Tietoturvallisuuden huomioiminen sovelluskehityksessä

Data Security Conference 2002
24.10.2002

Jari.Pirhonen@atbusiness.com
Senior Security Consultant, CISSP, CISA
AtBusiness Communications Oyj

www.atbusiness.com
www.iki.fi/japi/

Sisältö

- Tietoturva sovelluksissa
  - tietoturvahaasteita sovelluksille
  - tietoturvatavoitteet vs. sovellusprojektin tavoitteet
- Sovelluskehityksen haasteet
  - web-sovelluksen koostumus
  - tietoturvahaasteet sovelluskehityksessä
- Standardit & ohjeet
  - tärkeimpiä standardeja
- Tietoturvan huomioiminen sovellusprojektissa
  - määrittely
  - suunnittelu
  - arkkitehtuurit
  - toteutus
  - testaus
  - käyttöönotto
  - ylläpito

Sisältö

- Uusia haasteita
  - Web Services
  - XML
- Esimerkkejä sovellusongelmista
  - tunnettuja ongelmia
  - puskuriylivuoto
  - SQL injection
  - Format String Attack
  - Cross-site scripting
  - vastuun siirto käyttäjälle

Sisältö

- **Tietoturva sovelluksissa**
- Sovelluskehityksen haasteet
- Standardit & ohjeet
- Tietoturvan huomioiminen sovellusprojektissa
- Uusia haasteita
- Esimerkkejä sovellusongelmista

"If J. Random Websurfer clicks on a button that promises dancing pigs on his computer monitor, and instead gets a hortatory message describing the potential dangers of the applet - he's going to choose dancing pigs over computer security any day."

Bruce Schneier

Turvallista?

- “Palvelumme tietoturva perustuu www-selaimien ominaisuuksiin, asiakastunnuksiin sekä niitä täydentäviin tunnisteisiin. Näiden lisäksi käytämme alan kehittyneimpiä palomuuri- ym. sisäisiä tietoturva- ja varmistusratkaisuja, jotka ovat asiakkaillemme näkymättömiä.”
- “Turvallisuus perustuu SSL-turvatekniikkaan, joka salakirjoittaa koko tietoliikenteen jatkuvasti vaihtuvilla avaimilla. Ulkopuolisten ei ole mahdollista muuttaa tai lukea siirrettävää tietoa. “
- ”Järjestelmää voivat käyttää vain sopimuksen tehneen organisaation nimeämät käyttäjät. Selaimen ja järjestelmän välinen tietoliikenne on salattu. Palvelussa on nykyaikainen palomuuriratkaisu.”
- "At the core of HYDRA's security features is a biomorphic technology based on a field of mathematics called 'Chaotic Dynamics.' Using Chaos Theory, HYDRA can generate special groups of characters called Bodacions. Bodacions are impossible to guess, and never repeat. “

Tietoturva

Luottamuksellisuus
Eheys
Käytettävyys

Tunnistus
Valtuutus
Auditointi
Kiistämättömyys

Oikeellisuus
Ajantasaisuus
Yksityisyys
Anonymiteetti
Monitoroitavuus
Jäljitettävyys
Todistettavuus
Hallittavuus
Uskottavuus
Toimivuus
Laadukkuus

Miksi turvaratkaisuja tarvitaan?

atbusiness

- Hyvät turvaratkaisut **mahdollistavat** tietoverkkoja hyödyntävän liiketoiminnan.
- Mitä enemmän tietoverkkoja käytetään kriittiseen liiketoimintaan, sitä parempia ja varmempia turvaratkaisuja **halutaan**.
- Turvatoimet ovat **riskien hallintaa**
- Tietoturva parantaa sovellusten **laatua**

Ei käyttäjä-tunnistusta

tunnus + salasana

kertakäyttöiset salasanat

fyysinen laite + kertakäyttöiset salasanat

PKI, biometria

Haasteita

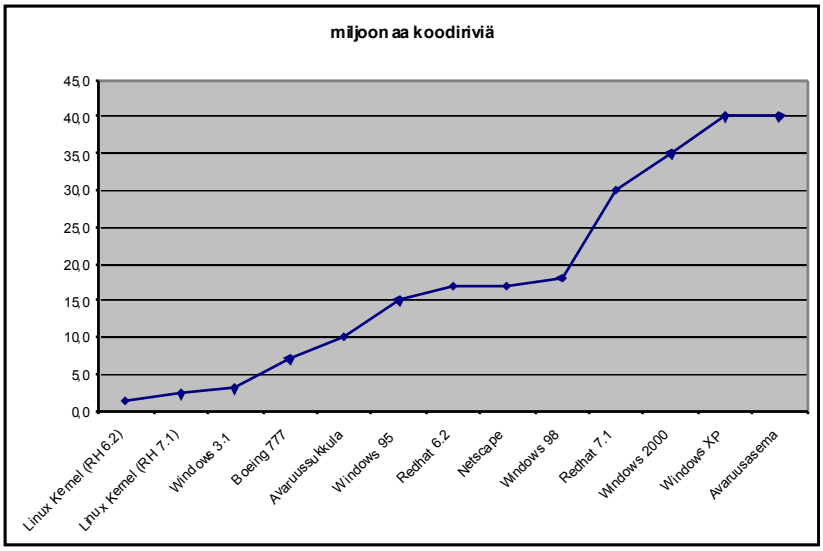
atbusiness

- **Verkottuminen**
  - yrityksen järjestelmät ”avataan” maailmalle
  - yhteydet sovelluksiin saatava kaikkialta
  - laillisten käyttäjien pääsyn helpottaminen avaa ovia myös väärinkäyttäjille
- **Laajennettavuus**
  - päivitykset verkon yli automaattisesti
  - plug-init: käyttäjän ohjelmisto muuttuu lennossa
  - mobile code: client-sovellus ”luovutetaan” käyttäjän armoille
- **Monimutkaisuus**
  - laajat, hajautetut järjestelmät, useille päätelaitteille, aina saatavilla
  - paljon valmiskomponentteja, protokollia, middleware-ohjelmistoja, rajapintoja,…

Maailma muuttuu…

- *Ympäristö* keskitetty, rajattu => *globaali, rajoittamaton*
- *Tietojärjestelmä* kuin linnake => *tori*
- *Tapahtumat* ennustettavissa => *yllätyksellisiä*
- *Vastuu* keskitetty => *jaettu*
- *Tietoturva* rajoite => *mahdollisuus*
- *Lähtökohtana* komponettien suojaus => *riskien hallinta*
- *Pääasia* turvallisuus => *selviytyminen*
- *Turvallisuusajattelu* => *laatu*

ja monimutkaistuu



Turvallisuus on kompromissi

atbusiness

Tietoturvatavoitteet

- **käytettävyys (availability)**
  - käyttökatkosten välttäminen
- **eheys (integrity)**
  - tiedot ja järjestelmät
- **luottamuksellisuus (confidentiality)**
  - tiedot vain oikeille henkilöille
- **jäljitettävyys (accountability)**
  - kuka teki mitä ja milloin?
- **luotettavuus (assurance)**
  - mistä tiedän riittävän turvallisuuden tason toteutuvan?

Sovellusprojektin tavoitteet

- **toiminnallisuus (functionality)**
  - usein tärkein (ainoa) kriteeri
- **käytettävyys (usability)**
  - tietoturva vaikeuttaa…
- **tehokkuus (efficiency)**
  - tietoturva hidastaa ja maksaa…
- **oikea-aikaisuus (time-to-market)**
  - kiire, kiire – missä oikaistaan…
- **yksinkertaisuus (simplicity)**
  - hyvä!

Tutkimus 45 sovelluksesta

atbusiness

tietoturvaongelman luokka	tietoturvaongelma löytyi sovelluksista	suunnitteluvirheiden osuus	vakavien suunnitteluvirheiden osuus
hallintaliittymä	31%	57%	36%
tunnistus/valtuutus	**62%**	**89%**	**64%**
konfiguroinnin hallinta	42%	41%	16%
salausalgoritmit	33%	93%	61%
tiedon keräys	47%	51%	20%
syötteen tarkistus	**71%**	**50%**	**32%**
parametrien manipulointi	33%	81%	73%
luottamuksellisen tiedon käsittely	33%	70%	41%
istunnon hallinta	40%	94%	79%

http://www.atstake.com/research/reports/atstake_app_unequal.pdf

Löydetyt tietoturvaongelmat ”Top 10”

- istunnon kaappaus/uudelleen ajo
- salasanakontrollit
- puskuriylivuodot
- haavoittuvien tiedostojen ja sovellusten haku
- heikko salaus
- salasanojen kuuntelu
- ”keksien” (cookies) manipulointi
- ylläpitomekanismit
- lokit
- virhekoodit

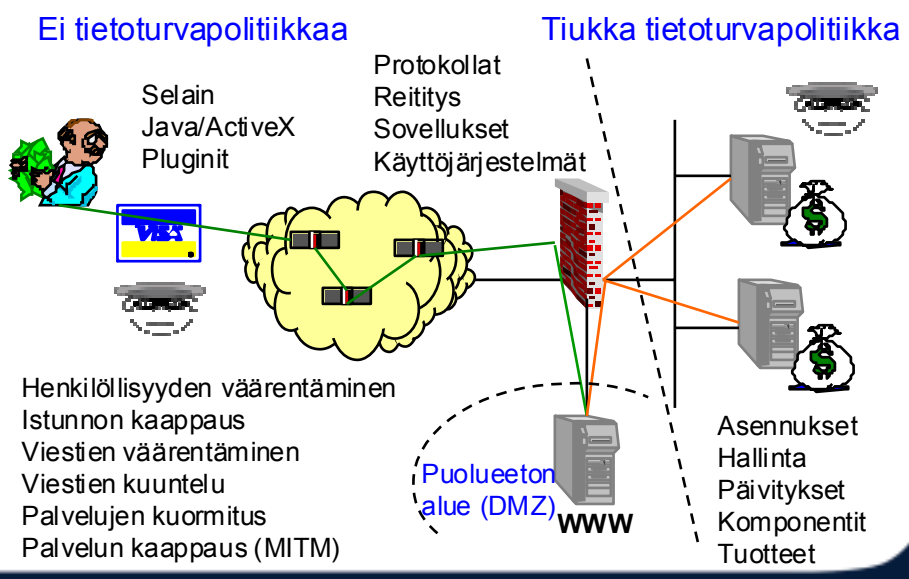
http://www.atstake.com/research/reports/atstake_app_unequal.pdf

Sisältö

- Tietoturva sovelluksissa
- **Sovelluskehityksen haasteet**
- Standardit & ohjeet
- Tietoturvan huomioiminen sovellusprojektissa
- Uusia haasteita
- Esimerkkejä sovellusongelmista

Even if you trust a man not to play certain cards, there's no point in dealing them to him.
Andrew Vachss

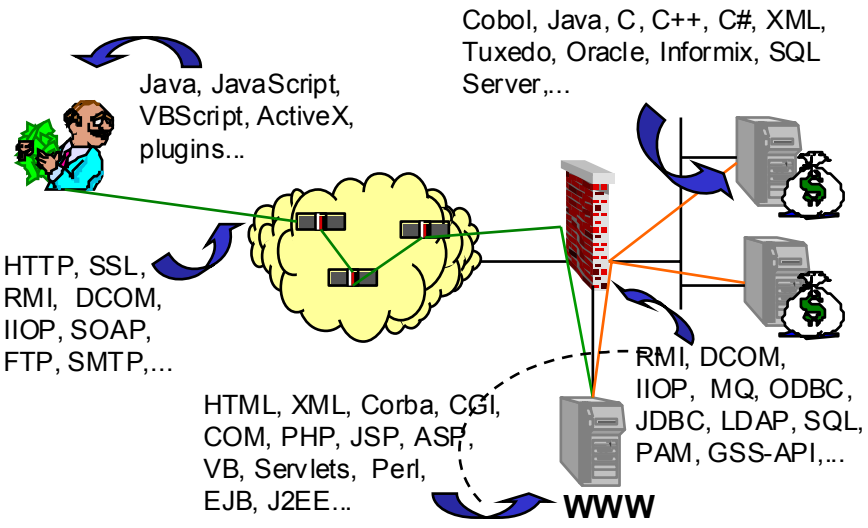
Verkkosovelluksen uhkia



Ratkaistavaa mm.

- Käyttäjätunnistus
- Käyttäjävaltuutus
- Viestien salaus
- Viestien muuttumattomuus
- Viestien aitous
- Sähköiset allekirjoitukset
- Istunnon suojaaminen
- Verkkokomponenttien suojaaminen
- Liitynnät operatiivisiin järjestelmiin
- Loki- ja seurantatietojen keräys
- Palvelujen saatavuus
- Virhetilanteista toipuminen

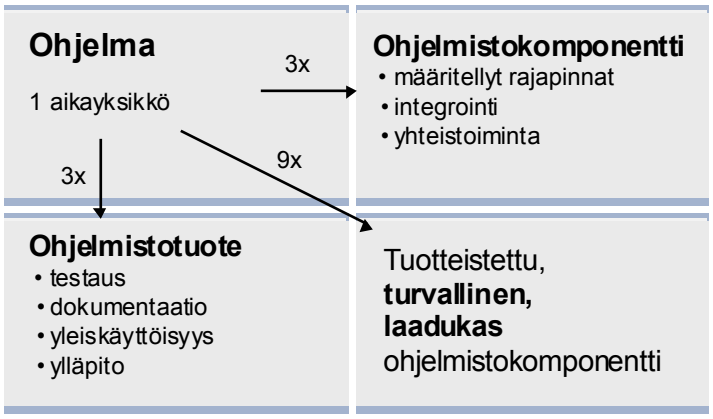
Sovellusnäkökulma



Sovelluskehityksen haasteet

- Kiire, kiire, kiire – “release now, fix it later”
- Kilpailevat standardit
- Välineiden ja protokollien runsaus, nopea kehitys
- Ympäristön vaikeus: hajautettu oliomalli, uudet tekniikat
- Valmiskomponentit – järjestelmä turvallisemmaksi kuin komponenttiensa summa?
- Sovellusten koon kasvaessa turvaongelmien määrä lisääntyy eksponentiaalisesti
- Murphy’s computer (safety) vs. Satan’s computer (security)
- Sovelluksista testataan toiminnallisuutta eikä tietoturvaa => toimiva sovellus ei välttämättä ole turvallinen
- Tietoturva koetaan jälkikäteen lisättävänä piirteenä
- Tietoturva ei ole hauskaa?
- Tietoturvaosaamisen puute

The Mythical Man-Month



Frederick P. Brooks

Sisältö

- Tietoturva sovelluksissa
- Sovelluskehityksen haasteet
- **Standardit & ohjeet**
- Tietoturvan huomioiminen sovellusprojektissa
- Uusia haasteita
- Esimerkkejä sovellusongelmista

"What would be the point of cyphering messages that very clever enemies couldn't break? You'd end up not knowing what they thought you thought they were thinking…"
Terry Pratchett

Standardeja, ohjeita

- TCSEC, ITSEC
- Common Criteria
- BS7799, ISO 17799
- SSE-CMM
- COBIT
- ISF Standard of Good Practice
- GASSP
- The Open Web Application Security Project, www.owasp.org
- Valtionhallinnon suositukset, www.vm.fi
  - tietojärjestelmäkehityksen tietoturvallisuussuositus (2000)
  - tietotekniikkahankintojen tietoturvallisuuden tarkistuslista (2001)

TCSEC / ITSEC

- (Käyttö)järjestelmän turvavaatimusten luokitus
  - TCSEC (Orange Book): D, C1, C2, B1, B2, B3, A1
  - ITSEC (Red Book): E0-E6
- Evaluoinnin kohteena koko järjestelmä laitteineen
- Käytössä lähinnä käyttöjärjestelmätoimittajilla kuvaamassa turvatasoa.
  - Normaalit järjestelmät (NT, W2K, Unix) tyypillisesti nostettavissa C2-tasolle.
  - (erikois) Unix-versioita esim. Trusted Solaris, HP Trusted Linux, Argus PitBull
- Standardista ei juurikaan apua sovellusprojekteille, korkeamman turvatason ”valmispaketeista” kylläkin
  - esim. HP VirtualVault

BS7799 / ISO 17799

- BS7799-1 ja ISO 17799-1: Code of Practice for Information Security Management
- BS7799-2: Specification for Information Security Management Systems
  - Tarkistuslista, BS7799 evaluoinnin perusta
  - Ei vastaavaa ISO-standardia
- Keskittyy tietoturvan hallintaan, ei sovelluskehitykseen
  - tietoturvapolitiikka
  - organisaatio
  - tietojen luokittelu
  - jatkuvuusssuunnittelu
  - jne.

Common Criteria

- Järjestelmän/tuotteen turvatason luokitus
  - Evaluation Assurance Levels: EAL1-EAL6
  - EAL4: menetelmällisesti suunniteltu, testattu ja katselmoitu
- Termejä
  - TOE, Target of Evaluation (kohde)
  - ST, Security Target (tavoite)
  - Assurance Documents (todistelu)
- Muodostetaan turvaprofiili (protection profile), joka kuvaa järjestelmän tietoturvallisuusvaatimukset
- Turvaprofiilin muodostaminen hyödyllistä sovellusprojektia ajatellen
  - tavoitteet, vaatimukset, kontrollit, EAL
  - standardi kuvaustapa, tarkistuslista
  - auditointi

SSE-CMM

- Systems Security Engineering – Capability Maturity Model
- Hyväksytty ISO standardiksi
- Ei kuvaa yhtä prosessia, vaan yleisesti hyväksi havaittuja käytäntöjä
- Laaja
  - kehitys, operointi
  - koko organisaatio
  - ohjelmisto, laitteet, ihmiset, ympäristö
  - sidosryhmät, hankinnat, hallinta
- Auttaa koko prosessin evaluoinnissa ja mittaamisessa
- Sisältää ohjeistuksen prosessien kypsyyden arvioimiseksi
- Isoille organisaatioille, turvatuotteita kehittäville?
- Osaamista huonosti saatavilla

Standardit - yhteenveto

- Standardit lähestyvät tietoturvaa eri näkökohdista
- Ei ole yhtä hyvää ohjetta/standardia, jonka mukaan sovellusprojektin saisi kuntoon tietoturvan huomioimisen osalta
- Käytä standardeja ja ohjeita soveltuvin osin – luo oma standardi
- CC, jos tarvitset tuotteelle virallisen turvasertifioinnin tai haluat määritellä tuotteen turvatavoitteen
- SSE-CMM suunnattu sovellus-/tuotekehitykseen

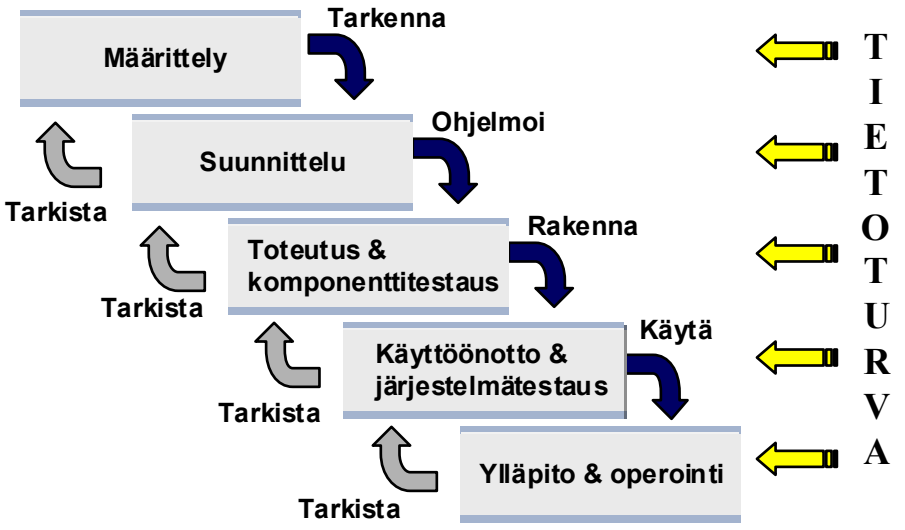
Sisältö

- Tietoturva sovelluksissa
- Sovelluskehityksen haasteet
- Standardit & ohjeet
- **Tietoturvan huomioiminen sovellusprojektissa**
- Uusia haasteita
- Esimerkkejä sovellusongelmista

The coding cowboy's day is done. There was a time when everyone admired the brilliant programmer who worked in self-imposed isolation, creating powerful functions and elegant user interfaces, ingratiating himself with the user community while ignoring his colleagues and his project manager. He had the panache of a rock star and the vanity of a prima donna. Unfortunately, when he rode off into the sunset his code turned out to be undocumented, unextendable, and unmaintainable."

Patricia Ensworth

Vesiputousmalli



Määrittely

atbusiness

Määrittely

Turvallisuus on kompromissi

atbusiness

Mihin halutaan panostaa?

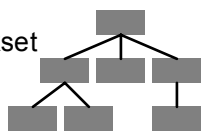
- Toiminnallisuus
  - mitä sovelluksen pitää tehdä?
  - mitä pitää suojata?
- Mekanismi
  - kuinka toiminnot toteutetaan?
  - kuinka suojaukset toteutetaan?
  - valitut/annetut komponentit, algoritmit, middleware,…?
- Toteutus
  - oikea, luotettava, testattu, aikataulussa?
- Käytettävyys
  - tietoturva vs. helppokäyttöisyys?

Määrittely ja tietoturva

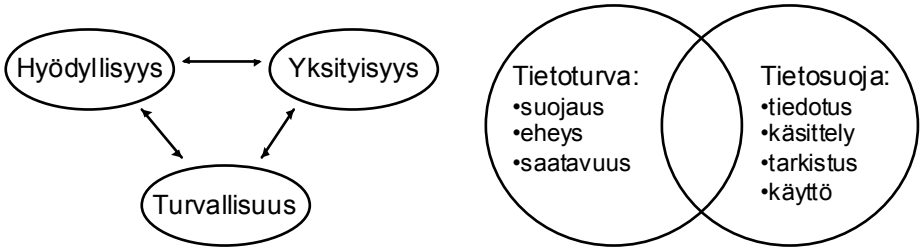
- Suojattavat tiedot
- Tietojen omistajat
- Tietojen arvo
- Riskit, uhka-analyysi
- Olemassa olevat tietoturva-ohjeistukset ja välineet
- Turvallisuusvaatimukset
- Tietosuojavaatimukset
- Kustannusarviot
- Paras, helpoin ja halvin vaihe tehdä tietoturvaratkaisut

Riskien hallinta

- Riskien (tietomurrot, käyttökatkokset) toteutumisesta johtuvat seuraamukset selvitettävä ei-teknisesti: rahan tai maineen menetys, markkinoiden menetys, kilpailukyvyn heikentyminen,…
- Esim. reititin tai sovellus altis hyökkäykselle, mitä se *oikeasti* tarkoittaa yritystoiminnalle?
- Uhka-analyysin kautta suojaukset kohdennetaan oikein ja järkevällä tasolla – ei arvauksiin perustuen
- Riskitaulukot
  - Riski = haavoittuvuus * uhka
  - haavoittuvuus, kustannukset riskin toteutuessa, todennäköisyys, suojaus, suojauksen kustannukset
- Hyökkäyspuut (attack trees)
  - ehdot hyökkäyksen tavoitteen toteutumiseksi
  - AND/OR, vaikeus, kustannukset

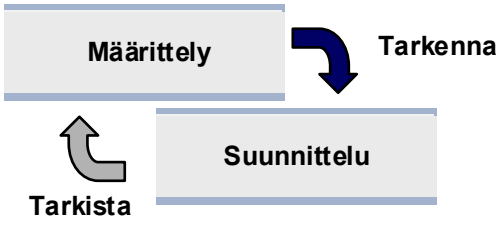


Tietoturva & tietosuoja



- Tietosuojavaatimukset ohjaavat parempiin tietoturvaratkaisuihin
- Lait ja ohjeet eivät anna teknisiä vaatimuksia
  - "tiedot eivät saa paljastua asiattomille"
- Tietoturva on kompromissi – tietosuoja ehdoton?

Suunnittelu



Suunnittelu ja tietoturva

- Suunnittelijoiden koulutus
- Turvakeinojen valinta
- Tietoturva-arkkitehtuuri
- Verkkoarkkitehtuuri
- Sovellusarkkitehtuuri
- Ohjelmointikielten, työkalujen ja tuotteiden tietoturvaominaisuudet (tai puutteet)

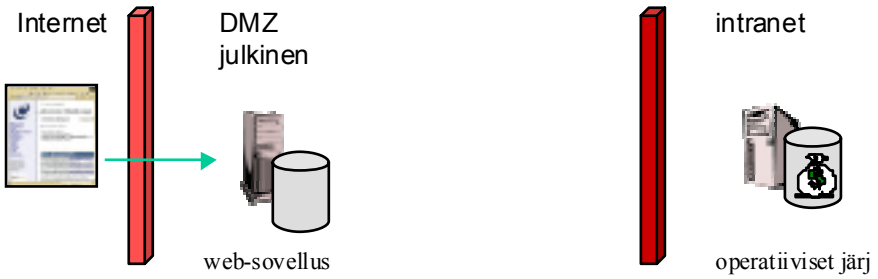
Perusohjeet suunnittelussa

- Suojaa heikoin kohta ensin
- Rakenna useita puolustuslinjoja
- Turvaa virhetilanteet
- Anna sovellukselle mahdollisimman vähän oikeuksia
- Erottele toiminnallisuudet ”turvatiloihin”
- KISS
- Huolehdi yksityisyyden suojasta
- Salaisuuksien pitäminen on vaikeaa
- Luota säästeliäästi
- Käytä testattuja välineitä ja komponentteja – kerää ja jaa kokemuksia

Lisävinkkejä

- Huomioi tietoturvanäkökohdat alusta pitäen
- Tietoturva-asiantuntijoita mukaan projektiin
- Perustellut vaihtoehdot tilaajalle:
  halpa – optimaalinen – erittäin turvallinen
- Tee sovelluksista joustavia ja siirrettäviä
- Eri ohjelmointikielet vaativat erilaista paneutumista turvallisuuteen
- Turvallisuus on prosessi, ei tarkistuslista
- Muista koulutus, tiedostaminen, ohjeistaminen, motivointi
- Koodikatselmointi tietoturvanäkökulmasta
- Koodari siivotkoon jälkensä (fix your own bugs)
- Tietoturvallisuus on laatukriteeri!

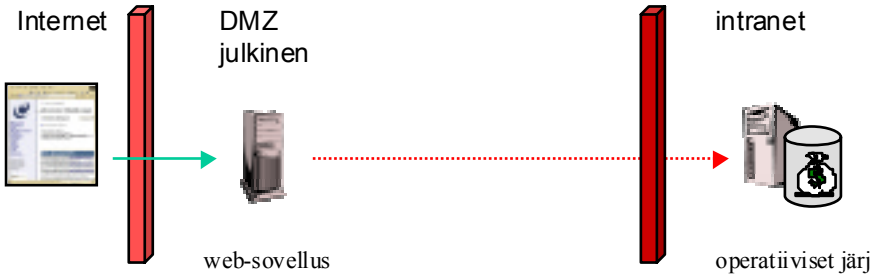
Arkkitehtuurit
stand-alone



+ Ei yhteyksiä sisäverkkoon

- Tietokanta haavoittuva
- Ei sovellu operatiivisia tietokantoja käyttäville sovelluksille

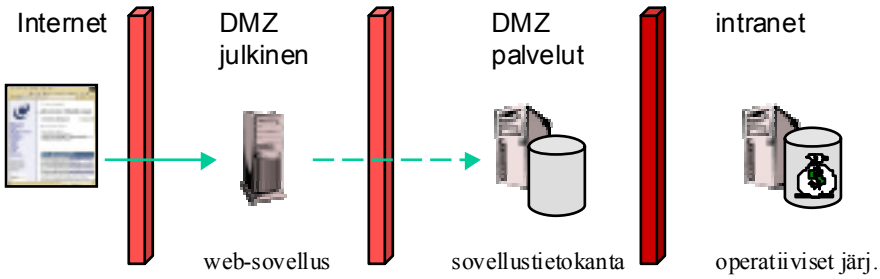
Arkkitehtuurit
tietokanta sisäverkossa



+ Operatiiviset tietokannat käytettävissä

- Tietokanta haavoittuva
- Tietokantayhteyttä voidaan käyttää sisäverkkoon tunkeutumiseen

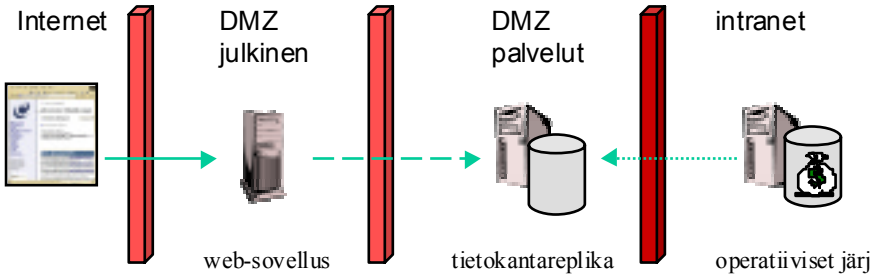
Arkkitehtuurit
erillinen tietokanta



+ Ei yhteyksiä sisäverkkoon

- Tietokanta paremmin suojattu, mutta haavoittuva
- Ei sovellu operatiivisia tietokantoja käyttäville sovelluksille

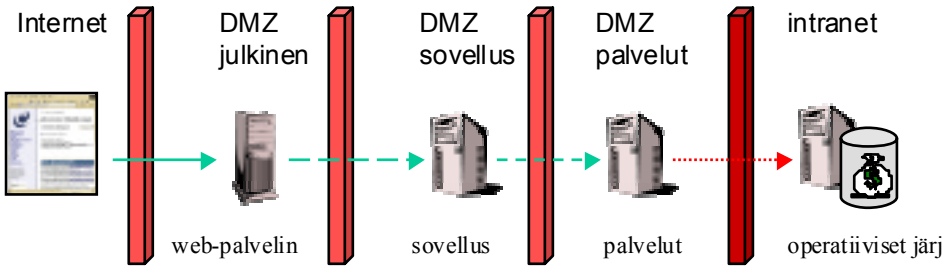
Arkkitehtuurit
replikoitu tietokanta



+ Ei yhteyksiä sisäverkkoon (replikointi ulospäin)

- Tietokanta “tuplattu”
- Monimutkainen
- Ei sovellus reaaliaikaiselle tiedolle tai suurille tietomassoille

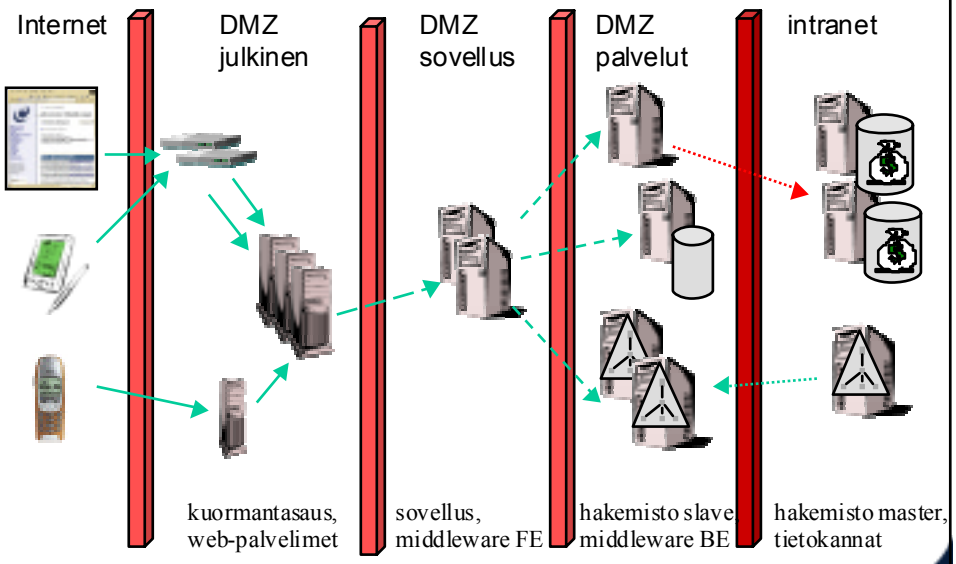
Arkkitehtuurit
monitasoarkkitehtuuri



+ Operatiiviset tietokannat käytettävissä
+ Turvallinen

- Monimutkainen

Arkkitehtuurit
todellisuus



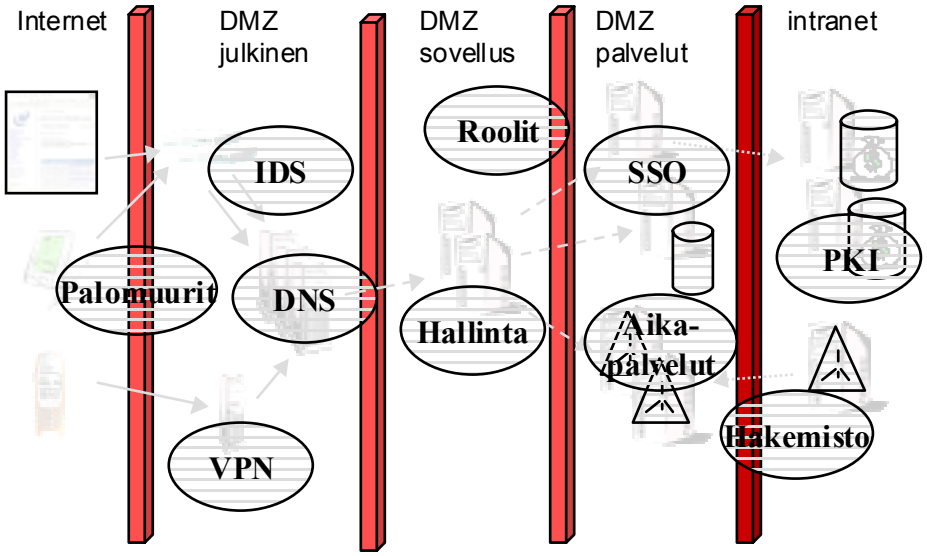
Sovellusten tietoturvapalvelut

- Sovellusarkkitehtuuria tukevien ja suojaavien taustapalvelujen rakentaminen vaatii usein oman (laajan) projektinsa:
  - hakemisto
  - PKI-järjestelmä
  - Single Sign-On
  - VPN
  - keskitetty käyttöoikeuksien hallinta/valtuutus, roolit

Arkkitehtuurit

turvattu todellisuus

atbusiness



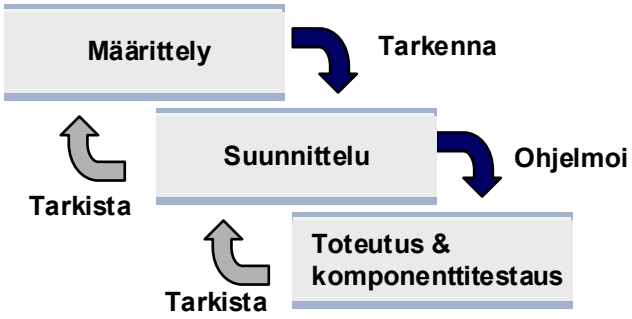
The Fundamental (design) Truths

RFC 1925

atbusiness

- It Has To Work.
- No matter how hard you push and no matter what the priority, you can't increase the speed of light.
- With sufficient thrust, pigs fly just fine.
- Some things in life can never be fully appreciated nor understood unless experienced firsthand.
- It is always possible to agglutinate multiple separate problems into a single complex interdependent solution. In most cases this is a bad idea.
- It is easier to move a problem around than it is to solve it.
- Good, Fast, Cheap: Pick any two
- It is more complicated than you think.
- For all resources, whatever it is, you need more.
- One size never fits all.
- Every old idea will be proposed again with a different name and a different presentation, regardless of whether it works.
- In design, perfection has been reached not when there is nothing left to add, but when there is nothing left to take away.

Toteutus



Toteutus ja tietoturva

- Ohjelmoijien ja testaajien koulutus
- Turvanäkökohdat ohjelmoinnissa
- Valmiskomponentit
- Lokit
- Koodikatselmoinnit
- Tietoturvallisuuden testaus
- Kehitys- ja testiympäristön turvallisuus
- C vs. C++ vs. Perl vs. Java …
- Komponenttien allekirjoitus

Ohjelmointikielistä

- C
  - Brian Kernighan: ”C is a fine scalpel, fit for a surgeon, although, in the hands of the incompetent, it can create a bloody mess.”
  - ”It’s easy to shoot yourself in the foot”
- C++
  - “You accidentally create a dozen instances of yourself and shoot them all in the foot. Providing emergency medical assistance is impossible since you can't tell which are bitwise copies and which are just pointing at others and saying, ‘That's me, over there.’"
- Perl
  - tainted-moodi: ei salli syöttötietojen käyttöä vaarallisissa funktioissa ilman tarkistusta
- Java
  - sandbox-toteutus, tietoturva suunnittelukriteeri
  - monimutkainen tietoturvamalli
  - turvallisuus riippuu JVM-toteutuksesta
  - mikä Java: J2SE, J2EE, J2ME, JavaCard,…?
  - tietoturvalaajennoksia: JAAS, JCE, JSSE

Windows 2000 lisenssi

NOTE ON JAVA SUPPORT

THE SOFTWARE PRODUCT MAY CONTAIN SUPPORT FOR PROGRAMS WRITTEN IN JAVA.

JAVA TECHNOLOGY IS NOT FAULT TOLERANT AND IS NOT DESIGNED, MANUFACTURED, OR INTENDED FOR USE OR RESALE AS ON-LINE CONTROL EQUIPMENT IN HAZARDOUS ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, SUCH AS IN THE OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH THE FAILURE OF JAVA TECHNOLOGY COULD LEAD DIRECTLY TO DEATH, PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE.

Sun Microsystems, Inc. has contractually obligated Microsoft to make this disclaimer

Vinkkejä koodareille

- Tarkista aina sovelluksen/komponentin ulkopuolelta tulevat tiedot
  - Älä luota käyttäjään
  - Älä luota client tai server-sovellukseen
  - Älä luota alla olevaan järjestelmään
  - Älä luota systeemin tilaan
- Älä pidä luottamuksellista tietoa salaamattomana tiedostoissa tai puskureissa (esim. salasanat)
- Tekeehän ohjelma vain ja vain sen mitä pitää?
- Mieti itse, kuinka sovelluksen voisi rikkoa
- Aseta aina turvalliset oletusarvot!

Testauksesta

- Testattavuus huomioitava sovellusta kehitettäessä
- Koodikatselmointi
- Arvioi komponenttien rajapintoihin liittyvät riskit
- Arvioi suunnittelijan/koodaajan olettamukset ja riko ne
- Käytä omaa testi-clientia (perl)
- Testaa normaalia sovellusrajapintaa alemmalla tasolla
- Huijaa
- Käytä aputyökaluja mm. vaarallisten funktioiden löytämiseksi, satunnaissyötteen generoimiseksi, virhetilanteiden aiheuttamiseksi,…
- Testisovellusten tulee olla laadukkaita

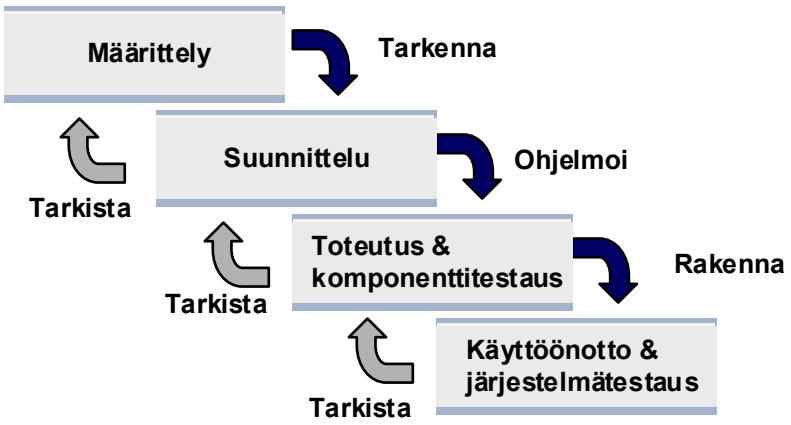
Lähdekoodin analysointi

- Flawfinder (www.dwheeler.com/flawfinder)
  - C/C++
- RATS (www.securesw.com/rats/)
  - C, C++, Python, Perl and PHP
- ITS4 (www.cigital.com/its4/)
  - C & C++
- Man Machine System (www.mmsindia.com) (€)
  - "Automated Java Testing and Quality Assurance Tools for the Serious Java Professional!"
- Rational Purify (www.rational.com) (€)
  - Java & C/C++
- Jtest (www.parasoft.com) (€)
  - Java

Sovellustason analysointi/testaus

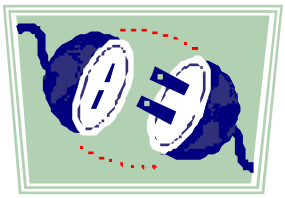
- Perl
- Netcat (http://www.atstake.com/research/tools)
- cURL(http://curl.haxx.se/)
- Nessus (www.nessus.org)
- Ethereal (www.ethereal.org)
- Nessus (www.nessus.org)
- Nikto (www.cirt.net/code/nikto.shtml)
- HTTPush (www.sourceforge.net/projects/httpush/)
- Spike (www.immunitysec.com/spike.html)
- WebProxy (http://www.atstake.com/research/tools)
- WebSleuth (http://www.geocities.com/dzzie/sleuth)
- Sanctum AppScan (www.sanctuminc.com) (€)
- spiDYNAMICS WebInspect (www.spidynamics.com) (€)
- Hailstorm (www.cenzic.com) (€)

Käyttöönotto



Käyttöönotto ja tietoturva

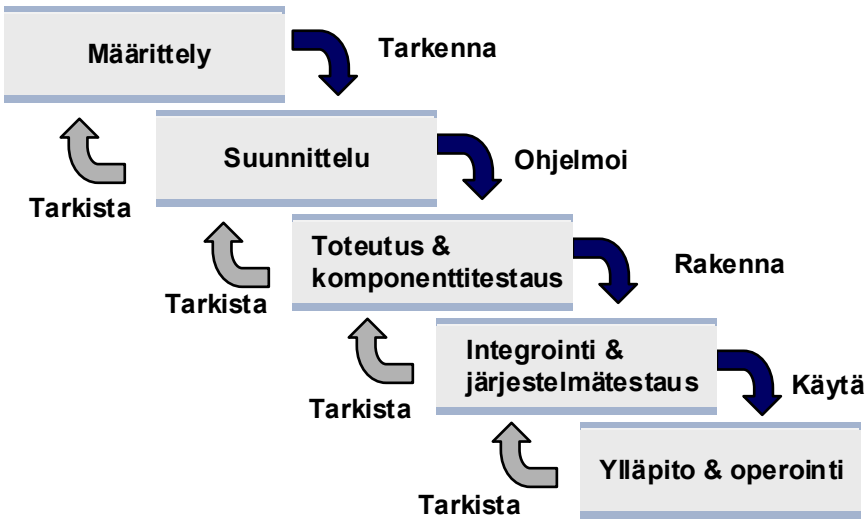
- Verkko- ja järjestelmäasiantuntijoiden koulutus
- Asennussuunnitelmat ja –dokumentit
- Turvalliset asennukset
- Liittymäkomponenttien konfigurointi
- Järjestelmän auditointi
- Käyttöönottokoulutus
- Kokonaisuuden testaus
  - suorituskyky
  - vikasietoisuus



Tietoturvatason mittaaminen

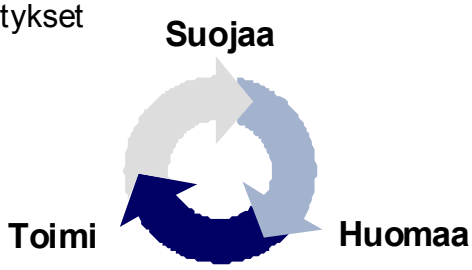
- Hankalaa
  - sovellusten tietoturvatarpeet vaihtelevat
  - riskinsietokyky vaihtelee
  - tekniset vs. hallinnolliset ratkaisut
- Koodikatselmoinnit
- Auditoinnit
  - speksit
  - arkkitehtuuri
  - asennukset
  - koodi
- Kuinka verrata eri projektien tietoturvatasoa?
- Kannattaa keskittyä omien prosessien ja menetelmien virittämiseen ja virheistä oppimiseen
- Arviointi suhteessa määritykseen

Ylläpito



Ylläpito ja tietoturva

- Ylläpidon ja tukihenkilöiden koulutus
- Järjestelmän turvallisuuden seuranta
- Turvakorjaukset
- Sovelluspäivitykset
- Muutoshistoria
- Lokien seuranta, hälytykset
- Reagointi



Tietoturvan liittäminen sovelluskehitysprosessiin

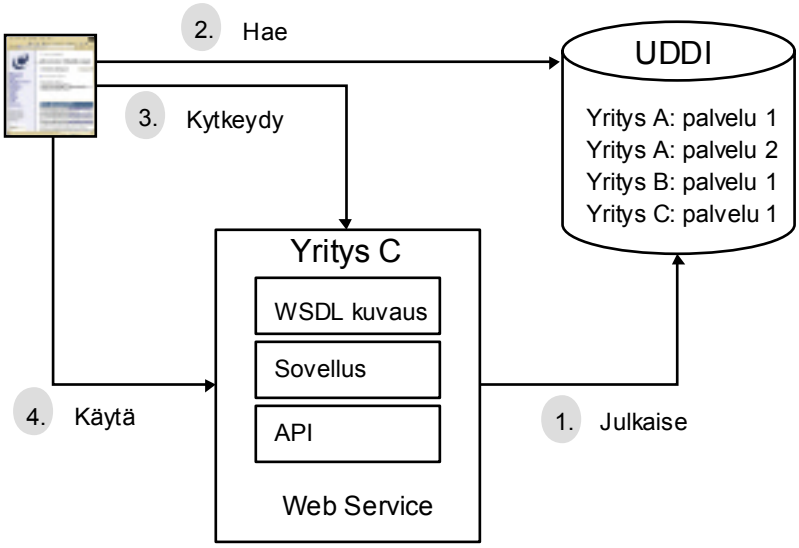
- Vastuu tietoturvasta kaikilla: projektipäällikkö, suunnittelija, speksaaja, koodaaja, testaaja, asentaja, ylläpitäjä, tukihenkilö – tietysti johdon tukemana.
- Nimitä tietoturvaan erikoistuva henkilö joka osa-alueelle
- Kouluta
- Pienin askelin eteenpäin
- Aloita määrittelystä, testauksesta ja turvallisista asennuksista
- Aluksi mukaan yleisesti tunnetut ja käytössä olevat turvaratkaisut
- Poimi standardeista ja ohjeista sopivat kohdat – sovella
- Vuosien työ

Sisältö

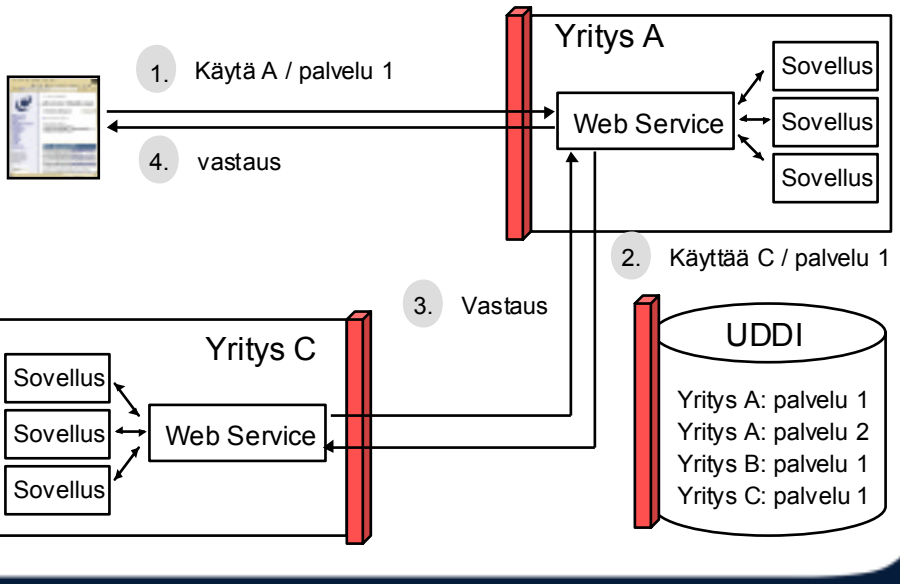
- Tietoturva sovelluksissa
- Sovelluskehityksen haasteet
- Standardit & ohjeet
- Tietoturvan huomioiminen sovellusprojektissa
- **Uusia haasteita**
- Esimerkkejä sovellusongelmista

"Now, here, you see, it takes all the running you can do just to stay in the same place."
Red Queen in Alice in the Wonderland

Web Services 1/3



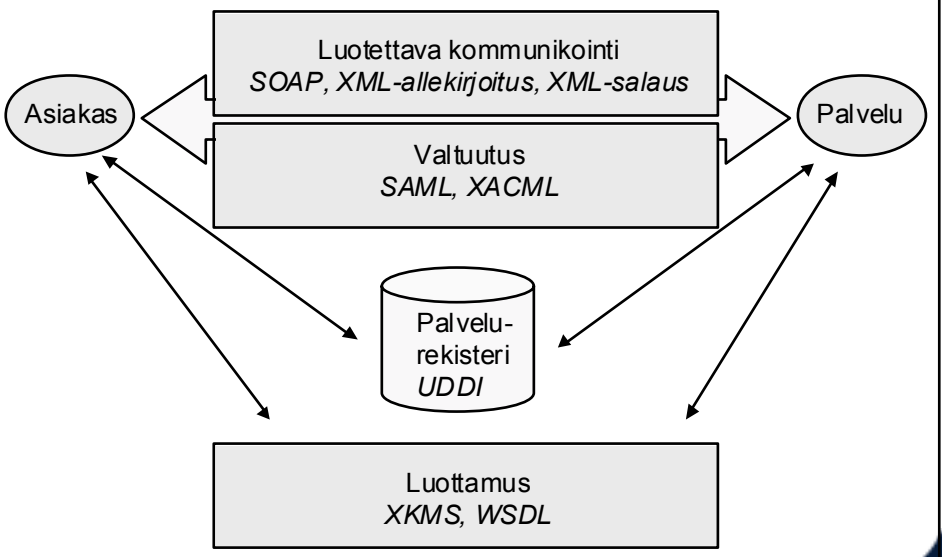
Web Services 2/3



Web-services 3/3

- SOAP
  - XML-pohjainen viestinvälitys
  - HTTP, SSL
- WSDL
  - XML-pohjainen verkkopalvelujen kuvaustapa
- UDDI
  - Web-palvelujen mainostus ja hakuprotokolla
- DISCO
  - Web-palvelujen (helppo, MS-spesifi) julkaisutapa

Web Services tietoturva 1/3



Web-services tietoturva 2/3

- XML-allekirjoitus
  - XML dokumentin tai sen osan digitaalinen allekirjoitus
  - dokumenttiin linkitetyn tiedon allekirjoitus
  - XML-muoto vs. käyttäjälle esitetty muoto
  - kanonisen (canonical) XML:n allekirjoitus
- XML-salaus
  - XML dokumentin tai sen osan salaus
- XACML
  - XML-pohjainen pääsylista (ACL)
  - tekijä, kohde, toiminto (luku, kirjoitus, luonti, tuhoaminen)
  - kohde jopa XML-dokumentin yksi elementti

Web-services tietoturva 3/3

- SAML
  - tunnistus, valtuutus, oikeudet
  - sovellusten ja yritysten välinen oikeustietojen välitys
- XKMS
  - XML-pohjainen PKI
  - X-KRSS: XML Key Registration Service Specification
  - XKISS: XML Key Information Service Specification
  - palvelut: avainten generointi, salaus, allekirjoitus, varmenteen haku ja käsittely, luottamusketjun käsittely,…
  - PKIX, SPKI, PGP

Muuta mielenkiintoista

- Open Source
- .NET
- Biometria

Sisältö

- Tietoturva sovelluksissa
- Sovelluskehityksen haasteet
- Standardit & ohjeet
- Tietoturvan huomioiminen sovellusprojektissa
- Uusia haasteita
- **Esimerkkejä sovellusongelmista**

"Today's computer and network security mechanisms are like the walls, moats, and drawbridges of medieval times. At one point, effective for defending against isolated attacks, mounted on horseback. Unfortunately, today's attackers have access to airplanes and laser-guided bombs!"

Gary McGraw

Tunnettuja ongelmia…

- Puskuriylivuodot
- Ajoitusongelmat (race condition, TOCTOU)
- Satunnaisuuden luominen
- Salausprotokollat
- Luottamussuhteet
- Cross-site scripting (XSS)
- SQL injection
- Istunnon kaappaus
  - HTML "piilotetut" kentät
  - salaamattomat cookiet
- Vanhojen tai backup-versioiden unohtaminen
- Paljastavat virheilmoitukset

Tunnettuja ongelmia…

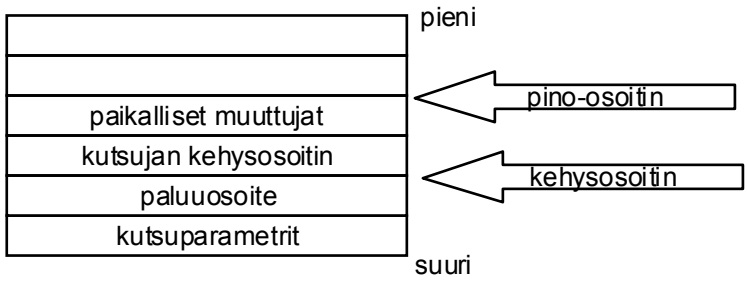
- Parametrien manipulointi
- Luottamuksellista tietoa ”vuodetaan”
  - Kirjoittamalla lokiin (HTTP GET)
  - SSI-tiedostoissa
  - Web-palvelimen tiedostoissa ja tiedostojen näkymistä ei ole estetty
  - HTML-sivun kommenteissa
  - ASP- tai JSP-sivuilla
    - `http://www.my.com/page.asp.`
    - `http://www.my.com/page.asp::$DATA`
    - `http://www.my.com/page.JSP` (WebLogic, WepSphere)
- Linkkien ”luova” käyttö
  - `http://host/something.php=<b>Hi%20mom%20I'm%20Bold!</b>`
  - `http://host/scripts/something.asp=../../WINNT/system32/cmd.exe?dir+e:\`
  - `http://host/files.asp%00.jpg`
- UTF8-merkistön tai ”%-merkistön” unohtaminen
  - `http://foo.com/cgi?file=%2F%65%74%63%2F%70%61%73%73%77%64`

Puskuriylivuoto…

- Erittäin yleinen ongelma
  - Cert-varoituksista lähes puolet
- Tiedolle on varattu säilytystilaa, mutta talletettavan tiedon määrää ei tarkisteta
- Pinon (stack) tai kasan (heap) ylivuoto
  - Pinon ylivuoto helpompi toteuttaa
- Oireet
  - Sovellus käyttäytyy “oudosti”
  - Sovellus “kaatuu”
  - Ei mitään (vaikea löytää testauksessa)
- Seuraukset
  - Kriittisen (esim. tietoturvaan liittyvän) tiedon ylikirjoitus
  - Ylimääräisen ohjelmakoodin suoritus

Puskuriylivuoto…

- X86 arkkitehtuurin ohjelmapino



- Paikallisten muuttujien ylivuoto mahdollistaa…
- Paluuosoitteen ylikirjoituksen, joka mahdollistaa…
- Ohjelmasuorituksen jatkumisen halutusta paikasta, joten…
- Voidaan suorittaa ylivuodon sisältämä ohjelmakoodi

Puskuriylivuoto

Sovelluksessa :

```
char str[10];
strcpy ( str, argv[1] );
```

Ajetaan ohjelma siten, että

```
argv[1] =NNNNNNNNNNNNSSSSSSSSSRRRRRRRRRRRRRRR
```

N = NOP

S = Shellcode (haluttu “murtokoodi”)

R = Shellcode aloituskohta tai jonkin NOP-komennon osoite; R-osion pitää ylikirjoittaa funktion paluukoodi

Korjaus:

```
strncpy(str,agv[1],10);
```

Format String Attack

- **Oikein**: `printf(format, str);`
- **Väärin**: `printf(str);`
- Väärässä tavassa voidaan mahdollisesti syöttää sovellukselle merkkijono, joka sisältää myös (vääriä) formaatteja (%n)
  - Ylikirjoitetaan (korvataan) ohjelman suorittama komento
  - Ylikirjoitetaan paluuosoitteen tilalle haluttu (väärä) osoite

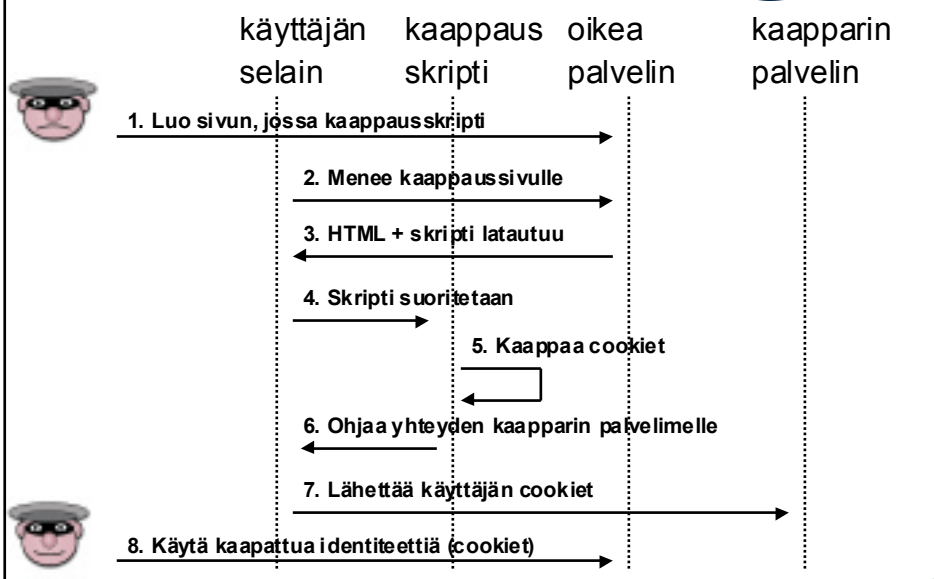
Cross-site Scripting (XSS)

- Mahdollista, jos web-palvelu antaa käyttäjän syöttää dataa muiden käyttäjien katsottavaksi
- Käyttäjä syöttääkin tekstikenttään esim. skriptin
- "Tietoa" katsovan käyttäjän selain suorittaa skriptin
  - Haitallisen skriptin suorittaminen käyttäjän selaimessa
  - Identiteetin (cookie) kaappaus
  - Käyttäjän yhteyden siirtäminen väärään osoitteeseen

```
www.site.com/search.pl?text=security

www.site.com/search.pl?text=<script>alert(document.cookie)</script>
```

Cross-site Scripting identiteetin kaappaus



SQL Injection

Haetaan tietokannasta käyttäjän hakuehdolla:

```
Set myRecordset = myConnection.execute
("SELECT * FROM myTable WHERE
someText ='" & request.form("inputdata") & "'")
```

Käyttäjä antaakin “hakuehdoksi”:

```
'exec master..xp_cmdshell 'net user test testpass /ADD' --
```

Ja koodissa käykin näin:

```
SELECT * FROM myTable WHERE someText =''
exec master..xp_cmdshell 'net user test testpass /ADD'--'
```

Vastuu siirtyy käyttäjälle?

- Voidaanko käyttäjän vastuulle antaa tietoturvapäätösten tekeminen?
  - Koodin lataus työasemaan?
  - Palvelinvarmenteen hyväksyminen?
  - Allekirjoituksen hyväksyminen?
  - Salausalgorimin valinta?
  - Sormenjälkitietojen tarkistus?
- Käyttäjältä ei pidä edellyttää tietoturva-asiantuntemusta!
- Kuinka moni tietoturvaexpertti (saati sovelluskehittäjä) oikeasti ymmärtää esim.
  - HTTP-protokollan toiminnan
  - SSL:n toiminnan?
  - Salausalgoritmien erot?
  - Sähköisen allekirjoituksen vaatimukset?
  - XML:n

Tietoturva on laatua