Teksti: Ilkka Pirttimaa

*Kirjoittaja työskentelee pääarkkitehtina konsernin tietohallinnossa Stockmannilla. Työssään hän pyrkii löytämään ja käyttöönottamaan menetelmiä, joilla saadaan kehitettyä liiketoiminnan tarpeisiin sovelluksia jotka ovat helppokäyttöisiä ja turvallisia.*

# Tietoturvan määritteleminen väärinkäyttötapauksia mallintamalla

**Perinteisesti järjestelmän määrittelytyössä tunnistetaan käyttäjät ja heidän käyttötapauksensa. Järjestelmän turvallisuus määritellään kuitenkin usein ei-toiminnallisena vaatimuksena, jolloin usein lopputuloksena on myös ei-toiminnallinen turvallisuus. Tässä artikkelissa esitellään menetelmä, kuinka väärinkäyttäjiä ja väärinkäyttötapauksia mallintamalla myös järjestelmän turvaominaisuudet saavat toteutuksessa paremmin tavoiteltavan muodon. Lopuksi esitetään, kuinka väärinkäyttötapausmallinnus voidaan liittää osaksi yritysarkkitehtuuritasoa ja mikä tärkeintä – Kuinka piirrosvälineet opetetaan uusille tavoille.**

## Järjestelmän toiminnallisuuden kuvaus käyttötapauksina

Lähes kahden vuosikymmenen ajan järjestelmän toiminnallisuuden määrittämiseksi on käytetty käyttötapauskuvauksia. Kaikki tuntevat UML:n visuaalisen tavan piirrellä järjestelmän käyttäjiä tai suunniteltavan järjestelmän ulkopuolisia järjestelmiä tikku-ukkoina. Nämä käyttäjät, aktorit, käyttävät käyttötapauksia, jotka mallinnetaan valkoisina soikioina. Käyttötapaukset voivat puolestaan kutsua toisia käyttötapauksia. Varsinaiset käyttötapaukset voidaan kuvata hyvinkin tarkkaan tekstinä, jossa kuvataan, miten käyttäjän ja järjestelmän vuorovaikutus kulkee, esiehtoineen ja poikkeuskäsittelyineen.

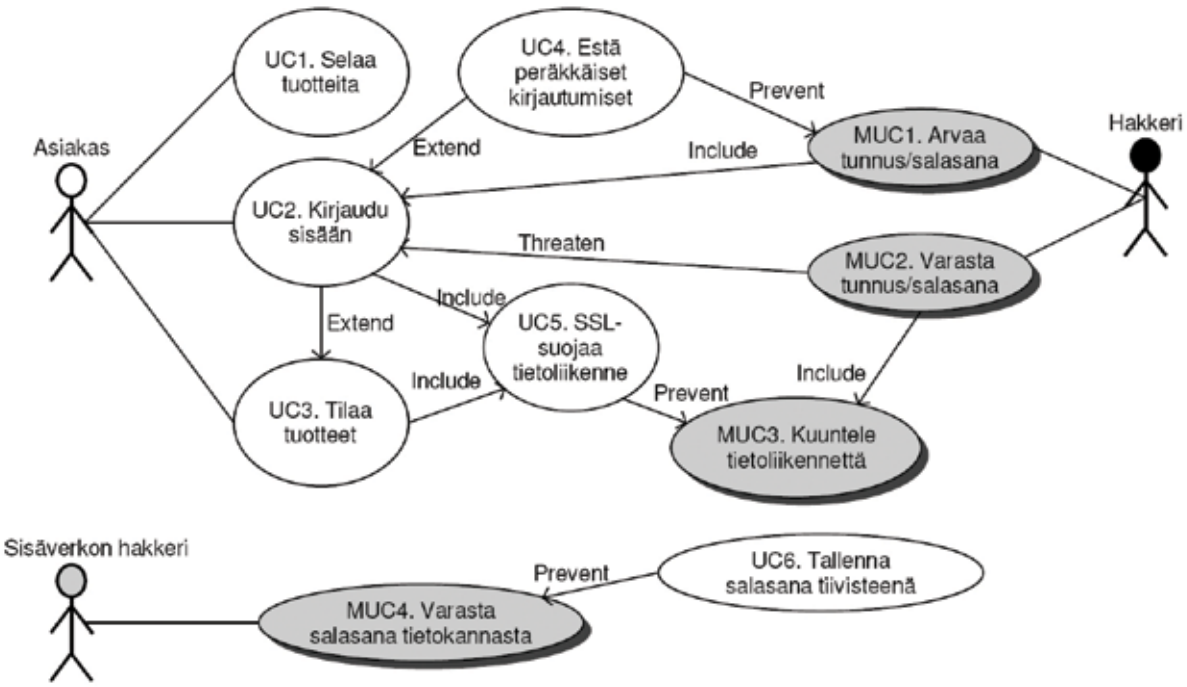
Tässä artikkelissa on esimerkkikaavio (Kuva 1.) kuvitteellisesta nettitilausjärjestelmästä, jonka liiketoimintavaatimuksina on tuoteselailu (UC1), sisäänkirjautuminen (UC2) ja tuotteiden tilaus (UC3).

## Järjestelmien tietoturvaongelmat

Kun järjestelmän käyttötapauksille määritellään testitapauksia, hyvä määrittelijä lisää mukaan testejä, joilla voidaan havaita myös järjestelmään mahdollisesti syntymiä tietoturvaongelmia. Tämä menettely ei kuitenkaan aina takaa, että järjestelmän turvallisuutta olisi kattavasti mietitty tai suunniteltu.

Usein tietoturvavaatimukset kuvataan yleisinä ei-toiminnallisina vaatimuksina, tyyliin ”Järjestelmän on oltava tietoturvallinen”. Tämä helposti kuitataan ajatuksella, että kunhan järjestelmä asennetaan palomuurin taakse ja ylläpitäjät tekevät alustaan tietoturvapäivitykset, niin järjestelmä on turvallinen. Käytännössä tietoturvaongelmat ovat nykyään ohjelmiston toiminnallisuuteen liittyviä ja näin ollen palomuurilla ne eivät häviä.

Tällaisia ongelmia ovat mm. mahdollisuus varastaa tunnuksia/salasanoja tai muuttaa järjestelmän tietokantaa SQL Injection-hyökkäyksillä puuttellisen syötteentarkistuksen takia.




*Kuva 2. Liiketoiminnan käyttötapaukset sovellukselle ovat UC1-3. Hyökkäjän käytttötapaukset ovat MUC1-4. Järjestelmään on määritelty UC4-6 estämään hyökkääjien aikeet.*

Myös järjestelmän toteutusta tarjoavalle toimittajalle ei-toiminnalliset vaatimusmääritykset ovat haastavia: Kyseessä on toiminnallisuus, joka ei ole mitään konkreettista. Mikäli järjestelmän tarjouksessa varaudutaan tarkemmin määrittelemättömien turvallisuusaspektien toteutukseen, on tarjous kalliimpi kuin kilpailijalla.

## Väärinkäyttäjät ja väärinkäyttötapaukset

2000-luvulla Norjalaiset Guttorm Sindre ja Andreas L. Opdahl esittivät laajennoksen perinteiseen käyttötapausmallinnukseen. Ajatuksena on tunnistaa järjestelmän käyttäjien lisäksi potentiaaliset väärinkäyttäjät. Väärinkäyttäjät kuvataan mustina tikku-ukkoina. Tämän jälkeen mietitään, mitkä ovat väärinkäyttäjien käyttötapaukset. Esimerkissäni hakkeri yrittää arvata brute-forcella käyttäjätunnuksia ja salasanoja (MUC1) sekä varastaa tunnuksia/salasanoja (MUC2). MUC1 sisältää käyttötapauksen UC1 (Kirjaudu sisään) – Näin on siis määritelty, kuinka hakkeri pyrkii väärinkäyttämään järjestelmää.

Kun väärinkäyttötapaukset on tunnistettu voidaan suunnitella uusia järjestelmään toteutettavia ominaisuuksia, joilla estetään väärinkäyttäjän käyttötapaus. Annetussa esimerkissä näin saatiin järjestelmään kaksi uutta toiminnallista ominaisuutta: UC4 estää peräkkäiset kirjautumisyritykset ja UC5 lisää vaatimuksen kirjautumistoiminnon ja tuotetilauksen tietoliikennesalaukselle. Samalla saadaan lisää systematiikkaa testitapausten määrittelylle: On helppo lisätä testit, joissa tarkistetaan UC4:n ja UC5:n toimivuus.

## Sisäverkon väärinkäyttötapaukset

Sindren ja Opdahlin menetelmästä löytyy myös oma harmaan värinen tikku-ukko sisäverkon väärinkäyttötapausten mallintamiseen. Mallissani olen piirtänyt tämän vasemmalle kuvaamaan paremmin järjestelmien käyttäjien keskuudesta tulevaa uhkaa. Esimerkissäni kuvataan uhka, jossa sisäverkon hakkerin väärinkäyttötapaus on järjestelmän käyttäjien tunnusten ja salasanojen varastaminen suoraan tietokannasta. Tämä väärinkäyttötapaus estetään yksinkertaisesti lisäämällä vaatimus, että salasanat tallennetaan tietokantaan tiivisteinä. Näin uhka on tunnistettu ja siihen on reagoitu suunnittelemalla järjestelmä paremmin.

## Muita mahdollisia väärinkäyttäjiä

Väärinkäyttäjien ja väärinkäyttötapausten mallintaminen mahdollistaa kommunikoinnin liiketoiminnan kanssa pohdittaessa riittävää tietoturvan tasoa : Mikä on oikea riskien vs. toteutuskustannusten hinta. On hyvä tunnistaa iso joukko potentiaalisia väärinkäyttäjiä ja kirjata myös päätökset, joissa riski on koettu niin pieneksi, ettei järjestelmään kannata rakentaa suojaus- tai valvontamekanismeja näitä vastaan. Hyviä ehdokkaita väärinkäyttäjiksi ovat esim:

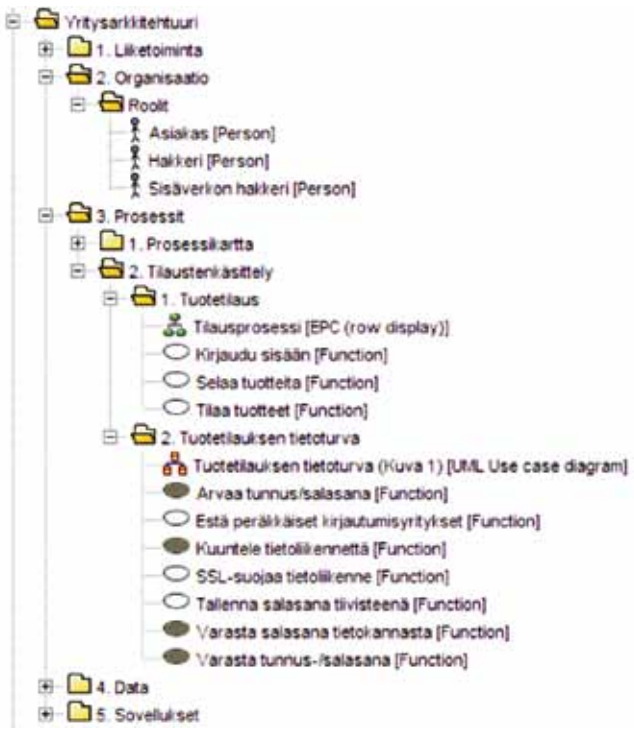
- Sosiaalinen hakkeri puhelimessa
- Sosiaalinen hakkeri työpaikalla
- Siivoojaksi rekrytoitunut hakkeri
- Irtisanottu työntekijä
- Lahjottu työntekijä
- Nettihakkeri
- Kilpailijan palkkaama hakkeri

## Väärinkäyttötapaukset osana yritysarkkitehtuureita

Väärinkäyttömallinnuksella on tärkeä osa tietoturvakysymyksissä yritysarkkitehtuuritasolla. On tärkeää, että kuvaukset tehdään johdonmukaisesti keskenään linkittäen ja jo olemassa olevia kuvauksia uudelleenkäyttäen.

Kuvassa 2 esitetään, kuinka myös väärinkäyttäjäroolit on määritelty yleiskäyttöisesti organisaatiohaaraan. Varsinainen liiketoimintaprosessi on kuvattu prosessikaaviona kansiossa ”1. Tuotetilaus” – Siinä muodossa, jossa liiketoiminta haluaa prosessin nähdä.

Kansioon ”2. Tuotetilausten tietoturva” on kerätty vain ne käyttötapaukset, jotka nähdään tietoturvan kannalta oleellisiksi – Osa on väärinkäyttäjän käyttötapauksia, osa on toiminnallisuutta jota tarvitaan niiden estämiseksi. Väärinkäyttötapauskaaviossa (Kuva 1.) varsinaisessa liiketoimintaprosessissa esiintyvät toiminnot ovat oleellisessa osassa – Ne voidaan raahata UML-kaavioon prosessikuvasta, jolloin niiden ulkomuoto muuttuu käyttötapauspallukaksi, objektin periessä kaikki muut tietonsa prosessikaaviosta.




*Kuva 2. Väärinkäyttötapaukset osana yritysarkkitehtuurikokonaisuutta.*

## Mallinnustyökalut

Artikkelia kirjoittaessani havaitsin, että mallinnintyökaluilla voi olla vaikea piirtää mustia/harmaita tikku-ukkoja ja käyttötapauksia. Käyttötapauskaavio ei välttämättä myöskään mahdollista uusien assosiaatioiden piirtämistä (uhkaa, estää). Useilla työkaluilla ”käyttötapausmalliin kuulumattomia” nuolityypityksiä pitää lisätä käsin erillisinä teksti- tai kommenttilaatikoina.

Artikkelin kuvat on piirretty ARIS-mallintimella. Senkään UML-kaaviotyyppi ei tue suoraan tarvittavia assosiaatioita, mutta sen ylläpitotyökaluilla ohjelmalle voidaan opettaa tarvittavat asiat. Objekteille opetettiin 2 uutta attribuuttia, joista ensimmäinen sisältää tarvittavat assosiaatioiden nimet (include, extend, threaten, prevent, …) ja jälkimmäinen (Kuva 3.) sisältää alasvetolistana tiedon, onko kyseessä käyttötapaus, väärinkäyttötapaus vai sisäinen väärinkäyttö. Valittu assosiaatio saadaan näkymään nuolissa ja tikku-ukon/käyttötapauksen väri saadaan vaihtumaan valitun tyypin mukaan. Näin saadaan vakioitua menettely, jolla syntyy Kuva 1:n mukaisia kaavioita.

MS Visiolla tarvittavat assosiaatiot tuodaan käyttöön seuraavasti :

Luodaan uudet stereotyypit:

1. Avaa UML/Stereotypes
2. Lisää 4 stereotyyppiä (include, extend, prevent, threaten) ja valitse niille Base Class=Dependency

*Kuva 3. ARIS-mallintimeen voidaan lisätä omia attribuutteja, joilla saadaan lisättyä tarvittava semantiikka*


Kaavion piirtäminen

1. Piirrä aktorit ja käyttötapaukset käyttäen ”UML Use Case”-piirrosobjekteja
2. Värjää väärinkäyttäjät ja väärinkäyttötapaukset siten, että fontti on valkoinen ja tausta on musta.
3. Piirrä käyttötapausten väliset suhteen käyttäen ”UML Static Structure”-piirrosobjektivalikoiman Dependency-nuolta
4. Tuplaklikkaa nuolta ja valitse alasvetolistalta haluttu stereotyyppi, jonka aiemmin lisäsit

## Yhteenveto

Tietoturvan huomioiminen ohjelmistokehityshankkeissa on usein abstrakti asia, johon ei välttämättä osata oikealla tavalla tarttua. Se, kenen vastuulla tietoturvan rakentaminen on, saattaa olla hämärän peitossa. Arkkitehdit luottavat koodareihin, koodarit luottavat palvelimen asentajiin, palvelimen asentajat luottavat tietoliikennekavereihin. Ilman oikeanlaista suunnittelua ja tietoturvariskikartoitusta järjestelmästä ei tule turvallinen ja sen korjaaminen jälkikäteen voi olla hyvin kallista. Juuri tällaiseen kokonaisvaltaiseen tietoturvasuunnitteluun kuvattu menetelmä on hyvä. Se tarjoaa tarttumapintaa sekä toteuttajille että järjestelmän testaajille – mitä turvaominaisuuksia järjestelmään pitää toteuttaa ja millä niiden toteutukset todennetaan ja testataan.

Olipa suunnittelutyökaluina sitten mikä tahansa ohjelma tai vaikkapa vain kynä ja paperi, suosittelen tämän suunnittelumenetelmän käyttöönottoa. Vain ottamalla menetelmän rohkeasti käyttöön voi työkaluohjelmiltakin odottaa kehitystä tukemaan tätä mainiota menetelmää. Myös UML kehittyy!

Artikkelissa käytetyt suomennokset

| **Käyttötapauskuvaus** | | **Väärinkäyttökuvaus** | |
|---|---|---|---|
| Actor | Käyttäjä | Misuser | Väärinkäyttäjä |
| Use case | Käyttötapaus | Misuse case | Väärinkäyttötapaus |
| Include | Sisältää | Threaten | Uhkaa |
| Extend | Laajentaa | Prevent, Mitigate | Estää |

*Lähteet:*

*Guttorm Sindre, Andreas L. Opdahl:*

*Capturing Security Requirements through Misuse Cases*