jamk

Haittaohjelmien analysointiin tarkoitettujen avoimen lähdekoodin hiekkalaatikkotuotteiden arviointi

Topias Tonteri

Opinnäytetyö
Joulukuu 2022
Tekniikan ala
Insinööri (AMK), tieto- ja viestintätekniikka

jamk | Jyväskylän ammattikorkeakoulu
University of Applied Sciences

jamk

Kuvailulehti

Tonteri, Topias

Haittaohjelmien analysointiin tarkoitettujen avoimen lähdekoodin hiekkalaatikkotuotteiden arviointi

Jyväskylä: Jyväskylän ammattikorkeakoulu. Joulukuu 2022, 46 sivua.

Tekniikan ala. Insinööri (AMK), tieto- ja viestintätekniikka. Opinnäytetyö AMK.

Julkaisun kieli: Suomi

Julkaisulupa avoimessa verkossa: Kyllä

Tiivistelmä

Opinnäytetyön toimeksiantajana toimi Elisa Santa Monica Oy.
Opinnäytetyön tarkoituksena oli selvittää vertailevan tutkimuksen avulla mitkä avoimen lähdekoodin haittaohjelmahiekkalaatikko tuotteet olisivat käytännöllisiä. Samaan aikaan tarkoituksena oli parantaa osaamista ja tietoa kyseisistä tuotteista. Vertailtaviksi tuotteiksi valikoitui Cuckoo-hiekkalaatikko ja CAPE-hiekkalaatikko. Vertailujen tarkoituksena oli selvittää valikoitujen hiekkalaatikkosovelluksien soveltuvuutta toimeksiantajan käyttöön.

Opinnäytetyön teoriaosuudessa käytiin läpi yleisesti haittaohjelmatyyppejä ja miten haittaohjelmia voidaan analysoida. Osuudessa käytiin myös läpi hiekkalaatikointi yleisesti ja miten haittaohjelmat välttelevät analyysia.

Opinnäytetyössä käytiin läpi valikoituneiden tuotteiden asennus prosessi, analysointi kyvykkyydet yleisellä tasolla, ominaisuudet ja tuotteiden ylläpidettävyys.

Tuotteiden asennus, tutkiminen ja testaus toteutettiin itse rakennetulla Oracle VirtualBox -virtualisointiohjelmalla virtualisoidussa ympäristössä.

Opinnäytetyön tuloksena saatiin käytyä läpi molempien vertailtavien tuotteiden asennus prosessi, analysointikyvykkyyksiä, ominaisuudet ja ylläpidettävyys. Samalla saatiin toteutettua kaksi toimivaa haittaohjelmien hiekkalaatikointiympäristöä. Tuotteiden läpi käydyt kategoriat vertailtiin, jolla saatiin selvitettyä tuotteiden soveltuvuutta toimeksiantajan ja lisättyä osaamista tuotteista.

Avainsanat (asiasanat)

Haittaohjelma-analyysi, Cuckoo sandbox, CAPE sandbox

Muut tiedot (salassa pidettävät liitteet)

jamk

Description

Tonteri, Topias

Open-source malware sandbox – product evaluation

Jyväskylä: JAMK University of Applied Sciences, December 2022, 46 pages.

Engineering and technology. Bachelor of Engineering, Information Technology. Bachelor's thesis.

Permission for open access publication: Yes

Language of publication: Finnish

Abstract

The thesis was assigned by Elisa Santa Monica Oy.
Objective of the thesis was to compare different open-source malware sandbox products. By comparing the product's main objective was to find out which product would be most useful. Secondary objective was to educate about malware sandbox products. Products chosen for the thesis were Cuckoo sandbox and CAPE sandbox.

The theory section of the thesis was to go through fundamentals about malware and how malware can be analyzed. Fundamentals of what is sandboxing and how malware tries to evade analysis was also gone through.

The thesis went through installing process, analysis capabilities, features, and maintainability of the chosen products.

Products were installed on self-made environment using Oracle VirtualBox virtualizing software.

Results of the thesis was two working malware sandbox environments and comparison of installing process, analysis capabilities, features, and maintainability from both chosen products. Resulting in increase of knowledge about the products and their suitability for Elisa Santa Monica.

Keywords/tags (subjects)

Malware analysis, Cuckoo sandbox, CAPE sandbox

Miscellaneous (Confidential information)

Sisältö

Kuviot

Taulukot

1 Johdanto

Haittaohjelmat kehittyvät jatkuvasti ja uusia haittaohjelma hyökkäyksiä tapahtuu päivittäin. Haittaohjelmilla hyötyvät rikolliset ovat alkaneet tienaamaan enemmän rahaa. Kiristyshaittaohjelmahyökkäyksien määrä on pandemian aikana noussut räjähdysmäisesti. VMware Carbon Black kyberturvallisuus yrityksen tutkimuksen mukaan kiristyshaittaohjelma hyökkäyksien määrä nousi maaliskuussa vuonna 2020 148 %. (Culafi. 2020.) Noususta kertoo myös rahan määrä, mitä kiristyshaittaohjelma hyökkääjät ovat tienanneet yrityksiltä. Vuonna 2019 kiristyshaittaohjelmaryhmät tienasivat 152 miljoonaa dollaria, kun taas vuonna 2020 ryhmät tienasivat 692 miljoonaa dollaria. Rahan määrä siis noin viisinkertaistui pandemian alettua. (Paganini. 2022.)

Haittaohjelmien analysointi turvallisessa ympäristössä on tärkeää. Haittaohjelman analysointi on tärkeää mahdollisen hyökkäyksen jokaisessa vaiheessa. Analysoimalla haittaohjelmia saadaan tietoa haittaohjelman toiminnasta, mitä ei välttämättä ole ympäristöstä vielä huomattu. Kun toiminta tunnetaan paremmin, saadaan vihjeitä siitä, mitä dataa on varastettu ja miten haittaohjelmalta voidaan puolustautua jatkossa. (How Malware Analysis Benefits Incident Response. n.d.)

Opinnäytetyössä tutkittiin haittaohjelmien analysointiin tarkoitettuja hiekkalaatikkotuotteita. Hiekkalaatikkotuotteella on tarkoitus ajaa haittaohjelmaa turvallisessa ympäristössä ja analysoida, miten haittaohjelma käyttäytyy. Opinnäytetyössä oli tarkoitus selvittää vertailemalla, avoimen lähdekoodin hiekkalaatikkotuotteita. Vertailtaviksi hiekkalaatikkotuotteiksi valikoitui Cuckoo ja CAPE tuotteet.

2 Toimeksiantaja

Opinnäytetyön toimeksiantajana toimii Elisa Santa Monica Oy. Elisa Santa Monica Oy on vuonna 2004 perustettu yritys. Yrityksen nimi perustettaessa oli Santa Monica Networks. Vuonna 2017 Santa Monica Networksin osti Elisa Oyj ja se liitettiin Elisa Oyj:n sisaryhtiöksi nimellä Elisa Santa Monica Oy. Elisa Santa Monica Oy:lla on henkilöstöä 144 ja sen liikevaihto on noin 45 miljoonaa euroa. (Taloustiedot n.d.) Elisa Santa Monica tuottamat palvelut voidaan jakaa neljään eri ryhmään: tietoverkot, datakeskukset, tietoturva ja koulutuspalvelut. Opinnäytetyön aihe on saatu tietoturvapalveluihin kuuluvalta Elisa kyberturvallisuuskeskukselta. (Palvelut ja ratkaisut n.d.)

3 Tutkimusmenetelmä

Opinnäytetyön tarkoituksena oli vertailla avoimen lähdekoodin haittaohjelmien analysointiin tarkoitettuja hiekkalaatikkotuotteita. Opinnäytetyössä arvioitiin hiekkalaatikkotuotteiden kyvykkyyksiä ja ylläpidettävyyttä. Tutkimuksen tavoitteiden perusteella valikoitui tutkimusmenetelmäksi vertaileva tutkimus. Tutkimus vastaa kysymykseen ”Mikä on arvioitavista avoimen lähdekoodin hiekkalaatikkotuotteista käytännöllisin toimeksiantajalle?”

Opinnäytetyössä on toimittu tutkimuseettisesti. Työn teossa ei ole varastettu tietoa, eikä rikottu lakia.

4 Haittaohjelmat ja analysointi

Haittaohjelma (eng. Malware) on mukana jollain tavalla suurimmassa osassa tietoturvahyökkäyksissä. Haittaohjelma on tietokoneohjelma, joka aiheuttaa jollain tavalla haittaa tietokoneelle, käyttäjälle tai tietoverkolle. (Sikorski & Honig 2012.) Haittaohjelmien analysointi on tärkeää, sillä on tärkeää ymmärtää mitä eri haittaohjelmat voivat järjestelmissä tehdä. Tämän avulla puolustava taho pystyy paremmin estämään potentiaalisen uhan. Analysoinnista voi löytyä eri näköisiä vaarantumisindikaattoreita, joiden avulla voidaan opettaa eri antivirus tai muita kehittyneitä tunnistusohjelmia tunnistamaan haittaohjelma.

4.1 Haittaohjelmatyypit

Haittaohjelmia on monia eri tyyppejä. Haittaohjelma tyypit ovat yleisesti nimetty haittaohjelman toiminnan tai leviämistavan perusteella. Haittaohjelmien luokittelu ei ole kuitenkaan aivan yksinkertaista, sillä tämänpäiväiset haittaohjelmat sisältävät monia eri osia, joilla on omat toiminnallisuutensa. Eri toiminnallisuudet useasti osuvat eri haittaohjelma luokkaan. Alla yleisimpiä haittaohjelma tyyppejä.

Adware eli mainosohjelma on haittaohjelma, jonka tarkoituksena on esittää käyttäjälle mainoksia tai myydä tämän tietoja. Esimerkkinä vuonna 2017 paljastunut kiinalainen Fireball mainosohjelma. Iso digitaalinen markkinointi yritys sai haittaohjelmansa 250 miljoonalla tietokoneelle. Ohjelma

muutti käyttäjän selainta vaihtaen käyttäjän kotisivun ja oletushakukoneen. Tavoitteena oli näyttää käyttäjälle enemmän mainoksia ja kerätä dataa käyttäjän verkkosurffailusta. (Perekalin 2017.)

Spyware eli vakoiluohjelma on haittaohjelma, jonka tarkoituksena on kerätä tietoa tietokoneesta ja käyttäjästä. Esimerkkinä vuonna 2014 paljastunut pohjoiskorealainen DarkHotel. Haittaohjelma levisi hotellien vieraille tarkoitettujen langattomien verkkojen avulla. Käyttäjän kirjauduttuaan hotellin verkkoon tietokone sai ilmoituksen ladata uusin Adobe-päivitys. Koneelle ladattiin päivityksen sijaan haittaohjelma DarkHotel. Ohjelma tallensi käyttäjän näppäinpainallukset ja lähetti nämä haittatoimijalle. (Zetter 2014.)

Ransomware eli kiristyshaittaohjelma. Haittaohjelman tavoitteena on salata laitteen tiedostot ja pyytää lunnaita salauksen purkuavainta vastaan. Esimerkkinä vuonna 2017 levinnyt WannaCry. WannaCry kiristyshaittaohjelma levisi matomaisesti käyttäen EternalBlue haavoittuvuutta Windows-käyttöjärjestelmän SMB-protokollassa. Haittaohjelma salasi laitteen tiedostot ja pyysi lunnaita salauksen purkuavainta vastaan. Haittaohjelmaa nähtiin 150 eri maassa tuhansilla koneilla. (What is WannaCry ransomware? n.d.)

Virus on haittaohjelma, joka kopioi itseään ja leviää laitteesta laitteeseen. Esimerkki viruksena Yhdysvaltojen valtion tuottama Stuxnet-virus. Virusta käytettiin Iranissa ydinvoimalan käyttöönoton häiritsemisessä. (What is Stuxnet? n.d.)

Trojan virus eli troijalainen virus on nimetty antiikin kreikan Troijan hevosen mukaan. Virus on naamioitunut haitattomaksi, mutta todellisuudessa on haitallinen. Esimerkkinä Zeus troijalainen virus. Zeus on talous petos haittaohjelma. Ohjelma tekeytyy haitattomaksi, mutta siinä on takaovi, jolla pystytään ohittamaan yritysten turvallisuus toimet. Haittaohjelma lisäsi tietyille verkkosivuille lisäkenttiä, joissa pyydettiin esimerkiksi luottokortin tietoja. (Baykal 2015.)

Worm eli mato on haittaohjelma, joka leviää laitteesta laitteeseen automaattisesti. Haittaohjelma ei vaadi minkäänlaista ihmisen aktivointia levitäkseen. (Computer Worm, n.d.) Esimerkkinä NotPetya kiristyshaittaohjelma, joka levisi matona Ukrainassa. Haittaohjelma salasi kaikki laitteen tiedostot ja siirtyi seuraavaan laitteeseen. (Banerjea 2018.)

Fileless Malware on haittaohjelma, joka on pelkästään tietokoneen muistissa. Haittaohjelma ei ole siis tiedostona laitteen massamuistissa, vaan piilee yleisimmin RAM-muistissa. Esimerkkinä WannaMine haittaohjelma. WannaMine käyttää samaa EternalBlue haavoittuvuutta kuin WannaCry ja NotPetya. Haittaohjelma louhii laitteelta kryptovaluutta Moneroa. (Paganini 2018.)

4.2 Haittaohjelman analysoinnin tavat

Haittaohjelmien analysointi on tärkeä osa haittaohjelmia vastaan puolustautumisessa. Kun ymmärretään, miten haittaohjelma käyttäytyy ja mikä sen tavoite on, on helpompi puolustautua sitä vastaan. On myös tärkeää saada tietoa jo tapahtuneissa haittaohjelma saastumissa mitä haittaohjelma on laitteella tehnyt. Haittaohjelmien analysoinnin voi jakaa kolmeen eri kategoriaan staattinen analyysi, dynaaminen analyysi ja hybridi analyysi. (Sihwail 2018.)

Staattinen analyysi on haittaohjelmien analyysi menetelmä, jolla analysoidaan tiedostoa ilman, että sitä ajetaan. Menetelmä toimii siten, että tiedostosta poimitaan kaikki mahdolliset staattiset tiedot. Esimerkiksi hash, strings, kirjastot, tuodut funktiot ja resurssit. Tämän avulla saadaan ymmärrystä mitä haittaohjelma tekee ilman, että sitä ajetaan. (Bencherchali 2019.) Virustorjuntaohjelmat myös käyttävät yleisesti staattista analyysiä skannauksessaan. Staattisella analyysillä voidaan saada selville, onko tiedosto haitallinen vai ei. Staattinen analyysi on kuitenkin helposti kierrettävissä ja tämän takia sitä ei voida pitää luotettavana. Staattinen analyysi on siis yksinkertainen ja nopea tapa, mutta ei hyvin toimiva hienostuneempia haittaohjelmia vastaan. (Sikorski & Honig 2012.) Kehittyneempi ja luotettavampi tapa on takaisin mallintaa haittaohjelma. Haittaohjelma ladataan purkajaohjelmaan ja analysoidaan mitä ohjeita prosessori suorittaa. Tällä menetelmällä saadaan selville tarkalleen mitä ohjelma tekee. Menetelmä on kuitenkin paljon vaikeampaa ja aika vievämpää kuin perusstaattinen analyysi. Menetelmä vaatii paljon erikoistunutta osaamista eri osa-alueilta. (Sikorski & Honig 2012.)

Dynaaminen analysointi on haittaohjelman analysointi menetelmä, jossa ideana on ajaa haittaohjelma laitteella ja tarkkailla miten ohjelma käyttäytyy. Tällä menetelmällä on tärkeää, että haittaohjelma ajetaan erillisessä ympäristössä. Ilman syvempää ohjelmointiymmärrystä on myös todennäköistä, että haittaohjelmasta jää huomaamatta tärkeitä toimintoja. Kehittyneempi tapa on dynaamisesti analysoida debuggerin avulla. Tällä tavalla saadaan parempi ymmärrys ohjelmasta. (Sikorski & Honig 2012.)

Hybridi-analyysissa on molemmat staattinen ja dynaaminen analyysi yhdistettynä. Ideana tällä on kattaa toistensa puutteet. Haittaohjelmassa voi olla esimerkiksi jotain, jota ei staattisella analyysillä helposti näe, mutta paljastuu helposti ohjelmaa ajettaessa dynaamisesti. (Sihwail 2018.)

4.3 haittaohjelmien tunnistusmenetelmät

Haittaohjelmien tunnistusmenetelmät voi jakaa karkeasti kahteen eri laajempaan kategoriaan, tunnistepohjainen ja käytöspohjainen. (Souri & Hosseini 2018.)

Tunnistepohjainen haittaohjelmien tunnistusmenetelmä on tunnistusmenetelmistä suosituin, sillä suurin osa antivirus-tuotteista käyttävät tätä menetelmää haittaohjelmien tunnistamiseen. Antivirus tuotteet purkavat epäilyttävän ohjelman ja etsivät erilaisia rakenteita ohjelman koodista. Rakenteiden avulla tarkistetaan ovatko nämä samoja kuin aiemmin tunnetuissa haittaohjelmissa. (Tahir 2018.)
Huonona puolena tällä menetelmällä on, että haittaohjelman tulee olla jo tunnettu. Menetelmällä ei siis voi tunnistaa uusia tai tuntemattomia haittaohjelmia. Hyvänä puolena menetelmällä on, että se on nopea ja tehokas tapa tunnistaa jo tunnettuja haittaohjelmia. (Heena & Mehtre 2021.)

Käytöspohjainen haittaohjelmien tunnistusmenetelmä on Aslanin ja Sametin (2020) mukaan esitetty toimivan käyttäen hyväkseen erilaisia monitorointityökaluja. Työkaluilla yritetään tunnistaa haittaohjelmien toimintaa. Aslan ja Samet toteavat myös, että vaikka haittaohjelman koodi voi muuttua ohjelman toiminta pysyy silti samana. Menetelmällä on siis tehokkaampi tunnistaa uusia haittaohjelmia. (Aslan & Samet 2020.)

4.4 Hiekkalaatikkomenetelmä

Haittaohjelman dynaamisessa analyysissa on tarkoitus ajaa haittaohjelmaa laitteella. Haittaohjelmaa ei kuitenkaan kannata ajaa omalla laitteella ja vaarantaa laite. Jotta haittaohjelman voi turvallisesti analysoida dynaamisesti on kehitetty hiekkalaatikkomenetelmä. (Anand 2019.)
Hiekkalaatikko on fyysinen tai virtuaalinen turvallinen ympäristö, jossa on turvallista ajaa haittaohjelmia ja analysoida näitä. (Arntz 2020.)

Virtualisiointi on yleinen tapa toteuttaa hiekkalaatikko ympäristö. Yksi tapa toteuttaa ympäristö on virtuaalikone. Virtuaalikoneella tarkoitetaan konetta, joka toimii ohjelmiston avulla. Yhdessä fyysisessä laitteistossa voi olla monta eri virtuaalista konetta. (What is a virtual machine (VM)? n.d.) Hiekkalaatikko ympäristön toteuttamisessa virtualisoinnilla on kuitenkin huono puoli. Edistyneemmät haittaohjelmat ovat kehitetty tunnistamaan, jos niitä ajetaan virtuaalikoneen sisällä. Yleisesti jos haittaohjelma tunnistaa olevansa virtuaalikoneen sisällä haittaohjelma ei suostu suorittamaan haitallista toimintojaan. Analysoidessa tämä aiheuttaa virheitä ja haittaohjelma voidaan tunnistaa haitattomaksi. (Abrams 2020.)

Kontitus on virtuaalikoneita nopeampi ja kevyempi tekniikka. Virtuaalikoneet ovat täysiä valmiita kokonaisuuksia, jotka sisältävät koko käyttöjärjestelmän. Kontit sen sijaan ovat valmiita itsenäisiä sovelluksia, jotka sisältävät vain tarvittavat riippuvuudet sovelluksen toimintaan. Kontit ovat eristettyjä itsenäisiä muusta laitteen käyttöjärjestelmästä. Koska kontit ovat valmiita paketteja niitä on myös nopea ottaa käyttöön. (Wallen 2020.)

Kontitus teknologiaa voi hyödyntää myös haittaohjelmien analysoinnissa. Konteilla voi ajaa haittaohjelmien analysointi ohjelmia, jolloin analysointi alustalla ei ole mitään ylimääräistä. Kontit ovat myös eristettyjä ympäristöjä. Koska kontit ovat nopeita ottaa käyttöön saa haittaohjelman analysointi ympäristönkin nopeasti käyttöön. (Zeltser n.d.)

4.5 Haittaohjelmakirjasto

Haittaohjelmien analysoinnin harjoittelemista varten tarvitaan turvallinen lähde, josta saadaan ajankohtaisia haittaohjelmia analysoitavaksi. Uusia ylläpidettyjä haittaohjelmanäytteitä ei ole helppo löytää jatkuvasti. Tätä varten on olemassa erilaisia ilmaisia ja maksullisia lähteitä, joista voi ladata itselleen haittaohjelmanäytteitä. (Gençaydın 2021.) TheZoo haittaohjelmavarasto on ilmainen kaikille avoin lähde, joka ei vaadi erillistä rekisteröitymistä. Palvelu tarjoaa haittaohjelmista kiinnostuneille haittaohjelmia analysointia varten. (Zeltser, 2021.)

4.6 Haittaohjelmien kiertotekniikat

Haittaohjelmat ovat kehittyneet paljon Internetin alkupäivistä, kuten ovat myös puolustusmenetelmät. Tästä syystä haittaohjelmien tuottajat ovat joutuneet kehittämään haittaohjelmia välttämään havaitsemisen. (Marpaung, Sain & Lee 2012.) Tässä luvussa tarkastellaan minkälaisilla eri tavoilla haittaohjelmat voivat vältellä analysointia. Haittaohjelmien välttely tekniikat voidaan jakaa kahden isomman kategorian alle samalla tavalla kuin luvussa 4.2. Luvussa keskitytään enemmän dynaamisen analyysin välttelymenetelmiin, sillä hiekkalaatikkotuotteet ovat dynaamista analysointia. Staattisen analyysin välttely menetelmät ovat kuitenkin hyvä tietää.

Dynaamisen analysoinnin välttelyssä haittaohjelma pyrkii tunnistamaan olevansa hiekkalaatikossa ja lopettaa toimintansa. Haittaohjelma yrittää siis tunnistaa millaisessa ympäristössä sitä ajetaan. Haittaohjelman tunnistaessa, että sitä ajetaan esimerkiksi hiekkalaatikkoympäristössä. Haittaohjelma ei aja haitallista koodiaan ja analyysin tulokseksi tulee, että haittaohjelma olisi turvallinen. Sekä Bulazel ja Yener (2017) että Afianian, Neksefat, Sadeghiyan ja Baptiste (2018) esittelevät tunnistepohjaisen menetelmän. Menetelmässä haittaohjelma etsii erilaisia merkkejä, joiden avulla se päättelee, ajetaanko sitä virtualisoidussa tai emuloidussa ympäristössä. Afianian ja muiden (2018) mukaan merkit voivat olla esimerkiksi asennetut laiteajurit, avoimet tiedostot ja rekisteriavaimet. Bulazel ja Yener (2017) lisäävät, että merkkejä voi myös olla käyttäjät, tiedostot, järjestelmän asetukset ja käynnissä olevat prosessit. (Bulazel ym. 2017; Afianian ym. 2018)

Staattisen analyysin välttely menetelmät perustuvat haittaohjelman koodin obfuskointiin eli tarkoituksenmukaiseen monimutkaistamiseen. Obfuskoinnin tarkoitus on peittää haitallinen koodi ja täten ohittaa esimerkiksi antivirus tuotteen analysointi tai analyytikon käsin analysointi. Yksinkertainen obfuskointitapa on vaihtaa haittaohjelman koodin järjestystä. Koodin uudelleen järjestelyllä pyritään vaikeuttamaan analysoijan työtä. Muita yksinkertaisia tapoja häiritä haittaohjelma analystin analyysiä on esimerkiksi ylimääräisen koodin lisääminen haittaohjelmaan. Ylimääräisen koodin lisäämisellä tarkoitetaan haittaohjelmaan koodin lisäämistä koodia, joka ei tee mitään. Eli koodilla ei ole minkäänlaista vaikutusta, miten haittaohjelma toimii. Tämä vaikeuttaa myös staattisen analysointi tuotteen toimintaa. Staattisilla analysointituotteilla on vaikeuksia tunnistaa, onko koodi turhaa vai ei. Staattisen analysoinnin tuotteita voidaan myös vältellä esimerkiksi salaamalla haittaohjelma. Alkuperäinen haittaohjelman koodi on salattu, joten sitä on mahdoton lukea ilman salauksen purkamista. Haittaohjelma kuitenkin tarvitsee tavan purkaa salauksensa, jotta sitä voidaan

ajaa. Salauksen purkava haittaohjelma on siten mahdollista tunnistaa ja tällä tavalla tunnistaa, että haitallista ohjelmaa yritetään ajaa. (Singh & Singh 2018.)

Reverse Turing Test -menetelmällä haittaohjelma yrittää päätellä onko ihminen käyttänyt laitetta, jossa sitä ajetaan. Haittaohjelma olettaa, että sitä ajetaan hiekkalaatikossa, jos se ei huomaa mitään ihmisen vuorovaikutusta. Haittaohjelmalla voi olla monia tapoja tunnistaa, ettei laite ole ihmisen käyttämä. Haittaohjelma voi esimerkiksi odottaa hiiren klikkausta tai hakea käyttäjän viimeisen syötön. Menetelmästä on tulossa suositumpi, sillä hiekkalaatikolla on vaikeata puolustautua tätä menetelmää vastaan. (Afianian ym. 2018.)

5 Tuotteiden vertailu

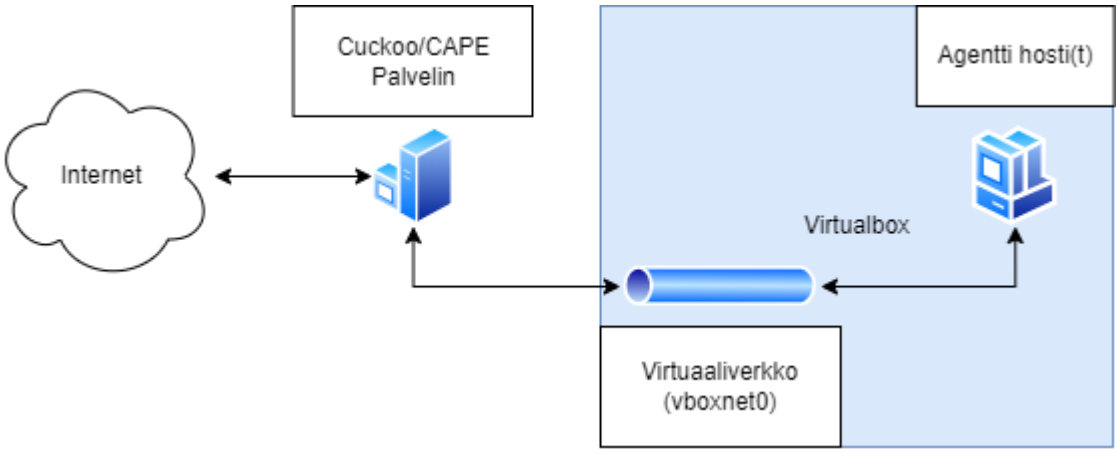
Opinnäytetyön tarkoituksena oli tutkia erilaisia haittaohjelma-analyysin hiekkalaatikkotuotteita. Avoimen lähdekoodin haittaohjelma-analyysin hiekkalaatikkotuotteita oli opinnäytetyön tekohetkenä yllättävän pieni valikoima saatavilla. Suurin osa tuotteista olivat joko maksullisia tai nettisivuilta tarjottavia palveluita. Vertailtaviksi tuotteiksi valikoitui Cuckoo ja CAPE tuotteet. Molemmat tuotteet ovat avoimen lähdekoodin tuotteita ja soveltuvat toimeksiantoon.

Tuotteita vertailtiin käyttäen pistejärjestelmää. Tuotteet saivat pisteitä eri kategorioista. Arvioitavina kategorioina oli tuotteiden asennus, tuotteiden ominaisuudet, tuotteiden analysointi kyvykkyydet ja tuotteiden ylläpidettävyys. Jokaisesta kategoriasta annettiin pisteitä yhdestä viiteen (1-5).

Tuotteiden vertailtavissa kategorioissa käytiin seuraavasti läpi asioita. Asennus-kategoriassa käytiin läpi tuotteiden asennus prosessi. Ominaisuus-kategoriassa käytiin läpi tuotteiden konfigurointi mahdollisuuksien kautta tuotteiden ominaisuuksia ja myös hieman merkittävämpiä ominaisuuksia, mitä konfiguraatio tiedostojen ulkopuolelta löytyi. Analysointi-kategoriassa tehtiin tuotteilla testianalyysi, jonka jälkeen tutkittiin mitä tuloksia tuotteet ovat tuottaneet. Ylläpidettävyys-kategoriassa käytiin läpi, miten tuotteita voisi päivittää.

5.1 Ympäristö

Haittaohjelma hiekkalaatikko tuotteiden vertailuun tarvittiin ympäristö, johon tuotteet voitiin asentaa. Ympäristöön tarvittiin palvelin, jossa itse hiekkalaatikko tuotetta ajettiin ja palvelimelle käytettäväksi agentti. Palvelimen käyttöjärjestelmäksi valikoitui Linux Ubuntu 20.04.5 LTS. Molemmat tuotteet on suositeltu asennettavaksi Linux käyttöjärjestelmälle. Agenttikoneet toteutettiin käyttäen Windows 7 käyttöjärjestelmää. Windows 7 käyttöjärjestelmä valikoitui uudemman Windows 10 sijasta, koska Windows 7 käyttöjärjestelmää oli suositeltu käytettäväksi tuotteiden dokumentaatiossa. Agentti koneet ovat virtualisoitu hiekkalaatikko palvelimella. Virtualisointi tuotteeksi valikoitui Oracle VirtualBox. VirtualBox soveltui molempiin tuotteisiin. VirtualBoxin avulla luotiin myös eristetty virtuaaliverkko, joka toimi verkkona minkä välillä hiekkalaatikkopalvelin ja agentti kone kommunikoi. Hiekkalaatikko palvelimella oli yhteys internettiin, joka oli mahdollista jakaa tarvittaessa myös virtuaaliverkolle ja tätä kautta agentti koneelle. Liikenne kuitenkin kulki hiekkalaatikko palvelimen kautta. Alla olevassa kuviossa 1 on kuvattuna tuotteiden vertailuun luotu ympäristö. Kuvattu ympäristö soveltui molempien tuotteiden käyttöön.



Kuvio 1. Hiekkalaatikkotuotteiden vertailu ympäristö.

Itse haittaohjelmien ajo tapahtuu agenttikoneella, joten konetta tarvitsi hieman parannella, jotta haittaohjelma ei tunnistaisi olevansa virtuaalikoneen sisällä. Mikäli haittaohjelma tunnistaa olevansa virtuaalikoneen sisällä tämä tekisi jotain ei haitallisen näköistä tai ei käynnistyisi ollenkaan. (Tavares 2022.) Konetta luodessa annettiin koneelle enemmän kuin minimi määrä resursseja, jotta kone näyttäisi normaalilta koneelta. Koneelle annettiin enemmän kuin yksi prosessorin ydintä käyttöön ja RAM-muistia myös enemmän kuin normaalisti virtuaalikoneelle annettaisiin. Koneelta

poistettiin VirtualBoxiin liittyvä virtualboxquestadditions ohjelma, jota ei asenneta muihin kuin virtuaalikoneisiin käytön helpottamiseksi. Koneelle asennettiin myös normaaleita ohjelmia, kuten Google Chrome selain, Windows Office ja Adobe PDF-lukija. Konetta myös käytettiin hieman, jotta koneelle generoituisi lokia ja muita käytönmerkkejä. Näillä toimilla parannettiin mahdollisuuksia, että haittaohjelma ajaisi normaalista ja täten sen analysointi onnistuisi.

5.2 Cuckoo-hiekkalaatikko

Cuckoo Sandbox on omien sanojensa mukaan, ”leading open source automated malware analysis system.” Eli johtava avoimen lähdekoodin automaattinen haittaohjelmien analysointijärjestelmä. Tuote oli selvästi opinnäytetyön valikoituneista tuotteista suosituin. Cuckoon tarina alkaa vuodesta 2010. Cuckoon ensimmäinen versio tehtiin Googlen Summer of Code leirillä. Seuraavana vuonna Cuckoosta julkaistiin ensimmäinen beta-versio. Vuonna 2012 Cuckoo voitti Rapid7:n pitämän Magnificent7 kilpailun ensimmäisen kierroksen. Cuckoon kehitystä on jatkettu vuosien varrella ja tämän hetken uusin versio on 2.0.7, joka julkaistiin vuonna 2019. Cuckoosta ei ole tämän jälkeen julkaistu uutta versiota. Cuckoota ollaan opinnäytetyön kirjoitus hetkellä uudelleen kirjoittamassa ja uusia päivityksiä ei ole lähihetkinä tulossa. (What is Cuckoo? n.d.)

5.2.1 Asennus

Cuckoo-hiekkalaatikko asennettiin kappaleessa 5.1 kuvattuun ympäristöön. Asennuksen voi jakaa kolmeen eri osaan, tarvittavien kirjastojen asennus, Cuckoon asennus ja agentin asennus. Cuckoo-hiekkalaatikko on toteutettu python-ohjelmointikielellä, joten ensimmäisenä vaiheena oli asentaa Python-komentotulkki ja Cuckoo hiekkalaatikon tarvitsevat kirjastot. Cuckoon dokumentaatiossa suositeltiin käyttämään Pythonin versiota 2.7. Python ja osa tarvittavista kirjastoista asennettiin seuraavalla komennolla:

sudo apt-get install python python-dev libffi-dev libssl-dev

sudo apt-get install libjpeg-dev zlib1g-dev swig

Tuotteen dokumentaatiossa suositeltiin myös Pythonin virtual enviromentin käyttöä, joten tämä asennettiin alla olevalla komennolla:

sudo apt-get install python virtualenv python-setuptools

Cuckoon web-käyttöliittymä on toteutettu käyttäen Djangoa, joka on Python pohjainen verkkokehys. Web-käyttöliittymä tarvitsee MongoDB tietokantaohjelmiston toimiakseen. MongoDB asennettiin seuraavalla komennolla:

sudo apt-get install mongodb

Cuckoo tarvitsee agenttikoneen, jota se käyttää haittaohjelmien ajamiseen. Agenttikone on virtualisoitu eristetty kone, jolla voidaan turvallisesti ajaa haittaohjelmia. Virtualisointi alustana käytettiin avoimen lähdekoodin Oracle VirtualBox tuotetta. VirtualBox asennettiin seuraavalla komennolla:

sudo apt install virtualbox

Jotta Cuckoo voisi tallentaa verkkoliikennettä Cuckoo-palvelimen ja agenttikoneen välillä, tarvittiin siihen työkalu. Cuckoo käyttää avoimenlähdekoodin tcpdump ohjelmaa. Lisäksi tarvittiin apparmor-utils, jotta voitiin sallia PCAP-tiedostojen muodostus Cuckoon toimesta. Tcpdump-ohjelma asennettiin alla olevalla komennolla:

sudo apt-get install tcpdump apparmor-utils

Ubuntu-käyttöjärjestelmässä on käytössä AppArmor-suojausmoduuli. Moduuli tulisi estämään PCAP-tiedostojen luonnin tcpdumpilla Cuckoon kansioon. Tcpdump sallittiin AppArmorissa seuraavalla komennolla:

sudo aa-disable /usr/sbin/tcpdump

Koska kyseessä on hiekkalaatikkoympäristö, jossa tullaan ajamaan haittaohjelmia, tulee Cuckoo asentaa käyttäjätunnukselle, jolla ei ole sudo-oikeuksia. Tämä on tärkeää, koska kyseessä on haittaohjelmien analysointia varten tehty ympäristö. Inhimilliset virheet ovat mahdollisia ja tällä tavalla vähennetään virheen sattuessa haittaohjelman suoritusoikeuksia virtuaalikoneella. Tätä varten luotiin ”cuckoo” -niminen käyttäjä, jolla ei ole ylimääräisiä tuotteen normaalin toimintaan tarpeettomia oikeuksia. Käyttäjä luotiin alla olevalla komennolla:

sudo adduser cuckoo

Jotta Cuckoo-hiekkalaatikko toimisi oikein tarvitsee käyttäjä määrätyn valikoiman oikeuksia, jotka ovat dokumentoitu Cuckoon ohjeistukseen. Cuckoon tulee kyetä hallita virtuaalikoneita, jotta se saa operoitua agenttikoneita, joilla haittaohjelmia ajetaan. Cuckoo käyttäjä lisättiin vboxusers ryhmään alla olevalla komennolla:

sudo usermod -a -G vboxusers cuckoo

Tcpdump ohjelma tarvitsee root-tason oikeudet toimiakseen. Turvallisuussyistä näitä oikeuksia ei aiemmin luodulle cuckoo-käyttäjätunnukselle kuitenkaan tule lisätä, joten ratkaisuna luotiin erillinen ryhmä, jonka alle tcpdump-sovellus siirrettiin. Lisäksi tcpdump-sovellukselle tuli asettaa tarvittavat oikeudet käyttäen setcap-komentoa. Pcap-niminen ryhmä luotiin, cuckoo-käyttäjä lisättiin pcap-ryhmän jäseneksi, tcpdump sovellus siirrettiin pcap ryhmään ja tarvittavat oikeudet tcpdump-sovellukselle lisättiin alla olevilla komennoilla:

sudo groupadd pcap

sudo usermod -a -G pcap cuckoo

sudo chgrp pcap /usr/sbin/tcpdump

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

Koska Cuckoo hiekkalaatikko on toteutettu Python 2 ohjelmointikieltä käyttäen, tarvittiin asentamiseen pip2-paketinhallintatyökalu. Pip2 ladattiin ja asennettiin alla olevilla komennoilla:

curl https://bootstrap.pypa.io/pip/2.7/get-pip.py --output get-pip.py

sudo python2 get-pip.py

Cuckoo asennettiin käyttäen cuckoo-käyttäjää. Lisäksi asennus suoritettiin pythonin virtuaaliympäristössä. Virtuaaliympäristöä käytettiin, jotta vältyttiin yhteensopivuus ongelmilta koneelle asennettujen jaettujen Python-kirjastojen kanssa, joita muut ohjelmat käyttöjärjestelmässä olisivat voineet aiheuttaa. Lisäksi virtuaaliympäristö mahdollistaa lisäkirjastojen asennuksen ilman root-tason oikeuksia. Virtuaaliympäristö luotiin ja aktivoitiin alla olevilla komennoilla:

virtualenv venv

. venv/bin/activate

Tarvittavat valmistelut itse Cuckoo hiekkalaatikko asentamiseen olivat nyt tehty. Cuckoo asennettiin alla oleville komennoilla:

pip2 install -U pip setuptools

pip2 install -U cuckoo

Seuraavaksi ajettiin kansiosta /home/cuckoo/.local/bin, cuckoo ensimmäisen kerran konfiguraatio- alla olevalla komennolla:

python2 cuckoo

Ensimmäisen kerran konfiguraation jälkeen kansio, jossa cuckoon konfiguraatio sijaitsee on: ” /home/cuckoo/.cuckoo”.

Seuraavana vaiheena oli asentaa cuckoolle agenttikone ja luoda palvelimen ja koneen välille virtuaalinen verkko. Verkko luotiin VirtualBox host-käyttöliittymä komennolla:

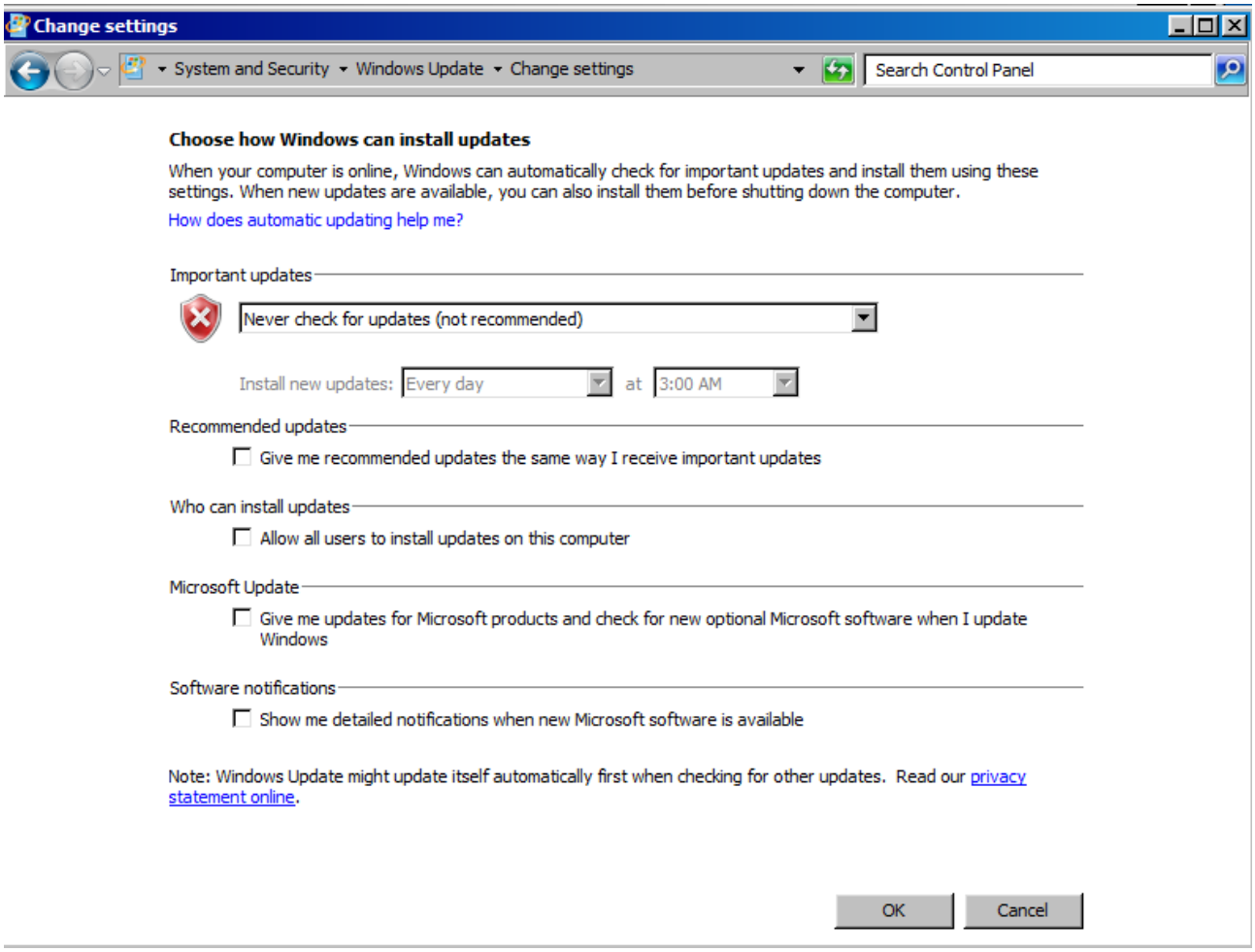
vboxmanage hostonlyif create

Komento loi virtuaaliverkon nimeltä vboxnet0. Seuraavaksi virtuaaliverkolle asetettiin oletusyhdyskäytävän IP-osoite ja verkkoalue alla olevalla komennolla:

vboxmanage hostonlyif ipconfig vboxnet0 --ip 192.168.56.1 –net-mask=255.255.255.0

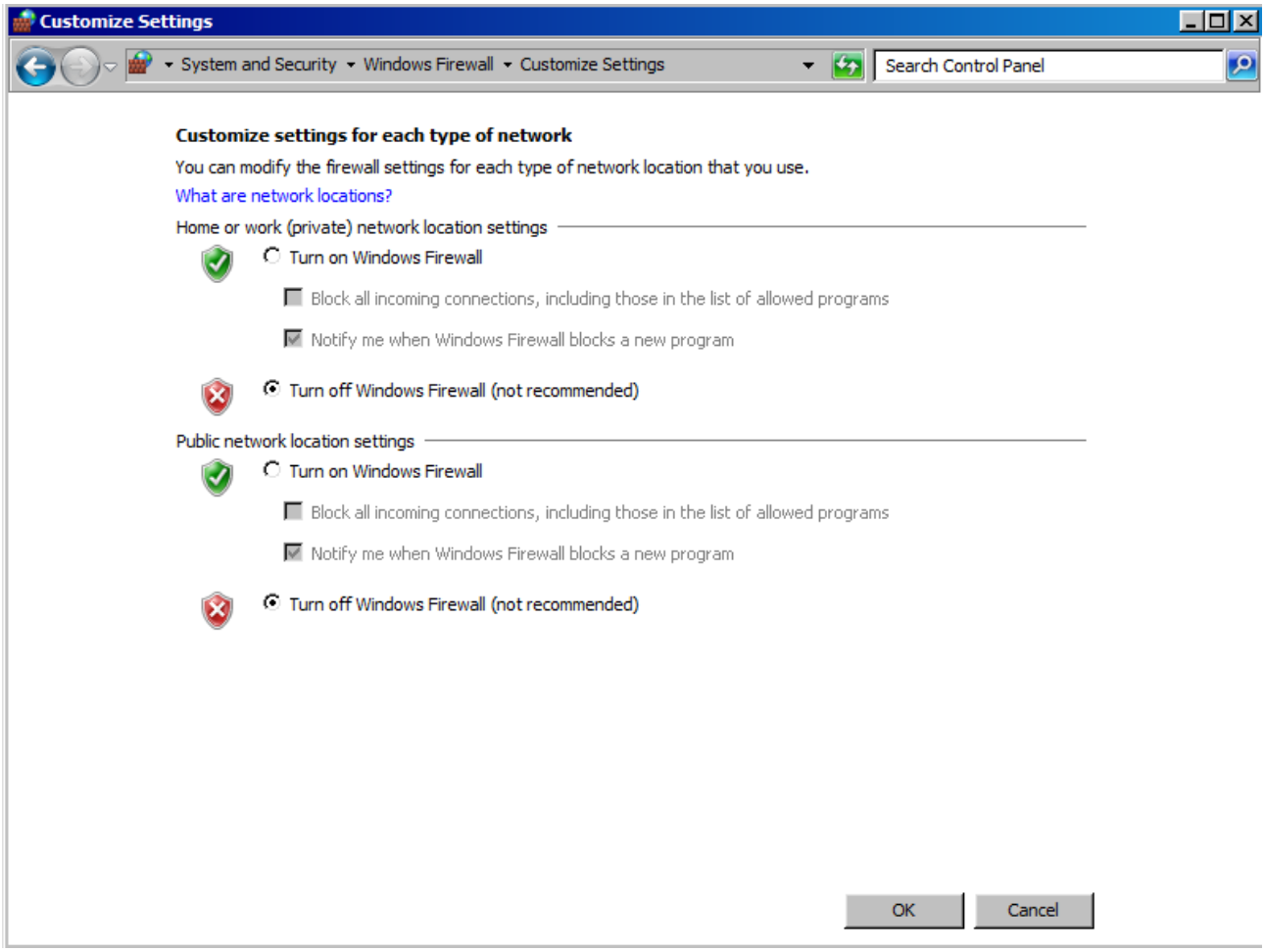
Seuraavaksi asennettiin agenttikone. Agentiksi luotiin Windows 7 virtuaalikone. Windows 7 oli suositeltu käyttöjärjestelmä haittaohjelmien hiekkalaatikoimiseen. Windows 7 virtuaalikone asennettiin käyttäen Oracle VirtualBox-virtualisointialustaa. Windows 7 virtuaalikone asetettiin samaan vboxnet0 virtuaaliverkkoon. Koneelle asetettiin myös IP-osoite samasta verkosta, jossa cuckoo palvelin sijaitsee. Koneelle asetettiin IP-osoite: 192.168.56.101. Virtuaalikoneen nimeksi asetettiin cuckoo1.

Cuckoo1-koneella tuli muuttaa muutamia asetuksia, jotta nämä eivät häiritsisi haittaohjelmien analysointia. Koneelta poistettiin käytöstä Windows Update-toiminnallisuus. Kyseinen toiminnallisuus voi generoida ylimääräistä liikennettä, joka voi häiritä haittaohjelman analysointia. Windows Update otettiin pois käytöstä kuvio 2:en mukaisesti.



Kuvio 2. Windows updaten poistaminen käytöstä.

Cuckoo1-koneelta on myös tärkeää poistaa Windows-palomuuri käytöstä. Windowsin oma palomuuri saattaisi muuten häiritä haittaohjelmien analysointia ja mahdollisesti myös jopa estää koko analysoinnin. Windows-palomuuri otettiin pois käytöstä alla olevan kuvio 3:n mukaisesti.



Kuvio 3. Windows palomuurin poistaminen käytöstä.

Seuraavaksi cuckoo1-koneelle ladattiin tarvittava Python 2.7 -komentotulkki. Cuckoo-hiekkalaatikon agentti ohjelma on kirjoitettu Python 2.7. -ohjelmointikielellä. Tämän jälkeen haettiin cuckoo1-koneelle cuckoo-palvelimelta agent.py -skriptitiedosto. Skriptitiedosto on automaattisesti generoitunut Cuckoo-hiekkalaatikon asennuksen aikana. Agent.py -skriptitiedosto lisättiin cuckoo1-koneelle kansioon C:\Users\%USERNAME%\AppData\Roaming\Microsoft\Windows\StartMenu\Programs\Startup. Lisäyksen ansiosta agent.py -skriptitiedosto suoritetaan automaattisesti, kun kone käynnistetään. Seuraavaksi Virtualboxilla snapshot -ominaisuutta käyttäen koneesta otettiin snapshot eli tilakopio koneen tilasta, joka on palautettavissa myöhemmin. Cuckoo käyttää tätä tilakopiota koneen nopeaan palauttamiseen analysoinnin jälkeen palautuen tilaan, jossa kone on puhtaassa, ei mahdollisesti haittaohjelman saastuttamassa tilassa.

Jotta Cuckoo-palvelin ja cuckoo1-kone voivat kommunikoida täytyi avata palvelimen palomuurista liikenne iptables-komennoilla. Liikenteen avaaminen palvelimen palomuurilta tapahtui alla olevilla komennoilla:

sudo iptables -t nat -A POSTROUTING -o enp0s3 -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

Huomiona, että yllä olevat komennot avaavat myös yhteyden internettiin cuckoo1 koneelle. Tätä ei välttämättä kaikissa tapauksissa haluta tehdä. IP-forwarding-toiminnallisuus täytyy myös laittaa palvelimelle päälle. Tämä tehtiin alla olevilla komennoilla:

echo 1 | sudo tee -a /proc/sys/net/ipv4/ip_forward

sudo sysctl -w net.ipv4.ip_forward=1

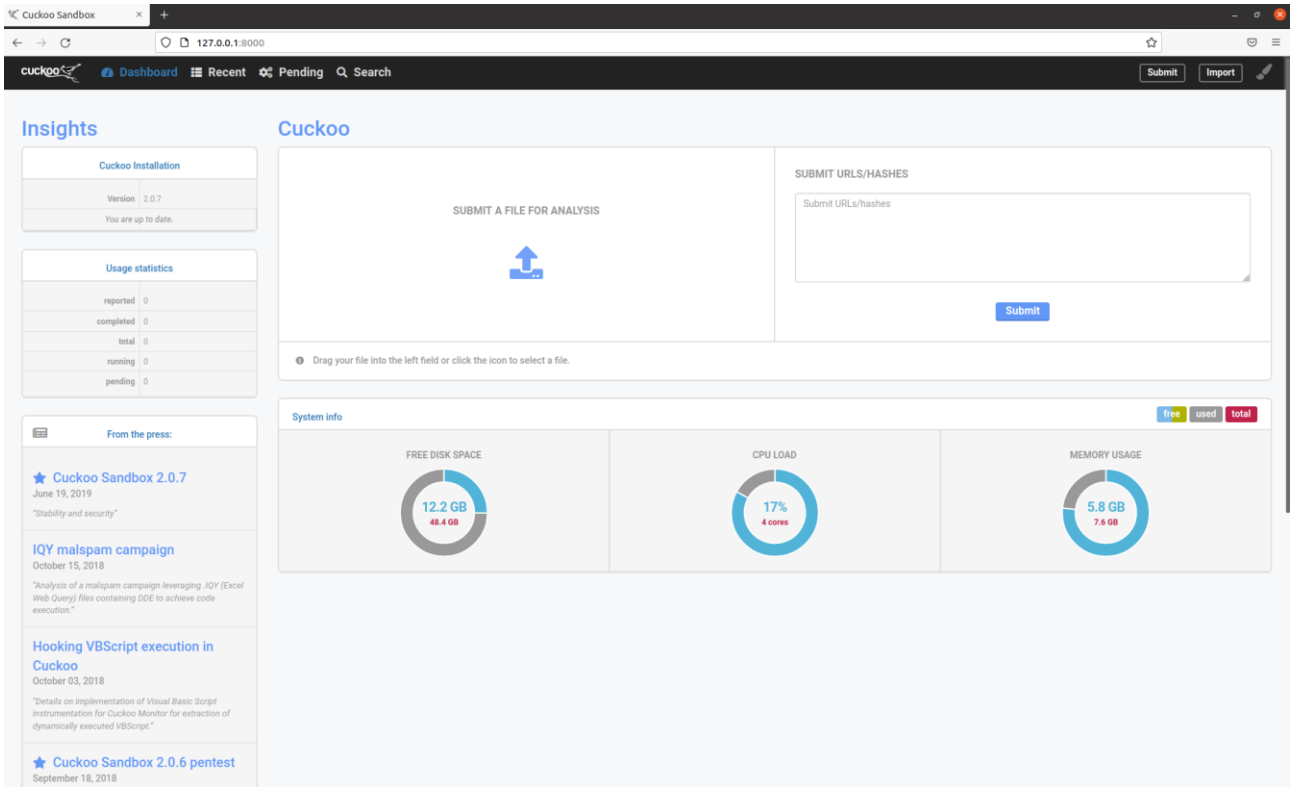
Yllä olevat komennot eivät kuitenkaan jää pysyviksi, vaan nämä komennot täytyy ajaa jokaisen uudelleenkäynnistämisen jälkeen. Tämä on kuitenkin tuotteen testaamiseen riittävä. Komennot saisi pysyviksi esimerkiksi cron apuohjelmalla, joka ajaisi komennot joka uudelleenkäynnistyksen jälkeen. Tuotteen testaamisessa halutaan kuitenkin testata myös Cuckoo-hiekkalaatikkoanalysointia ilman internet-yhteyttä, joten tätä ei tehty.

Oletuksena Cuckoo-hiekkalaatikolla ei ole web-käyttöliittymä päällä. Tämä saadaan päälle muuttamalla kansiossa ” /home/cuckoo/.cuckoo/conf” olevasta reporting.conf konfiguraatiotiedostosta mongodb kohdasta arvo enabled tilaan yes (enabled=yes). Seuraavaksi voidaan käynnistää Cuckoo-hiekkalaatikko. Käynnistys tapahtuu ajamalla kahdella eri terminaalilla komennot:

python2 cuckoo

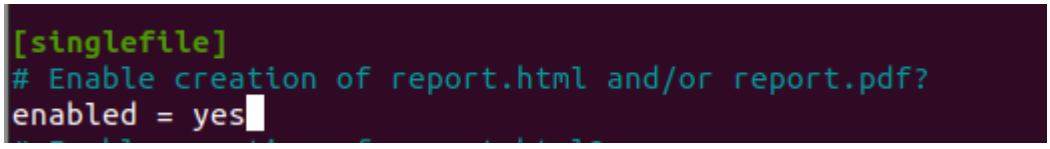
python2 cuckoo web runserver

Cuckoon web-käyttöliittymä on saavutettavissa localhost-osoitteesta 127.0.0.1:8000. Alla olevassa kuviossa 4 nähtävillä Cuckoon web-käyttöliittymän etusivu.



Kuvio 4. Cuckoo web-käyttöliittymän etusivu.

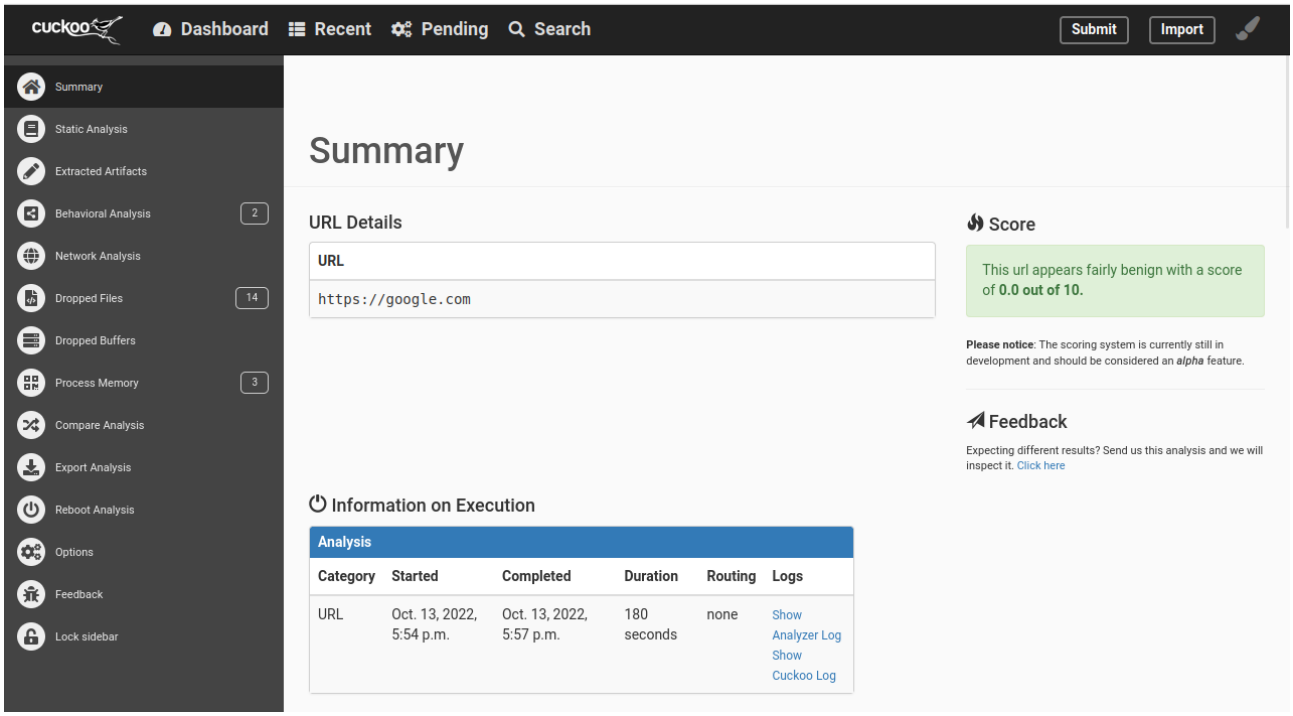
Cuckoon asetuksista täytyi myös erillisesti laittaa päälle analyysien raportointi. Tämä onnistui muuttamalla aiemmin mainitusta reporting.conf tiedostosta kuvion 5 mukaisesti singlefile kohdasta enabled = yes.



Kuvio 5. Analyysien raportoinnin päälle kytkeminen.

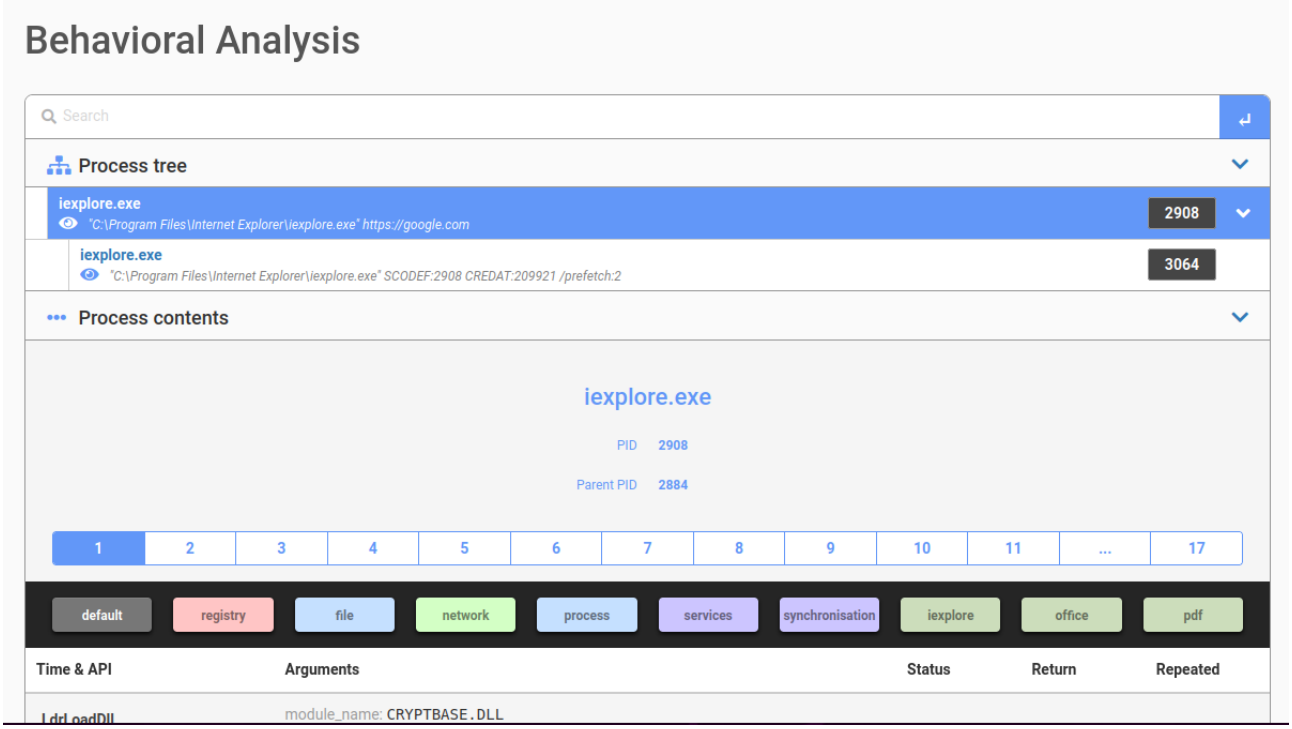
5.2.2 Analysointi

Nyt kun Cuckoo-hiekkalaatikko valmiina analysointia varten. Ensimmäisenä testianalyysinä Cuckoo pyydettiin analysoimaan URL-osoite https://google.com. Tämä onnistui yksinkertaisesti Cuckoo web-käyttöliittymän etusivulta löytyvästä ”Submit URL/HASHES” kohdasta. Kuviossa 4 nähtävissä web-käyttöliittymän etusivu. Analyysin valmistuttua tuotti Cuckoo analyysistä raportin. Raportin etusivu nähtävissä alla olevassa kuviossa 6. Raportin etusivulla on selkeästi esitetty Cuckoon arvio URL-osoitteen haitallisuudesta. Sivun vasemmalla puolella on nähtävissä valikoita, joista näkee lisää tietoa mitä koneella on tapahtunut.



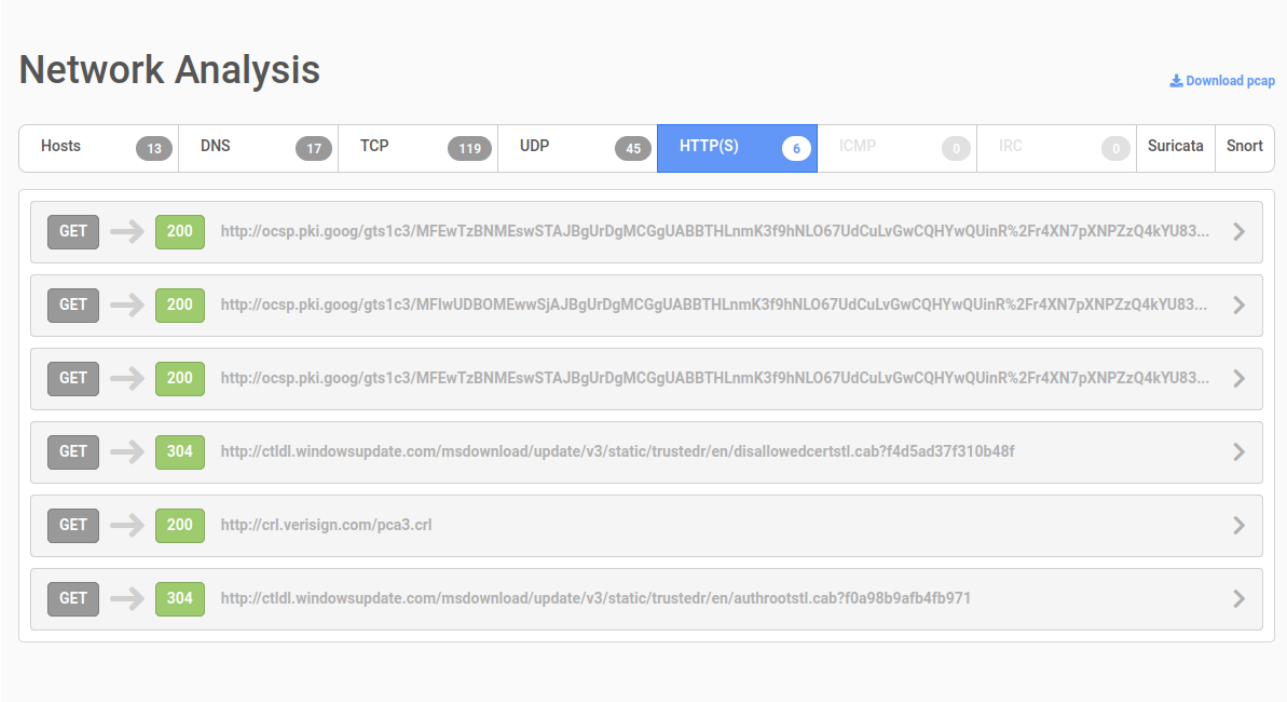
Kuvio 6. Raportin etusivu.

Behavioral Analysis valikosta nähdään jokseenkin hyvin prosessi rakenne ja mitkä moduulit on näihin prosesseihin ladattu. Kuviossa 7 nähtävissä Behavioral Analysis-välilehden sisältö. Nämä tiedot ovat tärkeitä haittaohjelmia analysoidessa ja paljastavat helposti analysoijalle epäilyttävän näköiset tapahtumat.



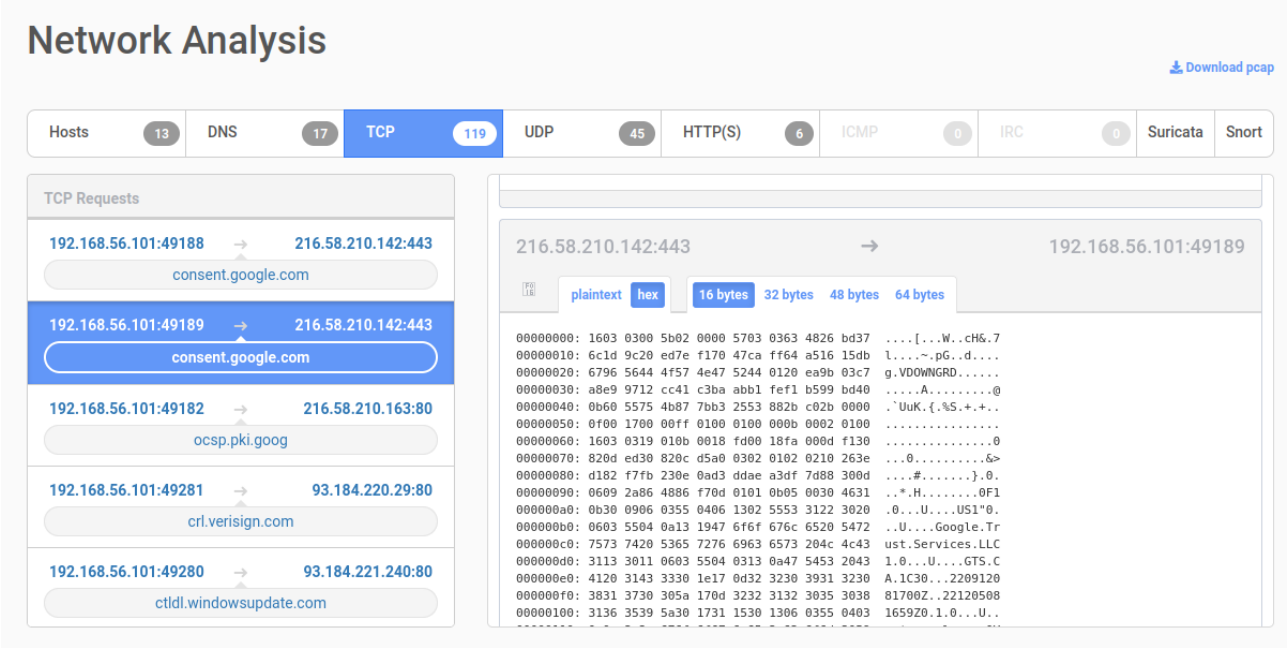
Kuvio 7. Behavioral Analysis -välilehden sisältö.

Network Analysis välilehdellä on nähtävissä verkkoliikenne, jota analyysin ajon aikana on tapahtunut. Verkko-liikenne on nähtävissä todella monipuolisesti. On myös mahdollista ladata packet capture (PCAP) -tiedosto liikenteestä. Kuviossa 8 nähtävissä, miltä http(s)-liikenne näyttää Network Analysis-välilehdellä. Välilehdellä hyvin nähtävissä käytetyt http-metodit ja minkä vastauksen http-kysely on saanut.



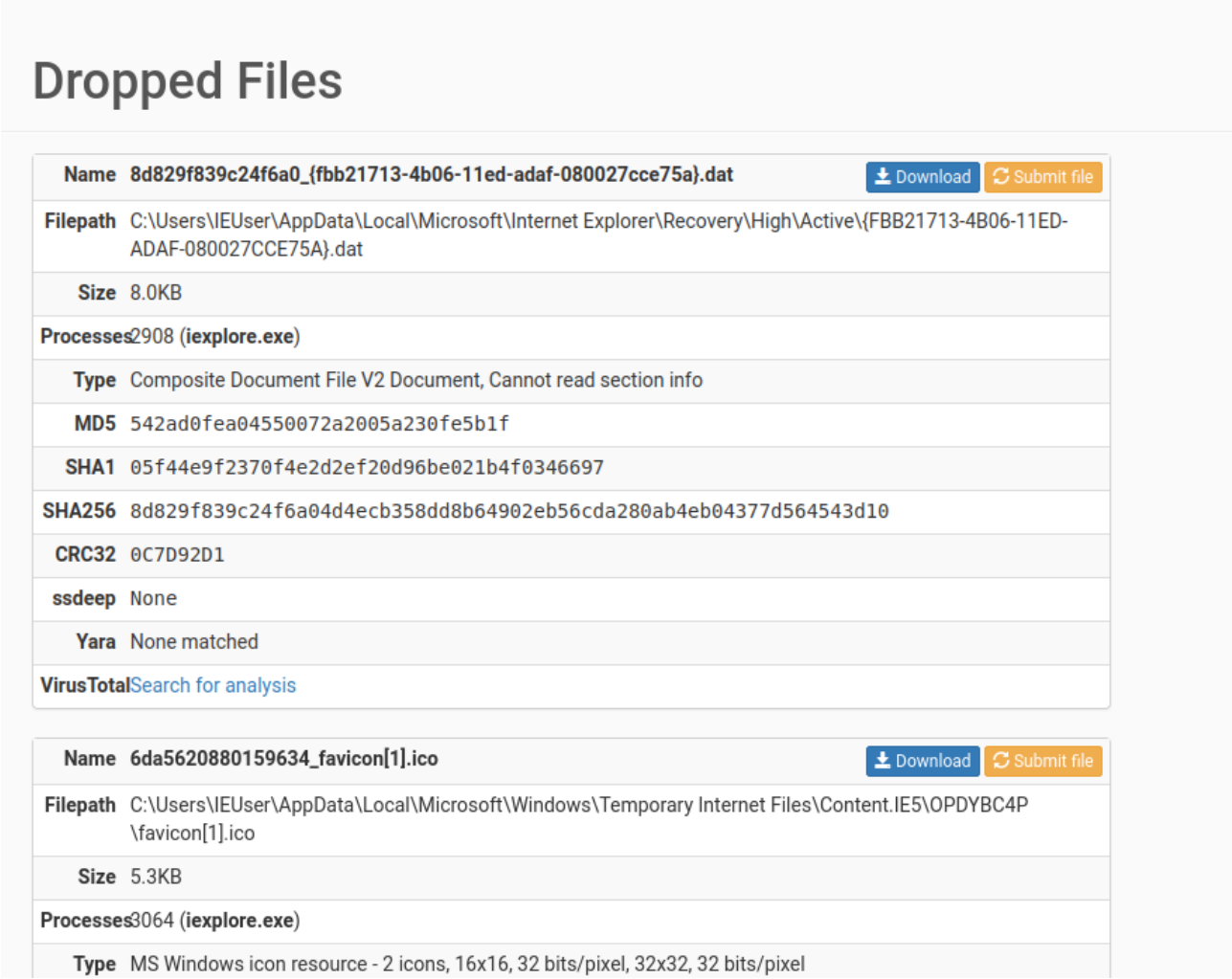
Kuvio 8. Network Analysis välilehden HTTP(S) kohta.

Network Analysis-välilehdellä on myös nähtävissä esimerkiksi TCP-kohdassa TCP-protokollan mukaiset pyynnöt, joita analyysin aikana on suoritettu. Hyvänä puolena tässä näkymässä on, että nähtävissä on itse TCP-paketin sisältö ilman, että PCAP-tiedostoa tarvitsee erikseen ladata ja tutkia ulkoisessa järjestelmässä. Kuviossa 9 on nähtävissä Network Analysis välilehden TCP kohta.



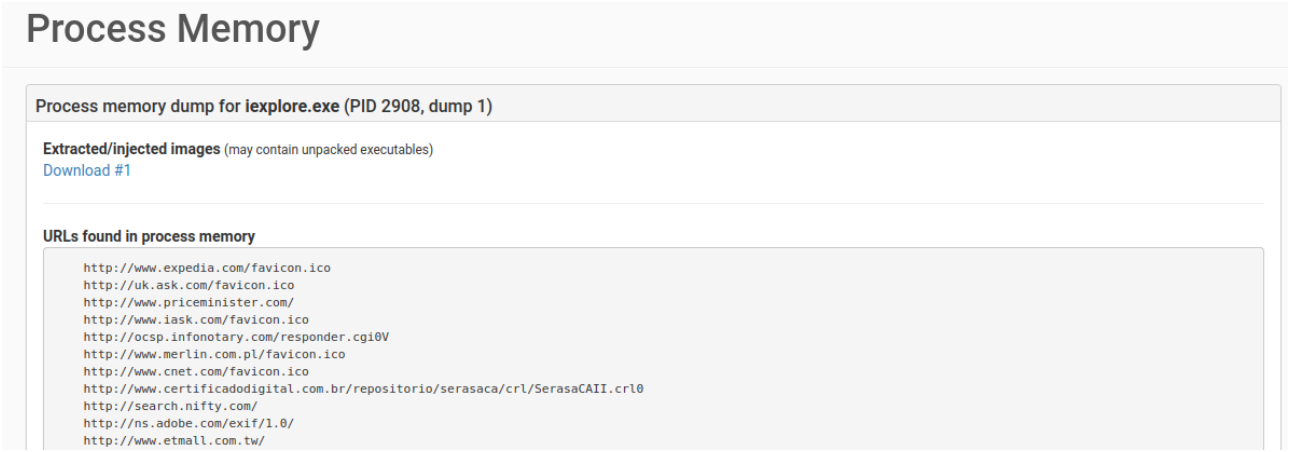
Kuvio 9. Network Analysis välilehden TCP kohta.

Dropped Files välilehdellä on nähtävissä tiedostoja mitä haittaohjelma on luonut tai ladannut ajonsa aikana. Tiedostoista on nähtävissä hyvin tietoa. Välilehdeltä on myös mahdollista ladata tiedosto tai lähettää tiedosto suoraan analysoitavaksi. Molemmat näistä ovat hyödyllisiä ominaisuuksia, sillä monelle haittaohjelmalle on normaalia kirjoittaa tiedostoja levylle suorituksen aikana. Tämän ominaisuuden avulla on mahdollista kätevästi analysoida myös näitä lisätiedostoja. Kuviossa 10 on nähtävissä Dropped Files-välilehti.



Kuvio 10. Dropped files -välilehti.

Process Memory-välilehdellä on mahdollista ladata prosessin muistivedos. Tämä on hyödyllinen ominaisuus, mikäli tarvitsee tehdä tarkempaa staattista analyysiä tapahtuneesta. Välilehdellä on myös nähtävissä mitä URL-osoitteita prosessin muistista on havaittu. Välilehden avulla voisi helposti nähdä haitallisen URL-osoitteen mitä haittaohjelma käyttää. Kuviossa 11 nähtävissä Process Memory -välilehti.



Kuvio 11. Process Memory -välilehti

5.2.3 Ominaisuudet

Cuckoo-hiekkalaatikon konfigurointi tapahtuu muuttamalla .conf päättyviä tiedostoja Cuckoon conf kansiossa. Konfiguraatio voidaan jakaa kuuteen erilaiseen tiedostoon cuckoo.conf, auxiliary.conf, <machinery>.conf, memory.conf, prosessing.conf ja reporting.conf. Näistä tiedostoista cuckoo.conf ja <machinery>.conf on pakko muokata, jotta Cuckoo hiekkalaatikon saa toimimaan. <machinery>.conf nimi määräytyy siitä mitä virtualisointi alustaa käytetään. Eli tässä tapauksessa tiedoston nimi on virtualbox.conf. Cuckoon jokainen konfiguraatio tiedosto on kommentein dokumentoitu ja selitetty mitä nämä muuttuvat. Seuraavaksi selitettynä hieman auki erikseen minkälaisia asetuksia pystytään muuttamaan muokkaamalla eri konfiguraatiotiedostoja.

Cuckoo.conf konfiguraatiotiedostosta löytyy Cuckoon geneeriset konfiguraatio mahdollisuudet. Tällä konfiguraatio tiedostolla voidaan esimerkiksi asettaa mitä virtualisointimoduulia Cuckoo käyttää. On myös mahdollista konfiguroida Cuckoo käyttämään eri palvelinta analyysin tuloksia varten. Huomiona tässä, että agenttikoneella pitää olla yhteys tähän palvelimeen mihin tulokset lähetetään. Oletuksena tulos palvelimeksi on asetettu Cuckoo palvelimen oma IP-osoite. Tiedostossa on myös muita analyysiin liittyviä asetuksia. Voidaan muokata montako analyysiä voi olla samaan aikaan ajossa. Voidaan myös asettaa analysoitavalla tiedostolle maksimi tiedostokoko. Tiedostokoko on oletuksena 128MB. Kuten kuviossa 9 nähtiin Network Analysis -välilehden TCP-kohdassa, oli nähtävissä verkko paketin sisältö. Tämän ominaisuuden voi myös kytkeä pois päältä, mikäli sitä ei haluta cuckoo.conf tiedostossa. Tiedostossa on myös mahdollista määrittää mitä tietokantaa halutaan käyttää.

Auxiliary.conf konfiguraatio tiedostosta löytyy haittaohjelmia analysoidessa ajettavien skriptien asetukset. Tiedostossa voidaan asettaa päälle tcpdump-ohjelman tuoma verkkoliikenteen vakoilu. Tämä on oletuksena päällä. Mikäli halutaan tallentaa verkkoliikennettä analyysin aikana muusta kohdasta, voidaan asetuksissa asettaa päälle man-in-the-middle (MITM) välipalvelin. MITM-toimintoa varten Cuckoo käyttää avoimen lähdekoodin mitmdump-ohjelmaa. Ohjelma tarjoaa samoja ominaisuuksia kuin tcpdump, mutta mahdollistaa lisänä myös liikenteen muokkaamisen ja salatun TLS-protokollan mukaisen liikenteen salauksen purun. TLS-liikenteen salauksen purku vaatisi myös omien TLS-sertifikaattien luonnin. Jotkin haittaohjelmat saattavat huomata omien sertifikaattien käytön ja estää haittakoodin suorituksen.

<machinery>.conf konfiguraatio tiedostosta löytyy asetukset virtualisointiohjelmalle. Tiedoston nimi määräytyy sen mukaan mitä virtualisointi alustaa käytetään Cuckoo-sovelluksen suoritukseen. Konfiguraatio tiedosto, jota VirtualBox virtualisointialustan kanssa tulee muokata, on virtualbox.conf. Virtualbox.conf -konfiguraatiotiedostosta löytyy muutamia asetuksia, joita täytyy muokata, jotta Cuckoon saa toimimaan. Virtuaaliverkon nimi täytyy muuttaa oikeaksi, Agenttikoneen tai koneiden nimet täytyy olla oikein ja agenttikoneen IP-osoite täytyy olla oikea.

Memory.conf konfiguraatio tiedostosta löytyy valinnaisen Volatility ohjelman asetukset. Volatility on avoimen lähde koodin muistiforensiikan- ja haittaohjelmien analysointiohjelma. Cuckoo-hiekkalaatikko tukee Volatilityä ja käyttää tätä muistivedoksien analysointiin. Memory.conf tiedostossa määritetään mitä moduuleja Volatilitystä halutaan käyttää. Volatilityn käyttö vaatii volatility kohdan käyttöön ottamisen prosessing.conf tiedostosta ja memory_dump kohdan käyttöönoton cuckoo.conf tiedostossa. Memory.conf tiedostossa voidaan, konfiguroida esimerkiksi poistetaanko muistivedos näiden analysoinnin jälkeen. Muistivedosten poistaminen säästää paljon massamuistia, sillä muistivedos tiedostot voivat olla kooltaan suuria.

Processing.conf -konfiguraatiotiedostossa voidaan konfiguroida Cuckoon prosessointimoduuleja. Prosessointi moduulit sijaitsevat cuckoo.processing moduulissa. Nämä moduulit määrittelevät kuinka Cuckoo käsittelee raa'an datan, joka kerätään analyysin aikana. Tiedostossa olisi mahdollista asettaa päälle ominaisuuksista, joiden avulla myös Android-käyttöjärjestelmän tiedostoja voisi myös analysoida. Tiedostossa on myös muita mielenkiintoisia asetuksia, joita ei oletuksena ole päällä. Olisi mahdollista asettaa päälle koko virtuaalikoneen muistivedos. Tätä muistivedosta voisi

sitten hyödyntää Volatily-sovelluksessa, jonka voisi laittaa päälle memory.conf tiedostossa. Cuckoo tukee myös Internet-pohjaisen Virustotal-palvelun käyttöä. Asetuksiin on mahdollista asettaa oma Virustotal API-avain integraatiota varten. Cuckoo voi integraatiota käyttäen rikastaa tietoja analysoitavista tiedostoista Virustotalin haittaohjelmatietokantaa hyödyntäen. On myös mahdollista asettaa päälle asetus, joka automaattisesti lähettää analysoitavan tiedoston Virustotaliin arvioitavaksi.

Reporting.conf -konfiguraatiotiedostossa on Cuckoon raportoinnin asetukset. Jostain syystä raportointi ei ollut oletuksena päällä, kun Cuckoo oli asennettu. Tämä täytyi erikseen käydä asettamassa päälle kyseisestä tiedostosta. MongoDB täytyi myös asettaa päälle tästä tiedostosta, jotta Cuckoon web-käyttöliittymä toimisi. Tiedostosta löytyy mahdollisuus kytkeä päälle monia eri lisätoimintoja Cuckoo-hiekkalaatikkoon. Mahdollisia lisätoimintoja ovat MISP, Elasticsearch, Moloch ja Mattermost integraatiot. MISP on avoimen lähdekoodin ohjelma, jolla voidaan kerätä, tallentaa ja jakaa haittaohjelmien ja muutenkin turvallisuuden indikaattoreita. Elasticsearch on avoimen lähdekoodin analytiikka ja hakuohjelma. Elasticsearchin avulla voisi Cuckoossa hakea tietoa ajetuista analyyseista. Moloch on avoimen lähdekoodin verkkoliikenteen keräys- ja indeksointi ohjelma. Ohjelmaan voidaan kerätä analyyseista syntyviä PCAP-tiedostoja. Näitä tiedostoja voidaan Molochin avulla indeksoida ja hakea sekä analysoida tehokkaasti. Mattermost on avoimen lähdekoodin kommunikointi ohjelma. Cuckoon voi asettaa ilmoittamaan Mattermost-tekstikanavalle tietoa analyyseista. Lisäksi on mahdollista asettaa Cuckoo ilmoittamaan analyyseista mille tahansa web-palvelimelle käyttäen HTTP POST -metodia.

Cuckoo-sovelluksessa on sisäänrakennettu representational state transfer (REST) API-rajapinta. API-rajapinta käyttämällä on mahdollista aloittaa analyyseja, saada tietoa analyyseista ja tietoa Cuckoo-palvelimen terveydestä suoraan API-rajapintakyselyä käyttäen. APIn käyttö toimii lähettämällä HTTP GET ja POST erilliselle Cuckoo API-palvelimelle. Cuckoon voisi halutessaan integroida muihin palveluihin käyttäen API-rajapintaa.

Cuckoo tarjoaa ladattavaksi Cuckoo yhteisön keräämiä haittaohjelma allekirjoituksia. Yhteisön allekirjoitukset voi ladata komennolla ”cuckoo community”. Yhteisö GitHub sivua ei kuitenkaan ole päivitetty muutamaan vuoteen, joten allekirjoitukset ovat hyvin todennäköisesti vanhentuneita. Esimerkiksi YARA-tunnistus-sääntöjä on mahdollista kuitenkin saada muista avoimista lähteistä.

Cuckoo-sovellukseen on mahdollista konfiguroida eri reititys vaihtoehtoja käytettäväksi analyysissä. Reititys-vaihtoehdot ovat pudotusreititys, Internet-reititys, InetSim-reititys, Tor-reititys ja VPN-reititys. Pudotusreitityksessä Cuckoo pudottaa kaiken liikenteen. Internet-reitityksessä yksinkertaisesti analysoinnissa päästetään liikenne Internettiin. InetSim-reitityksessä käytetään erikseen konfiguroitavaa simuloitua Internettiä. InetSim tuottaa simuloituja verkkopalveluita, joiden avulla voidaan huijata haittaohjelmaa uskomaan, että sillä on yhteys Internettiin. Tor-reitityksessä liikenne yksinkertaisesti reititetään Tor-verkon läpi. VPN-reitityksessä liikenne reititetään määritetyn VPN-palvelun kautta.

Cuckoo-hiekkalaatikkoon ei ole tullut päivitystä kirjoitushetkellä moneen vuoteen, joten tuotteen päivitystä ei voitu testata. Dokumentaation mukaan Cuckoon päivittäminen ei ole hankalaa. Cuckoossa on automaattinen päivitys toiminto, jonka pitäisi päivittää Cuckoo automaattisesti.

5.3 CAPE-hiekkalaatikko

CAPE hiekkalaatikko on Cuckoo-hiekkalaatikosta johdettu hiekkalaatikkotuote. CAPE:n alkuperäinen tavoite oli lisätä automatisoitu haittaohjelman purkaminen ja asetusten purkaminen. CAPE onkin akronyymi sanoista ”Config And Payload Extraction”. Automaattinen purkaminen mahdollisti haittaohjelmien luokittelun esimerkiksi YARA-tunnistus sääntöjen perusteella. Aiemmin luokittelu perustui pitkälti prosessien muistivedosten analysointiin. Tämän pystyivät haittaohjelmien tekijät estämään. CAPE on alkuperäisestä tavoitteestaan kehittynyt ja sillä on nykyään hyvät työkalut välttää haittaohjelmien analyysin välttelytekniikat. (What is CAPE? n.d.)

5.3.1 Asennus

CAPE hiekkalaatikko asennettiin kappaleessa 5.1 kuvattuun ympäristöön. CAPE hiekkalaatikko asentaminen oli tehty helpoksi, sillä hiekkalaatikon tuottajat ovat tehneet asennusta varten asennus skriptin. Skripti asentaa kaikki tarvittavat kirjastot ja asentaa alla olevat palvelut.

cape.service

cape-processor.service

cape-web.service

cape-rooter.service

Ennen asennusskriptin ajamista on kuitenkin asennettava Python 3 -komentotulkki. CAPE-hiekkalaatikko on toteutettu käyttäen Python 3 -ohjelmointikieltä. Python 3:n lisäksi tarvittiin pythonin paketinhallintaohjelma pip. Python 3 ja pip asennettiin alla olevilla komennoilla.

sudo apt install python3

sudo apt install python3-pip

Python 3:n asentamisen jälkeen voitiin ladata CAPE-hiekkalaatikon asennusskripti. Asennusskripti ladattiin CAPE-hiekkalaatikon GitHub-sivulta. Asennusskripti ladattiin alla olevalla komennolla.

wget https://raw.githubusercontent.com/kevoreilly/CAPEv2/master/installer/cape2.sh

Ennen asennusskriptin ajoa täytyi skriptille antaa tarvittavat oikeudet. Oikeuksien antamisen jälkeen asennusskripti voitiin ajaa. Asennusskriptillä on kaksi eri versiota, base ja full. Dokumentaation mukaan base ja full versioilla ei ole erona kuin virt-manager. Virt-manager on virtuaalikoneiden hallintaohjelma. Tätä ohjelmaa ei tähän toteutukseen tarvittu, joten asennus skripti on ajettu base-versiolla. Alla asennus skriptin oikeuksien asetus ja asennus skriptin ajo komennot.

chmod a+x cape2.sh

sudo ./cape2.sh base cape

Kun asennusskripti oli ajettu läpi, tarvitsi enään CAPE-hiekkalaatikon asennusta varten asentaa agenttikone, jolla haittaohjelmia analysoidaan. Agentti virtualisoitiin samalla tavalla kuin Cuckoo-hiekkalaatikon asennuksessa VirtualBox-virtualisointi ohjelmalla. Ohjelma asennettiin alla olevalla komennolla.

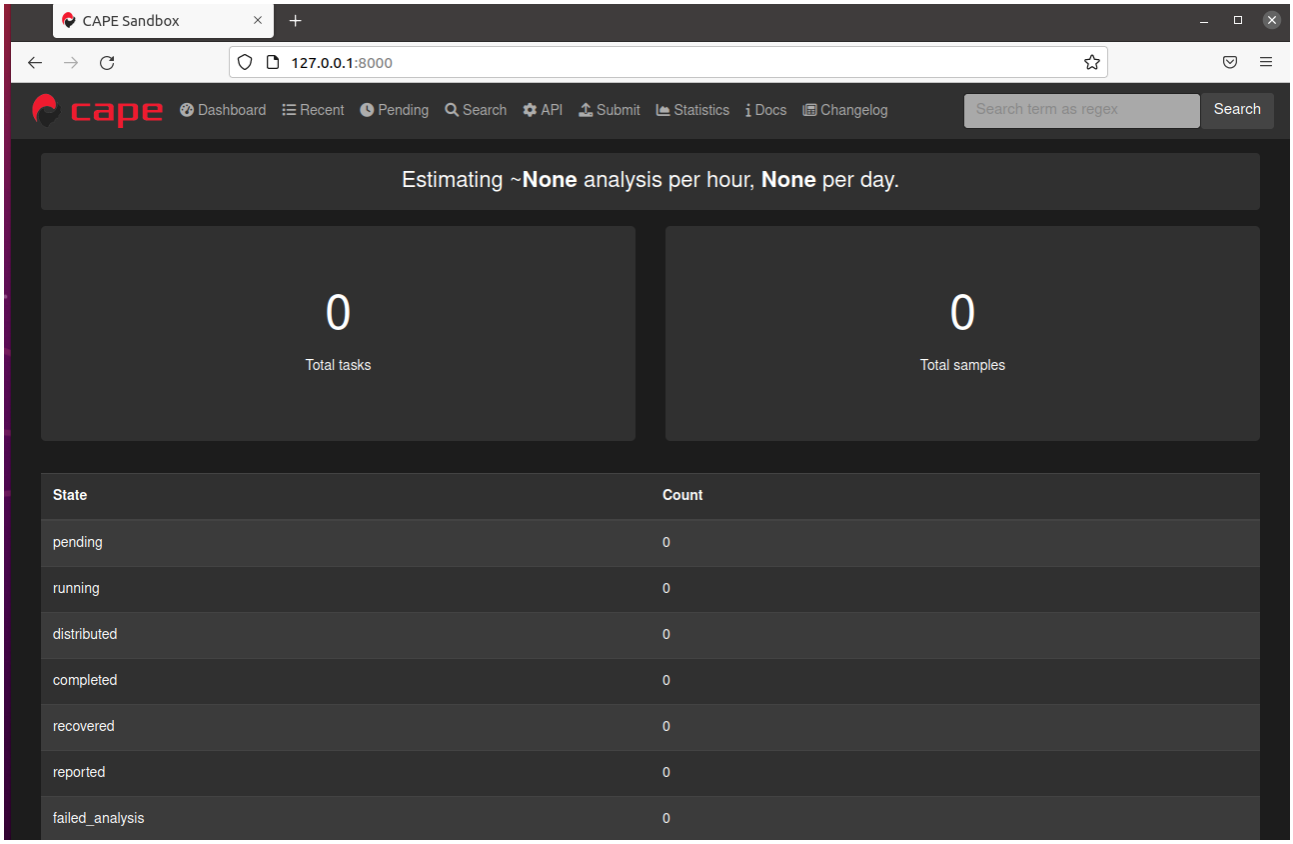
sudo apt install virtualbox

Agenttikoneena käytettiin CAPE-hiekkalaatikolla myös Windows 7 konetta. Virtuaalikoneelle annettiin nimeksi cape1. Cape1-koneelle täytyi asentaa Python 3.6, jotta sillä voisi ajaa CAPE:n agenttiohjelmaa. Python version täytyi olla 32-bit versio, jotta agentti toimii. Cape1-koneelle tehtiin

myös samat muutokset kuin Cuckoo hiekkalaatikossa käytettyyn Cuckoo1 koneelle. Koneelta kytkettiin pois päältä Windows Update -toiminnallisuus sekä Windows palomuuri. Koneelle asetettiin manuaalisesti IP-osoite aiemmin luodusta virtuaaliverkosta. Näiden muutoksien lisäksi koneelle asennettiin Python kirjasto pillow. Pillow mahdollistaa näyttökaappauksien oton analysoidessa. Pillow asennettiin alla olevalla komennolla.

pip install pillow

Cape1-koneen asennuksen jälkeen CAPE-hiekkalaatikko oli valmis analysoimista varten. Kuviossa 12 CAPE-hiekkalaatikon etusivu.

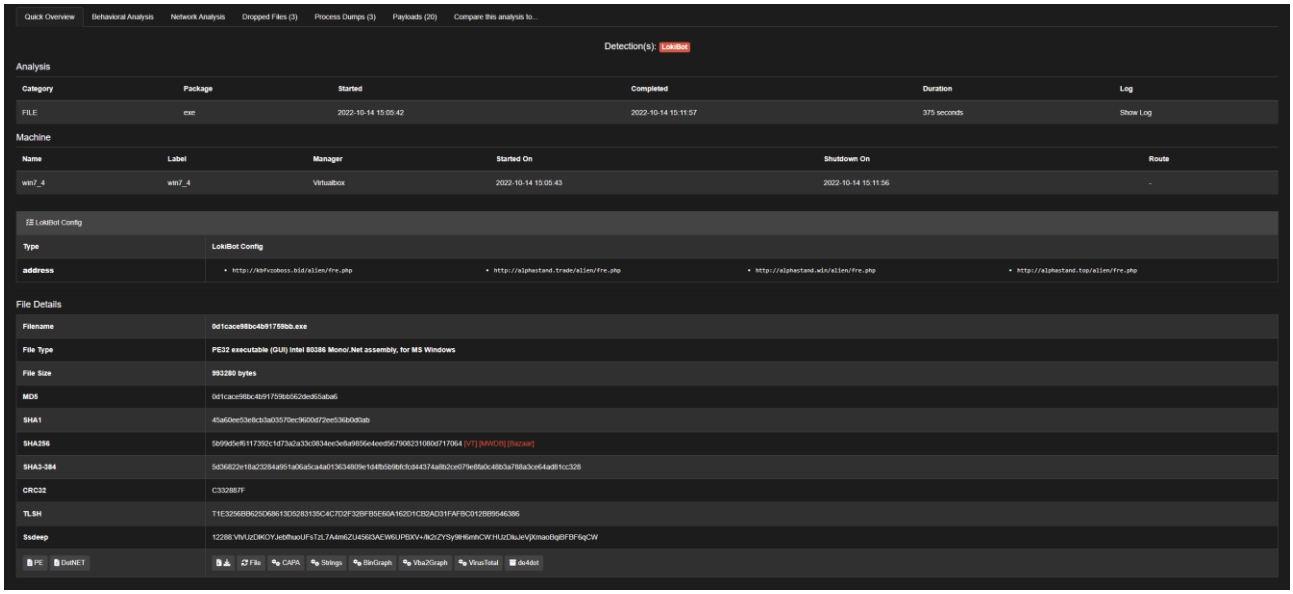


Kuvio 12. CAPE-hiekkalaatikon etusivu.

5.3.2 Analysointi

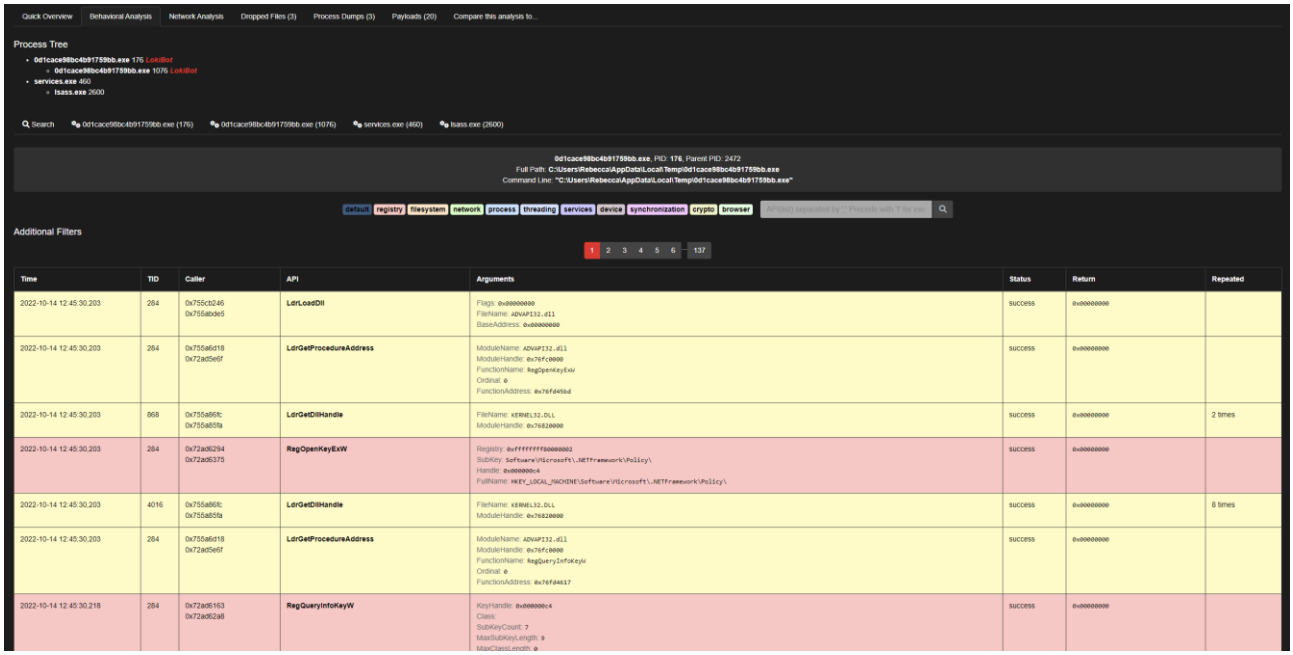
Testi analyysia varten hankittiin MalwareBazaar-haittaohjelma näyte sivustolta geneerinen Lokibot haittaohjelma. Tavoitteena oli testata miten CAPE hiekkalaatikon YARA-tunnistus-säännöt tunnistavat helposti tunnistettavan haittaohjelman. Kuviossa 13 CAPE-hiekkalaatikon analysointi raportin

etusivu. Etusivulta on nähtävissä nopeasti tietoa analyysin tuloksesta. Sivun yläpuolella löytyy valikot, joista löytyy tarkemmin tietoa analysoidusta ohjelmasta.



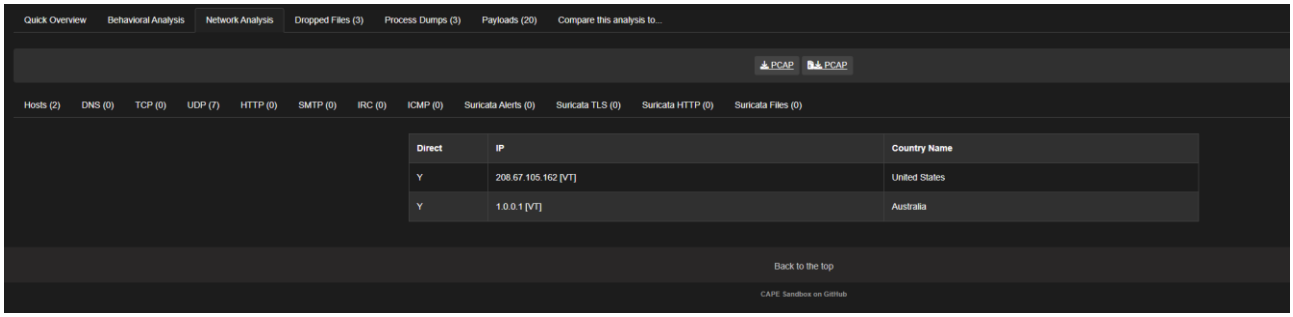
Kuvio 13. CAPE-hiekkalaatikon analysointi raportin etusivu.

Ensimmäisessä valikossa on nähtävissä CAPE hiekkalaatikon tekemä käyttäytymisanalyysi. Sivulla on nähtävissä prosessipuu, jossa on nähtävissä mitä prosesseja haittaohjelma on käyttänyt. Tästä alempana on nähtävissä kaikki toiminnot mitä haittaohjelma on tehnyt. Kuviossa 14 CAPE-hiekkalaatikon käyttäytymisanalyysivalikko.



Kuvio 14. CAPE-hiekkalaatikko käyttäytymisanalyysi valikko.

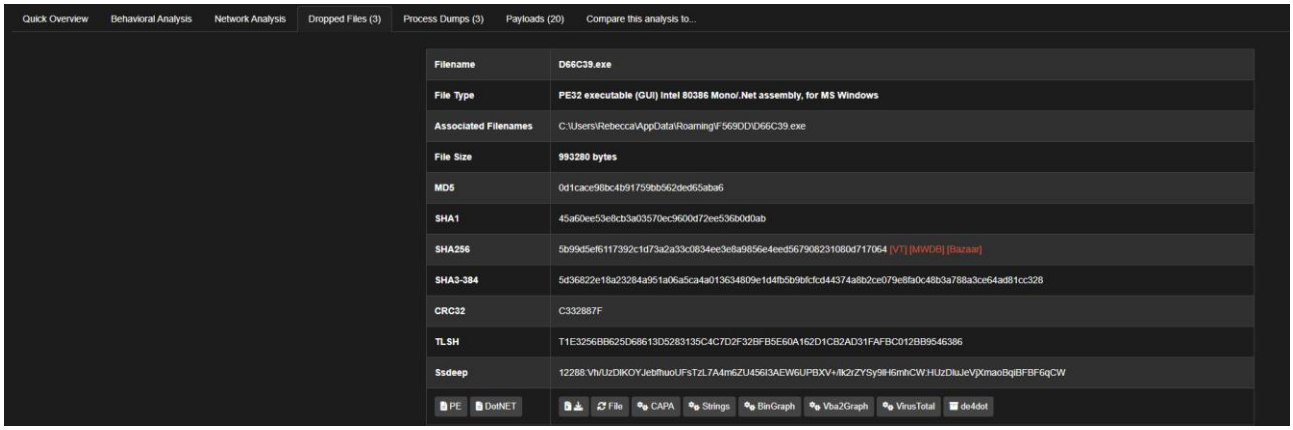
Toisena valikkona oli verkkoanalyysi valikko. Valikon takaa on nähtävissä mihin verkkoosoitteisiin haittaohjelma on kommunikoinut. Valikosta on mahdollista ladata PCAP-tiedosto tarkempaa tarkastelua varten. Valikossa ei kuitenkaan ole ilman PCAP-tiedoston lataamista mahdollista nähdä tarkemmin verkko pakettien sisältöä. Kuviossa 15 CAPE-hiekkalaatikon verkko analyysi valikko.



Kuvio 15. CAPE-hiekkalaatikko verkko analyysi valikko.

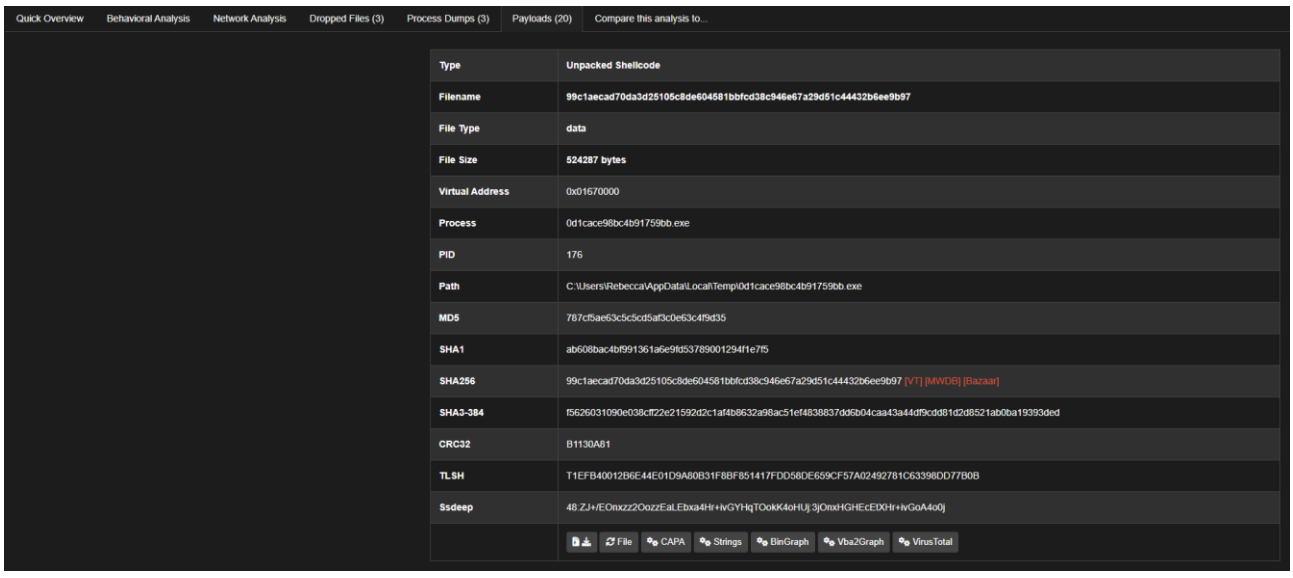
Seuraavana valikkona oli pudotetut tiedostot valikko. Valikon takaa löytyy tiedostot, joita haittaohjelma on kirjoittanut levylle sitä ajettaessa. Tiedoston voi tästä valikosta lähettää analysoitavaksi. Tiedoston voi myös tästä valikosta ladata salasanasuojatussa paketissa. Samat ominaisuudet löytyvät seuraavasta valikosta ”process dump”. Valikossa on nähtävissä samalla tavalla kuin pudotetut

tiedosto valikossa prosessien muistivedokset. Nämä on mahdollista ladata tarkempaa analyysiä varten. Koska CAPE käyttää Volatilityä automaattisesti on muistivedokset mahdollista lähettää myös analysoitavaksi CAPE:lla. Kuviossa 16 CAPE-hiekkalaatikon pudotetut tiedostot valikko.



Kuvio 16. CAPE-hiekkalaatikko pudotetut tiedostot valikko.

Viimeisessä valikossa on nähtävissä haittaohjelman hyötykuormat. Valikossa on nähtävissä kaikki haittaohjelman osat. Valikossa on mahdollista lähettää nämä analysoitavaksi tai ladata kuten pudotetut tiedostot valikossa. Tämä ominaisuus oli toinen mainostetuista CAPE hiekkalaatikon hyvistä puolista. Kuviossa 17 CAPE-hiekkalaatikon payload-valikko.



Kuvio 17. CAPE-hiekkalaatikko payload-valikko.

5.3.3 Ominaisuudet

CAPE-hiekkalaatikon konfigurointi onnistuu muuttamalla konfigurointitiedostoja, jotka sijaitsevat conf-kansiossa. Pääkonfigurointitiedostoja on kuusi cuckoo.conf, auxiliary.conf, <machinery>.conf, memory.conf, processing.conf ja reporting.conf. Koska CAPE on alun perin tehty Cuckoon pohjalle ovat konfigurointi tiedostot saman nimisiä ja konfiguroivat samoja asioita mitä Cuckoo-hiekkalaatikon ominaisuudet osiossa käytiin läpi. Joten seuraavaksi käydään läpi mitä eri ominaisuuksia CAPE-hiekkalaatikossa on, joita ei Cuckoo-hiekkalaatikossa ole.

Cuckoo.conf tiedostossa löytyy samat konfiguraatiomahdollisuudet kuin Cuckoo-hiekkalaatikon cuckoo.conf tiedostossa. Huomiona, että myös CAPEssa cuckoo.conf tiedostoon muuttama machinery-arvo päättää mitä <machinery>.conf tiedostoa käytetään. Koska CAPE on toteutettu tässä tapauksessa VirtualBoxin kanssa käytössä on virtualbox.conf.

Auxiliary.conf tiedostossa sen sijaan on joitakin ominaisuuksia mitä ei Cuckoon versiosta löytynyt. Tiedostossa on mahdollista muuttaa mitä moduuleita analysoiva agenttikone käyttää. Esimerkiksi on mahdollista asettaa näytönkaappaus koneelta analyysin aikana päälle ja tlsdump. CAPEn auxiliary.conf tiedostossa kuitenkaan ei ole man-in-the-middle (MITM) asetusta, mikä Cuckoon tiedostossa oli.

Virtualbox.conf tiedostosta löytyy täysin samat asetukset kuin Cuckoon virtualbox.conf tiedostossa.

Memory.conf tiedostossa on Volatility muisti-forensiikka-ohjelman asetukset. Volatility on valinnainen osa. CAPE-hiekkalaatikossa kuitenkin on enemmän ominaisuuksia, jotka tukevat Volatilityn käyttöä. Konfiguraatiotiedostot kuitenkin ovat molemmissa CAPE ja Cuckoo samat.

Processing.conf tiedostossa löytyy asetukset CAPE-hiekkalaatikon prosessointi moduuleista, jotka määrittävät miten analysoitua dataa pureskellaan. Tässä tiedostossa on selvästi eniten eroa Cuckoo-hiekkalaatikkoon. Tämä ei tullut yllätyksenä, sillä CAPE-hiekkalaatikosta oli mainostettu, että juuri nämä ominaisuudet olisivat paremmat kuin Cuckoossa. Konfiguraatio tiedostossa on monia ominaisuuksia, joita ei Cuckoon tiedostossa ollut. Mielenkiintoisimpina MaliciousMacroBot, jonka tarkoitus on tunnistaa haitallisia Microsoft Office dokumentteja. Malduck, joka on Puolan

valtion kyberturvallisuuskeskuksen tekemä avoimen lähdekoodin ohjelma. Ohjelma sisältää erilaisia skriptejä, joiden tarkoitus on nopeuttaa ja parantaa haittaohjelman analysointia. CAPA on avoimenlähdekoodin ohjelma, joka analysoi ohjelman ja yrittää kertoa mitä analysoitu ohjelma voi tehdä. Kaiken kaikkiaan CAPElla on monipuoliset analysointi mahdollisuudet.

Reporting.conf tiedostosta löytyy raportointiin liittyviä asetuksia. Raportoinnissa on joitakin eroja Cuckoohon verrattuna. CAPEn tiedostossa on pääosin samat tärkeimmät mitä Cuckoon tiedostossa, mutta tiedostossa on myös muita, joita ei Cuckoon tiedostossa ole. Esimerkiksi tiedostosta löytyy mahdollisuus lisätä MITRE integraatio. Yksi ominaisuus mitä ei CAPEn reporting.conf tiedostossa ollut on Mattermost integraatio. Tämä ominaisuus löytyy Cuckoo-hiekkalaatikosta.

CAPE-hiekkalaatikossa on samat ominaisuudet konfiguraatio tiedostojen ulkopuolella. Näitä ominaisuuksia käyty läpi Cuckoo-hiekkalaatikon ominaisuudet kohdassa. Mainitsemisen arvoisena kuitenkin CAPE-hiekkalaatikon yhteisö ominaisuus, josta on mahdollista ladata CAPE-yhteisön tekemiä sääntöjä ja allekirjoituksia suoraan CAPE:n sisään rakennettua ominaisuutta käyttämällä. CAPE:n yhteisö GitHub sivua päivitetään aktiivisesti toisin kuin Cuckoon, joten tuotteeseen saa helposti ajankohtaisia sääntöjä, joilla tunnistaa haittaohjelmia. Molemmissa tuotteissa olisi myös mahdollista tehdä python ohjelmointikielellä itse lisää ominaisuuksia.

6 Tulokset

CAPE-hiekkalaatikko on luotu Cuckoo-hiekkalaatikon pohjalta, joten molemmissa tuotteissa on havaittavissa paljon samaa. Molemmat kuitenkin ovat omalla tavallaan erikoistuneet tiettyihin asioihin ja täten tekevät joitain asioita paremmin kuin toinen.

6.1 Asennus

Asennusprosessi oli Cuckoo-hiekkalaatikossa monimutkainen ja vaati perehtymistä. CAPE-hiekkalaatikon asennusta varten oli tehty asennusskripti, joka teki kaikki vaiheet automaattisesti, jotka täytyi Cuckoo-hiekkalaatikon asennuksessa tehdä käsin. CAPE-hiekkalaatikon konfiguraatiotiedostoja ei myöskään tarvinnut muuttaa sillä asetukset olivat oikein heti asennettua. Toisin kuin Cuckoossa jossa täytyi esimerkiksi web-käyttöliittymä ja raportointi laittaa erikseen päälle. Kaiken kaikkiaan asennus prosessi oli tehty CAPE-hiekkalaatikkoon paljon helpommaksi.

6.2 Analysointi

Analysoinnin suhteen molemmissa tuotteissa on hyviä ja huonoja puolia. Molemmissa tuotteissa löytyvät tulosten raportoinnin osalta suurin piirtein samat valikot. Huomiona kuitenkin, että Cuckoo-hiekkalaatikon web-käyttöliittymä on selvästi käytännöllisempi ja analyysin tuloksia on selvempi tutkia. CAPE-hiekkalaatikon käyttöliittymästä kuitenkin löytyvät melkein kaikki samat tiedot, mutta hieman epäselvemmin. Ainoa tieto mitä CAPE-hiekkalaatikosta ei löydy ja Cuckoo-hiekkalaatikosta löytyy, on verkkoliikenteen paketin sisältö ilman, että täytyy ladata itse koko PCAP-tiedostoa ja tulkita sitä muussa sovelluksessa. Kuviossa 9 on kuvattu miltä tämä ominaisuus näyttää. CAPE hiekkalaatikossa löytyvä valikko mitä ei Cuckoo-hiekkalaatikosta on ”payload” valikko. Valikon sisältö on mikä eniten erottaa CAPE-hiekkalaatikon Cuckoosta ja osa syytä miksi CAPE-hiekkalaatikko on tehty. Kaiken kaikkiaan molemmista tuotteista löytyy tarvittavat tiedot haittaohjelman analysointia varten.

6.3 Ominaisuudet

Molemmissa tuotteissa on pitkälti samat ominaisuudet. Cuckoo-hiekkalaatikossa ainoana mainitsemisen arvoisena ominaisuutena mitä ei CAPE-hiekkalaatikosta löydy on Mattermost pikaviestintäsovelluksen integraatio. Tämä ei kuitenkaan ole mitenkään merkittävä ominaisuus, koska se ei lisää analysointikykyä. CAPE-hiekkalaatikossa on pitkälti samat ominaisuudet kuin Cuckoossa, lukuun ottamatta processing.conf konfiguraatio tiedostoa. CAPE-hiekkalaatikossa on paljon enemmän ominaisuuksia analysoidun datan pureskelemiseen. Datan pureskelu ominaisuuksia käyty läpi kappaleessa 5.3.3 processing.conf kohdassa. Nämä ominaisuudet ovat hyödyllisiä ja tämän ansiosta CAPE on selvästi parempi ominaisuuksien perusteella. Molemmissa tuotteissa on yhteisöominaisuus, jolla on mahdollista ladata yhteisön tekemiä haittaohjelmien tunnistussääntöjä ja sormenjälkiä. Valitettavasti Cuckoo-hiekkalaatikon yhteisösääntöjä ei oltu moneen vuoteen päivitetty. Toisin kuin CAPE-hiekkalaatikon, joka on saanut säännöllisiä päivityksiä.

6.4 Ylläpidettävyys

Ylläpidettävyys Cuckoo-hiekkalaatikossa on dokumentaation mukaan paljon kätevämpää kuin CAPE-hiekkalaatikon tapauksessa. Cuckoo-hiekkalaatikon tapauksessa päivityksen ajoa ei voitu testata, sillä Cuckoo-sovellukseen ei ole julkaistu päivityksiä moneen vuoteen. Cuckoota ollaan kään-

tämässä Python 3 -ohjelmointikieleen yhteensopivaksi ja päivityksiä vanhaan versioon ei ole näkyvissä. Cuckoossa kuitenkin dokumentaation mukaan on automaattinen päivitysmahdollisuus. Kun taas CAPE-hiekkalaatikossa ainoa tapa päivittää on asentaa koko CAPE uudestaan. Tämä kuitenkaan ei onneksi asennusskriptin ansiosta ole työlästä.

6.5 Piste-tulokset

Molemmat tuotteet arvosteltiin vielä pisteitä käyttäen perusteluina aikaisempien kappaleiden vertailun tulokset. Tuotteet arvosteltiin pisteillä yhdestä viiteen. Arvosteltava kategoriat ovat samat kuin vertailussa käytetyt. Eli asennus, analysointi, ominaisuudet ja ylläpidettävyys.

Seuraavaksi hieman perusteluja arvostelusta. Asennus oli Cuckoo-hiekkalaatikossa pitkä prosessi ja vaati tuotteeseen perehtymistä, joten tästä syystä asennuksesta Cuckoo-hiekkalaatikko sai kolme pistettä. CAPE-hiekkalaatikossa sen sijaan oli asennukseen tarkoitettu skripti ja asennus onnistui melkein kokonaan pelkän skriptin ajolla. Tästä syystä CAPE-hiekkalaatikko saa asennuksesta viisi pistettä. Analysointi oli molemmissa tuotteissa hyvä. CAPE-hiekkalaatikossa oli yksi valikko enemmän analysointia varten, joten CAPE-hiekkalaatikko sai tästä kategoriasta viisi pistettä ja Cuckoo neljä pistettä. Ominaisuudet osuudessa CAPE sai myös viisi pistettä ja Cuckoo neljä. CAPE-hiekkalaatikko sai enemmän pisteitä, sillä CAPEssa oli enemmän tärkeitä ominaisuuksia. Ylläpidettävyysnäkökulmasta Cuckoo-hiekkalaatikko on selvästi parempi, sillä tuotteessa on automaattinen päivitys mahdollisuus. CAPE-hiekkalaatikon päivittäminen onnistuu vain tuotteen uudelleen asentamalla. Cuckoo-hiekkalaatikko sai ylläpidettävyydestä viisi pistettä ja CAPE-hiekkalaatikko kaksi pistettä. Yhteensä pisteitä kertyi siis Cuckoo-hiekkalaatikolle 16 ja CAPE-hiekkalaatikolle 17 pistettä. Alla olevassa taulukossa nähtävissä piste tulokset.

Taulukko 1. Tuotteiden arvostelu pisteet.

	Cuckoo	CAPE
Asennus	3	5
Analysointi	4	5
Ominaisuudet	4	5
Ylläpidettävyys	5	2
Yhteensä	16	17

7 Yhteenveto

Työn tarkoituksena oli vertailemalla selvittää vertailemalla mikä avoimen lähdekoodin haittaohjelma hiekkalaatikko tuote olisi käytännöllisin samalla tavoitteena oli kasvattaa tietoa kyseisistä tuotteista. Tuotteiksi rajattiin kaksi suosituinta avoimen lähdekoodin tuotetta Cuckoo-hiekkalaatikko ja CAPE-hiekkalaatikko. Vertailtaviksi kategorioiksi rajattiin tuotteiden asennus prosessi, tuotteiden analysointi kyvykkyydet, tuotteiden ominaisuudet ja tuotteiden ylläpidettävyys.

Yhteenvetona molemmissa tutkituissa tuotteissa Cuckoo-hiekkalaatikko ja CAPE-hiekkalaatikko on ominaisuudet tarvittavalla tasolla. Vastauksena tutkimuksen kysymykseen ”Mikä on arvioitavista avoimen lähdekoodin hiekkalaatikko tuotteista käytännöllisin toimeksiantajalle?” on vertailun ja piste tuloksien perusteella CAPE-hiekkalaatikko. CAPE-hiekkalaatikossa oli paremmat analysointikyvykkyydet ja paremmat ominaisuudet kuin Cuckoo-hiekkalaatikolla. CAPE-hiekkalaatikkoa ylläpidetään myös aktiivisesti ja siihen tulee uusia ominaisuuksia jatkuvasti. CAPE hiekkalaatikon yhteisö ominaisuus on myös aktiivisesti päivitetty, joten tuotteeseen on saatavilla uusia ajankohtaisia haittaohjelman tunnistus sääntöjä helposti. Cuckoo olevat ominaisuudet ja analysointi kyvykkyydet eivät ole CAPE-hiekkalaatikon tasolla. Cuckoo-hiekkalaatikossa kuitenkin on selvä käyttöliittymä ja hyviä ominaisuuksia. Isona miinuksena Cuckoo-hiekkalaatikkoa ei ole aktiivisesti päivitetty moneen vuoteen ja tuotteen Python 3 -ohjelmointikielelle käännöstä ei ole tullut mitään uutisia.

Tutkimus kasvatti osaamista hiekkalaatikko tuotteiden käytöstä ja tuotteista yleisesti. Tutkimuksena saatiin myös toteutettua kaksi toimivaa haittaohjelmien hiekkalaatikointi ympäristöä. Jatkokehityksenä tuotteista voisi tarkemmin tutkia eri haittaohjelmien analysointia ja käydä syvällisemmin läpi analysointimahdollisuuksia.

Lähteet

Abrams, L. 2020. New Evasion Encyclopedia Shows How Malware Detects Virtual Machines. Viitattu 26.3.2021.
https://www.bleepingcomputer.com/news/security/new-evasion-encyclopedia-shows-how-malware-detects-virtual-machines/.

Afianian, A. & Niksefat, S. & Sadeghiyan, B. & Baptiste, D. 2018. Malware Dynamic Analysis Evasion Techniques: A Survey Tutkimus. Viitattu 12.04.2022.
https://arxiv.org/pdf/1811.01190.pdf.

Anand, A. 2019. Malware Analysis 101 – Sandboxing. Viitattu 26.3.2021
https://infosecwriteups.com/malware-analysis-101-sandboxing-746a06432334.

Arntz, P. 2020. Sandbox in security: what is it, and how it relates to malware. Viitattu 26.3.2021.
https://blog.malwarebytes.com/awareness/2020/09/sandbox-in-security/.

Banerjea, A. 2018. NotPetya: How a Russian malware created the world's worst cyberattack ever. Artikkeli business-standard verkkosivulla. Viitattu 22.3.2022.
https://www.business-standard.com/article/technology/notpetya-how-a-russian-malware-created-the-world-s-worst-cyberattack-ever-118082700261_1.html.

Baykal,A. 2015. A guide to the zeus virus. Artikkeli fedtechmagazine verkkosivulla. Viitattu 21.3.2022.
https://fedtechmagazine.com/article/2015/02/guide-zeus-virus.

Bencherchali, N. 2019. Malware Analysis Techniques — Basic Static Analysis. Blogi teksti medium verkkosivulla. Viitattu. 23.3.2022.
https://nasbench.medium.com/malware-analysis-techniques-basic-static-analysis-335a7286a176.

Bulazel, A. & Yener, B. 2017. A Survey On Automated Dynamic Malware Analysis Evasion and Counter-Evasion. Tutkimus. Viitattu 12.04.2022.

https://www.researchgate.net/publication/322588326_A_Survey_On_Automated_Dynamic_Malware_Analysis_Evasion_and_Counter-Evasion_PC_Mobile_and_Web.

Computer worm. N.d. Malwarebytes verkkosivut. Viitattu 20.11.2022. https://www.malwarebytes.com/computer-worm.

Culafi, A. 2020. Ransomware attacks see 148% surge amid COVID-19. Artikkeli techtarget verkkosivulla. Viitattu 25.4.2022. https://www.techtarget.com/searchsecurity/news/252481832/Ransomware-attacks-see-148-surge-amid-COVID-19.

Gençaydın, B. 2021. Malware Sample Sources — New & Maintained. Viitattu 27.3.2022. https://infosecwriteups.com/malware-sample-sources-a3c7f306adea.

Heena, Mehtre, B. M. 2021. Advances In Malware Detection-An Overview. Viitattu 19.3.2022 https://arxiv.org/pdf/2104.01835.pdf.

How Malware Analysis Benefits Incident Response. N.d. Solutionary yrityksen teettämä tutkimus. Viitattu 25.4.2022. https://informationsecurity.report/Resources/Whitepapers/51e831f9-aeef-41a4-b2e9-5162a2ac5f65_How%20Malware%20Analysis.pdf.

Marpaung, J.A.P & Sain, M. & Lee, H. 2012. Survey on malware evasion techniques: state of the art and challenges. Tutkimus. Viitattu 11.04.2022. https://ieeexplore.ieee.org/abstract/document/6174775/authors#authors.

Pagani, P. 2022. Organizations paid at least $602 million to ransomware gangs in 2021. Artikkeli securityaffairs verkkosivulla. Viitattu 25.4.2022. https://securityaffairs.co/wordpress/127974/cyber-crime/ransomware-payments-600m-2021.html.

Paganini, P. 2018. WannaMine, the sophisticated crypto miner that spreads via NSA EternalBlue exploit. Artikkeli securityaffairs verkkosivulla. Viitattu 22.3.2022. https://securityaffairs.co/wordpress/68518/malware/wannamine-nsa-eternalblue.html.

Palvelut ja ratkaisut. N.d. Elisa santa monican verkkosivu. Viitattu 25.4.2022 https://www.elisasantamonica.fi/.

Perakalin, A. 2017. Fireball: Adware with potential nuclear consequences. Blogiteksti Kaspersky verkkosivuilla. Viitattu 21.3.2022 https://www.kaspersky.com/blog/fireball-adware/17015/.

Samet, R. Aslan, Ö. A. 2020. A Comprehensive Review on Malware Detection Approaches. Viitattu 19.3.2022 https://ieeexplore.ieee.org/abstract/document/8949524.

Sihwail, R. 2018. A Survey on Malware Analysis Techniques: Static, Dynamic, Hybrid and Memory Analysis. Julkaisu International Journal on Advanced Science Engineering and Information Technolog lehdessä. Viitattu 23.3.2022. https://www.researchgate.net/profile/Rami-Sihwail/publication/328760930_A_Survey_on_Malware_Analysis_Techniques_Static_Dynamic_Hybrid_and_Memory_Analysis/links/5d73c61892851cacdb28d68f/A-Survey-on-Malware-Analysis-Techniques-Static-Dynamic-Hybrid-and-Memory-Analysis.pdf.

Sikorski, M & Honig, A. 2012. Practical Malware Analysis. San Francisco: No starch press. Viitattu 21.3.2022.

Singh, J. & Singh, J. 2018. Challenges of Malware Analysis: Obfuscation Techniques Tutkimus. Viitattu 11.04.2022. https://dergipark.org.tr/en/download/article-file/2160186.

Souri, A. Hosseini, R. 2018. A state-of-the-art survey of malware detection approaches using data mining techniques. Viitattu 19.3.2022. https://hcis-journal.springeropen.com/articles/10.1186/s13673-018-0125-x.

Tahir, R. 2018. A Study on Malware and Malware Detection Techniques. Viitattu 19.3.2022. https://www.mecs-press.org/ijeme/ijeme-v8-n2/IJEME-V8-N2-3.pdf.

Taloustiedot. N.d. Finder verkkosivu. Viitattu 25.4.2022 https://www.finder.fi/Tietoliikennepalvelut+tietoliikennelaitteet/Elisa+Santa+Monica+Oy/Helsinki/yhteystiedot/920995.

Taveres, P. 2022. Artikkeli infosecinstitute verkkosivulla. Viitattu 20.11.2022. https://resources.infosecinstitute.com/topic/popular-evasion-techniques-in-the-malware-landscape/.

Wallen, J. 2020. Containers: A cheat sheet for tech pros. Viitattu 26.3.2022. https://www.techrepublic.com/article/containers-the-smart-persons-guide/.

What is a virtual machine (VM)? N.d. Microsoftin verkkosivu. Viitattu 26.3.2022 https://azure.microsoft.com/en-us/overview/what-is-a-virtual-machine/.

What is CAPE? N.d. CAPE tuotteen ohjesivusto. Viitattu 17.10.2022. https://capev2.readthedocs.io/en/latest/introduction/what.html.

What is Cuckoo? N.d. Cuckoo tuotteen ohjesivusto. Viitattu 13.10.2022. https://cuckoo.readthedocs.io/en/latest/introduction/what/.

What is Stuxnet? N.d. Artikkeli Malwarebytes verkkosivulla. Viitattu 21.3.2022. https://www.malwarebytes.com/stuxnet.

What is WannaCry ransomware? N.d. Blogiteksti Kaspersky verkkosivuilla. Viitattu 21.3.2022. https://www.kaspersky.com/resource-center/threats/ransomware-wannacry.

Zeltser, L. N.d. Docker Containers for Malware Analysis. Viitattu 27.3.2022. https://zeltser.com/media/archive/docker.pdf

Zeltser, L. 2021. Free Malware Sample Sources for Researchers. Viitattu 19.11.2022. https://zeltser.com/malware-sample-sources/.

Zetter, K. 2014. DarkHotel: A Sophisticated New Hacking Attack Targets High-Profile Hotel Guests. Artikkeli Wired verkkosivulla 10.11.2014. Viitattu 21.3.2022 https://www.wired.com/2014/11/darkhotel-malware/.