Tomi Jormakka

OWASP TOP 10 -LISTAN RIITTÄVYYS KYBERTURVALLISESSA VERKKOSOVELLUSKEHITYKSESSÄ

JYVÄSKYLÄN YLIOPISTO
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA
2023

TIIVISTELMÄ

Jormakka, Tomi
OWASP Top 10 -listan riittävyys kyberturvallisessa verkkosovelluskehityksessä
Jyväskylä: Jyväskylän yliopisto, 2023, 27 s.
Tietojärjestelmätiede, kuvaileva kirjallisuuskatsaus
Ohjaaja: Saastamoinen, Anna

Tässä kandidaatintutkielmassa perehdytään kymmeneen yleisimpään verkkosovelluskehityksen riskiin pohjautuen OWASP verkkosivuilta löytyvään top 10 listaan. Tutkielma pyrkii vastaamaan kysymykseen, onko OWASP top 10 -lista riittävä puhuttaessa kyberturvallisesta verkkosovelluskehityksestä. OWASP (Open Worldwide Application Security Project) on kyberturvallisuuden alalla yksi tunnetuimpia kansainvälisiä yhteisöjä, jonka tarkoituksena on kehittää turvallisempaa verkkosovelluskehitystä. Tutkielma toteutetaan kuvailevana kirjallisuuskatsauksena. Tutkielman lähteet ovat kerätty seuraavista tietokannoista: Association for Computing Machinery ja Google Scholar. Tulokset osoittavat, että OWASP top 10 -lista on hyvä lähtökohta kyberturvalliselle verkkosovelluskehitykselle. Kuitenkaan top 10 -lista ei kata kaikkia näkökulmia kyberturvallisessa verkkosovelluskehityksessä. Lisäksi alan teknologia ja trendit kehittyvät, joten on suositeltavaa noudattaa jatkuvan oppimisen mallia verkkosovelluskehityksessä.

Asiasanat: OWASP, OWASP top 10, verkkosovelluskehitys, kyberturvallisuus.

ABSTRACT

Jormakka Tomi
Is OWASP Top 10 enough as a cybersecurity perspective in web software development
Jyväskylä: University of Jyväskylä, 2023, 27 pp.
Information system science, descriptive literature review
Supervisor(s): Saastamoinen, Anna

This bachelor's thesis delves into the ten most common risks in web application development, based on the OWASP (Open Web Application Security Project) Top 10 list available on their website. The thesis aims to address the question of whether the OWASP Top 10 list is sufficient when discussing cybersecurity in web application development. OWASP is one of the most well-known international communities in the field of cybersecurity, with a mission to enhance safer web application development practices. The thesis employs a descriptive literature review methodology. The sources for the thesis have been gathered from the following databases: the Association for Computing Machinery and Google Scholar. The findings indicate that the OWASP Top 10 list serves as a solid foundation for cybersecurity in web application development. However, it is acknowledged that the Top 10 list does not cover all aspects of cybersecurity in web application development, as the field continues to evolve along with its technologies and trends. Therefore, adopting a model of continuous learning is recommended for those engaged in web application development.

Keywords: OWASP, OWASP top 10, web application development, cybersecurity.

KUVIOT

SISÄLLYSLUETTELO

1 JOHDANTO

Tässä kuvailevassa kirjallisuuskatsauksessa käsitellään OWASP 10 -listan riittävyyttä verkkosovelluskehityksessä kyberturvallisuuden näkökulmasta. Tutkimus kysymys on OWASP top 10 -listan riittävyys kyberturvallisessa verkkosovelluskehityksessä. OWASP 10 yksi käytetyimmistä ohjenuorista verkkosovelluskehityksessä. Kyberturvallisuuden tärkeyttä ei voi liikaa korostaa nyky-yhteiskunnassa, sillä teknologiaa, jolla on pääsy maailmanlaajuiseen verkkoon, löytyy kaikkialta. Lisäksi Euroopan unioni on laatinut lain, jonka mukaan yritysten, organisaatioiden ja valtioiden on noudatettava GDPR lakia, joka takaa yksilöille oikeutta ja turvaa yksityisyyttä koskevissa asioissa. Tämä katsaus ei käsittele paikallisesti toimivien laitteiden tietoturvariskejä, jotka ovat yleisesti ottaen fyysisen maailman esteitä. Tässä kirjallisuuskatsauksessa ei myöskään keskitytä siihen, kuinka haavoittuvuuksista pääsisi eroon. Tietovuotojen, haittaohjelmien ja vakoiluyritysten kohteena ovat niin yksityishenkilöt, yritykset kuin valtiotkin. Kaikein suurimpana riskinä haavoittuvuuden kannalta on pidetty kriittistä infrastruktuuria kuten vesiverkostoja, sähköverkostoja sekä liikenne- ja terveydenhuoltojärjestelmiä. Mielestäni on myös hyvä muistaa se, että kaiken fyysisen teknologian sekä sen sisällä olevan aineettoman koodin on rakentanut ihminen. Näin ollen voidaan olettaa, että vaikka jokin järjestelmä olisi tehty tiiviiksi kyberhyökkäyksiltä, niin silti jokin näkökulma on voinut jäädä huomioimatta inhimillisen virheen vuoksi. Lisäksi kovaa vauhtia kehittyvä tekoäly sekä kielimallit voivat kehittää niin sanottuja day zero haavoittuvuuksia, joita emme osaa vielä ennustaa.

Artikkelissaan Andjelka Kelic (Kelic, 2019) tarkastelee kriittisen infrastruktuurin kyberturvallisuuden riskejä ja haasteita. Artikkelissaan hän huomauttaa, että moderni yhteiskunta on erittäin riippuvainen toimivasta infrastruktuurista, kuten tietoliikenteestä, energiaverkoista, vesihuollosta, liikenteestä ja terveydenhuollosta. Kelic korostaa, että kyberhyökkäykset voivat aiheuttaa vakavia häiriöitä kriittisissä infrastruktuureissa, mikä voi johtaa laajamittaiseen tuotannon laskuun, taloudellisiin menetyksiin ja jopa kansallisen turvallisuuden vaarantumiseen. Hän mainitsee esimerkkeinä vuoden 2015 Ukrainan

energiaverkkohyökkäyksen ja vuoden 2017 WannaCry-haittaohjelman, jotka molemmat osoittivat kriittisen infrastruktuurin alttiuden kyberuhille (Kelic, 2019).

Toinen esimerkki kyberhyökkäyksestä on StuxNet hyökkäys, joka kohdistui kriittiseen infrastruktuuriin. Tämän kaltaisia hyökkäyksiä voidaan kutsua jo sodan käynniksi, mutta asiasta puhuttaessa täytyy olla tarkka, sillä suurin osa hyökkäyksistä on jotain muuta. Samuel Greengard (Greengard, 2010) käy artikkelissaan läpi tätä kyberhyökkäystä, joka pelästytti koko maailman. Kyseessä oli verkossa leviävä virus, joka hyödynsi jopa neljää day zero haavoittuvuutta Windows pohjaisissa käyttöjärjestelmissä. Virus oli kuitenkin rakennettu niin taitavasti, että se levisi laitteiden välillä valtavaa vauhtia, mutta ei vahingoittanut niitä millään tavalla ja siitä syystä se pääsi leviämään hyvin laajalle. Viruksen ainoa tehtävä oli aiheuttaa vahinkoa teollisuudessa käytettyihin Siemensin valmistamiin ohjausyksiköihin, mitkä valvoivat muun muassa sähkövoimaloiden jäähdytystä, putkilinjastoja ja sähkökulutusta. Virus antoi käyttäjille väärää tieto esimerkiksi todellisesta generaattorin lämpötilasta, joka aiheutti suuria vaaratilanteita. Greengardin mukaan StuxNet oli levinnyt artikkelin tekohetkellä jo yli 30 000 teollisuuden laitteeseen. Viruksen leviämistä oli miltei mahdoton enää estää, mutta sen vaikutus saatiin eliminoitua paikkaamalla aiemmin mainitut tietoturvallisuusaukot (Greengard, 2010). Kuka tietää, vaikka virusta olisi vieläkin olemassa?

Tutkielman tavoitteena on selvittää, onko OWASP (Open Wordlwide Application Security Project) riittävä ohjenuora verkkosovelluskehitykseen kyberturvallisuuden näkökulmasta. OWASP on kansainvälinen voittoa tavoittelematon avoimen lähteen yhteisö. Heillä on myös testaustyökaluja, joilla voidaan koeolosuhteissa testata esimerkiksi jonkin tietyn sovelluksen toimintaa kyberhyökkäyksiä vastaan.

OWASP:n verkkosivuilta löytyy järjestön ylläpitämä OWASP top 10 -lista yleisimmistä ajankohtaisista kompastuskivistä, joihin törmätään verkkosovelluskehityksessä kyberturvallisuuden näkökulmasta. Huomioitavaa on se, että tässä tutkielmassa ei käydä lävitse, kuinka torjua näitä OWASP top 10 kriittisiä kompastuskiviä. Tutkielma toteutetaan kuvailevana kirjallisuuskatsauksena.

Toisessa luvussa käydään läpi tutkielmalle keskeisiä käsitteitä. Käsitteistöä avataan siltä osin, mitä tutkimuskysymyksen muotoilu sitä tuottaa. Käsitteiden läpikäynnin yhteydessä esiin nousee lisää käsitteitä, joista avataan vain tärkeimmät tämän tutkielman ymmärtämiseksi. Lisäksi jokaisen OWASP top 10 käsitteen yhteydestä löytyy lukijan ymmärtämistä helpottava kuva. Kolmannessa luvussa käydään läpi tutkimuskysymykselle kriittinen kirjallisuus. Lisäksi siellä määritellään kirjallisuuteen pohjautuen mitä on kyberturvallisuus sekä yleisimmät kyberhyökkäykset. Neljännessä luvussa käydään läpi tutkielman löydökset, johtopäätökset sekä jatkotutkimusaiheet.

Lähteet on valittu sureksi osaksi ACM tietokannasta sekä osittain Google Scholarista. Muutamia lähteitä on poimittu ei tieteellisistä lähteistä sillä ajankohtaisin tieto löytyy parhaiten esimerkiksi nettisivuilta, joita päivitetään useammin

tilanteiden muuttuessa. Kuten esimerkiksi lista yleisimmistä kyber- hyökkäyksittä.

2 KESKEISIÄ KÄSITTEITÄ

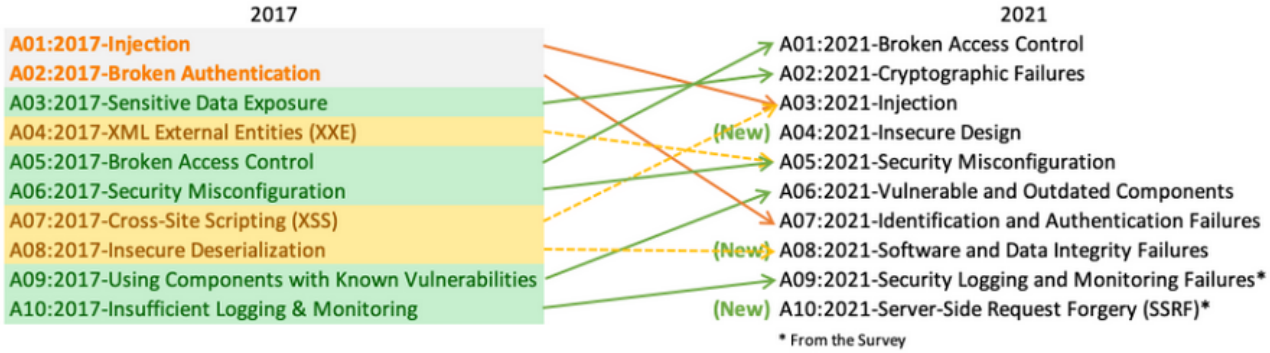
Tässä luvussa käydään läpi tutkielmalle keskeisiä käsitteitä. Ensimmäisessä alaluvussa käsitellään, mikä on OWASP. Toisessa alaluvussa pyritään selittämään OWASP top 10 -listan verkkosovellusriskit. Kaikki top 10 riskiä on myös kuvitettu helpottaakseen lukijan ymmärrystä aiheesta. Kolmannessa alaluvussa käydään läpi muita tutkielmassa esiintyviä keskeisiä käsitteitä.

2.1 OWASP

OWASP (OWASP, 2021) Open Worldwide Application Security Project on voittoa tavoittelematon organisaatio, jonka tarkoituksena on parantaa kyberturvallisuutta verkkosovelluskehityksessä. He tuottavat vapaasti saatavilla olevia ohjeita, artikkeleita, menetelmiä, dokumentaatioita, työkaluja ja teknologioita verkkosovellusten tietoturvan alalta. OWASP opastaa myös käytännön harjoittelun ja opetuksen avulla sovelluskehittäjiä kohti kyberturvallisempaa mallia. Lisäksi heiltä löytyy käytännön työkaluja sovellusten testaamiseen testiympäristöissä. OWASP järjestöä pidetään yhtenä merkittävimpänä verkkosovelluskehityksen edistäjistä kyberturvallisuuden näkökulmasta. OWASP pitää esimerkiksi konferensseja ympäri maailmaa ja siihen kuuluu tuhansia jäseniä. OWASP kattaa tällä hetkellä miltei kaikki maanosat kokonaisuudessaan (OWASP, 2021). Voidaan siis todeta, että organisaatiolla on laaja yhteisö ja se on laajalle levinnyt.

2.2 OWASP top 10

Nettisivuillaan OWASP (OWASP, 2021) on julkaissut OWASP top 10 ajankohtaisinta verkkosovellusriskiä, joiden tarkoituksena on lisätä sovellusten turvallisuutta tunnistamalla niiden kriittisimmät haavoittuvuudet. Lukijan ymmärrystä helpottaakseen riskit on listattu järjestyksessä yleisimmästä riskistä pienimpään. Lisäksi jokaisen määritelmän alla on kuvituskuva tilanteesta, jotta lukija saisi paremman käsityksen mistä on kyse. Suomennoksia kaikille käsitteille ei ole vielä vakiintunut, joten niitä ei myöskään ole kaikilta osin tutkielmaan laitettu. Lista on kehittynyt OWASP järjestön keräämän datan sekä osittain organisaation toteuttamien kyselyiden kautta. Lisäksi top 10 -lista näyttää muutoksen vuosien 2017 ja 2021 välillä (OWASP, 2021).



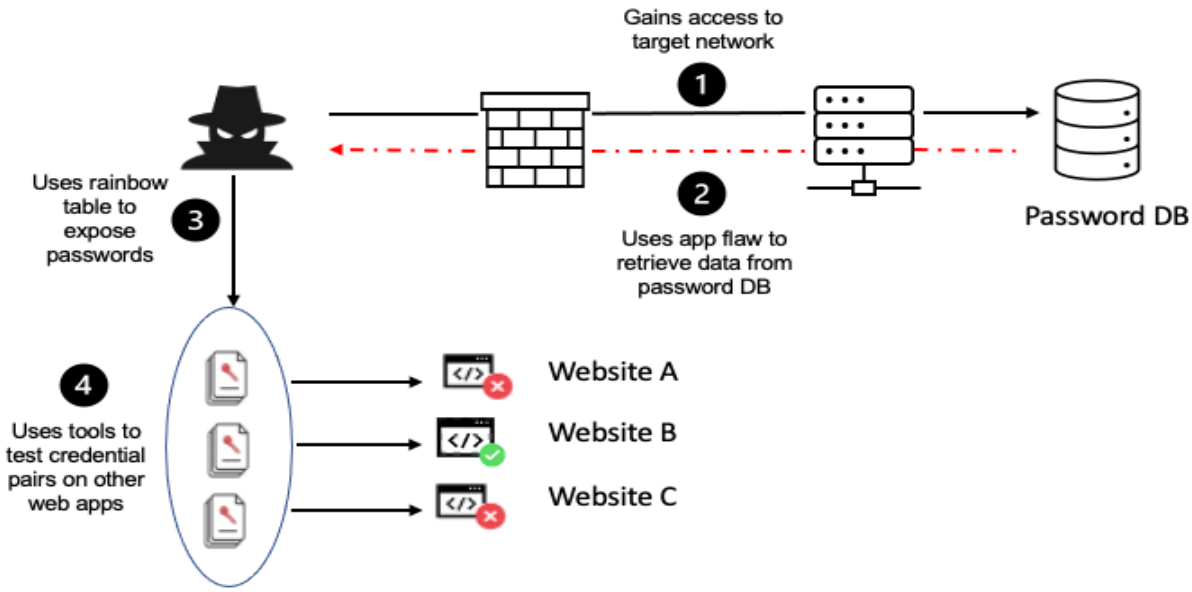
Kuva 1: OWASP top 10 lista (OWASP, 2023).

Broken Acces Control (BAC, rikkinäiset käyttöoikeudet) haavoittuvuudella tarkoittaa sitä, että asiaton käyttäjä pääsee käsiksi sovellukseen, johon hänellä ei ole oikeutta ja pääsee muuttamaan tai poistamaan tietoja. Pahimmassa tapauksessa hyökkääjä pääsee paljastamaan luvattomasti arkaluonteisia tietoja (OWASP, 2021). Artikkelissa (Hassan ym., 2018) kerrotaan yhdestä suurimmasta verkkosovellusriskistä, BAC. Sen mukaan jopa 40 % 330 testatusta verkkosivusta altistuivat tälle haavoittuvuudelle. Siinä tarkennettiin kuitenkin, että kriittisimmät haavoittuvuudet ovat virheellinen istunnon konfiguraatio tai huono syötteen validointi. Break Glass konseptia kuvataan myös herkäksi BAC hyökkäyksille (Hassan ym., 2018). Break Glass aihetta käsiteltiin artikkelissa (Brucker & Petritsch, 2009). Se on niin sanottu takaporttitoiminto sovelluskehityksessä, joka on tehty sen varalle, että virhetilanteessa rikkomalla niin sanotusti lasi päästään nopeasti käsiksi ohjelmiston koodiin. Tämä on kätevä tapa helpottaa hallintaa verkonvalvonnan puolella, mutta luo samalla ikävän väylän kyberhyökkääjille (Brucker & Petritsch, 2009).

Kuva 2: Broken Access Control (F5, 2023).

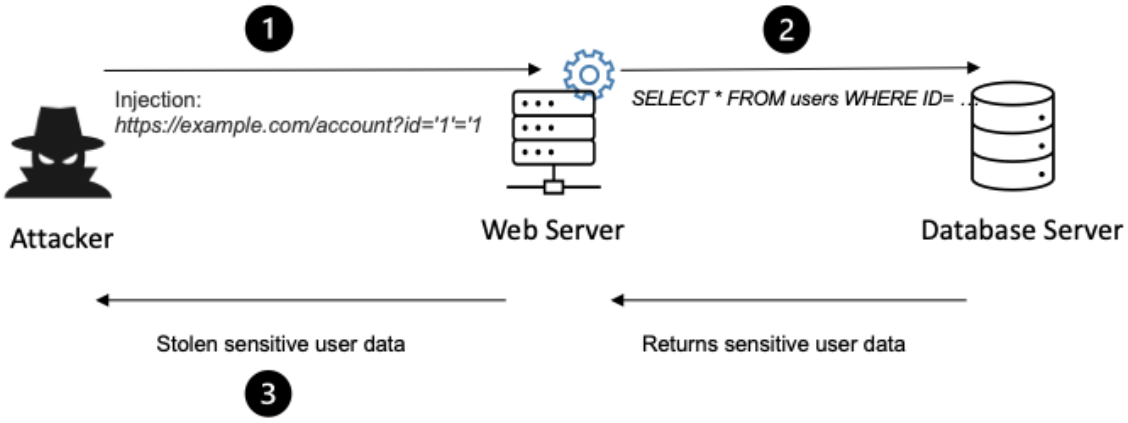
Cryptographic Fillures, salausvirheet syntyvät, kun arkaluonteista tietoa ei suojata säilytettäessä tai siirrettäessä riittävän hyvin (OWASP, 2021). Artikkelissa (Lazar ym., 2014) toteavat, että Cryptographic Failure on tilanne, jossa ohjelmiston salaus ei toimi tarkoitetulla tavalla. Tällaisessa tilanteessa esimerkiksi arkaluontoista tietoa voi vuotaa vääriin käsiin. He ovat listanneet useita syitä tähän

tilanteeseen ja niitä ovat muun muassa puutteellinen suunnittelu, heikko toteutus, virheet protokollassa, avainhallinnan haavoittuvuudet sekä huonot käyttöliittymät. Virhetilanteita voivat aiheuttaa myös itse loppukäyttäjät ja esimerkiksi heikko salasana on itsessään tallainen virhe (Lazar ym., 2014). Artikkelissa (Xiao ym., 2023) käsiteltiin entistä tarkemmin aihetta Cryptographic Failure. Heidän mukaansa nämä virheet tapahtuvat erityisesti kahden tai useamman ohjelman rajapinnassa, jossa ohjelmistot keskustelevat toisensa kanssa. Kyseisissä rajapinnoissa on monesti käytössä Java ohjelmistokieli ja virheet ovat mittakaavaltaan todella monimutkaisia (Xiao ym., 2023).



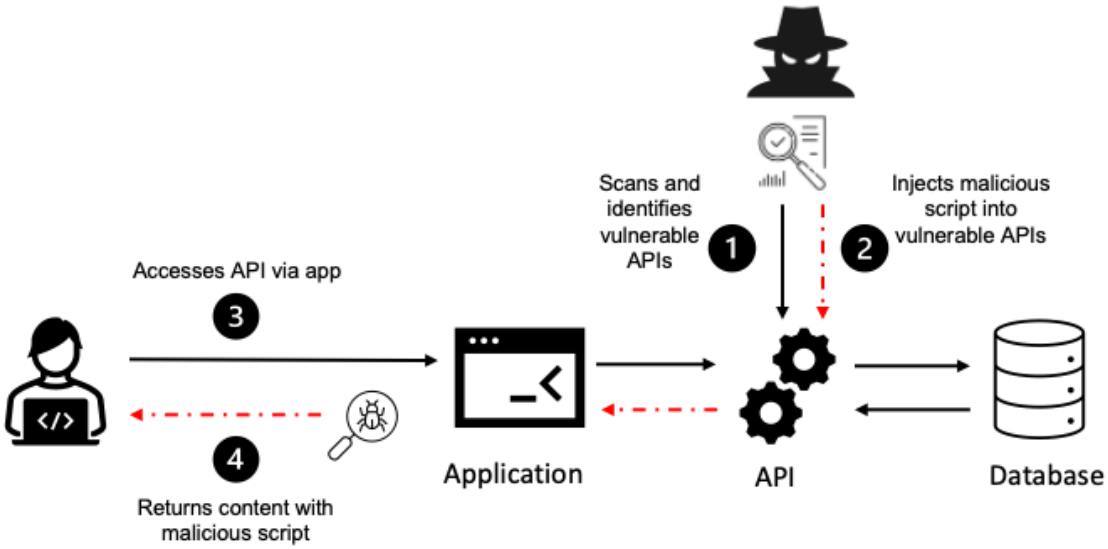
Kuva 3: Cryptographic Failures (F5, 2023).

Injection on haavoittuvuus, joka syntyy, kun hyökkääjä pääsee syöttämään epäluotettavaa tietoa sovellukseen (OWASP, 2021). Artikkelissa (Zhang & Zhang, 2018) kerrotaan SQL injektion hyökkäyksistä, minkä avulla hyökkääjä voi päästä käsiksi osaan tai koko tietokantaan. Tällainen hyökkäys syntyy tilanteessa missä käyttäjän syöttämät tiedot eivät ole lainkaan tai eivät ole tarpeeksi tarkasti validoituja ja hän syöttää omaa SQL koodia esimerkiksi kohdetietokantaan ja näin ollen saa sieltä reaktion. Pahimmassa tapauksessa tällainen tilanne voi jopa kaataa kyseisen tietokannan. Artikkelin mukaan on kuitenkin olemassa tehokkaita keinoja torjua ja ennaltaehkäistä näitä hyökkäyksiä (Zhang & Zhang, 2018). Artikkelissa (Liu ym., 2017) kerrotaan, että niin sanotut koneoppimismallitkaan eivät ole turvassa injektiohyökkäyksiltä. Esimerkiksi antamalla vääriä SQL pyyntöjä käyttäjä saattaa saada manipuloitua tietokannan paljastamaan arkaluontoista tietoa (Liu ym., 2017). Tässä aiheessa on vahva yhteys yllä mainittuun ohjelmistorajapinnoissa tapahtuvaan virheeseen. Koneoppimismallit ovat nimenomaan monessa ohjelmistorajapinnassa toimivia sovelluksia.



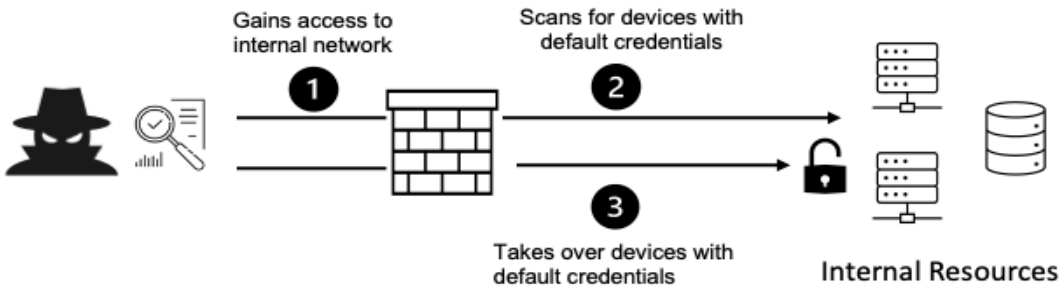
Kuva 4: Injection (F5, 2023).

Insecure Desing (epäluotettava suunnittelu) on laaja haavoittuvuus, joka keskittyy yleisesti kaikkiin verkkosovelluskehityksen elinkaaren suunnittelu- ja kehitysvirheisiin, etenkin ohjelmiston riskienhallinnan puutteeseen (OWASP, 2021). Artikkelissa (Kwon & Su, 2012) todetaan, että epäluotettava suunnittelu (Insecure Desing) on muun muassa sitä, että käytetään kolmannen osapuolen komponentteja fyysisen maailman laitteissa. Ne voivat olla myös kolmannen osapuolen ohjelmistokomponentteja. Ei voi olla siis täysin varmaa, onko kolmannen osapuolen komponenteilla sama protokolla tietoturvaan liittyvissä asioissa kuin muilla komponenteilla (Kwon & Su, 2012). Artikkelissa (Odlyzko, 2010) kuvaillaan epäluotettavan suunnittelun olevan tahattomia heikkouksia ja vikoja teknologiassa. Viat voivat ilmetä itse arkkitehtuurissa, ohjelmistoissa, komponenteissa tai protokollissa. Esimerkkinä hän käyttää heikkoa ohjelmiston salaustasoa sekä ohjelmistoihin tehdyt takaovet ovat myös heikkoa suunnittelua. Tämä niin kutsuttu takaovi on esimerkki BAC hyökkäyksistä. josta kerroin jo aikaisemmin tässä luvussa.



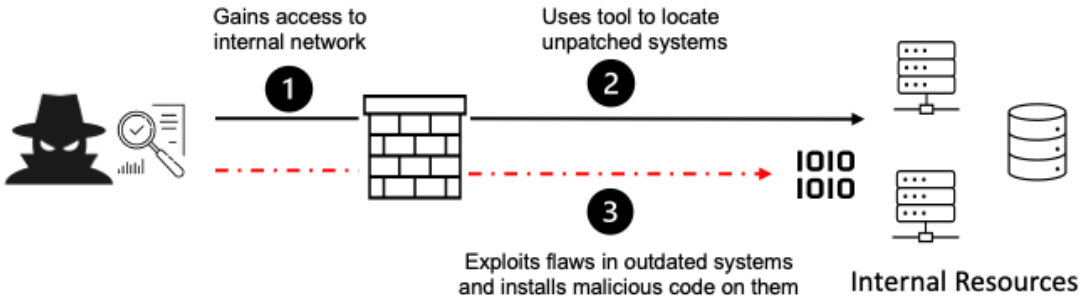
Kuva 5: Insecure Design (F5, 2023).

Security Misconfiguration, altistavat hyökkäyksille, jossa tietoturvahyökkäys on mahdollista suorittaa puutteellisten tietoturva-asetusten vuoksi. Haavoittuvuuden uhkaan kuuluu myös käyttöjärjestelmien päivittämättömyys (OWASP, 2021). Artikkelissa (Sulatycki & Fernandez, 2015) kuvaavat, että Security Misconfiguration on sitä, kun jonkin tietyn järjestelmän tietoturvaan liittyvät asetukset on määritetty heikosti, puutteellisesti tai ne jopa puuttuvat kokonaan. Tällaisia tilanteita ovat artikkelin mukaan muun muassa heikot salasanat, joita on helppo selvittää esimerkiksi brute force attack (vastahyökkäys) menetelmän avulla. Menetelmä ei vaadi juurikaan käyttäjältään ponnisteluja, sillä se on automatisoitu työkalu. Toisena keinona artikkeli listasi parametrien muuttamisen. Tässä kohtaa on hyvä huomata, että monet tavat horjuttaa tietoruvallisuutta solmiutuvat yhteen. Esimerkiksi parametrejä voidaan muuttaa aiemmin mainitsemani BAC hyökkäyksen avulla, mutta siinä saatetaan hyödyntää heikkoa turvallisuuden määrittelyä (Security Misconfiguration). Artikkelissa (Abbas ym., 2023) kerrotaan Security Misconfiguration vaaroista käytettäessä VPN yhteyksiä. Siinä pääsääntöisesti on kerrottu samat alttiutta lisäävät seikat, kun edellä mainitussa artikkelissa, mutta lisänä on heikko käyttäjänhallinta, liian avoimet VPN yhteydet sekä vanhentuneet protokollat. Yllättävää on myös se, että artikkelin mukaan tärkein löydös on nimenomaan puhuttaessa puutteellisista tietoturva-asetuksista VPN yhteyden heikkoudet. Tällä tarkoitetaan, että VPN yhteyksissä on väärin tai puutteellisesti asetetut turvallisuusasetukset (Abbas ym., 2023). Näin ollen on hyvin ymmärrettävää, että puhutaan suurelta osalta ihmisen tekemästä virheestä tai huolimattomuudesta.



Kuva 6: Security Misconfiguration (F5, 2023).

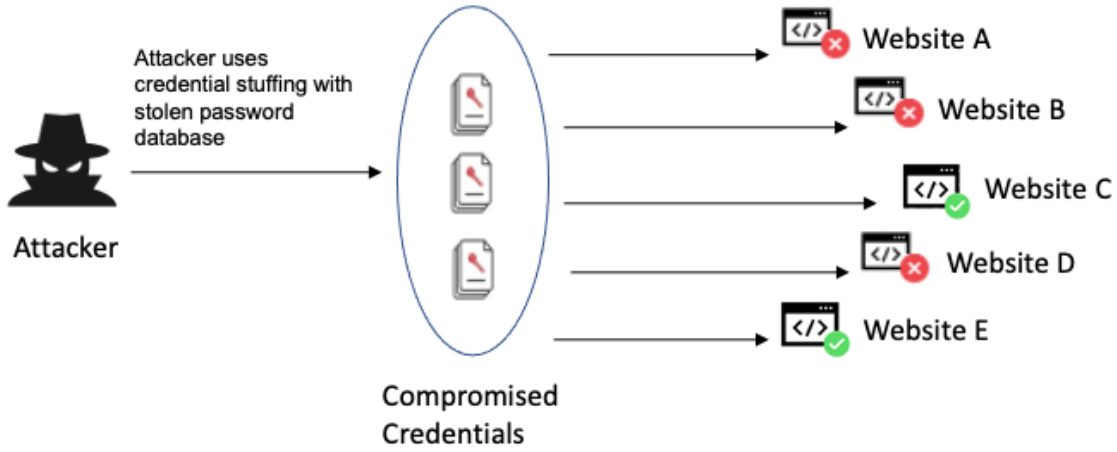
Vulnerable and Outdated Components voivat olla esimerkiksi vanhentuneita kirjastoja, lisäosia tai moduuleita ja niiden käyttö voi johtaa täydelliseen tietomurtoon (OWASP, 2021). Artikkelissa (Neuhaus ym., 2007) kerrotaan vanhentuneista ja haavoittuvaisista komponenteista (Vulnerable and Outdatet Components), joihin ei saa enää ohjelmistopäivityksiä ja ovat siten tietoturvallisuusriski. Vaikka komponentit toimisivat tietyssä olosuhteessa hyvin, ne saattavat aiheuttaa ongelmia silloin kun muut komponentit ja ohjelmistot kehittyvät ja näin ollen niistä tulee riskialttiita. Artikkelin mukaan on myös pystytty kategorioimaan tiettyjä komponenttimalleja, joita ohjelmistokehittäjien tulisi välttää (Neuhaus ym., 2007). Artikkelissa (Uchôa ym., 2021) kerrotaan empiirisen tutkimuksen pohjalta, että jopa 50 % tarkastelun alla olleista komponenteista olivat vanhentuneita ja niistä noin 11 % sisälsi haavoittuvuuksia. Yksi ehdotus olisi käyttää vain luotettujen lähteiden komponentteja. Käsitykseni mukaan nämä luotettavat lähteet pääsääntöisesti tarkoittaisi sitä, ettei käytettäisi kolmannen osapuolen komponentteja.



Kuva 7: Vulnerable and Outdated Components (F5, 2023).

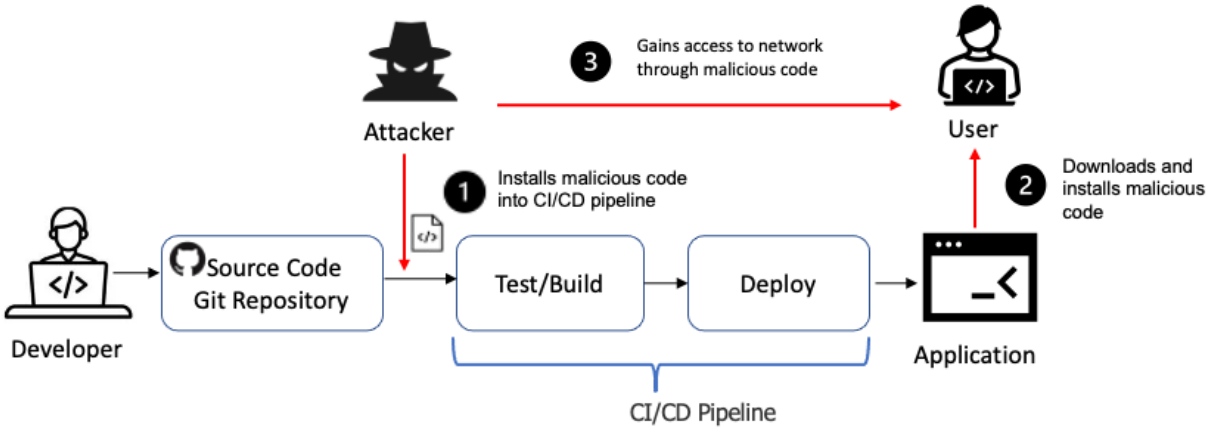
Identification and Authentication Failures ilmenevät yleensä järjestelmän puutteellisena tai helposti kierrettävänä tunnistautumisena (OWASP, 2021). Artikkelissa (Kaixin ym., 2017) kerrotaan tunnistus- ja todennus virheistä (Identification and Authentication Failures) ja siinä käy ilmi, että perinteiset tunnistautumismuodot kuten heikot salasanat ja tunnusluvut ovat erittäin alttiita erilaisille hyökkäyksille sekä niitä on helppo murtaa että varastaa. Artikkelissa käydään

myös läpi menetelmä, jolla voitaisiin tunnistaa ja identifioida ihmisiä suurella tarkkuudella. Kyseessä olisi hiiridynamiikkaan perustuva tunnistusmenetelmä. Jokaisen ihmisen käyttäytyminen on erilaista, jos tarkastellaan hiiren liikkeitä. Artikkelin mukaan näistä voisi luoda toimivan tunnistustietomallin (Kaixin ym., 2017). Toisessa artikkelissa (Lukács ym. 2022) käytiin läpi samaa aihetta. Identifikaatiovirheet ovat todella yleisiä. Tällaisissa tilanteissa hyökkääjä saa pääsyn esimerkiksi tietokantoihin, joihin heillä ei olisi oikeutta päästä. Artikkelissa käy myös ilmi, että näiden tilanteiden ennaltaehkäisy on kriittistä tietoturvalle. Tällaisissa tilanteissa tiedot ovat monesti päässeet vääriin käsiin heti murtautumisen jälkeen. Ennaltaehkäisyksi tuotiin ehdotus, että käytettäisiin CVE luetteloa auttamaan identifioinnissa (Lukács ym., 2022).



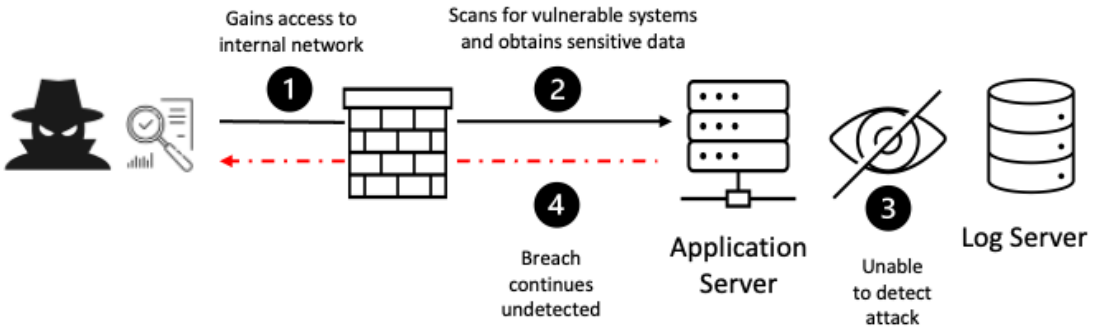
Kuva 8: Identification and Authentication Failures (F5, 2023).

Software and Data Integrity Failures (eheysvirheet) liittyvät ohjelmistopäivityksiin tai epävarmaan CI/CD-järjestelmään, esimerkiksi sovellukseen ladatut laajennukset, kirjastot ja moduulit epäluotettavista lähteistä (OWASP, 2021). Artikkelissa (Sivathanu ym. 2005) käydään läpi aihetta ohjelmiston ja tietojen eheydestä (Software and Data Integrity Failures). Tietojen ja ohjelmistojen tietojen eheyden menettäminen voi johtua todella monesta syystä muun muassa tietoturvaongelmista, tietojen korruptoitumisesta, ohjelmisto- ja laitteistovioista. Tiedon eheyden menetystä on varjeltava erityisen tarkoin kriittisissä infrastruktuureissa. Hyviä keinoja eheyden turvaamiseksi ovat tietojen salaus, varmuuskopiointi ja tarkistussummien käyttö (Sivathanu ym., 2005). Vastaavasti artikkelissa (Wu ym. 2018) kerrotaan varajärjestelmistä, jos tietojen eheydessä ilmenee ongelmia. Nopeasti kehittyvä teknologia aiheuttaa paineita kyseisen asian hallinnalle ja suunnittelijoiden on keksittävä erilaisia menetelmiä, jotta luotettavuus, suorituskyky, skaalautuvuus ja eheys säilyvät ongelmatilanteissa. Artikkelin mukaan monesti joku näistä osa-alueista jää heikommalle huolenpidolle ja siksi ehdotetaan tiedon eheyden ja saatavuuden osalle varajärjestelmää, joka käynnistyy pääjärjestelmässä ilmenneen vian jälkeen (Wu ym., 2018).



Kuva 9: Software and Data Integrity Failures (F5,2023).

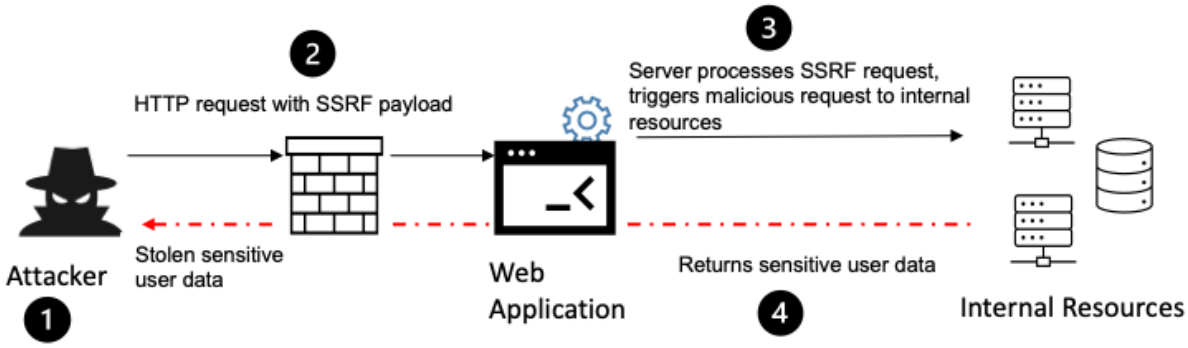
Security Logging and Monitoring Failures (puutteellinen loki ja valvonta) ilmenee, kun järjestelmän lokin kirjaukset ovat puutteellisia tai puuttuvat kokonaan. OWASP (2021) kertoo nettisivuillaan turvallisen kirjauksen ja monitoroinnin epäonnistumisesta (Security Logging and Monitoring Failures). Tähän kuuluu kirjausjärjestelmä, riittävä valvonta, seuranta sekä tietojen analyysi, jotta ongelmat pystymään havaitsemaan ajoissa. Puutteellinen monitorointi voi pahimmillaan johtaa muun muassa tietovuotoihin, haittaohjelmien leviämiseen, luvattomiin kirjautumisiin ja palvelunestohyökkäyksiin (OWASP, 2021).



Kuva 10: Security Logging and Monitoring Failures (F5, 2023).

Server-Side Request Forgery SSRF (palvelinpyynnön väärennös), hyökkääjät voivat päästä sovelluksen palomuurien takana oleviin suojattuihin järjestelmiin (OWASP, 2021). Artikkelissa (Jabiyev ym. 2021) keskitytään aiheeseen Server-Side Request Forgery, SSRF. Kyseinen altistuminen tapahtuu manipuloimalla palvelimen suorittamaa pyyntöä siten, että se lähettää pyynnön haitalliselle sivustolle. Artikkelissa esitellään myös muita tapoja ongelman estämiseksi, esimerkiksi

pyyntöjen rajoittaminen IP osoitteisiin, pyyntöjen tarkistus ennen lähettämistä ja pyyntöjen suoritus eristetyssä ympäristössä (Jabiyev ym., 2021). Myös artikkelissa (Wen ym. 2021) kerrotaan samasta aiheesta ja sen mukaan SSRF manipulointia voidaan käyttää muun muassa palvelintietojen lataamiseen, verkkosivujen haavoittuvuuksien löytämiseen sekä tietojen varastamiseen. Lisäksi SSRF hyökkäyksellä voidaan päästä sisäisiin, niin kutsuttuihin virtual private network liittymiin. Yhtenä keinon tällaisten hyökkäysten estoon esitettiin verkko-osoitteiden tarkistamista ja niiden jäljittämistä (Wen ym., 2021).



Kuva 11: Server-side Reguest Forgery (F5, 2023).

2.3 Muut tutkielmassa esiintyvät keskeiset käsitteet

Adrienn Lukács (Lukács, A. (2016) esittelee tekstissään yksityisyyden merkitystä nyt ja historian saatossa. Yksityisyys käsite on ollut käytössä jo kauan, mutta sen merkitys on hieman muuttunut ja lisääntynyt teknologian myötä. Nykyaikana yksityisyyttä suojaa lait, joiden on reagoitava muutoksiin. Artikkelissa kerrotaan, että erään eurooppalaisen ihmisoikeustutkijan mukaan ei ole mahdollista antaa tyhjentävää oikeudellista määritelmää yksityisyyden suojasta. Määritelmä kehittyy uusien tapauksien kautta. Artikkelissa kuitenkin kerrotan, että yksityisyys on kokonaisuus ihmisen arvokkuuteen, vapauteen ja yksilön itsenäisyyteen liittyviä asioita. Päätelmä yksityisyyden tarkasta määrittelemisestä jää kuitenkin osittain auki. Artikkelissa todetaan, että yksityisyyden määrittämiseen tarvitaan joustava perusta. Sen täytyy käsittää niin yksityishenkilöt kuin yritykset ja valtiot. Lisäksi määritelmään täytyy ottaa huomioon myös teknologia (Lukács, 2016).

Artikkelissa (Last, 2016) käydään läpi aihetta day zero haavoittuvuudet. Sen mukaan kyberturvallisuus asioissa ollaan aina kaksi askelta jäljessä suhteessa hyökkääjiin. Artikkeli pureutuu myös siihen, kuinka näitä haavoittuvuuksia voitaisiin ennustaa, jotta niitä vastaan pystyttäisiin puolustautumaan nopeammin. Day zero haavoittuvuudet ja uhat ovat siis vielä tuntemattomia uhkia. Se tekeekin niistä entistä pelottavampia ihmiskunnalle. Artikkelin mukaan valoa

on kuitenkin tunnelinpäässä, sillä kehitteillä on kaksi koneoppimiseen perustuvaa ennustustapaa sekä yksi regressioon pohjautuva ennustustapa (Last, 2016).

Nettisivuillaan Gdpr (2023) kertoo mitä GDPR on ja mihin sillä pyritään. GDPR (general data protection regulation) on säädetty voimaan toukokuussa 2018. Se antaa oikeutta ja suojaa yksilöille henkilökohtaisen ja arkaluontoisen tiedon leviämistä vastaan. GDPR pitää sisällään yksilöiden henkilötietojen suojaan liittyviä asioita, kuten parempaa kontrollia siihen, miten dataa säilötään, kuinka kauan ja mihin tarkoituksiin sitä käytetään. Lisäksi se säätää organisaatioille vaatimuksen tuoda selvästi asian esille, jos joitain henkilökohtaisia tietoja aiotaan säilöä. Lisäksi tämän lain varjolla organisaatiot ovat oikeuden edessä pakotettuja antamaan yksilöitä koskevat tiedot niitä pyydettäessä sekä poistamaan ne tarpeen päätyttyä. Laki myös säätää organisaatioille minimisakko vaatimukset, mikäli näitä ehtoja rikotaan. Sakkorangaistuksen suuruus riippuu organisaation globaalista brutto tuloksesta (Gdpr.eu., 2023).

Artikkelissa (Askarifar ym., 2018) käydään läpi kesäkuussa 2017 alkunsa saanutta WannaCry lunnasvaatimus kyberhyökkäystä. Tämä ohjelmisto lukitsee käyttäjien tietotekniset laitteet salausmetodilla, kunnes he maksavat lunnasvaatimukset. WannaCry lunnasvaatimus ohjelmistoa on löydetty yli 150 maanosasta ja raportin tekohetkellä lunnasohjelma on saavuttanut vähintään 100 000 organisaatiota maailmanlaajuisesti. Kyseisen ohjelmiston tuhoista selviämistä on vaikeuttanut se, että hyökkääjän tekijät ovat salanneet jokaisen saastutetun laitteen uudella avaimella. Heidän mukaansa WannaCry on alun perin NSA:n tutkimuslaboratoriosta, josta se vuoti kyberhyökkäyksen yhteydessä laajasti ihmisten saataville. WannaCry levisi verkossa hyödyntäen Windows pohjaisten tietokoneiden tietoturva-aukkoa (Askarifar ym., 2018).

Salminen (2020) kuvailee verkkosovellusta palveluksi, jossa verkon avulla voidaan ladata tietoa palvelimelta. Verkkosovelluksella voidaan myös ladata itse palvelimelle uutta tietoa. Myös kaikki nykyaikaiset verkkosivut ovat lähtökohtaisesti verkkosovelluksia, sillä ne ovat dynaamisia. Verkkosovellukset voidaan jakaa myös kahteen osaan. Frontend on visuaalinen osa käyttöliittymää, jonka loppukäyttäjät näkevät ja jolla he voivat hallita sovellusta visuaalisella tasolla. Backend puolestaan sisältää tietokannan sekä palvelimen. Nämä kaksi osasovelluksesta keskustelevat keskenään, josta syntyy yksi kokonaisuus. Verkkosovellukset käyttävät lähes poikkeuksetta http (hypertext transfer protocol) yhteyttä loppukäyttäjän ja palvelimen välillä (Salminen, 2020).

3 OWASP TOP 10 -LISTAN RIITTÄVYYS TIETOTURVALLISUUDEN NÄKÖKULMASTA

Tässä luvussa käydään läpi tutkimuskysymykselle olennaista kirjallisuutta. Tutkimuskysymys on seuraava: Onko OWASP top 10 -lista riittävä tietoturvallisuuden kannalta verkkosovelluskehityksessä. Lisäksi luvussa käsitellään mitä on kyberturvallisuus sekä yleisimmät kyberhyökkäykset. Näin ollen lukija saa tässä luvussa käsityksen siitä, millaista roolia OWASP top 10 suurinta verkkosovelluskehityksen riskiä näyttelee puhuttaessa kyberturvallisuudesta.

3.1 Mitä on kyberturvallisuus?

Singerin ja Friedmanin (2014, s. 34–44) mukaan kyberturvallisuus on tietoverkkojen, tietojärjestelmien ja digitaalisiin laitteisiin liittyviä riskejä ja riskien hallintaa. He tuovat myös esille näkökulman, jossa kyberturvallisuuteen liittyvät asiat eivät aina liity pelkästään teknologiaan, vaan ne voivat ylettyä laajasti aina ihmisyyteen ja politiikkaan asti. Lisäksi he kertovat teknologiaan kohdistuvan kyberturvallisuuden rakentuvan noin kuudesta pääelementistä.

Tietoturvalla tarkoitetaan tietojärjestelmien, tietoverkkojen ja digitaalisten laitteiden suojautumista kaikilta mahdollisilta hyökkäystyypeiltä. Lisäksi se kattaa tietojärjestelmien salauksen sekä päivitetyt palomuurit ja ohjelmistot. Se pitää myös sisällään yksilön ja organisaatioiden vastuulle jäävää toimintaa, joka saattaisi vaarantaa kyberturvallisuuden kuten vahvat salasanat ja monivaiheisen tunnistautumisen.

Verkkoturvallisuudella tarkoitetaan vastaavasti verkon turvallisuuteen liittyvää toimintaa, oli se sitten sisäinen verkko tai niin sanottu World Wide web. Tämä osio pitää sisällään verkkotoiminnan seuraamista, joka tunnetaan paremmin nimellä ylläpito ja valvonta. Siihen kuulu myös luvattoman toiminnan havaitsemisen lisäksi toiminnan estämistä ja oikeanlaista vastaamista virhetilanteisiin. Singer ja Friedman (2014) myös ehdottavat, että tähän kategoriaan kuuluisi muun muassa palvelunestohyökkäysten estäminen. Lisäksi asianmukaiset salausmetodit ja palomuurit kuuluvat samaan ryhmään.

Kolmantena on niin sanottu sovellusturvallisuus. Tällä tarkoitetaan kaikkia ohjelmistoiksi luokiteltavia ohjelmistoja tai verkkosovelluksia. Aihe pitää sisällään IT arkkitehtuurin, jolla tarkoitetaan sitä, että ohjelmistot on rakennettu oikein ja niin, että ne toimivat toisensa kanssa saumattomasti. Ohjelmistot on rakennettava lähtökohtaisesti myös salausta ja kyberturvallisuutta silmällä pitäen. Lisäksi ohjelmistot on päivitettävä, mikäli niistä löytyy tietoturva-aukkoja.

Kirjassa on mainittu vielä sisäisten verkkojen turvallisuus erikseen, joka pitää kuitenkin samat toimet kuin yllä mainittu verkkojen suojaus.

Viidentenä on yksityisyyden suojaaminen, jota voi parantaa esimerkiksi tietoisuuden levittämisellä tai kouluttamalla ihmisiä.

Viimeisenä on käytännön tietoturvallisuus. Se pitää sisällään muun muassa eheys, saatavuus ja luotettavuus periaatteen. Lisäksi yleisen tavan toimia teknologian kanssa kuten ajankohtaiset ohjelmistot, päivittäminen ja kriittinen tarkastelu (Singer & Friedman., 2014, s. 34–44).

3.2 Yleisimmät kyberhyökkäykset

Tässä luvussa käydään läpi yleisimmät kyberhyökkäykset pohjautuen IBM:n ylläpitämään verkkosivuun. Ensimmäinen on haittaohjelmat, jotka ovat tyypillisesti vakoiluohjelmia, viruksia tai hyödylliseksi ohjelmistoiksi naamioituja haittaohjelmia. Uusien haittaohjelmien havaitseminen vie aikaa ja varsinkin viruksen elinkaaren alkuvaiheessa ne aiheuttavat eniten vahinkoa.

Toisena listassa on uhkavaatimukset. Uhkavaatimus ohjelmistojen toimintaperiaate pohjautuu siihen, että se esimerkiksi joko lukitsee laitteet tai lähettää tiedostoja eteenpäin, ellei uhkavaatimusmaksua makseta. Viimeaikaisina trendeinä on nähty se, että hyökkäykset kohdistuvat kansalaisten luottamille sivuille tai tietokannoille, jolloin on suurempi houkutus maksaa lunnaat (IMB, 2023). Tästä hyvä esimerkki on jo aiemmin mainitsemani Wannacry kyberhyökkäys, joka lukitsi käyttäjän tietokoneen, ellei maksanut tiettyä summaa kryptovaluuttana hyökkääjien tilille. Toisena hyvänä esimerkkinä on juuri hiljattain sattunut Psykoterapiakeskus Vastaamon tietomurto, jossa asianomaiset saivat suoraan uhkavaatimusmaksukirjeet, jotta tietoja ei levitettäisi. Tässä tapauksessa itse hyökkäyksen kohteena ollut yritys jätettiin vaatimuksista käsittääkseni kokonaan pois.

Kolmantena (IBM, 2023) hyökkäyksenä on listattu kalastelu. Kalastelu on sosiaalisen manipuloinnin muoto, jossa käyttäjä sortuu antamaan arkaluontoista tietoa esimerkiksi, aidon näköisesti tehtyjen huijausvistien avulla. Tällainen uhka vaivaa eniten kokemattomia käyttäjiä.

Neljäntenä on sisäiset uhat. Sisäiseksi uhaksi voidaan luokitella kuka tahansa, jolla on ollut tai on edelleen pääsy esimerkiksi yrityksen sisäiseen verkkoon. Sisäisiä uhkia on myös hyvin vaikea selvittää, sillä palomuurit ja viruksentorjunta ohjelmat on suunniteltu havaitsemaan uhkia ulkoapäin. Esimerkiksi työntekijä voisi kävellä yrityksen palvelimelle konkreettisesti ja ladata sieltä arkaluontoista tietoa.

Viidentenä ovat DDos eli palvelunestohyökkäykset. Palvelunestohyökkäyksen ainoa tarkoitus on rasittaa joitain verkon olennaista osaa kuten palvelinta, jotta se kaatuu ja joutuu uudelleen käynnistymään. Tällä saatetaan aiheuttaa esimerkiksi hallaa sekä uudelleen käynnistys saattaa tarjota uusia mahdollisuuksia kyberhyökkäyksille.

Kuudentena on niin sanottu APT eli kehittynyt pysyvä uhka. Hyökkääjä jää esimerkiksi sisäiseen verkkoon istumaan niin ettei jätä jälkiä ja sisäinen ylläpito ei sitä huomaa. Näin ollen se voi pitkän aikaa vakoilla yrityksen toimintaa ja varastaa tietoja.

Seitsemäs ja viimeinen on mies keskellä hyökkäys. Tällainen hyökkäys tapahtuu tyypillisesti suojaamattomassa langattomassa verkossa, jolloin hyökkääjä saa kaiken tiedon itselleen kohteen ja langattoman verkon välillä. Hän luo itsesään niin sanotun tukiaseman, jonka kautta tieto kulkee (IMB, 2023).

3.3 OWASP top 10 kyberturvallisuus ohjeena

Artikkelissa (Sane, 2021) käsittelee OWASP:n yleisesti hyväksyttyä turvallisuusohjeistusta, joka on suunnattu verkkosovelluskehitykseen. Artikkelissa käsitellään, onko OWASP:n Top 10 -listaus riittävän kattava turvallisen koodin kirjoittamiseen vai onko kehittäjien syytä tarkastella muitakin turvallisuusriskejä. Artikkeli käy läpi OWASP:n Top 10 -listan sisältöä ja huomauttaa, että lista on erittäin hyödyllinen kehittäjille, jotka haluavat kehittää turvallisuutta verkko- ja verkkosovelluskehityksessä. Hän kuitenkin myös korostaa, että lista ei ole kattava, sillä siinä käsitellään vain kymmentä yleisintä turvallisuusuhkaa, jotka liittyvät verkkosovelluskehitykseen. On siis paljon muitakin seikkoja, jotka tippuvat top 10 listan alapuolelle. Artikkelissa esitellään useita muitakin turvallisuusriskejä, jotka eivät ole mukana OWASP:n Top 10 -listassa, kuten muun muassa "Man-in-the-middle" -hyökkäykset ja "Cross-Site Request Forgery" (CSRF) -hyökkäykset. Sane kertoo, että kehittäjien tulisi käyttää OWASP:n Top 10 -listaa vain lähtökohtana turvallisen koodin kirjoittamiseen, eikä pitää sitä ainoana mahdollisuutena kehittää turvallisia verkkosovelluksia. Hän kehottaa kehittäjiä tekemään laajempaa tutkimusta ja käyttämään muita turvallisuusohjeistuksia, jotta he voivat kehittää mahdollisimman turvallisia sovelluksia. Yhteenvetona artikkeli kertoo, että OWASP top 10 -lista on hyödyllinen ja hyvä lähtökohta. On kuitenkin otettava paljon muitakin asioita huomioon (Sane, 2021).

Artikkelissa (Marchand-Melsom & Nguyen Mai, 2020) tarkastellaan OWASP Top 10 -turvallisuushaavoittuvuuksia ja niihin liittyviä automaattisia korjausmenetelmiä verkko- ja sovelluskehityksessä. Artikkelissa käsitellään erilaisia automaattisia korjausmenetelmiä, kuten koodin muokkausta, lokianalyysiä ja verkkoliikenteen seurantaa. Näitä menetelmiä käytetään havaittujen turvallisuushaavoittuvuuksien korjaamiseen. Artikkelissa käsitellään myös erilaisia haasteita, jotka liittyvät automaattiseen korjaamiseen, kuten väärän positiivisen tunnistamisen riski ja joidenkin turvallisuushaavoittuvuuksien monimutkaisuus. Melsom toteaa, että automaattinen korjaaminen voi olla hyödyllinen lisä perinteiseen manuaaliseen tietoturvanhallintaan. Hän kuitenkin huomauttaa, että automaattinen korjaaminen ei ole yleisratkaisu kaikkiin tietoturvaongelmiin vaan manuaalinen tarkastelu ja korjaaminen ovat edelleen tärkeitä osia tietoturvahallinnassa. (Melsom ym., 2020).

Artikkelissa (Rotella, 2018) käsitellään tietoturva-aukkojen mittaamista ja vertailua ohjelmistoissa. Rotella korostaa tarvetta tarkastella tietoturva-aukkoja peruslinjana ja vertailla niitä muihin ohjelmistoihin, jotta voidaan havaita trendejä ja löytää parannusmahdollisuuksia. Täten voitaisiin löytää niin sanotusti jokin vakio, jonka perusteella voitaisiin koeponnistaa ohjelmistojen kyberturvallisuutta. Rotella käsittelee myös erilaisia ohjelmistoanalyysityökaluja, jotka auttavat löytämään tietoturva-aukkoja. Hän painottaa kuitenkin, että nämä työkalut eivät ole täydellisiä ja että ihmisen tarkastelu on aina välttämätöntä. Tietoturva-aukkojen mittaaminen ja vertailu on jatkuva prosessi, joka vaatii aikaa ja huolellisuutta. Tällaisen prosessin avulla voidaan kuitenkin parantaa ohjelmistojen tietoturvaa ja vähentää riskiä tietoturvaongelmista (Rotella, 2018).

4 Yhteenveto

Tutkielman tarkoituksena oli selvittää, onko OWASP Top 10 kriittisintä verkkosovellusriskiä riittävä määrä toimia, jotta voidaan puhua kyberturvallisesta verkkosovelluskehityksestä. Tutkielma toteutettiin kuvailevana kirjallisuuskatsauksena. Tutkielman aineistot on kerätty Association for Computing Machinery -tietokannasta ja Google Scholarista. Seuraavissa alaluvuissa käydään läpi tutkielman johtopäätökset sekä jatkotutkimusaiheet.

4.1 Johtopäätökset

Tutkielmassa on omat luvut kyberturvallisuudelle, kyberhyökkäyksille sekä OWASP Top 10 -verkkosovellusriskille, jotta tutkimuskysymykseen vastaaminen olisi kattavampaa. Lisäksi lukujen tarkoituksena on helpottaa lukijaa ymmärtämään aiheiden väliset erot. Helposti aiheesta lukiessa saattaisi luulla, että esimerkiksi OWASP Top 10 on kyberhyökkäyksiä, kun tosiasiassa ne ovat heikkouksia, jotka mahdollistavat kyberhyökkäykset.

OWASP Top 10 -lista käsittää tämänhetkiset yleisimmät verkkosovellusriskit, jotka mahdollistavat suuren osan kyberhyökkäyksistä. Näitä riskejä voidaan kuvailla avoimiksi oviksi, joiden kautta hyökkäys tapahtuu. Kirjallisuuteen pohjautuen voidaan todeta, että OWASP Top 10 -lista on hyvä lähtökohta ehkäisemään suurimman osan ajankohtaisista verkkosovellusriskeistä. Kirjallisuudesta käy kuitenkin ilmi, että ei ole hyvä turvautua pelkästään OWASP Top 10 -listaan, sillä sen lisäksi on olemassa myös muita verkkosovellusriskejä.

Kyberturvallisuus verkkosovelluskehityksessä on moninainen kokonaisuus, joten sitä on tarpeen tarkastella useammasta näkökulmasta. OWASP Top 10 -lista ei käsittele muun muassa day zero -haavoittuvuuksia. Ne ovat haavoittuvuuksia ja uhkia verkkosovellusten rakenteessa, joita emme ole vielä huomanneet. Lisäksi muita kyberturvallisuuteen vaikuttavia tekijöitä ovat inhimillisyystekijät, kuten heikko salasana tai huolimaton ohjelmistojen päivittäminen.

Alaluvussa 3.2 käytiin läpi seitsemän yleisintä kyberhyökkäystyyliä. OWASP Top 10 -lista on yhteydessä niistä vain viiteen hyökkäykseen, jotka ovat haittaohjelmat, uhkavaatimusvirukset, palvelunestohyökkäykset, APT-hyökkäykset ja man-in-the-middle-hyökkäykset. Näin ollen kalasteluhyökkäyksiä eikä sisäisiä uhkia OWASP Top 10 -lista käsittele lainkaan. On muistettava, että tapoja kyberhyökkäykseen on myös enemmän. Näitä tapoja ei kuitenkaan usein listata esimerkiksi niiden alhaisen esiintyvyyden vuoksi. Voidaan siis todeta, että OWASP Top 10 ei ole riittävä verkkosovelluskehityksessä, jos puhutaan kyberturvallisuudesta kokonaisuutena. Esimerkiksi kalasteluhyökkäykset vaatisivat loppukäyttäjien koulutusta ja tietoisuuden lisäämistä, sillä niissä tapauksissa loppukäyttäjät tekevät itse virheen esimerkiksi avaamalla epäilyttävän sähköpostiviestin. Sisäisten uhkien poistaminen ajoissa on puolestaan lähes

mahdotonta. Vaikka verkkojen sisäistä valvontaa tehdään, usein sisäiset uhat osaavat kiertää sen tai toimia tavalla, joka ei herätä huomiota.

Lisäksi OWASP Top 10 -listan kuudes kohta, haavoittuneet ja vanhentuneet komponentit, on hankala ehkäistä. Nämä kolmannen osapuolen komponentit ja hakemistot aiheuttavat merkittävän riskin kyberturvalliselle verkkosovelluskehitykselle. Vaikka se on listattu yhdeksi osaksi OWASP Top 10 -listaa, sille ei ole kehitetty täysin varmaa tapaa eliminoida kyseistä ongelmaa. Jos jokin kolmannen osapuolen tekijä unohtaa tai päättää lopettaa esimerkiksi verkkohakemiston tuen, se saattaa aiheuttaa huomaamatta verkkosovelluksille riskitekijöitä.

Alaluvusta 3.3 voidaan päätellä, että kyberturvallisuuden mittaamiselle ei ole vielä vakiintunutta tapaa, jolla saataisiin luotettavia ja vertailukelpoisia tuloksia. Aihe kaipaisi siis kokonaisvaltaisesti lisää tutkimustyötä. Luvussa myös todettiin, että ihmisen työtä tarvitaan aina luotettavien tuloksien saamiseksi, sillä mitattavat kohteet ovat uniikkeja ja mittauksen tulisi olla jatkuvaa.

4.2 Jatkotutkimusaiheet

Ensimmäisenä jatkotutkimusaiheena on paradoksi vakiintuneista nimityksistä kyberturvallisuuden ja tietotekniikan sanastossa. Sanoja kyberturvallisuus, tietoturvallisuus, kyberuhka, kyberhyökkäys ja tietoturvallisuusriski käytetään kirjallisuudessa sekä yleisessä keskustelussa säännöllisen epäsäännöllisesti sekaisin. Englanninkieliset nimet ovat kuitenkin mielestäni parhaiten vakiintuneet.

Toisena jatkotutkimusaiheena on security logging and monitoring failures, joista ei löytynyt juuri ollenkaan tieteellistä kirjallisuutta. Aiheesta mainittiin muutamissa asiayhteyksissä, mutta esimerkiksi määrittelyä näille tietoturvaongelmille ei löytynyt tieteenalan kirjallisuudesta.

Viimeinen jatkotutkimusaihe pyrkisi selvittämään tavan, jolla voitaisiin mitata ja vertailla eri verkkosovellusten kyberturvallisuutta. Tällä hetkellä vertailukelpoisten tulosten saaminen on vielä työläs ja epätarkka prosessi.

LÄHTEET

Abbas, H., Emmanuel, N., Amjad, M. F., Yaqoob, T., Atiquzzaman, M., Iqbal, Z., Shafqat, N., Shahid, W. bin, Tanveer, A., & Ashfaq, U. (2023). Security Assessment and Evaluation of VPNs: A Comprehensive Survey. *ACM Computing Surveys*. https://doi.org/10.1145/3579162

Askarifar, S., Rahman, N. A. A., & Osman, H. (2018). *A REVIEW OF LATEST WANNACRY RANSOMWARE: ACTIONS AND PREVENTIONS*. *7*.

Brucker, A. D., & Petritsch, H. (2009). Extending access control models with break-glass. *Proceedings of the 14th ACM symposium on Access control models and technologies*, 197–206. https://doi.org/10.1145/1542207.1542239

English (US). (2023). Noudettu 24. elokuuta 2023, osoitteesta https://www.f5.com/ https://doi.org/10.1145/3507682

General Data Protection Regulation (GDPR) Compliance Guidelines. (2023). GDPR.Eu. Noudettu 24. elokuuta 2023, osoitteesta https://gdpr.eu/

Greengard, S. (2010). The new face of war. *Communications of the ACM*, *53*(12), 20–22. https://doi.org/10.1145/1859204.1859212

Hassan, M. M., Ali, Md. A., Bhuiyan, T., Sharif, M. H., & Biswas, S. (2018). *Quantitative Assessment on Broken Access Control Vulnerability in Web Applications*.

IBM - United States. (2023, kesäkuuta 7). https://www.ibm.com/us-en

Jabiyev, B., Mirzaei, O., Kharraz, A., & Kirda, E. (2021). Preventing server-side request forgery attacks. *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, 1626–1635. https://doi.org/10.1145/3412841.3442036

Kaixin, W., Hongri, L., Bailing, W., Shujie, H., & jia, S. (2017). A User Authentication and Identification Model Based on Mouse Dynamics. *Proceedings of the 6th International Conference on Information Engineering*, 1–6. https://doi.org/10.1145/3078564.3078581

Kelic, A. (2019). Cyber Risk in Critical Infrastructure. *ACM SIGMETRICS Performance Evaluation Review*, *46*(2), 72–75. https://doi.org/10.1145/3305218.3305243

Kwon, T., & Su, Z. (2012). Detecting and analyzing insecure component usage. *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*, 1–11. https://doi.org/10.1145/2393596.2393599

Last, D. (2016). Forecasting Zero-Day Vulnerabilities. *Proceedings of the 11th Annual Cyber and Information Security Research Conference*, 1–4. https://doi.org/10.1145/2897795.2897813

Lazar, D., Chen, H., Wang, X., & Zeldovich, N. (2014). Why does cryptographic software fail?: A case study and open problems. *Proceedings of 5th Asia-Pacific Workshop on Systems*, 1–7. https://doi.org/10.1145/2637166.2637237

Liu, Y., Wei, L., Luo, B., & Xu, Q. (2017). Fault injection attack on deep neural network. *Proceedings of the 36th International Conference on Computer-Aided Design*, 131–138.

Lukács, A. (2016). *WHAT IS PRIVACY? THE HISTORY AND DEFINITION OF PRIVACY*.

Marchand-Melsom, A., & Nguyen Mai, D. B. (2020). Automatic repair of OWASP Top 10 security vulnerabilities: A survey. *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, 23–30. https://doi.org/10.1145/3387940.3392200

Neuhaus, S., Zimmermann, T., Holler, C., & Zeller, A. (2007). Predicting vulnerable software components. *Proceedings of the 14th ACM conference on Computer and communications security*, 529–540. https://doi.org/10.1145/1315245.1315311

Odlyzko, A. (2010). Providing security with insecure systems. *Proceedings of the third ACM conference on Wireless network security*, 87–88. https://doi.org/10.1145/1741866.1741867

OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation. (2023). Noudettu 24. elokuuta 2023, osoitteesta https://owasp.org/

Rotella, P. (2018). Software security vulnerabilities: Baselining and benchmarking. *Proceedings of the 1st International Workshop on Security Awareness from Design to Deployment*, 3–10. https://doi.org/10.1145/3194707.3194708

Sadlek, L., Čeleda, P., & Tovarňák, D. (2022). Current Challenges of Cyber Threat and Vulnerability Identification Using Public Enumerations. *Proceedings of the 17th*

International Conference on Availability, Reliability and Security, 1–8. https://doi.org/10.1145/3538969.3544458

Salminen, J. (2020). *Verkkosovelluksen haavoittuvuustestaus*.

Sane, P. (2021). Is the OWASP Top 10 List Comprehensive Enough for Writing Secure Code? *Proceedings of the 2020 International Conference on Big Data in Management*, 58–61. https://doi.org/10.1145/3437075.3437089

Singer, P. W., & Friedman, A. (2014). *Cybersecurity: What Everyone Needs to Know*. Oxford University Press.

Sivathanu, G., Wright, C. P., & Zadok, E. (2005). Ensuring data integrity in storage: Techniques and applications. *Proceedings of the 2005 ACM workshop on Storage security and survivability*, 26–36. https://doi.org/10.1145/1103780.1103784

Sulatycki, R., & Fernandez, E. B. (2015). Two threat patterns that exploit" security misconfiguration" and" sensitive data exposure" vulnerabilities. *Proceedings of the 20th European Conference on Pattern Languages of Programs*, 1–11. https://doi.org/10.1145/2855321.2855368

Uchôa, A., Assunção, W. K. G., & Garcia, A. (2021). Do Critical Components Smell Bad? An Empirical Study with Component-based Software Product Lines. *15th Brazilian Symposium on Software Components, Architectures, and Reuse*, 21–30. https://doi.org/10.1145/3483899.3483907

Wu, D., Xia, Y., Sun, X. S., Huang, X. S., Dzinamarira, S., & Ng, T. S. E. (2018). Masking failures from application performance in data center networks with shareable backup. *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, 176–190. https://doi.org/10.1145/3230543.3230577

Wen, S., Wu, Q., Wu, X., Ling, Y., & Ye, Z. (2021). Toward Tracing the Source of Web Attacks Targeted at Web Applications. *Proceedings of the 2021 International Conference on Pattern Recognition and Intelligent Systems*, 10–14. https://doi.org/10.1145/3480651.3480654

Xiao, Y., Zhao, Y., Allen, N., Keynes, N., Yao, D. (Daphne), & Cifuentes, C. (2023). Industrial Experience of Finding Cryptographic Vulnerabilities in Large-scale Codebases. *Digital Threats: Research and Practice*, *4*(1), 4:1-4:18.

Zhang, H., & Zhang, X. (2018). SQL Injection Attack Principles and Preventive Techniques for PHP Site. *Proceedings of the 2nd International Conference on Computer Science and Application Engineering*, 1–9. https://doi.org/10.1145/3207677.3277958