Shibboleth-järjestelmän riskianalyysi FTA-menetelmällä

Juha Kervinen

Helsinki 17. huhtikuuta 2003
Tietoturvallisuus nykyaikaisessa liiketoimintaympäristössä

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Sisältö

Tiivistelmä

Tässä seminaarityössä on tuotettu FTA-menetelmällä[1] suppea riskianalyysi Shibboleth-järjestelmän tietoturvasta. Lisäksi työssä arvioidaan FTA-menetelmän sopivuutta tietoturvariskien analysointiin. Analysoitava ohjelmisto, Shibboleth, on HTTP-palvelimeen asennettava on *middleware-komponentti* ja se on tarkoitettu organisaatioiden väliseen digitaalisen pääsynvalvonta- ja käyttäjätiedon välittämiseen. Tämän työn puitteissa ei analysoida syvällisesti järjestelmän teknisen tietoturvan hienouksia vaan tarkastellaan tietoturvaa yleisemmältä tasolta. Analyysin edetessä on esitetty huomioita sekä itse analyysiprosessista, että Shibboleth-järjestelmän analysoinnin aikana esiin nousseista tietoturva-asioista.

1 Johdanto

Shibbolethin avoin lähdekoodi ja protokolla ovat käyneet läpi huomattavan pitkät iteraatio- ja arviointiprosessit [1], joten tekniseltä tietoturvaltaan järjestelmää voidaan pitää ainakin kohtuullisen turvallisena. Joka tapauksessa järjestelmän teknisen tietoturvan analysointi on mahdotonta seminaarin laajuuden puitteissa. Tässä seminaarityössä on pyritty seuraaviin tavoitteisiin: Ensinnäkin tutkitaan, onko Shibboleth-järjestelmän tietoturvassa havaittavissa pintapuolisen analyysin perusteella joitakin ongelmia tai onko järjestelmän turvallisesta käytöstä löydettävissä joitakin yleisiä, ei-triviaaleja huomautuksia. Toiseksi tämän kirjoitelman aikana lukija saa jonkinlaisen käsityksen siitä, mitä FTA-analyysi on, mihin sitä kannattaa soveltaa ja mihin ei kannata. Kolmas tavoite on esittää perustellen muutamia huomioita FTA-menetelmän käytöstä tietoturvaan liittyvien asioiden yhteydessä.

Mainittuihin kolmeen tavoitteeseen pyritään varsin suoraviivaisesti. Ensimmäisen tavoitteen pohjustukseksi esitellään huomattavan paljon yksinkertaistettu, mutta totuudenmukainen [2] malli Shibboleth-ohjelmistosta ja sen käytöstä ja tarkoituksesta. Mainittakoon jo etukäteen, että varsin kummallisesti nimetylle, varmasti hieman vieraalle komponentille löytyy ihan oikeaa käyttöä. Shibbolethissa on potentiaalia monien ajankohtaisten ongelmien ratkaisijaksi [3, 4]. Tämän esittelyn pohjalta luodaan itse riskianalyysi, jossa analysoidaan viittä olennaisinta kohtaa, joissa Shibbolethin tietoturvaa vastaan voidaan kuvitella hyökättävän. Varsinainen analyysivaihe, missä yritetään vastata ensimmäiseen asetettuun tavoitteeseen,

[1] Fault Tree Analysis

jakautuu kolmeen osaan, joista jokaisesta on oma kappaleensa. Analyysi alkaa niiden rajoitteiden esittelyllä ja perustelulla jotka analyysille asetetaan. Toinen osuus on tunnistaa ne tapaukset, joita analysoidaan FTA-menetelmällä, eli ns. *päätason tapahtumat (top issues)* eli vastataan kysymykseen: *"Mitä voi mennä tutkittavassa järjestelmässä vikaan?"*. Analyysissä suoritetaan näiden tapahtumien pilkkominen osiin ja ehdotetaan näille osille ratkaisuja. Analyysin lopputuloksena on siis vastaus kysymykseen: *"Miten ongelmatilanteet voisi ratkaista?"*. Ongelmien ja niiden ratkaisujen ohella on mahdollista löytää myös säännöllisesti järjestelmän ylläpitoa rasittavia ongelmia ja yllättäviä riippuvuuksia tai sukulaisuuksia eri tietoturvaongelmien välillä.

Toiseen tavoitteeseen vastaaminen vaatii FTA-analyysin perusteiden esittelyä, ja tähän on pyritty kappaleessa 3.1. Varoitettakoon kuitenkin, että kyseessä on jälleen yksinkertaistettu, mutta pätevä [5] versio FTA-analyysistä. FTA-analyysistä käytetään myös suomennettua, ilmeisen epävirallista termiä *vikapuuanalyysi*. Viimeiseen tavoitteeseen eli FTA-analyysin sopivuuteen tietoturvariskien analysointivälineenä tullaan lopuksi, tämän seminaarityön loppupäätelmissä.

2 Shibboleth-järjestelmän esittely

Internet2 määrittelee Shibbolethin seuraavasti [1] (suomennettu, referoitu): *Shibboleth on aloite sellaisen ohjelmiston tuottamiseksi, jonka avulla useat toimijat, kuten korkeakoulut, valtion virastot ja vastaavat organisaatiot voivat siirtää turvallisesti tietoa käyttäjistään tai käyttäjäryhmistään. Tiedonvaihdon tarkoituksena on tyypillisesti päätellä, onko esimerkiksi www-selaimen käyttäjä oikeutettu pääsemään johonkin organisaation tietoresurssiin. Päätös tästä tehdään sen perusteella, onko henkilö jonkin ryhmän jäsen. Järjestelmä pyrkii mahdollisimmaan pitkälle suojelemaan yksityisyyttä ja käyttäjänsä identiteettiä eikä näinollen jaa ylimääräistä tietoa käyttäjästä. Käyttäjä voi itse päättää, mitä tietoja paljastaa itsestään. Shibboleth on avoin ratkaisu – sen arkkitehtuuri, toimintatapa ja lähdekoodi ovat kaikkien nähtävissä ja kommentoitavissa.*

Shibbolethin arkkitehtuurista esitellään tässä sen ymmärtämisen kannalta erittäin minimaalinen osa, vain olennaiset komponentit. Shibbolethin lopullista arkkitehtuuria tai toteutusta ei ole vielä toistaiseksi edes kiinnitetty [2], joten sen tarkempi esittely ei liene järkevääkään.

Yksi Shibbolethin tärkeimmistä tavoitteista on helpottaa järjestelmäylläpidon työtä käyttäjähallintaa vaativien sovellusten ylläpidossa. Sekä vanhanaikaisemmissa nimi/salasana-tietokannoissa, että julkisen avaimen infrastukstuurin varassa toimivissa käyttäjätietojärjestelmissä on perustavaa

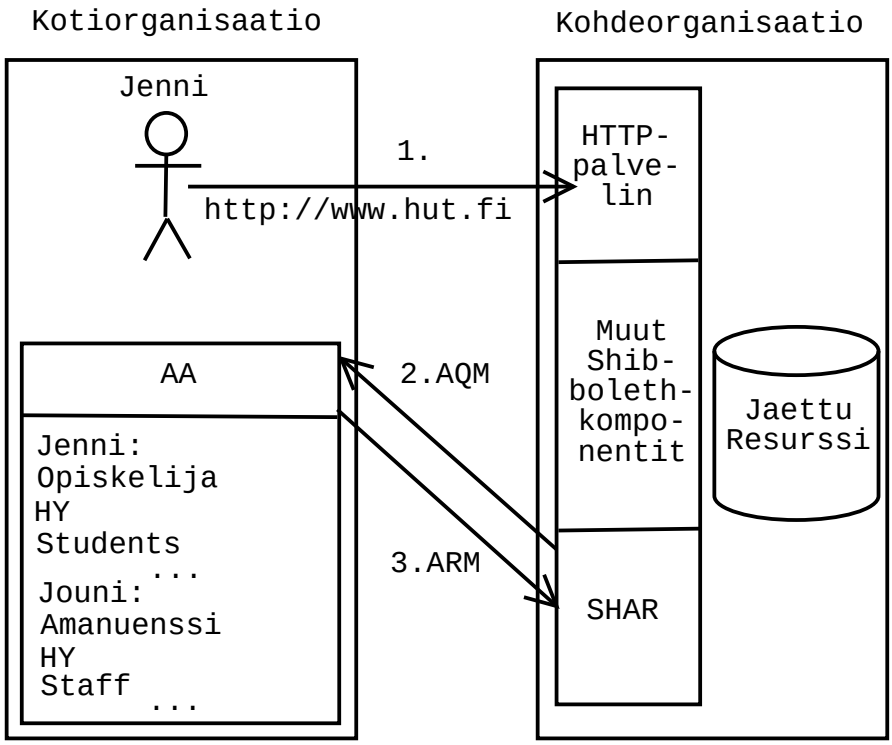
laatua oleva ongelma. Niissä käyttäjätiedon ylläpito rasittaa järjestelmäylläpitoa. Jokaisessa paikassa joudutaan ylläpitämään käyttäjätietorekisteriä, jonka ajantasaisuudesta on huolehdittava tai järjestelmä ei toimi tai pahimmassa tapauksessa sen turvallisuus murtuu. Käyttäjätiedon monimuotoisuus on lisäksi kaiken aikaa kasvussa, joten kasvavan tietomassan käsittely ei ainakaan helpotu. Poistuvien käyttäjien poistaminen rekistereistä ja uusien sinne lisääminen on suuri urakka jo yhden organisaation ylläpidolle, mutta jos tämä sama työ jouduttaisiin tekemään vastaavassa mittakaavassa tämän organisaation jokaisessa sidosorganisaatiossa, kokonaistyömäärä käyttäjähallinnan hoitamisessa kasvaisi eksponentiaalisesti. Lisäksi käyttäjätiedon säilyttäminen useassa paikassa ei välttämättä ole laillista tai ainakaan käyttäjien toiveiden mukaista [6].

Shibboleth pyrkii eroon näistä ongelmista pitämällä käyttäjätiedon pelkästään käyttäjän omassa organisaatiossa eli *kotiorganisaatiossa*. Shibboleth siirtää tietoa käyttäjäattribuutteina, tietopaketteina jotka sisältävät tietoa käyttäjästä. Attribuutti voi sisältää esimerkiksi käyttäjätunnuksen, mutta ei välttämättä. Monissa pääsynvalvontapäätöksissä riittää tietää kuuluuko käyttäjä esimerkiksi ryhmään[2] *students@cs.helsinki.fi* tai *member of university community@hut.fi*. Ei ole siis tarpeen siirtää käyttäjästä mitään muuta tietoa kuin tieto ryhmän jäsenyydestä.

Kohtuullisen paljon yksinkertaistettuna Shibboleth-järjestelmä koostuu käyttäjän kotiorganisaatiosta, useista kohdeorganisaatioista ja niitä yhdistävästä tietoverkosta (internet). Jokaisella kotiorganisaatiolla on attribuuttipalvelin *(engl. Attribute Authority, AA)*, jolta voidaan kysellä tietoja käyttäjästä. Kotiorganisaation AA toteuttaa jonkin attribuutinjulkaisupolitiikan, jonka käyttäjä voi periaatteessa määrätä itse, vaikkapa jonkin www-lomakkeen välityksellä. Tässä politiikassa määritellään mitä tietoja käyttäjästä mikäkin kohdeorganisaatio voi saada.

Kuva 1 esittää Shibbolethin toimintaperiaatteen yksinkertaistettuna kolmeen viestinvaihtoon. Ensimmäisessä vaiheessa kotiorganisaation käyttäjä pyytää kohdeorganisaation HTTP-palvelimelta tietoa liittyen johonkin jaettuun resurssiin, mikä voi olla vaikkapa käyttäjän tutkimusryhmän tietopankki. Shibboleth-yhteensopiva palvelin päättelee, että sillä ei ole tarpeeksi tietoa käyttäjästä, jotta se tietäisi mitä käyttäjän on mahdollista nähdä jaetusta resurssista. Latauspyynnön perusteella SHAR, eli *Shibboleth Attribute Requester* pyytää AA:lta käyttäjän attribuutteja AQM-viestillä *(Attribute Query Message)*. AA palauttaa attribuutit, jotka on määritelty (käyttäjän toimesta tai muuten) palautettavaksi vastauksena saapuneenkaltaisi-

[2]Tässä käytetty notaatio, attribuuttitunnus@organisaatio.tunniste näyttää sähköpostiosoitteelta, mutta se ei ole sitä.



Kuva 1: Shibbolethin toimintaperiaate yksinkertaistettuna kolmeen viestinvaihtoon (suomennettu ja mukautettu lähteestä [2]).

in kyselyihin. AA lähettää vastauksen, ARM-viestinä *(Attribute Response Message)* takaisin SHAR:lle. Tämän jälkeen tapahtuvaa viestienvaihtoa ei ole esitetty kuvassa 1. SHAR joka tapauksessa lähettää saamansa attribuutit organisaationsa resurssinhallintajärjestelmälle, joka yksinkertaisimmillaan on staattisia HTML-sivuja tarjoilevalla http-palvelimella oleva ohjelmakomponentti. SHAR tekee saamiensa attribuuttien perusteella päätöksen pääsystä jaettuun resurssiin.

Todellisuudessa järjestelmä voi olla hyvinkin paljon monimutkaisempi. Resurssienhallintajärjestelmä voi olla kokonaan toisella palvelimella tai hyvinkin kaukana alkuperäisen pyynnön vastaanottajasta. Monimutkaisempi tilanne voisi olla vaikkapa tutkimusrahoituksen kuittaus rahoitusorganisaatiolta tai päivityspyyntö opintorekisteriin.

Tämän seminaarityön laajuuteen sopiva tarkastelulaajuus on yllä esitetty suhteellisen yksinkertainen tilanne, missä organisaation HTTP-palvelin vastaa käyttäjien kotiorganisaatioista tuleviin pyyntöihin ja kyselee näiltä organisaatioilta lisäattribuutteja käyttäjistä.

3 Riskianalyysi

Tässä kappaleessa käydään ensin läpi FTA-menetelmän perusperiaatteet, sitten suoritetaan analyysin rajaus eli kuvataan tarkemmin tilanne mitä analysoidaan. Seuraava vaihe on analysoitavien tapahtumien eli päätason tapahtumien tunnistaminen ja lopuksi itse analyysi. Yleisenä huomautuksena kappaleesta todettakoon, että kaikki tilanteet kattavan riskianalyysin tekeminen mistään monimutkaisemmasta järjestelmäkokonaisuudesta on todella aikaavievä tehtävä. Tästä syystä tämän seminaarityön FTA-analyysi keskittyy vain pieneen osaan kaikista Shibbolethin käyttötavoista, joten tämän työn tuloksia ei voida yleistää muihin, ainakaan monimutkaisempiin käyttötapauksiin, verkkokonfiguraatioihin tai organisaatioihin. Toinen huomioitava seikka on, että seminaarin aihepiirin (tietoturva) takia tässä analyysissä keskitytään nimenomaan tietoturvariskeihin. Shibbolethin toimivuuden varmistaminen oikeassa tuotantokäytössä on huomattavasti tätä työtä monimutkaisempia asia ja tässä tunnistettujen riskien ja niiden ratkaisujen toimivuus nojaa oletuksiin, joita ei oikeassa ympäristössä voida koskaan välttämättä varmistaa, kuten esimerkiksi, että organisaation palomuuria ei voida kiertää jotain ennakoimatonta reittiä tai pakettisuodatussäännöt ovat varmasti oikeita [7, s.203]. Erityisesti organisaation sisältä tuleviin uhkiin varautuminen on Shibbolethin kaltaisessa järjestelmässä erityisen vaikea tehtävä [7, s.20].

3.1 FTA-menetelmästä

FTA-analyysissä on kaksi vaihetta. Hyvin suoraviivaisesti, nämä vaiheet ovat vikatilanteiden tunnistaminen (luku 3.3) ja niiden analysointi (luku 3.4. Näiden kahden vaiheen lisäksi tässä työssä on analyysin tietynlaisena esivalmisteluna toteutettu analyysin rajaus (luku 3.2), jonka tarkoitus on toisaalta esitystekninen eli karsitaan pois epäolennaisia rönsyjä selkeyden takaamiseksi ja toisaalta tekninen: Rajauksen tarkoitus on myös määritellä selkeästi, mitä Shibbolethin osia ja käyttötapauksia analyysissä käsitellään ja mitä mitkä ovat ne oletukset, joita järjestelmästä oletetaan.

FTA-analyysin varsinaisessa analyysiosuudessa tuotetaan järjestelmän tarkisteltavista tapahtumista vikapuu (*Fault Tree*). Vikapuussa käytetty notaatio on esitelty kahdessa oheisessa kuvassa. Kuvassa 2 on kaksi vikapuuta, joista ylimmäisessä on esitetty AND-portin toiminta ja alemmassa OR-portin toiminta sekä muutama tapahtumasymboli. Vikapuuta luetaan ylhäältä alaspäin. Ylemmässä kaaviossa siis AND-portti osittaa tapahtuman “Kaikki palvelimet sammuvat” tapahtumiksi (sekä) “WWW-palvelin sammuu” ja (että) “Tietokantapalvelin sammuu”. Alemmassa kuvassa OR-

portti esittää mahdollisia syitä tapahtumalle, että www-sivusto osoitteessa `http://www.cs.helsinki.fi` ei lataudu. Mahdollisia syitä on neljä ja niiden erilaisuutta kuvataan erilaisilla tapahtumasymboleilla. Vikapuiden kaaviomerkinnät on esitetty kuvassa 3.

- *Verkko-ongelma* on kehittymätön tapahtuma, koska sitä analysoitaessa on tultu johtopäätökseen, että analyysin jatkamiselle ei ole riittävästi dataa tai tämä tapahtuma menee analyysin rajauksen ulkopuolelle tai on muuten epäolennainen. Esimerkiksi verkko-ongelmat saattavat johtua internet-palveluntarjoajan runkoverkko-ongelmista, joiden analysointi ei ole oleellista analyysimme kannalta.

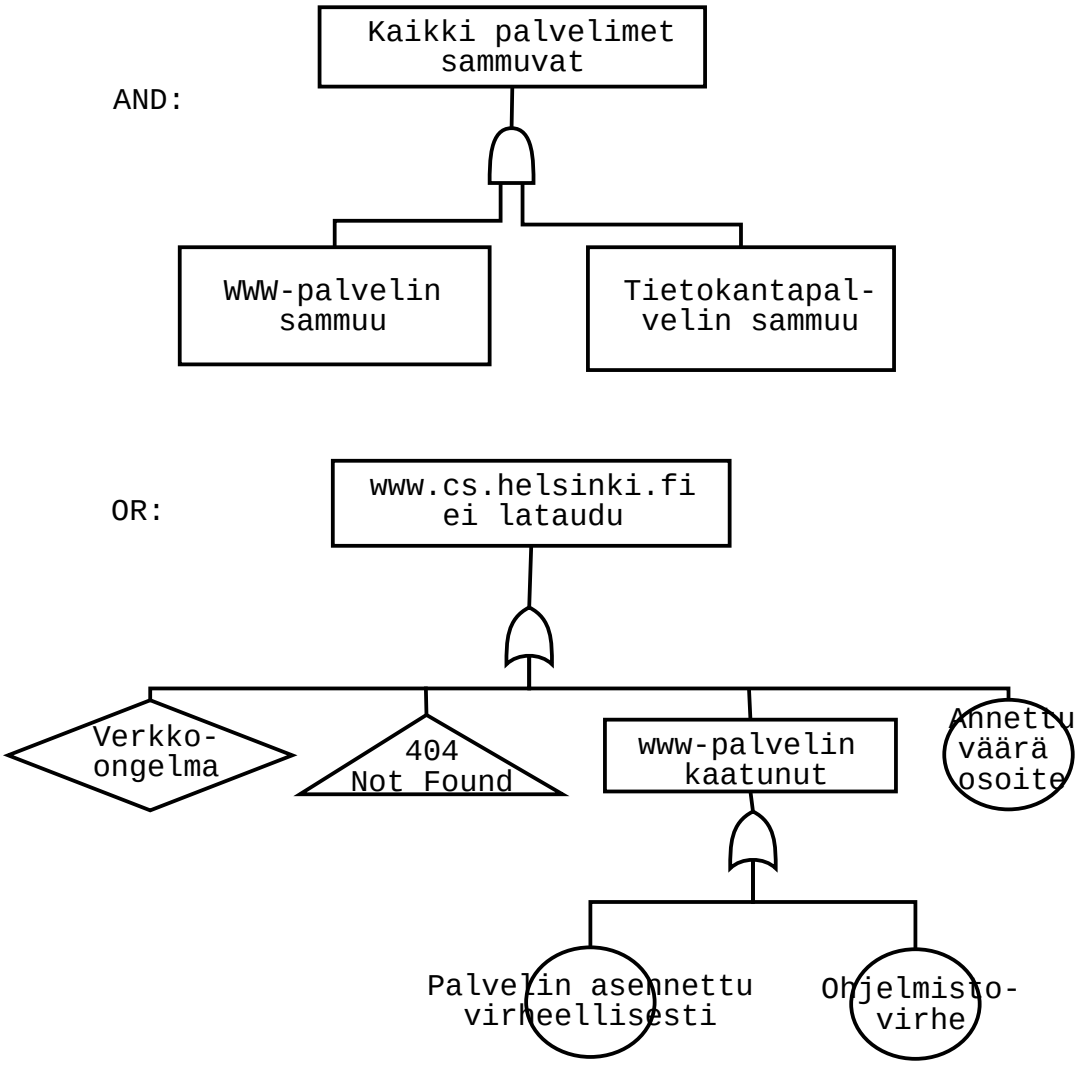
- *404 Not Found* eli palvelimen vastaus, että sivua ei löydy on merkitty vikapuussa kolmiosymbolina, mikä tarkoittaa sitä, että tämän tapauksen jatkoanalyysi on esitetty toisessa vikapuussa. Syytä siirtoon ei tarvitse antaa, mutta tyypillisesti syy voi olla esitystekninen (selkeys) tai sitten tapahtuman alitapahtumat ovat niin erityyppisiä kuin vikapuun muut tapahtumat, että siirto selkeyttää puun logiikkaa.

- *WWW-palvelin kaatunut* on välillinen tapahtuma, jonka analysointia jatketaa vikapuussa tapahtuman alapuolelle kaaviossa.

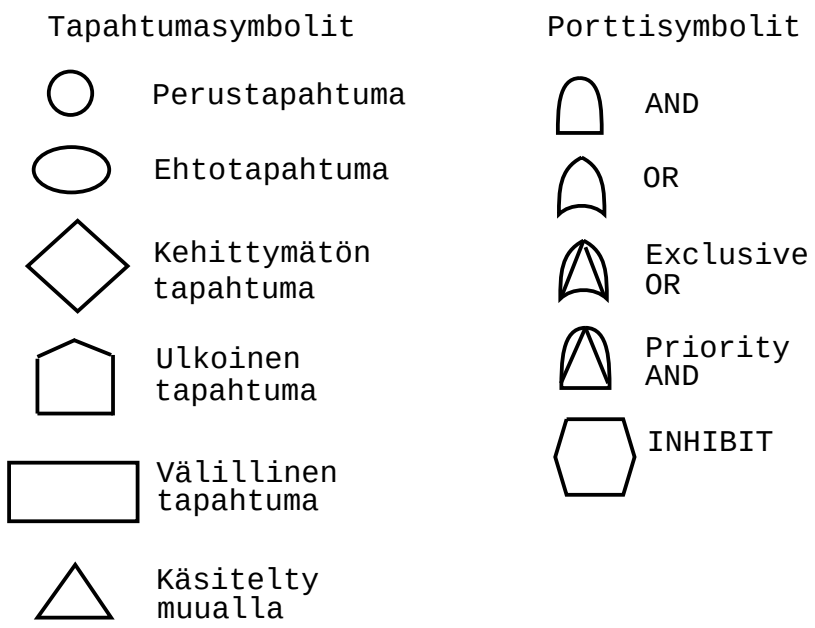
- Tapahtuma *Annettu väärä osoite* on merkitty perustapahtumaksi, eli sen jatkokehitys ei enää ole tarpeellista. Toisinsanoen tästä tapahtumasta on selvitetty kaikki tarvittava. Tällaisille tapahtumille voidaan kirjoittaa joka suoraan vikapuuhun tai muualle analyysin dokumentaatioon ratkaisurehdotuksia[3]. Kuvan tapauksessa ratkaisuehdotus voisi olla vaikkapa “Kirjoitetaan osoite uudelleen oikein”.

AND ja OR ovat varmasti käytetyimmät porttisymbolit, joten niiden toiminta on selvitettävä hieman tarkemmin, sillä niiden oikeaan käyttöön FTA-analyysissä sisältyy huomio, joka ei välttämättä ole selvää, vaikka loogiset operaatiot varmasti lukijalle ovatkin tuttuja. Syy-seuraussuhde ei nimittäin välity OR-portin läpi. Jokainen portille syötteenä toimiva vikatila on toisinsanoen yksittäistapaus portin tulostilasta. AND taas välittää tiedon tapahtumien kausaalisesta suhteesta. AND-portin syötteet edustavat yhteisesti tilaa, joka on portin tuloksena [8, s.39-41]. Syy-seuraussuhteen seuraaminen on tärkeää, sillä sen avulla vikapuista voidaan etsiä logiikkavirheitä.

[3]Tässä dokumentissa ratkaisuehdotukset esitetään tekstissä, silla tietoturva-asioden ratkaisuehdotuksia harvoin saa mahtumaan yhteen lauseeseen.



Kuva 2: Kaksi yksinkertaista vikapuuta (mukaillen lähdettä [5])



Kuva 3: FTA-analyysissä käytetyt symbolit (suomennettu ja mukautettu lähteestä [8])

Kuvassa 3 on siis esitetty myös muut vikapuuanalyysissä käytetyt symbolit[4]. Kuvan 2 esimerkeissä esiintymättömät tapahtumasymbolit ovat:

- *Ehtotapahtuma* eli tapahtuma, joka kuvaa tiettyä ehtoa, jolla jokin asia tapahtuu. Ehtotapahtuma liitetään siihen porttiin, jonka tapahtumisen ehtotapahtuma estää tai sallii. Ehtotapahtuma tarjoaa näin lisäkontrollin, jolla voidaan osoittaa se, miten tapahtumat riippuvat tietyistä muuttujista.
- *Ulkoinen tapahtuma* on tapahtuma, jonka odotetaankin tapahtuvan ja joka ei varsinaisesti ole vikatilanne. Tällaisia tapahtumia ovat esimerkiksi tilasiirtymät järjestelmässä

OR- ja AND-porttisymboleista on myös erityiset versiot, *Exclusive OR* ja *Priority AND*. Exclusive OR on tuttu XOR-operaatio, joka tapahtuu jos vain jokin (tasan yksi) syötetila on tosi. Priorisoidun AND-operaation tulostila tapahtuu vain, jos kaikki syötetilat tapahtuvat jossain tietyssä järjestyksessä. Järjestyksen määrää porttiin liitettävä ehtotapahtuma. INHIBIT on

[4]Esitystapa ja symbolien määrä vaihtelee hieman eri lähteiden välillä. Tässä esitetyt ovat peräisin teoksista [5] ja [8].

myös ehdollistava portti, jonka tulostapahtuma tapahtuu vain jos (yksi ainoa) syötetapahtuma tapahtuu porttiin liitetyn ehtotapahtuman osoittamassa tilanteessa.

3.2 Analyysin rajaus

Tässä työssä on analysoitu tilannetta, missä Shibboleth-järjestelmää katsotaan ikään kuin käyttäjän kotiorganisaation ylläpitohenkilökunnan silmin, esimerkiksi jonkin korkeakoulun laitoksen ATK-ylläpito -osastona, joka ylläpitää paikallista tietojärjestelmää. Analyysissä ollaan siis kiinnostuneita selvittämään mitä tietoturvariskitilanteita sisältyy kotiorganisaation ja sen AA:n teknisen ylläpitoon. Riskitilanteita ovat esimerkiksi sellaiset tilanteet, joissa

- kotiorganisaatiosta tai sen käyttäjästä lähtee sellaisia tietoja verkkoon, joiden ei haluta paljastuvan maailmalle tai tietoja jotka eivät pidä paikkaansa
- ulkopuolinen tahon on mahdollista kuunnella kotiorganisaation ja kohdeorganisaation välistä liikennettä
- ulkopuolinen taho saa selville järjestelmän tietoturvallisuutta uhkaavia asiota, kuten salasanoja
- järjestelmän tietoturva tai normaali toiminta muuten vaarantuu

Kuten jo edellä on todettukkin, järjestelmän sisäisten tahojen suunnalta tulevat uhkat joudutaan rajaamaan analyysistä pois niiden ennakoimattomuuden takia. Lisäksi analyysissä keskitytään vain teknisin keinoin toteutettavissa oleviin hyökkäyksiin. Hyökkäykset, joiden olennaisena osana on ns. *Human Engineering* eli ihmisten huijaaminen tietoturvan kompromisoimiseksi, jätetään niinikään huomioimatta.

3.3 Päätason tapahtumat

Vikapuuanalyysin tarkoitus eli virhetilanteiden analysoiminen implikoi analyysiin mukaan otettavista päätason tapahtumista muutamia asioita. Ensinnäkin on selvää, että ollakseen reaalimaailman virhetilanne, tapahtuman pitää olla mahdollista tapahtua[5]. Kyseessä ei ole itsestäänselvyys,

[5]Tällöin tapahtuman todennäköisyyden tulisi olla suurempi kuin nolla. Vikapuuanalyysissä nollan todennäköisyydekseen saavat tapahtumat ovat mahdottomia, sillä vikapuussa ne eivät silloin esiintyisi

vaikka se siltä kuulostaakin. Nimittäin tästä vaatimuksesta voidaan suoraan sanoa, että vikapuuanalyysi ei sovellu sen selvittämiseen, *voiko asia X tapahtua*. Tieto tästä on tietenkin täydellisen tietoturva-analyysin kannalta olennaista. Vikapuuanalyysi vastaa näinollen vain kysymykseen, *jos X tapahtuu, niin mitä siitä seuraa?* Näin ollen sellaisten tapahtumien kohdalla, joiden oletetaan tapahtuvan ja analyysi toteutetaan oletuksen perusteella, ei ole järjevää puhua vikapuun jonkin oksan todennäköisyydestä, sillä kaikki perustuu oletuksen todennäköisyyteen, mitä ei tiedetä. Toisaalta ollakseen analysoimisen arvoinen, tapahtuman täytyy jotenkin järkevästi jakaantua osakokonaisuuksiin. Yleisesti tietoturva-asioden yhteydessä todennäköisyyksistä puhuminen tuntuu vaikealta. Esimerkiksi oikeaan osunut arvaus siitä, milloin löydetään seuraava tietoturva-aukko jostain palvelinohjelmistosta lienee puhtaasti onnesta kiinni. Muutenkin nykyinen suuntaus näyttää olevan, että jos on tiedossa jokin teoreettinen murtomenetelmä, se ei ole teoreettinen kovin pitkään ja siksi aukko kannattaa paikata etukäteen. Näinollen tämän työn vikapuissa ei ole liitetty oksiin mitään todennäköisyyksiä.

Lisäksi on varmaankin syytä korostaa, että alla esitettyä päätason tapahtumien listaa ei varsinaisesti ole koottu tai sen kattavuutta vahvistettu minkään tieteellisesti hyväksytyn metodin perusteella. Todellinen tilanne sisältäisi suuren joukon (kehittymättömiä) puita joissa vain todetaan vikatilanne, mutta ei jatketa analyysiä, sillä puu kasvaisi liian suureksi triviaalien tilanteiden suuren määrän vuoksi. Tässä laajuudessaan päätason tapahtumat eivät varmasti kata koko totuutta tilanteesta. Alla luetellut tapahtumat analysoidaan seuraavassa kappaleessa.

(1) Hyökkääjä pääsee murtautumaan paikalliselle työasemalle. Tässä tapauksessa hyökkääjä saa tavalla tai toisella murrettua organisaation verkon ja työaseman turvatoimet ja saa pääsyn jollekkin organisaation työasemalle.

(2) Hyökkääjä pääsee murtautumaan Shibbolethin AA-osajärjestelmän palvelimelle. Palvelinmurto poikkeaa tavallisesta työasemamurrosta huomattavasti, kuten tulemme huomaamaan. Käytännössä peli on usein pelattu murtautujan päästyä palvelinjärjestelmään pääkäyttäjänä[6].

(3) Hyökkääjä saa käsiinsä käyttäjän selaimen ja kohdeorganisaation välisen siirtoyhteyden dataa. Tässä tapauksessa oletuksena on, että käyt-

[6]Käyttäjänä *root*

täjän selainohjelmisto on ajan tasalla ja käyttäjä käyttää nykyaikasta implementaatiota SSL-protokollasta ja käyttää mahdollisista SSL-protokollan työkalupakin[9, s.298] vaihtoehdoista turvallista vaihtoehtoa. Näinollen, hyökkääjälle jää suhteellisen paljon keinoja joista ei voi sanoa mitään varmaa. Tapaus on myös hyvä esimerkki siitä, miten FTA-analyysi ei välttämättä sovellu sellaisten tilanteiden analysoimiseen, joiden jakaantumisesta osiin ei voida sanoa mitään tarkkaa.

(4) Hyökkääjä saa haltuunsa AQM-viestin ja toistaa sen kotiorganisaation järjestelmälle. AQM-viestin saaminen siirtokanavaa kuuntelemalla ei ole mahdollista (käytössä on salattu kanava) joten tämän tilanteen syntymiseksi hyökkääjän on murtauduttava kohdeorganisaation tietojärjestelmään. Oletetaan että tämä on mahdollista.

(5) Hyökkääjä onnistuu esiintymään kotiorganisaation käyttäjänä. Tässä tapauksessa ongelma on se, että kotiorganisaation näkökulmasta vihollinen on oman organisaation käyttäjä ja uhkana on siis se, että kohdeorganisaatio luulee käyttäjän tulevan "oikeasta paikasta" kun näin ei ole.

3.4 Päätason tapahtumien analyysi

3.4.1 Tapaus 1. Työasemamurto

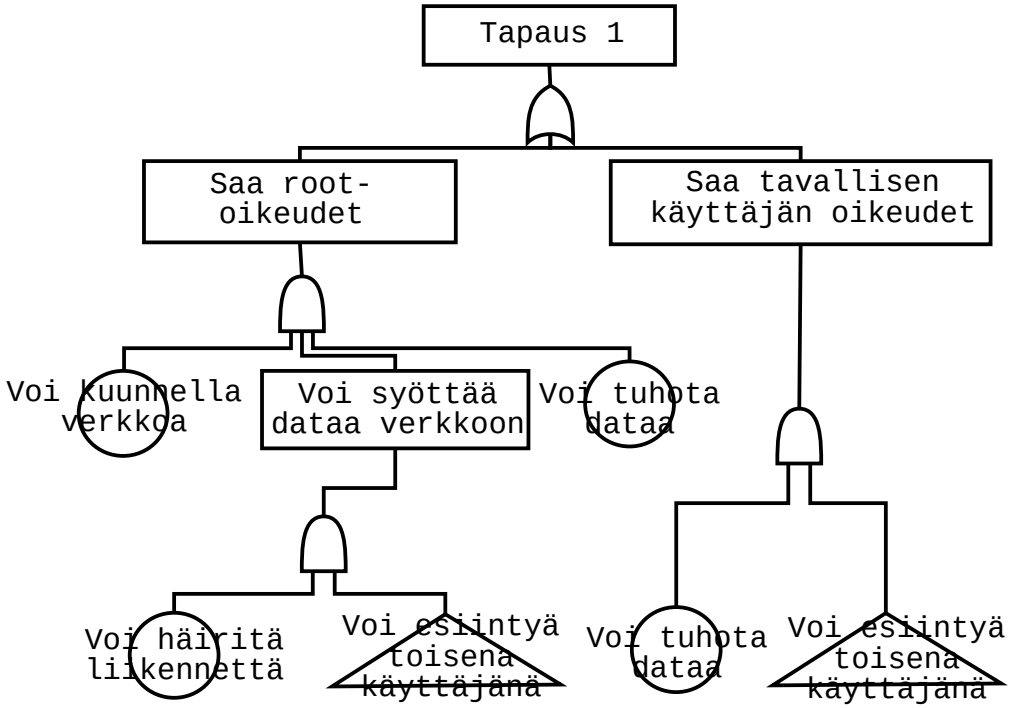
Työasemamurrossa (Kuva 4) hyökkääjä murtautuu jollekkin kotiorganisaation työasemakoneelle ja saa näin alustan jatkohyökkäykselle Shibboleth-järjestelmää vastaan. Oletuksena työasemassa on jokin Unixin kaltainen käyttöjärjestelmä[7]. Pääkäyttäjätilassa hyökkääjä saa täydet oikeudet kuunnella verkkoliikennettä ja lähettää omaa dataansa verkkoon. Hyökkäysmahdollisuudet ovat näin täysin eri luokkaa kuin tavallisena käyttäjänä.

Tapaus 1 siis korostaa erityisesti sitä faktaa, että organisaation on erittäin tärkeää suojella myös työasemakoneitaan root-oikeudet mahdollistavia hyökkäyksiä vastaan. Pääkäyttäjän mahdollisuus verkkoliikenteen kuuntelemiseen avaa luonnollisesti paljon mahdollisuuksia jatkaa hyökkäystä.

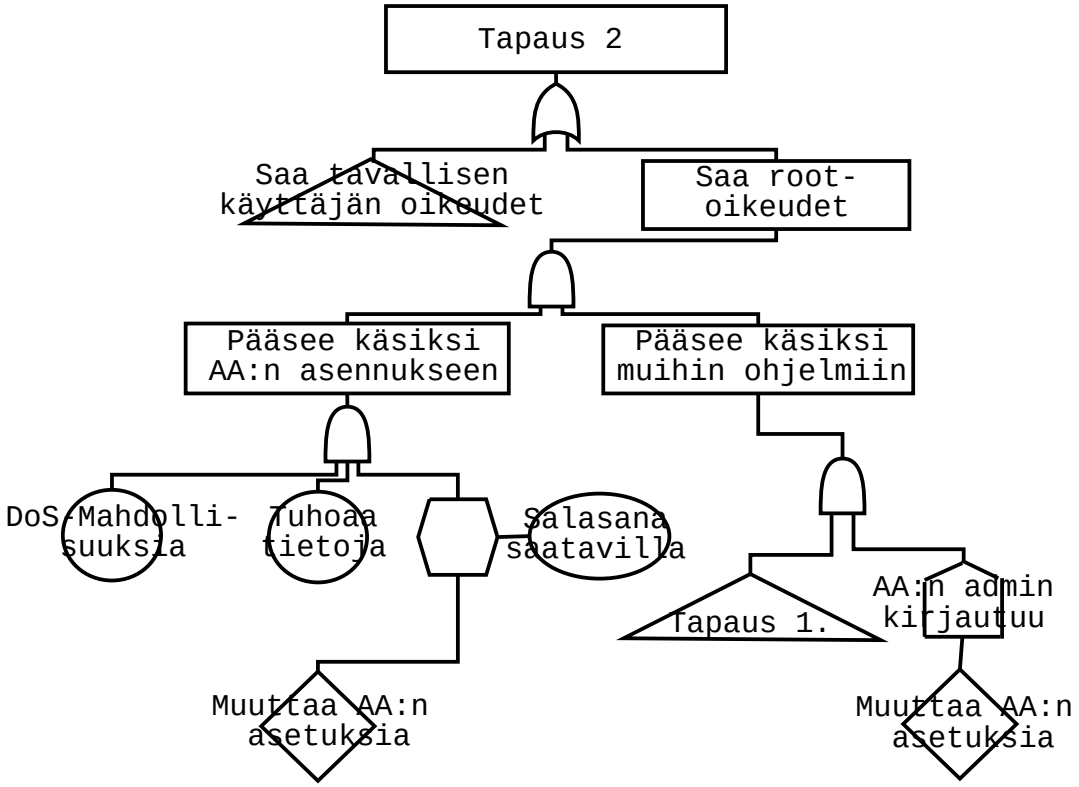
3.4.2 Tapaus 2. Palvelinmurto

Tapaus 2 (Kuva 5) eroaa ansimmäisestä tapauksesta niin, että hyökkääjä saa käyttöoikeudet kotiorganisaation AA-järjestelmää ajavalle palvelinkoneelle.

[7]Jos näin ei ole, niin oletamme, että käyttäjä saa suoraan pääkäyttäjän oikeudet.



Kuva 4: Vikapuu työasemamurrosta



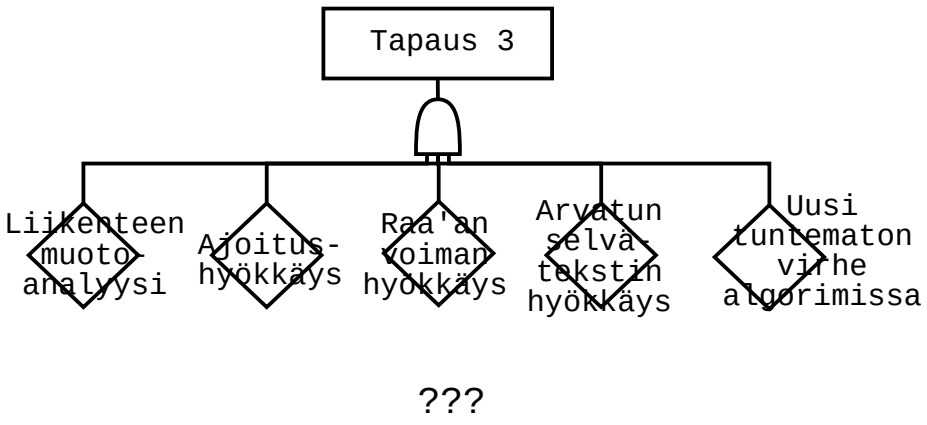
Kuva 5: Vikapuu palvelinmurrosta

Jos hyökkääjä saa vain tavallisen käyttäjän oikeudet, tilanne on olennaisesti sama kuin tapauksessa 1. Taas, jos hyökkääjä saa pääkäyttäjän tunnukset, tilanne on paljon huonompi.

Käytännössä palvelinkoneen pääkäyttätunnusten saaminen johtaa aina Shibboleth-järjestelmän AA-palvelimen murtoon, sillä vaikka AA:n salasana olisi palvelinkoneella salattuna (jollain salaisen avaimen menetelmällä) niin hyökkääjän ei tarvitse kuin asentaa näppäimistönkuunteluohjelma ja hän saa näin salasanan tietoonsa. Tapaus osoittaa, että palvelinkoneen root-oikeudet saava hyökkääjä pystyy mihin vain.

3.4.3 Tapaus 3. Man-in-the-middle

Tapauksessa 3 (Kuva 6) hyökkääjä on kotiorganisaation ja kohdeorganisaation välissä kuuntelemassa viestienvaihtoa. Koska Shibbolethin yhteydessä käytetään SSL-salausta [2], hyökkääjän mahdollisuudet ovat suhteellisen vähäiset, mikäli hän ei pysty päihittämään SSL-salauksen tieto-



Kuva 6: Vikapuu tilanteesta Man-in-the-middle

turvaa.

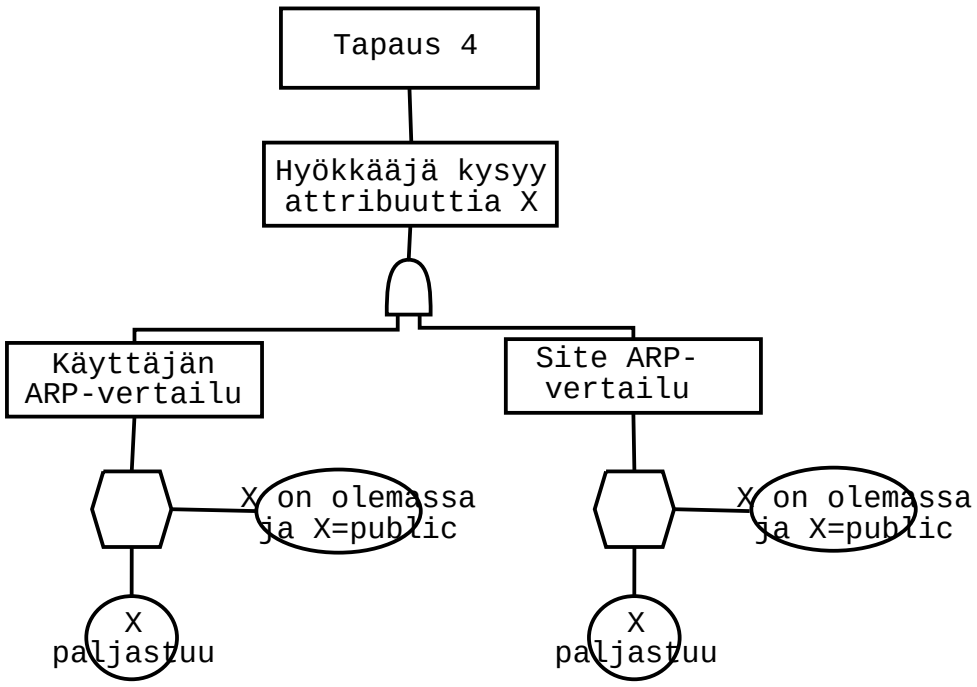
Tapaus osoittaa, miten vaikea FTA-analyysilla on hakea vastauksia tilanteisiin, jotka eivät selkeästi jakaudu joihinkin tiettyihin toimintoihin. Jokainen puun oksa degeneroituu yksitasoiseksi alipuuksi, missä voidaan vain todeta, että jokin on mahdollista.

3.4.4 Tapaus 4. Kohdeorganisaation murto

Tapaus on esitetty kuvassa 7. Shibbolethin kotiorganisaation kannalta kohdeorganisaatioon murtautuminen vaikuttaa niin, että enää ei voida luottaa siihen, että kohdeorganisaatiosta tulevat viestit ovat laillisten käyttäjien lähettämiä. Näinollen kotiorganisaation tietoturvan viimeinen suoja on attribuuttien julkaisupolitiikka (*ARP, Attribute Release Policy*)[8], joka toimii niin, että saapuneen AQM-viestin attribuuttikyselyä verrataan kotiorganisaatiossa määriteltyihin julkaisupolitiikkoihin ja näiden perusteella tehdään julkaisupäätös. Jos jokin julkaisupolitiikka sallii julkaisun, tieto julkaistaan. Julkaisupolitiikkoja on tyypillisesti kaksi. Kotiorganisaation määrittelemä ja käyttäjän itsensä määrittelemä.

Tapaus osoittaa, että jos kohdeorganisaatioon ei voi luottaa, tilanne on varsin huono. Johtopäätös on, että ARP kannattaa aina määritellä mahdollisimman tiukaksi. Lisäksi käyttäjän ei kannata välttämättä syöttää sellaisia arvoja joiden näkee olevan niin kriittisiä salassa pysymisen kannalta, että ei uskalla luottaa kotiorganisaation julkaisupolitiikkaan. Tosin käytännössä huoli tästä toivottavasti on turha. Yksikään vastuuntuntoinen ko-

[8]Voidaan suomentaan myös julkistuspolitiikaksi.



Kuva 7: Vikapuu kohdeorganisaation murrosta

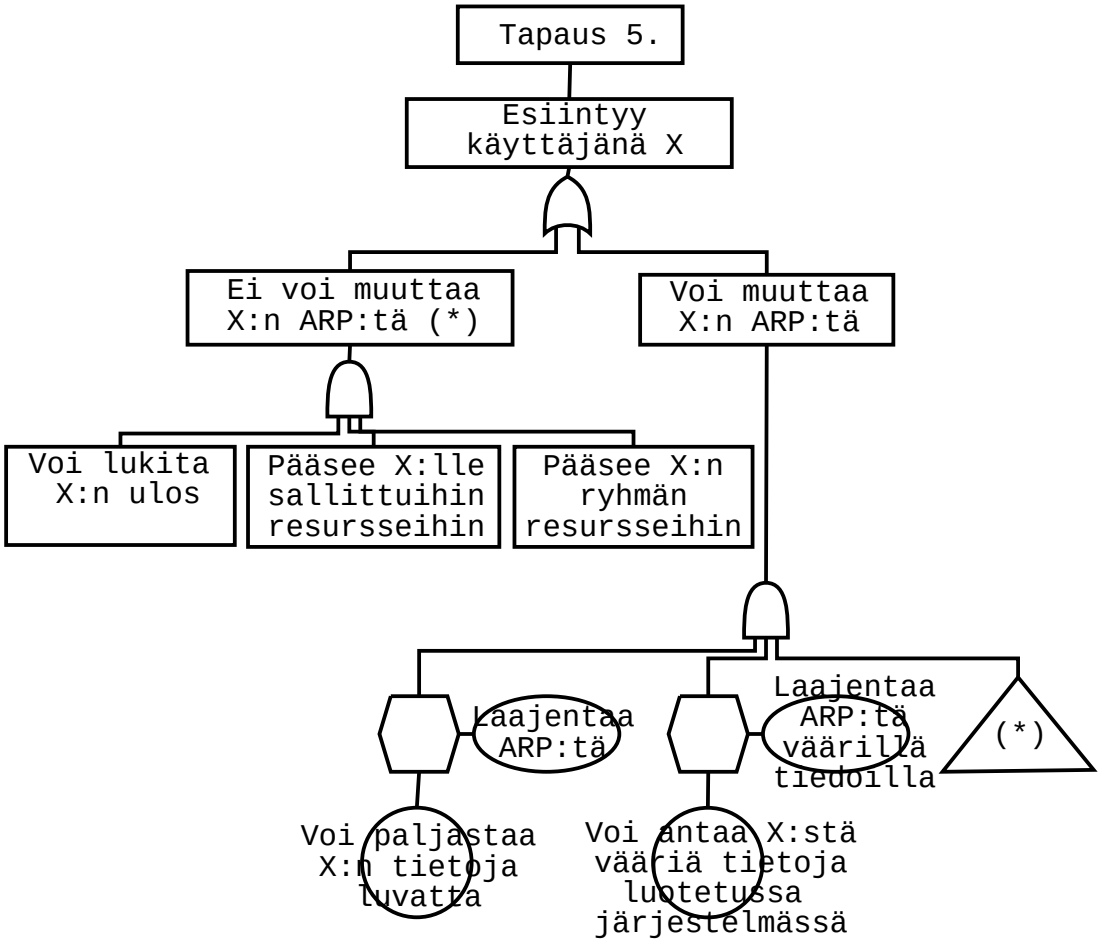
tiorganisaatio ei varmasti määrittele julkiseksi käyttäjiensä attribuutteja, joilla on jotain arvoa.

3.4.5 Tapaus 5. Paikallisena käyttäjänä esiintyminen

Tapauksien 1 ja 2 seurauksena yksi mahdollinen hyökkääjän strategia voi olla esiintyä kotiorganisaation käyttäjänä ja näin edetä murtotavoitteissaan.

Viimeisessä tapauksessa (Kuva 8) tilanne jakautuu kahtia sen mukaan, onko käyttäjänä X esiintyvällä hyökkääjällä mahdollisuutta vaihtaa käyttäjän ARP:tä. Jos ei ole, käyttäjän mahdollisuudet ovat rajatummat. Toisaalta seuraukset voivat olla vakavatkin, minkä osoittaa se, että esiintymällä käyttäjänä X hyökkääjä pääsee myös niihin resursseihin käsiksi mihin X, eli myös sen ryhmän resursseihin mihin X kuuluu. Tämä osoittaa sen, että ryhmän sisällä kannattaa noudattaa jonkinlaista “need to know”-mallia tiedon jakamisen suhteen – mikä ei ole pakko olla sallittua on kiellettyä.

Siinä tapauksessa, että hyökkääjä voi laajentaa käyttäjän ARP:tä paljastamaan enemmän tietoa käyttäjästä ulkomaailmaan, hänelle tarjoutuu mahdollisuuksia aiheuttaa lisäharmia X:lle ja X:n organisaatiolle. Nimit-



Kuva 8: Vikapuu hyökkääjän esiintymisestä laillisena käyttäjänä X

täin jos hyökkääjä laajentaa X:n ARP:tä, kohdeorganisaatioiden kyselyissä lähtee verkkoon käyttäjästä enemmän tietoa kuin alunperin käyttäjä olisi ikinä ollut valmis hyväksymään. Toisaalta vielä vakavampi tilanne muodostuu, jos on mahdollista laajentaa ARP:tä ja syöttää sinne tietoa, joka on valheellista. Koska Shibbolethin kaltaisessa järjestelmässä paljon perustuu siihen, että käyttäjät ja järjestelmät voivat luottaa toisiinsa, voivat seuraukset olla vakavat. Lisäksi organisaation maine kärsii, jos sen kautta leviää verkkoon valheellista tietoa. Käyttäjähän voi itse olla täysin syytön tilanteeseen, jos hyökkääjä on murtautunut organisaation työasemalle tai palvelimelle ja saanut pääkäyttäjän tunnukset.

4 Loppupäätelmät

Analyysin pohjalta voidaan sanoa, että Shibboleth on vain niin turvallinen kuin ympäristönsä. Tämä on tietenkin tulos, mikä pätee kaikkiin tietojärjestelmiin. Toisaalta käyttäjä ja kotiorganisaatio voivat ehkäistä väärinkäytöksiä määrittelemällä järjestelmänsä ja attribuutien julkaisupolitiikkansa mahdollisimman tiukiksi. Tämä on myös tulos, jonka pitäisi olla osaavalle ylläpidolle itsestäänselvyys[9], joskin kiireessä ja helppokäyttöisyyden tavoittelussa idea joskus unohtuu.

Väittäisinkin, että FTA-analyysin tekeminen tietoturvanäkökulmasta ei Shibbolethin kaltaisissa järjestelmissä ole kannattavaa, mikäli tarkoitus on saada kattava kuva monimutkaisen järjestelmän teknisestä tietoturvasta. Analyysimenetelmä pakottaa välttämättä yleistämään asioita. Lisäksi ei liene olemassa varmaa keinoa varmistua siitä, että ollaan varmasti tunnistettu kaikki päätason tapahtumat JA jaettu ne vikapuun oksiksi oikein ja kattavasti. Turhaa analyysin tekeminen ei kuitenkaan ole. Seminaarityöni loppupäätelmänä totean, että FTA-analyysin tekeminen tietoturvanäkökulmasta on hyödyllistä, mikäli

- halutaan kasvattaa systemaattisella tavalla ymmärrystään järjestelmän toiminnasta,
- tietoturvaongelmaa on tarkoitus vain jäsentää eikä niinkään ratkaista,
- tarkoituksena on luoda tarkistuslista yleisistä toimintaperiaatteista, joita jonkin järjestelmän asennuksessa tulee ottaa huomioon, tai

[9]Kyseessähän on vanha hyvä neuvo siitä, miten kaikki pääsynvalvontaan ja käyttöoikeuksiin liittyvä pitäisi aina olla oletusarvoisesti kielteisessä tilassa, eli noudattaa *Default deny policy*:a.

- jos tietojärjestelmän suunnittelu- tai testausvaiheeseen tarvitaan lisäksi menetelmä, jolla haetaan uutta näkökulmaa

Lisäksi mielenkiintoinen jatkokehitettävä idea olisi tutkia, miten tunnistaa jotenkin systemaattisesti päätason tapahtumia sellaisiksi, mitkä moniselitteisyydessään eivät sovi analysoitaviksi FTA-menetelmällä ja sellaisiksi jotka ovat jäsennettävissä selkeiksi kokonaisuuksiksi. Tämä voisi kertoa jotain tapahtuman tai tilanteen "luonteesta" tietoturvan kannalta.

Lähteet

1 Internet2 Shibboleth Project `http://shibboleth.internet2.edu` [17.4.2003]

2 Shibboleth-Architecture DRAFT v05 `http://shibboleth.internet2.edu/draft-internet2-shibboleth-arch-v05.html` [17.4.2003]

3 HSTYA-Projekti - Henkilön sähköinen tunnistaminen yliopistoissa ja ammattikorkeakouluissa - yhteistyöhanke. `https://hstya.funet.fi/` [17.4.2003]

4 HAKA-Projekti - Hakemistot käyttäjähallinnossa. `http://www.csc.fi/proj/hakemistot/haka.phtml` [17.4.2003]

5 Almut Herzog, Nahid Shahmehri, University of Linköping: *Towards Secure E-Services: "Risk analysis of a Home Automation Service"*, Proceedings of the 6th Nordic Workshop on Secure IT Systems (NordSec) 2001.

6 Tietosuojavaltuutetun toimisto - Ota oppaaksi henkilötietolaki! - Ohje Rekisterinpitäjille. `http://www.tietosuoja.fi/uploads/wvzmhnffsvyrhb5.pdf` [17.4.2003]

7 Elizabeth D. Zwicky, Simon Cooper, D. Brent Chapman: *Building Internet Firewalls 2nd edition*, O'Reilly & Associates 2000.

8 U.S. Nuclear Regulatory Commission, NUREG-0492, Washington DC 1981. Fault Tree Handbook. `http://www.nrc.`

gov/reading-rm/doc-collections/nuregs/staff
/sr0492/sr0492.pdf [17.4.2003]

9 Esa Kerttula: *Tietoverkkojen tietoturva*, Liikenneministeriö, Edita 1999. [17.4.2003]

10 Timing Attack on OpenSSL (OpenSSL Private Key Disclosure)
http://www.securiteam.com/unixfocus/5FP0C209FE.html
[17.4.2003]