



Jari Pirhonen
(CISSP, CISA, CSSLP)
toimii Oy Samlink Ab:n
turvallisuusjohtajana.
Hän edistää aktiivisesti
turvallisuuksiin myös
työtehtäviensä ulko-
puolella turvallisuusalan
yhdistyksissä ja verkos-
toissa. Jari on kokenut
kouluttaja ja
konferenssiintyjä.
(japi@iki.fi
<http://iki.fi/japi>)

Tietoturvallisia sovelluksia

Sovellustietoturvassa on sorruttu tietoturvan perisyntiin. Organisaatiot koettavat ratkoa tietoturvaasteita tuotteilla ja konsultoinnilla, vaikka ratkaisun avaimet ovat osaamisen ja prosessien parantamisessa.

Tietoisuus sovellustietoturvalisuuden tärkeydestä on viime vuosina lisääntynyt merkittävästi, mutta valitettavan moni organisaatio on jättänyt käytännön toimet sovellusten tietoturvan parantamiseksi liian vähälle huomiolle.

Kirjoitin sovellustietoturvasta Systeemyö-lehden^[1] tällä samalla otsikolla reilu kuusi vuotta sitten. Korostin silloin tietoturvan integroimista systeemyön kaikkiin vaiheisiin ja erityisesti tietoturva-vaatimusten määrittelyn tärkeyttä. Luin uudestaan vanhan kirjoitukseni ja en tiedä pitäisikö minun olla iloinen vai pettynyt siitä, että artikkeli on edelleen ajankohtainen. Olisi mukava todeta, että silloin mainitsemani perusasiat olisivat kunnossa suurimmassa osassa sovellustyötä tekeviä yrityksiä, mutta näin hyvin asiat eivät taida olla.

Kehittymistä täällä, jumiutumista tuolla

Miten sovellustyössä on kuudessa vuodessa sitten tietoturvanäkökulmasta kehitytty ja missä junnataan paikallaan?

Kehitystä on tapahtunut tietoturvatietoisuudessa, apuvälineiden kehityksessä, saatavilla olevassa kirjallisuudessa, ohjeistuksessa ja siinä, että valveutuneimmat yritykset ovat käynnistäneet sovellusturvallisuuteen liittyviä kehityshankkeita.

Yhä useammat tietoturva-asiantuntijat, sovelluskehittäjät ja parhaissa tapauksissa liiketoimintajohtokin ovat havahtuneet siihen, että liiketoimintakriittisten tietojen luottamuksellisuus on niitä käyttävien sovellusten varassa. Sovellusturvallisuuteen liittyvää kirjallisuutta on helppo löytää – tosin tuoretta tekstiä pääasiassa englanniksi. Verkko on pullollaan ohjeita, raportteja, podcasteja, webinareja, parhaita käytäntöjä ja tuoreimpana jopa malleja tietoturvan integroimiseksi systeemyöhön. Edistyneimmillä yrityksillä on ollut jo vuosia käynnissä hankkeita sovellus-

ten tietoturvan parantamiseksi. Joillekin voi tulla yllätyksenä se, että Microsoft on ollut edelläkävijä sovelluskehitykseen liittyvässä tietoturvatyössä ja on myös antanut runsaasti kokemuksiaan, dokumentaatiotaan ja työkalujaan vapaaseen käyttöön.

Ennallaan asiat ovat sikäli, että sovelluskehittäjiä aktiivisemmin sovellustietoturvan tärkeyttä tuovat esille tietoturva-asiantuntijat ja -auditoijat, ohjelmoijien tietoturvakoulutus on vielä satunnaisten tietoiskujen varassa ja liian paljon laskeaan valmiin sovelluksen tietoturva-auditoinnin varaan.

Tuntumani on, että valtaosin sovellusohjelmiojille, -arkkitehdeille ja –testaajille tietoturva on vielä melko vierasta. Tämä voi aiheuttaa melkoisen konfliktin, kun sovellustyön realiteetteja huonosti tuntevat tietoturva-asiantuntijat ja valmiin lopputuloksen auki repivät tietoturva-auditoijat kohtaavat sovelluskehittäjät. Tilanne on pahimmillaan se, että tietoturva-auditoija tulee arvioimaan sovelluksen juuri ennen käyttöönottoa ja kertoo sovelluksen olevan tietoturvaton ja käyttöönottokelvoton.

Sovelluskehittäjien työhönsä liittyvä tietoturvakoulutus on usein riittämätöntä, mahdollisesti vain muutamien tietoiskujen varassa. Nämäkin keskittyvät usein ”hakkerointiin” eli yksittäisten sovellusongelmien löytämiseen. Tärkeämpää olisi kouluttaa tietoturvasuunnittelun periaatteisiin ja hyviin tietoturvamalleihin. Sovellusten tietoturva-auditoijat elävät kulta-aikaa, koska monessa organisaatiossa halutaan pikaratkaisuja. Siellä missä vaadittaisiin laajaa koulutusta, prosessien uudistamista ja dokumentointityötä, ostetaan mieluummin ulkopuolinen auditoija kertomaan mitkä ovat sovelluksen pahimmat tietoturvaongelmat.

Tietoturvatehtävien integrointi systeemyöhön

Systeemyömalleissa ei oletusarvoisesti huomioida formaalisti tietoturvatehtäviä. Käytännössä yritykset joutuvat itse rakentamaan haluamansa tietoturvatehtävät sisään systeemyömalliinsa. Muutama vuosi sitten tekemässäni tutkielmassa arvioin kaupallista systeemyömallia tietotur-

Oy Samlink Ab
tukee finanssialan
asiakkaidensa liike-
toimintaa tarjoamalla
luotettavia tietoteknisiä
palveluja ja tukipalveluja,
kuten pankkitoiminnan
palveluja, verkkopalve-
luja sekä näihin liittyviä
tukipalveluja. Samlink
toimittaa monipankkijär-
jestelmiä mm. Aktialle,
Suomen Handelsbanke-
nille, Säästöpankeille ja
Paikallisosuuspankeille.
Samlinkin asiakaspan-
keilla on yhteensä yli
miljoona asiakasta ja
noin 500 000 asiakasta
käyttää kehittämäämme
Internet-pankkipalvelua.

vanäkökulmasta^[2]. Arvioinnin kohteena olleen mallin tietoturvaosuus rajoittui käytännössä sekalaiseen ohjeistukseen, joka ei systemaattisesti tukenut sovelluksen tietoturvatavoitteiden asettamista ja saavuttamista.

Tutkielmassani esitin ehdotuksia tietoturva-tehtäviksi systeemyön eri vaiheisiin. Käytännössä tietoturvatehtäviä ei kuitenkaan voi ottaa mukaan kertarysäyksellä, vaan niitä on lisättävä sopivissa erissä. Se, missä järjestyksessä ja millä aikataululla tietoturvatehtäviä otetaan mukaan, riippuu organisaation systeemyömallin nykytilasta ja systeemyön kypsyydestä. Mikäli tietoturvan integroiminen systeemyöhön on vielä alkuvaiheissaan, on varauduttava usean vuoden systemaattiseen työhön. Kyseessä on merkittävä systeemyökulttuurin muutos sekä koulutus- ja kehitysponnistus.

Kuva 1 esittää kaksi näkökulmaa tietoturvatehtäviin. Kuvassa on pitkän linjan sovellustietoturvalähtettilään, Gary McGrawn, sovellustietoturvan kuusi kosketuspintaa. Esitystapa on tarkoituksellisesti valittu niin, että se ei ota kantaa siihen, onko systeemyössä käytössä vesiputous-, spiraali-, ketterä vai joku muu malli. Kuvan yläosassa on Gary McGrawn näkemys tärkeimmistä tietoturva-tehtävistä eri vaiheissa^[3]. Tehtävät on priorisoitu esitetyn numerojärjestyksen mukaisesti. Kuvan alaosassa on vastaavasti Microsoftin näkemys asiasta^[4].

Kuvasta 1 näkee mainiosti, kuinka tehtävät ovat pääosin samankaltaisia, mutta käyttöönottojärjestys vaihtelee. McGraw esimerkiksi suosittelee koodikatselmointia ensimmäisenä käyttöönotettavaksi tietoturvatehtäväksi, Microsoft puolestaan suosittelee ensimmäiseksi tietoturva-vaatimusten dokumentointia.

Viime aikoina on vihdoin saatu hyvää dokumentaatiota auttamaan organisaatioita tietoturvan integroimisessa systeemyöhön. Suositeltavia lähteitä systeemyömallin kehittämisen tueksi ovat mm. The Building Security In Maturity Model^[5], Software Assurance Maturity Model^[6], Microsoft Security Development Lifecycle^[7] ja Safecode Fundamental Practices for Secure Software Development^[8]. Runsaasti lisää viitteitä löytyy julkisesta kirjanmerkkiluettelostani^[9].

Tietoa löytyy, mutta mitään ohjetta ei voi integroida suoraan organisaation tapaan työskennellä. Kukin joutuu miettimään itselleen parhaiten soveltuvan lähestymismallin olemassa olevia ohjeita soveltaen.

Viisi ensimmäistä keinoa parantaa sovellustietoturvaa

Esitän seuraavaksi omiin kokemuksiini perustuvan näkemykseni viidestä tärkeimmästä systeemyön tietoturvatehtävästä. Kaiken tietoturvatyön mahdollistamiseksi tarvittavien organisaation halun parantaa tietoturvallisuutta ja johdon sitoutumisen tietoturvan kehittämiseen oletan jo olevan kunnossa.

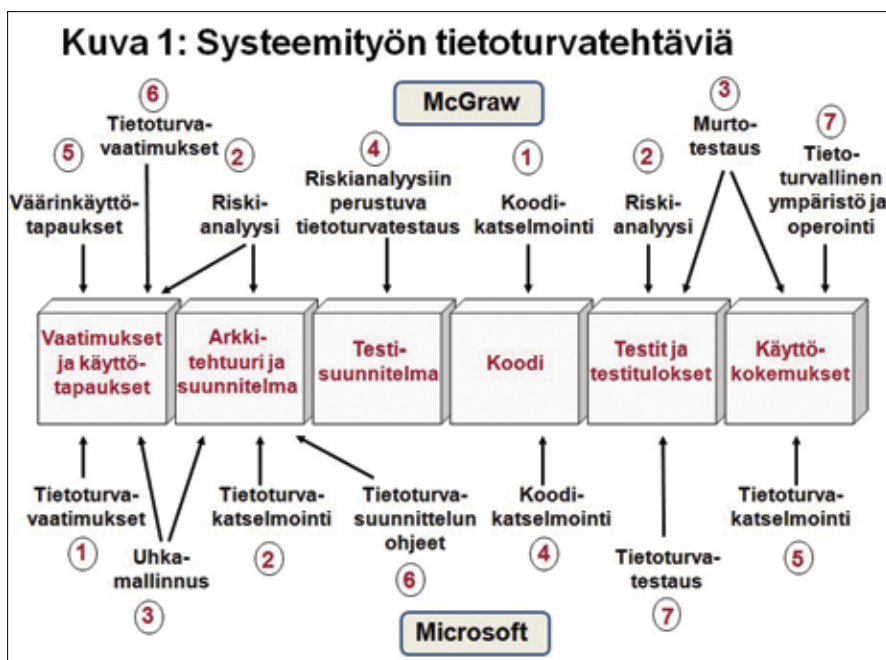
1. Käytä ulkopuolista tietoturva-auditoijaa

Mikäli sovellusten tietoturvatyötä ei vielä aktiivisesti tehdä, on parasta hankkia ulkopuolista apua. Kokenut tietoturva-auditoija havaitsee ainakin selkeimmät tietoturvaongelmat ja osaa opastaa niiden korjaamisessa. Korjatessaan virheellistä koodiaan ohjelmoijakin oppinee ainakin hetkeksi välttämään vastaavia virheitä. Muista kuitenkin, että tietoturva-auditointi ei takaa sovelluksen turvallisuutta.

Ideaalitilanteessa auditoinnin tuloksia pitäisi käyttää oman sovellustietoturvatyön parantamiseen. Oman osaamisen lisääntyessä ulkoista tietoturva-auditointia voisi lopulta käyttää vain tärkeimpien sovellusten yhteydessä tai pistokokeina varmistamaan toiminnan ja lopputuloksen laatu. Mahdolliset tietoturvaongelmat täytyy paitsi korjata, myös jäljittää niiden syyt. Olivatko tietoturva-vaatimukset puutteelliset? Onko ohjelmoijien osaamisessa kehittämistä? Onko testauksessa parantamisen varaa? Tietoturva-auditoinnin tulosten kautta kehitetään omia toimintatapoja ja työkaluja.

2. Kouluta sovellustietoturvaa

Tietoturvallisuus tehdään loppujen lopuksi yksittäisissä työtehtävissä ja jokainen tehtävä vaatii tietoturvaan omaa näkökulmaansa ja erikoisosaamistaan. Arkkitehdin täytyy ymmärtää tietoturvapalvelut, -protokollat ja tuotteiden tietoturvamahdollisuudet. Suunnittelijan on tunnettava tietoturvalliset suunnitteluperiaatteet. Ohjelmoijien on osattava välttää tietoturvaongelmia aiheuttavia virheitä ja tunnettava ohjelmointi-



kielikohtaiset sudenkuopat. Testaajan on pyrittävä tietoisesti rikkomaan sovellus ja siksi hänen on tunnettava tyypilliset ongelmat ja työkalut.

Kattavaa sovellustietoturvakoulutusta ei ole suomen kielellä yleisesti tarjolla. Perusteellista koulutusta haluava organisaatio joutuu joko hankkimaan räätälöityä koulutusta tai lähettämään henkilöstönsä koulutukseen ulkomaille.

OWASP^[10] on avoin, sovellustietoturvaa vapaaehtoisvoimin kehittävä organisaatio. Kuva 2 esittää OWASP-listan kymmenestä vakavimmasta tietoturvaongelmia aiheuttavasta virheestä. Vähintäänkin tämän listan asiat on jokaisen sovelluskehittäjän syytä ymmärtää ja tietää, miten ongelmat vältetään. Listaa päivitetään säännöllisesti ja kuvassa oleva lista on lopullista hyväksyntää vailla oleva uusi ehdotus. OWASP:n sivustolta löytyvät tarkemmat kuvaukset listassa esitetyistä ongelmista^[11]. Suomenokset^[12] ovat OWASP Helsinki-jaoston^[13] laatimia lukuun ottamatta kohtia 6 ja 8, jotka ovat omat suomenokseni listalle päässeiden uusien ongelmien osalta. OWASP-listan täydennykseksi sopii CWE/SANS lista 25 vaarallisimmasta ohjelmointivirheestä^[14].

Sovellusten tietoturva parantuisi huomattavasti jo sillä yksinkertaisella keinolla, että syöte muistettaisiin aina huolellisesti tarkistaa ennen käyttöä. Huomioiden erityisesti se, että syöte voi tulla muualtakin kuin käyttäjältä, kuten toiselta sovellukselta tai tietokannasta.

Toivottavasti tulevaisuudessa entistä useampi sovelluskehittäjällä on saanut kattavaa tietoturvakoulutusta ja jopa osaamisen osoittavan sertifikaatin. Sovelluskehitykseen liittyviä sertifikaatteja ovat mm. SANS GIAC Secure Software Programmer (GSSP) ja (ISC)2 Certified Secure Software Lifecycle Professional (CSSLP) sekä EC-Council Certified Secure Programmer ja Certified Secure Application Developer.

3. Selvitä tietoturva-vaatimukset

Sovelluksen tietoturva-vaatimukset on selvitettävä, dokumentoitava ja käsiteltävä kuten sovelluksen muutkin vaatimukset. Sovelluksen tietoturvaa ei saa jättää sattuman tai yksittäisten ohjelmoijien osaamisen varaan.

Tietoturva-vaatimukseen on erilaisia näkökulmia. Ulkoiset tietoturva-vaatimukset (lait, määräykset, toimialavaatimukset) edellyttävät liiketoimintaympäristön ja sidoryhmäympäristön tuntemista. Sisäiset tietoturva-vaatimukset (tietoturva-periaatteet ja -ohjeet, tietoturva-arkkitehtuuri) tulevat organisaation tekemistä päätöksistä edellyttäen oman tietoteknisen ympäristön ja toimintaprosessien tuntemista. Sovelluskohtaiset tekniset tietoturva-vaatimukset edellyttävät tyypillisten tietoturva-ongelmien ja -ratkaisujen tuntemista.

Erityisesti teknisten tietoturva-vaatimusten löytäminen on joskus työlästä, koska niiden formaaliin käsittelyyn ei ole totuttu. Hyvä työkalu tietoturva-vaatimusten löytämiseen on tietoturvariski-arvio tai uhka-arvio. Ensi vaiheessa riittää asiantuntijoiden avoriihessä löytämien tietoturvariskien kirjaaminen, arviointi ja priorisointi sekä niihin vastaavien tietoturva-vaatimusten dokumentointi. Myöhemmin tavoitteena on tietoturvariski-arvion jatkuva päivittäminen koko systeemyön ajan suunnitelmien ja ratkaisujen tarkentuessa.

Tietoturva-vaatimusten käsittelyä ja tietoturvariski-arvion tekemistä helpottaa termin "tietoturva" välttäminen. Tietoturva on yleiskäsite, joka tarkoittaa kaikille hieman eri asiaa. Tietoturvasta keskustelun sijaan on hyödyllisempää tunnistaa suojattavia tietoja, keskustella tietojen oikeellisuus- ja luottamuksellisuusvaatimuksista, arvioida tapahtumien jäljitettävyyksivaatimuksia, pohtia tietosuojatarpeita ja määrittää sovelluksen häiriöttömyysvaatimuksia.

Tietoturva-vaatimukset löytyvät, kun sovellus-, tietoverkko- ja tietoturva-asiantuntijat yhdessä pohtivat tietoihin kohdistuvia virhe-, väärinkäyttö- ja paljastumismahdollisuuksia tietoja käsiteltäessä, liikutettaessa ja talletettaessa.

4. Tietoturvakatselmoinnit

Tietoturvallisuus on sovelluksen laatu-kriteeri. Tietoturvallisuuden parantaminen parantaa sovelluksen laatua myös yleisesti. Vastaavasti laadunparannukseen tähtäävät toimenpiteet parantavat myös tietoturvallisuutta. Tietoturvakatselmoinnit ovat hyvä tapa parantaa sovelluksen laatua ja lisätä osaamista.

Tietoturvakatselmoitteja voidaan tehdä esim. määrittelyille, suunnitelmille, sovelluskoodille tai dokumentaatiolle. Pelkkä tietoisuus siitä, että kollega tai ulkopuolinen taho tulee katselmoimaan työtä tietoturvanäkökulmasta, vaikuttaa yleensä

Kuva 2: OWASP Top 10 sovellusongelmat

1. Taustajärjestelmäkyselyn rakenne ei säily (Injection)
2. Verkkosivun rakenne ei säily (Cross Site Scripting)
3. Puutteellinen tunnistusmenettely ja istunnonhallinta (Broken Authentication and Session Management)
4. Turvaton suora viittaus tietoalkioon (Insecure Direct Object References)
5. Puutteellinen pyynnön alkuperän tarkistus (Cross Site Request Forgery)
6. Tietoturvatonta tai virheellisiä asetuksia (Security Misconfiguration)
7. Rajoittamaton URL-tason pääsy (Failure to Restrict URL Access)
8. Tarkistamattomat linkkien uudelleenohjaukset (Unvalidated Redirects and Forwards)
9. Puutteellinen tietojen salaaminen (Insecure Cryptographic Storage)
10. Turvattomat tietoliikenneyhteydet (Insufficient Transport Layer Protection)

työn tekemiseen lopputulosta parantavasti. Katselmointi on mainio oppimistilaisuus sekä katselmoijalle että katselmoitavalle. Tehdyistä havainnoista keskustelu kehittää molempien osaamista ja roolit voivat vaihtua kohteittain.

Etenkin katselmointien aloitusvaiheessa erilaisista tietoturvan tarkistuslistoista on hyötyä. Katselmoinnin rutiininomaisissa osuuksissa ne toimivat hyvin muistin tukena, kunhan ymmärretään, että tarkistuslistat eivät koskaan voi olla kaiken kattavia.

5. Sovellustietoturvaryhmän perustaminen

Sovellustietoturvaryhmän perustaminen havaittiin tärkeimmäksi yhteiseksi tekijäksi tutkimuksessa, jossa arvioitiin yritysten sovellustietoturvan kehitysohjelmia^[15].

Perussyntejä organisaatioissa on, että kuvitellaan sovellustietoturvan syntyvän tietoturva- tai IT-asiantuntijoiden toimesta. Usein tietoturva-asiantuntijat ovat organisaatiossa ensimmäisiä, jotka huolestuvat sovellusten tietoturvasasta. Tietoturva- ja IT-asiantuntijoilla on kuitenkin harvoin ymmärrystä sovelluskehityksen kiemuroista ja haasteista, puhumattakaan ohjelmointikielistä ja sovelluskehitysohjelmista. Tietoturva- ja IT-asiantuntijat voivat toimia tukena, mutta sovellustietoturvan kehittämiseen tarvitaan sovelluskehityksen ammattilaisia. Sovellusten tietoturvan kannalta on huomattavasti hedelmällisempää opastaa sovelluskehittäjiä tietoturvaan, kuin koettaa tehdä tietoturva-asiantuntijoista sovelluskehittäjiä.

Yksi hyvä tapa kasvattaa organisaation osaamista on nimetä tietoturvavastuullisia eri osa-alueille ja määritellä tietoturvaosaamisen kehittäminen osaksi työnkuvaa. Arkkitehdit, ohjelmoijat, testaajat, projektipäälliköt – jokaiselta osa-alueelta voi löytää asiasta kiinnostuneen ja nimetä hänet tietoturvamentoriksi.

Sovelluskehitykseen osallistuvista voi koota ryhmän, joka aktiivisesti kehittää tietoturvatietämystään ja jakaa sitä kollegoilleen. Ryhmä voi myös toimia neuvonantajana systeemyön tietoturvallisuuden kehittämisessä.

Vaadi sovelluksilta tietoturvaa

Sovellusten tietoturvallisuus paranee vain, jos käyttäjät vaativat tietoturvaa ja sen toteutumisen osoittamista. Kysykää sovellustoimittajaltanne, millä perusteella sovellus on tietoturvallinen, miten tietoturvallisuus on huomioitu ja miten sovelluskehittäjät on koulutettu tietoturvallisuuteen. Mikäli sovellustoimittajan vastaus kyselyihinne on vain, että sovellus on suojattu palomuurilla ja yhteydet SSL-salauksella, vaihtakaa toimittajaa.

Vaatikaa riskilähtöistä lähestymistapaa, vaatikaa perusteltua ja dokumentoitua tietoturvaa, vaatikaa tietoturvaosaamista sovelluskehittäjiltä, vaatikaa ulkopuolista arviointia kriittisiltä sovelluksilta.

Autoteollisuudella kesti 50 vuotta, ennen kuin turvallisuuteen alettiin kiinnittää huomiota. Autojen turvaominaisuudet olivat vaatimattomat ja onnettomuudet – usein kuolemaan johtavat – katsottiin kuskin syyksi. Ralph Naderin 1965 julkaiseva kirja, "Unsafe At Any Speed", kyseenalaisti autoteollisuuden tavoitteet ja vastuut. Tämän seurauksena autonvalmistajilta alettiin vaatia turvallisia ajoneuvoja, alkaen turvavöistä ja kestävämmistä tuulilaseista. Sovelluskehitystä on tehty 50 vuotta – onko aika herätä vaatimaan tietoturvallisia sovelluksia?

Viitteet

1. *Tietoturvallisia sovelluksia, Systeemyölehti 2003-4 s. 14-17, Jari Pirhonen, <http://www.pcu.fi/sytyke/lehti/kirj/st20034/st034.pdf>*
2. *Erään systeemyömallin arviointi tietoturvanäkökulmasta, TKK Dipoli Turvallisuusjohdon koulutusohjelman tutkielma, Jari Pirhonen, <http://koti.welho.com/jpirhone/docs/tjk8.pdf>*
3. *Software Security – Building Security In, Addison-Wesley Professional, Gary McGraw*
4. *Patterns & Practices - Security Engineering Explained, Microsoft, <http://msdn.microsoft.com/en-us/library/ms998382.aspx>*
5. *The Building Security In Maturity Model, <http://www.bsi-mm.com/>*
6. *The Software Assurance Maturity Model, <http://www.opensamm.org/>*
7. *Microsoft Security Development Lifecycle, <http://www.microsoft.com/sdl>*
8. *Safecode Fundamental Practices for Secure Software Development, http://www.safecode.org/publications/SAFECode_Dev_Practices1108.pdf*
9. *Sovellustietoturvan linkkikokoelma, <http://koti.welho.com/jpirhone/security.html#prog>*
10. *The Open Web Application Security Project (OWASP), <http://www.owasp.org/>*
11. *OWASP Top Ten Project 2010 Release Candidate, http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project*
12. *OWASP Top Ten 2007 Finnish, http://www.owasp.org/index.php/Top_10_2007_Finnish*
13. *OWASP Helsinki Local Chapter, <http://www.owasp.org/index.php/Helsinki>*
14. *CWE/SANS TOP 25 Most Dagerous Programming Errors, <http://www.sans.org/top25-programming-errors/>*
15. *Software [In]security: You Really Need a Software Security Group, <http://www.informit.com/articles/article.aspx?p=1434903>*