Miikka Jarnola

BUG BOUNTYN HYÖDYT TIETOTURVATESTAUKSESSA

TAPAUSTUTKIMUS - LÄHITAPIOLA

JYVÄSKYLÄN YLIOPISTO
TIETOJENKÄSITTELYTIETEIDEN LAITOS
2018

TIIVISTELMÄ

Jarnola, Miikka
Bug Bountyn hyödyt tietoturvatestauksessa tapaustutkimus - LähiTapiola
Jyväskylä: Jyväskylän yliopisto, 2018, 46 s.
Järjestelmäkehitys/Tietoturva, Tietojärjestelmätiede, pro gradu -tutkielma
Ohjaaja: Siponen, Mikko

Tämän tutkimuksen tarkoituksena oli tuottaa tieteellistä dataa Bug Bountyn hyödyistä ja haitoista yrityksille. Tutkimuksella haluttiin avata uutta tutkimuskenttää Bug Bountyn parissa ja rohkaista yrityksiä ottamaan Bug Bounty käyttöön. Tutkimuskysymyksiä tässä tutkimuksessa oli kolme kappaletta. Ne olivat: Mitä hyötyjä LähiTapiolalle on ollut Bug Bountyn käyttöönotosta, mitä ongelmia ja riskejä Bug Bountysta on ollut LähiTapiolalle, sekä Mitä opittiin ja mitä vietiin käytäntöön? Tutkimus toteutettiin kirjallisuuskatsauksen ja empiirisen tutkimuksen yhdistelmänä. Tutkimustulokset olivat kohtalaisen yksimielisiä. Haastateltujen mukaan Bug Bounty on selvästi auttanut yritystä tuottamaan parempaa tietoturvaa sovelluksilleen. Hyötyinä nähtiin positiivinen julkisuus, vapautuneet resurssit ja haavoittuvuuksien konkreettinen vähentyminen. Suoranaisia ongelmia ei löytynyt. Riskit olivat potentiaalisia riskejä, jotka eivät koskaan toteutuneet. Nämä riskit olivat, negatiivinen julkisuus, palveluiden kaatuminen liiallisten käyttäjämäärien takia ja negatiivinen palaute. Nämä riskit nähtiin geneerisinä ja niiden todettiin pätevän lähes kaikkiin yrityksiin. Tutkimustulosten puitteissa tunnistettiin joitain haittoja, mitä Bug Bounty aiheutti LähiTapiolalle. Nämä haitat voivat aiheutua myös muille yrityksille. Niitä olivat palveluiden hitaus yhden päivän ajan Bug Bounty julkistamisesta, pitkä prosessi haavoittuvuuden löytymisen ja sen korjaamisen välillä, sekä kielimuuri, koska LähiTapiolan Bug Bounty on kansainvälinen. Opittuina asioina esille nousivat muun muassa uusi tapa suhtautua tietoturvaan ja sen parempi ymmärtäminen, ammatillisen osaamisen kehittyminen ja kansainvälisyyden aiheuttamat erot etiikan ja moraalin rajoissa. Empiirisen osion tulokset vastasivat hyvin pitkälti kirjallisuuskatsauksen aikana esitettyjä aiempien tutkimusten ja teorioiden tuloksia. Tämän perusteella tulosten voidaan todeta olevan käytettäviä jatkossa ja ne puoltavat yleistä linjaa, jonka mukaan Bug Bountysta on hyötyä yrityksille. Bug Bountyn käyttö on turvallista ja se tuo positiivista mainetta yrityksille. Tulokset ovat myös luotettavia.

Avainsanat: Bug Bounty, joukkoistaminen, tietoturva, CIA, penetraatiotestaus.

ABSTRACT

Jarnola, Miikka
Bug Bountyn hyödyt tietoturvatestauksessa tapaustutkimus - LähiTapiola
Jyväskylä: University of Jyväskylä, 2018, 46 p.
System development/Information security, Information Systems, Master's Thesis
Supervisor: Siponen, Mikko

Purpose of this study was to produce scientific data of benefits and disadvantages of Bug Bounty. With help of this study we wanted to open new research field in the information security research area. Second purpose of this study was to show companies that using Bug Bounties is safe. This research had three research questions. They were: what benefits LocalTapiola had with Bug Bounty, what risks and problems they had and what they learned and took into action. This study was executed as combination of empirical study and literature review. Results were quite unanimous. According to interviewees Bug Bounty has definitely helped company to produce better information security to its applications. Benefits was mentioned positive publicity for the company, freed resources and concrete decrease of vulnerabilities. There weren't any clear problems with Bug Bounty. LocalTapiola had identified some potential risks. These risks didn't realize. These risks were negative publicity, service unavailable because of too many users (DoS) and negative feedback. These risks were identified general risks, so they apply all companies. During research some disadvantages was recognized. These disadvantages can be happening to all companies. Recognized disadvantages were slowdown of services during one day after releasing Bug Bounty, long process from finding vulnerability before it was fixed and language barrier between some foreign countries. Lessons learned were mentioned new way of thinking and seeing information security and understanding it better. Interviewees mentioned that their vocational skills were developed and that internationally differences between ethics and moral were huge. Results of empirical section were quite close to earlier theories and studies presented in literature review. Based on this it can be said that results are beneficial in the future and they are trustworthy. Results just makes general concept stronger that Bug Bounty gives multiple benefits to companies and it is very recommendable to use. Bug Bounty brings positive reputation to the companies.

Keywords: Bug Bounty, crowdsourcing, information security, CIA, penetration testing.

KUVIOT

KUVIO 1 Tietoturvakolmio (Le Roux, 1993, 53-56, mukaillen) 12
KUVIO 2 Ohjelmistokehityksen elinkaarimalli (Braude & Bernstein, 2016, 1). . 16
KUVIO 3 Turvallisen ohjelmistokehityksen malli (Microsoft, 2018) 24
KUVIO 4 How a Bug Bounty Works (HackerOne, 2016, How Bug Bounties Work: A Comic) 35

TAULUKOT

TAULUKKO 1 Haastateltujen määrä organisaatioittain 30
TAULUKKO 2 Bug Bounty palkkioiden luokittelun havainnollistaminen 33
TAULUKKO 3 Bug Bounty ”All time metrics” marraskuu 2018 LähiTapiola ... 42
TAULUKKO 4 LähiTapiolan Bug Bounty raporttien suhde 42
TAULUKKO 5 Yleisimmät haavoittuvuustyypit LähiTapiolan Bug Bountyssa 42

SISÄLLYS

TIIVISTELMÄ 2

ABSTRACT 3

KUVIOT 4

TAULUKOT 5

SISÄLLYS 6

1 JOHDANTO 8
1.1 Tutkimusongelma ja tutkimuskysymykset 8
1.2 Motivaatio 8
1.3 Tutkimusmenetelmät 9
1.4 Saavutetut tulokset ja niiden merkitys 10
1.5 Keskeiset käsitteet 10
1.5.1 Bug Bounty 10
1.5.2 Joukkoistaminen 10
1.5.3 Joukkoistetun penetraatiotestauksen työkalut 11
1.5.4 Tietoturvatestaus / penetraatiotestaus 11
1.5.5 Tietoturvahaavoittuvuus 11
1.5.6 Defekti 11
1.5.7 CIA-kolmio / tietoturvakolmio 11

2 AIEMPI TUTKIMUS 13

3 VALITUT TEORIAT 15
3.1 Ohjelmistotuotanto 15
3.2 Laadunvarmistus 17
3.3 Joukkoistaminen 18
3.4 Standardit 19
3.5 Standardeihin kohdistunutta kritiikkiä 20
3.6 Joukkoistettu penetraatiotestaus aka. Bug Bounty 21
3.7 Turvallinen ohjelmistokehitys 22
3.8 Turvallisen ohjelmistokehityksen malli (SDLC) 23
24
3.9 Bug Bounty tyypit 25
3.10 Turvallisten tietojärjestelmien tutkimuksen historia ja tulevaisuus ..25
3.11 Viitekehys 26

4 TUTKIMUSMENETELMÄT 28
4.1 Tutkimuksen suunnittelu 28

4.2 Tiedonkeruumenetelmät 29
4.3 Haastateltavien valinta 30
4.4 Datan analysointi 30

5 TUTKIMUSTULOKSET 32
5.1 LähiTapiola 32
5.2 Miten Bug Bounty yleensä toimii 32
5.3 LähiTapiolan Bug Bounty 36
5.4 Hyödyt 37
5.5 Haasteet 38
5.6 Riskit 41
5.7 Tilastoja 41
5.8 Mitä opittiin ja vietiin käytäntöön 42

6 JOHTOPÄÄTÖKSET 44
6.1 Vastaukset tutkimuskysymyksiin 44
6.2 Tulosten merkitys, luotettavuus & käytettävyys 45
6.3 Rajoitukset & jatkotutkimusaiheet 46

7 YHTEENVETO 47

LÄHTEET 49

LIITE 1 53

1 JOHDANTO

Tämä pro gradu tutkielma käsittelee Bug Bountya ja sen hyötyjä LähiTapiolalle, sekä muille yrityksille. Tutkimuksen tulokset ovat hyödynnettävissä kaikille yrityksille ja toimialoille. Idea tutkimuksesta syntyi, koska aihetta on tutkittu todella vähän ja siitä kaivataan tieteellistä tutkimusta. Yrityksillä on selkeä tarve saada laajempaa tietoa Bug Bountysta ja sen hyödyistä, sekä haitoista. Tämän tutkimuksen avulla yritykset voivat suunnitella paremmin Bug Bountyn käyttöä. Tutkimus auttaa myös ymmärtämään Bug Bountyn hyötyjä paremmin. LähiTapiolan johto haluaa tieteellistä dataa tutkimuksen hyödyistä ja haitoista, jotta he voivat tehdä päätöksiä jatkon suhteen. Tutkimuksen toinen tarkoitus on osoittaa muille yrityksille Bug Bountyn hyötyjä sekä, että sen käyttö on turvallista. Tämä tullaan osoittamaan kartoittamalla Bug Bountyn riskejä ja hyötyjä. Hyötyjen osoittaminen tullaan tekemään aiheesta jo tehtyjen tutkimusten avulla.

1.1 Tutkimusongelma ja tutkimuskysymykset

Tämän pro gradun tutkimusongelma on seuraava: Mitä hyötyjä LähiTapiolalle on ollut Bug Bountyn käyttöönotosta? Tämä tutkimusongelma toimii myös pääasiallisena tutkimuskysymyksenä. Osaongelmia ovat seuraavat kaksi kysymystä: Mitä ongelmia ja riskejä Bug Bountysta on ollut LähiTapiolalle? Mitä opittiin ja mitä vietiin käytäntöön?

1.2 Motivaatio

Teknologian yleistyminen ja laajeneminen on tapahtunut hyvin nopeasti. Kaikki yritykset haluavat olla mukana kehityksessä. Valitettavasti tietoturva uusissa sovelluksissa ja laitteissa ei ole kovinkaan hyvällä tasolla monesti. Kasvavan tietoisuuden myötä yritykset ovat heränneet tarpeeseen panostaa tietoturvaan. He haluavat asiakkaidensa olevan turvassa ja luottavan heihin. Tämä tietoisuus

on luonut uusia avauksia tietoturvan varmistamiselle. Yksi näistä tavoista on tietoturvatestauksen ulkoistaminen joukkoistamalla eli Bug Bounty.

Bug Bounteista on maailmalla tehty toistaiseksi hyvin vähän tutkimusta. Kyseessä on kasvava trendi, joten tutkimuksen tarve on suuri. Monet yritykset karsastavat Bug Bountyn käynnistämistä. Tämä johtunee siitä, että niiden hyötyjä ei ole tutkittu hirveästi ja on epävarmaa, onko niistä hyötyä yrityksille. Tämän tutkimuksen on tarkoitus tuoda esille Bug Bountyn hyötyjä, sekä sitä kuinka Bug Bountya voidaan hyödyntää eri yrityksissä. Bug Bountyn parista aiempaa tutkimusta löytyy lähinnä siitä, kuinka hyvin valkohattuhakkerit ovat kiinnostuneet kyseisiin ohjelmiin osallistumisesta. (Chen, Grossklags, Zhao, 2014). Chen & kumppaneiden tutkimuksen mukaan hakkerit osallistuvat mielellään ohjelmiin, jos kannustimet ovat kunnossa. (Chen, Grossklags, Zhao, 2014).

Itse Bug Bountyn hyödyistä on tehty lähinnä tutkimuksia, joissa vertaillaan mm. Firefoxin ja Chromen palkinto-ohjelmia. Tämä tutkimus ei kuitenkaan kerro ohjelmien hyödyistä yrityksille suoranaisesti. (Finifter, Akhawe, Wagner 2013.). Tämä tiivistää aika hyvin Bug Bountyihin suoraan kohdistuvan olemassa olevan tutkimuksen ja puoltaa näin ollen selkää avausta tälle uudelle ja nousevalle trendin tutkimiselle.

Joukkoistaminen on Bug Bountyn perusta. Joukkoistamisen hyödyistä on olemassa aiempia tutkimuksia. Mm. Murturi, Kantarci ja Oktug (2015) ovat tutkimuksessaan kuvanneet mallin, kuinka joukkoistaminen toimii. Myös joukkoistamista penetraatiotestauksessa on tutkittu aiemminkin. Mm. Krishnamurthy ja Tripathi (2006).

Tämän tutkimuksen tavoitteena on siis tuottaa selkeä kuvaus yrityksille Bug Bountyn hyödyistä, haitoista ja haasteista. Tutkimuksen avulla avataan myös uutta tutkimuskenttää Bug Bountyn parista. Tutkimuksen luettuaan lukijalla on selkeä kuva siitä, miten Bug Bountya voidaan hyödyntää tietoturvatestauksen varmistavana tekijänä. Lukijalle on myös muodostunut kuva siitä mitä tutkimusta ja muuta tietoa aiheesta jo on. Kiinnostuneet saavat myös ideoita jatkotutkimusaiheiksi.

1.3 Tutkimusmenetelmät

Tutkimus toteutetaan tapaustutkimuksena. Tarkemmin sanottuna kyseessä on single-case study. Itse tutkimusmenetelmänä toimii laadullinen tutkimusmenetelmä. Jotta tutkimus voidaan tehdä halutulla tavalla käytetään siinä tulkitsevan ja selittävän case-metodin yhdistelmää. Datan keruu tapahtuu haastatteluiden avulla ja dokumentteja analysoimalla.

1.4 Saavutetut tulokset ja niiden merkitys

Saavutetut tulokset vastaavat kohtuu hyvin tässä tutkielmassa esitettyjen aiempien tutkimusten tuloksia ja teorioiden olettamuksia. Näin ollen niitä voidaan pitää käytettävinä ja merkittävinä. Tulokset ovat merkityksellisiä LähiTapiolalle ja muille yrityksille, jotka haluavat ottaa Bug Bountyn käyttöönsä. Tulokset puoltavat näkemystä siitä, että Bug Bounty parantaa yrityksen tietoturvaa ja sen käyttö on turvallista. Tulosten mukaan yritys saa myös positiivista mainetta Bug Bountyn käytöstä.

1.5 Keskeiset käsitteet

Tämä alaluku selventää tutkimuksessa käytettyjä käsitteitä. Käsitteet avataan tarkemmin ja joidenkin osalta valotetaan myös sitä miksi ne ovat oleellisia tutkimuksen kannalta. Käsitteiden nimet kerrotaan englanniksi ja suomeksi. Itse tutkimuksessa käytetään suomenkielisiä termejä siltä osin, kuin se on mahdollista. Käytettyjen termien ja käsitteiden määritelmät on poimittu akateemisista julkaisuista.

1.5.1 Bug Bounty

Bug Bounty on yleinen nimitys haavoittuvuuspalkinto-ohjelmille. Termille Bug Bounty ei ole virallista suomennosta. Bug Bounty tai haavoittuvuuspalkinto-ohjelma on yrityksen tai jonkin muun tahon järjestämä ohjelma. Kyseisen ohjelman ansiosta kaikki halukkaat voivat ennalta määrättyjen sääntöjen puitteissa osallistua yrityksen järjestelmien tietoturvatestaukseen. Yleensä osallistujat ovat teknisen taustan omaavia ihmisisä. Nämä ihmiset tunnetaan paremmin hakkereina. (Finifter, Akhawe, Wagner, 2013.). Sanan ohjelma kanssa on oltava tarkkana. Bug Bountyn yhteydessä se viittaa toimintaan vrt. kanta-asiakkuusohjelma. Ohjelma sanalla ei ole mitään tekemistä tietokoneen tai älypuhelinten sovellusten kanssa tässä yhteydessä. Bug Bounty on tärkeä käsite tutkimukselle, koska se muodostaa koko tutkimuksen ytimen.

1.5.2 Joukkoistaminen

Joukkoistaminen -termillä tarkoitetaan tavallisten ihmisten osallistamista tietyn asian tekemiseen. Termin englanninkielinen nimitys on ”crowdsourcing”. Tässä tutkielmassa joukkoistamiseen osallistuvat henkilöt ovat useimmiten hakkereita tai muita it-alan asiantuntijoita. Hammon & Hippner määrittelevät joukkoistamisen artikkelissaan seuraavasti: ”Joukkoistaminen on sarja yrityksen tai sen palveluntuottajan suorittamia tehtäviä, jotka tämän toimintamallin sijaan suorittaakin nyt yrityksen ulkopuoliset henkilöt. Tehtävät tulevat yleensä tarjolle Internetin välityksellä ja halukkaat voivat niitä tehdä”. (Hammon & Hippner,

2012, 163).

1.5.3 Joukkoistetun penetraatiotestauksen työkalut

Bugcrowd & HackerOne ovat yrityksiä, jotka tarjoavat alustoja haavoittuvuus-palkinto-ohjelmien toteuttamiseen. (Bugcrowd & HackerOne, 2016). Näiden alustojen avulla yritykset voivat hallinnoida Bug Bountyn toteuttamista. Alustan avulla voidaan jakaa hakkereille tietoa ohjelmasta. Yritykset laittavatkin usein alustalleen ohjeet siitä mitä saa testata ja miten. Tätä samaa kanavaa pitkin hakkerit voivat jättää raportteja löytämistään virheistä. (Bugcrowd & HackerOne, 2016.).

1.5.4 Tietoturvatestaus / penetraatiotestaus

Tietoturvatestauksella tarkoitetaan järjestelmän tai sovelluksen tietoturvan testaamista käyttäen siihen määriteltyjä työkaluja. Tietoturvatestauksen avulla järjestelmien heikkoudet ja haavoittuvuudet pyritään löytämään ennen, kuin niitä käytetään hyödyksi rikollisiin tarkoituksiin. (Tang, 2014.).

1.5.5 Tietoturvahaavoittuvuus

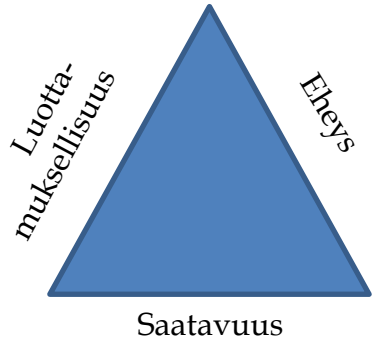
Tietoturvahaavoittuvuudella tarkoitetaan tietojärjestelmässä tai yksittäisessä ohjelmassa olevaa ohjelmointivirhettä. Tämä kyseinen virhe mahdollistaa luvattoman pääsyn järjestelmään. (Brauch ym., 2011.).

1.5.6 Defekti

Defekti tarkoittaa ohjelmointivirhettä ohjelman koodissa. Edellä mainittu virhe aiheuttaa ohjelman toimimattomuuden tai sen, että ohjelma ei toimi oikein. (Avizienis, Laprie, Randell, Landwehr 2004.). Tämä on tärkeä käsite, koska defektejä pyritään löytämään tietoturvatestauksella.

1.5.7 CIA-kolmio / tietoturvakolmio

CIA tarkoittaa tiedon luotettavuutta / luottamuksellisuutta, eheyttä, ja saatavuutta (Confidentiality, Integrity, Availability). Näiden ehtojen täyttyessä yrityksen tietoturva on lähtökohtaisesti kunnossa. (Le Roux, 1993.). Jatkossa k tästä termistä ja kuviosta käytetään nimitystä ”tietoturvakolmio”. Seuraava kuvio hahmottaa tietoturvakolmion rakennetta (kuvio 1). Kuvio on piirretty Le Rouxin artikkelin pohjalta.



KUVIO 1 Tietoturvakolmio (Le Roux, 1993, 53-56, mukaillen)

Tutkielma etenee seuraavasti. Ensin esitellään aiheen parista tehtyä tutkimusta ja annetaan lukijalle yleiskuva siitä mitä on tukittu ja mitä ei ole tutkittu. Seuraava luku hahmottelee tässä tutkimuksessa käytetyn teoriapohjan aiempien tutkimusten pohjalta. Seuraavaksi käsitellään empiiristä tutkimusta. Siinä käydään läpi tarkemmin mitä tehtiin, miten tehtiin ja miksi tehtiin. Loppupuolen luvut esittelevät tulokset ja johtopäätökset. Pro gradun lopuksi nivotaan yhteen tutkimuksen tärkeimmät asiat.

2 AIEMPI TUTKIMUS

Tässä luvussa esitellään lyhyesti Bug Bountyn ja siihen kiinteästi liittyvien tutkimusalueiden aiempia tutkimuksia. Esiteltävien tutkimusten avulla luodaan yleiskuva tehdyistä tutkimuksista ja kerrotaan lukijalle nykytilanne. Esiteltäviksi tutkimuksiksi on valittu sellaiset tutkimukset, jotka antavat hyvän yleiskuvan aihealueesta. Tämä luku luo myös pohjan seuraavalle luvulle, jossa esitellään tämän tutkimuksen teoriapohja.

Suomessa tai suomen kielellä tehtyjä tutkimuksia Bug Bountysta ja sen hyödyistä ei toistaiseksi ole olemassa. LähiTapiola on ensimmäisiä yrityksiä Suomessa, joka käyttää kyseistä ohjelmaa. LähiTapiola aloitti Bug Bountyn käytön 2015. (LähiTapiola, 2015.). Visma lanseerasi oman Bug Bountyn ohjelmansa 2016 (Visma, 2016). Tämä selittänee suomenkielisten tutkimusten puutetta. Näiden yritysten lisäksi Suomessa Bug Bounty on käytössä mm. Verolla, Bonus Waylla ja Väestörekisterikeskuksella. (Niemelä, 2018). Yhdysvalloissa ja maailmalla on tehty tutkimusta ohjelmien hyödyistä yrityksille (Akhawe, Finifter, Wagner, 2013). Tätäkin enemmän on tutkittu kyseisten ohjelmien kiinnostavuutta / houkuttavuutta valkohattuhakkereiden keskuudessa. (Chen, Grossklags, Zhao, 2014). Ulkomailla aihetta on myös tutkittu penetraatiotestauksen näkökulmasta (Schulz, 2014).

Finifter ym. on tehnyt tutkimusta yleisesti haavoittuvuuspalkinto–ohjelmista ja niiden käytöstä. Tutkimuksessaan he käsittelevät haavoittuvuuspalkinto-ohjelman hyötyjä Firefox ja Chrome selainten kehitykseen liittyen. Heidän löytämiään tuloksia käytetään vertailudatana tämän tutkimuksen osalta. (Finifter, Akhawe, Wagner 2013.). Täytyy toki muistaa, että LähiTapiolan kehitys tapahtuu suljetussa ympäristössä, kun taas esimerkiksi Firefox pohjautuu avoimeen lähdekoodiin. Ram Chillarege on tehnyt listauksen 28 parhaasta käytännöstä liittyen ohjelmistotestaukseen (Chillarege, 1999). Nämä käytänteet ovat edelleen valideja. Tässä tutkimuksessa niitä tullaan hyödyntämään kuvattaessa laadunvarmistuksen prosessia teoriassa. Gordon Schulmeyer on vastaavasti julkaissut kirjan nimeltä ”Handbook of Software Quality Assurance”. (Schulmeyer, 2007.). Tätä teosta hyödynnetään myös laadunvarmistuksen teoriapohjan määrittelyssä.

Joukkoistamiseen liittyen on tehty tutkimuksia laajemmin, kuin Bug Bountyyn liittyen. Näiden tutkimusten pohjalta on hyvä lähteä rakentamaan tämän tutkielman teoreettisia lähtökohtia. Murturi, Kantarci ja Oktug (2015) ovat tutkimuksessaan kuvanneet mallin, kuinka joukkoistaminen toimii. Samoin ovat tehneet myös Zhang ja Zhang (2011). Näiden mallien avulla kuvataan joukkoistamisen periaatteet seuraavassa luvussa. Arkin, Stender ja McGraw (2005) ovat kirjoittaneet selkeän dokumentin siitä, miten tietoturvatestaus ilmenee nykypäivän laadunvarmistuksessa. Tämän dokumentin avulla voidaan verrata LähiTapiolan tapaa hoitaa asiat yleisiin käytänteisiin nähden. IEEEn (2014) dokumentti laadunvarmistusprosessien standardeista täydentää seuraavassa luvussa käsiteltävää ISO standardien kokoelmaa.

Krishnamurthy ja Tripathi (2006) ovat tutkineet ”joukkoistettua penetraatiotestausta” avoimen ohjelmistokehityksen parissa. Tämä tutkimus antaa hyvän kuvan siitä, miten ohjelmoijia ja testaajia voidaan motivoida tekemään parempia ohjelmistoja. Tietoturvatestauksen ja ennen kaikkea ”joukkoistetun penetraatiotestauksen” ympärille on muodostunut myös mustan pörssin markkinat. Näillä markkinoilla on myynnissä löydettyjä haavoittuvuuksia. Virallisempi nimitys kyseisille markkinoille on ”Markets for Zero-Day Exploits”. Vapaasti suomennettuna nollapäivähaavoittuvuuksien markkinat. Kyseisillä markkinoilla toimivat tekijät kilpailevat niiden yritysten kanssa, jotka käyttävät ”joukkoistettua penetraatiotestausta” haavoittuvuuksien löytämiseksi. Yritykset pyrkivät joukkoistamalla löytämään haavoittuvuudet ennen, kuin rikolliset toimijat ehtivät ostaa niitä markkinoilta ja hyödyntää ostamiaan haavoittuvuuksia. (Egelman, Herley ja Oorschot, 2013.).

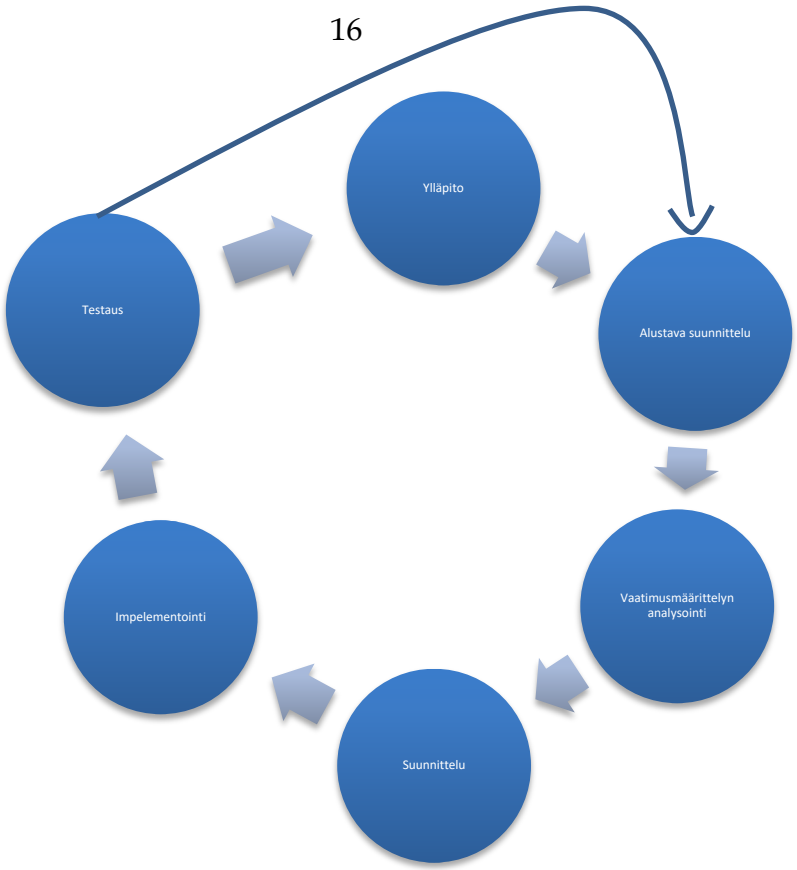
3 VALITUT TEORIAT

Tässä luvussa esitellään erilaisia teorioita Bug Bountyn, joukkoistamisen, laadunvarmistamisen ja ylipäänsä ohjelmistotuotannon saralta. Nämä teoriat muodostavat tutkimuksen tieteellisen pohjan. Tulevissa alaluvuissa hahmotellaan viitekehys tälle tutkimukselle. Viitekehyksen luominen aloitetaan kuvaamalla, kuinka ohjelmistotuotannon prosessi teoriassa etenee. Seuraavaksi kuvataan mitä laadunvarmistus on teoriassa ja kuinka se tulisi toteuttaa kirjan oppien mukaisesti.

Luvun keskivaiheilla esitellään ISO-tuoteperheen standardeja, joita käytetään laadunvarmistuksessa ja tietoturvan tuottamisessa. Kappaleen loppupuolella kuvataan vielä joukkoistaminen, joka on tärkeä osa tätä tukimusta. Tämän jälkeen lukijalle kerrotaan miksi juuri nämä teoriat valikoituivat tähän tutkimukseen. Samalla tuodaan esille, miten nämä tutkimukset ja teoriat löytyivät. Lopussa nivotaan yhteen luvun aikana luotu teoriapohja.

3.1 Ohjelmistotuotanto

Brauden ja Bernsteinin ohjelmistokehityksen elinkaarimalli kuvaa selkeästi ja yksinkertaisesti ohjelmistotuotannon / ohjelmistokehityksen prosessin (kuvio 2).



KUVIO 2 Ohjelmistokehityksen elinkaarimalli (Braude & Bernstein, 2016, 1).

Ohjelmistotuotanto muodostaa perustan ohjelmistokehitykselle. Näin ollen on luontevaa esitellä ensimmäisenä ohjelmistokehitystä ja jatkaa teoriapohjan luontia muilla teorioilla sen jälkeen. Brauden ja Bernsteinin mukaan ohjelmistokehitys noudattaa yllä esitettyä mallia. Kaavio lähtee liikkeelle alustavasta suunnitelmasta ja etenee nuolien osoittamassa suunnassa. Tässä ensimmäisessä vaiheessa tehdään alustava suunnitelma tarvittavasta tai halutusta tietokoneohjelmasta. Tietokoneohjelmalla tarkoitetaan tässä joko sovellusta tai käyttöjärjestelmää. Jatkossa termiä sovellus käytetään kuvaamaan tietokoneohjelmaa. Tämä ensimmäinen vaihe on myös uuden sovelluksen ideointia. (Braude & Bernstein, 2016.).

Jotta tämä malli pysyy yksinkertaisena, kuvataan tätä vain kahdella ohjelmistokehityksen toimintamallilla. Ensimmäinen malli on ohjelmistotalo, joka tekee asiakkailleen ohjelmistoja. Nämä ohjelmistot voivat olla valmisohjelmistoja tai räätälöityjä ohjelmistoja. Toinen malli on yritykset, jotka kehittävät ohjelmistoja omiin tarpeisiinsa. Tällaisia ohjelmistoja yritys käyttää sisäisesti tai sitten heidän asiakkaansa käyttävät niitä. Sisäiseen käyttöön tarkoitettu ohjelma voi olla esimerkiksi työajanseurannan työkalu. Asiakkaille tarkoitetusta ohjelmasta tai ohjelmistosta hyvä esimerkki on LähiTapiolan asiakkailleen Internetin kautta tarjoamat vakuutuspalvelut. Muun muassa itsepalveluportaali. Kun alustava suunnitelma tai idea on valmis, siirrytään toiseen vaiheeseen. Tässä vaiheessa analysoidaan vaatimusmäärittelyt. Tämä tarkoitta sitä, että selvitetään mitä vaatimuksia tulevalla ohjelmalla tai järjestelmällä tulee olla. Nämä

vaatimukset tunnetaan myös nimellä käyttötapaukset. (Braude & Bernstein, 2016.).

Kolmannessa vaiheessa aloitetaan varsinainen ohjelman suunnittelu ja määrittely. Tässä vaiheessa tehdään kehitettävän sovelluksen ohjelmointia valituilla menetelmillä ja ohjelmointikielillä. Implementointivaiheessa tehty sovellus siirretään testausympäristöön, jossa sitä voidaan huoletta testata. Jos kyseessä on jatkokehitys olemassa olevalle ohjelmistolle, tässä vaiheessa uusi tehty sovellus liitetään osaksi olemassa olevaa ohjelmaa tai ohjelmistoa. Viides vaihe on testausvaihe. Tässä vaiheessa ohjelmiston laatua testataan käyttötapausten pohjalta luotujen testitapausten avulla. Sovellukselle voidaan tehdä myös tutkivaa eli exploratiivista testausta. Tämä tarkoittaa sitä, että testaaja kokeilee erilaisia asioita. Hänen tavoitteena on saada sovellus mahdollisimman solmuun. Jos hän ei saa ohjelmaa rikki on se suhteellisen laadukas. Viimeinen vaihe on ylläpito. Tässä vaiheessa luotu sovellus siirretään ylläpitovaiheeseen. Tässä vaiheessa ohjelmisto on tuotannossa. Tuotannossa olevaa ohjelmistoa tulee ylläpitää muun muassa päivittämällä sitä. Sekä korjaamalla siitä löytyneitä virheitä eli bugeja. Nämä toimenpiteet suoritetaan myös tämän elinkaarimallin avulla. (Braude & Bernstein, 2016.).

Tässä esitetty malli antaa selkeän kuvan ohjelmistokehityksen prosessista. Mallin avulla voidaan tarkastella toimiiko kohdeyritys yleisesti hyväksytyn mallin mukaisesti omassa ohjelmistokehitystoiminnassaan. Tämä ylätason malli auttaa myös hahmottamaan mitä tutkimusalueen ja ohjelmistokehityksen osa-aluetta tässä pro gradu tutkielmassa tarkastellaan tarkemmin. Tarkemman tarkastelun kohteeksi pääsee viides vaihe, eli testaus, josta käytetään myös nimeä laadunvarmistus. Testauksen alta fokusoidaan tutkimukseen vain pieni osa, eli ”joukkoistettu penetraatiotestaus”.

3.2 Laadunvarmistus

Laadunvarmistus tunnetaan paremmin sen englanninkielisellä nimellään ”quality assurance. Testaus taas on menetelmä, jonka avulla ohjelmiston laatua voidaan varmistaa. (Ammann & Offutt, 2008.). Kuten aiemmin on mainittu tässä tutkimuksessa, laadunvarmistus on osa ohjelmistokehityksen elinkaarta. Testaus pitää sisällään neljä vaihetta. Nämä vaiheet ovat hyväksymistestaus, järjestelmätestaus, integraatiotestaus, moduulitestaus ja yksikkötestaus. (Amman & Offutt, 2008.). Penetraatiotestausta voidaan suorittaa jokaisen edellä mainitun vaiheen aikana. Tämä tutkimus keskittyy enimmäkseen tuotannossa olevan ohjelmiston penetraatiotestaukseen. Tämän tutkimuksen kannalta toinen oleellinen kohde tietoturvatestaukselle on hyväksymistestauksen aikana tehtävä penetraatiotestaus.

Chillarege kuvaa teoksessaan laadunvarmistuksen peruskäytänteet, joita tulisi noudattaa testauksen yhteydessä. Nämä ovat: toiminnalliset määrittelyt, katselmoinnit & tarkastukset, muodolliset aloitus ja lopetuskriteerit, toiminnallinen testaus, monella alustalla tapahtuva testaus, sisäiset betatestaukset, testiautomaatio, beta-ohjelmat ja yölliset tuotantoon otot. (Chillarege, 1999.). Kirjoit-

taja esittelee teoksessaan myös käytänteitä, jotka hänen mukaansa luovat vahvan perustan hyvälle laadunvarmistukselle. Näitä käytänteitä noudatetaan kuitenkin valitettavan harvoin. Tämän tutkielman kannalta niistä oleellisia ovat käyttötapaukset, testisuunnitelma ja käytettävyystestaus. (Chillarege, 1999.). Näiden lisäksi hän mainitsee niin kutsuttuja ”erityistyökaluja” erikoisempiin tilanteisiin. Teoriapohjan luomisen kannalta oleellisia erityistyökaluja ovat testaajien ja ohjelmoijien yhteistyö & Bug Bountyt. Jotta saadaan aikaan hyvä sovellus testaajien ja ohjelmoijien on tehtävä tiivistä yhteistyötä.

Chillarege mainitsee dokumentissaan kaiken kaikkiaan 28 hyvää käytäntöä liittyen laadunvarmistukseen. Edellä mainitut käytänteet ovat tämän tutkimuksen kannalta tärkeimmät. Vaikka dokumentti on useamman vuoden vanha, iso osa käytänteistä on edelleen käytössä. Näiden käytänteiden avulla voidaan määrittää kuinka laadunvarmistus tulisi yrityksessä hoitaa. Määritelmän avulla voidaan luoda perusteet sille, miten laadunvarmistuksen tulisi yrityksessä toimia.

3.3 Joukkoistaminen

Joukkoistamista on tutkittu jo pidempään. Tässä luvussa esitellään ja analysoidaan tämän tutkimuksen kannalta oleellisimpia aiheeseen liittyviä tutkimuksia. Murturi & kumppanit määrittelevät ”joukoistamiselle” viitteellisen mallin. Tämän mallin avulla ”joukkoistamista” voidaan hyödyntää palveluna. Englanniksi ”CaaS” Crowdsourcing as a service. (Murturi ym. 2015.). Tämä toimii samankaltaisesti, kuten muutkin XaaS –palvelumallit. ”Joukkoistaminen palveluna korvaa yrityksen perinteisen liiketoimintamallin”. Yksinkertaistettuna yritys käyttää ennalta valitsemaansa alustaa. Tälle alustalle yritys laittaa tarvittavat palvelut ja ohjeet siitä, kuinka ”joukkoistettua” –toimintaa harjoittavien ihmisten tulee toimia. Sovellettuna ”joukkoistettuun penetraatiotestaukseen” –esimerkkinä toimii seuraava: Yritys laittaa alustalle ohjeet siitä mitä järjestelmiä saa testata ja millä työkaluille. Yritys myös kertoo miten ja minne bugit raportoidaan. Näin yrityksellä on käytössään ”joukkoistaminen palveluna”. (Murturi ym. 2015.). Tätä mallia hyödynnetään tämän tutkimuksen teoriapohjassa. Mallin avulla hahmotellaan LähiTapiolan Bug Bounty –ohjelman toiminta käytännössä.

Doan ja kumppanit ovat tehneet tutkimusta ”joukkoistamisesta Internetissä”. He esittelevät ”joukkoistamisen” neljä avainhaastetta. Nämä ovat: kuinka rekrytoidaan osallistujia, mitä he voivat tehdä, miten yhdistetään heidän tukensa ja miten estetään väärinkäytökset. (Doan ym. 2011.). Kirjoittajien mukaan joukkoistamisen kohteesta pitää tehdä kiinnostava ja haastava. Yrityksen tulee määrittää mitä osallistujat saavat tehdä ja mitä he eivät saa tehdä. Yhteisenhengen luominen ja osallistujien työn yhdistäminen voi myös olla haastavaa. Yhteishenki syntyy yleensä siitä, että osallistujat ovat saman henkisiä ja haluavat samoja asioita. Työpanoksen tasapainottaminen vaatii selkeitä suunnitelmia siitä, kuka tekee ja mitä tekee. (Doan ym. 2011.). Kirjoittajat myös määrittelevät ”joukkoistamisen järjestelmän” (Crowdsourcing system). Vapaasti suomen-

nettuna määritelmä on heidän mukaansa seuraavanlainen: ”Joukko saman henkisiä ihmisiä yhdistää voimansa yhteisen tavoitteen saavuttamiseksi. Nämä ihmiset eivät rakenna mitään fyysistä esinettä. He vain tuovat tietotaitoaan projektiin. Tähän liittyvä yhteisö on olemassa vain sen hetken, kun kyseinen toiminta kestää”. (Doan ym. 2011.).

”Joukkoistamisesta” voidaan mainita hyvänä esimerkkinä Jolla C -älypuhelin. Jolla teki reilun 1000 kappaleen erän tätä matkapuhelinta. Se myytiin halukkaille ja nämä ostajat kuuluvat nyt Sailfish OS yhteisöön (Sailfish OS community). Tämän ”joukon” tarkoitus on rahoittaa Jollan puhelinta ja ohjelmistokehitystä. Samalla kyseisen puhelimen ostaneet sitoutuvat testaamaan sitä ja laatimaan raportteja Jollalle. (Jolla, 2016.). Edellä esiteltyjen ”joukkoistamisen” tutkimusten pohjalta on helppo lähteä rakentamaan perusteita tälle tutkimukselle. Murturin & kumppaneiden esittämän mallin pohjalta voidaan kuvata miten ”joukkoistamisen” tulisi toimia teoriassa. Tähän peilataan LähiTapiolan toimintaa. Doanin & kumppaneiden tutkimuksen pohjalta määritellään, miten ihmisiä tulisi osallistaa/rekrytoida ”joukkoistamiseen”.

3.4 Standardit

Tämä luku esittelee ja analysoi tutkimuksen kannalta lyhyesti oleelliset standardit laadunvarmistuksen ja tietoturvan saralta. ISO 27001 standardi käsittelee tietoturvallisuuden hallintaa. Sen avulla voidaan määritellä yrityksen tietoturvan perusteet. Näitä ovat mm. tietoturvakäytännöt ja tietoturvapolitiikka. 27001 standardiperheen avulla voidaan määritellä kenellä on pääsy mihinkin järjestelmään, kuka hallitsee tunnuksia jne. Kun yritys täyttää tämän standardin vaatimukset voivat he hakea ISO 27001 sertifikaattia yritykselleen. ISO 27002 standardi taas täydentää ISO 27001 standardia. Se antaa myös tarkempia ohjeistuksia siitä, miten tuon ylätason standardin vaatimukset täytetään. ISO 27031 standardin avulla määritellään yrityksen valmiudet ylläpitää liiketoiminnan jatkuvuutta ja valmiutta palautua ongelmatilanteista. (Gikas, 2010.).

ISO 22301 standardi taas määrittää vaatimukset sille, miten yrityksen tulee suunnitella, muodostaa, implementoida, toimia ja valvoa dokumentoituja järjestelmiä, jotta he voivat mahdollisimman nopeasti ja tehokkaasti reagoida ilmenneisiin ongelmiin. ISO 22313 tukee myös ISO 22301 standardin määrittelemiä asioita. Se myös tarkentaa ISO 22301 standardissa määriteltyjä asioita. (Gikas, 2010.).

Viimeisenä tämän tutkimuksen teorian kannalta merkittävänä mallina mainittakoon The System Security Engineering Capability Maturity Model (SSE-CMM). Tämän mallin avulla voidaan arvioida ja havaita tietoturvallisuuteen liittyviä ongelmia ohjelmistokehityksessä. Malli koostuu 22 avainprosessista ja kuudesta maturiteettitasosta. Edellä mainituista 22 prosessista 11 on tietoturvaan liittyviä, Nämä ovat: tietoturvakontrollien hallinta, vaikutusten arviointi, tietoturvariskien arviointi, uhkien arviointi, haavoittuvuuksien arviointi, argumenttien määrittäminen, tietoturvan koordinointi, tietoturvan toteutumisen valvonta, tietoturvaan liittyvän syötteen toimittaminen, tietoturvan tarpeiden

määrittäminen ja tietoturvan todentaminen ja varmistaminen. (Siponen & Willison, 2009.).

Edellä mainittujen standardien lisäksi löytyy ”ISF Standard of Good Practices for Information Security” –niminen standardi. Tämä standardi on maailman kattavin tietoturvan standardi. Se on laajempi kuin ISO-standardit. Edellä mainittujen seikkojen lisäksi tämä standardi sisältää valmiudet reagoida alati muuttuviin uhkiin ja validoinnin tietoturvajärjestelyihin kolmannen osapuolen kanssa. Se myös lisää tietoisuutta tietoturvasta ja tuo datan suojausta pilvipalveluihin. (Siponen, 2006a.).

ISO standardien avulla voidaan määrittää erinomainen tietoturvan perusta yrityksille. Näiden standardien avulla yritykset voivat myös osoittaa olevansa sertifioituja tietoturvan suhteen. ISF Standard of Good Practices for Information Security” –standardi taas tuo yritykselle keinot kehittää sovelluksia tietoturvallisesti. Tämä standardi huomioi myös ulkoistetut palveluntuottajat, kuten ohjelmistotalot. Näiden standardien lisäksi jokaisella yrityksellä tulee olla oma tietoturvasäännöstö, joka täydentää ja tarvittaessa ajaa yli standardien säännöistä. Edellä esitetyt standardit ovat käytössä myös LähiTapiolassa.

Siponen käsittelee tutkimuksessaan myös standardien hyödyntämiseen liittyviä haasteita. Yleisesti oletetaan, että kaikkien yritysten tulisi ottaa käyttöön tietyt yleiset standardit. Olettamuksen mukaan ne tulisi ottaa samalla tavalla ja samoilta osin käyttöön. (Siponen, 2006b.). Siposen mukaan tässä tulee määritellä mitkä standardit ja vaatimukset sopivat kullekin yritykselle. Kun oikeat standardit ja vaatimukset on löydetty, tulee ne ottaa käyttöön sovelletusti. Hänen mukaansa väärin tehty suojaus väärässä paikassa aiheuttaa haavoittuvuuksia yrityksen avainprosesseille ja palveluille. Tämä aiheuttaa myös paljon hukkaan heitettyä rahaa. Esimerkiksi pienen asianajotoimiston ja kansainvälisen vakoiluorganisaation tietoturvaa suojataan täysin eri tavalla. (Siponen, 2006b.).

3.5 Standardeihin kohdistunutta kritiikkiä

Tässä luvussa käsitellään standardeihin kohdistettua kritiikkiä. Tässä esitellyn kritiikin avulla pystytään empiirisessä osiossa ja tuloksia käsittelevässä osiossa argumentoimaan paremmin tutkimuksen tulokset.

Standardeja on myös kritisoitu paljon. Tutkijoiden mukaan standardit keskittyvät varmistamaan tiettyjen tietoturvallisuuteen liittyvien prosessien tai aktiviteettien olemassaolon. Standardit eivät kuitenkaan useimmiten kerro kuinka niiden vaatimukset käytännössä saavutetaan. (Siponen, 2006a.). Siposen mukaan standardit keskittyvät varmistamaan niissä mainittujen prosessien olemassaolon. Kun niiden todellisuudessa pitäisi kertoa kuinka standardien ehdottamat tai vaatimat prosessit käytännössä saavutetaan. (Siponen, 2006a.). Siponen mainitsee myös standardien tehokkaan hyödyntämisen mittapuuksi, kuinka hyvin jokin asia on tehty. (Siponen, 2006a.).

Siponen kritisoi myös standardien sisältämien prosessien ja periaatteiden olevan hyvin ylätasolla. Hänen mukaansa standardeissa on myös paljon tulkin-

nan varaa. Esimerkkinä mainitaan BS ISO/ESCI7799:2000, p.11 kohta 6.2. User Training. Vapaasti suomennettuna ”Huolehditaan, että käyttäjä ovat tietoisia mahdollisista tietoturvaan liittyvistä uhkista ja he pystyvät tukemaan ja toteuttamaan organisaation tietoturvapolitiikkaa päivittäisessä työssään”. (Siponen, 2006a, 98). Tästä päästään Siposen esittämään kysymykseen ”vaikka yritys järjestäisi koulutussession ja laittaa tietoturvapolitiikan kirjallisessa muodossa saataville millä varmistetaan, että työntekijät oikeasti noudattavat niitä?” (Siponen, 2006a, 98).

Myös The Standard of Good Practice for Information Security kärsii samoista ongelmista. Siponen kyseenalaistaa tutkimuksessaan myös sen mistä tietää, että tietoturvapolitiikka on hyvä tai edes riittävän hyvä. (Siponen, 2006a.). Siponen nostaa esille myös erilaisten standardien moraaliset kysymykset. Jos jossain maassa esimerkiksi piratismi on sosiaalisesti hyväksyttyä saattaa kyseissä maassa standardia noudattavan yrityksen työntekijät pitää asiaa hyväksyttävän. Kansainvälisessä maailmassa yritysten tulee kuitenkin huomioida maiden rajojen yli tapahtuva yhteistyö. (Siponen, 2006a.).

Siposen ja Willisonin mukaan (2009) standardit ovat useasti myös geneerisiä ja niitä ei ole testattu, esimerkiksi käyttäen tieteellisiä tutkimusmenetelmiä. Jotta yritysten on helppo soveltaa standardeja ja saada niistä riittävä hyöty irti tulee standardien olla kohdistettu niitä käyttävien yritysten tarpeisiin. (Siponen & Willison, 2009.). Niemimaan & Niemimaan tutkimuksessa huomautetaan myös, että standardit tulisi muuntaa yrityksen tarpeisiin sopiviksi käytänteiksi ja dokumenteiksi. Tämä muuntaminen ja sen ymmärtäminen aiheuttaa kuitenkin useille yrityksille haasteita. (Niemimaa & Niemimaa, 2017.). ”Tietoturvaa ei välttämättä sovelleta niille alueille, joissa se on välttämätöntä ja meillä ei ole todisteita, että standardien suuntaviivat ovat luotettavia” (Siponen & Willison, 2009, 269). Tämä kommentti tiivistää mielestäni erittäin hyvin standardien ongelmia. Se on myös yksi niistä syistä, jonka takia LähiTapiola on halunnut panostaa tietoturvan kehittämiseen ja pyysi tätä tutkimusta.

3.6 Joukkoistettu penetraatiotestaus aka. Bug Bounty

Tässä luvussa esitellään ja analysoidaan joukkoistetun penetraatiotestauksen parista tehtyä tutkimusta ja teoriaa. Esiteltävät tutkimukset ja teoriat käsittelevät aihetta eri näkökulmista. Finifter & kumppanit ovat tutkineet ”joukkoistettua penetraatiotestausta” empiirisestä näkökulmasta. (Finifter ym. 2013). Heidän mukaansa Bug Bountysta, joka tunnetaan myös nimellä haavoittuvuuspalkinto-ohjelma (Vulnerability reward program) on yritykselle useita hyötyjä.

Merkittävimpinä hyötyinä tutkijat mainitsevat hyvän kannustimen joukkoistaa osaavia tietoturvatutkijoita aka. valkohattuhakkereita etsimään tietoturva-aukkoja heidän ohjelmistoistaan ja sovellustoimittajien tehokkaamman toiminnan haavoittuvuuksien hallintaan. Kolmantena hyötynä Finifiter & kumppanit mainitsevat rahapalkkioiden motivoivan hakkereita raportoimaan bugit kyseiselle yritykselle, eikä myymään niitä haavoittuvuuksien harmailla/mustilla markkinoilla. Mustat markkinat tunnetaan paremmin nimellä black

hat markets. Viimeisenä merkittävänä hyötynä mainitaan Bug Bountyn ansiosta pienemmät määrät jäljellä olevia haavoittuvuuksia. Tämä heikentää haitallisten toimijoiden kykyä löytää nollapäivähaavoittuvuuksia, koska ne on jo löydetty ja paikattu. (Finifter ym. 2013.).

Egelmanin ja kumppaneiden tutkimuksessa on avattu vielä paremmin haavoittuvuuksien myynnin ympärillä pyöriviä mustan pörssin markkinoita. Joten tässä vielä tarkennuksen heidän määritelmänsä. Näillä markkinoilla yleensä pahantahtoiset tai rahan himon sokaisemat mustahattuiset hakkerit myyvät löytämiään haavoittuvuuksia. Ostajat ovat hyvin usein kilpailevia yrityksiä tai valtiollisia toimijoita. Toki myös rikolliset aikeet omaava hakkeri saattaa ostaa löydetyn haavoittuvuuden. (Egelman, Herley & Oorschot, 2013.).

Bug Bountyyn on myös kohdistettu kritiikkiä. Maailmalla on useita isoja toimijoita, jotka eivät ole Finfiterin & kumppaneiden tutkimuksen tekohetkellä lanseeranneet Bug Bounty -ohjelmia. Microsoft oli yksi näistä toimijoista. Heidän mielestään kyseinen ohjelma ei tarjoa parasta tuottoa sijoitukselle (ROI). Microsoft on myös argumentoinut, että ei ole varmaa onko maksettavat palkkiot riittävä kannustin hakkereille. Tätä he perustelevat sillä, että laittomilla markkinoilla haavoittuvuuksista maksetaan enemmän. Muun muassa Oracle ja Adobe ovat kertoneet samansuuntaisia ajatuksia julkisuuteen. (Finifter ym. 2013).

Finifiterin ja kumppaneiden tutkimuksen julkaisemisen jälkeen myös Microsoft on aloittanut Bug Bounty -ohjelman. Mielestäni Microsoftin ja monen muun sekä isomman, että pienemmän yrityksen mukaan tulo haavoittuvuuspalkinto-ohjelmien maailmaan osoittaa rohkeutta ja avoimuutta. Tämän perusteella voidaan nähdä yritysten menevän avoimempaan suuntaan. He haluavat selkeästi tuottaa laadukkaita ohjelmistoja asiakkailleen ja olla entistä läpinäkyvämpiä.

3.7 Turvallinen ohjelmistokehitys

Tässä luvussa kuvataan lyhyesti yleisimpiä ongelmia miksi ohjelmistokehityksestä puuttuu tietoturva-aspekti liian usein. Baskervillen mukaan tietojärjestelmiin kohdistuvat turvallisuusuhat ovat olleet riski jo 70-80 luvulta lähtien. Rikolliset ovat jo tuolloin osanneet hyödyntää tietojärjestelmiä laittomuuksiin. (Baskerville, 1993.).

Baskerville mainitsee tutkimuksessaan, että kehittyneet tietoturvalliseen kehitykseen liittyvät menetelmät laahaavat selkeästi normaalin ohjelmistokehityksen laadukkaiden menetelmien perässä. Tutkimuksen mukaan turvallisuuden tulee olla moniulotteista. Sen tulee kattaa, fyysinen turvallisuus, ihmiset, sekä tietokoneet. (Baskerville, 1993.). Baskervillen mukaan jo 1993 tietoturva-asiantuntijat ovat tiedostaneet tietoturvallisuuden integroinnin tärkeyden osaksi järjestelmäkehitystä. Tutkimuksessa korostetaan, että turvallisuus on tärkeä osa tietojärjestelmiä. Jotta tietoturvallinen ohjelmistokehitys mahdollistuu, tulee tietoturvalliset ohjelmistokehitysmenetelmät integroida osaksi normaaleja kehitysmenetelmiä.

Siposen & Baskervillen tutkimus vuodelta 2018 kuvaa, että näitä samoja ongelmia on edelleen olemassa. Tietoturvallisten ohjelmistokehitysmallien evoluutio ei ole edennyt toivotulla tavalla. Tutkijat myös kertovat, että verkkorikollisuus on kasvussa. (Siponen & Baskerville, 2018.). Tutkimusten perusteella vaikuttaa siltä, että huonot mallit ja tavat tehdä ohjelmistokehitystä ovat juurtuneet liian syvälle. Nyt uusien toimintatapojen omaksuminen on haastavaa. Baskervillen mukaan huonot analysointitaidot ja suunnittelumenetelmät johtavat huonoon kontrolliin laadun suhteen. Tämän takia syntyy tietoturvattomia sovelluksia. (Baskerville, 1993.).

3.8 Turvallisen ohjelmistokehityksen malli (SDLC)

Finifter & kumppanit esittelevät teoksessaan turvallisen ohjelmistokehityksen mallin (Secure software development lifecycle, SDLC). (Finifter ym. 2013). Tässä luvussa käydään mallin hyötyjä Bug Bountyn kannalta lävitse. Itse malli esitellään seuraavalla sivulla olevassa kuviossa (kuvio 3). Malli on alun perin lähtöisin Microsoftilta. Mallin avulla pystytään helposti määrittämään tietoturvallisen ohjelmistokehityksen polku. (What is the Security Development Lifecycle?, 2018.).

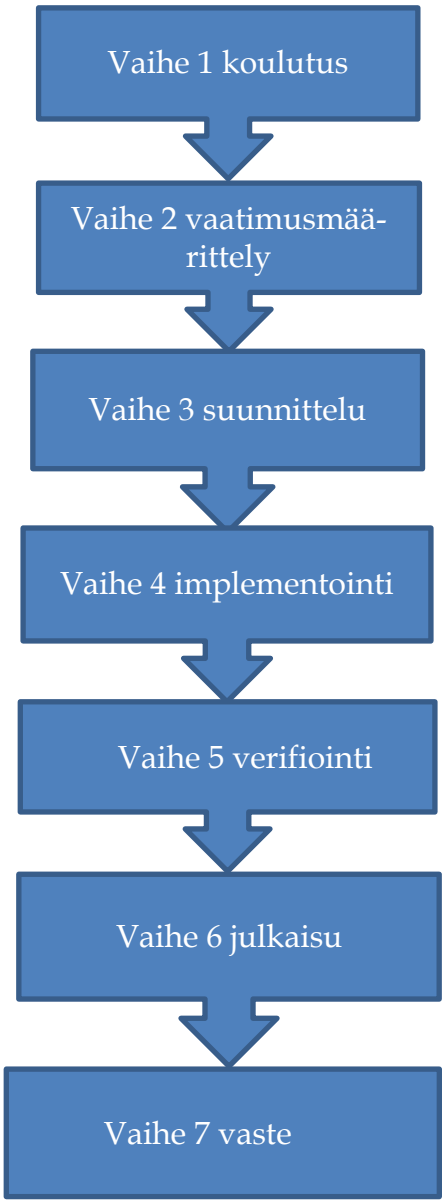
Malli koostuu seitsemästä vaiheesta. Vaiheet ovat koulutus, vaatimusmäärittely, suunnittelu, implementointi, verifiointi, julkaisu ja vaste. Vaiheiden tarkoituksena on määrittää minivaatimukset, joiden avulla ohjelmistoja voidaan kehittää tietoturvallisesti. (What is the Security Development Lifecycle?, 2018.). Koulutusvaihe on esivaatimus turvallisen ohjelmistokehityksen mallin soveltamiselle. Sen avulla määritellään pohja koko mallin tehokkaalle soveltamiselle. Tässä vaiheessa ohjelmoijat, testaajat ja ohjelmapäälliköt koulutetaan. Heiltä vaaditaan vähintään yhden koulutuksen käyminen vuodessa. (SDL process: training, 2018.).

Toinen vaihe on vaatimusmäärittely. Sen tarkoituksena on määritellä tietoturvallisuusvaatimukset ja luoda laatukriteerit. Tässä vaiheessa tehdään myös riskiarviot. SDL process: requirements, 2018.). Suunnitteluvaiheessa määritellään suunnitteluvaatimukset ja tehdään hyökkäyspinta-alaan liittyvät analyysit. Viimeisenä tehdään uhkamallinnus. SDL process: design, 2018.). Implementoinnin aikana määritellään lista hyväksytyistä työkaluista ja kyseenalaistetaan turvattomat / tarpeettomat toiminnot, joita ohjelmistolle on määritelty tai tehty. SDL process: implementation, 2018.).

Verifioinnin aikana varmistetaan ohjelman toimintoja tarkkailevat taustapalveluiden työkalut mm. muistin valvonta, käyttöoikeuksien hallinta ja niin edelleen. SDL process: verification, 2018.). Julkaisuvaiheessa luodaan varautumissuunnitelma (Incident Reponse Plan). Tämän suunnitelman avulla voidaan reagoida nopeasti mahdollisiin ongelmatilanteisiin, joita saattaa ilmetä sovelluksen julkaisemisen jälkeen. Tässä vaiheessa tehdään myös lopullinen turvallisuuskatsaus ohjelmistolle. SDL process: release, 2018.). Tämän mallin viimeinen vaihe on vaste (response). Tämän vaiheen tarkoituksena on varmistaa ja luoda

yritykselle valmius toteuttaa kuudennessa vaiheessa luotua varautumissuunnitelmaa. SDL process: repsone, 2018.).

Tietoturvallisen ohjelmistokehityksen malli mahdollistaa myös haavoittuvuuksien ennaltaehkäisemisen. Tämä tapahtuu seuraavien vaiheiden avulla: koodikatselmoinnit, tietoturvatestaus, dynaamisten ja staattisten analyysityökalujen käyttäminen, sekä haavoittuvuuspalkinto-ohjelmat eli Bug Bountyt. Nämä vaiheet sisältyvät tietoturvallisen ohjelmistokehitysmallin vaiheisiin. Finifter ym. 2013.).



KUVIO 3 Turvallisen ohjelmistokehityksen malli (Microsoft, 2018)

3.9 Bug Bounty tyypit

Tämän luvun tarkoituksena on kuvata lyhyesti yleisimmät Bug Bounty -ohjelman tyypit. Tyyppejä on kolme ja ne ovat globaalisti käytössä. Ensimmäinen on yksityinen Bug Bounty (private), toinen on kutsuihin perustuva (invitation based) ja kolmas on julkinen Bug Bounty (public). (Bacchus, 2017.). Yksityisestä Bug Bountysta ei ole mitään julkisia ilmoituksia. Siihen voi osallistua vain ennalta yrityksen valitsemat henkilöt. Nämä henkilöt voivat olla yrityksen sisäisiä työntekijöitä, konsultteja tai kolmansia osapuolia. (Bacchus, 2017.).

Kutsuihin perustuva Bug Bounty voi myös olla pelkästään yrityksen omassa tiedossa ja yritys sitten pyytää siihen haluamiaan henkilöitä. Useimmiten kutsuperusteinen Bug Bounty on toteutettu siten, että hakkerit saavat hakea siihen mukaan ja siitä on julkisesti tietoa saatavilla. Julkinen Bug Bounty taas on julkaistu yrityksen Internet-sivuilla. Siihen saa myös osallistua kuka tahansa. Useimmiten julkisista ohjelmista tiedotetaan yrityksen muissakin sosiaalisen median palveluissa. (Bacchus, 2017.).

Bug Bountyn julkaisun yhteydessä sitä varten määritetään osallistujille ohjeet, kuinka toimia. Ohjeissa on kerrottu mm. mitä haavoittuvuuksia ja miten saa testata. Ohjeissa voidaan esimerkiksi sanoa, että palvelunestohyökkäykset ovat kiellettyjä. Säännöissä on kerrottu miten ja minne defektit raportoidaan, milloin palkkiot maksetaan jne. Bug Bountyn pääperiaate on, että jos haavoittuvuuden löytäjä hyödyntää löydöstään esimerkiksi myymällä sen eteenpäin tai itse tekee sillä vahinkoa yritykselle ei palkkiota makseta. (Bacchus, 2017.).

Haavoittuvuuden saa usein julkistaa, kun yritys on sen korjannut. Haavoittuvuudet ovat usein määritelty ohjeissa eri vakavuusluokkiin. Kriittiset haavoittuvuudet pyritään korjaamaan mahdollisimman pian. Nollapäivähaavoittuvuudet ovat useimmiten niitä kriittisimpiä, sillä niille ei ole korjausta ja ne saattavat koskea useita eri palveluita ja yrityksiä. (Bacchus, 2017.).

3.10 Turvallisten tietojärjestelmien tutkimuksen historia ja tulevaisuus

Tässä luvussa esitellään hieman turvallisten järjestelmien (Secure Systems Design) historiaa ja tulevaisuutta. Luvun aikana lukijalle selviää nykyisen tutkimuksen hyvät ja huonot puolet. Luvun aikana tullaan myös esittämään kritiikkiä olemassa olevia tutkimussuuntia kohtaan. Luvun lopuksi kerrotaan mihin suuntaan turvallisten tietojärjestelmien kehitystä tulisi ohjata.

Baskerville on aloittanut secure systems design -kentän tutkimisen jo 1988. (Siponen, 2005a.). ISS metodit jaetaan usein kolmeen tai viiteen sukupolveen. Valitettavan usein kuitenkin vain alkupään metodeja käytetään. (Siponen, 2005a.). Siposen mukaan vanhemmat ns. olemassa olevat metodit kuuluvat neljään ensimmäiseen luokkaan. Tulevaisuuden ISS metodit muodostavat viidennen sukupolven. (Siponen, 2005b.). Kolmen luokan jaossa ensimmäinen luokka on yleensä niin kutsutut tarkistuslistat (checklists) tai standardit. Toinen luokka

on yleensä engineering-näkökulma. Kolmas luokka tai sukupolvi on loogisen mallinnuksen menetelmät. (Siponen, 2005b.). Baskervillen ja kumppaneiden mukaan lähes kaikki olemassa olevat turvallisen ohjelmistokehityksen mallit ovat teoreettisesti alikehittyneitä. Niistä puuttuu ns. vakava tutkimus. (Baskerville, Siponen & Heikka, 2006.).

Siponen on tutkinut eri turvallisten ohjelmistokehityksen mallien soveltuvuutta nykypäivän ohjelmistokehitykseen (Siponen, 2005ab). Vain muutama olemassa oleva malli soveltuu integroitavaksi käytössä oleviin ohjelmistokehityksen malleihin. Tämä on huolestuvaa, koska ohjelmistokehitysmallit ovat puutteellisia tietoturvan osalta. Ohjelmistokehitysmallit huomioivat tietoturvan heikosti, jos ollenkaan. (Siponen, 2005b.). & (Baskerville ym., 2006.).

Siposen (2005b) mukaan nykyiset mallit ja tutkimusmenetelmät ovat insinöörimäisiä. Siten niistä puuttuu sosiaalinen näkökulma. Järjestelmäkehityksen ja tietoturvan omatessa sosiaalisia ulottuvuuksia on myös tutkimusten ja mallien hyvä huomioida nämä ulottuvuudet. (Siponen, 2005a, Baskerville ym., 2006.). Tutkijoiden mukaan tällä hetkellä tarjolla ei ole yhtään sosiaalista metodia tietoturvalliseen järjestelmäkehitykseen. Heidän mukaansa pelkät tekniset menetelmät voivat johtaa ongelmiin sosiaalisissa organisaatioissa. Heidän mukaansa tekniset mallit kontrolloivat ihmisiä samalla tavalla, kuin järjestelmiä. Jos toimitaan tällä tavalla ihmiset eivät noudata esimerkiksi tietoturvaan liittyviä politiikkoja jne. (Siponen, 2005a.). Tämän takia sosiaalisten tietoturvan huomioivien mallien kehittäminen on erittäin tärkeää. Bug Bounty on yksi sosiaalinen osallistava malli, jolla voidaan varmistaa tietoturvaa.

3.11 Viitekehys

Edellä esitellyt teoriat ja tutkimukset luovat tämän tutkimuksen teoriapohjan. Luvun alussa mainittu ohjelmistotuotannon teoria valikoitui teoriaosuuden pohjaksi, koska se kuvaa hyvin yksinkertaisesti, kuinka ohjelmistoja tuotetaan. Luvussa 3.1 kuviossa 2 kuvattu malli on käytössä myös tapaustutkimuksen kohteena olevassa LähiTapiolassa. Näin ollen menetelmäosuudessa on helppo tarkastella LähiTapiolan toimintaa mallilla, joka realisoituu käytännössä.

Laadunvarmistus on mukana, koska tutkimus käsittelee tietoturvatestausta. Jotta tästä tutkielmasta saa laadukkaan on erittäin tärkeää määritellä mitä laadunvarmistus on. Joukkoistettu penetraatiotestaus on itseasiassa osa laadunvarmistusta. Tässä luvussa mainitut standardit taas valikoituivat mukaan sillä periaatteella, että ne ovat pakollisia vaatimuksia yritysten sovelluskehityksen laadunvarmistuksen saralla. Joukkoistettu penetraatiotestaus taas on mukana, koska se määrittelee mitä Bug Bounty käytännössä on. Eli Bug Bounty on yhdistetty joukkoistaminen ja penetraatiotestaus tiettyjen sääntöjen ja rajausten puitteissa. Turvallisen ohjelmistokehityksen malli valikoitui tutkimuksen teoriapohjaan mukaan, koska se on yksinkertainen ja toimiva menetelmä kehittää tietoturvallisia sovelluksia. Jotta Bug Bounty olisi toimiva konsepti tulee yritysten Bug Bounty -ohjelmien kohteena olevien sovellusten olla tehty tietoturvalli-

sesti. Sovellukset tulee testata yrityksen oman tietoturvatiimin toimesta ennen, kuin ne julkaistaan Bug Bounty -ohjelmassa. Tietoturvallisen ohjelmistokehityksen malli antaa loistavan pohjan tämän asian tekemiselle. Sen avulla määritellään myös selkeät suunnitelmat ongelmatilanteiden varalle, jotta niihin voidaan reagoida nopeasti.

Tässä mainitut teoriat valikoituivat tähän tutkimukseen niiden helpon yleistettävyyden ja soveltuvuuden vuoksi. Teorioita etsiessä vertailtiin eri teorioita ja valittiin niistä sopivimmat. Valintaperusteita olivat muun muassa tutkimusten ikä, tutkijoiden arvostus ja se, kuinka arvostetussa julkaisussa tutkimus on julkaistu. Tutkimuksia valittaessa tutustuttiin johdantoon ja yhteenvetoon. Näiden avulla sai hyvän kuvan dokumentista ja siitä kannattaako siihen tutustua tarkemmin. Haut suoritettiin Google Scholarin, Jykdokin, Googlen, IEEEXploren ja ACM:än tietokannoista. Hakusanoina toimivat tämän tutkielman keskeisimmät termit eli Bug Bounty, benefits for companies, Bug Bounty advantages for companies, software testing best practices, quality assurance best practices, vulnerability reward program, vulnerability reward program advantages, crowdsourcing, Bug Bounty, crowdsourcing Bug Bounty & quality assurance standards, ISO standards problems ja black hat vs. white hat hackers.

4 TUTKIMUSMENETELMÄT

Tässä luvussa kuvataan käytetyt tutkimusmenetelmät. Luvun alussa kerrotaan kuinka, tutkimus suunniteltiin. Suunnitteluosiossa valotetaan käytettyä tutkimusmenetelmää ja kerrataan tutkimuskysymykset. Seuraavaksi kerrotaan millaisilla menetelmillä data kerättiin ja miten. Tämä luku valottaa myös sitä, miten tutkimuskysymykset ja haastattelukysymykset nivoutuvat yhteen. Luvun lopussa valotetaan tarkemmin haastatteluiden tekotapaa ja kuinka niistä saatua dataa analysoitiin.

Eri tutkimusmenetelmille on olemassa useita vaatimuksia ja ohjeita (guidelines). Usein näiden tarkka noudattamatta jättäminen estää tutkimuksen julkaisemisen tietojärjestelmätieteen parhaissa lehdissä ja tutkimus nähdään huonona. (Holtkamp, Soliman & Siponen, 2019.).

Holtkamp ja kumppanit kritisoivat myös tiukkoja ohje -ja muotovaatimuksia mm. laadullisille tutkimuksille. Heidän mukaansa tämä voi poistaa luovuuden ja tutkimuksista ei välttämättä tule monipuolisia. Kirjoittajien mukaan liian tiukat vaatimukset saattavat myös aiheuttaa työn väärää tulkitsemista. Heidän mukaansa julkaistu tutkimus on tämän perusteella arvoitu enneminkin sillä perusteella noudattaako se muotovaatimuksia. Tutkimuksen oleellinen asia eli sen sisältö on jätetty täysin huomiotta. Tämä saattaa aiheuttaa laadukkaiden ja mielenkiintoisten tutkimusten jäämisen huomioitta. (Holtkamp ym., 2019.). Tähän kritiikkiin perustuen tämä tutkimus on toteutettu empiirisen tutkimuksen ja kirjallisuuskatsauksen yhdistelmänä. Tarkoitus on avata uutta tutkimuskenttää ja antaa tuloksille tulkinnanvaraa.

4.1 Tutkimuksen suunnittelu

Tutkimuksen suunnittelun lähtökohtana oli saada puolueetonta dataa siitä onko Bug Bounty -ohjelman lanseeraaminen hyödyttänyt LähiTapiolaa. Jos on niin miten? Sekä miten tulokset hyödyttävät muita yrityksiä. Tutkimuksen pääasiallinen tarkoitus oli vastata alla oleviin tutkimuskysymyksiin:

Tutkimuskysymys 1: Mitä hyötyjä LähiTapiolalle on ollut Bug Bountyn käyttöönotosta?

Tutkimuskysymys 2: Mitä ongelmia ja riskejä Bug Bountysta on ollut LähiTapiolalle?

Tutkimuskysymys 3: Mitä opittiin ja mitä vietiin käytäntöön?

Koska tutkimuksen pääasiallinen tarkoitus oli selvittää asiakasyrityksellä käytössä olevan palvelun hyötyjä, valikoitui tutkimusmenetelmäksi laadullinen tapaustutkimus. Tarkennettuna single case study. Tutkimuksessa käytettiin tulkitsevan ja selittävän casemetodin yhdistelmää. (Walsham, 1995). Tutkimuksen pohjaksi tehtiin ytimekäs kirjallisuuskatsaus yksinkertaisiin ja käytännössä hyödynnettäviin teorioihin. Nämä teoriat muodostavat tutkimuksen punaisen langan, joka kertoo miten tietoturvatestaus ja Bug Bounty tulisi hoitaa kirjan oppien mukaan. Haastattelukysymykset pohjautuivat yllä mainittuihin tutkimuskysymyksiin. Koska tutkimuskenttä on Bug Bountyn hyötyjen ja haittojen osalta kohtalaisen tuore. Ei haastatteluihin valmistauduttaessa ollut mitään selkeitä teorioihin perustuvia olettamuksia. Tavoite oli saada haasteltavilta käytännön kokemuksen antamaa tietoa. Haastatteluiden avulla pyrittiin myös saamaan irti ns. hiljaista tietoa, jota ei mistään aiemmista tutkimuksista ja teorioista välttämättä löydy.

4.2 Tiedonkeruumenetelmät

Haastattelututkimus tehtiin semi-strukturoidulla haastattelumenetelmällä. (Myers & Newman, 2007). Haastatteluita varten oli tehty runko, joka sisälsi 20 tukikysymystä, joiden avulla haastattelija saattoi viedä keskustelua eteenpäin. (Liite 1). Haastateltavia pyydettiin ensisijaisesti kertomaan tarkasti ja omin sanoin esimerkkejä ja tarinoita Bug Bountysta ja sen käytöstä sekä käyttöönotosta. Kysymykset lähetettiin haastateltaville etukäteen tutustuttaviksi. Tämän tarkoitus oli tehdä haastattelutilaisuudesta rento. Haastatteluita sovittaessa korostettiin, että kyseessä ei ole mikään ristikuulustelu.

Haastattelut toteutettiin suomenkielellä, koska kaikkein haastateltavien äidinkieli oli Suomi. Näin pystyimme eliminoimaan kielimuurin aiheuttamat epäselvyydet. Toki jos haastattelun aikana ilmeni epäselvyyksiä haastateltavia, pyydettiin tarkentamaan vastauksiaan. Haastattelut toteutettiin pari ja yksilöhaastatteluina. Parihaastattelut tehtiin henkilöille, ketkä tekevät tiiviisti töitä yhdessä samoilla osa-alueilla. Tällä metodilla saatiin hedelmällisempää keskustelua aikaiseksi. Haastateltavat henkilöt olivat keskenään samanarvoisia. Esimiehiä ja alaisia ei haastateltu samassa tilaisuudessa. Tällä tavoin vältettiin mahdolliset pelkotilat jättää vastaamatta tai vääristellä vastauksia. Haastatteluiden kesto vaihteli 0,5-2,5 tunnin välillä. Myöskään ryhmähaastatteluita ei haluttu tehdä. Näin vältettiin toisten haastateltavien tietoinen ja tiedostamaton

vaikutus toisiinsa. Yksilö ja parihaastatteluiden avulla tutkimusten data on luotettavampaa ja aitoa vs. ryhmähaastattelut.

Haastattelut tallennettiin digitaalisella sanelimella. Muita muistiinpanoja ei haastatteluiden aikana tehty. Näin haastattelija, joita oli vain yksi, pystyi keskittymään keskusteluun täysillä. Tallennetut haastattelut purettiin ja niistä kirjoitettiin ylös muistiinpanoiksi tutkimuksen kannalta oleellisimmat asiat. Digitaaliset versiot haastatteluista arkistoitiin, jotta niitä voidaan tarvittaessa käyttää myöhemmin. Haastattelukysymykset valittiin siten, että niillä saadaan mahdollisimman laaja-alainen kuva LähiTapiolan työntekijöiden suhtautumisesta Bug Bountyyn.

4.3 Haastateltavien valinta

Haastateltavat valittiin siten, että LähiTapiolan tietoturvatiimin, IT:n ja ohjelmistokehityksen osa-alueilta saatiin kunkin alueen spesifi osaaminen mukaan. Haastatteluun pyydettiin henkilöitä, jotka koordinoivat Bug Bountya LähiTapiolassa, tekevät tietoturvatestausta ennen sovellusten julkaisua tuotantoon ja Bug Bountyyn, sekä juridisesta näkökulmasta vastaava henkilö. Haastateltavia oli yhteensä yhdeksän kappaletta. Haastateltavat ehdotti LähiTapiolan tietoturvapäällikkö. Henkilöiden positioita, taustaa ja osaamista peilattiin tutkimuskysymyksiin. Niiden perusteella ehdotukset hyväksyttiin ja kyseiset henkilöt haastateltiin. Alla oleva taulukko (taulukko 1) kuvaa haastateltavien roolit ja organisaatiot.

TAULUKKO 1 Haastateltujen määrä organisaatioittain

Haastateltujen määrä	Organisaatio	Esimies
5	LähiTapiola	1 kpl
2	Yhteistyöyritys	-
2	Palveluntarjoaja	-

4.4 Datan analysointi

Haastatteluiden aikana saadut vastaukset eivät olleet mitenkään selkeästi jäsenneltyjä ja helposti poimittavissa. Koska vastaukset olivat kerronnallisia, täytyi jokaisesta haastattelusta erikseen poimia tutkimuksen kannalta relevantit kohdat. Tämä tehtiin lukemalla litteroidut haastattelut. Niistä poimittiin relevantit vastaukset. Vastukset jaoteltiin kuuteen kategoriaan. Nämä kategoriat olivat: hyödyt, haasteet, riskit, tilastolliset asiat, opit ja muut. Jokaisen kategorian alla olevan vastausnipun avulla pyrittiin vastaamaan loogisesti ja jäntevästi tutkimuskysymysten avulla kategorian kysymykseen.

Myös haastateltujen henkilöiden roolit, taustat, kokemus ja osaaminen huomioitiin vastausten analysoinnissa. Vastauksien validiutta arvoitiin vastaajan taustojen mukaan. Lähtökohtaisesti vastaukset olivat sitä mitä taustojen perusteella oletettiin niiden olevan. Saatuja vastauksia peilattiin tämän tutkimuksen alussa esitettyihin kirjallisuuskatsauksen teorioihin. Tällä toiminnalla analysointiin sitä, kuinka hyvin vastaukset täsmäävät kirjallisuudessa esitettyihin teorioihin. Samalla voitiin löytää vastauksista se haluttu hiljainen tieto.

5 TUTKIMUSTULOKSET

Tämä luku esittelee tutkimustulokset teorian ja haastatteluiden osalta.Tuloksia peilataan alussa esitettyihin teorioihin. Tässä luvussa esitellään myös lyhyesti asiakasayritys LähiTapiola ja heidän Bug Bounty -ohjelmansa. Luvun loppupuolella käsitellään Bug Bountyn haasteita, hyötyjä ja tulevaisuutta. Tämän luvun aikana lukijalle muodostuu kattava kuva tutkimustuloksista.

5.1 LähiTapiola

LähiTapiola on Suomalainen asiakkaidensa omistama vakuutus -ja sijoitusyhtiö. LähiTapiolalla on reilut 1,5 miljoonaa omistaja-asiakasta. LähiTapiola tuottaa palveluita myös yrityksille. (Tietoa yhtiöryhmästä, 2018).

> LähiTapiola-ryhmä muodostuu valtakunnallisesti toimivista LähiTapiola Vahinkoyhtiöstä, LähiTapiola Henkiyhtiöstä, LähiTapiola Varainhoidosta, LähiTapiola Kiinteistövarainhoidosta ja LähiTapiola Kiinteistöpääomarahastoista sekä 20 alueellisesta keskinäisestä vahinkovakuutusyhtiöstä. (Ryhmän rakenne ja johto, 2018, 1).

Ryhmällä on vajaa 3 500 työntekijää (Ryhmän rakenne ja johto, 2018). Yhtiön pääkonttori sijaitsee Espoossa. LähiTapiola syntyi 2012 Lähivakuutuksen ja silloisen Tapiolan fuusioituessa. Virallisesti yhtiön toiminta alkoi tammikuussa 2013. (Historia, 2018.).

5.2 Miten Bug Bounty yleensä toimii

Tämä luku esittelee lyhyesti, kuinka Bug Bounty -ohjelmat käytännössä toimivat. Yritykset voivat toki vapaasti muokata ohjelmia itsensä näköisiksi ja näin ollen kaikki Bug Bountyt eivät ole identtisiä. Tämä on vain hyvä asia, koska ohjelmien erilaisuus ja raikkaus saa hakkerit kiinnostumaan niistä. Muokkauk-

sista huolimatta Bug Bountyilla on tietyt periaatteet, joiden mukaan ne toimivat. Bug Bounty tyyppejä on kolme ja nämä esiteltiin jo luvussa 3.8. Useimmiten yritys julkaisee Bug Bounty -ohjelmansa niihin tarkoitetuilla alustoilla. Hackerone, Bugcrowd, Hackr.fi ja ZeroCopter ovat muun muassa yrityksiä, jotka tarjoavat Bug Bountyn julkaisualustoja.

Aloittaessa Bug Bounty -ohjelmaa yrityksen tulee päättää missä ja miten se haluaa julkaista ohjelmansa. Seuraavaksi päätetään onko kyseessä yksityinen, kutsuihin perustuva vai julkinen ohjelma Tässä yhteydessä valitaan usein myös yhteistyökumppanit. Kun ohjelman tyyppi on päätetty, tulee laatia tarkka rajaus mitkä yrityksen palvelut, ohjelmistot, verkkosivustot yms. ovat mukana Bug Bountyssa. Jos hakkeri löytää haavoittuvuuksia rajausten ulkopuolelta ei hänelle tarvitse maksaa palkkiota. On kuitenkin suositeltavaa ottaa vastaan myös rajausten ulkopuolella olevat löydökset. Jos nämä ovat vakavia useimmat yritykset maksavat myös niistä. Tämä antaa positiivista julkisuutta yritykselle ja saa hakkerit testaamaan jatkossakin yrityksen palveluita. (Bacchus, 2017.).

On myös suositeltava laatia säännöt Bug Bountyyn liittyen. Nämä säännöt kertovat mitä palveluita saa testata ja mitä ei. Niistä käy myös ilmi, miten ja minne haavoittuvuudet raportoidaan. Sääntöihin määritellään myös palkkioiden suuruudet ja maksuaikataulut. Palkkioiden suuruuden yritys voi päättää itse. Jotta Bug Bounty -ohjelma olisi houkutteleva hakkereille tulee palkkioiden olla riittävän suuria. Palkkiot luokitellaan usein niiden teknisen luokan ja mahdollisen yritykselle / liiketoiminnalle aiheutuvan vaikutuksen mukaan. Jos esimerkiksi haavoittuvuuden avulla saa yksittäisen käyttäjän etunimen tietoonsa se on lievä haavoittuvuus. Jos taas haavoittuvuuden avulla voidaan kaapata työntekijän sessio ja saada kaikkien käyttäjien yksityiskohtaiset arkaluintoiset tiedot haltuun on tämä vakava haavoittuvuus. Palkkiot voivat olla tarkkoja summia tai hintahaarukoita. Alla oleva taulukko (taulukko 2) havainnollistaa palkkioiden luokittelua. (Bacchus, 2017.).

TAULUKKO 2 Bug Bounty palkkioiden luokittelun havainnollistaminen

Vakavuus / vaikutus (1=vähäinen vaikutus, 2=erittäin suuri vaikutus)	**Haavoittuvuuden tyyppi**			
	A	B	C	D
1	200 €	1 000 €	5 000 €	10 000-30 000 €
2	300 €	1 500 €	6 000 €	40 000-60 000 €
3	500 €	2 000 €	7 000 €	70 000-90 000 €
4	700 €	2 500 €	8 000 €	100 000-200 000 €
5	900 €	3 000 €	9 000 €	500 000 €

Palkkiotaulukon lisäksi laaditaan usein kirjallisessa muodossa olevat tarkemmat määritykset minkä suuriset palkkiot mistäkin haavoittuvuudesta maksetaan. Yritys voi raportin saatuaan arvioida onko maksettava palkkio haarukan alapäässä vai yläpäässä. Yrityksen on hyvä perustaa tiimi, jonka vastuulla Bug

Bountyn pyörittäminen on. Jos yrityksen resurssit ovat niukat voidaan nimetä vain yksi henkilö koordinoimaan Bug Bountya. Kun nämä asiat ovat tehty voidaan Bug Bounty -ohjelma julkaista. (Bacchus, 2017.). Seuraavalla sivulla oleva kuvio (kuvio 4) havainnollistaa vielä erittäin yksinkertaisesti, kuinka Bug Bounty toimii.

KUVIO 4 How a Bug Bounty Works (HackerOne, 2016, How Bug Bounties Work: A Comic)

5.3 LähiTapiolan Bug Bounty

Tämä luku esittelee lyhyesti LähiTapiolan Bug Bountyn. LähiTapiola -ryhmä aloitti Bug Bounty -ohjelmansa syyskuussa 2015. Heidän Bug Bountynsa oli alkuun julkinen, sittemmin hetken yksityinen ja nyt taas julkinen. Palkkiot vaihtelevat välillä 50-50 000 Yhdysvaltain dollaria (USD). LähiTapiola käyttää Bug Bounty alustanaan Hackeronen tuottamaa palvelua. Hackeronen verkkosivuilta löytyy LähiTapiolan Bug Bountyn tiedot ohjeineen. (Hackerone, LocalTapiola, 2018.).

LähiTapiola noudattaa hyvin pitkälti luvussa 5.3 esiteltyjä Bug Bountyn periaatteita. Merkittävimpänä erona edellisessä luvussa mainittuun on LähiTapiolan tapa arvioida haavoittuvuuksia ja niistä maksettavia palkkioita. He käyttävät liiketoimintavaikutusanalyysia tähän. Mallissa LähiTapiola arvioi ne riskit, joita haavoittuvuus voi aiheuttaa yrityksen liiketoiminnalle. Yksinkertaistaen mitä suurempi vaikutus liiketoimintaan kohdistuu vihollisen hyödyntäessä löydettyä haavoittuvuutta sitä suuremman palkkion haavoittuvuuden löytäjä saa. (Niemelä, 2018.). Tämä malli koostuu yhdeksästä eri kohdasta. Kohdat ovat:

1. Haavoittuvuuden vaikutus LähiTapiolan palveluille
2. Koskeeko asiakkaita? Jos kyllä, kuinka isoa osaa?
3. Koskeeko LähiTapiolan työntekijöitä? Jos kyllä, kuinka isoa osaa?
4. Moneenko järjestelmään haavoittuvuus kohdistuu?
5. Voiko haavoittuvuutta hyödyntää ilman tunnistautumista?
6. Haavoittuvuuden hyödyntämisen vaikeus? Vaatiiko esimerkiksi vuorovaikutusta käyttäjän kanssa?
7. Hyödyntämisen aikaikkuna?
8. Haavoittuvuudessa hyödynnettävien tietojen arvo LähiTapiolalle?
9. Onko haavoittuvuus löydetty Bug Bounty -ohjelmassa mukana olevasta kohteesta?

(Niemelä, 2018, 21.).

LähiTapiolan Bug Bounty kohdistuu heidän julkisesti saatavilla oleviin verkkopalveluihin. Bug Bountyn kohteena olevat palvelut on määritelty tarkemmin Hackeronen verkkosivuilla. Samoin siellä on kerrottu selkeästi, että kaikki ne palvelut, jotka eivät ole mukana eli in scope ovat out of scope. (Hackerone, LocalTapiola, 2018.). Lopuksi mainittakoon vielä, että LähiTapiola on huomioinut myös GDPR:än ehdoissaan. Merkittävimpänä tästä on kohta, jossa määritellään valkohattuisten hakkereiden olevan osa heidän tietoturvaprosessiaan. ”Our bug bounty program is an additional strategic component in our ICT risk management process.” (Hackerone, LocalTapiola, 2018, 4).

5.4 Hyödyt

Tässä alaluvussa kuvataan mitä hyötyjä Bug Bountyn käyttöönotosta LähiTapiolalle tuli. Luku kokoaa yhteen kuuden erillisen haastattelusession aikana saadut tiedot Bug Bountyn hyödyistä.

Bug Bountyn on osa LähiTapiolan tietoturvaprosessia. Yhtenä hyötynä haastateltavat kokivat selkeästi osittaisen tietoturvatestauksen ulkoistamisen. Tällä tavalla LähiTapiolan tietoturva-asiantuntijoiden aikaa vapautetaan muiden tärkeiden tehtävien hoitamiseen. Kun tietoturva-asiantuntijat testaavat vain sovellusten ja palveluiden kriittiset osat. Joiden tietoturvan tule olla kunnossa ennen, kun ne voidaan julkaista Internetiin jää heille enemmän aikaa uusien palveluiden suunnitteluun ja kehittämiseen.

Toisena hyötynä haastateltavat kokivat LähiTapiolan saaman positiivisen julkisuuden. Tämä julkisuus on tunnustettu ja kiitelty LähiTapiolan ylimmän johdon toimesta. LähiTapiolan Bug Bounty on niittänyt mainetta myös maailmalla. Darkreading-tietoturvasivusto listasi LähiTapiolan Bug Bountyn yhdeksi maailman kiinnostavimmista Bug Bounty -ohjelmista. Samalla listalla ovat muun muassa Google ja Apple. Tämä valinta lisäsi hakkereiden kiinnostusta entisestään ohjelmaa kohtaan. Yhtenä merkittävänä LähiTapiolan arvostuksena Bug Bountya kohtaan yrityksen työntekijät ehdottivat LähiTapiolan tietoturvajohtaja Leo Niemelää vuoden tietoturvajohtajaksi vuonna 2016. Hän voitti tuon palkinnon kyseisenä vuonna.

Haastateltavat mainitsivat konkretisoituneena hyötynä laadukkaamman koodin. Bug Bountyn käyttöönoton jälkeen oli selkeästi havaittu suunnan muutos koodauksessa. Toimittajien ohjelmoijat tekevät laadukkaampaa koodia. Tässä laatu on parantunut nimenomaan tietoturvan osalta. Useimmat Bug Bountyn kautta löydetyt haavoittuvuudet peilautuvat LähiTapiolassa OWASP Top 10 listaukseen. Bug Bountyn löydösten avulla on voitu osoittaa ne yleisimmät virheet, jotka toistuvat lähes jatkuvasti koodissa. Bug Bountyn avulla on saatu myös sosiaalinen paine hyvää koodia kohtaan. Ohjelmoijat miettivät, että ”nyt ne testaavat minun koodiani siellä ja siellä on haavoittuvuuksia” -tyylisesti. Tämä johtaa automaattisesti toimintatavan muutokseen ja uusi koodi on parempaa. LähiTapiolan mukaan myös toimittajien yleinen laatu on parantunut ja Bug Bounty nähdään niin sanottuna vahvan välittämisen mallina.

Bug Bountyn kiinnostavuus ja tunnettuus maailmalla nähtiin myös hyvänä asiana. Hakkerit saavat pisteitä ja mainetta löytämistään haavoittuvuuksista. Kun Bug Bounty ohjelma on kiinnostava, houkutteleva ja tunnettu osallistuu siihen myös ne parhaat hakkerit. Näin ollen LähiTapiola saa hyvää mainetta myös hakkereiden keskuudessa ja pystyy osallistamaan ohjelmaansa ne kaikkein kyvykkäimmät osaajat.

Haastatteluissa nousi esille hyötynä myös se, että Bug Bountyn myötä saat tietää nopeammin asioista. Monesti jopa sellaisista haavoittuvuuksista, joita ei muuten koskaan löytyisi. Tähän liittyen hyötynä nähtiin myös nopealla tahdilla sisään tulevat löydökset. Tämän pohjalta haastateltavat myös totesivat, että oikein sovellettuna ja käytettynä Bug Bounty tukee ehdottomasti kokonaisvaltaista laadunvarmistusta.

Haastatteluista kävi myös ilmi, että tietoturvaan ei välttämättä ole budjetoitu kehitysrahaa ja siihen ei ole panostettu riittävästi. Bug Bounty on parantanut tilannetta. Nyt tietoturvaan panostetaan selkeästi enemmän ja tehokkaammin. Bug Bountyn tuoma avoimuus tietoturvan suhteen nähtiin hyvänä asiana. Yritys antaa ulospäin selkeän viestin ”haluamme panostaa tietoturvaan ja välitämme asiakkaistamme”. Avoimuus on LähiTapiolalaisten mielestä tulevaisuutta. Tämä lisää myös asiakkaiden luottamusta yritystä kohtaan. Haastateltavat näkevät Bug Bountyn myös ennakointina. Eli tehdään asioita ennen, kuin asiat räjähtävä käsiin.

Bug Bounty nähdään sisäisesti hyvänä työkaluna myös ostettaessa palveluita kumppaneilta. LähiTapiola osaa nykyään vaatia parempaa koodia ja tietoturvallisempia palveluita. Jos toimittaja ei toimita sovittua laatua voidaan kyseenalaistaa toimitus, eli oliko tämä nyt sovittua laatua vai ei? Haastateltujen mukaan LähiTapiolan toimittajat ovat suhtautuneet Bug Bountyyn positiivisesti. Toimittajien sovellusten laatu ja tietoturva ovat selkeästi parantuneet. Toimittajat ovat myös ottaneet Bug Bountyn palautteen ilolla vastaan ja parantavat toimittamiaan tuotteita ja työtapojaan jatkuvasti.

LähiTapiola on onnistunut myös luomaan Bug Bountyn avulla selkeän prosessin korjausten tuotantoon viennille. Korjaukset viedään tuotantoon normaalilla syklillä. Niiden viemisellä ei ole erityistä kiirettä, koska haavoittuvuudet julkistetaan vasta, kun ne ovat korjattu. Aiemmin LähiTapiolalla ei ollut selkeää kanavaa haavoittuvuuksien raportointiin ja oli jatkuvasti kiire, sekä pelko siitä, että haavoittuvuuden löytäjä julkaisee sen ennen aikojaan mediaan tms. Selkeä prosessi on poistanut kiirettä ja säätöä mitä ennen oli paljon. Löydettyjen haavoittuvuuksien julkaiseminen (public disclosure) nähdään myös hyvänä asiana. Sen ansiosta hakkerit oppivat mitä ja miten kannattaa ainakin LähiTapiolan palveluissa testata. Löydettyjen virheiden julkaiseminen auttaa myös muita yrityksiä parantamaan palveluitaan, koska usein samat asiat toistuvat muidenkin yritysten palveluissa. LähiTapiolan toimittajiensa ohjelmoijille järjestämät työpajat löydettyjen haavoittuvuuksien läpikäymiseksi koettiin myös erittäin hyödyllisiksi.

Haastateltavien mukaan virheraporttien määrä kertoo siitä, että hakkeriyhteisön voima on valtava. Se myös todistaa, että ohjelmalla on positiivisia vaikutuksia LähiTapiolan tietoturvaan. Raporttien määrän tasoittuminen ja palkkioiden määrä yhdessä kielivät myös parantuneesta tietoturvasta. Haastateltavien mukaan haavoittuvuuksia löytyy edelleen, mutta ei niin paljon. Myös tietyt OWASP Top 10 haavoittuvuudet ovat kadonneet lähes kokonaan kiitos tietoturvallisemman koodaustavan.

5.5 Haasteet

Tässä luvussa käsitellään haastatteluiden aikana esille tulleita haasteita LähiTapiolan Bug Bountyssa. Haastatteluiden perusteella haasteita oli ohjelmaa käynnistettäessä ja myös käynnistymisen jälkeen.

Ohjelman käynnistyessä haitoiksi mainittiin muun muassa tietty epäselvyys siitä mikä Bug Bounty on ja miten se toimii. Osa toimittajista ei halunnut palveluitaan ohjelman piiriin, koska pelkäsivät, että ne joutuvat mustahattuisten hakkereiden kohteeksi. Palvelut olisi ollut mahdollista lisätä ohjelmaan, mutta silloin LähiTapiola olisi joutunut maksamaan kaikki kulut, jos jotain olisi käynyt. Tästä seurasi seuraava haaste palvelut ovat julkisesti verkossa näkyvissä, mutta niihin ei voi laittaa tarraa, joka sanoo älä hakkeroi tätä. Tällöin Bug Bountyn puitteissa kyseisiin palveluihin kohdistui testausta ja hakkereiden suunnalta tuli ihmetystä, miksi ei saadakaan palkkioita. Tämä ongelma korjattiin listaamalla LähiTapiolan Bug Bounty -sivustolle ne palvelut, jotka ovat Bug Bountyn piirissä. Niiden perään lisättiin disclaimer, joka kertoo, että muut palvelut eivät ole ohjelman piirissä.

Toinen haaste alkuun oli Bug Bountyn julkaisemisen aiheuttama piikki verkkoliikenteessä LähiTapiolan palveluita kohtaan. Haastateltavien mukaan useat hakkerit testailivat LähiTapiolan palveluita ja tämä aiheutti palveluissa hitautta. Palvelut olivat kuitenkin pystyssä. Tämän ansiosta palveluiden tehokkuutta ja suorituskykyä parannettiin. Näin ongelma poistui. Bug Bountyn pyöriessä haasteeksi koettiin ohjelmistokehityksen raskaus. Malli on lähempänä vesiputousmallia, kuin agilea. Johtuen osittain finanssialan tiukoista säännöistä ja legacy-ohjelmistoista, joita ei voi agilesti päivittää. Muutosprosessi koettiin liian pitkäksi. Myös hakkerit kokivat prosessin liian pitkäksi. Heiltä tuli paljon ihmetystä siitä miksi haavoittuvuuksien korjaaminen kestää näin kauan ja miksi emme saa rahojamme.

Tästä seurasi haastateltujen mukaan muutama lisähaaste. Suomessa on pitkät lomat ja LähiTapiolalla myös kielto tehdä kesän aikana muutoksia. Tämän selittäminen hakkereille ei ollut helppoa. Esimerkiksi Google ja muut ison maailman firmat eivät lomia pahemmin vietä. Myös toimittajien näkökulmasta muutosprosessi koettiin hitaana. Toimittajat kokivat haasteena myös resursoinnin. Raportteja haavoittuvuuksista tuli paljon ja heillä oli vaikeuksia löytää tekijöitä korjauksille. Resurssien pitää ymmärtää teknisen tietoturvan näkökulmasta, miten ongelma korjataan. Toimittajat toivoivatkin kiinteitä resursseja korjauksia varten.

Haastatteluiden aikana ajan puute mainittiin myös ongelmaksi. Jokaisella toimittajalla on oma tiketöintijärjestelmänsä ja näiden kautta tulevissa muutospyynnöissä on vain Hackerone-alustalla olevan virheraportin tunniste. Toimittajan edustaja joutui siis selvittämään mistä on kyse ja tekemään tilauksen ohjelmoijalle. Hackeronen tiketeissä oli myös pitkiä keskusteluja ja niistä jyvälle pääseminen koettiin haastavaksi. Myöskään kaikilla ohjelmoijilla ei ollut pääsyä Hackerone-alustalle, jossa tekniset tiedot virheestä oli. Toki tietoturvaan liittyvien defektien raporttien tulee näkyä vain rajatuille henkilöille, joiden työtehtävien kannalta se on oleellista. Tämä tuo lisähaastetta asiaan. Ongelmaksi koettiin myös tilanteet, joissa toimittajan ohjelmoija kysyy LähiTapiolan tietoturvatestaajalta, miten korjaus pitäisi testata. Osalta toimittajia puuttuu osaaminen tietoturvasta sovelluskehityksessä. Tämä sitoo turhaan tietoturvatestaajan aikaa.

Haastateltavat kritisoivat myös korjausten kattavuutta ja sitä, että Bug Bounty -tiimillä ei ole keinoja seurata korjauksia saatikka tietoa, miten korjaus on tehty. Vastaan on tullut useita tapauksia, joissa korjaus on ollut laastari.

Esimerkiksi, jos virhe on löydetty syöttämällä järjestelmään A tämä on korjattu. Mutta jos syöttää B:n sama ongelma toistuu. Haastateltavien mukaan tämä ongelma johtuu osin siitä, että Bug Bounty -tiimi ja kehittäjät ovat eri yrityksissä. Haastateltavat korostivat kommunikaation tärkeyttä. Myös selkeämpää yhteistyömallia kaivattiin. Yksi ehdotus oli ottaa toimittajan ohjelmoijat aktiivisesti mukana relevantteihin palavereihin. Nyt viestit hajoavat matkalla liian usein. Laastarikorjaukset aiheuttavat ongelmia myös hakkereiden suuntaan. Koska LähiTapiolasta kerrotaan heille virheen olevan korjattu. Todellisuudessa näin ei olekaan. Tämä aiheuttaa haittaa LähiTapiolan maineelle.

Haastatteluissa nousi haasteeksi myös kommunikointi hakkereiden kanssa. Osa hakkereista ei ymmärrä Suomen lomakäytäntöjä, finanssialan toimintaa ja sääntöjä jne. Näiden asioiden kommunikointi koettiin haastavaksi heidän suuntaansa. Osa hakkereista myös olettaa Suomen ja LähiTapiolan olevan samassa mittakaavassa, kuin Yhdysvallat ja heidän suurimmat finanssialan yrityksensä. Todellisuudessa näin ei ole. Toinen kommunikaation kanssa esiin tullut haaste on hakkereiden toimittamat raportit. LähiTapiola on julkaissut selkeät ohjeet Bug Bounty sivustollaan raporteista. Siltä yllättävän moni raportti on puutteellinen tai täysin väärin tehty. Osa raporteista on myös selkeästi täysin geneerisiä ja niillä koitetaan vain saada rahaa LähiTapiolalta, vaikka mitään virheitä ei ole palveluista löydetty.

Haastateltujen mukaan ulkomaalaisten hakkereiden kanssa törmää valitettavan usein isoon kielimuurin. Osa heistä ymmärtää vain sanat yes ja no. Ei mitään muuta. LähiTapiolan Bug Bounty on julkinen ja kansainvälinen. Tämä näkyy haastateltavien mukaan erittäin selkeästi. Toki he painottivat, että myös hyviä ja erinomaisia raportteja tulee hakkereiden suunnalta. Huonot raportit osoittavat myös uuden haasteen. Eli Bug Bountyn ohjeita ei lueta. Näin ollen osa raporteista käsittelee esimerkiksi palveluita, jotka eivät kuulu ohjelman piiriin. Myös kulttuurierot esimerkiksi, miten eri maissa suhtaudutaan sääntöihin ja lakeihin koettiin haasteeksi. Osa LähiTapiolan palveluista vaatii kirjautumisen Suomalaisilla pankkitunnuksilla. Tämä rajoittaa ulkomaalaisten osallistumista. Haastateltujen mukaan tästä aiheutuu myös haasteita. Käytettävissä on vähemmän hakkereita.

Myös testi -ja tuotantoympäristöjen identtisyys nousi haasteeksi. Bug Bountyn piirissä olevia sovelluksia testataan tuotannossa. LähiTapiolan tuotanto -ja testiympäristöt eivät aina ole identtisiä. Näin ollen LähiTapiolan edustaja ei välttämättä voi toistaa/testata hakkerin löytämää virhettä. Tämän kommunikoiminen hakkereiden suuntaan voi myös olla ongelmallista. Yleisesti haastatteluista oli tulkittavassa, että Bug Bountyn pyörittäminen ei ole helppoa. Se on haastavaa, mutta erittäin palkitsevaa. Haasteet ovat osin yhteisiä yrityksille ja osin yrityskohtaisia. Jokainen haaste on kuitenkin taklattavissa. Haastateltavien mukaan sovelluksia on vaikea rakentaa turvallisiksi, mutta helppo rikkoa. Samoin tietoturvallisen koodin tekeminen on jatkuvaa työtä. Sitä ei opita yhdessä yössä ja siihen pitää panostaa.

Haastateltavat nostivat esille myös budjetin. Tietoturvaan budjetoitu raha ei ole itsestäänselvyys. Rahan puute aiheuttaa löydettyjen virheiden korjaamisessa haasteita. Myös testausorganisaation avainhenkilöiden tärkeyttä painotettiin. Jos esimerkiksi jostain asiasta tietää vain yksi ihminen ja hän jostain syystä

poistuu vahvuudesta saattaa kestää liian kauan ennen, kuin korvaaja on työn touhussa. Lopuksi haluttiin nostaa esille myös tietoturvattoman koodin aiheuttama tekninen velka, josta nyt maksetaan. Tosin Bug Bountyn yhdeksi hyödyksi voidaan laittaa teknisen velan pienentäminen tietoturvan osalta.

5.6 Riskit

Tässä luvussa käsitellään Bug Bountyn LähiTapiolalle aiheuttamia riskejä. Luvussa mainitaan myös, miten LähiTapiola kyseiset riskit hoiti. Haastateltavien mukaan LähiTapiola teki riskianalyysin ennen Bug Bountyn aloittamista. Osa analyysiä oli varmistaa yrityksen palveluiden tietty tietoturvan maturiteettitaso. He tekivät myös tiettyjä tietoturvatestauksia ennen julkistusta, jotta pystyivät todentamaan palveluidensa olevan kypsiä Bug Bountylle. Merkittävimpiä riskejä oli, että joku kohdistaa palvelunestohyökkäyksen palveluihin tai palvelut menevät alas liian ison kuorman takia. Jos näin olisi käynyt tai jokin muu riski olisi eskaloitunut Bug Bounty olisi lopetettu toistaiseksi. Muita riskejä oli muun muassa negatiivinen palaute ja ongelmat palkkioiden maksussa.

Palvelun käyttöönoton jälkeisiin riskeihin haastateltavat lukevat muun muassa löydöksen hyväksi käytön riskin. Etenkin jos löydös on Bug Bountyn ulkopuolella. Lähtökohtaisesti valkohattuhakkerit, jotka Bug Bounteihin osallistuvat eivät tätä tee. He arvostavat enemmän palkkioita ja mainettaan. Mustahattuiset hakkerit taas ovat riski ilman Bug Bountyakin. Ne yrityksen palvelut, kun ovat siellä verkossa joka tapauksessa. Myös hakkereiden tunnistamattomuus saattaa olla ongelma. Tämä riippuu paljolti siitä mitkä palvelut ovat kohteena ja kuinka ison riskin LähiTapiola haluaa ottaa. Haastateltavat muistuttavat, että suurin osa palveluista vaatii kirjautumisen. Näin ollen hakkereita on mahdollista jäljittää. Yleensä tämä ei ole ongelma, että löydöksiä käytetään väärin. Myös anonyymien hakkereiden kanssa kommunikointi on tietyn tasoinen riski. Haastatteluissa kommentoitiinkin, että ole aina ”vainoharhainen” ja älä sano mitään asiatonta tai kerro liikaa. Et voi tietää kuka siellä toisessa päässä oikeasti on.

Rikosoikeudellisten seuraamusten vaikeus ulkomaalaisia kohtaan on tiedossa. Eli jos ulkomailla oleva hakkeri tekee jotain laitonta hänen saamisensa oikeuden eteen, on vaikeaa ja aikaa vievää. Tässä on riski, joka tiedostetaan ja jos ei ole kyse isosta rikkeestä saattaa olla helpompi vain antaa asian olla. Toki yleensä rikollisiin tekoihin syyllistyvä hakkeri taas kerran tekee rikoksen ilman Bug Bountya. Lopuksi haastatteluiden aikana todettiin, että Bug Bountyyn osallistuvat eivät ole niitä pahoja hakkereita ja Bug Bounty ei tee rikollisista hyviä.

5.7 Tilastoja

Tässä luvussa kuvataan LähiTapiolan Bug Bountyyn liittyviä tilastoja lyhyesti. Tilastojen tarkoitus on auttaa hahmottaman tutkimuksen tuloksia paremmin.

Ensimmäinen taulukko (taulukko 3) kertoo LähiTapiolan Bug Bountyn keskiarvot mm. maksettujen palkkioiden ja ratkaistujen bugien mukaan.

TAULUKKO 3 Bug Bounty ”All time metrics” marraskuu 2018 LähiTapiola

Ratkaistut bugit	212
Kiitettyjen hakkereiden määrä	198
Palkittujen raporttien määrä	271
Maksettuja palkkioita yhteensä	126 090 $
Palkkioiden keskiarvo	465 $
Maksu% ratkaistuista raporteista	94, 8 %
Vasteaika	10 tuntia
Raportin luokittelu/lajitteluaika	25 päivää
Palkkion maksun keskimääräinen aika	Noin yksi (1) kuukausi
Ratkaisuaika	Kaksi (2) kuukautta

Seuraava taulukko (taulukko 4) havainnollistaa hyvien ja huonojen raporttien suhdetta LähiTapiolan Bug Bountyssa.

TAULUKKO 4 LähiTapiolan Bug Bounty raporttien suhde

Millainen raportti	**% osuus kaikista raporteista**
Ohjelman ulkopuolelta (Out-of-Scope)	20 %
Huono tai duplikaatti	20 %
Hyväksytty raportti	60 %

Alla oleva taulukko (taulukko 5) listaa vielä yleisimmät löydetyt haavoittuvuustyypit.

TAULUKKO 5 Yleisimmät haavoittuvuustyypit LähiTapiolan Bug Bountyssa

Haavoittuvuustyyppi
XSS
CTR
Injektiot
Tietojen vuotaminen
Sessionhallintaongelmat
Konfiguraatiovirheet

5.8 Mitä opittiin ja vietiin käytäntöön

Tämän viimeisen luvun tarkoituksena on lyhyesti kuvata mitä LähiTapiola on Bug Bounty projektinsa aikana oppinut. Opit pyritään kuvamaan siten, että

muutkin yritykset voivat hyödyntää niitä helposti omissa Bug Bounty projekteissaan.

Useat haastateltavat kertoivat oppineensa uusia tapoja ajatella tietoturvaa. Myös hakkereiden ajatusmalleja oli opittu. Yhteistyö hakkereiden kanssa on tehnyt LähiTapiolan tietoturva-asiantuntijoista itsestäänkin parempia hakkereita. Kaikki haastateltavat kertoivat ammatillisen osaamisensa kehittyneen huomattavasti. Bug Bountyn pyörittäminen on opettanut, että sillä voidaan parantaa tietoturvaa ja varmistaa tietoturvaa. Bug Bountyn myötä haastateltavat ovat heränneet siihen, että Suomi ei ole mikään lintukoto. Myös Suomalaisia palveluita yritetään hakkeroida.

Kansainvälisessä ympäristössä toimiminen on opettanut haastatelluille myös sen, että etiikan ja moraalin rajat määräytyvät eri maissa ihan eri tavoilla. Se mikä Suomessa on väärin voi toisessa maassa olla täysin hyväksyttävää. Bug Bounty hanke on opettanut myös sen, että tietoturvaan tulee suhtautua vakavasti ja siihen on syytä käyttää rahaa. Tämän projektin myötä haastatellut kertoivat myös oppineensa huomioimaan uusia asioita ja vaatimaan enemmän toimittajilta. Bug Bountyn myötä LähiTapiola on huomannut myös, että toimittajien tietoturvatestausta on syytä yhteismitallistaa, jotta se olisi samaa tasoa. Toimittajien tietoturvatestauksen ollessa liian eri tasolla keskenään tulee tästä ongelmia.

6 JOHTOPÄÄTÖKSET

Tämän tutkimuksen tavoitteena oli selvittää mitä hyötyjä Bug Bounty on tuonut LähiTapiolalle. Tavoitteena oli myös osoittaa olettama todeksi, jonka mukaan Bug Bountyn käyttöönotosta on ollut LähiTapiolalle enemmän hyötyä, kuin haittaa. Viimeisenä tavoitteena oli luoda pioneeritutkimusta Suomen yrityskentässä ja valottaa muille yrityksille mitä hyötyjä Bug Bountyn käytöstä on. Tutkimuksella haluttiin myös osoittaa Bug Bountyn olevan turvallinen tapa parantaa tietoturvaa. Tutkimus toteutettiin lyhyenä kirjallisuuskatsauksena, jonka pohjalta tehtiin tapaustutkimus.

6.1 Vastaukset tutkimuskysymyksiin

Tämä tutkielma koostui kolmesta tutkimuskysymyksestä. Tässä luvussa kerrataan kysymykset ja esitetään lyhyesti vastaukset niihin luvun 5 pohjalta.

Tutkimuskysymys 1: Mitä hyötyjä LähiTapiolalle on ollut Bug Bountyn käyttöönotosta?

Teorian mukaan Bug Bounty tuo yritykselle positiivista julkisuutta ja toimii tietoturvan varmistavana tekijänä. Tutkimuksen mukaan LähiTapiola on saanut positiivista julkisuutta Bug Bountyn myötä. He ovat onnistuneet parantamaan sovellustensa tietoturvaa ja luomaan selkeän prosessin tuotantoonvienneille. Myös LähiTapiolan asiakkaiden luottamus yritykseen on parantunut Bug Bountyn myötä. Bug Bountyn käyttöönotto on myös vapauttanut asiantuntijoiden aikaa LähiTapiolassa vaativampiin tehtäviin. Bug Bountyn avulla on onnistuttu myös vähentämään löytyneitä haavoittuvuuksia.

Tutkimuskysymys 2: Mitä ongelmia ja riskejä Bug Bountysta on ollut LähiTapiolalle?

Tutkimuksen mukaan Bug Bounty ei ollut tuonut LähiTapiolalle mitään varsinaisia ongelmia. Riskejä tunnistettiin kylläkin muutamia. Merkittävimpinä riskeinä nähtiin palvelunestohyökkäys palveluun Bug Bountyn aloittamisen jälkeen tai palveluiden kaatuminen suuren käyttäjämäärän vuoksi. Kumpikaan riski ei toteutunut. Riskeiksi tunnistettiin myös negatiivien julkisuus ja palaute.

Nämäkään riskit eivät toteutuneet. Kaikki muutkin riskit olivat spekulaatiota. Tutkimuksessa ei pystytty todentamaan yhtään varmaa riskiä, joka olisi realisoitunut. Ainoa merkittävä mahdollisesti realisoituva riski on löydöksen hyväksikäyttö rikollisiin tarkoituksiin. Tämänkään riskin ei voitu todeta johtuvan Bug Bountyn käyttöönotosta, sillä palvelut ovat verkossa saatavilla ilman Bug Bountya ja pahantahtoinen taho voi hyödyntää haavoittuvuutta ilman Bug Bountya.

Tutkimuksen puitteissa ei löydetty mitään varsinaisia ongelmia, joita Bug Bounty olisi LähiTapiolalle tuonut. Haittoja tosin tunnistettiin joitakin. Alkuun oli ollut hieman epäselvää mikä Bug Bounty on ja mitä nyt tapahtuu. Palvelun alkuvaiheessa LähiTapiolan palvelut olivat noin päivän verran hitaat, kun käyttäjämäärä kasvoi rajusti. Tämäkin haitta saatiin korjattua hyvin nopeasti ja nyt palvelu kestää isommankin kuorman. Haittoina tunnistettiin liian pitkä prosessi haavoittuvuuden löytymisestä sen korjaamiseen ja korjaamisen viemiseksi tuotantoon. Bug Bountyn kansainvälisyys aiheutti myös selkeitä haasteita. Kommunikaatio koettiin välillä vaikeaksi ulkomaille kielimuurin takia. Myös vähäinen budjetti tietoturvaa kohtaan ja sovellusten tekninen velka tunnistettiin haasteiksi.

Tutkimuskysymys 3: Mitä opittiin ja mitä vietiin käytäntöön.
Merkittävimmiksi opeiksi tunnistettiin uudet ajattelutavat tietoturvan suhteen ja tietoturvan parempi ymmärtäminen. Bug Bountyn voitiin todentaa myös kehittäneen kaikkien siihen osallistuneiden ammatillista osaamista. Bug Bountya voidaan pitää tietoturvan varmistavana tekijänä ja sillä voidaan parantaa tietoturvaa. Tutkimuksen aikana todettiin myös, että Suomalaisetkin palvelut ovat hakkereiden kohteena ja tietoturvaan pitää panostaa & käyttää rahaa. Viimeisenä löydöksenä mainittakoon etiikan ja moraalin rajat, jotka vaihtelevat paljon eri maiden välillä.

6.2 Tulosten merkitys, luotettavuus & käytettävyys

Saavutetut tulokset vastaavat kohtuu hyvin tässä tutkielmassa esitettyjen aiempien tutkimusten tuloksia ja teorioiden olettamuksia. Näin ollen niitä voidaan pitää käytettävinä ja merkittävinä. Tulokset ovat merkityksellisisä LähiTapiolalle ja muille yrityksille, jotka haluavat ottaa Bug Bountyn käyttöönsä. Tulokset puoltavat näkemystä siitä, että Bug Bounty parantaa yrityksen tietoturvaa ja sen käyttö on turvallista. Tulosten mukaan yritys saa myös positiivista mainetta Bug Bountyn käytöstä.

Tulosten luotettavuus on aiempiin tutkimuksiin ja teorioihin peilaten hyvä. Tämä tutkimus suoritettiin tapaustutkimuksena, joten haastateltavilta saatiin kerronnallisen keskustelun kautta selkeitä käyttäjäkokemuksia Bug Bountysta. Haastateltavat toimivat eri rooleissa kohdeyrityksessä. Näin ollen heiltä saatu kuva on laaja ja luotettavampi. Toki aina on mahdollista, että haastateltavat ovat tarkoituksella antaneet positiivisemman kuvan asioista, kuin todellisuus on. Haastattelutilaisuudet olivat rentoja ja kaikki haastateltavat olivat paikalla

omasta tahdostaan. He vaikuttivat aidosti kiinnostuneilta ja tyytyväisiltä aiheesta. Joten uskon, että heiltä saadut vastaukset ovat luotettavia.

6.3 Rajoitukset & jatkotutkimusaiheet

Tutkimustulosten rajoitteena oli haastateltujen vähäinen määrä ja vain yksi yritys. Jotta tuloksia voisi paremmin verrata Suomen yrityskenttään olisi hyvä haastatella ihmisiä vähintään kolmesta yrityksestä. Näin saisi luotettavaa vertailudataa ja voisi todentaa, että näkevätkö muut asian samalla tavalla. Nyt ainoa vertailukohde oli muutamat Yhdysvalloissa toimivat yritykset. On huomioitava, että Yhdysvaltojen yrityskenttä ja kulttuuri on erilainen, kuin Suomen. Tutkimustulosten vertaaminen aiempiin tutkimuksiin osoittautui muutenkin vaikeaksi. Tämä siksi, koska Bug Bountyn hyödyistä yrityksille on tehty melko vähän tutkimusta. Aihe on vielä suhteellisen tuore ja sen käyttö ei ole vakiintunut yrityksissä.

Jatkotutkimusaiheita Bug Bountyyn liittyen on vaikka, kuinka paljon. Tämän tutkimuksen kannalta olisi hyödyllistä tietää miten Bug Bountya voidaan hyödyntää vielä paremmin tietoturvan ulkoistamisessa ja varmistamisessa. Myös haavoittuvuuksien mustan pörssin markkinoita voisi tutkia ja miettiä miten pienennetään markkinaosuutta siellä ja saadaan houkuteltua ihmiset Bug Bountyn pariin.

7 YHTEENVETO

Tässä tutkielmassa selvitettiin Bug Bountyn hyötyjä yhtenä tietoturvan varmistavana tekijänä. Tutkielma selvitti myös Bug Bountyn hyötyjä laadunvarmistuksen yhtenä työkaluna. Tutkimuksen pääasiallinen tarkoitus oli selvittää mitä hyötyjä ja haittoja Bug Bountysta on. Samoin tutkimuksen tarkoitus oli tuottaa tietoa siitä voidaanko Bug Bountya pitää tietoturvan varmistavana tekijänä. Tulosten oli tarkoitus antaa yrityksille ajankohtaista tietoa siitä, miten he voivat hyödyntää Bug Bountya. Tutkimuksen avulla pyrittiin myös tuomaan esille Bug Bountyn sudenkuopat ja neuvoja, kuinka se otetaan onnistuneesti käyttöön yrityksissä.

Tutkimuksen tulokset olivat ennakko-odotusten mukaiset ja positiiviset. Bug Bountyn käyttöönoton suurimpina hyötyinä olivat positiivinen julkisuuskuva yrityksille, löydettyjen haavoittuvuuksien konkreettinen väheneminen ja vapautuneet resurssit. Ongelmia ei tutkimuksen aikana löydetty. Riskejä tunnistettiin useita. Näitä olivat muun muassa palveluiden kaatuminen, negatiivinen julkisuus ja hallitsemattoman suuri löydösten määrä. Tutkimuksen aikana tunnistettiin myös joitain haittoja. Näitä olivat lähinnä kielimuurin aiheuttamat epäselvyydet raporteissa ja sääntöjen ymmärtämisessä, löydösten korjaamisen hitaus ja hämärtyneet moraalikäsitykset.

Tutkimus osoitti Bug Bountyn myös lisänneen siihen osallistuneiden henkilöiden ammatillista osaamista. Työntekijät kertoivat hakkerointitaitojensa kohonneen. Myös tietoisuus tietoturvasta ja halu valaista sitä muille oli selkeästi kohonnut. Bug Bountyn huomattiin edesauttavan myös työntekijöiden kykyä toimia kansainvälisessä ympäristössä hyvin erilaisten ihmisten kanssa.

Tutkimustulosten perusteella Bug Bountyn voidaan sanoa toimivan erinomaisesti tietoturvan ja etenkin tietoturvatestauksen varmistavana tekijänä. Jos yrityksen maturiteetti on riittävällä tasolla on erittäin suositeltavaa ottaa Bug Bounty käyttöön. Tulosten perusteella voidaan todeta myös, että Bug Bountyn käyttö on turvallista. Bug Bountyn käytöstä ei voitu todeta aiheutuvan mitään erityisiä riskejä. Tutkimuksen aikana todettiinkin, että vaikka yrityksellä ei olisi Bug Bountya käytössään on ne verkkopalvelut saatavilla verkossa. Näin ollen yrityksen on mahdotonta tietää ja estää kaikkia hyökkäyksiä. Tutkimuksessa todettiin myös, että pahantahtoiset hakkerit iskevät yritykseen oli sillä Bug

Bounty tai ei. Tärkeintä Bug Bountyn käyttöönotossa todettiin olevan riittävä maturiteetti, ennakkoon tehty sisäinen tietoturvatestaus ko. palveluille, riittävät resurssit ja hyvä suunnitelma koko prosessista.

LÄHTEET

Akhawe, D., Finifter, M., Wagner, D. An Empirical Study of Vulnerability Reward Programs. University of California, Berkeley, 22nd USENIX Security Symposium 14.16.8.2013 Washington D.C, 1-16.

Ammann, P., Offutt, J. (2008). Introduction to software testing, Cambridge university press, New York USA.

Arkin, B., Stender, S., McGraw, G. (2005) Software Penetration testing, IEEE Computer Society, Vol 3 issue 1, 84-87.

Bacchus, A. (2017). Bug Bounty field manual: How to Plan, Launch, and Operate a Succesful Bug Bounty Program, Hackerone.

Baskerville, R. (1993). Information Systems Security Design Methods: Implications for Information Systems Development. ACM Computing Surveys Vol. 25, No 4, December 1993, 375-414.

Baskerville, R., Siponen, M., Heikka, J. (2006). Journal of the Association for Information Systems Vol. 7, 11, November 2006, 725-770.

Brauch, H.G., Oswald Spring, U., Mesjasz, C., Grin, J., Kameri-Mbote, P., Chourou, B., Dunay, P., Birkmann, J. Coping with Global Environment Change, Disasters and Security Threats, Challenges, Vulnerabilities and Risks, Springer, 2011, 1-47.

Braude, Eric., Bernstein, M. (2011). Software engineering, second edition, Waveland Press Inc, Illinois.

Bugcrowd. Bugcrowdin verkkosivut. Haettu 28.3.2016. Saatavilla osoitteessa https://bugcrowd.com/what-we-do.

Chen, K., Grossklags, J., Zhao, M. (2014). An Exploratory Study of White Hat Behaviors in a Web Vulnerability Disclosure Program, ACM DL Proceeding SIW ’14 Proceedings of the 2014 ACM Workshop on Security, 51-58.

Chillarege, R. (1999). Software Testing Best Practices. Center for Software Engineering IBM Research. Technical report RC 21457.

Doan, A., Ramakrishnan, R., Halevy, A. (2011). Crowdsourcing systems on the World-Wide-Web. ACM DL Communications of the ACM Vol 54, 4, april 2011, 86-96.

Egelman, S., Herley, C., Oorschot, P. (2013). Markets for zero-day exploits: ethics and implications, ACM NSPW '13 Proceedings of the 2013 workshop on New security paradigms workshop, 41-46.

Gikas, C. (2010). A General Comparison of FISMA, HIPAA, ISO 27000 and PCI-DSS Standards. Information Security Journal: A Global Perspective, 19:132-141, 2010.

Hammon, L., Hippner, H. (2012) Crowdsourcing. University of Bayreuth, Bayreuth, Business & Information Systems Engineering 3/2012, 163-166.

LähiTapiola (2015, 14. syyskuuta). LähiTapiola testaa verkkopalveluidensa luotettavuutta palkinto-ohjelman avulla. Haettu 26.3.2017 osoitteesta http://www.lahitapiola.fi/cs/Satellite?c=LTContent_C&cid=1310386131299&locale=fi&p=1302682498678&pagename=LahiTapiola%2FLTContent_C%2FLTNewsLayout.

Hackerone. HackerOnen verkkosivut. Haettu 28.3.2016. Saatavilla osoitteessa https://hackerone.com/product.

Hackerone. (2016, 9. kesäkuuta). How Bug Bounties Work: A Comic. Haettu 24.11.2018 osoitteesta https://www.hackerone.com/blog/how-a-bug-bounty-works-comic.

Hackerone. (2018, 28. elokuuta). LocalTapiola. Haettu 24.11.2018 osoitteesta https://hackerone.com/localtapiola?view_policy=true.

Holtkamp, P., Soliman, W., Siponen, M. (2019). Reconsidering the Role of Research Method Guidelines for Qualitative, Mixed-methods, and Design Sciene Research. 52nd Hawaii International Conference on System Sciences (HICSS).

IEEE. (2014). Standard for Software Quality Assurance Processes. IEEE Computer Society.

Krishnamurthy, S., Tripathi, k. (2006). The Economics of Open Source Software Development, 165-183.

Jolla. (2016). Jollan verkkosivut. Haettu 21.6.2016. Saatavilla osoitteessa https://sailfishos.org/community/.

LähiTapiola, (2018). Yhtiöryhmätietoa. Haettu 23.11.2018 osoitteesta https://www.lahitapiola.fi/tietoa-lahitapiolasta/lahitapiola-ryhma/yhtioryhmatietoa.

Le Roux, Y. (1993). Information security – the CIA model. ProQuest Central August 1993, 53-56.

Niemelä, L. (2018). LähiTapiolan Bug Bounty -ohjelma. Aalto-yliopisto, Aalto PRO 15. Turvallisuusjohdon kehitysohjelma.

Niemimaa, E., Niemimaa, M. (2017). Information systems security policy implementation in practice: from best practices to situated practices. European Journal of Information Systems 26 (2017), 1-20.

Murturi, A., Kantarci, B., Oktug, S. (2015). A reference model for crowdsourcing as a service. IEEE Conference Publications, 64-66.

Myers, D., Newman, M. (2007). The qualitative interview in IS research: Examining the craft. Information and Organization 17 (2007), 2-26.

Schulmeyer, G. (2007). Handbook of Software Quality, fourth edition, artech house books.

Schulz, P. (2014). Penetration Testing of Web Applications in a Bug Bounty Program. Karlstads universitet, Karlstad university, Karlstad, 1-57.

SDL process: training (2018). Haettu 17.11.2018 osoitteesta https://www.microsoft.com/en-us/SDL/process/training.aspx.

SDL process: requirements (2018). Haettu 17.11.2018 osoitteesta https://www.microsoft.com/en-us/SDL/process/requirements.aspx.

SDL process: design (2018). Haettu 17.11.2018 osoitteesta https://www.microsoft.com/en-us/SDL/process/design.aspx.

SDL process: implementation (2018). Haettu 17.11.2018 osoitteesta https://www.microsoft.com/en-us/SDL/process/implementation.aspx.

SDL process: verification (2018). Haettu 17.11.2018 osoitteesta https://www.microsoft.com/en-us/SDL/process/verification.aspx.

SDL process: release (2018). Haettu 17.11.2018 osoitteesta https://www.microsoft.com/en-us/SDL/process/release.aspx.

SDL process: response (2018). Haettu 17.11.2018 osoitteesta https://www.microsoft.com/en-us/SDL/process/response.aspx.

Siponen, M., Baskerville, R. (2018). Intervention Effect Rates as a Path to Research Relevance: Information Systems Security Example. Journal of the Association for Information Systems, 19 (4) 2018, 247-265.

Siponen, M. (2006a). Information Security Standards Focus on the Existence of Process, Not Its Content. ACM Communications of ACM Vol. 49, No. 8 August 2006, 97-100.

Siponen, M. (2006b). Secure-System Design Methods: Evolution and Future Directions. IT Pro May 1, June 2006, 40-44.

Siponen, M., Willison, R. (2009). Information security management standards: Problems and Solutions. Information & Management 46, 267-270.

Siponen, M. (2005a). An analysis of the traditional IS security approaches: implications for research and practice. European Journal of Information Systems 14, 303-315.

Siponen, M. (2005b). Analysis of modern IS security development approaches: towards the next generation of social and adaptable ISS methods. Information and Organization 15 (2005), 339-375.

Tang, A. (2014). A guide to penetration testing. ACM DL Journal Network Security Volume 2014 Issue 8 August 2014, 8-11.

Visma (2016, 3. lokakuuta). Hack us! Haettu 26.3.2018 osoitteesta http://media.visma.fi/pressreleases/hack-us-1586391.

Walsham, G. (1995). Interpretive case studies in IS research: nature and method. European Journal Information Systems (1995) 4, 74-81.

What is the Security Development Lifecycle? (2018). Haettu 17.11.2018 osoitteesta https://www.microsoft.com/en-us/SDL.

Zhang, L., Zhang, H. (2011). Research of crowdsourcing model based on case study. IEEE Conference Publications, 1-5.

LIITE 1

Taustaa kartoittavat kysymykset:

1. Nimi?
2. Asema yrityksessä?
3. Kauanko olet ollut kyseissä roolissa?
4. Missä muissa rooleissa olet ollut yrityksessä?
5. Oletko konsultti vai LähiTapiolan palveluksessa?
6. Kuinka kauan olet työskennellyt LähiTapiolassa?

Kysymysrunko:
Kertokaa mahdollisimman tarkasti omia kokemusperäisiä tarinoita/caseja Bug Bountysta LähiTapiolassa. Alla olevat kysymykset ovat tueksi. Sana on vapaa tässä kohtaa.

7. Kun Bug Bounty otettiin käyttöön, oliko sille vastarintaa?
8. Jos kyllä, niin millaista se oli (omin sanoin)?
9. Mitä mieltä olet osallistumisesta IT-osaston pyytämiin tietoturvan työpajoihin?
10. Viekö IT:n hommiin osallistuminen liikaa aikaasi?
11. Paljonko varsinainen tietoturvatestaus vie aikaasi?
12. Paljonko Bug Bounty yleisesti ottaen vie aikaasi?
13. Miten suhtaudut Bug Bountyyn?
14. Miten johto mielestäsi suhtautuu Bug Bountyyn?
15. Mitä haittoja Bug Bounty on mielestäsi tuonut yritykselle?
16. Onko Bug Bounty tuonut yrityksellenne riskejä tai ongelmia?
17. Miten näet Bug Bountyn juridiselta kannalta?
18. Mitä hyötyjä Bug Bounty on tuonut yrityksellenne?
19. Millaista palautetta olette saaneet Bug Bountysta? (Asiakkaat, hakkerit, johto, kollegat yms.)?
20. Mitä opitte
21. Mitä veitte käytäntöön?
22. Millaista on työskentely hakkereiden kanssa?
23. Millaisia raportteja teille tulee. Mitä hyvää/huonoa niissä on?
24. Kertokaa hieman löydöksistä. Mm. niiden vakavuudesta, korjausprosessista, raportointikäytännöistä yms.?
25. Noudattavatko hakkerit hyvin asettamianne sääntöjä?
26. Miten tämä mielestänne tukee kokonaisvaltaista laadunvarmistusta (QA)?
27. Onko testausorganisaatio mielestänne järjestäytynyt hyvin vai huonosti?
28. Kertokaa miten oma toimenkuvanne istuu tähän prosessiin?
29. Mitä muuta haluat sanoa tai tuoda esiin? Tämän tutkimuksen tarkoitus on kartoittaa miten olette kokeneet Bug Bountyn, mitä hyötyjä siitä on ollut ja miten asiat voidaan tehdä paremmin, joten nyt on mahdollisuus vaikuttaa.