Opinnäytetyö (AMK)

Tieto- ja viestintätekniikka

2018

Asko Salonen

HAITTAOHJELMIEN TUTKIMINEN MANUAALISIN JA AUTOMAATTISIN MENETELMIN

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tieto- ja viestintätekniikka

2018 | 39 sivua

Asko Salonen

HAITTAOHJELMIEN TUTKIMINEN MANUAALISIN JA AUTOMAATTISIN MENETELMIN

Haittaohjelmat ovat jo pitkään olleet osa nykyistä tietoyhteiskuntaa. Viime aikoina haittaohjelmat ovat käyneet entistä tuhoisammiksi uusien uhria kiristävien haittaohjelmien myötä. Näin ollen tarve haittaohjelmilta ja kohdistetuilta hyökkäyksiltä suojautumiseen käy entistä akuutimmaksi. Kunnolliseen suojautumiseen haittaohjelmien varalta tarvitaan ymmärrystä siitä, miten haittaohjelma onnistuu saastuttamaan uhrinsa ja mitä haittaohjelma tekee saastuneella tietokoneella. Vaarallisimpia ovat haittaohjelmat, jotka salakirjoittavat tiedostot niin, ettei käyttäjä pysty enää avaamaan tiedostojaan. Vaarallisia ovat lisäksi haittaohjelmat, jotka siirtävät arkaluontoista materiaalia, esimerkiksi kuvia, ja tietoa, kuten salasanoja, haittaohjelman kirjoittajalle.

Opinnäytetyön tavoitteena oli tarkastella menetelmiä haittaohjelmien analysoimiseksi, löytää ilmaisia haittaohjelmien analysointiin käytettäviä työkaluja ja suorittaa pienimuotoista haittaohjelmien analysointia käyttäen hyväksi teoriaosuudessa esiin tuotuja työkaluja.

Haittaohjelmien manuaalista analysointia varten asennettiin kaksi virtuaalikonetta. Haittaohjelmien analysointi suoritettiin virtuaalikoneilla, koska virtuaalikoneet on helppo palauttaa tilaan ennen infektiota, hankkimiskustannuksia ei ole ja ylläpito on edullista. Toinen virtuaalikoneista saastutettiin haittaohjelmalla, jonka jälkeen haittaohjelman toimintaa tutkittiin teoriaosuudessa löydetyillä analysointityökaluilla. Automaattinen analysointi suoritettiin käyttämällä ilmaista analysointityökalua, Cuckoo Sandboxia. Cuckoo Sandboxille syötettiin haittaohjelma, jonka Cuckoo Sandbox käynnisti omassa virtuaalikoneessaan. Cuckoo Sandbox listasi haittaohjelman tekemät muutokset käyttöjärjestelmään ja antoi haittaohjelmalle arvosanan haitallisuuden perusteella.

Työn pohjalta voidaan todeta, että kattava haittaohjelmien analysointi onnistuu täysin ilmaiseksi, manuaalisesti ja automaattisesti. Ainoana esteenä voidaan pitää taitotiedon puutetta, sillä koodi- ja muistianalyysi vaativat kattavaa ymmärrystä ohjelmoinnista ja yleisesti tietotekniikasta.

ASIASANAT:

haittaohjelma, analysointi, Cuckoo, virtuaalikone, takaisinmallinnus, virus, Windows

BACHELOR´S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information and Communications Technology

2018 | 39 pages

Asko Salonen

MALWARE ANALYSIS USING MANUAL AND AUTOMATIC METHODS

Malware has been part of the current digital society for a long time. In recent times, malware has become increasingly destructive with the recent trend of blackmailing malware. With the recent developments, it has become imperative to develop proper protection against malware and targeted attacks. For proper protection, one must understand how malware has infected the machine and the effects of the infection. The most dangerous types of malware are those that encrypt user's personal files and those that spy on the user and send the information gathered to the malware author.

The objective of this thesis was to examine methods to analyze malware and find free software to do simple malware analysis.

Two virtual machines were installed for manual malware analysis. Virtual machines are easy to return to the state before the infection and upkeep is nonexistent since there is no physical machines. One of the virtual machines were infected with malware. During and after infection changes made by the malware were closely inspected. Automatic analysis was done using automatic analysis tool called Cuckoo Sandbox. Cuckoo Sandbox were given a file which it launched in a virtual machine. After analysis Cuckoo Sandbox listed all the modifications that the malware had done to the system.

In conclusion, it is possible to perform a detailed malware analysis free. The only obstacle comes from the lack of knowledge and experience. Code and memory analysis requires a deep understanding of programming and IT field in general.

KEYWORDS:

malware, analysis, Cuckoo, virtual machine, reverse engineering, virus, Windows

SISÄLTÖ

SANASTO

CMD	Microsoft Windows-käyttöjärjestelmässä käytetty komentorivi.
Dynaaminen analyysi	Ohjelman analysointi laukaisemalla.
Floppy disk, levyke	Vanha tiedonsiirron muoto.
Haittakuorma	Haittaohjelmissa tuhoa aiheuttava osuus.
Hash	Arvo joka voidaan laskea tiedostosta matemaattisten funktioiden avulla. Yleisesti käytetyt hash-laskentakaavat ovat MD5, SHA1, SHA256 sekä SHA512.
IOC	”Indication of compromise”, kuvaa tiettyä toimintaa tai olemassa olevaa asiaa, jota voidaan pitää merkkinä infektiosta.
Kryptaus	Salakirjoitus: tarkoituksena, että ulkopuoliset eivät pysty lukemaan kirjoitettuja tietoja.
Nollapäivähaavoittuvuus	Haavoittuvuus, johon ei ole tehty korjausta mutta haavoittuvuudelle löytyy hyväksikäyttömenetelmä.
Ping	Työkalu, jolla pyritään selvittämään laitteen tila lähettämällä tietoliikennepaketteja kohteeseen ja odottamalla vastausta.
Snapshot	Virtualisoinnissa termillä tarkoitetaan virtuaalikoneesta otettavaa kopiota, johon virtuaalikone on helppo palauttaa.
SSH	Internetprotokolla, joka luo turvallisen yhteyden kahden eri laitteen välille.
Staattinen analyysi	Tiedoston ulkopuolinen tarkastelu ilman sen käynnistystä.
Sudo	Ohjelma, joka mahdollistaa käyttäjän komentojen suorittamisen toisen käyttäjän oikeuksin, yleensä pääkäyttäjänä eli root-oikeuksin.
Virtuaalikone	Emuloitu käyttöjärjestelmä toisen käyttöjärjestelmän päällä.

1 JOHDANTO

Kehitys tietokoneiden ja internetin suhteen on muuttanut normaalia elämää ja mullistanut yritysten liiketoiminnan. Mullistuksen sivuefektinä ovat tulleet myös verkkorikolliset. Kasvavat uhat kriittisiä infrastruktuureja, datakeskuksia sekä julkisia ja yksityisiä yrityksiä vastaan luovat uniikin haasteen kaikille, yksityishenkilöstä isoihin yrityksiin. Nämä kyberuhat tapahtuvat käytännössä haittaohjelmien avulla, ja tarkoituksena on rahallinen hyöty, vakoilu, sabotaasi tai poliittinen vaikutusvalta. Hyökkääjien kehittyessä on tärkeää, että hyökkäykset huomataan ajoissa ja niihin on varauduttu. Haittaohjelmien analysoinnista on tullut erittäin tärkeä osa taistelussa kohdennettuja ja kehittyneitä uhkia vastaan. Haittaohjelmien analysointi vaatii taitotietoa useasta eri tietotekniikan aihealueesta.

Tässä opinnäytetyössä keskitytään haittaohjelmien toimintaan, eri tapoihin analysoida haittaohjelmia sekä suoritetaan staattista ja dynaamista analyysiä. Lisäksi käydään läpi ongelmia, joita voi ilmetä haittaohjelmien analysointia suorittaessa. Tässä opinnäytetyössä ei käydä läpi koodianalyysiä taikka muistianalyysiä, joka on kehittyneempää ja taitotasoltaan vaativampaa haittaohjelmien analysointia. Haittaohjelmien manuaalisesti suoritettu staattinen sekä dynaaminen analysointi tehtiin käyttäen virtuaalikoneita ja erityisesti haittaohjelmien analysointiin tarkoitettuja työkaluja. Haittaohjelmia tarkastellaan myös automaattisin keinoin käyttäen ilmaista Cuckoo Sandbox -nimistä työkalua. Cuckoo Sandboxista käydään läpi asennus, käyttö ja tulosten analysointi.

Haittaohjelmista on tehty melko paljon opinnäytetöitä ja tutkimuksia jo aiemmin. Opinnäytetöitä, joissa ollaan käsitelty haittaohjelmien analysointia, on tehty esimerkiksi Jyväskylän ammattikorkeakoulussa (Hakkarainen 2015) ja Metropolian ammattikorkeakoulussa (Aikio 2014). Myös pelkästä Cuckoo Sandboxista on tehty opinnäytetyö Jyväskylän ammattikorkeakoulussa (Kultanen 2016). Jyväskylän ja Metropolian ammattikorkeakouluissa tehdyissä töissä ei käsitellä Cuckoo Sandboxia lainkaan. Tässä työssä tutkitaan myös Cuckoo Sandboxia, joka on hyvä lähtökohta haittaohjelmien analysoinnille. Cuckoo Sandbox tarjoaa vertailukohdan manuaaliselle analyysille, sillä tekemällä manuaalisen analyysin on mahdollista oppia samat asiat kuin Cuckoo Sandboxin tuloksista käy ilmi ja mahdollisesti asioita, jotka ovat jääneet Cuckoo Sandboxilta huomioimatta. Metropolian ammattikorkeakoulussa tehdyssä opinnäytetyössä keskityttiin myös ainoastaan Bot-verkkojen toimintaan, eikä analysoimaan haittaohjelmien toimintaa yleisesti, kuten tässä työssä.

2 HAITTAOHJELMAT

Haittaohjelmat ovat ohjelmia, jotka tekevät haitallisia toimenpiteitä käyttöjärjestelmään ja mahdollisesti myös tietokoneen muihin komponentteihin. Haittaohjelmat voivat olla suoritettavia tiedostoja, skriptejä, koodia tai mitä tahansa muita sovelluksia. Hyökkääjät käyttävät haittaohjelmia esimerkiksi varastaakseen salassa pidettävää tietoa, vakoillakseen käyttäjää, lähettämään uhrin nimissä sähköpostia tai kaappaamaan saastuneen tietokoneen osaksi bottiverkkoa. Haittaohjelmat kulkeutuvat tietokoneeseen usein ilman, että käyttäjä tiedostaa, että kyseessä on haittaohjelma. Haittaohjelma kulkeutuu tietokoneelle esimerkiksi sähköpostin liitetiedoston mukana, verkosta ladatun tiedoston ohessa tai muistitikkujen avulla.

2.1 Haittaohjelmatyypit

Haittaohjelmia on useita eri tyyppejä, jotka on nimetty leviämistavan ja käyttäytymisen perusteella. Haittaohjelman kategorisointi ei aina ole täysin yksiselitteistä, koska yksi haittaohjelma voi sisältää useita eri ominaisuuksia, jotka voivat pudota eri kategorioiden alle.

Adware toimittaa käyttäjälle mainoksia, joiden katselukerroista haittaohjelman kehittäjä saa rahallista korvausta. Adwarea saa useimmiten muiden hyödyllisten ohjelmien mukana joko kiinnittämättä huomiota ohjelman asennusprosessiin tai pahimmillaan käyttäjältä ei edes kysytä halutaanko mukana asentaa muita ohjelmia. Adwaren mukana tulee välillä myös spywarea.

Spyware kerää salassa, käyttäjän tietämättä, tietoa tietokoneen toiminnasta. Yleisiä tietojen keräyskohteita ovat muun muassa käyttäjän nettisurffailu ja näppäinpainallukset. Nettisurffailun tietoja kerätään, jotta haittaohjelma pystyy tarjoamaan paremmin kohdennettuja mainoksia. Vaarallisemmaksi osoittautuu näppäinpainallusten seuranta, jossa haittaohjelman kirjoittaja voi saada haltuunsa esimerkiksi pankkitunnukset tai sähköpostitunnukset. Spywaret pystyvät ottamaan myös kuvakaappauksia tietokoneen ruudusta sekä aktivoimaan web-kameran. Vakoillut tiedot spyware lähettää haittaohjelman kirjoittajalle internetin välityksellä. Varastetuilla kuvilla haittaohjelman kirjoittaja voi kiristää uhriaan maksamaan lunnaita, jottei kuvia esimerkiksi julkaistaisi internetissä. Spyware leviää yleensä muiden ohjelmien kyljessä adwaren tavoin sekä hyödyntämällä ohjelmistohaavoittuvuuksia.

Botverkko lisää saastuneen tietokoneen osaksi käskyjä vastaanottavaan bottiarmeijaan, jotka voivat hyökkääjän käskystä lähettää suuria määriä yhteyspyyntöjä verkkolaitteille. Suuri määrä vajavaisia ja turhia yhteyspyyntöjä voi aiheuttaa esimerkiksi palvelimen ylikuormittumisen. Ylikuormittunut palvelin ei pysty vastaamaan oikeiden käyttäjien yhteyspyyntöihin, joka voi aiheuttaa palvelimen omistajalle huomattavia rahallisia menetyksiä. Sama tilanne voi käydä myös ilman bottiarmeijan uhriksi joutumista esimerkiksi uuden suositun palvelun tullessa suuren yleisön käyttöön tai kaupallisena juhlapäivänä, jolloin palvelun ylläpitäjä ei ole välttämättä varautunut riittävästi suureen verkkoliikenteen määrään.

Ransomware, kiristyshaittaohjelma kryptaa tietokoneella olevan datan ja pyytää lähes aina lunnaita tiedostojen vapauttamisesta, osa kiristyshaittaohjelmista ei tarjoa mahdollisuutta tiedostojen palautukseen ja uhrit saavat ainoastaan ilkkuvan viestin. Vuonna 2018 havaittiin kiristyshaittaohjelma, joka vaati käyttäjää pelaamaan tunnin verran suosittua PlayerUnknown's Battlegrounds -videopeliä, haittaohjelma tosin tarjosi tiedostojen vapautuskoodin myös kiristysviestissään (Chalk 2018). Lunnaat pyydetään yleensä vaikeasti jäljitettävinä kryptovaluuttoina. Lunnaiden maksaminen ei tarkoita, että tiedostot vapautetaan, mutta kryptauksen purkaminen ei välttämättä ole mahdollista muilla keinoilla. Lunnaiden maksun jälkeen haittaohjelman kirjoittaja voi vapauttaa tiedostot. Tarina vapautuneista tiedostoista antaa uskoa muille uhreille, että tiedostot on mahdollista palauttaa maksamalla rahaa. Osa kryptausohjelmista on kirjoitettu niin huonosti, etteivät edes haittaohjelman kirjoittajat tiedä salauksen purkuavaimia, eivätkä näin ollen pysty purkamaan tiedostojen salausta. Kiristyshaittaohjelma on haittaohjelmista käyttäjälle näkyvin sekä käyttäjää eniten häiritsevä, sillä ilman ajantasaisia varmuuskopioita kaikki tietokoneella olevat tiedot ovat mahdollisesti menetetty ja tietokoneen normaalikäyttö on mahdotonta, kunnes tiedostot on vapautettu tai käyttöjärjestelmä asennettu uudestaan.

Rootkit antaa haittaohjelman kirjoittajalle pääsyn käyttäjän tietokoneelle käyttäjän siitä tietämättä. Kuten Spywaressa, haittaohjelman kirjoittaja näkee kaiken tietokoneella tapahtuvan sekä pääsee kontrolloimaan tietokoneen käyttäytymistä. Rootkit pureutuu syvälle käyttöjärjestelmään ja siitä eroon pääseminen on hankalaa. Nopein ja kivuttomin tapa on todennäköisesti tietokoneen internetistä irti ottaminen, tärkeiden tiedostojen varmuuskopiointi ja käyttöjärjestelmän uudelleen asennus. (DuPaul 2012)

Trojan-nimitys viittaa tapaan, jolla haittaohjelma kulkeutuu tietokoneeseen eikä niinkään siihen, mitä vahinkoa se aiheuttaa. Trojan näyttää luotettavalta sovellukselta, joka on

todellisuudessa haittaohjelma tai sisältää myös haittaohjelman. Trojan tarjoaa myös sisäänpääsyn järjestelmään muille haittaohjelmille.

Virus lisääntyy ja siirtyy muihin tietokoneisiin hyödyntämällä tietoturva-aukkoja esimerkiksi käyttöjärjestelmässä, selaimessa tai toimistotyökaluissa. Virus kopioi itsensä kiinni muihin ohjelmiin ja käynnistyy tartunnan saaneen ohjelman mukana. Virus-nimitystä käytetään usein virheellisesti kansankielessä kuvaamaan yleisesti kaikkia haittaohjelmia.

Worm kopioi itseään ja siirtyy työasemasta toiseen verkon yli. Toisin kuin virusten kohdalla, madon replikointi tapahtuu automaattisesti eikä mato tarvitse isäntäohjelmaa, johon tarrata kiinni.

2.2 Haittaohjelmien historia

Haittaohjelmia on ollut olemassa lähes yhtä kauan kuin hyödyllisiäkin ohjelmia. Alun perin haittaohjelmat olivat vain pienemmän piirin hupia, jossa tarkoituksena oli lähinnä maineen ja muutaman illan huvin irti saaminen. Nykyään haittaohjelmat ovat paljon yleisempiä ja niiden olemassaolo on paljon enemmän raha-, valta- ja informaatioperusteista. Haittaohjelmat ovat luonnollisesti kehittyneet ajansaatossa vaikeammaksi havaita ja tutkia.

Ensimmäinen julkisesti tiedossa oleva mato-haittaohjelma nähtiin 1970-luvulla Yhdysvaltojen armeijan käytössä olleessa ARPANET-nimisessä verkossa. Creeperiksi nimetty haittaohjelma pystyi liikkumaan itsenäisesti verkossa ja saastuttamaan työasemia verkon yli. Creeper ei sisältänyt minkäänlaista haittakuormaa, saastunut tietokone näytti ainoastaan viestin ”I'M THE CREEPER: CATCH ME IF YOU CAN.”. Creeperiä vastaan kehitettiin Reaper-niminen ohjelma, joka Creeperin tavoin liikkui itsenäisesti verkossa etsien Creeperin saastuttamia työasemia, tarkoituksena puhdistaa työasemat. Reaperia voidaan pitää ensimmäisenä haittaohjelmien poistotyökaluna. (Dominguez 2018)

Vuonna 1974 havaittiin toinen nopeasti itseään replikoiva virus, joka myöhemmin nimettiin Rabbitiksi ja Wabbitiksi, nopeasta replikoinnistaan johtuen. Toisinkuin aikaisemmat virukset, Rabbit heikensi työaseman toimintaa ja lopulta aiheutti työaseman kaatumisen. (Snyder 2010)

Vuonna 1981 Apple II tietokoneet olivat jo melko yleisiä koti- ja toimistoympäristöissä. Apple II tietokoneilla havaittiin haittaohjelma Elk Cloner, joka replikoitui toisiin työasemiin

levykkeiden avulla. Aina kun tietokone käynnistettiin saastuneelta levykkeeltä, virus aktivoitui. Virus jäi tarkkailemaan Floppy-asemaa uuden puhtaan levykkeen varalta saastuttaen senkin. Elk Clonerin haittakuorma käänsi kuvia ja väläytti vitsejä. (Rouse 2005)

> “ELK CLONER:
>
> THE PROGRAM WITH A PERSONALITY
>
> IT WILL GET ON ALL YOUR DISKS
>
> IT WILL INFILTRATE YOUR CHIPS
>
> YES, IT'S CLONER
>
> IT WILL STICK TO YOU LIKE GLUE
>
> IT WILL MODIFY RAM, TOO
>
> SEND IN THE CLONER!” (Rouse 2005)

Elk Cloner oli vasta alkua Apple II tietokoneille kirjoitetuista viruksista. Muutama vuosi myöhemmin ilmaantuivat Joe Dellingerin ja hänen opiskelijatoverien kirjoittamat Virus 1, Virus 2 ja Virus 3 -virukset. (Apple II History 2018)

Jon Hepps ja John Shock aloittivat projektin, jonka tarkoitus oli luoda ja tutkia matojen leviämistä, mutta ohjelmointivirheen johdosta madot alkoivat lisääntyä hallitsemattomasti ja projekti jouduttiin lopettamaan. Vuonna 1985 ilmaantui EGABTR-niminen troijalainen, joka lupasi parempaa graafista suorituskykyä tietokoneeseen, mutta todellisuudessa tyhjensi tietokoneen kovalevyn sekä näytti viestin ”Arf, arf, Gotcha!” näytöllä. (G Data 2018)

Vuonna 1986 kaksi pakistanilaista tietokonekaupan omistajaa, Basit Farooq Alvi ja Amjad Farooq Alvi, kehittivät oman viruksensa IBM PC -tietokoneille. Haittaohjelma oli lähes harmiton ja ainoastaan näytti tietokonekaupan omistajien nimet sekä kaupan osoitteen. Lopputulos tosin oli ensimmäinen ”hiljainen” Brain-virus, joka tarttui tuhansiin MS-DOS-tietokoneisiin. Käyttäjän tarkastellessa tartunnan saanutta asemaa, näytti virus ainoastaan saastuttamattomat tiedot. (Harán 2018)

Vienna.636.A-virusta voidaan pitää yhtenä haittaohjelmateollisuuden virstanpylväänä. Haittaohjelman kirjoittaja on tuntematon, mutta tärkeämpänä voidaan pitää Bernd Fixiä, joka onnistui neutralisoimaan viruksen. Berndin koodi päätyi osaksi Ralf Burgerin kirjoit-

tamaa kirjaa ”Computer Viruses: A High-tech Disease”. Kirja sisälsi viruksen neutralisointiin käytetyn koodin ja oli suunnattu tiedottamaan käyttäjiä haittaohjelmien vaaroista, vaikkakin se auttoi uusien haittaohjelmien kirjoittamisessa. (Malware Wikia 2018)

Vuonna 1987 huomattiin lukuisia uusia haittaohjelmia, geneeristen haittaohjelmien lisäksi vuoteen mahtui myös muutama maininnan arvoinen haittaohjelma. Lehigh-virus, joka huomattiin Pennsylvanian yliopiston verkossa, tuli kuuluisaksi lähinnä sen tuhoisuudesta. Virus infektoi neljä eri tiedostoa levyltä, jonka jälkeen se alkoi tuhota tiedostoja levyltä satunnaisessa järjestyksessä tuhoten myös itsensä prosessissa. Yliopiston työntekijät onnistuivat taltuttamaan viruksen, eikä se missään vaiheessa onnistunut pakenemaan muihin verkkoihin. (Kaspersky 2018)

Win32.Worm.Suriv.A-virus vaikutti syntyvän vahingossa, israelilaisen ohjelmoijan halutessa muokata sitä, miten .exe-tiedostot asentuvat käyttöjärjestelmään. Myöhemmin samana vuonna julki tullut viruksen uusi versio ”Jerusalem” poisti kaikki levyltä löytyvät .exe-tiedostot perjantai 13. päivä, lukuun ottamatta vuotta 1987. Haittaohjelma oli erittäin yleinen aikanaan, mutta modernit Windows-versiot eivät ole haavoittuvaisia. (Kaspersky 2018)

Loppuvuodesta julki tullut Cascade.1701 oli ensimmäinen virus, joka salakirjoitti haittakuormansa. Haittakuorma viruksessa oli lähes harmiton, käytännössä haittaohjelma siirsi näytöllä olleita kirjaimia alaspäin, kunnes kirjain osui näytön alareunaan tai toiseen kirjaimeen, tapahtuma muistutti Tetris-videopelissä tapahtuvaa toimintaa. Haittakuorma aktivoitui ensimmäisellä kerralla minuutin jälkeen infektiosta, myöhemmillä kerroilla haittakuorma aktivoitui 30 sekunnin välein. Joulun alla Saksassa ilmestyi vielä Christmas Tree Worm -mato, jonka haittakuorma muodosti näytölle joulukuusen kirjaimista. 13. joulukuuta mennessä haittaohjelma oli onnistunut tukkimaan EARNetin verkon. (Kaspersky 2018)

Haittaohjelmien laajamittainen leviäminen käynnisti uuden teollisuuden alan syntymisen. Tietoturvaan erikoistuneet yritykset alkoivat julkaista virustentorjuntaohjelmia sekä muita työkaluja taistelemaan haittaohjelmia vastaan. Luonnollisesti ensimmäiset skannerit ja virustentorjuntaohjelmat olivat yksinkertaisia ja sisälsivät usein immunisoivan komponentin. Immunisoiva komponentti lisäsi tiedostoihin koodia, joka huijasi viruksia luulemaan, että tiedosto on jo saastunut. Immunisointi toimi hyvin tiedettyjä viruksia vastaan, mutta ei tarjonnut aktiivista suojaa uusia tuntemattomia haittaohjelmia vastaan.

Torjuntaohjelmat eivät yleistyneet nopeasti, halvasta hinnastaan huolimatta, lisäksi internetin alkeellisuuden johdosta torjuntaohjelmien päivittäminen oli haasteellista. Ensimmäinen virustentorjuntafoorumi syntyi 1988. Samana vuonna, kolikon kääntöpuolella, julkaistiin haittaohjelmien rakennustyökalu. Työkalu tarjosi yksinkertaisen käyttöliittymän viruksen rakentamiseen.

Ensimmäinen Macintoshille kirjoitettu haittaohjelma Worm.Macos.Macmag.A tuli julki vuoden 1988 alkupuolella. Haittaohjelma lisättiin keskustelupalstalle ladattavaksi peitenimellä. Käyttäjän ladattua ja käynnistettyä haittaohjelman, asensi se taustaohjelman, joka näytti käynnistyksen yhteydessä viestin. (DaBoss 2013)

> “RICHARD BRANDOW, publisher of MacMag, and its entire staff would like to take this opportunity to convey their UNIVERSAL MESSAGE OF PEACE to all Macintosh users around the world.” (DaBoss 2013)

Kuten Reaper etsi Creeper-haittaohjelmaa, haittaohjelma Denzuko.A etsi ja korvasi Brain-haittaohjelman instansseja itsellään. Ohjelmointivirheiden takia haittaohjelma ei onnistunut saastuttamaan tiettyjä massamuistityyppejä vaan ainoastaan tuhoamaan kaiken tiedon muistista. Ensimmäisiä saastumisen merkkejä olivat näppäinyhdistelmän Ctrl+Alt+Del toimimattomuus uudelleenkäynnistykseen sekä viestin ”DEN ZUK” näkyminen ruudulla hetken ajan. (Virus Wikia 2018)

Haittaohjelmien kasvavaa määrää vastaan perustettiin ”Computer Emergency Response Team / Coordination Center (CERT / CC)”, organisaatio, joka on edelleen aktiivinen tänä päivänä. Ensimmäinen virustentorjuntaohjelma, Dr. Solomon's Anti-Virus Toolkit, julkaistiin brittiläisen ohjelmoijan kirjoittamana vuonna 1988 (Wikipedia 2018).

3 HAITTAOHJELMIEN ANALYSOINTI

Haittaohjelmien analysoinnin motiivi on saada tietoa haittaohjelmasta. Saatua tietoa käytetään haittaohjelmalta suojautumiseen. Haittaohjelmia analysoidessa pyritään saamaan selville, miten haittaohjelma on päässyt sisään, haittaohjelman kyvykkyys ja kuinka se voidaan havaita, taltuttaa sekä estää tulevaisuudessa. Haittaohjelman havaitsemista varten kerätään listaa tuntomerkeistä. Tästä listasta käytetään yleisesti lyhennettä IOC, joka tulee sanoista Indication of compromise. Tuntomerkkejä ovat esimerkiksi IP-osoitteet ja verkkotunnukset, joihin haittaohjelma ottaa yhteyttä, sekä tiedostonimet ja rekisterimuutokset, joita haittaohjelma tekee.

3.1 Takaisinmallinnusmenetelmä

Takaisinmallinnuksessa asia puretaan osiin ja tutkitaan yksittäisiä osaa, jotta ymmärretään, miten kokonaisuus toimii. Toimintatapa ei rajoitu pelkästään tietotekniikkaan, vaan asioita on purettu osiin läpi ihmishistorian. Tietotekniikasta puhuttaessa takaisinmallinnukseen käytetään varta vasten tehtyjä sovelluksia, jotka kääntävät tietokonekielen ohjelmointikieleksi, kuten Java ja C#. Takaisinmallinnuksessa ei onnistuta palauttamaan alkuperäistä lähdekoodia, vaan niin sanottua pseudokoodia. Alkuperäistä lähdekoodia ei pysty täysin palauttamaan kääntäjien tekemän optimoinnin takia. Koodissa olleita kommentteja, alkuperäisiä funktion ja muuttujien nimiä ei pysty palauttamaan. Haittaohjelmien takaisinmallinnus tapahtuu eristetyssä ympäristössä, usein virtuaalikoneen sisällä.

3.2 Haittaohjelmien analysointimenetelmät

Haittaohjelmien tutkimiseen on eri menetelmiä, jotka saavuttavat hieman eri asioita mutta jokainen menetelmä on yhtä lailla tärkeä. Mitä syvempää analyysiä halutaan tehdä, sitä todennäköisemmin jokaista menetelmää tullaan käyttämään.

Staattisessa analyysissä ohjelmaa tutkitaan ilman sen käynnistystä. Staattinen analyysi on analyyseistä helpoin, lopputuloksena on ohjelman metadataa. Staattinen analyysi ei välttämättä kerro kaikkea, mutta voi antaa suuntaa tuleville analyyseille. Staattisessa analyysissä saadaan esimerkiksi haittaohjelman hash-arvo, jolla voidaan hakea

haittaohjelmasta tietoa, käyttämällä internetissä olevia palveluita. Uusista ja kohdennetuista haittaohjelmista ei välttämättä löydy mitään tietoa kyseisistä palveluista ja haittaohjelmasta tiedon saaminen jää analysoijan varaan.

Dynaamisessa analyysissä ohjelma käynnistetään eristetyssä ympäristössä ja sen toimintaa tarkkaillaan. Analyysi on helppo tehdä, ja se antaa arvokasta tietoa, miten haittaohjelma käyttäytyy laukaisutilanteessa ja siitä seuraavassa käyttöjärjestelmän saastuneessa tilassa. Dynaaminenkaan analyysi ei välttämättä kerro kaikkia haittaohjelman kykyjä, sillä jotain voi jäädä huomaamatta tai haittaohjelma ei käyttäydy kuten sen pitäisi. Dynaamisessa analyysissä mahdollisesti vastaantulevia ongelmia käydään läpi tarkemmin seuraavassa kappaleessa.

Lähdekoodianalyysi on kehittynyt analyysin muoto, jossa koodia analysoidaan. Lähdekoodin analysointi paljastaa tietoa, jota ei välttämättä näe staattisen tai dynaamisen analyysin avulla. Lähdekoodianalysointi jaetaan edelleen staattiseen lähdekoodin analysointiin ja dynaamisen lähdekoodin analysointiin. Staattinen lähdekoodin analysointi tarkoittaa, että haittaohjelma puretaan osiin ja lähdekoodia tarkastellaan palasina. Dynaamisessa lähdekoodin analysoinnissa haittaohjelmaa käynnistetään rivi kerrallaan tai tiettyyn pisteeseen asti. Lähdekoodin analysointi vaatii laajaa ymmärrystä ohjelmointikielistä ja käyttöjärjestelmistä.

Muistianalyysissä tutkitaan tietokoneen keskusmuistia. Muistianalyysi tapahtuu ottamalla saastuneen tietokoneen muistista tapahtumahetken kuva ja analysoimalla sitä. Keskusmuistia analysoimalla saadaan tietoa käyttöjärjestelmässä tapahtuneista komennoista, prosesseista ja aktiivisista verkkoyhteyksistä. Usein kriittinen data haittaohjelmista ja kohdennetusta hyökkäyksestä sijaitsee ainoastaan keskusmuistissa. Kaikki käyttöjärjestelmässä ajettavat ohjelmat, olivat ne sitten haitallisia tai ei, ladataan keskusmuistiin suorittamista varten. Tästä syystä muistianalyysi on kriittistä tutkittaessa vaikeasti havaittavia haittaohjelmia ja kohdennettuja hyökkäyksiä.

3.3 Dynaamisessa analyysissä mahdollisesti ilmaantuvia ongelmia

Haittaohjelma ei välttämättä toimi oikein dynaamista analyysiä tehdessä. Tässä kappaleessa käydään läpi siihen johtavia syitä. Yksinkertaisin syy voi löytyä ohjelmointivirheestä, joka aiheuttaa haittaohjelman väärin toiminnan. Valtaosa uusista haittaohjel-

mista yrittää selvittää, onko haittaohjelma virtuaalikoneessa, jonka tarkoitus on analysoida haittaohjelmaa. Virtuaaliympäristöt ovat yleistyneet huimasti viime vuosina, eikä virtuaalikone tarkoita automaattisesti, että haittaohjelmaa yritetään analysoida.

Haittaohjelma pyrkii selvittämään ympäristöään erinäisin tavoin. Alitehoinen tietokone on merkki, että haittaohjelma on käynnistynyt virtuaaliympäristössä. Virtuaalikoneilla on yleensä käytössään ainoastaan yksi tai kaksi suorittimen ydintä sekä muutama gigatavu keskusmuistia, joka on nykyään erittäin vähän resursseja. Keskihintaisista puhelimistakin löytyy useita gigatavuja keskusmuistia ja moniytimisiä prosessoreita.

Haittaohjelma, joka on nimetty selkeillä analysointiin viittaavilla nimillä kuten ”sample” tai ”malware”, ei välttämättä toimi samoin kuin muissa ympäristöissä. Myös käyttäjänimet kuten ”User” tai ”Test” ja tietokoneen nimet kuten ”analysismachine” ja ”virtualmachine” antavat haittaohjelmalle indikaatiota, että kyseessä on analysointiin käytettävä virtuaalikone. VMwarelta ja VirtualBoxilta löytyvät omat ohjelmansa, jotka asentuvat virtuaalikoneelle ja niiden olemassaoloa voidaan pitää myös yhtenä indikaattorina, että kyseessä on virtuaalikone.

Muita ongelmia haittaohjelman oikein toimimisen kannalta saattaa tulla käyttöjärjestelmän tai suositun sovelluksen, kuten Microsoft Officen, väärästä versiosta. Kieliasetuksen olemassaolo tai puuttuminen saattaa myös vaikuttaa siihen, toimiiko haittaohjelma oikein. Esimerkiksi **Citadel**-niminen troijalainen ei tee tuhoja tietokoneissa, joissa on asennettuna kyrilliset aakkoset. Verkkoyhteyden toimimattomuus voi estää haittaohjelman oikein toimimisen. Miksi aktivoitua, jos kaapattuja tietoja ei pysty kuitenkaan siirtämään haittaohjelman julkaisijalle tai liittyä mukaan Bot-verkkoon? Vaikka verkkoyhteys toimisi, kunnollinen laukaisu voi jäädä tekemättä, koska yhteys tulee väärästä osoitteesta, esimerkiksi haittaohjelma, joka on kohdennettu yritykseen tai valtion virastoon. Väärä päivämäärä tai kellonaika ovat myös mahdollisia esteitä, haittaohjelma voi odottaa esimerkiksi joulupäivään asti käynnistymistä. Haittaohjelma voi myös odottaa esimerkiksi viikon ennen käynnistymistä vähentääkseen kiinnijäämisen riskiä.

4 MANUAALINEN ANALYYSI

Haittaohjelman tarkastelu vaatii turvallisen laboratorioympäristön, jossa on minimaalinen mahdollisuus haittaohjelman leviämiseen oikeisiin laitteisiin. Haittaohjelmalaboratorio voi olla pieni ja yksinkertainen tai massiivinen automatisoitu järjestelmä. Varmasti turvallisen haittaohjelmalaboratorion tekeminen ei ole mahdollista, etenkään suuremmissa mittakaavoissa. Haittaohjelmalaboratoriota tehdessä tavoitteena on riskien minimointi, tiedostaminen ja hyväksyminen.

Täysin aukottaman analysointiympäristön kehittämisen ongelma tulee haavoittuvuuksista, joiden olemassaolosta ei välttämättä tiedä kukaan haavan löytäjää lukuun ottamatta. Esimerkiksi 7.11.2018 julkaistiin nollapäivähaavoittuvuus, joka koskee kaikkia VirtualBoxin versioita 5.2.20 versiosta alaspäin (Zorz 2018). Haavoittuvuus hyväksikäyttää VirtualBoxin käyttämää oletusverkkokorttia, jolloin yhteyden ollessa NAT-tilassa hyökkääjä pääsee käsiksi Host-koneeseen. Haavoittuvuuteen ei ole paikkausta tällä hetkellä, mutta haavoittuvuudelta pystyy suojautumaan vaihtamalla virtuaalikoneen verkkokorttia tai käyttämällä toista verkkoasetusta.

4.1 Työvälineet

Seuraavaksi käydään läpi suosittuja haittaohjelman tutkimisen työkaluja. Etenkin alussa parhaimman lopputuloksen saa yleensä kokeilemalla eri ohjelmia. Analysointityökalujen käyttöliittymistä löytyy eroja, lisäksi tulokset voivat antaa erilaisia johtolankoja. Tässä kappaleessa mainitut ohjelmat ovat tehokkaita työkaluja Windows-ympäristöön, tästä johtuen virustentorjunta voi estää ohjelmien olemassaolon tai käynnistyksen.

4.1.1 Windows työasemien analysointiin käytettävät työkalut

Windows Sysinternals sisältää suuren määrän diagnostiikkatyökaluja levyn, verkkoyhteyksien, Windowsin prosessien, tietoturvan ja laitteistotietojen hallintaan, sekä opastetiedostoja. Työkaluja pystyy lataamaan erikseen sekä koko pakettina ilmaiseksi Microsoftin sivuilta. (Russinovich 2018)

Malware Analyst Pack sisältää suuren määrän diagnostiikkatyökaluja erinäisiin tarkoituksiin. Työkalut ovat kolmannen osapuolen kirjoittamia, eikä Microsoft ole kytköksissä niihin millään tavalla. (Zimmer 2018)

010 Editor on Heksaeditori Windowsille, Linuxeille ja Mac OS:lle, joka antaa yksinkertaisen näkymän tiedostojen muokkaukseen tavutasolla. 010 Editorista löytyy 30 päivän kokeilujaksoversio, mutta 30 päivää ylittävältä ajalta täytyy ohjelma ostaa. (Sweetscape 2018)

IDA kääntää konekielen ihmiselle luettavaan assembly-kieleen. IDA on erittäin suosittu ammattilaisten käytössä suuren ominaisuusvalikoiman ja muokkautuvuuden ansiosta. IDAsta löytyy versio Windowsille, Linuxeille sekä Mac OS:lle. IDAsta löytyy kahta eri maksullista versiota, Starter ja Professional, sekä kokeilujakso Starter-versioon. Mainittavin ero versioiden välillä on Starter versiosta puuttuva tuki 64-bittisille ohjelmille. (Hex-Rays 2017)

Cygwin tuo suuren kirjaston GNU:n ja muita avoimen lähdekoodin työkaluja Windowsille, käytännössä mahdollistaen Linuxin kaltaisen toiminnallisuuden Windowsilla. (Cygwin 2018)

Notepad++ on yksinkertainen tekstinkäsittelyohjelma ohjelmoijia varten. Notepad++ sisältää pitkän liudan hyödyllisiä ominaisuuksia normaaliin Notepadiin verrattuna. Notepad++:sta löytyy myös kattava lisäosaluettelo, jolla ominaisuuksia saa lisättyä entisestään. (Ho 2016)

Capture-BAT listaa kaikki työasemassa tapahtuvat muutokset, kuten levylle kirjoittamisen sekä rekisteri muutokset. Capture-BAT on ilmaiseksi ladattava ja käytettävä sovellus Windows-käyttöjärjestelmällä. (The Honeynet Project 2018)

INetSim tarjoaa mahdollisuuden simuloida eri verkkopalveluita, kuten esimerkksi verkkosivua, IRC-palvelinta tai SSH:ta. Yhteyspyynnön tullessa INetSim vastaa ennalta määritetyllä tavalla ja näin huijaa esimerkiksi haittaohjelmaa, että sillä on pääsy komentokanavaansa.

RegShot on yksinkertainen ohjelma, joka mahdollistaa rekisterin muutosten tarkastelun. Käytännössä RegShotilla pystyy ottamaan kuvan rekisteristä ennen haittaohjelman laukaisua ja sen jälkeen ja lopuksi vertailla eri tiloja.

PEiD yrittää selvittää, mikä ohjelma loi tiedoston. Käytännössä PEiD etsii tiettyjä osia koodista mitkä viittaavat tiettyyn koontiohjelmaan. (Aldeid 2017)

LordPE yrittää korjata hajonneet PE-tiedot, esimerkiksi OllyDbg:n käytön jälkeen. (Aldeid 2017)

OllyDbg mahdollistaa ohjelmien osiin purkamisen ja analysoinnin todella alhaisella tasolla. (OllyDbg 2014)

4.1.2 Verkkosivustot

Virustotal.com on Googlen omistama verkkosivusto, jonne pystyy lähettämään tiedoston, hakemaan hash-arvolla, verkkosivun nimellä sekä IP-osoitteella. Virustotal skannaa lähetetyt tiedostot usealle eri virustentorjuntaohjelmalla ja näyttää tulokset yksinkertaisesti haitallista- tai ei haitallista -arvoina. Tiedostoja lähettäessä täytyy huomioida, että kaikki lähetetyt tiedostot ovat julkisessa jakelussa, eikä palveluun kannata lähettää mitään arkaluontoisia tietoja sisältäviä tiedostoja.

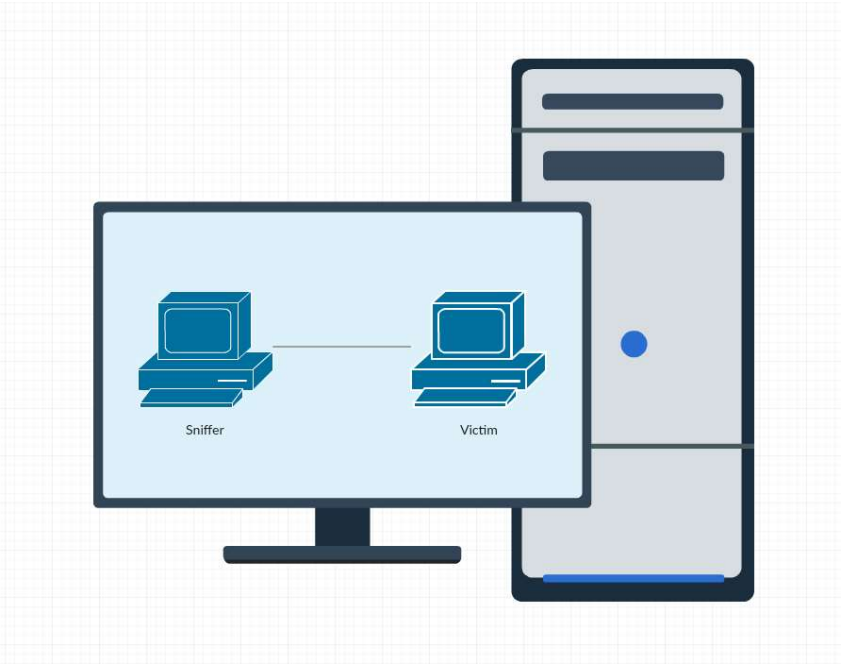
Hybrid-Analysis.com on palvelu, jonne voi lähettää tiedoston tai verkkosivun osoitteen tai hakea vanhoja tuloksia hash-arvoilla. Hybrid Analysis käynnistää syötetyn tiedoston tai vierailee verkkosivulla virtuaalikoneella sekä tulostaa yksityiskohtaisen raportin lopputuloksesta. Hybrid Analysis sisältää myös muutamia asetuksia, joilla analyysiä voidaan tarkentaa. Kaikki Hybrid Analyysiin lisätyt tiedostot ovat julkisia eikä palveluun kannata lähettää mitään arkaluontoisia tietoja sisältäviä tiedostoja.

Threatcrowd.org yhdistää eri verkkosivuja, IP-osoitteita sekä tiedostoja toisiinsa graafisessa näkymässä. Tämä on hyödyllistä tietoa esimerkiksi selvittäessä, mitkä muut verkkosivut voivat tarjota haittaohjelmia.

Urlscan.io mahdollistaa verkkosivujen tutkimisen ilman vierailua haitallisessa verkko-osoitteessa. Urlscan.io avulla pystyy selvittämään yksinkertaisesti esimerkiksi mihin muihin palveluihin ja IP-osoitteisin verkkosivu ottaa yhteyttä käyttäjän vieraillessa verkkosivulla.

4.2 Virtuaalikoneiden asennus manuaalista analyysiä varten

Staattisen analyysin avuksi käytettiin Malware Unicorn -projektia. Malware Unicornin sivuilta pystyy latamaan kaksi virtuaalikonetta, Sniffer- ja victim-virtuaalikoneet, jotka asennettiin VirtualBoxiin. Sniffer- ja victim-virtuaalikoneet ovat yhteydessä ainoastaan toisiinsa. Kuvassa 1 näkyy yksinkertainen verkkotopologia.



Kuva 1. Virtuaalikoneet kytkettynä toisiinsa

Asennus aloitettiin lataamalla isohkot Zip-tiedostot Malware Unicornin sivuilta. Sisältä löytyivät. ova-päätteiset tiedostot, jotka tuotiin VirtualBoxiin Import-toimintoa käyttäen. Asetuksia ei ollut tarpeellista muuttaa, vaan molemmat virtuaalikoneet asennettiin oletusasetuksilla. Virtuaalikoneiden tuonnin jälkeen asennettiin molemmille virtuaalikoneille VirtualBoxin Guest addons ja käynnistettiin virtuaalikoneet uudelleen. Seuraavaksi tarkistettiin, että uhrin asetuksissa ”Drag and Drop” sekä ”Clipboard” ovat Bidirectional tilassa, sekä molemmissa virtuaalikoneissa oli käytössä sisäinen verkko nimeltä ”intnet”. Lisäksi Windowsin kokeiluversio uudelleen aktivoitiin ajamalla järjestelmävalvojana käynnistetyssä cmd-ikkunassa komento.

```
slmgr /rearm
```

Sniffer-virtuaalikoneen näppäimistöasettelu vaihdettiin suomalaiseksi komennolla.

```
setxkbmap fi
```

Seuraavaksi virtuaalikoneista otettiin snapshot. Etenkin Victim-virtuaalikoneesta on tärkeää ottaa snapshot, jotta voidaan helposti palata puhtaaseen tilaan ennen infektiota. Sniffer-virtuaalikone kuuntelee Victim-virtuaalikoneelta tulevia yhteyspyyntöjä ja vastaa haittaohjelmien tekemiin yhteyspyyntöihin inetsim-ohjemalla. Inetsim on oletuksena käynnissä Victim-virtuaalikoneessa, mutta ohjelman käynnissä olo varmistettiin komennolla.

```
ps -ef | grep inetsim
```

Ohjelman ollessa käynnissä komento tuottaa pitkän listan avoimia portteja, joita kuunnellaan. Mikäli komento ei tuota pitkää listaa avoimista porteista, täytyy inetsim käynnistää manuaalisesti komennolla.

```
/etc/init.d/inetsim start
```

InetSimin lisäksi WireShark on tärkeä analysointityökalu. WireShark ei toimi kunnolla ilman muutoksia käyttäjän ryhmiin tai WireSharkin ajamista sudo-oikeuksin, tämä tosin ei ole suotavaa ja käytännössä käyttäjä lisättiin WireShark ryhmään komennoilla.

```
sudo dpkg-reconfigure wireshark-common
sudo gpasswd -a $USER wireshark
```

Seuraavaksi tarkistettiin saako Victim-virtuaalikone yhteyden Sniffer-virtuaalikoneeseen käyttämällä avuksi Windowsin komentoriviä ja pingaamalla osoitetta 192.168.0.1.

```
ping 192.168.0.1
```

Yhteyden toimivuuden varmistuttua asennettiin Victim-virtuaalikoneelle muutama Malware Unicornin tarjoamasta virtuaalikoneesta puuttuva ohjelma: 010 Editor, RegShot, Notepad++, Malcode Analyst Pack ja Capture-BAT. Ohjelmat löytyivät Googlen avulla ja asentuivat ongelmitta. Capture-BATin pikakuvakkeen käynnistysparametreihin tehtiin muutoksia, jotta Capture-BAT kirjoitti lokitiedostonsa työpöydälle ja kopioi haittaohjelmien poistamat tiedostot erilliseen kansioon. Uusiksi parametreiksi annettiin seuraavat:

```
”C:\Program Files\Capture\CaptureBAT.exe” -c -l “C:Users\victim\Desktop\logs.txt”
```

4.3 Haittaohjelmakirjasto

Githubista löytyy käyttäjänimen ytisf alta projekti nimeltä ”theZoo”, jossa on massiivinen määrä erilaisia haittaohjelmia. Kaikki haittaohjelmat on pakattu .zip-tiedostoihin ja tiedostot on suojattu salasanalla ”infected” vahinkojen minimoimiseksi. Haittaohjelmia löytyy myös muualta, mutta theZoosta löytyy kattava määrä erilaisia haittaohjelmia.

4.3.1 Dyre-haittaohjelma

Tarkastelu aloitettiin hakemalla Dyre-haittaohjelma haittaohjelmakirjastosta ja siirtämällä se Victim-virtuaalikoneelle. Dyre ladattiin kirjastosta pakattuna tiedostona ja avattiin analysointiympäristössä. Pakatun tiedoston sisällä havaittiin 3 eri kansiota. Original-nimisestä kansiosta löytyi PDF-dokumentiksi naamioitu .scr-päätteinen tiedosto sekä fax.exe-tiedosto. PDF-dokumentiksi naamioitu tiedosto avattiin 010 Editorilla. 010 Editorilla huomattiin, että tiedostomuoto alkaa kirjaimilla MZ, joka tarkoittaa, että tiedosto käyttäytyy kuten .exe tiedosto.

Seuraavaksi käynnistettiin Capture-BAT järjestelmävalvojan oikeuksin ja suoritettiin PDF-dokumentiksi naamioitu tiedosto. Hetken odottelun jälkeen Capture-BAT suljettiin ja tarkasteltiin Capture-BATin luomaa lokitiedostoa. Lokitiedostosta kävi ilmi, että haittaohjelma on tehnyt rekisterimuutoksia ja luonut tiedoston ”googleupdaterr.exe”. Seuraavaksi haittaohjelma käynnisti luomansa googleupdaterr.exe tiedostonsa ja poisti alkuperäisen PDF-dokumentiksi naamioidun tiedoston. Alkuperäisen haittaohjelman ja uuden tiedoston MD5 arvoja vertailtiin. MD5 arvot olivat samoja, joka tarkoittaa, että haittaohjelma siirsi itsensä uuteen polkuun massamuistissa. Haittaohjelman tekemät rekisterimuutokset mahdollistavat haittaohjelman käynnistymisen käyttäjän sisäänkirjautumisen yhteydessä.

Rekisterimuutokset varmistettiin myös Regshotilla. Uhri palautettiin puhtaaseen tilaan ja Regshot käynnistettiin. Ensimmäinen kuva otettiin ennen haittaohjelman käynnistystä ja toinen haittaohjelman käynnistyksen jälkeen. Regshotilla todettiin samat rekisterimuutokset, jotka näkyivät Capture-BATin lokitiedostossa. Rekisterimuutokset vahvistettiin myös SysInternals-työkalupakista löytyvällä autoruns.exe työkalulla.

4.3.2 IllusionBot-haittaohjelma

Tarkastelu aloitettiin hakemalla IllusionBot-haittaohjelma haittaohjelmakirjastosta ja siirtämällä se Victim-virtuaalikoneelle. Seuraavaksi Sniffer-virtuaalikoneessa käynnistettiin Wireshark, tarkistettiin että inetsim on käynnissä ja aloitettiin portin enp0s3 kuuntelu. Victim-virtuaalikoneesta pingattiin Sniffer-virtuaalikonetta, jotta varmistuttiin, että sniffer-virtuaalikoneessa oleva Wireshark vastaanottaa liikennettä. Seuraavaksi IllusionBot-haittaohjelman pakattu tiedosto purettiin työpöydälle ja käynnistettiin Capture-BAT, jonka

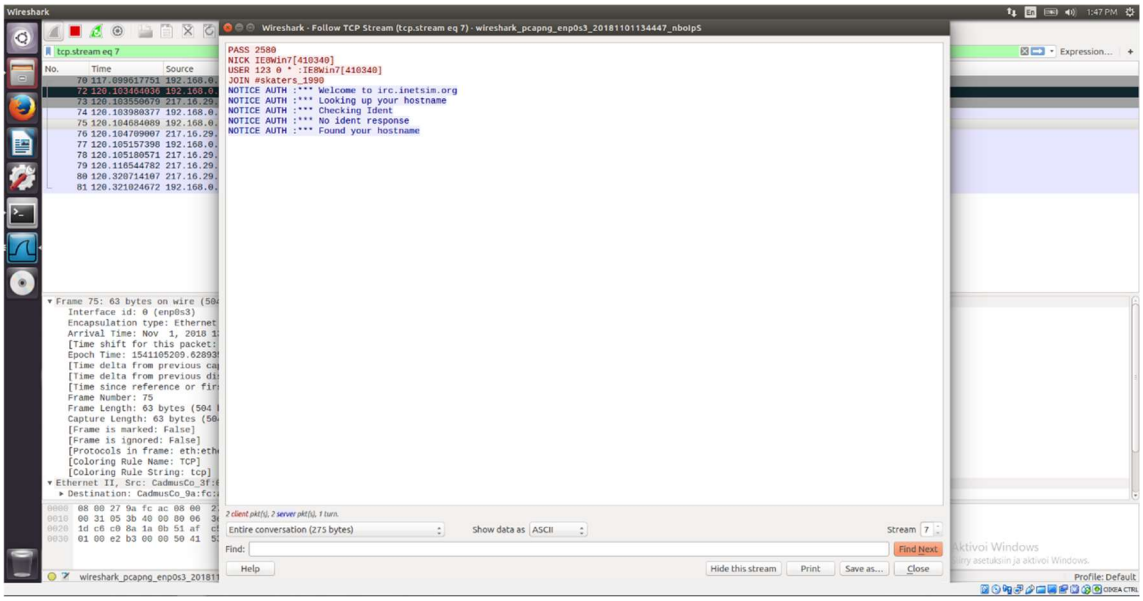
jälkeen käynnistettiin itse haittaohjelma. Wiresharkin avulla seurattiin, kuinka haittaohjelma yrittää ottaa yhteyttä komentokanavaansa. Seuraavaksi kaikki Victim-virtuaalikoneen liikenne ohjattiin Sniffer-virtuaalikoneelle komennolla.

```
route ADD 0.0.0.0 MASK 255.255.255.255 192.168.0.1
```

Sniffer-virtuaalikoneella tehtiin myös muutoksia reititykseen.

```
sudo iptables -t nat -A PREROUTING -i enp0s3 -j REDIRECT
```

Reititysmuutosten jälkeen Wiresharkista seurattiin, kuinka haittaohjelma luuli onnistuneensa ottamaan yhteyttä komentokanavaansa. Kuvassa 2 nähdään Wiresharkin kiinniottaman paketin sisältö.



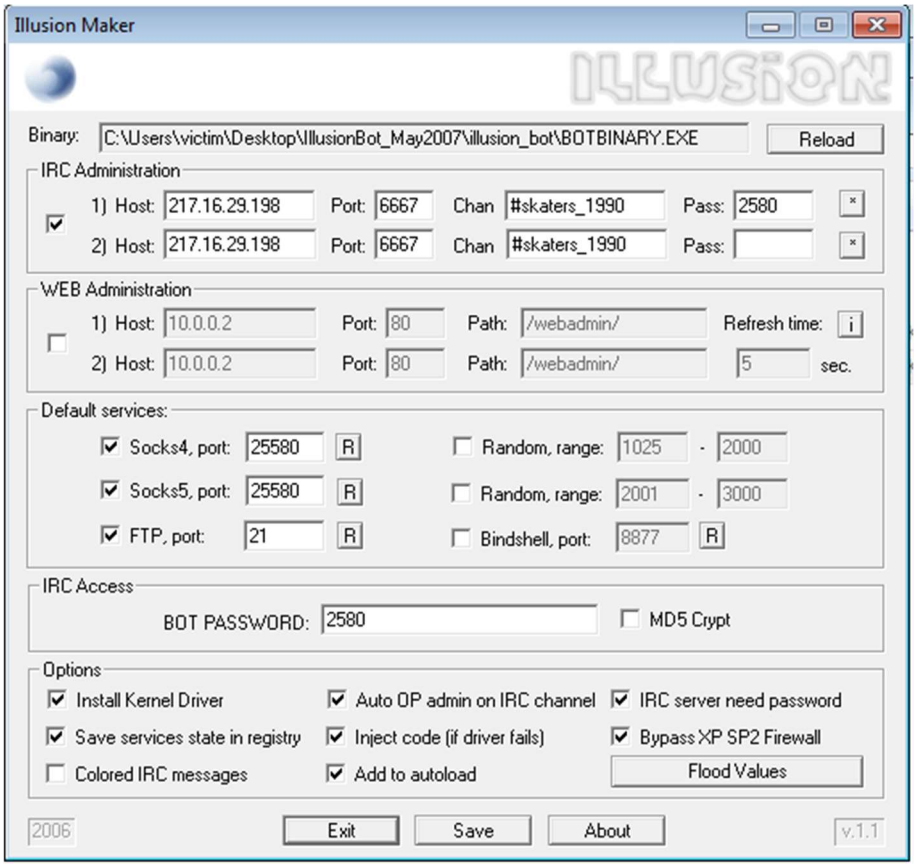
Kuva 2. Haittaohjelman lähettämän paketin sisältö Wiresharkissa

Oikeasti kyselyyn vastasi inetsim. Wiresharkin kaappaaman paketin sisällöstä voidaan vetää johtopäätös, että haittaohjelma yrittää kirjautua IRC-kanavalle. Kanavan nimi on #skaters_1990 ja salasana kanavalle on 2580. Käyttäjänimeksi haittaohjelma on antanut tietokoneen nimen. Seuraavaksi Inetsim pysäytettiin ja Netcat käynnistettiin parametrein.

```
nc -l -p 6667
```

Netcat mahdollistaa saman tiedon vastaanottamisen sekä komentojen lähettämisen haittaohjelmalle. Haittaohjelman komennot jäivät epäselväksi, mutta lähettämällä suuren määrän satunnaisia merkkejä haittaohjelman pystyi kaatamaan.

IllusionBot-kansiosta löytyi myös Build.exe. Build.exe ohjelmaa käyttämällä haittaohjelman toimintaa muokattiin. Kuvassa 3 näkyy Build.exen avulla muokattavat haittaohjelman parametrit.



Kuva 3. Build.exe ohjelmassa avattu IllusionBot-haittaohjelma

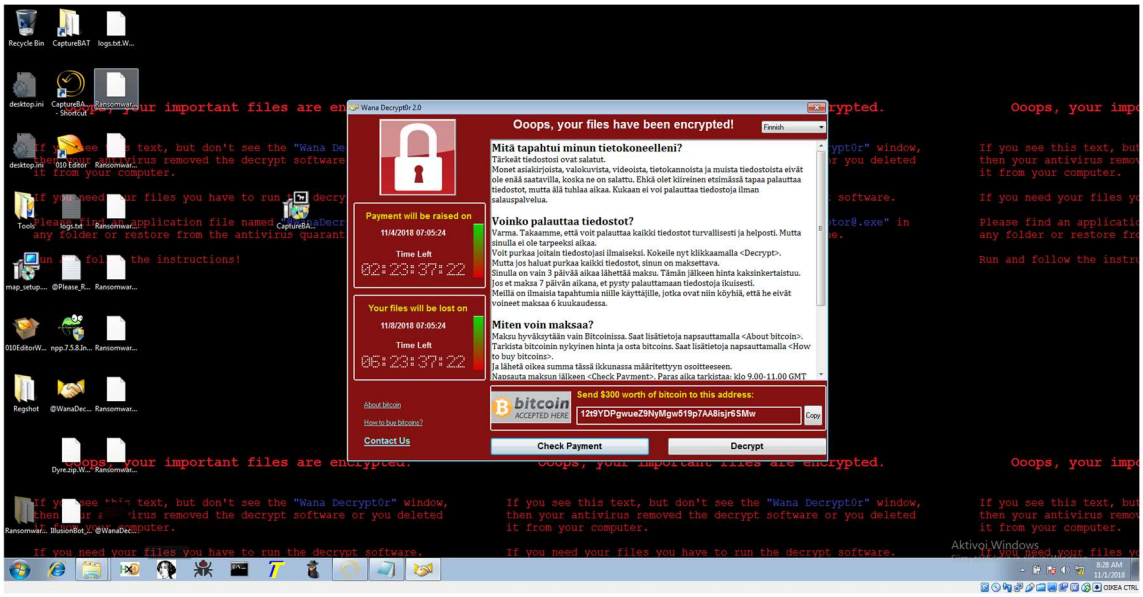
Build.exen avulla pystyi muokkaamaan FTP-palvelun porttia, IRC-palvelun osoitetta, porttia, kanavaa ja salasanaa. Build.exellä vaikuttaisi myös pystyvän aktivoimaan paikallisen hallintapaneelin haittaohjelmaan. Build.exessä olevien asetusten perusteella haittaohjelmalla on myös keino ohittaa Windows XP:n palomuuri. Windows 7-käyttöjärjestelmässä se vaikuta toimivan, sillä Capture-BATin lokitiedostossa ei havaittu muutoksia palomuurin asetuksiin. Capture-BATin lokitiedostosta havaittiin myös rekisterimuutos, jonka avulla haittaohjelma käynnistyy Windowsin käynnistyksen yhteydessä.

4.3.3 Cryptowall-kiristyshaittaohjelma

Tarkastelu aloitettiin lataamalla haittaohjelma haittaohjelmakirjastosta, avaamalla pakattu tiedosto ja käynnistämällä Capture-BAT. Sisältä löytyi cryptowall.bin-niminen tiedosto. Bin-tiedostoja ei pysty käynnistämään kuten normaaleja .exe-tiedostoja, joten haittaohjelma käynnistettiin komentorivin kautta. Haittaohjelman käynnistyttyä Capture-BATin lokitiedostosta nähtiin, kuinka haittaohjelma luo kansion ja uuden kansion sisälle 61b71d45.exe-tiedoston ja internat.exe-tiedoston. Tiedoston luonnin jälkeen haittaohjelma tekee rekisterimuutoksia, joiden seurauksena 61b71d45.exe ja internat.exe-tiedostot käynnistyvät Windowsin käynnistyksen yhteydessä. Haittaohjelma kopioi 61b71d45.exe-tiedoston myös AppData- ja Startup-kansioon. Startup-kansio on rekisterimuutosten ohella toinen tapa käynnistää ohjelma käyttäjän kirjautuessa sisään. Haittaohjelma lisää rekisteriin myös merkinnän, että 61b71d45.exe-tiedosto käynnistyy käyttäjän kirjautuessa sisään. Näiden lisäysten jälkeen haittaohjelma poisti rekisterilisäyksensä liittyen internat.exe-tiedostoon. Seuraavaksi haittaohjelma käynnisti Windowsista löytyviä työkaluja ja sammutti oman vanhan prosessinsa. Windowsin prosessien käynnistymisen jälkeen haittaohjelman seuraaminen Capture-BATin lokitiedoston avulla kävi hankalaksi ja useista yrityksistä huolimatta haittaohjelman ei havaittu kryptaavan levyllä olevia tiedostoja.

4.3.4 WannaCry-kiristyshaittaohjelma

WannaCry-haittaohjelman tarkastelu aloitettiin lataamalla haittaohjelma haittaohjelmakirjastosta, avaamalla pakattu tiedosto, ja käynnistämällä Capture-BAT. Pakatun kansion sisältä löytyi .exe-tiedosto, jonka laukaisu aloitti tiedostojen kryptauksen. Muutaman sekunnin odottelun jälkeen ruudulle ilmestyi kuvassa 4 näkyvä infoikkuna, jossa kerrottiin, että tiedostot on kryptattu, mutta ne voidaan vapauttaa maksamalla lunnaat Bitcoineina.



Kuva 4. WannaCry-haittaohjelma on kryptannut tietokoneella olevat tiedostot

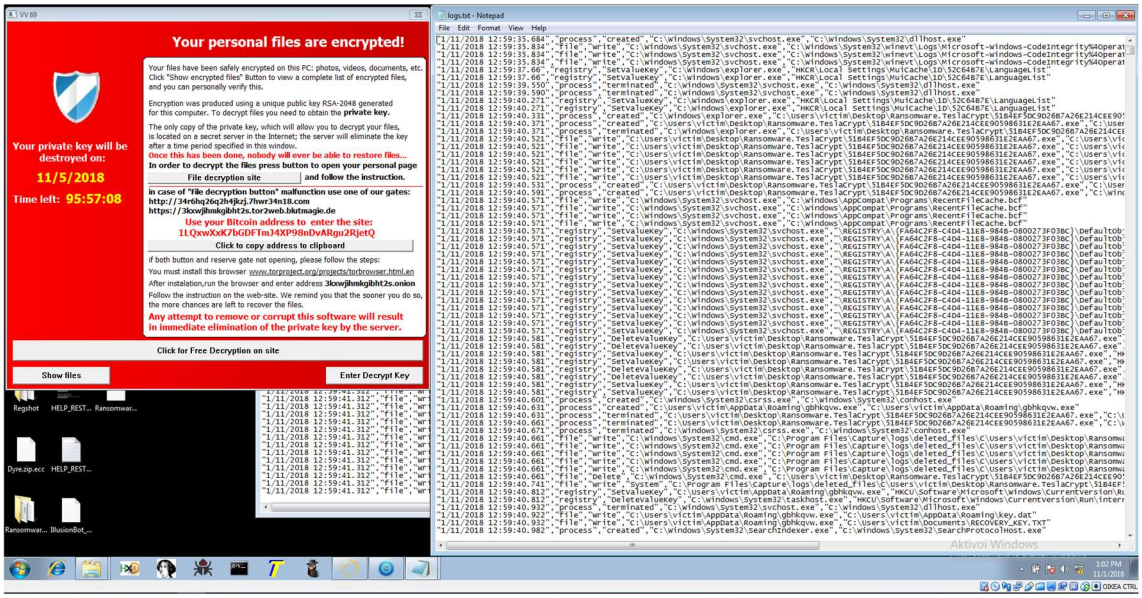
Infoikkunasta löytyy monta eri kielivaihtoehtoa ja myös suomen kieli löytyy valikosta. Suomen kieli on todennäköisesti käännetty kielenkääntäjäohjelmalla. Infoikkunasta löytyy myös vaihtoehto ”Decrypt”, jonka avulla pystyi palauttamaan muutaman tiedoston takaisin luettavaan muotoon, loput tiedostoista haittaohjelma suostui palauttamaan ainoastaan lunnaita vastaan.

Capture-BATin lokitiedosto joutui myös kryptauksen kohteeksi, joten virtuaalikone jouduttiin palauttamaan puhtaaseen tilaan ennen haittaohjelman laukaisua ja käynnistämään haittaohjelma uudestaan. Uudella laukaisukerralla lokitiedosto ehdittiin avata ennen kryptausta. Lokitiedosta nähtiin, kuinka haittaohjelma aloittaa täyttämällä oman käynnistyskansionsa muilla tiedostoilla, kuten infoikkunaohjelmallansa, TOR-selaimella ja kryptatuilla tiedostoilla. Tämän jälkeen haittaohjelma aloittaa muiden levyllä olevien tiedostojen kryptauksen. Haittaohjelman kryptattua kansion sisällön, se jättää usein jälkeen tekstitiedoston, mistä käy ilmi samat tiedot kuin haittaohjelman infoikkunasta. Haittaohjelma käynnistää myös muita prosesseja sekä tekee muutoksia rekisteriin. Kaikki virtuaalikoneelta löytyvät tiedostot on kryptattu muutamassa sekunnissa.

4.3.5 TeslaCrypt-kiristyshaittaohjelma

TeslaCrypt-haittaohjelman tarkastelu aloitettiin lataamalla TestaCrypt-haittaohjelma haittaohjelmakirjastosta, avaamalla pakattu tiedosto, ja käynnistämällä Capture-BAT.

TeslaCryptin kansion sisältä löytyi kolme tiedostoa, joista jokaisesta puuttui tiedostopääte. Kansion ensimmäiseen tiedostoon lisättiin .exe-tiedostopääte, jonka jälkeen haittaohjelman käynnistettiin. TeslaCrypt vaikutti kryptaavan tiedostoja huomattavasti hitaammin verrattuna WannaCry-haittaohjelmaan. Capture-BATin lokitiedoston sisältö muistutti WannaCry-haittaohjelman tekemiä toimia. Suurin osa lokista on levylle kirjoitusta ja rekisterimuutoksia. TeslaCryptistä löytyvä infoikkuna on saman kaltainen kuin WannaCryssä esiintullut. TeslaCryptin versiosta puuttuu kielenvaihto sekä mahdollisuus purkaa osa tiedostoista. Kuvassa 5 on TeslaCrypt-haittaohjelman infoikkuna vasemmalla ja CaptureBATin lokitiedosto oikealla.



Kuva 5. TeslaCrypt infoikkuna ja Capture-BAT lokitiedosto

5 AUTOMAATTINEN ANALYYSI

Tässä osassa opinnäytetyötä käydään läpi Cuckoo Sandboxin asennus, käyttö sekä Cuckoo Sandboxilla tehdyn analyysin tuloksia. Internetistä löytyy suuri määrä maksullisia ja maksuttomia haittaohjelmien analysointityökaluja, mutta Cuckoo Sandbox on tunnetuin ja mahdollisesti myös ainoa maksuton vaihtoehto, minkä voi asentaa omaan verkkoon. Ainuttakaan toista maksutonta ratkaisua ei tämän opinnäytetyön teon aikana löytynyt, joten Cuckoo Sandbox valikoitui automaattisesti alustaksi automaattista haittaohjelmien tutkimista varten.

5.1 Cuckoo Sandboxin yleiskatsaus

Cuckoo Sandbox on ilmainen avoimen lähdekoodin automaattinen analysointijärjestelmä. Cuckoo Sandboxille voi käytännössä tarjota minkä tahansa Windowsille, OS X:lle, Linuxille tai Androidille tarkoitetun tiedoston. Cuckoo Sandbox analysoi tiedoston käynnistämällä sen virtuaalikoneessa ja koostaa yksityiskohtaisen raportin tiedoston tekemistä toimista. Cuckoo Sandboxia pystyy käyttämään yhdeltä tietokoneelta lokaalisti taikka valjastaa sen usealle tietokoneelle ja hallita sitä etänä web-käyttöliittymän avulla. Cuckoo Sandboxiin löytyy suuri määrä kolmannen osapuolen lisäosia, jotka tekevät analyysistä tarkemman. (Stiching Cuckoo Foundation 2018)

Tulevissa kappaleissa käydään yksityiskohtaisesti läpi Cuckoo Sandboxin asennus. Cuckoo Sandbox asennetaan käsin, mutta Cuckoo Sandboxin asennusta varten on tehty lukuisia skriptejä, jotka asentavat Cuckoo Sandboxin muutamalla napin painalluksella.

5.2 Cuckoo Sandboxin asennuksen valmistelu

Cuckoo Sandboxia asennettaessa seurattiin Cuckoo Sandboxin virallista dokumentaatiota. Dokumentaatio löytyy osoitteesta ”https://cuckoo.readthedocs.io/en/latest/installation/”. Cuckoo Sandbox on kehitetty etupäässä Linux-pohjaisille käyttöjärjestelmille. Cuckoo Sandbox on mahdollista asentaa myös Windows-käyttöjärjestelmälle, mutta Linux valikoitui alustaksi yksinkertaisempana ja turvallisempana asennusalustana. Cuckoo Sandbox asennettiin kannettavalle tietokoneelle, kirjoitushetkellä Ubuntun uusimpaan 18.04 versioon. Muistitikku alustettiin Rufus-nimisellä ohjelmalla Ubuntun asennusta varten. Asennuksessa valittiin minimaalinen asennus, jotta ylimääräisiä ohjelmia ei asennu.

Tarpeettomat ohjelmat vievät tilaa massamuistista ja laajentavat haittaohjelmien hyökkäyspinta-alaa. Ubuntu päätettiin asentaa kaksoiskäynnistyksenä Windowsin rinnalle, koska tietokoneelta löytyi jo entuudestaan Windows-käyttöjärjestelmä. Kaksoiskäynnistyksen asennus onnistui ongelmitta ja molemmat käyttöjärjestelmistä toimivat ongelmitta rinnakkain. Päivitysten ajamisen jälkeen Ubuntu käynnistettiin uudelleen. Uudelleenkäynnistyksen jälkeen oltiin valmiita asentamaan Cuckoo Sandbox.

5.3 Vaadittavien komponenttien asennus

Cuckoo Sandboxin asennus käynnistettiin asentamalla Pythonin 2.7-versio. Python 2.7 asennettiin ajamalla alla olevat komennot komentorivillä.

```
$ sudo apt-get install python python-pip python-dev libffi-dev libssl-dev
$ sudo apt-get install python-virtualenv python-setuptools
$ sudo apt-get install libjpeg-dev zlib1g-dev swig
```

Pythonin lisäksi tarvittiin myös MongoDB sekä PostgreSQL. Cuckoo Sandboxin hallinta tapahtuu paikallisessa verkkosivussa, joka vaatii toimiakseen MongoDB ja PostgreSQL -tietokannat. Asennus suoritettiin ajamalla alla olevat komennot komentorivillä.

```
$ sudo apt-get install mongodb
$ sudo apt-get install postgresql libpq-dev
```

Cuckoo Sandboxin virtualisointi toteutettiin käyttämällä VirtualBox-virtuaalialustaa. VirtualBox asennettiin ajamalla alla olevat komennot komentorivillä.

```
$ echo deb http://download.virtualbox.org/virtualbox/debian bionic contrib | sudo tee -a /etc/apt/sources.list.d/virtualbox.list
$ wget -q https://www.virtualbox.org/download/oracle_vbox_2016.asc -O- | sudo apt-key add -
$ sudo apt-get update
$ sudo apt-get install virtualbox-5.2
```

Seuraavaksi asennettiin tcpdump työkalu, jolla on tarkoitus tarkastella, mihin osoitteisiin haittaohjelmat ottavat yhteyttä. Tcpdumpin asennus onnistui ajamalla alla olevat komennot komentorivillä.

```
$ sudo apt-get install tcpdump apparmor-utils
```

```
$ sudo aa-disable /usr/sbin/tcpdump
```

Tcpdump tarvitsee root-oikeudet, mutta koska Cuckoo Sandboxia ei haluta ajaa root-oikeuksin, tehtiin Tcpdump-työkalun oikeuksiin muutoksia.

```
$ sudo setcap cap_net_raw,cap_net_admin=eip /usr/sbin/tcpdump
```

Oikeuksien muutokset varmistettiin komennolla.

```
$ getcap /usr/sbin/tcpdump
/usr/sbin/tcpdump = cap_net_admin,cap_net_raw+eip
```

5.4 Cuckoo Sandboxin asennus

Cuckoo Sandboxin asennus aloitettiin luomalla käyttäjä nimeltä cuckoo.

Uusi käyttäjä luotiin komennolla.

```
$ sudo adduser cuckoo
```

Uusi cuckoo-käyttäjä lisättiin VirtualBoxin vboxusers ryhmään. Lisäys tapahtui ajamalla komento.

```
$ sudo usermod -a -G vboxusers cuckoo
```

Seuraavaksi VirtualBoxiin luotiin virtuaaliverkko virtuaalikoneelle. Virtuaaliverkko luotiin menemällä Global Toolseihin ja valitsemalla sieltä Host Network Manager. Host Network Manager -näkymässä luotiin uusi verkko painamalla Create. Verkkojen luonti viimeisteltiin komennoilla.

```
$ sudo iptables -t nat -A POSTROUTING -o enp2s0 -s 192.168.56.0/24 -j
MASQUERADE
$ sudo iptables -P FORWARD DROP
$ sudo iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
$ sudo iptables -A FORWARD -s 192.168.56.0/24 -j ACCEPT
$ sudo iptables -A FORWARD -s 192.168.56.0/24 -d 192.168.56.0/24 -j
ACCEPT
$ sudo iptables -A FORWARD -j LOG
```

Enp2s0 viittaa verkkokorttiin, jolla yhteys internettiin muodostetaan. Tämän pystyy tarkistamaan komennolla ”`ip address`”, tai aikaisempien Linux versioiden kohdalla komennolla ”`ifconfig`”. Iptables-muutoksien lisäksi aktivoitiin uudelleenreititys komennolla.

```
$ echo 1 | sudo tee -a /proc/sys/net/ipv4/ip_forward
$ sudo sysctl -w net.ipv4.ip_forward=1
```

Iptablesin muokkaukset säilyvät ainoastaan seuraavan tietokoneen sammutukseen asti. Iptables-muutoksista tehtiin pysyvät asentamalla iptables-persistent -ohjelma.

```
$ sudo apt install iptables-persistent
```

Cuckoo Sandboxista löytyy ominaisuus myös analyysikohtaiselle reititykselle. Tässä vaiheessa yksinkertainen reititys riittää, sillä ei ole tarkoitus, että haittaohjelmat pääsevät keskustelemaan internettiin tässä vaiheessa.

Oletuksena Cuckoo Sandboxin asetuksissa ei ole määritelty ohjelmaa ylläpitämään tietokantaa. Tietokantasovellukseksi valittiin aikaisemmin MongoDB ja se käytiin aktivoimassa Cuckoon Sandboxin asetustiedostossa. Asetustiedosto löytyy polusta `CWD/conf/reporting.conf` ja MondoDB:n asetukset keskivaiheelta tiedostoa.

5.5 Uhrin asennus

Uhrikäyttöjärjestelmäksi valikoitui 64-bittinen Windows 7 Home Premium -käyttöjärjestelmä. Windows 7 valikoitui uhriksi erittäin suurien käyttäjämäärien ja Windows 10:ä heikomman tietoturvan ansiosta. Kattavammassa analysoinnissa tulisi käyttää useampaa eri Windows-versiota sekä eri versioita esimerkiksi Microsoft Office sovelluksesta, mutta tässä vaiheessa opinnäytetyötä niitä ei asennettu. Windows 7-käyttöjärjestelmän lataaminen osoittautui ongelmallisemmaksi kuin alun perin oletettiin. Vuoteen 2015 asti Microsoft oli tarjonnut Windowsin ISO-imageja vapaasti ladattavaksi, mutta kirjoitushetkellä ainoastaan Windows 10 löytyy Microsoftin virallisilta sivuilta ladattavaksi. Tuoteavaimen avulla Windows 7:n pystyy lataamaan Microsoftin sivuilta, mutta ilmaista kokeiluversiota ei ole mahdollista enää mahdollista ladata. Ainoaksi vaihtoehdoksi osoittautui ISO-imagen lataus kolmannen osapuolen verkkosivulta. Tiedoston alkuperäisyys varmistettiin vertailemalla MD5-summia.

Windows asennettiin VirtualBoxin suosittelemilla resursseilla, koska varmuutta uhrivirtuaalikoneen resurssitarpeesta ei ollut ja määriä on mahdollista muokata tarpeen mukaan.

Käynnistyksen yhteydessä tulleesta virheviestistä ilmeni, että uusista tietokoneista löytyvä Secure Boot -ominaisuus estää VirtualBoxin käyttämän virtualisoinnin. Ongelma ratkesi käymällä BIOSissa ja poistamalla Secure Bootin käytöstä. Secure Bootin ollessa poissa käytöstä virtuaalikone käynnistyi normaalisti ja Windows asentui ongelmitta.

Seuraavaksi asennettiin Python-ohjelmointikirjaston uusin versio Windowsille. Pythonia haettaessa ilmeni, että Internet Explorer ei pysty näyttämään Python-projektin verkkosivuja oikein. Ongelma ratkaistiin asentamalla toinen selain, jonka avulla Pythonin lataus onnistui. Seuraavaksi tarkistettiin, että uhri-virtuaalikoneessa eivät ole päivitykset, palomuuri tai User Account Control käytössä. Tämän jälkeen uhri siirrettiin pois NAT-verkosta Host-only Adapter -virtuaaliverkkoon, joka luotiin aiemmassa vaiheessa.

Seuraavaksi siirrettiin Python-tiedosto Agent.py host-virtuaalikoneelta uhrivirtuaalikoneelle. Agent.py tiedosto on Cuckoo Sandboxin komponentti, joka lähettää tiedot uhrivirtuaalikoneelta Cuckoo Sandboxille. Tiedoston siirtoa varten Cuckoo Sandboxin dokumentointi suosittelee webserveriä, yksinkertaisempi ratkaisu oli siirtää tiedostot kansiojaon avulla. Webserverin avulla tapahtuva tiedonsiirto on mahdollinen kehityskohde. Agent.py tiedosto löytyy Cuckoo Sandboxin asennuspolun alta kansiosta /agent/. Agent.py-tiedostopääte muokattiin muotoon ”.pyw”, jotta tiedostoa ajettaessa ei syntyisi tyhjää komentoriviikkunaa. Lopuksi Agent.pyw lisättiin Windowsin mukana automaattisesti käynnistyvien ohjelmien listalle. Cuckoo Sandboxin mukana tulleen Python-skriptin lisäksi uhriin asennettiin Pillow-niminen Python-projekti, jonka avulla Cuckoo Sandbox pystyy ottamaan kuvia uhritietokoneen tapahtumista. Kaiken ollessa valmista uhrivirtuaalikone on hyvä käynnistää vielä kerran uudestaan ja varmistaa, että Python-skripti käynnistyy Windowsin käynnistyksen yhteydessä. Cuckoo Sandboxin Python-prosessi näkyy prosessien alla Task Managerissa nimellä pythonw.exe. Lopuksi uhrista otettiin vielä varmuuskopio. On erityisen tärkeää, että uhrivirtuaalikone on käynnissä varmuuskopion aikana.

5.6 Cuckoo Sandboxin käyttö

Cuckoo Sandboxin käyttämät pip- ja setuptools -työkalut päivitettiin ennen ensimmäistä käynnistyskertaa.

```
$ virtualenv venv
$ . venv/bin/activate
(venv)$ pip install -U pip setuptools
```

```
(venv)$ pip install -U cuckoo
(venv)$ cuckoo --cwd ~/.cuckoo
```

Cuckoo Sandbox käynnistetään komennolla.

```
cuckoo -d
```

Ensimmäisen käynnistyksen yhteydessä suositellaan lataamaan yhteisön tekemät haittaohjelma-allekirjoitukset, jotka auttavat Cuckoo Sandboxin analyysien tarkkuutta. Yhteisön tekemien allekirjoitusten lataus onnistuu komennolla.

```
$ cuckoo –cwd /home/*käyttäjä nimi*/.cuckoo community
```

Tiedostojen lisäys analysointia varten tekstipohjaisena tapahtuu komennolla.

```
(venv)$ cuckoo submit /tmp/sample1.exe
Success: File "/tmp/sample1.exe" added as task with ID #1
```

Analysointi prosessi aloitetaan komennolla

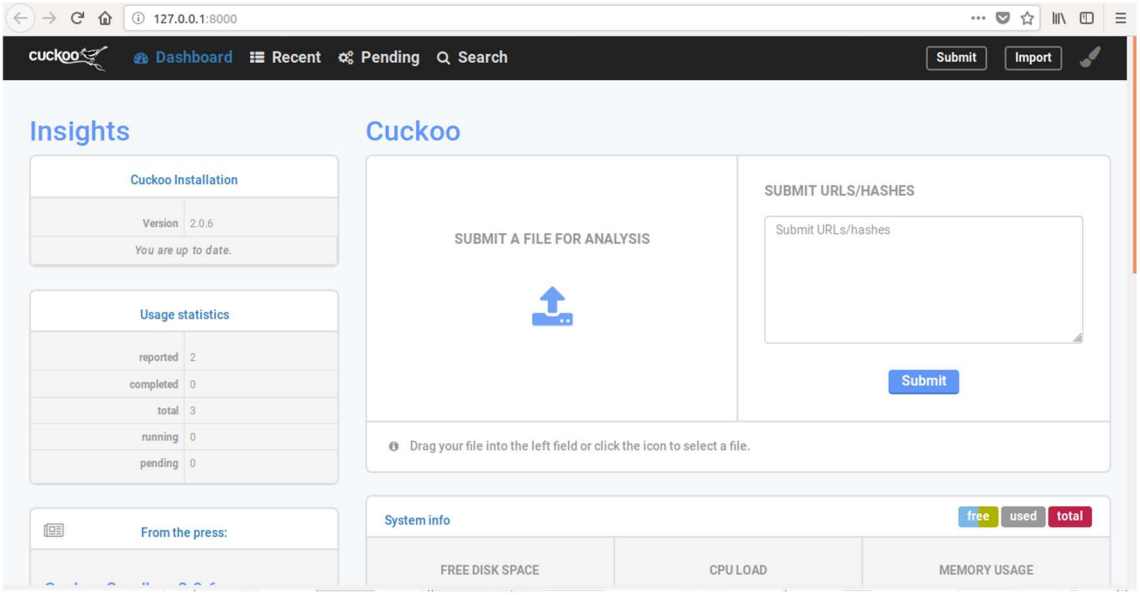
```
(venv)$ cuckoo -d -m 1
```

jossa -m 1 viittaa siihen, kuinka pitkään analyysi saa maksimissaan kestää minuutteina. Web-käyttöliittymä käynnistetään komennolla.

```
(venv)$ cuckoo web
Performing system checks...

System check identified no issues (0 silenced).
March 31, 2017 - 12:10:46
Django version 1.8.4, using settings 'cuckoo.web.web.settings'
Starting development server at http://localhost:8000/
Quit the server with CONTROL-C.
```

Cuckoo Sandboxin web-käyttöliittymän ollessa päällä Cuckoo Sandboxia pystyy hallitsemaan ottamalla yhteyden osoitteeseen http://localhost:8000/ tai http://127.0.0.1:8000/. Kuvassa 6 nähdään Cuckoo Sandboxin web-käyttöliittymän etusivu.



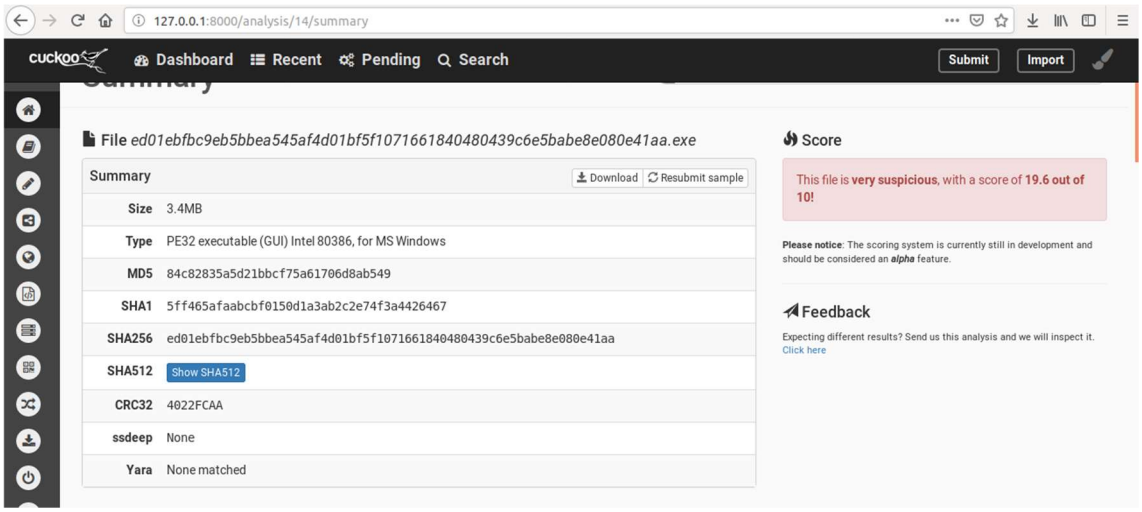
Kuva 6. Cuckoo Sandboxin web-pohjainen hallintapaneeli.

Uusien tiedostojen jonoon lisääminen graafisessa käyttöliittymässä onnistuu etusivulla joko raahaamalla taikka painamalla ”Submit a file for analysis” -kuvaketta. Jos saat tietokoneen uudelleen käynnistyksen jälkeen virheviestin tiedostoja lisätessä, ongelma todennäköisesti johtuu siitä, että VirtualBoxissa olevan virtuaalikoneen verkkokortti tai segmentti ei ole käynnissä. Ongelma korjaantuu käynnistämällä VirtualBox sekä virtuaalikone ennen Cuckoo Sandboxia.

5.7 Cuckoo Sandboxin analyysi

WannaCry-haittaohjelma, josta tehtiin myös dynaaminen analyysi, syötettiin Cuckoo Sandboxille. Cuckoo Sandbox antoi haittaohjelmalle arvioksi 19,6 / 10 vaarallisuudesta.

Arviointi on tällä hetkellä varhaiskehitysvaiheessa, eikä näin ollen täysin luotettava, kuten tuloksesta voidaan havaita. Kuvassa 7 nähdään tiivistelmä WannaCry-haittaohjelman tuloksista Cuckoo Sandboxissa.



Kuva 7. Tiivistelmä WannaCry-haittaohjelmasta Cuckoo Sandboxissa

Cuckoo Sandbox arvioi tapahtumia ja antaa niille värikoodin. Sininen kuvaa huomioitavaa infoa, keltainen on epäilyttävää toimintaa ja punainen on haitallinen tapahtuma. WannaCry-haittaohjelma tuottaa pitkän listan tapahtumia ja haitalliset tapahtumat käytiin läpi. Kuvassa 8 nähdään WannaCry-haittaohjelman tekemät haitalliset toimet virtuaalikoneella.

Kuva 8. WannaCry-haittaohjelman haitalliset tapahtumat

Cuckoo Sandboxin analysoinnin perusteella haittaohjelma kommunikoi useaan eri IP-osoitteeseen, todennäköisesti raportoidakseen uudesta saastuneesta tietokoneesta ja ottaakseen vastaan lisäohjeita tai esimerkiksi käskyn purkaa salaus. Haittaohjelma käynnistyy Windowsin mukana sekä vakoilee käyttäjän toimia ja kytkee pois käytöstä Windowsin Shadow Copy -varmuuskopiointiominaisuuden. Se poistaa suuren määrän tiedostoja ja lisää suureen määrään tiedostoja WannaCry-kiristysohjelman tiedostopäätteen, käyttää Windowsin työkaluja epäilyttävillä parametreillä sekä asentaa Tor-selaimen. Tor-selain mahdollistaa pääsyn niin sanottuun pimeään verkkoon. Pimeä verkko sisältää laittomuuksia ja sen käyttö on anonyymimpää kuin normaali verkkoselaus. Lunnaiden maksun voi suorittaa pimeässä verkossa ja siksi Tor-selaimen asennus on Cuckoo Sandboxin mielestä haitallista.

Extracted Artifacts -osio luettelee erikseen komentorivin komennot, sekä ohjelman, jossa rivi oli suoritettu. Behavioral Analysis -osio näyttää haittaohjelman tapahtumat järjestyksessä sekä prosessien sisällön. Network Analysis -osio kertoo haittaohjelman tekemät yhteyspyynnöt sekä niiden sisällön. Dropped Files -osio listaa haittaohjelman tallentamat tiedostot, salaushaittaohjelman kohdalla lähes koko kovalevyn sisältö listautuu. Cuckoo Sandbox mahdollistaa myös eri analyysien vertailun, esimerkiksi saman haittaohjelmien versioiden vertailemiseksi. Analyysidatan voi myös viedä raakadatana toiseen Cuckoo Sandbox -instanssiin, mutta tuloksien vienti esimerkiksi PDF- tai csv-tiedostona harmillisesti puuttuvat.

Mielenkiintoinen huomio tapahtui uhri-virtuaalikoneelle kirjauduttaessa. Cuckoo Sandbox ei ollut palauttanut uhrivirtuaalikonetta takaisin puhtaaseen tilaan. Tämän epäiltiin johtuvan analyysiin määritellystä kahden minuutin enimmäiskestosta tai haittaohjelma oli onnistunut rikkomaan Cuckoo Sandboxin komponentin, joka toteuttaa virtuaalikoneen palautuksen puhtaaseen tilaan. Huomion arvoista oli myös, että haittaohjelman infoikkunassa olevat tekstit olivat suomen kielellä. Suomenkielinen lunnasvaatimus oli mielenkiintoinen, koska uhrivirtuaalikoneella oleva Windows-käyttöjärjestelmä on englanninkielinen ja uhrikoneella ei ollut pääsyä internettiin. Pohdinnan jälkeen syyksi epäiltiin näppäinasettelua, joka oli asetettu suomen kielelle.

6 YHTEENVETO

Opinnäytetyön tarkoituksena oli analysoida haittaohjelmien toimintaa manuaalisin ja automaattisin menetelmin. Opinnäytetyössä käytiin läpi eri työvälineitä manuaalisen analyysin toteuttamiseksi Windows-alustalla. Lisäksi käytiin läpi menetelmät haittaohjelma-analyysin tekemiseen sekä automaattisen analysointityökalun asentaminen ja automaattisella työkalulla analysointi. Dynaaminen analyysi onnistui odotetusti ja haittaohjelmista pystyttiin erittelemään infektion merkkejä. Erityisesti kirityshaittaohjelmien analysointi oli mielenkiintoista, johtuen niiden ajankohtaisuudesta ja tuhovoimasta. Kiristyshaittaohjelmia analysoidessa keskitytään löytämään, miten haittaohjelma saastuttaa käyttöjärjestelmän ja miten haittaohjelman toiminnan pystyy estämään.

Cuckoo Sandbox -työkalu osoittautui helppokäyttöiseksi ja kevyeksi asentaa esimerkiksi vähälle käytölle jääneelle kannettavalle tietokoneelle. Cuckoo Sandboxin monipuolisuus vapaan lähdekoodin työkaluna ja paikallisesti asennettuna sopii myös hyvin yritysten käyttöön. Cuckoo Sandboxin käyttö ei vaadi samanlaista teknistä osaamista kuin koodi- ja muistianalyysi ja on näin hyvä aloituskohta haittaohjelmien analysointiin. Cuckoo Sandboxin etu verkosta löytyviin automaattisiin analysointipalveluihin on paikallisen asennuksen tuoma varmuus siitä, että arkaluontoiset tiedostot eivät päädy vääriin käsiin.

Työn pohjalta voidaan todeta, että haittaohjelmien analysointi manuaalisin ja automaattisin keinoin onnistuu myös yksityishenkilöltä ilman kustannuksia. Analyysin syvyys rajoittuu ainoastaan taitotiedon puutteeseen, sillä koodi- ja muistianalyysi vaativat kattavaa ymmärrystä ohjelmoinnista ja yleisesti tietotekniikasta.

Työtä pystyy laajentamaan ottamalla tarkasteluun myös koodi- ja muistianalysointia. Tässä työssä ei käsitelty koodi- tai muistianalyysia kuin teoriatasolla, vaativuudesta ja ajankäytännöllisistä syistä johtuen. Työtä pystyy jatkamaan myös tutkimalla muiden käyttöjärjestelmien haittaohjelmia sekä tutkimalla, miten virustentorjuntaohjelma onnistuu suojaamaan käyttöjärjestelmään haittaohjelmilta. Myös Cuckoo Sandboxista löytyy ominaisuuksia, joita ei otettu käyttöön tässä työssä, esimerkiksi analyysikohtainen verkotus ja muiden käyttäjien luomat moduulit.

LÄHTEET

Aikio, E. 2014. Bot-verkot pahantahtoisissa käyttötarkoituksissa. Metropolia ammattikorkeakoulu. http://urn.fi/URN:NBN:fi:amk-201403042832.

Aldeid. 2017. LordPE. Viitattu 13.10.2018. https://www.aldeid.com/wiki/LordPE.

Aldeid. 2017. PEiD. Viitattu 13.10.2018. https://www.aldeid.com/wiki/PEiD.

Apple II History. 2018. 23-Viruses. Viitattu 2.12.2018. https://apple2history.org/history/ah23/#03.

Chalk, A. 2018. A new kind of ransomware forces you to play PUBG to unlock your files. Viitattu 11.11.2018. https://www.pcgamer.com/a-new-kind-of-ransomware-forces-you-to-play-pubg-to-unlock-your-files/.

Cygwin. 2018. Cygwin FAQ. Viitattu 10.10.2018. http://www.cygwin.com/faq.html#faq.what.what.

DabBoss. 2013. Chapter 8 MacMag. Viitattu 25.11.2018. https://www.cknow.com/cms/vtutor/chapter-8-macmag.html.

Dominguez, A. 2018. History of computer viruses: Creeper and Reaper. Viitattu 2.12.2018. https://blog.pandorafms.org/creeper-and-reaper/.

Don, H. 2016. Notepad++ About. Viitattu 10.10.2018. https://notepad-plus-plus.org/.

DuPaul, N. 2012. Common Malware Types: Cybersecurity 101. Viitattu 14.10.2018. https://www.veracode.com/blog/2012/10/common-malware-types-cybersecurity-101.

G Data. 2018. History of malware. Viitattu 2.12.2018. https://www.gdatasoftware.com/seccuritylabs/information/history-of-malware.

Hakkarainen, J. 2015. Malware Analysis Environment for Windows Targeted Malware. Jyväskylän ammattikorkeakoulu. http://urn.fi/URN:NBN:fi:amk-2015061613468.

Harán, J. 2018. Malware of the 1980s: Looking back at the Brain Virus and the Morris Worm. Viitattu 2.12.2018. https://www.welivesecurity.com/2018/11/05/malware-1980s-brain-virus-morris-worm/.

Hex-Rays. 2017. IDA: About. Viitattu 10.10.2018. https://www.hex-rays.com/products/ida/index.shtml.

Kaspersky. 2018. History of malicious programs 1987. Viitattu 2.12.2018. https://encyclopedia.kaspersky.com/knowledge/year-1987/.

Kultanen, M. 2016. Haittaohjelmien analysointijärjestelmä : Cuckoo Sandbox. Jyväskylän ammattikorkeakoulu. http://urn.fi/URN:NBN:fi:amk-2016053010817.

Malware Wikia. 2018. Vienna. Viivattu 2.12.2018. http://malware.wikia.com/wiki/Vienna.

OllyDbg. 2014. OllyDbg. Viitattu 13.10.2018. http://www.ollydbg.de/.

Rouse, M. 2005. Elk Cloner. Viitattu 11.11.2018. https://searchsecurity.techtarget.com/definition/Elk-Cloner.

Russinovich, M. 2018. Sysinternals Suite. Viitattu 9.10.2018. https://docs.microsoft.com/en-us/sysinternals/downloads/sysinternals-suite.

Snyder, D. 2010. The very first viruses: Creeper, Wabbit and Brain. Viitattu 2.12.2018. http://infocarnivore.com/the-very-first-viruses-creeper-wabbit-and-brain/.

Stiching Cuckoo Foundation. 2018. Cuckoo. Viitattu 24.10.2018. https://cuckoosandbox.org/.

Sweetscape. 2018. 010 Editor. Viitattu 9.10.2018. http://www.sweetscape.com/010editor/.

The Honeynet Project. 2018. Capture-BAT Download Page. Viitattu 13.10.2018. https://www.honeynet.org/node/315.

Virus Wikia. 2018 Denzuko. Viitattu 2.12.2018. http://virus.wikia.com/wiki/Denzuko.

Wikipedia. 2018. Dr. Solomon's Antivirus. Viitattu 2.12.2018. https://en.wikipedia.org/wiki/Dr_Solomon%27s_Antivirus.

Zimmer, D. 2018. Malcode Analyst Pack. Viitattu 9.10.2018. http://sandsprite.com/iDef/MAP/.

Zorz, Z. 2018. VirtualBox Guest-to-Host escape 0day and exploit released online. Viivattu 7.11.2018. https://www.helpnetsecurity.com/2018/11/07/virtualbox-guest-to-host-escape-0day/.