Tietojenkäsittelytiede **25** Joulukuu 2006 sivut 10–27
Toimittaja: Antti Valmari

Tietojärjestelmien tietoturvavaatimusten ja -uhkien mallintaminen väärinkäyttötapausten avulla

Juhani Heikka
Oulun yliopisto
Tietojenkäsittelytieteiden laitos
juhani.heikka@tol.oulu.fi

Tiivistelmä

Verkottuvassa nyky-yhteiskunnassa on tärkeää kiinnittää huomiota kehitettävien tietojärjestelmien tietoturvallisuuteen. Nykyisin käytössä olevat tietojärjestelmien suunnittelumenetelmät ja mallinnuskielet eivät kuitenkaan tarjoa apua järjestelmien tietoturvaominaisuuksien käsittelyyn. Ratkaisuksi ongelmaan on esitetty väärinkäyttötapauksia, joiden avulla voidaan selvittää järjestelmän tietoturvavaatimuksia ja suunnitella toimivia suojausmekanismeja. Aiempi väärinkäyttötapausten tutkimus on keskittynyt käsitteelliseen tutkimukseen irrallaan käytännöstä. Tämä toimintatutkimus tarjoaa empiiristä aineistoa väärinkäyttötapausten integroinnista osaksi kehitysprosessia ja syntyneen lähestymistavan soveltamisesta käytännössä.

1 Johdanto

Yhdysvalloissa vuonna 2005 tehdyn tietoturvaselvityksen mukaan virusinfektiot ja tietojärjestelmiin murtautumiset ovat yleistyneet ja aiheuttavat merkittäviä taloudellisia menetyksiä eri alojen organisaatioille. Selvitykseen osallistuneista 700 organisaatiosta yli puolet (56 %) oli joutunut tietojärjestelmiin murtautumisen tai niiden luvattoman käytön kohteeksi viimeksi kuluneiden 12 kuukauden aikana [23]. Suomessa vuoden 2006 ensimmäisen puoliskon aikana on ilmoitettu yli 150 hyökkäysyritystä, joista joka kymmenes (17 kpl) on johtanut tietomurtoon [10].

OWASP:n tekemän tutkimuksen mukaan merkittävä osa järjestelmien tietoturvahaavoittuvuuksista johtuu kehitysprosessin aikana tehdyistä suunnitteluun ja toteutukseen liittyvistä virheistä [27]. Yksi merkittävä haaste tietoturvallisten tietojärjestelmien kehitykselle on se, etteivät olemassa olevat suunnittelumenetelmät ja mallinnuskielet tue tietoturvaominaisuuksien suunnittelua tai mallintamista [4, 37, 39]. Tietojärjestelmätieteen tunnetuimatkin ja yleisesti hyödynnetyt teokset, esim. [30, 32], tyytyvät kuittaamaan tietoturva-asiat muutamalla lyhyellä kommentilla. Toisin sanoen tietojärjestelmien kehityksessä tavallisimmin käytettävät suunnittelumenetelmät eivät tarjoa loogista tukea järjestelmien tietoturvaominaisuuksien suunnittelulle tai toteutukselle [3, 14, 15, 26].

OHJE KIRJAPAINOLLE: B5-arkin vasen- ja yläreuna kohdistetaan A4-arkin vasempaan ja yläreunaan. Näin pitäisi marginaaliksi tulla taitteen puolella noin 33 mm ja muualla noin 22 mm.

Kansallisesti ja taloudellisesti merkittävien järjestelmien, kuten energia-, taloushallinto- ja telekommunikaatiojärjestelmien, suojaamisessa hyödynnetään erilaisia teknisiä tietoturvaratkaisuja, kuten palomuureja ja virustorjuntaa. Teknisten ratkaisujen ja erillisten tietoturvapäivitysten ongelmana on se, että ne lisätään yleensä valmiisiin tietojärjestelmiin, eikä niiden ominaisuuksia oteta huomioon varsinaisessa järjestelmän kehitysprosessissa [4, 33]. Tällöin eri ratkaisujen yhteensovittamisessa voidaan joutua tekemään kompromisseja järjestelmän tietoturvan ja toiminnallisuuden välillä, mikä voi lyhentää järjestelmän elinkaarta tai pahimmassa tapauksessa estää sen käytön kokonaan [3].

Yhteensopivuus- ja tietoturvaongelmien välttämiseksi järjestelmien tietoturvaominaisuudet tulee ottaa huomioon läpi kehitysprosessin [9, 39, 42]. Ottamalla huomioon järjestelmän tietoturvaominaisuudet heti kehitysprosessin alusta asti voidaan järjestelmään tuottaa loppukäyttäjän kannalta hyödyllisiä ominaisuuksia [44]. Esimerkiksi SSH-yhteys toteuttaa samat toiminnot kuin Telnet, joiden lisäksi SSH:ssa on tietoturva on otettu huomioon salaamalla ja autentikoimalla yhteys [31].

Tietoturva-alan tutkimus nimittää ilmiötä kaksijakoisuusongelmaksi, joka tarkoittaa sitä, että tietojärjestelmä ja sen tietoturvaomaisuudet kehitetään toisistaan erillään [3, 4]. Kaksijakoisuudesta johtuen näillä erillisillä kehityslinjoilla on toisistaan poikkeavat tavoitteet: tietojärjestelmäkehittäjät pyrkivät maksimoimaan järjestelmän toiminnallisuuden ja tietoturvakehittäjät pyrkivät rajoittamaan riskialtista toiminnallisuutta [3]. Tilannetta hankaloittaa myös se, että eri kehitysryhmien välillä ei tapahdu riittävästi kommunikointia [49]. Tämä johtuu usein siitä, että ryhmät käyttävät eriäviä menetelmiä eikä ryhmillä ei ole tarvittavia työvälineitä, esimerkiksi selkeitä notaatioita, vaivattomaan kommunikointiin [26].

Tietoturvan huomioon ottavien suunnittelumenetelmien puuttuessa kehittäjät hyödyntävät työssään tietoturvatarkastuslistoihin ja -standardeihin perustuvia menetelmiä [3, 41, 42]. Tavallisesti kehityksessä hyödynnettävät tietoturvamenetelmät tai tarkistuslistat perustuvat kolmeen tunnetuimpaan tietoturvastandardiin, jotka ovat ISO/IEC 17799 [19], Standard of Good Practices [18] ja Common Criteria [11, 48]. Tarkastuslistatyyppisten menetelmien [12, 46] ongelmana on, että ne eivät ota huomioon asiakasorganisaatioiden yksilöllisiä tietoturvavaatimuksia vaan tarjoavat ratkaisuksi jo ennalta tunnettuja ratkaisuja tai yleispäteviä nyrkkisääntöjä [4, 40]. Tarkistuslistat ja standardit eivät myöskään tue tietoturvavaatimusten mallintamista tai suunnittelua ja soveltuvat siksi paremmin tietoturvavaatimusten auditoimiseen.

Yksi lähestymistapa kaksijakoisuusongelman ratkaisemiseksi on muokata olemassa olevia suunnittelumenetelmiä niin, että niiden avulla voidaan käsitellä tietoturvavaatimuksia [13, 42]. Oliopohjaisessa tietojärjestelmäkehityksessä tavallisimmin hyödynnetty mallinnuskeino on UML (Unified Modeling Language) ja sen avulla tuotetut suunnitelmat, kuten käyttötapaukset, luokka- ja sekvenssikaaviot. Edellä mainitut keinot soveltuvat hyvin toiminnallisten vaatimusten käsittelyyn eli niiden avulla voidaan mallintaa mitä järjestelmällä täytyy pystyä tekemään. Perinteisten UML-suunnitelmien ongelmana on etteivät ne sovellu riittävän hyvin ei-toiminnallisten vaatimusten käsittelyyn. Tietoturvavaatimukset ovat usein luonteeltaan ei-toiminnallisia vaatimuksia, kuten luottamuksellisuus, eheys tai saatavuus, joista ei suoraan selviä mi-

ten vaatimus tulee kehitysprosessin aikana käsitellä [14]. UML ei myöskään tue järjestelmän toimintaa rajoittavien vaatimuksien käsittelyä. Sen avulla ei esimerkiksi voida kuvata sitä mitä järjestelmällä ei pidä pystyä tekemään [37].

Tiedeyhteisö on ottanut tavoitteekseen tuottaa käytännöllisiä ja helposti omaksuttavia tietoturvamenetelmiä, koska suurin osa kehittäjistä ei tunne formaaleja tai matemaattisia mallinnusmenetelmiä [26]. Käytännöllisille tietoturvamenetelmille on tarvetta, koska kehittäjät joutuvat usein toteuttamaan tietoturvan kannalta kriittisiä sovelluksia, esimerkiksi telekommunikaatio- ja potilastietojärjestelmiä, eikä heillä ole siihen tarkoitukseen soveltuvia työkaluja.

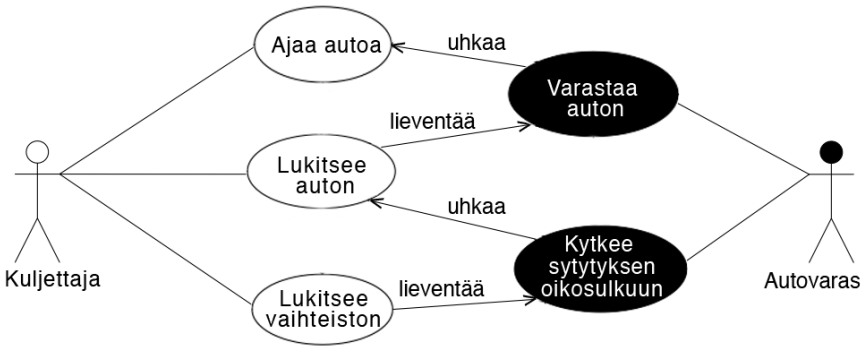
Yksi lupaavimmista tutkimuksen haaroista on UML-pohjaisten tietoturvamenetelmien kehitys UML:n helppokäyttöisyyden ja työkalutuen avulla saavuttaman suosion ansiosta. UML-pohjaisia tietoturvamenetelmiä on esitelty useita [16, 20, 21, 29, 28, 36, 38, 47]. Tässä tutkimuksessa keskitytään tutkimaan väärinkäyttötapausten [26, 37] soveltamista käytäntöön. Tavoitteena on kokonaisvaltainen lähestymistapa tietoturvavaatimusten käsittelyyn, jossa väärinkäyttötapauksia ja UML-mallinnuskieltä hyödyntämällä voidaan kattaa koko kehitysprosessi aina vaatimusmäärittelystä testaukseen ja ylläpitoon asti.

Koko kehitysprosessin kattava lähtökohta on tärkeää, jotta eri kehitysryhmien tai -vaiheiden välille ei jää mustia aukkoja, joissa tietoa pääsee katoamaan työvälineiden yhteensopimattomuudesta johtuen. Aukot saattavat aiheuttaa kommunikaatiokatkoksia tai erilaisia väärinkäsityksiä. Tämän tutkimuksen ensisijaisena tarkoituksena on tarjota kehittäjille väline tietoturvavaatimusten käsittelyyn. Lisäksi tavoitteena on tarjota yhteinen kieli kommunikaation edistämiseksi eri kehitysryhmien välillä. Ihannetilanteessa molemmat ryhmät voivat muodostaa tiettyä menetelmää hyödyntämällä yhteisen runkosuunnitelman, josta selviää järjestelmän toiminnallisuus, toimintoihin liittyvät tietoturvariskit ja riskeiltä suojautuminen.

2 Väärinkäyttötapausten esittely

Useat UML-mallinnuskielen tutkimukset keskittyvät mallien analysointiin ja hyödyntämiseen eri sovellusalueilla [22]. UML:n soveltamiseen tietoturvanäkökulmasta on esitetty useita vaihtoehtoisia malleja, joista tämän tutkimuksen lähtökohdaksi valittiin väärinkäyttötapaukset. Väärinkäyttötapauksiin päädyttiin siksi, että aiemmat käsitteelliset tutkimukset [26, 43] ovat osoittaneet menetelmän lupaavaksi lähestymistavaksi tietoturvavaatimusten ja -uhkien käsittelemissä. Väärinkäyttötapausten alustavan analyysin perusteella näytti siltä, että menetelmä on integroitavissa kehitysprosesseihin, joissa hyödynnetään UML-mallinnuskieltä. Täten lähestymistapa vaikutti hyvältä lähtökohdalta tämän tutkimusprojektin kannalta.

Oliopohjaisessa tietojärjestelmäkehityksessä käyttötapausten avulla kuvataan toiminnot, joita kehitettävässä järjestelmässä halutaan olevan. Käyttötapausten mallintamisessa UML:ssä hyödynnetään tikku-ukkoja ja ellipsejä, joiden avulla kuvataan kunkin järjestelmää käyttävän toimijan (aktorin) mahdollisesti hyödyntämät toiminnot. Perinteisten käyttötapauksien tai käyttötapauskaavioiden ongelmana tietoturvamielessä on se, että ne eivät huomioi toimintoja tai niihin liittyviä riskejä, joita järjestelmässä ei haluta tapahtuvan. Ratkaisuksi ongelmaan on esi-



Kuva 1: Yksinkertainen esimerkki väärinkäyttötapausten soveltamisesta [1]. Esimerkissä auton omistaja yrittää suojautua autovarkailta lukitsemalla auton ja vaihteiston.

tetty väärinkäyttötapauksia, joiden avulla kuvataan kehitettävän järjestelmän toimintaa uhkaavia tietoturvariskejä. Väärinkäyttötapauksia voidaan käyttää järjestelmässä olevien tietoturvan kannalta tärkeiden tietojen tunnistamiseen, joiden suojaamiseen järjestelmän kehitysprosessissa täytyy kiinnittää huomiota.

Väärinkäyttötapauksia hyödyntämällä voidaan analysoida tietoturvavaatimuksia ja muuntaa ei-toiminnallisia vaatimuksia loogisiksi toiminnoiksi ja suojausmekanismeiksi [37]. Esimerkiksi puhuttaessa sähköpostiviestin luottamuksellisuudesta tai oikeellisuudesta, voidaan väärinkäyttötapauksia hyödyntämällä analysoida viestin välittämiseen liittyviä riskejä ja johtaa tietoturvavaatimuksia. Väärinkäyttötapausten avulla voidaan tunnistaa tahot, joilla on oikeus viestin sisältöön ja esittää vaihtoehtoisia ratkaisuja, miten viestin luottamuksellisuus voidaan taata. Tässä tapauksessa kyseeseen voi tulla viestin tai yhteyden salaaminen tai sähköinen allekirjoittaminen. Väärinkäyttötapausten avulla johdetut tietoturvavaatimukset tai toiminnot voidaan kuvata käyttötapauksina, esimerkiksi lisäämällä käyttötapaus ”salaa viesti”.

Väärinkäyttötapaukset voidaan karkeasti jakaa kahteen ryhmään: 1) *järjestelmän luvallisten käyttäjien suorittama väärinkäyttö* ja 2) *ulkopuolisten tahojen suorittama väärinkäyttö* [26].

Esimerkki luvallisen käyttäjän suorittamasta väärinkäytöstä voi olla organisaation työntekijän yritys päästä käsiksi hänelle kuulumattomaan tietoon tai tiedostoon. Järjestelmän luvallisten käyttäjien aiheuttamat uhat on tärkeää ottaa huomioon, koska useassa tapauksessa väärinkäyttäjät tulevat organisaation sisältä [24, 25].

Luvallista väärinkäyttäjää kuvaava rooli on hyvä erottaa omaksi roolikseen ja nimetä se vastaavaksi pahantahtoiseksi käyttäjäksi, esimerkiksi ”pahantahtoinen ylläpitäjä”. Ulkopuolisten tahojen suorittama väärinkäyttö sisältää erilaiset virukset, murtautumisyritykset ja palvelunestohyökkäykset, joiden suorittamisesta vastaa jokin ulkopuolinen taho, esimerkiksi rikollisliiga.

Käyttötapauskaavioissa väärinkäyttö voidaan kuvata esimerkiksi mustalla. Kuvassa 1 on esitetty käyttötapauskaavio auton toiminnoista ja niihin liittyvistä riskeistä. Auton luvallinen käyttäjä eli kul-

jettaja voi ajaa autoa, lukita auton ja lukita vaihteiston. Kuljettajan ja auton kannalta haitallisia toimintoja (uhkia) on kuvattu ulkopuolisen tahon eli autovarkaan suorittamien väärinkäyttötapausten avulla. Esimerkissä autovaras yrittää varastaa auton, joka uhkaa auton kuljettajan toimintoa ”ajaa autoa”. Kuljettaja voi lieventää autovarkaista aiheutuvaa uhkaa lukitsemalla auton, joka toimii suojausmekanismina autovarkaita vastaan.

Menetelmä määrittelee kaksi uutta stereotypiaa: *prevents* ja *detects*. Prevents-stereotypia tarkoittaa, että tietty käyttötapaus tai toiminto sisältää ominaisuuden, jonka avulla pyritään ehkäisemään tai lieventämään väärinkäytöksiä. Esimerkiksi järjestelmään murtautumista voidaan ehkäistä rajoittamalla virheellisten salasana-käyttäjätunnus parien arvaaminen kolmeen kertaan, jonka jälkeen käyttäjätili lukitaan.

Detects-stereotypia tarkoittaa, että kuvatun toiminnon avulla pyritään havaitsemaan tai jäljittämään järjestelmässä mahdollisesti tapahtuvat väärinkäytökset. Esimerkkejä tällaisista tapauksista ovat lokitiedostot ja monitorointitoiminnot, joita hyödyntämällä voidaan tunnistaa erilaisia väärinkäytöksiä, kuten palvelunestohyökkäyksiä [37].

Menetelmän hyödyntämisen kannalta on tärkeää yksilöidä väärinkäyttäjät, jotta kehittäjät voivat arvioida väärinkäyttäjien käytössä olevia resursseja (raha, laitteet, yms.), taitoja ja motivaatioita. Näitä tietoja voidaan hyödyntää tietoturvariskien ja suojaustoimenpiteiden arvioinnissa. Esimerkiksi lyhyt salausavain riittää pitämään harrastelijat loitolla, mutta tiedusteluorganisaatioilta suojautumisessa on syytä käyttää pidempää salausavainta. Tämä johtuu siitä oletuksesta, että tiedusteluorganisaatioilla on todennäköisesti käytössään tarvittavaa erikoisosaamista, resursseja ja tehokkaammat välineet kuin yksittäisillä murtautujilla.

3 Väärinkäyttötapauksiin liittyvät tutkimukset

Väärinkäyttötapauksia on tutkittu usean vuoden ajan, mutta siitä huolimatta tutkimus on ollut osittain puutteellista. Tutkimukset [1, 26, 37, 38] ovat keskittyneet esittelemään menetelmää ainoastaan teoreettisesti, ilman empiiristä todistusaineistoa menetelmän hyödyllisyydestä käytännössä. Käytännön hyödyllisyyden tutkiminen on tärkeää, koska sen avulla voidaan selvittää menetelmässä mahdollisesti olevia heikkouksia, jotka eivät ole tulleet esille teoreettisissa esimerkeissä. Lisäksi turvallisten tietojärjestelmien kehityksen kannalta on tärkeää viedä menetelmiä käytäntöön ja tutkia, voidaanko niitä hyödyntämällä saavuttaa tietoturvan kannalta myönteisiä tuloksia.

Aiempi väärinkäyttötapausten tutkimus on tunnistanut muutamia kehityskohteita empiirisen tutkimusaineiston puutteen lisäksi. Yhtenä puutteena on nähty tietoturvavaatimusten priorisointimekanismin ja riskien hallintatyökalun puuttuminen [17, 43]. Tietoturvavaatimusten priorisointimekanismi on käytännön tietojärjestelmäkehityksen kannalta oleellinen, koska useassa tapauksessa budjetti on rajallinen eikä kaikkia ominaisuuksia voida toteuttaa. Tällöin on hyvä pystyä tekemään perusteltuja valintoja eri ominaisuuksien välillä.

Eri sidosryhmien yhteistyössä tekemien runkosuunnitelmien avulla voidaan tunnistaa toimintoja, jotka toteuttamalla voidaan täyttää yhteisiä tavoitteita [49]. Esimerkiksi verkkokauppajärjestelmässä on tärkeää pystyä jäljittämään tehdyt toiminnot tilauksen kiistämättömyys-

vaatimuksen toteuttamiseksi tai mahdollisten väärinkäytösten jäljittämiseksi. Tämä ominaisuus on helposti valjastettavissa esimerkiksi tilausten seurantaan, jolloin voidaan tarjota myös asiakkaalle hyödyllisiä toiminnallisia ominaisuuksia ja samalla ottaa huomioon järjestelmän tietoturvallisuus. Tällaisissa tilanteissa on tärkeä pystyä arvioimaan millaisia riskejä tietyn tietoturvaominaisuuden toteuttamatta jättämisestä voi aiheutua muualla järjestelmässä.

Useasti tietojärjestelmiä ja ohjelmistoja kehitetään iteratiivisesti eli järjestelmä rakennetaan valmiiksi useassa kehityssyklissä [6]. Tämä tarkoittaa sitä, että kaikkia prosessin alussa määriteltyjä toimintoja tai vaatimuksia ei välttämättä toteuteta samalla kertaa. Tällöin on tärkeää, että käytettävä versio väärinkäyttötapauksista soveltuu myös iteratiiviseen kehitykseen. Iteratiivista kehitystä voidaan tukea tarjoamalla kehittäjille tieto siitä, missä iteraatiossa tietty toiminnallisuus tai suojausmekanismi toteutetaan. Tämä voidaan tehdä lisäämällä dokumentteihin tietokenttä, joka määrittää iteraation, jossa tietty ominaisuus toteutetaan [43]. Tällöin toteutetun ominaisuuden oikea toiminta voidaan varmistaa ennen uusien ominaisuuksien lisäämistä järjestelmään.

4 Tutkimuksen suorittaminen kohdeorganisaatiossa

Tässä julkaisussa esitetyt tulokset perustuvat suuressa ohjelmistotalossa tehtyyn toimintatutkimukseen. Tutkimuksen kohdeorganisaatio kehittää tietoturvan kannalta tärkeitä tietojärjestelmiä, kuten taloushallinnon ja sähköisen asioimisen järjestelmiä.

4.1 Toimintatutkimuksen vaiheet

Tutkimuksen kannalta ihanteellinen lähtökohta tietoturvamenetelmien viemiseksi käytäntöön on toimintatutkimus, jossa on kaksi päätavoitetta: 1) tieteellisen aineiston tuottaminen ja 2) käytännön ongelman ratkaiseminen [5, 35, 39]. Täten toimintatutkimuksen avulla voidaan sekä tuottaa tieteellisesti arvokasta tietoa tietoturvamenetelmien käytöstä osana organisaation tietojärjestelmäkehitystä että tarjota kehittäjille työkaluja tietoturvavaatimusten ja riskien mallintamiseen.

Toimintatutkimus muodostuu viidestä tutkimusvaiheesta [5]. Ensimmäisenä suoritetaan *diagnoosivaihe*, jonka tarkoituksena on tunnistaa ja määritellä pääkysymykset, jotka ovat tutkimuksen kohdeorganisaation muutoshalun taustalla. Ongelmien tunnistamisen jälkeen määritellään ja suunnitellaan toimet, joiden avulla ongelmat tai haasteet pyritään ratkaisemaan. *Suunnitteluvaihetta* ohjaa yleensä teoreettinen viitekehys, joka määrittelee tavoiteltavan lopputilanteen ja toimet miten se saavutetaan.

Toteutusvaiheessa suoritetaan interventio eli edellisessä vaiheessa suunnitellut toimet, joiden avulla tavoitetila pyritään toteuttamaan, viedään käytäntöön. Toimien suorittamisen jälkeen *arvioidaan* toteutuivatko halutut muutokset ja saavutettiinko määritelty lopputilanne. Prosessi sisältää *oppimisvaiheen*, jonka tarkoituksena on tuottaa tietoa tutkimuksesta ja toimia pohjana tulevaisuudessa tehtäville interventioille. Yleensä toimintatutkimus nähdään syklisenä prosessina, jossa edellä mainittuja vaiheita toistetaan kunnes ratkaisu tyydyttää määritellyt tavoitteet [5].

4.2 Lähtötilanteen selvittäminen

Tutkimuksen alussa lähtötilannetta selvitettiin useiden kohdeorganisaatioiden asiantuntijoiden haastatteluilla. Haastatteluissa nousi esiin, että organisaatiossa hyödynnettiin tietoturvatarkistuslistoja, jotka ovat erillään tietojärjestelmien kehityksessä hyödynnettävistä työkaluista ja prosesseista. Tämä johti tilanteeseen, jossa vain osa kehitysprojekteista hyödynsi tarkastuslistoja.

Yhdeksi pääongelmaksi koettiin järjestelmällisen lähestymistavan puuttuminen tietoturvavaatimusten käsittelyssä. Useissa kehitysprojekteissa tarvittavien työvälineiden puuttuessa tietoturvaasioiden huomioimisen koettiin jäävän yksittäisten kehittäjien osaamisen varaan. Tästä syystä organisaation edustajat kokivat tärkeäksi, että tietoturvaominaisuudet kattavat koko kehitysprosessin, koska tilanne on usein se, että eri kehitysvaiheista vastaavat eri vastuuhenkilöt. Uuden lähestymistavan toivottiin tuovan apua myös tietoturvavaatimusten dokumentoimiseen.

4.3 Suunnitteluvaihe

Suunnitteluvaihe aloitettiin tapaamisella, jossa kartoitettiin eri mahdollisuuksia tunnistettujen haasteiden ratkaisemiseksi. Tietoturvamenetelmien tutkimus on osoittanut, että suurin osa menetelmistä ei ole integroitavissa osaksi yleisesti käytettäviä kehitysprosesseja [4, 39, 41]. Tämä johtuu siitä, että useat tietoturvamenetelmät hyödyntävät epästandardeja notaatioita pakottaen kehittäjät hyödyntämään niissä määriteltyjä prosesseja ja työkaluja.

Tietoturvamenetelmien joukosta erottui kaksi UML-pohjaista menetelmää, väärinkäyttötapaukset [26, 37] ja UMLsec [20, 21], joita on mahdollista soveltaa tässä yhteydessä. Kohdeorganisaation edustajien kanssa käytyjen keskustelujen perusteella päädyttiin hyödyntämään väärinkäyttötapauksia, koska ne sisälsivät kehittäjille tuttuja elementtejä ja notaation, jonka oletettiin edistävän eri sidosryhmien välistä kommunikointia kehitysprosessin aikana. Lisäksi kohdeorganisaation käytössä olevat CASE-työkalut soveltuvat väärinkäyttötapausten mallintamiseen ja käsittelyyn.

Alkuperäiset väärinkäyttötapaukset ovat osoittautuneet tietyiltä osin rajoittuneeksi ja vaativat hienosäätöä ennen kuin niitä voidaan soveltaa kohdeorganisaation tarpeisiin. Väärinkäyttötapaukset eivät sisällä esimerkiksi tietoturvavaatimusten priorisointiin tarvittavia työkaluja. Kohdeorganisaatiossa tietoturvavaatimusten priorisointi koettiin tärkeäksi ominaisuudeksi kehityksen tehostamisen kannalta. Vaatimusten priorisoinnin avulla asiakkaille voidaan aluksi toimittaa tärkeimmät toiminnot ja niihin liittyvät tietoturvaominaisuudet.

Hienosäädettyjen väärinkäyttötapausten lähtökohdaksi valittiin malli, johon on liitetty priorisointiin ja riskien hallintaan tarvittavia elementtejä [43]. Tietoturvavaatimusten priorisoinnissa hyödynnetään riskin tai väärinkäyttötapauksen arvioitua todennäköisyyttä, esiintymistiheyttä ja niiden toteutuessaan aiheuttamia kuluja. Mallissa riskienhallinnan apuvälineeksi on valittu yksinkertainen riskien hallintaprosessi [34], jonka istuttaminen kohdeorganisaatioon vaatii yhteensovittamista olemassa olevien käytäntöjen kanssa. Edellä kuvattu malli ja alkuperäiset väärinkäyttötapaukset on esitelty teoreettisella tasolla, joten lähestymistavan testaaminen ja edelleenkehittäminen tuottaa arvokasta tietoa lähestymistavan soveltuvuudesta käytäntöön.

Seuraavaksi suunnitteluvaiheessa tehtiin alustavat versiot väärinkäyttötapauk-

sista, jotka perustuivat organisaatiossa hyödynnettäviin käyttötapauksiin ja tutkijoiden esittelemiin väärinkäyttötapauksiin. Vaiheen tarkoituksena oli havainnollistaa kohdeorganisaation edustajille miten menetelmän tarjoamat tietoturvaominaisuudet voidaan integroida osaksi organisaation käytössä olevia dokumentteja ja prosesseja. Vaiheen päätteeksi kohdeorganisaation edustajat arvioivat dokumenttien koeversiot ja arviot käsiteltiin tapaamisessa.

Tapaamisessa kohdeorganisaation edustajat totesivat, että väärinkäyttötapaukset ovat hyvä lähtökohta ongelmien ratkaisemiseksi ja korostivat, että koko kehitysprosessin kattava lähestymistapa on organisaation kannalta tärkeä. Eräs kohdeorganisaation edustaja toivoi lisäksi menetelmää, jonka avulla voitaisiin havainnollistaa kehitettävän järjestelmän tietoturvariskejä asiakasorganisaation johdolle. Lähestymistavan avulla halutaan tarjota asiakkaille mahdollisuus vaikuttaa järjestelmän tietoturvaan liittyviin päätöksiin. Kohdeorganisaation tarvitseman lähestymistavan on oltava yksinkertainen, jotta sen avulla voidaan kuvata tietoturvariskejä myös henkilöille, jotka eivät ole teknisesti suuntautuneita.

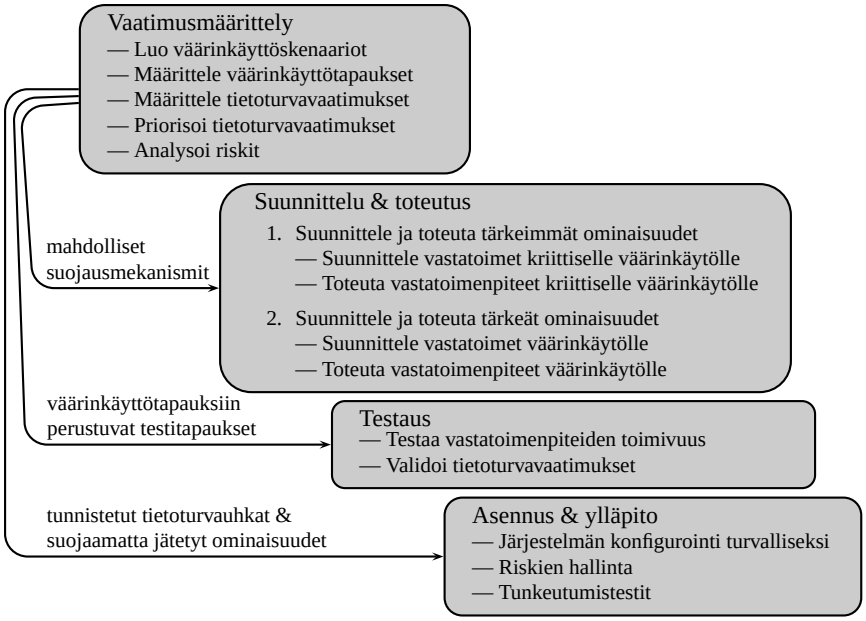
4.4 Väärinkäyttötapausten integrointi kehitysprosessiin

Toteutusvaiheessa väärinkäyttötapaukset integroitiin osaksi kohdeorganisaation kehitysprosessia kolmessa iteraatiossa. Ensimmäinen iteraatio keskittyi väärinkäyttötapausten liittämiseen osaksi olemassa olevia dokumenttipohjia niin, että eri dokumenteissa määritellyt tiedot kulkeutuvat läpi koko kehitysprosessin. Tämä tarkoittaa sitä, että eri dokumentit ja niissä määritellyt asiat täytyy linkittää toisiinsa aina vaatimusmäärittelystä testaukseen ja dokumentointiin asti. Täten vaatimukset ja niiden alkuperä voidaan tarvittaessa jäljittää kehitysprosessin myöhemmissä vaiheissa. Iteraation päätteeksi kohdeorganisaation edustajat arvioivat iteraation tuotokset ja antoivat palautetta tapaamisessa. Lisäksi tarkempaa tietoa ongelmakohdista kerättiin kohdeorganisaation edustajien haastatteluilla.

Kohdeorganisaation edustajat olivat sitä mieltä, että osa väärinkäyttötapauksiin liittyvistä tietoturvaelementeistä oli integroitu vääriin kohtiin dokumentaatiossa. Esimerkiksi väärinkäyttötapauksien määrittely haluttiin erottaa omaksi dokumentikseen, jotta niiden käsittely ei sotke varsinaisten käyttötapausten käsittelyä. Lisäksi osa linkityksistä eri dokumenttien välillä koettiin epäselväksi. He toivoivat, että väärinkäyttötapausten linkitys käännettäisiin toisin päin eli väärinkäyttötapaukset linkitettäisiin käyttötapauksiin eikä päinvastoin. Heidän mukaansa tämä helpottaa tilannetta, jossa yksittäisen väärinkäyttötapauksen loppuun saattaminen voi liittyä useaan käyttötapaukseen. Lisäksi he toivoivat ohjeistusta menetelmän käytöstä ja prosessikartan, joka kuvaa eri vaiheissa suoritettavat tietoturvaan liittyvät tehtävät.

Edellä mainittuja ongelmakohtia lähdettiin korjaamaan toisessa iteraatiossa, jossa kehitettiin yksinkertainen prosessimalli ja ohjedokumentti väärinkäyttötapausten hyödyntämisestä osana kohdeorganisaation käyttämää kehitysprosessia. Toisena tavoitteena iteraatiossa oli löytää ratkaisu siihen, miten asiakasorganisaation johdolle voidaan havainnollistaa kehitettävään järjestelmään liittyviä tietoturvauhkia.

Kehitysprosessin alussa kohdeorganisaatiossa hyödynnetään käyttöskenaarioita, joiden avulla hahmotellaan järjestel-



Kuva 2: Kohdeorganisaation käytössä olevan kehitysprosessin päävaiheet ja väärinkäyttötapauksista johdettujen tietoturvatehtävien suorittaminen.

män käyttäjärooleja ja heidän suorittamia toimintoja. Tutkijan ja kohdeorganisaation edustajien yhteistyönä kehitettiin vastaavantyyppinen työkalu tietoturvauhkien havainnollistamiseen korkealla tasolla. Työkalu nimettiin väärinkäyttöskenaarioiksi, jotka ovat sanallisia kuvauksia järjestelmässä olevasta toiminnallisuudesta ja niiden mahdollisesta soveltamisesta vääriin tarkoituksiin. Väärinkäyttöskenaarioiden avulla asiakas saa yleiskuvan järjestelmän käyttöön liittyvistä riskeistä ja voi varautua riskeiltä suojautumisesta aiheutuviin kuluihin. Väärinkäyttöskenaariot liitettiin osaksi aiemmin kehitettyä prosessia niin, että väärinkäyttöskenaariot toimivat syötteenä myöhemmin kehitysprosessissa tehtäville väärinkäyttötapauksille.

Toisen iteraation lopuksi kohdeorganisaation edustajat arvioivat vaiheen tulokset ja kommentoivat niitä tapaamisessa. Yhdeksi menetelmään liittyväksi ongelmaksi koettiin termistö, jota haluttiin yhtenäistää organisaation käytössä olevien termien kanssa. Väärinkäyttötapausten kuvauksien haluttiin olevan yhtenäisempiä käyttötapausten kanssa ja molemmissa käytettävien metaforien täytyy heidän mukaansa olla samoja. Lisäksi osa eri dokumenteissa määritellyistä tiedoista toistui useassa paikassa. Tämä voi aiheuttaa väärinkäsityksiä kehittäjien välillä tai vaaran, että tiedot eivät ole keskenään samat.

Kolmannen iteraation tarkoituksena oli poistaa eri dokumenteissa oleva tietojen toistaminen ja yhtenäistää dokumentaation termistö. Lisäksi aiemmin kehitetty prosessimalli oli muutosten johdosta vanhentunut, joten se päivitettiin vastaamaan nykyistä prosessia (kuva 2). Iteraa-

tion päätteeksi kohdeorganisaation edustajat arvioivat korjaustoimenpiteet ja totesivat lähestymistavan olevan valmis testaamista varten.

4.5 Väärinkäyttötapausten testaaminen osana kohdeorganisaation kehitysprosessia

Väärinkäyttötapausten integroimisen jälkeen menetelmää testattiin www-pohjaisen tietojärjestelmän kehitysprosessissa. Tietojärjestelmän tietoturva koettiin tärkeäksi, koska sen avulla on tarkoitus jakaa erilaisia ohjelmistopaketteja kohdeorganisaation asiakkaille.

Kehitysprosessin aluksi projektiryhmä lähti miettimään järjestelmän kannalta merkittäviä tietoturvauhkia ja kuvasi niitä kuudessa väärinkäyttöskenaariossa. Projektiryhmä priorisoi väärinkäyttöskenaariot ja totesi riskianalyysissa kolmen skenaarion olevan järjestelmän tietoturvan kannalta merkittäviä.

Vaatimusmäärittelyvaiheessa määriteltiin väärinkäyttötapaukset hyödyntäen tehtyjä käyttötapauksia ja väärinkäyttöskenaarioita. Käyttötapauksia hyödyntäen projektiryhmä yhdisti väärinkäyttötapaukset tiettyihin järjestelmän toimintoihin. Väärinkäyttötapauksista projektiryhmä onnistui johtamaan uusia tietoturvavaatimuksia, kuten tunnistamaan tietoja, jotka ovat tärkeitä tietokannassa tapahtuneiden toimintojen jäljittämisessä esimerkiksi lokitiedoista. Väärinkäyttötapausten avulla tunnistettiin myös järjestelmässä olevia luottamuksellisia tietoja, kuten sopimus- ja laskutustiedot ja käyttäjätunnuksiin liittyvät tiedot.

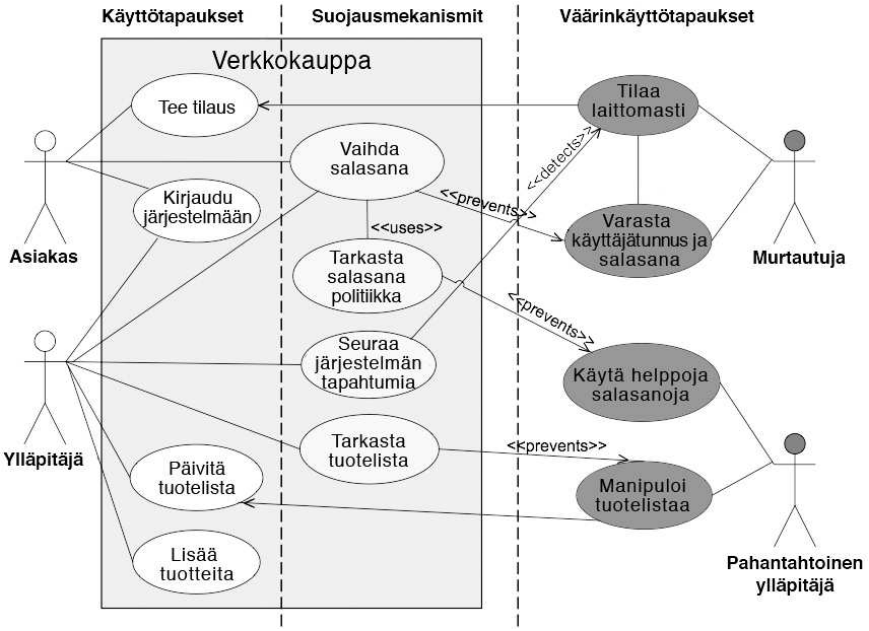
Väärinkäyttötapauksien avulla voitiin luontevasti kartoittaa näihin tietoihin liittyviä tietoturvauhkia ja määritellä niiden suojaamiseksi tarvittavia vaatimuksia ja toimenpiteitä. Yhdeksi tärkeäksi näkökulmaksi muodostui järjestelmän kautta jaettavan aineiston tarkastaminen, jotta järjestelmää hyödyntämällä ei voida tahallisesti tai tahattomasti aiheuttaa virhetilanteita asiakasorganisaation järjestelmissä.

Väärinkäyttötapausten mallintamisessa projektiryhmä totesi alkuperäisten versioiden [26, 37] olevan hieman rajoittuneita. Projektiryhmä halusi korostaa tietoturvavaatimuksia ja väärinkäyttötapausten estämiseksi tehtäviä vastatoimenpiteitä ja kehitti siihen tarkoitukseen kaksi työkalua: laajennetun väärinkäyttötapausmallinnuksen ja käyttötapausmatriisin.

Väärinkäyttötapausten mallintamista laajennettiin lisäämällä kaavioihin värikoodaus: normaalit käyttötapaukset kuvataan valkoisella, väärinkäyttötapaukset punaisella ja erilaiset suojaus- tai vastatoimenpiteet keltaisella (kuva 3). Selkeyden lisäämiseksi kaavio voidaan jakaa kolmeen vyöhykkeeseen: normaaleihin käyttötapauksiin, väärinkäyttötapauksiin ja suojausmekanismeihin. Laajennetusta kaaviosta kaikki kehittäjät voivat nähdä yhdellä silmäyksellä järjestelmän toiminnot, toimintoihin liittyvät tietoturvauhat ja uhilta suojautumiseen tarkoitetut mekanismit.

Käyttötapausmatriisin (taulukko 1) avulla kehittäjät halusivat kartoittaa mitkä käyttötapaukset ja väärinkäyttötapaukset käsittelevät samoja tietoja tai tietokokonaisuuksia. Matriisia hyödyntämällä haluttiin varmistaa, että kaikki järjestelmän tietoturvan kannalta oleelliset tiedot on tunnistettu ja otettu huomioon järjestelmän kehityksessä.

Edellä esitetty taulukko 1 toimii yhteenvetona, jota käytetään pohjana järjestelmän pääsyoikeuksien suunnittelussa. Tässä kehitysprojektissa pääsyoikeudet suunniteltiin CRUD-mallin perusoperaatioiden avulla (C=luonti, R=luku,



Kuva 3: Esimerkkikaavio kuvitteellisen verkkokaupan käyttötapauksista, järjestelmää uhkaavista väärinkäyttötapauksista ja suojausmekanismeista. (Selkeyden vuoksi kuvassa olevat väärinkäyttötapaukset on kuvattu varsinaisen järjestelmän ulkopuolelle.)

Taulukko 1: Käyttötapausmatriisi, jonka avulla voidaan analysoida mitkä käyttötapaukset ja väärinkäyttötapaukset käsittelevät samoja tietokokonaisuuksia.

	Käyttötapaus 1	Käyttötapaus 2	Käyttötapaus …
Väärinkäyttötapaus 1	Salasana ja käyttäjätunnus		
Väärinkäyttötapaus 2		Tilaustiedot	Resurssi x
Väärinkäyttötapaus …			Resurssi y

Taulukko 2: Pääsynhallintamatriisi, josta selviää järjestelmässä olevien tietojen käsittelyyn tarvittavat toiminnot ja kunkin roolin käyttöoikeudet toimintoihin.

	Käyttäjä 1	Käyttäjä 2	Käyttäjä …	Tarvittavat operaatiot
Tieto-objekti 1	R	CRUD	–	CRUD
Tieto-objekti 2	RU	–	C	CRU
Tieto-objekti …	R	R	CR	CR

U=päivitys, D=poisto). Suunnittelussa hyödynnettiin pääsynhallintamatriisia (taulukko 2), josta selviää kullekin järjestelmän käyttäjälle toteutettavat toiminnot suhteessa käsiteltäviin tieto-objekteihin.

Taulukon 2 esimerkissä käyttäjä 1 saa lukea tieto-objektia 1. Käyttäjä 2 voi luoda, lukea, päivittää ja poistaa tieto-objektin 1. Tämä voi kuvastaa tilannetta, jossa käyttäjä 1 on asiakaspalvelussa työskentelevä henkilö, jonka tarvitsee työssään ainoastaan lukea asiakastietoja. Käyttäjä 2 on asiakasrekisterin ylläpitäjä, jonka vastuulla on uusien asiakkaiden lisääminen, asiakastietojen päivittäminen ja vanhojen asiakastietojen poistaminen järjestelmästä. Taulukon oikeassa laidassa oleva yhteenvetosarake kertoo kehittäjille mitkä toiminnot tietyn tieto-objektin käsittelyyn tarvitaan. Esimerkiksi tieto-objektin 1 käsittelemiseksi järjestelmään täytyy toteuttaa kaikki neljä perusoperaatiota. Tieto-objektia 2 ei saa pystyä tuhoamaan, jolloin tuhoamisoperaatiota ei tarvitse toteuttaa ollenkaan.

Tarkastelemalla taulukon 2 yksittäisiä sarakkeita matriisista selviää kaikki tiettyyn käyttäjärooliin liitettävät toiminnot. Tämä helpottaa myöhemmässä vaiheessa tehtäviä käyttäjäroolien toteutuksia.

Tietoturvaominaisuuksien suunnittelussa projektiryhmä hyödynsi aiempien vaiheiden tuotoksia. Esimerkiksi väärinkäyttöskenaarioista ja -tapauksista johdettuja tietoturvariskejä silmällä pitäen projektiryhmä suunnitteli ratkaisun, jonka avulla voidaan jäljittää mahdollisia väärinkäytöksiä. Lisäksi tietoturvavaatimusten pohjalta luotiin käytäntöjä, joita hyödyntämällä voidaan varmistaa järjestelmän kautta jaettavan aineiston oikeellisuus. Eri kehitysvaiheista johdettiin tietoa järjestelmän tekniseen dokumentaatioon, jota hyödynnetään tulevaisuudessa järjestelmän jatkokehityksessä ja ylläpidossa.

4.6 Väärinkäyttötapausten arviointi

Väärinkäyttötapausten integroinnin ja testaamisen jälkeen koko tutkimusprosessin tulokset arvioitiin. Arvioinnin tarkoituksena oli selvittää saatiinko ongelmat ratkaistua ja aiheuttivatko suunnitellut toimet halutut muutokset kohdeorganisaatiossa. Arviointiin osallistuivat kehittäjät, jotka testasivat menetelmän www-tietojärjestelmän kehitysprojektissa. Tiedonkeruumenetelmänä oli teemahaastattelu.

Projektiryhmä totesi väärinkäyttötapaukset helppokäyttöisiksi ja hyödyllisiksi tietojärjestelmien tietoturvariskien ja -ominaisuuksien määrittelyssä. Kehittäjien mukaan väärinkäyttötapaukset istuivat käytössä olevaan prosessiin hyvin ja suurin hyöty niistä on se, että tietoturvavaatimukset tulevat kuvattua ja dokumentoitua. Myönteisenä ominaisuutena nähtiin, että tietoturvavaatimuksia ei tarvitse kuvata kehitysprosessin alussa liian teknisellä tasolla. Lisäksi kehittäjät arvioivat, että menetelmää voitaisiin soveltaa olemassa olevien järjestelmien kehityksessä, esimerkiksi järjestelmien tietoturvatason arvioinnissa ja uusien vaatimusten johtamisessa.

Yhtenä päätavoitteena tutkimuksessa oli kehittää koko kehitysprosessin kattava lähestymistapa kaksijakoisuuden välttämiseksi. Kehittäjien mukaan väärinkäyttöskenaariot auttavat kuvaamaan järjestelmän toimintaympäristöä siten, että tietoturvavaatimukset saadaan määriteltyä ja myöhemmin priorisoitua.

Väärinkäyttöskenaarioiden ja -tapausten huomioon ottaminen luo kehittäjien mukaan realistisen kuvan järjestelmän käyttöön liittyvistä tietoturvauhista ja suojausmekanismeista, mikä helpottaa järjestelmän tietoturvaominaisuuksien suunnittelua. Lisäksi myönteisenä asiana näh-

tiin, että väärinkäyttötapausten avulla voidaan täydentää testitapauksia. Tämä nähtiin hyödylliseksi testaustilanteissa, joissa toimittajan ja asiakkaan välillä on epäselvää onko kyseessä virhe vai määrittelemätön ominaisuus.

Yhteenvetona kehittäjät totesivat väärinkäyttötapausten ja siihen liittyvien dokumenttien auttavan siirtämään määriteltyä tietoa, kuten tietoturvavaatimuksia ja -suunnitelmia, eri kehitysvaiheiden välillä.

Testauksen aikana kehittäjät kohtasivat väärinkäyttötapausten soveltamiseen liittyviä haasteita. Yhdeksi ongelmaksi projektiryhmän edustajat tunnistivat alkuperäisen notaation [37], joka ei heidän mukaansa korostanut riittävästi tietoturvavaatimuksia tai suojausmekanismeja. Kehittäjät ratkaisivat ongelman laajentamalla väärinkäyttötapausten kuvausnotaatiota kohdeorganisaation tarpeisiin sopivaksi. Lisäksi väärinkäyttötapausten kuvaamisessa joudutaan kehittäjien mukaan soveltamaan UML-standardia, mutta se ei heidän mukaansa ole ongelma, kunhan projektiryhmän sisällä on selvyys siitä, mitä laajennetun notaation merkinnät tarkoittavat. Tarvittaessa kaavioita voidaan heidän mukaansa selventää lisäämällä niihin selitystekstejä.

Toiseksi ongelmaksi kehittäjät mainitsivat, etteivät alkuperäiset lähestymistavat auttaneet riittävästi järjestelmässä olevien tärkeiden tietojen ja toimintojen tunnistamisessa. Näiden tietojen tunnistaminen auttaa järjestelmän pääsyoikeuksien suunnittelussa. Tämän puutteen kehittäjät ratkaisivat kehittämällä yksinkertaisen käyttötapausmatriisin, jonka avulla voidaan tarkastaa, mitkä käyttötapaukset ja käyttäjäroolit sekä mitkä väärinkäyttötapaukset ja väärinkäyttäjät käsittelevät samoja tietokokonaisuuksia.

Kolmanneksi käytännön haasteeksi väärinkäyttötapausten soveltamisessa kehittäjät mainitsivat, että joissakin tapauksissa on hankala päättää mihin väärinkäyttötapausten määrittely lopetetaan. Esimerkiksi www-pohjaisissa järjestelmissä potentiaalisia uhkia voi määritellä loputtomiin. Tämän vuoksi väärinkäyttöskenaarioiden priorisointi todettiin tärkeäksi, jolloin voidaan keskittyä järjestelmän kannalta olennaisimpiin tietoturvauhkiin.

5 Yhteenveto ja pohdinta

5.1 Yhteenveto

Nykyinen tietoyhteiskunta perustuu erilaisiin tietojärjestelmiin, tietoverkkoihin ja niiden luotettavaan toimintaan. Tästä syystä tietojärjestelmien tietoturvaan tulee kiinnittää huomiota jo niiden kehitysvaiheessa, esimerkiksi ottamalla huomioon järjestelmän tietoturvavaatimukset ja -uhkat mahdollisimman kattavasti. Ongelmana tietoturvavaatimusten ja uhkien käsittelyssä on tarvittavien menetelmien ja työkalujen puute. Lisäksi menetelmien soveltamisesta käytännössä on vain vähän esimerkkejä. Ilman tarvittavia työkaluja tietoturvan huomioon ottaminen kehitysprosessissa jää usein yksittäisten kehittäjien taitojen varaan.

Tässä tutkimuksessa tietojärjestelmien tietoturvaominaisuuksien kehittämiseen liittyviä haasteita lähestyttiin toimintatutkimuksen avulla. Tutkimuksen tavoitteena oli lisätä tietämystä väärinkäyttötapausten soveltamiseen käytännössä ja ratkaista kohdeorganisaatiossa olleita tietojärjestelmien tietoturvaominaisuuksien käsittelyhaasteita. Toimintatutkimus osoittautui hyväksi lähestymistavaksi, koska sen avulla voitiin yhteistyössä kohdeorganisaation edustajien kanssa tuottaa käytännöllinen ratkaisu tietoturvavaatimusten mallintamiseksi osana koh-

deorganisaation käyttämää kehitysprosessia.

5.2 Kaksijakoisuusongelman ratkaisu

Tutkimuksessa kaksijakoisuusongelma saatiin ratkaistua integroimalla väärinkäyttötapaukset osaksi kohdeorganisaation tietojärjestelmäkehitysprosessia. Integroinnin tarkoituksena oli tarjota kehittäjille työkalu, jota hyödyntämällä he pystyvät määrittelemään, suunnittelemaan ja mallintamaan järjestelmien tietoturvaominaisuuksia ja täten ottamaan huomioon tietoturvaominaisuudet heti kehitysprosessin alusta alkaen. Näin voidaan välttää tilanne, jossa tietoturvaominaisuudet lisätään järjestelmään myöhäisessä vaiheessa kehitysprosessia (esimerkiksi kun testauksessa huomataan, että tarvitaan tietty tietoturvaominaisuus) tai pahimmassa tapauksessa vasta valmiiseen käytössä olevaan järjestelmään.

Kaksijakoisuuden ratkaiseminen on tärkeää myös kustannustehokkuuden kannalta. Tutkimusten mukaan ohjelmistotaloille on edullisempaa kiinnittää huomiota tietoturvaominaisuuksiin heti kehitysprosessin alussa, koska niiden korjaaminen jälkikäteen aiheuttaa merkittäviä lisäkustannuksia [33, 45].

Kovassa kilpailutilanteessa olevat ohjelmistotalot kuitenkin usein tinkivät ”asiakkaalle näkymättömistä tietoturvavaatimuksista”, koska markkinoiden valtaamisessa nopeus on valttia. Ehkä hieman lyhytnäköisestikin ohjelmistotalot kehittävät mieluummin asiakkaiden arvostamaa toiminnallisuutta ja jättävät tietoturvaominaisuudet tulevaisuuden huoliksi ristien sormensa etteivät heikkojen tietoturvaratkaisuiden aiheuttamat haitat tule esille [33]. Tilanteen parantamiseksi tietojärjestelmiä tai ohjelmistoja ostavien tahojen tulee vaatia tietoturvallisia järjestelmiä ja olla myös valmiita maksamaan näiden ominaisuuksien toteuttamisesta [2].

5.3 Tutkimustulosten luotettavuus

Baskerville ja Wood-Harper [7, 8] ovat määrittäneet seitsemänosaisen kriteeristön toimintatutkimuksen validiuden arviointiin. Tässä tutkimuksessa täytettiin kaikki seitsemän kriteeriä: 1) tutkimusympäristö sisälsi runsaasti vuorovaikutusta eri sidosryhmien, kuten tutkijan ja kohdeorganisaation edustajien, välillä, 2) tutkimuksen aikana suoritetut haastattelut dokumentoitiin nauhoitusten ja sähköpostien avulla, jotka analysoitiin tulkitsevassa viitekehyksessä, 3) tutkija suoritti intervention esittelemällä uuden tietoturvamenetelmän kohdeorganisaatiossa, 4) tutkimuksen aikana havainnointiin ja tutkija osallistui aktiivisesti eri tutkimusvaiheissa tehtyihin väärinkäyttötapauksien testauksiin ja havainnoi kehittäjien työskentelyä, 5) tilanteen kehittymistä organisaatiossa seurattiin läpi koko tutkimusprosessin ja väärinkäyttötapausten hyödyntämistä tutkittiin käytännön kehitysprojektissa, 6) kohdeorganisaatiossa ollut ongelma eli tietoturvamenetelmän puuttuminen sekä kaksijakoisuusongelma ratkaistiin tutkimusprosessin aikana ja 7) tutkija reflektoi tulokset takaisin teoriaan selittäen miten suoritetut toimenpiteet johtivat saatuun tulokseen.

5.4 Jatkotutkimushaasteet

Tämä tutkimus tuotti tietoturvamenetelmien edelleenkehityksen kannalta tärkeää empiiristä tietoa väärinkäyttötapausten soveltamisesta käytännössä. Tutkimus osoittaa, että väärinkäyttötapaukset autta-

vat tietoturvavaatimusten selvittämisessä ja niiden käsittelyssä eri kehitysvaiheissa. Menetelmää testanneet kehittäjät pitivät väärinkäyttötapauksia käytännöllisenä ja helppokäyttöisenä lähestymistapana.

Jatkotutkimusten kannalta mielenkiintoisia kysymyksiä ovat esimerkiksi CASE-työkalutuen laajentaminen ja menetelmän edelleenkehittäminen arkkitehtuurivaatimusten käsittelyyn. Työkalutukea voidaan parantaa esimerkiksi luomalla valmiita sapluunoja, joita tutkimalla voidaan selvittää tehostuuko väärinkäyttötapausten käsittely. Toinen haastava kysymys on tutkia voidaanko väärinkäyttötapausten avulla arvioida vaihtoehtoisten järjestelmäarkkitehtuurien paremmuutta järjestelmän tietoturvan kannalta.

Kiitokset

Haluan kiittää tutkimuksen yhteistyökumppaneita ja väitöskirjaohjaajaani FT, YTT Mikko Siposta. Lisäksi kiitokset rakentavista kommenteista TkT Antti Valmarille ja DI Antti Siirtolalle.

Viitteet

[1] Alexander, I. (2003). Misuse Cases: Use Cases with Hostile Intent. IEEE Software, Vol. 20, Issue 1, ss. 58–66.

[2] Anderson, R. (2001). Why Information Security is Hard — An Economic Perspective. Proceedings of 17th Annual Computer Security Applications Conference (ACSAC 2001), ss. 358–366.

[3] Baskerville, R. (1992). The Developmental Duality of Information System Security. Journal of Management Systems, Vol. 4, ss. 1–12.

[4] Baskerville, R. (1993). Information Systems Security Design Methods: Implications for Information Systems Development. ACM Computing Surveys (CSUR), Vol. 25, Issue 4, ss. 375–414.

[5] Baskerville, R. (1999). Investigating Information Systems with Action Research. Communications of the Association for Information Systems, Vol. 2, Issue 3, Article19, ss. 1–32.

[6] Baskerville, R., Levine, L., Pries-Heje, J., Ramesh, B. & Slaughter, S. (2003). Is Internet-Speed Software Development Different? IEEE Software, Vol. 20, Issue 6, ss. 102–107.

[7] Baskerville, R. & Wood-Harper, T. (1996). A Critical Perspective on Action Research as a Method for Information Systems Research. Journal of Information Technology, Vol. 11, ss. 235–246.

[8] Baskerville, R. & Wood-Harper, T. (1998). Diversity in Information Systems Action Research Methods. European Journal of Information Systems, Vol. 7, ss. 90–107.

[9] Booysen, H.A.S. & Eloff, J.H.P. (1995). A Methodology for the Development of Secure Application Systems. Proceedings of the 11th International Conference on Information Security (IFIP TC11), ss. 255–269.

[10] CERT-FI. (2006). Tietoturvaloukkausten havainnointi ja ratkaisu. http://www.ficora.fi/suomi/tietoturva/tilastot.htm (11.8.2006).

[11] Common Criteria. (2006). http://www.commoncriteriaportal.org/ (11.8.2006).

[12] Cranor, L.F. & Garfinkel, S. (2005). Security and Usability: Designing Secure Systems That People Can Use. O'Reilly Media Inc.

[13] Crook, R., Ince, D., Lin, L. & Nuseibeh, B. (2002). Security Requirements Engineering: When Anti-requirements Hit

the Fan. Proceedings of IEEE International Requirements Engineering Conference (RE'02), ss. 203–205.

[14] Devanbu, P. & Stubblebine, S. (2000). Software Engineering for Security: A Roadmap. Proceedings of the Conference on the Future of Software Engineering, ss. 227–239.

[15] Dhillon, G. & Backhouse, J. (2001). Current Directions in IS Security Research: Towards Socio-Organizational Perspectives. Information Systems Journal, Vol. 11, ss. 127–153.

[16] Doan, T., Demurjian, S., Ting, T.C. & Ketterl, A. (2004). MAC and UML for Secure Software Design. Proceedings of ACM Workshop on Formal Methods in Security Engineering, ss. 75–85.

[17] Heikka, J. & Siponen, M. (2006). Abuse Cases Revised: An Action Research Experience. Proceedings of the 10th Pacific Asia Conference on Information Systems (PACIS 2006), ss. 673–684.

[18] Information Security Forum (ISF). (2005). Standard of Good Practices for Information Security. http://www.isfsecuritystandard.com/index_ie.htm (11.8.2006).

[19] International Organization for Standardization (ISO). (2005). ISO/IEC 17799, Information Technology — Code of Practice for IS Security Management, Second Edition.

[20] Jürjens, J. (2002). UMLsec: Extending UML for Secure Systems Development. Lecture Notes in Computer Science, Vol. 2460, Springer-Verlag, ss. 412–425.

[21] Jürjens, J. (2005). Secure Systems Development with UML. Springer-Verlag.

[22] Koskimies, K., Koskinen, J., Maunumaa, M., Peltonen, J., Selonen, P., Siikarla, M. & Systä, T. (2004). UML työvälineenä ja tutkimuskohteena. Tietojenkäsittelytiede, Vol. 21, ss. 19–51.

[23] Lawrence, G.A., Loeb, M.P., Lucyshyn, W. & Richardson, R. (2006). 2005 CSI/FBI Computer Crime And Security Survey. http://americas.utimaco.com/encryption/fbi_csi_2005_p1.html (11.8.2006).

[24] Marcinkowski, S.J. & Stanton, J.M. (2003). Motivational Aspects of Information Security Policies. Proceedings of IEEE International Conference on Systems, Man and Cybernetics, ss. 2527–2532.

[25] Mattia, A. & Dhillon, G. (2003). Applying Douple Loop Learning to Interpret Implications for Information Systems Security Design. Proceedings of IEEE International Conference on Systems, Man and Cybernetics, ss. 2521–2526.

[26] McDermott, J. & Fox, C. (1999). Using Abuse Case Models for Security Requirements Analysis. Proceedings of the 15th Annual Computer Security Applications Conference (ACSAC '99), ss. 55–64.

[27] Open Web Application Security Project (OWASP). (2006). Top Ten Most Critical Web Application Security Vulnerabilities. http://www.owasp.org/documentation/topten.html (7.3.2006).

[28] Pauli, J.P. & Xu, D. (2005). Misuse Case-Based Design and Analysis of Secure Software Architecture. Proceedings of International Conference on Information Technology, ss. 398–403.

[29] Popp, G., Jürjens, J., Wimmel, G. & Breu, R. (2003). Security-Critical System Development with Extended Use Cases. Proceedings of the 10th Asia-Pacific Software Engineering Conference (APSEC'03), ss. 478–487.

[30] Pressman, R.S. (2000). Software Engineering, A Practitioner's Approach. European Adaptation, Fifth Edition, McGraw-Hill Publishing Company.

[31] Rosasco, N. & Larochelle, D. (2004). How and Why More Secure Technologies Succeed in Legacy Markets. In Economics of Information Security. Edited by Camp, L.J. & Lewis, S. Advances in Information Security, Vol. 12, Kluwer Academic Publishers, ss. 247–254.

[32] Rumbaugh, J., Jacobson, I. & Booch, G. (1999). The Unified Modeling Language Reference Manual. Addison Wesley.

[33] Råman, J. (2004). Network Effects and Software Development — Implications for Security. Proceedings of the 37th Hawaii International Conference on System Sciences (HICSS 2004), s. 70186a.

[34] Saltmarsh, T.J. & Browne, P.S. (1983). Data Processing — Risk Assesment. In Advantages in Computer Security Management, Vol. 2, John Wiley and Sons Ltd, ss. 93–116.

[35] Schein, E. (1987). Clinical Perspective in Fieldwork. Sage Publications.

[36] Shirazi, M.R.A., Jaferian, P., Elahi, G., Baghi, H. & Sadeghian, B. (2005). RUP-Sec: An Extension on RUP for Developing Secure Systems — Requirements Discipline. Transactions on Engineering, Computing and Technology, Vol. 4, ss. 208–212.

[37] Sindre, G. & Opdahl, A.L. (2000). Eliciting Security Requirements by Misuse Cases. Proceedings of 37th International Conference Technology of Object-Oriented Languages and Systems (TOOLS 2000), ss. 120–131.

[38] Sindre, G., Opdahl, A.L. & Firesmith, D. (2003). A Reuse-Based Approach to Determing Security Requirements. Proceedings of 9th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'03), ss. 104–114.

[39] Siponen, M. (2002). Designing Secure Information Systems and Software: Critical Evaluation of the Existing Approaches and a New Paradigm. Acta Universitatis Ouluensis. Oulun yliopistopaino.

[40] Siponen, M. (2003). Information Security Management Standards: Problems and Solutions. Proceedings of the 7th Pacific Asian Conference on Information Systems (PACIS 2003), ss. 1550–1561.

[41] Siponen, M. (2005). Analysis of Modern IS Security Development Approaches: Towards the Next Generation of Social and Adaptable ISS Methods. Information and Organization, Vol. 15, Issue 4, ss. 339–375.

[42] Siponen, M. & Baskerville, R. (2001). A New Paradigm For Adding Security Into IS Development Methods. In Advances in Information Security Management & Small Systems Security. Edited by Eloff, J., Labuschagne, L., von Solms, R. & Dhillon, G., Kluwer Academic Publishers, ss. 99–111.

[43] Siponen, M., Baskerville, R. & Kuivalainen, T. (2005). Integrating Security into Agile Development Methods. Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS 2005), s. 185b.

[44] Smetters, D.K. & Grinter, P.E. (2002). Moving from the Design of Usable Security Technologies to the Design of Useful Secure Applications. Proceedings of the New Security Paradigms Workshop, ss. 82–89.

[45] Soo Hoo, K., Sudbury, A.W. & Jaquith, A.R. (2001). Tangible ROI through Secure Software Engineering. Secure Business Quarterly, Vol. 1, Issue 2, ss. 1–4.

[46] Viega, J. & McGraw, G. (2004). Building Secure Software — How to Avoid Security Problems the Right Way. Addison-Wesley Professional Computing Series.

[47] Villarroel, R., Fernandez-Medina, E. & Piattini, M. (2005). Secure Information Systems Development — a Survey and Comparison. Computers and Security Vol. 24, Issue 4, ss. 308–321.

[48] Ware, M.S., Bowles, J.B. & Eastman, C.M. (2006). Using Common Criteria to Elicit Security Requirements with Use Cases. Proceedings of IEEE SoutheastCon 2006, ss. 273–278.

[49] White, E.F.R. & Dhillon, G. (2005). Synthesizing Information System Design Ideals to Overcome Developmental Duality in Securing Information Systems. Proceedings of the 38th Hawaii International Conference on System Sciences (HICSS 2005), ss. 186–195.