Aalto-yliopiston teknillinen korkeakoulu
Tietotekniikan osasto
Tietotekniikan tutkinto-ohjelma

# WWW-palvelujen tietoturva

Kandidaatintyö

21.12.2010

Maksim Luzik

AALTO-YLIOPISTO
TEKNILLINEN KORKEAKOULU
Informaatio- ja luonnontieteiden tiedekunta
Tietotekniikan tutkinto-ohjelma

KANDIDAATINTYÖN
TIIVISTELMÄ

| | | |
|---|---|---|
| **Tekijä:** | Maksim Luzik | |
| **Työn nimi:** | WWW-palvelujen tietoturva | |
| **Päiväys:** | 21. joulukuuta 2010 | **Sivumäärä:** 32 |
| **Pääaine:** | Mediatekniikka | **Koodi:** T610-2 |
| **Vastuuopettaja:** | prof. Ilkka Niemelä | |
| **Työn ohjaaja:** | TkL Sanna Suoranta | |

Tämä työ käsittelee www-palveluiden perustoimintaa, tietoturvaratkaisuja ja yleisempiä hyökkäyksiä palveluita vastaan. Työssä käydään läpi myös HTTP-protokollan toimintaa ja tutustutaan pintapuolisesti TCP/IP-protokollapinon yhteyskäytäntöön. Työssä ei käsitellä muita tiedonsiirtoprotokollia tai sovelluksia.

Verkkopalvelujen määrä kasvaa jatkuvasti. Monet ohjelmat ovat käytettävissä verkossa selaimen kautta tai ainakin tukevat jonkinlaista verkkopalveluiden integraatiota. Kun palveluiden määrä verkossa kasvaa, lisääntyy myös uhat ja hyökkäykset palveluita vastaan.

Tietoturva on jokaisen järjestelmän kriittinen komponentti. Verkkohyökkäykset kehittyvät jatkuvasti ja huijarit keksivät uusia tapoja manipuloida uhreja. Mikään järkevä tietoturvaratkaisu ei välttämättä takaa täyttä suojaa kaikilta uhilta. Ratkaisut voidaan kuitenkin toteuttaa niin, että pystytään suojautumaan yleisimmiltä hyökkäyksiltä. Pahimpia virheitä voi oppia välttämään etukäteen, jottei tarvitse kokea ikäviä tilanteita kantapään kautta.

| | |
|---|---|
| **Avainsanat:** | internet, tietoturva, www-palvelut, salausmenetelmät, hyökkäykset |
| **Kieli:** | Suomi |

## Käytetyt lyhenteet

| | |
|---|---|
| 3-DES | Triple-DES; DES-salausmenetelmän parannettu versio |
| AES | Advanced Encryption Standard; toistaiseksi murtumaton salausmenetelmä |
| DDOS | Distributed Denial of Service; Hajautettu palvelunestohyökkäys. Ero tavalliseen palvelunestohyökkäykseen on enemmän kuin yksi hyökkäyksen lähde. |
| DES | Data Encryption Standard; tunnetuin salausmenetelmä, joka on nykytietokoneilla heikko ja murrettavissa |
| DNS | Domain Name System; nimipalvelujärjestelmä |
| DOS | Denial of Service; Palvelunestohyökkäys |
| FTP | File Transfer Protocol; Tiedostonsiirtoprotokolla |
| HTTP | Hypertext Transfer Portocol; Hypertekstin siirtoprotokolla |
| HTTPS | Hypertext Transfer Protocol Secure; Laajaan käyttöön levinnyt turvallinen hypertekstin siirtoprotokolla, käyttää TLS:ää |
| IETF | Internet Engineering Task Force; Internetin protokollien standardoinnista vastaava organisaatio |
| MySQL | MySQL on suosittu SQL-tietokannan hallintajärjestelmä |
| PHP | Hypertext Preprocessor; Hyvin suosittu ohjelmointikieli, joka aikaisemmin tunnettiin nimellä Personal Home Pages |
| RSA | Ron Rivestin, Adi Shamirin ja Len Adlemanin kehittämä julkisen avaimen salausmenetelmä |
| S-HTTP | Secure Hypertext Transfer Protocol; Vaihtoehtoinen turvallinen hypertekstin siirtoon käytetty ratkaisu, joka ei saavuttanut samanlaista suosiota kuin HTTPS |

| | |
|---|---|
| SQL | Structured Query Language; standardoidu kyselykieli, joka on yleisessä käytössä tietokannan hallintajärjestelmissä. |
| TCP/IP | Transmission Control Protocol/Internet Protocol; Protokollapari, joka on käytössä verkossa. Vastaa pakettien siirrosta verkon ylitse, mm internetissä. |
| TLS | Transport Layer Security; Yleinen tiedon salaamiseen ja purkuun käytetty protokolla |
| UDP | User Datagram Procotol; Protokolla eroaa TCP-protokollasta sen yksinkertaisuutensa vuoksi. Toisin kuin TCP-protokolla, UDP ei missään vaiheessa tarkista, saapuiko paketti perille vai ei. |
| WWW | World Wide Web; Laaja internet-verkossa toimiva hypertekstijärjestelmä |

## Sisältö

# 1 Johdanto

Yhä useammin yritykset siirtävät tai tekevät rinnakkaispalveluja verkkoon. Monet työpöytäohjelmistot ovat käytettävissä suoraan selaimen kautta Internetissä. Maailmanlaajuisesti on tarjolla isoja määriä kriittisiä palveluita kuten verkkopankkeja, sähköpostipalveluita ja laajat yhteisöpalvelut, jotka tarvitsevat hyvän tietoturvan toimiakseen niille asetettujen vaatimusten mukaisesti.

Monet tietoturvaratkaisut on toteutettu puutteellisesti. Hyökkäyksen sattumista omalle kohdalle pidetään epätodennäköisinä. Vasta sen jälkeen, kun vahinko on tapahtunut, kiinnitetään huomiota tietoturvaan.

Tämän työn tarkoituksena on tutkia erilaisia menetelmiä ja ratkaisuja, joiden avulla voidaan parantaa WWW-palvelun tietoturvaa. Tarkoituksena on myös käydä läpi tavanomaisia hyökkäyksiä ja huijausyrityksiä, sekä ymmärtää paremmin, miten näiltä voidaan suojautua. Lisäksi tavoitteena on tuoda esille myös tietoturvan yleistä tietoa ja nostaa sen tärkeyttä enemmän esille, erityisesti www-palveluiden tekijöiden keskuudessa.

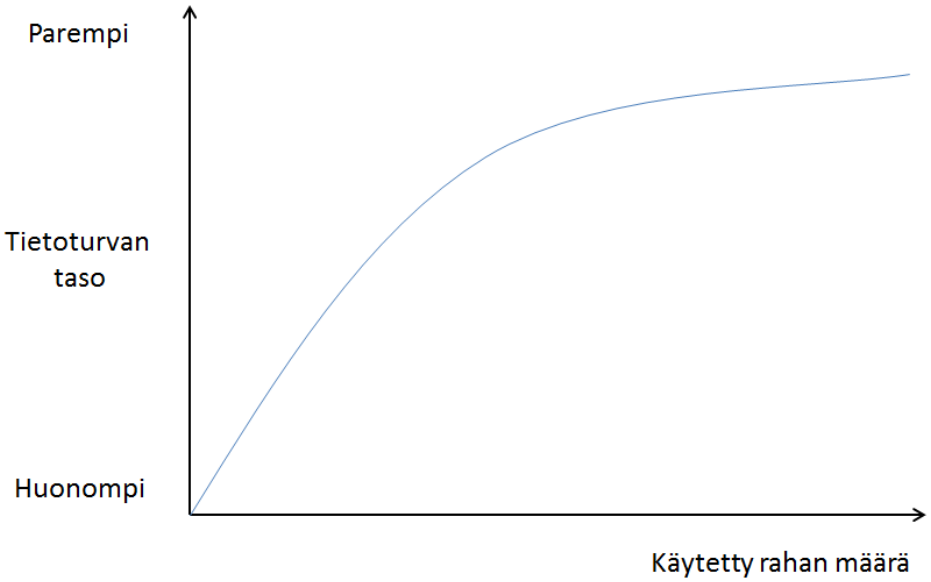
Tutkielmassa keskitytään WWW-sivustojen ja WWW-palvelujen tietoturvaan. Sivustojen ja palveluiden ratkaisuja käsitellään ainoastaan sen verran, että voidaan ymmärtää paremmin niiden toimintaa tietoturvan kannalta. Lisäksi tutustutaan tarkemmin HTTP- ja HTTPS-protokollaan. Tutkielma ei käsittele sähköpostin tai muiden tietoliikenneratkaisujen tietoturvaa, kuten esimerkiksi FTP-protokollaa.

Toisessa luvussa käsitellään tietoturvaa yleisellä tasolla ja tutustutaan keskeisimpiin käsitteisiin. Kolmannessa luvussa käydään läpi WWW-palveluiden ominaisuuksia ja niiden käyttämiä protokollia. Neljännessä luvussa syvennetään salauksen menetelmiin ja viidennessä luvussa käsitellään verkkohyökkäyksiä ja verkkohyökkäysten vastaisia suojausmenetelmiä ja turvatoimia.

## 2 Tietoturva

Tietoturva jaetaan yleensä kolmeen tekijään: saatavuuteen tai käytettävyyteen (engl. availability), luottamuksellisuuteen (engl. confidentiality) ja eheyteen (engl. integrity). Vaikka saatavuus tai käytettävyys voi kuulostaa hieman omituiselta, se on yhtä tärkeä ominaisuus tietoturvassa siinä missä muutkin. Saatavuudella varmistetaan se, että tietoon tai palveluun on pääsy niillä, joilla on siihen käyttöoikeus. Toisin sanoen se on saatavilla, kun sitä tarvitaan. Eristämällä palvelu koko yhteiskunnasta saadaan hyvin turvallinen ratkaisu, mutta sen hyöty on kuitenkin olematon, jos siihen ei ole minkäänlaista pääsyä. Saatavuudella tarkoitetaan myös palvelun toimintavarmuutta. Palvelun pitää olla saatavana, vaikka siihen kohdistuisikin erilaisia hyökkäysyrityksiä. Luottamuksellisuus puolestaan määrittelee oikeudet tietoon tai palveluun. Tietoa voi lukea tai muokata vain ne tahot, joilla on siihen oikeus. Esimerkiksi palvelua pääsevät käyttämään ainoastaan ne henkilöt, joilla on siihen oikeuttava tunnus. Eheys vastaa siitä, että tieto on sitä mitä sen pitäisi olla. Tieto ei siis saa muuttua mitenkään tahtomatta, esimerkiksi hyökkäyksessä. Jos tieto kuitenkin muuttuu, muutos ja itse muutoksen lähde pitää voida havaita.

Kehittäessä uutta www-sivustoa tai -palvelua on hyvä huomioida tietoturvan lisäksi myös kustannukset. Tietoturvaa voi parantaa ja kehittää jatkuvasti, sillä jopa pitkään aikaan tietoturvallisena pidetyssä palvelussa tai ratkaisussa voi paljastua vuosien päästä haavoittuvuus. Niin kauan kuin ihminen on luomassa uusia ratkaisuja ja palveluja, täydellistä tietoturvallisuutta ei voida taata. Tietoturvallisuuden tason ja uponneiden kustannusten välillä voi kuitenkin hahmottaa ”tasapainopisteen”. Esimerkiksi omalle kotisivulle ei välttämättä ole ajallisesti eikä rahallisesti kannattavaa hankkia maksullista todennussertifikaattia ja kirjautumisjärjestelmää tai edellyttää salausmenetelmän käyttöä yhteyden suojaamiseen, jos tarkoituksena on vain esitellä omat harrastuksensa. Alla oleva kuva havainnollistaa tilannetta paremmin.




**Kuva 1: Tietoturvan kustannus ja saavutettu taso (Alestalo, 2010)**

Keskeisiä tietoturvaan liittyviä käsitteitä ovat (Aura, 2010):

- uhka (engl. threat) – ikävä tapahtuma, joka voi tapahtua
- hyökkäys (engl. attack) – tahallaan aiheutettu ikävä teko
- hyväksikäyttö (engl. exploit) – täytteenpantu hyökkäys
- haavoittuvuus (engl. vulnerability) – heikkous tietojärjestelmässä, joka mahdollistaa hyökkäyksen
- riski (engl. risk) – hyökkäyksen todennäköisyys ja sen aiheuttaman vahingon kustannus

Canalin (2005) mukaan tietoturvaa ei voi ajatella läsnä olevana asiana, vaan se on pikemminkin poissa oleva tai tiedottamaton asia. Niin kauan kun ei ole sattunut mitään vahinkoja, ajatellaan oltavan puhtailla vesillä. Tietoturva on siis kontekstisidonnainen: kun ei ole mitään uhkaa, voi olla turvassa, vaikka ei suojautuisikaan mitenkään.

# 3 WWW-palvelut

WWW-palvelut tulevat varmasti lisääntymään entisestään. Ne ovat helposti saatavilla ja käytettävissä joka laitteella, jolla on Internet-yhteys. Palvelujen toteuttajan ei tarvitse välittää laitteiden käyttöjärjestelmistä tai fyysisistä kokoonpanoista, vaan selaimet toimivat palvelun ja käyttäjän välisenä rajapintana.

Palvelun toteutustapoja on suuri määrä. Ohjelmointikieliä ja tietokantoja on laidasta laitaan. Suuri valinnanvara lisää kehittäjien määrää ja innokkuutta tehdä palveluita, mutta toisaalta aiheuttaa tietoturva-asiantuntojoille lisää työmäärää ymmärtää jokaisen järjestelmän heikot kohdat.

## 3.1 WWW-palvelujen ominaisuudet

Toisin kuin paikallinen ohjelmisto työkoneella, WWW-palvelut ovat aina saatavilla verkossa. Niihin on helppo ottaa yhteyttä mieleisen selaimen kautta. Lisäksi WWW-palveluiden toiminta voidaan helposti jakaa eri palvelimien välille. Yksi palvelin voi esimerkiksi luoda käyttäjälle ainoastaan palvelun käyttöliittymän rajapinnan ja toinen palvelin puolestaan voi toimia tietokantana. Palvelun voi skaalata käyttäjämäärän ja vaatimusten mukaisesti, sillä palvelimia voi lisätä tarpeen mukaan.

Valitettavasti verkkoympäristö aiheuttaa usein myös uhkia. Verkkopalveluihin voidaan helposti kohdistaa esimerkiksi palvelunestohyökkäys (engl. Denial of Service). Palvelunestohyökkäyksellä pyritään lamauttamaan palvelu, jolloin palvelun käyttäjät eivät pääse siihen käsiksi. Tästä aiheutuu suuria kuluja palvelun tuottajalle ja harmaita hiuksia käyttäjille.

Toisaalta verkosta on myös paljon hyötyä. Palveluja voidaan ylläpitää helpommin. Esimerkiksi uuden haavoittuvuuden voi paikata suoraan palvelimella ilman, että käyttäjän tarvitsee asentaa erikseen mitään ohjelmistopäivitystä (ellei kyseessä ollut selaimen haavoittuvuus). Käyttäjiltä jäävät usein asentamatta tavallisten ohjelmistojen tietoturvapäivitykset, jolloin käyttäjä on silti haavoittuvainen, vaikka tietoturvapäivitys olisi julkaistu.

Näiden positiivisten ominaisuuksien takia verkkopalvelut ovat hyvin suosittuja ratkaisuja yritysten keskuudessa. Pienet kustannukset ja suhteellisen helppo integrointi yrityksen nykyisiin tietoratkaisuihin on suuri yllyke (Papazoglou, 2008 s. 4).

## 3.2 WWW-palvelujen käyttämät protokollat

Kaikki WWW-palvelut perustuvat HTTP-protokollaan (engl. Hypertext Tranfer Protocol), joka puolestaan toimii TCP/IP-protokollapinon päällä (engl. Transmission Control Protocol ja Internet

Protocol). IP-protokolla on pakettipohjainen protokolla, joka vastaa pakettien lähettämisestä verkossa. Koko internet on rakennettu IP-protokollan päälle. IP-protokolla yhdistää kaikki verkossa olevat koneet IP-osoitteiden avulla. IP-osoite on uniikki numeroyhdistelmä, joka yksilöi jokaisen tietokoneen tai laitteen verkkorajapinnan.

TCP-protokolla on tietoliikenneprotokolla, joka toimii IP-protokollan päällä. TCP on luotettava protokolla, jonka avulla voidaan varmistaa pääsikö lähetetty paketti määränpäähänsä. TCP huolehtii myös siitä, että paketit saapuvat oikeassa järjestyksessä. TCP-protokolla vastaa siis yhteyksistä kahden osapuolen välillä. IP-protokolla ei tiedä verkkoyhteyksistä mitään ja toimii ainoastaan pakettien välittäjänä. Vaikka TCP-protokolla huolehtii siitä, että paketit tulevat perille, se ei kuitenkaan ole luotettava protokolla tietoturvamielessä. Esimerkiksi kahden osapuolen välillä käydyn keskustelun voi helposti lukea henkilö, jolla on riittävä osaaminen ja fyysinen pääsy verkkoon.

## 3.3 HTTP-protokolla

HTTP-protokolla, eli hypertekstin siirtoprotokolla (engl. Hypertext Transfer Protocol) on sovellustason protokolla, jonka pääidea on hypermedian siirto ja joka toimii TCP-protokollan päällä. Se on tilaton protokolla, mikä tarkoittaa, että osapuolet eivät tallenna mitään tietoa yhteyden tilasta. Esimerkiksi identtisen pyynnön tekemisen pitäisi tuottaa aina sama tulos. (Fielding;ym., 1999)

WWW-palvelut tarvitsevat toimiakseen HTTP-protokollan. HTTP-protokolla vastaa viestinnästä selaimen ja palvelimen välillä. Kaikki käyttäjän selaimessa tekemät pyynnöt välitetään palvelimelle HTTP:n avulla ja se vastaa HTTP-viestein.

### 3.3.1 HTTP-metodit

HTTP käyttää kommunikointiin metodeja. Metodeja voidaan ajatella välineinä, joita HTTP-protokolla käyttää datan välitykseen. IETF (engl. Internet Engineering Task Force) määrittelee HTTP-protokollalle kahdeksan metodia:

- GET – Käytetään resurssien hakuun. Resurssi haetaan merkkijonosta (osoitteesta) ja näytetään asiakkaalle. Sivujen selaamiseen käytetään yleensä GET-metodia.
- HEAD – Metodilla voidaan pyytää vain verkkosivun otsaketiedot. Tämän avulla voidaan seurata onko sivustoa muutettu vai ei. Hakukoneiden robotit käyttävät tätä metodia tarkistaakseen onko sivustoon tullut muutos ja tarvitseeko sivua indeksoida uudestaan.

- POST – Käytetään tietojen lähettämiseen asiakkaalta palvelimelle, esimerkiksi verkkosivujen palautelomakkeiden pitäisi käyttää tätä metodia.
- OPTIONS – Kysely resurssien tai palvelimen ominaisuuksista.
- PUT – Tallennetaan merkkijonossa (osoitteessa) oleva resurssi.
- DELETE – Poistetaan merkkijonossa (osoitteessa) oleva resurssi.
- TRACE – Käytetään vikojen paikannukseen. Palvelin palauttaa asiakkaan lähettämän resurssin takaisin asiakkaalle, jolloin voidaan nähdä onko matkan varrella tapahtunut joku tahaton muutos.
- CONNECT – Yhteyden pyyntö, jota käytetään välityspalvelimien (engl. Proxy) kanssa.

Metodit voivat olla joko turvallisia (engl. safe), idempotentteja (engl. idempotent) tai ei-idempotentteja (engl. non-idempotent). Turvallisella metodilla tässä tapauksessa ei ole mitään viittauksia tietoturvaan. Sillä tarkoitetaan, että metodi ei vaikuta tai muuta palvelimen resursseja mitenkään. Sitä voidaan kutsua monta kertaa aiheuttamatta mitään sivuvaikutuksia. GET- ja HEAD-metodit ovat turvallisia, sillä ne ainoastaan lukevat resursseja, eivätkä muuta niitä mitenkään. Idempotentin metodin vaikutukset ovat samat riippumatta siitä, suoritetaanko se yhden tai useamman kerran. Idempotentti metodi voi olla esimerkiksi osoitetiedon haku tietokannasta, sillä se ei muuta tietokannan tilaa. Toisaalta osoitetiedon tallennus tietokantaan voi olla myös idempotentti, sillä metodin kutsuminen useita kertoja ei lisää identtisiä osoitetietoja tietokantaan. Idempotentteja metodeja ovat GET, HEAD, DELETE, OPTIONS ja TRACE. PUT-metodi puolestaan ei ole idempotentti, sillä se voi jokaisella kutsullaan muuttaa jollakin tapaa resursseja tai palvelimen tilaa. (Fielding;ym., 1999)

### 3.3.2 Metodien väärinkäyttö

Yksi pahimmista virheistä on käyttää HTTP-metodeja määrityksien vastaisesti. Esimerkiksi internetistä löytyy sivustoja, jotka käyttävät GET-metodeja tallentaakseen tietokantaan uutta tietoa tai päivittääkseen olemassa olevaa tietoa. Hakukoneet käyttävät nettisivujen läpikäyntiin GET-metodia, koska metodi on määritelty turvalliseksi (Fielding;ym., 1999 s. 51). Jos sivusto on toteutettu väärin, niin hakukoneet voivat tahattomasti muuttaa sivuston toimintaa tai resursseja, mikä voi aiheuttaa huomattavia ongelmia.

# 4 Salattu yhteys

Kuten TCP/IP-protokollapino, ei HTTP:kään ole luotettava protokolla tietoturvan kannalta. Tiedon suojaamiseen tarvitaan lisäksi erillinen protokolla. Tunnetuin ja käytetyin salausprotokolla on TLS (engl. Transport Layer Security), jonka edeltäjänä oli pitkään Netscapen vuonna 1994 kehittämä SSL (engl. Secure Socket Layer) (IBM).

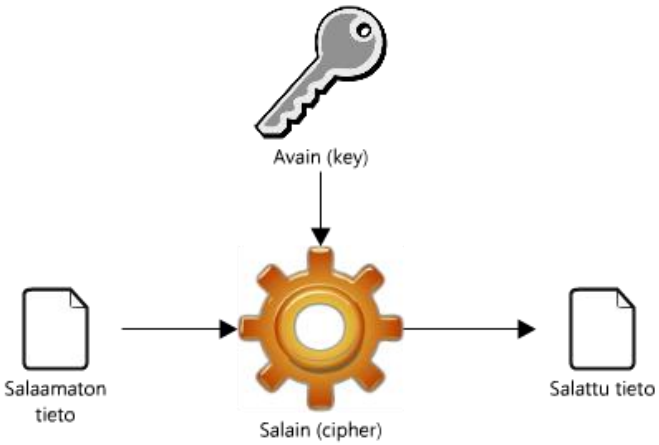
WWW-linkeissä on tavanomaisesti lisätty protokollan nimi, jonka näkee linkin alkuosasta. Linkki voi olla siis joko http://- tai https://-alkuinen. Ensimmäisessä linkissä tiedon siirtoon käytetään pelkkää HTTP:tä, kun taas jälkimmäisessä on kyse TLS-protokollasta, joka toimii HTTP:n alla. Yleisemmin HTTP ja TLS protokollapinoa kutsutaan HTTPS-protokollaksi (engl. Hypertext Transfer Protocol Secure). Jos linkissä ei ole lisätty erikseen http- tai https-alkuista osaa, tavanomaisesti selaimet ja ohjelmat käsittelevät pyyntöä suojaamattomana.

HTTPS-protokollaa ei saa sekoittaa S–HTTP-protokollaan, joka on hyvin vähän käytetty vaihtoehto HTTPS:lle. S-HTTP hävisi 90-luvulla käyneen kilpailun, mikä johtui pääosin Netscapesta ja Microsoftista, jotka tukivat HTTPS:ää. Tämän vuoksi HTTPS saavutti de facto -standardin aseman. (Rescorla;ym., 1999)

## 4.1 Salausmenetelmät

Oli kyse sitten sähköpostin, luottokorttitietojen tai reaaliaikaisen keskustelun salaamisesta, salauksen periaate on aina sama. Ehkä tunnetuimmaksi esimerkiksi on noussut Caesarin salaus, jota Rooman keisari Julius Caesar käytti 100 – 44 ennen ajanlaskun alkua. Jo silloin salauksen käytössä olivat samat pääelementit, mitä nykysalauksessa käytetään. (Järvinen, 2003)

Salaus muodostuu neljästä elementistä: alkuperäinen data, salain (engl. cipher), avain (engl. key) ja salattu data. Alkuperäinen data voi olla ihan mitä tahansa tavallisesta sähköpostiviestistä perhekuvaan. Salain on menetelmä ja salauksen koko ydin, jolla alkuperäinen data salataan. Avain puolestaan mahdollistaa salaimen käytön. Salain voi olla jopa julkisesti tiedossa, mutta avaimen pitää säilyä salassa kahden osapuolen välillä. Jos avain vuotaa jostain syystä julkisuuteen, se on kuitenkin helppo vaihtaa uuteen, eikä näin ollen tarvitse muuttaa koko salausmenetelmää. Hyvä salain on julkinen, sillä silloin salaimen luotettavuutta voi kokeilla myös muut kuin itse salaimen kehittäjät. (Järvinen, 2003 ss. 45-51)




**Kuva 2: Tiedon salaukseen tarvitaan salaimen lisäksi myös avain**

On virhe ajatella, että salattu tieto on täysin turvassa, jos sitä ei ole murrettu. Todellisuudessa myös salattua tietoa voi muuttaa tietämättä salauksessa käytettyä avainta tai ymmärtämättä salauksen menetelmää. Jos joku saa kaapattua salatun tiedon, hän voi arpapelillä muuttaa sitä ymmärtämättä siitä mitään. Pahimmassa tapauksessa kaappaajaa voi onnistaa muuttamaan tietoa siten, että vastaanottajan purettua salatun tiedon, hän ei huomaa tai tajua, että siihen on tullut muutos.

Tiedon muuttamattomuuden voi tarkistaa hajautusarvoilla, eli tiivisteillä. Tiivisteet ovat pieniä tietopaloja, jotka on muodostettu alkuperäisestä tiedosta. Niiden avulla voidaan vertailla onko esimerkiksi vastaanotettu tiedosto identtinen lähettäjän tiedoston kanssa. Tavanomaisesti tiivisteet allekirjoitetaan, jolloin tiivistettäkään ei voi muuttaa huomaamattomasti.

### 4.1.1 Caesarin salaus

Caesarin salausmenetelmä perustuu yksinkertaiseen tekniikkaan, jossa jokaiselle kirjaimelle on aakkoston asettamassa järjestyksessä vastaava numeerinen arvo. Salatessa tietoa, jokaisen kirjaimen arvoa siirretään eteenpäin tietyn määrän, jonka tietää ainoastaan viestin vastaanottaja ja lähettäjä. Caesar käytti menetelmää määräystensä salaamiseen.

Jos haluaisimme salata viestin ”Palvelu on murrettu”, Caesarin menetelmällä salattu viesti olisi muotoa ”Sdoyhox rq pxuuhwwx”. Jälkimmäinen viesti saadaan siirtämällä ensimmäisen viestin jokaista kirjainta kolme kertaa oikealle. Avain on tässä kolme ja salain, eli salausmenetelmä on siirtosalaus (engl. shift cipher).

Caesarin salausmenetelmä voidaan myös ilmaista muodossa:

$$C = (P + K) mod\ 29$$

jossa C on salattu kirjain, P on alkuperäinen kirjain ja K on avain (tässä esimerkissä 3). Modulo, eli mod varmistaa sen, että emme mene suomen kielen aakkoston ulkopuolelle, vaan pysytään 29 merkin sisällä. Esimerkiksi jos salataan kirjainta ä, sen salattu muoto Caesarin menetelmässä on b.

### 4.1.2 Modernit salausmenetelmät

Nykyään on käytössä useita eri salausmenetelmiä, joista tunnetuimpia ovat muun muussa 3-DES (engl. Triple-Data Encryption Stantard), AES (engl. Advanced Encryption Standard), IDEA (engl. International Data Encryption) ja RSA, jonka lyhenne muodostuu salausmenetelmän kehittäjien sukunimistä: Rivest, Shamir ja Adleman (Menezes;ym., 2001). Ensimmäinen menetelmä on paranneltu versio alkuperäisestä DES-salauksesta, joka otettiin Yhdysvalloissa vuonna 1976 käyttöön standardisalausmenetelmänä (Järvinen, 2003 s. 87). DES:n avaimen pituus on 56 bittiä. Toisen menetelmän, eli AES:n avainpituus voi olla joko 128, 192 tai 256 bittiä. Jos tarkastellaan numeroita pelkistetysti, niin luulisi AES:n olevan ainoastaan noin kaksi – neljä kertaa tehokkaampi kuin DES-menetelmä. Todellisuudessa AES 128 bit on noin 4,7 kvintilijoonaa ($10^{30}$) kertaa tehokkaampi salausmenetelmä kuin DES.

### 4.1.3 Salaimen vahvuus

Salaimen vahvuutta tavanomaisesti tarkastellaan avaimen pituudella. Tietotekniikassa avain koostuu biteistä. Yksi bitti voi olla joko 0 tai 1 ja kahdesta bitistä voi muodostaa neljä eri numeroa: 00, 01, 10 ja 11. Jos luvut muutetaan kymmenjärjestelmään, ne ovat: 0, 1, 2 ja 3. Esimerkiksi 10-bittisessä avaimessa on $2^{10} = 1024$ avainvaihtoehtoja. DES-algoritmissa avaimen pituus on 56 bittiä ja siinä eri vaihtoehtojen määrä $2^{56} = 72\ 057\ 594\ 037\ 927\ 936$. Vaikka luku voi näyttää suurelta, todellisuudessa tietokoneelle luku on hyvin pieni. DES-suojaus voidaankin murtaa nykykoneilla alle 24 tunnissa (SciEngines).

## 4.2 Julkinen ja yksityinen avain

Julkisen avaimen salaus (engl. Public-key encryption) on yksi salauksen menetelmistä, joka tunnetaan myös nimellä asymmetrinen salaus. Asymmetrisessä salauksessa käytetään kahta avainta, joista julkinen avain jaetaan kaikille halukkaille ja yksityinen avain pidetään vain omana tietona. Tieto, joka on salattu julkisella avaimella, voidaan purkaa vain ja ainoastaan yksityisellä avaimella. Yksityisellä avaimella tietoa puolestaan voidaan digitaalisesti allekirjoittaa ja allekirjoitus varmistaa julkisella avaimella.

Kahden tahon väliseen suojattuun keskusteluun ei yleensä käytetä pelkästään asymmetrisen salauksen perusperiaatetta, koska se vaatii paljon laskentatehoa. Asymmetristä salausta

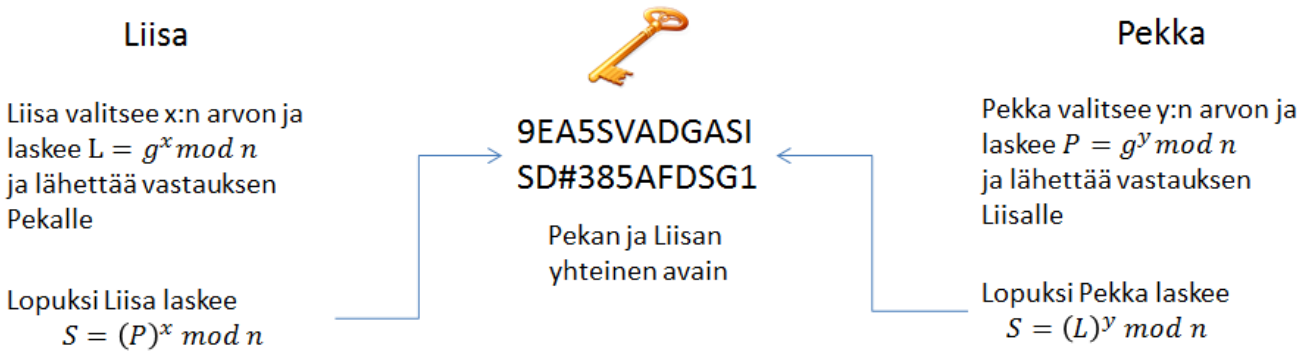
käytetäänkin pääosin digitaalisena allekirjoituksena. Koska yksityinen avain on tiedossa ainoastaan avaimen luojalla, voidaan esimerkiksi hänen lähettämän suostumuksen aitous tarkistaa julkisella avaimella.

## 4.3 Symmetrinen salaus ja avainvaihto

Symmetrisen salauksen periaate on se, että tiedon salaukseen ja purkuun käytetään samaa avainta. Symmetrinen salaus on hyvin yleinen salausmenetelmä monissa internet-palveluissa, koska se on tehokkaampi menetelmä kuin asymmetrinen salaus (Oh;ym., 2003).

Asymmetrisen salauksen avulla voidaan luoda myös symmetrinen avain. Menetelmä perustuu diskreettiin logaritmiin. Koska diskreetin logaritmin ratkaisuun ei ole tehokasta tapaa, voidaan matemaattista menetelmää käyttäen sopia kahden osapuolen välille symmetrinen avain julkista verkkoa käyttäen. Tätä metodia kutsutaan Diffie–Hellman avainvaihdoksi. (Menezes;ym., 2001)

Kun julkisessa tiedossa on generaattori *g* ja kunta *n*, Pekka ja Liisa voivat laskea yhteisen avaimen. Vain Liisa tietää *x* arvon ja vain Pekka tietää *y* arvon. Kuten kuvassa 3 esitetään, Liisa ja Pekka voivat luoda yhteisen salaisuuden julkisen verkon yli.




**Kuva 3: Diffie-Hellman avainvaihto**

Toinen tapa symmetriseen salaukseen Diffie-Hellman avainvaihdon metodin lisäksi on hybridisalaus. Hybridisalauksessa nimensä mukaan käytetään myös asymmetrista salausta apunaan, mikä mahdollistaa symmetrisen avaimen vaihdon turvallisesti. Tämä tapa on lähellä Diffie-Hellmanin menetelmää, mutta kuitenkin eroaa siinä, että symmetrisen avaimen muodostamiseen ja vaihtoon tarvitaan ainoastaan yhden osapuolen asymmetrista avainparia. Hybridisalauksen toimintaperiaate on kuvattu alla (käytetään esimerkkinä Liisan ja Pekan välistä keskustelua):

Jotta Pekka voisi lähettää Liisalle salatun viestin, hän ensiksi hankkii Liisan julkisen avaimen. Seuraavaksi Pekka generoi uuden symmetrisen avaimen ja salaa äskettäin luodulla symmetrisellä avaimella viestin. Tämän jälkeen Pekka vielä salaa symmetrisen avaimen Liisan julkisella avaimella ja lähettää sekä salatun viestin, että salatun avaimen Liisalle.

Purkaakseen salauksen Liisan pitää tehdä seuraava toimenpide: Liisa käyttää omaa yksityistä avaintaan purkaakseen Pekan lähettämän asymmetrisen avaimen, jota hän käyttää salatun viestin avaamiseen.

Symmetrinen salaus on laajalti käytössä www-palveluiden tietoturvaratkaisuina, koska se on tehokkaampaa, helpompaa ja halvempaa kuin asymmetrisen salauksen käyttö. Kun käytetään symmetristä avainta, tiedon salaus ja purkaminen on nopeampaa. Lisäksi symmetrisessä salauksessa molemmilla osapuolilla ei tarvitse olla sekä julkista, että yksityistä avainta. Kun symmetrinen avain on molemmilla osapuolilla, niin asymmetristä avainta ei tarvitse enää käyttää.

## 4.4 Salaus www-liikenteessä

Suurin osa salatusta www-liikenteessä on toteutettu TLS-protokollalla. TLS huolehtii yksityisyydestä ja tiedon eheydestä kahden keskenään keskustelevan sovelluksen välillä (Dierks;ym., 2008). TLS:stä on kehitetty kolme versiota (1.0 1.1 ja 1.2), jotka ovat tuettuna useimmissa selaimissa. TLS:n uusin versio sisältää tuen muun muussa DES, 3-DES, RSA ja AES salausalgoritmeihin (Dierks;ym., 2008). TLS:n ongelma on kuitenkin siinä, että esimerkiksi vanhentuneet selaimet eivät tue uusimpaa TLS:n versiota, mikä tekee TLS turvallisuudesta yhtä tehokkaan, kuin vahvin salausmenetelmä, jota sekä asiakkaan selain että palvelimen ohjelmisto tukee. Pahimmassa tapauksessa kriittisen palvelun välillä kommunikoidaan ainoastaan DES-algoritmin avulla. Vaikka DES onkin vanhentunut ja siten heikko salausmenetelmä, sitä käytetään edelleen salamaan tietoa internetissä. Esimerkiksi jos asiakkaan selain on vanhentunut versio, joka tukee ainoastaan DES-menetelmää, asiakkaan otettua yhteyden palvelimeen, palvelin muodostaa yhteyden käyttäen DES-algoritmia. Käyttäjä voi luulla, että sen lähettämä tieto on turvassa, mutta todellisuudessa sen voi kaapata ja murtaa jälkikäteen.

## 4.5 Asiakkaan ja palvelimen keskustelu TLS:n avulla

Asiakkaan ja palvelimen välistä keskustelun muodostamista kutsutaan usein kättelyksi. Asiakas lähettää palvelimelle ”Hello”-viestin, johon asiakkaan kone lisää listan selaimen tuetuista salausmenetelmistä. Palvelin vertaa asiakkaan ja omaa tuettujen salaimien listaa, josta se valitsee tehokkaimman salausmenetelmän. Lisäksi palvelin lähettää oman palvelinsertifikaatin, johon

sisältyy palvelimen julkinen avain. Lopuksi palvelin vastaa asiakkaalle ja kertoo mitä salainta sen tulee käyttää. (Microsoft, 2003)

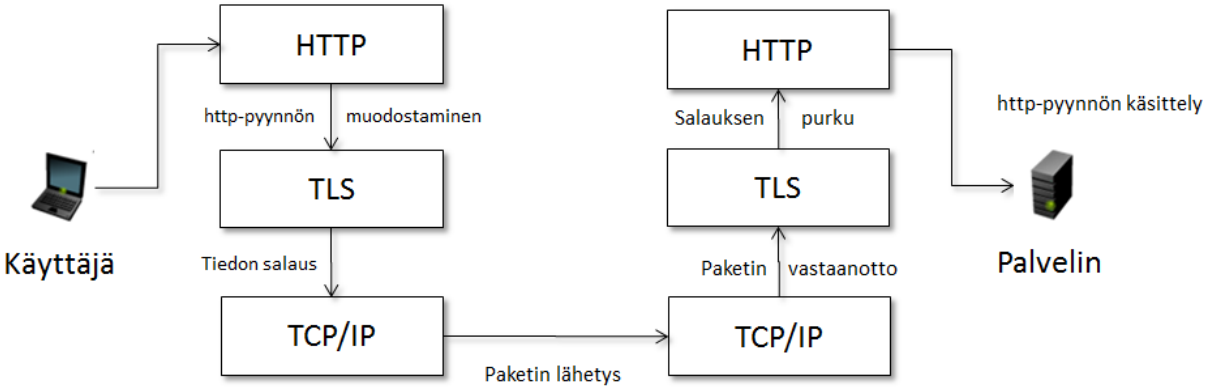
## 4.6 Julkisen avaimen infrastruktuuri

Palvelimen pitää jotenkin tunnistautua, jotta voidaan varmistua siitä, että palvelimen lähettämä julkinen avain on oikeasti palvelimen avain, eikä minkään muun tuntemattoman osapuolen avain. Tähän tarvitaan varmentaja. Varmentaja on toinen yritys tai taho, joka on varmistanut, että esimerkiksi www.nordea.fi osoitteesta lähetetty julkinen avain on oikeasti Nordea-pankin avain. Varmentaja allekirjoittaa sertifikaatin, eli varmenteen, joka sisältää palvelun nimen ja julkisen avaimen.

Selaimissa on määritelty valmiiksi tunnetuimmat ja luotetut varmentajat, sekä niiden julkiset avaimet. Jos joku taho, joka ei ole selaimessa määritelty luotettavaksi varmentajaksi, on varmentanut tietyn palvelun, käyttäjä saa siitä sertifikaattivaroituksen. Selain varoittaa käyttäjää myös, jos sertifikaatti on vanhentunut tai sertifikaatin sisältämä nimi ei vastaa palvelimen nimeä. Julkisen avaimen menetelmän käytön suurin hankaluus on tietää, että avain oikeasti kuuluu juuri sille vastaanottajalle, jolle sen uskotaan kuuluvan.

## 4.7 Salatun www-liikenteen kulku

Ensiksi selain muodostaa käyttäjän tarpeen mukaan HTTP-pyynnön. Toiseksi TLS salaa tiedon, ennen kuin se välittää sen eteenpäin TCP:lle. Kolmanneksi TCP/IP protokollapino välittää paketin salatun sisällön palvelimelle. Palvelimen päässä TCP/IP kerros vastaanottaa salatun paketin ja siirtää sen eteenpäin TLS-protokolalle. TLS purkaa salauksen, jonka jälkeen HTTP-pyyntö välitetään HTTP-palvelinohjelmalle. Palvelin vastaa asiakkaan pyyntöön samalla periaatteella.




**Kuva 4: Salatun tiedon kulku internetin protokollapinossa**

## 4.8 Kriittisten palvelujen tietoturvaratkaisut

Monet tärkeät ja suositut palvelut käyttävät aina salattuja yhteyksiä, sillä julkisuudella on myös haittapuolensa. Mitä enemmän käyttäjiä palvelu kerää, sitä suositummaksi hyökkäyskohteeksi se tulee. Myös palvelun kriittisyys on suuri yllyke ammattilaisille huijareille ja verkkohyökkääjille.

Istuntojen automaattinen vanhentuminen on hyvä tapa estää käyttäjätunnusten väärinkäyttöä. Ominaisuus korostuu etenkin, kun palveluja käytetään julkisilla koneilla ja paikoilla. Esimerkiksi pankeilla on käytössä vaihtuvat salasanat, mikä pienentää asiakkaan ja pankin riskiä. Monet verkkopakkien tietoturvaratkaisut ovat tehostettu istuntojen aikakatkaisulla. Myös Google käyttää palveluissaan istunnon vanhentumista, vaikka Googlen palvelut mahdollistavatkin käyttäjän muistamisen. Vaikka käyttäjä onkin valinnut vaihtoehdon ”muista minut”, Google silti kysyy tietyin väliajoin käyttäjän salasanaa. Tapaus voi ilmetä, kun käyttäjä yrittää seuraavana päivänä päästä esimerkiksi Googlen sähköpostipalveluun Gmailiin.

Jotta kriittisten palveluiden ratkaisut olisivat tehokkaita, myös niiden yhteistyökumppaneiden ja muiden palveluketjussa mukana olevien osapuolten ratkaisujen pitää olla kunnossa (Ylitalo, 2004). Esimerkiksi Facebook tarjoaa connect-ominaisuutta muille verkkosivuille, mikä mahdollistaa kommenttien lisäämisen kolmannen osapuolen verkkosivulle pelkällä Facebook-tunnuksen sisäänkirjautumisella. Jos kolmannen osapuolen verkkosivustossa on puutteita tietoturvaratkaisuissa, käyttäjien tiedot voivat olla vaarassa. Sama periaate pätee myös pankkiratkaisuissa ja niiden yhteistyökumppaneiden sekä asiakkaiden tapauksissa. Verkkokauppojen tietoturvan pitää olla kunnossa, jotta asiakkaan tiedot pysyvät turvassa hyökkääjiltä ja huijareilta. Lisäksi pankkien välisen toiminnan pitää olla moitteetonta. Näin ei valitettavasti aina ole.

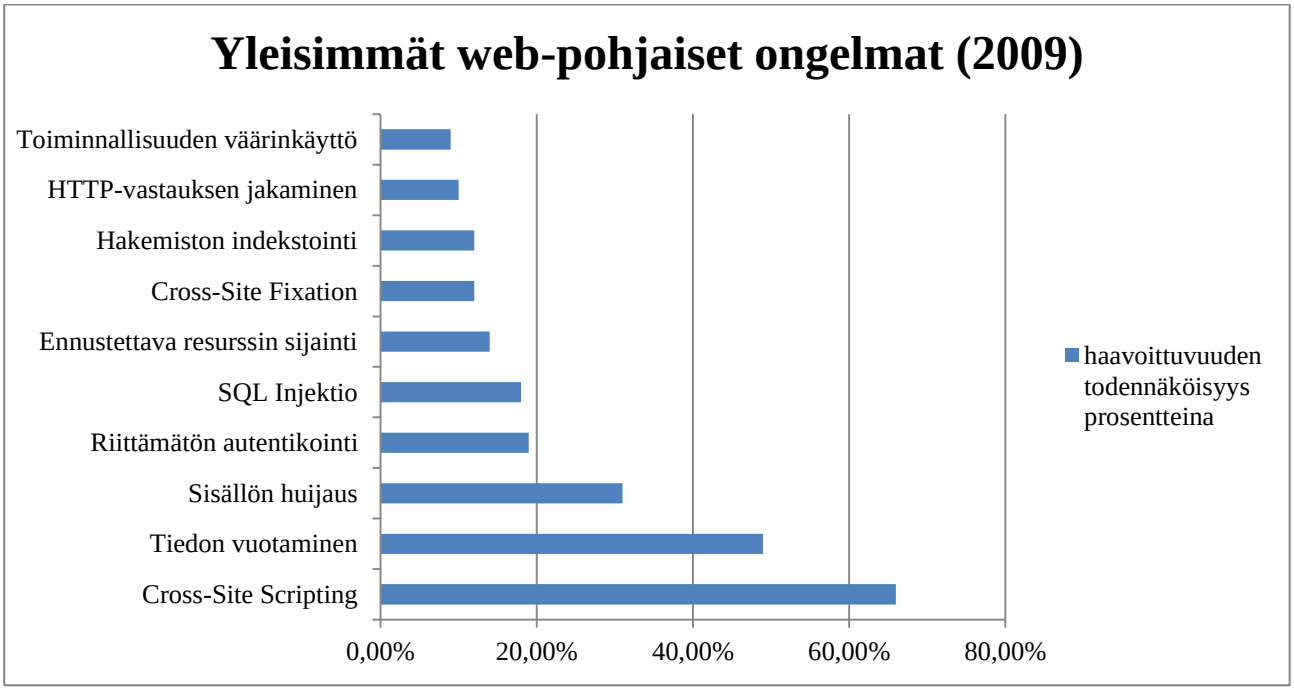
Anna Mård (2007) kertoo artikkelissaan, että tutkimusyhtiö Datamonitorin selvityksen mukaan pankkien tietoturva on Euroopassa edelleen erittäin hajanaista. Tutkimuksen mukaan ainoastaan alle puolet pankeista vaatii alihankkijoiltaan ja yhteistyökumppaneiltaan yhtä hyvää tietoturvaa kuin itseltään. Vaikka tutkimuksessa ei ollut pohjoismaiden pankkeja, tutkimuksen tekijä uskoo, että tilanne on täysin sama muuallakin. Datamonitorin selvityksen mukaan pankit tuntuvat ajattelevaan tietoturvaa pääosin it-näkökulmasta, vaikka tietoturvan suurin ongelma on ihminen. Suomessa on myös paljastunut tapaus, jossa pankin asiakastietoja imuroi yrityksen tietoturvapäällikkö itse (Mård, 2007).

Finanssialan Keskusliiton maksujärjestelmätoiminnan johtaja Timo Ylitalo on puolustellut suomalaisia pankkeja mainitsemalla, että turvallisuus ei ole absoluuttinen tekijä, vaan

liiketoiminnallinen asia. Ylitalon mukaan järjestelmät pitää suojata siten, että asiakkaat pystyvät vielä käyttämään palvelua. Lisäksi resurssien kulutus suhteessa uhkiin pitäisi pysyä järkevissä lukemissa. Ylitalo myös kiistää, että Suomessa pankit keskittyvät liikaa tietoturvan it-kysymyksiin. Hänen mukaan pankit eivät ole pitkään aikaan luottaneet pelkästään tietotekniikkaan, vaan asiakkaat on tunnistettu haavoittuvampana palvelun osana. (Taloussanomat, 2007)

# 5 Yleisimmät verkkohyökkäykset

Verkkohyökkäykset kehittyvät jatkuvasti ja ovat usein nopea ja tehokas tapa arkaluonteisen tiedon varastamiseen tai tuhon aikaansaamiseen. Verkkohyökkäyksiä voi verrata huligaaneihin ja ammattivarkaisiin. Osa hyökkäyksistä on amatöörien tekemiä ja niiden tarkoitus onkin ainoastaan tuho ja kerskailu omilla hyökkääjän taidoilla. Verkkohyökkäyksiä tekevät myös ammattilaiset. Ammattilaisten tarkoituksena on luottokorttitietojen varastaminen tai arkaluonteisen materiaalin hankinta ja myynti. Myös kiristys ja uhkailu, esimerkiksi palvelunestohyökkäyksillä, ovat menetelmiä, joita ammattilaiset harrastavat. Uutisissa kuulee välillä yrityksiin kohdistuvista tietomurroista, joiden takana arvellaan olevan kilpailevia yrityksiä. Vaikka verkkohyökkäykset ovat useimmissa maissa laittomia, verkkohyökkäyksiä käytetään saavuttaakseen kilpailuetua häiritsemällä tai lamauttamalla kilpailijoiden toimintaa tai hankkimalla liiketoiminnallisia salaisuuksia.




**Kuva 5: Yleisimmät verkkosivujen ongelmat ja haavoittuvuudet (Grossman, 2009)**

Kuvassa 5 esitellään verkkosivujen yleisimmät ongelmat ja haavoittovuudet. Yleisin ongelma web-pohjaisissa ratkaisuissa on Cross-Site scripting, eli XSS. Mitä laajempi ja sisällöltään käyttäjäpainotteisempi sivusto, sitä suurempi riski sivustolla on XSS-hyökkäykseen. XSS:ää ja SQL-injektiota käydään läpi tarkemmin luvussa 5.3 .

Tiedon vuotaminen voi tapahtua vahingossa, kun esimerkiksi sivustolla näytetään kaikille kävijöille sivustokäyttäjien henkilökohtaisia tietoja, ei-julkisen ohjelmiston lähdekoodia tai muuta

arkaluonteista tietoa inhimillisen erehdyksen takia tai sivuston väärän toiminnan seurauksena. Sisällön huijaus liittyy usein sosiaaliseen manipulointiin, jossa käyttäjää ohjataan alkuperäisen sivuston sijaan huijaussivulle. Sisällön huijausta käsitellään luvussa 5.5 .

Riittämätön autentikointi on käyttötilanne, jossa käyttäjä saa enemmän oikeuksia kuin hänelle on alun perin annettu tai määritelty. Tapaukset voivat liittyä myös osittain XSS:ään. Myös ennustettava resurssien sijainti voidaan yhdistää riittämättömään autentikointiin. Resurssien uskotaan olevan turvassa, jos niiden sijaintia, eli osoitetta ei paljasteta. Esimerkiksi jonkun yksityisen ohjelman lähdekoodi voi sijaita osoitteessa http://example.com/development/source. Automaattiset skannerit voivat kuitenkin löytää myös sellaiset osoitteet, joita ei ole missään vaiheessa annettu julkiseen tietoon, sillä ne generoivat ja käyvät tuhansia osoitearvauksia läpi (Grossman, 2009).

Cross-Site Fixation tai Cross-Site Request Forgery on menetelmä, jossa hyökkääjä voi käyttää toisen käyttäjän selainta hyökkäyksen työkaluna (Grossman, 2009). Hyökkääjä voi suorittaa murretun käyttäjän selaimella muita selainpyyntöjä, mitä tavallinen käyttäjä ei edes huomaa. Sivusto käsittelee pyynnöt, koska ne tulevat lailliselta käyttäjältä. Aihetta käsitellään lisää luvussa 5.2 .

Hakemiston indeksointi, http-vastauksen jakaminen ja toiminnallisuuden väärinkäyttö liittyvät kaikki sivuston ominaisuuksiin, toimintaan ja asetuksiin, jotka ovat hyvin yksityiskohtaisia ja niistä on hankala tehdä mitään yleistyksiä. Useimmat hyökkäykset vaativat tarkempaa tietoa sivuston rakenteesta ja lähdekoodista.

## 5.1 Palvelunestohyökkäys

Palvelunestohyökkäys (engl. Denial of Service) eli DOS on hyökkäys tiettyä palvelua tai resurssia vastaan. Hyökkäyksen idea on lamauttaa hyökkäyksen kohde, jolloin sen käyttö saadaan keskeytettyä lyhyeksi ajaksi tai kokonaan. Pääajatuksena on estää tavallisten oikeutettujen käyttäjien pääsy palveluun tai resurssiin, jolloin siitä aiheutuu suuria kustannuksia palvelun ylläpitäjälle ja haittaa käyttäjille. Palvelunestohyökkäykset voidaan jakaa kolmeen päätyyppiin: rajallisten resurssien kulutus, asetuksien muuttaminen ja fyysisten komponenttien tuho (Carnegie Mellon University, 1999).

Rajallisten resurssien kulutuksessa hyökkääjällä on useita vaihtoehtoja, joita se voi käyttää. Otetaan esimerkiksi www-palvelin, jonka hyökkääjä on valinnut sen kohteeksi. Palvelunestohyökkäyksen tekijä voi ylikuormittaa palvelimen TCP-paketeilla. Tämä onnistuu siten, että hyökkääjä lähettää yhteyspyynnön palvelimelle, mutta ei saata yhteydenmuodostusta koskaan loppuun ja lähettää

keskeneräisen yhteyspyynnön perään uuden yhteyspyynnön (Carnegie Mellon University, 2000). Palvelin saa valtavasti valheellisia yhteyspyyntöjä, joita se joutuu käsittelemään. Käsittelyssä kuluu enemmän aikaa ja resursseja kuin hyökkääjältä pyynnön lähettämisessä. Lopulta palvelin voi ylikuormittua niin paljon, ettei se pysty käsittelemään kenenkään muun asiakkaan yhteyspyyntöjä. On hyvä huomioida, että tässä tapauksessa hyökkääjän tarkoituksena ei ole verkon ylikuormitus, vaan mahdollisesti palvelimen kaataminen kokonaan. TCP-tulvaa vastaan voidaan suojautua asentamalla järjestelmään TCP-tulvanestopäivityksiä ja konfiguroimalla reitittimet oikein.

Toinen resurssien kulutuksessa käytetty menetelmä on käyttää yrityksen omia resursseja sitä vastaan. Esimerkiksi hyökkääjä voi saada kaksi yrityksen palvelinta lähettämään UDP-protokollan avulla (engl. User Datagram Protocol) kaiutettuja viestejä toisilleen (Carnegie Mellon University, 1997). Kaiutetulla viestillä tarkoitetaan viestiä, jonka palvelin palauttaa (kaiuttaa) viestin lähettäjälle. Virallisesti tätä menetelmää käytetään pääosin verkon toiminnan kokeilussa ja varmistamisessa, mutta tässä tapauksessa sitä voidaan käyttää myös palvelunestohyökkäyksen suorittamiseen. Tältä hyökkäykseltä voidaan suojautua, kun palvelimilta poistetaan käytöstä Echo-palvelut ja kaikki käyttämättömät UDP-palvelut.

Kolmas tapa on hyvin perinteinen: hyökkääjä luo turhia paketteja ja lähettää ne yrityksen verkon infrastruktuuriin. Paketit ylikuormittavat reitittimiä ja näin hidastavat verkkoa ja voivat mahdollisesti lamauttaa koko verkon. Usein DOS-hyökkäyksissä hyökkääjä käyttää apunaan useaa eri konetta, eli hajauttaa hyökkäyksen. Hajautettu palvelunestohyökkäys eli DDOS-hyökkäys (engl. Distributed Denial of Service) tehostaa palvelunestohyökkäysmenetelmää valtavasti. Hajautettua palvelunestohyökkäystä vastaan on hyvin hankalaa suojautua, mutta siihenkin on joitakin työkaluja. Esimerkiksi jatkuva pakettitulva tietystä IP-alueesta voidaan hetkellisesti pysäyttää. Reitittimiin voidaan määritellä esimerkiksi tilapäisen ajan, jolloin mitään paketteja ei vastaanoteta, jos ne tulevat samasta IP-alueesta.

Hyökkääjän ei tarvitse tyytyä pelkästään rajallisten resurssien kulutukseen. Jos palvelin tai reititin on huonosti suojattu, hyökkääjä voi saada pääsyn laitteisiin ja muokata tai poistaa niiden asetuksia. Tämä voi puolestaan estää palvelun tai resurssien käytön kokonaan.

Hyökkäysten alkuperä ei ole aina verkko. Joskus yritykset käyttävät tosi paljon resursseja suojautuakseen verkosta tulevia hyökkäyksiä vastaan, mutta unohtavat kokonaan laitteiden fyysisen turvallisuuden. Jos verkosta on miltei mahdotonta tehdä hyökkäystä, niin hyökkääjä etsii helpomman vaihtoehdon, eli konehuoneen. Jos koneet eivät ole lukitussa tilassa tai niitä ei vartioi

kukaan, niin hyökkääjä voi helposti käydä sotkemassa koko järjestelmän ja aiheuttaa DOS-tilanteen käyttämättä verkkoa.

## 5.2 Hyökkäykset selainta vastaan

XSS:n lisäksi pahimpia www-palveluiden tai web-ohjelmistojen heikkouksia ovat selaimet (Grossman, 2009). Monet selaimet tukevat kolmannen osapuolen tekemiä laajennuksia. Useimmiten käyttäjät eivät tarkista laajennuksien alkuperää tai turvallisuutta. Selainlaajennus voi pahimmassa tapauksessa olla troijalainen, eli ohjelma, joka naamioituu hyödylliseksi ohjelmaksi, vaikka todellisuudessa se on haittaohjelma. Haittaohjelma voi esimerkiksi lukea kaikki käyttäjän näppäimistöltä syöttämät tiedot, mukaan lukien käyttäjänimet ja salasanat.

Paljon kritiikkiä ja keskustelua on herättänyt Internet Explorerin ActiveX-laajennukset. ActiveX on komponentti, joka toimii normaalin ohjelman tavoin. Ongelma liittyy siihen, että ActiveX on tuonut myös haitallisia ohjelmia, joita käyttäjät ovat asentaneet tutustumatta ensin riskeihin.

Toiminnallisuutta on laajennettu myös muuten, esimerkiksi Javalla. Java on hyvin monipuolinen ohjelmointikieli, jolla voi tehdä myös web-pohjaisia ratkaisuja. Javan valtti ja heikkous on kuitenkin pääsy asiakkaan päätteeseen. Käyttäjä voi antaa Javalle oikeudet omaan koneeseen, jolloin ohjelma voi tehdä muutoksia käyttäjän koneella.

Hyökkäyksen selainta vastaan voi tehdä myös evästeiden eli keksien kautta (engl. Cookie). Keksit ovat tiedostoja, jotka selain tallentaa palvelimen pyynnöstä käyttäjän koneelle. Ne voivat sisältää tietoa istunnosta tai ihan mitä tahansa muuta dataa. Jos hyökkääjä pystyy muuttamaan uhrin keksin sisällön, verkkosivu tai palvelin voi luulla, että muutos oli laillinen, koska keksi sijaitsee käyttäjän koneella. Evästeen avulla hyökkääjä voi siis esiintyä toisena käyttäjänä.

## 5.3 Cross-site-scripting (XSS) ja koodin injektio

Cross-site-scripting, eli XSS on www-palveluiden tietoturva-aukko, joka mahdollistaa koodin injektion www-palvelun järjestelmään asiakkaan päätteestä (Bodmer, 2007 s. s. 3). XSS-hyökkäyksissä voi olla suuriakin eroja, sillä hyökkäyksessä käytetty kieli tai menetelmä voi olla hyvin poikkeava toisessa hyökkäyksessä käytettyyn menetelmään nähden. Onnistuneessa XSS-hyökkäyksessä tekijä voi saada korkeampia oikeuksia järjestelmään kuin tavalliselle käyttäjälle on tarkoitettu, saada käsiinsä arkaluonteista tietoa tai sotkea järjestelmän tai nettisivuston käyttökelvottomaksi (Bodmer, 2007 s. s. 4).

Kaikki mahdolliset sivuston lomakkeet, missä käyttäjä voi itse syöttää tietoja ovat XSS:n riskitekijöitä. Jos syötettä ei suodateta tai tarkisteta mitenkään, hyökkääjä voi syöttää tekstikenttiin omaa koodia, joka suoritetaan www-palvelun päässä. Jos hyökkääjä tietää palvelun rakenteesta ja käytetystä teknologiasta, hän voi helposti kohdistaa hyökkäykset suoraan teknologioiden heikkoihin kohtiin.

Oletetaan esimerkiksi sivusto, joka toimii PHP-ohjelmointikielen (engl. Hypertext Preprocessor) avulla ja käyttää tietokantana MySQL-tietokantaa. Tietokannan käskyt saadaan suoraan GET-parametrin tiedoista. Alla oleva esimerkki kuvaa linkkiä käyttäjälistaussivulle. Tämä on hyvin huonosti toteutettu ratkaisu ja sen tarkoitus on vain demonstroida koodin injektiota, joten näin sivustoa ei kannata missään tapauksessa toteuttaa:

```
http://example.com/index.php?mysql=”SELECT * FROM users;”
```

Nyt jos hyökkääjä tietää tietokannan nimen, hän voi helposti tuhota vaikka koko tietokannan suorittamalla seuraavan http-pyynnön:

```
http://example.com/index.php?mysql=”DROP DATABASE database_name;”
```

Tietokannan nimen voi saada selville kokeilemalla kaikkia yleisempiä tietokannan nimiä tai tässä tapauksessa suorittamalla seuraavan tietokantakyselyn: ”`SHOW TABLES;`”, joka listaa kaikki taulukot, sekä tietokannan nimen. Tämä on hyvin yksinkertainen esimerkki SQL-injektiosta (engl. Structured Query Language -injection).

Injektion ei aina tarvitse välttämättä olla tietokantaan kohdistuva hyökkäys, vaan esimerkiksi PHP:llä toteutettuun palautelomakkeeseen hyökkääjä voi lisätä omaa PHP-koodia, joka puolestaan voi sotkea sivuston ja rikkoa sen toiminnan.

```
<?php
  $string = "<script type=”text/javascript”>alert('Tämä on XSS');</script>";
  echo $string;
?>
```

Yllä oleva koodinpätkä kuvaa palautelomakkeeseen liitettyä viestiä, joka injektoi JavaScript-koodia sivustoon. Kun sivusto lataa palautetekstin, PHP prosessoi koodin, joka puolestaan lisää JavaScript

rivin ” `<script type=”text/javascript”>alert('Tämä on XSS');</script>` ” sivustoon. JavaScript koodi aiheuttaa sivustolla hälytyksen ”Tämä on XSS”.

## 5.4 Mies välissä-hyökkäys

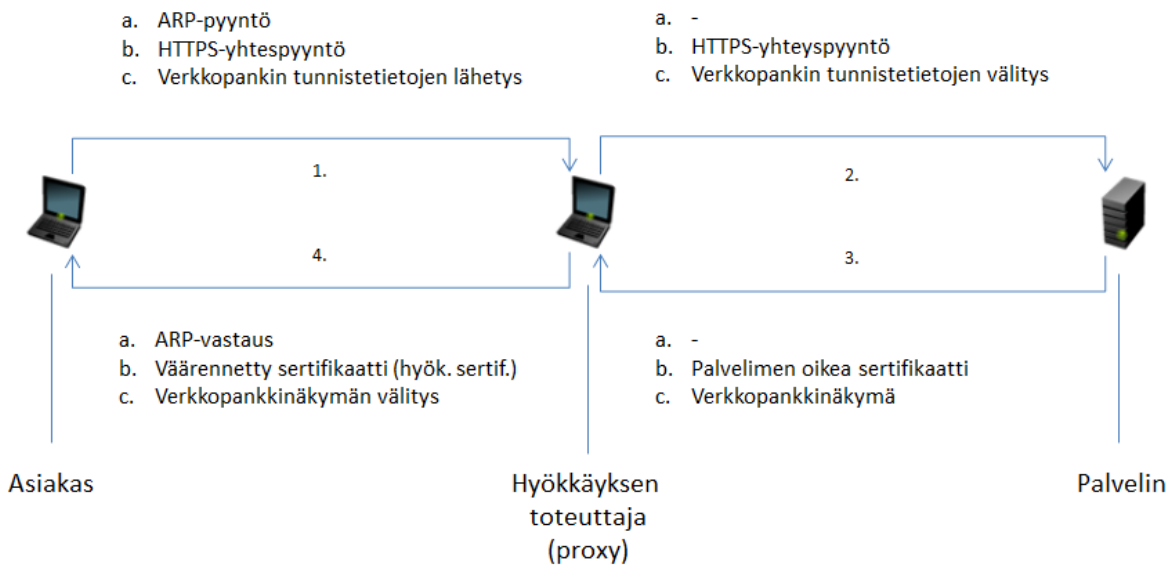
Mies välissä-hyökkäys (engl. Man in the Middle), eli MITM on tietoturvahyökkäys, jossa hyökkääjä asettuu kahden osapuolen väliin. Hyökkääjä siis toimii osapuolien tietoliikenteen välittäjänä ja halutessaan voi kaapata tai muuttaa tietoa (Burkholder, 2002). MITM-hyökkäyksiä vastaan on hyvin vaikea suojautua, koska hyökkäys tapahtuu suhteellisen matalalla tasolla, josta tavallisella käyttäjällä ei usein ole mitään tietoa.

Eräs tapa toteuttaa MITM-hyökkäys on lähettää uhrin sijaitsemaan verkkoon ARP-väärennös (engl. Address Resolution Protocol), kuten esitetään kuvassa 6. ARP on protokolla, joka mahdollistaa MAC-osoitteiden selvittämisen IP-osoitteen avulla. MAC-osoite on jokaisen fyysisen verkkokortin uniikki osoite, joka erottaa laitteet toisistaan. Kun uhri pyytää ARP:n avulla verkon internetiin yhdistävän reitittimen, eli yhdyskäytävän osoitteen, hyökkääjä voi lähettää oman vastauksen nopeammin kuin aito reititin, jolloin uhrin kone tallentaa sen yhdyskäytävän osoitteena. Tällöin uhrin kone lähettää kaiken tiedon hyökkääjän koneen kautta. (Burkholder, 2002)

Kun hyökkääjä on asettunut uhrin ja palvelimen väliin varsinainen tiedon salakuuntelu voi alkaa. MITM-hyökkäyksen voi toteuttaa myös salatun yhteyden kanssa. Silloin tarvitaan kaksi erillistä TLS-sessiota: yksi palvelimen ja hyökkääjän välille, ja toinen uhrin ja hyökkääjän välille. Kun uhri päättää mennä osoitteeseen https://www.nordea.fi, hyökkääjä välittää HTTP-pyynnön Nordean palvelimelle ja muodostaa samalla salatun yhteyden palvelimen kanssa esiintyen käyttäjänä. Tämän jälkeen hyökkääjä voi luoda oman sertifikaatin, jonka se lähettää palvelimen vastausviestin (verkkosivun) kanssa uhrille. Uhrin selain huomauttaa sertifikaattivaroituksesta, mutta käyttäjä voi sivuuttaa varoituksen, koska ei ymmärrä uhkaa. Joissakin tapauksissa hyökkääjä voi myös kiertää sertifikaattivaroituksen. Hyökkääjä voi käyttää Internet Explorerin 6.0 ja aikaisempien versioiden haavoittuvuutta tai käyttää virallista varmentajaa varmentaakseen oman sertifikaatin (Burkholder, 2002). Jälkimäiseen tapaukseen liittyy muitakin tekijöitä kuin pelkän varmennuksen hankinta, sillä pelkällä hyökkääjän virallisella varmennuksella sertifikaattihuijausta ei pysty tekemään.

MITM-hyökkäyksiä vastaan on olemassa ratkaisuja. Koska hyökkäys perustuu ARP-väärennöksiin, sivustojen pitäisi käyttää mahdollisuuksien mukaan staattisia ARP-taulukoita, sekä DNSSEC-laajennuksia (engl. Domain Name System Security Extensions), eli nimipalvelujärjestelmän suojauslaajennuksia (Burkholder, 2002). Verkkoon voi myös asettaa seurantaohjelmia, jotka voivat

huomata ja estää ARP-väärennöksiä ja MITM-hyökkäysyrityksiä. Käyttäjät voivat suojautua hyökkäyksiltä hyvien palomuurien avulla. Palomuurit valvovat yhteyksiä käyttäjän koneilla, ja jos koneesta ei ole lähetetty ARP-pyyntöä, niin ei myöskään mitään ARP-vastausta oteta vastaan.




**Kuva 6: Mies välissä -hyökkäyksen periaate.**
**Kohdat a-c kuvaavat toimintaa ja numerot 1-4 ilmaisevat toimintajärjestystä.**

## 5.5 Sosiaalinen manipulointi

Sosiaalinen manipulointi (engl. Social engineering) on menetelmä, jossa käyttäjää huijataan paljastamaan tai antamaan pääsy salattuihin tai henkilökohtaisiin tietoihin. Huijari voi saada uskottavuutta ja käyttäjän luottamuksen esiintymällä luotettavana tahona, kuten sivuston ylläpitäjänä. Parantaakseen luottamusta huijari voi kerätä etukäteen tietoa uhrista. Sosiaalinen manipulointi voi esiintyä missä muodossa tahansa, esimerkiksi kasvotusten, puhelinsoiton kautta, kirjeitse tai verkossa. Yleisimmin tapaukset ilmenevät kuitenkin internetissä, esimerkiksi sähköpostitse.

Julkisuuteen on noussut paljon pankkitunnusten khalastelu (engl. phishing). Khalastelu on sosiaalisen manipuloinnin harjoittamista verkossa. Monet suomalaiset pankkiasiakkaat, muun muussa Nordean ja Sammon asiakkaat, ovat joutuneet khalastelun uhreiksi. Asiakkaat ovat saaneet sähköpostiviestejä, joissa heitä pyydetään lähettämään omat pankkitunnuksensa vastausviestillä tai seuraamalla viestissä ollutta linkkiä. Linkki näyttää siltä, että se vie pankin viralliselle sivustolle, mutta todellisuudessa käyttäjä ohjataan väärennetylle sivustolle. Sivustolla käyttäjää pyydetään kirjautumaan sisään, jolloin huijarit saavat käsiinsä käyttäjän verkkopankkitunnukset. Huijareiden syyt tietojen tarpeeseen ovat olleet yleensä joko tarve päivittää käyttäjän tietoja, tietojärjestelmien

päivitys tai väittämät, että asiakkaan tileihin on murtauduttu ja asian korjaamiseksi asiakkaalta vaaditaan pankkitunnukset tunnustautumista varten. Myös tunnusten vanhentuminen on yksi tunnetuimpia väittämiä, joita huijarit käyttävät. Huijareiden tarinat kehittyvät jatkuvasti paremmiksi ja ne voivat olla hyvinkin uskottavia.

Paras menetelmä erottaa khalasteluyritykset oikeista viesteistä on lukea viestit tarkasti ja huolellisesti, ennen kuin lähtee tekemään mitään. Mahdolliset uhkaukset ja hoputtelu viesteissä on todennäköisesti merkki siitä, että kyseessä on huijausyritys. Lähettäjän sähköpostiosoite on hyvin helppo väärentää, joten ei kannata missään vaiheessa luottaa sokeasti lähettäjän aitouteen. Esimerkiksi `support@nordea.fi` osoitteesta lähetetty viesti voi todellisuudessa olla lähetetty ihan eri osoitteesta ja ainoastaan naamioitu Nordean sähköpostiosoitteeksi. Toistaiseksi suomenkieliset huijausviestit ovat usein kirjoitettu hyvin huonolla suomenkielellä ja ne voi helposti huomata huijausyritykseksi. Usein myös englanninkielisissä viesteissä esiintyy paljon kirjoitusvirheitä. On kuitenkin vain ajan kysymys, jolloin huijausviestit alkavat näyttää ja kuulostaa hyvin aidoilta ja kieliopillisesti miltei virheettömiltä viesteiltä.

Sosiaalista manipulointia vastaan on hankalaa ja miltei mahdotonta taistella. Tärkeintä on kuitenkin pitää käyttäjät ajan tasalla valistamalla heitä huijaustapauksista. Näissä tapauksissa ikinä ei ole liikaa korostaa käyttäjille, että ammattimaisen sivuston tai palvelun ylläpitäjät eivät kysy käyttäjiltä heidän salasanojan tai henkilökohtaisia tietoja missään tapauksessa.

# 6 Yhteenveto

WWW-palvelut ovat nykyään hyvin yleisiä ja kilpailukykyisiä ratkaisuja verrattuna tavallisiin ohjelmistoihin. WWW-palveluiden ongelmat ovat puutteelliset tietoturvaratkaisut. Kaikki palveluiden tuottajat eivät kiinnitä tarpeeksi huomiota tietoturvaan. Vaikka pelkästään it-pohjainen tietoturva ei ole ratkaisu kaikkeen ongelmaan, se on yksi tietoturvan peruselementeistä.

Hyvin toteutetun tietoturvaratkaisun heikoin kohta on usein käyttäjä. Tietoturvaan voi upottaa valtavia määriä rahaa, siksi on välttämätöntä tuntea oman järjestelmän vaatimukset ja tietoturvataso. Mitä suosituimmaksi palvelu tulee, sitä enemmän se vaatii tietoturvalta. Verkkohyökkäykset kehittyvät jatkuvasti, siksi tietoturvaa ei voi ajatella kertaluonteisena asiana. Tietoturvan pitää pysyä mukana kehityksessä, sillä vuosia sitten toteutettu turvallinen palvelu ei välttämättä enää ole turvallinen. Kuten sovellusohjelmistot myös www-palvelut tarvitsevat jatkuvaa ylläpitoa.

Kaikki salaukset ovat murrettavissa, sillä kyse on vain siitä, paljonko tarvitaan resursseja ja aikaa. Kun käytetään lyhyttä avainta, salausmenetelmä on aina turvaton. Pitkän avaimen käytössä menetelmä saattaa olla turvallinen. Vanha salausmenetelmä, jonka heikkoudet tunnetaan voi olla turvallisempi kuin uusi menetelmä, jota ei vielä ehditty riittävästi analysoida. Sillä vasta myöhemmin ajan kuluessa voidaan nähdä, miten turvalliseksi menetelmä osoittautuu. Tietoturvan kokonaisuuden kannalta salaustekniikoilla on lopulta kuitenkin vain marginaalinen merkitys. (Järvinen, 2003 s. 79)

WWW-palveluiden toteutuksessa uhkia ja palvelun heikkouksia kannattaa miettiä alkumetreiltä asti. Tietoturva unohtuu usein täysin tai siihen ei jakseta panostaa enää palvelun toteutuksen jälkeen. Jos vain mahdollista, käyttäjiä pitäisi kouluttaa ja tiedottaa verkon uhista. Ohjelmistojen päivitys on hyvin tärkeä tekijä tietoturvan kannalta. Käyttäjien ohjelmistot pitäisivät olla ajan tasalla, sillä usein uudet versiot eivät pelkästään lisää uusia ominaisuuksia, mutta myös paikkaavat ohjelmien vanhojen versioiden haavoittuvuuksia. Ottamalla huomioon mainitut asiat, useimpien palveluiden kannalta saavutetaan täysin riittävä tietoturvataso.

## Lähteet

**Alestalo, Antti. 2010.** WWW-palvelun tietoturva. *Noppa-Portaali.* [Online] 28. 1 2010. [Viitattu: 5. 12 2010.] https://noppa.tkk.fi/noppa/kurssi/t-111.4360/luennot/T-111_4360_tietoturva1.pdf.

**Aura, Tuomas. 2010.** Computer security overview. *Noppa-portaali.* [Online] 8. 9 2010. [Viitattu: 5. 12 2010.] T-110.4206 Information Security Technology course lecture. https://noppa.tkk.fi/noppa/kurssi/t-110.4206/luennot/T-110_4206_lecture_01__security_overview.pdf.

**Bodmer, Fabrice. 2007.** Cross-Site Scripting (XSS). *Department of Informatics.* [Online] 2007. [Viitattu: 27. 11 2010.] http://diuf.unifr.ch/drupal/tns/sites/diuf.unifr.ch.drupal.tns/files/Teaching/2006_2007/Computer_Security_Threats_and_Counter_Measures/Bodmer_CrossSiteScripting.pdf.

**Burkholder, Peter. 2002.** SSL Man-in-the-Middle Attacks. [Online] 1. 2 2002. [Viitattu: 20. 10 2010.] http://74.125.155.132/scholar?q=cache:EOpJ6WVr5OEJ:scholar.google.com/+SSL+man+in+the+middle&hl=fi&as_sdt=2000.

**Carnegie Mellon University. 1999.** CERT/CC Denial of Service. [Online] 1999. [Viitattu: 27. 11 2010.] http://www.cert.org/tech_tips/denial_of_service.html.

**—. 1997.** CERT® Advisory CA-1996-01 UDP Port Denial-of-Service Attack. [Online] 1997. [Viitattu: 27. 11 2010.] http://www.cert.org/advisories/CA-1996-01.html.

**—. 2000.** CERT® Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attack. [Online] 2000. [Viitattu: 27. 11 2010.] http://www.cert.org/advisories/CA-1996-21.html.

**Dierks, T ja Rescorla, E. 2008.** The Transport Layer Security Protocol. [Online] RFC5246 2008. [Viitattu: 21. 10 2010.] http://www.ietf.org/rfc/rfc5246.txt.

**Fielding, R;ym. 1999.** Hypertext Transfer Protocol -- HTTP/1.1. [Online] RFC2616 1999. [Viitattu: 19. 10 2010.] http://tools.ietf.org/html/rfc2616.

**Grossman, Jeremiah. 2009.** *WhiteHat Website Security Statistics Report.* CA : WhiteHat Security, 2009. A WhiteHat Security Whitepaper. http://www.whitehatsec.com/home/assets/WPStatsreport_100107.pdf.

**IBM.** History of SSL. [Online] [Viitattu: 20. 10 2010.] http://publib.boulder.ibm.com/infocenter/iseries/v5r3/index.jsp?topic=%2Frzain%2Frzainhistory.htm.

**Järvinen, Petteri. 2003.** *Salausmenetelmät.* Porvoo : Docendo, 2003. ISBN 951-846-183-X.

**Menezes, Alfred J.;van Oorschot, Paul C. ja Vanstone, Scott A. 2001.** Handbook of Applied Cryptography. *Handbook of Applied Cryptography.* [Online] 8 2001. [Viitattu: 21. 11 2010.] http://www.cacr.math.uwaterloo.ca/hac/. 0-8493-8523-7.

**Microsoft. 2003.** SSL/TLS in Detail. [Online] 31. 7 2003. [Viitattu: 22. 10 2010.] http://technet.microsoft.com/en-us/library/cc785811%28WS.10%29.aspx.

**Mård, Anna. 2007.** Pankkien tietoturva jää liian haperoksi. [Online] IT-viikko, 25. 10 2007. [Viitattu: 1. 12 2010.] http://www.itviikko.fi/tietoturva/2007/10/25/pankkien-tietoturva-jaa-liian-haperoksi/200726702/7.

**Oh, Soohyun;Kwak, Jin ja Won, Dongho. 2003.** *Networking Technologies for Enhanced Internet Services.* Heidelberg : Springer Berlin, 2003. ISBN 978-3-540-40827-7.

*On Information Security Paradigms.* **Canal, Vicente Aceituno. 2005.** Spain : s.n., 2005, ISSA Journal. http://www.issa.org/Library/Journals/2005/September/Aceituno%20Canal%20-%20On%20Information%20Security%20Paradigms.pdf.

**Papazoglou, Michael P. 2008.** *Web Services: Principles and Technology.* Harlow : Pearson Education Limited, 2008. 978-0-321-15555-9.

**Rescorla, E ja Schiffman, A. 1999.** The Secure HyperText Transfer Protocol. [Online] RFC2660 1999. [Viitattu: 20. 10 2010.] http://tools.ietf.org/html/rfc2660.

**SciEngines.** Break DES in less than a single day. [Online] [Viitattu: 22. 10 2010.] http://www.sciengines.com/company/news-a-events/74-des-in-1-day.html.

**Taloussanomat. 2007.** Pankkien tietoturva ei lipsu Suomessa. [Online] IT-viikko, 26. 10 2007. [Viitattu: 1. 12 2010.] http://www.itviikko.fi/tietoturva/2007/10/26/pankkien-tietoturva-ei-lipsu-suomessa/200726785/7.

**Ylitalo, Timo. 2004.** Luottamus ja tietoturva pankkipalveluissa. *Tietoyhteiskunnan kehittämiskeskus ry*. [Online] 21. 10 2004. [Viitattu: 1. 12 2010.] http://www.tieke.fi/mp/db/file_library/x/IMG/15044/file/TimoYlitalo.pdf.