Pentti Laitinen

# JAVA JA TIETOTURVA

JYVÄSKYLÄN YLIOPISTO
TIETOJENKÄSITTELYTIETEIDEN LAITOS
2013

# TIIVISTELMÄ

Laitinen, Pentti
Java ja tietoturva
Jyväskylä: Jyväskylän yliopisto, 2013, 26s.
Tietojärjestelmätiede, kandidaatin tutkielma
Ohjaaja: Hirvonen, Pertti

Tietoturvan tärkeys korostuu vuosi vuodelta. Java-alustaa käyttää yli miljardi tietokonetta ja vuosina 2011-2013 sitä kohtaan on kohdistunut tavallista enemmän haavoittuvuuksia. Tässä kirjallisuuskatsauksen muodossa suoritetussa tutkielmassa ttutkitaan, miten Javan tietoturva on kehittynyt sen versioiden aikana ja minkä tyyppisiä hyökkäyksiä siihen on kohdistunut. Tutkimuksessa käytetään apuna Javan dokumentaatiota ja muuta tietoturvakirjallisuutta.

Tutkimuksessa havaittiin, että Javassa on huomioitu tietoturva sen ensimmäisistä versioista asti. Lisäksi havaittiin, että malli, jolla Java toteuttaa tietoturvanhallinnan, on pysynyt pitkälti samana kautta Javan eliniän. Haavoittuvuudet kohdistuvat pääasiassa Javan selainliitännäiseen, joka mahdollistaa Javan suorittamisen selaimessa. Huolestuttavana asiana havaittiin, että useat käyttäjät käyttävät todella vanhoja versioita ohjelmistosta ja ovat siis alttiita hyökkäyksille.

Asiasanat: Java, tietoturva, ohjelmointikielet, haavoittuvuus

# ABSTRACT

Laitinen, Pentti
Java and security / Java and information security
Jyväskylä: University of Jyväskylä, 2013, 26 p.
Information Systems, Balchelor Thesis
Supervisor: Hirvonen, Pertti

Importance of information security has grown year by year. Popular software platform Java is installed on over billion computers across the world and it has been riddled with vulnerabilities during years 2011-2013. This dissertation, which has been made in form of literature review, tries to answer how Java information security has developed during its versions and what kind of vulnerabilities it has faced during this. Along with information security literature, documentation of Java is being used to answer these questions.

During examination it was noticed that Java has acknowledged importance of information security throughout its versions. Observations also concur that Java security model has not changed radically during Java's lifespan. Java vulnerabilities are mostly related to Java's browser plug-in, which allows Java to be ran inside browsers. It was worrying to find out how many users still use old versions of the software and expose themselves to security risks related to old software.

Keywords: Java, information security, programming languages, vulnerability

# KUVIOT

KUVIO 1: JDK 7 rakenne (Oracle, 2013m).....9
KUVIO 2: Java tietoturvamallit (Gong ym., 2013).....12
KUVIO 3: OWASP TOP 10 (OWASP Foundation, 2013b s.4).....14

# SISÄLLYS

TIIVISTELMÄ....2

ABSTRACT....3

KUVIOT....4

SISÄLLYS....5

1 JOHDANTO....7

2 JAVAN YLEISRAKENNE JA TIETOTURVAN KÄSITE....8

2.1 Javan rakenne....8
2.2 Tietoturva....9

3 JAVAN TIETOTURVAN HISTORIA....11

3.1 Versiot 1.0-1.1....11
3.2 Versiot 1.2 – 1.6....12
3.3 Java 7....13

4 MERKITTÄVIÄ HAAVOITTUVUUKSIA....14

4.1 Hybriditiedostohyökkäykset....15
4.2 Flashback-troijalainen....15

5 TURVALLISUUTTA EDISTÄVÄT KEINOT....17

5.1 Ohjelmointirajapinnat....17
5.2 Java-virtuaalikone ja SecurityManager....18

6 POHDINTA....19

6.1 Turvallisuuden parannuskeinoja....20
6.2 Java 8....20

7 YHTEENVETO....22

LÄHTEET.....24

# 1 JOHDANTO

Java-alusta toimii monilla erilaisilla laitteilla ja mahdollistaa Java-kielen sovellusten suorittamisen useilla käyttöjärjestelmillä. Java on internetin historian aikana ollut yksi suosituimmista kielistä ja on yhä TIOBEN suosituimpien ohjelmointikielien listan kärjessä (TIOBE Software BV, 2013). Yli miljardi tietokonetta ja kolme miljardia puhelinta käyttää Javaa (Oracle, 2013a).

Javaan on sen eliniän aikana kohdistunut useita haavoittuvuuksia. Uudet haavoittuvuudet ovat nousseet uutisotsikoihin vuosina 2011-2013. Viranomaistahot ovat suositelleet Javan poistamista tietokoneelta, jos käyttäjä ei sitä tarvitse (US-CERT, 2013). Suuren käyttäjämäärän takia voi haavoittuvuus Javassa pahimmillaan altistaa miljardeja käyttäjiä erilaisille riskeille.

Vuonna 1995 julkaistun Javan ensimmäisestä versiosta on kulunut jo 18 vuotta ja tänä aikana Java on saanut useita uusia versioita. Javan tietototurvaa koskevissa tutkimuksissa ei kuitenkaan ole tarkasteltu versioiden välisiä muutoksia ja näiden muutosten vaikutuksia pitkällä aikavälillä.

Javan tietoturvan historiaan keskittyvien tutkimusten määrä on vähäinen. Tässä tutkielmassa on tarkoitus tutkia aiheeseen liittyvän kirjallisuuden ja dokumenttien avulla, miten Javan tietoturva on kehittynyt sen elinkaaren aikana ja mihin Javan osiin haavoittuvuudet ovat kohdistuneet. Tutkielmassa käydään läpi Javan versiohistoriaa versioiden 1.0 ja 1.7.0_25 (Java 7 Update 25) välillä. Samalla esitellään aiheen kannalta keskeistä tietoturvatermistöä ja tuodaan esille muutamia Javaan kohdistuneita haavoittuvuuksia. Tutkielman lopussa pohditaan myös, miksi Java on hyökkäysten kohteena ja miten ongelmakohtia voitaisiin ratkaista.

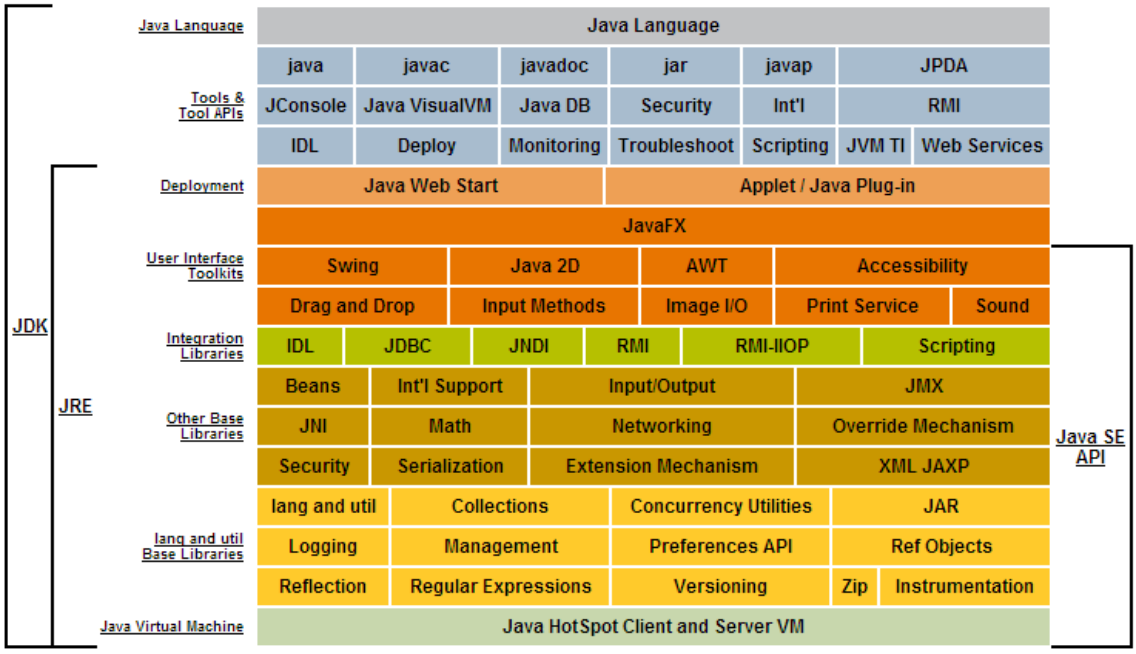
# 2 JAVAN YLEISRAKENNE JA TIETOTURVAN KÄSITE

Java-ohjelmointikielen ensimmäinen versio julkaistiin vuonna 1995 ja tästä lähtien Javan kehityksessä on korostettu tietoturvan tärkeyttä (Gong, 2009; Sun Microsystems, 1997a). Kieltä kuvattiin kaiken muun ohessa myös turvalliseksi, mutta tämä on entisen JavaSoftin tietoturva-arkkitehtuuri Li Gongin, mukaan (käännös): *"parhaimmillaan epätarkkaa ja usein täysin väärin kuvattaessa mitä tahansa systeemiä"* (Gong, 2009). Tässä luvussa käsitellään lyhyesti Javan rakennetta ja esitellään tietoturvatermistöä.

## 2.1 Javan rakenne

Versiosta 1.0 lähtien Javaan on tuotu täysin uutta sen merkittävissä versioissa kuten JDK 1.1, 1.2, 1.3 jne. Pienemmät päivitykset, esimerkiksi JDK 1.1.2 ja 1.2-.3, ovat sen sijaan sisältäneet vain ohjelmointivirheiden korjauksia ja pieniä lisäyksiä (Gong, 2009.) Java on saatavilla neljällä eri alustalla, jotka ovat Java Standard Edition, Java Enterprise Edition, Java Micro Edition ja JavaFX (Oracle, 2012). Javan nimeämismalli on myös muuttunut versioiden aikana muutamaan otteeseen. Esimerkiksi Java Standard Edition tunnettiin Java 2 Standard Editionina versiosta 1.2 versioon 1.5 asti. Nimeämisen takana oli versiossa 1.2 tulleiden merkittävien uudistuksien korostaminen (Gong, 2009). Versiosta 1.6 lähtien nimeämiseen on käytetty vain pääversionumeroa eli Java 6, Java 7 ja Java 8.

Java Development Kit eli JDK koostuu Java Runtime Enviromentistä (JRE), jonka mukana on Java Virtual Machine (JVM). Lisäksi JDK sisältää kaikki Javan ohjelmointirajapinnat (API) ja kirjastot (eng. library) (kuvio 1). Java Virtual Machine eli Javan virtuaalikone on se työkalu Javassa, jonka avulla Java-ohjelmia suoritetaan. Samalla se on vastuussa Javan tietoturvan hallinnasta ohjelmien suorituksen aikana. Luvussa 5 käsitellään Javan tietoturvasysteemiä ja miten Javan virtuaalikone ja SecurityManager-luokka toimivat.



KUVIO 1: JDK 7 rakenne (Oracle, 2013m)

## 2.2 Tietoturva

Tietoturva on Tietotekniikan liiton ATK-sanakirjan mukaan: *"tavoitetila, jossa tiedot, tietojärjestelmät ja palvelut saavat asianmukaista suojaa niin, että niiden luottamuksellisuuteen, eheyteen ja käytettävyyteen kohdistuvat uhat eivät aiheuta merkittävää vahinkoa yhteiskunnalle ja sen jäsenille"* (Tietotekniikan liitto ry, 2008). Javan tietoturvallisuutta tarkastellessa keskitytään ohjelmointikielen ja sen mukana tulevien työkalujen, kirjastojen, lisäosien ja laajennusten tietoturvaan. Tarkastelualueeseen eivät kuulu Java-kielellä toteutetut ohjelmistot, joissa haavoittuvuuden aiheuttajana on ollut ohjelmiston kirjoittajan virhe. Virhe voi olla muun muassa se, että ohjelmoija ei noudata hyviä ohjelmointitapoja, kuten esimerkiksi Oraclen turvallisen ohjelmoinnin ohjeistusta (Oracle, 2013b).

Sanastokeskus TSKn määritelmä haavoittuvuudelle on: *"Haavoittuvuus voi olla mikä tahansa heikkous, joka mahdollistaa vahingon toteutumisen tai jota voidaan käyttää vahingon aiheuttamisessa. Esimerkiksi ohjelmistossa voi olla haavoittuvuus, joka mahdollistaa järjestelmän väärinkäytön."* (Sanastokeskus TSK, 2004, s. 14). Suomen kielessä käytetään haavoittuvuus (eng. vulnerability) ja tietoturva-aukko termiä rinnakkain (Sanastokeskus TSK, 2004, s. 14). Java-haavoittuvuuksista puhuttaessa sillä tarkoitetaan siis jotakin heikkoutta, esimerkiksi Java Development Kitin osassa, jolla voidaan aiheuttaa vahinkoa tai jota voidaan käyttää vahingon aiheuttamiseen.

Suomen viestintävirastossa toimiva tietoturvaviranomainen CERT-FI on varoittanut vuoden 2013 aikana jo useampaan otteeseen Java-haavoittuvuuksista eli tietoturva-aukoista (CERT-FI, 2013). Myös muiden valtioiden tietoturvaviranomaiset, kuten Yhdysvaltojen Department of Homeland Security, ovat varoittaneet haavoittuvuuksista Javassa ja suositelleet Javan käytöstä poistamista, mikäli käyttäjällä ei ole sille tarvetta (US-CERT, 2013). Tietoturvan tärkeys on

kasvanut ympäri maailmaa, varsinkin kun yhä enemmän ja enemmän yhteiskunnalle tärkeistä toiminnoista on siirtynyt elektroniseksi verkkoon. Esimerkiksi yhdysvaltojen kansantalous muuttuu yhä riippuvammaksi ja riippuvammaksi internetistä, uhkien määrän samalla kasvaessa (Ghosh & McGraw, 2012).

Javassa on myös esiintynyt useampia nollapäivähaavoittuvuuksia (eng. Zero-day) vuosien 2011-2013 aikana. Nollapäivähaavoittuvuus tarkoittaa haavoittuvuutta, joka on paljastunut, mutta johon ei ole vielä olemassa korjausta (Valtionvarainministeriö, 2008, s. 67). Tietoturvan tärkeyden kasvusta hyvän kuvan antaa myös se, kuinka pienestä tietoturva-alasta on kasvanut nopeasti valtava viimeisessä kymmenessä vuodessa (Ghosh & McGraw, 2012).

# 3 JAVAN TIETOTURVAN HISTORIA

Tässä luvussa käsitellään merkittäviä Javan tietoturvaan kohdistuneita uudistuksia, etenemällä vanhimmista versioista aina versioon 7 asti. Tarkastelu kohdistuu Java Standard Editioniin ellei toisin mainita.

## 3.1 Versiot 1.0-1.1

Javassa tietoturva koettiin tärkeäksi jo ensimmäisessä versiossa. Tämä on ymmärrettävää, sillä Java toi kunnolla ensimmäisten joukossa selaimeen ladattavat ohjelmat, appletit eli sovelmat (eng. Applet) (Dean, Felten, & Wallach, 1996). Tämänkaltainen etänä ladattava koodi toikin uusia tietoturvaongelmia, joita Javan hiekkalaatikoinnin ja SecurityManagerin on tarkoitus estää (Dean ym., 1996; Gong, Oracle, & Sun Microsystems, 2013). Sovelmat mahdollistavat toisen tai kolmannen osapuolen hyökkäykset. Kolmannen osapuolen hyökkäys tarkoittaa sitä, että saastunut sovelma voi olla WWW-palvelimella, jolla ei ole tietoa sovelman saastuneisuudesta. Samalla saastunut sovelma lähettää suoraan hyökkääjälle tietoja käyttäjästä. (Dean ym., 1996.)

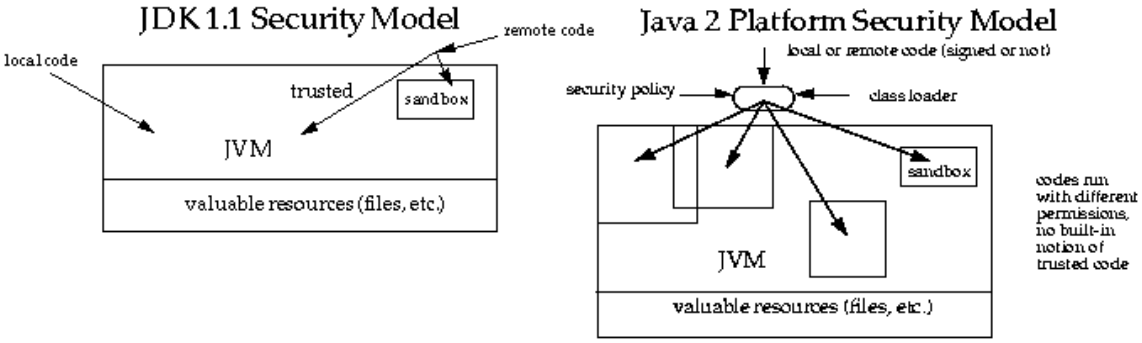
Vaikka tietoturvan tärkeyttä korostettiin, havaittiin jo ensimmäisessä versiossa (1.0) heikkouksia, joita voitiin hyödyntää palvelunestohyökkäyksiin (eng. Denial-of-service). Esimerkiksi HotJava-selaimessa pystyttiin hyökkäyksellä estämään käyttäjää lataamasta uusia web-osoitteita hyökkäyksen jälkeen. Netscape-selaimessa hyökkääjä pystyi hyödyntämään java.net.InetAddress-luokan heikkoutta ja estämään uusien verkkoyhteyksien muodostaminen (Dean ym., 1996). Lisäksi Javan tyyppisysteemiä pystyi kiertämään luomalla oman ClassLoader-luokan (Dean ym., 1996). ClassLoader-luokan avulla voidaan ladata muita luokkia virtuaalikoneeseen (Oracle, 2013c). Luomalla oman ClassLoader-luokan hyökkääjä pystyi muuttamaan luokkahierarkiaa ja asettamaan arvoja muuttujille, jotka yleensä ovat muokattavissa vain ohjelman suorituksen aikana (Dean ym., 1996). Version 1.0 tarkastelussa Dean & ym. huomasivat myös heikkouden Javan pakkausten (eng. package) nimien tarkastuksessa. Jos Javan pak-

kauksen nimeämisen aloitti pistemerkillä (”.”), niin merkki korvautuisi kauttaviivaksi (”/”) ohjelman suorituksen aikana. Tämä mahdollistaisi tiedostojen lataamisen mistä vaan tietokoneen kansiosta, sillä absoluuttiset tiedostosijainnit alkavat kauttaviivamerkillä (Dean ym., 1996).

Vuonna 1997 julkaistiin JDK 1.1, jonka suurin tietoturvan parannus versioon 1.0 verrattuna oli koodin allekirjoittaminen (eng. code signing). Tämän avulla sovelmien koodin pystyi palisesti allekirjoittamaan ja tämän allekirjoituksen tarkastaminen oli mahdollista sovelman suorituksen aikana (Gong, 2009). Useat edellä mainitut viat oli korjattu versioon 1.1 mennessä (Dean ym., 1996). Version 1.1 julkaisu-uutisessa mainitaan uusien tietoturvaohjelmointirajapintojen tulo Javan osaksi (Sun Microsystems, 1997b), mutta valitettavasti tarkka versiohistoria on vajavainen. Vanhojen versioiden tarkastelun tekee vaikeaksi Javan omistajan muuttuminen Sun Microsystemsistä Oracleen versioiden aikana. Oraclen ostaessa Sun Microsystems vanhat Sun Microsystemsin sivustot poistuivat verkosta. Tämä tarkoittaa sitä, että vanhoja versiohistorioita ei enää ole helposti saatavilla, sillä Oracle ei näitä omille sivuilleen siirtänyt.

## 3.2 Versiot 1.2 – 1.6

Java 2 eli versio 1.2 toi mukanaan suuria uudistuksia tietoturvaan. Sen suunnittelu aloitettiin jo vuoden 1996 loppupuolella ja se julkaistiin vuonna 1998 (Gong, 2009). Ennen versiota 1.2 toimittiin yksinkertaisen hiekkalaatikkomallin (eng. sandbox model) avulla (kuvio 2). Hiekkalaatikointi toimi periaatteella ”kaikki tai ei mitään”, joko sallien tai eväten kaikki oikeudet (Gong, 2011). Periaatteessa tässä mallissa kaikkea paikallista koodia pidettiin luotettavana ja sille sallittiin oikeudet systeemin resursseihin, kun taas etäkoodia (eng. remote code) käsiteltiin epäluotettavana ja se suoritettiin Javan hiekkalaatikossa rajoitetuin oikeuksin (Gong ym., 2013). Versiossa 1.2 tietoturvamallia uudistettiin siten, että etä- ja paikallinen koodi joutuvat käymään läpi samat pääsynvalvonnat (eng. access control) (kuvio 2) (Gong ym., 2013.) JDK 1.2 mukana tuotu hienojakoinen pääsynvalvontamalli on pysynyt nykyversioihin asti varsin samana (Gong, 2009).




KUVIO 2: Java tietoturvamallit (Gong ym., 2013)

Versio 1.2 oli suuri uudistus Javan tietoturvahallintaan ja sen jälkeiset versiot eivät ole sisältäneet yhtä suuria muutoksia rakenteeseen. Version 1.2 jälkeisissä versioissa on korjattu virheitä ja lisätty ominaisuuksia entisen päälle, rakenteen pysyessä samana (Oracle, 2000, 2003, 2004, 2011). Versio 1.6 oli viimeinen Sun Microsystemsin alla julkaistu pääversio ja se julkaistiin 2006.

## 3.3 Java 7

Java 7 julkaistiin vuonna 2011 ja on sisältänyt ennätysmäärän tietoturvapäivityksiä tähänastisista versioista. Perinteisestä kolmesta vuosittaisesta Java tietoturvajulkaisusta siirryttiin neljään julkaisuun vuonna 2013. (Ramani, 2013.) Oracle on luvannut myös julkaisevansa tietoturvapäivityksiä riippumatta aikataulusta, jos kriittisiä uusia haavoittuvuuksia ilmenee (Oracle, 2013d).

Tähän mennessä julkaistuissa päivityksissä (JDK 7 Update 25) on muun muassa lisätty paranneltu tietoturvahuomautus sovelmien suorittamisen yhteyteen, joka kehottaa käyttäjää päivittämään uuteen versioon, jos käytössä on vanha versio Javan Runtime Enviromentistä (Oracle, 2013e). Lisäksi käyttäjälle on annettu enemmän tietoturva-asetuksia Javan ohjauspaneeliin, joiden avulla käyttäjä voi paremmin kontrolloida Javan tietoturva-asetusten tasoa itselleen sopivaksi (Oracle, 2013f).

Ehkä merkittävin version seitsemän aikana tullut tietoturvapäivitys julkaistiin päivityksessä 21 (JDK 7 Update 21), kun allekirjoitettujen sovelmien (eng. signed applets) tietoturvamalli muutettiin, liitännäisiin liittyviä oletusasetuksia tiukennettiin ja sertifikaattien mustan listan tarkistus mahdollistettiin (Oracle, 2013g). Musta lista päivittyy päivittäin ja sen tarkoituksena on estää varastettujen ja kumottujen sertifikaattien käyttö (Oracle, 2013g).

# 4 MERKITTÄVIÄ HAAVOITTUVUUKSIA

Haavoittuvuuksien kohteet vaihtelevat vuosi vuodelta. Open Web Application Security Project (OWASP) pitää yllä listaa kymmenestä suurimmasta tietoturvariskityypistä, jotka kohdistuvat web-sovelluksiin (OWASP Foundation, 2013a). Vuotta 2013 edeltävinä vuosina listaa on päivitetty kolmen vuoden välein ja vuodelta 2007 on saatavilla erillinen Java EE (Enterprise Edition) listaus. Java EEn listan kärkikolmikossa ovat Cross Site Scripting (XSS), injektio-heikkoudet (eng. Injection Flaws) ja haitallisen tiedoston suoritus (eng. Malicious File Execution) (Rodrigues & Leader, 2007). Yleisessä listauksessa vuosien 2010 ja 2013 välillä ei ole ollut muutoksia listan kolmen kärkeen (kuvio 3) ja XSS ja injektio löytyvät myös yleisestä listauksesta (OWASP Foundation, 2013b s.4).

| OWASP Top 10 – 2010 (Previous) | OWASP Top 10 – 2013 (New) |
|---|---|
| A1 – Injection | A1 – Injection |
| A3 – Broken Authentication and Session Management | A2 – Broken Authentication and Session Management |
| A2 – Cross-Site Scripting (XSS) | A3 – Cross-Site Scripting (XSS) |
| A4 – Insecure Direct Object References | A4 – Insecure Direct Object References |
| A6 – Security Misconfiguration | A5 – Security Misconfiguration |
| A7 – Insecure Cryptographic Storage – Merged with A9 → | A6 – Sensitive Data Exposure |
| A8 – Failure to Restrict URL Access – Broadened into → | A7 – Missing Function Level Access Control |
| A5 – Cross-Site Request Forgery (CSRF) | A8 – Cross-Site Request Forgery (CSRF) |
| <buried in A6: Security Misconfiguration> | A9 – Using Known Vulnerable Components |
| A10 – Unvalidated Redirects and Forwards | A10 – Unvalidated Redirects and Forwards |
| A9 – Insufficient Transport Layer Protection | Merged with 2010-A7 into new 2013-A6 |

KUVIO 3: OWASP TOP 10 (OWASP Foundation, 2013b s.4)

Javan elinkaaren aikana siihen on kohdistunut useita erilaisia haavoittuvuuksia. Viimeaikaiset uutisissa mainitut haavoittuvuudet ovat kohdistuneet Javan selainliitännäiseen (eng. Java Browser Plug-in), joka mahdollistaa Javan suorittamisen selaimen sisällä (Ramani, 2013). Merkittäviksi haavoittuvuuksiksi voidaan luokitella esimerkiksi GIFAR-haavoittuvuus, koska se poikkesi tavan-

omaisista haavoittuvuuksista. Tässä luvussa käsitellään muutamia mielenkiintoisia haavoittuvuuksia ja mitä niillä on voinut saada aikaiseksi.

## 4.1 Hybriditiedostohyökkäykset

GIFAR- (Graphics Interchange Format Java Archive) ja PDFAR-hyökkäykset (Portable Document Format Java Archive) olivat hybriditiedostohyökkäyksiä, joissa esimerkiksi GIF-muotoiseen kuvatiedostoon oli yhdistetty JAR-tiedosto (Java Archive) (Stuttard & Pinto, 2011, s. 485-486). Kyseisen tyyppinen tiedosto mahdollisti tiedostontyyppitestin läpipääsyn, sillä GIF-formaatin metatieto on tiedoston alussa ja JAR-formaatin metatieto tallentuu tiedoston loppuun (Stuttard & Pinto, 2011, s. 485-486). Hyökkääjä pystyi käyttämään tämäntyyppistä tiedostoa esimerkiksi sivustoilla, joihin kuvatiedostojen lataaminen oli sallittua. Tiedoston avulla voitiin varastaa sivuston käyttäjien tietoja tai esimerkiksi selainistuntoja, mikäli käyttäjä oli aiemmin antanut oikeuden sivustolla sovelmien suoritukseen. Java näki GIFAR-tiedoston palvelimen omaksi tiedostoksi ja antoi oikeuden saman alkuperän perusteella sen suoritukselle (Dhanjani, Rios, & Hardin, 2009, s. 54-62; Stuttard & Pinto, 2011, s. 485-486).

Kyseinen hyökkäysmenetelmä korjattiin Javan versio 6 päivityksessä yksitoista (National Institute of Standards and Technology, 2011). GIFAR on hyvä esimerkki, kuinka hybriditiedostohyökkäys toimii ja kuinka menetelmää voidaan hyödyntää muihinkin tiedostotyyppeihin (Dhanjani ym., 2009, s. 54-62). Periaatteessa menetelmää pystyi tai pystyy hyödyntämään muihinkin kieliin ja se ei siten ollut pelkästään Javaan kohdistuva heikkous. Kuitenkin hyökkäysmenetelmä antaa hyvän kuvan, siitä kuinka yksinkertainen asia voi läpäistä tietoturvatarkistukset.

## 4.2 Flashback-troijalainen

Flashback oli vuonna 2011 löydetty troijalainen, jonka eri versiot hyödynsivät useampaa eri Java-haavoittuvuutta (Cheng, 2012; Oh, 2012). Flashback on erittäin kehittynyt Mac OS X haittaohjelma ja se oli ensimmäinen haittaohjelma, joka osasi sammuttaa OS X:n käyttöjärjestelmän oman haittaohjelmilta suojaavan ohjelman Xprotectin. Ensimmäisissä Flashbackin variaatioissa käyttäjän tietokone saastutettiin ohjaamalla käyttäjälle vahingolliselle sivustolle, esimerkiksi epäaidon Google hakusivuston kautta. Sivuston latautuessa käyttäjän eteen tuli kuva. Kuvan tarkoitus oli saada käyttäjä luulemaan, että Adoben Flashplayer on kaatunut ja se tarvitsee päivittämistä. Jos käyttäjä painoi päivitä näppäintä, haittaohjelma latautui koneelle. (Aquilino, 2012.)

Ensimmäiset versiot siis tarvitsivat käyttäjän luvan latautuakseen koneelle ja eivät hyödyntäneet Java-haavoittuvuuksia. Kuitenkin vuoden 2012 alkupuolen myöhempi versio troijalaisesta hyödynsi Java-haavoittuvuuksia CVE-2008-

5353 ja CVE-2011-3544, joiden avulla haittaohjelma pystyi latautumaan käyttäjän vain käydessä saastuneella sivustolla (eng. drive-by download) ja latautumaan ilman käyttäjän syötettä (Aquilino, 2012). Kyseisiin haavoittuvuuksiin oli jo olemassa korjaava Java-päivitys Flashbackin version ilmennyttyä. Jos troijalainen huomasi haavoittuvuuden korjaavan Java-päivityksen käyttäjän koneella, se hyödynsi vanhaa taktiikkaansa ja yritti huijata käyttäjää esittäen olevansa toinen ohjelma. (Aquilino, 2012.)

Applen hidas Java-päivityksien julkaisutahti OS X:lle mahdollisti jo korjatun Java-havoittuvuuden CVE-2012-0507 hyödyntämisen myöhemmissä Flashbackin versioissa. Flashback on tähän päivään mennessä onnistunut saastuttamaan yli 650 000 Mac-tietokonetta ja on laajiten levinnyt OS X haittaohjelma. (Aquilino, 2012.) Onkin ymmärrettävää, että Apple irtaantui Javasta ja poisti käyttöjärjestelmäpäivityksessään Javan selainlisäosan lokakuussa 2012 (Goodin, 2012).

# 5 TURVALLISUUTTA EDISTÄVÄT KEINOT

Tässä luvussa tarkastellaan, mitä Java Standard Edition Security pitää sisällään versiossa 7 ja miten Java estää haitallisen koodin suorittamista. Kuten aiemmin luvussa 2 mainittiin, Javan tietoturvaoikeuksista on vastuussa Java Virtual Machine, jonka sisäiset toiminnot jakavat oikeuksia tietyin periaattein Java-ohjelmille. Java on tyyppiturvallinen ohjelmointikieli eli kääntäjä voi huomata tyyppivirheitä koodissa ja huomauttaa näistä (Oracle, 2013b). Javassa on myös automaattinen roskienkeruu, muistin hallinta ja taulukkojen koon tarkistaminen (Oracle, 2013h). Oraclen mukaan nämä ominaisuudet samalla nopeuttavat ohjelmointia ja vähentävät virheitä koodissa, jolloin tuotettu koodi on turvallisempaa (Oracle, 2013h). Käsittelemme tässä luvussa näitä periaatteita ja muita Javan turvallisuutta edistäviä keinoja.

## 5.1 Ohjelmointirajapinnat

Javan mukana tulee paljon erilaisia ohjelmointirajapintoja (eng. API, Application Programming Interface). Ohjelmointirajapintojen on tarkoitus helpottaa ja nopeuttaa ohjelmointia. Osa Javan ohjelmointirajapinnoista on tarkoitettu parantamaan tietoturvallisuutta ja helpottamaan turvallisen koodin kirjoittamista. Java tarjoaa muun muassa rajapintoja salaustekniikoihin, julkisten avainten hallintajärjestelmiin (eng. PKI, Public Key Infrastructure), todennukseen ja pääsynhallinnan kontrollointiin (Oracle, 2013h).

Oraclen mukaan tietoturvallisuutta varten luodut ohjelmointirajapinnat on suunniteltu kolmen periaatteen mukaan. Ensimmäinen periaate on toteutuksen itsenäisyys (eng. Implementation independence) eli ohjelmien ei tarvitse itse toteuttaa tietoturvallisuutta vaan ne voivat kutsua useampia Java-alustan palveluita. Toisen periaatteen mukaan tarjoajat (eng. Provider) ovat käytettävissä yli ohjelmien (eng. Implementation interoperability), eli ohjelma ei ole sidottu yhteen tarjoajaan ja yksi tarjoaja ei ole sidottu yhteen ohjelmaan. Viimeinen periaate on algoritmien laajennettavuus (eng. Algorithm extensibility) eli mah-

dollistaa uusien ominaisuuksien, kuten uusien standardien, lisäämisen Javan yhteyteen. (Oracle, 2013i.)

## 5.2 Java-virtuaalikone ja SecurityManager

Java-virtuaalikone eli JVM on eräänlainen virtuaalinen tietokone ja se osa Javassa, joka mahdollistaa Java-koodin suorittamisen vapaasti eri käyttöjärjestelmissä ja eri laitteissa (Lindholm, Yellin, Bracha, & Buckley, 2013 s. 2). JVM osaa lukea vain class-muodossa olevaa tavukoodia. Tämä tiedostomuoto saadaan Javakääntäjän avulla Javan lähdekoodista. (Lindholm ym., 2013 s.2.) Class-muoto on siis eräänlainen välikieli. Tietoturvallisuutta edistääkseen JVM määrää vahvat syntaktiset ja muodolliset rajat Javan luokkatiedoston binäärikoodille (Lindholm ym., 2013 s.2). JVM on siis vastuussa luokkatiedostojen binäärikoodin oikeuden tarkistamisesta. Lisäksi JVM hoitaa automaattisen muistinhallinnan eli roskienkeruun (eng. Garbage collection) (Williams & Chitta). Roskienkeruu vähentää ohjelmoijan työtä automatisoimalla muistinhallinnan ja vähentää muistivuotojen riskiä. Roskienkeruun on myös tarkoitus estää puskurin ylivuotohyökkäykset ja se ennaltaehkäisee riskiä, että hyökkääjä pääsisi käsiksi ei haluttuihin tietoihin.

SecurityManager-luokka antaa ohjelmalle mahdollisuuksia luoda omia tietoturvasääntöjä. Luokan avulla ohjelma voi arvioida ennen suorittamista, ovatko operaatiot turvallisia vai ei ja täten sallia tai estää operaatioiden suorittamista (Oracle, 2013j). Jos ohjelmalle sallitaan oikeus, SecurityManager vain palaa suoritukseen, mutta mikäli operaatio estetään SecurityManager heittää SecurityException virheen (Oracle, 2013j). SecurityManagerin jakamat oikeudet on jaettu useampaan eri kategoriaan aina normaaleista tiedostonkäsittelyoikeuksista erilaisten yhteysoikeuksien kautta esimerkiksi Javan AWT-oikeuksiin (Abstract Window Toolkit) (Oracle, 2013j).

# 6 POHDINTA

Aiemmissa luvuissa on tarkasteltu kuinka, Java toteuttaa tietoturva-asiat ja minkälaisia haavoittuvuuksia Javassa on ollut. Mutta onko Java haavoittuvaisempi kuin muut kielet vai onko media vain liioitellut haavoittuvuuksia? Asiaa on vaikea tarkastella pelkän kirjallisuuden pohjalta ja tutkimuksessa tulisi ottaa huomioon kielten käyttäjien lukumäärät, kielten käyttötarkoitukset, pääasialliset käyttökohteet ja asioita, kuten kielten monipuolisuus. Haavoittuvuuksien esiintymismäärää on tutkittu. Esimerkiksi käytetyn kielen ja ohjelmoijan taustan pohjalta tehdyssä tutkimuksessa tultiin tulokseen, että molemmat asiat vaikuttavat haavoittuvuuksien esiintymismäärään (Bau, Wang, Bursztein, Mutchler, & Mitchell)

Javan 2013 huhtikuun kriittinen tietoturvapäivitys (Version 7 Update 21) sisälsi 42 uutta tietoturvakorjausta, mutta silti käyttäjät päivittivät keväällä hitaasti uuteen versioon (Websence Inc., 2013a). Päivitystä seuranneiden kahden päivän jälkeen vain kaksi prosenttia käyttäjistä oli päivittänyt uuteen versioon ja kuukaudessa päivittäneiden määrä oli noussut vain vähän yli seitsemään prosenttiin (Websence Inc., 2013a). Ongelma ei ole pelkästään kotikäyttäjien kohdalla vaan myös yritykset ovat hitaita päivityksien suhteen (Websence Inc., 2013a). Tietoturvayritys Websence on myös havainnut, että huolestuttavan moni käyttäjä käyttää yli vuodenkin vanhoja versioita Javasta (Websence Inc., 2013b). Tämä tekeekin hyökkääjän työstä varsin helppoa, sillä hän voi valita vain tuoreimman haavoittuvuuden ja samalla hyödyntää sitä valtaosaan käyttäjistä, sillä käyttäjät päivittävät niin hitaasti. Osasyynä hitaaseen päivitystahtiin voi olla se, että päivitykset tarvitsevat käyttäjältä järjestelmänhaltijan oikeuksia ja päivitysprosessi noudattaa kankeaa vanhaa kaavaa, joka ei ole päivittynyt useampaan vuoteen. Javan automaattinen päivittäjä tarjoaa myös päivityksen yhteydessä asennettavaksi Ask.com sivuston työkalupalkkia selaimeen ja haluaa asettaa sivuston selaimen kotisivuksi. Jos halutaan, että käyttäjät siirtyisivät nopeammin uusiin versioihin, pitäisi päivitysprosessin olla sulavampi ja mahdollisesti enemmän automatisoitu.

Java ei ole ainut, johon on kohdistunut vuoden 2013 aikana vakavia haavoittuvuuksia, joita on hyödynnetty. Ruby on Rails ohjelmistokehykseen (eng.

Framework) kohdistui keväällä vakava haavoittuvuus, jonka avulla hyökkääjät saivat helposti liitettyä saastuneita palvelimia bottiverkkoon (Goodin, 2013). TIOBE-indeksi seuraa vuosittain ohjelmointikielien suosiota. Vuoden 2013 tilastossa Java on pienellä erolla kärjessä ennen C ja C++ kieliä (TIOBE Software BV, 2013). Kielten suosion seuraaminen on hankalaa ja TIOBE-listaustakin tulisi käsitellä vain suuntaa antavana. On loogista ajatella, että mitä suositumpi kieli, sitä useampia hyökkäyskohteita on tarjolla hyökkääjälle. Vielä kun huomioon otetaan käyttäjien laiskuus päivittää uudempiin versioihin, niin ei jää epäselväksi, miksi Java on houkutteleva hyökkäyskohde.

## 6.1 Turvallisuuden parannuskeinoja

Selainvalmistajat ovat ottaneet askeleen suojatakseen käyttäjiään Java-hyökkäyksiltä estämällä Java-sovelmien suorittamisen ilman käyttäjän syötettä. Googlen Chrome selain kysyy käyttäjältä lupaa suorittaa Java-sovelmia ja antaa mahdollisuuden estää, sallia kerran tai sallia ja muistaa lupa suorittaa sivuston sovelmat (Oracle, 2013k). Tämä tuo uuden tietoturvakerroksen käyttäjän ja mahdollisen hyökkääjän välille, vähentäen riskiä joutua ”drive-by” hyökkäyksen kohteeksi.

Kuten aiemmissa luvuissa todettiin, on Javassa huomioitu tietoturvallisuus tärkeänä osana kieltä. Haavoittuvuudet ovat kohdistuneet pääasiassa selaimissa suoritettuun Javaan ja Java-sovelmiin. Kielessä on myös koitettu estää ohjelmoijaa tekemästä tietoturvaan vaikuttavia virheitä koodissaan, esimerkiksi roskienkeruulla ja laajalla apukirjastolla.

Jotta tietoturvariskeistä päästään eroon, tulee Oraclen saada käyttäjät päivittämään ohjelmistoaan nopeammin uusiin versioihin. Oracle on jo parantanut reagointikykyään uusiin haavoittuvuuksiin lisäämällä yhden vuosittaisen tietoturvajulkaisun vuosittaiseen aikatauluun (Reinhold, 2013).

## 6.2 Java 8

Java 8 piti alun perin julkaista vuoden 2013 puolella, mutta Oracle päätti jatkaa Javan tietoturvallisuuden parantamista kiihtyvällä tahdilla, joten Java 8 julkaisu myöhästyy vuoteen 2014 (Reinhold, 2013). Kehittäjät ovat siis pääasiassa keskittyneet korjaamaan tietoturvaongelmia ja Java 8 kehittämiseen on jäänyt vähemmän aikaa. Java-alustan pääarkkitehti Reinhold toteaakin blogissaan, että Java 8 oltaisiin voitu julkaista vuonna 2013, jos uusia ominaisuuksia olisi jätetty pois (Reinhold, 2013). Tämä olisi kuitenkin tarkoittanut ominaisuuksien, kuten lambdan eli anonyymien funktioiden poisjättämisen uudesta versiosta (Reinhold, 2013).

Koska uusi versio ei ole vielä valmis, ei ole vielä tarkkaa tietoa, mitä lopullisessa julkaisussa on lisätty ja paranneltu tietoturvan näkökulmasta. Oracle on

halunnut avata JDK:n kehitystä ja Java 8 kehitetään suurelta osalta avoimen OpenJKD projektin yhteydessä. OpenJDK:n sivuilla on listaus, johon on listattu ominaisuudet, jotka on tarkoitus lisätä uuteen versioon (Oracle, 2013l). Java 8 tulee sisältämään parannuksia jo olemassa oleviin ominaisuuksiin, kuten sertifikaattien mustan listan tarkistus ohjelmointirajapintaan (eng. Revocation list API) ja mahdollisuuden käyttää vahvempia algoritmeja salasanavarmennuksessa. Uusina ominaisuuksina on säädettävissä oleva turvallinen satunnaislukugeneraattori ja muita uusia tietoturvalaajennuksia.

# 7 YHTEENVETO

Java on kautta historiansa kokenut hyökkäyksiä etenkin sen omia web-sovelluksia eli sovelmia kohtaan. Näitä ongelmia korjatakseen Javaa on päivitetty useaan otteeseen pienillä korjauspäivityksillä. Myös suurempia korjauksia, kuten vuonna 1998 julkaistun version 1.2 tietoturvarakenteen uudistus, on tapahtunut Javassa. Vuoteen 2006 asti Java sai uuden pääversiopäivityksen kerran kahdessa vuodessa. Versioiden 1.6 ja 1.7 väliin muodostui melkein viiden vuoden väli, jonka aikana Java ei saanut uusia ominaisuuksia ja oli vain korjauspäivitysten kohteena. Vuonna 2010 Oracle osti Sun Microsystemsin ja sai omistukseensa myös Javan. Versio 7 julkaistiin vuonna 2011, ja sen tietoturvan toteutuksen rakenne on pysynyt siis kauttaaltaan saman kaltaisena kuin vuonna 1998 julkaistun version 1.2 rakenne. Tämä johtuu siitä, että Javan päivityksiä tehdessä on otettava huomioon, ettei vanhoja ominaisuuksia rikota ja että tuki vanhoilla versioilla tuotettuihin ohjelmiin säilyy. Tämä tekee rakenteen muuttamisesta hankalaa.

Java tarjoaa paljon tietoturvaa edistäviä ominaisuuksia, joiden avulla ohjelmoijat voivat toteuttaa helposti monimutkaisiakin tietoturva-asioita ohjelmistoissaan. Lisäksi Java hoitaa roskienkeruun automaattisesti Java-virtuaalikoneen avulla. Java-virtuaalikone on myös se osa Javassa, joka mahdollistaa ohjelmien suorittamisen erilaisissa käyttöjärjestelmissä ja erilaisilla laitteilla.

Javan suosio on houkutellut hyökkääjiä Javan kimppuun ja näistä hyökkäyksistä Java on joutunut kärsimään. Huomattavia hyökkäyksiä ovat olleet esimerkiksi Flashback-troijalainen, joka on laajimmalle levinnyt OS X haittaohjelma ja hybriditiedostohyökkäykset (Cheng, 2012). Vuonna 2013 Java on saanut useita tietoturvapäivityksiä, jotta haavoittuvuuksista päästäisiin eroon. Yhdysvaltojen Department of Homeland Security suositteli vuonna 2013 Javan poistamista useiden haavoittuvuuksien takia.

Nyt eletään muutoksen aikaa Javan tietoturvan suunnalla, kun Oracle on viivästyttänyt Java 8:n julkaisua (Reinhold, 2013). Tarkoitus on korjata tämänhetkisiä tietoturvaongelmia. Ongelmana ovat myös käyttäjät ja heidän hidas uusien Java-päivitysten käyttöönottotahti.

Kaiken tämän valossa herää kysymys, kuinka yrityksissä on reagoitu Javan ongelmiin. Websencen toteuttaman tutkimuksen pohjalta myös yritykset ovat olleet hitaita päivittämään ohjelmistojaan (Websence Inc., 2013a). Mitkä ovat syyt tähän, ja onko samankaltaisia ongelmia myös muiden ohjelmistojen kanssa? Lisäksi herää kysymys, kuinka haavoittuvuudet ovat vaikuttaneet ohjelmistoalan yrityksiin, jotka ovat projekteissaan käyttäneet Java-alustan tuotteita. Edellä mainitut asiat ovat mielenkiintoisia jatkotutkimusaiheita, koska niihin liittyvä tutkimus on vähäistä.

# LÄHTEET


Aquilino, B. I. (2012). Flashback OS X malware. Haettu 23.7.2013, osoitteesta http://www.f-secure.com/weblog/archives/Aquilino-VB2012.pdf

Bau, J., Wang, F., Bursztein, E., Mutchler, P., & Mitchell, J. Vulnerability Factors in New Web Applications: Audit Tools, Developer Selection & Languages. Haettu 27.7.2013, osoitteesta http://seclab.stanford.edu

CERT-FI. (2013). CERT-FI 2013 Haavoittuvuudet. Haettu 27.6.2013, osoitteesta https://www.cert.fi/haavoittuvuudet/2013.html

Cheng, J. (2012). Flashback trojan reportedly controls half a million Macs and counting. Haettu 1.7.2013, osoitteesta http://arstechnica.com/apple/2012/04/flashback-trojan-reportedly-controls-half-a-million-macs-and-counting/

Dean, D., Felten, E. W., & Wallach, D. S. (1996). Java security: from HotJava to Netscape and beyond. *Proceedings 1996 IEEE Symposium on Security and Privacy* (s. 190–200). IEEE Comput. Soc. Press. doi:10.1109/SECPRI.1996.502681

Dhanjani, N., Rios, B., & Hardin, B. (2009). *Hacking: the next generation* (1. p.). Sebastopol: O'Reilly Media.

Ghosh, A., & McGraw, G. (2012). Lost Decade or Golden Era: Computer Security since 9/11. *IEEE Security & Privacy Magazine*, *10*(1), 6–10. doi:10.1109/MSP.2012.12

Gong, L. (2009). Java Security: A Ten Year Retrospective. *2009 Annual Computer Security Applications Conference* (s. 395–405). IEEE. doi:10.1109/ACSAC.2009.44

Gong, L. (2011). Java Security Architecture Revisited. *Queue*, *9*(9), 30. doi:10.1145/2030256.2034639

Gong, L., Oracle, & Sun Microsystems. (2013). JavaTM SE Platform Security Architecture. Haettu 25.6.2013, osoitteesta http://docs.oracle.com/javase/7/docs/technotes/guides/security/spec/security-specTOC.fm.html

Goodin, D. (2012). Apple removes Java from all OS X Web browsers. Haettu 1.7.2013, osoitteesta http://arstechnica.com/apple/2012/10/apple-removes-java-from-all-os-x-web-browsers/

Goodin, D. (2013). Critical Ruby on Rails bug exploited in wild, hacked servers join botnet | Ars Technica. Haettu 9.8.2013, osoitteesta http://arstechnica.com/security/2013/05/critical-ruby-on-rails-bug-exploited-in-wild-hacked-servers-join-botnet/

Lindholm, T., Yellin, F., Bracha, G., & Buckley, A. (2013). *The Java ® Virtual Machine Specification Java SE 7 Edition*.

National Institute of Standards and Technology. (2011). Vulnerability Summary for CVE-2008-5343. Haettu 20.7.2013, osoitteesta https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2008-5343

Oh, J. W. (2012). Recent Java exploitation trends and malware. Haettu 1.7.2013, osoitteesta https://media.blackhat.com/bh-us-12/Briefings/Oh/BH_US_12_Oh_Recent_Java_Exploitation_Trends_and_Malware_WP.pdf

Oracle. (2000). Java 1.3 Security enhancements and changes. Haettu 29.6.2013, osoitteesta http://docs.oracle.com/javase/1.3/docs/guide/security/enhancements.html

Oracle. (2003). JavaTM 2 SDK, Standard Edition, version 1.4.2 Summary of New Features and Enhancements. Haettu 29.6.2013, osoitteesta http://docs.oracle.com/javase/1.4.2/docs/relnotes/features.html#security

Oracle. (2004). New Features and Enhancements J2SE 5.0. Haettu 29.6.2013, osoitteesta http://docs.oracle.com/javase/1.5.0/docs/relnotes/features.html

Oracle. (2011). Java™ 6 Security Enhancements. Haettu 29.6.2013, osoitteesta http://docs.oracle.com/javase/6/docs/technotes/guides/security/enhancements.html

Oracle. (2012). Differences between Java EE and Java SE - Your First Cup: An Introduction to the Java EE Platform. Haettu 9.8.2013, osoitteesta http://docs.oracle.com/javaee/6/firstcup/doc/gkhoy.html

Oracle. (2013a). Learn About Java Technology. Haettu 3.6.2013, osoitteesta https://www.java.com/en/about/

Oracle. (2013b). Secure Coding Guidelines for the Java Programming Language, Version 4.0. Haettu 11.8.2013, osoitteesta http://www.oracle.com/technetwork/java/seccodeguide-139067.html

Oracle. (2013c). ClassLoader (Java Platform SE 7). Haettu 20.7.2013, osoitteesta http://docs.oracle.com/javase/7/docs/api/java/lang/ClassLoader.html

Oracle. (2013d). Oracle's Security Vulnerability Remediation Practices. Haettu 9.8.2013, osoitteesta http://www.oracle.com/us/support/assurance/vulnerability-remediation/introduction/index.html

Oracle. (2013e). Java™ SE 7 Update 2 Release Notes.

Oracle. (2013f). Java™ SE Development Kit 7 Update 10 Release Notes. Haettu 7.8.2013, osoitteesta http://www.oracle.com/technetwork/java/javase/7u10-relnotes-1880995.html

Oracle. (2013g). Java™ SE Development Kit 7 Update 21 Release Notes. Haettu 7.8.2013, osoitteesta http://www.oracle.com/technetwork/java/javase/7u21-relnotes-1932873.html

Oracle. (2013h). Java SE Security. Haettu 3.6.2013, osoitteesta http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136007.html

Oracle. (2013i). Overview of Java SE Security. Haettu 3.6.2013, osoitteesta http://docs.oracle.com/javase/7/docs/technotes/guides/security/overview/jsoverview.html

Oracle. (2013j). Securitymanager (Java Platform SE 7). Haettu 20.6.2013, osoitteesta http://docs.oracle.com/javase/7/docs/api/java/lang/SecurityManager.html

Oracle. (2013k). Java and Google Chrome Browser. Haettu 11.8.2013, osoitteesta https://www.java.com/en/download/faq/chrome.xml

Oracle. (2013l). JDK 8 Features. Haettu 8.8.2013, osoitteesta http://openjdk.java.net/projects/jdk8/features#core/sec

Oracle. (2013m). Java Platform Standard Edition 7 Documentation. Haettu 25.6-.2013, osoitteesta http://docs.oracle.com/javase/7/docs/

OWASP Foundation. (2013a). OWASP Top 10 2013. Haettu 1.8.2013, osoitteesta https://www.owasp.org/index.php/Top_10_2013-Top_10

OWASP Foundation. (2013b). OWASP Top 10 - 2013 The Ten Most Critical Web Application Security Risks. Haettu 1.8.2013, osoitteesta http://owasptop10.googlecode.com/files/OWASP%20Top%2010%20-%202013.pdf

Ramani, N. (2013). Maintaining the security-worthiness of Java is Oracle's priority. Haettu 5.8.2013, osoitteesta https://blogs.oracle.com/security/entry/maintaining_the_security_worthiness_of

Reinhold, M. (2013). Java 8: Secure the train. Haettu 8.8.2013, osoitteesta http://mreinhold.org/blog/secure-the-train

Rodrigues, M., & Leader, O. (2007). OWASP TOP 10+ Java EE. *owasp.org*.

Sanastokeskus TSK. (2004). *Tiivis tietoturvasanasto*. Helsinki: Sanastokeskus TSK ry.

Stuttard, D., & Pinto, M. (2011). *The web application hacker's handbook: discovering and exploiting security flaws* (2. p.). Indianapolis: Wiley Publishing.

Sun Microsystems. (1997a). The Java Language Environment. Haettu 20.6.2013, osoitteesta http://www.oracle.com/technetwork/java/intro-141325.html

Sun Microsystems. (1997b). Sun Ships JDK 1.1. Haettu 29.6.2013, osoitteesta http://web.archive.org/web/20080210044125/http://www.sun.com/smi/Press/sunflash/1997-02/sunflash.970219.0001.xml

Tietotekniikan liitto ry. (2008). MOT Tietotekniikan liitto ATK-sanakirja. Haettu 24.7.2013, osoitteesta http://mot.kielikone.fi.ezproxy.jyu.fi/mot/jyu/netmot.exe

TIOBE Software BV. (2013). TIOBE Software: Tiobe Index. Haettu 9.8.2013, osoitteesta http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

US-CERT. (2013). 2013 Alerts US-CERT. Haettu 28.6.2013, osoitteesta https://www.us-cert.gov/ncas/alerts/

Valtionvarainministeriö. (2008). *Valtionhallinnon tietoturvasanasto*. Helsinki: Edita Prima Oy.

Websence Inc. (2013a). Majority of Users Still Vulnerable to Java Exploits. Haettu 28.7.2013, osoitteesta http://community.websense.com/blogs/security-labs/archive/2013/06/04/majority-of-users-still-vulnerable-to-java-exploits.aspx

Websence Inc. (2013b). How are Java attacks getting through? Haettu 5.8.2013, osoitteesta http://community.websense.com/blogs/securitylabs/archive/2013/03/25/how-are-java-attacks-getting-through.aspx

Williams, M. J., & Chitta, K. Java Garbage Collection Basics. Haettu 2.8.2013, osoitteesta http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/gc01/index.html