Teknillinen korkeakoulu
Tietoliikenneohjelmistojen ja multimedian laboratorio
Tik-110.401 Tietoturvallisuuden perusteet, syksy-97.

Uhkapuut

Tekijä: Ari Juhani Korhonen, Tik - jatko-opiskelija

Op. nro: 36784A

Päiväys: 18.12.1997

Uhkapuut

Ari Juhani Korhonen
Teknillinen korkeakoulu
Ari.Korhonen@hut.fi

Tiivistelmä

Uhkapuita on kirjallisuudessa tutkittu varsin vähän. Sekin vähä on lähinnä Yhdysvaltain sisällä tapahtunutta pragmaattista diskussiota mallista. Kansalliseksi rajoittuneen diskussion johdosta uhkapuiden formalismista on saatavana huomattavan vähän tietoa tai tieto on erittäin vaikeasti saatavissa. Tutkielmassa on esitetty näkemyksellinen yhteenveto uhkapuiden käsitteestä, historiasta ja sen käytännön merkityksestä tietoturvallisuuden kehittämisessä. Lisäksi on pohdittu niitä keinoja ja menetelmiä, jolla mallia ja sen formalismia voisi kehittää edelleen.

Johdanto

Uhka edustaa korkeinta tasoa tietokoneiden tietoturvallisuudessa [1]. Tietojärjestelmän haavoittuvuus ja tietojärjestelmää kohti tehdyt hyökkäykset ovat mielekkäitä ainoastaan sellaisessa viitekehyksessä, jossa tietojärjestelmään kohdistuu jokin uhka. Eräs analyyttinen menetelmä, jolla näitä uhkia pyritään määrittelemään on nimeltään uhkapuut.

Uhkapuut eivät ole ainoa lähestymistapa, menetelmä tai rakenne uhkien luokittelemiseksi. Se on kuitenkin saanut jalansijaa erityisesti Yhdysvalloissa, jossa menetelmän käyttöä suosittelee Yhdysvaltain hallituksen ja sotilasinsituutioiden antama standardi [4].

Pragmaattisena menetelmänä sitä on käytetty useissa käytännön tilanteissa, mm. AT&T:n projekteissa sekä StarWars-puolustusmekanismin kehitysohjelmassa [1]. Uhkapuiden hyvyyttä on usein perusteltu sen metodisesti paremmalla mallilla, verrattuna esimerkiksi uhkalistoihin. Uhkalistoissa kantavana ajatuksena on “aivoriihi”, joka tuottaa listan kaikista mahdollisista järjestelmään kohdistuvista uhista. Ongelmana tällöin on mm. tapaukset, joissa menetelmä johtaa ristiriitaisuuksiin, listojen rakentumaton kokonaiskuva sekä niiden riittämättömyys (ne eivät kata kaikkia mahdollisia uhkia).

Rakenteellisuuden lisäksi uhkapuut tarjoavat mahdollisuuden myös laskennalliseen malliin. Erityisesti tätä puolta on seuraavassa pyritty tarkastelemaan lähemmin.

Tämän tutkielman loppuosa koostuu

- uhkapuiden taustasta ja niiden kehittämiseen johtaneista syistä ja ideoista.
- lyhyesti uhkapuiden taustalta löytävästä vikaantumispuiden käsitteestä
- uhkapuiden tarkastelusta laskennallisena mallina sekä
- ajatuksista mallin kehittämiseksi edelleen
- lopuksi esitetään yhteenveto ja johtopäätökset.

Taustaa

Tehdyn lyhyen kirjallisuustutkimuksen perusteella uhkapuita on tutkittu varsin vähän. Edward Amoroson kirjassa [1] uhkapuita käsitellään yhden luvun verran. Kirjassa esitetään pragmaattinen lähestymistapa uhkapuiden kostruoimiseksi. Malli ei kuitenkaan sellaisenaan ole riittävä matemaattisen tarkastelun lähtökohdaksi. Teksti antaa ainoastaan viitteitä siitä kuinka mahdollisesti mallia voitaisiin käyttää analyyttisenä menetelmänä. Yksityiskohtien puuttuminen kuitenkin estää mallin hyödyntämisen sellaisenaan. Tässä tutkielmassa uhkapuista on käytetty sanaa malli, koska tällaisenaan ei voida vielä puhua varsinaisesti menetelmästä, kuten jatkossa tullaan esimerkein osoittamaan.

Kirjan kappaleen 2 lopussa olevat kirjallisuusviitteet eivät nekään ole riittäviä. Osa viitteistä on sellaisia, että niitä on erittäin vaikea, ellei jopa mahdotonta saada käsiinsä. Lisäksi annetut viitteet ovat pääosin konfferenssijulkaisuja ja standardeja, joten varsinaista pitkälle menevää kirjallista materiaalia ei juurikaan aiheesta ole saatavana.

Uhkapuiden taustalta voidaan löytää enemmän tutkittu vikaantumispuiden käsite (*fault trees*), joka juontaa juurensa syvälle laitteistopuolen ongelmiin kyetä arvioimaan jonkin annetun komponenteista (alunperin siis mekaaniset ja elektroniset komponentit) koostuvan järjestelmän luotettavuutta. Aihetta ovat tutkineet mm. Musa et al. [2]. Vikaantumispuuanalyysiin perustuvaa menetelmää, joka lähtee ajatuksesta pyrkiä arvioimaan kaikkein kriittisimpiä osia moduulitasolla, voidaan hyödyntää myös ohjelmistotuotannossa. Tällöin ajatuksena on kiinnittää huomiota juuri niihin ohjelmiston osiin, jotka ovat koko järjestelmän kannalta olennaisimpia, jotta järjestelmä toimisi moitteettomasti. Nämä osat siis nähdään komponentteina, jotka vaativat eniten huomiota. Tämä nähdään mallin erityisenä lisäarvona analysoitaessa järjestelmän turvallisuutta. Lisäksi on syytä huomata, että alkuperäisessä vikaantumispuuanalyysissä eri moduulit tai komponentit voivat vioittua eri tavoin. Toiset viat voivat olla katastrofaalisia kun taas toiset lähes ongelmattomia. Ohjelmistotuotannon puolella mallia on sovellettu siten, että eri moduulit voivat vikaantua vain yhdella tavalla ja kyseessä on aina jossain määrin vakava vika.

Uhkapuiden taustalla oleva tekniikka on siis johdettu edellä esitetystä vikaantumispuu-mallista. Mallin suora siirtäminen tietoturvaa analysoivaksi menetelmäksi on kuitenkin johtanut erinäisiin ongelmiin. Lieneekö tämä sitten syynä siihen, että mallista ei ole loppuun asti mietittyä konseptia ainakaan kovin laajassa levityksessä.

Vikaantumispuut

Vikaantumispuiden avulla on tarkoituksena tuottaa tutkittavasta järjestelmästä malli, joka esittää järjestelmän eri komponentteja puumaisena hierarkiana. Jokainen komponentti voidaan nähdä koostuvan alikomponenteista, joiden vikaantuminen vaikuttaa ylemmän tason komponentin toimintaan. Alimmalla tasolla ovat peruskomponentit, joille voidaan esittää vikaantumismalleja ja joiden vikaantumisen todennäköisyyttä voidaan arvioida esimerkiksi empiiristen tutkimusten perusteella.

Ohjelmistotuotannon näkökulmasta vikaantumispuut voidaan nähdä sellaisina, joissa esiintyy vain yhden tyyppisiä vikaantumisia. Tällöin ongelma jossain määrin yksinkertaistuu. Eri tyyppisille vioille voidaan tehdä erillinen analyysi. Tietoturvan kannalta voidaan kuitenkin antaa esimerkkejä, joissa alkuperäinen ajatus erilaisista vikaantumisista (tietoturvan yhteydessä puhutaan vikaantumisen sijasta uhista tai riskeistä) tulisi pystyä säilyttämään mallissa. Esimerkik-

si salasanalla suojatun salaisen RSA-avaimen joutuminen ulkopuolisen haltuun voidaan nähdä tällaisena turvallisuusriskinä, mutta se ei vielä tee koko järjestelmästä turvatonta (vähäisempi "vika"). Vasta avaimen joutuminen salaamattomana johtaa vakavaan "vikaan".

Mallissa tutkittavan järjestelmän komponentit koostuvat erilaisista alikomponenttien kombinaatioista. Yksinkertaisimmat kombinaatiot ovat loogisilla operaattoreilla AND ja OR muodostettuja. Lisäksi vikaantumispuiden yhteydessä puhutaan KN -rakenteesta (K-OUT-OF-N structure). Tällöin ajatuksena on, että komponentti toimii, jos K sen alikomponenteista (joita on yhteensä N) toimii. Tällaisten rakenteiden käsittely tietoturvamalleissa on kuitenkin rajattu tämän tutkielman ulkopuolelle.

Lisäksi on syytä huomata, että malleissa komponenttien vikaantumiset on oletettu toisistaan riippumattomiksi. Tämä aiheuttaa tietoturvamalleja ajatellen ongelmia. Esimerkiksi tilanne, jossa pyritään arvioimaan pankkikortin katoamisen aiheuttamaa tietoturvauhkaa sen kanssa, että myöskin kortin tunnusluku joutuu ulkopuolisen haltuun voidaan nähdä tietyissa tapauksissa toisistaan riippuviksi. Uhat ovat toisistaan riippumattomia vain, jos voidaan olettaa että jokainen kortin käyttäjä säilyttää tunnuslukua erillään kortista tai ei säilytä kortin tunnuslukua lainkaan kirjoitetussa muodossa. Tyypillisesti kuitenkin on olemassa aidosti positiivinen todennäköisyys, jolla annetussa kortissa tunnusluku on kirjoitettu itse korttiin.

Uhkapuut

Uhkapuiden perusajatus on adoptoitu vikaantumispuista. Menetelmän kantavana ajatuksena on luoda ensin abstrakti, koko tutkittavaa järjestelmää kuvaava uhan käsite (uhkapuun juuri), jota pyritään rakenteellisesti pilkkomaan pienempiin osiin. Mitä syvemmälle puussa edetään, sitä yksityiskohtaisemmiksi uhkien määrittelyt käyvät. Jokainen puun sisäsolmu määrittelee jonkin osajärjestelmän uhkakokonaisuuden ja on näinollen juurena alemman tason uhille. Lopulta lehtisolmuissa kuvataan konkreettinen uhka, joka voidaan esittää esimerkiksi uhkalistana. Tällainen listaesitys on jotakin sellaista, jota ei voida esittää järkevästi puumaisena struktuurina.

Etuja ja haittoja

Uhkapuiden etuna on, että **hyvin** rakennettuna ne pystyvät tietyissä tapauksissa kattamaan kaikki mahdolliset järjestelmään kohdistuvat uhat. Tämä edellyttää puun haarautumista uhaksi ja sen komplementiksi (esimerkiksi sairaalan tietojärjestelmässä potilaan henkeä uhkaavat uhat ja tämän komplementti, ei potilaan henkeä uhkaavat uhat). Ongelmana on luonnollisesti se, että tällainen lähestymistapa ei aina ole mahdollinen tai mielekäs ja se, että lopulta puun lehdissä tulisi pystyä alimman tason uhat määrittelemään kaikenkattavasti. Jälkimmäiseen Amoroso esittää ratkaisuksi uhkalistoja, jotka hän toisaalla esittää riittämättömiksi. Ensin mainittu ongelma esiintyy esimerkiksi tapauksessa, jossa uhkia on enumeroituva määrä. Ajatellaanpa tapausta, jossa palvelun esto -uhka koostuu useasta eri aliuhasta. Jos tällaisia uhkia on enumeroituva määrä (tai siis lähtökohtana on, että kaikkia uhkia on mahdoton tietää ennalta), niin puun haarautuminen "kaikenkattavaksi" jää aina ongelmaksi. Mikäli taas jokin haara määritellään käsitteellä "muut uhat", ollaan tilanteessa, jossa uhkalistoilla uhkia ei voida määritellä loppuun saakka. Uhkapuiden suunnittelu siis vaatii tarkkuutta ja huolellista analyysiä kohdejärjestelmästä.

Toinen keskeinen etu on, että tuloksena saatava analyysi toimii myös järjestelmän uhkien dokumenttina. Saatu dokumentti on helppo ymmärtää ja sitä on helppo tarvittaessa päivittää.

Kolmas uhkapuiden etu on sen rakenteellisuus, joka mahdollistaa mallin matemaattisen tarkastelun. Tätä tarkastelua voidaan myös automatisoida, jolloin erilaisilla CASE-välineillä voidaan suunnittelua tehostaa. Mallin täysimittainen matemaattinen tarkastelu kuitenkin edellyttää mallin kehittämistä edelleen, ainakin siltä osin kuin se on Amoroson kirjassa [1] esitetty. Mallia kohtaan voidaan jopa esittää kritiikkiä, koska se on lähestymistavaltaan liian pragmaattinen.

Neljäs etu on, että uhkapuita voidaan varsin suoraviivaisesti hyödyntää System Security Engineering -menetelmässä.

Seuraavassa on käsitelty mallin laskennallisia ominaisuuksia hieman laajemmin sekä lyhyesti kuvattu mallin käyttöä System Security Engineering -menetelmässä.

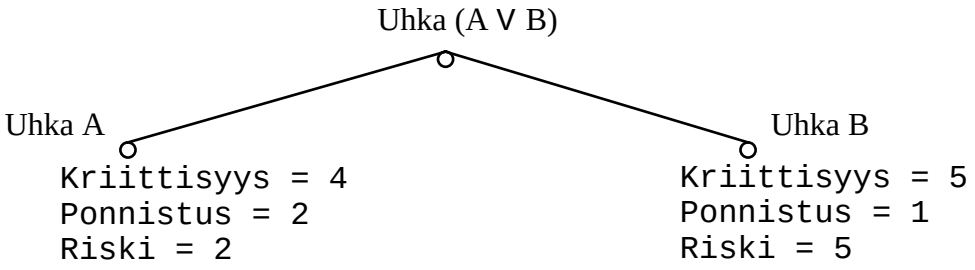
Laskennallinen analyysi

Kuten edellä jo mainittiin, eräs keskeinen uhkapuiden ominaisuus on niiden mahdollistama mallin matemaattinen tarkastelu. Kuten vikaantumispuiden yhteydessä jo todettiin, yksinkertaisimpia loogisia relaatioita eri alipuden välillä ovat AND ja OR-relaatiot. Uhkapuissa nämä voidaan nähdä relaatioina, joissa jokin sisäsolmun uhka toteutuu, jos

1. kaikki aliuhat toteutuvat samanaikaisesti (AND-relaatio) tai
2. jokin aliuhista toteutuu (OR-relaatio).

Tähän asti on oletettu, että samalla tasolla olevat aliuhat edustavat yhtä suuria uhkia. Mallin eri solmuille voidaan kuitenkin antaa painoarvoja sen mukaan, kuinka kriittisestä uhasta on kysymys tai kuinka suuren työn (ponnistus) potentiaalinen hyökkääjä joutuu tekemään, jotta uhka toteutuisi. Esimerkiksi kuvassa 1 on esitetty eräs uhka, joka koostuu kahdesta aliuhasta A ja B. Uhka nähdään toteutuvana, jos toinen sen aliuhista toteutuu (OR-relaatio). Aliuhille on annettu painoarvot, jotka kuvaavat ponnistuksen määrää sekä uhan kriittisyyttä. Uhkaan liittyvä riski on tällöin suoraan verrannollinen uhan kriittisyyteen ja kääntäen verrannollinen ponnistukseen.

Kuva 1: Uhkapuu mallin laskennallisessa tarkastelussa.



Maalaisjärjellä ajateltuna mahdollinen hyökkääjä pyrkisi tässä tapauksessa yrittämään vähemmän ponnisteluja vaativaa uhkaa. Tällöin ylemmän tason uhan riskiksi voitaisiin arvioida suurempi alemmista uhista, eli Riski(A V B) = max { Riski(A), Riski(B) } = 5. Tällöin uhan B pienentäminen olisi järjestelmän turvallisuuden kannalta mielekkäin ratkaisu. Jos esimerkiksi suojauksien parantaminen uhan B torjumiseksi johtaisi tilanteeseen, jossa Ponnistus(B) = 5, saataisiin uhan B uudeksi riskiksi Riski(B) = 5/5 = 1. Tällöin analysoitaessa järjestelmää uudelleen, uhan A suojauksien nostaminen tulisi järjestelmän kannalta paremmaksi vaihtoehdoksi (Riski(A) = 2 > Riski(B) = 1).

OR-relaatiossa siis ylemmän tason riski on aina suurin kaikista alemman tason riskeistä. Näin on laita myös kriittisyyden sekä ponnistuksen osalta. **[**Tässä yhteydessä herääkin kysymys, onko malli tässä suhteessa oikea? Ajatellaanpa tilannetta, jossa kuvan 1 mukaisessa tilanteessa uhat A ja B muodostavat aliuhat luvattomalle pääsylle johonkin järjestelmään. Tällöin A voisi olla esimerkiksi luvaton pääsy käyttäen salasanaa ja B vastaavasti käyttäen avainkorttia. Ylemmän tason uhka toteutuu, jos jompi kumpi aliuhista toteutuu, ts. jos jompi kumpi avaimista joutuu vääriin käsiin. On kuitenkin mahdollista, että ylemmän tason kriittisyys (sille, että järjestelmään päästään luvattomasti) on riippumaton siitä kumpi aliuhista toteutuu. Malli kuitenkin antaa aliuhille mahdollisuuden tilanteeseen, jossa kriittisyys on eri suuri ja jossa *ylemmän tason uhan **kriittisyys** riippuu alemman tason uhan **riskistä***. Ongelma siis syntyy, jos alemman tason uhkia (kriittisyyttä) tarkastellaan erikseen. Toisaalta voidaan miettiä esimerkkiä, jossa ylemmän tason uhkana on vaikkapa Unix käyttöjärjestelmän luvaton käyttö (ylemmän tason uhka), jossa vaihtoehtoina ovat murtautuminen pääkäyttäjän (*root*) tunnuksella (A) tai tavallisen käyttäjän tunnuksella (B). Tällöin varmastikin uhka A on kriittisempi kuin uhka B. Ongelmana on määrittää kriittisyys ylemmän tason uhalle, johon malli ei anna yksiselitteistä vastausta, koska kriittisyys riippuu alemman tason uhkien riskeistä. Edellä esitetystä seuraa, että ilman huolellista tarkastelua voidaan joutua tilanteeseen, jossa saadut lukuarvot eivät olekaan halutunlaisia eikä laskennallinen malli vastaa todellisuutta. Johtuuko tämä siitä, että alkuperäisessä mallissa on jotakin konseptuaalisesti pielessä? Tähän on vaikea yksiselitteisesti vastata, mutta ainakaan tällaisenaan malli ei mielestäni ole täysin ristiriidaton.**]**[1]

Lisäksi (Amoroson puutteellisesti esittämänä) menetelmänä malli ei anna vastausta siihen, kuinka vastaavat arvot voidaan johtaa AND-relaatiolla. Lisäksi kritiikkiä voidaan esittää sitä kohtaan, että annetut mitat ovat kovin subjektiivisia, eikä niillä ole (numeerisia) vastineita reaalimaailmassa. Tässä suhteessa esimerkiksi käsite “todennäköisyys sille, että jokin riski toteutuu” voisi olla mielekkäämpi. Kolmanneksi mallissa ei ole otettu kantaa siihen, onko tässä kohtaa tehty oletus, että eri uhat ovat toisistaan riippumattomia vai ei.

System Security Engineering -menetelmä (SSE) ja uhkapuut

Uhkapuita voidaan luontevasti soveltaa myös SSE-menetelmässä. Menetelmän ensimmäisellä askeleella määritellään järjestelmäarkkitehtuuri sekä identifioidaan järjestelmään kohdistuvat uhat (tässä yhteydessä voidaan puhua myös järjestelmän haavoittuvuudesta). Apuna voidaan nyt luontevasti käyttää uhkapuita, jolloin toisella askeleella eri osa-alueiden riskit voidaan laskea. Mikäli koko jäjestelmälle saatu riskitekijä on tarpeeksi alhainen, voidaan iteraatio lopettaa. Muutoin siirrytään kolmannelle askeleelle, jossa uusia suojauksia asennetaan niihin osiin järjestelmää, joissa saatu riskitekijä oli suurin (haavoittuvuuksien priorisointi ja suojauksien identifiointi sekä asennus). Tämän jälkeen iteraatio aloitetaan jälleen ensimmäisestä askeleesta.

Ajatuksia uuden uhkapuumallin kehittämiseksi

Kuten edellä todettiin, malli ei ole täysin ongelmaton. Lisäksi Amoroson esittämän mallin syvällisempi tarkastelu jäi puutteelliseksi, koska Amoroson antamia viitteitä [3,4] oli erittäin vaikea, ellei mahdoton jälijittää. Kysymysmerkiksi jäikin, pystyykö Amoroson viittaamat alkuperäiset paperit ja julkaisut vastaamaan edellä esitettyihin kysymyksiin. Eräs tällainen kes-

1. Tässä kohtaa itsekritiikkini heräsi ja päädyin epäilemään kritiikkini aiheellisuutta. Koko mallin idea on, että ylemmän tason uhan kriittisyys riippuu alemman tasin uhan riskistä. Oleelliset kysymykset kuitenkin ovat: onko **aina** näin ja pitäisikö jossakin tapauksessa puun sisäsolmussa sallia kriittisyyden eksplisiittinen määrittely.

keinen julkaisu oli J. D. Weissin konfferenssijulkaisu [3], joka käsittelee toista Amoroson viitettä (Yhdysvaltain sotilasstrandardi MIL-STD-1785 [4]), SSE-menetelmää sekä kuinka näitä voidaan hyödyntää yhdessä uhkapuiden kanssa hierarkisessa riskienlaskennassa.

Henkilökohtainen näkemykseni on, että uhkapuut mallina ovat varsin kehityskelpoinen ajatus. Tällöin mallin kehittäminen vastaamaan paremmin matemaattista tarkastelua voisi johtaa kahteen eri tutkimushaaraan. Toisaalta mallia voisi pyrkiä kehittämään todennäköisyyslaskuun ja sen tuomiin valmiisiin laskusääntöihin paremmin sopivaksi, jolloin mallin käsitteet pitäisi formuloida toisin. Mikäli käsitteet halutaan pitää yhtenäisinä Amoroson esittämien kanssa, voisi mallia kehittää esimerkiksi sumean logiikan näkökulmasta.

Todennäköisyyslasku ja uhkapuut

Todennäköisyyslaskuun perustuvassa mallissa lähtökohtana voisi olla eri uhkien määrittäminen todennäköisyyksien avulla uhkapuun lehtitasolla. Tällöin uhan kriittisyyttä voisi arvioida todennäköisyydellä nollasta yhteen. Käsitteen ponnistus määrittelyn kohdalla tulisi olla huolellinen. Mikäli käsite “jätetään laskuista” tai sitä arvioitaisiin sopivalla vakiolla (esim. 1), redusoituisi järjestelmä tapaukseen, jossa uhan toteutumisen todennäköisyydelle (riski) saadaan sama arvo kuin sen kriittisyydelle. Joka tapauksessa saatuja lukuarvoja voitaisiin tällöin hyödyntää laskemalla ylemmän tason solmuille uhan todennäköisyys käyttäen sopivia todennäköisyyslaskun kaavoja. Käytettävät kaavat riippuvat tehdyistä tutkittavaa järjestelmää koskevista perusoletuksista. Tällöin tietyin rajoituksin järjestelmässä voitaisiin ottaa kantaa myös esimerkiksi aikaisemmin mainittuun riippumattomuuteen. Tähän tarjoaa työkalut sovelletun todennäköisyyslaskun ehdollinen todennäköisyys. Todennököisyys U:lle ehdolla V määritellään perinteisesti seuraavasti:

$$P(U|V) = P(U \text{ AND } V) / P(V) \qquad (1)$$

Kaava johtaa monimutkaisempaan malliin, jossa alemmalla tasolla oleville uhille ja riskeille (kriittisyyksille) voidaan laskea toisistaan riippuvia ehdollisia todennäköisyyksiä. Esimerkiksi kahdelle aliuhalle A ja B, jotka riippuvat toisistaan voidaan laskea riskin todennäköisyydet P(A|B) (ja/tai P(B|A)) ja tämän avulla sopivaa metodia käyttäen laskea todellinen riski ylemmän tason uhalle (riippuen siitä oliko kyseessä AND vai OR relaatio).

Sumea logiikka ja uhkapuut

Eräs mielenkiintoinen näkökulma uhkapuihin on nähdä ne sumean logiikan eräänä sovelluskohteena. Analogia sumeaan logiikkaan syntyy Amoroson tavasta määritellä uhkapuiden eri käsitteet. Riskin laskentaan käytetyt muuttujat *kriittisyys* ja *ponnistus* voidaan nähdä “sumeina muuttujina”, koska ne ovat ensinnäkin varsin subjektiivisia ja siksi toisekseen niille voidaan antaa myös sanalliset “arvot”. Esimerkiksi ponnistusta voidaan arvioida sanoilla “vähäinen”, “keskisuuri” ja “suuri”. Lisäksi analogiaa lisää se, että sumeassa logiikassa samoin kuin uhkapuissa operoidaan loogisilla operaattoreilla (AND, OR). Sumeaa logiikkaa käytetään mm. erilaisissa säätöjärjestelmissä. Tässä yhteydessä “säädettävänä” komponenttina voidaan nähdä tutkittavan järjestelmän suojaukset (*safeguards*). Sumeaa logiikkaa on ilmeisesti tutkittu vikaantumispuiden ja tapahtumapuiden (*event trees*) yhteydessä mm. eräässä konfferenssijulkaisussa [5], jota en myöskään saanut valitettavasti käsiini (ei tule TKK:n kirjastoon).

Yhteenveto

Uhkapuu edustaa mallina erästä analyyttistä menetelmää, jolla tietoturvauhkia voidaan määritellä rakenteellisesti. Malli uhkapuiden konstruoimiseksi on adoptoitu alun perin mekaanisille ja elektronisille laitteille suunnitellusta vikaantumispuu-mallista ohjelmistotuotantoon ja siitä edelleen johdettu tietoturvapuolelle analyyttiseksi menetelmäksi. Tehdyn kirjallisuustutkimuksen perusteella menetelmää on tutkittu melko vähän ja siitä kirjoitettuja julkaisuja on erittäin vaikea saada käsiin, ainakin tälle työlle annetun aikarajan puitteissa[1].

Menetelmän keskeiset edut ovat seuraavat:

1. Se tarjoaa hierarkisen rakenteen uhille,
2. se toimii itsessään dokumenttina,
3. se tarjoaa työkaluja laskennalliseen analyysiin sekä
4. sitä voi hyödyntää suoraviivaisesti SSE-menetelmässä.

Menetelmää kohtaan voidaan esittää myös seuraavanlaista kritiikkiä:

1. Se on liian pragmaattinen,
2. siinä on tehty liian yleistäviä oletuksia (esim. riippumattomuus)
3. |laskennallinen analyysi voi johtaa ristiriitaisiin tuloksiin ja]
4. mallin käsitteet ovat subjektiivisia.

Johtopäätökset

Menetelmänä uhkapuut tarjoavat potentiaalisen vaihtoehdon pyrittäessä kartoittamaan järjestelmien turvallisuutta. Sellaisenaan malli ei kuitenkaan ole täysin riittävä. Mallin käytännön hyödyntäminen edellyttää jotakin tai kaikki seuraavista:

1. Alkuperäisten menetelmää koskevien julkaisujen etsimisen,
2. mallin edelleenkehittämisen sovelletun todennäköisyyslaskun sovelluskohteeksi ja/tai
3. mallin edelleenkehittämisen sumean logiikan sovelluskohteeksi.

Aiheena uhkapuut ovat jatkotutkimisen arvoisia. Tällöin menetelmänä se voisi toimia esimerkiksi erikoistyön tai miksei pidemmällekin menevän tutkimuksen aiheena.

1. Sain tilattua Weissin [3] paperin eräältä henkilöltä USA:sta, mutta sitä ei kahden viikon kuluessa kuulunut.

Kirjallisuus

[1] Amoroso, Edward G.; “Fundamental of Computer Security Technology”. AT&T Bell Laboratories. Prentice Hall, USA, 1996. ISBN 0-13-108929-3.

[2] Musa John D.; Iannino Anthony; Kazuhira Okumoto; “Software Reliability: Measurement, Prediction, Application”. AT&T Bell Laboratories, Whippany, NJ. USA. McGraw-Hill, 1987. ISBN 0-07-100208-1.

[3] Weiss, Jonathan D.; “A System Security Engineering Process”. AT&T Bell Laboratories. 14th National Computer Security Conference (NIST/NCSC), Oct 1991, Washington, USA, pp. 572-581.

[4] Department of Defence; “System Security Engineering Program Management Requirements”. MIL-STD-1785, June 20, 1988.

[5] Ross, Timothy J.; Donald, Sunil; “Fuzzy logic paradigm for fault trees and event trees in risk assessment”. Univ of New Mexico . Albuquerque, NM. USA. Lehti: Computing in Civil Engineering (New York). 1996. p 369-375. Konferenssi: Proceedings of the 1996 3rd Conference on Computing in Civil Engineering. Anaheim, CA. USA.