|  |
| --- |
|  |
| **Turvallisen sovelluskehityksen käsikirja** |
|  |
|  |
|  |

Sisällysluettelo

[1 Johdanto 3](#_Toc5165078)

[1.1 Millaiseen ohjelmistotuotantoon ohje on suunnattu? 3](#_Toc5165079)

[1.2 Ohjeen kohdeyleisö 3](#_Toc5165080)

[2 Vaatimustenmukaisuus 5](#_Toc5165081)

[3 Tietoturva ja tietosuoja ohjelmistokehitysprosessissa 7](#_Toc5165082)

[3.1 Yleiskuvaus 7](#_Toc5165083)

[3.2 Vaatimukset ohjelmistokehitysprosessille 8](#_Toc5165084)

[3.3 Vastuunjako 9](#_Toc5165085)

[3.4 Erityishuomioita tiettyihin ohjelmistokehitys- ja kulttuurisiin malleihin 13](#_Toc5165086)

[4 Ohjelmistoturvallisuuden seurannan ja ulkoisen tuen malli 14](#_Toc5165087)

[4.1 Jatkuva tietoturvatyön näkyvyys ja työn jälkikäteinen auditoitavuus 14](#_Toc5165088)

[4.2 Tukipalveluiden työnohjausmalli 14](#_Toc5165089)

[Liite 1: Tuotantoon viennin vaatimukset 15](#_Toc5165090)

[Liite 2: Tietoturvallisuuden yleisperiaatteet 17](#_Toc5165091)

[Tietoturvallisen arkkitehtuurin ja suunnittelun yleisperiaatteet 17](#_Toc5165092)

[Tietoturvallisten palvelurajapintojen ja käyttöliittymien yleisperiaatteet 19](#_Toc5165093)

[Tietoturvatestauksen ja teknisen tarkastuksen yleisperiaatteet 21](#_Toc5165094)

[Auditoitavuuden yleisperiaatteet 22](#_Toc5165095)

[Kehitysympäristöjen ja tuotantoon viennin yleisperiaatteet 24](#_Toc5165096)

[Käyttöpalveluympäristöt ja operationaaliset yleisperiaatteet 26](#_Toc5165097)

[Tietosuojan suunnittelun yleisperiaatteet 27](#_Toc5165098)

[Liite 3: Uuden tai muuttuneen toiminnallisuuden tietoturva- ja tietosuojatyön tarkastuslista 29](#_Toc5165099)

[Liite 4: Tietoturva-arkkitehtuurin dokumentoinnin vähimmäisvaatimukset 31](#_Toc5165100)

[Liite 5: Uhkamallinnuksen toteutusohje 32](#_Toc5165101)

[Uhkamallinnus yleisesti 32](#_Toc5165102)

[Uhkamallinnuksen aikataulutus 32](#_Toc5165103)

[Uhkamallinnus käytännössä 32](#_Toc5165104)

[Uhkamallinnuksen laatu 34](#_Toc5165105)

[Liite 6: Tietosuojan toteutumisen varmistaminen 35](#_Toc5165106)

[Liite 7: Tekninen liite 36](#_Toc5165107)

Turvallisen sovelluskehityksen käsikirja

# Johdanto

Tässä dokumentissa kuvataan <organisaatio> sovelluskehityksen keskeiset tietotur­vallisuuden periaatteet, vaatimukset sekä kontrollit. Dokumenttia käytetään <organisaatio> sovelluskehityksessä.

## Millaiseen ohjelmistotuotantoon ohje on suunnattu?

Ohje on kirjoitettu olettaen, että ohjelmistotuotannossa seurataan nykyaikaiselle ohjelmistotuotannolle ominaisia periaatteita:

1. Kehitettävän kohteen toiminnallisuutta kehitetään iteratiivisesti ja kehityksen kohde voi muuttua nopeastikin (*agile* eli ketterä kehittäminen).
2. Tuotantoon vienti-prosessissa ei ole estettä sille, etteikö integraatiota ja tuotantoon vientiä voitaisi automatisoida (*continuous integration / delivery / deployment*). Automaation ei tarvitse olla vielä olemassa, mutta tuotantoon vienti-prosessissa ei saisi olla tätä estäviä vaiheita.
3. Kehitystiimi on suurelta osin autonominen ja yhdessä tuoteomistajan kanssa kantavat liiketoimintavastuun myös tietoturvasta ja joissakin tapauksissa myös tuotannonaikaisista toimenpiteistä (*DevOps*).

Tilanteessa, jossa näitä periaatteita ollaan ottamassa käyttöön, ohjeen voi ottaa käyttöön soveltuvin osin samalla kun periaatteet kehittyvät.

Käsikirjan kuvaamat työnkulut on tarkoitettu käytettäväksi <organisaatio> kokonaisketterän kehityksen viitekehyksessä. Se kuitenkin soveltuu käytettäväksi myös yksittäisissä projekteissa, jotka eivät ole kokonaisketterän kehityksen alaisuudessa.

Mikäli ohjelmistotuotanto on osin ulkoistettu - esimerkiksi ostavalla organisaatiolla on tuoteomistaja, mutta itse ohjelmistokehitys tehdään toimittajan toimesta toisaalla - ohjetta voi soveltaa niiden roolien osalta, jotka ovat ostavan organisaation hallussa. Muut ohjeen osat tulisi vaatia toimittajalta sopimuksin.

Useat ohjeen tietoturvaperiaatteet on kirjoitettu niistä lähtökohdista, että arkkitehtuuri voi olla nk. pilvinatiivi ja palvelu- tai mikropalveluorientoitunut, ja että kehitystiimillä voi olla valtaa ja vastuuta käytönaikaisista toimenpiteistä. Ohjetta voi kuitenkin soveltaa, vaikka näin ei olisikaan.

## Ohjeen kohdeyleisö

Ohjeen kohdeyleisönä ovat seuraavat **roolit**. Riippuen organisaatiosta, sama henkilö voi toimia useissa eri rooleissa. Tässä ohjeessa roolien nimiä käytetään siten kuin ne useimmiten ohjelmistotuotannossa ymmärretään, eikä niillä ole suoraa vastaavuutta virkanimikkeisiin. Usein esimerkiksi pääkehittäjällä on vastuita, jotka ulottuvat kaikkien alla mainittujen roolien alueille.

Tarkempi roolien vastuunjako on kuvattu kappaleessa Vastuunjako.

* **Hankepäälliköt.** Hankepäälliköt vastaavat kokonaisketterän kehitystyön vastuidensa puitteissa siitä, että portfolio- ja suunnittelutason toiminnoissa tehtävä tietoturvatyö mahdollistuu.
* **Tuoteomistajat (*product owner*).** Tuoteomistaja on palvelun tai sovelluksen liiketoiminnallinen omistaja, joka tulkitsee asiakkaan tarpeita toiminnallisiksi ja ei-toiminnallisiksi vaatimuksiksi ja erityisesti kaupallisissa yhteyksissä saattaa kantaa myös tulosvastuuta. Tuoteomistaja päättää viime kädessä eri toiminnallisuuksien ja muun ohjelmistokehitystyön prioriteeteista, joten hänellä on tällä tasolla myös valtaa ajan ja muiden resurssien käyttöön ja jäännösriskien hyväksyntään. Tuoteomistaja vastaa viime kädessä siitä, että järjestelmän tai palvelun kehityksessä on otettu huomioon tietoturva- ja tietosuojavaatimukset.
* **Arkkitehdit.** Arkkitehti vastaa kehitettävän palvelun tai sovelluksen sopivuudesta sitä ympäröivään tietotekniseen ympäristöön sekä on merkittävässä roolissa käytettyjen strategisten tekniikkavalintojen suhteen. Arkkitehdin suunnitteluhorisontti on yleensä koko palvelun elinkaaren mittainen, ja häneltä odotetaan näkemystä palvelun toteutukseen myös erityisnäkökulmista, joista yksi on tietoturva-arkkitehtuuri. Arkkitehdin roolissa toimivat esimerkiksi järjestelmäarkkitehdit, ”koodaavat arkkitehdit” sovelluskehitystiimeissä ja järjestelmäpäälliköt.
* **Ohjelmistokehittäjät.** Ohjelmistokehittäjät toimivat useimmiten pienehköissä tiimeissä ja ovat vastuussa toiminnallisuuden toteuttamisesta, testaamisesta, tuotantoon viennistä ja joissakin tilanteissa käytönaikaisista toimenpiteistä. Tietoturvamielessä ohjelmistokehittäjillä on päävastuu tietoturvan teknisestä toteutumisesta.
* **Tietoturva-asiantuntijat.** Tuoteomistajat, arkkitehdit ja ohjelmistokehittäjät voivat tarvittaessa käyttää apuna erityisesti tietoturvaan keskittyneitä henkilöitä, joita ovat kehitysprojektiin varatut tietoturvatestaajat, -arkkitehdit tai -asiantuntijat. Tietoturva-asiantuntijat tukevat tyypillisesti tietoturvariskien löytämisessä ja arvioinnissa, suunnitteluperiaatteiden ja ohjelmakoodin tarkistamisessa sekä tutkivassa tietoturvatestauksessa. Myös ohjelmistokehitystiimeissä itsessään voi olla tietoturvaan erikoistunut henkilöitä. Tietoturva-asiantuntijat ovat tässä ohjeessa konsultatiivisessa roolissa, ja lopullinen vastuu tietoturvan toteutumisesta on yllämainituilla rooleilla.
* **Käyttöpalvelujen toimittajat.** <organisaation> on voitava luottaa siihen, että käyttöpalvelujen toimittaja takaa tietoturvan toteutumisen omalta osaltaan. Joissakin tapauksissa ohjelmistokehittäjät voivat kantaa itse vastuuta osasta käyttöpalveluita (erityisesti DevOps-tiimeissä); tämä on kuitenkin aina sovittava erikseen.

# Vaatimustenmukaisuus

Tietoturvavaatimukset jaetaan neljään ryhmään niiden pakollisuuden mukaan.

Jokaisesta pakollisesta vaatimuksesta sekä valinnaisista toteutetuista vaatimuksista on luotava joko vaatimus tuotteen tehtävälistalle tai vaatimuksen täyttymistä on ajettava jollakin tietoturva-aktiviteetilla, josta jää todiste (kuten esimerkiksi testitulos). Tällä tavoin vaatimuksista syntyy auditoitavissa oleva todiste myöhempää tarvetta varten.

Pakolliset (lakiin perustuvat) vaatimukset otetaan huomioon kokonaisketterän kehitysmallin portofiokanbanissa. Tuotantoon viennin vaatimukset otetaan huomioon pääosin kokonaisketterän kehityksen suunnittelutasolla. Nämä johtavat todennäköisesti lopulta tehtävälistan tehtäviin, joilla vaatimustenmukaisuus syntyy.

Yleiset tietoturvaperiaatteet on koottu ryhmiksi, joista kussakin roolissa olevan henkilön on omaksuttava omaan rooliinsa liittyvät periaatteet ja huolehtia periaatteiden noudattamisesta oman työnsä viitekehyksessä. Esimerkiksi käyttöliittymäsuunnittelijoiden tulisi tutustua rajapintoja ja käyttöliittymiä koskeviin yleisiin tietoturvavaatimuksiin.

Viitteelliset vaatimukset on tarkoitettu ensisijaisiksi lähteiksi tietoturva-aktiviteetteja laajennettaessa esimerkiksi asiakasvaatimusten pohjalta. Tuoteomistajan tulee tunnistaa asiakasvaatimuksista tietoturvatarpeet ja näihin on vastattava ensisijaisesti viitteellisten vaatimusten keinoin.

|  |  |  |  |
| --- | --- | --- | --- |
| Vaatimustyyppi | Vaatimuslähde | Kuka voi päättää vaatimuksen tekemättä jättämisestä | Huomioita |
| Pakolliset vaatimukset | Kansallinen lainsäädäntö, GDPR[[1]](#footnote-1), eIDAS[[2]](#footnote-2) ja tuleva ePrivacy-asetus |  | Huomioidaan portfoliokanbanissa.  Lain ja asetusten vaatimuksia ei voi jättää tekemättä. <organisaation> tietosuojavastaava voi tarjota apuaan niiden tulkinnassa. |
| Tuotantoon viennin vaatimukset | Liite 1: Tuotantoon viennin vaatimukset | Tuoteomistaja konsultoituaan <ORGANISAATIO>:n tietosuojavastaavaa tai tietoturvahenkilöstöä | Huomioidaan suunnittelukanbanissa. Tekemättä ­­jättämis­päätös on dokumentoitava |
| Yleiset tietoturvaperiaatteet | Liite 2: Tietoturvallisuuden yleisperiaatteet | Tuoteomistaja konsultoituaan <ORGANISAATIO>:n tietosuojavastaavaa tai tietoturvahenkilöstöä | Kussakin roolissa toimiva henkilö tutustuu itselleen olennaisiin periaatteisiin ja huolehtii niiden täyttymisestä oman työnsä kontekstissa. Tekemättä­­ jättämis­päätös on dokumentoitava |
| Viitteelliset vaatimukset | VAHTI 1/2013[[3]](#footnote-3), OWASP ASVS[[4]](#footnote-4), A29WP WP248[[5]](#footnote-5), CIS Benchmarks[[6]](#footnote-6) | Tuoteomistaja | Nämä vaatimukset otetaan mukaan vain tarvittaessa, esimerkiksi kun tarvitaan tietyn aihealueen tarkempaa ohjeistusta. |

# Tietoturva ja tietosuoja ohjelmistokehitysprosessissa

## Yleiskuvaus

Tietoturva ja tietosuoja tuodaan tuotekehitysprosessiin tekemällä ne näkyväksi työnohjauksessa. Yleisellä tasolla tärkein perusperiaate on esitetty kuvassa.



Tietoturva- ja tietosuojatyö alkaa kokonaisketterän kehitysmallin portfolio- ja suunnittelutasoilla. Näillä tasoilla luodaan edellytykset suurimpien tietoturvavaatimusten noudattamiselle.

Valtaosa käytännön suunnittelun ja toteutuksen tietoturva- ja tietosuojatyöstä tuodaan näkyväksi tuotteen tehtävälistalla (engl. *product backlog*). Perustapauksessa tämä on kaksivaiheinen prosessi: tuoteomistaja huomaa uhkamallinnusta (tai tietosuojatapauksessa tietosuojavaikutusten arviointia) tarvitsevat tehtävät erillisen ohjeen mukaan, ja ohjelmistokehittäjät suorittavat uhkamallinnuksen ja lisäävät sen perusteella löytyneet uudet tietoturvatehtävät tehtävälistalle.

Nämä tietoturvatehtävät luokitellaan (*label*) yhteisesti sovittuihin luokkiin, ja linkitetään siihen toteutustehtävään, johon ne liittyvät. Näin organisaation tietoturvahenkilöstö voi seurata tietoturvatyön kertymistä ja etenemistä ja mahdollisessa auditointitilanteessa on helpompi löytää todisteet tehdystä työstä.

Tärkeimmät dokumentissa mainitut tietoturvatehtävät ovat **uhkamallinnus**, **tietosuojavaikutusten arviointi**, **tietoturva-arkkitehtuurin ylläpito** ja **tietoturvatestaus**. Näiden välinen suhde on tiivis. Käytännössä kaikissa tehtävissä on yhteisiä osa-alueita, ja ne hyötyvät toisistaan näiltä osin - sama työ tehdään vain kertaalleen. Suhdetta voidaan kuvata seuraavasti:



Tekstin sujuvuuden vuoksi dokumentissa puhutaan jatkossa pääosin tieto**turva**työstä vaikka sillä tarkoitetaan myös tieto**suoja**työtä. Samaten tietoturva**tehtävä** voi olla minkä tasoinen tehtävä vain, esimerkiksi käyttäjätarina tai yksinkertainen bugikorjaus.

## Vaatimukset ohjelmistokehitysprosessille

Jotta ohjelmistokehitysprosessi mahdollistaa tietoturvallisen kehityksen ja tietosuojan tason tämän ohjeen mukaisesti, sen on täytettävä joitakin perusvaatimuksia. Ohjelmistokehitys, jossa seuraavat vaatimukset eivät täyty, ei ole tämän tietoturvaohjeen mukainen. Tämä koskee ohjelmistokehitystä myös alihankintatilanteessa, vaikka alihankkijan prosessit ja työkalut eivät olisikaan ostajalle näkyvissä.

Huoltovarmuuskriittisten järjestelmien toteutuksessa näiden vaatimusten toteutumisesta, esimerkiksi kehitys- ja tuotantoonvientityökalujen saatavuudesta, on huolehdittava myös jatkuvuussuunnittelumielessä.

Ohjelmistotuotantoprosessissa on oltava käytössä:

* **Tuotteen tehtävälista** (*product backlog*), joka on priorisoitu luettelo työtehtävistä, joita tuotteen (palvelun tai sovelluksen) toteuttamiseksi on tehtävä. Tehtävillä voi olla eri tasoja, kuten *epic*, kertomus eli *story* ja tehtävä eli *task*. Tässä dokumentissa "tehtävä" voi olla minkä tahansa tasoinen.
* **Tuoteomistajan** (*product owner*) **rooli**, joka on kuvattu aiemmin tämän ohjeen kohderyhmänä.
  + **Tietoturva- ja tietosuojariskien hyväksyminen** on vastuutettu tälle roolille kuten yllä olevassa vaatimusluokittelussa on kuvattu.
  + **Tietoturvatyön priorisointi** on oltava vastuutettu tälle roolille, eikä esimerkiksi erilliselle ohjausryhmälle.
* **Tiketöintiin** **pohjautuva työnohjausjärjestelmä, joka tukee luokituksia (*label* tai *tag*)**. Tämä tarkoittaa sitä, että tehtävälistalla olevien työtehtävien edistymistä voidaan erikseen ja yksittäin seurata, työlistalle ilmestyvät tietoturva- ja tietosuojatehtävät voidaan helposti huomata, ja työtehtävien teosta jää auditoitava jälki. Useimmiten tuotteen tehtävälista ja työnohjausjärjestelmä toimivat samassa työkalussa, mutta välttämättä näin ei ole.
* **Koodin versionhallintajärjestelmä, joka mahdollistaa koodin vertaiskatselmoinnin koordinoidun suorittamisen osana toteutustyötä**. Tällainen voi olla esimerkiksi nk. *pull request* -menettely.

Tehtävälistan ja työnohjausjärjestelmän on käytännössä tarpeen elää työkalussa, joka on standardoitu useamman projektin välille, koska esimerkiksi linkkien luominen eri projektien työtehtävien välillä on tärkeää. Tällainen työkalu on <organisaatio> käyttämä JIRA-järjestelmä.

## Vastuunjako

Ohjeen kohdeyleisö -kappaleessa luetelluilla rooleilla on tarkasti määritellyt vastuualueet. Vastuu seuraa rooleja eikä henkilöitä; henkilöllä voi olla useita rooleja ja toisaalta rooli voi olla jaettu useammalle henkilölle.

Mikäli rooli on jaettu useammalle henkilölle, näiden henkilöiden osalta on sovittava, kuka kantaa päävastuut seuraavassa luetelluista tietoturvavastuista. Mikäli päävastuullista henkilöä ei ole jaetun roolin tilanteessa määritelty, ohjelmistokehitysprosessi ei ole tämän tietoturvaohjeen mukainen.

Vastuutus on viestittävä kohdeyleisötahoille.

|  |  |
| --- | --- |
| Rooli | Vastuut (tarkemmin alla) |
| Hankepäällikkö | * Tietoturva- ja tietosuojatyön onnistumisen edellytyksistä huolehtiminen |
| Tuoteomistaja | * Vaatimuksista poikkeamisesta päättäminen * Tietoturvavaatimusten vienti tuotteen tehtävälistalle * Uuden ja muuttuvan toiminnallisuuden tietoturvavaikutusten arviointi * Uuden ja muuttuvan toiminnallisuuden tietosuojavaikutusten arviointi * Tietoturva- ja tietosuojatyön tuominen näkyväksi työnohjausjärjestelmässä |
| Arkkitehti | * Yleisten tietoturvaperiaatteiden noudattaminen suunnittelussa * Yleisen tietoturva-arkkitehtuurin rakentaminen ja ylläpito |
| Ohjelmistokehittäjä | * Tietoturvatyön tiketöinti ja siirto tuotteen tehtävälistalle * Tietoturvatyön suorittaminen |
| Tietoturva-asiantuntija | * Sovitun tietoturvatyön suorittaminen ja dokumentointi (siltä osin kun sovittu ohjelmistokehittäjien kanssa) |
| Käyttöpalvelun toimittaja | * Sovitun tietoturvatyön suorittaminen ja dokumentointi |

### Tuoteomistajan vastuut

#### Tuoteomistajat: Vaatimuksista poikkeamisesta päättäminen

Mikäli ohjelmistotuotantoprosessissa on tarve poiketa tietoturva- tai tietosuojavaatimuksista, tuoteomistajan tai tarvittaessa organisaatiossa hänen yläpuolellaan olevan henkilön on tehtävä päätös tästä aiheutuvan riskin hyväksynnästä. Päätös on dokumentoitava (esimerkiksi tietoturva-arkkitehtuuridokumentissa). Päätäntävalta on kuvattu tarkemmin kappaleessa Vaatimustenmukaisuus.

#### Tuoteomistajat: Tietoturvavaatimusten vienti tuotteen tehtävälistalle

Tuoteomistajan on identifioitava tietoturvavaatimukset, joiden lähteet on lueteltu kappaleessa Vaatimustenmukaisuus. Nämä vaatimukset on muutettava tuotteen tehtävälistalle tehtäviksi. Tehtävät voivat olla eri tasoisia (esimerkiksi kehitysaihio- (*epic-*) tai kertomustasoisia).

Laeista ja asetuksista kumpuavat vaatimukset voivat olla haastavia identifioitavia. Tämän vuoksi tuotantoonvientivaatimukset (Liite 1: Tuotantoon viennin vaatimukset) ja yleiset tietoturvavaatimukset (Liite 2: Tietoturvallisuuden yleisperiaatteet) on tehty kattamaan useimmat lakitekniset vaatimukset. On kuitenkin erityistilanteita, joissa ne eivät välttämättä riitä, jolloin tilanteen analysointi yhdessä lain tulkitsijan – esimerkiksi organisaation juristin – kanssa on tarpeen.

Joissakin tapauksissa pakollinen vaatimus ei välttämättä ole yksittäinen toiminnallisuus tai tehtävä, vaan esimerkiksi vaatimus tehdä ohjelmistokehitystä tietyllä tavalla, kuten esimerkiksi vaatimus tehdä jatkuvaa koodikatselmointia. Jatkuvia tai yhä uudelleen toistuvia tehtäviä ei voi järkevästi viedä tuotteen tehtävälistalle. Tällöin tuoteomistaja muuttaa sen prosessikehitystehtäväksi esimerkiksi "kehittäjät määrittelevät koodikatselmointiperiaatteet ja kouluttavat sen kaikille tiimiläisille" tai portfoliotyössä mukaan otetaan tarvittavia sidosryhmiä, jotka muutoin eivät olisi olleet mukana.

Käytännössä monien toiminnallisten vaatimusten vienti tehtävälistalle voidaan tehdä kertaluonteisesti merkittävän uuden kehityshankkeen alussa. Ylläpitovaiheessa tilanne voidaan katselmoida esimerkiksi vuosittain.

#### Tuoteomistajat: Uuden ja muuttuvan toiminnallisuuden tietoturvavaikutusten arviointi

Tuoteomistajan on tunnistettava sellainen uusi tai muuttunut toiminnallisuus, jolla voi olla tietoturva- tai tietosuojavaikutuksia, ja vaadittava toteutuksen hyväksyntäkriteereinä tietoturvariskien selvitys ("uhkamallinnus") sekä mahdollisesti soveltuvan tasoinen tietoturvatarkistus. Uhkamallinnuksen yleinen ohje on Liite 5: Uhkamallinnuksen toteutusohje.

Käytännössä tuoteomistajalla on käytössään lyhyt ja korkealla tasolla kirjoitettu tarkastuslista (Liite 3: Uuden tai muuttuneen toiminnallisuuden tietoturva- ja tietosuojatyön tarkastuslista), jonka perusteella riskialttiit toiminnot voidaan tunnistaa. Uhkamallinnustehtävä lisätään tuotteen tehtävälistalle.

Yleensä tuoteomistajan on helpointa tehdä tämä osana tuotteen tehtävälistan parannustyötä (*backlog maintenance, backlog grooming*).

#### Tuoteomistajat: Uuden ja muuttuvan toiminnallisuuden tietosuojavaikutusten arviointi

Vastaavasti kuin edellä tietoturvan osalta, myös tietosuojaan vaikuttava uusi tai muuttunut toiminnallisuus on tunnistettava. Tämä tehdään samalla tavalla kuin tietoturvan osalta, mutta tietoturvauhkamallinnuksen sijaan tai lisäksi tuoteomistaja luo tuotteen tehtävälistalle vaatimuksen tietosuojavaikutusten arvioinnista (DPIA, *data protection impact assessment*). Käsitteellisesti kyse on samantyyppisestä työstä. Ohje on dokumentissa Liite 6: Tietosuoja.

#### Tuoteomistajat: Tietoturva- ja tietosuojatyön tuominen näkyväksi työnohjausjärjestelmässä

Tuoteomistaja on vastuussa siitä, että tietoturvaan ja tietosuojaan liittyvät tehtävät on merkitty tikettipohjaisessa työnohjausjärjestelmässä erityisin luokituksin (*label*). Näin organisaation tietoturva- ja tietosuojaorganisaatiot voivat seurata näiden osa-alueiden etenemistä.

Luokituksissa käytettävät avainsanat on kuvattu dokumentissa Liite 3: Uuden tai muuttuneen toiminnallisuuden tietoturva- ja tietosuojatyön tarkastuslista.

Luokitukset on tarkoitettu ainoastaan auditoitavuus- ja seurantatarkoituksiin. Luokituksilla ei ole vaikutusta priorisointiin eikä työtehtävien elinkaaren vaiheisiin. Niitä ei myöskään käytetä ryhmittelemään työnohjaustikettejä.

#### Tuoteomistajat: Tietoturva- ja tietosuojatyön priorisointi suhteessa muuhun työhön

Tuoteomistaja määrittelee tuotteen tehtävälistalla olevien tehtävien keskinäisen prioriteetin. Käytännössä tämä tarkoittaa, että tuoteomistaja määrittelee tehtäviin käytettävissä olevan ajan.

Koska tietoturva- ja tietosuojatyö on näkyvissä tuotteen tehtävälistalla, tämä tarkoittaa sitä, että tuoteomistaja päättää myös tämän työn prioriteeteista ja sitä kautta hyväksyy jäännösriskin, mikäli tietoturvatehtävät väistyvät muiden (toteutus)tehtävien tieltä.

### Arkkitehtien vastuut

#### Arkkitehdit: Yleisten tietoturvaperiaatteiden noudattamisen suunnittelussa

Arkkitehti on vastuussa siitä, että yleisiä arkkitehtuuriperiaatteita noudatetaan ohjelmiston suunnittelutoiminnassa. Arkkitehtuuriperiaatteet on kuvattu kappaleessa Tietoturvallisen arkkitehtuurin ja suunnittelun yleisperiaatteet.

#### Arkkitehdit: Yleisen tietoturva-arkkitehtuurin määrittely ja ylläpito

Arkkitehtiroolin vastuulla on myös yleisen tietoturva-arkkitehtuurin ylläpito yli kaikkien toteutustiimien. Tähän liittyvät muun muassa tiimien väliset tietoturvaolettamat ja viestiminen käyttöpalveluiden tuottajien kanssa siitä, miten käyttöpalvelut vaikuttavat sovellustason tietoturva-arkkitehtuuriin.

### Ohjelmistokehittäjien vastuut

#### Ohjelmistokehittäjät: Tietoturvatyön tiketöinti ja siirto tuotteen tehtävälistalle

Jotta tietoturvatyö tulisi näkyväksi ja sille varattaisiin riittävästi aikaa, ohjelmistokehittäjien tärkeimpiä vastuita on varmistua siitä, että tietoturvatyö näkyy työnhallintajärjestelmässä. Kun tietoturvatyö on kirjattu tehtäväksi tikettiin, esimerkiksi työjaksojen sisältöä määriteltäessä on selkeämpää, paljonko aikaa tietoturvatyö vie.

Mikäli ohjelmistokehittäjä havaitsee uutta tietoturvatyötä, jota pitäisi tehdä, heidän vastuullaan on lisätä se tuotteen tehtävälistalle. Tyypillisimmin tämä tapahtuu uhkamallinnuksen seurauksena, jolloin löydettyjen riskien hallitsemiseksi luodaan uusia tehtäviä.

Näkyvyyden varmistamiseksi ohjelmistokehittäjien tulee myös luokitella tiketit kuten Liite 3: Uuden tai muuttuneen toiminnallisuuden tietoturva- ja tietosuojatyön tarkastuslista kuvaa.

#### Ohjelmistokehittäjät: Tietoturvatyön suorittaminen ja dokumentointi

Vastuu tietoturvatyön suorittamisesta on ohjelmistokehittäjillä.

On selvää, että kaikki ohjelmistokehitystiimit eivät aina pysty suorittamaan kaikkia näitä tietoturvatehtäviä aika- tai kompetenssirajoitteista johtuen. Vastuu ei kuitenkaan tällöin siirry, vaan rajoite on ratkaistava joko koulutuksella, priorisoinnilla tai tiimin ulkopuolisiin resursseihin tukeutumalla.

Joissakin tapauksissa ympäröivä organisaatio voi luoda palvelun, jonka käyttö on kustannustehokkaampaa kuin tiimin sisäisten resurssien käyttö. Esimerkiksi tutkivan tietoturvatestauksen asiantuntijaresurssit voi olla tehokkaampaa hankkia keskitetysti koko organisaation käyttöön.

Toteutusaikainen tietoturvatyö koostuu yleisimmin seuraavista tehtävistä:

* Kehitysympäristöjen ja -työkalujen tietoturvasta vastaaminen (ks. kappale Kehitysympäristöjen ja tuotantoon viennin yleisperiaatteet)
* Uhkamallinnus (ks. Liite 5: Uhkamallinnuksen toteutusohje ja kappale Tietoturvallisen arkkitehtuurin ja suunnittelun yleisperiaatteet)
* Tietosuojavaikutusten arviointi (ks. Liite 6: Tietosuoja)
* Koodikatselmointi tai staattinen analyysi (eli koneellinen koodikatselmointi)
* Automaattisten testitapausten toteutus (ks. kappale Tietoturvatestauksen ja teknisen tarkastuksen yleisperiaatteet)
* Tutkiva tietoturvatestaus (ks. kappale Tietoturvatestauksen ja teknisen tarkastuksen yleisperiaatteet)
* Ohjelmistoriippuvuuksien haavoittuvuusseuranta ja paikkaus (ks. kappale Tietoturvallisen arkkitehtuurin ja suunnittelun yleisperiaatteet)
* Tuotantoon viennin tietoturvadokumentaation ylläpito (ks. kappale Kehitysympäristöjen ja tuotantoon viennin yleisperiaatteet)

Tehdyn tietoturvatyön dokumentointi tehdään ensisijaisesti tiketöintijärjestelmään ja versiohallittuihin dokumentteihin (esimerkiksi wiki).

Mikäli ohjelmistokehittäjät tukeutuvat joiltakin osin käyttöpalvelutoimittajan komponentteihin (esimerkiksi käyttöjärjestelmään), ohjelmistokehittäjien vastuulla on määritellä vastuunjaon raja käyttöpalvelutoimittajaan nähden. Käyttöpalvelutoimittajan vastuualueista on sovittava sopimuksellisesti.

### Tietoturva-asiantuntijan vastuut

#### Tietoturva-asiantuntijat: Sovitun tietoturvatyön suorittaminen ja dokumentointi

Tietoturva-asiantuntijan vastuulla on häneltä pyydetyn tietoturvatyön toteuttaminen riskilähtöisesti ja parhaan asiantuntemuksensa mukaisesti. Hänen on myös dokumentoitava tietoturvatyön tulokset. Dokumentointi tehdään ensisijaisesti tiketöintijärjestelmään ja versiohallittuihin dokumentteihin (esimerkiksi wiki). Tietoturvatestaukseen ja -tarkastukseen liittyvät vaatimukset on lueteltu kappaleessa Tietoturvatestauksen ja teknisen tarkastuksen yleisperiaatteet.

### Käyttöpalvelun toimittajan vastuut

#### Käyttöpalvelun toimittajat: Sovitun tietoturvatyön suorittaminen ja dokumentointi

Käyttöpalvelun toimittajilta odotetaan, että se suorittaa käyttöpalveluja hankkivan organisaation kanssa sovitun tietoturvatyön. Viitteellisesti nämä kattavat kappaleissa Auditoitavuuden yleisperiaatteet ja Käyttöpalveluympäristöt ja operationaaliset yleisperiaatteet luetellut asiat.

Muut käyttöpalvelutoimittajan osalle siirretyt vastuut on määriteltävä sopimuksellisesti.

## Erityishuomioita tiettyihin ohjelmistokehitys- ja kulttuurisiin malleihin

Ketterää ja *lean*-ohjelmistokehitystä voidaan tehdä useilla eri malleilla. Ohessa on lyhyesti kuvattu, miten yllä mainitut vastuut voidaan ottaa mukaan kussakin mallissa.

### Scrum

Tietoturvatyötehtävät sisällytetään Scrum-työjaksoihin (*sprint*) aivan kuin ne olisivat toiminnallisten vaatimusten toteutusta. Esimerkiksi uhkamallinnuksen ja tutkivan tietoturvatestauksen tarve tuodaan esille tehtävänä tai tehtävään sidottuna hyväksyntäkriteerinä. Yleisiä kaikkiin kehitystehtäviin liittyviä *Definition of Done* -kriteereitä ei käytetä tietoturvatyön ohjaamiseen, koska tällöin niille ei erikseen varata aikaa. Ohjelmistokehittäjät saavat toki omatoimisesti tehdä tietoturvaan liittyviä Definition of Done -kriteerejä oman laadunvalvontansa tueksi.

Jos tietoturvatyö luodaan tehtävälistalle erillisenä tehtävänä, se merkitään yleensä riippuvuudeksi toiminnallisuuden toteutuksen valmistumiselle. Tämä tehdään tyypillisessä tiketointijärjestelmässä linkittämällä tehtävät toisiinsa.

Koska Scrum-työjakson suunnitteluvaiheessa (*sprint planning*) on yleensä sitouduttava tiettyyn työmäärään, esimerkiksi uhkamallinnustyö kannattaa yleensä tehdä aiemmassa työjaksossa kuin missä varsinainen toiminnallisuus toteutetaan. Uhkamallinnus saattaa muuttaa työmääräarvioita voimakkaastikin.

### Kanban

Tietoturvatyötehtävien hallinta noudattelee Scrumin mallia (yllä). Koska kanban-menetelmässä ei ole työjaksoon perustuvaa työmäärän (*Work in Progress*) rajoitusta, esimerkiksi uhkamallinnustarve on luonnollisemmin toteutettavissa toiminnallisuuden hyväksyntäkriteerinä sen sijaan, että se olisi erillinen työtehtävä.

### DevOps

Kulttuurisena käsitteenä *DevOps* eli autonomisten kehitys- ja tuotantoaikaisista toimenpiteistä vastaavien tiimien luominen on tietoturvan kannalta lähes aina positiivinen asia. DevOps-kypsyyden parantaminen on tämän vuoksi suositeltavaa.

Käytönaikaisten toimenpiteiden (Ops) osalta vastuunjaon sopiminen käyttöpalvelutoimittajan kanssa on erityisen tärkeää.

### Automaatio, jatkuva integraatio ja toimitus

Ne tietoturvatehtävät, jotka ovat luonteeltaan nopeasti toistuvia tai jatkuvia, kuten staattinen analyysi, automaattiset haavoittuvuusskannaukset, riippuvuuksien haavoittuvuusseuranta ja käytönaikainen tietoturvaan liittyvä monitorointi, pitäisi mahdollisuuksien mukaan automatisoida.

Kun automatiikka on riittävän kypsää, jatkuvan integraation (*continuous integration* tai *CI*) järjestelmään voidaan rakentaa logiikkaa, joka estää tuotantoonviennin, jos automatiikka huomaa mahdollisen tietoturvariskin.

Tietoturvatehtävien automatisointi on itsessään työtä, ja sikäli kun ohjelmistokehittäjät toteuttavat automatisoinnin, tämäkin kehitystyö ohjataan tuotteen tehtävälistan kautta. Tällöin sen kustannukset ja priorisointi tulevat käsitellyksi.

# Ohjelmistoturvallisuuden seurannan ja ulkoisen tuen malli

## Jatkuva tietoturvatyön näkyvyys ja työn jälkikäteinen auditoitavuus

Jotta tietoturvatyön edistymistä voidaan seurata reaaliaikaisesti, tietoturvatyötä ohjaavat tehtävät luokitellaan ja ne linkitetään siihen toiminnallisuuteen, mihin ne liittyvät. Organisaation tietoturva- ja tietosuojaosastot voivat tällöin identifioida ne projektit, joissa luokittelua ei tapahdu, ja reagoida asiaan aikaisessa vaiheessa. Tietoturva-aktiviteettien toteutumisen näkyvyys kokonaisketterän mallin portfoliotasolle voidaan tehdä samalla tavalla.

Luokittelu on kuvattu kappaleessa Liite 3: Uuden tai muuttuneen toiminnallisuuden tietoturva- ja tietosuojatyön tarkastuslista. Linkitys tehdään käytetyn tiketöintijärjestelmän linkkejä käyttäen.

Jos ohjelmistoja teetetään ulkoisena hankintana, jossa kehitystä ei tehdä ostajan työnohjauksen alaisuudessa, ostaja voi halutessaan pyytää näkymää työnohjausjärjestelmään ja vaatia sopimuksellisesti vastaavan luokittelun. Näin ostaja voi varmistua siitä, että tietoturva- ja tietosuojatyötä tehdään jatkuvasti koko kehitystyön ajan.

## Tukipalveluiden työnohjausmalli

Ohjelmistokehittäjille, arkkitehdeille ja tuoteomistajille voidaan tarjota erilaisia tietoturvan ja tietosuojan tukipalveluita. Tyypillisin näistä on tutkiva tietoturvatestaus, joka tehdään usein ohjelmistotiimin ulkoisia resursseja käyttäen.

Organisaatio luo tukipalveluiden toimittamista varten oman erillisen tehtävälistansa (*backlog*), johon kuka tahansa organisaation kehitystehtävissä toimiva henkilö voi lisätä tarvitsemansa tukipalvelutarpeen kuvauksen.

Mikäli tehtävälistaa ylläpidetään samassa tiketointijärjestelmässä kuin tuotteen tehtävälistaa, tukipalvelupyyntö voidaan linkittää ristiin näiden tehtävälistojen välillä. Näin tukipalveluiden tarve tehdään näkyväksi.

Tuotekehitystiimiä ympäröivä organisaatio tuottaa tukipalveluita keskitetysti tämän tehtävälistan pohjalta ja määrittelee prioriteettijärjestyksen toimitettaville tukipalveluille.

# Liite 1: Tuotantoon viennin vaatimukset

Seuraavat hyväksyntäkriteerit on täytettävä ennen kuin kehitettävän toiminnallisuuden saa viedä tuotantoon. *Näistä kriteereistä poikkeaminen on sallittua vain <ORGANISAATIO>:ssa tuoteomistajan yläpuolella olevan henkilön erillisellä päätöksellä. Päätös on dokumentoitava.*

Tuotantoon viennin vaatimukset otetaan kokonaisketterässä kehityksessä mukaan suunnittelukanbanissa.

Jotta jatkuva tuotantoon vienti on mahdollista, näiden vaatimusten on oltava sellaisia, jotka voidaan joko

1. täyttää suunnittelu- tai kehitysaikaisesti ennen kuin koodi menee automaattiseen integraatioon tai tuotantoon vientiin; tai
2. voidaan toteuttaa kertaluontoisesti vain yhden kerran koko järjestelmän elinaikana, tai
3. voidaan toteuttaa automaattisin työkaluin osana tuotantoon vientiä.

Dokumentointi- ja auditoitavuusvaatimukset tehdään ensisijaisesti joko työnohjauksessa käytetyn tiketin sisällössä tai tilassa, versiohallitussa wiki-sivussa tai automaattisen työkalun tuottamassa tiedossa, joka tallennetaan. Erillisiä vaatimustenmukaisuusdokumentteja kirjoitetaan vain, jos niiden tuottamiselle on erillinen tehtävä tehtävälistalla.

* Ohjelmiston yleisarkkitehtuurille ja pääasiallisille toiminnallisuuksille on toteutettu uhkamallinnus suunnitellussa laajuudessa ja mahdollinen jäännösriski on dokumentoidusti hyväksytty. Uhkamallinnuksen toteutus ja sen tulokset ovat auditoitavissa. Tämä vaatimus voi täyttyä yhdistelemällä kertaluontoisia uhkamallinnuksia ja tämän prosessin kuvaamia toiminnallisuuskohtaisia pienemmistä uhkamallinnuksista.
* Ohjelmistolle on suoritettu tietosuojavaikutusten arviointi, mikäli se on lakisääteisesti tarpeellista[[7]](#footnote-7). Suoritettu tietosuojavaikutusten arviointi ja sen tulokset ovat auditoitavissa.
* Mikäli ohjelmisto käsittelee henkilötietoja, ohjelmistosta on olemassa ajantasainen kuvaus henkilötietovirroista ja niiden tallennuspaikoista, henkilötietotyyppikohtaisesti eriteltyinä. Tämä vaatimus on erillinen tietosuojavaikutusten arvioinnista ja sen tulee toteutua kaikille henkilötietoja käsitteleville ohjelmistoille.
* Ohjelmiston tietoturvatestaustarpeet on määritelty ja ne on dokumentoitu (esimerkiksi tiketteinä tuotteen tehtävälistalle).
* Ohjelmistosta on olemassa ajantasainen tietoturva-arkkitehtuurikuvaus, joka täyttää minimivaatimukseltaan kappaleen Liite 4: Tietoturva-arkkitehtuurin dokumentoinnin vähimmäisvaatimukset vaatimukset.
* Ohjelmiston ajallisesti rajoitetut komponentit (kuten varmenteet) on dokumentoitu, ja niiden hallinta on siirretty mahdollisen tuotantoa hoitavan tahon haltuun.
* Pääsy tuotannon ajoympäristöjen ylläpitoon (esimerkiksi virtuaalikoneet tai orkestrointijärjestelmät) on rajattu vain tuotantoa hoitaville tahoille.
* Tuotannon ajoympäristöjen (esimerkiksi virtuaalikoneet, konttijärjestelmät ja -varastot sekä orkestrointijärjestelmät) omistajuus ja päivitysperiaatteet on vastuutettu ja kuvattu selkeästi.
* Tuotannon ajoympäristöt täyttävät niille erikseen asetetut kovennusvaatimukset[[8]](#footnote-8). Käyttöpalvelutoimittajan vastuulla.
* Ohjelmiston tuottamat lokit siirretään keskitettyyn lokijärjestelmään.

# Liite 2: Tietoturvallisuuden yleisperiaatteet

## Tietoturvallisen arkkitehtuurin ja suunnittelun yleisperiaatteet

|  |  |
| --- | --- |
| Periaate | Kuvaus |
| Järjestelmän ei-julkinen tieto on tunnistettu ja sen käsittelyperiaatteet on määritelty | Tunnistetaan, mitä ei-julkista tietoa järjestelmässä käsitellään, ja sille määritellään käsittelyn tietoturva- ja tietosuojavaatimukset. Vaatimusten tulisi päätyä tuotteen tehtävälistalle tulkittuna arkkitehtuurin toteutusvaatimuksiksi, ja tieto välitetään myös ohjelmistokehittäjille, jotta vaatimukset voidaan ottaa huomioon uhkamallinnuksessa. |
| Palvelun keskeiset käyttötilanteet on tunnistettu | Tunnistetaan palvelun käyttötapaukset (*use case*) ja erityisesti tietosuojamielessä käyttökokemus (*user experience*). Erityisen kiinnostavia ovat poikkihallinnolliset käyttötilanteet, joissa tietoa siirretään organisaatioiden välillä, sekä ylläpito- ja hallintakäyttötilanteet. Tämäkin tieto välitetään ohjelmistokehittäjille, jotta vaatimukset voidaan ottaa huomioon uhkamallinnuksessa. |
| Arkkitehtuurin tietoturva ja tietosuoja on dokumentoitu | Ks. Liite 4: Tietoturva-arkkitehtuurin dokumentoinnin vähimmäisvaatimukset |
| Arkkitehtuuri on modulaarinen | Arkkitehtuuri rakentuu määritellyin rajapinnoin eriytetyistä komponenteista tai palveluista, jotka toteuttavat itsenäisesti tietyn toiminnallisuuden tai tarjoavat pääsyn palvelun tarvitsemaan tietoon. Mikropalvelut (*microservices*) ovat yksi tapa tuottaa tämänkaltainen arkkitehtuuri. |
| Tietojen käsittely ja rajapintojen syntaktinen monimutkaisuus on minimoitava | Palvelun tulee toteuttaa vain palvelun kannalta välttämättömät toiminnot. Tämä pätee sekä tiedon käsittelyyn että rajapintojen toiminnallisuuteen ja ilmaisuvoimaan.  Henkilötietoja saa käsitellä vain niin siinä laajuudessa, kuin palvelun tekniseen toteuttamiseen vaaditaan.  Tietojen käsittely toteutetaan kokonaan käyttömuistissa eikä niitä kirjoiteta levylle, mikäli mahdollista. |
| Arkkitehtuuriin on luotava selvät luottamusalueet, jotka on dokumentoitu tietoturva-arkkitehtuurissa | Jokaisen komponentin osalta on määriteltävä, mihin muihin komponentteihin se luottaa, ja tämä määrittelee luottamusalueet. Eri luottamusalueiden välille on suunniteltava verkko- ja ajoympäristöjen riittävä erottaminen ja tietovirtojen sisällön oikeellisuuden tarkistus ja todennus.  Komponentit, joilla on rajapinta loppukäyttäjään tai kolmannen osapuolen palveluun eriytetään omaan luottamusalueeseensa.  Jos luottamusalueilla on keskinäinen hierarkia (esimerkiksi suojaustaso), suojaustasojen väliset eheys- ja luottamuksellisuusvaatimukset on määriteltävä. |
| Tietovarannot on eriytettävä luottamusalueiden välillä | Kaikki tietovirrat luottamusalueiden välillä on toteutettava rajapinnoilla. Suorat tietovarantopääsyt luottamusalueelta toisille (esimerkiksi vapaamuotoiset tietokantakyselyt tai tiedostojärjestelmän luku) on arkkitehtuurin tasolla estettävä.  Luottamusalueella, jolla on suora yhteys loppukäyttäjään tai kolmannen osapuolen palveluun ei saa sijaita tietovarantoja, vaan tiedot on noudettava tarvittaessa rajapinnan kautta. |
| Tietovirtojen tietoturva- ja tietosuojavastuut on selkeästi määritelty | Vastuu tietovirtojen sisällön oikeellisuuden tarkistamisesta ja niihin luottamisesta on aina vastaanottajalla.  Tietosuojan osalta vastuu siitä, että tietoja ei luovuteta toiselle komponentille lain vastaisesti, on aina lähettäjällä. |
| Järjestelmän saatavuustavoitteiden on oltava selkeitä vaatimuksia | Saatavuustavoitteet on määriteltävä ja niille on löydettävä tekninen ratkaisu esimerkiksi hajauttamalla ja automaattisella skaalautumisella. Erityishuomiota on kiinnitettävä vanhojen järjestelmien integraatiopisteisiin ja komponentteihin, joita voi käsitteellisesti olla vain yksi kappale. |
| Valmiita toteutuksia ja ohjelmistoriippuvuuksia käytettäessä tietoturvavastuu on määriteltävä | Mikäli käytetään jonkin muun tahon kirjoittamia ohjelmistoja, tietoturvavastuu kuuluu sille komponentille, joka valmista toteutusta käyttää. Vastuun voi siirtää sopimuksellisesti, mikäli käyttö perustuu sopimukseen. Tyypillisessä avoimen lähdekoodin tapauksessa käyttö perustuu vain rajoitettuun lisenssiin, jolloin tietoturvavastuuta ei voida siirtää. |
| Kolmannen osapuolen komponenttien laatu on selvitettävä ja seurattava | Ohjelmistokehityksessä käytettävien, ulkopuolelta hankittavien (esim. avoin lähdekoodi tai valmis ostettu komponentti) haavoittuvuustilannetta on seurattava jatkuvasti. Tämä tulisi tehdä automatisoidulla järjestelyllä. Käyttöön otettaessa ulkopuolisen komponentin tausta ja ylläpitotilanne on selvitettävä ja siihen liittyvät riskit on hallittava. |

## Tietoturvallisten palvelurajapintojen ja käyttöliittymien yleisperiaatteet

|  |  |
| --- | --- |
| Periaate | Kuvaus |
| Käyttöliittymät ja rajapinnat eriytetään käyttäjätyyppien tarpeiden perusteella | Käyttäjätyypit jaotellaan käyttötapaustensa mukaan, esimerkiksi viranomaiset, kuluttajat/kansalaiset/asiakkaat, ylläpito, yritykset ja avoimen datan rajapinnat. Käyttötapausten salliessa rajapinnat eriytetään siten, että samasta rajapinnasta ei saa erityyppistä palvelua pelkästään esimerkiksi todennetun käyttäjätyypin perusteella, vaan esimerkiksi rajapinnan osoite tai polku on erilainen ylläpito- ja loppukäyttäjätoiminnallisuudessa. Käyttöliittymät rajataan toiminnallisesti sisältämään vain käyttäjäryhmän tarvitsema minimitoiminnallisuus. |
| Käyttövaltuudet hallitaan keskitetysti ja ne sovitetaan kulloiseenkin tarpeeseen | Käyttövaltuudet hallitaan keskitetystä paikasta, jotta vältetään hajautetun valtuuksien hallinnan mahdolliset ristiriitaisuudet ja katvealueet. Käyttäjillä on käyttökokemukseen ja -tarpeeseen sovitetut käyttövaltuustasot ja tarvittaessa mahdollisuus vaihtaa niitä, jotta kaikkia toimintoja ei tarvitse tehdä aina korkeimmalla tasolla. |
| Henkilötietojen käyttöön liittyvät tiedot ja valinnat tuodaan käyttäjille oikea-aikaisesti ja oikeassa viitekehyksessä | Henkilötietojen käsittelyyn (erityisesti keräämiseen ja siirtoon) liittyvät tiedotteet ja luvat näytetään käyttäjille käyttökokemuksen perusteella parhaiten soveltuvassa tilanteessa ja ilmaisu on tehty käyttäjän ymmärrystason huomioon ottaen. Henkilötietojen keräyksestä informoidaan asteittain tarkentuvalla tiedolla sen mukaan, kuinka tarkkoja tietoja käyttäjä tietojen käytöstä haluaa (nk. *layered notice*). |
| Suostumuksen kysymisen käyttöliittymässä on oltava auditoitavissa | Suostumusta käyttäjältä kysyttäessä sen perumisen on oltava yhtä helppoa kuin sen antaminen. Suostumuksen antamisesta ja perumisesta on jäätävä aikaleimattu merkintä. |
| Käyttäjä saa tehdä tietoturvaan vaikuttavan päätöksen vain, jos se on tarpeellista | Käyttäjä saa tehdä tietoturvaan vaikuttavan päätöksen (esimerkiksi poikkeustilanteen hyväksynnän) vain, jos käyttäjä voi todellisuudessa tuoda päätökseen lisätietoa, jota järjestelmällä itsellään ei ole. Käyttöliittymän on uskottavasti pystyttävä takaamaan, että käyttäjä ymmärtää päätöksensä seuraukset. Muussa tapauksessa järjestelmän tulee automaattisesti tehdä turvallinen päätös käyttäjän puolesta, eikä käyttäjä saa pystyä ohittamaan päätöstä. |
| Rajapinnat toteutetaan kieliopiltaan mahdollisimman yksinkertaisiksi | Rajapintojen hyväksymien syötteiden kieliopin on oltava mahdollisimman yksinkertainen, jotta syötteen oikeellisuuden tarkastaminen on helppoa tai edes mahdollista. |
| Käyttöliittymät ja rajapinnat hylkäävät väärät syötteet kokonaisuudessaan | Mikäli rajapintaan tulee väärä syöte, koko syöte hylätään mahdollisimman varhaisessa vaiheessa. Syötettä ei yritetä tulkita, mikäli se on osittainkin virheellinen. |
| Käyttöliittymät ja rajapinnat piilottavat tietovarantojen sisäisen toiminnallisuuden | Tietovarantojen sisäiseen toiminnallisuuteen liittyvät asiat, kuten tietokantarivien järjestysnumerot tai palvelimen tiedostojärjestelmän nimet eivät ole näkyvissä rajapinnan tai käyttöliittymän läpi. |
| Jokainen rajapinta vaatii todennuksen ja valtuutuksen riippumatta verkkoympäristöstä | Jokaisen tarjottavan rajapinnan on todennettava kutsujan ja tarkistettava sen valtuutus. Rajapintoja ei saa avata edes luotettuun verkkoon niin, että kuka tahansa verkon alueelta pystyy kutsumaan rajapintaa tunnistautumatta. Tästä ainoana poikkeuksena ovat tilanteet, joissa rajapintojen välillä on pisteestä toiseen toteutettu VPN-yhteys tai jos rajapinta on julkisen tiedon haku- tai monitorointirajapinta. |
| Rajapinnoissa suositaan standardeja | Rajapintojen toteutuksissa suositaan protokollia ja tietojen esitys- ja siirtomuotoja, jotka ovat laajalti hyväksyttyjä ja käytössä.  Organisaation eri osissa standardoidaan ja yhtenäistetään samoihin tarkoituksiin tarkoitetut rajapinnat ja näiden toteutukset. |
| Rajapintojen todennuksessa ja valtuutuksessa käytetään standardoituja protokollia ja algoritmeja | Todennus ja valtuutus perustuvat kryptografiaan. Kryptografisten protokollien ja algoritmien suunnittelu itse ei ole sallittua. Yleisesti hyväksyttyjen todennus- ja valtuutusprotokollien toteutus itse on sallittua, mutta niille on tehtävä tarkka suunnittelu- ja toteutuskatselmointi sekä testaus. |

## Tietoturvatestauksen ja teknisen tarkastuksen yleisperiaatteet

|  |  |
| --- | --- |
| Periaate | Kuvaus |
| Tietoturvatestauksen ja -tarkastuksen kohteet määritellään hyökkäyspinnan ja mahdollisen vaikutuksen mukaan | Tietoturvatarkastukset ja niiden tyyppi kohdennetaan arvioimalla eri toiminnallisuudelle suhteellinen riski hyökkäyspinnan (riskin todennäköisyys) ja toiminnallisuuden tärkeyden tai käsiteltyjen tietojen (riskin vaikutus) mukaan. Suositeltavaa on tehdä tämä uhkamallinnuksen yhteydessä. Tietoturvatestaus jakaantuu pääasiassa automaattisiin testitapauksiin ja tutkivaan testaukseen; tutkiva testaus kohdistetaan ensisijaisesti niihin kohteisiin, jotka ovat suhteessa riskialttiimpia. |
| Tietoturvatarkastuksen kattavuus ja tyyppi määritellään hyökkäyspinnan ja mahdollisen vaikutuksen mukaan | Tietoturvatarkastuksen tarvittava taso voidaan määritellä OWASP ASVS -dokumentin tasojen mukaan (1-3), mikäli se on kohteeseen sovellettavissa. OWASP ASVS -dokumenttia voi käyttää myös testauksen tarkastuslistana ostettaessa web-sovelluksen tietoturvatarkastusta. Joissakin tapauksissa tarkastuksen tyypin on mahdollisesti oltava esimerkiksi Viestintäviraston akkredointi. |
| Tietoturvatestaustarpeet ja tekniset tarkastukset merkitään tehtävinä tuotteen tehtävälistalle | Automaattisen tietoturvatestauksen kehitystehtävät merkitään tuotteen tehtävälistalle kehittäjille tarkoitettuina työtehtävinä. Tekniset tarkastukset, vaikka ne tekisikin ohjelmistokehitystiimin ulkopuolinen taho, merkitään myös omina tehtävinään tuotteen tehtävälistalle, jotta niiden tilasta jää jälki. Ulkoisen testauksen työlistatehtävien omistaja on oletusarvoisesti tuoteomistaja, ellei niitä erikseen delegoida. |
| Tietoturvatestauksen ja teknisen tarkastuksen löydökset merkitään tuotteen tehtävälistalle | Kaikki tietoturvatestauksesta ja tarkastuksesta kumpuavat löydökset lisätään tuotteen tehtävälistalle niin, että tehtävänä on joko korjata ongelma, tuottaa kompensoiva kontrolli tai hyväksyä riski. |
| Tutkivasta tietoturvatestauksesta ylläpidetään vallitsevan tilanteen kuvausta | Tutkiva tietoturvatestaus tuottaa tuloksenaan löydösten lisäksi tietoturvan vallitsevan tilanteen kuvauksen, jota ylläpidetään kohdejärjestelmäkohtaisesti versioituna dokumenttina. Tämän dokumentin päivitys vastuutetaan tutkivan tietoturvatestauksen suorittajalle. |

## Auditoitavuuden yleisperiaatteet

|  |  |
| --- | --- |
| Periaate | Kuvaus |
| Järjestelmien tulee tuottaa riittävästi lokitietoa oleellisista tapahtumista | Osana uhkamallinnusta kartoitetaan ne toiminnat, mistä lokitietoa on kerättävä, sekä mitä tietoa on kerättävä. Useimmiten lokitiedon keräys tehdään rajapintakutsun yhteydessä.  Erityisesti henkilötietojen lukemisesta ja muuttamisesta on aina tehtävä lokimerkintä, jonka perusteella voidaan selvittää henkilö, joka on suorittanut lukemisen tai tehnyt muutoksen.  Sisään- ja uloskirjautumisista sekä sovelluksen että alustajärjestelmien tasolla, käyttöoikeustason muutoksista, asetustiedostojen muutoksista ja luvusta sekä ulkoisesta syystä aiheutuneista virhetilanteista on aina tehtävä lokimerkintä. |
| Henkilötietojen tallennusta lokijärjestelmään on vältettävä | Lokijärjestelmään ei saa tallentaa henkilötietoja, ellei sitä voida teknisesti muutoin toteuttaa. |
| Eri järjestelmien lokien on oltava korreloitavissa keskenään | Kaikki lokitallennus synkronoidaan koordinoituun yleisaikaan (UTC) luotetun aikapalvelimen avulla ja jokainen lokimerkintä varustetaan aikaleimalla, jonka tarkkuus on mahdollisimman suuri mutta vähintään sekuntitasolla.  On lisäksi suositeltavaa, että järjestelmän ulkopuolelta tuleviin pyyntöihin sidotaan satunnainen pyyntötunniste (*request ID*), joka välitetään taustajärjestelmältä toiselle ja kirjataan lokiin. Näin lokeista voidaan nähdä, mitkä toimenpiteet liittyivät mihinkin ulkopuolelta tulleeseen pyyntöön ja miten pyynnöt ketjuuntuvat (mikro)palveluarkkitehtuurissa. |
| Rajapintoihin tulevien pyyntöjen lähdeosoite on kirjattava lokiin | Rajapintoihin tulevien pyyntöjen lähdeosoite (IP-osoite) on kirjattava lokiin, myös mikäli rajapinta sijaitsee luotetussa verkossa. Luotettujen verkkojen IP-osoitteiden varaamisesta käyttöön on pidettävä loki, josta käy ilmi IP-osoitteiden ja sitä käyttäneiden laitteiden laitteisto-osoitteiden vastaavuus. Julkisesta Internetistä tulevien pyyntöjen lähde-IP-osoite on kirjattava lokiin sillä palvelimella, joka terminoi ulkoverkkoyhteyden ennen verkko-osoitemuunnosta. |
| Auditointilokimerkinnät on siirrettävä säännöllisesti pois niitä luovilta palvelimilta | Riippumatta lokienhallinnan ja -analyysin ympäristöstä, lokit tulee siirtää säännöölisesti ja mielellään reaaliaikaisesti pois kohteesta, jota lokitetaan. Tällä varmistetaan lokien luotettavuus, mikäli kohteen eheys vaarantuu. Siirrosta riippumatta lokeja säilytetään myös lähdepalvelimella 14 päivän ajan lokieheyden varmistamiseksi. |
| Lokimerkinnät on luotava standardoidulla tavalla | Organisaation omissa sovelluksissa lokimerkinnät on tehtävä JSON-muotoisina, jotta niiden analyysi ja jäsennys on helposti tehtävissä. Valmisohjelmistojen osalta on pyrittävä yleisesti käytössä olevien konventioiden seuraamiseen. |
| Monitorointirajapinnan toteutus saatavuusongelmien havaitsemiseen | Järjestelmän osien on toteutettava sisäinen monitorointirajapinta tai monitorointi voidaan suorittaa lokeja seuraamalla.  Uusissa monitorointirajapinnoissa tulisi pyrkiä julkaisemaan onnistuneiden pyyntöjen määrä aikayksikköä kohti, epäonnistuneiden pyyntöjen määrä aikayksikössä ja pyynnön käsittelyaika sekunneissa (pienin, isoin, keskimääräinen, mediaani) aikayksikössä (eli nk. *RED­*-tiedot: *rate, errors, duration*). |

## Kehitysympäristöjen ja tuotantoon viennin yleisperiaatteet

|  |  |
| --- | --- |
| Periaate | Kuvaus |
| Ohjelmistokehitykseen käytettävät tietokoneet ja muut päätelaitteet on suojattava | Päätelaitteet, joilla otetaan yhteyksiä ohjelmistokehityksen palvelimiin tai joilla tehdään ylläpidollisia toimia, on tuotava organisaation tietotekniikkaylläpidon määrittelemien tietoturvavaatimusten piiriin (vakiointi, päivitykset, haittaohjelmilta suojautuminen, jne).  Tämä vaatimus ei päde, mikäli päätelaitteella tehdään vain testausta julkisiksi tarkoitettuja käyttöliittymiä tai rajapintoja vasten. |
| Ohjelmistokehitykseen käytettävien palvelinten on oltava lokienhallinnan piirissä | Ohjelmistokehityksessä käytettävät palvelimet, kuten versionhallinta-, integrointi- ja muut palvelimet, on tuotava lokienhallinnan piiriin. Lokienhallinnan vaatimukset sovelletaan kappaleen Auditoitavuuden yleisperiaatteet perusteella. |
| Kehitysympäristöt ja työkalut on suojattava ja niiden saatavuus on taattava | Kehitysympäristöjen ja -työkalujen tietoturvavastuu kuuluu oletusarvoisesti ohjelmistokehittäjille. Vastuu voidaan siirtää erikseen sopimalla tai käyttämällä organisaation toimittamia standardoituja ympäristöjä ja työkaluja. Kehitysympäristöjen ja -työkalujen haavoittuvuuksien seuranta on järjestettävä kuten kappaleessa Tietoturvallisen arkkitehtuurin ja suunnittelun yleisperiaatteet on kuvattu. |
| Kehitys- ja tuotantoinfrastruktuurin käyttöoikeudet on rajoitettava mahdollisuuksien mukaan | Kehitys- ja tuotantoinfrastruktuurissa on vältettävä jaettuja tunnuksia ihmisten käytössä, paitsi jos sitä ei voi teknisesti välttää. Automaation käyttämien tunnusten oikeudet on rajoitettava niihin toimintoihin, jotka automaatio tarvitsee. |
| Ohjelmistokehittäjien käyttöoikeudet on sidottava projektissa työskentelyyn | Ohjelmistokehittäjien oikeudet on poistettava, mikäli he eivät enää työskentele projektissa. Mikäli käytetään jaettua tunnusta, jaetun tunnuksen tunnistustiedot on uusittava aina, kun jokin tunnusta jakanut henkilö ei enää työskentele projektissa. |
| Ihmisten kehitysjärjestelmien käyttäjätunnuksissa on käytettävä kaksivaiheista tunnistusta | Ihmisten käytössä olevilla käyttäjätunnuksilla on oltava käytössä kaksivaiheinen tunnistus. Nämä voivat olla esimerkiksi versionhallintaa käytettäessä tietokoneen kiintolevyn salausavain ja tällä kiintolevyllä säilytettävä ssh-avain, tai pilvipalvelukonsoliin käytettävä salasana ja erikseen laskettava aikapohjainen tunniste. |
| Salaisuuksien hallinta on järjestettävä | Salaisuudet, esimerkiksi salaus- ja rajapinta-avaimet, on hallittava niin, että ne voidaan tarvittaessa helposti vaihtaa niin testaus- kuin tuotantoympäristöissäkin, ja että pääsy salaisuuksiin on vain nimetyillä henkilöillä. Salaisuudet on vietävä tuotantoon konfiguraationa. Salaisuuksien säilytys osana lähdekoodia on kielletty. |
| Pilvipalveluiden resurssit hallitaan versionhallinnassa säilytettävän konfiguraation kautta | Pilvipalvelun konfiguraatiota hallitaan versionhallinnassa. Käsityönä tehtävää pilvipalveluiden hallintaa ei tuotannossa saa tehdä salaisuuksien hallintaa lukuun ottamatta, eikä sitä suositella myöskään testiympäristöille.  Mikäli pilvipalvelun tarjoaja antaa käyttöön inventaariorajapinnan, sitä käytetään apuna ylimääräisten pilviresurssien havaitsemisessa. |
| Ohjelmistoriippuvuuksien ja koodin versionhallinnan saatavuus on turvattava | Mikäli järjestelmä on huoltovarmuuskriittinen, ohjelmistoriippuvuuksia on käytettävä Suomessa sijaitsevan kopion välityksellä. Versionhallinnan sekä tuotantoonvientiautomaation on sijaittava kokonaisuudessaan Suomessa. |
| Ohjelmistoriippuvuuksien ja koodin eheys on taattava | Mikäli ohjelmistoriippuvuus tukee käytettävän komponentin eheyden tarkastamista kryptografisin menetelmin (esimerkiksi digitaalinen allekirjoitus), se on tehtävä.  Mikäli koodin versionhallinta tukee digitaalisin allekirjoituksin toteutettua eheystarkistusta, sen käyttöönottoa tulisi harkita. |
| Ohjelmakoodi katselmoidaan | Ohjelmakoodille tehdään koodikatselmointia (*code review* tai automaattisena staattista analyysiä, *static analysis*). Ohjelmistokehitystiimi valitsee heille parhaiten sopivan tavan suorittaa tätä. Suosituksena on nk. *pull request* -menettelyn käyttö koodikatselmointiin. Erityistä tietoturvakatselmointia tarvittaessa tämä on syytä pyytää ja tiketöidä erikseen. |
| Tuotantoon viedyistä komponenteista tallennetaan kopio forensiikkatarkoituksiin | Luodaan kyky selvittää jälkikäteen, mitä koodia tuotannossa on ollut ajossa. Esimerkiksi kontteja käytettäessä tuotantoon vietävästä kontin kuvasta voidaan tehdä arkistokopio. |

# 

## Käyttöpalveluympäristöt ja operationaaliset yleisperiaatteet

|  |  |
| --- | --- |
| Periaate | Kuvaus |
| Käyttöpalveluympäristön on toteutettava turvallisuusarkkitehtuurin luottamusalueet | Tietoturva-arkkitehtuurissa kuvattujen luottamusalueiden (ks. Tietoturvallisen arkkitehtuurin ja suunnittelun yleisperiaatteet) on oltava näkyvissä ja teknisesti toteutettu verkko- ja ajoympäristöjen toteutuksessa, sikäli kun niiden toteutus on tällä tasolla mahdollista. |
| Kehitystä tukeva pääsy tuotantoaikaisiin käyttöpalveluympäristöihin on toteutettava turvallisesti | Mikäli kehitystä varten on avattava pääsy tuotantoaikaisiin käyttöpalveluympäristöihin, pääsyn on oltava avoin vain erikseen nimetyille henkilöille, tehdyistä toimenpiteistä on jäätävä auditoitava merkintä, ja pääsy on rajattava verkkoteknisesti pienimpään mahdolliseen. |
| Auditointilokien seuraaminen on järjestettävä | Auditointilokeja (ks. Auditoitavuuden yleisperiaatteet) on seurattava automaattisin hälytyksin. Hälytykset määritellään perustuen tietoturva-arkkitehtuuriin ja tärkeimpiin käyttötapauksiin (ks. Tietoturvallisen arkkitehtuurin ja suunnittelun yleisperiaatteet ja Tietoturvallisten palvelurajapintojen ja käyttöliittymien yleisperiaatteet). |
| Palvelun saatavuuden monitorointi on järjestettävä | Palvelun saatavuutta on monitoroitava sekä palvelun käyttäjille tarkoitetusta rajapinnasta että monitorointirajapintojen kautta (ks. Auditoitavuuden yleisperiaatteet). Saatavuusongelmiin on määriteltävä soveltuvat hälytykset. |
| Ajoympäristöt on kovennettava | Palvelun sovellusten ajamiseen käytettävät ajoympäristöt on kovennettava. Suositeltava kovennustapa on CIS-kovennusohjeiden noudattaminen soveltuvin osin, sekä sovellusten ajaminen rajoitetuissa ajoympäristöissä (esim. kontit) mahdollisuuksien mukaan. |
| Liikenteen salaus | Kaikki julkisen Internetin yli siirrettävä tieto on salattava kulloinkin vahvana pidettävällä tavalla. Jos muuta ei ole määritelty, soveltuva tapa on uusin TLS-protokollan versio. |
| Tallennettavan tiedon salaus | Kaikilla henkilötietoja tai turvallisuusluokiteltua materiaalia sisältävillä tallennusmedioilla on oltava käytössä tallennusmediatason salaus. Muilta osin salauksen tarve voidaan vaatia riskiperusteisesti erikseen. |

## Tietosuojan suunnittelun yleisperiaatteet

Nämä periaatteet pätevät ainoastaan järjestelmiin, joissa käsitellään henkilötietoja. Osa vaatimuksista on mainittu muissa yleisperiaatelistoissa, jolloin tässä on annettu niihin vain viite.

|  |  |
| --- | --- |
| Periaate | Kuvaus |
| Tietosuoja otetaan huomioon uhkamallinnuksessa | Kun suoritetaan uhkamallinnusta (ks. Tietoturvallisen arkkitehtuurin ja suunnittelun yleisperiaatteet), henkilötietovirtojen osalta keskustelussa on käytävä läpi myös tietosuoja-asiat. |
| Henkilötietoja ei käsitellä turhaan | Henkilötietojen käsittelyn tulee olla teknisesti välttämätöntä halutun toiminnallisuuden toteuttamiseksi. Jos toiminnallisuus voidaan toteuttaa ilman jotakin tiettyä henkilötietoa, sitä ei pidä käsitellä lainkaan. |
| Henkilötietojen siirto järjestelmien välillä on tehtävä rajapintojen avulla | Henkilötietoja ei saa siirtää tiedostovientien tai suoran tietokanta- tai tiedostojärjestelmäpääsyn kautta, vaan niiden järjestelmien välillä siirtämistä varten on luotava rajapinta. |
| Tietosuoja-asetuksen näin vaatiessa järjestelmän on toteutettava väliaikainen käsittelyn keskeytys | Järjestelmän on toteutettava kullekin järjestelmässä rekisteröidylle henkilölle tila, jossa järjestelmä lopettaa kyseisen henkilön tietojen käsittelyn (ml. lukemisen) muihin kuin erikseen määriteltyihin ylläpidollisiin tai viranomaistarkoituksiin. Tila on oltava kytkettävissä päälle ja pois. Tätä ominaisuutta tarvitaan vain, jos tietosuoja-asetus käsittelyn perusteet huomioon ottaen tätä vaatii (artiklat 18 ja 21). |
| Henkilötiedot on pystyttävä tuomaan järjestelmästä | Järjestelmän on toteutettava riittävät rajapinnat sille, että tietyn rekisteröidyn henkilön henkilötiedot ja hänen järjestelmän käytöstä syntyneet tiedot voidaan tuoda ulos koneluettavassa muodossa. |
| Henkilötiedot on pystyttävä poistamaan | Järjestelmän on toteutettava rajapinta henkilön poistamiselle järjestelmästä, mikäli järjestelmän käyttötarkoitus mahdollistaa poistopyynnöt lain nojalla tai jos henkilötietoja käsitellään suostumukseen perustuen. Tämä asettaa vaatimuksia käytetylle tietovarantorakenteelle, joka ei saa joutua epäkonsistenttiin tilaan henkilöiden poistamisen vuoksi. |
| Henkilötietojen viennistä toiselle rekisterinpitäjälle on jäätävä merkintä | Mikäli henkilötietoja viedään toiselle rekisterinpitäjälle, viennistä on jäätävä merkintä. Merkinnän on oltava saatavissa, jos rekisteröity käyttää lakisääteistä tietojenpoisto-oikeuttaan, ja nämä luovutustiedot on pystyttävä tuomaan järjestelmästä. Jos vientejä säännönmukaisesti tehdään vain ja ainoastaan tietyille muille rekisterinpitäjille, tätä ominaisuutta ei välttämättä tarvita. |
| Henkilötietojen käsittelyyn pyydettävästä suostumuksesta ja sen perumisesta on jäätävä merkintä | Ks. Tietoturvallisten palvelurajapintojen ja käyttöliittymien yleisperiaatteet |
| Henkilötietojen käyttöön liittyvät tiedot ja valinnat tuodaan käyttäjille oikea-aikaisesti ja oikeassa viitekehyksessä | Ks. Tietoturvallisten palvelurajapintojen ja käyttöliittymien yleisperiaatteet |
| Henkilötietojen siirto on tehtävä aina salattua yhteyttä käyttäen | Ks. Käyttöpalveluympäristöt ja operationaaliset yleisperiaatteet |
| Henkilötiedot on salattava silloin, kun ne on tallennettu | Ks. Käyttöpalveluympäristöt ja operationaaliset yleisperiaatteet. Tapauskohtaisesti on harkittava, tarvitaanko myös tietovarantotasoista (esimerkiksi tietokantataulu- tai rivikohtaista) salausta. |
| Henkilötietoihin kohdistuvista luku- ja muutostapahtumista on tehtävä merkintä | Ks. Auditoitavuuden yleisperiaatteet |
| Henkilötietoja paljastavan rajapinnan on aina tarkistettava tunnistus ja valtuutus | Ks. Tietoturvallisten palvelurajapintojen ja käyttöliittymien yleisperiaatteet |
| Viitteenä henkilöön tulisi käyttää satunnaista, käyttötapauskohtaista tunnistetta | Ellei ole erityistä syytä muuhun, mikäli henkilöön pitää antaa viite esimerkiksi toiselle palvelulle tai sidosryhmälle tai lokitiedostoon, viitteeksi pitäisi luoda kryptografisesti satunnainen ja käyttötapauskohtainen tunniste. Tunnisteen korrelointi henkilöön tulisi järjestelmän ulkopuolella olla vaikeaa tai mahdotonta. |
| Henkilötiedoille on määritelty enimmäissäilytysaika | Tallennettaville henkilötiedoille on määritelty maksimisäilytysaika, joka perustuu lakiin tai operatiiviseen tarpeeseen. On myös määritelty, miten tiedot teknisesti poistetaan maksimisäilytysajan umpeuduttua. |

# Liite 3: Uuden tai muuttuneen toiminnallisuuden tietoturva- ja tietosuojatyön tarkastuslista

Tuoteomistaja päättää, minkä tason toiminnallisuuskuvauksiin tarkastuslistaa käytetään. Yleisimmin tämä tehdään käyttäjäkertomustasoisille (*user story*) tehtäville, mutta erityisesti tietosuojakysymyksiin voidaan usein vastata korkeammalla tasolla (*epic*). Joskus taas tarkastuslistan läpikäyminen voi olla tarpeen myös hyvin matalan tason tehtävillekin - esimerkiksi jos todennusmekanismissa suoritetaan bugikorjaus.

|  |  |
| --- | --- |
| Liittyykö uusi tai muuttunut toiminnallisuus... | |
| ...uuden tietovarannon kuten tietokannan käyttöönottoon? | * Lisätään tuotteen tehtävälistalle tehtävä (1) joka linkitetään toiminnallisuuden toteuttavaan tehtävään * Lisätään lisätylle tehtävälle (1) luokitus "uhkamallinnus" |
| ...todennuksen, valtuutuksen tai istuntojen hallinnan toteutukseen? |
| ...järjestelmän tuotantoonviennin periaatteen muutoksiin? |
| ...uuden kolmannen osapuolen riippuvuuden käyttöönottoon? |
| ...uuden rajapintaintegraation käyttöönottoon? |
| ...olemassa olevan oman rajapinnan toiminnallisuuden laajentamiseen? |
| ...muutokseen henkilötietojen käsittelyssä (mitä tai miten käsitellään)? | * Lisätään tuotteen tehtävälistalle tehtävä (2) joka linkitetään toiminnallisuuden toteuttavaan tehtävään * Lisätään toiminnallisuuden toteuttavalle tehtävälle luokitus "tietosuoja" ja lisätylle tehtävälle (2) luokitus "pia" * Lisätään tuotteen tehtävälistalle tehtävä (1) joka linkitetään toiminnallisuuden toteuttavaan tehtävään * Lisätään lisätylle tehtävälle (1) luokitus "uhkamallinnus" |
| ...muutokseen analytiikassa, mainostuksessa tai profiloinnissa? |
| ...henkilötietojen laajamittaisen käsittelyn, kattavan profiloinnin, järjestelmällisen valvonnan tai tietojen EU:n ulkopuolelle viennin aloittamiseen? | * Lisätään tuotteen tehtävälistalle tehtävä (3) joka linkitetään toiminnallisuuden toteuttavaan tehtävään * Lisätään toiminnallisuuden toteuttavalle tehtävälle luokitus "tietosuoja" ja lisätylle tehtävälle (3) luokitus "pia" * Lisätään tuotteen tehtävälistalle tehtävä (1) joka linkitetään toiminnallisuuden toteuttavaan tehtävään * Lisätään lisätylle tehtävälle (1) luokitus "uhkamallinnus" |
| ...muutokseen henkilöiden arvioinnissa, automaattiseen oikeusvaikutusten luomisessa? |
| ...muutokseen arkaluontoisten tai haavoittuvassa asemassa olevien henkilöiden tietojen käsittelyssä? |
| ...uuden tai innovatiivisen henkilötietojen hankinnan tai käsittelytekniikan käyttöönottoon? |

Tehtävä (1): "Suoritetaan uhkamallinnus toiminnallisuudelle käyttäen STRIDE-menetelmää (tietoturvaohjeen Liite 5: Uhkamallinnuksen toteutusohje). Uhkamallinnuksessa löytyvien riskien ehkäisemiseen tarkoitetut vaatimukset kirjataan tuotteen tehtävälistalle."

Tehtävä (2): "Suoritetaan tietosuojavaikutusten arviointi (tietoturvaohjeen Liite 6: Tietosuoja). Tietosuojavaikutusten arvioinnissa löytyvien riskien ehkäisemiseen ja lain noudattamiseen tarkoitetut vaatimukset kirjataan tuotteen tehtävälistalle."

Tehtävä (3): "Suoritetaan tietosuojavaikutusten arviointi tietosuoja-asetuksen määräämällä tavalla (DPIA) (tietoturvaohjeen Liite 6: Tietosuoja). Tietosuojavaikutusten arvioinnissa löytyvien riskien ehkäisemiseen ja lain noudattamiseen tarkoitetut vaatimukset kirjataan tuotteen tehtävälistalle."

# Liite 4: Tietoturva-arkkitehtuurin dokumentoinnin vähimmäisvaatimukset

Tietoturva-arkkitehtuurikuvaus tulisi mieluiten tehdä versiohallittuun kirjoitusalustaan (esimerkiksi wikiin), jotta erillisten, vanhentuvien dokumenttien leviämistä ehkäistään.

Tietoturva-arkkitehtuurin kuvauksen täsmällinen muoto ja sisältö riippuu toteutettavasta palvelusta, mutta minimissään se sisältää kolme tietovuokuvausta (*Data Flow Diagram*):

1. Järjestelmän kontekstitason kuva, joka näyttää tietovirrat ja eriteltyinä henkilötietovirrat järjestelmän ja sen ulkopuolisten liityntöjen välillä, mukaan lukien käyttäjät. Kontekstitason yhteydessä kirjoitetaan myös auki keskeisimmät käyttötapaukset lyhyesti; käyttötapauksissa mainittujen järjestelmien tulisi olla näkyvissä kuvassa.
2. Järjestelmän komponenttitason kuva, joka näyttää
   1. järjestelmän rakenteen yksittäisten käyttöjärjestelmäprosessien tasolle; sekä
   2. näiden prosessien väliset tietovirrat, henkilötietoja sisältävät tietovirrat erikseen merkittyinä; sekä
   3. järjestelmän luottamusalueet eli mitkä komponentit luottavat absoluuttisesti toisiinsa eli eivät tarvitse keskinäistä todennusta eivätkä valtuutusta; sekä
   4. monimutkaisemmista protokollista tai interaktioista viestisekvenssikaaviot (*Message Sequence Chart*).
3. Tuotantoon viennin tietovirtakaavio, joka näyttää tuotantoon viennin järjestelmät ja tietovirrat. Näitä järjestelmiä ovat muun muassa kehittäjien koneet, versionhallinta, integraatiopalvelimet ja salaisuuksien hallinnassa käytettävät palvelut.

Kuvat voidaan tuottaa osana uhkamallinnusta ja henkilötietoihin liittyvä osuus tietosuojavaikutusten arvioinnin yhteydessä.

Mikäli arkkitehti hallitsee töitään tiketöinti pohjaisen työnohjauksen kautta, arkkitehtuuridokumentaation päivittäminen kannattaa myös tiketoida, jotta sille tulee varattua riittävästi aikaa.

Mikäli organisaatio ei ole osoittanut tietosuojavaikutusten arviointien tuloksille erillistä paikkaa, myös tietosuojavaikutusten arviot kirjataan tietoturva-arkkitehtuurikuvauksen jatkoksi. Tämän dokumentaation sisältö on kuvattu kappaleessa Liite 6: Tietosuoja.

# Liite 5: Uhkamallinnuksen toteutusohje

## Uhkamallinnus yleisesti

Uhkamallinnus (*threat modeling* tai *architectural risk analysis*) on menetelmä, jossa kartoitetaan arkkitehtuurin ja tietovuokaavion avulla mahdollisia sovellusturvallisuuden ja tietosuojan riskejä sekä riskien hallintamekanismeja.

Tässä ohjeessa suositellaan käytettäväksi Microsoftin STRIDE-menetelmää[[9]](#footnote-9). Projektin tietoturva-asiantuntijaa, mikäli tällainen on nimetty, voi käyttää uhkamallinnuksen fasilitaattorina, jolloin mukana olijat oppivat menetelmän työn ohessa. Lyhyt selitys menetelmästä on myös alempana.

Uhkamallinnuksen tuloksena syntyy lista mahdollisista tietoturvaheikkouksista tai haavoittuvuuksista ja yleensä myös suoraan ehdotuksia asioista, mitä näille voisi tehdä. Tyypillisiä tuloksia ovat esimerkiksi testaustarpeet, tietyt yksittäiset testitapaukset, tarve muuttaa arkkitehtuuria tai matalamman tason suunnittelua, tai jonkin tietoturvaominaisuuden lisääminen.

Nämä löydökset siirretään tuotteen tehtävälistalle uusina tehtävinä tai olemassa olevien tehtävien hyväksyntäkriteereinä. Uudet tehtävät luokitellaan avainsanalla "uhkamallinnus".

## Uhkamallinnuksen aikataulutus

Paras tulos uhkamallinnukselle saadaan, jos sitä vaaditaan osana tuotteen tehtävälistaa tärkeimpien toiminnallisten vaatimusten hyväksyntäkriteerinä tai erillisenä tehtävänä, joka on linkitetty toiminnalliseen vaatimukseen. Näin mallinnustehtävät säilyvät järjellisen kokoisina ja asioista keskustellaan ajoissa, jolloin muutosten teko on helpointa.

Mikäli käytetään Scrum-mallia, uhkamallinnusta ei yleensä kannata tehdä osana Sprint Planningia vaan pikemminkin työjakson (sprintin) kestäessä. Yksi vaihtoehto on poimia työjakson sisältöön esimerkiksi tulevien työjaksojen toiminnallisuuksien uhkamallinnusta, jolloin uhkamallinnus kulkee 1-2 työjaksoa toteutusta edellä. Uhkamallinnuksen tekeminen edeltävällä työjaksolla ehkäisee myös sitä, että työjakso epäonnistuisi ilmenevän lisätyön vuoksi. Uhkamallinnuksen käytännön aikataulutus on myös vapaampaa, jos se ei kohdistu juuri sillä hetkellä toteutuksen alaisena olevaan toiminnallisuuteen. Ks. myös kappale Erityishuomioita tiettyihin ohjelmistokehitys- ja kulttuurisiin malleihin.

## Uhkamallinnus käytännössä

Tyypillinen uhkamallinnus suoritetaan työpajatyyppisesti. Paikalle pyydetään kaikki ne henkilöt, jotka tietävät tekniset yksityiskohdat arkkitehtuurista ja toteutuksesta. Käytännössä noin viisi henkilöä on sopiva määrä. Paikalle kannattaa pyytää myös tuoteomistaja ja käyttökokemuksesta vastaava henkilö, mikäli mallinnettava toiminnallisuus on täysin uutta. Työpajalle on hyvä varata 1-2 tuntia aikaa. Pienet toiminnallisuudet hoituvat nopeammin, mutta kaksi tuntia ylittää yleensä ihmisten sietokyvyn. Jos kyseessä on laaja analyysi, esimerkiksi jo rakennetun toiminnan retrospektiivinen mallinnus, sessioita on todennäköisesti järjestettävä useita.

Työpajassa piirretään tussitaululle tietovuokaavio (*data flow diagram*, DFD) tai viestisekvenssi-kaavio (*message sequence chart*, MSC). Viestisekvenssikaavio kannattaa mahdollisesti tuottaa jo ennakkoon niiden monimutkaisuuden vuoksi.

Tietovuokaaviota käytetään, jos mallinnetaan ajossa olevaa järjestelmää (sen prosesseja, tie­tovarastoja ja eri komponenttien välistä kommunikaatiota) tai tuotantoon viennin arkkitehtuuria, kuten yhteyksiä versionhallintajärjestelmiin.

Viestisekvenssikaaviota käytetään, jos mallinnetaan useita viestejä edestakaisin lähettäviä toiminnallisuuksia - esimerkiksi todennusta tai toiminnallisuutta, jossa selainta ohjataan eri rajapintoihin tai palveluihin toiminnon aikana.

Kaavion piirtäminen tussitaululle on olennainen osa mallinnusta. Valmiita kuvia voi käyttää referenssinä, mutta niitä ei pitäisi ottaa suoraan pohjaksi. Usein käy niin, että valmis kuva ei vastaakaan todellisuutta tai kaikkien osallistujien käsitys arkkitehtuurista ei ole yhteneväi­nen. Nämä olettamusten erot ovat kokemusperäisesti usein tietoturvaongelmien lähde.

Kaavion voi piirtää esimerkiksi yksi käyttäjätarina kerrallaan, jolloin mukaan otetaan ne komponentit, jotka kulloinkin tarvitaan. Kaaviosta ei pidä unohtaa ihmisiä, joita ovat sekä käyttäjät että ylläpito.

Kun kuva on valmis, jokainen piirretty tietovuo ja tietovarasto käsitellään keskustelemalla kuutta eri aihealuetta (STRIDE):

1. **Spoofing** eli todennukseen liittyvät asiat (esim. komponenttien välisen liikenteen päätepisteiden todennus)
2. **Tampering** eli eheyteen liittyvät asiat (esim. hyökkääjän mahdollisuus muuttaa tietoa)
3. **Repudiation** eli kiistettävyyteen liittyvät asiat (esim. auditointilokin luonti ja mitä sinne tallennetaan)
4. **Information disclosure** eli luottamuksellisuuteen liittyvät asiat (esim. pääseekö hyökkääjä käsiksi salaamattomaan tietoon, tai vuotaako tietoa sivukanavia pitkin)
5. **Denial of Service** eli saatavuus ja palvelunestohyökkäykset (esim. jos tietty komponentti ei vastaa, mitä se aiheuttaa, tai onko jonkin komponentin kautta mahdollisuus generoida liikaa kuormaa jollekin toiselle komponentille)
6. **Elevation of Privilege** eli käyttövaltuuksien ylittäminen (esim. koodi-injektiohyökkäykset

Keskustelun voi ohjata esimerkiksi niin, että kehittäjät ottavat yhden tietovuon kerrallaan ja lausuvat ääneen teknisen perusteen sille, miksi esimerkiksi todennus on turvallisesti tehty. Muiden tehtävä on kyseenalaistaa tätä argumenttia ja tehdä tarkentavia kysymyksiä. Kun kukaan ei enää pysty keksimään heikkoja kohtia argumentaatiosta (tai argumentti on osoitettu vaillinaiseksi ja tarvittavat toimenpiteet on kirjattu), voidaan siirtyä eteenpäin. Seuraava keskustelun kohde voidaan valita mekanistisesti STRIDE-mallin mukaan tai vapaammin mielenkiinnon ja prioriteettien mukaan.

Mikäli uhkamallinnusta tehdään henkilötietoja käsittelevälle toiminnallisuudelle, keskustelua voidaan jatkaa nk. TRIM-lisäyksellä:

1. **Transfer** of personal data: Siirretäänkö tietovuossa henkilötietoja maan tai sopimusteknisen rajan ylitse, ja onko tähän oikeus ja lupa
2. **Retention / Removal**: Jos komponentti tallentaa henkilötietoja, onko määritelty aika- tai muu kriteeri, jonka perusteella tallennetut tiedot tuhotaan; miten teknisesti toteutetaan yksittäisen henkilön tietojen poisto tarvittaessa ja miten yksittäisen henkilön tietojen käsittely voidaan pyynnöstä keskeyttää.
3. **Informed consent**: Kun komponentti lähettää henkilötietoja eteenpäin, onko loppukäyttäjää riittävästi informoitu siitä, että juuri nämä tiedot siirretään, onko käyttäjä ymmärtänyt tämän merkityksen, ja voisiko tämän tehdä vielä oikea-aikaisemmin.
4. **Minimisation**: Kun komponentti lähettää henkilötietoja eteenpäin (esimerkiksi johonkin raja-pintaan), onko tämä henkilötietojen joukko teknisesti ottaen pienin mahdollinen, vai voisiko sitä vielä entisestään pienentää teknisen toiminnallisuuden kärsimättä?

**TRIM ei korvaa tietosuojavaikutusten arviointia** (PIA, *privacy impact assessment* tai DPIA, *data protection impact assessment*), mutta se toimii alimman tason turvaverkkona sen suhteen, että jos henkilötietojen käsittelystä ei ole aiemmin tehty vaatimuksia, asia nousee viimeistään tässä vaiheessa keskusteluun.

Mikäli TRIM-analyysissä ilmenee, että jokin tietosuojanäkökanta on huonosti selvitetty, lisätään tehtävälistalle uusi tehtävä tietosuojavaikutusten arvioinnista (ks. Liite 6: Tietosuoja), joka luokitellaan avainsanalla "pia".

## Uhkamallinnuksen laatu

Uhkamallinnuksen tulosten laatuun vaikuttaa usein melko paljon sitä tekevien henkilöiden kokemus. Monet heikkoudet nimittäin ilmenevät eri järjestelmissä uudestaan ja uudestaan, ja kokemus mahdollistaa tehokkaan uhkamallinnuksen paremmin kuin vain listojen läpikäynti.

Kuitenkin on käynyt kokemuksen pohjalta selväksi, että nimenomaan STRIDE-tyyppinen, suhteellisen vapaamuotoinen mutta kuitenkin ohjattu analyysimenetelmä tuottaa varsin hyviä tuloksia ammattimaisten ohjelmistokehittäjien käsissä jo yhden tai kahden fasilitoidun esimerkkityöpajan jälkeen.

Yksi tärkeimmistä laadun takeista on olla hyväksymättä oletuksia. Jos keskustelussa käy ilmi, että jostakin tietovirrasta tai sen sisällöstä ei ole täyttä varmuutta, se kannattaa varmistaa vaikka koodia lukemalla, rajapintakuvauksesta tai protokolla-analysaattorilla. Samaten kuvan piirtämisessä kannattaa olla huolellinen; notaatiolla ei juuri ole merkitystä, kunhan on selvää, mikä komponentti keskustelee minkäkin kanssa.

# Liite 6: Tietosuojan toteutumisen varmistaminen

**Tietosuojavaikutusten arviointi yleisesti**

Tietosuojavaikutusten arvioinnin tarvittavat taustatiedot ja monet toimenpiteetkin ovat käytännössä samat kuin uhkamallinnuksen ja tietoturva-arkkitehtuurin, joten samat henkilöt, jotka ovat tekemässä uhkamallinnusta ja dokumentoimassa arkkitehtuuria ovat todennäköisesti parhaita tekemään myös tietosuojavaikutusten arvioinnin. Tarvittaessa apua voidaan pyytää organisaation tietosuojafunktiolta.

Tietosuojavaikutusten arvioinnin vaiheet ovat:

1. Määritellään kohde. Yleensä tämä on tietyn tuotteen tehtävälistan tehtävän kuvaama toiminnallisuus tai laajempi kokonaisuus. Tietosuojavaikutusten arvioinnista avataan dokumentointia varten oma tehtävä, joka linkitetään kohteena olevaan tehtävään ja luokitellaan avainsanalla "pia".
2. Selvitetään kohteen henkilötietovirrat ja tallennuspaikat konteksti- ja komponenttitasolla. Tämä on sama vaatimus kuin mikä tarvitaan tietoturva-arkkitehtuurin dokumentaatioon (ks. Liite 4: Tietoturva-arkkitehtuurin dokumentoinnin vähimmäisvaatimukset). Henkilötietovirroista on selvitettävä niissä siirrettävät henkilötietotyypit ja se, millä tavalla siirron tai tallennuksen suojaus on toteutettu.
3. Varmistetaan, että kohteen uhkamallinnus (ks. Liite 5: Uhkamallinnuksen toteutusohje TRIM-lisäyksin) on tehty.
4. Kohdassa 2 selvitetyille henkilötietovirroille ja -varannoille kirjataan:
   1. Käsittelyn laillinen peruste
   2. Henkilötietojen maantieteellinen sijainti, ja peruste niiden viemiselle pois EU:n alueelta, jos näin tapahtuu
   3. Luettelo ulkopuolisista tietojen käsittelijöistä, jotka saavat henkilötietoja, mukaan lukien mahdolliset käyttöpalvelun toimittajat; jos näitä on, varmistetaan, että sopimuksissa on huomioitu tietosuoja-asetuksen vaatimukset.
   4. Käydään läpi kappaleen Tietosuojan suunnittelun yleisperiaatteet vaatimukset ja varmistetaan, että ne täyttyvät
   5. Käsiteltävien henkilötietojen määrä ja laatu (esimerkiksi arkaluonteiset henkilötiedot).
5. Mikäli kohdan 4 keskustelussa ilmenee puutteita tavoitetilaan (esimerkiksi vastausta ei osata antaa), puutteiden korjaamisesta luodaan tehtävä tuotteen tehtävälistalle. Tämä tehtävä linkitetään kohteena olevaan tehtävään ja luokitellaan avainsanalla "pia".
6. Muut kohdan 4 tiedot dokumentoidaan mieluiten versioituun kirjoitusalustaan kuten wikiin, ja luodut korjaustehtävät linkitetään tähän dokumentaatioon. Organisaation tietosuojatoiminto voi osoittaa tälle dokumentaatiolle keskitetyn paikan. Muussa tapauksessa dokumentaatio kirjataan tietoturva-arkkitehtuurikuvauksen yhteyteen.
7. Mikäli yllä kuvattujen analyysien ja uhkamallinuksen pohjalta päädytään siihen, että henkilötietoihin kohdistuu todennäköisesti korkea riski, otetaan yhteyttä <ORGANISAATIO>:n tietosuojavastaavaan. Tietosuojavastaava käynnistää tarvittaessa tietosuoja-asetuksen vaatiman tietosuojavaikutusten arvioinnin (DPIA). Todennäköisesti korkea riski muodostuu aina kun:

* käsitellään suuria määriä henkilötietoja
* käsitellään laajamittaisesti arkaluonteisia tietoja

**Tietosuoja-asetuksen vaatima tietosuojavaikutusten arviointi (DPIA)**

<ORGANISAATIO>:n tietosuojavastaavan käynnistämässä ja ohjaamassa tietosuojavaikutusten arvioinnissa kehitysprojekti toimittaa ainakin seuraavat tiedot:

1. järjestelmällinen kuvaus suunnitelluista käsittelytoimista ja käsittelyn tarkoituksista; ja
2. arvio käsittelytoimien tarpeellisuudesta ja oikeasuhteisuudesta tarkoituksiin nähden; ja
3. arvio rekisteröityjen oikeuksia ja vapauksia koskevista riskeistä; ja
4. suunnitellut toimenpiteet riskeihin puuttumiseksi, mukaan lukien suoja- ja turvallisuustoimet ja mekanismit.

Kohdan 4 osalta kehitystiimin tulisi pystyä identifioimaan ne tuotteen tehtävälistan tehtävät, joissa suunnitellut toimenpiteet on kuvattu.

Tietosuojavaikutusten arvioinnissa yllämainittujen kohtien teemat keskustellaan tietosuojavastaavan, tuoteomistajan ja tarvittavien kehitystiimin teknisten asiantuntijoiden kesken. Keskustelu dokumentoidaan. Mikäli keskustelun tuloksena vaaditaan toiminnallisia muutoksia, ne avataan uusina tehtävinä tuotteen tehtävälistalle samoin kuin uhkamallinnuksesta seuraaville uusille vaatimuksille tehdään. Nämä uudet tehtävät leimataan avainsanalla ”pia”.

# Liite 7: Tekninen liite

Erillinen liitedokumentti *Liite 7\_Tekninen liite\_eng\_fin.docx*

1. General Data Protection Regulation: Euroopan parlamentin ja neuvoston asetus (EU) 2016/679 luonnollisten henkilöiden suojelusta henkilötietojen käsittelyssä sekä näiden tietojen vapaasta liikkuvuudesta ja direktiivin 95/46/EY kumoamisesta [↑](#footnote-ref-1)
2. Euroopan parlamentin ja neuvoston asetus (EU) N:o 910/2014 sähköisestä tunnistamisesta ja sähköisiin transaktioihin liittyvistä luottamuspalveluista sisämarkkinoilla ja direktiivin 1999/93/EY kumoamisesta [↑](#footnote-ref-2)
3. Sovelluskehityksen tietoturvaohje VAHTI 1/2013 [↑](#footnote-ref-3)
4. OWASP Application Security Verification Standard [↑](#footnote-ref-4)
5. Article 29 Working Party Opinion WP248 rev. 1 tai uudempi (tietosuoja-asetuksen mukaisen tietosuojavaikutusten arvioinnin ohje) [↑](#footnote-ref-5)
6. Center for Internet Security (CIS) Benchmarks (kovennusohjeet) [↑](#footnote-ref-6)
7. Tietosuojavaikutusten arviointitarpeen määrittelee yleisen tietosuoja-asetuksen 35. artikla sekä Article 29 Data Protection Working Partyn ohje WP248 ‘Guidelines on Data Protection Impact Assessment (DPIA) and determining whether processing is “likely to result in a high risk” for the purposes of Regulation 2016/679’. [↑](#footnote-ref-7)
8. Mikäli muuta ei ole vaadittu, käytetään Center for Internet Securityn (CIS) kovennusvaatimuksia kulloisellekin käytetylle tekniikalle. [↑](#footnote-ref-8)
9. Adam Shostack: *Threat Modeling: Designing for Security*. Wiley, 2014. [↑](#footnote-ref-9)