly on muun muassa automatisoinut monet työtehtävät, sitä on alettu käyttää myös kybervalvonnassa. Tekoälyn yleistymisen seurauksena ovat myös hakkerit opetelleet hyödyntämään ja kehittämään omia tekoälyjärjestelmiä, jotka auttavat tunnistamaan muita älykkäitä ohjelmistoja. Verkkorikolliset kehittävät taitojaan yhtä nopeasti ja tehokkaasti, kuin tekoälyä kehitetään. Sen takia pelkkä kybervalvonta ei auta ehkäisemään kyberhyökkäyksiä, vaan tarvitaan myös järjestelmien testausta mahdollisten haavoittuvuuksien löytämiseksi. Tähän käytetään usein penetraatiotestaajia. Koska tekoälyä käytetään jo lähes kaikkialla, sitä voisi hyödyntää myös kybertestauksessa.

Edellä mainitut seikat toimivat tämän insinöörityön innoittajina. Tarkoituksena on selvittää kolme tapaa, jolla tekoälyä voidaan hyödyntää penetraatiotestauksessa. Lopuksi on tarkoitus edellä mainittuja tapoja hyödyntäen luoda oma yksinkertainen penetraatiotestaustyökalu, joka hyödyntää toiminnassaan tekoälyä.

Jotta on mahdollista perehtyä tarkemmin tekoälyn käyttöön penetraatiotestauksessa, on tärkeää myös pohjustaa raporttia yleisesti kyberturvallisuuden ja

tekoälyn menetelmistä sekä niiden käsitteistä. Jo ainoastaan tekoälystä on kirjoitettu useita satoja, ellei tuhansia kirjoja ja artikkeleita. Voidaan siis olettaa, että käsiteltävä aihe on äärettömän laaja, joka vaatii jonkinasteista etukäteisvalmistelua ennen varsinaiseen asiaan siirtymistä.

Kyberturvallisuuden ja tekoälyn määrittäminen on pyritty tekemään mahdollisimman tiivistetysti käyttäen luettavuuden kannalta helposti ymmärrettäviä termejä. Raportissa käsiteltävä tieto on vain pisara valtameressä, ja käsitteitä ja termejä on todellisuudessa merkittävästi enemmän. Tarkoituksena on avata vain sellaisia termejä, jotka ovat tämän insinöörityön eri osioissa mainittu ja jättää ulkopuolelle ne käsitteet, joita ei raportissa mainita lainkaan, jotta lopputulos säilyisi mahdollisimman selkeänä. Tutkimustyön lähteinä ovat toimineet erilaiset artikkelit, eri teknologiayritysten verkkosivustot, aiheeseen liittyvä kirjallisuus, videot, tutkimus- ja opinnäytetyöt sekä oma aiemmin hankittu osaaminen ja tieto muun muassa erilaisten opintokokonaisuuksien kautta. Lisäksi oman penetraatiotestaustyökalun luomisessa Pythonilla on käytetty apuna ChatGPT:tä.

2 Kyberturvallisuus

Tässä luvussa käydään lyhyesti läpi, mitä kyberturvallisuus todellisuudessa on sekä käsitellään yleisempiä kyberhyökkäysmenetelmiä haittaohjelmista käyttäjän manipulointiin. Luvun lopussa keskitytään kyberuhilta suojautumiseen muun muassa kybervalvonnan ja penetraatiotestauksen avulla.

2.1 Määritelmä

Termit kyberturvallisuus ja tietoturva sekoitetaan usein toisiinsa, ja arkipuheessa näitä käytetäänkin monesti toisiaan vastaavina. Tieteellisessä kirjallisuudessa kyberturvallisuudella tarkoitetaan kuitenkin verkkoympäristössä olevan datan, tietojärjestelmien ja laitteiden suojaamista vaarantumiselta ja hyökkäyksiltä. Tietoturvallisuus taas on käsitteenä laajempi, ja siihen kuuluvat digitaalisen ympäristön turvaamisen lisäksi datan fyysinen tallentaminen ja sen valvominen. Tietoturvalla pyritään estämään kaikenlainen tiedon luvaton levittäminen sekä torjumaan väärää tietoa. Koska kyberturvallisuuden ja tietoturvan välillä on havaittavissa päällekkäisyyksiä, ovat ne kuitenkin erilaisia ja kyberturvallisuutta pidetään tietoturvallisuuden yhtenä osa-alueena. [1; 2.]

Kyberturvallisuus ei koske pelkästään tietokoneiden ja älylaitteiden turvaamista, vaan siihen kuuluu myös yhteiskunnalle elintärkeiden tavaroiden ja palvelujen toimitusketjujen suojaaminen, kuten esimerkiksi sähkönjakelu, vesi- ja jätehuolto, tele- ja rahaliikenne sekä sairaaloiden ja lääkinnällisten laitteiden toiminta. Toisin sanoen, kyberturvallisuudella pyritään suojaamaan kaikkea, missä digitaalisia järjestelmiä käytetään hyödyksi. [3, s. 110.]

2.2 Kyberuhat

Seuraavaksi käsitellään yleisimpiä kyberhyökkäysmenetelmiä. Huomionarvoista on, että kyberuhkia ja hyökkäystapoja on todellisuudessa paljon enemmän kuin tässä raportissa mainittu ja samoja menetelmiä saatetaan käyttää useissa eri tilanteissa.

2.2.1 Haittaohjelmat

Haittaohjelmilla (engl. Malware) tarkoitetaan tunkeutuvia ohjelmia, joiden tarkoituksena on hyväksikäyttää laitteita tai järjestelmiä, harhauttaa käyttäjiä sekä välttää suojaustoimintoja, asentaakseen itsensä laitteelle salaa, ilman käyttäjän lupaa. Näin on mahdollista saada laite käyttäytymään ei-toivotulla tavalla. Tarkoituksena on useimmiten varastaa arkaluonteista dataa, kuten esimerkiksi henkilötietoja tai vakoilla laitteen käyttäjää. [4.] Yleisimpiä haittaohjelmatyyppejä ovat kiristyshaittaohjelmat, troijalaiset ja vakoiluohjelmat.

Kiristyshaittaohjelma

Kiristyshaittaohjelman (engl. Ransomware) tarkoituksena on salata tiedostot laitteella tai lukita koko laite, jonka jälkeen vaaditaan uhria eli laitteen käyttäjää maksamaan lunnaat verkkorikollisille salauksen purkamiseksi. Lunnaat vaaditaan maksamaan useimmiten kryptovaluuttana, yleensä bitcoineina. Lunnaiden maksaminen bitcoineina vaikeuttaa rikollisten jäljittämistä. Uhria saatetaan myös uhkailla tietojen verkkoon vuotamisella, jos lunnaita ei makseta. [5.]

Troijalainen

Tämä hyökkäysmenetelmä on saanut nimensä vanhan legendan mukaan, jossa antiikin kreikkalaiset valtasivat Troijan kaupungin lahjoittamalla troijalaisille puisen hevosen, jonka sisälle mahtui piiloutumaan 40 miestä. Pahaa-aavistamattomat troijalaiset kuljettivat hevosen kaupungin muurien sisälle, jolloin piiloutuneet sotilaat päästivät loput joukoista sisään. Näin antiikin kreikkalaiset saivat vallattua kaupungin. Troijalaisen viruksen periaate on sama kuin tarinan Troijan hevosen. Haittaohjelma naamioituu tavalliseksi ohjelmistoksi tai tiedostoksi tavoitteena huijata käyttäjää päästämään se laitteeseen. Tarkoituksena on kaapata laite, kerätä arkaluonteisia tietoja tai vakoilla laitteen toimintaa. Troijalainen piilotetaan usein harmittomalta vaikuttavan sähköpostin liitteeksi tai ilmaiseksi ladattavaksi tiedostoksi. [4; 6.]

Vakoiluohjelma

Vakoiluohjelma voi tunkeutua laitteelle esimerkiksi haitallisen linkin tai sähköpostiliitteen kautta. Nimensä mukaisesti vakoiluohjelma vakoilee laitetta ja käyttäjän tekemisiä kyseisessä laitteessa. Useimmiten tavoitteena on varastaa arkaluonteisia tietoja, kuten luottokorttitietoja, verkkopankkitunnuksia, salasanoja ja käyttäjänimiä sekä tarkkailla käyttäjän toimia verkossa. [7.] Ohjelma saattaa myös tallentaa muun muassa laitteen näppäimistön painalluksia sekä ottaa kuvankaappauksia käyttäjän tietämättä [4].

2.2.2 Palvelunestohyökkäykset

Palvelunestohyökkäysten (engl. Denial-of-Service Attack eli DoS Attack) tai hajautettujen palvelunestohyökkäysten (engl. Distributed Denial-of-Service Attack eli DDoS Attack) kohteena ovat palvelimet, verkkosivustot tai muut verkkoresurssit. Hyökkäyksen tavoitteena kuormittaa näitä palveluja tai pahimmillaan kaataa ne. Kun esimerkiksi jokin verkkosivusto on palvelunestohyökkäyksen kohteena, sivuston käyttäjät eivät pääse kyseiselle sivustolle tai sivusto toimii poikkeuksellisen hitaasti. Hyökkäyksen kohteeksi joutuvat useimmiten suuret yritykset, verkkokaupat, sairaalat ja valtiolliset sivustot. Ne voivat aiheuttaa erittäin laajamittaisiakin katkoksia ja häiriöitä sekä usein myös merkittäviä taloudellisia seuraamuksia [4; 8.]

DoS- js DDoS-hyökkäykset ovat keskenään hyvin samantyyppisiä, muutamia eroavaisuuksia lukuun ottamatta. DoS-hyökkäys on peräisin yhdestä internetiin liitetystä laitteesta, ja se voidaan toteuttaa tähän tarkoitukseen kehitetyllä ohjelmalla, kun taas DDoS-hyökkäyksessä hyödynnetään botnettiä. Botnet on verkosto, joka muodostuu useammasta haittaohjelmalla kaapatusta laitteesta. Useampi laite yhdessä kykenee aiheuttamaan merkittävästi enemmän liikennettä verkossa ja sen seurauksena myös enemmän vahinkoa kuin yksi laite. Kun hyökkäys on peräisin useammasta laitteesta, sitä on myös vaikeampi torjua ja jäljittää. [8.]

2.2.3 Tietojenkalastelu

Tietojenkalastelua (engl. Phising) on monenlaista. Pääosin tässä hyökkäysmuodossa verkkorikollinen tekeytyy henkilöksi tai organisaatioksi, johon käyttäjä luottaa ja houkuttelee näin käyttäjää luovuttamaan arkaluonteisia tietoja kuten esimerkiksi luottokorttitietoja, käyttäjätunnuksia, salasanoja, sosiaaliturvatunnuksen tai huijaa lataamaan haittaohjelman laitteelle. Yleisin tapa kalastaa arvokkaita käyttäjätietoja on muun muassa kehotus avata esimerkiksi sähköpostin tai tekstiviestin kautta lähetetty liitetiedosto tai linkki ja täyttää lomake omilla henkilötiedoilla. Tällaiset viestit näyttävät ensivilkaisulla tulevan luotettavasta lähteestä, kuten pankista tai tunnetulta henkilöltä ja saattavat sisältää jonkinlaisen hälytysviestin, joka kertoo, että esimerkiksi käyttäjän pankkitilin kanssa on jokin ongelma ja pyytää näin käyttäjää tunnistautumaan käyttäjätunnuksellaan ja salasanallaan viestissä olevan linkin kautta. [9.] Todellisuudessa viestin takana on verkkorikollinen.

Tietojenkalasteluhuijaukset ovat valitettavasti yleistyneet yhä enemmän ja kokeneemmankin internetin käyttäjän on yhä vaikeampi tunnistaa näitä, sillä kalasteluviesteistä on tullut entistä parempia ja uskottavampia. Lisäksi ovat yleistyneet soittamalla tapahtuvat tietojenkalasteluhyökkäykset, jossa soittaja tekeytyy esimerkiksi poliisiksi tai muuksi ja pyytää vahvistamaan puheluun vastanneen henkilön tiedot kuten henkilötunnuksen.

2.2.4 SQL-injektiohyökkäykset

SQL-injektiohyökkäyksistä on puhuttu jo 1990-luvulta lähtien ja ne ovat edelleen suosituimpia hyökkäysmenetelmiä. Tässä hyökkääjä käyttää koostettua kyselykieltä (engl. Structured Query Language eli SQL) manipuloidakseen hyökkäyksen kohteena olevaa tietokantaa tavoitteenaan päästä käsiksi arkaluonteisiin tietoihin, kuten esimerkiksi potilastietoihin tai -kertomuksiin. SQL on kyselykieli, jolla voidaan ohjelmoinnissa lähettää pyyntö tietokannalle päästä käsiksi tallennettuihin tietoihin esimerkiksi muokatakseen ja poistaakseen niitä. Useimmat

verkkosovellukset ja -sivustot käyttävät SQL-pohjaista tietokantaa, joka tekee tästä hyökkäystavasta erityisen suositun ja samalla myös uhkaavimman.

Tyypillisesti esimerkiksi verkkolomake, johon syötetään sisäänkirjautumistiedot, on suunniteltu hyväksymään vain tietynlaisia ja -tyyppisiä tietoja, kuten nimi ja salasana. Käyttäjälle sallitaan sisäänkirjautuminen vain, jos verkkolomakkeeseen syötetyt tiedot täsmäävät tietokannassa oleviin tietoihin. Muussa tapauksessa pääsy evätään. Useimmiten ylimääräisten tietojen lisäämistä lomakkeille ei kuitenkaan voida estää ja tätä ongelmaa verkkorikolliset käyttävät hyväksi. Lomakkeen syöttökenttien kautta tietokantaan voidaan lähettää omia SQL-pyyntöjä ja samalla päästä käsiksi arkaluonteisiin tietoihin, jolloin niitä voidaan muokata, poistaa tai esimerkiksi julkaista muualla verkossa. Pahimmassa tapauksessa verkkorikollinen voi päästä koko organisaation yleiseen järjestelmään tietämättä käyttäjätietoja. [10.]

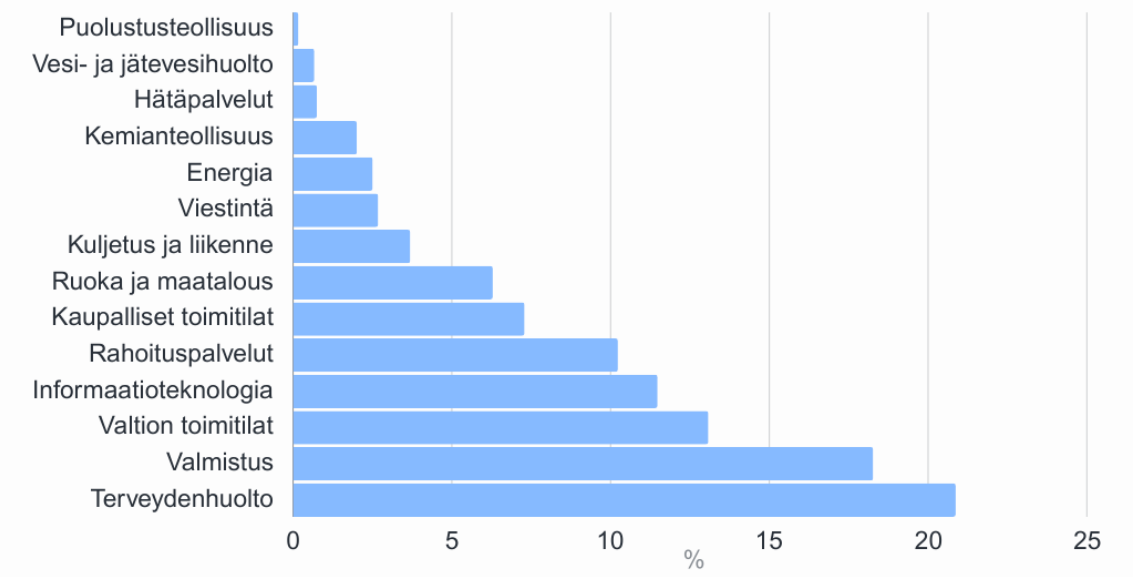
2.2.5 Käyttäjän manipulointi

Käyttäjän manipulointi (engl. Social Engineering) on huijaustaktiikka, jossa usein hyödynnetään ihmisten hyväntahtoisuutta ja inhimillisiä virheitä. Käyttäytymisen ja psykologian ymmärtäminen ovat tärkeimmässä roolissa tässä huijausmetodissa. Hyökkääjät varastavat rahaa, henkilötietoja, käyttäjätunnuksia ja muuta arkaluonteista tietoja manipuloimalla uhrejaan. Kohteina ovat sekä kokonaiset organisaatiot että yksityishenkilöt. Joskus tavoitteena on päästä käsiksi fyysiseen laitteeseen tai organisaation tiloihin.

Social engineering -hyökkäys alkaa yleensä kohteen tunnistamisella ja tiedonkeruulla, jonka jälkeen kohdetta usein lähestytään käyttäen väärennettyä henkilöllisyyttä ja keksittyä tarinaa. Hyökkäys toteutetaan, kun hyökkääjä on onnistunut saamaan uhrin luottamuksen. Onnistuneen hyökkäyksen jälkeen jäljet siivotaan. Social engineering -hyökkäyksiä voidaankin kutsua ihmisen hakkeroimiseksi, koska hyökkäys perustuu psykologiaan ja ihmisen manipulointiin. [11.] Kyber- ja tietoturvallisuuden heikoin lenkki onkin usein ihminen itse.

2.3 Kyberhyökkäyksiltä suojautuminen

Suojausmenetelmät ja niiden jatkuva kehittäminen ovat yhteiskunnan kannalta vähintäänkin yhtä merkittävässä roolissa kuin esimerkiksi poliisin työ kansalaisten suojelemisessa ja rikollisten nappaamisessa. Kun otetaan huomioon, miten monia laitteita sekä elintärkeitäkin järjestelmiä ja palveluita on kytketty verkkoon ilman aktiivista suojautumista, olisimme jatkuvassa vaarassa. Esimerkkinä kuvassa 1 näkyy, miten vuonna 2023 Yhdysvalloissa haittaohjelmahyökkäykset jakautuivat yhteiskunnan eri sektoreiden välillä. Eniten haittaohjelmahyökkäyksiä toteutettiin terveydenhuollon järjestelmiin ja palveluihin. [12.]



Kuva 1. Haittaohjelmahyökkäysten jakautuminen prosentteina yhteiskunnan eri sektoreiden välillä Yhdysvalloissa vuonna 2023 [12].

Kuva 1 kuvastaa hyvin sitä, miten tärkeää kyberturvallisuus yhteiskunnalle on. Mitä tapahtuisi, jos Suomen kaikki terveydenhuollon järjestelmät joutuisivat yhtäkkiä kyberhyökkäyksen kohteeksi ja palvelut lakkautuisivat? Emme voisi esimerkiksi ostaa reseptilääkkeitä apteekista. Tai pahimmassa tapauksessa emme edes huomaisi, kun muun muassa arkaluontoisia tietojamme levitettäisiin ympäri maailmaa.

2.3.1 Kybervalvonta

Kybervalvonta on merkittävä osa kyberturvallisuutta ja sen apuna käytetään monenlaisia menetelmiä, työkaluja ja ohjelmistoja. Yksi kybervalvonnan osa-alueista on verkonvalvonta, jonka tarkoituksena on tarkkailla tietoliikenteessä tapahtuvia poikkeamia sekä sen suorituskykyä. Poikkeaman syynä voi esimerkiksi olla käyttäjän virhe tai verkkorikollisen yritys päästä käsiksi verkkoliikenteeseen ja mahdollisesti vakoilla tai manipuloida sitä. [13, s. 2.] Verkkovalvonnan avulla voidaan havaita ja ehkäistä muun muassa DoS- ja DDoS-hyökkäyksiä, koska valvonnassa käytettävät työkalut tuottavat varoituksia ja hälytyksiä verkossa tapahtuvista muutoksista [14, s. 28].

Tietoturvatapahtumien vastaanottamisen lisäksi valvontaan kuuluu myös varoitusten ja hälytysten analysointi, lajittelu ja vastatoimet. Poikkeamien analysoinnin tarkoituksena on saada tietoa mahdollisesta hyökkäyksestä ja hyökkäyksen kohteena olevasta järjestelmästä sekä harkita vastatoimien tarvetta. Vastatoimiin voi kuulua esimerkiksi kohteena olevien järjestelmien suojaaminen, korjaaminen ja päivittäminen. [14, s. 29.]

Kybervalvontaan kuuluu myös fyysisten järjestelmien, kuten päätelaitteiden, reitittimien, palvelimien ja palomuurien valvonta. On huolehdittava muun muassa laite- ja ohjelmistopäivityksistä sekä huolto- ja korjaustoimenpiteistä. [14, s. 30.] Lisäksi esimerkiksi organisaation henkilökunnan kulunvalvonta on iso osa fyysistä kybervalvontaa.

2.3.2 Penetraatiotestaus

Kyberturvan valvonnan lisäksi toinen iso tekijä turvallisuuden varmistamiseksi on laitteiden, järjestelmien, ohjelmistojen sekä lisäksi ihmisten testaus. Tätä kutsutaan penetraatiotestaukseksi eli toisin sanoen suojattavaan tietojärjestelmään tunkeutuminen organisaation tai omistajan luvalla [14, s. 30]. Penetraatiotestaus on hakkerointia ja tätä voidaan kutsua myös termillä eettinen hakkerointi. Hakkerilla, jolla ei ole rikollisia, aatteellisia tai poliittisia aikeita mielessä kutsutaan

usein nimellä eettinen hakkeri tai valkohattuhakkeri (engl. Ethical Hacker ja White Hat Hacker). Penetraatiotestaajat käyttävät tietojärjestelmien ja ihmisten testaamisessa samoja työkaluja kuin muut hakkerit sekä verkkorikollisiksi luokitellut mustahattuhakkerit (engl. Black Hat Hacker).

Tietoturvayhtiö Kasperskyn mukaan penetraatiotestaajan menetelmiin ja työkaluihin kuuluu muun muassa:

1. Sosiaalinen manipulointi: Aiemmin mainittua social engineeringiä. Testaaja kerää tietoa kohdeorganisaatiosta, käyttää mahdollisesti kyseenalaisia, mutta laillisia toimia manipuloidakseen yhtä tai useampaa henkilöä organisaatiossa. Tavoitteena on saada henkilö luovuttamaan salassa pidettäviä tietoja, kuten kirjautumistietoja tai tietoteknisesti arvokkaiden laitteiden sijainteja.

2. Läpäisytestaus: tavoitteena löytää organisaation puolustuksen ja päätepisteiden heikkoudet.

3. Tiedustelu ja tutkimus: Tarkoituksena selvittää fyysisen ja tietoteknisen infrastruktuurin haavoittuvuudet. Tietoa pyritään saamaan mahdollisimman paljon laillisesti ilman, että mitään vahingoitetaan tai rikotaan.

4. Ohjelmointi: Testaaja luo verkkorikollisille aidolta vaikuttavia kohdeorganisaation järjestelmiä eli niin sanottuja hunajapurkkeja, joilla houkutellaan rikollinen sisälle organisaation järjestelmään. Näin rikollisen tekemisiä voidaan tarkkailla ja sen avulla selvittää kuinka todellisesta tietojärjestelmästä voitaisiin tehdä turvallisempi.

5. Digitaalisten ja fyysisten työkalujen käyttö: testaaja pyrkii pääsemään käsiksi organisaation tietojärjestelmiin käyttäen apuna muun muassa erilaisia haittaohjelmia. [15.]

Penetraatiotestauksen tavoitteena on siis selvittää, millaisia menetelmiä ja työkaluja todelliset verkkorikolliset voisivat käyttää päästäkseen sisälle

organisaation tietojärjestelmiin. Tämän perusteella organisaatio voi suojata laitteet ja järjestelmät entistä tehokkaammin sekä kehittää niistä turvallisempia.

3 Tekoäly

Tämän luvun tarkoituksena on esitellä pintapuolisesti tekoälyn (engl. Artificial Intelligence eli AI) toimintaa sekä siihen liittyviä käsitteitä. Tekoälyä on hyvin vaikea määritellä yksiselitteisesti, koska siihen liittyy monia erilaisia menetelmiä ja se saattaa myös tarkoittaa monia eri asioita, riippuen keneltä kysyy. Tässä raportissa keskitytään tekoälyyn ohjelmoituna, ihmisen aivotoimintaa simuloivana koneena ja niin sanottuna avustajana. Ensimmäisen kerran termiä ”artificial intelligence” käytti Stanfordin yliopiston tietojenkäsittelytieteen professori John McCarthy jo vuonna 1956 Darthmouth Collegen kesäseminaarissa [3, s. 25].

3.1 Määritelmä

Tekoälyn ajatellaan olevan tietokonejärjestelmä tai -ohjelma, jota voidaan opettaa toimimaan useimmiten ihmisen, mutta joskus myös muun biologisen organismin tavoin sekä reagoimaan erilaisiin ärsykkeisiin halutulla tavalla. Tekoäly muodostuu algoritmeista eli yksityiskohtaisista ohjeista, jotka perustuvat matemaattisiin kaavoihin. Algoritmien avulla tekoäly osaa tulkita muun muassa tekstiä, kuvia, videoita tai ääntä sekä lisäksi oppia näistä jotain. Opittua tietoa tekoäly käyttää hyväkseen esimerkiksi päätöksenteossa. [3, s. 28; 16.]

Yksinkertaistettuna tekoäly voidaan jakaa kahteen kategoriaan, heikkoon ja vahvaan tekoälyyn. Heikko tekoäly ei kykene ihmisen tapaiseen älykkääseen ajatteluun eikä sillä ole itsetietoisuutta. Se osaa suorittaa vain ihmisen määräämiä tehtäviä. Mitä tarkemmin tehtävä on rajattu, sitä paremmin tekoäly suoriutuu, yhtä hyvin tai jopa paremmin kuin ihminen. [16.]

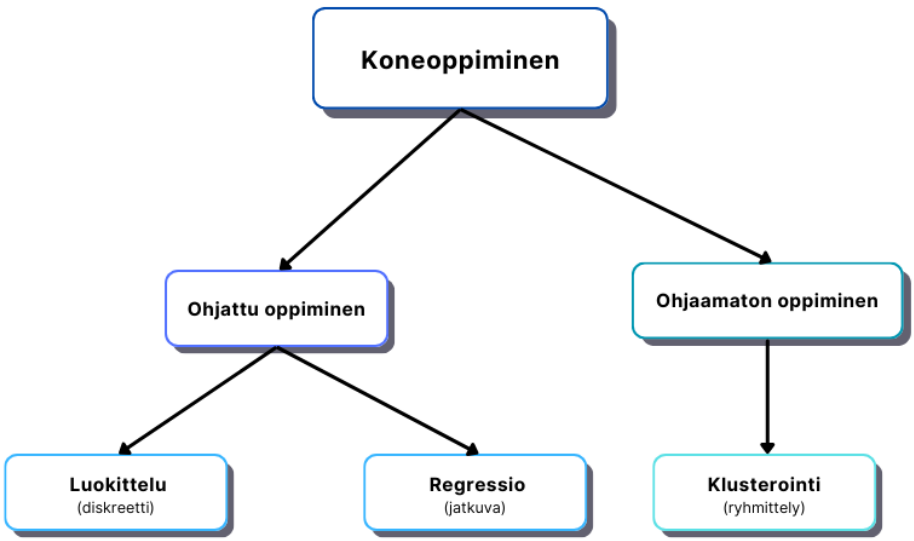
Vahvalla tekoälyllä tai supertekoälyllä on laajempi ymmärrys itsestään sekä ympärillä tapahtuvista asioista, ja se kykenee itsenäiseen ajatteluun ja ongelmanratkaisuun. Vahva tekoäly on lähimpänä ihmisen älykkyyttä ja sillä on oma arvomaailma. Tähän mennessä tällaista vahvaa tekoälyä ei kuitenkaan ole onnistuttu luomaan. [16; 17, s. 11.]

3.2 Yleiset käsitteet ja termit

Seuraavaksi käsitellään tekoälyyn liittyviä yleisiä käsitteitä ja termejä. Tähän osioon on kerätty tiivistetysti vain sellaisia termejä, joita käsitellään myöhemmin tässä raportissa tai mitkä liittyvät muulla tavoin olennaisesti tämän tutkielman tavoitteisiin.

Koneoppiminen

Koneoppiminen (engl. Machine Learning eli ML) kuuluu yhteen tekoälyn osa-alueisiin. Sen perusperiaate on saada ohjelmisto tai järjestelmä suoriutumaan entistä paremmin käyttäen hyväksi perustietoja ja mahdollisia käyttäjän toimia. Toisin sanoen se oppii toistuvista tapahtumista ilman, että ihminen erikseen opettaa sitä. Käyttäen monimutkaisia algoritmeja, koneoppivaa tekoälyä voidaan automatisoida tiedon tulkinnan ja koneen havainnointikyvyn laajentamisella. [3, s. 316.]

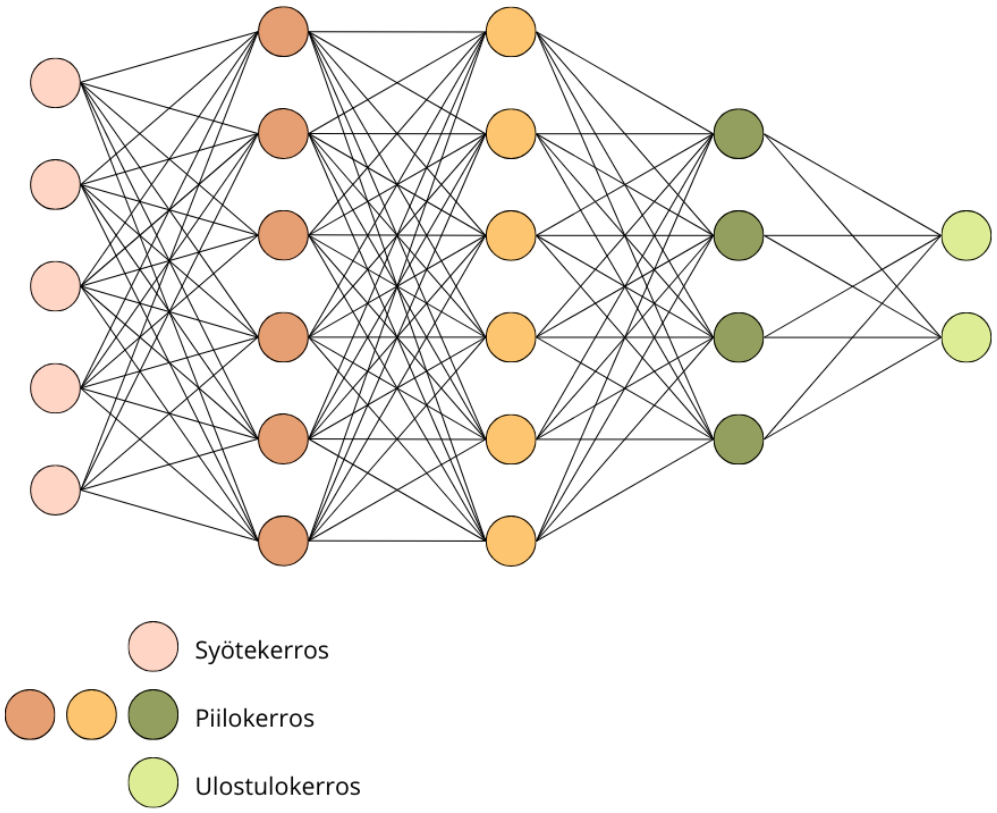


Kuva 2. Koneoppimisen jakautuminen ohjattuun ja ohjaamattomaan oppimiseen [3, s. 316].

Ohjatussa oppimistilanteessa koneelle opetetaan erilaisten tilanteiden tarkkailua ja huomioimista. Ohjaamattomassa oppimistilanteessa kone taas tunnistaa itse tilanteita ja ehdottaa niiden lisäämistä seurantaan. [3, s. 316.]

Syväoppiminen

Syväoppiminen (engl. Deep Learning) on koneoppimisen osa-alue, joka jäljittelee ohjelmallisella tavalla ihmisen aivojen toimintaa monikerroksisten neuroverkkojen avulla. Sitä käytetään muun muassa kuvan-, tekstin- ja äänentunnistuksessa. Se on tekniikka, joka kykenee parantamaan itseään ilman ihmisen valvontaa raakadatan perusteella, toisin kuin perinteinen koneoppiminen, joka edellyttää manuaalista piirteiden lajittelua. Syvät neuroverkot saattavat sisältää miljoonia ohjattavia muuttujia ja jokaisella piilokerroksella on oma tehtävänsä. Siitä syystä koneen opettaminen syväoppimismenetelmillä vaatii valtavaa määrää opetusdataa. [3, s. 318; 18.]



Kuva 3. Syvä neuroverkko [3, s. 318].

Kuva 2 on yksinkertaistettu havainnekuva syväoppimistekniikan toiminnasta. Piilokerroksella tarkoitetaan neuroverkkoon kuuluvaa solmusarjaa, jonka tarkoituksena on muuttaa syötekerroksen tulosignaalit ulostulostulosignaaleiksi matemaattisten algoritmien avulla [19].

Big data

Big data (suom. massadata) on jatkuvasti kasvavan, jäsennellyn ja ei-jäsennellyn tiedon, kuten esimerkiksi tekstiä, kuvia, videoita ja äänitteitä sisältävien tietojoukkojen kokoamista, varastointia ja hyödyntämistä. Massadataa syntyy esimerkiksi verkkosivustojen ja laitteiden käyttö- ja toimintatiedoista, sää- ja navigointidatasta sekä terveydenhuollon tiedoista. [3, s. 314.]

Vahvistusoppiminen

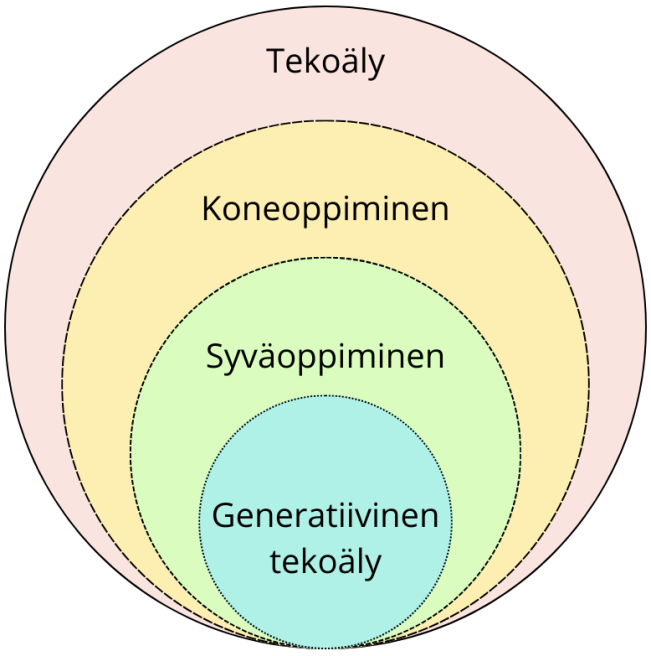
Vahvistusoppimisella (engl. Reinforcement Learning eli RL) tarkoitetaan koneoppimistekniikka, jota käytetään tietokoneohjelmistojen ja -järjestelmien kouluttamisessa opettaakseen niitä tekemään oikeita päätöksiä toivottujen tulosten saavuttamiseksi. Tässä oppimistekniikassa tekoälyjärjestelmälle ei anneta tehtävien suorittamiseen tarkkoja ohjeita tai dataa, vaan se oppii toiminnan perusteella saadusta positiivisesta tai negatiivisesta palautteesta. Näin järjestelmä oppii ajan myötä ihanteelliset toimintatavat parhaimman lopputuloksen saavuttamiseksi. [20; 21.]

Laaja kielimalli

Laaja kielimalli (engl. Large Language Model eli LLM) perustuu syväoppimiseen. Se on kehitetty tunnistamaan ja tuottamaan ihmiskieltä tai muunlaista monimutkaista tietoa. Laajan kielimallin koulutuksessa käytetään yleensä massiivisia tietokantoja, jotka sisältävät jopa miljoonien gigatavujen edestä tekstiä. Koulutuksen aikana mallin on ennustettava seuraava sana tai lauseen osa annetun sisällön perusteella. Biljoonien lauseiden analysoinnin jälkeen se oppii tunnistamaan kieliopilliset rakenteet, sanastot ja sanojen merkitykset vaihtelevissa yhteyksissä sekä luoda omia lauseitaan. [22; 23.]

Generatiivinen tekoäly

Generatiivisella tekoälyllä (engl. Generative AI eli GenAI) tarkoitetaan kehittynyttä syväoppimisen osa-aluetta, jonka tarkoitus on pystyä luomaan uutta sisältöä, joka on samankaltaista kuin alkuperäinen data, jolla se on koulutettu. Se voi olla melkein mitä tahansa kuvista ja musiikista kolmiulotteisiin malleihin. Generatiivista tekoälyä voi kouluttaa oppimaan ja ymmärtämään muun muassa ihmiskieltä, ohjelmointikieliä, kemiaa, biologiaa ja monia muita edistyneitä aiheita. Sitä käytetään esimerkiksi kuvantunnistuksessa, puheentunnistuksessa, luonnollisen kielen käsittelyssä, automaattisten käännösten sekä jopa taideteoksien luomisessa. [24; 25; 26.] Suosittuja generatiivista tekoälyä käyttävistä työkaluista on muun muassa ChatGPT ja Dall-E.



Kuva 4. Generatiivinen tekoäly on syväoppimisen osa-alue [27].

Deepfake

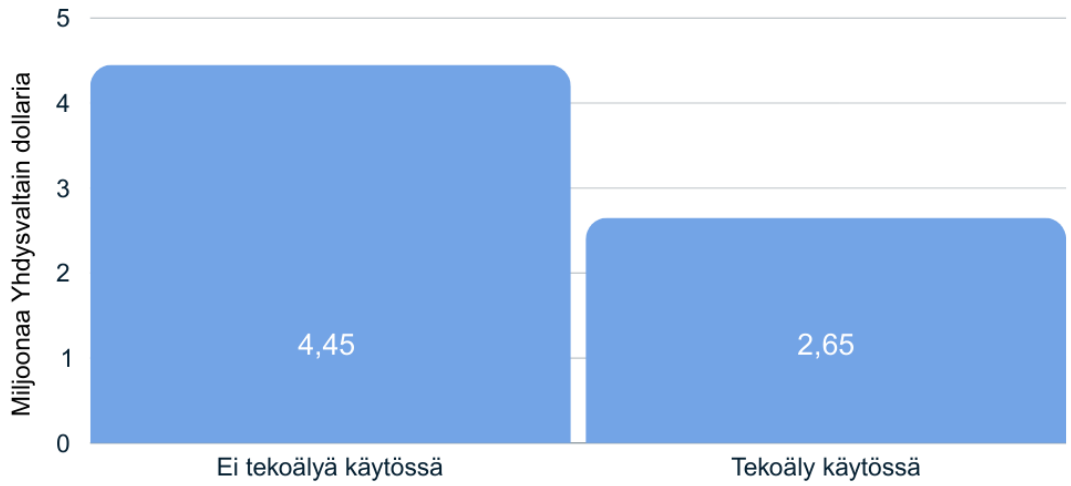
Deepfake (suom. syväväärennös) on syväoppimismallin avulla luotua ja aidolta vaikuttavaa kuva-, video- tai ääniaineistoa. Syväväärennöksiä voidaan tuottaa sulauttamalla tai muokkaamalla olemassa olevia kuvia sekä lyhyitäkin ääni- tai

videonäytteitä. [19; 28.] Monimutkaisten syväväärennösten tuottamisessa käytetään useimmiten kahta algoritmia, joista yksi on koulutettu kehittämään parhaita mahdollisia väärennöksiä ja toinen havaitsemaan näitä väärennöksiä. Kun algoritmit asetetaan toimimaan vastakkain, voidaan tuottaa erittäin aidolta vaikuttavia jäljennöksiä, joita ihmissilmän tai -korvan on vaikea tunnistaa. [19.]

3.3 Hyödyt ja ongelmat

Vaikka tekoälystä onkin tullut ihmiselle mahtava apulainen, liittyy sen käyttöön myös monia haittoja ja haasteita. Tekoäly osittain vähentää inhimillisiä virheitä, erityisesti rutiininomaisissa tehtävissä. Toisaalta vaikka ihmisen aivotoimintaa yritetään jäljitellä ja ymmärtää matemaattisten kaavojen avulla, on tekoälyn vaikeaa kehittää itselleen esimerkiksi omaa identiteettiä. Tekoälyllä ei ole omaa tahtoa eikä omatuntoa, eikä se ei kykene itsenäisesti erottamaan esimerkiksi moraalisesti oikeaa väärästä, vaan se toimii niillä tiedoilla, jotka sille on ihminen antanut. [3.] Tämä voi riippuen tilanteesta olla joko hyödyllistä tai haitallista.

Jotta tekoälystä saataisiin kaikki hyöty irti, kannattaa tekoäly esimerkiksi kyberturvallisuusjärjestelmässä integroida järjestelmän kaikille tasoille, jolloin hyökkäyksiltä suojautuminen ja ennaltaehkäiseminen voidaan suorittaa kokonaisuutena eikä vain yksittäisinä toimenpiteinä [3, s. 308]. Tekoälyjärjestelmien käyttöönotto saattaa tosin olla melko kallistakin. Toisaalta taas pitkällä aikavälillä sijoitus saattaa tulla hyvinkin edulliseksi ja alentaa kyberturvallisuuden kehittämiseen liittyviä kokonaiskustannuksia. Kuvassa 4 näkyy, miten tekoälyn hyödyntäminen kyberturvallisuudessa on alentanut tietomurroista aiheutuneita kustannuksia 1,8 miljoonalla Yhdysvaltain dollarilla vuonna 2023 [29].



Kuva 5. Tietomurroista aiheutuneet keskimääräiset kustannukset maailmanlaajuisesti vuonna 2023 [29].

3.3.1 Edut

Tekoälyn tarkoituksena on avustaa ja nopeuttaa ihmisen työskentelyä muun muassa käsittelemällä erittäin suuria määriä tietoa. Tiedon määrä voi olla niin suurta, että ilman tekoälyn avustusta, ihmisaivoilla menisi vuosia käsitellä sitä. [16.] Lisäksi massadatan hallitseminen ja analysoiminen on perinteisillä tietokantatyökaluillakin joko mahdotonta tai erittäin haastavaa [3, s. 314].

Kyberturvallisuudessa tekoälyn käyttö suurien tietomäärien analysoinnissa on erityisen hyödyllistä, koska näin voidaan tunnistaa uhkia, jotka eivät ihmisanalyytikolle ole välittömän ilmeisiä. Lisäksi tekoälyä voidaan käyttää rutiinitehtävien, kuten esimerkiksi järjestelmien korjaamisen ja päivittämisen automatisoinnissa. Usein myös raporttien ja hälytysten tuottaminen voidaan jättää tekoälyn tehtäväksi. [30.] On onnistuttu osoittamaan, että väärien hälytysten määrää voidaan vähentää merkittävästi ottamalla tekoälyn mukaan kybervalvontaan [31]. Näin monimutkaisemmat tehtävät jätetään ammattilaisille ja työskentely tehostuu. Tämä parantaa muun muassa tietoturvauhkien havaitsemista ja tunnistamista sekä nopeuttaa reagoimista ja vastatoimiin ryhtymistä. [30.]

Tekoälyä voidaan käyttää myös kyberhyökkäysten ennustamiseen. Ennustaminen tapahtuu historiallisten tietojen avulla sekä analysoimalla käyttäjien verkkokäyttäytymismalleja. Se huomaa pienetkin muutokset käyttäytymismalleissa ja ihmissilmälle mitättömiltä vaikuttavia yhteyksiä tapahtumien välillä. Tekoäly oppii näistä tiedoista ja parantaa samalla uhkien tunnistamis- ja ennustamiskykyään ajan myötä yhä tarkemmaksi. Tällaisen tekoälyjärjestelmän avulla voidaan ryhtyä ennaltaehkäiseviin toimenpiteisiin ennen kuin hyökkäys on ehtinyt edes tapahtua ja minimoida mahdolliset vahingot tai jopa estää hyökkäys kokonaan. [31; 32.]

3.3.2 Haasteet

Tekoälyn kehitykseen liittyy myös useita haasteita. Samalla, kun kybervalvonta tehostuu tekoälyn ansiosta, käyttävät myös verkkorikolliset tekoälyä hyödyksi kehittääkseen yhä vaarallisempia ja monimutkaisempia kyberuhkia. Sitä voidaan käyttää muun muassa realististen kalasteluviestien luomiseen, haittaohjelmien kehittämiseen ja levittämiseen sekä uskottavien syväväärennösten luomiseen. Samaan aikaan, kun tekoälyä kehitetään päivä päivältä paremmaksi, löytävät verkkorikolliset uusia luovia tapoja tekoälyn hyödyntämiseen kyberhyökkäyksissä. Kyberturvallisuuden ammattilaisten ja verkkorikollisten välillä tämä saattaa johtaa niin kutsuttuun kilpavarusteluun. [30.]

Jotta tekoäly voisi työskennellä mahdollisimman sujuvasti ihmisen kanssa, sen täytyisi ymmärtää ihmisen fysiikkaa sekä käyttäytymistä ja eleitä. Sen olisi osattava tulkita näitä ominaisuuksia ja toimia niiden perusteella. Lisäksi ryhmädynamiikan ja vuorovaikutuksen ymmärtäminen olisi tekoälyn ja ihmisen väliselle toimivalle yhteistyölle hyödyksi. [3, s. 34.] Tekoäly on kuitenkin vain niin hyvä kuin tiedot, joilla sitä on koulutettu [16]. Vaikka tekoälyn hyödyistä puhuttaessa mainitaankin usein inhimillisen virheen vähentyminen, täytyy muistaa, että tekoälyn algoritmeineen on kehittänyt ihminen. Tämän takia esimerkiksi tekoälyn koulutuksessa käytetyt tiedot saattavat olla puolueellisia, epätäydellisiä tai jopa virheellisiä [26].

Aiemmin mainittiin, että tekoälyn käyttö kyberturvallisuudessa tehostaisi kybervalvontaa muun muassa rutiinitehtävien automatisoinnilla. Tämä voi koitua ongelmaksi, kun tekoälyn annetaan tehdä päätöksiä ilman ihmisen valvontaa. Turvallisen, tekoälyä hyödyntävän automatisoidun järjestelmän edellytyksenä on, että päätöksentekoprosesseissa käytetään edelleen ihmisälyä. [30.]

4 Tekoälypohjainen penetraatiotestaus

Seuraavaksi tarkastellaan kolmea mahdollista penetraatiotestauksessa käytettävää tekoälypohjaista hyökkäys- ja tietojenkalastelutapaa. Kuten aiemmin mainittu, penetraatiotestaajat käyttävät työssään samanlaisia työkaluja ja tekniikoita kuin verkkorikolliset ja hakkerit. Termi ”hyökkäys” voi hämäävästi luoda ajatuksen verkkorikollisen suorittamasta hyökkäyksestä. Kuitenkin tässä osiossa kyseistä termiä käytetään viittaamaan tunkeutumiseen testausmielessä eikä näitä menetelmiä suositella missään nimessä käytettäväksi rikollisessa toiminnassa. Minilex sanoo tietomurroista seuraavasti:

> Tietomurrosta voidaan tuomita sakkorangaistukseen tai enintään kahden vuoden pituiseen vankeuteen. Tietomurron yritys on säädetty rangaistavaksi. Mikäli tietomurto tehdään erityisen suunnitelmallisesti, tai osana järjestäytyneen rikollisryhmän toimintaa, on kyseessä laissa erikseen säädetty törkeä tekomuoto. [33.]

4.1 Hunajapurkki

Hunajapurkin (engl. Honeypot) ideana on jäljitellä organisaation oikeaa laitetta tai järjestelmää eli toimia virtuaaliansana verkkorikollisille. Tavoitteena on välttää rikollisen tunkeutumista oikeaan järjestelmään. Se on luotu tarkoituksella haavoittuvaiseksi ja sitä valvotaan tarkasti. Hunajapurkki pyritään luomaan aina niin, ettei se olisi suoraan konfiguroituna muuhun organisaation verkkoon tai toisiin laitteisiin, minimoidakseen verkkorikollisten mahdollisuudet päästä käsiksi muihin järjestelmiin. Ansan avulla voidaan tarkkailla verkkorikollisen toimintaa järjestelmässä ja kerätä tunkeutujasta esimerkiksi seuraavia tietoja:

- IP-osoite ja sijainti
- tunkeutumistapa
- varastetut tiedot
- käytetyt salasanat.

Hunajapurkkeja luodaan erilaisiin tarkoituksiin. Haittaohjelma- ja tietokantahyökkäysten tutkiminen ja valvominen ovat yleisempiä kohteita, jossa

hunajapurkkeja käytetään. [34.] Perinteisten staattisten hunajapurkkien käyttöön liittyy kuitenkin monia rajoituksia. Seuraavaksi käsitelläänkin tekoälyn roolia hunajapurkkijärjestelmissä.

Tekoäly hunajapurkissa

Tekoälyä hyödyntävä hunajapurkki mukautuu ja kehittyy itsenäisesti oppimansa perusteella, toisin kuin perinteinen hunajapurkki. Tämä on tärkeää, koska jatkuvan tekoälyn kehityksen myötä myös verkkorikolliset kehittävät itseään ja taitojaan yhä paremmaksi sekä luovat työkaluja, jotka tunnistavat hunajapurkkeja. Hunajapurkkien älyllistäminen on parantanut näiden työkalujen oppimis- ja tunnistamisominaisuuksia. Vahvistusoppiminen on osoittautunut yhdeksi tehokkaimmaksi tekoälyn opetusmenetelmäksi, kun puhutaan tekoälyavusteisen hunajapurkin hallinta- ja päätöksentekotaidoista. Tämä auttaa luomaan mahdollisimman tarkkaan oikeaa järjestelmää simuloivan ympäristön, joka lisää hunajapurkin houkuttelevuutta.

Uuden sukupolven hunajapurkkien oppiessa ja kehittyessä ne hidastavat verkkorikollisten tunkeutumista järjestelmään ja levittävät muun muassa ennustusmallien avulla uudenlaisia ansoja, joiden avulla tunkeutujista ja heidän toiminnastaan saadaan yhä tarkempia tietoja. Tekoäly oppii hakkereiden käyttäytymismalleista, ja tämän ansiosta se pystyy muovautumaan aina jokaiseen tilanteeseen sopivammaksi houkutellakseen verkkorikollisia esimerkiksi muuttamalla järjestelmän konfiguraatiota tai manipuloimalla tiedostoja reaaliajassa. Tekoälyn ansiosta jopa oikea organisaation käytössä oleva järjestelmä voidaan muuttaa hunajapurkiksi, mikä saattaa pakottaa verkkorikolliset ansaan. Kun tekoäly havaitsee tunkeutujan, se kykenee keräämään ja analysoimaan tietoja sekä reagoimaan nopeammin kuin perinteinen hunajapurkki. Näin tekoälypohjainen dynaaminen hunajapurkki voi antaa välittömän hälytyksen tai jopa ryhtyä itse vastatoimiin esimerkiksi estämällä hakkerin pääsyn tiettyihin resursseihin. [35; 36.]

4.2 Uuden sukupolven chatbot-hyökkäykset ja syväväärennökset

Tekoälypohjaisiksi keskusteluboteiksi (engl. Chatbot) kutsutaan tietokoneohjelmaa, jonka tarkoituksena on käydä keskustelua ihmisen kanssa. Koska ne on useimmiten ohjelmoitu ymmärtämään luonnollista kieltä, ne ymmärtävät käyttäjän kysymyksiä sekä osaavat vastata niihin automaattisesti ja ihmismäisesti. Mikäli chatbot käyttää generatiivista tekoälyä ja laajaa kielimallia, osaa se automaattisten vastausten lisäksi luoda myös kokonaan uutta sisältöä, kuten korkealaatuisia tekstejä, kuvia, ääniä ja videoita. [37.]

Koska generatiivisesta tekoälystä on tullut ihmiselle oiva aputyökalu, on se myös syy miksi erityisesti käyttäjän manipuloinnin ja tietojenkalasteluhyökkäysten suosio chatbottia apuna käyttäen on noussut merkittävästi parin vuoden sisällä [38].

4.2.1 Chatbot-hyökkäykset

Koska tekoäly kykenee analysoimaan valtavia tietomääriä lyhyessä ajassa ja oppimaan käyttäjän käyttäytymishistoriasta, voidaan chatbottien avulla luoda uskottavia, yksilöllisesti kohdennettuja kalasteluviestejä. Lisäksi tekoälyn avulla voidaan ohittaa organisaation perinteiset turvatoimet analysoimalla ja jäljittelemällä aitoja viestintämalleja, jolloin vahingollisen sisällön havaitsemisesta tulee haastavampaa. Näin käyttäjä saattaa helposti luulla viestien tulevan luotettavalta taholta ja luovuttaa verkkorikollisille arkaluontoisia tietoja. Vaihtoehtoisesti kalasteluviestin lähettäjä saattaa esiintyä esimerkiksi esihenkilönä ja pyytää käyttäjää suorittamaan jonkun tehtävän, joka edistää verkkorikollisten tunkeutumista organisaation tietojärjestelmiin jotakin muuta reittiä. [39.]

Vielä muutama vuosi sitten kalasteluviestit oli helppo tunnistaa, sillä jos ne olivat kirjoitettu jollakin muulla kielellä kuin englanniksi, ne olivat usein täynnä kirjoitusvirheitä tai muuten kieliopillisesti huonosti muotoiltu. Nykyään tämä ongelma on ratkaistu laajaa kielimallia käyttävillä chatboteilla, joita voidaan pyytää kirjoittaa tekstejä lähes millä kielellä tahansa ja vieläpä kieliopillisesti oikein.

Näin saadaan myös huijaussivustoista, jotka yrittävät jäljitellä jotakin oikeaa, kuten esimerkiksi valtion verkkosivua, aidon näköisiä. [38.]

Generatiivista tekoälyä voidaan käyttää myös muuhun kuin uuden sisällön luomiseen. Sille voidaan antaa myös esimerkiksi valmis tekstisyöte, jota se voi yhdistellä, tiivistää tai muulla tavoin muuttaa. Näin voidaan helpommin kerätä esimerkiksi hyödyllistä tietoa kohdeorganisaatiosta tai siellä työskentelevistä henkilöistä, ja käyttää näitä tietoja hyödyksi sosiaalisessa manipuloinnissa. [40.]

4.2.2 Deepfake-huijaukset

Syväväärennöksillä voidaan luoda entistä uskottavampia huijauksia esimerkiksi puheluiden tai videoiden avulla. Yksi uusimmista ja edistyneimmistä huijaustavoista on uhrin tunteman henkilön, useimmiten erittäin läheisen, puhelinnumeron kaappaaminen, jonka jälkeen kyseisestä numerosta soitetaan uhrille ja pyydetään esimerkiksi lähettämään pikaisesti rahaa bussilippuun tai taksiin päästäkseen kotiin. Uhrin on vaikea epäillä puhelua huijaukseksi, koska se tulee tutusta puhelinnumerosta ja puhuja kuulostaa numeron alkuperäiseltä käyttäjältä, sillä ääni on luotu tekoälyn avulla. Pahimmassa tapauksessa voidaan jäljitellä myös henkilön kasvoja ja soittaa näin videopuhelu, jonka avulla saadaan tilanne näyttämään entistä uskottavammalta. Tämä on yksi tehokkaimmista uhrin manipulointikeinoista, koska tässä hyödynnetään henkilöiden tunnesidettä ja painostetaan uhria kiireellä, jolloin hän ei välttämättä kerkeä huomata puhelua huijaukseksi.

Jotta henkilön äänestä ja kasvoista voidaan luoda syväväärennöksiä, tarvitaan näistä näytteitä esimerkiksi äänitteistä tai videoista, jossa kyseinen henkilö puhuu tai valokuvista, joissa esiintyy. Tämän takia soittaja saattaa esiintyä usein myös julkisuuden henkilönä tai poliitikkona, koska heistä on saatavilla verkossa paljon näytemateriaalia. [41; 42.]

4.3 Ohjelmakoodin luominen ja analysointi

Mahdollisimman turvallisen ohjelmakoodin kirjoittaminen luo kyberturvallisuudelle kehykset. Yksi yleinen keino tunkeutua tietojärjestelmiin on nimenomaan hyödyntää ohjelmistokoodin haavoittuvuuksia. Tämä voi tapahtua joko hyödyntämällä jo entuudestaan tiedossa olevia tietoturvahaavoittuvuuksia, joita ei ole korjattu tai yrittää löytää ja hyväksikäyttää uusia haavoittuvuuksia ohjelmistokoodista.

Penetraatiotestaaja voi tässäkin käyttää apunaan tekoälyä. Tekoäly kykenee analysoimaan ohjelmakoodia ja muunlaista dataa nopeammin kuin ihminen. Koneoppimisalgoritmeja käyttävillä staattisen koodin analysointityökaluilla voidaan muun muassa skannata lähdekooditiedostoja. Näiden työkalujen avulla voidaan havaita yleisiä koodivirhemalleja ja -poikkeavuuksia, jotka mahdollistavat esimerkiksi SQL-injektiohyökkäyksiä sekä monia muita tunkeutumistapoja. Analysointityökalut mahdollistava heikkouksien tunnistamisen jo ennen varsinaista koodin käyttöönottoa. Lisäksi tekoäly voi tarjota automaattisesti myös korjausehdotuksia, mikä helpottaa ja nopeuttaa sekä koodin kehittäjän, että kyberturvaammattilaisten työtä.

Tekoälypohjaisten työkalujen avulla voidaan luoda myös suojattuja koodipätkiä useilla ohjelmointikielillä perinteisissä ohjelmointitehtävissä esimerkiksi käyttäjän todennusta sekä tietojen vahvistamista ja kryptausta varten. Lisäksi voidaan analysoida kolmansien osapuolten koodikirjastoja ja ehdottaa vaihtoehtoisia kokoonpanoja riskien minimoimiseksi. Penetraatiotestaaja voi tekoälyä hyödyntäen luoda myös testihaittaohjelmia, jotka jäljittelevät oikeita verkkorikollisten käyttämiä haittaohjelmia. [40; 43.]

Tarpeeksi suuritehoisen, tekoälyä hyödyntävän ohjelmakoodin analysointityökalun luominen on kuitenkin toistaiseksi hyvin työlästä. Syynä on tekoälymallien kouluttamiseen käytettävän opetusdatan pieni määrä ja laatu. Jotta tekoäly kykenisi tehokkaampaan koodianalyysiin, pitäisi sen opetusmateriaalien sisältöä laajentaa merkittävästi. Ongelmaksi koituu kuitenkin se, että uudenlaisia ja

entistä monimutkaisempia haavoittuvuuksia löytyy nopeammin kuin niitä kyetään lisäämään tekoälyn koulutusdataan. Lisäksi jo käytössä olevia syväoppimismalleja pitäisi kehittää paremmiksi tai luoda kokonaan uusia tekniikoita, koska tässä jää vielä toistaiseksi generatiivinenkin tekoäly jälkeen. [40.]

5 Yksinkertainen honeypot ChatGPT:n avulla

ChatGPT (engl. Chat Generative Pre-Trained Transformer) on yksi suosituimmista generatiivista tekoälyä hyödyntävistä chatboteista [44]. Näyttääkseen miten helppoa on käyttää chatbottia ohjelmakoodin luomiseen, pyydettiin ChatGPT:tä luomaan Pythonilla yksinkertainen tekoälyä hyödyntävä hunajapurkki, joka tunnistaa haitallisen toiminnan ja tallentaa nämä toiminnot sekä tunkeutujan IP-osoitteen erilliseen tiedostoon. Seuraava koodipätkä on ChatGPT:n luoma ensimmäinen versio.

```
import logging
from twisted.internet import reactor, protocol

logging.basicConfig(filename='honeypot.log', level=log-
ging.INFO, format='%(asctime)s - %(message)s')

class HoneyPotProtocol(protocol.Protocol):
    def connectionMade(self):
        print("Connection from:", self.transport.client)

    def dataReceived(self, data):
        print("Received:", data)
        if self.isMalicious(data):
            print("Malicious activity detected! Log-
ging...")
            logging.info("Malicious activity detected from
{}: {}".format(self.transport.client[0], data))
        else:
            print("No malicious activity detected.")

    def isMalicious(self, data):
        if b'hack' in data.lower():
            return True
        else:
            return False

class HoneyPotFactory(protocol.Factory):
    def buildProtocol(self, addr):
        return HoneyPotProtocol()

def main():
    port = 22
    reactor.listenTCP(port, HoneyPotFactory())
```

```
    print("Honeypot running on port", port)
    reactor.run()

if __name__ == "__main__":
    main()
```

Esimerkkikoodi 1. Ensimmäinen versio hunajapurkista [45; 46; 47].

Esimerkkikoodissa 1 käytetään Twisted-kehystä, joka kykenee käsittelemään samanaikaisesti useita verkkotapahtumia asynkronisesti. Honeypot kuuntelee saapuvia yhteyksiä, tässä tapauksessa portilla 22 ja kirjaa havaitun haitallisen toiminnan lokitiedostoon. `HoneyPotProtocol`-luokka vastaa yksittäisen TCP-yhteyden käsittelystä. Se kirjaa ylös IP-osoitteen ja tarkistaa avainsanan `hack` avulla, onko vastaanotettu data haitallista. Funktio `main` asettaa honeypotin kuuntelemaan ennalta määritettyä porttia ja käynnistää toiminnon, joka vastaanottaa ja käsittelee yhteyksiä.

TCP (engl. Transmission Control Protocol) huolehtii datan siirtämisestä internetissä laitteelta verkkopalvelimelle. Sen avulla muodostetaan yhteys lähettäjän ja vastaanottajan välille sekä varmistetaan, että tieto saapuu perille alkuperäisenä. [48.] IP (engl. Internet Protocol) on sääntöjoukko, jota käytetään säätelemään verkossa lähetetyn datan rakennetta [49].

Seuraavaksi pyydettiin ChatGPT:tä luomaan yksinkertainen esimerkkidataa sisältävä tekoälymalli verkkoliikenteen haitallisen toiminnan havaitsemiseksi.

```
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

def generate_example_data(num_samples):
    X = []
    y = []

    for _ in range(num_samples // 2):
        data = "This is a normal network packet."
        X.append([len(data), data.lower().count("mal-
ware"), data.lower().count("exploit"), data.lo-
wer().count("attack")])
```

```
            y.append(0)

        for _ in range(num_samples // 2):
            data = "This is a malicious network packet contai-
ning malware and exploit."
            X.append([len(data), data.lower().count("mal-
ware"), data.lower().count("exploit"), data.lo-
wer().count("attack")])
            y.append(1)

        return np.array(X), np.array(y)

X, y = generate_example_data(1000)
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

clf = RandomForestClassifier(n_estimators=100, ran-
dom_state=42)
clf.fit(X_train, y_train)

y_pred_train = clf.predict(X_train)
train_accuracy = accuracy_score(y_train, y_pred_train)
print("Training accuracy:", train_accuracy)

y_pred_test = clf.predict(X_test)
test_accuracy = accuracy_score(y_test, y_pred_test)
print("Test accuracy:", test_accuracy)

import joblib
joblib.dump(clf, 'malicious_detection_model.joblib')
```

Esimerkkikoodi 2. Esimerkkidataa hyödyntävä tekoälymalli [45; 46; 50; 51].

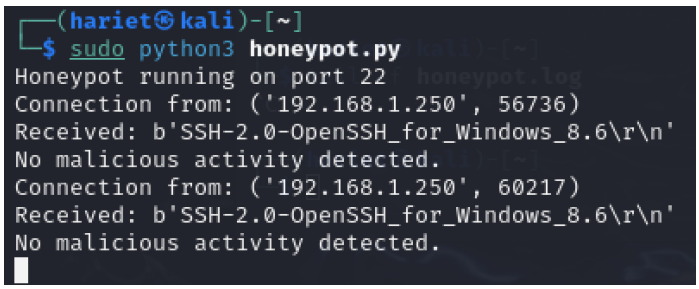
Malli käyttää scikit-learn-kirjastoa koneoppimisen toiminnallisuuksien toteuttamiseen muun muassa jakamalla datan koulutus- ja testisetteihin, satunnaismetsämallin luomiseen ja kouluttamiseen sekä suorituskyvyn arviointiin koulutus- ja testidataan nähden. Funktio `generate_example_data` luo esimerkkidatan kouluttamista varten ja määrittää kuinka monta näytettä otetaan. Data jaetaan ei-haitalliseksi ja haitalliseksi, jonka jälkeen malli koulutetaan käyttämällä koulutusdataa. Koulutusdatan ennustettu tarkkuus tulostetaan ja malli tallennetaan tiedostoon `malicious_detection_model.joblib`.

```
import logging
import numpy as np
from twisted.internet import reactor, protocol
import joblib

def load_model():
    return joblib.load('malicious_detection_model.joblib')

def extract_features(data):
    features = [len(data), data.lower().count(b"malware"),
data.lower().count(b"exploit"), data.lower().count(b"at-
tack")]
    return np.array(features).reshape(1, -1)

class HoneyPotProtocol(protocol.Protocol):
    def __init__(self):
        self.model = load_model()

    def connectionMade(self):
        print("Connection from:", self.transport.client)

    def dataReceived(self, data):
        print("Received:", data)
        if self.isMalicious(data):
            print("Malicious activity detected! Log-
ging...")
            logging.info("Malicious activity detected from
{}: {}".format(self.transport.client[0], data))
        else:
            print("No malicious activity detected.")

    def isMalicious(self, data):
        features = extract_features(data)

        if self.model:
            prediction = self.model.predict(features)
            if prediction > 0.5:
                return True

        return False

class HoneyPotFactory(protocol.Factory):
    def buildProtocol(self, addr):
        return HoneyPotProtocol()

def main():
```

```
    port = 22  # SSH port, change this to any other port
you want to emulate
    reactor.listenTCP(port, HoneyPotFactory())
    print("Honeypot running on port", port)
    reactor.run()

if __name__ == "__main__":
    main()
```

Esimerkkikoodi 3. Lopullinen hunajapurkki, joka hyödyntää yksinkertaista tekoälymallia [45; 46; 50; 51; 52].

Yllä oleva koodi on lopullinen hunajapurkki, joka hyödyntää aiemmin luotua yksinkertaista koneoppimismallia. Ensin ladataan ennalta koulutettu koneoppimismalli tiedostosta `malicious_detection_model.joblib`. Funktio `extract_features(data)` käy saapuvan datan perusteella läpi ominaisuudet, jotka muodostavat koneoppimismallin syötteen. Ominaisuudet sisältävät datan pituuden sekä sanojen ”malware”, ”exploit” ja ”attack” esiintymien lukumäärät. Kyseisiä sanoja ei luonnollisestikaan kannata välttämättä käyttää oikeassa hunajapotissa, koska harva verkkorikollinen nimeää esimerkiksi haittatiedostojaan malwareksi. Luokka `HoneyPotProtocol` käsittelee saapuvaa dataa ja sen ominaisuuksia koneoppimismallin avulla. Tämän jälkeen se ennustaa datan haitallisuuden sekä kirjaa havainnot.



Kuva 6. Kuvakaappaus virtuaalikoneen komentorivistä [53].

Hunajapurkkia testattiin virtuaalikoneella Kali Linux -ympäristössä. Yhteys porttiin 22 otettiin isäntäkoneella Windows-ympäristössä. Valitettavasti koodi ei toiminut ihan odotusten mukaisesti, mutta hunajapurkki onnistui kuitenkin

tunnistamaan ei-haitallisen toiminnan niin kuin kuvasta 6 näkyy. Syynä voi olla ihan vain koodivirhe, Pythonin ja Joblibin välinen toimintavirhe tai ongelma SSH:n konfiguroinnissa virtuaalikoneella. Valitettavasti todellista syytä siihen ei ehditty selvittämään eikä ongelmaa korjaamaan yrityksistä huolimatta.

6 Yhteenveto

Insinöörityössä pyrittiin luomaan aloittelevalle penetraatiotestaajalle yleiskäsitys siitä, mihin kaikkeen tekoälyä todellisuudessa voikaan käyttää ja minkä tyyppisiä tekoälyä hyödyntäviä testaustyökaluja ja -menetelmiä on jo olemassa. Tässä raportissa tutkittavien testausmenetelmien määrä rajattiin tarkoituksella kolmeen, koska jo etukäteen oli tiedossa pelkästään tekoälyn suuri laajuus aiheena. Siitä syystä voitiin olettaa, että kyberturvallisuudessa käytettäviä tekoälyä hyödyntäviä menetelmiä on kokonaisuudessaan liian paljon käsiteltäväksi yhdessä insinöörityössä.

Raportille asetetut tavoitteet täyttyivät pääsääntöisesti odotusten mukaisesti. Kahdessa ensimmäisessä osiossa onnistuttiin käsittelemään kyberturvallisuutta ja tekoälyä yleisellä tasolla kohtalaisen tiiviisti niin kuin oli tarkoituskin, mutta kuitenkin tarvittavan laajasti ymmärtääkseen tekoälyn hyödyt ja haitat sekä sen käyttömahdollisuudet kyberturvallisuudessa, erityisesti testauspuolella. ChatGPT:n avulla luotu hunajapurkki ei kuitenkaan toiminut täysin toivotulla tavalla. Hunajapurkki tunnisti kyllä ei-haitallisen toiminnan, kun virtuaalikoneeseen otettiin etäyhteys SSH:lla, mutta haitallisen toiminnan havaitseminen ja erilliseen tiedostoon tallentaminen ei valitettavasti toiminut.

Ottaen huomioon, että kovin laajaa aiempaa kokemusta tai tietoa tekoälystä ei entuudestaan ollut, työn aikana tuli aiheesta opittua odotettua enemmän. Kaikkea opittua ei kuitenkaan tuotu esille raportissa, jotta pysyttäisiin pääaiheessa eli tekoälyn käyttöön penetraatiotestauksessa. Vaikka penetraatiotestaus oli aihealueena entuudestaan kohtalaisen tuttu, tuli siitäkin opittua paljon uutta, erityisesti edelleen käytössä olevista perinteisistä testausmenetelmistä ja tämän ansiosta voikin lähteä kyberturvallisuusalalle töihin hieman itsevarmemmalla asenteella.

Tutkimustyön aikana nousi monesti esiin tekoälyn erityisen nopea kehitys, mutta samaan aikaan tuli tietoon monia osa-alueita, joissa tekoälyn käyttö ei vielä ole kovin tehokasta. Ihmisen on esimerkiksi edelleen toimittava

varsinaisena päätöksentekijänä. Lisäksi henkilökunnan kouluttaminen organisaation tasolla tekoälyn käytössä vaatii paljon resursseja. Koulutuksen on oltava jatkuvaa, sillä tekoäly kehittyy enemmän päivä päivältä. Lisäksi vaikka tekoäly kykeneekin analysoimaan esimerkiksi massadataa ja ennen päätöksentekoa käymään läpi erilaiset skenaariot seurauksineen huomattavasti nopeammin kuin ihminen, on yllättävää, että esimerkiksi laajojen ohjelmakoodikokonaisuuksien tehokkaaseen analyysiin ei tekoäly vielä kykene. Samaan aikaan se on myös ymmärrettävää, koska on vielä paljon asioita, joita ihminenkään vielä täysin oivaltanut.

Karu todellisuus on se, että yksikään verkkoon liitetty laite ei tule todennäköisesti koskaan olemaan täysin turvassa kyberhyökkäyksiltä ja verkkorikollisilta, koska uudenlaisia hyökkäysmenetelmiä ilmaantuu päivä päivältä enemmän. Sama koskee tekoälyä. On esimerkiksi asioita, joihin tekoäly ei tämän insinöörityön kirjoitushetkellä kyennyt, mutta todennäköisesti raportin julkaisun jälkeen monia tekoälyyn liittyviä ongelmia on jo onnistuttu ratkaisemaan. Jo käytettyjen lähteiden kohdalla tuli huomattua, että parikin kuukautta vanha artikkeli saattaa sisältää osittain vanhaa tietoa, puhumattakaan esimerkiksi vuoden vanhoista tutkimuksista. Onkin mielenkiintoista seurata, mihin tekoäly kykenee esimerkiksi vuoden päästä.

Tätä tutkimustyötä voisi laajentaa muun muassa perehtymällä tarkemmin esimerkiksi eri kone- ja syväoppimistekniikoihin algoritmitasolla. Tämän auttaisi ymmärtämään tekoälyn toimintaa vielä syvällisemmin sekä pohtimaan yksityiskohtaisemmin tekoälyjärjestelmien mahdollisuuksia kyberturvallisuudessa. Lisäksi syyn ChatGPT:n luoman hunajapotin toimimattomuudelle olisi hyvä selvittää.

Lähteet

1 Mitä on kyberturvallisuus? Verkkoaineisto. F-Secure <https://www.f-secure.com/fi/articles/what-is-cyber-security>. Luettu 17.4.2024.

2 What is cybersecurity? Verkkoaineisto. SAP. <https://www.sap.com/finland/products/financial-management/what-is-cybersecurity.html>. Luettu 17.4.2024.

3 Siukonen, Timo & Neittaanmäki, Pekka. 2019. Mitä tulisi tietää tekoälystä. Jyväskylä: Docendo.

4 Kyberhyökkäyksen estäminen. Verkkoaineisto. Kaspersky. <https://www.kaspersky.fi/resource-center/preemptive-safety/how-to-prevent-cyberattacks>. Luettu 17.4.2024.

5 Mikä on ransomware? Verkkoaineisto. F-Secure. <https://www.f-secure.com/fi/articles/what-is-a-ransomware-attack>. Luettu 17.4.2024.

6 Mikä on troijalainen? Verkkoaineisto. F-Secure. <https://www.f-secure.com/fi/articles/what-is-a-trojan>. Luettu 17.4.2024.

7 Mitä ovat vakoiluohjelmat? Verkkoaineisto. F-Secure. <https://www.f-secure.com/fi/articles/what-is-spyware>. Luettu 17.4.2024.

8 Mikä on palvelunestohyökkäys (DDoS)? Verkkoaineisto. F-Secure. <https://www.f-secure.com/fi/articles/what-is-ddos>. Luettu 17.4.2024.

9 Kaikki tietojenkalasteluhuijauksista ja niiden estämisestä: Tämä sinun tulee tietää. Verkkoaineisto. Kaspersky. <https://www.kaspersky.fi/resource-center/preemptive-safety/phishing-prevention-tips>. Luettu 17.4.2024.

10 Mikä on SQL-injektio? Määritelmä ja selitys. Verkkoaineisto. Kaspersky. <https://www.kaspersky.fi/resource-center/definitions/sql-injection>. Luettu 17.4.2024.

11 Mitä on käyttäjän manipulointi? Verkkoaineisto. F-Secure. <https://www.f-secure.com/fi/articles/what-is-social-engineering>. Luettu 17.4.2024.

12 Internet Crime Report. 2024. Verkkoaineisto. Federal Bureau of Investigation. <https://www.ic3.gov/Media/PDF/AnnualReport/2023_IC3Report.pdf>. 6.3.2024. Katsottu 4.5.2024.

13 Tuominen, Jukka. 2017. Verkonvalvonta. Opinnäytetyö. Lahden ammattikorkeakoulu. Theseus-tietokanta.

14 Ojala, Henri. 2020. Perusyhtymän johtamisjärjestelmän kybervalvonta. Pro gradu -tutkielma. Maanpuolustuskorkeakoulu. Doria-julkaisuarkisto.

15 Mustahattu-, valkohattu- ja harmaahattuhakkereiden määritelmä ja selitys. Verkkoaineisto. Kaspersky. <https://www.kaspersky.fi/resource-center/definitions/hacker-hat-types>. Luettu 18.4.2024.

16 Kolari, Jukka & Kallio, Aleksi. 2023. Tekoäly 123: matkaopas tulevaisuuteen. E-kirja. Jyväskylä: Docendo.

17 Hautala, Santeri. 2023. Tekoälyn rooli tietoturvassa. Opinnäytetyö. Turun ammattikorkeakoulu. Theseus-tietokanta.

18 Numminen Lari. 2023. Mitä on syväoppiminen? Verkkoaineisto. Finnish Up. <https://www.finnishup.com/mita-on-syvaoppiminen/>. 16.10.2023. Luettu 26.4.2024.

19 What the heck is a deepfake? Verkkoaineisto. University of Virginia. <https://security.virginia.edu/deepfakes>. Luettu 28.4.2024.

20 Tekoälyn perusteet. 2023. Opintomateriaali. Metropolia Ammattikorkeakoulu.

21 What is Reinforcement Learning? Verkkoaineisto. Amazon Web Services. <https://aws.amazon.com/what-is/reinforcement-learning/>. Luettu 28.4.2024.

22 Numminen, Lari. 2023. Mitä ovat suuret kielimallit ja miten ne toimivat? Verkkoaineisto. Finnish Up. <https://www.finnishup.com/mita-ovat-suuret-kielimallit-ja-miten-ne-toimivat/>. 17.10.2023. Luettu 28.4.2024.

23 What is a large language model (LLM)? Verkkoaineisto. Cloudflare. <https://www.cloudflare.com/learning/ai/what-is-large-language-model/>. Luettu 28.4.2024.

24 Numminen, Lari. 2023. Mitä on generatiivinen tekoäly? Verkkoaineisto. Finnish Up. <https://www.finnishup.com/mika-on-generatiivinen-ai/>. 17.10.2023. Luettu 28.4.2024.

25 What is Generative AI? Verkkoaineisto. Amazon Web Services. <https://aws.amazon.com/what-is/generative-ai/>. Luettu 28.4.2024.

26 What is Generative AI? Verkkoaineisto. Nvidia. <https://www.nvidia.com/en-us/glossary/generative-ai/>. Luettu 28.4.2024.

27 Wagh, Amol. 2023. What's Generative AI? Explore Underlying Layers of Machine Learning and Deep Learning. Verkkoaineisto. Medium. <https://medium.com/@amol-wagh/whats-generative-ai-explore-underlying-layers-of-machine-learning-and-deep-learning-8f99272e0b0d>. 26.3.2023. Luettu 28.4.2024.

28 Kun jokainen päivä voi olla aprillipäivä - Mistä deepfakeissa on kysymys? 2024. Verkkoaineisto. Kyberturvallisuuskeskus. <https://www.kyberturvallisuuskeskus.fi/fi/ajankohtaista/kun-jokainen-paiva-voi-olla-aprillipaiva-mista-deepfakeissa-kysymys>. 1.4.2024. Luettu 28.4.2024.

29 Techopedia. 2024. Average cost of data breach worldwide in 2023, by cybersecurity type (in million U.S. dollars). Verkkoaineisto. Statista. <https://www.statista.com/statistics/1450975/global-data-breach-cost-by-cybersecurity-type/>. 16.2.2024. Katsottu 4.5.2024.

30 MacKay, James. Exploring the Benefits and Challenges of AI in Cyber Security. Verkkoaineisto. MetaCompliance. <https://www.metacompliance.com/blog/data-breaches/benefits-and-challenges-of-ai-in-cyber-security>. Luettu 18.4.2024.

31 Amos, Zac. 2024. How AI Reduces the Cost of a Data Breach. Verkkoaineisto. Unite.ai. <https://www.unite.ai/how-ai-reduces-the-cost-of-a-data-breach/>. 23.1.2024. Luettu 18.4.2024.

32 The Role of AI in Cybersecurity: Anticipating and Preventing Attacks. 2023. Verkkoaineisto. BDO Digital. <https://www.bdodigital.com/insights/cybersecurity/the-role-of-ai-in-cybersecurity-anticipating-and-preventing-attacks>. 15.12.2023. Luettu 18.4.2024.

33 Rikoslaki ja tietomurto. Verkkoaineisto. Minilex. <https://www.minilex.fi/a/rikoslaki-ja-tietomurto>. Luettu 30.4.2024.

34 Klausaité, Laura. 2022. Mikä on hunajapurkki – ja miksi hakkerit vihaavat niitä? Verkkoaineisto. NordVPN. <https://nordvpn.com/fi/blog/hunajapurkki/>. 21.2.2022. Luettu 19.4.2024.

35 Sun, Chongxin; Bu, Youjun; Chen, Bo; Zhang, Desheng; Chen, Zhonglei; Lu, Xiangyu; Zhang, Surong; Sun, Jia. 2022. Application of Artificial Intelligence Technology in Honeypot Technology. Verkkoaineisto. IEEE Xplore. <https://ieeexplore.ieee.org/document/10013349>. 22.12.2022. Luettu 19.4.2024.

36 Anand, Kunal. 2023. Organizations Must Embrace Dynamic Honeypots to Outpace Attackers. Verkkoaineisto. Infosecurity Magazine. <https://www.infosecurity-magazine.com/opinions/embrace-dynamic-honeypots-outpace/>. 28.11.2023. Luettu 19.4.2024.

37 What is a chatbot? Verkkoaineisto. IBM. <https://www.ibm.com/topics/chatbots>. Luettu 25.4.2024.

38 Gurinaviciute, Juta. 2024. Why AI-Powered Chatbots Could Be Cyber Threats To Businesses. Verkkoaineisto. Forbes. <https://www.forbes.com/sites/forbestechcouncil/2023/04/24/why-ai-powered-chatbots-could-be-cyber-threats-to-businesses/>. 24.4.2023. Luettu 25.4.2024.

39 AI Chatbots Have Gone Rogue: The Terrifying New Face of Cyber Attacks! 2023. Verkkoaineisto. Stan's Garage. <https://www.youtube.com/watch?v=sRCXfAXpz9c>. 14.9.2023. Katsottu 25.4.2024.

40 Hamin, Maia & Scott, Stewart. 2024. Hacking with AI. Verkkoaineisto. The Digital Forensic Research Lab. <https://dfrlab.org/2024/02/15/hacking-with-ai/>. 15.2.2024. Luettu 26.4.2024.

41 Hughes, Alex. 2023. AI: Why the next call from your family could be a deepfake scammer. Verkkoaineisto. BBC Science Focus Magazine. <https://www.sciencefocus.com/future-technology/ai-deepfake-scam-calls>. 26.8.2023. Luettu 25.4.2024.

42 Deep-Fake Audio and Video Links Make Robocalls and Scam Texts Harder to Spot. 2024. Verkkoaineisto. Federal Communications Commission. <https://www.fcc.gov/consumers/guides/deep-fake-audio-and-video-links-make-robocalls-and-scam-texts-harder-spot>. Päivitetty 8.2.2024. Luettu 25.4.2024.

43 Sharma, Ashwani. 2023. Enhancing Cybersecurity with AI: Simulation and Vulnerability Analysis. Verkkoaineisto. Signity Software Solutions. <https://www.signitysolutions.com/tech-insights/enhancing-cybersecurity-with-ai>. 25.9.2023. Luettu 1.5.2024.

44 Rouse, Margaret. 2024. ChatGPT. Verkkoaineisto. Technopedia. <https://www.techopedia.com/fi/sanasto/chatgpt>. 13.2.2024. Luettu 5.5.2024.

45 Python. Verkkoaineisto. <https://docs.python.org/3/>. Luettu 5.5.2024.

46 ChatGPT. <https://chat.openai.com>. Vierailtu 5.5.2024.

47 Twisted. Verkkoaineisto. <https://docs.twisted.org/en/stable/>. Luettu 5.5.2024.

48 Zieniūtė, Ugnė. 2022. Mitä TCP ja UDP ovat? Yksinkertainen selitys. Verkkoaineisto. NordVPN. <https://nordvpn.com/fi/blog/tcp-udp-protokolla/>. 6.3.2022. Luettu 5.5.2024.

49 IP-osoite – määritelmä ja selitys. Verkkoaineisto. Kaspersky. <https://www.kaspersky.fi/resource-center/definitions/what-is-an-ip-address>. Luettu 5.5.2024.

50 NumPy. Verkkoaineisto. <https://numpy.org/doc/>. Luettu 5.5.2024.

51 scikit-learn. Verkkoaineisto. <https://scikit-learn.org/0.21/documentation.html>. Luettu 5.5.2024.

52 Joblib. Verkkoaineisto. <https://joblib.readthedocs.io/en/stable/>. Luettu 5.5.2024.

53 Kali Linux. Verkkoaineisto. <https://www.kali.org/docs/>. Luettu 5.5.2024.