

Ville Solonen

Houkutinansojen käyttö kyberuhkien havaitsemisessa

Opinnäytetyö
Tieto- ja viestintätekniikka

2020



**Kaakkois-Suomen
ammattikorkeakoulu**

Tekijä/Tekijät	Tutkinto	Aika
Ville Solonen	Insinööri (AMK)	Huhtikuu 2020
Opinnäytetyön nimi Houkutinansojen käyttö kyberuhkien havaitsemisessa		40 sivua 3 liitesivua
Toimeksiantaja Kaakkois-Suomen ammattikorkeakoulu XAMK		
Ohjaaja Vesa Kankare		
Tiivistelmä <p>"Hunajapurkki" on jokin tarkoituksella haavoittuvaksi tehty kohde jonka tarkoituksena on houkuttaa hyökkääjiä tunkeutumaan kohteeseen, jotta saadaan tuotettua tietoa jonka avulla voidaan kehittää kyberturvallisuutta. Tässä työssä käytetään termiä "hunajapurkki" tarkoittamaan yleisesti näitä houkutinansoja. Opinnäytetyön tavoitteena oli lähteä selvittämään "hunajapurkkien" käyttötapoja sekä hyötyä nykypäivän kyberturvallisuudessa sekä saada vastaus tutkimusongelmaan. Tämän työn tutkimusongelma on se, että meillä ei ole tarpeeksi näkyvyyttä verkkoon havaitaksemme lateraalista liikettä sekä nähdäksemme IoT-laitteisiin kohdistuvia uhkia. Työ aloitettiin perehtymällä yleisesti siihen mitä "hunajapurkit" ovat sekä selventämällä niiden välisiä eroja, hyötyjä ja mahdollisia heikkouksia. Tässä tutkimuksessa myös syvennyttiin tarkemmin muutamaa valittuun "hunajapurkkiin", pohtien niiden tarjoamia ominaisuuksia sekä niiden mahdollista hyötyä tutkimuksen kannalta. Kerättyjen tietojen avulla voitiin tähän tutkimukseen valita sopivat "hunajapurkit".</p> <p>Työ toteutettiin Kaakkois-Suomen ammattikorkeakoulun Kotkan kampuksen ICTLABin tiloissa. Työn aikana toteutettiin kaksi eri testiä. Ensimmäisen testin tarkoituksena oli pyrkiä havaitsemiseen lateraalista liikettä. Siinä hyödynnettiin Raspberry Pi -pienoistietokonetta alustana, joka ensin valmisteltiin asentamalla "hunajapurkki", jonka jälkeen laite liitettiin julkisen IP-osoitteen alle DMZ-alueelle keräämään tietoa saapuvista yhteyksistä. Toisessa testissä käytettiin Philips Hue Bridgeä houkuttimena ja siihen saapuvia yhteyksiä tarkkailtiin ja tallennettiin käyttäen Ethernet LAN tap -laitetta sekä tcpdump-pakettianalysoijaa.</p> <p>Vaikka lateraalista liikettä ei voitu testien tuloksista havaita, voidaan kuitenkin todeta sen havaitsemisen olevan teoreettisesti mahdollista esimerkiksi "hunajaverkon" avulla ja hyvä aihe jatkotutkimukselle. "Hunajaverkko" on useista "hunajapurkeista" muodostuva verkko-arkkitehtuuri, jonka tarkoituksena on tutkia kokonaisen verkon tietoliikennettä yhden kohteen sijaan. Tässä työssä käytetään termiä "hunajaverkko" tarkoittamaan tämän tyyppisiä useista houkutinansoista koottuja verkkoja. IoT-laitteisiin kohdistuvan tietoliikenteen tarkkailu suoritettiin onnistuneesti ja tietoliikenteen analysointia suoritettiin, jotta voitiin todeta haitallisen liikenteen esiintyminen. Testeissä todetuista haitallisista tietoliikennetapahtumista saatiin tietoa, jonka avulla löydettiin tutkimuskysymyksiin vastaukset.</p>		
Asiasanat kyberturvallisuus, lateraalinen liike, houkutinansa		

Author (authors)	Degree	Time
Ville Solonen	Bachelor of Engineering	April 2020
Thesis title		40 pages 3 pages of appendices
The use and benefit of honeypots in cybersecurity		
Commissioned by		
South-Eastern Finland University of Applied Sciences XAMK		
Supervisor		
Vesa Kankare		
Abstract		
<p>The objective of this thesis was to define the use of honeypots and the potential benefit in today's cyber security as well as means to increase the visibility in the networks to notice lateral movement and detect the threats directed towards IoT devices. Four honeypots were selected for a closer examination with regards to their features and possible benefit. With the gathered information, a suitable honeypot for thesis could be selected.</p> <p>The examination was performed in the ICTLAB environment on the South-Eastern Finland University of Applied Sciences campus in Kotka. During the examination, two tests were completed. The purpose of the first test was to observe lateral movement. A Raspberry Pi miniature computer was used as a platform, on which the selected honeypot was installed, and which was then connected with a public IP address and placed in the DMZ to collect information about the incoming connections. In the second test, Philips Hue Bridge was used as a decoy and the incoming connections were observed and recorded using Ethernet LAN tap device and tcpdump packet analyser.</p> <p>Even though lateral movement could not be seen from the test results, one could argue that it is theoretically possible to observe it by using a honeynet, which might be a good topic for a follow-up study. The observation of network traffic towards IoT devices was completed successfully, and the traffic was analysed so that the existence of malicious activities could be detected. Malicious traffic was seen during the tests, from which information could be gathered that clearly indicated the benefits gained from the use of a honeypot.</p>		
Keywords		
cyber security, lateral movement, honeypot		

SISÄLLYS

1	JOHDANTO.....	6
2	TUTKIMUKSEN TAVOITTEET	7
2.1	Tutkimusongelma ja tutkimuskysymykset.....	7
2.2	Tutkimusmenetelmä	9
3	"HUNAJAPURKKIEN" KATEGORISOINTI	10
3.1	IDS ja IPS	11
3.2	Hunajapurkin sijoittaminen verkkoon	12
3.3	Alhainen ja korkea vuorovaikutustaso	13
3.4	Palvelin- ja asiakastyypiset.....	14
3.5	Fyysiset ja virtuaaliset.....	15
3.6	"Hunajaverkko"	15
4	HUNAJAPURKIN VALINTA.....	19
4.1	HoneyD.....	19
4.2	Dionaea	20
4.3	KFSensor.....	20
4.4	Cowrie	21
4.5	Taulukointi ja valinta	22
5	TYÖN ETENEMINEN	23
5.1	ICTLAB hunajapurkki.....	23
5.1.1	Raspberry Pi	24
5.1.2	Hunajapurkin asennus	25
5.1.3	Julkinen hunajapurkki	26
5.2	IoT-Hunajapurkki	28
5.2.1	Kontrolloitu testi	28
5.2.2	Julkisen verkon testi.....	29
6	TULOKSET JA PÄÄTELMÄT	33
	LÄHTEET.....	36

LIITTEET

Liite 1. Cowrien asennus ja käyttöönotto

1 JOHDANTO

Teknologian kehittyessä yhä useampi laite on yhteydessä internettiin. Vaikka tämä teknologian kasvu tuo koteihin tuotteita kuten älyvalaisimet, on hyvä pitää mielessä, että se myös tarjoaa hyökkääjille lisää hyökkäyspinta-alaa, eli kohteita, joihin hyökätä. Näiden Internet of Things- eli IoT-laitteiden turvallisuus jättää usein toivomisen varaa ja niiden heikkouksia käytetään jatkuvasti hyväksi. Tämä voidaan todeta tarkasteltaessa esimerkiksi Impervan (Simonovich 2019) havaitsemaa bottiverkon palvelunestohyökkäystä, jossa käytössä oli noin 400 000 laitetta, joista suurin osa oli IoT-laitteita. Hyökkäys jatkui 13 päivän ajan, jona aikana yhteyspyyntöjen määrä kävi hieman alle 300 000:ssa per sekunti. Imperva onnistui hyökkäyksen kuitenkin torjumaan ilman että asiakkaat kärsivät häiriöajasta. Tämä ei kuitenkaan tarkoita sitä, että voidaan jatkaa samalla linjauksella eteenpäin, sillä IoT-laitteiden kasvun myötä suurenevat myös käytettävät bottiverkot ja sen mukana myös hyökkäysten koot. Vielä merkittävämmän tästä tekee se, että IoT-laitteeseen pääsyn jälkeen on hyökkääjällä mahdollista lähteä levittämään hyökkäyspinta-alansa käyttäen hyväkseen niin kutsuttua lateraalista liikettä, jossa yhden laitteen murtamisen jälkeen pyritään etenemään verkossa oleviin muihin laitteisiin.

”Hunajapurkki” on jokin tarkoituksella haavoittuvaksi tehty tai haavoittuvalta näyttävä laite, jonka tarkoituksena on houkutella hyökkääjiä tunkeutumaan kohteeseen (Göbel & Dewald 2010, 7). Tässä työssä käytetään termiä ”hunajapurkki” tarkoittamaan näitä houkutinansoja. ”Hunajapurkkeja” tullaan tarkastelemaan lähemmin myöhemmissä luvuissa. Tämän tutkimuksen tarkoituksena on selvittää, voidaanko ”hunajapurkkia” käyttää IoT-laitteisiin kohdistuvien uhkien sekä lateraalisen liikkeen havaitsemiseen. Aiheen työlleni sain koulun kautta ideapankista, ja tämä kyseinen aihe vaikutti mielenkiintoiselta, joten päätin ottaa sen työn alle. En myöskään omaa aiempaa kokemusta ”hunajapurkeista”, joten on se myös siinä mielessä erittäin mielenkiintoinen haaste henkilökohtaisesti itselleni.

Työ koostuu pääosin kolmesta osuudesta, jotka ovat teoriaosuus, käytännön osuus sekä tulokset ja päätelmät. Teoriaosuudessa käydään läpi tutkimuksen tavoitteet, tutkimusongelma sekä tutkimuskysymykset. Teoriaosuudessa perehdytään myös ”hunajapurkkien” toimintaperiaatteisiin, selvitetään eri ”hunajapurkkityyppien” eroja sekä pyritään selvittämään minkä tyyppinen ”hunajapurkki” olisi tutkimukseen parhaiten soveltuva. Käytännön osuudessa toteutetaan teoriaosuudessa valittu ”hunajapurkki” ICTLAB ympäristössä. Valitun ”hunajapurkin” avulla suoritetaan testejä, joista saaduista tuloksista pyritään saamaan vastaus tutkimusongelmaan tämän opinnäytetyön tulososiossa. Tulosten ja pohdinnan osuudessa pohditaan muun muassa tuotettujen tulosten merkityksen lisäksi myös sitä, kuinka hyvin työssä onnistuttiin sekä mahdollisia jatkotutkimusaiheita. Osuudessa pyritään myös vastaamaan tutkimuksessa asetettuihin tutkimuskysymyksiin ja sitä kautta saamaan vastaus itse tutkimusongelmaan.

2 TUTKIMUKSEN TAVOITTEET

Tämän työn tarkoituksena on pohtia ”hunajapurkkien” käyttötarkoitusta sekä niistä saatavaa hyötyä. Tarkemmin sanottuna tarkoituksena on pohtia, voidaanko ”hunajapurkin” avulla tarkkailla hyökkääjien liikkeitä tarpeeksi yksityiskohtaisesti, jotta olisi mahdollista todeta lateraaliseen liikkeen esiintyminen. Tarkoituksena on myös selvittää, voidaanko ”hunajapurkin” avulla ylipäättään nähdä testilaitteisiin kohdistuvaa haitallista tietoliikennettä. Tutkimuksen tavoitteena on myös kartoittaa hieman mahdollisia käytettäviä ”hunajapurkkeja” ja löytää niistä tähän työhön sopivat. Työhön sopivien ”hunajapurkkien” löydyttyä, pyritään niistä kerättyjen tietojen avulla ratkaisemaan tutkimusongelma.

2.1 Tutkimusongelma ja tutkimuskysymykset

Tietoverkoissa yksikin heikkous voi johtaa siihen, että koko muu toteutettu kyberturva romahtaa ja hyökkääjä pääsee yhden laitteen murtamisen jälkeen etenemään verkossa sisältä käsin laitteesta laitteeseen etsien arvokasta tietoa. Tätä tapaa liikkua verkossa kutsutaan lateraaliseksi liikkeeksi, ja sitä voi olla hyvinkin vaikea huomata sisäverkossa. Vaikean huomata siitä tekee se, että hyökkääjä voi saada käyttöönsä verkossa käytettyjä tunnuksia, joiden

avulla sisäverkossa liikkuminen voi vaikuttaa tavanomaiselta liikenteeltä (National Cyber Security Centre 2018). On myös epätodennäköistä, että itse sisäverkkoa edes valvotaan yhtä tarkasti kuin ulkoapäin tulevia yhteyksiä, ja vaikka valvottaisiinkin, on lateraalista liikettä erittäin vaikea huomata tavanomaisin keinoin. Työhön liittyvä tutkimusongelma onkin se, että meillä ei ole tarpeeksi näkyvyyttä verkkoon havaitaksemme lateraalista liikettä sekä nähdäksemme IoT-laitteisiin kohdistuvia uhkia.

Ongelmaan lähdetään etsimään ratkaisua tutkimuskysymyksillä kuten minkälaisia ”hunajapurkkeja” voidaan tutkimuksessa hyödyntää, kuinka hyökkääjä toimii kohteessa sekä kuinka tutkimuksesta saatua tietoa voidaan hyödyntää kyberturvallisuuden kehittämisessä.

Ensimmäinen tutkimuskysymys on se, että

- minkälaisia ”hunajapurkkeja” voidaan tutkimuksessa hyödyntää?

”Hunajapurkin” hyöty voidaan mitata esimerkiksi sen tuottaman tiedon määrän ja yksityiskohtaisuuden avulla. ”Hunajapurkin” valinnassa on huomioitava myös se, minkälaista tietoa halutaan tuottaa. Jos tavoitteena on lähinnä saada selvää hyökkäyksien määrästä, voi alhaisen vuorovaikutuksen ”hunajapurkki” olla sopiva tutkimukseen. Mutta jos halutaan tietoa esimerkiksi siitä, että miten hyökkääjä levittää hyökkäyspinta-alaansa, voi olla tarpeen ottaa käyttöön korkean vuorovaikutuksen ”hunajapurkki”. ”Hunajapurkkia” valittaessa on myös syytä hyödyntää aiempia tutkimuksia, ja eräs tässä työssä käytettävä on A survey on honeypot software and data analysis (Nawrocki ym. 2016). Edellä mainitussa tutkimuksessa jaotellaan ”hunajapurkkeja” käyttötarkoituksen sekä toiminnallisuuden mukaan, sekä perehdytään yksityiskohtaisesti jokaiseen ja selvitetään niiden tarjoamia ominaisuuksia. Uskon siis tutkimuksen olevan tärkeä tiedonlähde ”hunajapurkkia” valittaessa.

Toisessa tutkimuskysymyksessä pohditaan sitä, että

- kuinka hyökkääjä toimii kohteessa?

Tärkeä kysymys tutkimusongelmaa mietittäessä on se, kuinka hyökkääjä toimii kohteeseen pääsyn jälkeen. Sillä jos voidaan yksityiskohtaisesti seurata hyökkääjän toimia kohteessa, voidaan mahdollisesti saada viitteitä siitä, kuinka kyberturvaa olisi mahdollista lähteä tehostamaan. Hyökkääjän toiminnan näkeminen on myös oleellinen osa lateraalisen liikkeen havainnoimisessa. Hyökkääjän toiminnan näkemisen kannalta on tarpeellista myös osata valita oikeantyyppinen ”hunajapurkki”, jotta saadaan tuotettua tutkimuksen kannalta hyödyllistä tietoa. Oikeantyyppisen ”hunajapurkin” valintaan perehdytään tarkemmin myöhemmässä luvussa.

Kolmas tutkimuskysymys on

- kuinka tutkimuksesta saatua tietoa voidaan hyödyntää kyberturvallisuuden kehittämisessä?

Ei siis riitä, että kerätään tietoa, vaan on tärkeä myös tutkia kerätyn tiedon hyötyä ja pohtia millä tavoin sitä voitaisiin tulevaisuudessa hyödyntää kyberturvan parantamiseen. Voidaan esimerkiksi tutkia kerättyä tietoa ja etsiä uusia lähestymistapoja, joita hyökkääjät käyttävät tai voidaan pyrkiä löytämään ja paikkaamaan heikkouksia, joita käytetään hyökkäyksissä hyväksi. Tavoitteena on tutkimuksesta saadun tiedon avulla pystyä toteuttamaan oikeantyyppinen ”hunajapurkki” oikeantyyppiseen tilanteeseen ja näin ollen tehostaa verkon valvontaa.

2.2 Tutkimusmenetelmä

Tutkimusmenetelmänä toimii pääosin kvantitatiivinen eli määrällinen tutkimus koska uskon tutkimuksen tarvitsevan tuloksia laajemmalla otannalla. Heikkilän (2014, 15) mukaan, ”sen avulla selvitetään lukumääriin ja prosenttiosuuksiin liittyviä kysymyksiä. Se edellyttää riittävän suurta ja edustavaa otosta. Asioita kuvataan numeeristen suureiden avulla ja tuloksia voidaan havainnollistaa taulukoin ja kuvioin.” Mutta on todennäköistä, että tutkimuksen edetessä tulee vastaan yksittäistilanteita, joiden toimintaa seuraamalla saadaan selville jotain mitä ei koko joukkoa tutkimalla voida havaita. Tässä tapauksessa poiketaan kvalitatiivisen eli laadullisen tutkimuksen puolelle ja tutkitaan tarkemmin yksi-

lön toimintaa, jos nähdään sen olevan tutkimuksen kannalta oleellista. Pitkärannan (2014, 13) mukaan, ”laadullisen tutkimuksen tehtävä on lisätä ymmärrystä, mahdollistaa erilaisia tulkintoja, antaa asioille merkityksiä ja tuottaa asioista mallinnuksia.” Onkin siis syytä todeta, että tutkimuksen lopullinen tutkimusmenetelmä on triangulaatio. Kanasen mukaan (2014, 123) triangulaatiolla tarkoitetaan usean tutkimusmenetelmän yhdistämistä samassa tutkimuksessa, yhdistäminen voi tapahtua käyttämällä useampaa menetelmää kuten esimerkiksi laadullista ja määrällistä tutkimusta. Tämän tutkimuksen tapauksessa lähdetäänkin siis yhdistämään laadullista sekä määrällistä metodologiaa, koska uskon kokonaisjoukon sekä muutaman valitun yksittäistapauksen tutkimisen tuovan työlle halutun lopputuloksen.

3 ”HUNAJAPURKKIEN” KATEGORISOINTI

”Hunajapurkin” tarkoituksena on olla jokin ulkoa katsottuna houkuttavalta näytävä kohde, esimerkiksi vaikka web-palvelin, jolle ei sijoiteta mitään varsinaista sisältöä ja jolla ei kulje minkäänlaista ylimääräistä tietoliikennettä. Tämän houkuttimen ainoa tehtävä on taltioida kaikki siihen kohdistuva tietoliikenne, sillä voidaan olettaa, että kaikki tähän palvelimeen yhteyttä ottavat ovat mahdollisia hyökkääjiä. Näin voidaan saada tuotettua arvokasta tietoa siitä, kuinka hyökkääjät toimivat ja jopa saada selville haavoittuvuuksia, joita ei aiemmin tiedetty. Tutkimuksessa ei tulla enää jatkossa merkitsemään termiä ”hunajapurkki” lainausmerkkeihin, vaan tarkoitetaan sillä tässä määriteltä laitetta, sovellusta tai ilmiötä.

Hunajapurkkien avulla on myös mahdollista löytää nollapäivähaavoittuvuuksia tarkoittaen haavoittuvuuksia, joihin ei ole vielä korjausta, ja joita voi muutoin olla hyvin vaikea löytää (Göbel & Dewald 2010, 8). Hunajapurkkeja voidaan jakaa eri kategorioihin muun muassa sen mukaan, kuinka paljon tilaa ne antavat hyökkääjälle kohteessa, miten ne saavuttavat tavoitteensa sekä sen mukaan millaisella alustalla hunajapurkki on toteutettu. Tässä luvussa tarkoituksena on selvittää eri tyyppien eroja sekä hyötyjä sekä saada käsitys siitä, millaisia hunajapurkkeja voidaan toteuttaa.

3.1 IDS ja IPS

On hyvä ymmärtää, minkä tyyppinen puolustus hunajapurkki on. Puolustuksia voidaan jakaa karkeasti kahteen eri pääkategoriaan: IDS (intruder detection system) eli hyökkäyksen tunnistusjärjestelmät sekä IPS (intruder prevention system) eli hyökkäyksen estojärjestelmät. IDS on järjestelmä, joka seuraa ja analysoi verkon tapahtumia, ja pyrkii löytämään mahdollisesti haitalliset tapahtumat. IPS puolestaan on järjestelmä, jonka tarkoituksena on kerätyn tiedon avulla estää haitallisen liikenteen eteneminen verkossa. (What is IDS and IPS? s.a.) IPS-järjestelmiin lukeutuu esimerkiksi palomuuuri, jonka tehtävänä on suodattaa paketteja esimäärättyjen sääntöjen avulla ja rajoittaa pääsyä verkkoon estäen ei toivottujen yhteyksien etenemisen (Cisco s.a.). IDS-järjestelmiin taas lukeutuu tämänkin tutkimuksen aiheena esiintyvät hunajapurkki-järjestelmät, joiden tehtävänä yleisesti ottaen on lähinnä tarkkailla saapuvaa liikennettä ja taltioida tapahtumat myöhemmää tutkimista varten. Vaikka yleinen ratkaisu on turvata verkko palomuurilla, on syytä sen lisäksi pohtia hyötyä siitä tiedosta mitä hunajapurkki voi tuottaa.

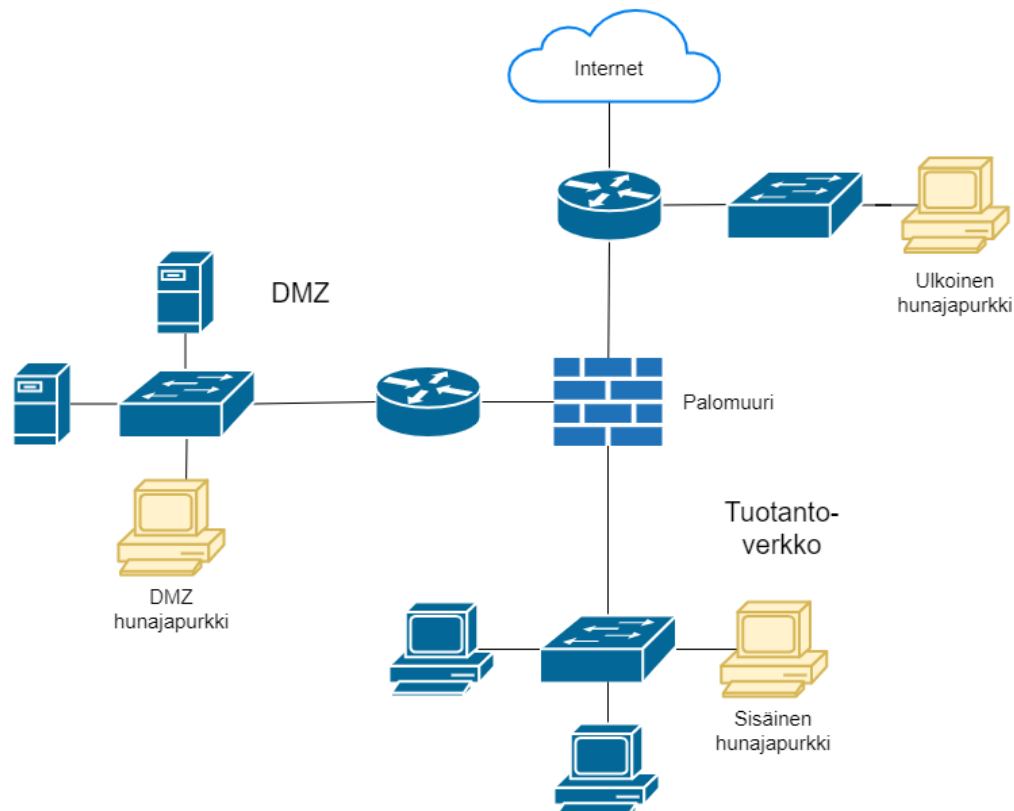
IDS-järjestelmien tietoliikenteen tulkinta voidaan Philokyprosin ym. (2018) mukaan jakaa seuraaviin kolmeen eri tapaan: signature-based eli allekirjoitus pohjaiset, anomaly-based eli poikkeuspohjaiset sekä specification-based eli määrityspohjaiset. Allekirjoitus pohjaiset IDS-järjestelmät havaitsevat hyökkäyksiä sen perusteella, onko hyökkääjän käyttämän hyökkäyksen allekirjoitus jo tallennettuna tietokantaan. Poikkeuspohjaiset havaitsevat hyökkäyksiä normaalista tietoliikenteestä luodun profiilin perusteella, jonka jälkeen poikkeukset normaalista voidaan havaita. Määrityspohjaiset järjestelmät toimivat samalla tavalla kuten poikkeuspohjaiset, mutta ne luovat normaalin tietoliikenteen profiilin ottamalla huomioon suojauskäytännöt sekä muut verkon toiminnot.

Verkon tietoliikenteen valvonnassa on kaksi tapaa sijoittaa järjestelmä: out-of-band sekä inline. Out-of-band on pois suoralta tietoliikenteen linjalta sijoitettu, pakettien kopioita vastaanottava ja tarkkaileva järjestelmä. Järjestelmän voidaan sanoa olevan pois suoralta tietoliikenteen linjalta, jos sen poistaminen ei vaikuta muun verkon toimintaan. Usein IDS-järjestelmät sijoitetaan pois lin-

jalta, sillä tarkoituksena on lähinnä tutkia ja analysoida liikennettä eikä suoraan reagoida saapuviin paketteihin kuten IPS-järjestelmät. Inline tarkoittaa sitä, että laite on suoraan tietoliikenteen linjalle sijoitettu ja tarkastelee alkupe-
räisiä paketteja eikä niiden kopioita kuten out-of-band-järjestelmä. Esimerk-
kinä toimii palomuuuri, jonka täytyy sijaita linjalla koska reaaliaikainen paket-
tienvalvonta on oleellinen osa sen toimintaa. Koska inline-järjestelmä on suo-
raan tietoliikenteen linjalla, on sen toiminta suoraan vaikutuksissa muun ver-
kon toimintaan ja sen hajoaminen tai poistaminen linjalta johtaa verkon alas-
ajoon. (Bromley 2016.) Inline järjestelmiä varten on kehitetty bypass switch eli
ohituskytkin, joka sijoitetaan inline-verkkolaitteiden sekä valvontatyökalujen
väliin sallien valvontatyökalujen hajoamisen tai poiston häiritsemättä muun
verkon toimintaa (ixia s.a.).

3.2 Hunajapurkin sijoittaminen verkkoon

Hunajapurkki on mahdollista sijoittaa muutamaankin eri verkon alueeseen riip-
puen siitä, millaista tietoa halutaan tuottaa. Nämä sijoituskohdat ovat kuvassa
1 näkyvät ulkoinen alue, DMZ (demilitarized zone) eli demilitarisoitu alue sekä
sisäinen alue (Grimes 2005, 23).



Kuva 1. Hunajapurkin sijoituskohdat verkossa

Ulkoisessa sijoituskohdassa hunajapurkki on suoraan avoinna internetille, joka tarkoittaa sitä, että välissä ei ole palomuuria estämässä tietoliikennettä ja saadun tiedon määrä on näin ollen suurin mahdollinen. Jos tavoitteena on lähinnä saada mahdollisimman suuri otanta, on sijoitus ihanteellinen. Sijoitus on myös hyvä aloituspiste hunajapurkkien käyttöön, sillä se ei vaadi juurikaan suurempaa suunnittelua muun verkon toiminnan suhteen. (Mt.)

Sisäisessä sijoituskohdassa hunajapurkki on sijoitettu palomuurin taakse tuotantoverkkoon muiden laitteiden kuten työasemien yhteyteen. Tässä tavassa ei välttämättä saada tuotettua yhtä suurta määrää tietoa kuten ulkoisessa, mutta palomuurin läpi pääsevät haitalliset yhteydet voidaan huomata ajoissa ja voidaan niihin täten myöskin reagoida ajoissa. On kuitenkin otettava huomioon myös se, että hunajapurkin sisäisen sijoituskohdan vuoksi se on erinomainen kohde hyökkääjälle saada haltuun, joten on sen turvallisuutta syytä pohdita. Sisäisessä sijoituskohdassa voidaan myös hyödyntää kytkimistä löytyvää portin peilausta tietoliikenteen tutkimiseen. (Mt.) Portin peilaus on tapa tarkkailla tietoliikennettä, jossa tiettyyn porttiin tai koko VLANin saapuvat paketit voidaan kopioida ja lähettää eteenpäin tutkimista varten (Juniper 2019).

DMZ-sijoituksessa hunajapurkki sijoitetaan nimensä mukaisesti verkon DMZ-alueelle. DMZ on verkon alue, joka sijoittuu ulkoisen sekä sisäisen verkon väliin, josta yhteydet on sallittu vain ulkoverkon suuntaan. Tämä mahdollistaa sen, että alueella voidaan ylläpitää ulkoverkon palveluja pitäen sisäverkon suojassa mahdollisilta ulkoverkosta lähestyviltä hyökkääjiltä. (CISA s.a.). Hunajapurkki DMZ-alueella on hyvä vaihtoehto, jos halutaan tutkia alueella oleviin palvelimiin kohdistuvia uhkia. DMZ-sijoituksessa voidaan myös hyödyntää samaa kytkimen portin peilausta tietoliikenteen tarkkailuun kuten sisäisessä sijoituskohdassa. (Grimes 2005, 23.)

3.3 Alhainen ja korkea vuorovaikutustaso

Hunajapurkkeja voidaan jakaa eri vuorovaikutustasoihin sen mukaan, kuinka paljon ne antavat hyökkääjälle tilaa kohteessa ja kuinka interaktiivisia ne ovat,

eli kuinka paljon ne kommunikoivat takaisin hyökkääjälle päin. Matalan vuorovaikutuksen hunajapurkki voi olla vaikka vain yksinkertainen palvelu, jonka hyviin puoliin kuuluu ylläpidon helppous, mutta huonoihin puoliin lukeutuu saadun tiedon alhainen määrä. Matalan vuorovaikutuksen hunajapurkki voi esimerkiksi luoda valeyhteystilan siihen yhteyttä ottavalle mahdolliselle hyökkääjälle. Valeyhteystilassa hyökkääjälle annetaan käyttöön vain tiettyjä komentoja simuloiden oikeaa kohdetta. Matalan vuorovaikutuksen hunajapurkki soveltuukin hyvin havaitsemaan perinteisiä haittaohjelmia, sillä ne tarjoavat juuri tarpeeksi vuorovaikutusta hyökkääjälle hyökkäyksen aloitukseen. (Göbel & Dewald 2010, 8.) Korkean vuorovaikutuksen hunajapurkki voi puolestaan antaa hyökkääjälle käyttöön vaikka koko käyttöjärjestelmän, jolloin voidaan saada aikaan paljon tietoa mutta sen ylläpitäminen vaikeutuu myös huomattavasti. Korkeassa vuorovaikutuksessa hyökkääjälle avautuu mahdollisuuksia, joita ei matalassa vuorovaikutuksessa pystytä tarjoamaan. Hyökkääjä pystyy käyttämään hyväkseen ei pelkästään käyttöjärjestelmän heikkouksia, mutta myös mahdollisesti asennettujen applikaatioiden heikkouksia. Tämän vuoksi korkean vuorovaikutuksen hunajapurkki soveltuukin hyvin nollapäivähaavoittuvuuksien löytämiseen. (Mts. 9.) Täytyy kuitenkin ottaa huomioon myös se, että enemmän tilaa ja valtaa antaessa avautuu aina riski siitä, että hyökkääjä pääsee levittämään hyökkäyspinta-alaansa kohteesta eteenpäin.

3.4 Palvelin- ja asiakastyypiset

Vuorovaikutustasojen lisäksi voidaan hunajapurkkeja jakaa myös palvelin- ja asiakastyypisiin. Palvelintyyppin hunajapurkin tarkoituksena on antaa kaikki valta hyökkääjälle ja olla itse passiivisessa roolissa tallentaen sekä tutkien tietoliikennettä (mts. 11). Palvelintyyppin hunajapurkin ominaisuutena voi olla esimerkiksi emuloida www-palvelinta kuten Microsoftin IIS-palvelinta. Hunajapurkki vastaa pyyntöihin, sekä isännöi verkkosivuja maskeeraten sen hyvin oikean oloiseksi www-palvelimeksi tehden hunajapurkiksi tunnistamisen hyökkääjälle vaikeaksi. (KFSensor Features s.a.)

Asiakastyypin hunajapurkki puolestaan toimii aktiivisessa roolissa, pyrkien itse löytämään ja tallentamaan mahdollisesti haitallista sisältöä. Asiakastyypin hunajapurkki imitoi käyttäjää, joka selaa verkkosivuja sekä dokumentteja. (Göbel

& Dewald 2010, 11.) Hyvänä esimerkkinä asiakastyypisistä hunajapurkista toimii YALIH, joka on Mansoorin ym. (2014) kehittämä hunajapurkki, jonka tarkoituksena on tutkia esimerkiksi sähköpostiin saapuvien linkkien takaa löytyviä mahdollisesti haitallisia web-sivuja. YALIH koostuu kolmesta pääkomponentista, jotka ovat URL collector, visitor agent ja analysis engine. URL collector on vastuussa linkkien keräämisestä esimerkiksi sähköpostiin saapuneista viesteistä. Visitor agent on virtuaalinen selain, joka emuloi oikeaa selainta, ja sen tehtävänä on käydä URL collectorin keräämissä linkeissä sekä noutaa ja tallentaa mahdollisesti haitallisten sivujen sisältö myöhempää tarkastelua varten. Analysis engine tehtävänä on nimensä mukaisesti analysoida tallennettujen sivujen sisältöä ja tunnistaa mahdollisesti haitallinen sisältö.

3.5 Fyysiset ja virtuaaliset

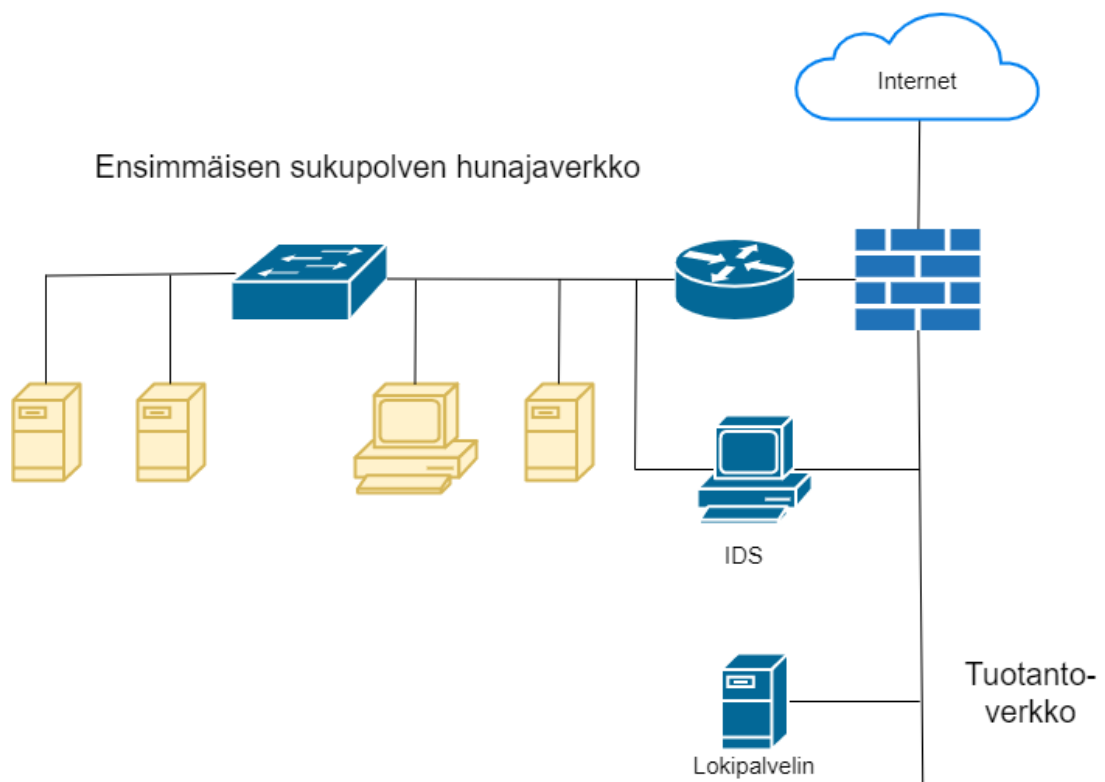
Hunajapurkki voidaan toteuttaa joko fyysisenä tai virtuaalisena järjestelmänä. Fyysisesti toteutettaessa ajetaan järjestelmä suoraan fyysisessä koneessa, jolloin se voi hyökkääjän näkökulmasta vaikuttaa oikeammalta kohteelta kuin virtuaalinen vastine. Fu ym. (2006, 2) mukaan tämä johtuu lähinnä siitä, että virtuaalisessa hunajapurkissa linkkiviive eroaa fyysisestä vastineesta ja tämä voi johtaa hunajapurkin paljastumiseen. Täytyy kuitenkin ottaa huomioon, että nykypäivänä, jolloin käytetään yhä enemmän virtualisointia, ei tämä välttämättä enää täysin pidä paikkaansa. Virtualisoimalla hunajapurkki voidaan myös säästää kustannuksissa ja sen ylläpito helpottuu huomattavasti, sillä samalla fyysisellä laitteella voi samanaikaisesti sijaita monta eri virtuaalista hunajapurkkia (mt.). Fyysisessä laitteessa hyökkäyksen jälkeinen järjestelmän aloituspisteeseen palauttaminen on työlästä ja aikaa vievää, kun taas virtualisoinnissa voidaan tämä hoitaa huomattavasti nopeammin palaamalla aikaisempaan snapshot-tilaan eli aikaisempaan järjestelmän tilaan, joka on tallennettu muistiin. Tämä aikaisempaan tilaan palaamisen ominaisuus löytyy muun muassa virtualisointiohjelmasta VMware Workstationin versiosta 4. (VMware s.a.)

3.6 ”Hunajaverkko”

”Hunajaverkko” on askel eteenpäin yksittäisestä hunajapurkista. Se on useasta hunajapurkista rakennettu kokonainen verkkoarkkitehtuuri, jonka avulla

voidaan tarkemmin tutkia kokonaisen verkon toimintaa yksittäisen kohteen si-
jaan. (Rouse 2007). ”Hunajaverkon” toiminta voidaan Fan ym. (2015, 5) mu-
kaan jakaa kolmeen pääkategoriaan eli tiedonhallintaan, tiedonkeruuseen,
sekä tiedon tallentamiseen. Tiedonhallinnan tarkoituksena on hallita hyök-
käystä niin että se ei leviä ”hunajaverkon” ulkopuolelle. Haasteena on se, että
hyökkääjälle tarvitsee antaa tarpeeksi tilaa, jotta tietoa voitaisiin kerätä mutta
enemmän tilaa annettaessa myös turvallisuusriski kasvaa. Tiedonkeruussa on
kolme päätekijää, jotka ovat palomuurin lokitiedostot eli saapuvat ja lähtevät
yhteydet, verkossa tapahtuva tietoliikenne eli jokainen paketti ja niiden sisältö
sekä järjestelmätoiminta eli muun muassa hyökkääjän kohteessa syöttämät
komennot ja muokatut tiedostot. Tiedon tallentamisessa kerätään tieto talteen
johonkin ulkoiseen tallennuspisteeseen, sillä hunajapurkit itsessään eivät
luontonsa vuoksi ole turvallisia tiedon sijoituskohteita. Tutkimuksessa ei enää
jatkossa tulla merkitsemään termiä ”hunajaverkko” lainausmerkkeihin, vaan
tarkoitetaan sillä tässä määriteltä laitetta, sovellusta tai ilmiötä.

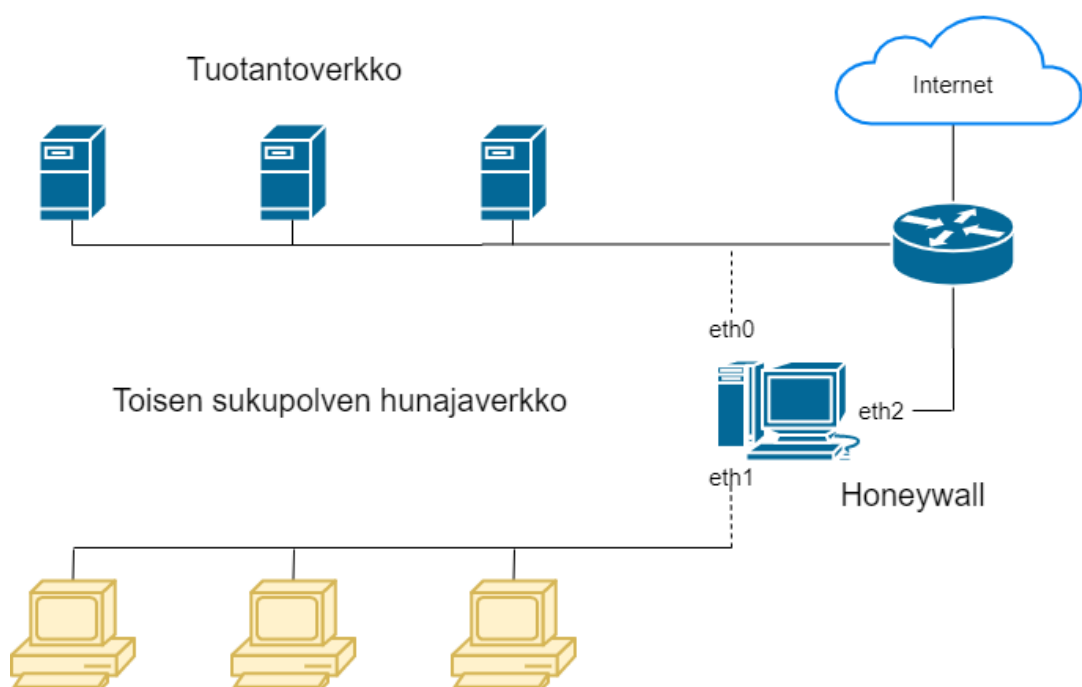
Hunajaverkko ei suinkaan ole uusi keksintö, sillä se on saanut alkunsa jo
vuonna 1999 Honeynet Projectin kehittämän ensimmäisen sukupolven hunaj-
javerkon myötä (kuva 2).



Kuva 2. Ensimmäisen sukupolven hunajaverkko

Ensimmäisen sukupolven hunajaverkossa palomuuuri suorittaa tiedonhallintaa rajoittaen yhteyksiä tietyn yhteysmäärän ylittyessä sekä tiedonkeruuta tallentaen hunajaverkkoon saapuvia sekä siitä lähteviä yhteyksiä. Hunajaverkossa tiedonhallintaa suorittaa myös verkossa oleva reitin, jolla voidaan rajoittaa yhteyksiä esimerkiksi IP-osoitteiden perusteella. IDS on verkon fyysisessä kytkimessä kiinni hyödyntäen kytkimestä löytyvää portin peilausta pakettien tarkasteluun. Kaikki tieto mitä hunajaverkosta saadaan, lähetetään eteenpäin loki-palvelimelle, joka sijaitsee hunajaverkon ulkopuolella. Tämän tyyllisessä ensimmäisen sukupolven hunajaverkossa on heikkouksia ja rajoituksia. Heikkoutena on muun muassa palomuurin tiedonhallinta, jolla yhteys voidaan estää tietyn yhteysmäärän ylittyessä. Tämä voi johtaa siihen, että hyökkääjä tunnistaa kohteen hunajaverkoksi. Toisena heikkoutena on se, että palomuuuri on helposti havaittavissa, sillä sen läpi kulkevan tietoliikenteen TTL (time to live) arvo laskee. Rajoituksena on myöskin se, että tämän tyylinen hunajaverkko on sijoitettava erilleen muusta verkosta, jottei muun verkon turvallisuus kärsi. (Mts. 6.)

Kuvassa 3 näkyvä toisen sukupolven hunajaverkko tuli muutaman vuoden myöhemmin vuonna 2001, jolloin Honeynet Project julkaisi ”hunajamuurin” nimeltään Eeyore.



Kuva 3. Toisen sukupolven hunajaverkko

"Hunajamuuri" on perinteisesti OSI-mallin toisella kerroksella toimiva siltaava laite, jossa on kolme verkkoliitäntää. Kahta verkkoliitäntää eth0 sekä eth1 käytetään erottamaan tuotantoverkon liikenne hunajaverkon liikenteestä. Kolmas verkkoliitäntä eth2 ei ole välttämätön ja sitä voidaan hyödyntää etäkäyttöön. "Hunajamuuri" yhdistää palomuurin sekä IDS:n toiminnot yhteen laitteeseen, joka voi suorittaa sekä tiedonhallintaa että tiedonkeruuta. Toisen sukupolven hunajaverkossa on monia hyötyjä edelliseen sukupolveen verrattuna. Kun ensimmäisen sukupolven hunajaverkon palomuri on hyökkääjälle havaittavissa oleva laite, on "hunajamuuri" täysin näkymätön, sillä vaikka kaikki tietoliikenne kulkee siitä läpi, ei se vähennä TTL:n arvoa koska ei tapahdu pakettien reititystä. Toinen hyöty on se, että toisen sukupolven hunajaverkossa toimiva teknologia nimeltä IDS-yhdyskäytävä osaa tunnistaa haitallisen liikenteen normaalista. Lisähyötynä edelliseen sukupolveen on myöskin se, että siinä missä ensimmäisen sukupolven hunajaverkon täytyy sijaita eristettynä muusta verkosta turvallisuussyistä, voidaan toisen sukupolven hunajaverkkoa pitää osana tuotantoverkkoa, sillä "hunajamuuri" toimii hunajaverkon sekä muun verkon välissä. (Mts. 7.)

Sukupolvi 3 julkaistiin vuonna 2005 Honeynet Projectin toimesta, jolloin myös julkaistiin uusi "hunajamuuri" nimeltä Roo. Tällä kertaa hunajaverkon arkkitehtuuri ei muuttunut, vaan siinä käytetään samaa verkkoarkkitehtuuria kuten sukupolven 2 hunajaverkossa. Uudessa sukupolvessa keskityttiinkin parantamaan sen "hunajamuuria", joka paransi tiedonkeruuta uudella hflow-tietokanta rakenteella sekä pcap-apilla pakettien keruuta varten. Se myös paransi tiedon analysointia web-pohjaisella työkalulla nimeltä walleye. (Mts 8.) "Hunajamuuri" Roo on saatavilla asennettavaksi honeywall CDROM -nimellä kulkevana pakettina, jossa tulee mukana tarvittavat osat sekä työkalut, joilla implementoida "hunajamuuri" (The Honeynet Project s.a.).

Hunajaverkko on myös mahdollista toteuttaa virtuaalisena. Virtuaaliset hunajaverkot voidaan jakaa kahteen kategoriaan: itsenäisiin sekä hybrideihin. Itsenäinen virtuaalinen hunajaverkko perustuu siihen, että koko arkkitehtuuri sijaitsee yhdessä fyysisessä laitteessa virtuaalisena. Laitteessa toimii useita virtuaalikoneita hunajapurkkeina sekä tiedonhallinta ja tiedonkeruu. Tämän toteu-

tuksen hyviin sekä huonoihin puoliin lukeutuu sen sijaitseminen yhdellä fyysisellä laitteella. Toisaalta se tekee ratkaisusta halvan toteuttaa, mutta koska kaikki sijaitsee yhdellä laitteella johtaa sen hajoaminen koko hunajaverkon alasajoon. (Fan 2015, 8.)

Toinen virtualisointitapa on hybridi, jossa tiedonhallinta sekä tiedonkeruu erotetaan toiseen fyysiseen laitteeseen, jolloin virtuaaliset hunajapurkit ovat toisessa ja tiedonhallinta sekä tiedonkeruu toisessa fyysisessä laitteessa. Tämä jakaminen auttaa hunajaverkkoa olemaan vikasietoisempi, joskin toteutus on hieman kalliimpi itsenäiseen verrattuna. Se ei myöskään ole enää yhtä liikuteltava kuin yhden laitteen ratkaisu. (Mts. 9.)

4 HUNAJAPURKIN VALINTA

Hunajapurkkeja voidaan jakaa moneen eri tyyppiin, ja jopa saman tyyppiset eroavat usein toisistaan tarjoamiensa toimintojensa puolesta. On siis tutkimuksen kannalta oleellista perehtyä tutkimaan tarkemmin muutamaa eri vaihtoehtoa, jotta saadaan selville mitä hunajapurkkia tulisi tutkimuksessa käyttää. Voidaan olettaa, että tutkimuksessa on tarpeellista käyttää palvelintyyppin hunajapurkkia, sillä tutkimuksessa ollaan kiinnostuneita lähinnä kohteeseen saapuvista yhteyksistä. Voidaan myös olettaa, että matalan vuorovaikutuksen hunajapurkki on työhön riittävä, joten ei ole tarpeellista tutkia korkean vuorovaikutuksen ohjelmia tarkemmin. Tässä tutkimuksen osiossa tullaan käymään läpi tarkemmin muutama valittu hunajapurkki, niiden toimintoja sekä pohditaan niiden tarjoamaa hyötyä tutkimuksen kannalta. Hunajapurkkien ominaisuudet ja hyödyt tullaan myös taulukoimaan valinnan helpottamiseksi.

4.1 HoneyD

HoneyD on yksi tunnetuimpia ja myöskin vanhimpia hunajapurkkeja. Sen merkittävimpanä ominaisuutena toimii sen virtualisointimahdollisuudet. Se voi samanaikaisesti simuloida kymmeniä tuhansia virtuaalisia isäntiä, testattu korkein määrä on 65 536 joka on B-luokan osoiteavaruuden maksimiosoitte-määrä. Näitä isäntiä voidaan konfiguroida näyttämään siltä, että ne ylläpitävät palveluita, sekä niiden ”sormenjälkiä” voidaan muokata niin että ne vaikuttavat oikeilta käyttöjärjestelmiltä. Kun puhutaan ”sormenjäljistä” tarkoitetaan sitä,

miten eri käyttöjärjestelmät muodostavat vastauksena TCP-yhteyksissä. Eri käyttöjärjestelmät käyttävät muun muassa eri kokoisia paketteja, tai eri TTL-arvoja jolloin voidaan tunnistaa kohteen käyttöjärjestelmä sen ”sormenjäljestä”. Eri käyttöjärjestelmiä voidaan simuloida TCP/IP-tasolla opettelemalla TCP-”sormenjälkiä” esimerkiksi verkon skannausohjelman nmapin ”sormenjälki”-tiedostoista, mikä johtaa yleisten skannausohjelmien kuten mainitun nmapin huijaamiseen. HoneyD ei pelkästään simuloi isäntiä, vaan on sillä mahdollista myös simuloida kokonainen reititystopologia reitittimineen ja reitteineen. Reiteissä voidaan myös simuloida viivettä sekä pakettien menetystä, jotta ne vaikuttaisivat realistisimmalta. Virtuaalikoneita on myös mahdollista tavoittaa ping- sekä traceroute-komennoilla. (Provos 2004.)

4.2 Dionaea

Dionaea on hunajapurkki, jonka tarkoituksena on toimia ansana haittaohjelmille pyrkien saamaan niistä kopion analysointia varten. Dionaea on Nepenthes-hunajapurkin jälkeläinen, pyrkimyksenä on parantaa jotain heikkouksia, joita Nepenthes-hunajapurkissa on esiintynyt kuten puuttuva ipv6-tuki ja siirtyminen C++ -ohjelmointikielestä Pythoniin. (Nawrocki ym. 2016, 7.) Dionaea toimii emuloimalla palveluita kuten MySQL sekä SMB ja niistä löytyviä heikkouksia. Dionaea tarkkailee palveluihin kohdistuvia hyökkäyksiä pyrkien tuottamaan kopion käytetystä haittaohjelmasta. Se käyttää Libemu-kirjastoa hyökkääjien lähettämien haittakuormien havaitsemiseen sekä tutkimiseen. (Tan 2014.) Dionaea mahdollistaa tavallisen tekstitiedoston lisäksi lokien lähettämisen sen ihandler-toiminnolla esimerkiksi VirusTotal-verkkosivustolle (Dionaea 2015). VirusTotal on verkkosivusto, johon lähetetyt tiedostot tarkistetaan muun muassa antivirus-ohjelmien skannereilla tavoitteena saada selville, onko tiedostossa haitallista sisältöä (VirusTotal s.a.).

4.3 KFSensor

KFSensor on hunajapurkki, joka tulee usein vastaan, kun kysytään hyvää vaihtoehtoa Windows-hunajapurkille. Sen grafiikkapohjainen käyttöliittymä takaa helpon käytön sekä selkeän näkymän tapahtumiin. KFSensor toimii pääasiallisesti emuloimalla palveluita kuten esimerkiksi Microsoftin omaa IIS-verk-

kopalvelinta. KFSensor tarjoaa myös mahdollisuuden luoda hälytyksiä hyökkäyksien tapahtuessa. Se tarjoaa mahdollisuuden hälytyksen työpöytäilmoituksen lisäksi myös automaattisen sähköpostiviestin lähettämiseen. Hälytys voidaan lähettää esimerkiksi, kun hyökkäys tapahtuu tai se voidaan rajoittaa lähettämään hälytys vain tiettyntyyppisen hyökkäyksen vuoksi. Viestiin sisältyy hyökkäyksen tiedot kuten IP-osoite, porttinumero sekä ajankohta. Ohjelma on kuitenkin maksullinen, joten ennen sen käyttöönottoa on verrattava sen mahdollisesti tuottamaa hyötyä sen hintaan nähden. Ohjelmaa on mahdollista kuitenkin kokeilla 30 päivän testiversion avulla. Jos työssä tullaan käyttämään Windows-pohjaista järjestelmää, on mahdollista, että KFSensor tulee olemaan hyvä vaihtoehto, joskin on silloin tarpeellista ottaa huomioon tuo 30 päivän kokeiluajan rajoitus. KFSensor tarjoaa mahdollisuuden ottaa lokitiedostot talteen XML-, HTML-, tab separated- ja CSV-muodoissa, joten tuo kokeiluajan rajoitus koskee oikeastaan vain tiedon keräämistä (KFSensor Features s.a.).

4.4 Cowrie

Cowrie on Michael Oosterhofin ylläpitämä Kippo-hunajapurkkiin pohjautuva ja Python-ohjelmointikieleen perustuva ohjelma, joka mahdollistaa Telnet- sekä SSH-yhteyksien tarkemman tutkimisen emuloimalla UNIX-tiedostojärjestelmää, tai toimimalla välipalvelimena tallentaen muihin järjestelmiin läpi kulkevaa liikennettä (Cowrie SSH and Telnet Honeypot s.a.). Esimerkkinä Cowrien voi määrittää toimimaan SSH-portissa 22 jolloin varsinainen SSH-yhteys tulee siirtää toimimaan eri portissa, jotta etäyhteys itse järjestelmään voidaan halutessa säilyttää. Kun hunajapurkki on toiminnassa, kaikki kohteeseen SSH:n avulla yhteyttä ottavat eivät pääse itse kohteeseen, vaan ovat Cowrien emuloimassa SSH-yhteystilassa, jossa hyökkääjällä ei ole yhteyttä varsinaiseen järjestelmään. Tässä tilassa hyökkääjällä on rajoitettu määrä mahdollisuuksia, joskin Cowrie tarjoaa kyllä Debianiin pohjautuvan valetiedostojärjestelmän, jossa on mahdollisuus esimerkiksi luoda ja muokata tiedostoja. (Nawrocki ym. 2016, 9.) Hunajapurkin käyttö on yksinkertaista ja Cowrien dokumentaatio tarjoaa pätevät ohjeet asennukseen sekä asetusten vaihtamiseen. Dokumentaatio tarjoaa myös ohjeita muun muassa MySQL-tietokannan käyttöönottoon, joka puolestaan tarjoaa siistimmän ja helppokäyttöisemmän lokipalvelun Cow-

rielle kuin sen alkuperäinen lokitiedosto, joka on vaikeaselkoinen ilman muutoksia, varsinkin suuremman tietomäärän kanssa. Cowrie vaikuttaa hunajapurkilta, joka olisi työhön sopiva suurimmalta osalta sen dokumentaation kattavuuden sekä tarjoamien ominaisuuksien vuoksi.

4.5 Taulukointi ja valinta

Tässä luvussa tuodaan yhteen hunajapurkkien tietoja sekä hyötyjä taulukon muodossa (taulukko 1). Taulukossa olevat ominaisuudet ovat ne mitä hunajapurkit itse tarjoavat eikä siinä oteta kantaa muiden kuin hunajapurkkien kehittäjien itsensä luomiin ratkaisuihin.

Taulukko 1. Hunajapurkkien ominaisuuksien vertailu

	HoneyD	Dionaea	KFSensor	Cowrie
Avoin lähdekoodi	Kyllä	Kyllä	Ei	Kyllä
Kaupallinen	Ei	Ei	30 päivän ko- keiluaika / Kyllä	Ei
Vuorovaikutustaso	Matala	Matala	Matala	Matala/kor- kea
Lokitiedostotuki	Kyllä	Kyllä	Kyllä	Kyllä
Käyttöjärjestelmätuki	Windows / Linux	Linux	Windows	Linux
Hälytysjärjestelmä	Ei	Ei	Kyllä esim. sähköposti- hälytys	Ei
Erikoistuminen	Virtualisointi	Haittaohjelmien tutkiminen	Allekirjoitus pohjainen tunnistaminen	SSH / telnet yhteyksien tallentaminen
GUI	Ei	Ei	Kyllä	Ei

Tähän työhön lateraalisen liikkeen havaitsemisen testiin valittu hunajapurkki on Cowrie. Valinta muodostui sillä perusteella, että Cowrie on ilmainen käyttää, siitä löytyy tuki lokien tallentamiseen MySQL-tietokantaan sekä tärkeimpänä sen SSH-yhteyksien seuraamisen ominaisuus, joka on olettavasti tärkeä hyökkääjien tarkempia toimia seurattaessa.

5 TYÖN ETENEMINEN

Työ toteutettiin Kaakkois-Suomen ammattikorkeakoulun Kotkan kampuksen ICTLABin tiloissa. Tavoitteena työllä oli pyrkiä löytämään lateraalista liikettä sekä tutkia voidaanko IoT-laitteisiin kohdistuvia uhkia havaita. Jotta tavoitteisiin päästäisiin, päätettiin toteuttaa kaksi eri hunajapurkkia. Toinen hunajapurkeista oli perinteisempi Raspberry Pi -laitteeseen asennettu hunajapurkki, jonka tarkoituksena oli pyrkiä tallentamaan mahdollista lateraalista liikettä sekä muuta tarkempaa tietoa hyökkääjien liikkeistä laitteessa. Toinen hunajapurkki oli IoT-hunajapurkki, jonka pohjana toimi Philips Hue Bridge. Tämän hunajapurkin tarkoituksena oli saada selville IoT-laitteisiin kohdistuvia uhkia. Tässä luvussa käydään läpi edellä mainittujen hunajapurkkien asennusvaiheita sekä testien eri vaiheita.

5.1 ICTLAB hunajapurkki

ICTLABin tuotantoverkkoon sijoitettiin hunajapurkki, jolle annettiin osoite palvelimien aliverkosta, jotta se olisi kohteena mahdollisimman uskottava. Tähän työhön sopivaksi hunajapurkiksi osoittautui Cowrie. Cowrie on matalan vuorovaikutustason palvelintyyppin hunajapurkki, ja se tarjoaa työhön riittävät ominaisuudet. Ilman muutoksia sen tuottamat lokitiedostot eivät ole työhön kuitenkaan riittävät, joten on tarpeellista ottaa käyttöön MySQL-tietokanta. Cowriessa on MySQL-yhteensopivuus, joten on se hyvä valinta lokien sijoittamiselle. Työn tarkoituksena oli asentaa hunajapurkki koulun sisäverkkoon ja tarkkailla siihen saapuvaa tietoliikennettä noin viikon ajan.

5.1.1 Raspberry Pi

Koska työssä ei juurikaan suurempaa suoritustehoa vaadita päätettiin hunajapurkki asentaa Raspberry Pi 4 -laitteeseen, joka koululta jo valmiiksi löytyi. Kuvassa 4 esitetty Raspberry Pi on pienoistietokone, jonka on luonut The Raspberry Pi Foundation tarkoituksenaan tuottaa kustannustehokas tietokone esimerkiksi opetuskäyttöön.



Kuva 4. Raspberry Pi käyttötilanteessa.

Sen tallennustilana toimii SD-muistikortti, jolle myös sen käyttöjärjestelmä asennetaan. Käyttöjärjestelmänä toimii usein jokin Linux-pohjainen järjestelmä kuten Raspbian, joka on virallinen tuettu käyttöjärjestelmä Raspberry-laitteille (Raspbian s.a.).

Asennuksen alkuvaiheessa ongelmaksi osoittautui tarvittavan micro-HDMI-kaapelin puute, joka olisi tarvittu kuvan näyttämiseen laitteesta. Ratkaisuna päätettiin suorittaa niin sanottu hiljainen asennus laitteelle, eli päätettiin muokata muistikortin sisältöä niin että laite suorittaisi käyttöjärjestelmän asennuksen itsenäisesti virran kytkennän jälkeen. Käytettävä muistikortti oli esiasennettu NOOBS-kortti, josta löytyi valmiiksi muutama käyttöjärjestelmä sekä tarvittavat tiedostot asennusta varten. Muistikorttia oli kuitenkin tarvetta muokata hieman, jotta voitaisiin toteuttaa hiljainen asennus. Kortilta löytyvään `recovery.cmdline`-tiedostoon lisättiin komento `silentinstall`, joka aktivoi hiljaisen

asennuksen käynnistyksen yhteydessä. Sen lisäksi oli tarpeellista poistaa kortilta ylimääräiset käyttöjärjestelmät ja jättää vain haluttu käyttöjärjestelmä, jotta hiljainen asennus toimisi halutusti ja asentaisi valitun käyttöjärjestelmän.

Tässä tapauksessa käyttöjärjestelmä oli kortilta valmiiksi löytyvä Raspbian, joka on siis Debian-pohjainen järjestelmä ja soveltuu hyvin hunajapurkiksi.

Tarvetta oli myös muokata cmdline.txt-tiedostoa ja lisätä siihen komento `IP=x.x.x.x`, jossa `x.x.x.x` oli ennalta valittu osoite hunajapurkille. Tämä tehtiin, jotta laitetta olisi mahdollista hallita SSH-yhteydellä asennuksen jälkeen.

Raspberryn turvapäivityksen vuoksi oli tarvetta myös lisätä tyhjä tiedosto nimellä SSH muistikortin boot-kansioon, jotta SSH aktivoituisi (Long 2016).

5.1.2 Hunajapurkin asennus

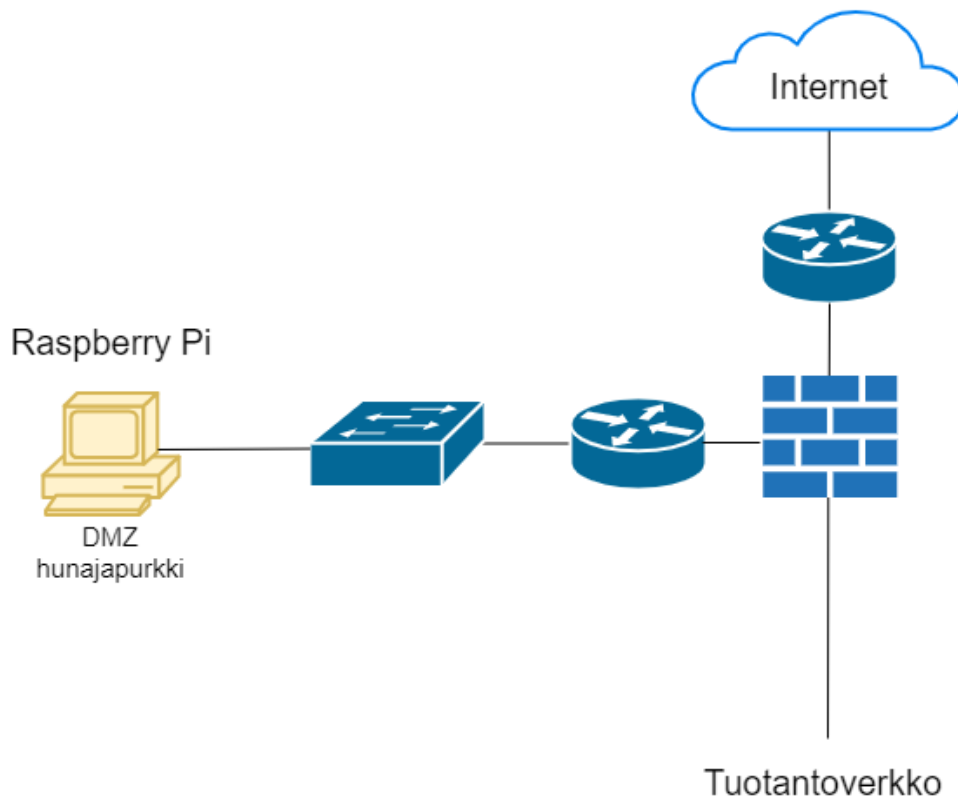
Käyttöjärjestelmän asennuksen jälkeen oli aika asentaa itse hunajapurkki.

Cowrien asennukseen käytettiin sen dokumentoinnista löytyviä asennusohjeita (Ks. Oosterhoof 2018). Tarkemmat ohjeet asennukseen löytyvät liitteenä olevista asennusohjeista (liite 1). Asennuksessa on tärkeää vaihtaa laitteen SSH-palvelu portista 22 johonkin toiseen porttiin, sillä portti 22 tulee jatkossa olemaan hunajapurkin emuloimaan yhteyteen johtava portti. Uusi portti SSH-palvelulle valittiin porteista 49 152–65 535, jotka ovat IANA-organisaation määrittämien yksityiseen tai tilapäiseen käyttöön tarkoitettuja portteja (Service Name and Transport Protocol Port Number Registry 2020). Tässä tapauksessa SSH siirrettiin porttiin 64 295. Itse portin siirto suoritettiin muokkaamalla Raspbianista löytyvää `/etc/ssh/sshd_config`-tiedostoa, jonka jälkeen `ssh service restart` -komennolla muutos saatiin voimaan. Laitteen SSH-portin vaihdon jälkeen voidaan porttiin 22 tulevat SSH-yhteydet uudelleenohjata porttiin 2 222 joka on Cowrien käyttämä portti. Tämä uudelleenohjaus toteutuu komentorivin komennolla `iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-port 2222`. Cowriessa itsessään on valmiiksi määritetty SSH päälle, joten ei sen asetustiedostoa tarvitse tässä vaiheessa muokata. Tämä voidaan vielä varmistaa komennolla `sudo netstat -tulpn | grep LISTEN`, joka näyttää käytetyt portit sekä niitä käyttävät prosessit.

Ilman muutoksia Cowrie tallentaa lokitiedostonsa omaan cowrie.log-tiedostoon, mutta hiemankin suuremman tietomäärän lukeminen siitä vaatii tiedoston muokkaamista. Toinen vaihtoehto, jota myös tässä työssä käytettiin, on ottaa tapahtumat talteen tietokantaan. Tietokanta on jokin järjestelmällinen informaation tallennuskohde, kuten esimerkiksi matkapuhelimesta löytyvä valokuvagalleria. MySQL on avoimeen lähdekoodiin perustuva tietokantojen hallitsemisjärjestelmä, joka on saanut alkunsa jo 90-luvun puolivälissä. Sen käyttäjäkuntaan kuuluvat tänä päivänä muun muassa sellaiset yritykset kuten Google, Facebook ja Cisco (MySQL s.a.). MySQL:n Cowrieen liittämiseen löytyy myös tarvittavat ohjeet Cowrien dokumentoinnista (Ks. Oosterhoof 2018b). Hieman ohjeista poiketen käytettiin työssä kuitenkin MariaDB-tietokantaa, joka on MySQL:n tekijöiden uusi alkuperäisestä haarautunut versio ja teoriassa melkein täysin identtinen lukuun ottaen käytettävät komennot. Tietokantaan luotiin käyttäjä sekä itse tietokanta Cowrie-nimellä. Itse Cowrien asetustiedoston muokkaus oli tässä vaiheessa myös tarpeellista, ja se toteutettiin luomalla uusi tiedosto cowrie.cfg, johon lisättiin tietokannan tiedot. Tämän jälkeen pikaisen Cowrien uudelleenkäynnistämisen jälkeen oli tietokanta toiminnassa ja valmiina taltioimaan tapahtumat. Tietokanta taltioi tietoa kätevästi omiin kategorioihin kuten esimerkiksi kirjautumistiedot ja käyttäjän syötteet, joista tietoa on helppo hakea ja tarvittaessa suodattaa vielä lisää.

5.1.3 Julkinen hunajapurkki

Alkuperäisen suunnitelman mukaan hunajapurkki oli ICTLABin palvelinten aliverkossa tuotantoverkon puolella. Viikon tiedonkeräyksen jälkeen voitiin todeta, että tuotantoverkossa ei liikkunut epätavallista liikennettä tai sitä ei pystytty tutkimuksen aikana havaitsemaan. Päätös oli siirtää hunajapurkki julkisen IP-osoitteen alle DMZ:lle, jotta voitiin alkaa keräämään tutkimuksen kannalta oleellista tietoa (kuva 5).



Kuva 5. Raspberry Pi verkon DMZ:lla

Julkisen IP-osoitteen alla ollessa, kuten odottaa saattoi, alkoi hunajapurkki välittömästi saada liikennettä. Ensimmäinen yhteys havaittiin jo sekunteja laitteen yhteyden avaamisen jälkeen. Monet yhteyksistä vaikuttivat automatisoitujen bottien toteuttamilta hyökkäyksiltä, sillä yhteydet kestivät vain muutaman sekunnin ajan. Tavoitteena yhteyksien aikana vaikutti suurilta osin olevan saada ladattua jokin haitallinen tiedosto laitteelle, sekä saada tarkempaa tietoa laitteesta. Kuvassa 6 voidaan nähdä, miten tietoa pyrittiin saamaan esimerkiksi komennolla `uname`, joka on komento, jolla voidaan saada kernelistä tarkempaa tietoa kuten versionumero, nimi tai vaikka julkaisupäivämäärä (Linux Information Project 2006).

38	c0282d71d067	2020-01-31 17:52:07	NULL	1	uname	
39	c0282d71d067	2020-01-31 17:52:08	NULL	1	free -m	
40	c0282d71d067	2020-01-31 17:52:10	NULL	1	ps -x	
41	c0282d71d067	2020-01-31 17:52:12	NULL	1	cat /proc/cpuinfo	
42	9974ecf874d8	2020-01-31 22:36:31	NULL	1	echo "cd /tmp; wget http://46.246.37.212/wget.sh curl http://46.246.37.212/curl.sh -o curl.sh; chmod +x *.sh; ./wget.sh; ./curl.sh" sh	
43	9974ecf874d8	2020-01-31 22:36:31	NULL	1	cd /tmp; wget http://46.246.37.212/wget.sh curl http://46.246.37.212/curl.sh -o curl.sh; chmod +x *.sh; ./wget.sh; ./curl.sh	

Kuva 6. MySQL tietokantaan tallentuneet hyökkäystiedot

Tietoa etsittiin myös tiedostosta `cpuinfo`, joka tarjoaa tarkat tiedot laitteen prosessorista kuten valmistajan ja ydinten määrän. National Cyber Security Centren mukaan (2018b) tiedon kerääminen kohteesta on ensimmäinen askel lateraalisessa liikkeessä, mutta muita merkkejä, kuten muun verkon tutkimista, siitä ei kuitenkaan lokeista paljastunut. Voidaan siis todeta, että lateraalista liikettä on mahdollisesti aloitettu, mutta hyökkääjä on joko todennut laitteen hunajapurkiksi tai hunajapurkki ei tarjonnut hyökkääjälle mahdollisuutta jatkaa eteenpäin.

5.2 IoT-Hunajapurkki

IoT-hunajapurkin osalta tarkoituksena on tutkia Philips Hue Bridgeen kohdistuvaa liikennettä, ja selvittää onko mahdollista havaita haitallista liikennettä. Suunnitelmana on löytää tutkimukseen soveltuva tapa tutkia laitteeseen kohdistuvaa liikennettä, jonka jälkeen Hue kytketään julkiseen verkkoon, jossa siihen kohdistuvaa liikennettä tarkkaillaan noin viikon ajalta.

5.2.1 Kontrolloitu testi

Philips Hue Bridge kytkettiin ensin Virtual Lab -ympäristöön, jossa voitiin kontrolloidussa ympäristössä tarkkailla tietoliikennettä, jota laite verkkoon tuottaa ja pohtia suunnitelmaa siihen kohdistuvan tietoliikenteen tarkkailuun. Laitteen tietoliikennettä tarkkailtiin Virtual Labista löytyvän cable tap monitor -ominaisuuden avulla, joka siis avaa valitun yhteyden väliin Wireshark-pakettianalysoijan, jolla voidaan tutkia siinä kulkevaa tietoliikennettä. Tällä tavoin voidaan

nähdä kaikki Huen tietoliikenne ja halutessa myös tallentaa se. Kuten oletettua Philips Hue Bridge ei itsessään tuottanut juurikaan suurempaa tietoliikennettä verkkoon. Testin aikana saatiin tuotettua suunnitelma siihen kohdistuvan tietoliikenteen tarkkailuun ja tallentamiseen. Suunnitelmaksi muodostui käyttää Ethernet LAN tap -laitetta ja liittää siihen esimerkiksi tutkimuksessa aiemmin käytetty Raspberry Pi. Raspberryllä voidaan käyttää pakettianalysijaa kuten Wiresharkia tutkimaan tietoliikennettä sekä tallentamaan tietoliikennettä myöhempää analysointia varten.

5.2.2 Julkisen verkon testi

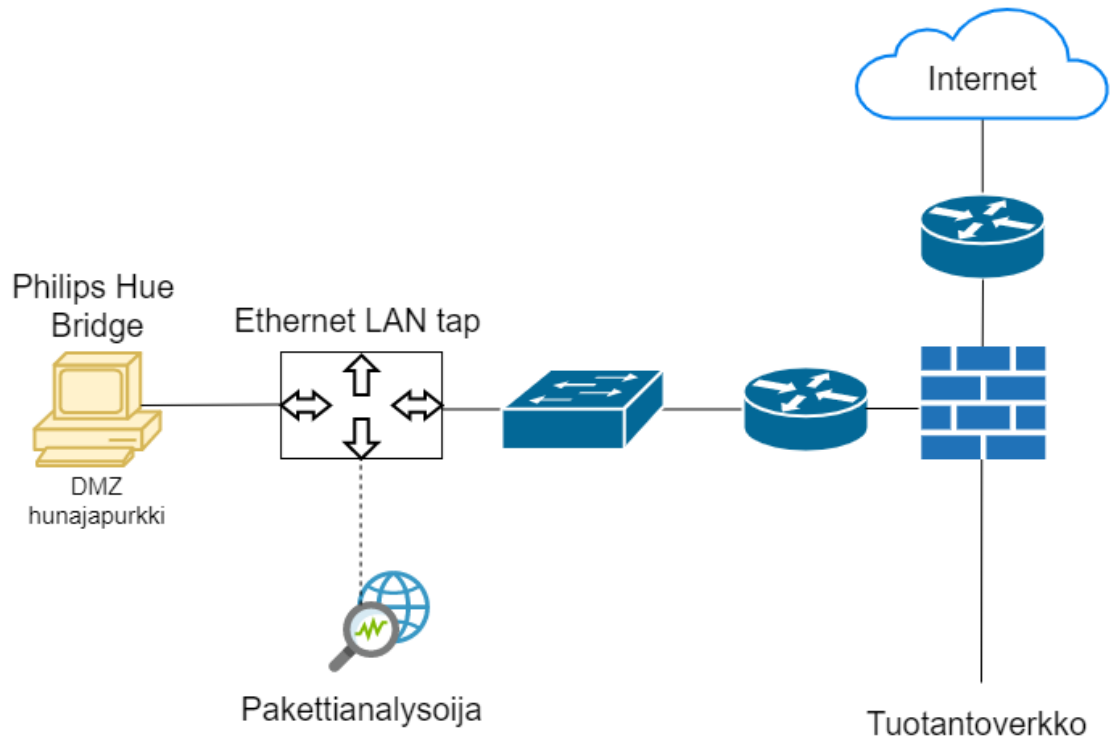
Kontrolloidun testin jälkeen Philips Hue Bridge siirrettiin julkiseen verkkoon DMZ:lle, jossa siihen kohdistuvaa liikennettä tarkkailtiin Ethernet LAN tapin avuin (kuva 7).



Kuva 7. Ethernet LAN tap

Ethernet LAN tap on laite, joka sijoitetaan fyysisen verkkoyhteyden väliin missä se jakaa yhteydessä kulkevan liikenteen kahteen ylimääräiseen niin sanottuun monitor porttiin eli valvontaporttiin. Valvontaportti on portti, johon tarkkailtava yhteys peilataan ja josta sitä voidaan tarkemmin tutkia kolmantena osapuolena. Valvontaportti jakaa liikenteen kolmannelle osapuolelle, mutta se

ei jaa liikennettä kolmannelta osapuolelta takaisin verkkoliikenteeseen. Näin voidaan pitää valvontaportissa kiinni oleva laite näkymättömänä tutkittavalle verkolle. (Hak5 s.a.) Tuohon valvontaporttiin voidaan kytkeä haluttu laite, vaikka aiemmin tutkimuksessa käytetty Raspberry Pi, jolla voidaan tutkia yhteydessä kulkevaa tietoliikennettä esimerkiksi Wiresharkin tai muun vastaavan pakettianalysoidin avulla (kuva 8).



Kuva 8. Ethernet LAN tapin sijoitus verkossa

Tutkimuksessa käytettäväksi ohjelmaksi valittiin tcpdump, joka on komentorivillä toimiva erittäin muokattavissa oleva pakettianalysoidin. Tcpdump tulostaa vakioasetuksillaan, eli pelkällä tcpdump komennolla, ruudulle verkkoliikenteeseen saapuvat ja siitä lähtevät paketit, sekä niiden saapumis- tai lähtöajat sekunnin murto-osan tarkkuudella. Tcpdump tukee vaihtoehtoisia lisäkomentoja, joiden avulla on mahdollista muokata ohjelman käyttäytymistä. Voidaan esimerkiksi määrittää ohjelmalle -w-lisäkomento, joka tallentaa verkkoliikenteen tiedoston myöhempää analyysiä varten (The Tcpdump group 2020). Tiedoston tallentaminen muodossa pcap mahdollistaa sen tutkimisen myös muilla pakettianalysoidinilla, kuten Wiresharkilla. Lopulliseksi komennoksi muodostui tcpdump host x.x.x.x -U -w - | tee huelog.pcap | tcpdump -r -, jossa x.x.x.x oli tarkkailtavan kohteen IP-osoite. Komento mahdollisti kohteeseen saapuvan liikenteen

tarkkailun reaaliajassa, sekä sen tallentamisen myöhempää tarkempaa analysointia varten. Kuvassa 9 voidaan nähdä sen reaaliajassa tuottamaa tietoa kohteen tietoliikenteestä.

```

pi@raspberrypi:~
File Edit Tabs Help
tcpdump 0,nop,nop,sackOK], length 0
11:59:36.269287 IP 92.63.196.7.1211 > 193.167.61.198.6668: Flags [S], seq 4022454483, win 200, options [mss 1300,nop,wsc
le 0,nop,nop,sackOK], length 0
11:59:36.878016 IP 92.63.196.7.1211 > 193.167.61.198.6668: Flags [S], seq 3161661493, win 200, options [mss 1300,nop,nop,
sackOK], length 0
11:59:38.450693 IP 184.178.158.12.60711 > 193.167.61.198.microsoft-ds: Flags [S], seq 1044315813, win 8192, options [mss
1300,nop,wscale 2,nop,nop,sackOK], length 0
11:59:39.109649 IP 184.178.158.12.64807 > 193.167.61.198.microsoft-ds: Flags [S], seq 1302594423, win 8192, options [mss
1300,nop,wscale 2,nop,nop,sackOK], length 0
11:59:39.764705 IP 184.178.158.12.23847 > 193.167.61.198.microsoft-ds: Flags [S], seq 1360179431, win 65535, options [mss
1300,nop,nop,sackOK], length 0
11:59:41.369272 IP 34.90.173.53.https > 193.167.61.198.36278: Flags [F], ack 79, win 251, length 0
11:59:59.539720 IP 216.239.35.8.ntp > 193.167.61.198.46738: NTPv4, Server, length 48
12:00:04.525214 ARP, Reply 193.167.61.193 is-at 00:e0:ed:2e:d7:9e (oui Unknown), length 46
12:00:08.066966 IP 180.189.153.249.53291 > 193.167.61.198.microsoft-ds: Flags [S], seq 3508366046, win 8192, options [mss
1300,nop,wscale 2,nop,nop,sackOK], length 0
12:00:08.913722 IP 180.189.153.249.53291 > 193.167.61.198.microsoft-ds: Flags [S], seq 3095075453, win 8192, options [mss
1300,nop,wscale 2,nop,nop,sackOK], length 0
12:00:11.449865 IP 34.90.173.53.https > 193.167.61.198.36278: Flags [F], ack 79, win 251, length 0
12:00:17.343217 IP 177.93.247.16.56188 > 193.167.61.198.microsoft-ds: Flags [S], seq 2986108074, win 8192, options [mss 1
300,nop,wscale 2,nop,nop,sackOK], length 0
12:00:18.090208 IP 177.93.247.16.56188 > 193.167.61.198.microsoft-ds: Flags [S], seq 1852343214, win 8192, options [mss 1
300,nop,wscale 2,nop,nop,sackOK], length 0
12:00:21.707710 IP 187.11.252.242.61373 > 193.167.61.198.microsoft-ds: Flags [S], seq 3212273788, win 64800, options [mss
1300], length 0
12:00:23.597715 IP 216.239.35.12.ntp > 193.167.61.198.46897: NTPv4, Server, length 48
12:00:26.203225 IP 217.197.225.38.50590 > 193.167.61.198.microsoft-ds: Flags [S], seq 3842076299, win 8192, options [mss
1300,nop,wscale 2,nop,nop,sackOK], length 0
12:00:26.725297 IP 217.197.225.38.50590 > 193.167.61.198.microsoft-ds: Flags [S], seq 3349936301, win 8192, options [mss
1300,nop,wscale 2,nop,nop,sackOK], length 0
12:00:27.644389 IP 216.239.35.0.ntp > 193.167.61.198.58476: NTPv4, Server, length 48
12:00:28.667676 IP 216.239.35.4.ntp > 193.167.61.198.50917: NTPv4, Server, length 48
12:00:38.355671 IP 200.53.28.238.27990 > 193.167.61.198.microsoft-ds: Flags [S], seq 2161896646, win 8192, options [mss 1
300,nop,wscale 2,nop,nop,sackOK], length 0

```

Kuva 9. tcpdump reaaliaikaista tietoa

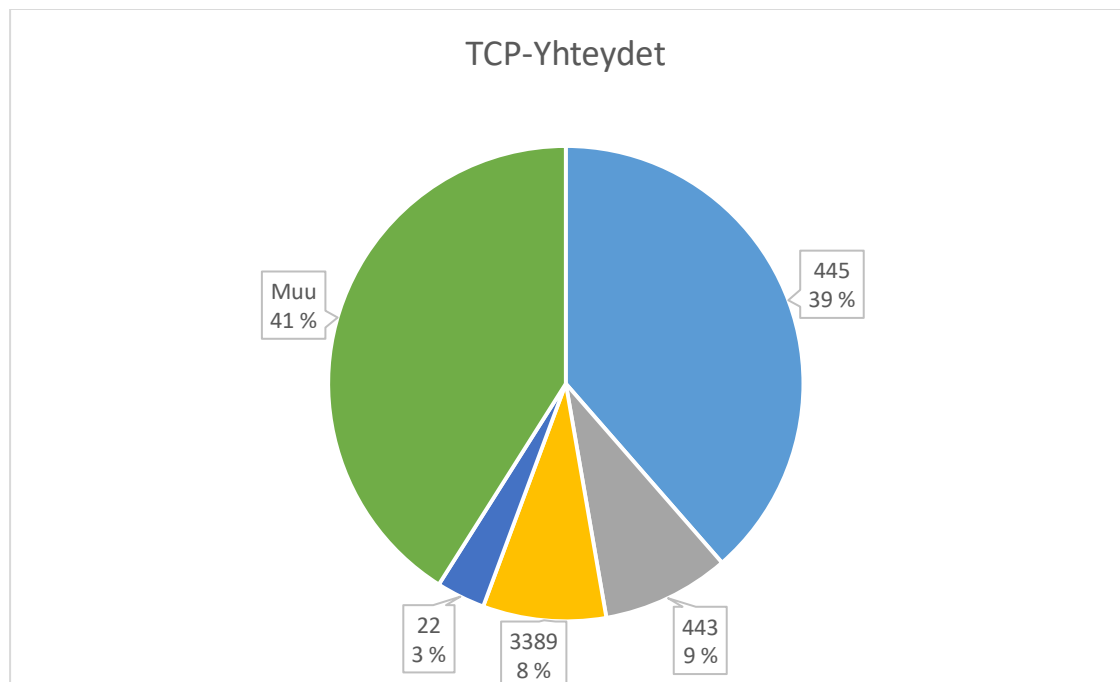
Tiedon keräämisen jälkeen voitiin tcpdumpin tuottamaa pcap-tiedostoa lähteä analysoimaan tarkemmin Wiresharkin avulla. Wireshark mahdollistaa tallenne-
tun tietoliikenteen tutkimisen yhteyskohtaisesti, ja jokaisesta yhteydestä on
mahdollista saada selville paljon yksityiskohtaista tietoa. Tästä esimerkkinä
kuvassa 10 on ote kerätystä tiedosta avattuna Wiresharkissa.

Time	Source	Destination	Protocol	Host	Info
2020-03-11 06:20:34,014690	120.92.123.150	193.167.61.198	TCP		45598 → 8080 [SYN] Seq=0
2020-03-11 06:20:35,013668	120.92.123.150	193.167.61.198	TCP		[TCP Retransmission] 455
2020-03-11 06:20:35,339558	120.92.123.150	193.167.61.198	TCP		45598 → 8080 [ACK] Seq=1
2020-03-11 06:20:35,340569	120.92.123.150	193.167.61.198	HTTP	193.167.61.198:8080	POST /users?page=&size=5
2020-03-11 06:20:35,344087	120.92.123.150	193.167.61.198	TCP		[TCP Dup ACK 2123#1] 45
2020-03-11 06:20:35,671323	120.92.123.150	193.167.61.198	TCP		45598 → 8080 [ACK] Seq=3
2020-03-11 06:20:35,671438	120.92.123.150	193.167.61.198	TCP		45598 → 8080 [FIN, ACK]
2020-03-11 06:20:36,664563	120.92.123.150	193.167.61.198	TCP		[TCP Retransmission] 455
2020-03-11 06:20:36,709446	120.92.123.150	193.167.61.198	TCP		45598 → 8080 [ACK] Seq=3

Wireshark · Follow TCP Stream (tcp.stream eq 14726) · huelog.pcap	
POST /users?page=&size=5 HTTP/1.1	
Host: 193.167.61.198:8080	
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.0;en-US; rv:1.9.2) Gecko/20100115 Firefox/3.6)	
Content-Length: 119	
Connection: close	
Content-Type: application/x-www-form-urlencoded	
Accept-Encoding: gzip	
username[&this.getClass().forName("java.lang.Runtime").getRuntime().exec("touch /tmp/su")]=&password=&repeatedPassword=	

Kuva 10. Ote Wiresharkista

Otteessa on näkyvissä yhteys, jossa on pyritty toteuttamaan CVE-2018-1273-nimellä kulkeva hyökkäys, jossa tavoitteena on saada Spring Data REST -haavoittuvuutta hyväksi käyttämällä suoritettua haitallista koodia kohteessa (National Institute of Standards and Technology 2019). Wiresharkista saatujen tietojen avulla voidaan myös visualisoida hunajapurkkiin kohdistuneita yhteyksiä. Kuvassa 11 esitetty ympyräkaavio sisältää visualisoinnin saapuneista TCP-yhteyksistä sekä niiden kohdeporteista.



Kuva 11. TCP yhteyksien kohdeportit visualisoituna

Yhteensä TCP-yhteyksiä havaittiin noin 28 000, joista 10 802 kohdistui porttiin 445, joka on Windows Server Message Block -tiedostonjakopalvelun (SMB) käytössä oleva portti (IANA 2020). Portti 445 on suosiossa erityisesti tunnetun WannaCry-lunnasohjelmiston kohteena ja hyvästä syystä, sillä vuonna 2017 tehdyssä tutkimuksessa havaittiin yli miljoona laitetta, joissa portti 445 oli avoinna internetiin (Hodgman 2017). Muita huomioon otettavia portteja on muun muassa 443, joka on HTTPS-liikenteen portti, 3 389 joka on Microsoft WBT Server -portti tarkoitettu esimerkiksi etäkäyttöyhteyksille sekä 22, joka on SSH palvelun portti (IANA 2020).

Pcap-tiedostoa on mahdollista myös tarkkailla käyttämällä esimerkiksi PacketTotalin tarjoamaa analysointipalvelua. PacketTotal käyttää hyödykseen avoimeen lähdekoodiin perustuvia Zeek- sekä Suricata IDS -järjestelmiä tiedon

analysoimiseen, sekä Elasticsearchia tiedon varastointiin sekä noutamiseen (PacketTotal s.a.). Se tarjoaa mahdollisuuden selainpohjaiseen pakettianalysointiin, jossa käyttäjän syöttämä pcap-tiedosto kategorisoidaan sekä visualisoidaan esimerkiksi pylväskaavioiden avulla.

6 TULOKSET JA PÄÄTELMÄT

Testien tarkoituksena oli saada kerättyä tietoa lateraalisesta liikkeestä ja hyökkääjien tavoista toimia, sekä pohtia tapaa havaita IoT-laitteisiin kohdistuvia uhkia. Ensimmäisessä testissä päätavoitteena oli lateraalisen liikkeen havainnointi, ja tarkoituksena oli pystyä havaitsemaan tarkemmin hyökkääjien liikkeitä kohteessa. Hyökkääjien liikkeitä onnistuttiin testissä havaitsemaan, mutta päätavoitetta ei testin aikana saavutettu, eli lateraalista liikettä ei havaittu. Tämä voi johtua siitä, että tietoa kerättiin lyhyemmältä ajalta kuin olisi tarvittu, tai myös siitä miten tietoa kerättiin. Vaikka testiin valittu hunajapurkki mahdollisti hyökkääjän toimimisen kohteessa emuloidussa tilassa voi olla, että se ei tarjonnut riittäviä ominaisuuksia lateraalisen liikkeen aloittamiseen. Testi oli lateraalisen liikkeen havaitsemisen kannalta siis osittain onnistunut, sillä voidaan todeta, että vaikka itse tutkimuksessa ei sitä päästy havaitsemaan kokonaisuudessaan, on hunajapurkin avuin todennäköisesti mahdollista sitä havaita. On kuitenkin todettava, että tarkempaa tietoa haluttaessa on mahdollisesti tarjottava hyökkääjälle enemmän mahdollisuuksia kohteessa, jotta olisi mahdollista saada tarkempaa tietoa hyökkääjien tavoista toimia. Vaihtoehtona jatkotutkimukselle voisi olla jonkin tyyppisen hunajaverkon rakentaminen, jossa voitaisiin todeta lateraalisen liikkeen kaikki vaiheet sekä mahdollisesti siirtyminen laitteesta laitteeseen.

Toisessa testissä tavoitteeksi oli asetettu IoT-laitteisiin kohdistuvien uhkien havaitseminen. Testi aloitettiin keräämällä tietoa siitä, kuinka IoT-laitteisiin tulevaa tietoliikennettä voitaisiin ylipäätään tarkkailla ja pohdinnan jälkeen päädyttiin tapaan, jossa voitiin kolmantena osapuolena tietoliikennettä tutkia. Tällä tavalla tietoliikennettä tutkiessa ei voida toteuttaa samankaltaista hunajapurkkia kuten edellisessä testissä, jossa hyökkääjällä olisi mahdollisuus toimia kohteessa. Voidaan kuitenkin nähdä kaikki tietoliikenne kohteeseen, ja sitä analysoimalla pyrkiä erottamaan haitallinen tietoliikenne normaalista. Testi oli

onnistunut ja IoT-laitteeseen kohdistuvaa tietoliikennettä pystyttiin tarkkailemaan. Jatkotutkimuksena voisi esimerkiksi pohtia miten tuota tietoliikennettä voitaisiin analysoida tarkasti, tehokkaasti ja mahdollisesti automatisoidusti erottaen haitallisen liikenteen normaalista.

Testien päätyttyä voitiin saatujen tulosten avulla selvittää vastaus teoriaosuudessa määritettyihin tutkimuskysymyksiin.

- Minkälaisia hunajapurkkeja voidaan tutkimuksessa hyödyntää?

Tutkimuksessa voitiin hyödyntää lähinnä palvelintyyppisiä hunajapurkkeja, sillä tavoitteena oli nähdä, miten hyökkääjä toimii kohteessa ja saada sitä kautta kerättyä tutkimuksen kannalta oleellista tietoa. Matalan vuorovaikutuksen hunajapurkki oli tutkimukseen tällä kertaa riittävä, joskin täytyy mainita, että korkean vuorovaikutuksen hunajapurkillä olisi mahdollisesti voitu saada vielä tarkempia tuloksia. Tämä lähinnä johtuu siitä, että matalan vuorovaikutuksen hunajapurkki ei välttämättä antanut hyökkääjille tarpeeksi tilaa kohteessa, jotta lateraalista liikettä olisi ollut mahdollista suorittaa. Tärkeää hunajapurkissa tutkimuksen kannalta oli myös se, että siitä saataisiin mahdollisimman yksityiskohtaiset lokitiedostot. Mitä yksityiskohtaisemmat lokitiedostot ovat, sitä todennäköisempää on havaita hyökkääjien lopulliset aiheet verkossa.

- Kuinka hyökkääjä toimii kohteessa?

Useimmat hyökkääjistä olivat selvästi automatisoituja botteja, joiden tehtävänä oli lähinnä saada ladattua kohteeseen jokin haitallinen tiedosto ja sen kautta joko saada kohde liitettyä bottiverkkoon tai saada lisää tietoa kohteen verkosta. Muutamissa hyökkäyksissä tarkoituksena tuntui olevan saada lisää tietoa itse kohteesta, kuten prosessorin tai kernelin tarkempia tietoja. Kuitenkaan yhdessäkään hyökkäyksessä ei pyritty saamaan tietoa kohteen verkosta, mikä voi johtua osittain myös hunajapurkin tuomista rajoituksista komennoissa ja toiminnoissa.

- Kuinka tutkimuksesta saatua tietoa voidaan hyödyntää kyberturvallisuuden kehittämisessä?

Voidaan todeta, että hunajapurkit ovat hyvä lisä tehostamaan verkon turvallisuutta. Vaikka ne eivät suoranaisesti hyökkääjää pysäytä niistä saatavan tiedon avulla voidaan havaita haavoittuvuuksia, joiden tunnistaminen muutoin olisi aikaa vievää, ellei jopa mahdotonta. Hunajapurkki itsessään on myös erittäin laaja käsite nykypäivänä ja varmaa on, että jokaiseen tilanteeseen löytyy sopiva tapa toteuttaa sellainen. Ja vaikka tämä tutkimus ei suoraan vastausta tarjoaisi, antaa se toivottavasti käsitystä siihen miten oikeanlaista hunajapurkia voisi etsimään lähteä.

Tämän työn tavoitteena oli pohtia hunajapurkeista saatavaa hyötyä sekä niiden käyttöä nykypäivänä kyberturvallisuudessa. Tunnen, että vaikka työssä saavutettiin joitain tavoitteita, jäi osa kuitenkin vaille tarkempaa vastausta. Aiheesta jää paljon tilaa jatkotutkimuksille, mikä voidaan kuitenkin nähdä positiivisena asiana. Näitä jatkotutkimusaiheita ovat mainitut hunajaverkot sekä haitallisen liikenteen tarkempi ja tehokkaampi analysointitapa. Luulen kuitenkin, että tutkimusta ja sen tuloksia on mahdollista käyttää joko jatkotutkimuksia suunnitellessa, tai muuten hunajapurkkien käyttöä pohtiessa. Voidaan siis todeta, että opinnäytetyö oli onnistunut ja vaikka kaikkia tavoitteita ei täysin työssä saavutettu saatiin silti tuotettua aikaan tietoa, jonka avulla nuo tavoitteet voitaisiin mahdollisesti jatkossa saavuttaa.

LÄHTEET

Bromley, K. 2016. Inline and Out-of-Band: The ABCs of Network Visibility. Blogi. Saatavissa: <https://www.ixiacom.com/company/blog/network-visibility> [viitattu 27.3.2020].

CISA. s.a. Control System Security DMZ. WWW-dokumentti. Saatavissa: https://www.us-cert.gov/ics/Control_System_Security_DMZ-Definition.html [viitattu 26.3.2020].

Cisco. s.a. What Is a Firewall? WWW-dokumentti. Saatavissa: <https://www.cisco.com/c/en/us/products/security/firewalls/what-is-a-firewall.html> [viitattu 28.3.2020].

Coty, S. 2013. Attacking Kippo. Blogi. Saatavissa: <https://blog.alertlogic.com/blog/attacking-kippo/> [viitattu 29.3.2020].

Cowrie. s.a. Cowrie SSH and Telnet Honeypot. WWW-dokumentti. Saatavissa: <https://www.cowrie.org/> [viitattu 7.3.2020].

Dionaea. 2015. Logging (ihandler). WWW-dokumentti. Saatavissa: <https://dionaea.readthedocs.io/en/latest/ihandler/index.html> [viitattu 29.3.2020].

Fan, W., Fernández., D. & Du, Z. 2015. Adaptive and Flexible Virtual Honeynet. PDF-dokumentti. Saatavissa: https://www.researchgate.net/publication/285598988_Adaptive_and_Flexible_Virtual_Honeynet [viitattu 28.3.2020].

Fu, X., Yu, W., Cheng, D., Tan., X., Streff, K. & Graham, S. 2006. On Recognizing Virtual Honeypots and Countermeasures. PDF-dokumentti. Saatavissa: <https://ieeexplore.ieee.org/Xplore/home.jsp> [viitattu 26.3.2020].

Grimes, R. 2005. Honeypots for Windows. E-kirja. Saatavissa: <https://books.google.fi/> [viitattu 26.3.2020].

Göbel, JG. & Dewald, A. 2010. Client-Honeypots: Exploring Malicious Websites. E-Kirja. Berliini: Walter de Gruyter GmbH. Saatavissa: <https://kaakkuri.finna.fi> [viitattu 2.3.2020].

Hak5. s.a. THROWING STAR LAN TAP. WWW-dokumentti. Saatavissa: <https://shop.hak5.org/products/throwing-star-LAN-tap> [viitattu 27.3.2020].

Heikkilä, T. 2014. Tilastollinen tutkimus. E-kirja. Edita Publishing Oy. Saatavissa: <https://kaakkuri.finna.fi> [viitattu 27.3.2020].

Hodgman, R. 2017. WannaCry Update: Vulnerable SMB Shares Are Widely Deployed And People Are Scanning For Them (Port 445 Exploit). Blogi. Saatavissa: <https://blog.rapid7.com/2017/05/16/update-on-wannacry-vulnerable-smb-shares-are-widely-deployed-and-people-are-scanning-for-them/> [viitattu 26.3.2020].

IANA. 2020. Service Name and Transport Protocol Port Number Registry. WWW-dokumentti. Saatavissa: <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml> [viitattu 7.3.2020].

Ixia. s.a. Bypass Switches. WWW-dokumentti. Saatavissa: <https://www.ixia.com.com/products/iBypass> [viitattu 27.3.2020].

Juniper. 2019. Understanding Port Mirroring. WWW-dokumentti. Saatavissa: https://www.juniper.net/documentation/en_US/junos/topics/concept/port-mirroring-qfx-series-understanding.html [viitattu 26.3.2020].

Juniper. s.a. What is IDS and IPS? WWW-dokumentti. Saatavissa: <https://www.juniper.net/us/en/products-services/what-is/ids-ips/> [viitattu 7.3.2020].

Kananen, J. 2014. Toimintatutkimus kehittämistutkimuksen muotona: Miten kirjoitan toimintatutkimuksen opinnäytetyönä? E-kirja. Suomen Yliopistopaino Oy: Juvenes Print. Saatavissa: <https://kaakkuri.finna.fi> [viitattu 25.3.2020].

KFSensor. s.a. KFSensor Features. WWW-dokumentti. Saatavissa: <http://www.keyfocus.net/kfsensor/features/> [viitattu 7.3.2020].

Linux Information Project. 2005. The uname Command. WWW-dokumentti. Päivitetty 1.6.2006. Saatavissa: <http://www.linfo.org/uname.html>. [viitattu 7.3.2020].

Long, S. 2016. A security update for Raspbian PIXEL. Blog. Saatavissa: <https://www.raspberrypi.org/blog/a-security-update-for-raspbian-pixel/> [viitattu 7.3.2020].

Mansoori, M., Welch, I. & Fu, Q. 2014. YALIH, yet another low interaction honeyclient. PDF-Dokumentti. Saatavissa: https://www.researchgate.net/publication/263507637_YALIH_yet_another_low_interaction_honeyclient [viitattu 7.3.2020].

MySQL. s.a. WWW-dokumentti. Saatavissa: <https://www.mysql.com/> [viitattu 7.3.2020].

National Cyber Security Centre. 2018. Preventing Lateral Movement. WWW-dokumentti. Saatavissa: <https://www.ncsc.gov.uk/guidance/preventing-lateral-movement> [viitattu 7.3.2020].

National Cyber Security Centre. 2018b. Preventing Lateral Movement. WWW-dokumentti. Saatavissa: <https://www.ncsc.gov.uk/guidance/preventing-lateral-movement> [viitattu 7.3.2020].

National Institute of Standards and Technology. 2018. CVE-2018-1273 Detail. WWW-dokumentti. päivitetty 2019. Saatavissa: <https://nvd.nist.gov/vuln/detail/CVE-2018-1273#vulnCurrentDescriptionTitle> [viitattu 11.3.2020].

Nawrocki, M. Wählisch, M. Schmidt, T. C. Keil, C. & Schönfelder, J. 2016. A survey on honeypot software and data analysis. PDF-dokumentti. Saatavissa: <https://arxiv.org/pdf/1608.06249.pdf> [viitattu 29.11.2019].

Oosterhoof, M. 2018. Installing Cowrie in seven steps. WWW-dokumentti. Saatavissa: <https://cowrie.readthedocs.io/en/latest/INSTALL.html> [viitattu 7.3.2020].

Oosterhoof, M. 2018b. How to Send Cowrie Output to a MySQL Database. WWW-dokumentti. Saatavissa: <https://cowrie.readthedocs.io/en/latest/sql/README.html> [viitattu 7.3.2020].

PacketTotal. s.a. docs. WWW-dokumentti. Saatavissa: <http://docs.packetotal.com/> [viitattu 12.3.2020].

Ioulidou, P., Vassilakis, V., Moscholios, I. & Logothetis, M. 2018. A Signature-based Intrusion Detection System for the Internet of Things. PDF-dokumentti. Saatavissa: https://www.researchgate.net/publication/326376629_A_Signature-based_Intrusion_Detection_System_for_the_Internet_of_Things [viitattu 28.3.2020].

Pitkäranta, A. 2014. Laadullinen tutkimus opinnäytetyönä. E-kirja. e-Oppi Oy. Saatavissa: <https://kaakkuri.finna.fi> [viitattu 27.3.2020].

Provos, N. 2004. HoneyD General Information. WWW-dokumentti. Saatavissa: <http://www.honeyd.org/general.php> [viitattu 28.3.2020].

Raspberry Pi Foundation. s.a. Raspbian. WWW-dokumentti. Saatavissa: <https://www.raspberrypi.org/downloads/raspbian/> [viitattu 7.3.2020].

Rouse, M. 2007. honeynet. WWW-dokumentti. Saatavissa: <https://searchsecurity.techtarget.com/definition/honeynet> [viitattu 27.3.2020].

Simonovich, V. 2019. Imperva Blocks Our Largest DDoS L7/Brute Force Attack Ever (Peaking at 292,000 RPS). Blog. Saatavissa: <https://www.imperva.com/blog/imperva-blocks-our-largest-ddos-l7-brute-force-attack-ever-peaking-at-292000-rps/> [viitattu 21.11.2019].

Tan, E. 2014. Dionaea – A Malware Capturing Honeypot. Blogi. Saatavissa: <https://www.div0.sg/single-post/dionaea-malware-honeypot> [viitattu 29.3.2020].

The Tcpdump Group. 2020. Manpage of TCPDUMP. WWW-dokumentti. Saatavissa: <https://www.tcpdump.org/manpages/tcpdump.1.html> [viitattu 7.3.2020]

VirusTotal. s.a. How it works. WWW-dokumentti. Saatavissa: <https://support.virustotal.com/hc/en-us/articles/115002126889-How-it-works> [viitattu 29.3.2020].

VMware. s.a. VMware Workstation 4. WWW-dokumentti. Saatavissa: https://www.vmware.com/support/ws4/doc/preserve_snapshot_ws.html [viitattu 26.3.2020].

.

Cowrien asennus ja käyttöönotto

SSH portin vaihto sekä uudelleenohjaus

```
/etc/ssh/sshd_config
```

```
Port 64295
```

```
ssh service restart
```

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-port  
2222
```

Portin vaihtumisen voi vielä varmistaa komennolla:

```
Netstat -tan
```

Cowrien riippuvaisuuksien sekä MySQL:n lataaminen

```
apt-get install git python-virtualenv libssl-dev libffi-dev build-essential libpy-  
thon3-dev python3-minimal authbind virtualenv
```

```
apt-get install mysql-server libmysqlclient-dev python-mysqldb
```

Käyttäjän luonti

```
adduser --disabled-password cowrie
```

Cowrien lataus

```
git clone http://github.com/cowrie/cowrie
```

Virtuaaliympäristön luonti ja aktivointi sekä päivitysten lataus

```
virtualenv --python=python3 cowrie-env
```

```
source cowrie-env/bin/activate
```

```
pip install --upgrade pip
```

```
pip install --upgrade -r requirements.txt
```

MySQL asennus

```
source cowrie/cowrie-env/bin/activate
pip install mysqlclient
mysql -u root -p
CREATE DATABASE cowrie;
GRANT ALL ON cowrie.* TO 'cowrie'@'localhost' IDENTIFIED BY 'PASS-
WORD HERE';
FLUSH PRIVILEGES;
exit
```

Navigoi kansioon /cowrie/docs/sql/

```
mysql -u cowrie -p
USE cowrie;
source /cowrie/docs/sql/mysql.sql;
exit
```

Cowrie.cfg muokkaus

Cowrien asetustiedosto löytyy polusta /cowrie/etc/, on suositeltavaa joko kopioida koko cowrie.cfg.dist ja nimetä se cowrie.cfg tiedostoksi, tai luoda kokonaan uusi tiedosto nimellä cowrie.cfg. Cowrie hakee asetukset aina ensin cowrie.cfg tiedostosta, joten ei ole tarvetta muokata cowrie.cfg.dist tiedostoa.

```
[output_mysql]
host = localhost
database = cowrie
username = cowrie
password = PASSWORD HERE
port = 3306
debug = false
enabled = true
```

Käynnistetään Cowrie

Bin/cowrie start

Varmistetaan toiminta

Ota hunajapurkkiin SSH-yhteys testataksesi toiminta. Kirjautumistiedot pitäisivät nyt näkyä MySQL tietokannassa.

```
mysql -u cowrie -p  
use cowrie;  
select * from auth;
```

Ongelmanratkonta

Suurimmat ongelmat näkyvät Cowrien käynnistämisen jälkeen lokitiedostossa

```
cat /var/log/cowrie/cowrie.log
```