Mikko Hautaviita

OIKEUSHALLINNON TIETOTEKNIIKKAYMPÄRISTÖN UHKAMALLINNUS

JYVÄSKYLÄN YLIOPISTO
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA
2024

TIIVISTELMÄ

Hautaviita, Mikko
Oikeushallinnon tietotekniikkaympäristön uhkamallinnus
Jyväskylä: Jyväskylän yliopisto, 2024, 92 s.
Kyberturvallisuus, pro gradu -tutkielma
Ohjaaja: Frantti, Tapio

Tutkimuksessa selvitettiin oikeushallinnon tietotekniikkaympäristön keskeiset riskit ja niiden hallintakeinot. Lisäksi tutkimuksessa selvitettiin, miten uhkia voidaan tunnistaa uhkamallinnuksen avulla.

Oikeushallinnon tietotekniikkaympäristö uhkamallinnettiin ohjelmistokeskeisellä uhkamallinnuksella STRIDE-metodologian avulla. STRIDE koostuu kuudesta elementistä, jotka ovat huijaus *(engl. spoofing)*, peukalointi *(engl. tampering)*, kiistäminen *(engl. repudiation)*, tietojen paljastaminen *(engl. information disclosure)*, palvelunesto *(engl. denial of service)* ja oikeuksien korottaminen *(engl. elevation of privilege)*. Uhkamallinnuksen apuna käytettiin Elevation of Privilege- ja Backdoors & Breaches -korttisarjoja, joista käsiteltiin yhteensä 80 korttia. Uhkamallinnus toteutettiin työpajatyöskentelyllä kohdeorganisaation tietoturvapäällikön ja it-erityisasiantuntijan kanssa.

Uhkamallinnuksessa tunnistettiin yhteensä 105 riskiä, joiden vaikutus luokiteltiin vähäiseksi tai kohtalaiseksi. Uhkamallinnuksessa tunnistettiin myös merkittäviä ja kriittisiä riskejä, mutta tietoturvasyistä kyseisiä riskejä ei käsitellä tässä tutkimuksessa. Tutkimuksen liitteessä kaksi tutkija antaa esimerkin kriittisen uhkan käsittelystä luotettavuuden parantamiseksi.

Uhkamallinnus on keskeinen osa kohdeorganisaation riskienhallintaa, koska osa tunnistetuista riskeistä siirrettiin riskienhallintaprosessissa jatkokäsittelyä varten. Uhkien mallintamisessa keskitytään tarkastelemaan järjestelmän heikkouksia ja haavoittuvuuksia, jotka vaikuttavat kohdejärjestelmään. Uhkamallinnuksen tarkoituksena on tunnistaa uhkia, joita muut menetelmät eivät havaitse.

Tutkimuksessa havaittiin tunnistettujen uhkaskenaarioiden avulla käyttäjien tietoturvakoulutuksen tärkeys, koska koulutuksen avulla ehkäistään lukuisia tietoturvapoikkeamia. Teknisestä näkökulmasta tarkasteltuna havaittiin, että EDR, UEBA ja DLP ovat nykyajan digitaalisessa toimintaympäristössä merkittäviä työkaluja, jotka parantavat organisaatioiden tietoturvaa. Muita merkittäviä tekijöitä tietoturvan parantamiseen havaittiin muun muassa turvallisen ohjelmointikoodin kirjoittaminen, penetraatiotestaus ja salasanattomuus.

Asiasanat: uhkamallinnus, riskienhallinta, STRIDE, uhka, riski

ABSTRACT

Hautaviita, Mikko
Legal administration information technology environmental threat modeling
Jyväskylä: University of Jyväskylä, 2024, 92 pp.
Cyber Security, Master's Thesis
Supervisor: Frantti, Tapio

This research investigated the key risks of the information technology environment of the legal administration and the means of managing them. In addition, the research figured out how threats can be identified using threat modeling.

The information technology environment of the legal administration was threat modeled with software-based threat modeling using the STRIDE methodology. STRIDE consists of six elements, which are spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privilege. The Elevation of Privilege and Backdoors & Breaches card sets were used to help with threat modeling, and a total of 80 cards were processed. The threat modeling was carried out by working in a workshop with the information security manager of the target organization and an IT specialist.

A total of 105 risks were identified in the threat modeling, the impact of which was classified as low or moderate. Significant and critical risks were also identified in the threat modeling, but for information security reasons, these risks are not dealt with in this research. In appendix two of the research, the researcher gives an example of handling a critical threat to improve reliability.

Threat modeling is a key part of the target organization's risk management because some of the identified risks were transferred for further processing in the risk management process. Threat modeling focuses on examining system weaknesses and vulnerabilities that affect the target system. The purpose of threat modeling is to identify threats that other methods do not detect.

The research found the importance of information security training for users, because with the help of training, numerous information security deviations are prevented. From a technical point of view, it was found that EDR, UEBA and DLP are important tools in today's digital operating environment that improve the information security of organizations. Other significant factors for improving information security were found, including writing secure programming code, penetration testing and passwordless authentication.

Keywords: threat modeling, risk management, STRIDE, threat, risk

KUVIOT

TAULUKOT

SISÄLLYS

1 JOHDANTO

If It's smart, It's vulnerable.
- Mikko Hyppönen

Jos laite on älykäs, se on haavoittuva. Tästä sitaatista muodostui Hyppösen laki, joka saattaa kuulostaa pessimistiseltä, mutta sitaatissa on perustavanlaatuinen ajatus. Mikäli laitteisiin lisätään toimintoja, niistä tehdään samanaikaisesti haavoittuvia. (Hyppönen 2022, 130.) Jokaiselle haavoittuvalle laitteelle on tarpeellista tehdä uhkamallinnus, koska uhkamallinnuksen avulla tunnistetaan ne uhkat, jotka voivat aiheuttaa kriittisiä riskejä esimerkiksi organisaatioiden liiketoiminnalle. Uhkamallinnus on tärkeä prosessi osana organisaatioiden riskienhallintaa, koska riskianalyysin jälkeen osa riskeistä siirtyy organisaatioiden riskienhallintaprosessiin jatkokäsittelyä varten.

Uhkamallinnus on ennakoivaa toimintaa ja riskeihin perustuvaa arviointia siitä, mitkä hyökkäysskenaariot ovat todennäköisiä tai mahdollisia tarkasteltavassa kohteessa. Uhkamallinnuksessa on oleellista tehdä selväksi, mitkä ovat arvokkaita kohteita hyökkääjälle. Uhkia ovat kaikki tekijät, jotka aiheuttavat haittaa organisaatiolle tai yksittäisille henkilöille. Riski sisältää todennäköisyyden siitä, että uhka toteutuu. Uhkamallinnuksen tekemiseen on useita menetelmiä, ja tässä tutkimuksessa uhkamallinnuksessa käytetään ohjelmistokeskeistä uhkamallinnusta ja hyödynnetään STRIDE-metodologiaa. (Ullman 2023, 7.) Haavoittuvuus puolestaan tarkoittaa jotain heikkoutta järjestelmässä, ohjelmistossa tai ympäristössä, joka voi altistaa organisaation hyökkäyksille, mikäli hyökkääjä käyttää haavoittuvuutta hyväkseen (Ucedavélez & Morana 2015, 259).

Tutkimuksen kohteena on oikeushallinnon tietotekniikkaympäristö, joka uhkamallinnettiin ohjelmistokeskeisellä uhkamallinnuksella STRIDE-metodologian avulla. STRIDE muodostuu kuudesta elementistä, jotka ovat huijaus *(engl. spoofing)*, peukalointi *(engl. tampering)*, kiistäminen *(engl. repudiation)*, tietojen paljastaminen *(engl. information disclosure)*, palvelunesto *(engl. denial of service)* ja oikeuksien korottaminen *(engl. elevation of privilege)*. Uhkamallinnuksen apuna käytettiin Elevation of Privilege- ja Backdoors & Breaches -

korttisarjoja. Tässä tutkimuksessa käsiteltiin yhteensä 80 korttia. Tutkimuksen tarkoituksena on selvittää keskeiset oikeushallinnon tietotekniikkaympäristön riskit ja niiden hallintakeinot. Lisäksi tutkimuksen tavoitteena on selvittää, miten uhkia voidaan tunnistaa uhkamallinnuksen avulla. Uhkamallinnuksessa tunnistettiin yhteensä 105 riskiä, joiden vaikutus on luokiteltu vähäiseksi tai kohtalaiseksi. Tutkimuksessa nostetaan esille ne riskit, jotka ovat keskeisiä oikeushallinnon tietotekniikkaympäristössä.

Toinen pääluku käsittelee tutkimuksen toteutusta ja tutkimuksessa käytettyjä työkaluja. Kolmannessa pääluvussa käsitellään uhkamallinnusta yleisellä tasolla ja esitetään erilaisia tapoja tehdä uhkamallinnusta: niitä ovat omaisuuskeskeinen, hyökkääjäkeskeinen ja ohjelmistokeskeinen uhkamallinnus. Neljäs pääluku keskittyy STRIDE-metodologian analysointiin ja siihen, miten uhkia voidaan tunnistaa STRIDE:n avulla ja miten kyseisiä uhkia voidaan lieventää. Uhkien tunnistamisessa ja lieventämiskeinoissa hyödynnetään Shostackin (2014) havaintoja, joita täydennetään teoksista ja tutkimusartikkeleista saatujen havaintojen perusteella. Lisäksi neljännessä pääluvussa esitetään tyypillisimpien uhkien lieventämiskeinoja. Tyypillisimmät uhkat perustuvat Stallingsin (2019) esittämään listaan. Viides pääluku käsittelee kohdeorganisaatiota yleisellä tasolla, ja siinä esitellään oikeushallinnon tietotekniikkaympäristö. Kuudennessa pääluvussa esitellään oikeushallinnon tietotekniikkaympäristön keskeiset riskit ja niiden hallintakeinot sekä pohditaan tutkimuksen luotettavuutta ja eettisyyttä ja esitetään jatkotutkimusaihe. Viimeisessä pääluvussa nostetaan esille tutkimuksen keskeiset havainnot.

2 TUTKIMUKSEN TOTEUTUS

Toisessa pääluvussa käsitellään tutkimuksen toteutus ja tutkimuksessa käytetyt työkalut. Tutkimuksessa tehty uhkamallinnusprosessikuvaus on kuvattu liitteessä yksi. Tutkimuksen tietoteoreettisena lähtökohtana käytetään konstruktiivista tutkimusotetta.

2.1 Tutkimusongelma ja -kysymykset

Tutkimuksen tarkoituksena on selvittää oikeushallinnon tietotekniikkaympäristön keskeiset riskit. Ensimmäinen alatutkimuskysymys tukee päätutkimuskysymystä, koska on oleellista selvittää myös riskien hallintakeinot. Toisen alatutkimuskysymyksen tarkoituksena on selvittää, miten ohjelmistokeskeisen uhkamallinnus ja eritoten pelikorttien käyttäminen soveltuvat organisaatiokohtaiseen uhkamallinnukseen.

Päätutkimuskysymys:

- Mitkä ovat keskeiset oikeushallinnon tietotekniikkaympäristön riskit?

Alatutkimuskysymykset:

- Miten näitä riskejä voidaan hallinnoida?
- Miten riskejä voidaan kartoittaa uhkamallinnuksen avulla?

2.2 Konstruktiivinen tutkimusote tietoteoreettisena lähtökohtana

Konstruktiivista tutkimusotetta käytetään ongelmien määrittelemiseen ja ratkaisemiseen sekä olemassa olevan järjestelmän tai suorituskyvyn parantamiseen. Konstruktiivista tutkimusotetta on sovellettu tekniikan alalla esimerkiksi

uuden tuotteen kehittämisessä. (Oyegoke 2011, 578–579.) Lisäksi konstruktiivisessa tutkimusotteessa keskitytään muun muassa toimintaympäristön ja organisaation ongelmiin, jotka on tarkoitus ratkaista (Heikkinen & Söderqvist 2005, 38). Keskeistä konstruktiivisessa tutkimusotteessa on se, että tutkimuksen tavoitteena on tuottaa uudenlaisia konstruktioita, joiden avulla yritetään ratkaista reaalimaailman haasteita. Tutkimuksen keskeinen teema on konstruktio, joka on ratkaisu reaalimaailman haasteeseen. Konstruktio rakentuu neljästä tekijästä, jotka ovat ongelman ja ratkaisun käytännöllinen merkitys, ratkaisun toimivuus käytännössä, yhteys aikaisempaan teoriaan ja tutkimuksen teoreettinen kontribuutio. (Lukka 2014.) Virtasen (2006, 49) mukaan konstruktiivisessa tutkimuksessa tuotetaan yksi ratkaisu organisaation dilemmaan. Huomioitavana on se, että ratkaisu on siirrettävissä muihin yrityksiin. Tutkimuksen pääpaino on ratkaisun esittelyssä sekä siinä, että ratkaisu on esitettävä opinnäytetyön puitteissa. Lukka (2000) kuvaa konstruktiivisen tutkimusprosessin rakentuvan seuraavista vaiheista:

1. Löydä käytännön ongelma, joka mahdollistaa teoreettiseen kontribuutioon.
2. Tutki mahdollisuuksia pitkän aikavälin yhteistyöhön kohdeorganisaation kanssa.
3. Syvenny tutkittavaan aiheeseen käytännöllisesti ja teoreettisesti.
4. Kehitä ratkaisuidea ja ongelmanratkaisurakenne, joka mahdollistaa teoreettiseen kontribuutioon.
5. Toteuta ja testaa kehitetty ratkaisu.
6. Pohdi ratkaisun sovellettavuutta.
7. Tunnista ja analysoi teoreettinen kontribuutio.

Tutkimusongelmana oli selvittää oikeushallinnon tietotekniikkaympäristön riskit. Tutkittavan aiheen tarpeellisuus on ensisijaisen tärkeä, koska tutkimuskohteena on kriittinen valtiollinen toimija, jolle on asetettu lakisääteiset velvoitteet ja jonka häiriötilanteet voivat vaikuttaa yhteiskuntaan merkittävästi. Kohdeorganisaatiolla oli tarve toteuttaa tietotekniikkaympäristön uhkamallinnus, ja yhteistyö alkoi syyskuun alusta vuonna 2023.

Tutkimus alkoi aineiston keräämisellä, joka tehtiin laadullisen aineiston analyysin perusteella. Tavoitteena oli luoda tutkittavaan aihepiiriin selkeyttä, ja tämän avulla tuotettiin lisää tietoa tutkittavasta kohteesta. Lähteet valittiin harkiten käyttäen ajantasaisia ja tutkittavan aihepiirin kannalta oleellisia teoksia ja tutkimusartikkeleita. Tutkijan tavoitteena oli luoda tiivistetty, mutta kattava aineisto uhkamallinnuksesta keskittyen ohjelmistokeskeiseen uhkamallinnukseen STRIDE:n avulla. (Ks. Taanila 2007.)

Tutkimusongelmaa lähdettiin selvittämään työpajaperusteisesti; työpajoihin osallistuivat tutkijan lisäksi kohdeorganisaation tietoturvapäällikkö ja iterityisasiantuntija. Ensimmäinen työpaja toteutettiin 19.9.2023 ja viimeinen työpaja 9.1.2024. Yhteensä työpajoja oli yksitoista, ja niiden kesto oli yhdestä tunnista seitsemään tuntiin. Yhteensä aikaa työpajatyöskentelyyn kului noin kaksikymmentäyhdeksän tuntia. Työpajat toteutettiin virtuaalityöpajoina, joissa en-

simmäiseksi tietotekniikkaympäristö mallinnettiin OWASP Threat Dragon -ohjelman avulla ja tämän jälkeen alettiin käsitellä Elevation of Privilege- ja Backdoors & Breaches -korttisarjojen uhkakortteja. Työpajat mukailivat ryhmäkeskustelun mukaista formaattia, koska tarkoituksena oli selvittää uhkaskenaarioiden lieventämiskeinot ja pohtia, miten eri uhkat vaikuttavat omaan toimintaympäristöön. Ryhmäkeskusteluun osallistuneilla henkilöillä on monen vuoden työkokemus tietoturvasta, ja he ovat olleet samassa yksikössä töissä myös usean vuoden ajan. Ryhmäkeskustelussa mielipiteet ja asenteet käsitetään henkilöiden henkilökohtaisiksi ominaisuuksiksi, jotka henkilöt jakavat tai jättävät jakamatta. (Ks. Valtonen 2005, 226.)

Työpajatyöskentely ja korttisarjojen käyttäminen osoittautui hyväksi ratkaisuksi tutkimusongelman kannalta. Ensimmäisestä työpajasta alkaen saatiin tutkimustuloksia, joista saatiin tärkeää tietoa tutkimusongelmasta. Tutkimusongelman kannalta oleelliset tiedot on taulukoitu luvussa kuusi taulukoissa 18–29. Tutkimustuloksia voidaan soveltaa muuhun organisaatioon, mutta huomioitavana asiana on järjestelmien erilaisuus. Jokainen lieventämiskeino täytyy tarkastella järjestelmäkohtaisesti, mutta uhkien lieventämiskeinot on kuvattu hyvin yleisellä tasolla, mikä voi helpottaa lieventämiskeinojen jatkokäyttöä uuden järjestelmän tarkastelemisessa. Lisäksi teoriaosassa tarkasteltiin uhkien havaitsemista ja lieventämiskeinoja, joita voidaan soveltaa yleisellä tasolla myös muuhun järjestelmään.

Tutkimuksen empiirinen kontribuutio muodostui luvun kuusi saatujen havaintojen perusteella. Käytännössä tämä tarkoittaa uuden konstruktion syntymistä, koska teoriaosasta saadut havainnot tukevat tätä (ks. Lukka 2014).

2.3 Tutkimuksessa käytetyt työkalut

Seuraavaksi esitellään tutkimuksessa käytetyt työkalut, jotka ovat OWASP Threat Dragon -uhkamallinnustyökalu ja Elevation of Privilege- ja Backdoors & Breaches -korttisarjat. Lisäksi viimeisessä alaluvussa esitellään, miten riskit on luokiteltu.

2.3.1 Elevation of Privilege- ja Backdoors & Breaches -korttisarjat

Adam Shostack on kehittänyt Elevation of Privilege -korttipelin uhkamallinnuksen helpottamiseksi. Tarkoituksena oli helpottaa ohjelmistokehittäjiä, jotka eivät ole tietoturvan harjoittajia tai asiantuntijoita. (Agile Stationery 2024.) Korttisarja on mahdollista ladata GitHubista, mutta Oikeusrekisterikeskus osti fyysisen korttipakan. Tutkimuksessa uhkamallinnuksessa ei käytetty Shostackin mallia, joka on käytännössä korttipeli. Tässä tutkimuksessa käytettiin korttipakan uhkaskenaarioita, joista jokainen uhka käsiteltiin, ja mietittiin lievennystoimenpiteet. Yhteensä korttipakassa on seitsemänkymmentäkaksi korttia. Elevation of Privilege -korttipakka perustuu STRIDE-metodologiaan, ja jokainen

STRIDE:n elementti sisältää kaksitoista valmista uhkaskenaariota. Valmiit uhkaskenaariot helpottivat uhkamallinnuksen tekemistä, koska uhkia ei tarvinnut miettiä ryhmäkeskusteluissa.

Elevation of Privilege -korttipakan lisäksi käsittelimme myös Backdoors & Breaches -korttipakan uhkaskenaariot. Tässä korttipakassa on yhteensä viisikymmentäkaksi korttia. Korttipakan sisältö jakautuu seuraaviin aihealueisiin: *engl. Initial Compromise (10), Pivot & Escalate (7), Persistence (9), Procedures (10) ja Injects (10).* (Ks. Black Hills 2024.) Tämä korttisarja erosi Elevation of Privilege -korttipakasta uhkaskenaarioiden perusteella, ja lisäksi jokainen uhka mietittiin STRIDE-metodologian mukaisesti.

2.3.2 OWASP Threat Dragon

OWASP Threat Dragon on ilmainen avoimen lähdekoodin mallinnustyökalu, jota käytetään uhkamallinnuskaavioiden tekemiseen. Työkalun avulla on mahdollista tallentaa uhkia ja kirjoittaa uhkien lievennystoimenpiteistä ja uhkamallinnusta on mahdollista tehdä STRIDE-, LINDDUN-, CIA- ja DIE-taksonomioita hyödyntäen. (Owasp 2024.) Työkalu on mahdollista asentaa Windows-, Linux- ja Mac-käyttöjärjestelmille, ja Threat Dragon mahdollistaa käsiteltyjen uhkien raportointiominaisuuden. Raportissa Threat Dragon luetteloi jokaisen komponentin sisältämät uhkat ja lievennystoimenpiteet. (Tarandach & Coles 2020, 129–130.)

OWASP Threat Dragon -uhkamallinnustyökalu osoittautui tässä tutkimuksessa hyödylliseksi uhkamallinnuksen apuvälineeksi. Työkalua oli helppo käyttää, ja ongelmia työkalun käyttämisessä ei esiintynyt. Negatiivisena asiana oli se, että työkalu ei tarjonnut valmiita lievennysehdotuksia ja jokaisen uhkaskenaarion lievennystoimenpiteet jouduttiin itse tekemään. Työn nopeuttamiseksi valmiit lievennystoimenpiteet olisivat nopeuttaneet uhkamallinnuksen tekemistä. Huolimatta siitä, että työkalu tarjoaisikin valmiita lievennystoimenpiteitä, niin jokainen lievennystoimenpide joudutaan tarkastelemaan tapauskohtaisesti. Työpajatyöskentelyn jälkeen tarkastelimme raportin, jonka kokonaispituus on neljätoista sivua.

2.3.3 Riskianalyysi

Riskianalyysin tekijät voivat olla laadullisia, määrällisiä tai puolimäärällisiä. Laadullisissa tekniikoissa käytetään määreisiin perustuvaa asteikkoa, esimerkiksi pieni, keskisuuri ja suuri. Määrällisissä tekniikoissa käytetään numeerista asteikkoa, esimerkiksi rahallinen arvo, ja puolimäärällisillä tekniikoilla käytetään laadullista asteikkoa määritetyillä arvoilla. Riskianalyysin tarkoituksena on kohdistaa huomio niihin riskeihin ja hallintakeinoihin, jotka oikein hallittuna parantavat todennäköisyyttä organisaation tavoitteiden saavuttamiseksi. (ISO/IEC 27005:2022, 24.) Riskianalyysissä huomioidaan uhkat, haavoittuvuudet, todennäköisyys ja vaikutukset organisaation toimintaan. Riskianalyysi on

jatkuva prosessi, jota tehdään koko järjestelmän kehityksen elinkaaren ajan. (NIST SP 800-53 2020, 240.)

Ucedavélezin ja Moranan (2015, 265–266) mukaan riskien määritteleminen on osa todennäköisyyden ja vaikutuksen huomioon ottamista. Edellä mainittu lause voidaan muuntaa kaavaksi, joka on

Riskin prioriteettiarvo = Todennäköisyys (P) x Vaikutus (I)

Laadullisen riskianalyysin tavoitteena on määrittää riskien tasot, jotka on esitetty taulukossa yksi.

TAULUKKO 1 Esimerkki riskien luokittelusta

Asteikko	**Todennäköisyys**	**Vaikutus luottamuksellisuuteen, eheyteen ja saatavuuteen**
Erittäin matala	Erittäin epätodennäköistä	Erittäin vähäinen
Matala	Voi esiintyä satunnaisesti	Vähäinen
Kohtalainen	Yhtä todennäköistä, että tapahtuu/ei tapahdu	Kohtalainen
Korkea	Todennäköisesti tapahtuu	Korkea
Kriittinen	Lähes varmaa, että tapahtuu	Kriittinen

(Ks. Ucedavélez & Morana 2015, 265.)

TAULUKKO 2 Riskikartta

Riski = P x I	Erittäin matala (1)	Matala (2)	Kohtalainen (3)	Korkea (4)	Kriittinen (5)
Kriittinen (5)	5	10	15	20	25
Korkea (4)	4	8	12	16	20
Kohtalainen (3)	3	6	9	12	15
Matala (2)	2	4	6	8	10
Erittäin matala (1)	1	2	3	4	5

(Ks. Ucedavélez & Morana 2015, 266.)

Taulukon oikeassa reunassa on arvio todennäköisyydestä (P), ja yläpuolella on vaikutuksen arviointi (I). Värikoodit tarkoittavat seuraavaa:

- Vihreä: erittäin matala/ matala, 1–4 pistettä.
- Keltainen: kohtalainen, 5–9 pistettä.
- Punainen: korkea/ Kriittinen, yli 9 pistettä.

Tutkimuksessa hyödynnettiin Ucedavélezin ja Moranan (2015) riskianalyysiä, mutta tässä tutkimuksessa käytettiin seuraavaa asteikkoa: vähäinen, kohtalainen, merkittävä ja kriittinen.

- Vähäinen, 1–4 pistettä.
- Kohtalainen, 5–8 pistettä.
- Merkittävä, 9–12 pistettä.
- Kriittinen, 13–16 pistettä.

3 UHKAMALLINNUS

Shostackin (2014) mukaan uhkamallinnus koostuu vaiheista, jotka muodostavat uhkamallinnuksen ytimen. Keskeiset kysymykset, joiden avulla uhkamallinnuksessa saadaan kattava kuvaus tarkasteltavasta kohteesta, ovat seuraavat:

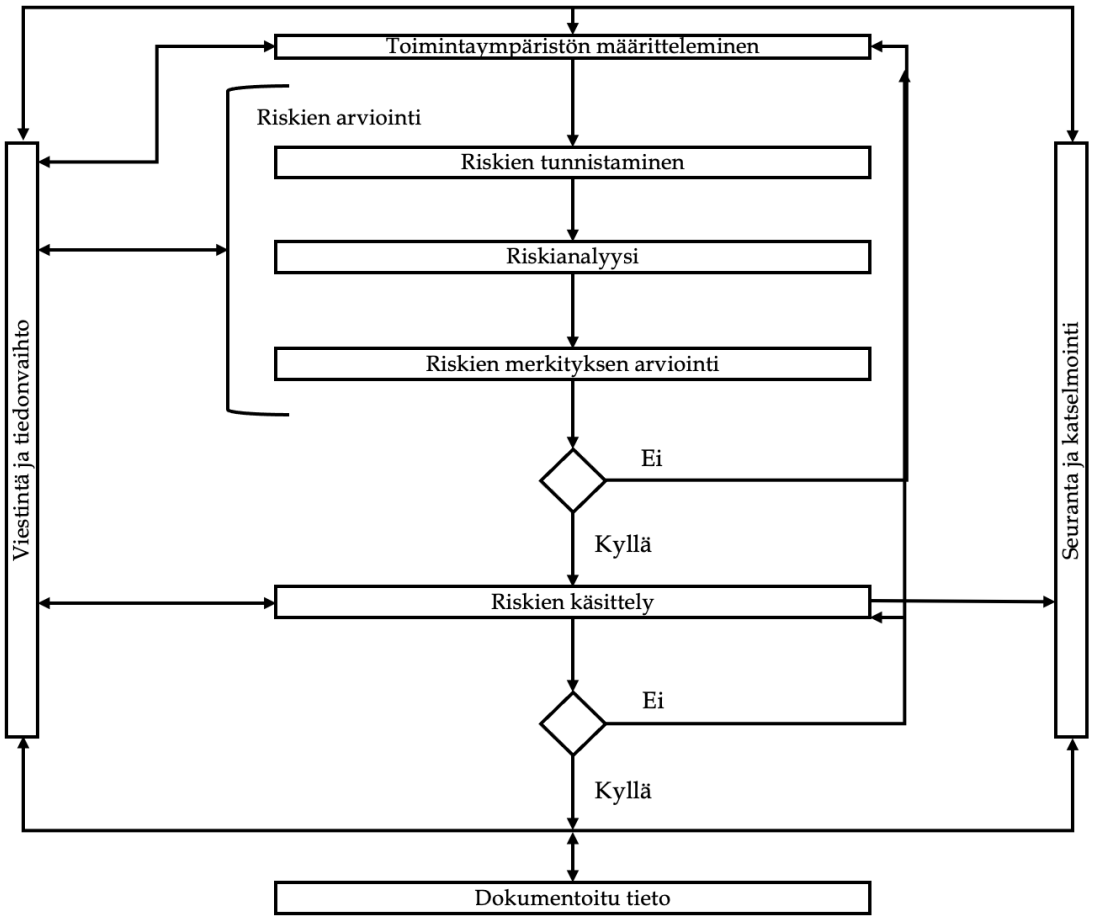
- Mitä sinä rakennat?
- Mikä voi epäonnistua, kun se on rakennettu?
- Mitä voit tehdä asioille, jotka voivat epäonnistua?
- Teitkö kattavan analyysityön?

Uhkamallinnus on prosessi, jonka avulla järjestelmää analysoidaan ja etsitään heikkouksia. Prosessin aikana on tarkoitus tunnistaa järjestelmän heikkoudet mahdollisimman aikaisessa vaiheessa, koska korjauskustannukset voivat olla kalliita. Keskeinen ajatus suunnitteluvaiheessa tehtävässä uhkamallinnuksessa on tunnistaa järjestelmän heikot kohdat ja pienentää riskejä omistajien ja käyttäjien kannalta hyväksyttävälle tasolle. (Tarandach & Coles 2020.) Uhkamallinnus on ennakoivaa riskeihin perustuvaa arviointia, jossa tarkastellaan hyökkäysten todennäköisyyttä. Vaihtoehtoisesti voidaan tarkastella tietyn skenaarion vaikutusta ympäristöön, henkilöön tai henkilöihin. Uhkien mallinnusprosessissa tarkastellaan heikkouksia tai haavoittuvuuksia, jotka vaikuttavat organisaatioon tai järjestelmään. (Ullman 2023, 7.)

3.1 Uhkamallinnus osana riskienhallintaa

Tietoturvariskien hallintaprosessi on kuvattu standardin ISO/IEC 27005:2022 mukaan kuviossa yksi. Tietoturvariskienhallintaprosessi perustuu yleiseen riskienhallintaprosessiin, joka on määritelty standardissa ISO 31000. (ISO/IEC 27005:2022, 12.) Riskienhallinta on osa organisaatioiden sisäistä valvontaa. Riskienhallinta on koordinoitu ja jatkuva prosessi, jonka avulla tunnistetaan, analysoidaan, arvioidaan, käsitellään ja seurataan riskejä. Riskienhallinta on organisaatioiden keino varmistaa tavoitteiden saavuttaminen ja toiminnan jatku-

vuuden turvaaminen. (Valtiovarainministeriö 2021, 15.) Tutkimuksen tarkoituksena ei ole kuvata riskienhallinnan tematiikkaa syvällisesti, vaan tavoitteena on havainnollistaa lukijalle, mihin kokonaisuuteen uhkamallinnus liittyy kohdeorganisaation riskienhallintaprosessissa.



KUVIO 1 Tietoturvariskien hallintaprosessi

Tutkimuksessa määriteltiin ensiksi toimintaympäristö, joka on oikeushallinnon tietotekniikkaympäristö. Toimintaympäristön määritteleminen perustuu sisäisen ja ulkoisen toimintaympäristön muodostamiseen tietoturvariskien hallintaa tai -riskien arviointia varten (ISO/IEC 27005:2022, 12). Toimintaympäristön määrittelemisen jälkeen aloitettiin uhkamallinnusprosessi, jonka avulla saatiin luettelo erilaisista riskeistä. Uhkien tunnistamisessa on keskeistä tunnistaa uhkatekijät, jotka vaikuttavat tarkastelussa olevaan järjestelmään. Uhkatekijät voidaan luokitella kolmeen osa-alueeseen, jotka ovat ympäristöön, liiketoiminnan resursseihin ja vihamielisiin toimijoihin liittyvät uhkat. (Stallings 2019, 89.) Tutkimuksessa keskityttiin tarkastelemaan tietotekniikkaympäristöön kohdistuvia uhkia. Uhkalla tarkoitetaan ilmiötä, joka voi aiheuttaa vaurioita järjestelmään. Riski on tietoturvatapahtuman vaikutuksen ja todennäköisyyden yhdistelmä. (ISO/IEC 27005:2022, 7–8.) Uhkamallinnuksen avulla siis tunnistetaan erilaisia uhkia ja riskianalyysin jälkeen uhkasta muodostuu riski, joka on muodostettu uhkan vaikutuksen ja todennäköisyyden arvioinnista. ISO/IEC (27005:2022, 10)

mukaan riskianalyysin tarkoituksena on määrittää riskitaso, johon sisältyy arvio riskin suuruudesta. Riskianalyysi on kuvattu tarkemmin luvussa 2.3.3. Riskien merkityksen arvioinnin tarkoituksena on määrittää, onko riski hyväksyttävä tai siedettävä (ISO/IEC 207005:2022, 10). Tutkimuksessa osa tunnistetuista riskeistä siirtyy riskienhallintaprosessissa jatkokäsittelyä varten. Tutkimuksessa on esitetty ainoastaan ne riskit, jotka luokiteltiin vähäisiksi ja kohtalaisiksi.

Walls, McMullen, Heiser ja Gopal (2023) haastavat perinteisen riskienhallinnan, koska perinteinen riskienhallinta johtaa tehottomiin kyberturvallisuusinvestointeihin ja huonoon kyberturvallisuuteen. Yritysjohtajien tulisi keskittyä uhkien hallintaan ja liiketoimintavaikutusten analysointiin. Tutkijat perustelevat näkemystään sillä, että tuntemattomia tekijöitä muuttuvassa digitaalisessa ympäristössä ei voida mitata numeerisesti. Teknologiaympäristöt ovat monimutkaisia kokonaisuuksia, ja teknologinen muutos johtaa uusiin uhkiin. Riskienarviointiprosessit ovat hitaita, ja ne perustuvat vanhentuneeseen tietoon uhkista. Uhkien hallitsemisen lähestymistapana on rakentaa havaitsemis-, ehkäisy- ja palautumismekanismeja tunnettujen uhkavektoreiden pohjalta ja minimoida uhkien kohteena oleva hyökkäyspinta. Edellä mainittu toimenpide ei ole riskienhallintaa, vaan uhkien hallintaa. Tutkijat nostavat esille kiristyshaittaohjelmaviestit, joiden toteutumistodennäköisyys on sata prosenttia. Yritykset investoivat automatisoituihin toimenpiteisiin vähentääkseen altistumista kiristyshaittaohjelmille. Uhkan olemassaolo on kuitenkin sata prosenttia, mutta mahdollisen vaikutuksen aste pienenee. Toinen esimerkki on nollapäivähaavoittuvuudet ja uhkatoimijat, jotka tuottavat lukuisia uhkia ja näitä uhkia ei ole määritelty. Määritelmän puutteellisuus johtaa siihen, että epävarmuustekijöillä ei voida hallita uhkia, koska todennäköisyyttä epävarmuuden keskiössä ei voida numeerisesti mitata. Epävarmuus tekee sen, ettei perinteinen riskimalli sovellu riskien laskemiseksi. Perinteinen riskimalli muodostuu vaikutuksesta ja todennäköisyydestä, joka tuottaa illuusion riskien hallittavuudesta ja ennustettavuudesta. Tutkijat korostavat, että määritellyt uhkat ovat varmuutta, eivät riskejä. Riskiarvioiden perusteella tehdyt investoinnit yrittävät ratkaista menneisyyden ongelmia, ja tämän vuoksi organisaatioiden tulisi keskittyä uhkatiedon ja -tutkimuksen perusteella tunnistamaan epävarmuudesta nousevia uhkia. Riskienarviointiprosessit ja riskienhallinnan taustalla olevat mallit vääristävät olennaisia tietoja, joita yritysjohtajat tarvitsevat investointien priorisointiin. Tutkijat ehdottavat, että yritysjohtajien tulisi luopua perinteisistä riskienarviointikäytännöistä ja riskeistä puhuttaessa tulisi käyttää termiä uhkille altistumisen hallinta. Tietoturvatiimin tulisi esittää yritysjohtajille selkeä kuvaus yrityksen altistumisesta uhkille, joiden perusteella tehdään investointeja uhkien lieventämiseksi. Uhkien tiedusteluun ja tutkimustoimintaan tulisi keskittyä, jotta vähennetään epävarmuutta tulevaisuuden uhkien kentässä.

Walls, McMullen, Heiser ja Gopal (2023) tarjoavat hyvin poikkeuksellisen keinon tietoturvariskienhallintaan. Tutkijat ovat kuitenkin siinä oikeassa, että todennäköisyyden ennustaminen on haastavaa. Täytyy kuitenkin huomioida se, että ISO/IEC (27005:2022, 12, 28) mukaan riskien arviointiin voidaan käyttää paljon aikaa, mutta tehokkaan päätöksen teon tueksi alustavat arviot riskien

vaikutus- ja todennäköisyysarvioista voivat riittää. Lisäksi riskien arvioinnin jälkeen organisaation täytyy määrittää riskien hyväksyntäkriteerit, jotka määrittävät riskienkäsittelyn jatkotoimenpiteet. Asiantuntijoiden ja riskien omistajien välisessä keskustelussa vahvistetaan riskitaso. Riskien tärkeysjärjestyksen kriteereissä täytyy huomioida organisaation sopimukset, lainsäädäntö ja viranomaisten vaatimukset. Wallsin, McMullenin, Heiserin ja Gopalin (2023) tutkimuksen havaintoihin vaaditaan enemmän tieteellistä näyttöä, koska kyseinen malli haastaa muun muassa ISO/IEC 27005:2022 -standardin tietoturvariskien hallintaprosessin.

3.2 Uhkamallinnuksen hyödyt

Heikkouksien havaitseminen ajoissa mahdollistaa kustannusten pienentämisen prosessin toteuttamisen myöhäisemmässä vaiheessa. Selkeät ja konkreettiset vaatimukset mahdollistavat johdonmukaiset ratkaisut, koska uhkien lieventämisellä ja vaatimuksilla on merkittävä riippuvuussuhde. Uhkien mallintamisessa voidaan huomata, että osa uhkista ei ole organisaatioiden toimintaympäristössä merkityksellisiä, minkä takia kyseisiin uhkiin ei kannata reagoida. Keskeinen asia uhkamallinnuksessa on se, että siinä havaitaan ongelmakohtia, joita muut menetelmät eivät havaitse. Uhkamallinnuksessa ei tulisi keskittyä ongelmakohtiin, jotka muut menetelmät havaitsevat: esimerkiksi tietokantaan liittyvät uhkat, kuten SQL-injektiohyökkäykset, käsitellään vain nopeasti uhkamallinnuksen yhteydessä. (Shostack 2014.) Asiakkaiden henkilötietojen ja organisaation arkaluonteisten tietojen suojaamiseksi organisaatiot yleensä käyttävät tietoturvastandardien tietoturva- ja riskienhallintaprosesseja riskien hallitsemiseksi (Ucedavélez & Morana 2015, 64). Uhkamallinnus tekee arkkitehtuurin suunnittelusta helpompaa ja järjestelmällisempää, ja sillä saavutetaan tarkemmat arkkitehtuurikuvaukset luottamusrajoja myöten (Tarandach & Coles 2020).

Uhkamallinnusta voidaan tehdä monella eri tavalla, mutta lopputuloksena työ kohdistuu mahdollisten uhkien tunnistamiseen. Uhkamallinnusprojektiin osallistuvat henkilöt suuntaavat huomionsa uhkiin, joita muut työkalut eivät havaitse, ja mahdollisten virheiden tekeminen pienenee. (Poston 2022, 12.) Keskeinen idea uhkamallinnuksessa on tunnistaa järjestelmää koskevat uhkat ja vähentää niitä. Alun perin menetelmä kehitettiin parantamaan ohjelmistoturvallisuutta suunnitteluvaiheessa, mutta myöhemmin menetelmää on laajennettu käsittelemään myös teollisuuden ohjausjärjestelmiä (ICS). ICS-ympäristöt käsittävät monia kriittisiä infrastruktuurin osia, kuten voimalaitokset, vedenjakelulaitokset ja tuotantolaitokset. Itsessään nämä ovat kriittisiä toimijoita yhteiskunnalle, minkä takia ne ovat alttiita kyberhyökkäyksille. Onnistuneet hyökkäykset voivat johtaa vakavaan häiriötilaan ja pahimmassa skenaariossa jopa ihmisuhreihin. (Khalil, Bahsi & Korõtko 2024.) Beyer (2020, 111) kysyi haastattelussaan Adam Shostackilta uhkamallinnuksen hyödyistä. Shostackin mukaan uhkamallinnus antaa strukturoidun ja kokonaisvaltaisen lähestymistavan turvallisuuden tarkastelemiseen. Uhkamallinnuksen avulla kyetään tunnis-

tamaan, mikä voi suunnittelussa epäonnistua, ja varmistumaan siitä, että uhkien tunnistaminen on tehty kattavasti.

Uhkamallinnuksessa käytetään edistyneitä tiedon keräämisen tekniikoita, joiden tarkoituksena on käsitellä tarkasteltava järjestelmä mahdollisimman kattavasti. (Ucedavélez & Morana 2015, 146.) Arkaluonteisen tiedon tallentaminen ja säilyttäminen tekee järjestelmästä otollisen kohteen hyökkääjälle. Hyökkääjä voi hyödyntää saamiaan tietoja ja/tai tehdä vaurioita uhrin järjestelmässä. Onnistuneet hyökkäykset aiheuttavat huomattavat taloudelliset tappiot, minkä takia uhkamallinnusta täytyisi tehdä jokaisessa sovelluskehityksen vaiheessa. Uhkien tunnistaminen on ensiaskel ennakoivaa tietoturvaprosessia. (Kohnfelder & Garg 1999.) Uhkien mallintamista tarvitaan muuttuvan toimintaympäristön vuoksi, koska turvallisuusongelmia on vaikeaa ratkaista yhdellä kertaa. Hyökkäykset ja haittaohjelmat kehittyvät jatkuvasti, ja uusia haavoittuvuuksia ilmaantuu, mikä vaatii pitkän aikavälin tarkastelua. Hyökkäysten ja tietoturvaongelmien kehittyminen johtaa siihen, että turvallisuuspuolen on kehityttävä samaan tahtiin. (Souppaya & Scarfone 2016, 9.) Nweke ja Wolthusen (2020) mainitsevat Souppayan ja Scarfonen tapaan, että uhkien kenttä muuttuu jatkuvasti. Hyökkäyskentän evoluutio on jatkuva prosessi, ja puolustuksen on pysyttävä tässä prosessissa mukana.

Uhkat vahingoittavat tietojärjestelmien luottamuksellisuutta, eheyttä tai saatavuutta informaation luvattoman paljastamisen, väärinkäytön, muuttamisen tai tietojen poistamisen kautta. Uhkamallinnus on prosessina oleellinen tekijä muuttuvan uhkien toimintaympäristön tarkastelussa, koska uhkamallinnus tarjoaa menetelmiä ja työkaluja uhkien havaitsemiseen, priorisointiin ja lievennyskeinojen toteuttamiseen. (Nweke & Wolhusen 2020.) Melkein jokainen järjestelmä kohtaa toiminnassaan uhkia, jotka voivat olla sisäisiä ja ulkoisia uhkia, joiden vaikutukset saattavat olla vakavia. Uhkamallinnus on kriittinen osa turvallista järjestelmänkehitystä, ja sen avulla varmistetaan, että jokainen mahdollinen uhka on huomioitu ja uhkille on tehty tarvittavat lievennystoimenpiteet. (Möckel & Abdallah 2011, 346.)

3.3 Strategioita

3.3.1 Omaisuuskeskeinen uhkamallinnus

Omaisuudella tarkoitetaan uhkamallinnuksen yhteydessä yleensä kolmea asiaa, jotka ovat

- hyökkääjän haluamat asiat
- asiat, joita puolustaja haluaa suojella
- yhteydet edellä mainittuihin tekijöihin.

Näitä kolmea tekijää ei tulisi käsitellä erillisinä tyyppeinä, vaan yhtenä kokonaisuutena, koska edellä mainituilla tekijöillä on kiinteä yhteys toisiinsa. (Shos-

tack 2014, 37.) Omaisuuskeskeinen lähestymistapa keskittyy riskeihin tai omaisuuteen liittyviin kokonaisuuksiin, jotka ovat yhteydessä tietojen menettämiseen tai liiketoiminnan resursseihin. Huomioitavaa on se, että tämä lähestymistapa ulottuu laajempaan kontekstiin kuin ainoastaan hyökkääjän aikomusten tunnistamiseen tai tietoturva-aukkojen havaitsemiseen. Painopisteenä on ymmärtää skenaarioiden vaikutus liiketoimintaan. (Ucedavélez & Morana 2015, 167.)

Hyökkääjän haluamat asiat ovat yleensä käyttäjien kirjautumistiedot, käyttäjien henkilökohtaiset tunnistamistiedot, luottokorttien tiedot ja arkaluonteiset yritystiedot. Puolustajan suojelemat asiat ovat muutakin kuin vain aineellisia asioita. Yrityksen maineen suojeleminen itsessään on vaikeaa teknisillä ratkaisuilla. Tämän takia puolustajan täytyy suojella asioita, koska asioiden suojeleminen suojaa myös yrityksen mainetta. Arkaluonteisten tietojen vuotamisella on laajamittaiset seuraukset, jotka hyvin suurella todennäköisyydellä vaikuttavat maineen menetykseen. Yhteydet puolustajan ja hyökkääjän haluamiin asioihin voidaan nähdä siltana asioiden välillä, koska tekniset komponentit avaavat reittejä muihin oleellisiin komponentteihin, joita hyökkääjä voi hyödyntää. Toisin sanoen hyökkääjä voi ”uida” komponentista komponenttiin saaden kriittistä tuhoa aikaan. (Shostack 2014, 38.)

Omaisuuskeskeisessä uhkamallinnuksessa on keskeistä tehdä luettelo suojeltavasta omaisuudesta. Tämän jälkeen puolustaja miettii, kuinka hyökkääjä voi päästä omaisuuteen käsiksi. Kun skenaariot on käyty läpi, niin puolustaja miettii, miten kyseisiä uhkia voidaan käsitellä. Omaisuusluettelon jälkeen puolustaja liittää jokaisen komponentin kohdejärjestelmään, minkä jälkeen mallinnetaan koko tietojärjestelmä. Mallinnusprosessissa on keskeistä tunnistaa luottamusrajat, koska yleensä uhkat ryhmittyvät luottamusrajojen ympärille. Luottamusrajat tekevät myös selkeäksi sen, mitä rajoja järjestelmässä on ja miten niitä suojataan. STRIDE tai hyökkääjäkeskeinen aivoriihi on oivallinen tapa tarkastella, miten kohteena olevaan järjestelmään voidaan hyökätä. (Shostack 2014, 39, 50–51.)

Riskin laskemiseksi voidaan käyttää DREAD-mallia, jonka tarkoituksena on havainnollistaa hyökkäyksen kannattavuutta liiketoiminnan perusteella. Työkalun avulla hyökkäykset voidaan jakaa seuraavasti:

- Vahinkopotentiaali *(engl. Damage Potential)*: hyökkäyksen kokonaisvaikutus tunnistetun haavoittuvuuden avulla
- Toistettavuus *(engl. Reproducibility)*: hyökkäyksen toistettavuus
- Hyödyntämispotentiaali *(engl. Exploitability)*: tunnistetun haavoittuvuuden hyödyntäminen
- Vaikutus käyttäjiin *(engl. Affected Users)*: vaikutusten ennustettavuus käyttäjiin, resursseihin tai sovellusympäristöön
- Löytämisen todennäköisyys *(engl. Discoverability)*: haavoittuvuuden löytäminen kohdeympäristöstä.

DREAD sisältää laadulliset matalan, keskitason ja korkean riskikuvaajat ja määrällisen riskiarvon 1, 2 ja 3. (Ucedavélez & Morana 2015, 167.)

3.3.2 Hyökkääjäkeskeinen uhkamallinnus

Hyökkääjäkeskeistä lähestymistapaa voidaan tarkastella seuraavien kysymysten avulla:

- kohde: mitä hyökkääjä haluaa?
- motiivi: miksi hyökkääjä haluaa tehdä tämän?
- vaikutus: mitä hyökkääjä voi tehdä kyseisillä tiedoilla?
- hyökkäys: miten hyökkääjä pääsee tavoitteeseensa?

(Ucedavélez & Morana 2015, 157.) Hyökkääjäkeskeinen lähestymistapa on yleinen tapa tehdä uhkamallinnusta. Luonnollisesti ajatellaan, että jos kukaan ei aio hyökätä, niin miksi edes tehdä uhkamallinnusta. Hyökkääjäkeskeinen lähestymistapa ei kuitenkaan ole ideaalein lähestymistapa, mutta tietyissä tapauksissa se on kuitenkin kannattavaa, esimerkiksi penetraatiotestauksen suunnittelussa. Kyseessä ovat samat skenaariot kuin omaisuuskeskeisessä uhkamallinnuksessa, esimerkiksi keskittyminen asiantuntijoihin, vähemmän tekniseen panostukseen prosessissa tai priorisointiin. (Shostack 2014, 40.)

Hyökkääjäkeskeisessä uhkamallinnuksessa käytetään tyypillisesti hyökkääjäluetteloita, joiden käyttö mahdollistaa aivoriihen tietoturva-asiantuntijoiden keskuudessa. Tämä lähestymistapa tekee mahdolliseksi tuoda esille erilaisia mahdollisuuksia, jotka ovat ihmiskeskeisiä, koska ihmisistä puhumisen avulla uhkista on helpompi tehdä todellisia. Vakoojan tai hakkerin näkökulmasta hyökkäysmenetelmät eroavat todennäköisesti, mutta uhkat ja niiden tuomat lieventämisstrategiat muuttuvat todellisiksi. (Shostack 2014, 40.) Beyerin (2020, 111) haastattelussa Shostack kertoo, että hyökkääjäkeskeisessä uhkamallinnuksessa hyökkääjille rakennetaan persoonia, jotka voivat hyökätä suojattavaan järjestelmään. Hyökkääjäkeskeinen lähestymistapa voi kuitenkin johtaa harhaan, koska hyökkääjien motiiveja, taktiikoita ja tekniikoita ei voida ennustaa täysin.

Hyökkääjäluettelot eivät itsessään ole kattava peruste tulosten tulkitsemiseen, koska aivoriihen tuomat harhat saattavat ohjata tekemään ratkaisuja, joita hyökkääjä ei koskaan tekisi. Ihmislähtöisyys on otettava huomioon, ja jokainen hyökkäysskenaario on tietyllä tapaa uniikki eikä niihin voida saada yksiselitteistä ratkaisua puolustajan aivoriihen tuloksena. Hyökkääjän motiivit ja strategiat saattavat erota puolustajan aivoriihen lopputuloksesta. (Shostack 2014, 41.)

Sovellusuhkien mallintamisessa puhutaan yleensä hyökkääjäkeskeisestä uhkamallinnuksesta, koska mallinnuksessa on tarkoitus listata tietoturva-aukkoja, joita hyökkääjä kykenee hyödyntämään kohdeympäristöä vastaan. Tavoitteena on tunnistaa ne uhkat, joiden avulla hyökkääjä kykenee tekemään tuhoa kohdejärjestelmään. Tietoturvakeskeisyys sovellusten uhkien mallintamisessa keskittyy tunnistamaan hyökkääjän motiivit ja identiteetin. Hyökkääjän identiteetin selville saaminen mahdollistaa sen, että puolustaja kykenee arvioimaan hyökkääjän kyvyt ja resurssit. Identiteettiä ei kuitenkaan ole tarkoitus rajata vain yksittäiseen henkilöön, koska se on muuttunut hyvin vaikeaksi kasvavien hyökkäysmenetelmien ja välityspalvelinpohjaisten hyökkäysten ansiosta. Hyökkääjän identiteetin yksilöiminen mahdollistaa sovellusuhkamalleissa

hyökkäyspuun liittämisen, tietoja hyökkääjän tyylistä ja taktiikasta. Haasteena hyökkääjän identiteetin selvittämisessä on ajan ja resurssien puute analyysin toteuttamiseksi. (Ucedavélez & Morana 2015, 156–157.)

3.3.3 Ohjelmistokeskeinen uhkamallinnus

Ohjelmistokeskeinen uhkamallinnus on paras strukturoitu lähestymistapa uhkamallinnuksen tekemiseen. Ohjelmistokeskeisessä uhkamallinnuksessa keskitytään rakennettavaan malliin tai käyttöön otettavaan järjestelmään. Keskeistä on dokumentoida tarkasteltava järjestelmä kaavioita hyödyntämällä, esimerkiksi UML-kaavioiden avulla. Kaavioiden tekeminen on erittäin hyödyllistä, koska sen avulla kyetään havaitsemaan, miten järjestelmän osat sulautuvat yhteen. Uhkien tarkastelun perusteella kaaviot selkiyttävät tarkasteltavaa järjestelmää, ja uhkien havaitsemisesta tulee huomattavasti helpompaa. (Shostack 2014, 41–42.) Ohjelmistokeskeistä uhkamallinnusta kutsutaan usein järjestelmäkeskeiseksi uhkamallinnukseksi, joka rakentuu ohjelmistomallien ympärille. Ohjelmistomallien rakentamisessa on hyödyllistä lisätä luottamusrajoja, koska ne helpottavat uhkien tarkastelemista, koska uhkat ovat yleensä luottamusrajojen ympärillä. (Hajrić, Smaka, Baraković & Husić 2020, 57.) Ohjelmistokeskeisessä uhkamallinnuksessa mallinnettava järjestelmä on keskiössä. Keskeistä on ymmärtää järjestelmän arkkitehtuuri hyvin ennen uhkamallinnusprosessin aloittamista. Järjestelmät saattavat olla kompleksisia, ja mallinnettaessa on syytä purkaa monimutkainen järjestelmä laajempiin kokonaisuuksiin. (Nweke & Wolhusen 2020.) Yleensä ohjelmistokeskeistä uhkamallinnusta tehdään ohjelmistosuunnittelun aikana haavoittuvuuksien vähentämiseksi (Souppaya & Scarfone 2016, 9).

Ohjelmistokeskeisessä uhkamallinnuksessa järjestelmä puretaan esimerkiksi prosesseihin, tietokantoihin, tietovirtoihin, luottamusrajoihin ja ulkoisiin kokonaisuuksiin uhkien tunnistamista varten. Tarkoituksena on etsiä erilaisia hyökkäystyyppejä järjestelmän jokaista elementtiä vastaan. UML-kaavioiden (*engl. Unified Modeling Language*) lisäksi voidaan käyttää tietovirtakaavioita (*engl. data Flow Diagram*). Tietovirtakaaviota käytetään yleensä ohjelmistokehityksen elinkaaren suunnitteluvaiheessa, minkä tarkoituksena on kuvata elementtien vuorovaikutusta keskenään ja muiden järjestelmien kanssa. (Satapathy 2014, 7–8, 21.) Ohjelmistosuunnittelussa tietoturvauhkien olemassaololla on kriittinen vaikutus järjestelmän luotettavaan ja turvalliseen käyttöön, sekä toimintaan (Rouland, Hamid & Jaskolka 2021). Ohjelmistokeskeisessä lähestymistavassa on oleellista ottaa ohjelmistokehittäjät mukaan uhkamallinnukseen, koska heillä on paras tuntemus kehittämästään järjestelmästä (Shostack 2014, 43). Taulukossa kolme on esitetty tietovirtakaavion elementit, joita käytettiin tässä tutkimuksessa.

TAULUKKO 3 Tietovirtakaavion elementit

Elementti	**Muoto**	**Selitys**	**Esimerkki**
Prosessi	Ympyrä	Käynnissä oleva koodi	Ohjelmointikielet, esimerkiksi Python
Tietovirta	Nuoli	Kommunikointi järjestelmän osien välillä	Verkkoyhteydet, esimerkiksi HTTP
Tietokanta	Suorakulmio ilman pystyviivoja	Varastoivat dataa	Tiedostot, tietokannat ja jaetut muistisegmentit
Ulkoinen kokonaisuus	Suorakulmio	Tekijät, jotka eivät ole järjestelmän kontrollissa, esimerkiksi käyttäjät	Asiakas, käyttäjät jne.
Luottamusraja	Katkoviiva	Raja, jossa kaksi tai useampi elementti vaihtaa dataa keskenään	Esimerkiksi dataa liikkuu lokien ja tietokannan välillä

(Ks. Shostack 2014, 45.)

4 STRIDE-UHKAMALLINNUS

Neljännessä pääluvussa käsitellään sitä, miten uhkia voidaan tunnistaa STRIDE:n avulla ja miten näitä uhkia voidaan lieventää. Viimeisessä alaluvussa keskitytään tyypillisimpiin uhkiin ja niiden lieventämiskeinoihin. Alaluvut alkavat taulukoilla, joissa kuvataan STRIDE ja sen elementit. Viimeisessä alaluvussa kuvataan taulukon avulla tyypillisimmät uhkat ja niiden lieventämiskeinot. Taulukoissa on nostettu esille keskeiset tekijät, jotka on avattu tarkemmin tekstissä.

4.1 Uhkien tunnistaminen STRIDE:n avulla

STRIDE on uhkamallinnusmenetelmä, jonka Microsoft otti käyttöön osana järjestelmien kehityksen elinkaarta. Microsoft ja monet tunnetut organisaatiot ovat käyttäneet STRIDE-metodologiaa vuosikymmeniä uhkien tunnistukseen. STRIDE koostuu kuudesta elementistä, jotka ovat huijaus *(engl. spoofing)*, peukalointi *(engl. tampering)*, kiistäminen *(engl. repudiation)*, tietojen paljastaminen *(engl. information disclosure)*, palvelunesto *(engl. denial of service)* ja oikeuksien korottaminen *(engl. elevation of privilege)*. Nämä kuusi kokonaisuutta analysoidaan kuuteen ominaisuuteen, jotka ovat todennus *(engl. authentication)*, eheys *(engl. integrity)*, kiistämättömyys *(engl. non-repudiation)*, luottamuksellisuus *(engl. confidentiality)*, saatavuus *(engl. availability)* ja valtuutus *(engl. authorization)*. Tätä metodologiaa hyödynnetään, kun järjestelmä mallinnetaan tietovirtakaavioksi *(engl. dataflow diagram)*. (Khalil, Bahsi & Korõtko 2024.) STRIDE mahdollistaa tietoturvallisen lähestymistavan sovellusten uhkien tunnistamiseen, koska se on helppokäyttöinen ja ymmärrettävä tietoturva-ammattilaisille ja ohjelmistokehittäjille (Ucedavélez & Morana 2015, 160). STRIDE mahdollistaa uhkien ennakoivan havaitsemisen ja nopean reagoinnin tapahtumiin (Mathew & Kazi 2024). STRIDE-metodologiaa voidaan käyttää omaisuus- ja ohjelmistokeskeisessä uhkamallinnuksessa (Shostack 2014). Tässä tutkimuksessa menetelmää tarkastellaan ohjelmistokeskeisestä näkökulmasta.

TAULUKKO 4 STRIDE

Uhka	**Tietoturvan elementit**	**Selitys**	**Vaikuttaa useimmiten**	**Esimerkki**
Spoofing *(Huijaus)*	Authentication *(Todennus)*	Huijataan olevamme joku, joka emme ole	Prosessit, ulkoiset resurssit, henkilöstö	Väitetään, että olemme tietoturvayrityksen konsultteja ja tarvitsemme tietokoneen IP-osoitteen.
Tampering *(Peukalointi)*	Integrity *(Eheys)*	Muutetaan jotain, esimerkiksi verkossa tai muistissa	Tietokannat, tietovirrat, prosessit	Muokataan, lisätään tai poistetaan tietoja esimerkiksi tietokannasta
Repudiation *(Kiistäminen)*	Non-Repudiation *(Kiistämättömyys)*	Väitetään, ettei ole tehty jotain toimenpidettä, oli se sitten totta tai valetta	Prosessit	En ole avannut tuota linkkiä
Information Disclosure *(Tietojen paljastaminen)*	Confidentiality *(Luottamuksellisuus)*	Tiedon antaminen henkilölle, jolla ei ole oikeutta sitä nähdä	Prosessit, tietokannat, tietovirrat	Pääsyn salliminen tiedostoihin, sähköpostiin tai tietokantoihin
Denial of Service *(Palvelunesto)*	Availability *(Saatavuus)*	Estetään/hidastetaan järjestelmän käyttö	Prosessit, tietokannat, tietovirrat	Ruuhkautetaan tietoliikenne ja tämän avulla hidastetaan/estetään järjestelmän käyttö
Elevation of Privilege *(Oikeuksien korottaminen)*	Authorization *(Valtuutus)*	Henkilö tekee, jotain, mihin hänen käyttövaltuutensa eivät riittäisi	Prosessit	Tavallinen käyttäjä kykenee tekemään toimintoja järjestelmänvalvojan oikeuksilla

(Ks. Shostack 2014, 62–63.)

Spoofing

Spoofing tarkoittaa, että teeskennellään, että ollaan joku, joka ei oikeasti olla (Shostack 2014, 64). Hyökkääjä on saanut käyttäjän kriittiset tiedot tai haltuunsa jotain, minkä avulla hän pääsee järjestelmään sisälle. Uhkat liittyvät yleensä siihen, että hyökkääjä esiintyy luotettavana käyttäjänä tai resurssina päästäkseen järjestelmään sisälle ja vaarantamaan järjestelmän turvallisuuden. (Kohnfelder & Garg 1999.) Hyökkäykset jäljittelevät toista identiteettiä tai esittävät olevansa se. Identiteetti ei rajoitu pelkästään käyttäjiin tai asiakkaisiin, vaan uhkat voivat kohdistua myös protokollien tai niitä käyttävien palvelinten manipulointiin. Huijausuhkat ovat kiinteässä yhteydessä myös muihin todennusmenetelmiin, joita käytetään monimenetelmätunnistuksessa. Nämä kattavat digitaaliset sertifikaatit, biometriset hallintalaitteet ja kaikki muut tunnistustavat, jotka voidaan siepata tai arvata. (Ucedavélez & Morana 2015, 160.)

TAULUKKO 5 Spoofing

Kohde	**Esimerkkikeino**
Prosessin huijaaminen	Tiedoston luominen ennen prosessia
	Uudelleen nimeäminen tai linkittäminen
	Uudelleen nimeäminen
Tiedoston huijaaminen	Paikalliseen hakemistoon uuden tiedoston luominen
	Linkin luominen ja sen muuttaminen
	Monien tiedostojen luominen hakemistoon
Tietokoneen huijaaminen	ARP-, IP- ja DNS-huijaus
	IP-osoitteen uudelleen ohjaus
Henkilön huijaaminen	Käyttäjätilin varastaminen

(Ks. Shostack 2014, 65.)

Mikäli hyökkääjä ehtii luomaan tiedoston ennen prosessia, niin hän kykenee hyödyntämään koodin tietoja, koska koodi tulkitsee, että ne ovat luotettavia. Tiedosto-oikeuksien hallinnan puutteellisuus mahdollistaa sen, että hyökkääjä voi luoda päätepisteen, jonka avulla hyökkääjä kykenee luomaan ja muokkaamaan tiedostoja. Väärennettyjen tiedostojen tai prosessien luominen voi toimia etäkoneella, mikäli hyökkääjä on onnistunut saamaan järjestelmänvalvojan oikeudet tai huijaamalla olevansa se. (Shostack 2014, 65–66.) Tiedostojen rajoittamattomien lataushaavoittuvuuksien takia hyökkääjä kykenee lataamaan ja suorittamaan haitallisia komentosarjoja palvelimissa. Jos hyökkääjä on onnistunut lataamaan tiedoston, mahdollistaa se monien hyökkäysten toteuttamisen. (Huang, Li, Zhang & Dai 2019.)

Hyökkääjä voi huijata koneita useilla eri menetelmillä, esimerkiksi huijaamalla ARP-pyyntöjä ja IP- ja DNS-paketteja. ARP-pyyntöjen on oltava paikallisia, ja IP-osoitteen huijaamisessa lähdeosoite väärennetään, joten paketit näyttävät tulevan jostain, mistä ne eivät oikeasti tule. Hyökkääjä voi huijata DNS-kyselyn vastauksen kyselyyn, jonka käyttäjän oletetaan tekevän. DNS-huijaus onnistuu, kun käyttäjä tekee eteen- ja taaksepäin haun. DNS-tietueiden hallinta on mahdollista, kun käyttäjän koodi tekee käänteisen haun, esimerkiksi IP-osoitteen kääntäminen täysin hyväksytyksi verkkotunnukseksi (*engl. Fully Qualified Domain Name, FQDN*). (Shostack 2014, 66.) Hyvin yleistynyt hyökkäystekniikka huijausuhkissa on Man-in-the-middle (MITM-hyökkäys), jossa hyökkääjä saa luvattoman pääsyn kahden tai useamman päätepisteen väliseen viestintäkanavaan, jossa viestintä tapahtuu. Päästyään viestintäkanavaan hyökkääjä voi salaa tarkkailla, varastaa ja muokata dataa. (Sowah, Ofori-Amanfo, Mills & Koumadi 2019; Kumar, Dey, Guelton, Bali & Singh 2024, 3.)

Tyypillisesti ihmisten huijaamisessa tavoitteena on saada uhrin käyttäjätili haltuun. Tietojenkalastelu on yksi yleisimmistä keinoista saada pääsy uhrin tilille. (Shostack 2014, 66.) Evil twin -hyökkäys on eräs keino huijata käyttäjiä.

Hyökkäys alkaa, kun käyttäjä yhdistää laitteensa valetukiasemaan, minkä jälkeen hyökkääjä kykenee seuraamaan käyttäjän dataliikennettä. Evil Twin -hyökkäys avaa mahdollisuuden muille hyökkäyksille, muun muassa SSL Strip ja DNS- ja IP-huijaukselle. (Muthalagu & Sanjay 2021.) Klassinen esimerkki on troijalainen hevonen, jossa hyökkääjä on luonut aidolta vaikuttavan kopion järjestelmän kirjautumissivusta. Käyttäjät syöttävät kirjautumistietonsa haitalliselle sivulle, ja hyökkääjä pääsee käsiksi käyttäjien tietoihin. (Kohnfelder & Garg 1999.)

Tampering

Tampering tarkoittaa jonkin asian muuttamista esimerkiksi levyllä, verkossa tai muistissa (Shostack 2014, 67). Tietojen siirtäminen tai muokkaaminen onnistuu esimerkiksi lähettämällä tietoliikenteeseen tunnistamattoman datapaketin, ja pienikin muutos kriittisessä tiedostossa muodostaa merkittävän uhkan. (Kohnfelder & Garg 1999.) Hyökkäykset voivat kohdistua siirrettävään- tai staattisesti tallennettuun dataan. Siirrettävään dataan kuuluu kaikki langaton tiedonsiirto, esimerkiksi Wi-Fi, ja staattisesti tallennettuun dataan kuuluvat muun muassa konfiguraatiotiedostot ja tietokantoihin tallennettu data. (Ucedavélez & Morana 2015, 160–161.)

TAULUKKO 6 Tampering

Kohde	**Esimerkkikeino**
Tiedoston muuttaminen	Tiedoston muokkaaminen ja käyttäjä luottaa kyseiseen tiedostoon
	Tiedoston muuttaminen tiedostopalvelimessa
	Linkkien muuttaminen
Muistin muuttaminen	Koodin muuttaminen
	API:n toimittamien tietojen muuttaminen
Verkkoliikenteen muuttaminen	Tietovirran muuttaminen
	Verkon yli liikkuvan datan muuttaminen

(Ks. Shostack 2014, 67.)

Hyökkääjä kykenee muokkaamaan tiedostoja, joihin hänellä on kirjoitusoikeus. Tiedosto on mahdollisesti haavoittuva, jos koodin on perustuttava tiedostoihin, joita myös muut kirjoittavat. Yleisesti muokkaaminen tapahtuu paikallisella levyllä. Tietoturva voi vaarantua, kun hyökkääjä muokkaa tiedostoja esimerkiksi puutteellisten käyttöoikeuksien ja/tai käyttöoikeuksien korottamisen takia. (Shostack 2014, 68.) Hyökkääjä voi käyttää SQL-injektiota, jossa hyökkääjä suorittaa haitallisia kyselyitä tietokantaan, ja tämän avulla hyökkääjä kykenee muuttamaan ja lisäämään tietokantaan uusia tiedostoja. (Yunus ym. 2018.) SQL-injektiolla on myös kiinteä yhteys tietojen paljastamisuhkiin, koska yleisesti ottaen onnistunut SQL-injektiohyökkäys mahdollistaa hyökkääjän näkemään tie-

toja, joita hyökkääjä ei saisi nähdä, esimerkiksi tietokannan rakenteet, kirjautumistiedot tai muut sovelluksen tiedot (Crespo-Martínez ym. 2023, 3).

Muistin muuttaminen voi onnistua, jos hyökkääjällä on samat valtuudet kuin käyttäjällä. Muistin muokkaamisen puolustaminen on hankalampaa, jos ohjelmointirajapinta käsittelee dataa viitteellä. Tämän ansiosta hyökkääjällä on mahdollisuus muokata dataa suojaustarkistuksen jälkeen. (Shostack 2014, 68.) Tiedostojen tietojen muuttaminen ilman valtuutustarkistuksia muodostaa uhkan muun muassa puskurin ylivuoto -hyökkäyksille (Kohnfelder & Garg 1999).

Verkon muuttamiseen liittyy useita erilaisia menetelmiä, kun hyökkääjä tuo uhrin järjestelmästä dataa omalle koneelleen ja data lähetetään eteenpäin ehjänä tai muutettuna. Wi-Fi:n ja bluetoothin ansiosta ei välttämättä tarvita erityisiä toimenpiteitä, koska dataa liikkuu huomattavan paljon. (Shostack 2014, 68.) BGP-kaappauksen avulla voidaan peukaloida internetin reititysprotokollaa, jossa kaapataan IP-osoiteavaruuksia ja ohjataan verkkoliikennettä eri tavoin (Zhang, Zhang & Hu 2007, 2). Chen, Liu ja Su (2024) korostavat artikkelissaan tietoliikenteen muuttamiseen liittyviä uhkia. Kyberfyysiset järjestelmät sisältävät ohjaus-, viestintä- ja tietojenkäsittelyn kaltaisia teknologioita, ja onnistuneet hyökkäykset kyberfyysisiin järjestelmiin turmelevat päätöksentekoa ja valvontaa. Tämä johtaa mahdollisesti tuotannon tehokkuuden heikkenemiseen ja jopa laitevaurioihin. Huomioitavaa on se, että hyökkääjä kykenee käynnistämään nämä hyökkäykset, mikäli hyökkääjä voi syöttää ja/tai muuttaa tietoliikenteeseen lähetettävää dataa.

Repudiation

Repudiation tarkoittaa toimintojen kiistämistä tai tilannetta, jossa käyttäjä väittää, ettei ole vastuussa tapahtuneesta. Väite saattaa olla totta tai valetta. (Shostack 2014, 69.) Laittoman toiminnan suorittaminen ilman jäljitettävyyttä muodostaa merkittävän uhkan organisaation näkökulmasta. Todistusaineiston puuttuminen aiheuttaa sen, ettei toimenpiteitä voida todistaa todeksi tai epätodeksi. (Kohnfelder & Garg 1999.) Kiistämisuhkat ovat yleisiä melkein jokaisessa tapauksessa, koska hyökkääjät eivät halua jättää todistusaineistoa tutkintaa varten. Muutamissa tapauksissa kiistämisuhkat ovat epätodennäköisiä, jolloin kokemattomat hyökkääjät eivät ota jälkien peittämistä toiminnassaan huomioon tai hyökkääjä ei välitä siitä, jääkö toimenpiteestä todistusaineistoa, koska mahdollinen hyöty on suurempi kuin rangaistus. (Ucedavélez & Morana 2015, 161.)

TAULUKKO 7 Repudiation

Kohde	**Esimerkkikeino**
Toiminnan kiistäminen	”En ole avannut tuota”
	”En ole vastaanottanut tuota”
	Toisen henkilön tilin käyttäminen
Hyökkääminen lokeihin	Hyökkäys lokeihin sekoittaakseen lokit/lokin luku-

	koodin/henkilön, joka lukee lokeja

(Ks. Shostack 2014, 69.)

Tarkasteltaessa toiminnan kiistämistä on syytä huomioida se, että suurimmassa osassa tapauksista, joissa henkilö kiistää tehneensä jotain, on harvoin hyökkääjä. Esimerkiksi roskapostisuodatin on voinut estää sähköpostin tai henkilö on huomaamattaan avannut linkin. (Shostack 2014, 70.) Digitaalisten allekirjoitusten yhdistäminen identiteetin tunnistukseen tai aikaleimoihin voi parantaa turvallisuutta ja ehkäistä kiistämisongelmia (Fang ym. 2020, 13).

Lokien puuttuminen on erittäin huono skenaario, koska kiistämisongelmia ei voida todentaa (Shostack 2014, 69). Tähän liittyen havaitsemattomat murtautumisyritykset käyttäjien tileille ja kirjautumistarkastusten puute muodostavat kriittisen uhkan (Kohnfelder & Garg 1999). Wiesner (2023, 139–140) mainitsee, että moduulilokin käyttöön ottaminen mahdollistaa hyökkääjien komentojen havaitsemisen Windows-järjestelmässä. Moduuliloki voidaan ottaa käyttöön nykyisessä istunnossa, tai se voidaan ottaa käyttöön pysyvästi. Huomioitavaa on se, että vianmäärityksessä kannattaa käyttää nykyistä istuntoa, mutta hyökkääjän komentojen havaitsemiseen pysyvä moduulilokin käyttöönotto on kannattavampaa. Nykyisen istunnon asettaminen voidaan tehdä seuraavalla komennolla:

```
> Get-Moduuli Tapahtumalista
> (Get-Moduuli Tapahtumalista) .LokiPipelineSuoritusTiedot = $true
> (Get-Moduuli Tapahtumalista) .LokiPipelineSuoritusTiedot
True
```

Mikäli järjestelmänvalvoja ei halua ottaa Windows-laitteilla moduulia käyttöön manuaalisesti jokaisessa koneessa, niin ryhmäkäytännön ottaminen käyttöön on tähän sopiva vaihtoehto. Toimenpiteessä luodaan uusi ryhmäkäytäntöobjekti (*engl. Group Policy Object, GPO*).

Information Disclosure

Information Disclosure tarkoittaa, että oikeudeton henkilö näkee tietoja, joita hänen ei kuuluisi nähdä (Shostack 2014, 70). Tietojen paljastumisuhkia ovat käyttäjien kyky lukea tietoja, joihin heille ei ole myönnetty oikeutta, ja hyökkääjän kyky lukea tietoja esimerkiksi siirrettäessä dataa koneiden välillä. Tämä uhka eroaa huijausuhkista siinä, että hyökkääjän ei tarvitse huijata päästäkseen käsiksi tietoihin. (Kohnfelder & Garg 1999.) Yleensä tämän tyypin hyökkäykset keskittyvät tietojen luvattomaan hankkimiseen, esimerkiksi luottokorttitietoihin, potilastietoihin, käyttäjätunnuksiin ja niin edelleen. Hyökkääjän motiivit liittyvät todennäköisesti tietojen jälleenmyyntiin. (Ucedavélez & Morana 2015, 161.)

TAULUKKO 8 Information Disclosure

Kohde	**Esimerkkikeino**
Tietojen paljastaminen prosessista	Datan poimiminen virheilmoituksista
	Virheilmoitusten lukeminen käyttäjätunnuksista/salasanoista suuriin tietokantatauluihin
Tietojen paljastaminen tietokannasta	Puuttuvien käyttövaltuusmääritysten hyödyntäminen
	Puutteellisten tietokantaoikeuksien hyödyntäminen
	Salausavainten löytäminen
	Tiedostojen lukeminen, kun ne kulkevat verkossa
	Tietojen hakeminen lokeista
Tietojen paljastaminen tietovirroista	Verkossa olevan datan lukeminen
	Liikenteen uudelleenohjaus, joka mahdollistaa datan lukemisen
	Paljastukset tarkkailemalla DNS:ää

(Ks. Shostack 2014, 70–71.)

Tietojen paljastaminen prosessista voi tapahtua, kun prosessi vuotaa muistiosoitteita tai suoritetaan arkaluonteisen tiedon- tai tiettyjen yksityiskohtien poimiminen virheilmoituksista (Shostack 2014, 71).

Suurin syy tietojen paljastumiseen tietokannasta johtuu puutteellisista turvamekanismeista, esimerkiksi lupien asettamatta jättämisellä. Erikoistapauksena ovat salausavaimet, jotka altistavat lisähyökkäyksille. Tiedostonimien tiedot usein unohdetaan tarkasteltaessa lisähyökkäyksiä. Tietokannassa voi olla tiedosto, esimerkiksi Bob_varoituksen_antaminen.pdf. Kyseinen tiedostonimi paljastaa jo tietoja. (Shostack 2014, 71.)

Tiedon virratessa verkon yli tietovirrat ovat hyvin alttiita tietojen paljastushyökkäyksille. Ennalta tiedusteltuun tietokoneeseen voidaan hyökätä, kun käytetään esimerkiksi epäluotettavan tarjoajan pilvipalvelua. Verkossa olevaa tietoliikennettä voidaan uudelleen ohjata, kun hyökkääjä huijaa ohjausprotokollaa. (Shostack 2014, 72.)

Cross-site scripting -hyökkäyksen (XSS-hyökkäyksen) avulla hyökkääjä lähettää haitallista koodia käyttäjän selaimeen. Hyökkäyksen avulla hyökkääjä kykenee varastamaan esimerkiksi käyttäjän kirjautumistiedot. Tämän avulla hyökkääjä kykenee näkemään tietoja, joita hänen ei kuuluisi nähdä. (Sarmaha, Bhattacharyyaa & Kalitab 2018.)

Denial of Service

Denial of Service -hyökkäys kuluttaa järjestelmän kapasiteettia, jota tarvitaan palveluiden tarjoamiseen (Shostack 2014, 72). Teoriassa tietoliikennepohjainen palvelunestohyökkäys voi tapahtua millä tahansa OSI-mallin tasolla, ja haas-

teen tekee se, että palvelunestopohjaisia hyökkäyksiä on haastavaa puolustaa. Hyökkääjän motiivina on tyypillisesti sivuston tai käytettävyyden häiritseminen ja mahdollisesti haittaohjelmien tuominen järjestelmään sisälle palvelunestohyökkäyksen avulla. Hajautetut palvelunestohyökkäykset ovat tehokkaampia, koska haitallinen liikenne tulee eri paikoista. (Ucedavélez & Morana 2015, 161–162.)

TAULUKKO 9 Deanial of Service

Kohde	**Esimerkkikeino**
Palvelunesto prosessia vastaan	Muistin- ja prosessorin kuluttaminen
Palvelunesto tietokantaa vastaan	Tietokannan täyttäminen
	Pyyntöjen lähettäminen järjestelmän hidastamiseksi
Palvelunesto tietovirtaa vastaan	Verkkoresurssien kuluttaminen

(Ks. Shostack 2014, 72.)

Tietoliikenteeseen kohdistuvat palvelunestohyökkäykset voidaan jakaa kolmeen luokkaan, jotka ovat volumetriset-, protokolla- ja sovellustason hyökkäykset. Volumetrisilla hyökkäyksillä tarkoitetaan sitä, kun tietoliikenne määrä ylittää palvelun kapasiteettiresurssin. (Kyberturvallisuuskeskus 2016a, 1.) Volumetriset palvelunestohyökkäykset muodostavat kasvavan uhkan nykyajan internetissä. Suurin osa volumetrisista palvelunestohyökkäyksistä perustuu pakettityyppeihin, jotka ovat samanlaisia aitojen pakettien kanssa. Eron tekee se, että hyökkääjä lähettää paketteja enemmän kuin käyttäjät. (DeLaughter 2023.) Tietoliikenteen täyttyminen tarkoittaa hyökkäyksen kohteena olevan verkon täyttämistä haitallisella liikenteellä, joka estää normaalia liikennettä pääsemästä hyökkäyksen kohteena olevaan järjestelmään (Khajuria & Srivastava 2013, 1).

Protokollahyökkäykset tarkoittavat resurssien kuluttamista muun muassa palvelimissa, palomuureissa ja kuormanjakajissa (Kyberturvallisuuskeskus 2016a, 1). Resurssien kuluttamisen tavoitteena on sitoa hyökkäyksen kohteena olevan järjestelmän resursseja. Hyökkäys kohdistuu tyypillisesti palvelimeen tai prosessiin, jolloin ne eivät kykene käsittelemään palvelupyyntöjä. (Khajuria & Srivastava 2013, 1.)

Sovellustason hyökkäyksissä hyödynnetään kohteen haavoittuvuuksia ja tämän avulla kaadetaan kohde (Kyberturvallisuuskeskus 2016a, 1). *Apache Range Headerin* haavoittuvuus altisti palvelunestohyökkäykselle. Haavoittuvuuden ansiosta hyökkääjä kykeni lähettämään haitallisia http-pyyntöjä, jotka aiheuttivat muistin loppumisen ja palvelimen kaatumisen. Nykyään kyseinen haavoittuvuus on korjattu. (Apache 2011.)

Elevation of Privilege

Elevation of Privilege tarkoittaa sitä, että henkilö voi tehdä jotain, mihin hänellä ei ole valtuuksia, esimerkiksi koodin suorittaminen järjestelmänvalvojan oi-

keuksilla (Shostack 2014, 73). Oikeuksien korottaminen hyökkääjän näkökulmasta tekee sen, että hyökkääjällä on muun muassa järjestelmänvalvojan oikeudet, mikä aiheuttaa mahdollisesti suuria vaurioita järjestelmässä. (Kohnfelder & Garg 1999.) Rajoitetut käyttöoikeudet pienentävät hyökkääjän onnistumistodennäköisyyttä. Oikeuksien korottaminen on hyökkääjän näkökulmasta hyödyllinen keino tehdä tuhoa kohdejärjestelmään, koska oikeuksien korottamisella voidaan ottaa käyttöön lisäkomentoja ja tuoda lisää haittaohjelmia. Tyypillisesti hyökkääjät käyttävät avointen lähteiden tiedustelua ja ylipäätään tiedustelumenetelmiä hyväkseen saadakseen lisätietoa sovellusympäristöstä, infrastruktuurista, tietokannoista ja protokollista. (Ucedavélez & Morana 2015, 163.)

TAULUKKO 10 Elevation of Privilege

Kohde	**Esimerkkikeino**
Oikeuksien korottaminen muuntamalla prosessia	Syötteiden lähettäminen, mitä koodi käsittelee virheellisesti
	Pääsy kirjoittamaan tai lukemaan muistia
Oikeuksien korottaminen ohitettujen valtuustarkastusten kautta	Valtuutusta ei tarkisteta jokaisella tasolla
Oikeuksien korottaminen virheellisten valtuustarkastusten kautta	Valtuutusta ei tarkisteta jokaisella tasolla
Oikeuksien korottaminen muuttamalla dataa	Bittejä muokkaamalla tekevät toimintoja, joita valtuutetun käyttäjän ei ole aikomus tehdä

(Ks. Shostack 2014, 73.)

Prosessin muuntaminen mahdollistaa hyökkääjän saavan vaikutusvaltaa tai ohjelman tietovirran hallinnan. Tietovirran seuraaminen on tärkeää, koska sen avulla kyetään huomaamaan, missä oikeuksia voidaan korottaa. Valtuutustarkistusten epäonnistuminen tarkoittaa sitä, ettei tarkistuksia tehdä jokaisella tasolla. (Shostack 2014, 74.)

4.2 Uhkien lieventämiskeinot

Uhkien tunnistamisen jälkeen uhkamallintajilla on lista erilaisia uhkia, joiden käsittelyyn on neljä erilaista tapaa: uhkien lieventäminen, poistaminen, siirtäminen ja hyväksyminen. Uhkien lieventäminen tarkoittaa toimenpiteitä, jotka vaikeuttavat hyökkääjien hyödyntämästä uhkaa. Vahvojen salasanojen vaatiminen on yksi tapa hankaloittaa hyökkääjän keinoja murtautua järjestelmään. (Shostack 2014, 12.)

Uhkien poistaminen tarkoittaa käytännössä ongelmallisten toiminnallisuuksien poistamista. Uhka saattaa olla esimerkiksi järjestelmän toiminnallisuudessa, ja tietyissä tapauksissa toiminnallisuuden poistaminen on kustannustehokkaampi vaihtoehto kuin toiminnallisuuden korjaaminen. (Shostack 2014, 12.)

Uhkien siirtäminen tarkoittaa sitä, että uhka siirretään jonkun muun henkilön tai tahon käsiteltäväksi. Luottamusrajojen valvonta palomuurituotteilla on yksi tapa siirtää riskejä jonkun muun käsiteltäväksi. (Shostack 2014, 13.) Kybervakuutukset ovat myös yksi tapa siirtää uhkia muun tahon käsiteltäväksi. Kybervakuutus voi sisältää muun muassa asiantuntija-apua tietojen palauttamiseen. (OP 2024.)

Riskien hyväksyminen on myös yksi tapa uhkien käsittelemiseksi. Jokainen järjestelmä sisältää jonkin verran riskejä, ja jokaisen riskin poistaminen on hyvin vaikeaa ja kustannuksia nostattavaa. Tietyissä tapauksissa riskin poistamisen kustannukset voivat olla suurempia kuin riskistä aiheutunut vahinko. Jokainen riski täytyy kuitenkin käsitellä, mutta jos todetaan riskin arvioinnin jälkeen, että sen vaikutus on hyvin pieni, niin riski on hyväksyttävä ja toimenpiteitä voidaan jatkaa. Kun riski on hyväksytty, niin siitä ei tarvitse enää huolehtia. Mikäli riski aiheuttaa huolta, niin siinä tapauksessa se ei ole käsitelty ja riski vaatii uudelleen tarkastelemista. (Shostack 2014, 13.)

Spoofing

TAULUKKO 11 Huijausuhkien lieventämisstrategia ja -tekniikka

Uhka	**Strategia**	**Tekniikka**
Prosessin huijaaminen	Käyttöjärjestelmän hyödyntäminen	• Käyttöjärjestelmillä on yleensä jokin todennusprotokolla, joka on käyttöjärjestelmän ominaisuus • Windows-todennus (NLTM) • Kerberos
Tiedoston huijaaminen	Käyttöjärjestelmän hyödyntäminen	• Täydelliset polut • ACL • Digitaaliset allekirjoitukset • Hash-toiminnot
Verkko-osoitteiden huijaaminen	Salausteknologia	• DNSSEC • HTTPS • IPsec • SSH-isäntäavaimet • PKI, esimerkiksi TLS-varmenteet
Henkilön huijaaminen	Tunnistus ja todennus	• Käyttäjänimet, oikeat nimet • Salasanat • Pääsyoikeudet • MFA

(Ks. Shostack 2014, 14, 148.)

Käyttöjärjestelmillä on yleensä todennusprotokolla, joista esimerkkeinä käytetään NTLM:ää ja Kerberosta. NTLM-todennusprotokolla todentaa käyttäjät ja

tietokoneet vastausmekanismien perusteella ja osoittaa palvelimella käyttäjän aitouden. NTLM-todennus on edelleen käytössä, mutta tästä todennusprotokollasta kehittyneempi versio on Kerberos. (Microsoft 2016.) Kerberos on todennusprotokolla, jota käytetään oletusprotokollana Windows 2000:ta uudemmissa Windows-versioissa. Kerberos on hyvin pitkälti korvannut NTLM-todennusprotokollan, koska Kerberos tarjoaa paremman suorituskyvyn, turvallisemman vaihtoehdon ja monimenetelmätunnistautumisen (*engl. Multi Factor Authentication, MFA*). (Microsoft 2018.) Kerberos on hajautettu pääsynvalvontajärjestelmä, josta on tullut osa todennuksen perusmekaniikkaa Windows- ja Linux-käyttöjärjestelmille. Todennuksen perusmekaniikka korostuu tilanteissa, joissa koneet jakavat resursseja lähiverkon kautta. Perusrakenteena on se, että Kerberos mahdollistaa skaalautuvan pääsynhallinnan. (Anderson 2020, 141–142.) Kerberos on laajalti käytetty todennusprotokolla, joka suojaa hajautettuja palveluita esineiden internetissä (*engl. Internet of Things)* ja suurdataa (*engl. Big Data)*. Hajautetussa skenaariossa entiteettien on todistettava henkilöllisyytensä luotettavalle kolmannelle osapuolelle käyttämällä, esimerkiksi salaisia avaimia. Huomioitavaa on se, että Kerberoksen heikkoutena on tunnistetietojen paljastuminen, mikä voi vaarantaa järjestelmän turvallisuuden. Ongelman ratkaisemiseksi tutkijat ovat ehdottaneet käytettäväksi hajautettua Kerberos Secure Service-Management (DKSM) -protokollaa, joka perustuu lohkoketjuteknologiaan ja salakirjoituspolitiikkaan. (Chen ym. 2023.)

Mikäli tiedosto sisältää arkaluonteista tietoa, niin avaamisen jälkeen on syytä tarkistaa suojauselementit, esimerkiksi käyttöoikeudet ja tiedoston omistaja. Tiedoston kuvauksen varmistaminen mahdollistaa kilpailuolosuhteiden välttämisen, koska kuvauksen tarkistaminen saattaa auttaa siinä, ettei hyökkääjä voi muuttaa tiedoston oikeuksia. Käyttöjärjestelmässä tulee pyrkiä käyttämään täydellisiä polkunimiä kirjastoille huijausten vähentämiseksi. Mikäli käyttöjärjestelmä suojaa jotakin kohdetta, niin asiantuntijoiden täytyy varmistaa, että käyttöoikeudet tekevät, mitä niiden oletetaan tekevän. Käyttäjätietokannan (*engl. Active Directory*) ja hakemistopalvelujen käyttöön tarkoitetun verkkoprotokollan (*engl. Lightweight Directory Access Protocol*) käyttäminen on kannattavaa verkotetuissa järjestelmissä yhdessä luottamustoimialueessa, joita käytetään käyttöoikeuksien tarkistamiseen. Mikäli järjestelmässä on useita luottamustoimialueita, niin siinä tapauksessa kannattaa käyttää julkisen avaimen infrastruktuuria (*engl. Public Key Infrastructure*). (Shostack 2014, 14, 147.)

PKI-järjestelmään kuuluvat varmenteen myöntäjä, rekisteröinnin myöntäjä, varmenteiden peruutusluettelo, sertifiointikäytäntö ja digitaalinen sertifikaatti. Varmenteen myöntäjä (*engl. Certification Authority, CA*) on vastuutaho digitaalisten varmenteiden myöntämiseen. Rekisteröinnin myöntäjä (*engl. Registration Authority, RA*) vahvistaa digitaalisten allekirjoitusten pyynnöt ja suosittelee CA:n myöntämään sertifikaatteja. Varmenteiden peruutusluettelo (*engl. Cerficate Revocation List, CRL*) sisältää CA:n peruuttamat digitaaliset varmenteet. Peruutettuihin varmenteisiin ei tulisi enää luottaa. Sertifiointikäytäntö (*engl. Certification Practice Statement, CPS*) määrittelee käytännöt ja prosessit digitaalisista varmenteista. Digitaalinen sertifikaatti on sähköinen asiakirja ja se varmis-

taa julkisen avaimen omistajuuden. Digitaalinen sertifikaatti sisältää tiedot avaimesta ja tiedot omistajasta. (Doshi 2023, 359.) Digitaalisen sertifikaatin ideana on sitoa identiteetti julkiseen avaimeen, jota voidaan hyödyntää salausalgoritmeissa turvallisuuden parantamiseksi (Baumeister 2011).

TLS (*engl. Transport Layer Security*) tarjoaa julkiseen avaimeen perustuvan todennuksen ja suojatun istuntoavaimen luomisen. Monet sovellukset ovat riippuvaisia TLS-protokollan turvallisuudesta, ja hyökkääjät ovat käyttäneet MITM-hyökkäystä TLS-protokollaa vastaan. Lal Damas ja Samdaria ehdottavatkin soft-token-pohjaista lähestymistapaa käyttäjien todentamisen varmistamiseksi. Kyseinen lisäominaisuus tehostaa TLS-protokollan turvallisuutta. (Lal Das & Samdaria 2014.) Toisin sanoen TLS-protokolla käyttää kahta eristettyä protokollaa viestintäkanavien suojaamiseksi; ne tarjoavat kaksi suojauskerrosta, muun muassa todennuksen ja salauksen. (Satapathy & Livingston 2016.)

Salausteknologioiden käyttäminen on myös yksi hyvä tapa puuttua huijausuhkiin, koska tässä tapauksessa avain sidotaan tiettyyn henkilöön, joka muodostaa yhteyden tai todennuksen (Shostack 2014, 147). ACL-luetteloita (*engl. Access Control List*) käytetään muun muassa verkkoliikenteen tai -pakettien sallimiseksi tai hylkäämiseksi. ACL on verkkoliikenteen suodatustoiminto, joka suodattaa verkkoliikennettä ja -paketteja, ja sen avulla kyetään suojaamaan tietoverkkoa. ACL on liikenteen/pakettien suodatuskehys, jota voidaan käyttää staattisen tai dynaamisen liikenteen/pakettien käsittelyyn. Staattinen ACL voi tunnistaa vain ennalta määritetyt hyökkäykset käyttämällä ACL-käytäntöjä. (Jayaprakash & Seethalakshmi 2021.) Eheyden varmistamisen voi tehdä muun muassa hash-toiminnoilla ja digitaalisilla allekirjoituksilla (Shostack 2014, 149). Hash-toiminnolla datalle lasketaan tiivistearvo eli hash. Hash-toiminto voi tuottaa myös tekstidatasta kiinteän pituisen salatun tekstin. (Macharia 2021.) Digitaalinen allekirjoitus on salausprimitiivi, jonka avulla varmistetaan, että allekirjoituksen tehnyt henkilö on lähettänyt viestin ja viestin tiedot eivät ole muuttuneet. Digitaalisen allekirjoituksen avulla kyetään estämään tietojen muuttaminen ja viestin lähettäminen toisen henkilön nimellä. (Lizama-Pérez 2022, 3; Huang, Chen & Qu 2009.)

Tietokoneen ja järjestelmän huijaamisessa on tarpeellista käyttää salausmenetelmiä, kuten HTTPS, IPsec ja DNSSEC. Käyttämällä edellä mainittuja salausmenetelmiä varmistutaan siitä, että tietokone muodostaa yhteyden oikeaan paikkaan. DNSSEC:llä ja SSH-tunneloinnilla on mahdollista parantaa todennusta. Osa verkkopalveluiden tarjoajista suodattaa lähtevän tietoliikenteen, koska tämä tekee huijausuhkista vaikeampia toteuttaa. Toimenpide ei ole kuitenkaan aukoton, eikä siihen voi täysin luottaa. (Shostack 2014, 14, 147.) SSH-protokolla mahdollistaa isäntätodennuksen, joka käyttää isäntiin liitettyjä julkisia avaimia käyttäjien todentamiseen. Julkisia avaimia käyttämällä pyritään estämään MITM-hyökkäykset, jotka voivat aiheuttaa muun muassa salasanojen varastamisen tai komentojen lisäämisen todennuksen jälkeen. (Ylonen 2019.) SSH-avaimet eivät ole sidottuja yhteen käyttäjään, ja useat käyttäjät voivat jakaa yksityisen avaimen palvelimelle. Tämä tarkoittaa sitä, että vaarantunutta avainta voidaan käyttää esimerkiksi palvelimiin tunkeutumisessa. (Haber & Hibbert

2018, 38.) SSH-protokolla on eräs yleisimmin käytetyistä tavoista muodostaa etäyhteys palvelimeen. SSH mahdollistaa yksityisyyden ja luottamuksellisuuden salaamalla verkkoliikenteen palvelimen ja asiakkaan välillä. Huomioitavaa on se, että SSH-palvelu voi olla hyökkääjien kohde. Huonosti konfiguroidut palvelimet voivat joutua raa'an voiman salasanahyökkäysten kohteiksi. Hyökkääjät voivat myös käyttää SSH:ta toisen vaiheen hyökkäyshyötykuormien siirtämiseen tai vaikuttaa komento- ja ohjauspalvelimiin. (Sentanoe & Reiser 2022.)

DNSSEC (*engl. Domain Name System Security Extensions*) laajentaa DNS:ää käyttämällä julkisen avaimen tekniikkaa, joka tarjoaa digitaalisen allekirjoituksen DNS-tietueille. DNSSEC:n käyttäminen parantaa turvallisuutta, mutta kustannuksia ja hyötyjä tulee tarkastella tapauskohtaisesti. (Lian, Rescorla, Shacham & Savage 2013.) DNS-tietueet ovat digitaalisesti allekirjoitettuja, ja niiden avulla luodaan RRSIG-tietue hyökkäyksiltä suojautumiseen (Neil 2020, 202). IPsec (*engl. Internet Protocol Security*) on yksi vanhimmista VPN-protokollista, mutta se on edelleen yleisin käytetty VPN-protokolla. IPsecin tarkoituksena on lisätä todennusta ja salausta verkkoliikenteeseen. (Hauser, Häberle & Menth 2020.) IPSecin avulla luodaan suojattu istunto tietokoneen ja palvelimen välillä. Toimenpiteellä estetään, ettei hyökkääjä voi varastaa verkkopaketteja istunnon ajalta tai palvelimelta. (Neil 2020, 188.)

Henkilön huijaamisen välttämisessä on varmistuttava, että jokaisella on identifioitu käyttäjätunnus ja riittävä todennustapa. Tyypillisimmin tämä tehdään salasanojen avulla ja hyödyntämällä monimenetelmätunnistautumista. Lisäksi koodiin kohdistuvat huijausuhkat ovat mahdollisia ja ne esiintyvät useissa muodoissa, esimerkiksi levyllä ohjelman väärentäminen, portin käyttöön ottaminen, yhdistäminen ja etäkoneen huijaaminen. Eräs suuri ja monimutkainen teema on ihmisten huijaaminen, jonka vähentäminen vaatii kontrolloituja integrointeja järjestelmäkerrosten välillä. Esimerkiksi järjestelmän ulkopuolinen arkkitehti tarvitsee pääsyn tietokantaan, mutta ikuisuuskysymys on se, miten varmistetaan henkilön luotettavuus. (Shostack 2014, 14, 146.) Käyttäjien todennus on ensiarvoisen tärkeää digitalisaation aikakaudella. MFA (*engl. Multi Factor Authentication*) tarjoaa laajemman todennuksen verrattuna kaksivaiheiseen tunnistautumiseen. MFA tarjoaa käyttäjille riittävän turvallisuuden ja todentamisen arkaluonteisiin tietoihin pääsemiseksi. (Ometov ym. 2018, 18–19.) Vuonna 2020 Microsoft havaitsi, että 99,9 prosenttia sen seuraamista vaarantuneista tileistä ei käyttänyt MFA:ta (Nahari 2021).

Tampering

TAULUKKO 12 Peukalointiuhkien lieventämisstrategia ja -tekniikka

Uhka	**Strategia**	**Tekniikka**
Tiedoston peukalointi	Käyttöjärjestelmä	• ACL • Tiedostojen eheyden valvonta • EDR
	Salausteknologia	• Digitaaliset allekirjoitukset • Hash-toiminnot
	Analyysityökalut	• UEBA

		• EDR
Verkkopaketin peukalointi	Salausteknologia	• HTTPS • IPSec • SSH • Digitaaliset allekirjoitukset • IPS/IDS

(Ks. Shostack 2014, 15, 150.)

Karkeasti määriteltynä peukalointiuhkien torjumiseksi on kolme pääkeinoa eli käyttöjärjestelmän suojauksiin turvautuminen, salausteknologioiden käyttäminen ja lokiteknologian sekä auditointitoimintojen käyttäminen pelotteena. Toimintojen suojaaminen käyttöoikeuksien avulla voi suojata tiedostoja, tietokannan tietoja tai verkkopalvelimen polkuja, jos elementit ovat käyttömekanismien alla. (Shostack 2014, 149.) Neil (2020, 57) ehdottaa eheyden varmistamiseksi, että tallennetut tiedot hajautetaan tiedostopalvelimella, koska tämän avulla voidaan todistaa, onko tietoja muutettu. Huomioitavaa on se, että eheyden varmistamiseen vaaditaan tiivistearvon eli hashin laskeminen. Digitaalisten allekirjoitusten käyttäminen sähköpostipalveluissa on helppo keino varmistaa, ettei sähköpostia ole muutettu viestin kuljettamisen aikana.

Tiedostojen peukalointi on hyökkääjälle suhteellisen helppoa, mikäli hyökkääjällä on pääsy- tai käyttäjätili uhrikoneessa (Shostack 2014, 15). Tyypillinen hyökkäyskohde on tiedostojärjestelmä, koska ne sisältävät arkaluonteista tietoa, esimerkiksi valtuutustietoja. Haitallisen toiminnan havaitsemiseksi tiedostojen eheyden valvonta on tarpeellinen toimenpide, jonka avulla kyetään havaitsemaan tiedostojen muokkaustoimenpiteet. Hyökkääjien tavoitteena on murtautua järjestelmään ja piilottaa jälkensä muokkaamalla kriittisiä tiedostoja, muun muassa järjestelmälokeja ja suoritettavia lokeja. FIM (*engl. File Integrity Monitoring*) on eräs suosittu tapa tarkkailla haitallista käyttäytymistä, esimerkiksi lokien muokkaamista, troijalaisten lisäämistä ja takaovien liittämistä. (Jin, Xiang, Zou, Zhao, Li & Yu 2010.) FIM-työkalujen avulla tunnistetaan, mitä tiedostoja tai hakemistoja on vahingoitettu tai muokattu. Työkalujen avulla kyetään havaitsemaan, kuka käyttäjä on tehnyt muokkauksia ja mihin kellonaikaan. (Zlatkovski, Mileva, Bogatinova & Ampov 2018.) UEBA:n käyttäminen FIM-ratkaisujen tukena mahdollistaa turvallisuuden parantamisen (Martín, Beltrán, Fernández-Isabel & Martiín de Diego 2021). UEBA (*engl. User and Entity Behavior Analytics*) käyttää analytiikkateknologiaa, muun muassa koneoppimista tunnistamaan järjestelmän käyttäjien poikkeavaa käyttäytymistä. UEBA luo uuden profiilin, jonka avulla se seuraa käyttäjien normaalia käyttäytymistä. Analyysi perustuu siihen, että käyttäjien poiketessa normaalista käyttäytymisestään UEBA tunnistaa sen. Hyökkääjien on haastavaa matkia käyttäjien normaalia käyttäytymistä, ja tämän takia UEBA on tehokas keino kehittyneiden uhkatoimijoiden (*engl. Advanced Persistent Threat, APT*) torjuntaan. APT-hyökkäysten suunnittelu voi kestää hyvin pitkän aikaa, ja hyökkääjän tavoitteena on päästä järjestelmään sisälle ja viettää järjestelmän sisällä mahdollisimman kauan. UEBA:n avulla tunnistetaan tämänkaltainen epänormaali toiminta. (Diogenes & Ozkaya 2022, 78–79.)

Haittaohjelmien havaitsemiseksi ja estämiseksi on olemassa erilaisia tekniikoita, joita ohjelmisto- ja laitteistotoimittajat ovat ottaneet käyttöön käyttöjärjestelmissä, esimerkiksi tietojen suoritussuojaus (*engl. Data Execution Protection, DEP*), osoiteavaruuden asettelun satunnaistaminen (*engl. Address Space Layout Randomization, ASLR*), strukturoidun poikkeuskäsittelijän päällekirjoitussuojaus (*engl. Structured Exception Handler Overwrites Protection, SEHOP*) ja pakollinen eheyden valvonta (*engl. Mandatory Integrity Control, MIC*) (Marpaung, Sain & Lee 2012, 744). EDR (*engl. Endpoint Detection and Response*) on kehittyneempi ratkaisu virustorjuntaan tai perinteisiin haittaohjelmasuojauksiin verrattuna, koska EDR valvoo jatkuvasti tietokonetta ja se hälyttää automaattisesti, kun havaitsee uhkan. EDR käyttää koneoppimista uhkien havaitsemiseen, ja se kykenee havaitsemaan myös tiedostottomat virukset. (Neil 2020, 337.) EDR on oivallinen ratkaisu muun muassa kehittyneiden uhkatoimijoiden torjuntaan, koska nykyajan järjestelmämurrot tapahtuvat erittäin huomaamattomasti. Kehittyneet uhkatoimijat aiheuttavat merkittäviä turvallisuusuhkia muun muassa hallituksille ja suurille yrityksille, ja EDR:stä on muodostunut vakioratkaisu kehittyneiden uhkatoimijoiden torjuntaan. (Chen ym. 2023.) EDR:n heikkoutena on se, että se tuottaa huomattavan paljon vääriä hälytyksiä, ja uhkahälytysten käsittely vaatii manuaalista työtä (Hassan, Bates & Marino 2020).

Verkkopakettien peukalointiuhkien estämiseksi vaaditaan huijaus- ja peukalointiuhkien torjuntaa. MITM-hyökkäyksissä hyökkääjä voi huijata olevansa luotettu käyttäjä ja hyökkääjällä on pääsy datan peukaloimiseksi. Yleisimmin käytetty torjuntamekanismi on IPSec. (Shostack 2014, 15.) MITM-hyökkäyksiä voidaan lieventää myös käyttämällä suojattuja istuntotunnisteita, aikaleimoja ja suojattua DNS:ää (Auger, Scott, Helmus & Nguyen 2021, 16). IDS (*engl. Intrusion Detection Systems*) ja IPS (*engl. Intrusion Prevention Systems*) tutkivat haitallista liikennettä, mutta toiminnaltaan ne ovat erilaisia. IDS hälyttää, jos se havaitsee normaalista poikkeavaa verkkoliikennettä. Toisin sanoen IDS tarkkailee verkkoliikennettä muun muassa analysoiden verkkosegmentille tarkoitetut paketit hyökkäyksen havaitsemiseksi. IPS havaitsee ja estää manuaalisilla ja automaattisilla toiminnoilla haitallisia toimenpiteitä. Hyökkääjä voi yrittää hyökätä TCP-portti 80:een, joka mahdollistaa pääsyn www-palvelimeen. IDS ilmoittaa mahdollisesta hyökkäyksestä, ja IPS aloittaa vastatoimet, esimerkiksi katkaisemalla tämän yhteyden. (Ashoor & Gore 2011, 1.) Hajautetut palvelunestohyökkäykset ja edistyneet hyökkäykset aiheuttavat nykyajan toimintaympäristössä varsin suuren uhkan. Verkkotunkeutumisen uhkasta on tullut eräs merkittävä ongelma, jonka ratkaisemiseksi tutkijat työskentelevät jatkuvasti suunnitellakseen turvallisen ja tehokkaan verkkoon tunkeutumisen havaitsemisjärjestelmän. (Liu, Gao & Hu 2021, 1.) Tunkeutumisen havaitsemisjärjestelmät (IDS) ovat ensisijaisia työkaluja organisaatioiden tietoverkoissa tapahtuvien hyökkäysten havaitsemiseen (Bajtoš, Sokol & Kurimský 2024).

Repudiation

TAULUKKO 13 Kiistämisuhkien lieventämisstrategia ja -tekniikka

Uhka	Strategia	Tekniikka
Lokien puuttuminen	Lokien käyttöönotto	• Tietoturvaan liittyvien toimintojen kirjaaminen • Lokien käyttäminen • Digitaaliset allekirjoitukset • Laadukkaasti laaditut lokit • Tiedonhallintalain noudattaminen
Lokit hyökkäyksen kohteena	Lokien suojaaminen	• ACL • Suojatut lokivarannot • Luotettavat kolmannet osapuolet

(Ks. Shostack 2014, 14, 153.)

Kiistämisuhkat edellyttävät, että järjestelmässä on lokitiedot käytössä ja niitä säilytetään ja suojataan asianmukaisesti (Shostack 2014, 16). Lokien käyttäminen perustuu lakiin, joka vaatii viranomaisen käyttämään lokeja tietojärjestelmässään (Laki julkisen hallinnon tiedonhallintalaista 906/2019, 17 §). Eräs tietoturvaongelma on SQL-injektiohyökkäykset ja useimmat olemassa olevat ratkaisut käyttävät suojautumistoimenpiteinä lokianalyysiä ja koneoppimismenetelmiä. Monivaiheisen lokianalyysiarkkitehtuurin käyttäminen, jossa yhdistyvät hahmontunnistus ja koneoppimismenetelmät, havaitsee SQL-injektiohyökkäykset tehokkaammin. Menetelmä käyttää hyökkäyksen aikana luotuja lokeja tunnistamaan hyökkäykset, ja se auttaa estämään tulevat hyökkäykset. (Moh, Pininti, Doddapaneni & Moh 2016.)

Jos järjestelmässä ei ole lokitietoja käytössä, niin siinä tapauksessa ei voida todistaa mitään. Lokien käyttäminen ja ylläpitäminen on ensiarvoisen tärkeää, jotta voidaan tutkia, mitä on tapahtunut kiistämisuhkien ilmetessä. Lokit ovat tyypillinen tekniikka kiistämisongelmien ratkaisuun. Lokien määrä riippuu tapahtumista, mutta ne sisältävät yleensä allekirjoitukset tai IP-osoitteen. (Shostack 2014, 16, 151.) Lokitiedot ovat oleellinen osa tietojärjestelmän suojauksen diagnosoinnissa ja mahdollisten ongelmien havaitsemisessa. Ongelmat voivat olla monimutkaisia, mutta laadukkaasti laaditut lokitiedot mahdollistavat ongelmatilanteiden ratkaisemisen huomattavasti nopeammin. (Azizi, Azizi & Elboukhari 2019.) Laadukkaasti laaditut lokit rakentuvat seuraavista tekijöistä: aikaleima eli tapahtuma-ajan kirjaaminen, tapahtuma ja toimija eli toimenpiteiden kirjaaminen ja kuka teki, käyttöoikeus eli millä oikeuksilla toimenpide tehtiin, lähde eli missä toimenpide tehtiin, toimenpiteen tila eli onnistuiko vai epäonnistuiko toimenpide (Kyberturvallisuuskeskus 2023a).

Esimerkkinä Windows-käyttöjärjestelmän lokityypeistä ovat turvallisuuslokit, jotka sisältävät tietoa järjestelmän turvallisuuteen liittyvistä kokonaisuuksista. Sovelluslokit sisältävät tietoa toiminnoista, joita sovellukset tekevät. Järjestelmälokeja hyödynnetään muun muassa virheiden määritykseen. Nämä kolme lokityyppiä sisältävät suuren toimintovalikoiman, ja ne ovat hyödyllisiä määriteltäessä, mitä toimintoja on tapahtunut järjestelmässä. (Johansen 2022, 294.) Azahari ja Balzarotti (2024) havaitsivat merkittävän tekijän sovelluslokien

hyödyntämisestä. Sovelluslokit voivat parantaa merkittävästi tietoturva-analyysiä, koska ne tarjoavat tietoja esimerkiksi käyttäjien toiminnasta. Tutkijat tarkastelivat kuusikymmentä avoimen lähdekoodin sovelluksen kirjaustoteutusta. He havaitsivat, että monista lokeista puuttui oleellisia elementtejä. Kaksikymmentäyhdeksän sovellusta jätti aikaleimat kirjaamatta, ja kaksikymmentäkolme sovellusta ei tunnistanut uusien toimintojen yksilöllisiä tunnisteita (UID). Tutkijat tulivat siihen lopputulokseen, että nykyiset sovelluslokit eivät yksin ole riittäviä perusteellista tietoturva-analyysiä varten.

Hyökkääjä saattaa hyökätä lokitietoihin, ja tavoitteenaan on täyttää lokit hyökkäyksen havaitsemisen estämiseksi. Eräs mahdollinen skenaario on se, että hyökkääjä tekee vääriä hälytyksiä ja todellinen hyökkäyskohde hukkuu väärien hälytysten joukkoon. Huomioitavaa on myös se, että lokien lähettäminen verkon kautta altistaa muille uhkille, joihin täytyy varautua. (Shostack 2014, 17.) Lokien turvallisuuden varmistamiseksi tärkeät lokitiedot ohjataan keskitetylle ja suojatulle lokipalvelimelle. Oleellista on varmuuskopioida tiedot päivittäin turvallisuusluokan vaatimaan ympäristöön, ja lokitietojen säilytysajat tulee määritellä tapauskohtaisesti. (Katakri 2020, 85.) Mikäli hyökkääjä on päässyt käsiksi valvonta- ja suojauslokeihin, niin hyökkääjä kykenee muuttamaan lokeja ja tämän avulla piilottamaan jälkensä (Wiesner 2023, 200). Tämän takia lokien tarkastelu ja käsittely on tarpeellista lokittaa (Kyberturvallisuuskeskus 2016b, 4).

Information Disclosure

TAULUKKO 14 Tietojen paljastamisuhkien lieventämisstrategia ja -tekniikka

Uhka	**Strategia**	**Tekniikka**
Verkonvalvonta	Salaus	• HTTPS • IPSec • Hyväksytty avainten hallinta
Hakemisto tai tiedostonimet	Käyttöjärjestelmän hyödyntäminen	• ACL
Viestinnän sisältö	Salaus	• Sipulireititys • Steganografia
Tiedoston sisältö	Käyttöjärjestelmän hyödyntäminen	• ACL • RBAC-ratkaisut • PAM
	Salausteknologia	• Tiedostosalaimet, esimerkiksi Bitlocker
Ohjelmointirajapinnan tietojen paljastaminen	Suunnittelu	• Asianmukaisesti toteutettu suunnittelu ja sen valvonta

(Ks. Shostack 2014, 17, 155.)

Verkonvalvonnassa hyödynnetään useiden verkkojen arkkitehtuuria tietoliikenteen valvomiseksi. Huijaus- ja peukalointiuhkat ovat oleellisia tekijöitä ottaa huomioon käsiteltäessä tietojen paljastamisuhkia. Mikäli näitä uhkia ei oteta huomioon, hyökkääjä kykenee vaikuttamaan lähettäjään ja vastaanottajaan tietoliikenteen näkökulmasta tarkasteltuna. Mikäli käyttöjärjestelmä hallitsee tietoihin pääsyä, voidaan sen pääsyoikeuslistoja käyttää, mutta muussa tapauk-

sessa data on salattava. (Shostack 2014, 18, 154.) Avainhallintapolitiikan on oltava hyvin suunniteltua ja se koskee jokaista organisaatiota. Oleellista on määritellä yksityisten ja salaisten avainten turvallinen jakelu, määritellä avainten elinkaaret, korvata avaimet elinjakson lopussa, tehdä toimenpiteet avainten hätäperuuttamiseen, tarkistaa tietojen eheys elinjakson päättyessä ja luoda asianmukaiset menettelytavat avainmateriaalin tuhoamiseksi. (Barker & Barker 2018, 17–18.)

Tiedostojen- ja hakemistojen nimet paljastavat itsessään tietoja, minkä takia on syytä harkita päähakemiston luomista. Päähakemistolle määritellään neutraali nimi ja hyödynnetään käyttöjärjestelmän ACL-luetteloita ja pääsyoikeuksia. Jos tiedosto sisältää arkaluonteista tietoa, niin siinä tapauksessa tulee käyttää salausmekanismeja tai ACL-luetteloita. Tietojen paljastamisen estämiseksi on yleisesti ottaen kaksi päätapaa, jotka ovat ACL-luetteloiden käyttäminen järjestelmän rajoissa ja rajojen ulkopuolella salauksen käyttäminen. Viestinnän sisällön suojaamiseksi tavallinen salaus riittää, mutta jos halutaan salata viestinnän osapuolet, niin tähän vaaditaan esimerkiksi sipuliverkon käyttämistä. Steganografiaa vaaditaan, jos halutaan salata viestinnän olemassaolo. (Shostack 2014, 18, 154.) Steganorafian avulla on mahdollista piilottaa tiedon olemassaolo, ja sille on kolme perusvaatimusta: salaaminen, kestävyys, hyvä ja asianmukainen steganografiakyky. (Zhang, Cao, Jahanshahi & Mou 2023.) Sipulireititys protokolla on laajasti käytetty tekniikka, jossa piilotetaan viestinnän osapuolet. Protokollan käyttäminen tarjoaa käyttäjälle salaustoimenpiteet, joiden avulla salakuuntelu- ja tietoliikenteenanalyysihyökkäyksiä on vaikea toteuttaa. (AlQahtani & El-Alfy 2015, 122.)

Roolipohjainen pääsynvalvonta (*engl. Role Based Access Control, RBAC*) on verrattain vanha toimintamalli, joka yleisesti otettiin käyttöön 1990-luvulla. Sen tarkoituksena on käyttöoikeuksien liittäminen rooleihin, missä perusajatus on yksinkertaistaa tietoturvan hallintaa. (Orchilles 2010.) Organisaatioissa tiedot ovat yleensä arvokkain omaisuus, minkä takia organisaatioiden täytyy varmistaa, että vain luvalliset käyttäjät pääsevät tietoihin käsiksi. ACL-luettelot auttavat rajoittamaan, mihin käyttäjällä on pääsy. Huomioitavaa on se, että kun käyttäjä suorittaa prosessin, niin prosessilla on sama pääsy tietoihin kuin käyttäjällä. Mikäli käyttäjä käyttää haitallista ohjelmistoa, niin tietoja voidaan poistaa tai siirtää organisaation ulkopuolelle. Applocker mahdollistaa tämän tyyppisen tietoturvaongelman lieventämistä, koska Applockerin avulla voidaan rajoittaa tiedostoja, joita käyttäjä voi suorittaa. (Microsoft 2024.) Applocker on tarkoitettu asennettujen sovellusten hallintaan, mutta sitä ei saa sekoittaa Bitlockeriin, joka on tarkoitettu työaseman salauksen hallintaan (Orchilles 2010). SELinux tarjoaa pakollisen pääsynhallintapolitiikan sovelluksille, prosesseille ja tiedostoille (Wonga, Chekolea, Ochoab & Zhoua 2023). Pääsynhallintapolitiikan ansiosta SELinux on osoittautunut tehokkaaksi ratkaisuksi useita oikeuksien korottamisuhkia vastaan (Radhika, Kumar, Shyamasundar & Vyas 2020).

Purban ja Soetomon (2018) mukaan ISO 27001 on eräs laajimmin hyväksytyistä ja tunnetuimmista tietoturvastandardeista. Standardi käsittelee lähes jokaisen tietoturvan näkökulman, mukaan lukien PAM-ratkaisut (engl. Privileged

Access Management), joka on yksi mahdollinen tapa osoittaa pääkäyttäjätunnusten huolellista hallintaa ISO 27001 sertifiointi- ja auditointiprosessissa. Etuoikeutettujen tilien hallitsemiseksi organisaatio voi käyttää PAMia kriittisten IT-resurssien suojaamiseksi, vaatimustenmukaisuussäännösten noudattamiseksi ja tietomurtojen estämiseksi. PAM mahdollistaa automaattisen salasanan ja istunnon hallintaratkaisun, joka tarjoaa suojatun pääsynhallinnan, tarkastuksen ja tallennuksen jokaiselle etuoikeutetulle tilille (Haber & Hibbert 2018, 107).

Ohjelmointirajapinnan (*engl. Application Programming Interface, API*) suunnittelussa on oleellista tunnistaa, mitä tietoja paljastetaan, kun tietoja siirretään luottamusrajan ylitse. Oletuksena on syytä käyttää sitä, että tiedot välitetään muille käyttäjille, minkä takia täytyy olla huolellinen siitä, mitä tietoja paljastetaan. Yleinen puute on se, että verkkosivustojen virheet voivat paljastaa käyttäjätunnuksen ja salasanan tietokantaan. (Shostack 2014, 18.) Tietojen paljastamisuhka esiintyy tapauksessa, jossa API ja sitä tukeva ohjelmisto jakaa arkaluonteisia tietoja oikeutetuille käyttäjille. Tietoja voidaan paljastaa API-vastauksista tai julkisista lähteistä. Esimerkiksi WordPress-sovellusliittymää käyttävä sivusto voi jakaa käyttäjätietoja kenelle tahansa, joka siirtyy API-polun sivustoon https://<URL-osoite>/wp/-json/wp/v2/users. (Ball 2022, 54–55.) Ohjelmistovirhe johtuu yleensä suunnittelu- tai ohjelmointivirheestä. Mahdollinen virhe voi olla syötteen vahvistusvirhe esimerkiksi käyttäjän syötteen luotettavuutta ei ole arvioitu oikein haitallisten merkkijonojen varalta, jotka voivat mahdollistaa hyökkäyksen toteuttamisen. (Souppaya & Scarfone 2016, 3.) Huonosti suunnitellut APIt ovat usein kaikkien palvelunestohyökkäysten kohteita esimerkiksi skenaariossa, jossa API ei estä haitallisia pyyntöjä (Sharieh & Ferworn 2021, 291).

Luottamuksellisuuden korostaminen on ensiarvoisen tärkeää yrityksen liiketoiminnan kannalta. Tietojen paljastamisen kustannuksia ei voida täysin määritellä taloudellisten tappioiden perusteella. Aineettomat menetykset, muun muassa maineen menettäminen, voivat olla yritystoiminnan jatkuvuuden kannalta kriittinen tekijä. Nykyajan hybridi- ja/tai etätyömaailmassa työntekijöiden tulisi kiinnittää huomiota kotiverkon turvallisuuteen. L2TP / IPSec VPN-tunnelin ja AES (*engl. Advanced Encryption Standard*) -salausmenetelmän käyttäminen on yksi tapa tietojen paljastamisen ehkäisemiseksi myös kotiverkossa. (Neil 2020, 57.) AES sisältää symmetrisen algoritmin, joka voi käyttää samaa avainta viestien salaukseen ja salauksen purkamiseen. AES on todettu hyväksi viestien ja tiedostojen salausmenetelmäksi toimintanopeuden, avainturvallisuuden ja yksinkertaisen rakenteen osalta. Heikkoutena AES:lla on muistin varauksen käyttö, koska sitä käytetään paljon. (Kaffah ym. 2020.)

Denial of Service

TAULUKKO 15 Palvelunestouhkien lieventämisstrategia ja -tekniikka

Uhka	**Strategia**	**Tekniikka**
Tietoverkon tulva	Kuormittuneiden resurssien etsiminen	• Joustavat resurssit ja varmistetaan, että hyökkääjän resurssien kuluttaminen on yhtä suuri tai suurempi

		kuin suojattavan kohteen • Tietoverkkojen ACL • Suodattimet • Kaistan rajoittaminen • Ylimääräinen kaistanleveys • Pilvipalvelut
Ohjelman resurssit	Huolellinen suunnittelu	• Joustava resurssienhallinta
Järjestelmän resurssit	Käyttöjärjestelmän hyödyntäminen	• Käyttöjärjestelmän asetusten käyttäminen

(Ks. Shostack 2014, 19, 157.)

Alkuvuonna 2024 kymmeniä organisaatioita listattiin palvelunestohyökkäysten kohteiksi, ja niistä uusia palvelunestohyökkäyksen kohteita oli kunta- ja koulutussektorilta. Vuonna 2023 finanssi-, logistiikka- ja liikenne- ja valtionhallinnon toimijoihin kohdistui palvelunestohyökkäyksiä. (Traficom 2024.) DDoS-hyökkäyksissä tavoitelluin puolustustekniikka on ennaltaehkäisy, koska DDoS-hyökkäykset aiheuttavat suuren vaaratekijän järjestelmän resursseille, infrastruktuurille ja tietoverkon kaistanleveydelle. Aloitettu ja onnistunut hyökkäys vaarantaa uhrin järjestelmän, minkä takia ennaltaehkäisy DDoS-hyökkäyksiä vastaan on paras suojautumistekniikka. (Mahjabin, Xiao, Sun & Jiang 2017, 13.)

Tietoverkon tulvassa on ymmärrettävä staattisten rakenteiden lukumäärä ja tapahtumaketju, jos nämä ruuhkautuvat. Väärennetystä osoitteesta ei saa hyväksyä verkkodataa eikä palauttaa dataa. Palomuurit tarjoavat suhteellisen hyvän kerroksen tietoliikenteen vastaanottamiseen ja lähettämiseen. Yleisesti ottaen palomuurit ovat hyödyllisiä palvelunestohyökkäysten lieventämisessä. (Shostack 2014, 19.) Suodatustekniikat ovat hyvä suojautumiskeino, ja ne estävät huomattavan paljon DDoS-hyökkäyksiä, koska niiden avulla varmistetaan, ettei haitallista liikennettä pääse järjestelmään. Yleinen suodatustekniikka on sisään- ja ulostulosuodatus, jonka avulla estetään väärennetyillä IP-osoitteilla olevaa tietoliikennettä pääsemästä suojattavaan järjestelmään. Sisääntulosuodatus suodattaa haitallisen liikenteen, ja ulostulosuodatus hylkää kohdejärjestelmästä lähtevän haitallisen liikenteen. (Mahjabin ym. 2017, 13.)

Pilvipalvelut tarjoavat hyvän joustavuuden, skaalautuvuuden ja saatavuuden useille sovelluksille ja palveluille. Pilvipalveluita käyttämällä voidaan tasata resursseja, ja tämän avulla vähennetään kuormittavuutta. Huomioitavaa on kuitenkin se, että pilvipalvelut ovat myös alttiita palvelunestohyökkäyksille, ja tämän takia koneoppimisalgoritmit ovat nousseet lupaaviksi lieventämismenetelmiksi palvelunestohyökkäysten torjumisessa. Koneoppimisalgoritmit kykenevät käsittelemään suuren määrän verkkoliikennettä ja oppimaan hyökkäyksistä, minkä avulla ne kykenevät tunnistamaan uusia hyökkäyksiä. (Polu & Bapuji 2024, 341.) Khadke ja Madankar (2016) ovat myös huomanneet, että pilvipalveluiden tarjoamat virtuaaliset palvelimet tarjoavat ohjelmistoja, infrastruktuuria ja useita muita resursseja. Pilvipalveluiden käyttämisen etuna on laaja käytettävyys ja kustannussäästöt.

Resurssien suunnittelussa tulee huomioida se, mitä ohjelma hallinnoi itsenäisesti. Huomioitavaa on tunnistaa hyökkääjän tekemä toimenpide ja huomata, kuluttaako tämä enemmän puolustajan kapasiteettia kuin hyökkääjän. Esimerkiksi hyökkääjä lähettää kohdejärjestelmään paketin ja puolustaja tekee salaustoimintoja verrattuna siihen, että salaustoimenpiteet tekee ensin hyökkääjä. (Shostack 2014, 20.) Hunajaverkot voivat estää palvelunestohyökkäyksiltä, koska hunajaverkkojen avulla huijataan hyökkääjää hyökkäämään niihin suojeltavan verkon sijasta. Toiminta mahdollistaa lisätietojen saamisen hyökkääjästä, hyökkäystavasta ja -työkaluista, ja tietoja voidaan hyödyntää hyökkääjän havaitsemiseen ja hyökkäyksen estämiseen. Haittapuolena hunajaverkkojen käyttämiselle on niiden passiivinen ja staattinen luonne, joka ei anna täydellistä suojaa. (Mahjabin ym. 2017, 15.) Resurssien suunnittelussa täytyy myös huomioida se, että ulkoisten palveluiden käytön estymisellä ei ole vaikutusta sisäisten palveluiden käyttämiselle. Vaihtoehtona on muun muassa segmentoida ulkoiset verkkopalvelut tai hyödyntää CDN:ää (*engl. Content Delivery Network*). (Kyberturvallisuuskeskus 2016a, 6.) CDN:n käyttäminen parantaa verkkosivustojen suorituskykyä ja skaalautuvuutta, joiden avulla kyetään jakamaan tietoliikennekuormaa tasaisesti, mikä auttaa palvelunestohyökkäyksen estämisessä (Guo, Li, Liu, Haot & Zhang 2020). Verkkojen segmentoinnissa jaetaan verkko pienempiin osiin, jotka tunnetaan aliverkkoina. Tarkoituksena on parantaa verkkojen suorituskykyä ja turvallisuutta. (Enoka 2023, 26.) Eräs esimerkki segmentoinnista on fyysinen segmentointi. Tämä tarkoittaa sitä, että sallitaan käyttäjien pääsy internetiin, mutta se erotetaan organisaation lähiverkosta ja asetetaan suojattuun aliverkkoon. (Neil 2020, 194.) Yleisesti ottaen segmentointia tarkasteltaessa kuvataan menetelmää, jossa erotetaan tai sisällytetään tietyt laitteet muista laitteista verkossa. Langattoman infrastruktuurin kannalta tärkeimpiä segmentointimenetelmiä ovat ensimmäisen kerroksen fyysinen segmentointi, toisen kerroksen segmentointi reitittämättömien VLAN-verkkojen tai muiden menetelmien kautta, kolmannesta kerroksesta neljänteen kerrokseen segmentointi ACL-luetteloilla, kolmannesta kerroksesta seitsemänteen kerrokseen segmentointi ohjelmiston määrittämillä verkoilla tai virtualisoiduilla verkoilla ja kerroksen seitsemän segmentointi tietoverkkopohjaisella mikrosegmentoinnilla nollaluottamusarkkitehtuureissa. (Minella 2022, 23.)

Käyttöjärjestelmät ohjaavat koodin tarvitsemien resurssien käyttöä. Suunnittelussa täytyy ottaa huomioon se, mitä resursseja käyttöjärjestelmä hallitsee, esimerkiksi muistia tai levyn käyttöä. Järjestelmän käyttöön ottamisessa täytyy tunnistaa resurssit, joita hyökkääjä voi kuluttaa, ja etsiä tapoja kyseisten resurssien rajoittamiseksi tapauskohtaisesti. (Shostack 2014, 20, 156.) Kuormantasaajien käyttäminen on oleellinen osa joustavaa suunnittelua, jonka tarkoituksena on tasapainottaa kuormia palvelinten välillä. Tämän avulla optimoidaan järjestelmän käyttäminen myös häiriötilanteessa. Kerrosten käyttäminen on eräs palvelunestohyökkäysten estokeino, jossa luodaan overlay-verkkoja, ja ne muodostavat yhteyden suojattuun verkkoon. NIDS (*engl. Network Inrusion Detection System*) mahdollistaa verkkoliikenteen analysoimisen, missä se valvoo verk-

koon tulevaa ja sieltä lähtevää tietoliikennettä kaikista verkkoon kytketyistä laitteista. (Songa 2022.)

Elevation of Privilege

TAULUKKO 16 Oikeuksien korottamisuhkien lieventämisstrategia ja -tekniikka

Esimerkki	Strategia	Tekniikka
Datan tai koodin sekaannus	Työkalujen ja arkkitehtuurin hyödyntäminen, jotka erottavat datan koodista	• Tietojen tallennus tietokantaan, selkeä erottelu kanonisille muodoille
Kontrollivirran tai muistin korruptiohyökkäys	Tyyppiturvallisen ohjelmointikielen käyttäminen	• Koodikatselmointi
	Käyttöjärjestelmän hyödyntäminen muistin suojaamiseksi	• ASLR • Hiekkalaatikko

(Ks. Shostack 2014, 20–21.)

Ongelmakohdat ovat niitä, joissa dataa käsitellään koodina. XSS-hyökkäykset hyödyntävät HTML:n koodia ja dataa, koska HMTL voi sisältää JavaScriptiä ja tekstidataa. Tärkeää on tarkastella lähetettävää dataa, ettei se sisällä ylimääräisiä merkkijonoja, muun muassa #-, <-, >-, -tai &-merkkejä. (Shostack 2014, 21.) Kehittäjät saattavat käyttää piilotettuja kenttiä arvojen siirtämiseen, esimerkiksi <input type=”hidden” id=”userLevel” value=”1”>. Piilotetut kentät sisältävät tietoturvariskin, koska kentät näkyvät käyttäjille ja mahdollistavat niiden manipuloimisen. Hyökkääjä kykenee muuttamaan tyhjien kenttien arvoja HTML-lähdekoodissa ja käyttää tätä hyväkseen. Mikäli verkkosovellus käyttää piilotettua kenttää esimerkiksi alennuskupongin lisäämiseen, niin hyökkääjä voi kyetä ostamaan tuotteen jopa ilmaiseksi. Hyökkäyksen onnistumisessa on kyse siitä, että verkkosovellus ei tarkista piilotetun kentän palautusarvon ja lähtevän arvon samankaltaisuutta. (Lia, Xiea, Jinb & Liub 2010, 2264.)

XSS-hyökkäyksen avulla hyökkääjä kykenee lähettämään suoritettavan komentosarjan uhrin selaimeen, ja tämän avulla hyökkääjä kykenee tekemään haitallisia toimintoja, esimerkiksi kaappaamaan käyttäjäistunnon. SQL-injektio voi onnistua XSS-hyökkäyksen tapaan, jos verkkosovellus ei suodata tai muunna käyttäjien syötteisiin sisältyviä SQL-komentoja. Esimerkki ohjelmointivirheestä on

*SQLKysely = ”SELECT * FROM Käyttäjät WHERE (Käyttäjänimi= ´ ”*
+ strKäyttäjänimi + ” ´) AND (Salasana= ´ ”+ strSalasana + ” ´);”;
if SQLKyselyTulos (SQLKysely) == 0
then authenticated = false;
else authenticated = true;

Hyökkääjä voi manipuloida ajettavaksi kyselyksi:

*SELECT * FROM Käyttäjät WHERE (Käyttäjänimi = ´X´OR´A´ = ´A´)*
AND(Salasana = ´X´OR ´A´= ´A´);

Hyökkäyksen avulla hyökkääjä ohittaa todennuksen ja pääsee käsiksi käyttäjättauluun. (Li ym. 2010, 2264–2265.)

Kontrollivirran tai muistin korruptiohyökkäyksissä hyökkääjä käyttää C-tyyppisten kielten staattisia rakenteita, minkä takia mahdollisuuksien mukaan kannattaa käyttää tyyppiturvallisempia ohjelmointikieliä, esimerkiksi Java- tai C#-kieltä. Hyökkäys on huomattavasti vaikeampi toteuttaa, kun puolustaja käyttää edellä mainittuja ohjelmointikieliä. (Shostack 2014, 21–22.) Puskurin ylivuoto on klassinen esimerkki muistinkorruptiohyökkäyksessä; siinä hyökkääjä muuttaa ohjelman muistin tilaa siten, että koneen hallinta siirtyy hyökkääjälle. Onnistuneessa hyökkäyksessä hyökkääjä lähettää ylimääräistä syötettä kohteeseen. Ohjelma, joka ei tarkista, ylittääkö syötteen tulo muistipuskurin kokoa, kopioi ylimääräiset tiedot, ja tiedot vuotavat toiseen puskuriin. (Ruwase & Lam 2004.) Puskurinylivuotohyökkäysten estämisen käytännöllisimpiä keinoja on tehdä perusteellinen koodikatselmointi eli lähdekoodin auditointi ennen ohjelmiston käyttöön ottamista (Kuperman, Brodley, Ozdoganoglu, Vijaykumar & Jalote 2005, 53).

Nykyään käyttöjärjestelmissä on mahdollisuus käyttää muistin suojaus- ja satunnaistoimintoja, esimerkiksi ASLR (*engl. Address Space Layout Randomization*). Ominaisuudet eivät välttämättä ole vakioasetuksia, minkä takia toiminnot täytyy testata toiminnan varmistamiseksi. Toinen vaihtoehto on hiekkalaatikoiden käyttäminen. (Shostack 2014, 22.) ASLR on laajasti käytössä käyttöjärjestelmissä, ja sen tarkoituksena on estää koodin uudelleenkäyttö hyökkäykset. Hyökkääjä voi kuitenkin ohittaa ASLR:n käyttämällä muistin korruptiohaavoittuvuuksia ja suorittaa muistin paljastamishyökkäyksiä. (Jin, Liu, Du & Zou 2018.) Hiekkalaatikoita käytetään muun muassa eristämään testaamaton ja epäluotettava koodi. Toimenpide suojaa kriittisiä resursseja, esimerkiksi palvelimia ja niissä olevaa tietoa muutoksilta. (Bak, Manamcheri, Mitra & Caccamo 2011.) Hiekkalaatikko tarjoaa eristetyn ympäristön, jossa sovellukset toimivat itsenäisesti, mutta se rajoittaa sovellusten pääsyä järjestelmän kriittisiin osiin. Hiekkalaatikko hallitsee sovellusten käyttöoikeuksia ja seuraa sovellusten toimintaa ja käyttäytymistä. (Verma & Nand 2023, 119.)

4.3 Uhkatyypit

Uhkilla tarkoitetaan haitallisia tai muita tapahtumia, jotka vaarantavat järjestelmän turvallisuuden. Uhkilla on haitallisia vaikutuksia, ja ne voivat aiheuttaa järjestelmän kaatumisen, arkaluonteisen tiedon lukemisen tai datan muokkaamisen ja niin edelleen. (Kohnfelder & Garg 1999.) Beyerin (2020, 110) haastattelussa Adam Shostack luonnehtii uhkan tulevaisuuden haitaksi, joka saattaa tapahtua. Tyypillisten uhkien tyypittelyssä hyödynnetään Stallingsin (2019, 90–92) mallia, jossa uhkat on jaoteltu taulukon seitsemäntoista kategorioihin. Lieventämiskeinoina on käytetty pääluvun neljä alaluvusta 4.2 saatuja havaintoja. Tiettyjä lieventämiskeinoja ei ole mainittu teoriaosassa, mutta ne avataan taulukon yhteenveto-osiossa. Uhkia ei ole luokiteltu STRIDE:n mukaisiin kategorioi-

hin, koska uhkien kuvaukset on esitetty geneerisellä tasolla, eikä niitä ole sidottu tiettyyn kontekstiin. Lisäksi STRIDE on työkalu, jonka avulla tarkastellaan uhkia, eikä siinä keskitytä tarkkaan kategorisointiin (Shostack 2014, 12).

TAULUKKO 17 Tyypillisiä uhkia ja niiden lieventämiskeinot

Uhka	**Kuvaus**	**Lieventämiskeinot**
Mato	Toimivat kohdejärjestelmässä itsenäisesti ja tekevät haitallisia toimintoja verkon isännille.	• Ohjelmistopäivitykset • Käyttäjien koulutus • IDS/IPS • Vähimpien oikeuksien periaate • Sähköpostisuodatus • EDR • SOC • Verkkosuodattimet • Sovellusten valkolistaus
Kiristyshaittaohjelma	Estävät käyttäjiä käyttämästä esimerkiksi tietokonettaan.	• Käyttäjien koulutus • Palomuuri • Sähköpostisuodatus • Verkkosuodattimet • IDS/IPS • Vähimpien oikeuksien periaate • EDR • Sovellusten valkolistaus
Looginen pommi *(engl. Logic bomb)*	Hyökkääjä on onnistunut saamaan haitallista koodia uhrin tietokoneelle. Ohjelma käynnistyy, kun jokin hyökkääjän asettama ehto täyttyy.	• Koodikatselmoinnit (ennen tuotteen julkaisua) • EDR • UEBA • Penetraatiotestaus
Takaovi *(engl. Backdoor)*	Piilotettu toiminto, joka sallii pääsyn järjestelmän toimintoihin.	• Penetraatiotestaus • Koodikatselmoinnit • Vähimpien oikeuksien periaate • EDR • SOC • Palomuuri
Hyväksikäyttö *(engl. exploit)*	Hyökkääjä hyödyntää järjestelmän haavoittuvuutta.	• Penetraatiotestaus • Tietoturva-auditoinnit • EDR • IDS/IPS
Tiputin *(engl. dropper)*	Ohjelmalogiikka, joka lataa haittaohjelman järjestelmään; esimerkiksi haittaohjelma voi olla piilotettu tavalliseen tiedostoon.	• Käyttäjien koulutus • Sähköpostisuodatus • Vähimpien oikeuksien periaate • EDR • IDS/IPS • Verkkosuodattimet • Sovellusten valkolistaus
Auto-rooter	Hyökkääjän käyttämä haitallinen	• Penetraatiotestaus • EDR

	työkalu, jonka avulla hyökkääjä pyrkii hankkimaan pääkäyttäjäoikeudet järjestelmään.	• IDS/IPS • SOC • Segmentointi • Kovennukset • Sovellusten valkolistaus
Vakoiluohjelma	Kerää tietoja uhrin laitteesta ja lähettää tietoja eteenpäin.	• Käyttäjien koulutus • Sähköpostisuodatus • Vähempien oikeuksien periaate • EDR • Lokit • UEBA • Sovellusten valkolistaus
Datan käsittelyyn liittyvä ohjelmointivirhe *(engl. Injection flaw)*	Hyökkääjä käyttää ohjelmointivirhettä hyväkseen saadakseen ohjelman suorittamaan hyökkääjän koodia.	• Koodikatselmoinnit • Penetraatiotestaus • Staattinen ja dynaaminen ohjelmistotestaus
Salasanahyökkäys	Hyökkääjä pyrkii saamaan käyttäjän salasanan ja käyttäjätunnuksen haltuunsa.	• Henkilöstön koulutus • MFA • UEBA

Haittaohjelmat muun muassa virukset ja troijalaiset, muodostavat merkittävän uhkan internetin käyttäjille. Lieventämistoimenpiteiden täytyy olla ajan tasalla, koska niiden avulla tunnistetaan ja mahdollisesti poistetaan haittaohjelmat. (Enoka 2023, 117.) Anderson (2020, 629–630) mainitsee erilaisia tapoja puolustautua haittaohjelmia ja verkkohyökkäyksiä vastaan, muun muassa haavoittuvuusskannerin käyttäminen; se tutkii jatkuvasti tietoverkkoa tunnettujen haavoittuvuuksien varalta. Rajavalvontalaitteilla esimerkiksi palomuureilla ja välityspalvelimilla, tarkoituksena on suodattaa vierailtavien verkkosivustojen URL-osoitteet ja kriittisten sovellusten välityspalvelimet. Uhkatietoalustan käyttämisellä integroidaan syötteitä palveluntarjoajilta varoittamaan muun muassa haitallisista IP-osoitteista, ja lokianalyysityökalun avulla voidaan selvittää tapahtumien kulku.

Tyypillisten uhkien lieventämiskeinoista mainittakoon verkkosuodattimien käyttäminen, joiden tarkoituksena on estää pääsy haitallisille sivuille (Du, Safavi-Naini & Susilo 2003). Lisäksi sovellusten valkolistaus on eräs hyvä lieventämiskeino, koska valkolistauksen avulla määritellään, mitkä tiedostot saavat tehdä toimintoja järjestelmässä (Romana, Jha, Reddy, Pareek & Eswari 2015). Suurilla organisaatioilla on yleensä käytössä turvallisuusoperaatiokeskus (*engl. Security Operation Center, SOC*), joka on vastuussa tietojen käsittelystä, poikkeamien havaitsemisesta ja tutkimisesta, uhkien tiedustelusta, haavoittuvuuksien hallinnasta ja penetraatiotestauksesta. SOC:lla on suora pääsy tapahtumien hallintaan, tunkeutumisen estoon ja havaitsemiseen sekä virustorjuntajärjestelmään. Organisaatioiden tietoturvan kannalta SOC on erittäin tärkeässä asemassa uhkilta suojautumisessa. (Johansen 2022, 11, 28; Mughal 2022, 1–2.)

5 OIKEUSREKISTERIKESKUS

Viidennessä pääluvussa kuvataan Oikeusrekisterikeskus valtiollisena toimijana, jonka toiminta perustuu lakiin Oikeusrekisterikeskuksesta (2012/625). Tietotekniikkaympäristö on kuvattu yleisellä tasolla hahmottamaan oikeushallinnon tietotekniikkaympäristöä.

5.1 Oikeusrekisterikeskus valtiollisena toimijana

Oikeusrekisterikeskuksen toiminta perustuu lakiin Oikeusrekisterikeskuksesta. ”Oikeusrekisterikeskus on oikeusministeriön hallinnonalaan kuuluva virasto, jonka tehtävänä on:

1) toimia oikeusministeriön hallinnonalan tietojärjestelmien ja rekisterien rekisterinpitäjänä sen mukaisesti kuin niistä erikseen säädetään sekä välittää hallinnonalan viranomaisten ilmoittamia tietoja muille viranomaisille; (8.5.2020/348)

2) huolehtia sakkoihin, menettämisseuraamuksiin, maksuihin ja saataviin liittyvistä täytäntöönpanotehtävistä sekä käyttää valtion puhevaltaa näissä tehtävissä;

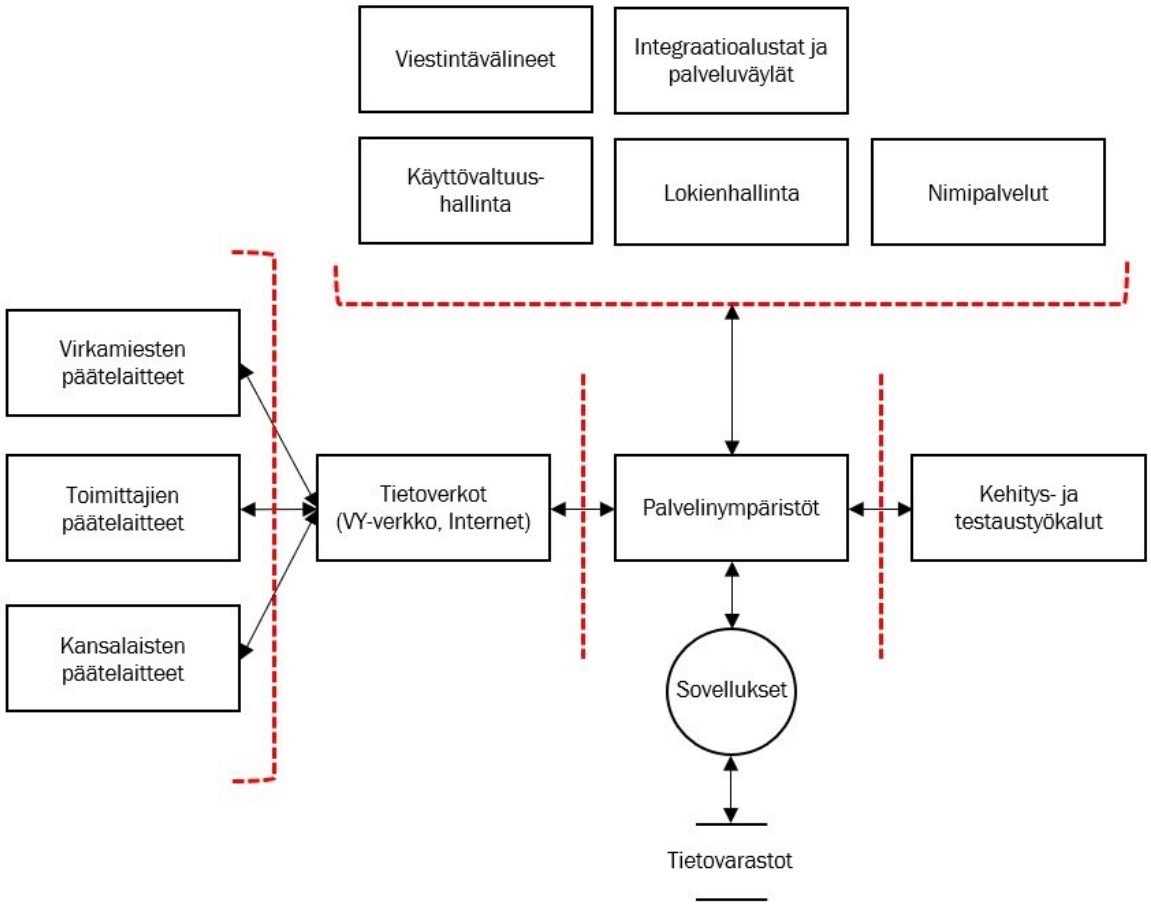
3) huolehtia oikeusministeriön hallinnonalaan kuuluvan yksikön toimeksiannosta tutkimustarkoituksiin käytettävien rekisterien ylläpidosta ja niihin liittyvistä tehtävistä;

4) huolehtia oikeusministeriön hallinnonalan tietojärjestelmien ylläpidosta ja kehittämisestä yhteistyössä hallinnonalan virastojen kanssa siten kuin palvelusopimuksissa on sovittu.

Oikeusrekisterikeskus tuottaa tarvittavat palvelut itse tai hankkii ne muilta palveluntuottajilta.” (Laki Oikeusrekisterikeskuksesta 16.11.2012/625 § 1.) Oikeusrekisterikeskus on oikeusministeriön alainen virasto, joka on kriittinen yhteiskunnallinen toimija. Tämän takia digiturvallisuus on varmistettava jokaisella toiminnan tasolla. Digitaalisen turvallisuuden kokonaisuus muodostuu riskienhallinnasta, toiminnan jatkuvuudesta ja varautumisesta, tieto- ja kyber-

turvallisuudesta ja tietosuojasta. Varmistamalla digitaalisen turvallisuuden luodaan pohja turvallisille ja laadukkaille digitaalisille palveluille. Digitaalisen turvallisuuden asianmukaisella toteuttamisella varmistetaan yhteiskunnan luottamuksen ylläpitäminen nykyisiä ja uusia toimintatapoja kohtaan. Riskit keskeisissä tietojärjestelmissä voivat vaarantaa hallinnonalan toiminnan, tietoturvallisuuden ja tietosuojan. (ORK 2023, 5.)

5.2 Oikeushallinnon tietotekniikkaympäristö



KUVIO 2 Oikeushallinnon tietotekniikkaympäristö

Oikeushallinnossa on noin 10 000 työntekijää ja päätelaitteita on yhteensä yli 10 000 kappaletta. Tämän lisäksi Oikeusrekisterikeskus hyödyntää toimittajien tarjoamia palveluita työtehtävissä. Kansalaisilla on mahdollisuus kirjautua Oikeusrekisterikeskuksen palveluihin, joihin voi tutustua Oikeusrekisterikeskuksen verkkosivuilla.

Oikeushallinto käyttää tietoverkkoina valtion yhteistä tietoverkkoa ja internetiä. Tietotekniikkaympäristö muodostuu yli sadasta tietojärjestelmästä, jotka ovat eri-ikäisiä ja ne käyttävät eri teknologioita. Palvelinympäristö on laaja

kokonaisuus, koska se sisältää useita satoja palvelimia, jotka sijaitsevat eri paikoissa. Lisäksi palvelimet käyttävät eri käyttöjärjestelmiä. Sovellus- ja tietovarastoja kuvastaa monipuolisuus, koska oikeushallinnolla on paljon erilaisia sopimusmalleja, teknologioita ja käyttötarkoituksia. Kehitys- ja testauskalut ovat myös laaja kokonaisuus, sillä osa järjestelmistä on matalakoodin kehitysalustaa (*engl. low-code development platform, LCDP*) ja osa symbolista konekieltä ja pääosa järjestelmistä sijoittuu LCDP:n ja symbolisen konekielen välille. Viestintälaitteet-, integraatioalustat ja palveluväylät-, käyttövaltuushallinta-, lokienhallinta- ja nimipalvelut-komponentit vastaavat suurta it-taloa, ja näitä kuvastaa kompleksisuus.

Kuvan nuolet vastaavat tietovirtoja, ja punainen katkoviiva kuvaa luottamusrajoja. Tietovirrat ovat tärkeitä hahmottaa kuvasta, koska niissä on aina jokin teknologiarajapinta. Huomioitavaa on se, että tämän organisaation sisällä on myös organisaatioiden välisiä rajapintoja tietyissä osioissa.

6 TULOKSET JA POHDINTA

Kuudennessa pääluvussa esitetään oikeushallinnon tietotekniikkaympäristön keskeiset riskit ja niiden hallintakeinot. Tutkimuksen viimeiset alaluvut käsittelevät luotettavuutta, eettisyyttä ja mahdollista jatkotutkimusaihetta.

6.1 Tunnistetut uhkat

Tunnistetut uhkat on mallinnettu luvun 5.2 tietotekniikkaympäristön mukaisesti. Aineistosta nousi esille uhkia, joiden vaikutus luokiteltiin vähäiseksi, kohtalaiseksi, merkittäväksi ja kriittiseksi. Merkittäviä ja kriittisiä uhkia havaittiin, mutta niitä ei ole esitetty tässä tutkimuksessa. Tässä tutkimuksessa esitellään tietoturvasyistä ainoastaan ne uhkat, joiden vaikutus on vähäinen tai kohtalainen. Tutkimuksen liitteessä kaksi tutkija antaa esimerkin kriittisen uhkan käsittelystä. Alaluvut alkavat taulukoilla, joihin on kirjattu käsitellyt uhkakortit, vaikutuksen arviointi ja lieventämistoimenpiteet. Tarkasteluun on nostettu keskeisimmät oikeushallinnon tietotekniikkaympäristön riskit ja riskien hallintakeinot.

6.1.1 Virkamiesten päätelaitteet

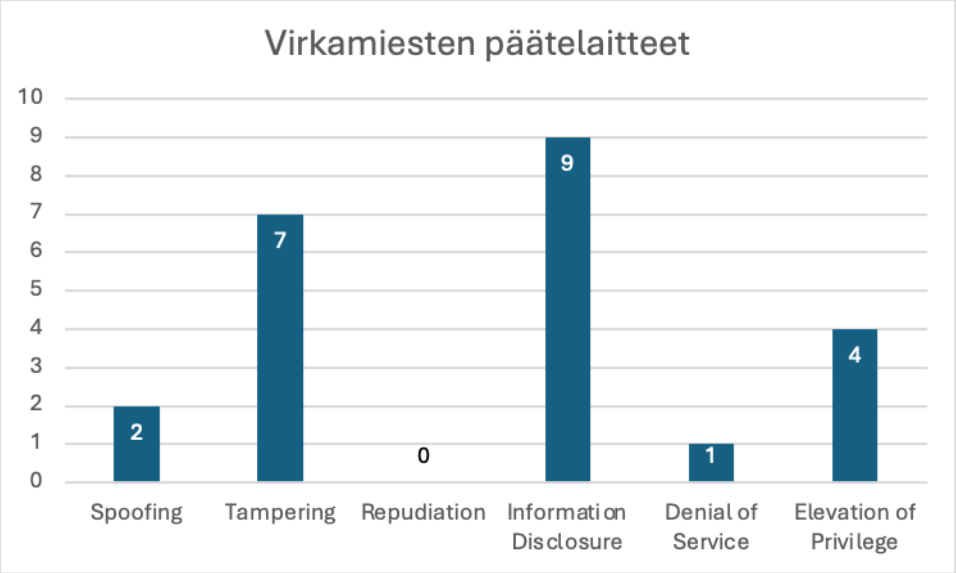
TAULUKKO 18 Virkamiesten päätelaitteet

Kortin skenaario	Kuvaus	Luokittelu	Lieventämistoimenpiteet
Bring your own (exploited device)	(Mobiilipäätelaitteiden) tietoverkkoon on mahdollista kytkeä omia turvattomia laitteita, jotka vaarantavat edelleen tietoverkon turvallisuuden.	Vähäinen/ Spoofing	• Pääsynvalvonta, esimerkiksi 802.1X. Yksittäinen käyttäjä ei pysty ottamaan uutta päätelaitetta käyttöön itsenäisesti.
Etäyhteyden käyttö omilta päätelaitteilta	On mahdollista siirtää VPN myös omaan päätelaitteeseen.	Kohtalainen/ Spoofing	• Virkamiesten ohjeistaminen.
Event Trigged Malware	Käytössä olevassa ohjelmistossa on piilotettuna lepäävä haittaohjelma, joka käynnistyy tietyn ennalta määritellyn tapahtuman yhteydessä.	Kohtalainen/ Tampering	• Ohjelmistojen lähdekoodin tarkistus. • Ohjelmistojen tietoturva-arvioinnit. • Ohjelmistojen allekirjoitukset.

			• Hiekkalaatikko tai EDR, joka estää ei-valkolistatun käyttäytymisen. • Tietoliikenne- ja prosessilokien valvonta.
Hallintatyökaluissa haittaohjelma	Hallintatyökaluun upotettu haittaohjelma vaarantaa ympäristön turvallisuuden.	Kohtalainen/ Tampering	• Hallintatyökalujen turvallisuus on tarkistettu. • Verkkoyhteyksien rajoittaminen ja valvonta. • Toimittajat eivät saa ottaa itsenäisesti käyttöön erilaisia hallintatyökaluja. • Versiopäivitysten tarkistusten yhteydessä WhiteBox-testaus. • Virkamiesten ohjeistus.
DLL Attacks	Hyökkääjä manipuloi DLL-tiedostojen latausjärjestystä ja saa haavoittuvan sovelluksen käynnistämään haittakoodia.	Kohtalainen/ Tampering	• Käyttöoikeuksien rajoittaminen. • Applocker. • EDR. • Prosessilokien valvonta.
Automaattisesti käynnistyvät ohjelmat	Hyökkääjä lisää automaattisesti käynnistyviin ohjelmiin haittaohjelman (rekisteriavaimet, autorun, loginskriptit, palvelut).	Kohtalainen/ Tampering	• Prosessien lokivalvonta. • EDR. • Automaattisesti käynnistyvien ohjelmien katselmointi. • Tietoturva-arvioinnit. • Käyttöjärjestelmän kovennukset.
Windows Service Recovery Actions	Hyökkääjä käyttää Windowsin palvelujen toipumismekanismia oman koodin suorittamiseen.	Kohtalainen/ Tampering	• EDR. • Prosessien lokivalvonta. • Palvelujen asetusten tarkistaminen.
New User Added	Hyökkääjä lisää uuden käyttäjätilin, jolla suorittaa operaatioitaan edelleen.	Kohtalainen/ Tampering	• Lokivalvonta. • Käyttöjärjestelmän kovennukset.
Malware Injection into Client Software	Haittaohjelma injektoidaan käynnissä olevaan prosessiin. Erityisen kiinnostava tapaus, jos kohdeprosessi on käynnissä korkeammalla käyttöoikeustasolla.	Kohtalainen/ Tampering	• Ohjelmistojen lähdekoodin tarkistus. • Ohjelmistojen tietoturva-arvioinnit. • Ohjelmistojen allekirjoitukset. • Sovellusten valkolistaus tai EDR, joka estää ei-valkolistatun käyttäytymisen. • Tietoliikenne- ja prosessilokien valvonta.
Credential stuffing	Internetistä löytyviä käyttäjätunnus ja salasanapareja voidaan käyttää myös meidän palveluissamme. (Virkamiehet voivat käyttää työympäristön palveluiden salasanoja uudestaan.)	Vähäinen/ Information Disclosure	• Henkilöstön koulutus. • Aito kertakirjautuminen. • Avointen lähteiden tiedustelu (OSINT). • MFA.
Windows Background Intelligent Transfer Service (BITS)	BITS-protokollaa käytetään komentokanavana.	Vähäinen/ Information Disclosure	• Zero trust -arkkitehtuuri. • Tietoliikenteen inspektointi. • Lokivalvonta. • EDR.
Exfiltration over physical medium	Hyökkääjä salakuljettaa salassa pidettävää tietoa ulos esimerkiksi muistitikulla tai papereilla.	Kohtalainen/ Information Disclosure	• PAM. • DLP. • Työasemakovennukset. • Koulutus. • Turvallisuusselvitykset. • Yhdyskäytäväratkaisut.

			• USB-laitteiden käytön rajoittaminen.
Tiedostojen käyttöoikeudet	Hyökkääjä voi lukea arkaluonteista tietoa tiedostosta, jossa on väärin tai puutteellisesti määritelty käyttöoikeus.	Kohtalainen/ Information Disclosure	• Tietoturva-arviointi. • Tiedosto-oikeuksien skannaus ja analysointi.
Asiakirjan muutoshistoria	Tietoa vuotaa asiakirjan muutoshistoriassa tai metatiedoissa.	Kohtalainen/ Information Disclosure	• Toimisto-ohjelmien asetusten koventaminen. • DLP-ominaisuudet. • Käyttäjien ohjeistus.
DNS as C2	DNS-protokollaa käytetään komentokanavana.	Kohtalainen/ Information Disclosure	• Zero trust -arkkitehtuuri. • Tietoliikenteen inspektointi. • DNS-kyselylokien tarkastaminen.
Gmail, tumbler, salesforce, twitter as C2	Hyökkääjä käyttää yleisiä verkkopalveluita komentokanavana.	Kohtalainen/ Information Disclosure	• Zero trust-arkkitehtuuri. • Tietoliikenteen inspektointi. • Lokivalvonta. • EDR.
Kovalevynsalauksen murtaminen	Hyökkääjä voi murtaa tiedosto- tai kovalevysalauksen kokeilemalla mahdollisia salasanoja.	Kohtalainen/ Information Disclosure	• Aikaa vievät salausalgoritmit. • Kovalevynsalauksen tietoturva-arviointi. • CAA-prosessi. • Työaseman automaattinen lukkiutuminen ja sammutus.
HTTP/S as EXFIL	HTTP/S-protokollaa käytetään komentokanavana.	Kohtalainen/ Infromation Disclosure	• Zero trust -arkkitehtuuri. • Tietoliikenteen inspektointi. • DNS-kyselylokien tarkastaminen.
Virrankulutus	Hyökkääjä aiheuttaa akun tyhjentymisen lisäkuormituksella.	Vähäinen/ Denial of Service	• Sähkön jakelun varmistaminen. • EDR. • Työasemien kuorman seuranta.
Hyökkääjä pääsee fyysisesti toimitiloihin	Hyökkääjä pääsee fyysisesti toimitiloihin ja siitä edelleen tietoverkkoihin.	Kohtalainen/ Elevation of Privilege	• Turvallisuussopimus. • Kulunvalvonta. • Vartiointipalvelu. • Työasemat lukitsevat itsensä, kun ovat tarpeeksi kauan käyttämättä. • Laitetunnistus, esimerkiksi 802.1X. • Verkon turvallisuuden valvonta (IDS/IPS). • Päätelaitteissa kovalevynsalaus. • Painotetaan etätöitä. • Vierailijakäytänteet.
Dirty USB	USB-laite päätyy kiinni verkossa olevaan laitteeseen. USB-laite voi olla muistitikku tai jotain muuta, sekä verkossa oleva laite voi olla työasema tai jotain muuta, esimerkiksi tulostin.	Kohtalainen/ Elevation of Privilege	• Hyväksyttyjen USB-laitteiden käyttö. • Lokivalvonta (USB-laitteet). • Erilliset skannauskoneet. • Virtuaalityöasemat. • Henkilöstön koulutus. • Turvallisuusselvitykset kaikista tiloihin pääsevistä. • Toimitusketjujen hallinta.
Sovelluksen oikeustaso	Hyökkääjä voi hyödyntää sovelluksille myönnettyjä liiallisia oikeuksia.	Kohtalainen/ Elevation of Privilege	• Sovellusten ajaminen pienimmillä mahdollisilla oikeuksilla. • Käyttöjärjestelmän oletusasetusten koventaminen.

			• Sovellusten käyttöoikeuksien katselmointi. • Penetraatiotestaus.
XSS	Hyökkääjä voi heijastaa syötettä takaisin käyttäjille.	Kohtalainen/ Elevation of Privilege	• Kehitystyön aikainen tietoturvatestaus. • Tietoturvavaatimukset. • Penetraatiotestaus.



KUVIO 3 Virkamiesten päätelaitteet

Virkamiesten päätelaitteet -komponentissa havaittiin yhteensä kaksikymmentäkolme uhkaa, joista peukalointi- ja tietojenpaljastamisuhkia oli eniten. Huomioitavaa on se, että oikeushallinnon tietotekniikkaympäristössä virkamiesten päätelaitteita on yli 10 000 kappaletta, joten virkamiesten tietoturvakoulutuksen tärkeyttä on syytä korostaa.

Teknisestä näkökulmasta tarkasteltuna EDR:n käyttäminen auttaa uhkien lieventämistä. Toinen keskeinen lieventämiskeino on lokienvalvonta, koska sen avulla kyetään selvittämään tapahtumien aika ja se, mitä on tapahtunut. Korttisarjojen uhkaskenaariot perustuvat hyvin pitkälti hyökkääjän toimenpiteisiin ja nimenomaan ulkoisiin uhkiin. Sisäiset uhkat ovat myös relevantteja skenaarioita, ja ne voivat aiheuttaa vakavia riskejä. Sisäinen uhka tarkoittaa sitä, että työntekijällä tai konsultilla on pääsy tietojärjestelmiin, tietoverkkoihin ja dataan, joiden avulla voidaan tehdä haitallisia toimenpiteitä tarkoituksella tai tahattomasti. Mahdollisia uhkaskenaarioita voivat olla esimerkiksi seuraavat:

- Henkilö poistaa tietokannan ja tietokannan varmuuskopion.
- Henkilö ohjelmoi järjestelmään takaoven, jota käytetään haitallisten toimenpiteiden suorittamiseen.
- Henkilö poistaa tietoja vahingossa ja muuttaa lokeja tapahtuman jälkeen virheen peittelemiseksi.

Tahattomia uhkaskenaarioita voivat olla esimerkiksi seuraavat:

- Henkilö käyttää ohjelmistoa väärin ja aiheuttaa ohjelmiston häiriötilaan.
- Henkilö poistaa vahingossa kriittisiä tietoja.

(Janca 2021, 8–9.) Oleellisista edellä mainituissa uhkaskenaarioissa on korostaa vähimpien oikeuksien periaatetta. Mikäli jokaisella työntekijällä on pääsy järjestelmän jokaiseen osaan, niin se muodostaa mahdollisen riskin. Lieventämistoimenpiteenä on myöntää työntekijälle vain työnkuvan mukaiset oikeudet. Lokienhallinta on myös tässä tapauksessa oleellinen lieventämistoimenpide. Tietojen paljastamisuhkien lieventämiskeinona on syytä käyttää tietovuotojen estojärjestelmää (*engl. Data Loss Prevention, DLP*), joka auttaa tunnistamaan henkilöt, jotka yrittävät viedä järjestelmästä kriittisiä tietoja (Anderson 2020, 630).

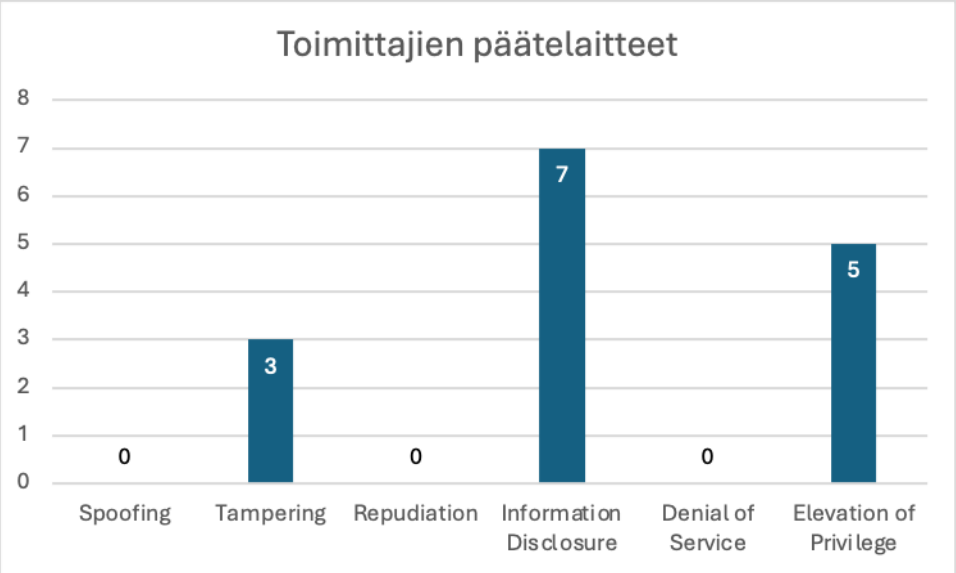
6.1.2 Toimittajien päätelaitteet

TAULUKKO 19 Toimittajien päätelaitteet

Kortin skenaario	Kuvaus	Luokittelu	Lieventämistoimenpiteet
Event Trigged Malware	Käytössä olevassa ohjelmistossa on piilotettuna lepäävä haittaohjelma, joka käynnistyy tietyn ennalta määritellyn tapahtuman yhteydessä.	Kohtalainen/ Tampering	• Ohjelmistojen lähdekoodin tarkistus. • Ohjelmistojen tietoturva-arvioinnit. • Ohjelmistojen allekirjoitukset. • Hiekkalaatikko tai EDR, joka estää ei-valkolistatun käyttäytymisen. • Tietoliikenne- ja prosessilokien valvonta.
New User Added	Hyökkääjä lisää uuden käyttäjätilin, jolla suorittaa operaatioitaan edelleen.	Kohtalainen/ Tampering	• Lokivalvonta. • Käyttöjärjestelmän kovennukset.
Malware Injection into Client Software	Haittaohjelma injektoidaan käynnissä olevaan prosessiin. Erityisen kiinnostava tapaus, jos kohdeprosessi on käynnissä korkeammalla käyttöoikeustasolla.	Kohtalainen/ Tampering	• Ohjelmistojen lähdekoodin tarkistus. • Ohjelmistojen tietoturva-arvioinnit. • Ohjelmistojen allekirjoitukset. • Hiekkalaatikko tai EDR, joka estää ei-valkolistatun käyttäytymisen. • Tietoliikenne- ja prosessilokien valvonta. • Injektoinnin estävät ympäristön kovennukset.
Windows Background Intelligent Transfer Service (BITS)	BITS-protokollaa käytetään komentokanavana.	Vähäinen/ Information Disclosure	• Zero trust -arkkitehtuuri. • Tietoliikenteen inspektointi. • Lokivalvonta. • EDR.
Exfiltration over physical medium	Hyökkääjä salakuljettaa salassa pidettävää tietoa ulos esimerkiksi muistitikulla tai papereilla.	Kohtalainen/ Information Disclosure	• PAM. • DLP. • Työasemakovennukset. • Koulutus. • Turvallisuusselvitykset. • Yhdyskäytäväratkaisut. • USB-laitteiden käytön rajoittaminen.
HTTP/S as EXFIL	HTTP/S-protokollaa käytetään komentokanavana.	Kohtalainen/ Information Disclosure	• Zero trust -arkkitehtuuri. • Tietoliikenteen inspektointi. • DNS-kyselylokien tarkastaminen.

Asiakirjan muutoshistoria	Tietoa vuotaa asiakirjan muutoshistoriassa tai metatiedoissa.	Kohtalainen/ Information Disclosure	• Toimisto-ohjelmien asetusten koventaminen. • DLP-ominaisuudet. • Käyttäjien ohjeistus.
Tiedostojen käyttöoikeudet	Hyökkääjä voi lukea arkaluonteista tietoa tiedostosta, jossa on väärin tai puutteellisesti määritelty käyttöoikeus.	Kohtalainen/ Information Disclosure	• Tiedostojärjestelmän käyttöoikeuksien oletusten kovennukset. • Tietoturva-arviointi. • Tiedosto-oikeuksien skannaus ja analysointi.
DNS as C2	DNS-protokollaa käytetään komentokanavana.	Kohtalainen/ Information Disclosure	• Zero trust -arkkitehtuuri. • Tietoliikenteen inspektointi. • DNS-kyselylokien tarkastaminen.
Kovalevynsalauksen murtaminen	Hyökkääjä voi murtaa tiedosto- tai kovalevysalauksen kokeilemalla mahdollisia salasanoja.	Kohtalainen/ Information Disclosure	• Aikaa vievät salausalgoritmit. • Kovalevynsalauksen tietoturva-arviointi. • CAA-prosessi. • Työaseman automaattinen lukkiutuminen ja sammutus.
Hyökkääjä pääsee fyysisesti toimitiloihin	Hyökkääjä pääsee fyysisesti toimitiloihin ja siitä edelleen tietoverkkoihin.	Kohtalainen/ Elevation of Privilege	• Kulunvalvonta. • Vartiointipalvelu. • Työasemat lukitsevat itsensä, kun ovat tarpeeksi kauan käyttämättä. • Laitetunnistus, esimerkiksi 802.1X. • Verkon turvallisuuden valvonta (IDS/IPS). • Päätelaitteissa kovalevynsalaus. • Painotetaan etätöitä. • Vierailijakäytänteet.
Toimittajaportaali	Hyppykoneportaali mahdollistaa suunnittelemattoman laajan pääsyn ympäristöön.	Kohtalainen/ Elevation of Privilege	• Käyttövaltuuksien ajantasaisuuden valvonta. • Nopea tietoturvapäivitysten asentaminen. • Jatkuva palvelun turvallisuuden skannaus internetistä.
Sovelluksen oikeustaso	Hyökkääjä voi hyödyntää sovelluksille myönnettyjä liiallisia oikeuksia.	Kohtalainen/ Elevation of Privilege	• Sovellusten ajaminen pienimmillä mahdollisilla oikeuksilla. • Käyttöjärjestelmän oletusasetusten koventaminen. • Sovellusten käyttöoikeuksien katselmointi. • Penetraatiotestaus.
Infected authorized vendor laptop	Kiinteistöautomaation, kulunvalvonnan, turvatekniikan tai esitystekniikan ylläpidossa käytetty työasema sisältää haittaohjelman.	Kohtalainen/ Elevation of Privilege	• Ei kytketä mainittuja palveluja Valtion yhteiseen tietoverkkoon. • Mainitut palvelut kytketään erilliseen IoT-verkkoon. • Mitään näitä ei kytketä koskaan internetiin. • Tietoturvapäivitykset. • Tietoturva-auditointi. • Tietoliikennelokien valvonta. • Laitetunnistus.
Tietomurto toimittajalla	Hyökkääjä on päässyt teknologiatoimittajan järjestelmiin.	Kohtalainen/ Elevation of Privilege	• Toimitusketjujen hallinta. • Turvallisuussopimukset. • Tietoturva-arvioinnit. • Valvonta- ja suodatusjärjestel-

			mät. • Offline-varmuuskopiot. • Haavoittuvuusskannaukset. • Hyväksyntäkäytännöt.



KUVIO 4 Toimittajien päätelaitteet

Toimittajien päätelaitteiden uhkaskenaariot mukailevat virkamiesten päätelaitteiden uhkaskenaarioita. Eron tekee se, että oikeuksien korottamisuhkat nousevat enemmän esille.

6.1.3 Kansalaisten päätelaitteet

TAULUKKO 20 Kansalaisten päätelaitteet

Kortin skenaario	Kuvaus	Luokittelu	Lieventämistoimenpiteet
Kansalaiset käyttävät omia päätelaitteitaan	Mikäli päätelaite on esimerkiksi saastunut, tämä voi vaikuttaa siten, että sieltä tulee meidän suuntaamme erilaista materiaalia.	Vähäinen	• Palvelujen käyttöön liittyvän ohjeistuksen yhteydessä huomautetaan kansalaisille heidän päätelaitteidensa turvallisuudesta (erityisesti Traficomin ohjeet). • Kansalaisilta tulevien materiaalien turvallisuuden tarkistaminen (tallennuspalvelu, sähköpostien suodatus). • Websovelluspalomuuri.

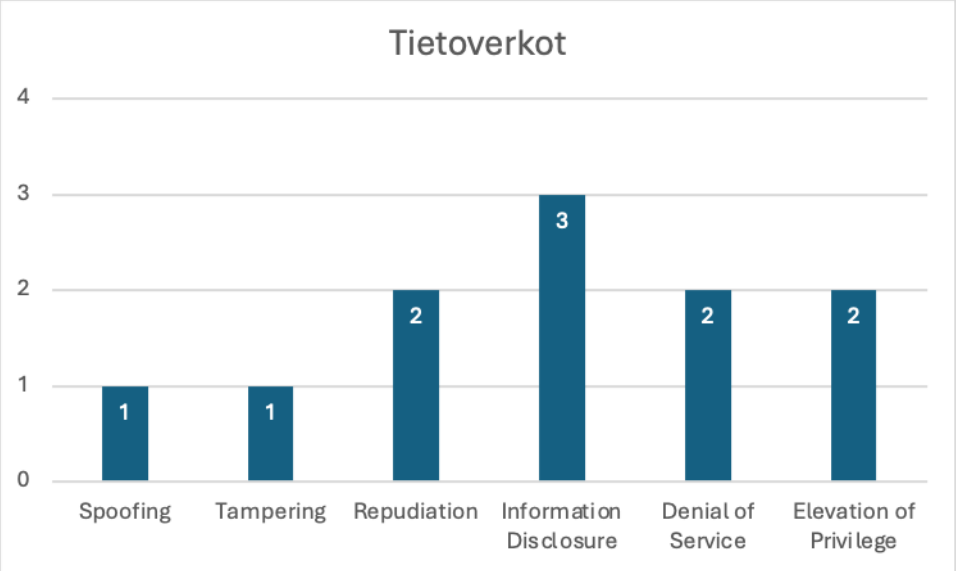
Tämänkaltainen uhkaskenaario on mahdollinen, mutta oleellisina lieventämistoimenpiteinä on varmistaa tulevan materiaalin turvallisuus ja websovelluspalomuurin käyttäminen.

6.1.4 Tietoverkot

TAULUKKO 21 Tietoverkot

Kortin skenaario	Kuvaus	Luokittelu	Lieventämistoimenpiteet
Kalastelusivustot	Hyökkääjä voi hämätä asiakasta, koska on liian monta tapaa identifioida palvelin tai palvelu.	Kohtalainen/ Spoofing	• Selkeä nimipolitiikka. • Varmennepolitiikka. • Tietoliikenteen inspektointi. • DNS-kyselylokien valvonta.
Broadcast/Multicast protocol poisoning	LLMNR-, NetBios- ja mDNS-välimuistien manipulointi	Kohtalainen/ Tampering	• UEBA/vastaava. • Verkkolaitteiden lokitietojen ja kuorman seuranta.
Stale Network Address Configurations	Hyökkääjä käyttää hyväkseen kiinteästi konfiguroitua vanhaa osoitetta, esimerkiksi vanhentunut domain, IP-osoite tai S3-bucket Huom! Erityisesti internetissä oleviin resursseihin liittyvä, mutta myös sisäverkossa varsinkin internetissä olevia resursseja käytettäessä.	Vähäinen/ Repudiation	• Riippuvuusskannaukset. • Huolellinen konfiguraationhallinta. • Koulutus. • Verkkotunnuspolitiikka.
DNS-tietojen väärentäminen	Nimipalvelutietoja voi väärentää.	Kohtalainen/ Repudiation	• DNSSEC. • Palvelinympäristöjen sisäiset hosts-tiedostot.
Salaamaton tietoliikenne	Hyökkääjä voi lukea viestiliikennettä, joka ei ole salattu.	Kohtalainen/ Information Disclosure	• Riittävän salauksen käyttö. • Jokaisen tietoverkossa liikennöivän palvelun tietoturva-arviointi. • Salaamattoman tietoliikenteen seulonta ja käsittely poikkeamana.
Puutteellisesti salattu tietoliikenne	Hyökkääjä voi lukea huonolla tai epästandardilla algoritmilla salattua dataa.	Kohtalainen/ Information Disclosure	• Määriteltyjen turvalliseksi tiedettyjen salausasetusten käyttö. • Jokaisen tietoverkossa liikennöivän palvelun tietoturva-arviointi. • CAA-prosessi. • Salaamattoman tietoliikenteen seulonta ja käsittely poikkeamana.
Man in the middle	Hyökkääjä voi kaapata tietoliikennettä olemalla välissä, koska päätepisteitä ei tunnisteta.	Kohtalainen/ Information Disclosure	• Zero trust -arkkitehtuuri. • Salaamattoman liikenteen esto verkkotasolla.
(D)DOS	Hyökkääjä estää ylikuormituksella sovelluksen tai tunnistautumispalvelun toiminnan.	Kohtalainen/ Denial of Service	• Palvelunestohyökkäysten estopalvelu. • Kuormantasaimet. • CDN-ratkaisut. • Skaalautuva sovellusarkkitehtuuri. • Valvontajärjestelmät. • Sovelluspalomuurit.
(D)DOS-amplifikaatio	Hyökkääjä saa vahvistettua palvelunestohyökkäystään, jossain palvelussa olevan haavoittuvuuden vuoksi.	Kohtalainen/ Denial of Service	• Amplifikaation mahdollistavien palvelujen (esim. DNS) tunnistaminen, koventaminen ja tietoturva-arviointi.
Exploitable API	Verkkoon on liitetty IoT-laitteita, jotka yrittävät tarjota hallintarajapinnan internettiin (esimerkiksi kahvinkeittimet).	Kohtalainen/ Elevation of Privilege	• Kytketään vain niille tarkoitettuun verkkoon. • Laitteiden tietoturva-arviointi. • Verkon ja lokien valvonta

			(esim. DNS-lokit).
Internal Password Spray	Hyökkääjä käyttää IoT-laitteiden oletussalasanoja saadakseen pääsyn näihin laitteisiin.	Kohtalainen/ Elevation of Privilege	• IoT-laitepolitiikka. • IoT-laitteiden selkeä vastuutus tai palveluvastaavat. • Oletussalasanojen vaihto. • Verkkosegmentointi.



KUVIO 5 Tietoverkot

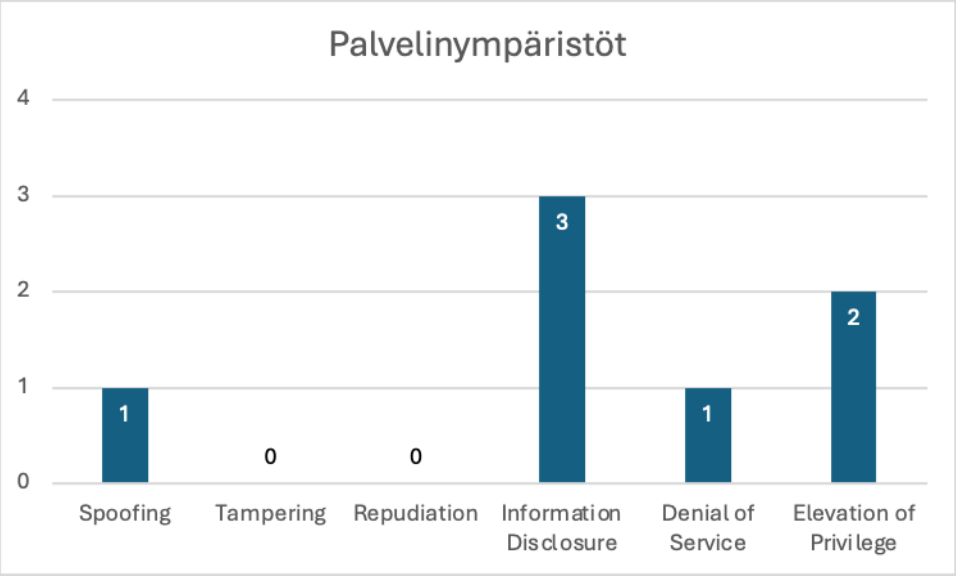
Tietoverkot-komponentissa esiintyi jokaista STRIDE-metodologian uhkatyyppiä. Tietoverkkojen uhkaskenaarioissa on syytä kiinnittää huomiota tietoverkkojen suojaustoimenpiteisiin. Yleisesti ottaen tietoverkkojen ja erityisesti internetin kautta tehtyjen tietoturvahyökkäysten estäminen ja havaitseminen on oleellisin osa tietoturvasuunnittelua. Verkkohyökkäysten puolustamiseksi voidaan luetella neljä tekijää, jotka auttavat puolustautumaan verkkohyökkäyksiltä. Ensimmäinen tekijä on järjestelmän pitäminen ajan tasalla ja konfiguroituna, koska tämä vähentää hyökkäyspinta-alaa. Suodattimien käyttö, esimerkiksi palomuuri, vähentää verkkohyökkäyksiä, ja tunkeutumisen havaitsemisohjelmat auttavat tarkkailemaan kohdeverkkoa epäilyttävän toiminnan varalta. Viimeisenä tekijänä ovat erilaiset salausprotokollat, jotka mahdollistavat suojautumisen tietyiltä hyökkäyksiltä. (Anderson 2008, 652, 677.)

6.1.5 Palvelinympäristöt

TAULUKKO 22 Palvelinympäristöt

Kortin skenaario	Kuvaus	Luokittelu	Lieventämistoimenpiteet
Haitallisen palvelun piilottaminen	Hyökkääjä piilottaa haittaohjelman palvelimen normaalisti käyttämään porttiin.	Kohtalainen/ Spoofing	• Tietoliikenteen inspektointi. • Tietoliikennelokien valvonta.
Exfiltration over physical medium	Hyökkääjä salakuljettaa salassa pidettävää tietoa ulos esimerkiksi muistitikulla tai papereilla.	Kohtalainen/ Information Disclosure	• PAM. • DLP. • Työasemakovennukset. • Koulutus.

			• Turvallisuusselvitykset. • Yhdyskäytäväratkaisut. • Materiaalien ja medioiden käsittelyn politiikka.
Windows Background Intelligent Transfer Service (BITS)	BITS-protokollaa käytetään komentokanavana.	Vähäinen/ Information Disclosure	• Välityspalvelimen käytön pakottaminen. • Lokivalvonta. • EDR.
Tiedostojen käyttöoikeudet	Hyökkääjä voi lukea arkaluonteista tietoa tiedostosta, jossa on väärin tai puutteellisesti määritelty käyttöoikeus.	Kohtalainen/ Information Disclosure	• Tiedostojärjestelmän käyttöoikeuksien oletusten kovennukset. • Tietoturva-arviointi. • Tiedosto-oikeuksien skannaus ja analysointi.
Pilvitilien rajat	Hyökkääjä käyttää pilvitilin rahavarauksen tyhjäksi.	Kohtalainen/ Denial of Service	• Rajoittamattomat pilvitilit. • Pilvitilin kustannusten valvonta. • Pilvitilin käyttövaltuushallinta. • Kovennetut resurssit pilvitilillä.
External cloud access	Hyökkääjä saa pääsyn pilviresurssiin, kuten palvelin, sovellus tai tietovarasto. Hyökkääjä käyttää tätä hyväkseen hyökätäkseen muualle.	Kohtalainen/ Elevation of Privilege	• Pilviympäristöresurssien tietoturvallisuuden valvonta. • Käyttövaltuuksien katselmointi. • Käyttövaltuuksien jatkuva automaattinen tarkastus. • Verkon- ja lokienvalvonta. • Pilvi- ja on-premises-konesaleja ei yhdistetä suoraan toisiinsa. • Eri pilvialustoja ei yhdistetä toisiinsa. • Tarkka segmentointi pilvialustoilla.
SMB Weakness	Hyökkääjä hyödyntää SMB-protokollan vanhojen versioiden ominaisuuksia saadakseen pääsyn resursseihin.	Vähäinen/ Elevation of Privilege	• Haavoittuvuusskannaukset. • Kovennetut konfiguraatiot. • Lokivalvonta.



KUVIO 6 Palvelinympäristöt

Palvelinympäristöissä havaittiin yhteensä seitsemän uhkaa. Huomioitavaa on se, että palvelinympäristöt ovat laaja kokonaisuus, johon sisältyy useita satoja

palvelimia, ja nämä käyttävät eri käyttöjärjestelmiä. Valvontatoimenpiteillä, muun muassa lokien ja tietoliikennelokien valvonnalla, kyetään reagoimaan hyökkääjän tekemiin haitallisiin toimenpiteisiin. EDR on myös eräs tehokas keino haitallisen toiminnan havaitsemisessa. Uhkaskenaarioista keskeisenä nostona on tietojen suodatustekniikka (*engl. Data Exfiltration)*, jota hyökkääjät käyttävät arkaluonteisen tiedon saamiseksi. Kyseinen hyökkäys voidaan tehdä manuaalisesti tai etänä, ja sitä voi olla vaikea havaita. Vaikeus muodostuu siinä, että toiminta muistuttaa normaalia toimintaa verkkoliikenteessä. Yleisimpiä tekniikoita on tietojenkalasteluhyökkäykset ja tietojen varastaminen esimerkiksi USB-muistitikulta. Lieventämisstrategioina voidaan käyttää muun muassa DLP- tai EDR-ratkaisua ja hyviä tiedonhallintakäytäntöjä. (Peiris, Pillai & Kudrati 2022, 147.)

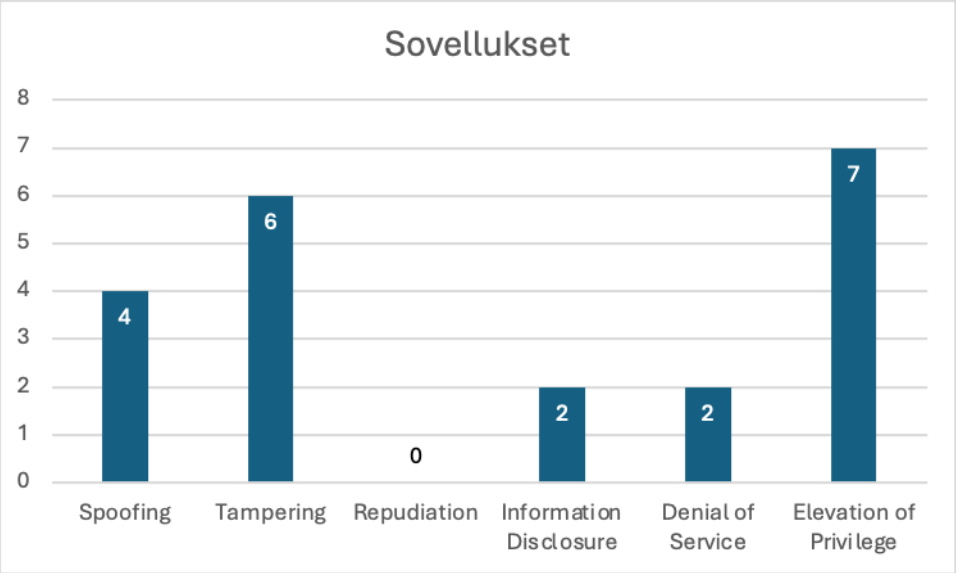
6.1.6 Sovellukset

TAULUKKO 23 Sovellukset

Kortin skenaario	Kuvaus	Luokittelu	Lieventämistoimenpiteet
Tunnistautumisen kiertäminen	Hyökkääjä voi yhdistää palveluun ilman tunnistautumista, koska tunnistautumisen on oletettu tapahtuvan toisella kerroksella.	Kohtalainen/ Spoofing	• Järjestelmien arkkitehtuurin katselmointi. • Penetraatiotestaus.
Palvelimena esiintyminen	Hyökkääjä voi väittää olevansa palvelin, koska asiakasohjelmisto ei tarkasta palvelimen identiteettiä.	Kohtalainen/ Spoofing	• Varmenteiden tarkistukset. • Penetraatiotestaus. • Järjestelmien tietoturva-arvioinnit.
Tunnistautumisen kiertäminen	Hyökkääjä voi yhdistää palveluun ilman tunnistautumista käyttäen autentikoimatonta tai salaamatonta kanavaa.	Kohtalainen/ Spoofing	• Tietoverkko- ja sovellusarkkitehtuuri. • Penetraatiotestaus. • Verkonvalvonta.
Salausavaimien vuotaminen	Hyökkääjä voi varastaa palvelimen salaus- tai rajapinta-avaimet ja käyttää niitä.	Kohtalainen/ Spoofing	• Salaisuuksienhallinta. • Avainten säilyttäminen HSM:llä. • Palvelinten koventaminen. • DLP.
Hallintatyökaluissa haittaohjelma	Hallintatyökaluun upotettu haittaohjelma vaarantaa ympäristön turvallisuuden.	Vähäinen/ Tampering	• Hallintatyökalujen turvallisuus on tarkistettu. • Verkkoyhteyksien rajoittaminen ja valvonta. • Toimittajat eivät saa ottaa itsenäisesti käyttöön erilaisia hallintatyökaluja. • Versiopäivitysten tarkistusten yhteydessä WhiteBox-testaus. • Hallintatyökaluja koskevat sopimusehdot.
Haavoittuva pääsynhallinta	Sovelluksen pääsynhallintapäätöksiä tehdään useissa paikoissa.	Kohtalainen/ Tampering	• Tiukka MVC-mallin käyttö. • Pienet kovennetut pääsynhallintakomponentit. • Koodikatselmoinnit. • Penetraatiotestaus.
Kryptohaavoittuvuus	Hyökkääjä hyväksikäyttää ei-standardia avaintenvaihtoa tai eheystarkastusta.	Kohtalainen/ Tampering	• Standardien salausprotokollien käyttö. • Koodikatselmoinnit.

			• Penetraatiotestaus. • Kansallinen CAA-prosessi.
Parametrien manipulointi	Hyökkääjä voi manipuloida parametreja luottamusrajan yli tai validaation jälkeen.	Kohtalainen/ Tampering	• Muuntamattomat parametrit. • Keskeisten tietojen allekirjoitus. • Defensiivinen ohjelmointi. • Koodikatselmoinnit. • Penetraatiotestaus.
Virhe resurssien nimien käsittelyssä	Hyökkääjä voi ohittaa oikeustarkistuksen, koska resurssien nimiä ei kanonisoida ennen käyttöä.	Kohtalainen/ Tampering	• Case insensitive -pyyntöjen käsittely. • Koodikatselmoinnit. • Penetraatiotestaus.
Sovelluksen tilan manipulointi	Hyökkääjä voi manipuloida sovelluksen tilaa.	Kohtalainen/ Tampering	• Tiukka MVC-mallin käyttö. • Middleware-arkkitehtuuri. • Defensiivinen ohjelmointi. • Koodikatselmoinnit. • Penetraatiotestaus.
Virheilmoituksien sisältö	Hyökkääjä voi nähdä virheilmoituksia, joissa on turvallisuutta vaarantavaa tietoa.	Kohtalainen/ Information Disclosure	• Virheilmoitusten sisällön katselmointi. • Koodikatselmoinnit. • Tietoturva-arvioinnit. • Penetraatiotestaus.
Kiinteät salausavaimet	Hyökkääjä saa selville kiinteän salausavaimen.	Kohtalainen/ Information Disclosure	• Salausten toteutusta koskevat vaatimukset. • Salaisuuksienhallinta. • Avainten hallinnan prosessi. • Tietoturva-arvioinnit. • CAA-prosessi.
Persistentti (D)DOS	Hyökkääjä estää ylikuormituksella sovelluksen tai tunnistautumispalvelun toiminnan niin, että vaikutus jatkuu hyökkäyksen päätyttyäkin (esimerkiksi levyt täyttyneet).	Kohtalainen/ Denial of Service	• Palvelunestohyökkäysten estopalvelu. • Kuormantasaimet. • CDN-ratkaisut. • Skaalautuva sovellusarkkitehtuuri. • Valvontajärjestelmät. • Sovelluspalomuurit.
Lokituksen kuormitus	Hyökkääjä estää lokituksen toiminnan, esimerkiksi ylikuormittamalla.	Kohtalainen/ Denial of Service	• Riittävä lokienhallintakapasiteetti keskitetyssä lokienhallintajärjestelmässä. • Riittävä tietoliikennekapasiteetti.
Exploitable external service	3. osapuolen palvelussa on konfiguraatiovirhe tai haavoittuvuus. Esimerkiksi valtion yhteinen SAAS-palvelu, yhteistyökumppanin portaali/palvelu tai EU-palvelu.	Kohtalainen/ Elevation of Privilege	• Tietoturva-arviointi. • Haavoittuvuustiedotteiden seuranta. • Zero trust -arkkitehtuuri. • Käyttäjien koulutus ja käyttöpolitiikka.
Tietojen uudelleenohjaus	Hyökkääjä voi ohjata tietoja eri kanaviin, joista tulee eri tuloksia. Esimerkiksi virustorjunnan ohitus.	Kohtalainen/ Elevation of Privilege	• Sovellusarkkitehtuuri, joka pakottaa käsittelyreitin ja -tavat. Penetraatiotestaus.
Sovelluksen oikeustaso	Hyökkääjä voi hyödyntää sovelluksille myönnettyjä liiallisia oikeuksia.	Kohtalainen/ Elevation of Privilege	• Sovellusten ajaminen pienimmillä mahdollisilla oikeuksilla. • Käyttöjärjestelmän oletusasetusten koventaminen. • Sovellusten käyttöoikeuksien katselmointi. • Penetraatiotestaus.
Tietojen muokkaamisen jatkaminen	Hyökkääjä voi syöttää dataa, joka validoidaan ennen kuin hyökkääjän kontrolli datan sisältöön päättyy.	Kohtalainen/ Elevation of Privilege	• Muuttamattomat parametrit sovelluspinoissa. • Penetraatiotestaus.

Tietoturvaoletukset	Käyttäjät eivät tiedä, mitä tietoturvaoletuksia on tehty.	Kohtalainen/ Elevation of Privilege	• Tietoturvaoletusten ja -päätösten loki. • Tietoturva-arvioinnit. • Järjestelmän turvallisuuden dokumentointi.
Haitallinen sisältö	Sovelluksessa on käyttäjän generoimaa haitallista sisältöä, kuten linkkejä.	Kohtalainen/ Elevation of Privilege	• Kehitystyön aikainen tietoturvatestaus. • Tietoturvavaatimukset. • Penetraatiotestaus.
Sovelluksen oikeustaso	Hyökkääjä voi syöttää komentoja, jotka sovellus ajaa korkeammalla oikeustasolla.	Kohtalainen/ Elevation of Privilege	• Kehitystyön aikainen tietoturvatestaus. • Tietoturvavaatimukset. • Penetraatiotestaus.



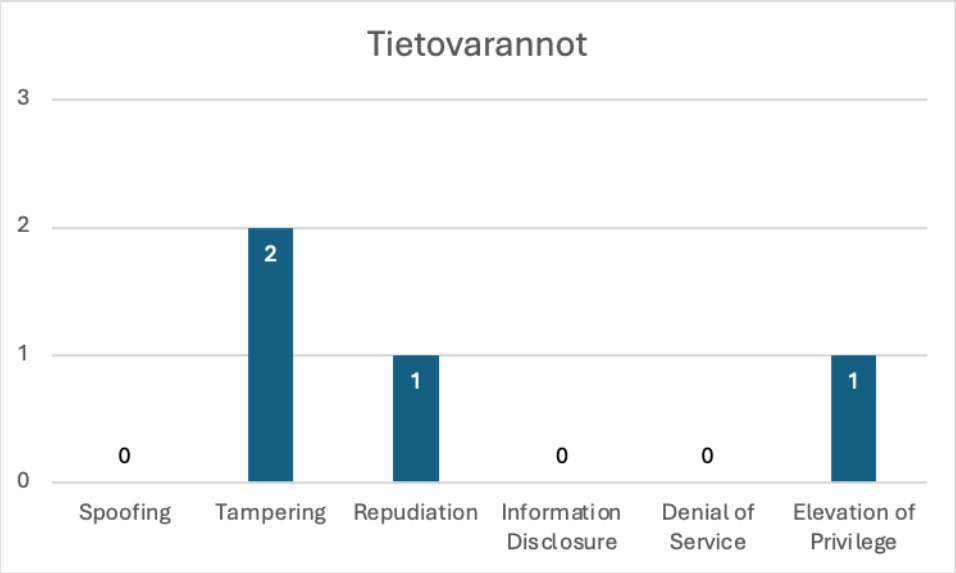
KUVIO 7 Sovellukset

Sovelluskomponentissa tunnistettiin yhteensä kaksikymmentäyksi uhkaa, ja melkein jokaisessa uhkaskenaarion lieventämistoimenpiteenä esiintyy penetraatiotestaus. Penetraatiotestauksella varmistetaan sovellusten turvallisuus, mutta muita merkittäviä lieventämistoimenpiteitä ovat sovellusarkkitehtuurin hyödyntäminen ja koodikatselmoinnit. Syväpuolustus on eräs huomioitava kokonaisuus tarkasteltaessa sovellusten turvallisuutta. Syväpuolustuksen voidaan ajatella koostuvan suojakerroksista, jotka ovat muun muassa turvalliset koodauskäytännöt, turvallisuustestaus ja sovelluspalomuurit. Suojakerrokset voidaan toteuttaa esimerkiksi ohjelmistoa luotaessa: turvallisuusvaatimusten määrittäminen, turvallisten koodaustekniikoiden käyttäminen, tietoturvatestaus, testaus usealla työkalulla ja uhkamallinnuksen toteuttaminen. Verkonsuojauksessa huomioitavia tekijöitä ovat valvonta, SIEM, IPS/IDS ja palomuurit. Viimeisenä tekijänä on fyysisen turvallisuuden varmistaminen: vierailijakäytänteet, videokamerat ja vartiointi. (Janca 2021, 9–10.)

6.1.7 Tietovarannot

TAULUKKO 24 Tietovarannot

Kortin skenaario	Kuvaus	Luokittelu	Lieventämistoimenpiteet
Pääsy tietovarantoon	Hyökkääjä voi kirjoittaa sovelluksen käyttämään tietovarantoon.	Kohtalainen/ Tampering	• ORM:it. • Asetusten katselmointi. • Koodikatselmoinnit. • Penetraatiotestaus.
Tietovarannon käyttöoikeudet	Hyökkääjä voi muokata tietoa tietovarannossa väljien käyttöoikeuksien takia.	Kohtalainen/ Tampering	• Sovelluksen käyttöoikeuksien rajoittaminen. • Write-once-arkkitehtuuri. Keskeisten tietojen allekirjoitus. • Koodikatselmoinnit. • Penetraatiotestaus.
Tietojen muuttaminen	Hyökkääjä voi muunnella viestiliikennettä, ohjelmistoja ja muita tiedostoja.	Vähäinen/ Repudiation	• Autentikoidut salatut tietoliikenneyhteydet. • Turvalliset salausasetukset (data at rest, data in transit). • Ohjelmistopakettien tietoturvan arviointi. • Keskeisten tietojen ja tiedostojen sähköiset allekirjoitukset.
Tietojen validointi	Tietojen käyttäjät eivät tiedä, miten ne on validoitu.	Kohtalainen/ Elevation of Privilege	• Validoinnista kertominen kriittisten tietojen yhteydessä.



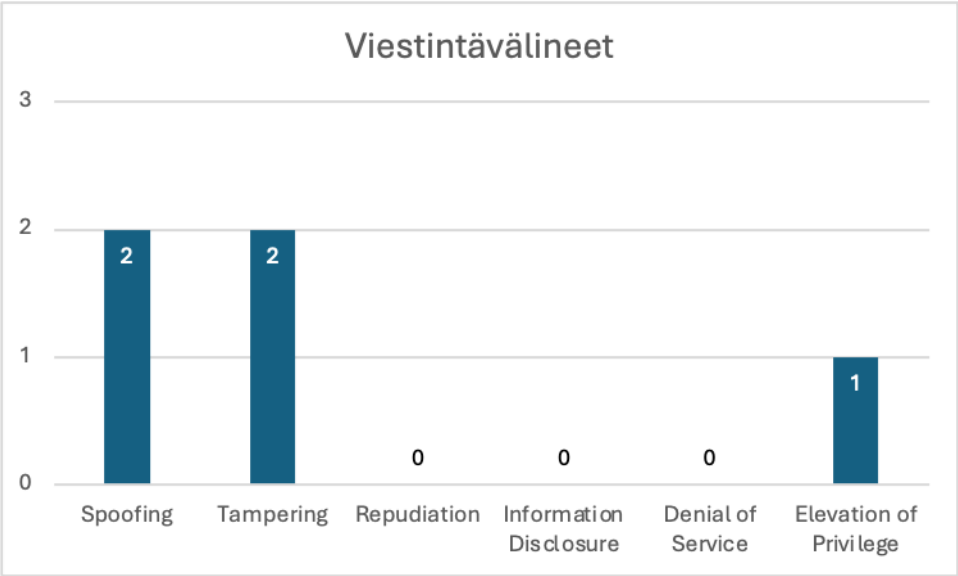
KUVIO 8 Tietovarannot

Tietovarannot komponentissa tunnistettiin yhteensä neljä uhkaa. Merkittävimpänä lieventämiskeinona ovat koodikatselmoinnit ja tyyppiturvallisen koodikielen käyttäminen. Tämä mahdollistaa sen, ettei hyökkääjä kykene hyödyntämään koodausvirheen aiheuttamaa haavoittuvuutta.

6.1.8 Viestintävälineet

TAULUKKO 25 Viestintävälineet

Kortin skenaario	Kuvaus	Luokittelu	Lieventämistoimenpiteet
Tietojen kalastelu	Kalastelusähköpostit. Tarkoitetaan pääosin kohdentamatonta toimintaa.	Kohtalainen/ Spoofing	• Henkilöstön koulutus. • Ilmoitusnäppäin sähköpostisovelluksessa. • Zero trust -arkkitehtuuri. • Automaattiset suodatukset. • Räätälöidyt suodatussäännöt. • Lokivalvonta. • MFA.
Internal Sprearphishing	Tilin kaapannut hyökkääjä lähettää organisaation omasta sähköpostiosoitteesta huijaussähköposteja.	Kohtalainen/ Spoofing	• Henkilöstön koulutus. • Ilmoitusnäppäin sähköpostisovelluksessa. • Zero trust -arkkitehtuuri. • Automaattiset suodatukset. • Räätälöidyt suodatussäännöt. • Lokivalvonta. • MFA. • User behaviour analytics.
Hallintatyökaluissa haittaohjelma	Hallintatyökaluun upotettu haittaohjelma vaarantaa ympäristön turvallisuuden.	Vähäinen/ Tampering	• Hallintatyökalujen turvallisuus on tarkistettu. • Verkkoyhteyksien rajoittaminen ja valvonta. • Toimittajat eivät saa ottaa itsenäisesti käyttöön erilaisia hallintatyökaluja. • Versiopäivitysten tarkistusten yhteydessä WhiteBox-testaus. • Hallintatyökaluja koskevat sopimusehdot.
Malicious Email Rules	Hyökkääjä luo sähköpostiin käsittelysääntöjä (esimerkiksi sähköpostien edelleen lähetys hyökkääjän sähköpostiin)	Kohtalainen/ Tampering	• Sähköpostijärjestelmän asetuskovennukset. • Käyttäjien lisäämien käsittelysääntöjen katselmointi.
Linkkien hyödyntäminen	Hyökkääjä voi validoitavissa olevan datan sijasta antaa linkin, joka ylittää luottamusrajan.	Kohtalainen/ Elevation of Privilege	• Zero trust -arkkitehtuuri. • Tietoliikenteen inspektointi. • Käyttäjille esitettävät varoitukset. • Käyttäjien ohjeistus ja koulutus.



KUVIO 9 Viestintävälineet

Viestintävälineet-komponentissa tunnistettiin yhteensä viisi uhkaa. Käyttäjien koulutuksen tärkeyttä lieventämiskeinona on syytä korostaa, koska sen avulla vältytään usealta mahdolliselta hyökkäykseltä. Käyttäjien toimenpiteiden lisäksi monimenetelmätunnistautumisen ja nollaluottamusarkkitehtuurin (*engl. zero-trust-architecture*) käyttäminen parantaa tietoturvaa. Minellan (2022, 455) mukaan nollaluottamusarkkitehtuurin huomioon ottaminen nykyajan turvallisuus- ja verkkoarkkitehdeille on tärkeä ja ajankohtainen asia.

6.1.9 Integraatioalustat ja palveluväylät

TAULUKKO 26 Integraatioalustat ja palveluväylät

Kortin skenaario	Kuvaus	Luokittelu	Lieventämistoimenpiteet
Hallintatyökaluissa haittaohjelma	Hallintatyökaluun upotettu haittaohjelma vaarantaa ympäristön turvallisuuden.	Kohtalainen/ Tampering	• Hallintatyökalujen turvallisuus on tarkistettu. • Verkkoyhteyksien rajoittaminen ja valvonta. • Toimittajat eivät saa ottaa itsenäisesti käyttöön erilaisia hallintatyökaluja. • Versiopäivitysten tarkistusten yhteydessä WhiteBox-testaus. • Hallintatyökaluja koskevat sopimusehdot.
Viestien muokkaus	Hyökkääjä voi manipuloida järjestelmien välisiä tai sisäisiä viestejä.	Kohtalainen/ Tampering	• Aikaleimat ja sekvenssinumerot. • Allekirjoituksen sisältävät protokollat. • Sovelluksen tilan validointi. • Koodikatselmoinnit. • Penetraatiotestaus.

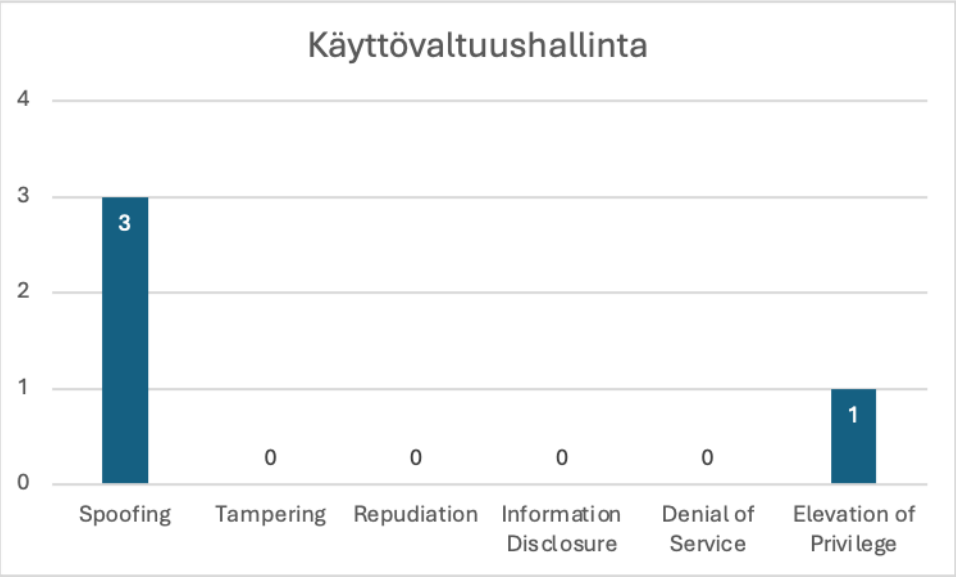
Integraatioalustat ja palveluväylät -komponentissa tunnistettiin kaksi uhkaa ja näiden uhkaskenaarioiden oleellisena tekijänä on varmistaa käytettävien hallin-

tatyökalujen turvallisuus erilaisilla testaustoimenpiteillä muun muassa White-Box-testauksella. Viestin muokkauksen estämisen lieventämiskeinoina ovat muun muassa aikaleimojen käyttäminen, validointi, koodikatselmoinnit ja penetraatiotestaus.

6.1.10 Käyttövaltuushallinta

TAULUKKO 27 Käyttövaltuushallinta

Kortin skenaario	Kuvaus	Luokittelu	Lieventämistoimenpiteet
Password spray	Hyökkääjä kokeilee yleisiä salasanoja meidän verkossamme olevaan tai 3. osapuolen palveluun.	Kohtalainen/ Spoofing	• Toteutettu salasanapolitiikka. • Salasanattomuus. • Lokivalvonta. • Avointen lähteiden tiedustelu (OSINT). • User and Entity Behavior Analytics.
Internal Password Spray	Hyökkääjä kokeilee yleisiä tunnettuja salasanayhdistelmiä organisaation verkossa.	Kohtalainen/ Spoofing	• Toteutettu salasanapolitiikka. • Salasanattomuus. • Lokivalvonta. • Avointen lähteiden tiedustelu (OSINT). • User and Entity Behavior Analytics.
Credential Stuffing	Hyökkääjä käyttää hyväkseen ympäristöstä löytyneitä Active Directory -tunnisteita sisältäviä tiedostoja.	Kohtalainen/ Spoofing	• Salasanattomuus • Salaisuuksienhallinta ohjelmistot. • Haavoittuvuusskannaukset (erityisesti konfiguraatiot).
IT compromised with shared domain trust	Luottosuhteen kautta tapahtuva murtautuminen	Kohtalainen/ Elevation of Privilege	• Azure AD:n tehostettu valvonta (UEBA). • Lokivalvonta. • Riittävä penetraatiotestaus.



KUVIO 10 Käyttövaltuushallinta

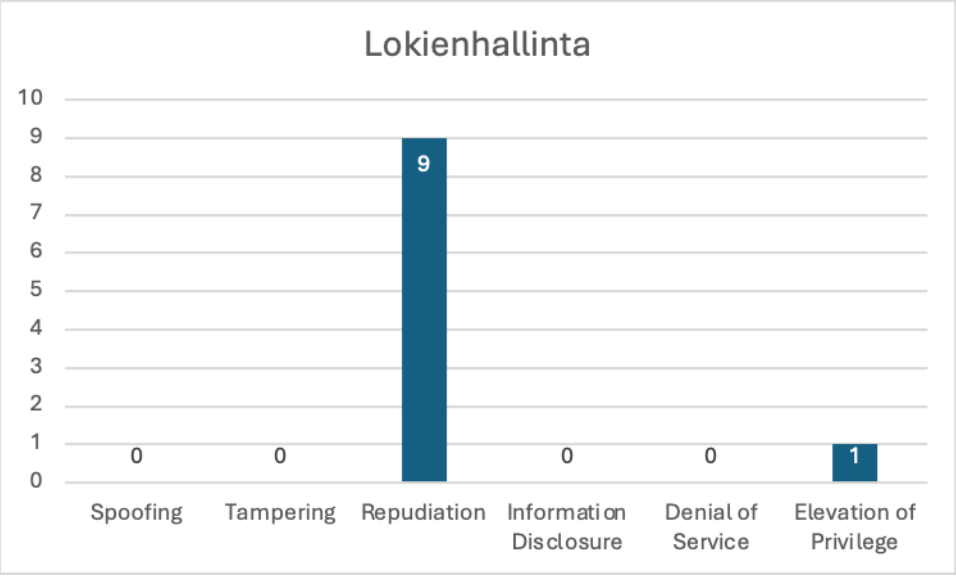
Käyttövaltuushallinta-komponentissa tunnistettiin neljä uhkaa. Salasanojen spaying-hyökkäyksissä oleellisena tekijänä on hyvin toteutettu salasanapolitiikka, mutta huomioitavaa on se, että salasanattomuudella estetään heikkojen salasanojen käyttäminen. Mikäli hyökkääjä liikkuu kohdejärjestelmässä, niin UEBA:n avulla normaalista poikkeava käyttäminen kyetään havaitsemaan. Salasanattomuudessa on useita hyötyjä, koska vahvoja salasanoja on vaikea muistaa ja samaa salasanaa käytetään useassa paikassa. Lisäksi käyttäjät voivat paljastaa vahingossa salasanansa tietojenkalasteluhyökkäyksessä. Microsoft Authenticator -sovelluksella on mahdollista kirjautua jokaiselle Azure AD -tilille ilman salasanaa. Tekniikkana Microsoft Authenticator käyttää avainpohjaista todennusta, joka on liitetty laitteeseen. (Peiris, Pillai & Kudrati 2022, 211.) 1980-luvun puolivälistä lähtien on tutkittu, millaisia salasanoja ihmiset käyttävät. Tulokset ovat olleet huonoja, koska ihmiset käyttävät puolisoidensa nimiä, yksittäisiä kirjaimia tai painavat rivinvaihtoa antaen tyhjän merkkijonon. (Anderson 2020, 102.)

6.1.11 Lokienhallinta

TAULUKKO 28 Lokienhallinta

Kortin skenaario	Kuvaus	Luokittelu	Lieventämistoimenpiteet
Haittakoodin lähettäminen lokeissa	Hyökkääjä voi ladata haittakoodia lokienhallintajärjestelmään.	Kohtalainen/ Repudiation	• Lokien toimitustavan rajoittaminen. • Lokitietojen siivoaminen.
Lokien manipulointi	Hyökkääjä voi manipuloida lokienhallintajärjestelmään lähetettävää lokia saatuaan lähettävän koneen hallintaansa.	Kohtalainen/ Repudiation	• Lähettävien ohjelmistojen koventaminen. • Lokien lähetyksen viiveettömyys. • Tehokkaat SOC-prosessit.
Lokitietojen tiedustelu	Hyökkääjä voi lukea turvallisuutta vaarantavaa lokia matalilla käyttöoikeuksilla.	Kohtalainen/ Repudiation	• Lokitietojen käyttöoikeuksien rajoittaminen. • Paikallisten lokitietojen määrän rajoittaminen. • Tiedostojen käsittelyn valvontasäännöt. • Lokitietojen sisällön katselmointi ja rajaaminen.
Lokitietojen ajan väärentäminen	Hyökkääjä voi luoda lokimerkinnän ilman aikaleimaa tai väärentää aikaleiman.	Kohtalainen/ Repudiation	• Aikaleimojen loogisuustarkastukset. • Aikalähteen kirjaaminen lokiin. • Luotettujen aikalähteiden käyttö. • Allekirjoitettujen aikalähteiden käyttö. • Lokienhallinta lisää puuttuvan ajan.
Liikaa lokitietoja	Hyökkääjä generoi niin paljon lokia, että lokitiedosto aloitetaan uudelleen tai lokitila täyttyy.	Kohtalainen/ Repudiation	• Keskitetyn lokienhallinnan käyttö. • Levytilan käytön valvonta. • Riittävä resursointi.
Lokimerkintöjen hävittäminen	Hyökkääjä hävittää lokimerkintöjä.	Kohtalainen/ Repudiation	• Keskitetty lokienhallinta. • Lokien käyttäjille ei tarjota

			muokkaustoiminnallisuuksia. • Palvelinympäristön valvonta. • Lokien varmuuskopiointi. • Write-only lokienhallintajärjestelmä datadiodin takana.
Lähettäjän väärentäminen	Hyökkääjä lähettää lokitietoja eri palvelimen tai järjestelmän identiteetillä.	Kohtalainen/ Repudiation	• Suojatuilla avaimilla allekirjoitetut lokit.
Lokitietojen muokkaaminen	Hyökkääjä voi muokata lokitietoja.	Kohtalainen/ Repudiation	• Keskitetty lokienhallintajärjestelmä. • Lokien käyttäjille ei tarjota muokkaustoiminnallisuuksia. • Lokienhallintajärjestelmän koventaminen ja valvonta. • Lokitietojen allekirjoittaminen.
Puutteelliset lokitiedot	Hyökkääjä väittää, ettei ole tehnyt jotain toimenpidettä, eikä tätä voida osoittaa vääräksi.	Kohtalainen/ Repudiation	• Lokien sisällön riittävyyden katselmointi. • Laadukkaat aikaleimat lokeissa. • Laadukas jäljitysketju lokeista käyttäjän identiteettiin. • Lokitietojen allekirjoittaminen.
Data historian compromise	Tietomurto lokienhallinnassa mahdollistaa pääsyn edelleen.	Kohtalainen/ Elevation of Privilege	• Kredentiaaleja ei lokiteta. • Palvelinympäristön EDR. • Lokittavien palvelujen vahva tunnistautuminen. • Lokienhallinnan vahva tunnistautuminen. • Lokienhallinnan tietoturvapäivitykset. • Tietoliikenneavausten tarkistaminen.



KUVIO 11 Lokienhallinta

Lokienhallinta-komponentissa tunnistettiin kymmenen uhkaa. Huomioitavaa puolustajan näkökulmasta on se, että lokienhallinnan tietoturvan tärkeyttä täytyy korostaa, koska lokienhallinnalla kyetään selvittämään, mitä on tapahtunut. Skenaariot lokeihin hyökkäämisestä ovat mahdollisia, ja tämän takia lokien tie-

toturva täytyy pitää ajan tasalla. Yleisesti ottaen lokien käyttämisessä ja suojaamisessa tulee huomioida, että lokitiedostot ja kirjausketjut on suojattu ja varmuuskopioitu, ainoastaan valtuutetuilla henkilöillä on pääsy lokeihin, lokeja seurataan järjestelmällisesti, lokit tulee tallentaa salatussa muodossa, lokit on tallennettu suojatulle palvelimelle tai muulle suojatulle alustalle ja lokitietojen hävittämisessä noudatetaan yleisesti hyväksyttyjä periaatteita. (Janca 2021, 102–103.) Huomioitavaa on se, että lokeihin tallentamisessa tulisi välttää muun muassa henkilötunnuksia, luottokorttinumeroita, valtuutustietoja, salasanoja ja käyttäjien välisen viestiliikenteen sisältöä, koska liian tarkka tapahtumaseuranta yhdistettynä henkilön identiteettiin voi rikkoa yksilön tietosuojaa (Kyberturvallisuuskeskus 2016b, 5).

6.1.12 Nimipalvelut

TAULUKKO 29 Nimipalvelut

Kortin skenaario	Kuvaus	Luokittelu	Lieventämistoimenpiteet
Hallintatyökaluissa haittaohjelma	Hallintatyökaluun upotettu haittaohjelma vaarantaa ympäristön turvallisuuden.	Kohtalainen/ Tampering	• Hallintatyökalujen turvallisuus on tarkistettu. • Verkkoyhteyksien rajoittaminen ja valvonta. • Toimittajat eivät saa ottaa itsenäisesti käyttöön erilaisia hallintatyökaluja. • Versiopäivitysten tarkistusten yhteydessä WhiteBox-testaus. • Hallintatyökaluja koskevat sopimusehdot.
DNS-tietojen väärentäminen	Nimipalvelutietoja voi väärentää.	Kohtalainen/ Repudiation	• DNSSEC. • Palvelinympäristöjen sisäiset host-tiedostot.

Nimipalvelut-komponentissa tunnistettiin kaksi uhkaa. Hallintatyökaluissa oleva haittaohjelma liittyy useampaan komponenttiin, muun muassa integraatioalustoihin ja palveluväyliin, minkä takia hallintatyökalujen tietoturvan varmistaminen on merkittävä tekijä tietoturvan varmistamisen kannalta. Nimipalveluiden väärentämisen lieventämiskeinoina ovat muun muassa DNSSEC ja palvelinympäristöjen sisäiset isäntätiedostot.

6.2 Luotettavuuden arviointi ja tutkimuksen eettisyys

Tutkimuksen luotettavuuden arvioinnin tapoja on useita, ja tässä tutkimuksessa luotettavuutta arvioitiin Tuomen ja Sarajärven (2018, 163–164) listan mukaisesti. Siihen kuuluvat tutkimuksen kohde ja tarkoitus, aineiston keruu, tiedonantajat ja tutkijan välinen suhde, tutkimuksen kesto, aineiston analyysi, luotettavuus ja tutkimuksen raportointi.

Tutkimuksen tarkoituksena oli selvittää keskeiset oikeushallinnon tietotekniikkaympäristön riskit. Tutkimuksen päätutkimuskysymyksen tueksi alatutkimuskysymyksissä selvitettiin kyseisten riskien hallintakeinot ja se, miten riskejä voidaan kartoittaa uhkamallinnuksen avulla. Päätutkimuskysymykseen esitettiin vastaus luvun 6.1 alaluvuissa, jossa esitettiin oikeushallinnon tietotekniikkaympäristön keskeisimmät riskit. Luotettavuuden parantamiseksi lukijalle tulisi esittää myös ne riskit, jotka arvioitiin merkittäviksi ja kriittisiksi. Uhkamallinnusprosessissa havaittiin riskejä, jotka arvioitiin merkittäviksi ja kriittisiksi, mutta tietoturvasyistä niitä ei esitetä tässä tutkimuksessa. Keskeisimpien riskien hallintakeinot esitettiin luvun 6.1 alaluvuissa. Kyseiset hallintakeinot kuvattiin mahdollisimman yleisellä tasolla suojellen kohdeorganisaation tietoturvaa ja lisäten samalla tulosten hyödynnettävyyttä muissa organisaatioissa. Tutkimuksessa esitettiin yksi tapa tehdä uhkamallinnusta, ja se oli ohjelmistokeskeinen uhkamallinnus Elevation of Privilege- ja Backdoors & Breaches -korttisarjojen avulla. Havaintoina tästä saatiin, että korttisarjojen käyttäminen soveltuu organisaatiokohtaiseen uhkamallinnukseen.

Aineiston kerääminen toteutettiin työpajatyöskentelyllä kohdeorganisaation tietoturvapäällikön ja it-erityisasiantuntijan kanssa. Kummallakin henkilöllä on usean vuoden työkokemus tietoturvasta, ja he ovat työskennelleet kohdeorganisaatiossa usean vuoden ajan. Huomioitavaa on se, että työskentely pienessä ryhmässä saattaa ohjata näkemyksiä ja osa asioista saattaa jäädä huomioimatta. Kuitenkin työpajatyöskentelyyn osallistuvilla henkilöillä oli kokemusta uhkamallinnuksen tekemisestä. Tutkimuksessa pyrittiin valitsemaan lähteiksi vertaisarvioituja tutkimusartikkeleita ja ajantasaisia teoksia. Artikkeleita valittiin Google Scholarin, IEEE Exploren ja Sciencedirectin avulla.

Tutkimus alkoi syyskuussa 2023, jolloin tutkija teki alustavan tutkimussuunnitelman ja aloitti aineiston keräämisen. Aineiston keräämisen aikana aloitettiin työpajatyöskentely, joka alkoi 19.9.2023, ja viimeinen työpaja järjestettiin 9.1.2024. Työpajoja oli yhteensä yksitoista, ja ne kestivät yhdestä tunnista seitsemään tuntiin. Työpajatyöskentelyyn käytettiin aikaa yhteensä noin kaksikymmentäyhdeksän tuntia. Työpajat järjestettiin virtuaalityöpajoina, jotka olivat toimiva ratkaisu. Kasvokkain pidettäviä työpajoja ei järjestetty.

Tutkimuksen tietoteoreettisena lähtökohtana käytettiin konstruktiivista tutkimusotetta, mikä osoittautui tämän tutkimuksen kannalta oivalliseksi valinnaksi. Tutkimus rakennettiin Lukan (2000) konstruktiivisen tutkimusotteen vaiheiden mukaisesti, ja ne esitettiin luvussa 2.2. Tutkimuksessa pyrittiin aineiston analyysivaihe kuvaamaan mahdollisimman selkeästi, mikä auttaa myös muita organisaatioita huomioimaan erilaisia riskienhallintatoimenpiteitä. Tutkimuksen tärkein osa on raportointivaihe, joka esitettiin luvun 6.1 alaluvuissa. Raportointivaiheessa päädyttiin käyttämään taulukointia, koska se osoittautui selkeimmäksi vaihtoehdoksi. Tutkija ei käyttänyt taulukoita apuna ensimmäisessä vaiheessa, mutta raportti muodostui epäselväksi ja tämän takia päädyttiin käyttämään taulukoita. Taulukoiden lopuksi nostettiin muutama merkittävä havainto esille havaintojen selkiyttämiseksi. Kokonaisuudessaan luotettavuuden parantamiseksi myös merkittävät ja kriittiset riskit sekä niiden hallintakei-

not esitettäisiin tutkimuksessa. Tietoturvasyistä merkittäviä ja kriittisiä riskejä ei voida esittää tässä tutkimuksessa kohdeorganisaation suojelemiseksi. Luotettavuuden parantamiseksi tutkija on esittänyt liitteessä kaksi esimerkin kriittisen uhkan käsittelystä.

Eettisesti kestävälle tutkimukselle voidaan asettaa normatiiviset kriteerit, jotka on jaettu kolmeen luokkaan. Ensimmäisessä osassa käsitellään kestävää tieteellistä käytäntöä, johon kuuluvat rehellisyys, huolellisuus ja tarkkuus. Toinen osa käsittelee hyvän käytännön loukkauksia, jotka jaetaan piittaamattomuuteen ja vilppiin tieteellisessä tutkimuksessa. Viimeinen osa käsittelee säännöksiä tutkijan oikeusturvasta. (Hirvonen 2006, 31–32.) Tutkimuksessa pyrittiin noudattamaan normatiivisia kriteereitä, ja tavoitteena oli tehdä eettisesti kestävä tutkimus, josta olisi hyötyä myös muille organisaatioille, koska vastaavanlaista tutkimusta tutkija ei löytänyt.

Tutkimusaineiston säilyttämiseen liittyviä kysymyksiä tulisi tarkastella huolellisesti. Aineistosta tulisi säilyttää ainoastaan ne osat, jotka ovat tutkimuksen kannalta oleellisia. (Mäkinen 2006, 120.) Tutkimuksesta tehtiin kohdeorganisaatiolle raportti, joka on sen hallussa. Varsinaisesta raportin säilyttämisestä vastaa kohdeorganisaatio.

6.3 Jatkotutkimusaihe

Tutkimusta tehdessä tutkijalle nousi esille keskeinen jatkotutkimusaihe kehittyneiden uhkatoimijoiden tutkimisesta. Kehittyneet uhkatoimijat tunnetaan nimellä APT-ryhmät, ja ne ovat suorittaneet kyberhyökkäyksiä valtiollisiin ja muihin toimijoihin. APT-ryhmät ovat kehittyneitä ja joustavia, ja niillä on erinomainen kyky käyttää hyökkäysvektoreita ja sisääntulopisteitä. Nopeimmillaan APT-ryhmät murtautuvat tietoverkkoon minuuteissa ja välttävät paljastumista jopa vuosia. APT-ryhmien toiminnasta on kehitetty erilaisia malleja, jotka yleisimmin kuvataan ketjumallilla. (Lehto 2022, 122.)

Jatkotutkimus perustuu hyökkääjäkeskeiseen uhkamallinnukseen, ja esimerkkinä voidaan käyttää esimerkiksi Mitre ATT&CK -mallia kuvaamaan APT-ryhmien toimintaa. Toiminta muodostuu seuraavista vaiheista:

1. Tiedustelu (*engl. Reconnaissance*)
2. Resurssien kehittäminen (*engl. Resource Development*)
3. Jalansija tai pääsy kohdejärjestelmään (*engl. Initial Access*)
4. Haittaohjelman asennus ja suorittaminen (*engl. Execution*)
5. Vakiinnutetaan jalansija (*engl. Persistence*)
6. Käyttöoikeuksien korottaminen (*engl. Privilege Escalation*)
7. Puolustuksen välttely (*engl. Defence Evasion*)
8. Tunnistetietojen käyttäminen (*engl. Credential Access*)
9. Havainnointi (*engl. Discovery*)
10. Sivuttaisliike (*engl. Lateral Movement*)
11. Tietojen kerääminen (*engl. Collection*)

12. Haittaohjelman komento- ja kontrollointiyhteyden käyttö (*engl. Command and Control*)
13. Tietojen suodattaminen salausta ja/tai lähettämistä ja/tai tuhoamista varten (*engl. Exfiltration*)
14. Hyökkäyksestä aiheutuneet vaikutukset (*engl. Impact*)

Mitre ATT&CK-verkkosivuilla on kerätty jokaisesta toiminnan vaiheesta toiminnan kuvaus, hyökkääjän käyttämät tekniikat ja lieventämiskeinot. (Mitre ATT&CK 2024a.)

APT-ryhmiä on monia, mutta tutkittava ryhmä voisi olla APT29, jonka on katsottu kuuluvan Venäjän ulkomaan tiedustelupalveluun. APT29 on toiminut ainakin vuodesta 2008, ja se on kohdistanut hyökkäyksiä Euroopan unionin ja Naton jäsenvaltioiden hallitusten tietoverkkoihin ja tutkimuslaitoksiin. (Mitre ATT&CK 2024b.) Tutkimuksesta saatuja havaintoja voidaan vertailla tämän tutkimuksen havaintoihin, ja samalla saataisiin tietoa, kumpi uhkamallinnus tyyppi soveltuu paremmin valtiollisen toimijan tarkasteluun.

7 YHTEENVETO

Tutkimuksen kohteena oli oikeushallinnon tietotekniikkaympäristö, joka uhkamallinnettiin ohjelmistokeskeisellä uhkamallinnuksella. Uhkamallinnuksessa käytettiin Elevation of Privilege- ja Backdoors & Breaches -korttisarjoja. Tutkimuksessa on esitetty oikeushallinnon tietotekniikkaympäristön keskeiset riskit ja niiden hallintatoimenpiteet. Uhkamallinnuksen avulla tunnistettiin erilaisia uhkia, jotka luokiteltiin vähäisiksi, kohtalaisiksi, merkittäviksi ja kriittisiksi. Luokittelun jälkeen voidaan keskustella riskeistä, jotka liittyvät Oikeusrekisterikeskuksen riskienhallintaan. Uhkamallinnus on oleellinen osa riskienhallintaa, koska riskien tunnistamisen jälkeen osa riskeistä siirtyy viraston riskienhallintaprosessiin jatkokäsittelyä varten.

Saatujen havaintojen perusteella käyttäjien tietoturvakoulutuksen tärkeyttä on kriittistä korostaa. Oikeushallinnossa on noin 10 000 työntekijää, ja päätelaitteita on yli 10 000 kappaletta ottaen huomioon muun muassa työpuhelimet. Työntekijöiden järjestelmällisellä ja säännöllisellä koulutuksella voidaan ennaltaehkäistä mahdolliset tietoturvapoikkeamat. Tietoturvasta huolehtiminen kuuluu jokaisen organisaation työntekijälle, ja mahdollisista virheistä tulee ilmoittaa organisaation tietoturvatiimeille tai tietoturvasta vastaavalle henkilölle. Työntekijöiden näkökulmasta tarkasteltuna salasanattomuus parantaa merkittävästi tietoturvaa, koska vahvoja salasanoja on vaikeaa muistaa ja samaa salasanaa mahdollisesti käytetään useassa paikassa.

Turvallisen ohjelmointikoodin kirjoittaminen ja penetraatiotestaus on merkittävässä asemassa. Tutkimuksessa käsiteltiin erilaisia hyökkäystekniikoita haavoittuvaa ohjelmistokoodia vastaan, mikä voidaan ehkäistä edellä mainituilla tekniikoilla. Tyypillisimpien uhkien ja tutkimuksessa saatujen havaintojen perusteella EDR ja UEBA parantavat tietoturvaa merkittävästi. EDR ja UEBA ovat oivallisia työkaluja muun muassa kehittyneiden uhkatoimijoiden torjuntaan, koska tietomurrot voivat olla nykyään huomaamattomia ja seuraukset voivat olla vakavia. Nykyajan tietomurrot voivat olla huomaamattomia, ja normaalista poikkeavan käyttäytymisen havaitseminen ja reaaliaikainen valvonta on kriittistä. DLP-työkalun käyttäminen parantaa merkittävästi tietoturvaa, koska se tarjoaa valvontaa ja suojausta arkaluonteisen tiedon suojaamiseksi.

Yksi tapa tehdä uhkamallinnusta on pelikorttien käyttäminen. Internetistä on saatavilla ilmaiseksi lukuisia korttisarjoja, joiden avulla voidaan tehdä uhkamallinnusta, ja kehityksessä oleviin tietojärjestelmiin ohjelmistokeskeinen uhkamallinnus on oivallinen valinta. Tämä tutkimus keskittyi ohjelmistokeskeiseen uhkamallinnukseen, mutta toinen mahdollinen tapa tehdä uhkamallinnusta on hyökkääjäkeskeinen uhkamallinnus. Mitre ATT&CK-verkkosivuilla on lukuisia esimerkkejä kehittyneiden uhkatoimijoiden hyökkäystekniikoista ja -strategioista. Huomioitavaa on myös se, että kehittyneet uhkatoimijat ovat vaikuttaneet globaalilla tasolla valtiollisiin toimijoihin, minkä takia tulevaisuudessa valtion virastojen on syytä kiinnittää huomiota myös tähän näkökulmaan.

Wallsin, McMullenin, Heiserin ja Gopalin (2023) havainnot perinteisen riskienhallinnan toimimattomuudesta on suhteellisen radikaali näkemys verrattuna ISO/IEC 27005:2022-standardiin. Aihe herättää uhkamallinnuksen näkökulmasta keskustelua, koska uhkamallinnus liittyy kiinteästi riskienhallintaan. Tutkijoiden näkemykseen vaaditaan huomattavan paljon tieteellistä näyttöä siitä, miten konkreettisesti riskienhallinta tulisi toteuttaa. Tieteellistä näyttöä kaivataankin myös kustannustehokkuuteen, koska tutkijat mainitsevat, että organisaatioiden tulee investoida uhkientiedusteluun, jonka perusteella tehdään investointeja. Uudet tietoturvaratkaisut tuottavat organisaatioille lisäkustannuksia, mutta niin tuottaa myös uhkien tiedustelu. On totta, että ennustettavuus on vaikeaa muuttuvassa digitaalisessa toimintaympäristössä ja uudet tietoturvaratkaisut saattavat olla jo vanhentuneita, kun ratkaisut saadaan käyttöön. Huomioitavaa on se, että uhkien tiedustelun perusteella saadut havainnot saattavat ohjata päätöksentekoa harhaan, mikä aiheuttaa suuria tappioita organisaatiolle.

LÄHTEET

Agile Stationery (2024). *Elevation of Privilege (EoP) Threat Modeling Game.* Viitattu: 6.3.2024. https://agilestationery.com/pages/elevation-of-privilege-eop-threat-modeling-card-game.

AlQahtani, A. A. & El-Alfy, E-L. S. (2015). *Anonymous connections based on onion routing: A review and a visualization tool*. 6th International Conference on Ambient Systems, Networks and Technologies, ANT 2015. Procedia Computer Science 52 (2015) 121–128.

Anderson, R. (2020). *Security Engineering, A Guide to Building Dependable Distributed Systems. Third Edition.* Indianapolis: Wiley Publishing, Inc.

Anderson, R. (2008). *Security Engineering, A Guide to Building Dependable Distributed Systems. Second Edition.* Indianapolis: Wiley Publishing, Inc.

Apache. (2011). *Apache HTTPD Security Advisory*. Viitattu: 15.12.2023. https://httpd.apache.org/security/CVE-2011-3192.txt.

Ashoor, A. S. & Gore, S. D. (2011). *Intrusion Detection System (IDS): Case Study*. International Journal of Scientific & Engineering Research Volume 2, Issue 7, July-2011. ISSN 2229-5518.

Auger, G., Scott, J., Helmus, J. & Nguyen, K. (2021). *Cybersecurity Career Master Plan*. Birmingham: Packt Publishing Ltd.

Azahari, A. & Balzarotti, D. (2024). *On the inadequacy of open-source application logs for digital forensics*. Forensic Science International: Digital Investigation 49 (2024) 301750. Elsevier Ltd.

Azizi, Y., Azizi, M. & Elboukhari, M. (2019). *Log files Analysis Using MapReduce to improve Security*. Second international Conference on Intelligent Computing in Data Sciences (ICDS 2018). Procedia Computer Science 148 (2019) 37–44.

Bajtoš, T., Sokol, P. & Kurimský, F. (2024). *Processing of IDS alerts in multi-step attacks*. Software Impacts 19 (2024) 100622. Elsevier B.V.

Bak, S., Manamcheri, K., Mitra, S. & Caccamo, M. (2011). *Sandboxing Controllers for Cyber-Physical Systems*. ResearchGate. DOI:10.1109/ICCPS.2011.25.

Ball, C. J. (2022). *Hacking APIS, Breaking Web Appilication programming Interfaces*. San Francisco: No Starch Press.

Barker, E. & Barker, W. C. (2018). *Recommendation for Key Management – Part 2: Best Practices for Key Management Organizations*. National Institute of Standards and Technology Draft (2nd) NIST Special Publication (SP) 800-57 Part 2 Rev. 1.

Baumeister, T. (2011). *Adapting PKI for the Smart Grid*. ResearchGate. DOI: 10.1109/SmartGridComm.2011.6102327.

Beyer, J. (2020). *Adam Shostack on Threat Modeling*. IEEE Computer Society. ISSN 0740-7459.

Black Hills (2024). *Backdoors & Breaches.* Viitattu: 6.3.2024. https://www.blackhillsinfosec.com/projects/backdoorsandbreaches/.

Chen, T., Liu, Z. & Su, H. (2024). *Tampering attack detection for remote interval observer*. Journal of the Franklin Institute 361 (2024) 71–84. Elsevier Inc.

Chen, J., Xiao, H., Zheng, Y., Hassan, M. M., Ianni, M., Guzzo, A. & Fortino, G. (2023). *DKSM: A Decentralized Kerberos Secure Service-Management Protocol for Internet of Things*. Internet of Things 23 (2023) 100871.

Chen, T., Zheng, C., Zhu, T., Xiong, C., Ying, J., Yuan, Q., Cheng, W. & Lv, M. (2023). *System-level data management for endpoint advanced persistent threat detection: Issues, challenges and trends*. Computers & Security Volume 135, December 2023, 103485. Elsevier Ltd.

Crespo-Martínez, I. S., Campazas-Vega, A., Guerrero-Higueras, A. M., Riego-DelCastillo, V., Álvarez-Aparicio, C. & Fernández-Llamas, C. (2023). *SQL injection attack detection in network flow data*. Computers & Security 127 (2023) 103093. Elsevier Ltd.

DeLaughter (2023). *Redistributing the Costs of Volumetric Denial-of-Service Mitigation*. Massachusetts: Computer Science and Artificial Intelligence Lab.

Diogenes, Y. & Ozkayal, E. (2022). *Cybersecurity – Attack and Defense Strategies. Third Edition*. Birmingham: Packt Publishing Ltd.

Doshi, H. (2023). *CISA – Certified Information Systems Auditor Study Guide, Second Edition*. Birmingham: Packt Publishing Ltd.

Du, R., Safavi-Naini, R. & Susilo, W. (2003). *Web filtering text classification*. The 11th IEEE International Conference on Networks.

Enoka, S. (2023). *Cybersecurity For Small Networks, A No-Nonsense Guide For The Reasonably Paranoid*. San Francisco, CA: No Starch Press.

Fang, W., Chen, W., Zhang, W., Pei, J., Gao, W. & Wang G. (2020). *Digital signature scheme for information non-repudiation in blockchain: a state of the art review*. EURASIP Journal on Wireless Communications and Networking (2020) 2020:56.

Faraj, S. & Sambamurthy, V. (2006). *Leadership of Information Systems Development Projects*. IEEE Transactions on Engineering Management, 53(2), 238–249.

F-Secure. (2023). *Kolme yleistä kyberuhkaa vuonna 2023*. Viitattu: 3.5.2024. https://www.f-secure.com/fi/articles/3-trending-cyber-threats-in-2023.

Guo, R., Li, W., Liu, B., Haot, S. & Zhang, J. (2020). *CDN Judo: Breaking the CDN DoS Protection with Itself*. Network and Distributed Systems Security (NDSS) Symposium 2020 23-26 February 2020, San Diego, CA, USA. ISBN 1-891562-61-4.

Haber M. J., & Hibbert B. (2018). *Privileged Attack Vectors: Building Effective Cyber-Defense Strategies to Protect Organizations.* ISBN-13 (pbk): 978-1-4842-3047-3

Hajrić, A., Smaka, T., Baraković, S. & Husić, J. B. (2020). *Methods, Methodologies, and Tools for Threat Modeling with Case Study*. Telfor Journal, Vol. 12, No. 1, 2020.

Hauser, F., Häberle, M. & Menth, M. (2020). *P4-IPsec: Site-to-Site and Host-to-Site VPN With IPsec in P4-Based SDN*. Chair of Communication Networks, University of Tübingen, 72076 Tübingen, Germany. IEEE Access. Digital Object Identifier 10.1109/ACCESS.2020.3012738.

Hassan, W. U., Bates, A. & Marino, D. (2020). *Tactical Provenance Analysis for Endpoint Detection and Response Systems*. IEEE Symposium on Security and Privacy.

Heikkinen, V. & Söderqvist, M. (2005). *Konstruktiivinen tutkimusote majoitus- ja ravitsemisalan ilmiöiden analysointivälineenä*. Ammattikasvatuksen aikakauskirja, 7(2), 37–45.

Hirvonen, A. (2006). *Eettisesti hyvä tutkimus*. Teoksessa: J. Hallamaa, V. Launis, S. Lötjönen & I. Sorvali (toim). *Etiikka ihmistieteille*. Helsinki: Hakapaino Oy. 31–49.

Huang, J., Li, Y., Zhang, J. & Dai, R. (2019). *UChecker: Automatically Detecting PHP-Based Unrestricted File Upload Vulnerabilities.* 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). DOI: 10.1109/DSN.2019.00064.

Huang, Y., Chen, F. & Qu, P. (2009). *Research on Digital Signature Based on Digital Certificate.* Proceedings of 14th Youth Conference on Communication.

Hyppönen, M. (2022). *If It's Smart, It's Vulnerable*. Hoboken: John Wiley & Sons, Inc.

ISO/IEC 27005:2022:fi. *Tietoturvallisuus, kyberturvallisuus ja tietosuoja. Ohjeita tietoturvariskien hallintaan. Information security, cybersecurity and privacy protection. Guidance on managing information security risks*. Suomen Standardoimisliitto.

Janca, T. (2021). *Alice & Bob Learn Application Security*. Indianapolis: Wiley Publishing, Inc.

Jayaprakash, R. & Seethalakshmi, V. (2021). *Mitigation of Malicious Flooding in Software Defined Networks Using Dynamic Access Control List*. Wireless Personal Communications https://doi.org/10.1007/s11277-021-08626-6.

Jin, H., Liu, B., Du, Y. & Zou, D. (2018). *BoundShield: Comprehensive Mitigation for Memory Disclosure Attacks via Secret Region Isolation*. National 973 Fundamental Basic Research Program. IEEE Access.

Jin, H., Xiang, G., Zou, D., Zhao, F., Li, M. & Yu, C. (2010). *A guest-transparent file integrity monitoring method in virtualization environment*. Computers and Mathematics with Applications 60 (2010) 256–266.

Johansen, G. (2022). *Digital Forensics and Incident Response, Incident response tools and techniques for effective cyber threat response*. Third Edition. Birmingham: Packt Publishing Ltd.

Kaffah, F. M., Gerhana, Y. A., Huda, I. M., Rahman, A., Manaf K. & Subaeki, B. (2020). *E-Mail Message Encryption Using Advanced Encryption Standard (AES) and Huffman Compression Engineering*. 2020 6th International Conference on Wireless and Telematics (ICWT).

Katakri. 2020. *Tietoturvallisuuden auditointityökalu viranomaisille*. Traficomin julkaisusarja. ISSN 2669-8757, verkko.

Khajuria, A. & Srivastava, R. (2013). *Analysis of the DDoS Defence Stratigies in Cloud Computing*. International Journal of Enhanced Research in Management & Computer Applications. Vol. 2, Issue 2, 2013. Issn no: 2319-7471.

Khalil, S. M., Bahsi, H. & Korõtko, T. (2024). *Threat modeling of industrial control systems: A systematic literature review*. Computers & Security 136 (2024) 103543.

Kohnfelder, L. & Garg, P. (1999). *The threats to our products*. © 2009 Microsoft Corporation.

Kumar, R., Dey, R., Guelton, K., Bali, A. & Singh, U. (2024). *Adaptive control for cyber-physical systems under man-in-the-middle attacks with false data injections*. Journal of the Franklin Institute 361 (2024) 106661. Elsevier Inc.

Kuperman, B. A., Brodley, C. E., Ozdoganoglu, H., Vijaykumar, T. N. & Jalote, A. (2005). *Detection and Prevention Buffer Overlow Attacks*. Communications Of The Acm. November 2005/Vol. 48, No. 11.

Kyberturvallisuuskeskus. (2023a). *Näin keräät ja käytät lokitietoja*. Viitattu: 29.1.2024. https://www.kyberturvallisuuskeskus.fi/fi/ajankohtaista/ohjeet-ja-oppaat/nain-keraat-ja-kaytat-lokitietoja?toggle=Lokeja%20koskeva%20lainsäädäntö.

Kyberturvallisuuskeskus (2023b). *Tietomurtoaalto leviää organisaatiosta toiseen – katkaise tietojenkalastelu*. Viitattu: 3.5.2024. https://www.kyberturvallisuuskeskus.fi/fi/tietomurtoaalto-leviaa-organisaatiosta-toiseen-katkaise-tietojenkalastelu.

Kyberturvallisuuskeskus. (2016a). *Palvelunestohyökkäysten ehkäisy ja torjunta*. Kyberturvallisuuskeskus, Viestintävirasto. Ohje 3/2016 liite 1.

Kyberturvallisuuskeskus. (2016b). *Lokien keräys ja käyttö. Ohje lokitietojen tallentamiseen ja hyödyntämiseen*. Kyberturvallisuuskeskus, Viestintävirasto. Ohje 4/2016.

Laki julkisen hallinnon tiedonhallintalaista 906/2019, 17 §. Finlex. Viitattu: 29.1.2024: https://www.finlex.fi/fi/laki/alkup/2019/20190906.

Laki Oikeusrekisterikeskuksesta 16.11.2012, 1§. Finlex. Viitattu: 7.3.2024: https://www.finlex.fi/fi/laki/alkup/1995/19951287.

Lal Das, M. & Samdaria, N. (2014). *On the security of SSL/TLS-enabled applications*. Applied Computing and Informatics (2014) 10, 68–81.

Li, N., Xie, T., Jin, M. & Liu, C. (2010). *Perturbation-based user-input-validation testing of web applications*. The Journal of Systems and Software 83 (2010) 2263–2274. DOI:10.1016/j.jss.2010.07.007.

Lizama-Pérez, L. A. (2022). *Digital signatures over HMAC entangled chains*. Engineering Science and Technology, an International Journal 32 (2022) 101076.

Lehto, M. (2022). *APT cyber-attack modelling - building a general model*. Teoksessa: R. P. Griffin, U. Tatarand, & B. Yankson (toim)., ICCWS 2022: *Proceedings*

of the 17th International Conference on Cyber Warfare and Security (17, pp. 121-129). Academic Conferences International Ltd. The proceedings of the 17th International Conference On Cyber Warfare And Security. https://doi.org/10.34190/iccws.17.1.36.

Lian, W., Rescorla. E., Shacham, H. & Savage, S. (2013). *Measuring the Practical Impact of DNSSEC Deployment*. 22nd USENIX Security Symposium. August 14–16, 2013, Washington, D.C., USA. ISBN 978-1-931971-03-4.

Liu, J., Gao, Y. & Hu, F. (2021). *A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM*. Computers & Security 106 (2021) 102289. Elsevier Ltd.

Lukka, K. (2014). *Konstruktiivinen tutkimusote*. Viitattu: 10.4.2023. https://metodix.fi/2014/05/19/lukka-konstruktiivinen-tutkimusote/.

Lukka, K. (2000). *The Key Issues of Applying the Constructive Approach to Field Research*. Teoksessa: T. Reponen (toim). *Management Expertise in the New Millennium*. In Commemoration of the 50th Anniversary of Turku School of Economics and Business Administration, Series A-1:2000, Publications of Turku School of Economics and Business Administration, Turku, 113-128.

Macharia, K. W. (2021). *Cryptographic Hash Functions*. ResearchGate.

Mahjabin, T., Xiao, Y., Sun, G. & Jiang, W. (2017). *A survey of distributed denial-of-service attack, prevention, and mitigation techniques*. International Journal of Distributed Sensor Networks 2017, Vol. 13(12). DOI: 10.1177/1550147717741463.

Marpaung, J. A. P., Sain, M. & Lee, H-J. (2012). *Survey on malware evasion techniques: state of the art and challenges*. Department of Ubiquitous IT, Graduate School of General, Dongseo University. ISBN 978-89-5519-163-9.

Martín, A. G., Beltrán, M., Fernández-Isabel, A. & Martiín de Diego, I. (2021). *An approach to detect user behaviour anomalies within identity federations*. Computers & Security 108 (2021) 102356.

Mathew, M. & Kazi, F. (2024). *Hardware-in-Loop (HIL) Testbed Design of Thermal Power Plant for Threat Modeling and Attack Vector Analysis*. International Journal of Critical Infrastructure Protection (2024), doi: https://doi.org/10.1016/j.ijcip.2024.100675.

Microsoft. (2024). *Applocker*. Viitattu: 29.4.2024. https://learn.microsoft.com/en-us/windows/security/application-security/application-control/windows-defender-application-control/applocker/applocker-overview.

Microsoft (2018). *NTLM vs Kerberos*. Viitattu: 22.1.2024. https://answers.microsoft.com/en-us/msoffice/forum/all/ntlm-vs-kerberos/d8b139bf-6b5a-4a53-9a00-bb75d4e219eb.

Microsoft (2016). *NTLM Overview*. Viitattu: 22.1.2024. https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/hh831571(v=ws.11)?redirectedfrom=MSDN.

Minella, J. (2022). *Wireless Security Architecture, Designing and Maintaining Secure Wireless for Enterprise*. New Jersey: John Wiley & Sons, Inc., Hoboken.

Mitre ATT&CK (2024a). *ATT&CK*. Viitattu: 11.4.2024. https://attack.mitre.org.

Mitre ATT&CK (2024b). *APT29*. Viitattu: 11.4.2024 . https://attack.mitre.org/groups/G0016/.

Mitre ATT&CK (2024c). *Phishing*. Viitattu: 3.5.2024. https://attack.mitre.org/techniques/T1566/.

Moh, M., Pininti, S., Doddapaneni, S. & Moh, T-S. (2016). *Detecting Web Attacks Using Multi-Stage Log Analysis*. 2016 IEEE 6th International Conference on Advanced Computing. DOI 10.1109/IACC.2016.141.

Mughal, A. A. (2022). *Building and Securing the Modern Security Operations Center (SOC)*. International Journal of Business Intelligence and Big Data Analytics.

Muthalagu, R. & Sanjay, S. (2021). *Evil Twin Attack Mitigation Techniques in 802.11 Networks*. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 12, nro 6, 2021.

Mäkinen, O. (2006). *Tutkimusetiikan ABC.* Vaajakoski: Gummerus Kirjapaino Oy.

Möckel, C. & Abdallah, A. E. (2011). *Understanding the Value and Potential of Threat Modeling for Application Security Design – An E-Banking Case Study*. Journal of Information Assurance and Security. ISSN 1554-1010 Volume 6 (2011) pp. 346–356.

Nahari, S. (2021). *Best Defence? Our Red Team Lead Reveals 4 MFA Bypass Techniques*. Viitattu: 27.4.2024. https://www.cyberark.com/resources/threat-research-blog/mfa-bypass-techniques-from-red-team-research.

Neil, I. (2020). *CompTIA Security+: SY0-601 Certification Guide, Second Edition*. Birmingham: Packt Publishing Ltd.

NIST SP 800-53 (2020). *Security and Privacy Controls for Information Systems and Organizations*. NIST Special Publication 800-53 Revision 5.

Nweke, L. O. & Wolhusen, S. D. (2020). *A Review of Asset-Centric Threat Modelling Approaches*. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 11, No. 2, 2020.

Ometov, A., Bezzateev, S., Mäkitalo, N., Andreev S., Mikkonen, T. & Koucheryavy, Y. (2018). *Multi-Factor Authentication*. A Survey. Cryptography 2018, 2, 1; doi:10.3390/cryptography2010001.

OP. (2024). *Mitä kybervakuutus korvaa?* Viitattu: 7.5.2024 https://www.op.fi/yritykset/vakuutukset/toiminnan-vakuutukset/kybervakuutus.

Orchilles, J. (2010). *Microsoft Windows 7 Administrator's Reference*. https://doi.org/10.1016/B978-1-59749-561-5.00011-5.

ORK. (2023). *Oikeusministeriön ja Oikeusrekisterikeskuksen välinen tulossopimus vuosille 2024–2027*. Viitattu: 7.3.2023. https://www.oikeusrekisterikeskus.fi/material/sites/ork/tulossopimukset/w7q6np9v9/OMORK_Tulossopimus_2024_002.pdf.

Owasp. (2024). *Owasp Threat Dragon*. Viitattu: 6.3.2024. https://owasp.org/www-project-threat-dragon/.

Owasp. (2024). *Owasp Risk Rating Methodology*. Viitattu: 1.5.2024. https://owasp.org/www.community/OWASP_Risk_Rating_Methodology.

Oyegoke, (2011). *The constructive research approach in project management research*. International Journal of Managing Projects in Business, Vol. 4 Iss 4 pp. 573–595.

Peiris, C., Pillai, B. & Kudrati, A. (2022). *Threat Hunting in the Cloud, Defending AWS®, Azure® and Other Cloud Platforms Against Cyberattacks*. Hoboken: John Wiley & Sons, Inc.

Polu, S. & Bapuji, V. (2024). *Mitigating DDoS attacks in cloud computing using machine learning algorithms*. Brazilian Journal of Development. ISSN: 2525-8761.

Poston, H. E. (2022). *Blockchain Security from the Bottom Up, Securing and Preventing Attacks on Cryptocurrencies, Decentralized Applications, NFTs, and Smart Contracts*. New Jersey, Hoboken: John Wiley & Sons, Inc.

Purba, A. & Soetomo, M. (2018). *Assessing Privileged Access Management (PAM) using ISO 27001:2013 Control*. Proceedings of Annual Conference on Management and Information Technology (ACMIT) 2018

Tarandach, I. & Coles. M., J. (2020). *Threat Modeling, A Practical Guide for Development Teams*. United States of America: O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

Ucedavélez, T. & Morana, M. (2015). *Risk Centric Threat Modeling, Process for Attack Simulation and Threat Analysis*. New Jersey, Hoboken: John Wiley & Sons, Inc.

Ullman, C., J., L. (2023). *The Active Defender, Immersion in the Offensive Security Mindset*. New Jersey, Hoboken: John Wiley & Sons, Inc.

Radhika, B. S., Kumar, N., Shyamasundar, R. K. & Vyas, P. (2020). *Consistency analysis and flow secure enforcement of SELinux policies*. Computers & Security 94 (2020) 101816. Elsevier Ltd.

Romana, S., Jha, A., Reddy, J., Pareek, H. & Eswari, L. (2015). *Practical Application Whitelisting*. ResearchGate.

Rouland, Q., Hamid, B. & Jaskolka, J. (2021). *Specification, detection, and treatment of STRIDE threats for software components: Modeling, formal methods, and tool support*. Journal of Systems Architecture 117 (2021) 102073.

Ruwase, O. & Lam, M. S. (2004). *A Practical Dynamic Buffer Overflow Detector*. National Science Foundation.

Sarmaha, U., Bhattacharyyaa, D. K & Kalitab, J. K. (2018). *A survey of detection methods for XSS attacks*. Journal of Network and Computer Applications (2018), doi: 10.1016/j.jnca.2018.06.004.

Satapathy, A. & Livingston, J. (2016). *A Comprehensive Survey on SSL/ TLS and their Vulnerabilities*. International Journal of Computer Applications (0975–8887) Volume 153 – No 5, November 2016.

Satapathy, S. R. (2014). *Threat Modeling in Web Applications*. Department of Computer Science and Engineering National Institute of Technology Rourkela Rourkela, Odisha, 769 008, India.

Sentanoe, S. & Reiser, H. P. (2022). *SSHkex: Leveraging virtual machine introspection for extracting SSH keys and decrypting SSH network traffic*. Forensic Science International: Digital Investigation 40 (2022) 301337. Elsevier Ltd.

Sharieh, S. & Ferworn, A. (2021). *Securing APIs and Chaos Engineering*. 2021 IEEE Conference on Communications and Network Security (CNS).

Shostack, A. (2014). *Threat modeling, designing for security*. Indianapolis, Indiana: John Wiley & Sons, Inc.

Songa, A. V. (2022). *A Review of DDoS Attacks and its Countermeasures in Cloud Computing*. ResearchGate. DOI:10.1109/ISCON52037.2021.

Souppaya, M. & Scarfone, K. (2016). *Data-Centric System Threat Modeling*. Draft NIST Special Publication 800-154. National Institute of Standards and Technology.

Sowah, R. A., Ofori-Amanfo, K. B., Mills G. A. & Koumadi, K. M. (2019). *Detection and Prevention of Man-in-the-Middle Spoofing Attacks in MANETs Using Predictive Techniques in Artificial Neural Networks (ANN)*. Journal of Computer Networks and Communications Volume 2019, Article ID 4683982.

Stallings, W. (2019). *Effective Cybersecurity, Understanding and Using Standards and Best Practices*. Pearson Education, Inc.

Taanila, A. (2007). *Laadullinen aineiston analyysi*. Kansanterveystieteen ja yleislääketieteenlaitos. Oulun yliopisto.

Tuomi, J. & Sarajärvi, A. (2018). *Laadullinen tutkimus ja sisällönanalyysi*. Helsinki: Kustannusosakeyhtiö Tammi.

Traficom (2024). *Kybersää, Tammikuu 2024*. Liikenne- ja viestintävirasto, Kyberturvallisuuskeskus. Viitattu: 13.3.2024. https://www.kyberturvallisuuskeskus.fi/sites/default/files/media/file/Kybersää%20tammikuu%202024.pdf

Valtiovarainministeriö. (2021). *Valtioneuvostotasoisen riskienhallinnan kehittäminen, Työryhmän loppuraportti*. Helsinki: Valtiovarainmisteriön julkaisuja – 2021:28.

Valtonen, A. (2005). *Ryhmäkeskustelut – Millainen metodi?* Teoksessa: J. Ruusuvuori & L. Tiittula. (toim). *Haastattelu. Tutkimus, Tilanteet ja Vuorovaikutus.* Jyväskylä: Gummerus Kirjapaino Oy.

Verma, M. & Nand, P. (2023). *A Profile-Based Privacy Protection Method using Sandbox Environment and k-Anonymity: Computer Data Privacy*. 2023 International Conference on Communication, Security and Artificial Intelligence (ICCSAI).

Virtanen, A. (2006). *Konstruktiivinen tutkimusote. Miten koulutus ja elinkeinoelämän odotukset kohtaavat ammattikorkeakoulun opinnäytetöissä*. Ammattikasvatuksen aikakauskirja, 8(1), 46–52.

Walls, A., McMullen, L., Heiser, J. & Gopal, D. (2023). *Maverick Research: Risk Management Produces Bad Cybersecurity*. Gartner.

Wiesner, M. C. (2023). *PowerShell Automation and Scripting for Cybersecurity, Hacking and defense for red and blue teamers*. Birmingham: Packt Publishing Ltd.

Wonga, A. Y., Chekolea, E. G., Ochoab, M. & Zhoua, I. (2023). *On the Security of Containers: Threat Modeling, Attack Analysis, and Mitigation Strategies*. Computers & Security 128 (2023) 103140. Elsevier Ltd.

Ylonen, T. (2019). *SSH Key Management Challenges and Requirements*. in 2019 10th IFIP International Conference on New Technologies, Mobility and Security: Proceedings of NTMS 2019 Conference and Workshop. International Conference on New Technologies Mobility and Security, IEEE, IFIP International Conference on New Technologies, Mobility and Security, Canary Islands, Spain, 24/06/2019.

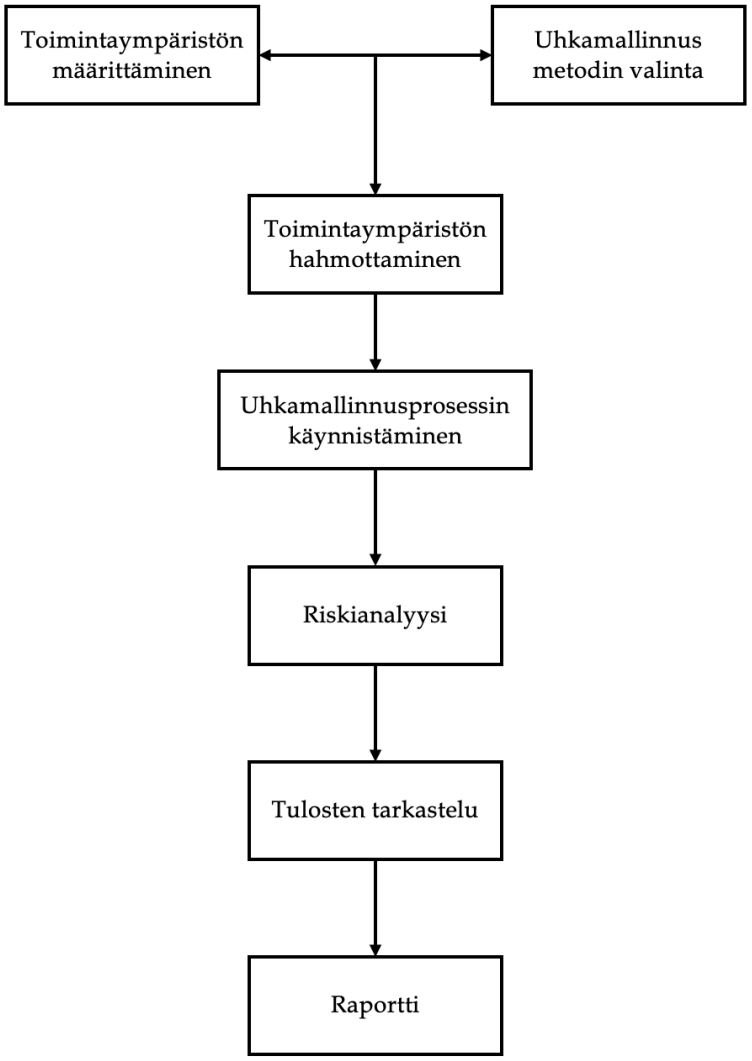
Yunus, M. A., Brohan, M. Z., Nawi, N. M., Surin, E. S. M., Najib, N. A. M. & Liang, C. W. (2018). *Review of SQL Injection: Problems and Prevention*. International Journal on Informatics Visualization. Vol 2 (2018) N0 3–2 ISSN : 2549-9610.

Zhang, Z., Cao, Y., Jahanshahi, H. & Mou, J. (2023). *Chaotic color multi-image compression-encryption/ LSB data type steganography scheme for NFT transaction security*. Journal of King Saud University - Computer and Information Sciences 35 (2023) 101839.

Zhang, Z., Zhang, Y. & Hu, Y. C. (2007). *Practical Defenses Against BGP Prefix Hijacking*. ECE Technical Reports. Paper 364.

Zlatkovski, D., Mileva, A., Bogatinova, K. & Ampov, I. (2018). *A New Real-Time File Integrity Monitoring System for Windows-based Environments*. Faculty of Computer Science.

LIITE 1 UHKAMALLINNUSPROSESSI



KUVIO 12 Uhkamallinnusprosessi

Kuviossa 12 on esitetty tutkimuksessa tehty uhkamallinnusprosessi. Prosessi lähti käyntiin toimintaympäristön määrittämisellä ja uhkamallinnus metodin valinnalla. Uhkamallinnus metodin valintaa ohjaa pitkälti tarkastelun kohteena oleva toimintaympäristö. Koska tarkasteltavana kohteena oli oikeushallinnon tietotekniikkaympäristö, niin uhkamallinnus tehtiin ohjelmistokeskeisellä uhkamallinnuksella STRIDE metodologialla.

Toimintaympäristön hahmottaminen tehtiin OWASP:in Threat Dragon työkalun avulla ja tämän jälkeen aloitettiin uhkamallinnusprosessi. Uhkaskenaarioiden tarkastelut tehtiin Elevation of Privilege ja Backdoors & Breaches -korttisarjojen avulla. Käsitellyt uhkaskenaariot tallennettiin OWASP Threat Dragon-työkaluun. Uhkamallinnustyöpajat toteutettiin virtuaalityöpajoina, joi-

ta oli yhteensä yksitoista. Käsiteltyjen uhkaskenaarioiden jälkeen suoritettiin riskianalyysi, jossa käytettiin apuna laadullista riskianalyysiä.

Viimeisessä vaiheessa suoritettiin tulosten tarkastelu ja aloitettiin raportin kirjoittaminen. Uhkamallinnus työ liittyy olennaisesti kohdeorganisaation riskienhallintaan ja osa riskeistä siirtyy organisaation riskienhallintaprosessiin jatkokäsittelyä varten.

LIITE 2 ESIMERKKI KRIITTISEN UHKAN KÄSITTELYSTÄ

TAULUKKO 30 Esimerkki kriittisen uhkan käsittelystä

Skenaario	Kuvaus	Luokittelu	Lieventämistoimenpiteet
Sosiaalinen manipulointi	Hyökkääjä lähettää tietojenkalastelu viestejä päästääkseen uhrin järjestelmään.	Kriittinen/ Spoofing	• Henkilöstön koulutus. • Zero trust -arkkitehtuuri. • Automaattiset suodatukset. • Räätälöidyt suodatussäännöt. • Lokivalvonta. • MFA. • Automaattiskannaukset.

(Ks. Mitre ATT&CK 2024c.)

Taulukossa 30 on annettu esimerkki kriittisen uhkan käsittelystä. Kyseistä uhkaskenaariota ei ole käsitelty tässä tutkimuksessa, mutta tutkimuksen luotettavuuden parantamiseksi tutkija on halunnut antaa esimerkin, miten kriittinen uhka voidaan käsitellä. Uhka on käsitelty vaikuttavuuden ja todennäköisyyden perusteella taulukossa 31.

Sosiaalinen manipulointi on hyökkäys, jonka avulla manipuloidaan ihmistä suorittamaan toimintoja tai luovuttamaan tietoja. Tietojenkalastelu eli haitallisten sähköpostien lähettäminen on yksi tunnetuimmista sosiaalisen manipuloinnin tekniikoista. Haitallisessa sähköpostiviestissä voidaan houkutella käyttäjää muun muassa avaamaan saastunut liite. (Ullman 2023, 83.) Uhkaskenaario nostetaan kriittiselle tasolle, koska muun muassa kehittyneet uhkatoimijat käyttävät tätä tekniikkaa, joka mahdollistaa pääsyn kohdejärjestelmään (*engl. Initial Access*). Kun hyökkääjä saa pääsyn kohdejärjestelmään, niin hyökkääjä voi mahdollisesti asentaa haittaohjelmia kohdejärjestelmään (*engl. Execution*) ja tämän avulla vakiinnuttamaan jalansijan kohdejärjestelmässä (*engl. Persistence*). (Mitre ATT&CK 2024c.)

Tietojenkalasteluhyökkäykset ovat hyvin yleisiä nykyajan toimintaympäristössä. Kyberturvallisuuskeskus (2023b) on julkaissut artikkelin, jossa Suomalaisten organisaatioiden sähköpostitilejä on kaapattu tietojenkalastelukampanjan avulla. F-Secure (2023) listasi artikkelissaan kolme yleistä kyberuhkaa vuonna 2023, joista yksi oli tietojenkalastelu.

TAULUKKO 31 Esimerkki kriittisen uhkan luokittelusta

Riski = P x I	Erittäin matala (1)	Matala (2)	Kohtalainen (3)	Korkea (4)	Kriittinen (5)
Kriittinen (5)	5	10	15	20	25
Korkea (4)	4	8	12	16	20

Kohtalainen (3)	3	6	9	12	15
Matala (2)	2	4	6	8	10
Erittäin matala (1)	1	2	3	4	5

(Ks. Ucedavélez & Morana 2015, 266.)

Riskin prioriteettiarvo = Todennäköisyys (4) x Vaikutus (4) = 16

Uhkaskenaarion todennäköisyys- ja vaikutusarvo on luokiteltu tasolle neljä eli kokonaisarvo on kuusitoista, joka tarkoittaa kriittistä uhkaa.

- Erittäin matala/ matala, 1–4 pistettä.
- Kohtalainen, 5–9 pistettä.
- Korkea 10–14 pistettä.
- Kriittinen, yli 14 pistettä.