jamk.fi

Organisaation sisäverkon tilannekuvan parantaminen hunajapurkkituotteita hyödyntäen

Markus Hänninen

Opinnäytetyö
Marraskuu 2020
Tekniikan ja liikenteen ala
Insinööri (AMK), Tieto- ja viestintätekniikan tutkinto-ohjelma
Kyberturvallisuus

Jyväskylän ammattikorkeakoulu
JAMK University of Applied Sciences

jamk.fi

Kuvailulehti

Tekijä(t)	Julkaisun laji	Päivämäärä
Hänninen, Markus	Opinnäytetyö, AMK	Marraskuu 2020
	Sivumäärä 58	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty: kyllä

Työn nimi
Organisaation sisäverkon tilannekuvan parantaminen hunajapurkkituotteita hyödyntäen

Tutkinto-ohjelma
Tieto- ja viestintätekniikka (kyberturvallisuus)

Työn ohjaaja(t)
Vajaranta, Markku & Immonen, Jani

Toimeksiantaja(t)
Suomen Erillisverkot Oy

Tiivistelmä

Yhteiskunnan siirtyessä toimimaan yhä kasvavissa määrin erilaisten verkkopalveluiden varassa, on myös rikollisuus alkanut siirtyä enenevissä määrin verkkoon. Tämän vuoksi useissa organisaatioissa on herätty kasvavaan tarpeeseen tietoturvan parantamiseksi. Usein organisaation tietoturva on pyrkinyt keskittymään pitämään mahdolliset hyökkääjät omien ympäristöjen ulkopuolella kaikin mahdollisin keinoin, mutta samalla omien sisäverkkojen tilanne on jäänyt heikommalle. Pahimmillaan organisaatioilla ei ole lainkaan näkyvyyttä omien verkkojensa tilannekuvasta. Opinnäytetyössä etsittiin ratkaisuja organisaation sisäverkon tilannekuvan parantamiseksi avoimen lähdekoodin hunajapurkki tuotteita hyödyntäen.

Opinnäytetyön tavoitteena oli tutustua hunajapurkkien tietoperusteisiin ja erilaisiin tarjolla oleviin avoimen lähdekoodin hunajapurkkiratkaisuihin. Vertailuun valittiin muutamia sopivia ehdokkaita, joista valittiin yksi tarkempaa läpikäyntiä varten. Valitun tuotteen ominaisuuksia, resurssien käyttöä ja skaalautuvuutta arvioitiin ja sille suoritettiin testiasennus virtuaaliympäristöön.

Työn aikana kerättiin tietoa myös erilaisista teknologioista, joita voidaan hyödyntää hunajapurkkien käyttöönotossa ja skaalaamisessa laajempaan käyttöön. Tutkimuksessa selvitettiin myös, kuinka hunajapurkkien kautta saatavaa tietoa olisi mahdollista hyödyntää mahdollisimman tehokkaasti integraatioilla taustajärjestelmiin, kuten esimerkiksi tiketöinti- ja valvontajärjestelmiin.

Avainsanat (asiasanat)
Hunajapurkki, Honeypot, T-Pot, Cowrie, kyberturvallisuus, tilannetietoisuus, tilannekuvan parantaminen, Docker, konttiteknologia, kontitus, kontti, killchain

Muut tiedot (salassa pidettävät liitteet)

jamk.fi

Description

Author(s)	Type of publication	Date
Hänninen, Markus	Bachelor's thesis	November 2020
		Language of publication: Finnish
	Number of pages: 58	Permission for web publication: yes

Title of publication

Increasing the organizations internal network situational awareness with honeypots

Degree programme

Supervisor(s)

Vajaranta, Markku & Immonen, Jani

Assigned by

Suomen Erillisverkot Oy

Abstract

As digitalization increases, society is moving its crucial services more and more to networks. Meanwhile the criminal activity has been increasing in networks on yearly basis. Criminals are moving to internet as well. This has caused organizations to notice an increased need for information and cyber security. Typically, organizations have been trying to keep the attackers out from their environments by securing their defenses on the outer edges. At the same time their internal networks are often lacking visibility and suitable defense mechanisms. On worst case scenario, if the attacker has breached the perimeter, the organization might not have any visibility on attackers' doings on their own network. This thesis focused on topic how to increase the situational awareness on organizations internal networks with open source honeypot products.

The goal in this thesis was to study theory basics behind honeypot products and find out what kind of open source honeypot products currently exists. Couple of honeypots were selected to be studied more. After going through the basics of these products, one was selected to be researched more. The selected honeypot product was researched about its technical properties, resource usage, and its scalability were compared. Selected honeypot was installed to a virtual environment for further tests.

During the thesis, other technologies which could benefit the installation and implementation of the chosen honeypot, were also researched to increase the efficiency and scalability of the chosen product. There was also some research about enabling the communication from honeypot to existing ticketing and monitoring systems.

Keywords/tags (subjects)

honeypot, T-Pot, Cowrie, situational awareness, Docker, Container, Containerization

Miscellaneous (Confidential information)

Sisältö

1 Johdanto 6

1.1 Työn tausta 6

1.2 Tutkimusongelma 10

1.3 Tutkimusmenetelmät 10

2 Tilannetietoisuus ja hyökkäysmenetelmien kuvaaminen 12

3 Hunajapurkit ja niiden jaottelut 19

3.1 Käyttötarkoitus 21

3.2 Käytettävät OSI-mallin tasot 24

3.3 Vuorovaikutteisuus 25

3.4 Sijoittaminen verkkoon 27

3.5 Hunajapurkki vai hunajaverkko 29

4 Skaalautumista helpottavat työkalut 30

4.1 Ansible 30

4.2 Konttiteknologiat 31

5 Vertailtavat tuotteet 36

5.1 Cowrie 37

5.2 OpenCanary 38

5.3 T-Pot 39

6 Tuotteen valitseminen 46

7 Testiasennus 47

8 Yhteenveto 57

9 Pohdinta 59

Lähteet 62

Liitteet 67

Liite 1. T-Potin arkkitehtuurikuvaus 67

Kuviot

Kuvio 1. Kyberrikosten kehitys 2010 -2018 aikavälillä (Lähde: Tilastokeskus, rikos- ja pakkokeinotilasto (11cg -- Tietoon tulleet rikokset ja niiden selvittäminen rikosnimikkeittäin, 2010-2018) 7
Kuvio 2. Tilannetietoisuuden vaikutus päätöksen teossa ja toiminnassa 13
Kuvio 3. Kybertilannekuvan muodostaminen 14
Kuvio 4. Kolmen sovelluksen ajaminen konteissa yhdellä isäntäkoneella 33
Kuvio 5. Web-käyttöliittymän kirjautumisikkuna 48
Kuvio 6. Kibanan avulla helposti havainnoitavaan muotoon visualisoidut listat eniten käytetyistä käyttäjätunnuksista ja salasanoista. 50
Kuvio 7. Hyökkääjän Cowrien emuilomalla palvelimella ajamat komennot listattuna 50
Kuvio 8. Cockpitin tarjoama näkymä palvelimen resurssien käyttötilastoihin. 51
Kuvio 9. Näkymä konttien hallinnasta Cockpitin web-käyttöliittymässä. 53
Kuvio 10. Kontin luonnin parametrien asettaminen levykuvaa käytettäessä. 54
Kuvio 11. CyberChef työkalun käyttöliittymä. 55
Kuvio 12. Tietojen hakuehtoja spiderfootin käyttöliittymästä. 56
Kuvio 13. Sicherheitstacho eli reaaliaikainen kartta T-Pot ympäristöihin kohdistuvista hyökkäyksistä. 56

Sanasto:

API	Application Programming Interface	Ohjelmointirajapinta
APT	Advanced Persistent Threat	Kehittyneet uhkatoimijat
CPU	Central Processing Unit	Laskentayksikkö
DMZ	Demilitarized zone	Demilitarisoitu alue on fyysinen tai looginen alue, joka yhdistää organisaation oman verkon turvattomampaan alueeseen
DoS	Denial of Service	Palvelunestohyökkäys
DTK	Deception Toolkit	Yksi varhaisimpia hunajapurkkeja
IDS	Intrusion Detection System	Tunkeutujan havaitsemisjärjestelmä
IP	Internet Protocol	Verkkokerroksen protokolla, joka huolehtii IP-tietoliikennepakettien toimittamisesta perille pakettikytkentäisessä Internet-verkossa
IPS	intrusion Prevention System	Tunkeilijanestojärjestelmä
MITM	Man in the Middle	Välimieshyökkäys on hyökkäys, jossa hyökkääjä pääsee kiinni kahden viestijän väliseen liikenteeseen
OSI	Open Systems Interconnection model	OSI-malli kuvaa tiedonsiirtoprotokollien keskinäistä toimintaa seitsemälle kerrokselle jaettuna.
PID	Process Identifier	Prosessin tunniste
RAM	Random Access Memory	Keskusmuisti eli käyttömuisti
RDP	Remote Display Procol	Työpöydän etäkäyttö protokolla
SMS GW	Short Message Service GateWay	Yhdyskäytävä tekstiviestien lähettämiseen
SQL	Structured Query Language	Kyselykieli relaatiotietokantojen käyttämiseksi

SSH	Secure Shell	Salattuun tietoliikenteeseen tarkoitettu protokolla
TCP	Transmission Control Protocol	Tietoliikenneprotokolla tietokoneiden väliseen luotettavaan tiedonsiirtoon
UID	User Identifier	Käyttäjätunniste
YAML	Yaml Ain’t Markup Language	Merkintäkieli, jota käytetään esimerkiksi Ansiblen pelikirjoissa

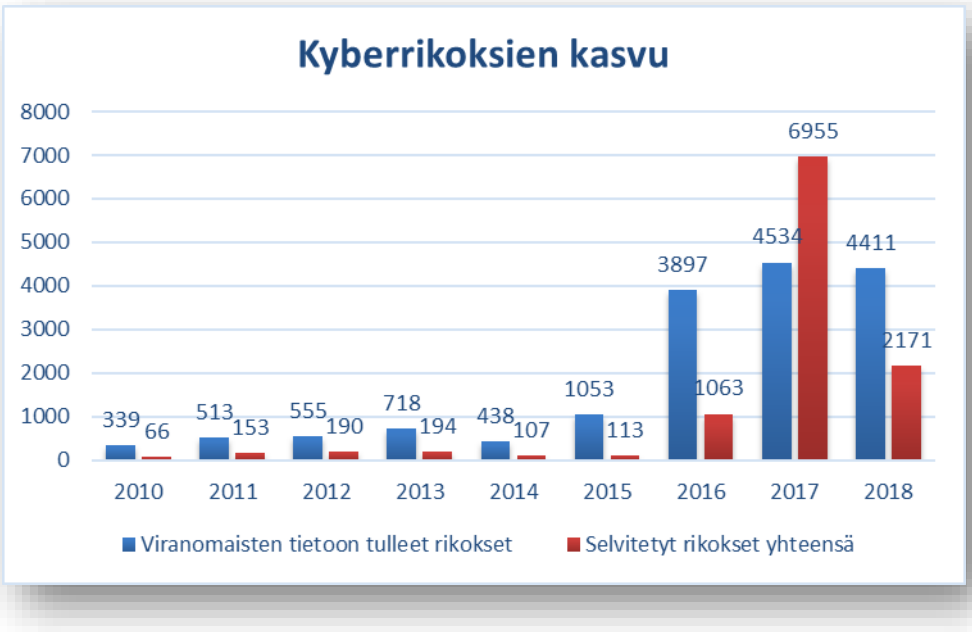
1 Johdanto

1.1 Työn tausta

Yhteiskunnan jatkaessa digitalisoitumistaan on informaatioteknologia sulautunut yhä kiinteämmin osaksi työtä ja arkea. Digitalisaation edetessä ihmisten normaalissa arjessa, myös rikollisuus on siirtymässä verkkoon. Rikostentorjuntaneuvosto on oikeusministeriön yhteydessä toimiva asiantuntija- ja yhteistyöelin, jonka mukaan kyberrikokset voidaan jakaa karkeasti sen mukaan, ovatko tietoverkot ja -tekniikka rikoksen kohteena vai käytetäänkö niitä apuna perinteisempien rikosten teossa. Kyberrikosten määrät ovat viime vuosina kasvaneet huomattavasti, mutta suhdeluvusta kokonaisrikollisuuteen ei ole vielä olemassa kattavia koottuja tilastotietoja. (Kyberrikokset n.d.)

Kyberrikollisuudesta puhuttaessa tarkka rajanveto perinteisen ja kyberrikollisuuden välillä on melko haastavaa, koska kyseessä on todella moniulotteinen ilmiö. Tämän vuoksi rajanveto joudutaan usein harkitsemaan jopa tapauskohtaisesti. Samaan aikaan hyökkäystekniikat ovat kehittyneet, niiden takana olevien rikollistahojen alkaessa erikoistua ja jakaa tietoa keskenään. Tämän lisäksi ihmiset saattavat mieltää verkossa tehtävät rikokset eri tavalla kuin fyysisessä maailmassa tapahtuvat. Verkko-identiteetin suoma, toisinaan tosin näennäinen, anonymiteetti ja kiinni jäämisen riskin pieni todennäköisyys vaikuttavat usein yksilöiden harkintakykyyn. (Limnéll 2013.)

Jonkinlaista kuvaa kyberrikosten määrän kasvusta on kuitenkin saatavilla, sillä tilastokeskus julkaisee Rikos- ja pakkokeinotilastoa, josta voidaan tarkastella myös poliisin tietoon tulleita tietoliikenteeseen ja tietojärjestelmiin kohdistuneita rikoksia ja vastaavien rikoksien ratkaistuja tapauksia. Voimakasta kasvua vuoden 2015 jälkeen selittää omalta osaltaan vuonna 2015 voimaan tulleet uudet kyberrikosmuodot, kuten identiteettivarkaus ja datavahingonteko.



Kuvio 1. Kyberrikosten kehitys 2010 -2018 aikavälillä (Lähde: Tilastokeskus, rikos- ja pakkokeinotilasto (11cg -- Tietoon tulleet rikokset ja niiden selvittäminen rikosnimikkeittäin, 2010-2018)

Kuviossa 1 on nähtävissä kasvu kyberrikosten määrässä vuosien 2010 ja 2018 välisenä aikana. Samassa taulukossa on myös nähtävillä viranomaisen tietoon tulleiden rikosten määrä suhteessa ratkaistuihin tapauksiin. Kyseiseen kuvioon on kirjattu muutamia yleisimpiä kyberrikoksia eri vakavuusasteina ja yrityksinä, jotka myös ovat nykyään rangaistavia tekoja. Tällaisia ovat esimerkiksi datavahingoteko, identiteettivarkaus, tietoliikenteen häirintä, tietomurto ja tietojärjestelmän häirintä. Kuviosta käy myös ilmi, että ratkaistujen rikosten osuus poliisin tietoon tulleista rikoksista on melko pieni, pois lukien vuonna 2017, jolloin poliisi oli Tilastokeskuksen mukaan selvittänyt tasan 5000 törkeää tietomurtoa. Useimpina vuosina kuitenkin selvitettyjen rikosten osuus kaikista kyberrikoksista on huomattavan pieni. Kyberrikollisuuden yksi erityispiirre onkin nimenomaan, että rikollisen on huomattavasti fyysistä toimintaympäristöä helpompaa piilottaa oma identiteettinsä verkossa ja pyrkiä tietoisesti vaikeuttamaan kiinnijäämistään reaalimaailmassa tapahtuvaan rikollisuuteen verrattuna. (Limnéll 2013.)

Yrityksiin kohdistuneiden kyberrikosten osalta on tarjolla niukasti tilastoja tai tietolähteitä. Tämä mahdollisesti selittyy sillä, että tietoon tulleita tapauksia ei haluta ilmoittaa

viranomaisille tai yrityksen ulkopuolelle ylipäänsä, koska sellaiset usein nähdään huonona julkisuutena, joka pahimmillaan saattaisi vaikuttaa yrityksen osake- tai markkina-arvoon. (Näsi 2019.)

Yksi ajankohtainen esimerkki kyberrikosten aiheuttamista ongelmista on lokakuussa 2020 julkisuuteen noussut Psykoterapiakeskus Vastaamon tietomurto vuodelta 2018. Tämän hetkisen tiedon mukaan hyökkääjät onnistuivat pääsemään käsiksi Vastaamon arkaluontoisia potilastietoja sisältäviin tiedostoihin ja kopioida ne itselleen. Tämän jälkeen hyökkääjä esitti kiristysviestin yritykselle vaatien suuria rahallisia lunnaita, jotta varastettuja tietoja ei julkaistaisi. Vastaamo kieltäytyi maksamasta, jonka jälkeen hyökkääjät julkaisivat osan varastamistaan tiedoista. Tapaus koski kymmeniätuhansia Vastaamon asiakkaita. Yrityksen ja koko psykoterapia-alan kärsimä mainehaitta tapauksen johdosta oli valtava, puhumattakaan yksityishenkilöiden henkilökohtaisista kärsimyksistä. (Uusimmat tiedot Vastaamon tietomurrosta 2020.)

Koska mahdollisilla tietomurroilla on voimakkaita vaikutuksia yrityksien imagoon, ovat organisaatiot heränneet tarpeeseen parantaa ja kehittää omia ympäristöjään siten, että tämän kaltaisilta ongelmilta voitaisiin välttyä. Yhtenä tällaisena keinona toimii näkyvyyden parantaminen omissa verkoissa tapahtuviin mahdollisiin tietoturvapoikkeamiin jo mahdollisimman varhaisessa vaiheessa. Usein ongelmana kuitenkin on, että hyökkäykset havaitaan vasta siinä vaiheessa, kun tietomurto tai vastaava hyökkäys on jo ohi. Yleisesti ottaen tietoturvan parantaminen tulisi aloittaa kriittisten ja suojattavien tietojen tunnistamisella, jotta tiedetään mitä tarkkaan ottaen lähdetään suojaamaan. Hyvänä pohjana tietoturvan kehittämiselle toimii ikään kuin sipulimainen suojautumisen ajatusmalli. Tällöin yhden kontrollin pettäessä, sen alla odottaa toinen, eikä suojattava kohde suoraan vaarannu. Tällaisen lähtötilanteen päälle on hyvä alkaa kehittämään yrityksen kyberkyvykkyyttä syvemmällä tasolla. Taloudellisen hyödyn perässä olevat kyberrikolliset menevät usein sinne, minne helpoimmalla pääsevät, jolloin jo perusasioiden kuntoon laittamisella on mahdollista välttyä kohteeksi joutumiselta. (Limnéll 2013.)

Tämä opinnäytetyö pyrkii omalta osaltaan kehittämään organisaatioiden sisäistä kyvykkyyttä kyberhyökkäyksien havaitsemiseen mahdollisimman varhaisessa vaiheessa parantamalla yrityksen sisäistä tilannekuvaa. Sisäverkossa oleva hunajapurkki toimii

havainnoivana kontrollina, jonka tarkoituksena on saada indikaatio hyökkääjän mahdollisesta pääsystä sisäverkkoon. Ulkoreunalle asennettu hunajapurkki toimii osaltaan myös havainnoivana, mutta niissä voi olla myös ennaltaehkäisevää toiminnallisuutta. Tässä mallina voisi toimia: ennaltaehkäisy – havainnointi – reagointi, jossa pääpainopiste tulisi olla hyökkääjän mahdollisimman aikaisessa estämisessä. Työn tavoitteena on tutustua erilaisiin avoimen lähdekoodin hunajapurkkituotteisiin ja tutkia erilaisia menetelmiä, joilla edellä mainittujen tuotteiden käyttöönotto, ylläpito ja skaalautuminen laajempaan käyttöön sujuisi mahdollisimman tehokkaasti. Rajaus avoimen lähdekoodin hunajapurkkiratkaisuihin tuli toimeksiantajalta, mutta se on myös tietoturva näkökulmaa ja kustannustehokkuutta tukeva rajaus. Avoimen lähdekoodin tuotteita käytettäessä käyttäjä voi halutessaan perehtyä tarkemmin tuotteeseen ja varmistua, ettei tuotteessa ole tarkoituksellisesti jätetty mitään haitallista koodia. Mikäli käyttäjällä on tarpeeksi osaamista, voi hän arvioida myös erilaisten teknisten toteutuksien turvallisuutta. Mikäli asia koetaan tarpeeksi kriittiseksi ja organisaation sisältä ei löydy tarvittavaa osaamista, voidaan koodikatselmointi myös tarvittaessa ostaa ulkopuolisilta tahoilta. Avoimen lähdekoodin tuotteet mahdollistavat läpinäkyvän ja yleensä kustannustehokkaan käyttöönoton. Tästä syystä tutkimuksen keskittämisestä avoimen lähdekoodin ratkaisuihin voi olla hyötyä muillekin kuin työn alkuperäiselle toimeksiantajalle.

Tilannetietoisuuden parantamisen työkaluksi valikoitui nimenomaan hunajapurkit, koska ennemmin tai myöhemmin kaikki perinteiset tietoturvakeinot, jotka pyrkivät estämään hyökkääjien pääsyn sisään järjestelmään, voivat murtua. Oli kyseessä sitten ongelmat palomuurissa, IDS:ssä tai henkilöstön puutteellisessa koulutuksessa, lähtökohtaisesti ylläpitäjien tulisi aina lähteä oletuksesta, että tarpeeksi päättäväisen ja kattavilla resursseilla varustetun hyökkääjän onnistuu päästä läpi ulkoisista puolustuksista niin halutessaan ennemmin tai myöhemmin. Hyökkääjä saattaa myös tulla sisään ympäristöön jotain muuta, kuin verkon kautta. Nykyään yleisimpinä tapoina alkaa olla sosiaalinen manipulointi ja hyökkäykset tapahtuvat esimerkiksi sähköpostin, selaimen tai siirrettävien medioiden kautta. Tässä tilanteessa hunajapurkki saattaa toimia ylläpitäjille viimeisenä hälyttimenä ympäristön murtamisesta.

1.2 Tutkimusongelma

Tutkimus sijoittuu kyberturvallisuuden kontekstiin ja sen toimeksiantajana toimii Suomen Erillisverkot Oy. Tutkimuksen tavoitteena on kerätä tietoa hunajapurkkien jaottelusta, käyttötapauksista, sekä hunajapurkeista saatavan tiedon hyödyntämisestä organisaatioiden sisäisen tilannekuvan parantamiseksi. Lisäksi tavoitteena on etsiä ja vertailla yrityskäyttöön sopivia avoimen lähdekoodin hunajapurkkituotteita, joiden avulla voitaisiin parantaa näkyvyyttä yrityksen sisäverkoissa mahdollisesti tapahtuviin haitallisiin tapahtumiin. Toimeksiantajan puolelta esitettiin tiettyjä toiveita tutkittavien tuotteiden osalta, mutta pääpiirteissään tutkittavien tuotteiden valinta kohdistuu avoimen lähdekoodin hunajapurkkituotteisiin, joiden avulla on mahdollista kuvata mahdollisimman laajasti erityyppisiä yrityksistä yleisesti löytyviä, palveluita ja prosesseja. Yllä mainittujen toiveiden lisäksi tuotteen käyttöönotto, asennukset ja ylläpito tulisi olla suoraviivaista ja tuotteiden tulisi olla tarvittaessa helposti skaalattavissa myös laajempaan käyttöön.

Tuotteiden valinnassa tulisi kiinnittää huomiota tuotteiden mahdollisiin tietoturvapuutteisiin, jotta tutkittavien tuotteiden mahdollisilta väärinkäytöksiltä hyökkääjien toimesta vältyttäisiin. Työssä pyritään myös sivuamaan mahdollisuuksia yhdistää tutkittavien tuotteiden raportointitoiminnallisuuksia, eri organisaatioiden käytössä oleviin taustajärjestelmiin, kuten esimerkiksi tiketöinti- ja valvontajärjestelmiin, niiltä osin kuin se on mahdollista.

Tutkimuskysymyksiä määrittyi lopulta kolme:

1. **Millaisia avoimen lähdekoodin hunajapurkki ratkaisuja on tarjolla?**
2. **Kuinka yrityksen sisäistä tilannekuvaa olisi mahdollista parantaa hunajapurkki tuotteita hyödyntämällä?**
3. **Kuinka mahdollinen käyttöönottoprosessi voitaisiin toteuttaa skaalautuvasti ja tehokkaasti?**

1.3 Tutkimusmenetelmät

Opinnäytetyö tulee olemaan pääasiallisesti kvalitatiivista, eli laadullista tutkimusta. Laadullisen tutkimuksen tavoitteena on lisätä ymmärrystä tutkittavasta kohteesta,

mahdollistaen myös vaihtoehtoiset tulkinnat. Lisäksi laadullisella tutkimuksella voidaan antaa asioille merkityksiä ja tuottaa mallinnuksia asioista. Koska kyseessä on toimeksiantajalta saatu selvitystyö tämän hetkisistä tarjolla olevista hunajapurkkituotteista ja niiden käyttöönotosta, osuu tutkimus soveltavan tutkimuksen kategoriaan.

Työ koostuu kahdesta erillisistä osioista. Käsitteellinen osuus käy läpi aiheesta jo olemassa olevia tutkimuksia avaten eri käsitteitä ja erityyppisten hunajapurkkituotteiden käyttötarkoituksia ja sijoittelua. Kokeellisessa osuudessa puolestaan arvioidaan eri kehittäjien hunajapurkkituotteiden hyviä ja huonoja puolia suhteessa toimeksiantajaorganisaation tuotteille asettamiin toivomuksiin. Eri tuotteiden vertailussa hyödynnetään tutkijan empiirisiä havaintoja, jotta tarjolla olevista tuotteista voidaan rajata pienempi otos. Tätä otosta lähdetään vertailemaan ja tutkimaan lisää. Lopulta valitaan yksi tai useampi tuote, joiden asennusprosessia ja testaamista käydään tarkemmin läpi.

Aineistonkeruu suoritetaan tutustumalla aikaisempiin tutkimuksiin, joita tutkimusalueelta on jo julkaistu, mutta koska kyseinen aihealue päivittyy usein yhteisön havaintojen ja tuotteiden kehittäjien toimesta, tullaan lähdemateriaalina käyttämään myös ei akateemisia lähteitä. Koska kyseessä on soveltava tutkimus, jossa tullaan tutkimaan ja vertailemaan tarjolla olevia tuotteita, ei aiheesta välttämättä löydy sopivaa vertaisarvioitua lähdemateriaalia tai se saattaa olla usein olla jo vanhentunutta tietoa alalla nopeasti tapahtuvien muutoksien vuoksi. Aineistonkeruussa pyritään pitämään kiinni lähdemateriaalien luotettavuudesta, mutta tutkimuksen lukijan on hyvä tiedostaa, että osa tuotteita koskevista lähdemateriaaleista eivät ole akateemisia tai ainakaan vertaisarvioituja, vaan ne saattavat pahimmillaan pitää sisällään tarkoituksellisesti harhaanjohtavaa tietoa tai julkaisijan oman edun tavoitteluun liittyviä tausta-agendoja.

Tutkimusprosessissa tullaan noudattamaan Baimyrzaevan (2018, 18) julkaisemaa viisi askelmista prosessia, jotka ovat:

1. Selvitä tutkimuksen rajaus.
2. Tutustu aikaisemmin julkaistuihin tutkimuksiin aiheesta.
3. Pohdi tutkimussuunnitelmasi eri tehtävävaiheet ja käytettävät menetelmät.
4. Kerää, analysoi ja tee johtopäätökset tutkimuksestasi.
5. Julkaise tutkimus.

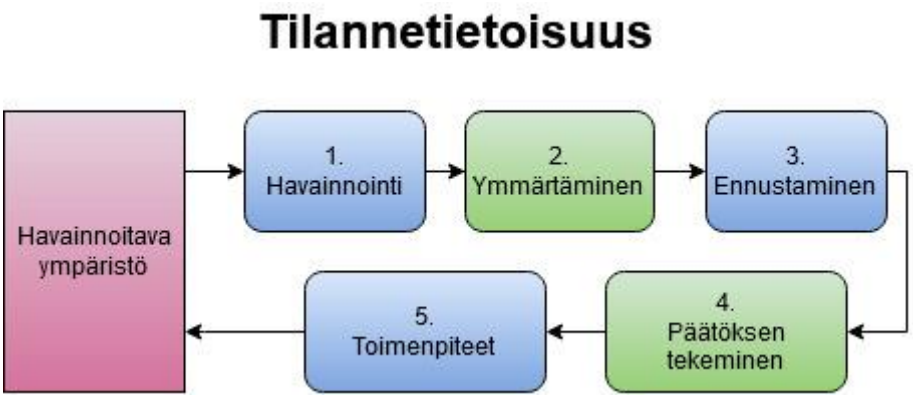
Aihepiiri on rajattu käsittelemään hunajapurkkituotteiden käyttöä organisaation sisäverkon tietoturvan parantamiseksi, eikä siinä käsitellä juurikaan esimerkiksi tutkimuksellisia hunajapurkkituotteita, joita käytetään useammin esimerkiksi tiedonkeruussa hyökkääjien toimintamalleista ja uusista mahdollisesti ennen tuntemattomista hyökkäysmenetelmistä. Tietyiltä osin kyseiset toiminnot saattavat kuitenkin limittyä keskenään.

Tutkimuksen tuloksena käydään läpi kyberhyökkäyksen etenemistä ympäristössä, hunajapurkkituotteiden historiaa ja teoriaa, erityyppisiä toteutuksia tuotteista, sekä erilaisia käyttötapauksia tarpeiden ja tuotteiden sijoituksien mukaan. Samalla perehdytään erityyppisten hunajapurkkituotteiden ylläpidon tuomiin haasteisiin ja mahdollisiin saavutettaviin lisäetuihin toisiin tuotteisiin verrattuna.

2 Tilannetietoisuus ja hyökkäysmenetelmien kuvaaminen

Tilannekuva on reaaliaikainen kuvaus jostain seurattavasta tilanteesta. Se koostuu sillä hetkellä kaikesta saatavilla olevasta aiheeseen liittyvistä tiedoista eli tilannetiedoista. Tilannetietoa voi olla esimerkiksi hunajapurkilta saatava havainto, jonka perusteella saadaan tieto haitallisesta toiminnasta organisaation ympäristöissä tai jonkin muun järjestelmän tai asiakirjan pohjalta saatava aiheeseen liittyvä tieto. Tilannekuva on yksilön omaan tilannetietoisuuteen verrattuna objektiivisempi, minkä vuoksi sitä voidaan paremmin jakaa yksilöiden kesken.

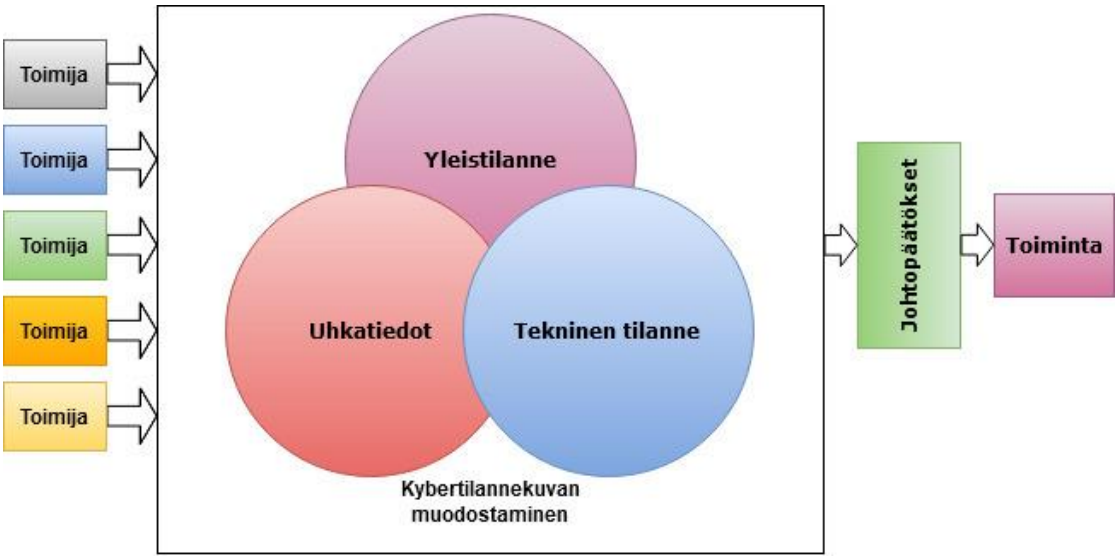
Tilannetietoisuus taas on yksilön omaa tulkintaa omien kokemuksiensa kautta ja siihen vaikuttaa muun muassa yksilön taustat, kokemukset ja esimerkiksi väsymys tai stressi. Tilannetietoinen henkilö kykenee havainnoimaan mitä ympäristössä tapahtuu tällä hetkellä, miten tilanne mahdollisesti etenee tulevaisuudessa ja mitä toimenpiteitä hänen on mahdollista tehdä. Tilannetietoisuutta on mahdollista kuvata kehänä, jossa henkilö havainnoi ympäristöä ja pystyy tilannetietoisuutensa avulla ennakoimaan mahdollisia tulevia tapahtumia ja suunnittelemaan ja toteuttamaan toimenpiteitä havainnoitujen tietojen pohjalta. Tilannetietoisuus on siis vahvasti tulevaisuusorientoitunutta. Tilannetietoisuuden pohjalta tehtävien päätöksien syklinen kuvaus Endsleyn mukaan on nähtävissä kuviossa 2. (Koistinen 2011.)



Kuvio 2. Tilannetietoisuuden vaikutus päätöksen teossa ja toiminnassa

Tilannekuvan ja tilannetietoisuuden tarkat määritelmät vaihtelevat jonkin verran havainnoitavista asioista riippuen, mutta yhteisenä tekijänä kaikille toimii tilannekuvasta saatujen tietojen toimiminen tukena päätöksien tekemiselle. Tämän ansiosta päätökset, resursointi ja toiminta saadaan tehtyä parhaimmalla mahdollisella tavalla ja oikeaaikaisesti kuhunkin tilanteeseen sopien. (Mt.)

Kyberturvallisuuden näkökulmasta tilannekuvalla pyritään havainnoimaan ja ymmärtämään organisaation järjestelmiin ja tietoihin kohdistuvia ajankohtaisia uhkia, sekä ennakoimaan mahdollisia tulevaisuudessa esiin nousevia uhkia. Kybertilannekuvaa ei välttämättä muodosteta ainoastaan saatavilla olevista teknisistä tilannetiedoista, vaan kybertilannekuvassa tulisi olla myös seurantaa voimassa olevista uhkatiedoista ja ympäristössä yleisesti vallitsevasta tilanteesta. Kyberturvallisuuden tilannekuvasta puhuttaessa on tärkeää huomata, että sitä seuraavilla ja tuottavilla henkilöillä on käytössään riittävä tietämys olemassa olevista erilaisista potentiaalisista uhkista ja mahdollisten haitallisten toimijoiden yleisimmistä käyttäytymismalleista. Tällä hetkellä tärkein toimija kybertilannekuvan muodostamisessa on edelleen ihmiset, jotka pystyvät koostamaan kerätyistä tiedoista selkeän kokonaisuuden. Joissain tapauksissa toimijana voi toimia myös automatiikka tai kokonaan erillinen tarkoitukseen tehty järjestelmä. Kaikki toimijat pitävät huolta ennakkoon sovituista omista osa-alueistaan, jotta tilannekuvasta saadaan riittävän kattava, mutta turhalta päällekkäiseltä työltä vältytään. Kybertilannekuvan muodostaminen on nähtävissä kuviossa 3. (Laari, Flyktman, Härmä, Timonen & Tuovinen 2019 52-53.)



Kuvio 3. Kybertilannekuvan muodostaminen

Koska digitaalisesta toimintakentästä muodostettava kuva on usein todella laaja ja monimutkainen, saattaa yksittäinen ihminen sokeutua jatkuvaan herätepaineeseen. Yksi suurimmista ongelmista on erilaisista tietojärjestelmistä kumpuavien herätteiden määrä. Suurten herätemäärien käsittely on usein ihmiselle jopa mahdotonta (Ihanus 2019.). Usein organisaatioilla on käytössään jonkinlainen tilannekuvajärjestelmä, joka kerää ja analysoi tilannetietoja. Se saattaa esimerkiksi yhdistää useammasta lähteestä tulevia herätteitä ja tehdä johtopäätöksen, että tapahtuman tarkistukseen ja käsittelyyn tarvitaan ihmisen huomiota. Tilannekuvajärjestelmä tarjoaa myös yleensä keinon esittää kerättyjä tilannetietoja mahdollisimman ymmärrettävällä ja havainnollistavalla tavalla, jotta henkilöt voivat hyödyntää kerättyjä tietoja mahdollisimman tehokkaasti päätöksien tekemisessä jatkotoimenpiteiden suhteen (Laari, Flyktman, Härmä, Timonen & Tuovinen 2019, 52−53.). Toimivan tilannekuvan avulla ylläpitäjät pystyvät arvioimaan, sillä hetkellä vallitsevaa tilannetta omissa ympäristöissään ja kohdistamaan resurssejaan erityisesti sellaisiin osa-alueisiin, joissa havaitaan esimerkiksi tuoreita haavoittuvuuksia tai joihin hyökkääjät muuten kohdistavat paljon hyökkäyksiään.

Jotta tilannekuvaa pystytään tuottamaan ja hyödyntämään oikein, tulee käyttäjien ymmärtää myös heihin kohdistuvia uhkatekijöitä. Hyökkääjien toimintaa kohteissaan kuvataan usein erilaisina hyökkäysketjuina. Hyökkäysketju terminä juontaa juurensa alun perin armeijan käyttöön. Alkuperäisessä käyttötarkoituksessaan termillä kuvattiin

kohteen tunnistamista, hyökkäykseen valmistautumista, hyökkäykseen aseistautumista ja kohteen tuhoamista (Hospelhorn 2020). Kyberturvallisuuden näkökulmasta termi nousi käyttöön Lockheed Martinin tietojärjestelmätieteen tutkijoiden toimesta vuonna 2010. Kyberhyökkäyksen hyökkäysketjulla pyritään kuvaamaan tyypillistä systemaattista prosessia, jolla hyökkäys etenee APT:n (Advanced (Kehittynyt, tässä tapauksessa esimerkiksi kohdistettu, koordinoitu ja tarkoituksenmukainen) Persistent (Pysyvä eli kestoltaan pitkäaikainen) Threat (Uhka eli henkilö), eli ammattimaisen osaavan hyökkääjän toteuttaessa organisoitua hyökkäystä valitsemaansa organisaatiota vastaan. (Hutchins, Cloppert & Amin 2010, 4−5.)

Lockheed Martinin malli koostuu seitsemän vaiheisesta ketjusta:

1. Tiedustelu: Hyökkääjä valitsee kohteen ja tutkii sitä yrittäen löytää mahdollisimman paljon tietoa hyökkäyksen kohteena olevasta organisaatiosta. Tiedustelu alkaa yleensä passiivisella tiedustelulla julkisia lähteitä käyttäen, mutta saattaa sisältää myös aktiivista tiedustelua esimerkiksi verkon yli kohdeorganisaation palveluihin. Myös erilaiset sosiaalisen manipulaation (Social Engineering) keinoja käytetään usein tiedon hankintaan. Tässä vaiheessa hyökkääjä pyrkii etsimään tietoa myös kohdeorganisaation käyttämistä laitteista, ohjelmista ja palveluista, löytääkseen mahdollisesti hyödynnettäviä haavoittuvuuksia kohteesta.
2. Aseistaminen: Hyökkääjä räätälöi hyökkäystään juuri kohteeseen sopivaksi. Useimmiten tällä tarkoitetaan hyökkäyksen haitallista asiaa tekevän osuuden laatimista. Tämä voidaan naamioida näyttämään kohteeseen sopivalta hyödylliseltä tiedolta tai toiminnolta. Hyökkääjä hyödyntää kohteen naamioinnissa ensimmäisessä vaiheessa kerättyä tiedustelutietoa, eikä kyseinen hyökkäystapa välttämättä toimi muualla kuin kyseisessä kohdeympäristössä.
3. Toimitus: Hyökkääjä toimittaa edellisessä vaiheessa laatimansa työkalun organisaation sisällä toimivalle kohteelle. Toimitus tapahtuu usein sähköpostin liitetiedostoilla, USB-tikuilla tai ohjaamalla uhri lataamaan haitallinen tiedosto verkkosivulta.
4. Hyväksikäyttäminen: Uhrille toimitettu hyökkäystyökalu aktivoituu ja suorittaa haitallista koodia. Tässä vaiheessa hyödynnetään usein sovelluksen tai verkon tiedossa olevia haavoittuvuuksia, mutta työkalu saattaa myös hyödyntää yksinkertaisesti järjestelmän automaattista koodin ajamista.
5. Asennus: Uhrina olevaan järjestelmään asennetaan etähallintatyökalu, jonka ansiosta hyökkääjä saavuttaa paremman pääsyn ympäristöön jatkossa.

6. Komento ja kontrolli: Usein saastuneet kohdelaitteet soittavat sisäverkosta ulospäin hallintapalvelimelle kysyen näin toimintaohjeita. Tässä vaiheessa hyökkääjällä on käytännössä oma näppäimistö kiinni saastuneessa ympäristössä.
7. Varsinainen toiminta kohteessa: Vasta viimeisessä vaiheessa hyökkääjä pääsee toteuttamaan alkuperäistä agendaansa kohteena olevassa ympäristössä.

Yllä kuvatuista askelista käy selkeästi ilmi, miksi toiminnasta käytetään termiä hyökkäysketju. Jokainen askel on hyökkääjän näkökulmasta merkittävä ja epäonnistuessaan saattaa katkaista koko hyökkäysketjun. (Mts. 4−5.)

Alkuperäinen Lockheed Martinin hyökkäysketju havaittiin kuitenkin riittämättömäksi kuvaajaksi tietyissä tilanteissa, joten monet organisaatiot ovat kehittäneet omia hyökkäysketjujaan, joilla he pyrkivät kuvaamaan hyökkääjän toimintaa tarkemmin. Alkuperäistä Lockheed Martinin hyökkäysketjua on arvosteltu esimerkiksi siitä, että se kuvaa hyökkäysketjua lähtökohtaisesti ulkoapäin tulevien hyökkäyksien näkökulmasta, eikä huomioi lainkaan organisaation sisältäpäin tulevia hyökkäyksiä. Yleensä kuitenkin juuri organisaation sisältä tulevat hyökkäykset ovat kaikista haitallisimpia ja aiheuttavat keskimääräisesti suurimmat kustannukset. (Reidy 2013.)

Alkuperäinen Lockheed Martinin laatimaa hyökkäysketjun kuvausta on kritisoitu myös siitä, että se ohjaa puolustajaa panostamaan hyökkäysketjun katkaisemiseen nimenomaan hyökkäyksen alkuvaiheessa. Lisäksi kritiikkiä on esitetty valheellisesta turvallisuuden tunteesta, johtuen kuvitelmasta, että hyökkäysketjun katkaiseminen, mistä tahansa kohtaa, riittäisi todella pysäyttämään hyökkääjän toiminnan. Malli myös keskittyy erityisesti haittaohjelmalähtöisiin hyökkäysnäkökulmiin ja organisaation sisä- ja ulkoverkon välisen rajan puolustuksen painottamiseen. (Pols 2017). Samaan lopputulokseen on tullut myös Malone (2016), joka kuvaa perinteistä kyberhyökkäysketjua ”Perimeter Breach Kill Chain” termillä eli ulkokehän murtamisen hyökkäysketju. Termi onkin tältä osin kuvaavampi. Osaltaan näiden syiden takia, Lockheed Martinin mallin rinnalle on noussut myös useita muita vastaavia kyberhyökkäysketjuja eri toimijoilta, jotka kuvaavat laajemmin hyökkääjän toimia varsinaisen tunkeutumisen jälkeisinä aikoina. (Malone 2016.)

Malonen esittelemässä mallissa hyökkäysketju on jaettu kolmeen osaan. Perinteiseen kyberhyökkäysketjuun, jolla hyökkääjä pyrkii pääsemään sisään kohde ympäristöönsä. Tätä seuraa sisäinen hyökkäysketju, jossa hyökkääjä tekee jälleen tiedustelua, pyrkii

nostamaan käyttöoikeuksiaan, etsimään ja hyödyntämään tehokkaammin käytettävissä olevia haavoittuvuuksia, levittämään hallintaansa lateraalisesti ja manipuloimaan kohdettaan haluamaansa suuntaan. Kolmannessa vaiheessa hyökkääjä tekee varsinaiset tavoitteena olleet toimenpiteensä kohteessa, joka voi olla esimerkiksi tietojen kaappaamista tai vahingon aiheuttamista järjestelmille tai häiriöitä tarjottaviin palveluihin. (Mt.)

Paul Pols on puolestaan laatinut yhdistetyn hyökkäysketjun, joka yhdistää perinteiseen Lockheed Martinin kyberhyökkäysketjuun muutamia muita hyökkäysketjuja MITRE:n ATT&CK malliin. Kyseinen malli koostuu 18 eri vaiheesta ja se tarjoaa kattavamman ja monitahoisemman kuvauksen hyökkäysketjun eri vaiheista:

1. **Tiedustelu**: Kohteen valinta ja tutkiminen, mahdollisimman laaja tiedon keruu käyttäen aktiivista ja passiivista tiedustelua.
2. **Aseistaminen**: Kohteeseen sopivan haittasovelluksen räätälöinti ja naamioiminen kohteeseen sopivaksi.
3. **Puolustuksen väistäminen**: Toimenpiteet, joita hyökkääjä tekee välttääkseen paljastumisensa.
4. **Toimittaminen**: Hyökkääjä toimittaa kohteelle luodun haittaohjelman
5. **Hyväksikäyttäminen**: Tekniikat, joilla hyökkääjä pääsee hyödyntämään järjestelmissä olevia haavoittuvuuksia ja ajamaan omaa koodiaan.
6. **Pysyvyyden saavuttaminen**: Hyökkääjä pyrkii saavuttamaan pysyvän jalansijan kohdeorganisaation ympäristössä. Yksinkertaisimmillaan haittaohjelma käynnistyy uhrin työaseman käynnistyessä.
7. **Komento ja kontrolli**: Koska kohdeorganisaatioiden ympäristöt ovat usein suljettu paremmin ulkoa sisäänpäin, kuin sisältä ulos, käyttävät hyökkääjät usein erilaisia hallintapalvelimia. Näiltä palvelimilta saastuneet koneet käyvät hakemassa toimintaohjeita. Tässä vaiheessa hyökkääjällä on käytännössä oma näppäimistö kohdeorganisaation sisällä olevassa koneessa.
8. **Pivotointi**: Liikenteen tunnelointi jo hallinnassa olevien kohteiden kautta kohteisiin, joihin suora pääseminen ei ole syystä tai toisesta mahdollista suoraan.
9. **Oikeuksien kasvattaminen**: Erilaisia tekniikoita, joilla hyökkääjä saa kasvatettua hallussa olevien tunnustensa käyttöoikeuksia.
10. **Tutkinta**: Hyökkääjä tutkii kohdeorganisaation järjestelmiä ja sisäistä verkkoa.
11. **Lateraalinen liikkuminen**: hyökkääjä laajentaa hallinnassaan olevia kohteita rinnakkaisiin kohteisiin.

12. **Toteuttaminen**: Hyökkääjä ajaa omaa haitallista koodiaan valitsemassaan kohteessa
13. **Tunnuksiin käsiksi pääseminen**: Hyökkääjän tekniikat, jotka mahdollistavat hyökkääjälle pääsyn ja/tai hallintamahdollisuuden järjestelmään, palveluun tai toimialueen tunnuksiin.
14. **Kohteen manipulointi**: Hyökkääjä suorittaa toimenpiteet, jotka olivat alkuperäisen hyökkäyksen tarkoitusperinä.
15. **Tiedonkeräys**: Hyökkääjän käyttämät tekniikat, joilla hyökkääjä kerää tietoa kohteen verkkoympäristöstä ennen jälkiensä siivousta ja poistumista
16. **Jälkien siivous**: Hyökkääjä toimittaa haltuunsa saamat tiedot tai tiedostot ulos kohdejärjestelmästä ja siivoaa mahdollisesti jättämänsä jäljet ympäristöstä. (Pols 2017, 28.)

Yhdistetty hyökkäysketju on siis kokoelma erilaisia tekniikoita ja toimenpiteitä, joita hyökkääjä yleensä käyttää toimiessaan. Lockheed Martinin kyberhyökkäysketjuun verrattuna, yhdistetty hyökkäysketju sisältää kattavammin hyökkääjän suorittamia toimenpiteitä myös kohdeverkkoon pääsemisensä jälkeen. Tässä mallissa Hyökkäys ei myöskään välttämättä pysähdy estämällä jokin tietty askel ketjusta, mikä vastaa paremmin todellisen maailman tilanteita. (Mt.)

Hunajapurkkien avulla on mahdollista saada indikaatioita mahdollisesta käynnissä olevasta hyökkäyksestä, joka muuten saattaisi jäädä huomaamatta järjestelmän ylläpitäjiltä. Hunajapurkkien kautta saatava ennakkovaroitus kohdistuu Lockheed Martinin kyberhyökkäysketjussa erityisesti tiedusteluvaiheeseen, mikäli hyökkääjä tekee aktiivista tiedustelua suoraan kohdeorganisaation ympäristöihin. Tällöin tiedustelusta tai hyökkäyksestä indikoivat merkit saadaan nimenomaan ympäristön ulkoreunoille sijoitetuilta hunajapurkeilta.

Toinen vaihe, jossa hyökkääjällä on suuri todennäköisyys jäädä kiinni hunajapurkkiin, tapahtuu siinä vaiheessa, kun hyökkääjä on jo päässyt sisään ympäristöön ja alkaa tutkimaan sisäverkkoa tai lateraalisen liikkumisen yhteydessä. Yhdistetyssä hyökkäysketjussa näitä vaiheita kuvataan askelissa 8-15. Näissä vaiheissa hunajapurkki on sijoitettu sisäverkkoon, ja siitä saatavat indikaattorit varoittavat ylläpitäjiä, että muista estävistä toimenpiteistä ja työkaluista huolimatta hyökkääjä on päässyt sisään organisaation

ympäristöön. Ennakkovaroitus saattaa kuitenkin mahdollistaa hyökkääjän pysäyttämisen ennen kuin hän pääsee toteuttamaan lopullisen tavoitteensa hyökkäyksen kohteena olevassa ympäristössä.

3 Hunajapurkit ja niiden jaottelut

Vapaasti suomennettuna Spiznerin määritelmän mukaan hunajapurkki on: ”*Tietojärjestelmän resurssi, jonka arvo on riippuvainen resurssin luvattomasta tai oikeudettomasta käytöstä*” (Spizner 2003). Hunajapurkki on hieman normaalista poikkeava tietoturvatyökalu. Oletuksena ylläpitäjien tavoitteena on pyrkiä tekemään tietoverkoistaan ja sovelluksistaan mahdollisimman tietoturvallisia, jolloin mahdolliset pahantahtoiset toimijat saataisiin pidettyä poissa omista tietoverkoista jo lähtökohtaisesti. Hunajapurkit ovat kuitenkin suunniteltu toimimaan juuri päinvastoin. Kyseiset tuotteet ovat rakennettu matkimaan todellisia palveluita ja sovelluksia tietoverkoissa. Normaaleista sovelluksista ja palveluista poiketen, ne ovat useimmiten jätetty haavoittuvaisen näköisiksi. Tarkoituksena tälle on hämätä mahdollista hyökkääjää tai mahdollistaa tietojen keruun hyökkääjän tai hänen käyttämänsä automatisoidun botin ajamista komennoista ja niiden mahdollisista haavoittuvuuksien hyväksikäytöstä. (What is a honeypot? How it can lure cyberattackers n.d.)

Kirjallisuudessa on mainintoja hunajapurkkia vastaavista sovelluksista jo vuodelta 1990, mutta ensimmäinen julkinen hunajapurkkitoteutus löytyy vuodelta 1998, kun Fred Cohen julkaisi Deception ToolKit-toteutuksensa. DTK oli Perl- ja C-kielellä kirjoitettu kokoelma koodeja, jotka asennettiin käyttöjärjestelmään, saaden käyttöjärjestelmän vaikuttamaan siltä, että se sisältäisi useita tuohon aikaan tiedossa olevia haavoittuvuuksia. DTK tallensi hyökkääjien toimintaa antaen ympäristön ylläpidolle varoituksen mahdollisesti tulossa olevista todellisista hyökkäyksistä jo ennakkoon. (Cohen 1998.)

Cohen (1998) mainitsee hunajapurkin yhdeksi hyödyksi myös hyökkääjän resurssien haaskaamisen. Esimerkiksi hyökkääjä käyttää omia voimavarojaan turhaan, kun hän yrittää avata Unixin salasanatiedolta vaikuttavaa DTK:n tarkoituksella luomaa harhautustiedostoa. Nykyaikana hyökkääjät ovat siirtyneet käyttämään automatisoituja hyökkäysrobotteja, jotka toimivat tiettyjen ehtojen mukaisesti tehden ohjelmoituja

toimenpiteitä väsymättä. Samalla myös laskentatehon hinta on laskenut merkittävästi, jolloin hyökkääjän ”uuvuttamisen” merkitys on vuosien saatossa vähentynyt jonkin verran. (Mt.)

Samoilla jalanjäljillä jatkoi myös 1999 vuonna perustettu Honeynet-projekti. Projekti alkoi muutaman tietoturva-asiantuntijan postituslistasta. Projektin tavoite on vapaasti suomennettuna ”Oppia työkaluista, taktiikoista ja motiiveista, jotka liittyvät tietoverkkoihin ja jakaa opittuja asioita”. Honeynet Projekti on nykyään voittoa tavoittelematon tietoturvallisuutta tutkiva organisaatio, jonka toiminnassa on mukana lukemattomat innokkaat vapaaehtoiset. (ABOUT US n.d.)

Yhtenä reaalimaailman esimerkkinä hunajapurkkien käytöstä toimii Simo Kemppaisen tutkimus, Tietoturvaloukkausten analysointi hunajapurkkijärjestelmien avulla. Tutkimuksessaan Kemppainen asensi hunajapurkkeja julkiseen verkkoon 49:ksi päiväksi ja sai kerättyä dataa lähes puolesta miljoonasta kirjautumisyrityksestä. Samalla Kemppainen sai kerättyä dataa hyökkääjien käyttämistä käyttäjätunnus ja salasanapareista ja hunajapurkissa ajetuista komennoista. Kerätyn datan perusteella vaikuttaisi, että suurin osa hyökkäyksistä olisi automaattisten ”haistelijoiden” suorittamia, sillä vain murto-osassa onnistuneista kirjautumisista, hyökkääjä päätyi ajamaan varsinaisia komentoja palvelimilla. Suurin osa ajetuista komennoista keskittyi lataamaan ja ajamaan erillisen ohjelman kohdekoneelle ja kirjautumaan ulos. Hyökkääjien käyttämien automaattisten hyökkäystyökalujen toiminta etenee siis pitkälti hyökkäysketjujen mukaisesti. (Kemppainen 2016.)

Hunajapurkkeja on mahdollista jaotella useiden eri ominaisuuksien perusteella. Kirjallisuuslähteissä korostuvat erityisesti jako eri vuorovaikutustasojen perusteella. Tämän lisäksi jakoja voidaan suorittaa esimerkiksi käyttötarkoituksen, hunajapurkin sijoituksen verkkoon, asiakas- ja palvelintyyppisyyden tai sen mukaan koostuuko toteutus yhdestä vai useammasta laitteesta. Seuraavissa kappaleissa tutustutaan tarkemmin edellä mainittuihin jakoihin.

3.1 Käyttötarkoitus

Hunajapurkkien käyttötarkoitus on jaettavissa kahteen pääryhmään: Tutkimuksellisiin ja tuotannollisiin hunajapurkkeihin. Nimensä mukaisesti tutkimuksellisia hunajapurkkeja rakentavat ja käyttävät tutkijat, jotka pyrkivät oppimaan lisää blackhat-yhteisöiden käyttämistä tekniikoista, taktiikoista ja menetelmistä. Opittuja tietoja käytetään parempien IDS-työkalujen kehittämiseen ja antamaan tietoa uusista hyökkäysmenetelmistä ja mahdollisista ohjelmointivirheistä nykyisissä protokollissa. Kerätystä datasta voidaan saada myös havaintoja uusista nollapäivähaavoittuvuuksista tai sitä voidaan käyttää forensiikka- ja tilastolliseen analyysiin. Esimerkiksi jos hyökkääjiltä tulee huomattava määrä tietyntyyppisiä skannauksia, jotka keräävät tietoa nimenomaan jonkun tietyn protokollan käytöstä, on mahdollista, että kyseisestä protokollasta on löytynyt jokin hyödynnettävä haavoittuvuus. Tämä ei välttämättä vielä ole tietoturvayhteisön tiedossa. (Jain & Singh 2011.)

Tutkimukselliset hunajapurkit on laitettava kiinni verkkoon siten, että havainnoitavat hyökkääjät pääsevät verkon yli kiinni hunajapurkkiin, muttei kuitenkaan niin ilmeisen avonaiseksi, että hyökkääjä tulisi epäluuloiseksi ja huomaisi olevansa monitoroinnin alaisena tai jopa suoraan käyttävänsä hunajapurkkia. Hunajapurkin tulisikin vaikuttaa toiminnoiltaan samankaltaiselta kuin vastaavassa verkon osassa oleva todellinen tuotannossa oleva palvelin. (Mt.)

Tuotannolliset hunajapurkit sijoitetaan yleensä osaksi organisaatioiden tietoturvainfrastruktuuria eli käytännössä tuotantojärjestelmien joukkoon tai rinnalle. Tämä mahdollistaa ennakkovaroituksien saamisen ennen varsinaisten hyökkäystoimenpiteiden alkua. Tuotannolliset hunajapurkit täydentävät osaltaan jo olemassa olevaa tietoturvainfrastruktuuria. Niitä voidaan myös tietyin edellytyksin käyttää validoimaan esimerkiksi yrityksen käyttämiä tunkeutujan tunnistamisjärjestelmiä (IDS). (mt.)

Tuotannollisissa hunajapurkeissa pyritään usein saamaan ennakkohavaintoja erityisesti hyökkääjän lateraalisesta liikkumisesta organisaation sisäverkossa. Mikäli hyökkääjä on päässyt tavalla tai toisella jo aikaisemmin kiinni yrityksen sisäverkkoon, hän pyrkii todennäköisesti etsimään lisätietoa verkosta ja siihen liitetyistä palveluista. Joissain tilanteissa organisaation sisäverkossa saattaa pyöriä esimerkiksi vanhempia haa-

voittuvia legacy-järjestelmiä, joita ei ole välttämättä käytännön syistä tai tietämättömyydestä johtuen saatu eriytettyä omaan suljettuun ympäristöönsä tai edes omaan verkkosegmenttiinsä. Organisaatioiden sisäverkot ovat huomattavasti haavoittuvaisempia sen jälkeen, kun pahantahtoinen toimija on saanut jalansijan sisäverkkoon verrattuna kokonaan organisaation ulkopuolelta tulevaan hyökkäykseen. Osaltaan tähän vaikuttaa hyökkäyspinta-alan kasvaminen ja estävien toimenpiteiden keskittyminen ulkorajalle. Harva organisaatio tarjoaa kaikkia sisäisiä palveluitaan suoraan julkiseen verkkoon ainakaan tarkoituksella. Tietyissä määrin sisäverkkoa on mahdollista suojata erilaisilla IDS toteutuksilla, mutta tuotannollisilla hunajapurkeilla voidaan havaita suoraan liikenne sinne kuulumattomista osoitteista. Samalla voidaan nähdä suoraan hyökkääjän käyttämiä menetelmiä, joiden perusteella muita palveluita on mahdollista lähteä kovettamaan tarpeen mukaisesti, kerätä tietoja hyökkääjästä ja pyrkiä estämään tämän toimintaa yleisesti. (Spitzner 2003; Jain 2011.)

Tuotannolliset hunajapurkit pyrkivät havaitsemaan tietoturvapoikkeamia organisaation sisäverkossa meneillään olevien normaalien tapahtumien joukossa. Normaaleja tapahtumia organisaation sisäverkossa ovat esimerkiksi kirjautuminen järjestelmään, jonkun palvelun kaatuminen tai sähköpostin lähettäminen. Tietoturvapoikkeama eroaa normaaleista tapahtumista siten, että ne indikoivat organisaation järjestelmien tai datan olevan mahdollisesti vaarantuneina. (Pham 2001, 2−5.)

Molemmilla käyttötyypeillä on tietyt perustoiminnot, joiden tulisi olla kunnossa käyttötyypistä riippumatta. Näistä tärkein on lokien tuottaminen, joka mahdollistaa hunajapurkin keräämien tietojen varsinaisen hyödyntämisen. Lokitus eli lokitietojen tallennus ja hyödyntäminen, olisi hyvä saada eriytettyä varsinaisen hunajapurkin toiminnallisuudesta, jotta hyökkääjä ei hunajapurkin paljastuessa pääsisi ainakaan suoraan tuhoamaan jättämiään jälkiä. Lokitusta voidaan tehdä suoraan erilliseen tekstitiedostoon, mutta varsinkin laajemmassa käytössä tietokantaan tallennuksen mahdollisuutta tulisi harkita. Tietokantaan tallentaminen mahdollistaa lokitietojen joustavamman käsittelyn, hakemisen ja rajaamisen, mutta tuo samalla mukaan vaatimuksen ylimääräisestä komponentista tietokannan muodossa. (Jain 2011.)

Lokituksen lisäksi hunajapurkin tulisi pystyä tarvittaessa hälyttämään epäilyttävästä toiminnasta ylläpitäjiään. Näin ylläpitäjät saavat tiedon mahdollisista tietoturvauhista

automaattisesti, eikä hunajapurkin lokeja tarvitse käydä erikseen tarkistamassa säännöllisin väliajoin. Hälytykset on mahdollista toteuttaa esimerkiksi konsolille tulostettavalla hälytysviestillä, mutta käytännöllisempää olisi toimittaa hälytykset esimerkiksi sähköpostiviestillä suoraan ylläpitäjän sähköpostiin tai käytettävissä oleviin tiketöintijärjestelmiin. Mikäli käytössä on SMS GW, voidaan hälytyksestä ilmoittaa ylläpitäjille esimerkiksi https:n tai sähköpostin yli tapahtuvalla tekstiviestin lähetyksellä. Yksi nykyään yleistynyt vaihtoehto on hyödyntää API-rajapintaa ja viedä tieto tätä kautta johonkin jo olemassa olevaan järjestelmään. (Mt.)

Näiden kahden ominaisuuden lisäksi olisi hyvä, että hunajapurkki olisi konfiguroitavissa tarpeen mukaiseksi. Näin hunajapurkki voidaan tarvittaessa siirtää uuteen ympäristöön ja uusien hunajapurkkiympäristöjen luonti onnistuu helpommin. Hunajapurkkien olisi myös hyvä pystyä selvittämään hyökkäyksen lähdettä esimerkiksi reitinselvityksen avulla. Näin saadaan tieto reitistä hyökkääjän osoitteesta kohteeseen. (mt.)

Hunajapurkkeja jaotellaan toisinaan myös käyttötarkoituksen perusteella houkuttimiin ja sensoreihin. Houkuttimien suurin hyöty on tuotantoympäristöihin sijoitettavissa hunajapurkeissa, jossa niiden tehtävänä on nimensä mukaisesti pyrkiä houkuttelemaan haitallista toimijaa pois todellisista arvokkaista kohteista ja ohjata tätä tuhlaamaan resurssejaan houkuttelevan näköiseen kohteeseen. Todellisuudessa kohteet ovatkin vain hyökkääjän näkökulmasta arvottomia houkutuslintuja. Tämän kaltaisen hunajapurkin täytyy vaikuttaa ulkoisesti juuri siltä, mitä hyökkääjä pyrkii organisaation verkosta etsimään. Houkuttimen tarkoituksena on ohjata hyökkääjää harhaan ja tuhlaamaan resurssejaan. (Higgins 2018, 28−29.)

Sensorina toimiva hunajapurkki puolestaan pyrkii keräämään tietoa hyökkääjän liikkeistä ja tämän suorittamista toimenpiteistä. Tutkimukselliset hunajapurkit ovatkin usein nimenomaan sensoreina toimivia ja niiden tehtävänä on kerätä tietoa hyökkäävien tahojen käyttämistä menetelmistä ja käytösmalleista. Myös tuotannolliset hunajapurkit voidaan periaatteessa nähdä sensoreina, niiden hälyttäessä hyökkääjän liikkeistä ja toimista alueilla ja palveluissa, joissa normaaleissa käyttötilanteissa toimintaa ei tulisi olla. (Mts. 28-29.)

3.2 Käytettävät OSI-mallin tasot

Suunniteltaessa hunajapurkkituotetta, on tärkeää pohtia, millä OSI-mallin tasoilla toimintaa halutaan simuloida, tallentaa ja analysoida. OSI-mallilla kuvataan tiedonsiirrossa käytettävien protokollien keskinäistä toimintaa jaettuna seitsemälle eri tasolle. Tasot on jaettu niin, että alemmat kerrokset tarjoavat palveluitaan yläpuolelleen ja käyttää alapuolellaan olevia palveluita itse. Teoriassa hunajapurkilla on mahdollista imitoida mitä tahansa OSI mallin tasoa. (Grimes 2005a.)

Mikäli hunajapurkkina on todellinen käyttöjärjestelmä, sillä voidaan melko helposti toteuttaa fyysisen- ja siirtokerroksen asettamat vaatimukset. Eri hyökkääjät käyttävät usein eri kerroksia hyödyksi toiminnassaan. Jotkut hyökkääjät saattavat keskittyä puhtaasti esimerkiksi sovellustason muistin ylivuotoa käyttäviin hyökkäyksiin, kun taas jotkut pyrkivät vääristämään ja muokkaamaan IP-paketteja verkkotasolla. Nykypäivänä useat hyökkääjät erikoistuvat tietyn tyyppisiin hyökkäyksiin, eivätkä välttämättä lähde laajentamaan muille osa-alueille itse. Valitettavasti ennakkoon ei voi tietää, minkälainen hyökkääjä sattuu kohdalle ja minkälaisia heikkouksia hän pyrkii hyödyntämään. Lähtökohtaisesti ajatusmalli, jossa hyökkääjä pyrkii pääsemään käsiksi esimerkiksi arkaluontoisiin tietoihin, on tietyissä tilanteissa väärä. Jotkut hyökkääjät saattavat vain pyrkiä aiheuttamaan ongelmia tuotannollisiin ympäristöihin ja saada aikaan erilaisia DoS-hyökkäyksiä kohteitaan vastaan. Tämän kaltaisissa hyökkäyksissä saatetaan usein käyttää OSI-mallin matalimpia tasoja. (Mt.)

Hyökkääjän motiivina saattaa esimerkiksi olla kiristää yritystä, uhkaamalla estää todellisten asiakkaiden pääsyn yrityksen tarjoamiin palveluihin aiheuttaen näin taloudellista haittaa yritykselle. Näin toimi esimerkiksi eräs ryhmittymä, joka esitti ylläkuvatun kaltaisia kiristyksiä kohteilleen. Mikäli kohde kieltäytyi maksamasta, yritys joutui palvelunestohyökkäyksen kohteeksi tilanteissa, joissa katkoksesta aiheutui huomattavaa taloudellista ja maineellista haittaa. Alkuperäinen ryhmittymä jäi kiinni, mutta vielä tämän jälkeenkin toinen ryhmä päätyi hyödyntämään aiemmin hankittua mainetta lähettämällä perättömiä uhkauksia alkuperäisen ryhmän nimissä. Kun uhrit tutkivat saamaansa uhkausta, he löysivät tietoja toteutuneista hyökkäyksistä ja päätyivät usein maksamaan ilman, että nimellä ratsastavan ryhmän tarvitsi edes koskaan toteuttaa uhkauksiaan. (Hyppönen & Tuominen 2019.)

Organisaation näkökulmasta on siis parasta varautua kykyjensä ja resurssiensa puitteissa mahdollisimman laaja-alaisesti kaikilla OSI-mallin kerroksilla tapahtuviin hyökkäyksiin. Tämän vuoksi myös hunajapurkin tulisi organisaation tarpeen mukaan olla mahdollisimman monipuolinen. Mitä monipuolisemmin se pystyy imitoimaan OSI-mallin kerroksia, sitä todennäköisemmin hyökkääjä erehtyy luulemaan sitä todelliseksi kohteeksi. (Grimes 2005a.)

Matalammat kerrokset ovat usein staattisia ja ennakkoon määritetty. Matalampia kerroksia saattaa olla teknisesti vaikeampaa ymmärtää, mutta niiden vakiomuotoisuuden takia niille löytyy valmiiksi tarjolla useita eri hunajapurkkiratkaisuja. Tällaisten ratkaisujen ylläpitäminen on myös yleensä vaivattomampaa korkeampien kerroksien ratkaisujen sijaan. (Mt.)

OSI-mallin korkeampien kerroksien päällä, esimerkiksi sovellustasolla pyörivät ratkaisut, ovat hieman haastavampia mallintaa. Jos kyseinen ratkaisu mallintaa esimerkiksi FTP-palvelinta, huomioon otettavien asioiden lista on huomattavasti pidempi, kuin matalampien kerroksien päällä toimivissa ratkaisuissa. Kehityksessä tulee ottaa huomioon, kuinka pitkälle todellisen maailman palvelua kyseinen hunajapurkki halutaan mallintaa. Tähän vaikuttaa luonnollisesti myös hunajapurkin tarve eli onko kyseessä tuotannollinen vai tutkimuksellinen hunajapurkki. Luonnollisesti tutkimuksellisen hunajapurkin olisi tärkeämpää imitoida todellisen maailman esikuvaansa mahdollisimman tarkkaan, jotta hyökkääjä ei tulisi epäluuloiseksi liian nopeasti. Yksi mahdollisuus on tarjota hunajapurkissa todellista palvelua, mutta tämä lisää huomattavasti riskejä hunajapurkin väärinkäytölle. Hyökkääjä saattaa onnistua väärinkäyttämään tarjolla olevaa palvelua jollain odottamattomalla tavalla ja esimerkiksi hyödyntämään sitä hyökkäyksiin toisia palveluita tai laitteita vastaan. (mt.)

3.3 Vuorovaikutteisuus

Hunajapurkit voidaan kahteen tai kolmeen ryhmään vuorovaikutteisuuden tason mukaan, sen perusteella kuinka paljon valtaa ja tilaa toimia ne hyökkääjälle antavat. Useat korkean vuorovaikutteisuuden hunajapurkit toimivat kaikilla OSI-mallin tasoilla ja niissä pyörii todellisen maailman sovelluksien tai palveluiden näköisversiot tai peräti

käyttöjärjestelmät. Kaikista korkeimman tason hunajapurkeissa, palveluun emuloidaan sisältöä ja muokataan esimerkiksi tiettyjä asetuksia. Näin muokkauspäivämäärät pysyvät tuoreina ja hyökkääjälle on helpompaa uskotella hänen yrittävänsä murtautua todelliseen, käytössä olevaan palveluun. Korkeamman vuorovaikutustason hunajapurkeissa korostuu riskienhallinnan merkitys. Koska ratkaisu keskustelee vapaammin hyökkääjän kanssa, on sen oltava myös pidemmälle ohjelmoitu. Joissain tilanteissa korkeamman vuorovaikutuksen hunajapurkeissa voidaan hyödyntää myös todellisia palveluita, joiden lokittamisen tasoa on vain nostettu taustalla. Korkeampi vuorovaikutustaso kuitenkin antaa hyökkääjälle huomattavan paljon lisää hyökkäyspinta-alaa hunajapurkkia itseään kohtaan. Pelkän käyttöjärjestelmän lisäksi hyökkääjälle on käytettävissään myös mahdollisten asennettujen sovellusten ja palveluiden olemassa olevat haavoittuvuudet, jotka pahimmillaan saattavat vaarantaa hunajapurkin lisäksi myös muita kohteita. (Yahyaoui 2014.)

Korkeampi vuorovaikutustaso mahdollistaa kuitenkin hyökkääjän helpomman vakuuttamisen todellisesta ympäristöstä, jonka vuoksi tämän kaltainen ratkaisu sopiikin erityisesti tutkimuksellisiin hunajapurkkeihin, joilla yritetään havaita esimerkiksi uusia nollapäivähaavoittuvuuksia. Korkeammalla vuorovaikutustasolla on siis mahdollista saada kerättyä enemmän tietoa, mutta samalla kasvavat hunajapurkkiympäristöstä karkaamisen riskit. Korkean vuorovaikutustason hunajapurkiksi voidaan myös katsoa kokonainen käyttöjärjestelmätasoinen hunajapurkki. Tällainen hunajapurkki asennetaan kokonaan omalle palvelimelleen joko virtuaalisesti tai fyysisesti. Tällainen toteutus vaatii kuitenkin huomattavan suurta työpanosta käyttöönotossa ja ylläpidossa. Siinä on myös suuri riski, että hyökkääjä saisi kaapattua kyseisen ympäristön hallintaansa. (Mt.)

Yleensä hunajapurkit jaetaan kahteen vuorovaikutustasoon, mutta joissain tilanteissa voidaan puhua myös kolmannesta, keskitason vuorovaikutuksella olevasta hunajapurkista. Tällaiset järjestelmät ovat käytännössä matalan interaktion hunajapurkkeja, mutta saattavat tarjota hyökkääjälle valheellista näkyvyyttä laajempaan, aitoa käyttöjärjestelmää ulkoisesti muistuttavaan ympäristöön, johon hyökkääjälle ei kuitenkaan anneta pääsyä. Hyökkääjä ei myöskään pääse vaikuttamaan hunajapurkin ulkopuoliseen ympäristöön, koska vaikutusmahdollisuudet ovat tiukasti rajoitetut. Keskitason eli emuloidut hunajapurkit saattavat kuitenkin olla tietyissä tilanteissa hyökkääjälle

helpommin havaittavia. Tällaisia tilanteita saattaa esiintyä esimerkiksi, jos hunajapurkki kuvastaa jotain tietyn valmistajan verkkolaitetta, mutta hyökkääjä pääsee kuitenkin näkemään, että taustalla pyörii esimerkiksi Linux- tai Windows-palvelin. Emuloitujen hunajapurkkien hyvät puolet ovat kuitenkin kattavia. Hyökkääjän on vaikeampaa ottaa tällainen hunajapurkki haltuunsa, havaitessaan hyökkäyksensä kohdistuvan vain hunajapurkkiin. Mikäli näin kuitenkin pääsisi tapahtumaan, emuloidun ympäristön hyödyntäminen hyökkäyksissä muita kohteita vastaan on huomattavasti haastavampaa, sillä kyseessä ei ole kokonainen järjestelmä. Ne ovat myös korkean vuorovaikutuksen hunajapurkkeihin verrattuna helpompia ylläpitää ja usein myös kustannustehokkaampia. Lokitukset ja monitorointi onnistuu parhaimmillaan suoraan isäntäkoneelta ja vanhan ongelmia kohdanneen hunajapurkin palautus onnistuu yleensä suoraviivaisesti. (Grimes 2005a.)

Yksinkertaisimmillaan matalan vuorovaikutustason hunajapurkki saattaa olla vain jotain tiettyä porttia kuunteleva prosessi, joka tallentaa porttiin tulevat paketit erilliseen lokitiedostoon. Tämänkaltainen yksinkertainen hunajapurkki on mahdollista toteuttaa käyttämällä esimerkiksi Netcat-työkalua, jota voidaan käyttää tiettyjen porttien kuunteluun ja putkittaa saapuvat paketit tekstitiedostoon. Tällaista voisi käyttää esimerkiksi verkon tarkkailussa havaitsemaan, yrittääkö joku ottaa yhteyksiä porttiin 22, joka on SSH-yhteyksien oletusportti. Mikäli lokitiedosto luotaisiin, osaisi järjestelmä nostaa automaattisesti hälytyksen. (Mt.)

3.4 Sijoittaminen verkkoon

Hunajapurkin sijoituspaikka verkossa riippuu hyvin pitkälle käytössä olevan hunajapurkin tyypistä ja käyttötapauksesta. Yleensä sijoituspaikkojen suhteen vaihtoehtoja on lähtökohtaisesti kolme:

- **Ulkoinen**, joissa hunajapurkki sijoitetaan palomuurin ulkopuolelle ja avoimeksi julkiseen internetiin.
- **DMZ** (demilitarized zone), jossa hunajapurkki on sijoitettuna DMZ:lle.
- **Sisäinen**, jossa hunajapurkki sijaitsee organisaation sisäverkossa

Ulkoinen sijoituspaikka soveltuu parhaiten tutkimuksellisiin hunajapurkkeihin, jossa tavoitellaan suurta otantaa. Koska hunajapurkki on avoimena julkiseen internettiin, on

hyökkääjillä suurempi mahdollisuus löytää kyseinen hunajapurkki ja päätyä hyökkäämään sitä vastaan. Kyseinen sijoituspaikka on palomuurin ulkopuolella, ei palomuuri rajoita mahdollisen hyökkäysliikenteen määrää tyypistä ja laadusta riippumatta. Ulkopuolelle sijoitettu hunajapurkki kerää täten suurimman määrän tietoa, mutta on myös samalla haavoittuvaisemmassa asemassa, kuin palomuurin suojiin sijoitettu hunajapurkki eli hyökkääjän pakeneminen ulkopuolelle sijoitetusta hunajapurkista ja hunajapurkin väärinkäyttö on todennäköisempää tämän takia. (Grimes 2005a.)

Sisäinen hunajapurkki sijoitetaan palomuurin sisäpuolelle varsinaiseen tuotantoverkkoon. Tällainen hunajapurkki pyörii todellisten palveluiden ja työasemien joukossa. Tämän takia hunajapurkille ei oletuksena tule yhtä paljon liikennettä, kuin ulkoiselle hunajapurkille. Hyökkääjällä tulee olla hunajapurkkiin törmätäkseen jo valmiiksi pääsy organisaation sisäverkkoon. Sisäinen hunajapurkki toimiikin parhaiten tuotannollisena hunajapurkkina, jolloin järjestelmän ylläpitäjät saavat sen kautta havainnon mahdollisesta hyökkääjän pääsystä organisaation sisäverkkoon. Sisäisessä hunajapurkissa on kuitenkin omat riskinsä. Mikäli hyökkääjä pääsee murtautumaan hunajapurkin sisältä ulos ja hän onnistuisi väärinkäyttämään hunajapurkkia, olisi hänellä pahimmillaan suuremmat vaikutusmahdollisuudet organisaation sisäverkon palveluihin, kuin mitä hänellä olisi ollut ilman hunajapurkkia. Kyseinen riski tulee ottaa aina huomioon, ennen sisäverkkoon sijoitetun hunajapurkin käyttöönottoa. (Mt.)

Koska sisäverkkoon sijoitettavat hunajapurkit ovat yleensä tuotannollisia ja niiden tehtävänä on toimia ennakkovaroitusjärjestelmänä, ei sisäverkkoon kannata välttämättä sijoittaa korkean vuorovaikutteisuusasteen hunajapurkkia, niiden suuremman hunajapurkista pakenemisriskin vuoksi. Sisäverkkoon sijoitettavissa tuotannollisissa hunajapurkeissa tarpeisiin riittää usein jo pelkkä matalan- tai keskitason vuorovaikutteisuuden hunajapurkki. Hyökkääjän on vaikeampaa päästä pakenemaan matalan- tai keskitason vuorovaikutteisuuden hunajapurkeista, ja vaikka ne paljastuvatkin helpommin hunajapurkeiksi korkeamman vuorovaikutustason tuotteisiin verrattuna, on hyökkääjästä usein jo jäänyt ympäristöön jotain varoittavia jälkiä tämän käynnistä. Koska matalan vuorovaikutteisuustason tuotannollisilla hunajapurkeilla ei normaalisti oletuksena ole juuri lainkaan liikennettä, riittää tietyissä tilanteissa jo yritys tutkia tai päästä kyseiselle hunajapurkille, kertomaan järjestelmän ylläpitäjille mahdollisen hyökkääjän läsnäolosta organisaation sisäverkossa. (mt.)

DMZ:lle sijoitettu hunajapurkki on kahden yllä selitetyn välimuoto, joka sijoitetaan yleensä organisaation DMZ:lle (demilitarized zone). DMZ sijaitsee palomuurin takana, mutta yhteydet ovat sallittu vain ulkoverkon suuntaan. Näin ollen sijoitettaessa hunajapurkkia tänne, tulee muut DMZ:lla olevat laitteet ja palvelut suojata siltä varalta, että hyökkääjä onnistuu saamaan hunajapurkin hallintaansa ja päätyy hyödyntämään sitä omiin tarkoitusperiinsä. (Norrgård 2013, 9−10.)

3.5 Hunajapurkki vai hunajaverkko

Hunajaverkot ovat verkkoja, jotka koostuvat useista yhteen liitetyistä hunajapurkeista. Niiden avulla on mahdollista saada kerättyä kattavampaa tietoa hyökkäysmenetelmistä ja toimintamalleista, kuin yksittäisellä hunajapurkilla. Niiden avulla on mahdollista esimerkiksi tutkia hyökkäyksen etenemistä yhden hunajapurkin kautta toiselle, niiden simuloidessa useita erilaisia ympäristöjä mahdollistaen kattavamman alueen erilaisia hyökkäyspinta-aloja. Tästä syystä hunajaverkkojen käyttämät hunajapurkit ovat usein korkean vuorovaikutustason hunajapurkkeja. Tämä mahdollistaa hyökkääjien kattavamman seurannan ja paremman tiedon keruun, mutta hallinnointiin ja ylläpitoon käytettävän työnmäärä kasvaa huomattavasti. Tämän lisäksi hyökkääjällä on suurempi todennäköisyys päästä karkaamaan hunajapurkkien sisältä, verrattuna yksittäisiin matalan vuorovaikutustason hunajapurkkeihin. Pahimmillaan hyökkääjä pääsisi hyödyntämään hunajapurkkeja haitallisesti. Samalla tavoin kuin hunajapurkkien osaltakin, kaikki yhteydet hunajaverkkoon kuuluviin laitteisiin voidaan lähtökohtaisesti olettaa haitallisiksi. (Higgins 2018, 35−36.)

Hunajaverkkojen yksi suuri etu on mahdollisuus linkittää yhteydet Honeywall-nimisen yhdyskäytävän läpi. Kyseinen laite yhdistää ulkopuoliset järjestelmät hunajaverkkoon, mutta kaikki liikenne hunajaverkkoon kulkee Honeywall yhdyskäytävän läpi. Toisen sukupolven Honeywall on pyritty piilottamaan hyökkääjän näkyvistä. Paketit kulkevat sen läpi, mutta pakettien elinajan rajoitin (Time to Live) ei vähene tästä huolimatta. Honeywall toimii OSI-mallin toisella kerroksella siltaavana laitteena. Kyseinen tuote mahdollistaa täyden monitoroinnin ja voi jopa toimia niin kutsuttuna keyloggerina, tallentaen hyökkääjän näppäinten painallukset. Lyhyesti, hunajaverkon avulla on mahdollista saada kaikki hyökkääjän hunajaverkossa tekemät toimenpiteet näkyviksi. Hyökkääjän kattavassa hunajaverkkoympäristössä tekemien toimien perusteella on

mahdollista kerätä arvokasta tietoa, joiden avulla ympäristöjä on mahdollista laatia tietoturvallisemmiksi jatkossa. (Mts. 35−36.)

4 Skaalautumista helpottavat työkalut

Koska toimeksiannon yhtenä tavoitteena oli myös tutkia hunajapurkkituotteiden skaalautuvuutta, päätettiin tutkimukseen lisätä myös tähän tarkoitukseen usein käytettyjä valmiita työkaluja. Käsiteltävien työkalujen avulla on mahdollista tarvittaessa nostaa suuria määriä hunajapurkkituotteita pystyyn mahdollisen pienellä manuaalisella työllä ja mahdollisuuksien mukaan vähentää palveluiden pyörittämiseen vaadittujen resurssien määrää.

4.1 Ansible

Puhuttaessa tarpeesta nostaa pystyyn suuria määriä hunajapurkkituotteita ja hallinnoida niitä, herää myös kysymys näiden tehtävien automatisoinnista. Tähän tarpeeseen vastaa Redhatin rahoittama avoimen lähdekoodin yhteisöprojekti Ansible. Ansiblen avulla on mahdollista pystyttää, ylläpitää ja orkestroida palvelimia ja palveluita laaja-alaisesti. Tuotteesta löytyy kattava valikoima erilaisia moduuleita ja lähes kaikkiin yritysverkoissa oleviin tarpeisiin löytyy sopiva moduuli valmiina. (How Ansible Works 2020.)

Käytännössä Ansiblen toiminta perustuu kahdenlaisiin noodeihin. Ohjausnoodiin ja kontrolloitaviin noodeihin. Vastaavista muista tuotteista Ansible erottuu erityisesti sillä, ettei tuote vaadi erillisien lisäsovellusten asennusta kontrolloitaville kohdelaitteille. Ainoastaan ohjausnoodilla tulee olla Ansible asennettuna. Yhteydet kohdelaitteisiin hoituvat SSH yhteyksien kautta yleensä erillisten SSH-avaimien kautta. Tuki pelkän salasanan käyttämiselle löytyy myös, mikä on usein koekäytössä helpoin toteuttaa. Tuote tukee myös useita muita autentikaatiomenetelmiä. Ansible toimii yhdistämällä ohjausnoodilta kohdelaitteeseen SSH:lla ja puskemalla sinne pienen ohjelman, jota kutsutaan Ansible moduuliksi. Nämä moduulit toimivat ohjeena halutulle tilalle kohteessa. Tämän jälkeen Ansible toteuttaa toivotut muutokset kohteessa SSH:n yli ja poistaa moduulit. (Mt.)

Ansibleen on mahdollista tallentaa inventaarioita (Inventory). Inventaariot ovat listauksia hallinnoitavista laitteista, joita voidaan kategorisoida käyttäjän tarpeiden mukaisesti esimerkiksi web-palvelimiin ja tietokantapalvelimiin. Tämän jälkeen laitteille on mahdollista ajaa keskitetysti esimerkiksi tietty versio käytössä olevasta sovelluksesta ja ohjeistaa kyseinen muutos koskemaan ainoastaan tietokantapalvelimia. Komennoissa on mahdollista käyttää tilaperusteisia resurssimoduuleja, mutta myös puhtaita komentokehotekomentoja voidaan tarvittaessa käyttää. (mt.)

Ansiblen varsinainen arvo nousee esiin siinä vaiheessa, kun yhtälöön lisätään pelikirjojen käyttö (playbooks). Pelikirjat ovat YAML-merkintäkielellä kirjoitettuja yksinkertaisia ohjeita, joiden perusteella Ansible toteuttaa tehtäviä. Pelikirjoihin on esimerkiksi mahdollista määritellä kohteeksi aiemmin mainittujen tietokantapalvelimien ryhmä, joille asetetaan rooli Tietokanta. Roolissa voi olla erilaisia määrityksiä mutta yksinkertaisimmillaan rooli voi olla, että rooliin kuuluvalla kohteella pitää olla asennettuna tietokantasovellus. Mikäli näin ei vielä ole, asentaa Ansible sen. Kohteelle on mahdollista asettaa useampia rooleja, esimerkiksi kaikilla palvelin rooleilla tulee olla asennettuna käytössä oleva valvontasovellus. Tällöin pelikirjaa ajettaessa tietokantapalvelimille asentuisi ensin palvelin-roolissa määritetyt ohjelmat, jonka jälkeen tietokantapalvelimille haluttu tietokantasovellus asentuisi palvelimelle. (mt.)

Samaan tapaan Ansiblen avulla on mahdollista valita hunajapurkki tuotteita nostettavaksi pystyyn nopeasti tarpeen niin vaatiessa. Myös hunajapurkkien hallinnointi ja erilaiset tarvittavat konfiguraatiomuutokset on mahdollista toteuttaa keskitetysti Ansiblen avulla. Pelikirjoihin on mahdollista laatia valmiiksi ohjeet hunajapurkkien nostamiseksi esimerkiksi tiettyjä palveluita kuvastamaan tiettyyn verkon osaan. Havaittaessa toisenlaista tarvetta muualla, voidaan ajaa toinen pelikirja, joka nostaa pystyyn erilaista palvelua kuvastavan hunajapurkin eri osaan sisäverkkoa.

4.2 Konttiteknologiat

Hunajapurkkituotteita olisi hyvä saada sijoitettua kuvastamaan useita erilaisia palveluita eri puolille organisaation verkkoinfrastruktuuria. Tällaisessa tilanteessa on mah-

dollista käyttää erillisiä fyysisiä tai virtuaalisia palvelimia, mutta tällaiset ratkaisut kuluttavat usein paljon resursseja. Mikäli hunajapurkkeja halutaan sijoittaa suuria määriä alkaa resurssien tarve nopeasti kumuloitumaan huomattavan suureksi.

Perinteisten fyysisten ja virtuaalisten palvelinten rinnalle on viime vuosina alkanut ilmestymään erilaisia konttiratkaisuja. Koska varsinkin matalan vuorovaikutustason hunajapurkit ovat usein yksinkertaisia ja melko kevyitä, ei niiden pyörittämiseen kannata välttämättä käyttää erillistä virtuaalikonetta, vaan pyöritettävät hunajapurkit saattavat olla kustannustehokkainta kontittaa.

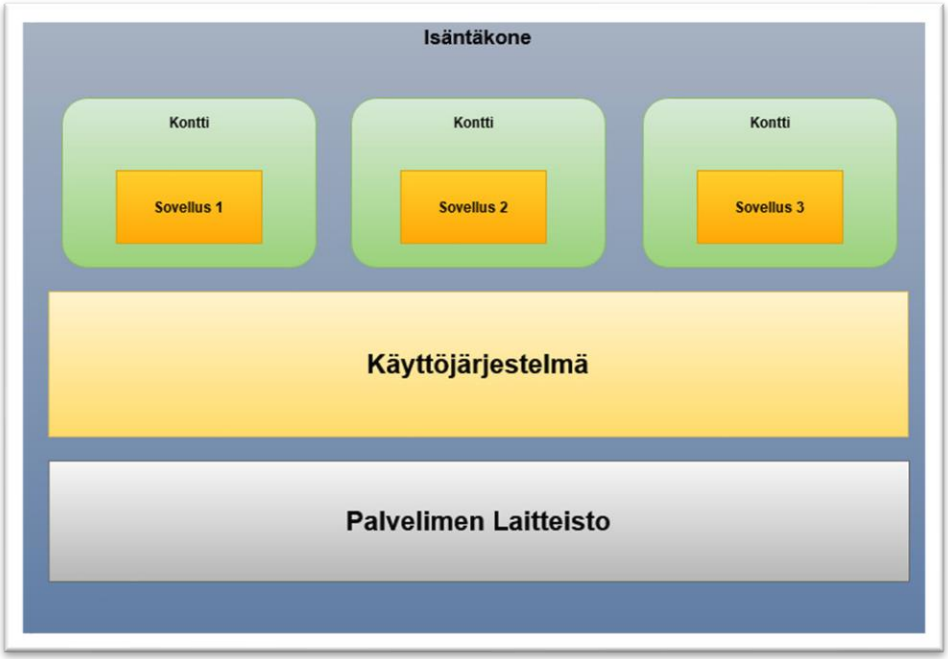
Useat suuret yritykset ovat tuoneet markkinoille omia ratkaisujaan ja siihen liittyviä tuotteita, mutta konttiteknologioiden tunnetuin nimi on kuitenkin edelleen alun perin vuonna 2013 julkaistu Docker. Alun perin Docker-kontit toimivat puhtaasti Linux-ympäristöissä, mutta nykyään myös Microsoftin tuotteista löytyy tuki Docker konteille. Microsoft on sittemmin julkaissut myös oman, Hyper-V virtualisoitiin perustuvan, kontitusratkaisunsa. (Rubens 2017.)

Kontit vastaavat osaltaan ongelmaan, kuinka sovellukset on mahdollista saada toimimaan luotettavasti siirrettäessä niitä erilaisten ympäristöjen välillä. Tässä mielessä kontit IT-maailmassa muistuttavat hyvin pitkälti myös todellisen maailman kuljetuskontteja. Kontteja voidaan käyttää siirrettäessä sovelluksia esimerkiksi ohjelmistokehittäjän kannettavalta testiympäristöön ja testiympäristöstä tuotantoympäristöön. Toisena esimerkkinä voidaan käyttää sovelluksen siirtämistä fyysisestä tietokoneesta konesalissa virtuaalikoneelle pilveen. (Mt.)

Käytännössä kontit siis yhdistävät ajonaikaisesti tarvittavan ympäristön yhdeksi. Sisältä löytyy siis sovellus kaikkine riippuvuuksineen, kirjastoineen ja binaareineen ja konfiguraatiotiedostoineen, joita sovellus tarvitsee toimiakseen. Kontituksella on mahdollista saada häivytettyä alla olevien käyttöjärjestelmien ja infrastruktuurin eroavaisuuksia eli käytännössä samaa konttia on mahdollista ajaa isäntäkoneiden käyttöjärjestelmistä riippumatta. (mt.)

Kontittaminen eroaa virtualisoinnista siinä, että virtuaalikoneissa siirrettävä pakettikokonaisuus, joka voidaan siirtää, on virtuaalikone, joka sisältää kokonaisen käyttöjärjestelmän ja sovelluksen. Fyysinen kone, joilla pyöritetään kolmea virtuaalikonetta,

koostuisi hypervisorista ja kolmesta erillisestä käyttöjärjestelmästä sen päällä. Konttiteknologiaa hyödyntämällä ajettaisiin vain yhtä käyttöjärjestelmää, jonka ydintä (kerneliä) kontit jakaisivat. Tämä on esitetty kuviossa 4 alla.



Kuvio 4. Kolmen sovelluksen ajaminen konteissa yhdellä isäntäkoneella

Näin ollen konttien avulla toteutettuna ympäristö olisi kevyempi ja vaatisi vähemmän resursseja toimiakseen. Konttien koot saattavat olla vain kymmeniä megatavuja, kun taas virtuaalikoneet omine käyttöjärjestelmineen saattavat olla useita gigatavuja. Tämän ansiosta yksittäinen isäntäkone voi samoilla resursseilla pyörittää huomattavasti enemmän kontteja verrattuna virtuaalikoneisiin. (mt.)

Kevyemmän resurssien kulutuksen lisäksi konteissa pyörivät palvelut myös käynnistyvät nopeammin verrattuna erillisen virtuaalikoneen ja sen palveluiden pystyyn nostamiseen. Näin ollen kontteja on mahdollista pystyttää lähes reaaliaikaisesti ja lopettaa, kun tarve kontille päättyy. (mt.)

Kolmas hyöty kontittamisessa on siirrettävyyden lisäksi modulaarisuus. Sen sijaan, että ajettaisiin monimutkaisia sovelluksia yksittäisen kontin sisällä, voidaan sovellus hajottaa useammiksi mikropalveluiksi. Näin esimerkiksi sovelluksen front-endistä vastaava palvelu voidaan jakaa omaan konttiinsa ja vaikkapa tietokanta toiseen. Näin sovelluksen hallintaa ja ylläpitoa saadaan mahdollisesti helpommaksi, koska esimerkiksi yhden osan muuttamiseksi koko palvelua ei tarvitse luoda uudelleen. Näin voidaan ikään kuin

lennossa tehdä muutoksia sovelluksen yksittäisiin osa-alueisiin. Koska kontit ovat lähtökohtaisesti todella kevyitä ja nopeita pystyttää, voidaan kontteja käynnistää ja sammuttaa tarpeen mukaan lähes reaaliaikaisesti. (mt.)

Konttien toiminnallisuuden mahdollistavat ominaisuudet ovat Linux-käyttöjärjestelmästä lähtöisin olevissa ydinteknologioissa (kernel). Sen ansiosta sovellukset on mahdollista eristää toisistaan ohjelmistotasolla. Käytännössä nämä ydinteknologioiden ominaisuudet antavat mahdollisuuden käyttäytyä hieman kuin virtuaalikone, mutta kuitenkin eri tavalla. Kontit ovat prosesseja, jotka käyttävät samaa jaettua Linuxin ydintä (Kernel). Koska ohjelmistot eivät pyöri omilla virtuaalikoneillaan, riittää konttien pyörittämiseen yksi käyttöjärjestelmä. (Evans n.d.)

Kuten aikaisemmin mainittiin, konttien toiminta on mahdollista Linuxin ydintoimintojen avulla. Kyseiset ydintoiminnot ovat nimiavaruudet (Namespaces) ja hallintaryhmät (cgroups). Kernel mahdollistaa prosessien eristämisen luomalla erillisiä nimiavaruuksia konteille. Nimiavaruudet voivat sitten tehdä varauksia Linux-käyttöjärjestelmän tarjolla olevista resursseista. Näin useat kontit voivat käyttää samaa resurssia yhtäaikaisesti aiheuttamatta konflikteja. (Chapter 1. Introduction to Linux Containers. n.d.)

Nimiavaruuksia on useita erityyppisiä:

- **Tiedostojärjestelmä nimiavaruudet (Mount namespaces)** eristää eri prosessiryhmiin kuuluvat prosessit siten, että ne voivat käyttää erilaista näkyvyyttä tiedostojärjestelmän hierarkiaan. Näiden nimiavaruuksien ansiosta jokaisella kontilla voi olla omat /tmp tai /var kansionsa. Samalla voidaan estää konttia näkemästä nimiavaruutensa ulkopuolella olevia muiden konttien tai isäntäkoneen tiedostoja.
- **UTS nimiavaruudet (UTS namespaces)** eristää kaksi järjestelmätunnistetta, jonka ansiosta konteille on mahdollista määrittää omat host- ja domain-nimet.
- **IPC nimiavaruudet IPC (namespaces)** mahdollistaa, että kaksi konttia voivat luoda jaetun muisti segmentin ja semaforia samalla nimellä, mutta eivät pysty vaikuttamaan muiden konttien muisti segmentteihin tai jaettuun muistiin.
- **PID nimiavaruudet (PID namespaces)** mahdollistavat konttien sisäisille prosesseille saman nimiset PID numerot. Konteilla ei ole näkyvyyttä toisten konttien prosessien tietoihin, vaan niiden näkyvyys rajoittuu omiin prosesseihinsa. Isäntäkoneen käyttöjärjestelmä näkee konttien sisällä pyörivät prosessit, mutta täällä määritellään konttien PID nimille omat numeronsa.

- **Verkko nimiavaruudet (network namespaces)** mahdollistavat konttien verkkokontrollien eristämisen, jonka ansiosta jokaiselle kontille voidaan määrittää oma IP-osoitteensa, reititystaulunsa ja jopa omat palomuurit.

Näiden nimiavaruuksien lisäksi on olemassa myös **käyttäjä nimiavaruudet (user namespaces**, joka muistuttaa PID nimiavaruutta. Niiden avulla voidaan määrittää tiettyjä UID:ita toimimaan konteissa. Prosessille voidaan antaa täydet root-oikeudet konttien sisällä ajettaviin tehtäviin ja samaan aikaan pienemmillä käyttöoikeuksilla varustetut oikeudet kontin ulkopuolella tapahtuviin tehtäviin. (Mt.)

Nimiavaruuksien lisäksi konttien eristämiseen ja hallintaan käytetään myös hallintaryhmiä (cgroups). Hallintaryhmillä voidaan rajoittaa, kuinka paljon resursseja kyseisen ryhmän sovelluksilla, on käytettävissään. Hallintaryhmillä voidaan esimerkiksi tarkkailla ja rajoittaa ryhmään kuuluvien sovelluksien muistin, prosessointitehon, verkon ja levytilan käyttöä. Ryhmien avulla on myös mahdollista käynnistää ja lopettaa sovelluksia. Tämän avulla on mahdollista estää esimerkiksi tilanne, jossa jokin prosessi menee häiriötilaan ja kuluttaisi kaiken muistin tai prosessointitehon koneelta. Sen sijaan, että kaikki resurssit annettaisiin kyseisen prosessiin käyttöön, voidaan hallintaryhmien avulla määritellä kyseiselle ryhmälle enimmäisarvoksi vaikka 25 % isäntäkoneen muistinkäytöstä. Näin ongelmallinen prosessi käyttäisikin vain oman ryhmänsä käyttöön luovutetun osion muistista, mutta muut palvelut eivät häiriintyisi vikatilasta lainkaan. (Koutoupis 2018.)

Konttien käyttäminen on siis prosessien hallitsemista hyödyntäen Linuxin ytimestä jo valmiiksi löytyviä työkaluja. Konttien avulla on mahdollista luoda kevyitä ja eristettyjä sovelluksia, mutta tämä vaatisi huomattavaa osaamista Linuxin hallinnasta ja käyttämisestä. Juuri tämän vuoksi alalle on ilmestynyt helppokäyttöisempiä, valmiita kontitusratkaisuja, joista tunnetuin on jo aikaisemminkin mainittu Docker. Docker tarjoaa myös muita hyödyllisiä ominaisuuksia aiemmin mainittujen osien helpottamisen lisäksi. (Evans n.d.)

Käytännössä Docker tuo isäntäkoneen käyttöjärjestelmän päälle yhden lisätason, jonka sisällä Docker-kontit ajetaan. Docker esiteltiin alun perin vuonna 2013 maaliskuussa, jonka jälkeen se sai nopeasti mainetta ja tunnettavuutta tietotekniikanalan ihmisten piirissä. Docker on avoimen lähdekoodin projekti, jonka avulla on mahdollista

rakentaa, jakaa, testata ja ajaa konttisovelluksia. Docker-projekti jaettiin melko pian alun jälkeen useammasta osasta koostuvaksi kokonaisuudeksi. (Yegulalp 2019.)

Docker-kontit ovat kevyitä ja eristettyjä pienoisympäristöjä, jotka sisältävät kaiken tarvittavan sovelluksen ajamiseksi. Yhtenä tärkeimmistä ominaisuuksista Dockerin dokumentaatiossa mainitaankin sen kyky eriyttää konttien sisällöt muista konteista ja isäntäkoneesta siten, että kontti on vuorovaikutuksessa ainoastaan oman yksityisen tiedostojärjestelmänsä kanssa. Tämän tiedostojärjestelmän kontti saa Docker-levykuvasta (Docker image). Docker-levykuva sisältään kaiken, mitä kontti tarvitsee sovelluksen ajamiseen. Joissain tilanteissa kuitenkin isäntäkoneelta on mahdollista kopioida tiedostoja kontin sisälle, mikäli tarve sellaista vaatii. (What is a Container? A standardized unit of software 2020.)

5 Vertailtavat tuotteet

Tutkimuksen toimeksiannossa määriteltiin tiettyjä toiveita ja ehdotuksia koskien tutkittavia hunajapurkkiratkaisuja. Toiveena oli, että valittavia hunajapurkkeja olisi mahdollista käyttää simuloimaan useita eri palveluita. Myös hallittavuuden ja ylläpidon tulisi olla suoraviivaista. Tietoturvallisuus katsottiin yhdeksi tärkeäksi osa-alueeksi, minkä vuoksi mahdollisen hyökkääjän karkaaminen hunajapurkista nähtiin suurempana riskinä, kuin hunajapurkin paljastuminen mahdolliselle hyökkääjälle. Tällä tarkoitetaan lähinnä sitä, että vaikka korkean vuorovaikutustason hunajapurkkituotteet ovat uskottavamman näköisiä hyökkääjälle, on niissä paljastuessaan usein laajempi hyökkäyspinta-ala, jota hyökkääjä saattaa pystyä hyödyntämään. Tuote tulisi myös olla tarvittaessa helposti skaalattavissa laajempaan käyttöön. Näiden lisäksi vertailtavat tuotteet päätettiin rajata koskemaan pääasiallisesti sisäverkon sensoreina toimivia tuotannollisia hunajapurkkeja, eli tutkimukselliset hunajapurkit rajattiin pääsääntöisesti pois tutkimuksesta. Tässä kuitenkin menetetään hyökkääjien käyttämistä menetelmistä kerättävistä tiedoista saatava hyöty. Näiden toiveiden takia tutkimuksessa keskityttiin pääasiassa matalan vuorovaikutustason ja keskitason hunajapurkkeihin, mutta tuotteissa pyrittiin ottamaan huomioon myös mahdollinen hybridimalli tuotannollisen ja tutkimuksellisen hunajapurkin välillä.

Tutkimuksessa oli myös tavoitteena tutkia mahdollisia käyttöönottomenetelmiä, joiden avulla hunajapurkkeja olisi mahdollista lisätä ja poistaa tarpeen vaatiessa nopeastikin. Tämän vuoksi tutkimuksessa perehdyttiin myös jonkin verran kontitukseen ja hunajapurkkien käyttöönoton ja käynnistämisen automatisointiin esimerkiksi Ansiblea hyödyntäen. Useissa hunajapurkeissa kuvataan todellisia palveluita, joita vastaavilla palvelimilla pyörisi muutenkin. Tämän takia hunajapurkit myös kuuntelevat usein pieninumeroisia portteja, jotka vaativat lähtökohtaisesti kohotettuja käyttöoikeuksia. Hunajapurkin ajaminen kohotetuilla käyttöoikeuksilla on oletusarvoisesti riski. Tämän vuoksi hunajapurkkeja kannattaisi ajaa jonkin sopivan porttien uudelleenohjausmenetelmän kautta.

5.1 Cowrie

Cowrie on keskitason vuorovaikutustason hunajapurkki, jota ylläpitää Michel Oosterhof. Cowrie toimii varhaisemman Kippo-projektin manttelinperijänä. Cowrie on Python-ohjelmointikielellä tehty avoimen lähdekoodin hunajapurkki, joka pyrkii emuloimaan hyökkääjälle olevansa todellinen UNIX-palvelin, johon on auki SSH ja telnet yhteydet. Hunajapurkilla on mahdollista uskotella hyökkääjälle, että tämä olisi päässyt kirjautumaan sisään palvelimelle SSH yhteydellä portin 22 kautta. Tällöin oikea SSH:n kautta tuleva hallintayhteys tulee siirtää toimimaan toisesta portista käsin. (What is Cowrie 2018.)

Kun hyökkääjä on päässyt ”kirjautumaan” sisään palvelimelle, aukeaa hänelle näennäisesti oikeaa muistuttava ympäristö, joka perustuu oletuksena Debian 5.0 käyttöjärjestelmälle. Palvelimelle on mahdollista luoda valetiedostoja, jotka saattavat kiinnittää hyökkääjän huomion. Hyökkääjälle annetaan myös mahdollisuus lisätä ja poistaa tiedostoja. (Mt.)

Todellisuudessa kuitenkin kaikki hyökkääjän tekemät toimenpiteet kirjautuvat palvelimen lokitiedostoihin ylläpitäjän myöhemmin analysoitavaksi. Cowrie tarjoaa kattavan dokumentaation ja mahdollisuuden datan keräämiseksi MySQL-tietokantaan, kuten myös työkalut kerätyn datan visualisoimiseksi helpommin käsitettävään muotoon. Oletuksena Cowrie tallentaa lokeihinsa kaikki hyökkäykset JSON-formaatissa, jota

useat tarjolla olevat työkalut tukevat. Cowriesta löytyy myös tuki ELK stackin käyttämiseksi. (mt.)

Cowrielle löytyy valmiiksi tuettu versio Docker-kontissa ajettavaksi ja sillä on laaja käyttäjäkunta. Tämän johdosta mahdolliset suuremmat ongelmat, joita sovelluksissa usein alussa on, on todennäköisesti saatu jo korjattua. Cowrieta päivitetään ja sen laaja ja paneutunut yhteisö tarjoaa nopeat vasteajat kysymyksiin ja havaittuihin ongelmiin. Se on laajalti käytössä sekä tutkimus-, että tuotannollisena hunajapurkkina. Cowrie on myös parantanut edeltäjäänsä Kippoa vaivanneita, hunajapurkkiin joutumisen hyökkääjälle paljastavia, tunnistetietoja koskevia ongelmia. (mt.)

5.2 OpenCanary

OpenCanary on luotu paljastamaan mahdollisten hyökkääjien lateraalista liikehdintää organisaatioiden sisäverkoissa. OpenCanary on Thinkstin tuottama avoimen lähdekoodin hunajapurkkituote. Samalta tekijältä löytyy myös maksullinen hunajapurkki toteutus nimeltä Canary, joka tarjoaa muutamia lisäominaisuuksia avoimen lähdekoodin versioon verrattuna, kuten helpomman käyttöönoton, paremman hallittavuuden ja paremman datan visualisoinnin helppokäyttöisestä web-pohjaisesta käyttöliittymästä. (Wahl 2019.)

OpenCanaryn ideana on virittää tuotantoverkkoon ulkoisesti aidon näköisiä palveluita, joilla ei kuitenkaan normaalitilanteessa tulisi olla mitään liikennettä. Mikäli hunajapurkki havaitsee, että joku tutkii tai yrittää käyttää jotain sen palveluista, tekee hunajapurkki hälytyksen halutulla tavalla. (Configuration 2018.)

OpenCanaryn yhtenä vahvuutena on kattava valikoima erilaisia palveluita, joita tuote osaa emuloida ja nostaa niiden väärinkäytöstä hälytyksen:

- **SSH:** Nostaa hälytyksen, mikäli joku yrittää kirjautua sisään palvelimelle SSH-yhteydellä.
- **FTP:** Tiedostojen siirtoon käytettävä protokolla. Palvelin nostaa hälytyksen kirjautumisyrityksistä.
- **Git:** Git-protokolla, joka hälyttää repositorion kloonauksesta.
- **HTTP:** HTTP-palvelin, joka hälyttää kirjautumisyrityksistä.

- **HTTPproxy:** http web proxy, joka hälyttää yritettäessä ohjata liikennettä toiselle sivustolle.
- **MSSQL:** MSSQL palvelin, joka hälyttää kirjautumisyrityksistä.
- **MySQL:** MySQL palvelin, joka hälyttää kirjautumisyrityksistä.
- **telnet:** Telnet palvelin, joka hälyttää kirjautumisyrityksistä.
- **SNMP:** SNMP palvelin, joka hälyttää OID (Object IDentifier) pyynnöistä.
- **SIP:** SIP-palvelin, joka hälyttää SIP-pyynnöistä.
- **VNC:** VNC-palvelin, joka hälyttää kirjautumisyrityksistä.
- **Redis:** Redis-palvelin, joka hälyttää toimenpiteistä.
- **TFTP:** TFTP-palvelin, joka hälyttää kaikista pyynnöistä.
- **NTP:** NTP-palvelin, joka hälyttää NTP pyynnöistä.
- **TCPbanner:** TCPbanner-palvelu, joka hälyttää yhteyksistä ja seuraavista dataa vastaanottavista tapahtumista.
- **SMB:** Palvelu poikkeaa hieman yllä olevista, sillä se vaatii todellisen Windowsin tiedoston jakopalvelun. OpenCanary voidaan säätää seuraamaan tiettyä tiedostojakoa ja nostamaan hälytyksen, mikäli joku pyrkii avaamaan tai muokkaamaan sitä.
- **PortScan:** Iptablesin kanssa yhdessä toimiva lisäosa, joka hälyttää, kun OpenCanaryyn kohdistuu porttiskannauksia. Tällä hetkellä palvelu tukee tunnistusta nmap OS, nmap FIN, nmap NULL ja normaaleille porttiskannauksille.

OpenCanary tukee useita eri tapoja hälytyksen toimittamiseksi valvovalle taholle, mutta dokumentaatiossa pääpaino on keskittynyt sähköpostiin tuleviin hälytyksiin. Dokumentaatiosta löytyy myös valmiita uskottavaksi palvelimeksi luotuja pohjia sekä Windows, että Linux puolelle. OpenCanary on myös mahdollista ajaa kontista käsin, mutta kyseessä on kolmannen osapuolen tekemä toteutus, eikä Thinkst ole ainakaan virallisesti mukana siinä.

5.3 T-Pot

T-Pot ei varsinaisesti ole yksittäinen hunajapurkkituote, vaan se yhdistää lukuisia tunnettuja hunajapurkkisovelluksia yhdeksi tuotteeksi (All-in-One honeypot). T-Pot on Deutsche Telekomin hunajapurkkiprojekti, joka on käynnistetty jo vuonna 2010. (Introduction to Deutsche Telekom's Honeypot Project 2015.) Deutsche Telekom on saksalainen telekommunikaatioalan yritys ja suurin toimialallaan Saksassa. (Company Profile 2020.)

T-Pot projektin kehitys alkoi jo vuonna 2010 yhdistämällä muutamia hyväksi havaittuja hunajapurkkeja toimimaan yhdessä. Tuotteen kehitys jatkuu edelleen ja vuonna 2020 tarjolla oleva tuote on kattava valikoima erilaisia hunajapurkkituotteita ja niiden käyttöä tukevia lisätyökaluja. Tuote on käytössä Telekomin asiakkailla ympäri maailmaa, toimien sensoriverkostona, joka havainnoi hyökkäyksiä. T-Pot sisältää myös toiminnallisuuden, jonka avulla halukkaat voivat jakaa tietoa havaituista hyökkäyksistä yhteisölle. Kevyimmillään tämä voidaan nähdä sicherheitstacho-työkalusta, jonka avulla hyökkäyksien määrää ympäri maailmaa voi seurata visuaalisesta käyttöliittymästä. Sivustolla näkyvät tiedot ovat kuitenkin melko rajoittuneita ja toimivat pääasiassa visuaalisesti toteutettuina tilastoina. (tpotce 2020.)

Tutkimukselliset hunajapurkit ovat usein raskaita pystyttää ja niiden ylläpito vaatii huomattavan suurta työpanosta. T-Pot pyrkii yhdistämään useita erityyppisiä hunajapurkkeja yhdeksi helposti pystytettäväksi ja hallittavaksi kokonaisuudeksi. Osittain tämän vuoksi T-Pot on rakennettu hyödyntämään konttiteknologiaa. Koko järjestelmä on rakennettu käyttämään volatiileja kontteja, jotka voidaan ongelmatilanteessa poistaa ja luoda uudelleen. Koko järjestelmä perustuu avoimuuteen ja läpinäkyvyyteen, eli halutessaan varmistua tuotteen tietoturvasta, voi tuotteen kasata myös alusta loppuun itsenäisesti. Tällä hetkellä tuotteesta löytyy seuraavien hunajapurkkien kontitetut versiot (Mt.):

- **ADBHoney** on matalan vuorovaikutustason hunajapurkki, joka käyttää Android Debug Bridgeä (ADB) TCP-yhteyden yli. ADB protokollan tarkoituksena on pitää kirjaa kyseiseen isäntäkoneeseen yhteydessä olevista oikeista ja virtuaalisista puhelimista, televisioista ja mediantallennuslaitteista. Kyseistä palvelua käytetään yleensä kehittäjien toimesta vianselvityksessä ja sisällön toimittamisessa laitteille. Jos yhteys on kuitenkin jo muodostettu, ADB:stä löytyy toiminto TCP/IP yhteyden ottamiseksi laitteelle. Toisin kuin USB-yhteyttä käytettäessä, ei yhteydessä ole samanlaista suojausta ja hyökkääjä voi ajaa esimerkiksi shell-komentoja kohteessa. ADBHoney hunajapurkki toimii paljastaakseen tämän kaltaisia hyökkäyksiä. (ADBHoney 2019.)
- **Cisco ASA honeypot** on matalan vuorovaikutustason hunajapurkki, joka pyrkii kuvastamaan Ciscon ASA laitteessa olevaa haavoittuvuutta CVE-2018-0101 ja keräämään tietoa siihen liittyvistä palvelunestohyökkäyksistä ja haitallisen koodin suorittamisesta. (Cisco ASA Honeypot 2018.)

- **CitrixHoneypot** on Citrixin tuotteisiin liittyvään haavoittuvuuteen CVE-2019-19781 liittyvä hunajapurkki, jonka avulla pyritään saamaan tietoa skannauksista ja haavoittuvuuden hyödyntämisyrityksistä. Kyseinen haavoittuvuus liittyy Citrix Application Delivery Controlleriin ja Citrix Gatewayhin. (Honeypot for CVE-2019-19781 (Citrix ADC) 2020.)
- **Conpot** on matalan vuorovaikutustason hunajapurkki, joka pyrkii kuvastamaan teollisuudessa laajalti käytössä olevia hallintajärjestelmiä, eli kyseessä on ICS/SCADA hunajapurkki. Conpotin avulla on mahdollista rakentaa järjestelmä, joka näyttää ulkoisesti tarvittaessa monimutkaiseltakin teollisuuden hallintajärjestelmältä. Hunajapurkki on toteutettu niin, että sen avulla voidaan kuvata esimerkiksi kovan käyttökuormituksen alla olevan järjestelmän viiveitä. Conpotia kehitetään Honeynet Projektin alaisuudessa ja se on edelleen aktiivisessa kehityksessä. (Welcome to Conpot's documentation! 2018.)
- **Cowrie** joka käydään tarkemmin läpi aikaisemmin omassa kappaleessaan
- **Dicompot** on Go:lla kirjoitettu DICOM standardia käyttävää palvelua kuvastava hunajapurkki. DICOM tulee sanoista Digital Imaging and Communications in Medicine, joka on lääketieteellisten kuvien tiedonsiirrossa käytetty standardi. Dicompot käyttäytyy kuin oikea DICOM palvelin, mutta todellisuudessa kerää tietoa hyökkääjästä. (Dicompot - A Digital Imaging and Communications in Medicine (DICOM) Honeypot 2020.)
- **Dionaea** on pythonilla kirjoitettu hunajapurkki, jonka tarkoituksena on kerätä tietoa haittaohjelmien käyttämistä haavoittuvuuksista. Dionaea tarjoaa hunajapurkkipalvelua lukuisille eri protokolille kuten: blackhole, epmap, ftp, http, memcache, mirror, mqtt, mssql, mysql, pptp, sip, smbt, tftp, upnp. Dionaean pohjimmainen tarkoitus on saada kaapattua talteen kopio hyökkääjän käyttämästä haittaohjelmasta. Dionaea on edelleen aktiivisessa kehityksessä ja sillä vaikuttaa olevan vahva kehittäjäyhteisön tuki. (Welcome to dionaea's documentation! 2020.)
- **Elasticpot** kuvastaa Elastisearch palvelinta, joka on avattu vahingossa ulkoverkkoon. Hunajapurkki vastaa saamiinsa kyselyihin riippuen siitä, kuinka palvelin on konfiguroitu. Käytettäessä oletusvastauksia hyökkääjän on mahdollista tunnistaa olevansa hunajapurkin sisällä, joten kehittäjä on julkaissut vahvan suosituksen vastauksien muokkaamisesta. T-Potin omasta dokumentaatiosta ei ole suoraan nähtävissä millä asetuksilla kyseistä kontissa ajettavaa palvelua ajetaan. Tämän vuoksi Elasticpot konttia käytettäessä asia kannattaa käydä tarkistamassa kyseisen kontin omista asetuksista. (ElasticPot 2020.)

- **Glutton** on Go:lla kirjoitettu kaiken liikenteen vastaanottava hunajapurkki, joka toimii usein välittäjänä hyökkääjän ja muiden hunajapurkkien välillä. Se mahdollistaa myös liikenteen kaappaamisen, lokittamisen ja analysoinnin. Glutton kuuntelee kaikkia portteja ja toimii sitten oman sääntötaulunsa (rules.yaml) pohjalta. Gluttonin päätehtävä T-Potissa on ohjata liikennettä tilanteeseen sopiville muille hunajapurkeille. (Sheikh 2018.)
- **Heralding** on yksinkertainen matalan vuorovaikutustason hunajapurkki, jonka ainut tehtävä on kerätä kirjautumistietoja kirjautumisyrityksistä. Tällä hetkellä Heralding tukee seuraavia protokollia: ftp, telnet, ssh, rdp, http, https, pop3, pop3s, imap, imaps, smtp, vnc, postgresql and socks5. (Heralding 2020.)
- **Honeypy** Matalan tai keskitason vuorovaikutustason hunajapurkki, riippuen käytettävistä lisäosista. Honeypy on kirjoitettu Python 2:lla. Tuotetta ei enää kehitetä aktiivisesti tekijän toimesta ja Python 2:n tuki on jo päättynyt. Tämä tulee huomioida työkalua käyttöönotettaessa. Honeypystä löytyy tuki muutamille erilaisille lisäosille, joiden valinnan perusteella määrittyy, onko kyseessä matalan- vai keskivuorovaikutustason hunajapurkki. Tuotteesta löytyviä lisäosia ovat: DNS (vastaa DNS pyyntöihin satunnaisilla IP-osoitteilla), Echo (toistaa yhdistettyjen asiakaskoneiden lähettämän datan), Esticsearch (emuloi Elasticsearchia), HashCountRandom (Hämäyspalvelu, joka lisää laskurin lukuarvoa jokaisella yhteydellä ja palauttaa md5 tiivistesumman laskurin arvosta ja satunnaista dataa), MOTD (palauttaa TCP ja UDP yhteysyrityksissä viestin ja katkaisee yhteyden), NTP (hyväksyy NTP pyynnöt ja palauttaa järjestelmän ajan), Random (palauttaa jokaiselle saadulle data instanssille vastauksena satunnaista dataa), SIP (Istunnon aloitusprotokollan emulointi), SMTP (Yksinkertainen SMTP palvelinta kuvaava palvelu), TFTP (TFTP tiedonsiirron emulaatio), Telnet (kuvaa telnet palvelua), Web (Yksinkertainen web-palvelimen emulaatio, jolta löytyy esimerkiksi /robots.txt ja wp-login.php, kuten myös muutamia muita erilaisia admin sivustoja. Palauttaa muihin pyyntöihin ”200 OK” vastauksen). Kyseessä on melko monipuolinen, mutta osittain vanhentunut tuote. (HoneyPy Plugins 2020.)
- **Honeysap** on matalan vuorovaikutustason avoimen lähdekoodin hunajapurkki, jonka tehtävänä on kuvastaa tiettyjä SAP-palveluita. Sen tavoitteena on saada kerättyä tietoa erilaisista käytetyistä tekniikoista ja motivaatioista SAP-järjestelmien kimppuun hyökkääjien osalta. Tuotteen kehittäjä ei suosittele tuotteen käyttöä suoraan tuotantoympäristöissä ainakaan ilman, että sen osalta käydään läpi tarkasti järjestelmäkehi-

tyksen elinkaarimallin mukainen kartoitus. Tämä kannattaa ottaa huomioon valittaessa T-Potin asennuksen yhteydessä käyttöön otettavia tuotteita. (HoneySAP: SAP Low-interaction honeypot 2020.)

- **Honeytrap** on matalan vuorovaikutustason hunajapurkki, jonka tavoitteena on kerätä tietoa TCP ja UDP portteihin kohdistuvista hyökkäyksistä. Honeytrap vastaa pyyntöihin ja saa joissain tilanteissa asiakaskoneen tai hyökkääjän luulemaan kyseessä olevan todellinen palvelu. Koska kyseessä on kuitenkin matalan vuorovaikutustason hunajapurkki, paljastuu palvelu kuitenkin huijaukseksi nopeasti. T-Pot hyödyntää Honeytrappia ja aikaisemmin mainittua Glutton hunajapurkkia vastaamalla niiden avulla portteihin, joiden takana ei ole jo muita palveluita tarjoavia hunajapurkkeja. (Honeytrap 2007.)
- **IPP Honey** eli Internet Printing Protocol Honeypot on hunajapurkki, joka kuvastaa vahingossa julkisessa verkossa kiinni olevaa tulostinta. (IPP Honey 2020.)
- **Mailoney** on SMTP hunajapurkki, joka on kirjoitettu Pythonilla. Koostuu kolmesta moduulista, joista kaksi kuvastavat avointa linkkipalvelinta (open relay). Eroina on pääasiassa kerättyjen lokitietojen määrä, toisen kerätessä ainoastaan viestien sisällöt, toisen kerätessä talteen kaiken saamansa tiedon. Kolmas moduuli kerää kirjautumistunnuksia ja salasanoja kirjautumisyrityksien yhteydessä. (Mailoney 2018.)
- **Medpot** on Go:lla kirjoitettu HL 7:n FHIR (Fast Healthcare Interoperability Resources standardia esittävä hunajapurkki, joka tallentaa yhteysyritykset. (Medpot 2018.)
- **RDPY** on Pythonilla kirjoitettu Microsoftin etätyöpöytä protokollaa (RDP, Remote Desktop Protocol) kuvaava hunajapurkki. RDPY tukee RDP:n turvatasoa (security layer), etätyöpöytä yhteyttä SSL ja NLA (ntlmv2 tunnistautumisprotokollaa käyttäen) tunnistautumisia käyttäen. RDPY koostuu man-in-the-middle hyökkäystä käyttävästä välityspalvelimesta, jonka kautta ylläpitäjä voi tallentaa istunnon aikana tapahtumia, varsinaisesta RDP hunajapurkista, RDP ruutukaappaajasta (RDP screenshoter, joka tallentaa kirjautumisikkunan tiedostoksi), RDP clientistä, VNC clientistä, VNC ruutukaappaajasta (VNC screenshoter, joka tallentaa ensimmäiset ruutupäivityksen tiedostoon), ja RSS (record session) toistimesta. (RDPY 2020.)
- **Snare / Tanner** on kahdesta osiosta koostuva yhdistetty työkalu. Snare on web-sovellusta matkiva hunajapurkki. Sen tehtävänä on toimia sensorina ja houkutella haitallisia toimijoita hyökkäämään itseään vastaan verkon yli. Se kykenee muokkaamaan olemassa olevia sivustoja hyökkäyspinnoiksi. Tanner toimii aivoina Snaren takana. Jokainen tapahtuma, jonka Snare saa napattua lähetetään Tannerille arvioitavaksi, jonka jälkeen Tanner tekee päätöksen siitä, kuinka Snaren tulee vastata hyökkääjälle. Tämän

> ansiosta sensoreina toimivat Snare-hunajapurkit voivat muokata käytösmallejaan lähes reaaliaikaisesti. Tanneriin on ohjelmoitu lukuisa määrä erilaisia olemassa olevia haavoittuvuuksia, jonka vuoksi Snare pystyy vastaamaan hyökkääjältä tulleisiin pyyntöihin sopivan haavoittuvalla vastauksella. (MushMush Foundation n.d.)

Kuten listauksesta käy ilmi, löytyy T-Potista todella kattava määrä erilaisia hunajapurkkeja. Joukossa on kuitenkin muutamia hieman toimiala spesifejä hunajapurkkeja, joiden käyttö olisi todennäköisempää esimerkiksi sosiaali- ja terveyspuolen tietojärjestelmissä. Koska T-Pot hyödyntää erityyppisiä hunajapurkkituotteita todella kattavasti, löytyy heiltä myös räätälöityjä asennuksia eri toimialoille. Ohjeistuksesta löytyy listaukset käyttöönotettavista tuotteista esimerkiksi teollisuuden, terveysalan ja tutkimuskäyttöön tuleville asennuksille. Näin ollen jokaiseen ympäristöön ei ole tarvetta asentaa kaikkia tuotteita. (tpotce 2020.)

T-Pot on suunniteltu pyörimään Debian pohjaisella käyttöjärjestelmällä, jonka päälle asennetaan Docker-konttien ajamiseen vaadittavat työkalut. T-Pot hakee Docker levykuvat (images) Telekom-Securityn omista GitHub repositorioista ja käynnistää tämän jälkeen konteissa pyörivät palvelut.

Järjestelmän hallinnointi tapahtuu pääasiassa web-pohjaisesta käyttöliittymästä, mutta palvelimelle on mahdollista sallia myös SSH-yhteyden kautta tapahtuva hallinnointi. Järjestelmä tukee myös kaksivaiheista tunnistautumista, jonka avulla tietoturvaa on mahdollista parantaa kirjautumisten osalta. Järjestelmästä löytyy myös fail2banilla toteutettu bruteforce-hyökkäyksien esto kirjautumisen osalta. Palvelut itsessään ovat eriytettyinä omissa konteissaan, ja niitä on mahdollista tuhota ja rakentaa uudelleen tarpeen vaatiessa. Havaittaessa jossain kontissa tapahtuvan jotain ongelmallista, voidaan kyseinen kontti vain tuhota ja luoda uudelleen automaattisesti. Konttien tuottamat lokitiedot haetaan konteista erikseen, eli vaikka kontti jouduttaisiin poistamaan ja luomaan uudelleen, säilyisivät lokitiedot edelleen ympäristössä. Järjestelmän tarkempi arkkitehtuurikuvaus on nähtävissä liitteessä 1. (mt.)

Tuotteesta löytyy myös sisäänrakennettuna konteissa ajettavina sovelluksina ELK stackin työkalut. Koska palvelut T-Potin sisällä on hajautettuna jokainen omiin konteihinsa, tarvitsee tuote myös keskitetyn lokitietojen hallinnan ja visualisointityökalun. Tässä käyttöön on valittu aiemmin mainittu ELK Stack. (tpotce 2020.)

ELK koostuu kolmesta täysin avoimella lähdekoodilla toteutusta erillisestä sovelluksesta, joita kehittää, ylläpitää ja hallitsee yritys nimeltä Elastic. Nimen E-kirjain tulee sanasta ElasticSearch. ElasticSearch on ELK stackin ehkä tärkein osa, sillä sitä käytetään lokitietojen tallentamiseen. L-kirjain tulee sanasta LogStash, joka on palvelinpuolen datan käsittelyyn erikoistunut sovellus, joka kerää dataa useista lähteistä ja muokkaa sen sopivaan muotoon käyttäjän toiveiden mukaisesti. K-kirjain tulee sanasta Kibana, joka toimii ELK stackin käyttöliittymänä ja tarjoaa mahdollisuuden kerätyn tiedon visualisoimiseen ja tutkimiseen. (Senanayaka 2018.) Sisäänrakennettu ELK stack mahdollistaa T-Potin keräämien tietojen helpon hyödyntämisen esimerkiksi organisaation poikkeamanhallintaryhmän työskentelyn tukena. T-Potin asennuksen mukana tulee myös kontissa ajettavana Elasticsearch Head, joka on web-käyttöliittymä Elasticsearch klusterin selaamiseen ja hallinnointiin.

Yllä mainittujen työkalujen lisäksi järjestelmässä ajetaan konttien sisällä myös muita työkaluja. Näihin kuuluu Cockpit, joka on kevyt web-käyttöliittymä konteille ja käyttöjärjestelmälle ja joka toimii reaaliaikaisena resurssien käytön seurantana. Loput konteissa ajettavat työkalut keskittyvät pääasiassa kerätyn datan analysointiin. Cyberchef on websovellus, jonka avulla voidaan salata ja avata salauksia ja erilaisia pakkauksia. Työkalua voidaan käyttää kerätyn datan analysoimiseen ja purkamiseen selkokieliseen muotoon. Fatt työkalun avulla voidaan kerätä verkkoliikenteestä metadataa ja sormenjälkiä joko pcap-tiedostoista tai reaaliaikaisesti suoraan verkosta. SpiderFoot on avoimia lähteitä hyödyntävän tiedustelun automatisointi työkalu, jolla voidaan hankkia tietoa esimerkiksi IP-osoitteesta, domainista tai sähköpostiosoitteesta. T-Potin osalta työkalulla on mahdollista tarkistaa, onko hyökkääjän IP-osoite listattuna haitalliseksi osoitteeksi jo aikaisemmin. Lisäksi palvelimella pyörii myös Suricata, joka on verkkouhkien havainnointiin ja estämiseen tehty työkalu. Kontissa ajettava Suricata-ohjelma on reaaliaikainen IDS-, IPS- ja verkkomonitorointityökalu, jonka avulla voidaan prosessoida PCAP-tiedostoja myös offline tilassa.

6 Tuotteen valitseminen

Testiasennukseen asti päätyväksi tuotteeksi valikoitui lopulta Telekomin tuotteistama T-Pot. Tuotannolliseksi sensori hunajapurkiksi soveltuisi hyvin myös kevyempi OpenCanary. OpenCanaryn ominaisuudet soveltuisivat mahdollisesti sisäverkkoon sijoitettaviin hyökkäyksen paljastavaksi kevyemmiksi hunajapurkeiksi. T-Pot valittiin kuitenkin testiasennukseksi sen monipuolisen hunajapurkkituotteiden tarjonnan, tuotteen vahvan tuen ja jatkuvan kehitystyön vuoksi. OpenCanaryn suhteen huolta herätti erityisesti sen jatkuvuuden takaaminen, sillä Thinkst tarjoaa myös vastaavaa kaupallista kilpailevaa tuotettaan vapailla markkinoilla. Cowrie itsenäisenä hunajapurkkina päätettiin jättää kokonaan pois, sillä se löytyy myös osana T-Potin hunajapurkkivalikoimaa. T-Pot on lähtökohtaisesti avoimeen lähdekoodiin perustuva, eikä sillä ole yrityksen sisäistä kilpailijaa, joka mahdollisesti veisi resursseja kehitys- ja ylläpitotyöstä jatkossa. Tällainen tilanne oli OpenCanaryn osalta huolena. T-Pot oli myös tuotteistettu huomattavasti pidemmälle verrattuna muihin vastaaviin avoimen lähdekoodin järjestelmiin. T-Potin valintaa tuki myös järjestelmän ylläpidon helppous ja hallinta- ja valvonta järjestelmien suoraviivaisuus. Hallinta- ja valvontajärjestelmien käyttäminen on mahdollista toteuttaa suoraan tuotteesta löytyvän web-käyttöliittymän kautta.

Monipuolisesta hunajapurkkituotteiden tarjonnasta johtuen, tuote vaatii alustaympäristöltään jonkin verran enemmän resursseja kuin kilpailevat hunajapurkit. Tätä kuitenkin helpottaa ominaisuuksien jako omiin eriytettyihin kontteihinsa, joista loppukäyttäjän on mahdollista valita tiettyjä ominaisuuksia jätettäväksi pois tai lisättäväksi myöhemmin. Turvallisuuden näkökulmasta lisäsuojaa saadaan konttien eristämisestä omiksi yksiköikseen, mutta myös siitä, että kontit on mahdollista tuhota ja pystyttää tarvittaessa nopeasti uudelleen. T-Pot tarjoaa tuen myös ympäristön pystyttämiseksi Ansiblen avulla ja heidän Githubistaan löytyy valmiit pelikirjat asennuksia varten, joita on mahdollista muokata omiin tarpeisiin sopiviksi. Tämän ansiosta T-Pot on mahdollista ottaa nopeasti käyttöön ja pystyttää palvelimia ajamaan palvelua tarpeen vaatiessa eri puolille organisaation sisäverkkoa Ansiblea käyttämällä. Tällöin tulee kuitenkin huomioida käyttöönotettavien komponenttien osalta niiden tarpeellisuutta, elin-

kaarta ja esimerkiksi sitä, paljastaako tietyt tuotteet palvelun hunajapurkiksi jo ennakkoon. Viimeisin kohta ei toki ole niin oleellinen, kun kyseessä on tuotannollinen hunajapurkki, joka toimii pääasiallisesti sensorina.

T-Pot on pyrkinyt lähtökohtaisesti avoimuuteen ja läpinäkyvyyteen tuotteensa osalta. Tämän ansiosta käyttäjän on mahdollista asentaa tuote alusta asti myös itse, jolloin käyttäjä voi arvioida mahdollisia vaikutuksia tietoturvaan omatoimisesti. Tuotteesta on tarjolla myös valmis ISO-tiedosto, jonka avulla tuote on mahdollista asentaa omaan virtuaalikoneeseensa helposti ja nopeasti, esimerkiksi testausta varten.

7 Testiasennus

Testiasennus päätettiin suorittaa virtuaalikoneella, VirtualBox sovellusta käyttäen. VirtualBox on avoimen lähdekoodin virtualisoimiseen tarkoitettu ohjelma, joka tukee muun muassa Linux-, macOS ja Windows-käyttöjärjestelmiä. T-Pot on rakennettu toimimaan Debianin viimeisimmän vakaan version (Asennusohjeissa suositeltiin Debian 10) päällä, joten VirtualBox soveltui tehtäväänsä hyvin.

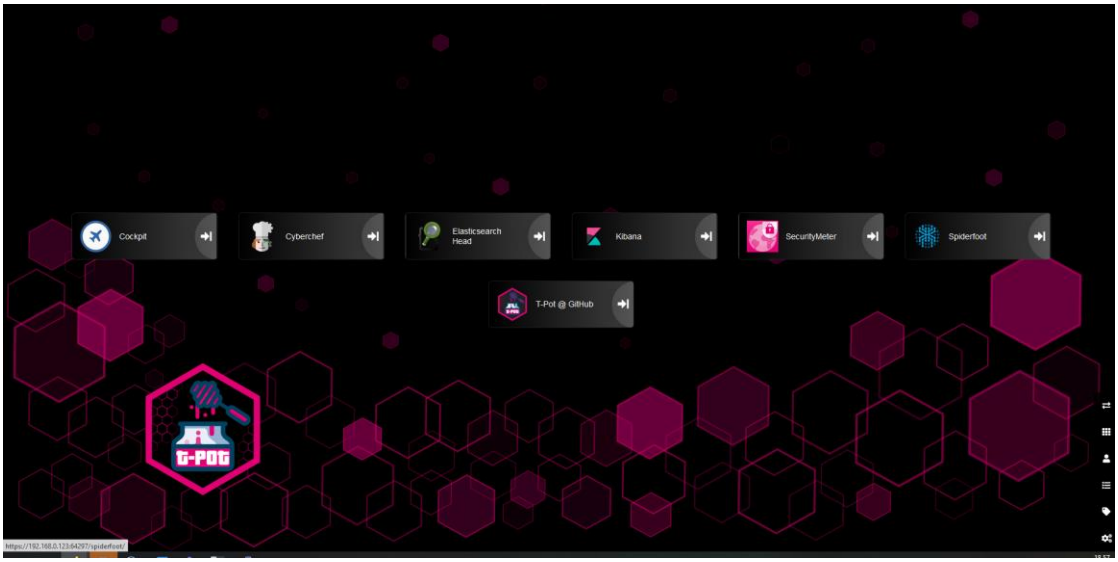
Asennusta varten VirtualBoxiin luotiin uusi kone vaadituilla resursseilla ja aloitettiin asennus valmiista ISO-tiedostosta. Verkkokortti tuli asettaa siltaavaan tilaan, sillä asennuksen aikana jouduttiin lataamaan jonkin verran tiedostoja verkon yli.

T-Potin asennusohjeissa käydään läpi myös järjestelmän sijoituspaikkaa verkossa. Mikäli järjestelmästä haluttaisiin saada mahdollisimman paljon tietoa hyökkääjien toimintatavoista ja hyökkäyksistä, tulisi palvelin sijoittaa suoraan kiinni julkiseen verkkoon. Tässä testiä varten asennetussa palvelimessa, T-Pot sijoitettiin kuitenkin sisäverkkoon, jolloin voitiin välttyä ulkopuolisten vaikuttajien häiritseviltä toimilta ja saatiin tutkittua tuotteen ominaisuuksia ilman riskiä ulkopuolisten aiheuttamista häiriöistä.

Asennus itsessään oli sangen yksinkertainen, eikä eronnut juurikaan Debianin asennuksesta. Asennuksen jälkeen järjestelmään luotiin käyttäjätunnus ja webkäyttäjätunnus ja valittiin minkä tyyppinen asennus palvelimelle haluttiin luoda. Tässä asennuksessa käytettiin standard-tyyppistä asennusta, joissa asentui kontit hunajapurkeille: adbhoney, ciscoasa, citrixhunajapurkki, conpot, cowrie, dicompot, dionaea, elasticpot,

heralding, honeysap, honeytrap, mailoney, medpot, rdpy, snare ja tanner. Näiden lisäksi asennettiin kontit seuraaville työkaluille: cockpit, cyberchef, ELK, fatt, elasticsearch head, ewsposter, nginx / heimdall, spiderfoot, p0f ja suricata. Kontit asentuivat ja käynnistyivät automaattisesti palvelimen uudelleenkäynnistymisen jälkeen.

Tämän jälkeen palvelimelle on mahdollista kirjautua sisään joko SSH:n yli tai käyttämällä web-käyttöliittymää, jonka etusivu näkyy kuviossa 5. Web-käyttöliittymä vastaa oletuksena portista 64297 ja SSH yhteys 64294. Web-käyttöliittymästä löytyy pääsy myös terminaaliin Cockpitin kautta (Cockpit vastaa oletuksena portista 64294).



Kuvio 5. Web-käyttöliittymän kirjautumisikkuna

Kun palvelin käynnistetään, käynnistyy ensin itse virtuaalikone, tämän jälkeen käynnistyvät tarvittavat palvelut, kuten Docker ja viimeisenä kaikki ajettavat kontit. Haluttaessa muokata asennuksen tyyppiä tai koettaessa tarvetta lisätä jokin toisen asennustyypin mukana tulevista työkaluista, voi käyttäjä muokata /opt/tpot/etc/compose alla olevia .yml tiedostoja tarpeidensa mukaisiksi ja ajaa sitten käytössä olevien tuotteiden päivittävän skriptin.

Haluttaessa varmistua Docker-levykuvien oikeellisuudesta vielä laskentasummien tarkistamisen lisäksi, voitaisiin kyseiset levykuvat ladata omaan ympäristöön ja perustaa oma repositorio täysin T-Potin käyttöön. Tämä kuitenkin lisää käyttöönotossa ja ylläpidossa vaadittavaa työmäärää ja asettaa omat haasteensa levykuvien ajan tasalla pitämiseksi. T-Potin monipuolinen palvelutarjonta tarkoittaa myös laajempaa ylläpitoa omaa repositoriota käytettäessä.

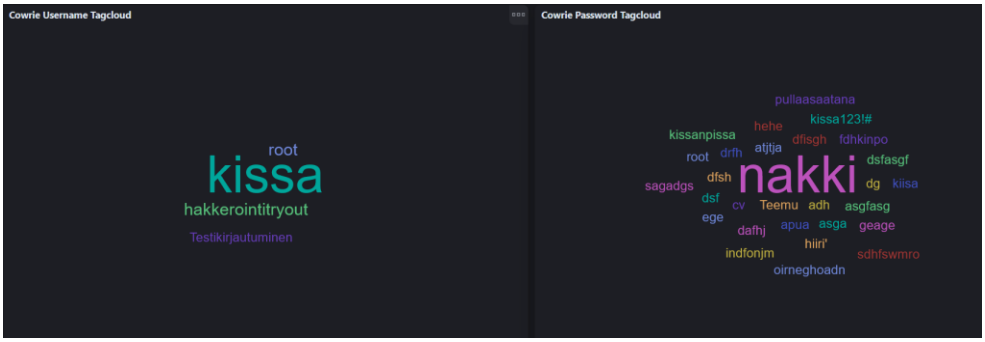
Käyttöönotettaessa käyttäjän kannattaa harkita käyttötapauksen mukaan myös lokien säilytysajan muokkaamista. Oletuksena lokitiedot ja konttien keräämät tiedot säilyvät 30 päivää, mutta asetusta on mahdollista muuttaa logrotate.conf tiedostossa.

T-Potin yhtenä ongelmana havaittiin sen tarve olla yhteydessä internettiin. T-Pot pyrkii hakemaan päivityksiä, erilaisia haavatiedotteita ja IP-osoitteiden mainetietoja julkisista lähteistä. Asennuksen yhteydessä palvelin tarvitsee lisäksi ICMP/TRACEROUTE avaukset hakeakseen tiedon lähimmistä ja nopeimmista tiedostojen latauspaikoista. Lisäksi asetuksista voi halutessaan estää tietojen jakaminen T-Potin yhteisön kanssa. Ohjeistuksessa suositellaan, ettei tätä estettäisi, mutta tämä on kuitenkin tehty mahdolliseksi. Estäminen tapahtui poistamalla Ewsposter palvelun käynnistämisen asetukset /opt/tpot/etc/tpot.yml tiedostosta kyseistä palvelua koskevat rivit.

Päivitykset on kuitenkin toteutettu siten, että kaikki konfiguraatiomuutokset, joita käyttäjä itse on tehnyt, tullaan yliajamaan päivityksien yhteydessä, eli kaikista itsetehdyistä muutoksista on hyvä ottaa varmuuskopiot ennen päivityksien tekemistä.

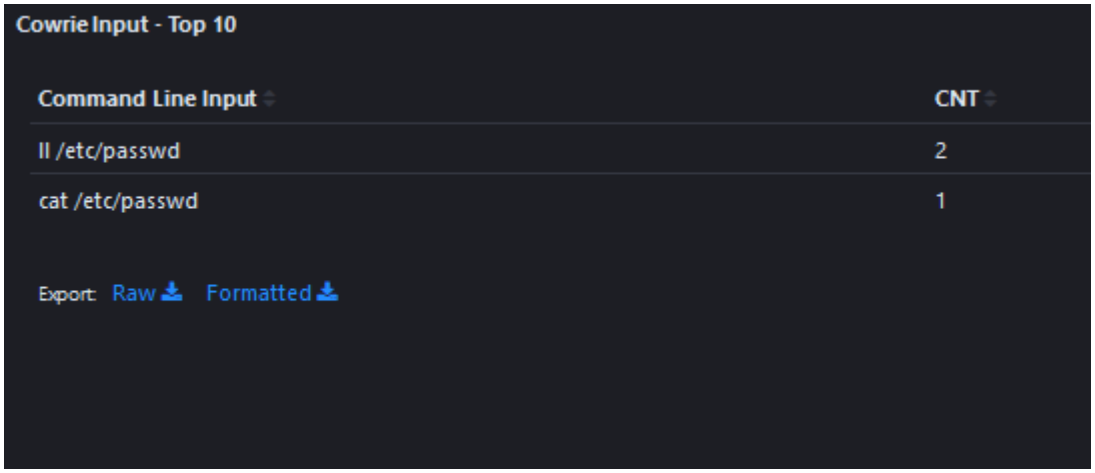
Palvelun toimivuutta testattiin ottamalla SSH-yhteys porttiin 22, jonka takana vastaa Cowrie palvelu. Cowrieta käydään läpi tarkemmin omassa osiossaan. Kyseinen tuote näyttää siltä, että hyökkääjä pääsisi yrittämään kirjautumista SSH yhteyden yli ja esimerkiksi bruteforce-menetelmää käyttämällä, pääsisi sisään ympäristöön satunnaisen yritysmäärän jälkeen. Ympäristö näyttää ulkoisesti oikealta, mutta todellisuudessa hyökkääjä ei voi tehdä järjestelmässä juuri mitään. Samaan aikaan Cowrie todellisuudessa tallentaa hyökkääjän käyttämät käyttäjätunnukset, salasanat ja palvelimella ajamat komennot.

Kibanan avulla kerättyä dataa on mahdollista visualisoida tehokkaasti ja useimmiten käytetyt sanat ja termit nousevat helposti havaittavaksi sanapilveksi, kuten kuviossa 6 näkyvät Kissa käyttäjänimi ja nakki useimmin käytettynä salasanana.



Kuvio 6. Kibanan avulla helposti havainnoitavaan muotoon visualisoidut listat eniten käytetyistä käyttäjätunnuksista ja salasanoista.

Kibanan kautta on myös nähtävissä suoraan statistiikkaa mistä maasta hyökkäykset saapuvat ja ovatko hyökkäyksen takana olevat IP-osoitteet esimerkiksi jo valmiiksi pahamaineisten IP-osoitteiden listalla. Cowrie tallentaa myös hyökkääjien sinne lataamat tiedostot ja osoitteet. Koska kyseisessä testiympäristössä ajettiin testejä vain sisäverkosta, ei statistiikkaa testauksen aikana saatu juurikaan kerättyä. Hyökkääjien palvelimella ajamat komennot listautuvat kuitenkin myös niiden käyttömäärien mukaiseen järjestykseen, joka on nähtävissä kuviossa 7.



Kuvio 7. Hyökkääjän Cowrien emuilomalla palvelimella ajamat komennot listattuna

T-Pot ympäristöstä on saatavilla Kibanan kautta helposti sisäistettävää vastaavanlaista dataa myös muista ajossa olevista hunajapurkeista. Muiden hunajapurkkien tiedoille löytyy valmiit pohjat, joihin on kerätty kyseisten hunajapurkkien keräämiä oleellisimpia tietoja valmiina visualisoitavaksi.

Nykyisessä T-Pot versiossa ei löytynyt suoraa mahdollisuutta sähköposti-ilmoituksen lähettämiseksi hälytyksen sattuessa, mutta kyseinen ominaisuus on tulossa tuottee-

seen lähitulevaisuudessa. Ominaisuus on kuitenkin teoriassa mahdollista ottaa käyttöön hyödyntämällä logstashista valmiiksi löytyvää hälytys toimintoa. Tarkempi selvitys sähköpostien lähettämiseksi hälytyksien yhteydessä jätettiin kuitenkin tekemättä, koska viralliseen julkaisuun on mahdollisesti tulossa erillinen ominaisuus tätä varten.

Kaiken kaikkiaan T-Potin asennus ja konfigurointi ovat todella yksinkertaisia ja käyttöönotto perusasetuksilla sujuu helposti tuotteen omia dokumentaatioita noudattaen. Dokumentaatiosta löytyy myös valmis ohjeistus tuotteen käyttöönottoa varten hyödyntäen Ansiblea, eli järjestelmän pystytys on todella helppo automatisoida tarpeen mukaan.

T-Potin asettamat järjestelmävaatimukset ovat raskaat muihin hunajapurkkituotteisiin verrattuna. Dokumentaatiossa suosituksena on vähintään 8 GB RAM muistia ja 128 GB vapaata levytilaa. Mikäli vaatimuksia halutaan keventää, voidaan ympäristö asentaa Sensor-pohjan asennuksella, jolloin järjestelmästä jätetään pois kokonaan ELK stack. Tällöin muistin osalta suosituksena on 4 GB. T-Pot on myös mahdollista asentaa alhaisimmilla resursseilla, mutta tällöin tuotteen toimivuutta ja vakautta ei voida kuitenkaan taata.

Cockpitin web-käyttöliittymästä on mahdollista seurata T-Potin käyttämiä resursseja graafimuodossa. Näkymä järjestelmän resurssien käytöstä Cockpitin kautta on nähtävissä kuviossa 8.

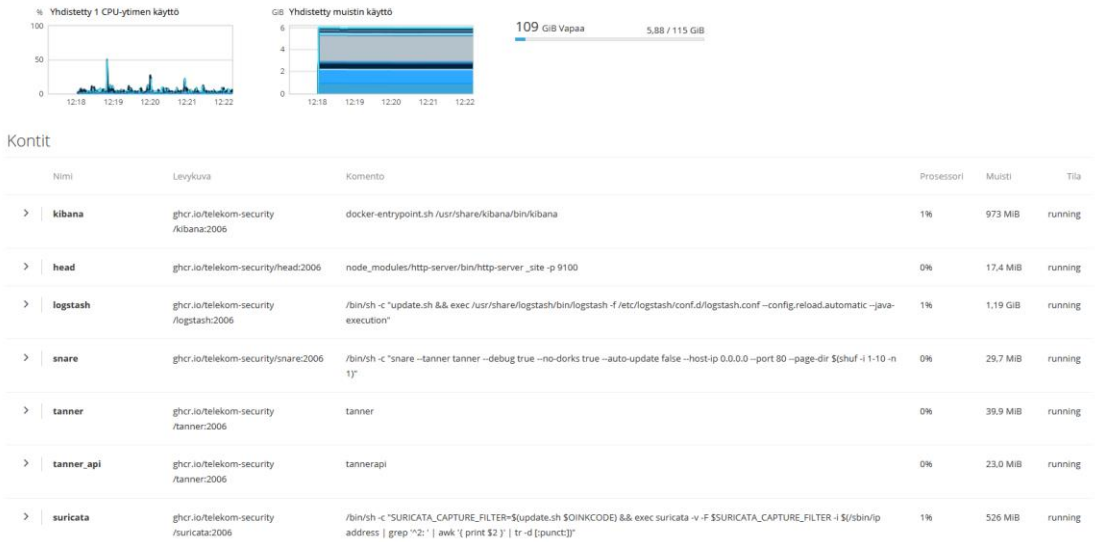
Kuvio 8. Cockpitin tarjoama näkymä palvelimen resurssien käyttötilastoihin.

Testiympäristössä T-Potille on annettu käyttöön 1 CPU, 8 GB muistia ja 128 GB levytilaa. Palvelimen muistin käyttö pysyi koko testauksien ajan tasaisesti 7 GB riippumatta

siitä, tehtiinkö palvelimella mitään tai ajettiinko hunajapurkkeihin hyökkäyksiä. Muiden resurssien käyttö oli maltillista, eikä suurempia piikkejä saatu aiheutettua muutamista yrityksistä huolimatta. Prosessorin käyttö pysyi jatkuvasti alle 25 % lukuun ottamatta muutamia hieman korkeampia piikkejä käytössä. Piikkienkin aikana pysyttiin kuitenkin alle 50 % käyttöasteessa. Levyn käyttö pysyi testien aikana alle 10 GB, joten minimivaatimuksena annettu 128 GB levytila tuntui testikäytössä korkeahkolta. Mikäli palvelimelle kuitenkin alettaisiin hyökätä aktiivisemmin, tulevat kerätyt lokitiedot ja hyökkääjien mahdollisesti käyttämät tiedostot viemään potentiaalisesti nykyistä enemmän tilaa. Samoin mahdolliset PCAP-tiedostot voivat viedä paljon tilaa, mikäli verkkoliikennettä halutaan tallentaa enemmän tutkittavaksi. T-Pot on suunniteltu olemaan nimenomaan yhteydessä ulkoverkkoon mahdollisimman suuren otannan saavuttamiseksi, joten pienimuotoisessa testauksessa sisäverkossa, tulokset eivät yllä samaan kuin julkiverkkoon sijoitetussa kohteessa.

Testiympäristössä pyöri kaiken kaikkiaan 33 konttia, mutta koska testitilanteessa kontit lähinnä kuuntelevat ja olivat valmiudessa yhteyksien varalta, ei resurssien käyttö ollut kovin suurta muistin käyttöä lukuun ottamatta. Toisaalta käyttötilanne vastaa hyvinkin pitkälle organisaation sisäverkkoon sijoitetun T-Pot palvelimen tilannetta. Palvelimen tehtävänä on kuunnella ja havaittaessa jotain poikkeavaa, aktivoitua toimimaan. Ajateltaessa tuotetta organisaation sisäverkkoon sijoitettavana tuotannollisena sensorihunajapurkkina, tuotteesta voisi huoletta jättää pois ELK stack komponentit. Tällöin palveluin käyttämää muistinkäyttöä saataisiin laskettua jonkin verran. Myös asennettavista hunajapurkkikonteista olisi mahdollista tinkiä ja valita ympäristöön sopivimmat. Tietoturvan näkökulmasta kontit, jotka kuvaavat palveluita, joita organisaatiossa ei ole käytössä, toimivat lähinnä hunajapurkin paljastavina sovelluksina ja turhana hyökkäyspinta-alana ja näin ollen turhana riskinä.

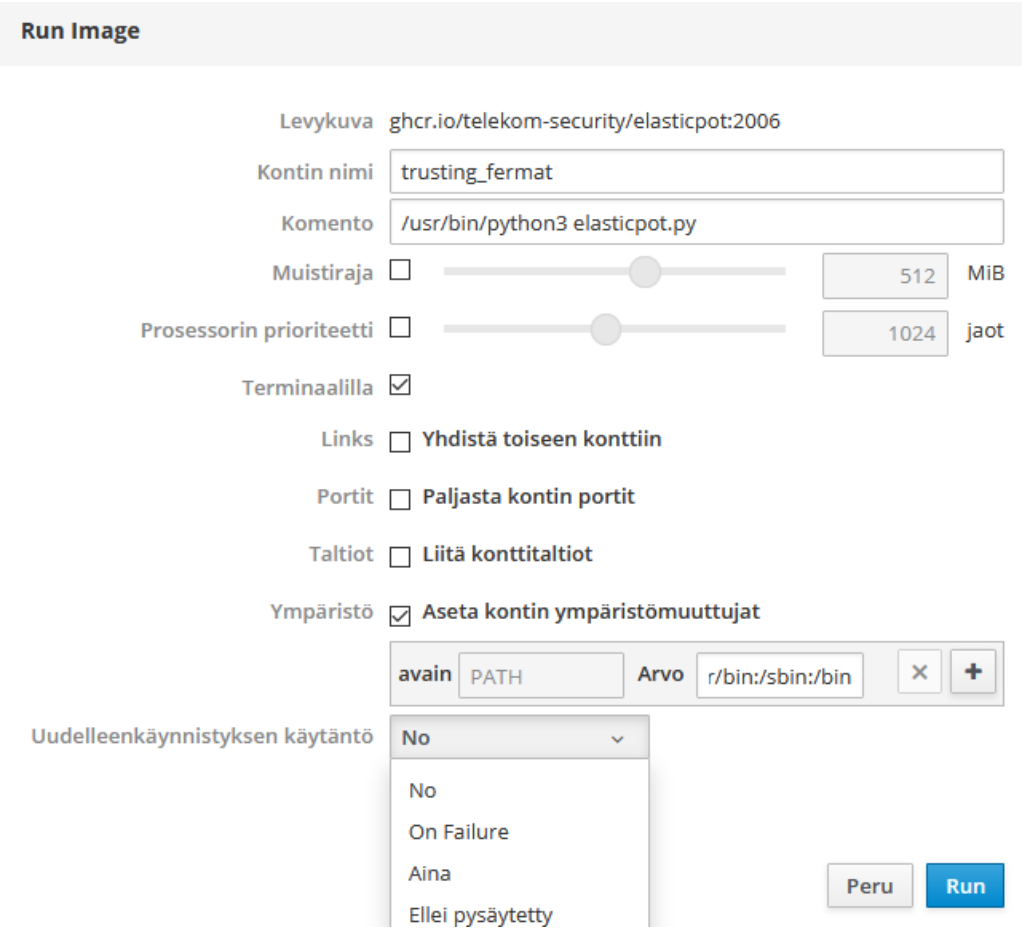
Cockpitin kautta on myös mahdollista hallinnoida ajossa olevia kontteja helposti graafisen käyttöliittymän kautta. Samasta käyttöliittymästä nähdään helposti myös eri konttien reaaliaikainen prosessorin ja muistin käyttö, sekä kontin tila. Sivulta nähdään myös komennot, joilla kontit ovat käynnistettynä ajoon sillä hetkellä. Yleisnäkymä konttien hallinnointisivustosta Cockpitin kautta on nähtävissä kuviossa 9.



Kuvio 9. Näkymä konttien hallinnasta Cockpitin web-käyttöliittymässä.

Käyttöliittymän kautta voidaan helposti nähdä, mikäli jossain ajossa olevassa kontissa on ongelmia. Valitsemalla jonkin ajossa olevista konteista, käyttäjä pääsee kyseistä konttia koskevaan näkymään. Näkymästä voidaan nähdä suoraan mistä levykuvasta kyseinen kontti on rakennettu ja muita hyödyllisiä lisätietoja konttia koskien. Samalta sivulta konttia on mahdollista myös hallinnoida myös muilla tavoin. Kontti voidaan esimerkiksi pysäyttää väliaikaisesti, käynnistää uudelleen ja tarvittaessa poistaa. Näiden lisäksi kontille on mahdollista asettaa yksilöllisiä raja-arvoja esimerkiksi muistin käytön suhteen.

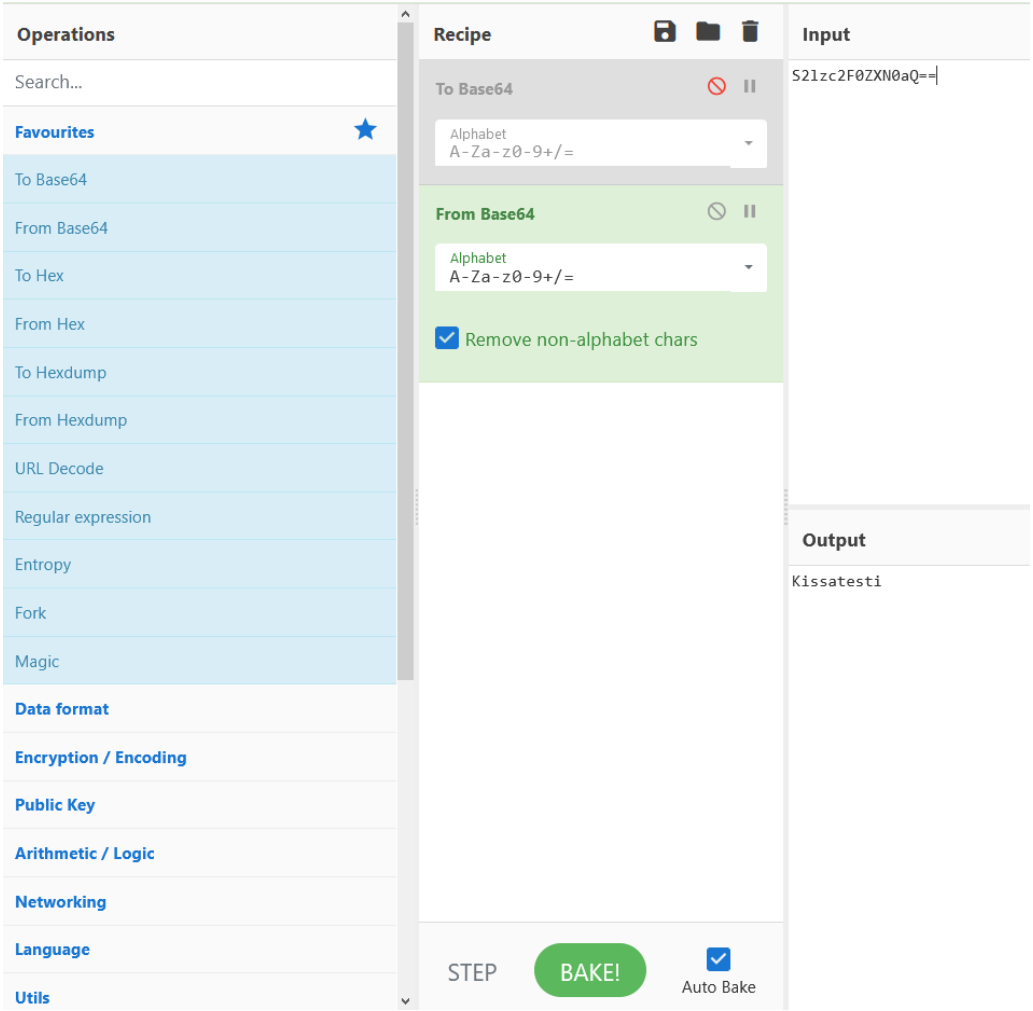
Kontit välilehdeltä löytyy myös listaus käytössä olevista levykuvista. Avattaessa levykuvia tarkemmin tutkittavaksi, nähdään suoraan mitkä ajossa olevat kontit käyttävät niitä. Mikäli jokin kontti on jouduttu aikaisemmin lopettamaan ja poistamaan esimerkiksi vikaantumisen vuoksi, voidaan levykuvista suoraan käynnistää uusi kontti web-käyttöliittymän kautta. Uuden kontin käynnistämisessä voidaan muokata kontin nimeä, komentoa kontin käynnistämiseksi, prioriteetteja ja muistirajoja, sekä erilaisia linkityksiä konttien välillä ja kontin käyttämiä portteja, sekä ympäristömuuttujia. Samalla ajossa olevalle kontille voidaan muokata uudelleenkäynnistyksen käytäntöjä, eli missä tilanteissa kontti ajetaan uudelleen käyntiin. Kontin käynnistämisen asetukset levykuvaa käytettäessä on nähtävissä kuviossa 10.



Kuvio 10. Kontin luonnin parametrien asettaminen levykuvaa käytettäessä.

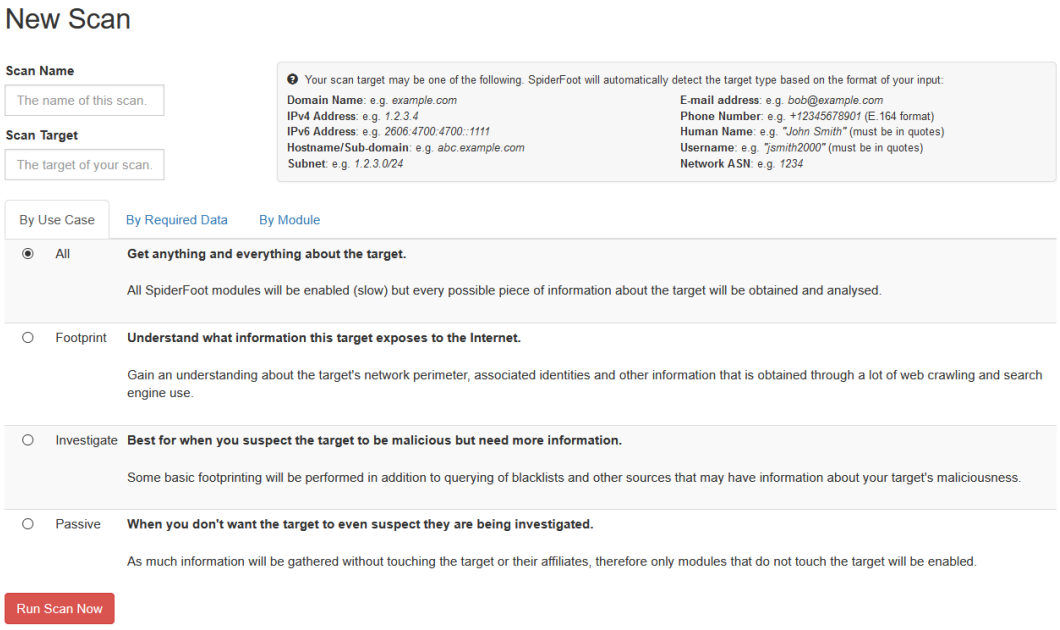
Oletusarvoisesti ajossa on ainoastaan asennuksen yhteydessä valitut kontit. Kyseiset kontit käynnistyvät palvelimen käynnistyessä oletuksena, mutta tätä on mahdollista muokata. Käynnistettävissä konteissa on valittu päälle asetus, että kontin kaatuessa, se poistetaan automaattisesti ja tilalle luodaan uusi puhdas kontti levykuvasta. Näin mahdollisesti saastunut tai kaatunut kontti ei pääse aiheuttamaan ongelmia vaan se tuhotaan välittömästi. Uudelleenluonnista huolimatta esimerkiksi hunajapurkkien keräämät lokitiedot ja mahdolliset muut kerätyt tiedostot säilytetään palvelimella tallessa. Käytännössä kontin luonti uudelleen tapahtuu välittömästi, eikä palveluun tule ihmissilmälle näkyvää käyttökatkosta.

Myös muiden asennettujen työkalujen käyttäminen onnistuu web-käyttöliittymää hyödyntämällä sujuvasti. Tästä esimerkkinä CyberChef, jonka avulla voidaan salata ja avata salauksia, muokata tekstiä eri koodauksista selkokielisiksi ja toisinpäin. Kuviossa 11 nähdään base64 muotoinen tekstikappale, joka on muokattu selkokieliseksi CyberChefiä käyttämällä.



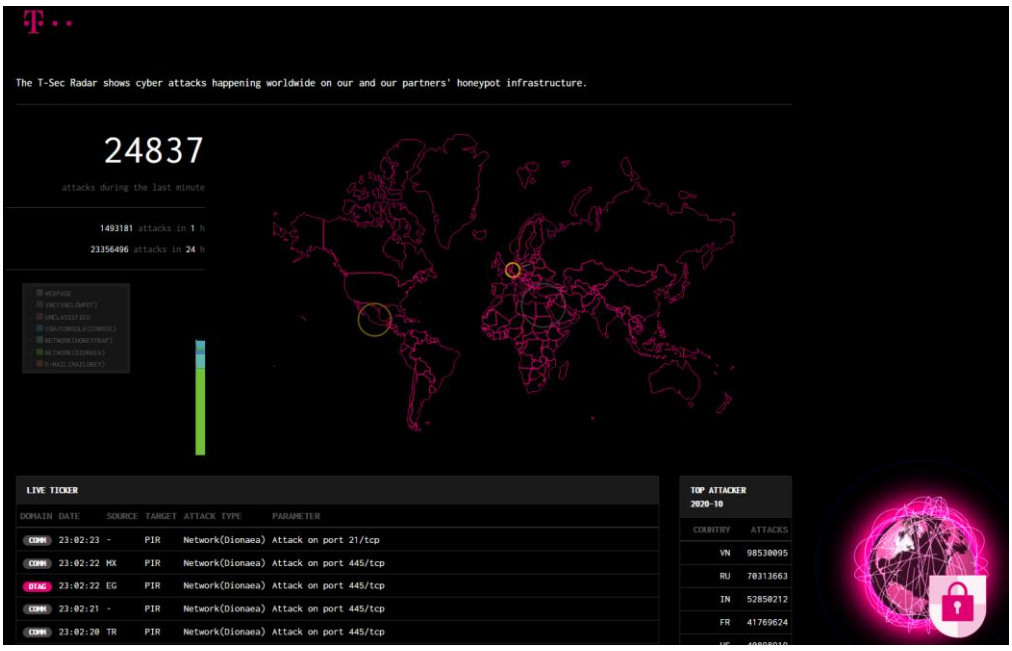
Kuvio 11. CyberChef työkalun käyttöliittymä.

Myös Spiderfootille on oma selkeä käyttöliittymänsä, johon pääsee helposti T-Potin webportaalin kautta. Työkalun kautta on mahdollista saada etsittyä tietoa esimerkiksi hyökkääjän käyttämistä IP-osoitteista. Käyttöliittymä uudelle haulle on nähtävissä ala-puolella olevassa kuviossa 12.



Kuvio 12. Tietojen hakuehtoja spiderfootin käyttöliittymästä.

Yllä mainittujen lisäksi T-Potin portaalin kautta löytyy painike visuaaliseen karttakuvaan, joka näyttää T-Pot hunajapurkkeihin kohdistuneet hyökkäykset ympäri maailman reaaliaikaisesti. Kartassa näytetään ainoastaan niiden ympäristöjen hyökkäystiedot, jotka eivät ole estäneet tietojen jakamista T-Potin yhteisölle. Palvelun nimi on Sicherheitstacho ja se on julkisessa verkossa oleva palvelu, joka ei vaadi erillistä kirjautumista. Näkymä Sicherheitstacho palvelusta nähdään kuviossa 13.



Kuvio 13. Sicherheitstacho eli reaaliaikainen kartta T-Pot ympäristöihin kohdistuvista hyökkäyksistä.

Testiasennuksesta saatujen tulosten perusteella T-Pot on hyvin toteutettu, monipuolinen ja helposti laajempaan käyttöön skaalattavissa oleva tuote, mutta muutamat sen käytössä olevat hunajapurkkituotteet ovat elinkaarensa loppupuolella ja niiden käyttämisen osalta on suositeltavaa vähintään arvioida mahdollisia riskitekijöitä. T-Potilla itsellään on asialleen omistautunut kehittäjä ja käyttäjäyhteisö, joka pyrkii aktiivisesti kehittämään tuotetta ja reagoimaan havaittuihin ongelmiin.

Tuotteen käyttöönotto oletusasetuksilla on todella suoraviivainen prosessi ja tuotteen käyttäminen on intuitiivisesti toteutettu. Yksi asia, jota dokumentaatiosta ei suoraan löytynyt, on hälytyksien edelleen ohjaaminen, joko sähköpostitse, SMS yhdyskäytävää tai rajapintoja hyödyntäen. Sähköpostin osalta kyseinen ominaisuus on ilmeisesti kehityksen alla parhaillaan, mutta varsinkin tuotantoverkkoon sensoriksi sijoitettavassa hunajapurkissa ominaisuudelle on suuri tarve. Yksi mahdollinen ratkaisu olisi palvelun asentaminen sensori-tyylisesti tarvittaviin pisteisiin yrityksen verkossa ja lokien toimittaminen jollain tavalla keskitetylle lokienhallinta palvelulle, jolloin sensoriverkoston keräämien datan ja mahdollisten hälytysten seuranta olisi suoraviivaisempaa ja nopeampaa.

8 Yhteenveto

Tutkimuksessa pyrittiin tutustumaan hunajapurkkien toimintaperiaatteisiin ja jaotteluihin erilaisten toiminnallisuuksien perusteella. Saatujen tietojen perusteella lähdettiin tutustumaan tällä hetkellä tarjolla oleviin avoimen lähdekoodin hunajapurkkituotteisiin. Tutkimuksen aikana kävi selväksi, että erilaisia tuotteita on tarjolla erittäin kattavasti ja suurin osa organisaatioissa yleisesti käytössä olevista sovelluksista löytyy myös hunajapurkki muodossa.

Varsinkin sensoreina toimivista hunajapurkkituotteista on mahdollista saada arvokasta lisätietoa organisaation sisäverkossa mahdollisesti toimivista hyökkääjistä, joita perinteiset tietoturvatuotteet eivät välttämättä havaitse. Hunajapurkkituotteista saatava tilannetieto saattaa joissain tilanteissa olla viimeinen varoitus, jonka ylläpitäjät ylipäänsä saavat järjestelmiensä murtamisesta. Hunajapurkkituotteista erityisesti T-Pot tarjosi suoraan visuaalisesti helposti havainnoitavaa tilannekuvaa ympäristöstään.

Lisäksi verkon ulkorajalle asetettuna sen kautta on mahdollista kerätä tietoa hyökkääjien mahdollisesti hyödyntämistä hyökkäysmenetelmistä, jonka perusteella ylläpitäjät pystyvät tarkistamaan kyseisiä haavoittuvuuksia omasta ympäristöstään.

Käytännön toteutuksissa ja dokumentaatioissa on kuitenkin usein huomattaviakin eroja eri tuotteiden välillä. Osaltaan tilanne selittyy kehittäjien skaalan vaihteluilla. Osa kehittäjistä saattavat olla yksittäisiä henkilöitä, jotka ovat alkaneet ohjelmoimaan omaa hunajapurkkituotettaan joko omasta tarpeestaan tai puhtaasti oppimiskokemuksena. Joukossa on kuitenkin myös yrityksien tekemiä ja ylläpitämiä pitkälle tuotteistettuja ratkaisuja. Myös kaupallisia ratkaisuja hunajapurkeista löytyy, mutta näiden osalta kyseessä harvemmin on enää avoimen lähdekoodin tuotteet, mikä oli asetettu rajoittavaksi tekijäksi tutkimuksen alussa. Osalla yrityksistä oli tarjolla oma avoimen lähdekoodin tuote ja sen lisäksi huomattavasti pidemmälle tuotteistettu versio samankaltaisesta tuotteesta, johon oli lisätty asennusta, käyttöä ja ylläpitoa helpottavia työkaluja, jotka tuottavat lisäarvoa käyttäjilleen.

Kerättyjen tietojen perusteella valittiin muutama avoimen lähdekoodin hunajapurkkituotteista tarkempaa tutkimista varten. Tutkimuksessa perehdyttiin tarkemmin valittujen hunajapurkkien ominaisuuksiin ja niistä tarjolla olevaan dokumentaatioon. Lopulta tuotteista valittiin yksi vielä tarkempia tutkimuksia varten ja siitä suoritettiin testiasennus virtuaalikoneelle.

Testiasennuksen perusteella arvioitiin tuotteen käyttöönotossa mahdollisesti vastaantulevia ongelmia ja haasteita. Lisäksi suoritettiin eri toiminnallisuuksien testauksia ja niiden vaikutusta palvelimen käyttämiin resursseihin. Resurssien käytön perusteella saatiin laadittua jonkinlaista arviota tuotteen mahdollisesti skaalautuvuudesta tarpeiden mukaisesti. Lopulliseen tarkempaan tutkimukseen valikoitui Saksalaisen Telekomin alla kehitetty T-Pot hunajapurkki. T-Pot ei varsinaisesti ole itse hunajapurkki, vaan pikemminkin alusta, jonka päällä loppukäyttäjän on helppo hallinnoida ja ylläpitää useista kontitetuista hunajapurkeista koostuvaa kokonaisuutta. T-Pot on pitkälle tuotteistettu ja pääsääntöisesti hyvin toimiva kokonaisuus, jonka avulla organisaation on helppo parantaa näkyvyyttään omaan verkkoonsa kohdistuviin uhkakuviin ja mahdollisesti sisäverkkoon päässeiden hyökkääjien toimintaan. Tuotetta ei tulisi lähteä viemään käyttöön todelliseen tuotantoympäristöön, ilman kaikkien käyttöönotettavien hunajapurkkikonttien täysimääräistä arviointia tietoturvan osalta.

Useimmista vähänkään isommilla resursseilla tuotetuista hunajapurkkituotteista löytyi joko suoraan tai kolmannen osapuolen tekemät valmiit pelikirjat asennusta varten. Kyseiset pelikirjat olivat luettavissa, joko tuotteiden varsinaisilla sivuilla tai kolmannen osapuolen sivuilla. Useissa tapauksissa yksittäiset käyttäjät olivat laatineet valmiit pelikirjat Ansiblea varten ja julkaisseet ne sitten esimerkiksi Githubissa. Valmiita pelikirjoja hyödyntämällä useimmat hunajapurkkituotteet on mahdollista pystyttää tarpeen vaatiessa nopealla aikataululla ja tarpeen mukaisesti. Tätä ennen tulee tietysti varmistua asennettavan hunajapurkin ja pelikirjan luotettavuudesta. Useista tunnetuimmista hunajapurkkituotteista löytyi myös yhteisön tuottamia valmiita Docker-levykuvia, joiden avulla kyseiset tuotteet on mahdollista ottaa helposti ajoon konttien sisältä.

9 Pohdinta

Opinnäytetyön tavoitteena oli tutustua hunajapurkkeja koskeviin aiempiin tutkimuksiin ja hunajapurkkien jaotteluihin niiden perusteella, sekä hunajapurkeista saataviin hyötyihin yrityksen sisäverkon näkyvyyttä parantavana tekijänä. Aiheesta löytyy huomattavan paljon jo olemassa olevia tutkimuksia, joiden ansiosta tuotteista saatiin laadittua laajahko tietoperusta erilaisia hunajapurkkeja ja niiden jaottelua koskien. Suurin osa tutkimuksista keskittyivät nimenomaan tutkimuksellisiin hunajapurkkeihin ja niiden kautta kerättäviin tietoihin hyökkääjien toiminnasta. Tuotannollisista, sensoreina toimivista hunajapurkeista, tuntui olevan huomattavasti vähemmän tietoa saatavilla. Tiedon kerääminen ja kokoaminen työn tietoperustaksi vei jonkin verran enemmän aikaa, kuin olin alun perin suunnitellut. Myös tutkimuksen teoriaosuuden laajuuden rajoittaminen koskemaan vain aihepiiriin liittyvää relevanttia oli haastavaa. Mielestäni lopputuloksena olevan tietoperustan rakentamisessa onnistuttiin kuitenkin lopulta hyvin.

Omasta mielestäni hunajapurkkeja on mahdollista hyödyntää huomattavasti nykyistä kattavammin organisaatioiden sisäisen tilannekuvan parantamiseksi ja samoilla linjoilla on myös Grimes, joka painottaa hunajapurkkituotteiden merkitystä perinteisten tietoturvatuotteiden ohessa (Grimes, 2005b). Liian usein yrityksissä keskitytään pitämään hyökkääjät ulkopuolella, mutta sisälle päästyään hyökkääjästä ei ole mahdollista saada minkäänlaista informaatiota. Valheellinen turvallisuudentunne siitä, kun kaikki

panokset on laitettu estämään hyökkääjän pääsy sisäverkkoon, on nykypäivänä lähinnä toivotonta optimismia. Usein hyökkääjälle riittää sisäänpääsyyn yksikin pieni murtuma vahvassa ulkomuurissa, jonka jälkeen ympäristön ylläpitäjien näkyvyys saattaa kadota lähes täysin. Tähän ongelmaan on mahdollista saada huomattavaa apua hunajapurkeista. Joissain tilanteissa hyökkääjillä on ollut pääsy järjestelmiin kuukausien tai jopa vuosien ajan, kenenkään tietämättä, eli sisäverkon näkyvyyden parantaminen on todella kriittistä monille yrityksille.

Ensimmäisestä osiosta saatuja tietoja pyrittiin hyödyntämään opinnäytetyön toisessa osiossa, tarjolla olevia avoimen lähdekoodin hunajapurkkituotteita vertailtaessa. Tarjolla olevien tuotteiden skaala oli laaja ja eläväinen. Useiden tuotteiden osalta löytyi viittauksia aikaisempiin vastaaviin ohjelmiin, joihin kyseinen sovellus perustui. Tutustuttaessa varhaisempaan versioon, voitiin kuitenkin usein dokumentaatiossa havaita viittaus vielä varhaisempaan edeltävään tuotteeseen. Usein hyvinkin erityyppiset hunajapurkit oli mahdollista jäljittää perustuvan samaan varhaisempaan hunajapurkkiin tai kehittäjäryhmään. Tutkimukselle asetetut rajaukset hunajapurkkien valintaan liittyen, auttoivat kuitenkin nopeuttamaan valittavien tuotteiden tutkimista. Hunajapurkkien tarjontaa tutkittaessa korostui kuitenkin myös teoriaosuudessa havaittu ilmiö, eli suurin osa tarjolla olevista hunajapurkeista oli nimenomaan tutkimuksellisia hunajapurkkeja ja sensoreina toimivien tuotannollisten hunajapurkkien määrä oli huomattavasti vähäisempi. Tuotannollisista hunajapurkeista parhaiten tehtävään olisi vastannut mahdollisesti Thinkstin OpenCanary, mutta aikaisemmin läpikäydyistä syistä johtuen tarkemmin tutkittavaksi tuotteeksi valikoitui kuitenkin lopulta Telekomin T-Pot. Kyseinen tuote on lähtökohtaisesti tarkoitettu tutkimukselliseksi hunajapurkiksi, mutta tarjoaa kuitenkin mahdollisuuden myös sensorityyppiseen asennukseen.

Valitun tuotteen käyttöönottoa organisaation sisällä pyrittiin emuloimaan asentamalla valittu tuote VirtualBoxissa toimivaan virtuaalikoneeseen ja tutkimalla tuotteen ominaisuuksia ja suorituskykyä. Testiasennuksesta saatuja tietoja pyrittiin arvioimaan tarkemmin pohdittaessa tuotteen skaalautuvuutta laajempaan käyttöön organisaatiossa. Mielestäni lopputuloksena saavutettiin tietoa kyseistä tarkempaan tutkiskeluun otetun tuotteen osalta ja erityisesti sen suhteen, mitä tuotetta käyttöönotettaessa tulisi ottaa erityisesti huomioon. Vaikka T-Pot onkin valmiinoloinen tuote, ei sitä voi varauksetta ja ilman jatkotutkimuksia, suositella käyttöönotettavaksi tuotantokäyttöön.

Tuotteesta löytyy muutamia hunajapurkkeja, joiden kehitystyö on jo päättynyt tai joiden osana käytetään kieliä ja kirjastoja, joiden tuki on loppunut jo aikaisemmin. Ainakin näiden lisäosien tarkempi arviointi täytyy toteuttaa ennen T-Potin varsinaista käyttöönottoa missään todellisessa tuotantoympäristössä. Mielestäni testiympäristöön asennuksella saavutettiin kuitenkin suuntaa antava kuva asioista, joita tulee ottaa huomioon varsinaisen tuotantoympäristöön tehtävän käyttöönoton yhteydessä ja laajamittaisempaa käyttöä pohdittaessa. Tuotteen resurssien käyttöön ja sen skaalautumiseen tulee kuitenkin suhtautua pienellä varauksella, sillä testiasennuksessa palvelimelle ei kohdistunut missään vaiheessa suurempaa kuormaa, jollaista se saattaisi kohdata esimerkiksi todellisen hyökkäyksen yhteydessä. Testaukset myös toteutettiin pienessä verkkoympäristössä, jossa ei kulje taustalla juuri mitään muuta liikennettä.

Todellisessa organisaation sisäverkossa taustalla saattaa olla järjestelmiä ja palveluita, jotka saattavat joissain tilanteissa laukaista hunajapurkeista vääriä hälytyksiä (false positives). Käytännössä kuitenkin jokainen organisaatio joutuu selvittämään tähän liittyvät mahdolliset ongelmat joko oman ympäristönsä laajan tietämyksen kautta tai perinteisellä kantapään kautta oppimisella.

Mikäli mahdollisissa lisäselvityksissä T-Potin käyttämiin hunajapurkkeihin liittyen ei löydy mitään akuuttia ja ongelmalliset lisäosat kytketään pois asennuksesta, on T-Pot suoraviivainen ja helposti käyttöönotettava tuote useimpiin organisaatioihin. Tuotetta ei kuitenkaan voi suositella käyttöönotettavaksi internetistä täysin eriytettyihin tai organisaatioiden toiminnan kannalta kriittisiin ympäristöihin tiettyjen sen käyttämien osien ja toiminnallisuuksien takia.

Lähteet

ABOUT US. N.d. Honeynet Projektin viralliset kotisivut. Viitattu 23.05.2020. https://www.honeynet.org/about/

ADBHoney. 2019. ADBHoney honeypot dokumentaatio. Viitattu 27.10.2020. https://github.com/huuck/ADBHoney

Architechture. 2020. T-Pot tuotteen arkkitehtuurikuvaus, tuotteen omasta dokumentaatiosta. Viitattu 22.10.2020. https://github.com/telekom-security/tpotce/blob/master/doc/architecture.png

Chapter 1. Introduction to Linux Containers. N.d. Red hat Linuxin dokumentaatiota konttiteknologiasta. Viitattu 20.10.2020. https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux_atomic_host/7/html/overview_of_containers_in_red_hat_systems/introduction_to_linux_containers

Cisco ASA honeypot. 2018. Cisco ASA tuotteeseen liittyvän hunajapurkin dokumentaatio. Viitattu 27.10.2020. https://github.com/Cymmetria/ciscoasa_honeypot

Cohen, F. 1998. The Deception ToolKit. Cohenin alkuperäinen julkinen ilmoitus Deception ToolKitin julkaisusta. Viitattu 20.05.2020. http://catless.ncl.ac.uk/Risks/19.62.html#subj11

Company Profile. 2020. Telekomin kotisivut. Viitattu 22.10.2020. https://www.telekom.com/en/company/company-profile

Configuration. 2018. OpenCanary Documentation. Viitattu 20.10.2020. https://opencanary.readthedocs.io/en/latest/starting/configuration.html#

Dicompot - A Digital Imaging and Communications in Medicine (DICOM) Honeypot. 2020. Dicompotin dokumentaatio. Viitattu 27.10.2020. https://github.com/nsmfoo/dicompot

ElasticPot. 2020. ElasticPotin gitlab sivusto. Viitattu 27.10.2020. https://gitlab.com/bontchev/elasticpot

Evans, J. N.d. What even is a container: namespaces and cgroups. Blogikirjoitus teknologioista konttien taustalla. Viitattu 20.10.2020. https://jvns.ca/blog/2016/10/10/what-even-is-a-container/

Grimes, R. 2005a. Honeypots for Windows. Viitattu 12.10.2020. https://books.google.fi/books?id=vT2j480fAdoC&printsec=frontcover&dq=hunajapurkkis+for+windows&hl=fi&sa=X&ved=2ahUKEwj135LzgobrAhXaBhAIHSxKCrIQ6AEwAHoECAAQAg#v=onepage&q&f=true

Grimes, R. 2005b. Honeypots as an early warning system. Viitattu 11.11.2020 https://www.infoworld.com/article/2673437/honeypots-as-an-early-warning-system.html

Heralding. 2020. Heralding hunajapurkin Github-sivusto. Viitattu 27.10.2020. https://github.com/johnnykv/heralding

Hyppönen, M. & Tuominen, T. Herrasmieshakkeri. F-Securen podcast. Viitattu 1.11.2020. https://www.f-secure.com/fi/business/podcasts/herrasmieshakkerit

Higgins, A. 2018. Adaptive Containerised Honeypots for Cyber-Incident Monitoring. University of Dublin, Trinity College. Viitattu 17.10.2020. https://www.scss.tcd.ie/Stefan.Weber/PDFs/Amber%20Higgins%20-%20MAI%20Dissertation%202018.pdf

Honeypot for CVE-2019-19781 (Citrix ADC). 2020. Citrix ADC hunajapurkin dokumentaatio. Viitattu 27.10.2020. https://github.com/MalwareTech/CitrixHoneypot

HoneyPy Plugins. 2020. HoneyPy hunajapurkin dokumentaatio. Viitattu 27.10.2020. https://honeypy.readthedocs.io/en/latest/plugins/

HoneySAP: SAP Low-interaction honeypot. 2020. HoneySap hunajapurkin dokumentaatio. Viitattu 27.10.2020. https://github.com/SecureAuthCorp/HoneySAP

Honeytrap. 2007. Honeytrap hunajapurkin dokumentaatio. Viitattu 27.10.2020. https://github.com/armedpot/honeytrap/

Hospelhorn, S. 2020. What is The Cyber Kill Chain and How to Use it Effectively. Viitattu 11.10.2020. https://www.varonis.com/blog/cyber-kill-chain/

How Ansible Works. 2020. Ansiblen dokumentaatio. Viitattu 13.11.2020 https://www.ansible.com/overview/how-ansible-works

Hutchins, E., Cloppert, M. & Amin, R. 2010. Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. Lockheed Martin Corporation. Viitattu 11.10.2020. https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf

Ihanus, J. Teknologian sokaisema kybertilannekuva, osa 1/3 – Inhimillinen tekijä. 2019. Viitattu 11.11.2020 https://huld.io/fi/nakemyksia/ajankohtaista/teknologian-sokaisema-kybertilannekuva-osa-1-3-inhimillinen-tekija/

Introduction to Deutsche Telekom's Honeypot Project. 2015. Blogikirjoitus Telekom Securityltä. Viitattu 22.10.2020. http://github.security.telekom.com/2015/02/hunajapurkkis-introduction.html

IPP Honey. 2020. IPP honey nimisen hunajapurkin dokumentaatio. Viitattu 27.10.2020. https://gitlab.com/bontchev/ipphoney

Jain, Y. & Singh, S. 2011. Honeypot based Secure Network System. International Journal on Computer Science and Engineering. Viitattu 12.10.2020. http://www.enggjournals.com/ijcse/doc/IJCSE11-03-02-030.pdf

Koutoupis, P. 2018. Everything You Need to Know about Linux Containers, Part I: Linux Control Groups and Process Isolation. Viitattu 19.10.2020. https://www.linuxjournal.com/content/everything-you-need-know-about-linux-containers-part-i-linux-control-groups-and-process

Koistinen, M. 2011. Tilannetietoisuus ja tilannekuva operatiivisessa liikenteenhallinnassa. Liikenneviraston tutkimuksia ja selvityksiä. Viitattu 11.11.2020 https://julkaisut.vayla.fi/pdf3/lts_2011-54_tilannetietoisuus_ja_tilannekuva_web.pdf

Kral, P. 2011. The Incident Handlers Handbook. Viitattu 17.10.2020 https://www.sans.org/reading-room/whitepapers/incident/incident-handlers-handbook-33901

Kyberrikokset. N.d. Rikoksentorjuntaneuvosto. Rikoksentorjuntaneuvoston verkkosivut. Oikeusministeriö. Viitattu 10.04.2020. https://rikoksentorjunta.fi/kyberrikokset

Laari, T., Flyktman, J., Härmä, K., Timonen, J. & Tuovinen, J. 2019. #kyberpuolustus: Kyberkäsikirja Puolustusvoimien henkilöstölle. Viitattu 11.11.2020 https://www.doria.fi/bitstream/handle/10024/173254/%23kyberpuolustus%20verkko%20%28interaktiivinen%20pdf%29%20%28002%29.pdf?sequence=1&isAllowed=y

Limnéll, J. 2013. Kyberrikollisuus on liian helppoa. Sitran julkaisema blogikirjoitus. Viitattu 10.04.2020. https://www.sitra.fi/blogit/kyberrikollisuus-liian-helppoa/

Malone, S. 2016. Using an expanded cyber kill chain model to increase attack resiliency. Black hat USA 2016. Viitattu 12.10.2020. https://www.blackhat.com/docs/us-16/materials/us-16-Malone-Using-An-Expanded-Cyber-Kill-Chain-Model-To-Increase-Attack-Resiliency.pdf

Mailoney. 2018. Mailoney hunajapurkin dokumentaatio. Viitattu 27.10.2020. https://github.com/awhitehatter/mailoney

Medpot. 2018. Medpot Hunajapurkin dokumentaatio. Viitattu 27.10.2020. https://github.com/schmalle/medpot

MushMush Foundation. N.d. Snare/Tanner hunajapurkin ylläpitäjien kotisivut. Viitattu 27.10.2020. http://mushmush.org/

Näsi, M. & Kaakinen, M. 2019. Kyberrikollisuus. Rikollisuutilanne 2018 Helsingin yliopisto, Kriminologian ja Oikeuspolitiikan Instituutti. Rikollisuuskehitys tilastojen ja tutkimusten valossa. Toimittanut Danielsson, P. Viitattu 10.04.2020. https://helda.helsinki.fi/bitstream/handle/10138/307111/Katsauksia_36_Rikollisuustilanne_2018_2019.pdf?sequence=2&isAllowed=y

Pols, P.2017. The Unified Kill Chain: Designing a Unified Kill Chain for analyzing,comparing and defending against cyber attacks. Cyber Security Academy. Viitattu 11.10.2020 https://www.csacademy.nl/images/scripties/2018/Paul-Pols---The-Unified-Kill-Chain.pdf

RDPY. 2020. RDPY hunajapurkin dokumentaatio. Viitattu 27.10.2020. https://github.com/citronneur/rdpy

Reidy, P. 2013. Combating the Insider Threat at the FBI. Presentation at Black Hat USA 2013. Viitattu 11.10.2020. https://media.blackhat.com/us-13/US-13-Reidy-Combating-the-Insider-Threat-At-The-FBI-Slides.pdf

Rubens, P. 2017. What are containers and why do you need them? Viitattu 18.10.2020. https://www.cio.com/article/2924995/what-are-containers-and-why-do-you-need-them.html

Senanayaka, L. 2018. What is Elastic stack which everyone is talking about???. Viitattu 22.10.2020. https://medium.com/@lakinisenanayaka/what-is-elastic-stack-which-everyone-is-talking-about-6c5d94d83983

Sheikh, M. 2018. An analysis of Glutton — All Eating honeypot. Viitattu 27.10.2020. https://medium.com/@cstayyab/an-analysis-of-glutton-all-eating-honeypot-625adf70a33b

Spitzner, L. 2003. Honeypots, Definitions and Value of honeypots. Viitattu 12.10.2020. https://crysp.uwaterloo.ca/courses/cs458/W11-lectures/local/www.spitzner.net/hunajapurkkis.html

tpotce. 2020. T-Pot tuotteen Github sivusto, jossa käydään tuotetta läpi. Viitattu 22.10.2020. https://github.com/telekom-security/tpotce

Uusimmat tiedot Vastaamon tietomurrosta. 2020. Ylen seuranta Vastaamon tietomurtoa koskien. Viitattu 29.10.2020. https://yle.fi/uutiset/3-11612399

Wahl, M. 2019. A seed for OpenCanary. Viitattu 20.10.2020. https://medium.com/@michael.wahl/a-seed-for-opencanary-23a43e4d5085

Welcome to Conpot's documentation! 2020. Conpot-hunajapurkin dokumentaatio. Viitattu 27.10.2020. https://conpot.readthedocs.io/en/latest/

Welcome to dionaea's documentation!. 2020. Dionaean oma dokumentaatio. Viitattu 27.10.2020. https://dionaea.readthedocs.io/en/latest/index.html

What is a honeypot? How it can lure cyberattackers. N.d. Nortonin ylläpitämä tietoturvaan liittyvä blogi. Viitattu 20.05.2020. https://us.norton.com/internetsecurity-iot-what-is-a-hunajapurkki.html

Yahyaoui, A. 2014. Testing Deceptive honeypots. Master's Thesis. Naval Postgraduate School. Viitattu 12.10.2020. https://www.hsdl.org/?view&did=760442

Yegulapl, S. 2019. What is Docker? The spark for the container revolution. Viitattu 20.10.2020. https://www.infoworld.com/article/3204171/what-is-docker-the-spark-for-the-container-revolution.html

Liitteet

Liite 1. T-Potin arkkitehtuurikuvaus