Onni Tolonen

USB-laitteiden hyökkäyksiä isäntälaitetta vastaan

Tietotekniikan kandidaatintutkielma

7. tammikuuta 2025

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Onni Tolonen

Yhteystiedot: toloojxz@student.jyu.fi

Ohjaaja: Tuomo Rossi

Työn nimi: USB-laitteiden hyökkäyksiä isäntälaitetta vastaan

Title in English: USB attack methods towards the host

Työ: Kandidaatintutkielma

Sivumäärä: 30+0

Tiivistelmä: Tutkielman tarkoituksena on tarkastella ja koota kirjallisuudesta löytyvien usb-laitteiden avulla suoritettuja isäntälaitetta vastaan tehtyjä hyökkäyksiä ja puolustautumiskeinoja näiden varalle. Tutkielman tutkimusmenetelmät koostuvat Scopuksen ja Google Scholarin käytöstä tiedonhakuvälineinä.

Avainsanat: Tietoturva, USB, tietoturvahyökkäykset

Abstract: The purpose of this thesis is to take a look at and categorize attacks made using USB devices. The attacks are mostly limited to ones that are directed towards the host. The thesis will also cover defenses for these attacks found in literature. The research methods consist of Google Scholar and Scopus that have been used to find relevant academic literature on the topic.

Keywords: information security, USB

Sisällys

1 Johdanto

USB (engl. “Universal Serial Bus”) on maailmanlaajuisesti levinnyt standardoitu sarjaväylärakkitehtuuri, joka on muodostunut ensisijaiseksi oheislaitteiden liitäntätavaksi. Se on yhä useammin käytössä fyysisenä tiedonsiirtotapana ja näyttäisi jatkavan leviämistään yhä useampaan käyttötarkoitukseen. Nykypäivänä USB:tä ei käytetä kuitenkaan ainoastaan tiedonsiirtoon mutta myös virranjakoon esimerkiksi puhelimien ja useiden oheislaitteiden tapauksessa. Koska USB on löydettävissä monista eri laitetyypeistä on myös sen käyttäjäkunta laaja koskien ryhmiä keskiverto puhelinkäyttäjistä tiedustelupalvelujen tietoturvatietoisiin ammattilaisiin. USB:n valtavasta levinneisyydestä ja sen sisältävien laitteiden monimuotoisuudesta huolimatta USB:n tietoturva on jäänyt usein taka-alalle niin käyttäjien kuin USB:tä standardisoivan organisaation taholta. Tämä tutkimus valottaa osittain myös USB:n avoimuuden ja helppokäyttöisyyden vaikutuksia sen kautta tehtävien tietoturvahyökkäysten toteuttamiseen.

USB:n hyödnytäminen tietoturvahyökkäyksiin ei ole kuitenkaan ole uusi ilmiö. USB 2.0 julkistettiin jo vuonna 2000 (J. Tian ym. 2018) ja vuonna 2005 tietoturvatutkija Abe Usher toi laajemman yleisön tietoisuuteen tavan hyödyntää Applen Ipod -laitteita varastamaan tietoja USB:n kautta siihen liitetyltä tietokoneelta (Al-Zarouni 2006). USB:n yleistyessä on tullut esiin yhä useampia USB:tä hyödyntäviä hyökkäystapoja joiden monimuotoisuus toimintamallien ja tarkoitusperien osalta on varsin laaja.

Tutkimuksen teema on ajankohtainen ja tärkeä sikäli, että se koskettaa miltei kaikkia USB-laitteita hyödyntäviä käyttäjäryhmiä tavallisista käyttäjistä yrityksiin ja jopa valtiollisiin tahoihin. USB:n kautta tehtävät hyökkäykset vaativat kyllä fyysisen pääsyn hyökkäyksen kohteena olevaan laitteeseen mutta usein riittää saada vaarasta tiedostamaton käyttäjä liittämään pahantahtoisesti muunneltu mutta ulkoisesti vaarattomalta näyttävä laite hyökkäyksen kohteena olevaan laitteeseen. Tämä on erityisesti vaara yrityksille joilla on usein riittämättömät suojatoimet USB-hyökkäyksiä vastaan. Haitallista ohjelmistoa tai jopa haitallisia loogisia piirejä sisältävä laite voi olla esimerkiksi niin USB-muistitikku kuin USB-johto.

Tämän tutkimuksen aiheena on tutkia millaisia eri USB:tä hyödyntäviä hyökkäystapoja on olemassa, miten niitä voidaan luokitella ja millaisia suojautumistapoja niille on. Tutkimus

käy siis myös läpi käyttöjärjestelmiin ja laitteisiin sijoitettavia uhkia lievittäviä ja estäviä ominaisuuksia sekä parannuksia. Tutkimus käsittää ainoastaan oheislaitteiden taholta isäntälaitetta vastaan käytyjä hyökkäyksiä, joihin lasketaan kuuluvaksi myös tarkoitusperäinen informaation isäntälaitten ja laitteen-välillä käydyn kommunikaation vuotaminen tai lähetys elektromagneettisten radioaaltojen tai muiden tapojen välityksellä. Tutkimus ei siis ota huomioon USB-laitetta vastaan käytyjen hyökkäysten kuten kryptografisesti suojatun muistitikun tietojen varastamisen tapoja.

Tutkimusmenetelmänä toimii kirjallisuuskartoitus. Hakuprosessi muodostuu olemassaolevien tutkimusten etsimisestä valituilla hakukoneilla. Hakukoneina toimivat sekä Google Scholar, sekä Scopus.

Luvun 2 tarkoituksena on antaa tarvittava ymmärrys USB:n laitteistoon ja sen toimintalogiikkaan myöhemmin esiteltyjen hyökkäysten toimintatapojen hahmottamiseksi. Luku kolme koskee valittuja kategorioita ja luvussa neljä käydään läpi hyökkäystapoja jaoteltuna OSI-malliin pohjautuviin kategorioihin. Luku viisi käsittelee suojautumistapoja luvussa neljä esitellyille hyökkäystavoille samoihin kategorioihin perustuvien luokitusten avulla.

2 USB:n toimintatapa

2.1 Laitteisto

USB-kommunikaatio tapahtuu laitteen (engl. device) ja isännän (engl. host) välillä. Isäntään voidaan liittää monia USB-laitteita mutta laitteella voidaan käydä kommunikaatiota vain yhden isäntälaitteen kanssa kerrallaan. Esimerkiksi tietokone johon on liitetty USB-muistitikku, hiiri ja näppäimistö on yksi tälläinen laitekonfiguraatio. Isäntälaitteeseen kuuluu kontrolleri, jonka kautta isäntälaite käy kommunikaatiota liitettyjen laitteiden kanssa (Jodeit ja Johns 2010). Kontrolleriin kuuluu sulautettu hub-laite, jota kutsutaan juurihubiksi (engl. root-hub) (Jodeit ja Johns 2010; Nissim, Yahalom ja Elovici 2017). Hub-laitteet tarjoavat mahdollisuuden liittää useita laitteita yhtä kontrolleria kohden (Nissim, Yahalom ja Elovici 2017). USB-laitteita on useita ja niille on määritelty omat luokkakoodinsa niiden käyttötapojen mukaisesti ("Defined USB Class Codes" 2023). Tyypillisimpiä USB-laitteita ovat esimerkiksi massamuistilaitteet, HID (engl. human interface device) -laitteet kuten hiiret ja näppäimistöt, sekä edellä mainitut hubit. Näistä käytetään usein myös nimitystä "oheislaitteet". Yhdellä fyysisellä laitteella voi kuitenkin olla monta USB-spesifikaation määrittelemää käyttötapaa, joita kutsutaan rajapinnoiksi (engl. interface) (D. Tian ym. 2016). Isäntäkontrollerille jokainen rajapinta näyttäytyy omana tietueenaan, jolle ladataan oma ajurinsa (D. Tian ym. 2016). Hyvä esimerkki tällaisesta komposiittilaitteesta on esimerkiksi webcam, joka tarjoaa mikrofonin ja videontallennuksen yhden laitteen muodossa (Nissim, Yahalom ja Elovici 2017). Näiden lisäksi yhdellä tai usealla rajapinnalla on ainakin yksi konfiguraatio, joka tarjoaa tietoja laitteen rajapinnoista ja tarpeista. Konfiguraatioita voi olla monia tarjoamaan esimerkiksi laitteen oman virransyötön- tai USB-väylän virransyötön toiminnallisuuksia (Murphy ja Family 2014). Vain yksi konfiguraatio voi olla käytössä kerrallaan (Jodeit ja Johns 2010; J. Tian ym. 2018).

2.2 Enumeraatio

Kun USB-laite liitetään isäntälaitteen USB-väylään määrittää kontrolleri aluksi laitteiden välisen tiedonsiirtonopeuden (J. Tian ym. 2018; Nissim, Yahalom ja Elovici 2017). Laite

resetoidaan ja alkaa enumeraatioprosessi, jonka tarkoituksena on saada selville liitetyn laitteen laitetiedot ajurin valintaa ja sujuvaa kommunikaatiota varten. Enumeraation laiteinfo lähetetään USB-standardin määrittelemien deskriptoreiden muodossa. GetDeviceDescriptors kysyy aluksi laitteen tuotteen- ja valmistajan tunnusnumerot (engl. vendor ID ja product ID), tuotenumeron (engl. serial number) (J. Tian ym. 2018), sekä muita tietoja kuten tuetut USB-versiot ("USB Device Enumeration" 2009), joiden prosessoinnin jälkeen GetConfigDescriptors-kutsulla haetaan konfiguraatioiden tiedot. (J. Tian ym. 2018) Viimeisenä kysytään GetInterfaceDescriptors rajapintojen selvitystä varten. (J. Tian ym. 2018)

2.3 Ajurien alustus

Kun tarvittavat tiedot on luettu, siirtyy isäntälaitteen käyttöjärjestelmä etsimään sopivia ajureita laitetta varten (Jodeit ja Johns 2010; J. Tian ym. 2018). Tarkemmat prosessit ajurien valintaan riippuvat käyttöjärjestelmästä. Ajureiden valinta Microsoft Windowsin osalta tapahtuu tuote- ja valmistaja ID:itten perusteella (Jodeit ja Johns 2010). Myös Linux valitsee ajurit laitteen ilmoittamien tietojen ja ajurien ilmoittamien tietojen yhtensopivuuden mukaan (Corbet, Rubini ja Kroah-Hartman 2005; Madieu 2017). Käyttöjärjestelmät hyödyntävät usein luokka-ajureita, jotka toimivat useiden yhteen USB-luokkaan kuuluvien ja niiden spesifikaatioita noudattavien USB-laitteiden kanssa (Jodeit ja Johns 2010).

2.4 USB:n toimintamallin vaarat

USB:n käyttökokemus on suunniteltu olemaan mahdollisimman vaivaton käyttäjälle. Tämä tarkoittaa käytännössä sitä, että USB-laitteet hoitavat esitellyt alustusprosessit automaattisesti käyttäjältä piilotettuna. Tietoturvan kannalta tämä muodostuu ongelmaksi, sillä USB:tä standardoiva USB-IF ("USB Implementers Forum") on katsonut USB-laitteiden tietoturvan olevan käyttäjien ja laitteiden valmistajien vastuulla (J. Tian ym. 2018). Käyttäjien tulisi USB-IF:n mukaan varmistua itse käyttämiensä laitteiden turvallisuudesta ja laitteiden valmistajien tulisi implementoida tarvittavat tietoturvaa parantavat ominaisuudet (J. Tian ym. 2018). J. Tian ym. (2018) kuitenkin huomauttaa, että vastuun asettaminen käyttäjille on ongelmallinen, sillä käyttäjillä ei ole tapaa varmistua käytettyjen laitteiden turvalli-

suudesta. Aikaisemmin spesifikaatio luotti täysin, että kaikki USB-kommunikaatioon liitetyt laitteet ovat turvallisia (Kang ja Saiedian 2017) ja minkäänlaista autentikaatiota ei ole erikseen määritelty (Neuner ym. 2018) kunnes vuonna 2016 USB-IF esitteli USB 3.0 C-tyypille määritellyn autentikaatiotavan (J. Tian ym. 2018).

USB:n helppokäyttöisyys ja sen joustavuus aiheuttavat ongelmia USB:n tietoturvan kannalta. USB-protokolla luottaa laitteiden ilmoittamiin yksilöiviin enumeraatiovaiheessa ilmoitettuihin parametreihin kuten valmistajan tunnusnumeroon ja tuotenumeroon, jotka ovat kuitenkin helposti valehdeltavissa. Rajapinnat mahdollistavat monien ominaisuuksien käytön yhden laitteen muodossa samanaikaisesti mutta tietoturvan kannalta tämä voi muodostaa ongelman, sillä käyttäjällä ei ole tietoa onko laite ohjelmoitu käyttämään rajapintoja, joita laite ei ulkomuodoltaan edusta. Yksi esimerkki tällaisesta haitallisiin tarkoituksiin ohjelmoiduista laitteista on muistitikku, joka on uudelleenohjelmoitu sisältämään normaalin massamuistitoimintojen ohella myös näppäimistönä toimivan rajapinnan, joka syöttää isäntälaitteelle käskyjä hyökkääjän määrittelemällä tavalla. Näitä hyökkäyksiä käydään tarkemmin luvussa 4.2.1.

3 Kategoriat

Tämän tutkimuksen aihe-alue on rajattu koskemaan vain isäntälaitetta vastaan tehtäviä hyökkäyksiä USB-laitteen taholta, johon katsotaan kuuluvaksi myös USB:n langatonta tiedonsiirtoa hyödyntävät hyökkäystavat. Jokainen esitelty hyökkäys siis vaatii, että haitallisesti toimiva laite liitetään fyysisesti hyökättävään laitteeseen tai USB:n langattoman tiedonsiirron toimintamatkan etäisyydellä on pahantahtoinen toimija tai laite. Haitallista toimintalogiikkaa sisältävä laite voi siis päästä toimintaetäisyydelle joko uhasta tietämättömän ihmisen tai pahantahtoisen toimijan avulla.

USB:n tietoturvaongelmia tutkivassa kirjallisuudessa on esitetty useita taksonomioita hyökkäysten lajitteluun. Nissim, Yahalom ja Elovici (2017) kategorisoivat hyökkäykset hyökkäyksen toteuttamiseen tarvittavan USB-laitteiston mukaisesti, jossa kategoriat ovat ohjelmoitavat mikrokontrollerit, USB-laitteet ja erikseen suuren jännitteen virtapiirejä hyödyntävät laitteet. Näistä USB-laitteet jakautuvat kahteen alikategoriaan, joissa USB-laitteet ovat joko uudelleenohjelmoituja tai ohjelmoimattomia laitteita. (Nissim, Yahalom ja Elovici 2017) Useat hyökkäykset kuitenkin ovat laitteistoriippumattomia ja sama hyökkäys voidaan toteuttaa usella Nissim, Yahalom ja Elovici (2017) esittelemällä kategorialla. Esimerkiksi mikrokontrollerilla voidaan toteuttaa suuri osa uudelleenohjelmoitujen USB-laitteiden hyökkäyksistä.

J. Tian ym. (2018) jakaa USB-kommunikaation neljään kerrokseen: ihmis-, sovellus-, kuljetus- ja fyysiseen kerrokseen. Ihmiskerros sisältää hyökkäyksen osallisten kommunikaation ja toiminnan. Sovelluskerrokseen liittyvät käyttäjätason sovellukset kommunikaatioon osallistuvilla laitteilla. Kuljetuskerrokseksi katsotaan USB-laitteen laitteisto-ohjelmistot sekä isäntälaitteen käyttöjärjestelmä USB-kommunikaatiota toteuttavine ohjelmistoineen (“USB stack”) ja fyysinen kerros käsittää kommunikaation USB-väylällä. Jokainen hyökkäys asetetaan yhteen näistä kerroksista sen mukaan, mitä kukin hyökkäys pääasiallisesti hyödyntää toimiakseen. Jokaiseen kerrokseen liitetään kuitenkin myös alakategoria, joka ryhmittelee hyökkäykset niiden toimintamekanismien ja lopputulosten mukaisesti.

Tässä tutkielmassa hyödynnetään J. Tian ym. (2018) esittelemää taksonomiaa. Kerrosmalli

minimoi tehokkaasti päällekkäisyyttä eri kategorioiden välillä hyökkäyksien osalta ja identifoi hyökkäysten pääasialliset toimintamekanismit USB:n toimintamalliin suhteutettuna. Koska reaalimaailman hyökkäykset voivat hyödyntää juurikin monia toimintamekanismeja lopputuloksen saavuttamiseksi on erityisen tärkeää selkeyttää pääasiallinen hyökkäystapa. Ihmiskerros jätetään käsittelemättä, sillä se jää tutkimuksen aihepiirin ulkopuolelle.

4 Hyökkäystavat

Reaalimaailman USB hyökkäykset voivat käyttää useaa alempana esiteltyjen kerrosten hyökkäystapaa tavoitteen saavuttamiseksi. Jokainen alikappale kerää yhteen identifioituun hyökkäysprimitiiviin kuuluvat hyökkäykset ja esittelee niiden taustaa.

4.1 Sovelluskerros

Sovelluskerrokseen kuuluvat käyttäjätason (engl. “user space”) ohjelmistot ja niiden interaktiot liitetyn USB-laitteen kanssa (J. Tian ym. 2018). Sovelluskerroksen alikappaleissa tarkastellaan datan piilottamista usb laitteilla sekä koodi-injektoita. Koodi-injektiot ovat termi hyökkäyksille, joissa hyökkääjä saattaa oman koodinsa sovelluksen tulkittavaksi tai suoritettavaksi (“Code Injection”, n.d.). Tiedon (engl. ”data”) piilottaminen koskee tiedon piilottamista usb laitteelle. Tieto voi olla isäntälaitteelta anastettua tai hyökkääjän asettamia haitallisia tiedostoja, joita voidaan hyödyntää hyökkäämiseen.

4.1.1 Tiedon piilottaminen

Tiedon piilottaminen on tapa, jota voidaan hyödyntää sekä koodi-injektoiden, että tietojen kaappaamisessa. Koodi-injektioissa injektoitavat ohjelmistot voidaan piilottaa USB laitteelle estäen käyttäjää tai viruksentorjuntaohjelmistoja havaitsemasta niitä. Nissim, Yahalom ja Elovici (2017) esittelevät seitsemän erilaista tapaa piilottaa tiedostoja, joista neljä ensimmäistä perustuvat visuaaliseen piilottamiseen tai käyttäjän harhaanjohtamiseen ja kolme jälkimmäistä koskevat tiedon piilottamista hyödyntämällä tiedostojärjestelmien ominaisuuksia.

Tiedostot voidaan piilottaa visuaalisesti käyttäjältä siten, etteivät ne näy esimerkiksi Windows Explorerissa käyttäjän selatessa liitetyn USB-laitteen tiedostoja. Tämä voidaan saavuttaa muuttamalla tiedoston attribuutteja piilotettavaksi tai merkitsemällä tiedoston tärkeäksi käyttöjärjestelmään kuuluvaksi tiedostoksi. Toinen tapa piilotukseen on asettaa tiedosto kansioon, jonka ikoni ja nimi ovat läpinäkyviä näin piilottaen kansion. Kolmas ja neljäs tapa ovat tapoja johtaa käyttäjä harhaan merkitsemällä tiedosto eri formaatiksi kuin mitä se todellisuudessa on tai sulauttamalla tiedostoon NTFS-tiedostojärjestelmän sallima vaihtehtoinen

tietovirta (engl. “alternate data stream”). (Nissim, Yahalom ja Elovici 2017)

Nissim, Yahalom ja Elovici (2017) esittelevät myös tapoja hyödyntää tiedostojärjestelmää datan piilottamiseen. Tiedostot voidaan asettaa tiedostojärjestelmän toimintaan varatuille sektoreille tai kokonaan tiedostojärjestelmän merkitsemän data-osion ulkopuolelle. Toinen tapa on siirtää tiedostot viallisiksi merkityille sektoreille. Tämä voidaan saavuttaa manipuloimalla tiedostojärjestelmän metadataa merkitsemällä myös toiminnalliset sektorit viallisiksi NTFS tai FAT tiedostojärjestelmissä.

4.1.2 Autorun ja koodi-injektiot

Autorun-hyökkäykset olivat erityisesti 2000-luvun ensimmäisen vuosikymmenen ongelma USB:n kautta suoritettavien hyökkäysten historiassa (Nissim, Yahalom ja Elovici 2017). Autorun on alun perin Windows95 käyttöjärjestelmään luotu toiminto, joka lukee liitetyllä media-laitteella kuten CD-ROM levyllä sijaitsevan autorun.inf tiedoston automaattisesti liittämisen yhteydessä ja suorittaa sen ilmoittaman ohjelman (“Beyond Autorun: Exploiting vulnerabilities with removable storage” 2011). Windows 7:ssä Autorun muutettiin toimimaan ainoastaan CD ja DVD -levyille (“Beyond Autorun: Exploiting vulnerabilities with removable storage” 2011). Autorun funktionaliteetti on sittemmin otettu pois käytöstä useimmissa käyttöjärestelmissä (Nissim, Yahalom ja Elovici 2017) mutta käyttöjärjestelmäbugien takia samantyyppisten hyökkäysten uhka ei ole täysin poissuljettu (J. Tian ym. 2018).

Autorun tyyppiseen hyökkäykseen USB laitteelta suojaamattomaan versioon Windows-käyttöjärjestelmästä riittää sijoittaa haluttu haittaohjelma sekä autorun.inf USB laitteen juuritiedostoon, joka näkyy käyttöjärjestelmälle. Simppelin autorun.inf tiedoston tulee sisältää ohjelman suorittamiseksi ainakin rivit “[autorun]"ja tämän alle “open=”, jonka jälkeen ilmoitetaan ajettavan ohjelmiston nimi kuten esimerkiksi: “open=haittaohjelma.exe”.(“Beyond Autorun: Exploiting vulnerabilities with removable storage” 2011) Ohjelma suoritetaan tällöin enumeraation jälkeen automaattisesti ilman käyttäjän hyväksyntää.

Joitakin tunnettuja hyökkäyksiä, jotka hyödyntävät autorun.inf-tiedostoja, ovat esimerkiksi Conficker ja Flame, jotka hyödynsivät toimintaansa myös nollapäivähaavoittuvuuksia (J. Tian ym. 2018). Tunnetuin USB-laitteelta suoritettu koodi-injektiota hyödyntävä hyökkäys

on Stuxnet. Stuxnet käytti ensisijaisesti hyökkäysvektorikseen .LNK tiedoston läpikäyvän käyttöjärjestelmän haavoittuvuutta hyväksseen. USB-laitteella sijaitseva .LNK tiedosto ilmoittaa tiedostoa varten näytettävän ikonin mutta Stuxnetin onnistui saada käyttöjärjestelmän suorittamaan .LNK tiedoston ja sen lukemistavan haavoittuvuuden avulla USB laitteella sijaitseva haittaohjelma. (Nissim, Yahalom ja Elovici 2017)

4.2 Kuljetuskerros

Kuljetuskerros sisältää liitettävän laitteen laiteohjelmiston ja isäntälaitteen käyttöjärjestelmän, jossa USB-kommunikaatiosta vastaavat ohjelmistot sijaitsevat. Kuljetuskerroksiin kohdistuvat hyökkäykset voidaan jakaa naamioituviin ja korruptoiviin hyökkäyksiin. (J. Tian ym. 2018). Kuljetuskerroksen hyökkäykset hyödyntävät nimenomaan USB protokollan toimintaa. Naamioitumiskategorian nimi perustuu siihen, että laite voi sekä ulkoisesti, että funktionaliteetiltaan edustaa normaalia käyttäjän olettamaa laitetta mutta sen toiminta voi olla protokollan sallimissa rajoissa jotain muuta. Protokollan korruptio pyrkii nimensä mukaisesti hyödyntämään itse USB-viestintää, jotta isäntälaitteen kommunikaatiosta vastaavat ohjelmistot saataisiin toimimaan hyökkääjän tarkoitusten mukaisesti. Tällöin paketteja voidaan esimerkiksi korruptoida eli muokata protokollan määrämien sääntöjen vastaisesti tai lähettää vastaanottavien ohjelmistojen odottamien tapojen vastaisesti.

4.2.1 Protokollan naamioiminen

Protokollan naamioiminen käyttää hyväkseen USB:n rajapintoja. Koska yksi laite voi sisältää monta rajapintaa tarjoten useamman toimintonnon yhdelle laitteelle, voi hyökkääjä uudelleenohjelmoida USB-laitteen laiteohjelmiston käyttämään sille tarkoituksettomia rajapintoja. Kyseessä ei kuitenkaan tarvitse olla tavallinen USB-laite vaan myös mikrokontrollereja voidaan ohjelmoida toimimaan esitellyllä tavalla. Käyttäjän kannalta tilanne on vaarallinen sikäli, että laite voi ulkomuodoltaan molemmissa tapauksissa näyttää harmittomalta. Yleisemmin näitä USB-laitteiden laiteohjelmistjen muokkausta hyödyntäviä hyökkäyksiä, jotka piilottavat rajapintoja, kutsutaan nimellä “BadUSB” (Lu ym. 2021). Neuner ym. (2018) ja Sun, Lu ja Liu (2021) käyttävät termiä BadUSB kuitenkin ainoastaan kaupallisille USB-laitteille, joiden laiteohjelmistoa on muokattu valmistajan oletusten vastaisesti jättäen ohjel-

moitavat mikrokontrollerit BadUSB:n kategorian ulkopuolelle.

Rajapintoja voidaan hyödyntää esimerkiksi HID-injektioihin. Jos kyseessä on muistitikku, jonka laiteohjelmistoa voidaan vapaasti muokata, on mahdollista ohjelmoida tavallisen muistitikun toiminnallisuuden lisäksi laite käyttämään HID-rajapintaa. Tällöin käyttäjä kykenee käyttämään laitetta normaalisti mutta sen ohella laite voi injektoida näppäinpainalluksia tai hiiren liikkeitä. Näppäininjektiohyökkäykset toimivat usein hyödyntäen käyttöjärjestelmän pikanäppäimiä avatakseen komentorivin hyökkäyksen toteuttamiseen. Tämän jälkeen mahdollisten hyökkäysten kirjo on hyvin laaja ja voi johtaa hyvinkin vakaviin lopputuloksiin. Hyökkääjä voi esimerkiksi ladata ja suorittaa haitallisia ohjelmia ja kopioida sensitiivisiä tietoja (Lu ym. 2021). Brandao (PhD) ja Scanavez (Degree) (2021) näyttivät miten ATTiny85 mikrokontrollerilla voidaan esimerkiksi tehdä näppäininjektiohyökkäys, joka muokkaa Windowsin ACL (“Access Control List) -asetuksia jakaen sensitiivisiä tietoja hyökkääjän saataville. Hyökkäykset voivat rajoittua käyttäjän oikeuksien mukaan mutta nollapäivähaavoittuvuudet ja käyttöoikeuksien laajentaminen (engl. “priviledge escalation”) ovat mahdollisia HID-injektioiden avulla sivuuttaen käyttäjän oikeudet.

HID-injektioiden ohella on myös löydetty muita rajapintojen piilotukseen perustuvia hyökkäyksiä. Laiteohjelmisto voidaan muokata esimerkiksi käyttämään USB ethernet-rajapintaa, jolloin on mahdollista tehdä palvelunestohyökkäys tai DNS-väärennys (engl. “DNS-spoofing”). Jos isäntälaitteiston käyttöjärjestelmän DHCP:stä vastaava ohjelmisto ei autentikoi DHCP-serverin identiteettiä on hyökkääjän mahdollista antaa isäntälaitteelle ethernet rajapinnan kautta väärennettyjä osoitteita ja ohjata käyttäjä hyökkääjän sivustolle aidon sivuston sijaan. Väärennetty sivusto voi näyttää esimerkiksi pankkisivustolta lähettäen syötetyt tiedot suoraan hyökkääjän serverille. (Nissim, Yahalom ja Elovici 2017)

4.2.2 Protokollan korruptio

Protokollan korruptiossa ongelmien lähteinä ovat haavoittuvuudet isäntälaitteen USB-kommunikaatiosta vastaavissa ohjelmistoissa. Hyökkääjät voivat hyväksikäyttää odottamattomien syötteiden prosessointia esimerkiksi kaataakseen käyttöjärjestelmän tai saadakseen aikaan hallinnoimatonta koodin suorittamista käyttöjärjestelmätasolla. Käyttöjärjestelmiä tes-

tatessa “fuzzauksen” (engl. “fuzzing”) avulla on löydetty laajalti haavoittuvuuksia USB-kommunikaatiosta vastaavien käyttöjärjestelmien ohjelmistojen parista niin Linuxin, Windowsin ja FreeBSD:n osalta (J. Tian ym. 2018). Yhä uudempien fuzzing-menetelmien ilmaantuessa löydetään uusia haavoittuvuuksia osoittaen ongelman olevan aiheellinen nykyistenkin käyttöjärjestelmien osalta. Peng ja Payer (2020) löysivät esittelemällään fuzzaustyökalulla 26 uusi uutta haavoittuvuutta Linuxin USB-alisysteemeistä joista 16 oli vakavia muistinhallintaan liittyviä haavoittuvuuksia.

4.3 Fyysinen kerros

Fyysinen kerros käsittää kahden kytketyn laitteen USB-väylien välisen yhteyden. Fyysiseen kerrokseen kuuluu siis esimerkiksi USB-väylällä kulkeva signaali isäntälaitteen ja liitetyn USB-laitteen välillä. Kyseiseen kerrokseen kohdistuviin hyökkäyksiin liittyvät erityisesti signaalin manipulointi eli signaalin eheyden rikkominen ja signaalin vuoto ulkopuolisille toimijoille joka rikkoo luottamuksellisuutta (J. Tian ym. 2018). Näistä muodostuvat kaksi alikategoriaa: signaalin vuoto ja signaalin manipulointi. Saatavuutta rikkovia hyökkäyksiä ovat analogiset signaali injektiot, jotka pyrkivät tuhoamaan isäntälaitteen komponentteja sähkövirran avulla (J. Tian ym. 2018).

4.3.1 Signaali-injektiot

Nimensä mukaisesti signaalin manipulointi kohdistuu isännän ja laitteen välisen kommunikaation muunteluun tai injektointiin signaalin tasolla. Tunnetut hyökkäykset ovat injektiohyökkäyksiä, jolloin hyökkäyksen toteuttamiseksi käyttäjällä on oltava HID-laite kuten USB-näppäimistö tai hiiri liitettynä hyökkäyksen kohteena olevaan laitteeseen. Hyökkäyksiä on sekä langattomia että fyysisiä. Fyysinen signaalin injektio vaatii erillisen laitteen, joka on liitettävä normaalin HID-laitteen ja isäntälaitteen väliin. Esimerkkinä tästä on ”turnipschool”-nimellä kulkenut NSA Playset -tutkijoiden esittelemä USB-kaapeliin sulautettu laite, joka kykenee kommunikoimaan hyökkääjän kanssa radioaaltojen välityksellä (Nissim, Yahalom ja Elovici 2017). Laite kykenee siis vuotamaan kaiken kaapelin kautta kulkevan informaation ja injektoimaan hyökkääjän käskyjä etäältä reaaliajassa. Hyökkäykseen siis riittävät hyökkääjän tai laitteen läsnäolo radioaaltojen kantomatkan etäisyydellä ja muunnellun USB-

johdon sijoittaminen HID-laitteen ja isäntälaitteen välille.

Langattomat injektiohyökkäykset ovat “langallisia” injektiohyökkäyksiä helpompia toteuttaa, sillä ne eivät vaadi hyökkääjän erillistä fyysistä laitetta liitettäväksi käyttäjän USB-väylään ja hyökkääjä kykenee toimimaan etäältä. Langattomat injektiohyökkäykset hyödyntävät pääasiallisesti langattomien USB HID laitteiden puutteellista tietoturvaa. Langattomaan USB hiireen tai näppäimistöön kuuluu itse laite ja sen kanssa kommunikoiva langaton vastaanotin (engl. “dongle”), joka liitetään isäntälaitteen USB-porttiin. Kaikki langattomat HID-laitteiden vastaanottimet eivät varmista, että vastaanottimelle kommunikoiva lähettäjä on autentikoitu laite mahdollistaen autentikoimattomien USB-pakettien hyväksymisen, sillä vastaanotin ei osaa erottaa ulkopuolista laitetta legitiimistä käyttäjän laitteesta vastaanotettujen pakettien perusteella, jolloin hyökkääjä voi täysin vapaasti syöttää isäntälaitteelle näppäinkomentoja tai hiiren liikettä.

Vuonna 2016 Bastille toi julki “mousejack”-nimellä kulkevan hyökkäystavan, johon kuuluu kolme erilaista tapaa injektoida paketit hyödyntäen langattomia USB-vastaanottimia. Ensimmäinen hyödyntää USB-hiirien langattomien USB-vastaanottimien tapaa käsitellä saapuvia paketteja. Jos vastaanotin ei varmista, että saapuneen USB-paketin ja sen lähettäneen USB-laitteen tyypit kuuluvat yhteen, voi hiirelle tarkoitetulle vastaanottimelle vapaasti syöttää näppäinpainalluksia. Hyökkäys olettaa, että saapuvia paketteja ei autentikoida. Toinen hyökkäys hyödyntää joidenkin USB-vastaanottimien tapaa kommunikoida näppäimistöjen kanssa. Joissakin tapauksissa näppäimistön paketit on salattu kolmannelle osapuolelle tiedon vuotamisen estämiseksi mutta vastaanotin ei kaikissa tapauksissa oleta salausta vaan hyväksyy ulkopuoliset näppäinpainallukset. Kolmas hyökkäys koskee myös joidenkin valmistajien vastaanottimia, jotka voidaan pakottaa yhdistämään ulkopuolinen laite vastaanottimen kanssa ilman käyttäjän toimintaa. Normaalisti USB-laite voitaisiin yhdistää esimerkiksi kadonneen laitteen tilalle salauksesta huolimatta mutta tämä vaatisi normaalisti käyttäjän aloittamaa yhdistämistilaa (engl. “pairing mode”). (“MouseJack Technical Details” 2016; Nissim, Yahalom ja Elovici 2017)

Sähkövirtaa itsessään hyökkäystarkoituksiin hyödyntävistä USB-laitteista on vakiintunut nimitys USB Killer. USB Killer liitetään isäntälaitteen USB-porttiin jolloin se alkaa lataamaan laitteen sisältämiä kondensaattoreita kunnes täysien kondensaattorien sähkövirta vapaute-

taan kerralla isäntälaitteen USB-väylän datalinjoja pitkin ja sama toistetaan kunnes laite irrotetaan tai isäntälaitteen virransyötön komponentit hajoavat (J. Tian ym. 2018). USB Killer:in saatavilla oleva kaupallinen versio (4) kykenee vapauttamaan 215 volttia kerrallaan (“usbkill.com” 2023).

4.3.2 Salakuuntelu

Salakuuntelu kytkeytyy vahvasti myös singaali-injektioihin, sillä ne toimivat osittain samojen periaatteiden mukaisesti. Mainittu turnipschool toimii esimerkiksi myös informaatiota vuotavana laitteena. Samoin langattomien HID-laitteiden vastaanottimien heikkouksia voidaan hyödyntää salakuunteluun. Radio- tai muilla taajusaalloilla liikkuvat salaamaattomat HID-paketit ovat helposti saatavilla radio-vastaanottimen ja tietokoneen avulla jopa lähes sadan metrin päästä (“MouseJack Technical Details” 2016).

Fyysisesti asetettavista salakuunteluun tarkoitetuista laitteista turnipschool:in lisäksi on myös NSA:n esittelemä Cottonmouth-1, joka toimii samoin kuin turnipschool mutta kykenee lähettämään ainoastaan johdon läpi kulkevat USB-paketit hyökkääjälle ilman hyökkääjän syötettä (Nissim, Yahalom ja Elovici 2017). USB-johtoihin upotettavat laitteet eivät kuitenkaan aina sisällä langattomaan tiedonsiirtoon tarvittavaa sirua. Yleisemmin näppäin painalluksia tallentavista laitteista tai ohjelmistoista käytetään sanaa näppäilytallennin “keylogger”. USB:n tapauksessa kyseessä on pieni laite, joka asetetaan USB-porttiin tallentavaksi välikädeksi, johon liitettynä esimerkiksi USB-näppäimistön jokainen painallus kirjataan näppäintallentimen muistiin myöhempää keruuta varten (J. Tian ym. 2018). Näppäintallentimen käytössä luotetaan siihen, ettei käyttäjä huomaa tai ymmärrä näppäintallentimen länsäoloa tai tarkoitusperää. Sensitiivisiä tietoja on myös mahdollista urkkia ilman, että hyökkäävä laite toimii välikätenä usb tiedonsiirrossa. Neugschwandtner, Beitler ja Kurmus (2016) näyttivät miten ennen USB 3.0 spesifikaatiota liitetyn USB-laitteen on mahdollista kerätä isäntälaitteen toisille USB laitteille osoitettua tietoa (J. Tian ym. 2018). USB:n 2.0 spesifikaation mukaan isäntälaitteen tietylle laitteelle osoittamat paketit välitetään kaikille laitteille, jolloin niiden oletetaan jättämään huomiotta niille kuulumattomat paketit (Neugschwandtner, Beitler ja Kurmus 2016). Tällöin riittää, että viaton USB laite ja pahantahtoinen USB laite ovat liitettynä samaan USB hubiin, jolloin pahantahtoinen laite voi kerätä sille kuulumatonta tietoa

(Neugschwandtner, Beitler ja Kurmus 2016).

Kirjallisuudesta löytyy myös useita isäntälaitetteen konfiguraatiota urkkivia hyökkäyksiä. J. Tian ym. (2018) kutsuvat näitä sormenjäljennyshyökkäyksiksi (engl. “fingerprinting attacks”). Liitetty USB-laite voi urkkia tietoja isäntälaitteesta eri tavoin. Laite voi esimerkiksi hyödyntää URB (engl. “USB request block”) pakettien sisältämiä tietoja sekä USB-pakettien ajoituksista pääteltäviä tietoja (J. Tian ym. 2018). Isäntälaitteen konfiguraatiota koskevia tietoja voidaan käyttää hyväksi kohdennettuun hyökkäykseen (J. Tian ym. 2018), jossa hyökkäysmekanismi perustuu johonkin tietyn konfiguraation haavoittuvuuteen.

5 Suojautumistavat

Puolustuskeinojen ryhmittelyssä on käytetty J. Tian ym. (2018) esittelemää tapaa tunnistaa puolustusimplementointien hyödyntäviä primitiivejä ja niiden kategorisointia sen mukaan, mitä kerrosta ne puolustavat. Puolustusprimitiivit voivat olla koko kokonaisia jolloin ne tarjoavat kokonaisvaltaisen suojan tiettyä hyökkäystä kohtaan tai osittaisia, jolloin niiden tarjoama suoja toimii vain tiettyjen olettamusten toteutuessa J. Tian ym. (2018). Puolustuskeinoissa tuodaan myös esille kirjallisuudessa esiteltyjä primitiivejä hyödyntäviä implementaatioita ja niitä koskevia huomioita.

5.1 Sovelluskerros

Isäntälaitteen karaistaminen on yksi tapa puolustautua sovelluskerrosta hyväksikäyttäviä hyökkäyksiä vastaan. Kategoriaan kuuluvat esimerkiksi modifikaatiot ja lisäykset isäntälaitteen käyttöjärjestelmään puolustuskyvyn parantamiseksi. Yksi tapa puolustautua sovelluskerroksen tasolla on kiristää käyttöjärjestelmän perusasetuksia estämään sekä allekirjoittamattomien ohjelmistojen suorittaminen tai kaikki automaattinen suorittaminen USB-laitteilta (J. Tian ym. 2018).

Viruksentorjuntaohjelmistot ovat yksi yleinen tapa parantaa käyttöjärjestelmän puolustusta haittaohjelmia vastaan. Näiden ohjelmistojen toimintalogiikka perustuu haittaohjelmien signatuurien vertailuun tunnettujen haittaohjelmien signatuurien tietokantaa vasten, jolloin haittapuolina ovat kyvyttömyys löytää nollapäivähaavoittuvuuksia ja jatkuva tarve tietokannan päivittämiseksi, jotta uudet haittaohjelmat voidaan havaita (Griscioli ja Pizzonia 2021). Viruksentorjuntaohjelmistot ovat voimattomia BadUSB-hyökkäyksiä vastaan Griscioli ja Pizzonia (2021), sillä niillä ei ole pääsyä USB laitteen laiteohjelmistoon (J. Tian ym. 2018; Kharraz ym. 2019) ja koska BadUSB:n toiminta on yleisesti ottaen protokollan mukaista hyökkäyksestä riippuen. On kuitenkin mahdollista, että viruksentorjuntaohjelmisto voi estää hyökkäyksen etenemisen myöhemmässä vaiheessa Griscioli ja Pizzonia (2021). Näiden huomioiden nojalla viruksentorjuntaohjelmistot voidaan katsoa osittaisiksi puolustuskeinoiksi USB:n kautta leviäviä haittaohjelmia vastaan.

Isäntälaitetta emuloivia virtuaaliympäristöjä voidaan hyödyntää eristämään tutkittava USB-laite isäntälaitteesta ja tutkimaan laitteen käyttäytymistä. Tian, Bates ja Butler (2015) esittelemä GoodUSB uudelleenohjaa tutkittavan laitteen QEMU-virtuaalikoneeseen tarvittaessa ja monitoroi sen käyttäytymistä . Angel ym. (2016) esittelemä Cinch taas kaventaa isäntälaitteen hyökkäysvektoreita eristämällä isäntälaitteen USB kontrollerin ohjaamalla USB kommunikaation kulkemaan virtuaalikoneen kautta. Tämä virtuaalikone voi sisältää sekä ohjelmistokerroksen, että kuljetuskerroksen torjuntamenetelmiä kuten viruksentorjuntaohjelmistoja ja ajureihin perustuvien kontrollien menetelmiä. Virtuaalikoneen tarve ja laitteen käytön keskeytyminen tekevät tästä puolustuskeinosta kuitenkin epäkäytännöllisen peruskäyttäjälle. (J. Tian ym. 2018)

5.2 Kuljetuskerros

Kuljetuskerroksen puolustusprimitiivit tähtäävät pääasiallisesti BadUSB:n torjumiseen. Yksi tapa lähestyä ongelmaa on laiteohjelmiston varmistus. Laiteohjelmiston varmistus voi rajoittua itse laitteeseen jolloin ilman salaista avainta USB laite ei hyväksy uuden laiteohjelmiston asentamista laitteelle (Griscioli ja Pizzonia 2021) estäen täten laiteohjelmiston muokkaamisen hyökkääjän toimesta. Vaikka laiteohjelmiston allekirjoitukset ovat yleisesti hyvä varotoimi luo kolmanteen osapuoleen luottaminen uuden hyökkäysvektorin (J. Tian ym. 2018). Tian, Bates ja Butler (2015) kuitenkin huomauttavat, että valtiollisille tahoille salattujen avainten hankkiminen ei kuitenkaan ole ongelma kuten Stuxnet osoittaa. Isäntälaitteet ovat myös suojauttuja ainoastaan niin kauan kuin niihin liitetään vain suojattuja USB laitteita joka taas kaventaa huomattavasti käyttökelpoisten laitteiden joukkoa (Griscioli ja Pizzonia 2021) vaikeuttaen keskivertokäyttäjän valinnanmahdollisuuksia.

5.2.1 Laiteohjelmiston varmistus ja palomuuri

Jos laiteohjelmisto on saatavilla laitteelta analysoitavaksi luotettavasti, voidaan sitä myös tarkastella BadUSB:tä varten injektoidun logiikan varalta. FirmUSB hyödyntää symbolisen suorituksen (engl. “symbolic execution”) menetelmää USB-laitteelta hankitun laiteohjelmiston tarkistamiseksi. Laiteohjelmisto on kuitenkin harvoin saatavilla. Laiteohjelmistojen muokkaamattomuutta on myös yritetty varmistaa ajoituksiin perustuvilla menetelmillä.

VIPER esittelee systeemin jossa isäntälaite lähettää liitetylle USB-laitteelle haasteita, joiden suorittamisaikaa mitataan. Kyseinen tapa kuitenkin vaatii haasteita tukevan laiteohjelmiston ja täten myös valmistajan tuen. (J. Tian ym. 2018)

Kuljetuskerroksen toinen primitiivi soveltaa tietoverkkojen puolustuksesta tuttua palomuuria USB kommunikaatioon. Usbfilter on USB pakettien palomuuri Linuxille, jonka perusideana on liittää Usb laitteen rajapinta ja käyttöjärjestelmän prosessi niin, että USB-laite voi kommunikoida vain sallittujen prosessien kanssa (D. Tian ym. 2016). Käytännössä esimerkiksi USB-mikrofonin rajapinta voidaan liittää Microsoft Teamsiin jolloin USB-laite voi lähettää paketteja vain ja ainoastaan Teamsiin. Lisäksi jokainen USB-paketti seulotaan Usbfilterin kautta käyttäjän määrittelemien sääntöjen mukaisesti (D. Tian ym. 2016) ja tarvittaessa säännökset voidaan määritellä hyvinkin hienojakoisiksi. Usbfilter toimii osittaisena puolustuskeinona BadUSB:ta vastaan. Palomuuri voidaan esimerkiksi asettaa hyväksymään HID-paketteja ainoastaan luotetuilta laitteilta (D. Tian ym. 2016) jolloin protokollan maskeeraamishyökkäys epäonnistuu.

Usbfilterin ongelmia ovat kuitenkin luotto laitteen ilmoittamiin tietoihin kuten sarjanumeroon laitetta identifioidessa sekä sen kyvyttömyys tunnistaa Badusb-hyökkäys. Laitteen identiteetin ongelmaa lievittää kuitenkin mahdollisuus huomioida USB-portti johon laite on liitetty, jolloin luotettavaa laitetta imitoidakseen laitteen tulisi olla myös tietyssä portissa kiinni (D. Tian ym. 2016). On huomionarvoista, että USBFilter nojaa vahvasti käyttäjän kykyyn asettaa hyvin määritellyt säännökset suodattimille. Tilanteessa, jossa halutaan esimerkiksi tutkia onko laite luotettava, jotta se voidaan muistaa luotettavana HID-laitteena, ei USBfilter tarjoa suojaa. USBFilter on niin sanotusti deterministinen torjuntakeino, joka huomioi vain tunnettuja hyökkäyksiä eikä esimerkiksi anomalioita (Neuner ym. 2018) kuten protokollan korruptiota hyödyntäviä hyökkäyksiä. Ongelmaksi muodostuvat myös BadUSB hyökkäykset joissa laite hyödyntää sille kuuluvia HID rajapintoja kuten esimerkiksi näppäimistö jonka laiteohjelmistoa on muokattu toteuttamaan näppäininjektiohyökkäys.

5.2.2 Ajureihin perustuvat kontrollit ja fuzzaus

Ajureihin perustuvat kontrollit (engl. ”driver based access controls”) on primitiivi joka pyrkii säätelemään USB ajureiden latausta hyökkäysten torjumiseksi (J. Tian ym. 2018). GoodUSB on yksi tätä primitiiviä hyödyntävä konsepti joka antaa käyttäjälle mahdollisuuden valita ennen enumeraatiota ne rajapinnat joita liitetty USB-laite pääsee käyttämään. Käyttäjän verifioimien rajapintojen jälkeen laitteen tiedot talletetaan tietokantaan, jossa laite on identifioitu sen ilmoittamien tietojen mukaisesti hyväksyttyjen rajapintojen kanssa. Jos laitteen ilmoittamat identifioivat tiedot vastaavat tietokannasta löytyvää laitetta, kysytään käyttäjältä ovatko laitteelle assosioidut rajapinnat ja tiedot oikein. Tapauksessa jossa käyttäjä ei tunnista laitteen tietoja oikeiksi laitteen enumerointi ja kommunikaatio siirretään virtuaalikoneeseen jossa sen kommunikaatio ja tiedot kirjataan hyökkäyksen analysointia varten. (Tian, Bates ja Butler 2015)

Samoin kuin USBFilter on GoodUSB tehokas suoja maskeeraavia BadUSB hyökkäyksiä vastaan mutta tehoton torjumaan laitteen ulkomuotoa vastaavia ajureita hyödyntäviä BadUSB hyökkäyksiä (J. Tian ym. 2018). Kuitenkin poiketen USBFilteristä GoodUSB:n virutaalikonemoduuli antaa mahdollisuuden tutkia laitteen käyttäytymistä ennen kuin sen annetaan toimia (Tian, Bates ja Butler 2015). Tämä kuitenkin vaatii proaktiivisuutta käyttäjän taholta varmistamaan laitteen luotettavuuden ennen sen käyttöönottoa. Tian, Bates ja Butler (2015) huomauttavat, että tapauksissa, joissa laitteen pyytämä rajapinta on laitevalmistajariippuvainen, voi olla epäselvää mitä toimenpiteitä laitteelle ollaan sallimassa jolloin on samoin syytä siirtää laite profiloitavaksi virtuaalikoneeseen ennen sen käyttöönottoa.

USB-ohjelmistojen testaus odottamattomien syötteiden varalta on tapa parantaa käyttöjärjestelmän puolustuskykyä ohjelmistotasolla protokollan korruptiota vastaan. Fuzzaus (engl. “fuzzing”) on tehokas tapa etsiä haavoittuvuuksia syöttämällä satunnaisille mutaatioille altistettuja syötteitä ohjelmistolle (Godefroid, Levin ja Molnar 2008). Jodeit ja Johns (2010) mutatoivat USB-paketteja väliintulotaktiikalla isäntälaitteen USB-ajureiden haavoittuvuuksien löytämiseksi. Peng ja Payer (2020) löysivät kehittämänsä USBFuzz-fuzzaustyökalun avulla 16 uutta vakavaa muistihaavoittuvuutta Linux käyttöjärjestelmästä.

5.2.3 Dynaamiset havaitsemissysteemit

Viimeisin puolustusprimitiiviksi noussut torjuntakeino ovat dynaamiset havaitsemissysteemit (engl. dynamic detection systems"). Deterministisistä tai ihmisten tunnistuskykyyn luottavista systeemeistä poiketen dynaamiset systeemit pyrkivät havaitsemaan ja estämään hyökkäykset niiden tapahtumahetkellä. Uusimmat havaitsemismetodit perustuvat koneoppimiseen, jossa systeemi opetetaan ensin luotetulla USB-kommunikaatiolla, jonka jälkeen sen tehtäväksi jää tunnistaa tavallisesta poikkeavaa toimintaa. (Denney, Babun ja Uluagac 2020)

Kirjallisuudesta löytyvät dynaamiset puolustuskeinot keskittyvät BadUSB:n torjumiseen. Kharraz ym. (2019) esittelevät ohjelmistopohjaisen Usbesafen, joka koostuu kolmesta moduulista. Ensimmäinen moduuli valvoo ja kerää USB-paketteja, toinen moduuli tarkistaa kerättyjen pakettien perusteella onko kommunikaatio normaalia ja kolmas moduuli ilmoittaa käyttäjälle jos yllättävää kommunikaatiota tapahtuu. Kommunikaation poikkeavuutta valvova moduuli tekee päätökset valitun koneoppimisalgoritmin perusteella. Havaitessaan anomalian Usbesafe estää laitteen käyttämän rajapinnan toiminnan ja ilmoittaa tästä käyttäjälle. (Kharraz ym. 2019) Denney, Babun ja Uluagac (2020) ehdottavat samankaltaista ratkaisua, joka perustuu isäntälaitteen käyttöjärjestelmään sijoitetettujen ohjelmistoisten sijasta erilliseen laitteistoon. Usbwatch liitetään esimerkiksi isäntälaitteen USB-väylään jolloin se välikätenä suojaa isäntälaitetta niin kauan kun käyttäjä liittää USB-laitteet vain Usbwatchiin. Sekä USBwatch, että Usbesafe keräsivät tietoaineistoihinsa monen erilaisen USB-laitteen lähettämiä USB-paketteja. Suurimmat erot näiden kahden suojauksen välillä ovat USBwatchin laitepohjaisuus ja Usbwatchin kyky ottaa puolustaa tehokkaammin hyökkäyksiä vastaan joissa laitteelta tulevien syötteiden viive on vaihteleva. Denney, Babun ja Uluagac (2020) testasivat höykkäyksiä myös Usbesafen käyttämää mallia ja parametreja vastaan, jolloin puolustus ei hälyttänyt kun syötteet tulivat 100ms ja 150ms välisellä viiveellä. Usbwatchin esittelemä versio hälytti kaikissa testatuissa tilanteissa. Laitepohjaisen puolustuksen hyödyt tulevat esille kehittyneissä hyökkäyksissä, joissa isäntälaitteen eheys voi olla vaarantunut. Laitteistopohjainen ratkaisu ei myöskään kuluta isäntälaitteen resursseja toisin kuin ohjelmistopohjainen ratkaisu.

Dynaamiset systeemit tarjoavat monia hyötyjä. Puolustussysteemin ei tarvitse luottaa ihmisen harkintakykyyn vaan ne voivat toimia autonomisesti. Niillä ei ole myöskään tarvetta mo-

difioida USB-protokollaa tai USB-laitteiden käyttötapaa. Kharraz ym. (2019) huomauttavat, että treenattavan tietoaineiston määrä ja laatu määrittävät anomalioiden havaitsemistodennäköisyyden. Tietoaineisto voidaan opettaa yhden käyttäjän syötteillä tai monen käyttäjän syötteillä. Kharraz ym. (2019) huomasivat, että Usbsafen tapauksessa monen käyttäjän tietoaineistolla väärien hälytysten osuus nousi 0.21 prosentista 0.93 prosenttiin ja aitojen löytöjen osuus laski 95.7 prosentista 94.9 prosenttiin verrattuna yhdeltä käyttäjältä kerättyyn aineistoon. Denney, Babun ja Uluagac (2020) eivät täsmentäneet oliko testaamisessa käytetty aineisto yhden vai monen käyttäjän tuotos. Uhkina dynaamisten systeemien kohdalla ovat esimerkiksi käyttäjän kirjoitustavan matkiminen (Kharraz ym. 2019) ja erilaiset tavat huijata koneoppimismallia (Denney, Babun ja Uluagac 2020).

5.3 Fyysinen kerros

Fyysisen kerroksen puolustusmenetelmiä harkitsevaa kirjallisuutta on varsin vähän (J. Tian ym. 2018). Salakuuntelua vastaan kirjallisuudessa on esitetty tiedonsiirron salausta ja sormenjäljittämistä vastaan isäntälaitteen USB-kommunikaatiota satunnaistavia metodeja. Signaali-injektioille ei tunneta puolustuskeinoja (J. Tian ym. 2018).

Sormenjäljittämistä vastaan voidaan puolustautua varmistamalla, ettei USB-kommunikaatiosta voida päätellä käyttöjärjestelmille ominaisia piirteitä. Pakettien ajoittamista voidaan yrittää hämätä esimerkiksi nopeuttamalla tai hidastamalla laitteen ja isännän välistä tiedonsiirtoa. On myös mahdollista torjua paketteihin perustuvaa identifikointia muokkaamalla enumeraation aikana lähetettävien pakettien lähetysjärjestystä. (J. Tian ym. 2018)

Salakuuntelua vastaan on ehdotettu USB-tiedonsiirroon salaamista laitteen ja isäntälaitteen välillä. Angel ym. (2016) näyttivät, että on mahdollista salata tietoliikenne isäntälaitteen ja USB-laitteen välillä hyödyntäen tietoliikennettä salaavaa adapteria ja isäntälaitteelle asennettavaa Cinch:iä. Neugschwandtner, Beitler ja Kurmus (2016) pyrkivät salaamaan tietoliikenteen USB-laitteiden välillä Uscramble-ohjelmiston avulla neuvottelemalla salausavaimen liitetyn USB-laitteen kanssa (J. Tian ym. 2018). On kuitenkin epäselvää kuinka moni USB-laite tukee salausta. Salaus laitteiden välillä on tehokas keino torjumaan USB-väylällä urkkivia USB laitteita, jotka ovat liitetty USB 2.0 spesifikaatiota noudattaviin hubeihin. Lan-

gattomien USB laitteiden langatonta tiedonsiirtoa urkkivia tahoja vastaan esitellyt keinot eivät kuitenkaan auta sillä ne voivat salata tiedonliikenteen vain fyysisesti liitettyjen laitteiden välillä. Näissä tapauksissa on luotettava, että USB-laitteiden valmistaja on impelementoinut riittävät salaus- ja autentikaatioprotokollat laitteen ja vastaanottimen välillä.

6 Yhteenveto

USB:n kautta tehtävien tunnettujen hyökkäystapojen määrä on varsin laaja. Myös kirjallisuudesta löytyvien yksittäisten puolustuskeinojen määrä on laaja mutta ne ovat osittaisia ja rajoittuvat usein vain yhden hyökkäyksen torjumiseen, jolloin laajempaan torjuntaan useamman puolustuksen käyttöönotto rasittaa loppukäyttäjää. Yksi tulevisuuden tutkimusaiheista voisikin olla useamman puolustusprimitiivin yhdistäminen yhdeksi systeemiksi laajan puolustusksen aikaansaamiseksi. Uusimapana tulokkaana dynaamiset systeemit näyttäytyvät melko lupaavina keinoina badusb-hyökkäyksiä vastaan ja koneoppimismallien kehitys todennäköisesti parantaa tulevien puolustuskeinojen vahvuutta.

Väittäisin että moni hyökkäys voitaisiin torjua vahventamalla autentikaatiota isäntälaitteen ja laitteen välillä protokollan tasolla, jolloin vastuu käyttäjältä siirtyisi USB-standardin kehittäjille. Tämä kuitenkin vaatii myös laitteiden valmistajien tuen, joka voi herättää vastustusta uusien protokollien adoptiossa jos autentikaatio olisi pakollinen. Ongelmaksi muodostuvat myös USB:n tuki aiemmille protokollille, joiden kielto aiheuttaisi suunnattoman määrän käyttökelvottomia laitteita. Siirtymisvaihe turvallisempien laitteiden aikakaudelle ei kuitenkaan tarvitse tapahtua hetkessä ja EU:n puuttuminen tietoturvaa kohtaan lainsäädännön keinoin voisi mahdollistaa myös USB-laitteiden ekosysteemin muutoksen tulevaisuudessa.

Lähteet

Angel, Sebastian, Riad S. Wahby, Max Howald, Joshua B. Leners, Michael Spilo, Zhen Sun, Andrew J. Blumberg ja Michael Walfish. 2016. “Defending against malicious peripherals with Cinch”. Teoksessa *25th USENIX Security Symposium (USENIX Security 16),* 397–414. Viitattu 30. joulukuuta 2024. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/angel.

“Beyond Autorun: Exploiting vulnerabilities with removable storage”. 2011. Viitattu 13. kesäkuuta 2024. https://media.blackhat.com/bh-dc-11/Larimer/BlackHat_DC_2011_Larimer_Vulnerabiliters_w-removeable_storage-wp.pdf.

Brandao (PhD), Pedro ja Rohan Scanavez (Degree). 2021. “Bad USB: why must we discuss this threat in companies?” Number: 3, *Research Review* 2, numero 3 (4. joulukuuta 2021): 561–567. ISSN: 2693-5007, viitattu 21. helmikuuta 2024. https://researchreview.in/index.php/rr/article/view/65.

“Code Injection”. n.d. Viitattu 13. syyskuuta 2024. https://owasp.org/www-community/attacks/Code_Injection.

Corbet, Jonathan, Alessandro Rubini ja Greg Kroah-Hartman. 2005. *Linux Device Drivers, 3rd Edition.* O’Reilly Media. ISBN: 9780596005900; 0596005903.

“Defined USB Class Codes”. 2023. Viitattu 10. maaliskuuta 2024. https://www.usb.org/defined-class-codes.

Denney, Kyle, Leonardo Babun ja A. Selcuk Uluagac. 2020. “USB-Watch: a Generalized Hardware-Assisted Insider Threat Detection Framework”. 7 citations (Crossref) [2024-02-21], *Journal of Hardware and Systems Security* 4, numero 2 (1. kesäkuuta 2020): 136–149. ISSN: 2509-3436, viitattu 21. helmikuuta 2024. https://doi.org/10.1007/s41635-020-00092-z. https://doi.org/10.1007/s41635-020-00092-z.

Godefroid, Patrice, Michael Y. Levin ja David A. Molnar. 2008. “Automated whitebox fuzz testing.” Teoksessa *NDSS,* 8:151–166. Viitattu 30. joulukuuta 2024. http://pxzhang.cn/paper/concolic_testing/FuzzTesting.pdf.

Griscioli, Federico ja Maurizio Pizzonia. 2021. “USBCaptchaIn: Preventing (un)conventional attacks from promiscuously used USB devices in industrial control systems”. Publisher: IOS Press, *Journal of Computer Security* 29, numero 1 (1. tammikuuta 2021): 51–76. ISSN: 0926-227X, viitattu 7. maaliskuuta 2024. https://doi.org/10.3233/JCS-191404.

Jodeit, Moritz ja Martin Johns. 2010. “USB Device Drivers: A Stepping Stone into Your Kernel”. Teoksessa *2010 European Conference on Computer Network Defense,* 46–52. 10 citations (Crossref) [2024-02-21], 2010 European Conference on Computer Network Defense. Lokakuu. Viitattu 21. helmikuuta 2024. https://doi.org/10.1109/EC2ND.2010.16.

Kang, M. ja H. Saiedian. 2017. “USBWall: A novel security mechanism to protect against maliciously reprogrammed USB devices”. 7 citations (Crossref) [2024-02-21], *Information Security Journal* 26 (4): 166–185. ISSN: 1939-3555. https://doi.org/10.1080/19393555.2017.1329461.

Kharraz, A., B.L. Daley, G.Z. Baker, W. Robertson ja E. Kirda. 2019. “USBESAFE: An end-point solution to protect against USB-based attacks”, 89–103. RAID 2019 Proceedings - 22nd International Symposium on Research in Attacks, Intrusions and Defenses. ISBN: 978-1-939133-07-6.

Lu, Hongyi, Yechang Wu, Shuqing Li, You Lin, Chaozu Zhang ja Fengwei Zhang. 2021. “BADUSB-C: Revisiting BadUSB with Type-C”. Teoksessa *2021 IEEE Security and Privacy Workshops (SPW),* 327–338. 2021 IEEE Security and Privacy Workshops (SPW). Toukokuu. Viitattu 7. maaliskuuta 2024. https://doi.org/10.1109/SPW53761.2021.00053.

Madieu, John. 2017. *Linux Device Drivers Development.* Packt Publishing. ISBN: 9781785280009; 1785280007; 9781782174752; 1782174753.

“MouseJack Technical Details”. 2016. Viitattu 12. huhtikuuta 2024. https://www.bastille.net/research/vulnerabilities/mousejack/technical-details.

Murphy, Robert ja Associated Part Family. 2014. “USB 101: An introduction to universal serial bus 2.0”. *no* 1:26.

Neugschwandtner, Matthias, Anton Beitler ja Anil Kurmus. 2016. “A transparent defense against USB eavesdropping attacks”. Teoksessa *Proceedings of the 9th European Workshop on System Security*. EuroSec ’16. London, United Kingdom: Association for Computing Machinery. ISBN: 9781450342957. https://doi.org/10.1145/2905760.2905765.

Neuner, Sebastian, Artemios G. Voyiatzis, Spiros Fotopoulos, Collin Mulliner ja Edgar R. Weippl. 2018. “USBlock: Blocking USB-Based Keypress Injection Attacks”. Teoksessa *Data and Applications Security and Privacy XXXII,* toimittanut Florian Kerschbaum ja Stefano Paraboschi, 278–295. Lecture Notes in Computer Science. Cham: Springer International Publishing. ISBN: 978-3-319-95729-6. https://doi.org/10.1007/978-3-319-95729-6_18.

Nissim, N., R. Yahalom ja Y. Elovici. 2017. “USB-based attacks”. 48 citations (Crossref) [2024-02-21], *Computers and Security* 70:675–688. ISSN: 0167-4048. https://doi.org/10.1016/j.cose.2017.08.002.

Peng, Hui ja Mathias Payer. 2020. “USBFuzz: A Framework for Fuzzing USB Drivers by Device Emulation”.

Sun, Chengzhi, Jiyu Lu ja Yunqing Liu. 2021. “Analysis and Prevention of Information Security of USB”. Teoksessa *2021 International Conference on Electronic Information Engineering and Computer Science (EIECS),* 25–32. 2 citations (Crossref) [2024-02-21], 2021 International Conference on Electronic Information Engineering and Computer Science (EIECS). Syyskuu. Viitattu 21. helmikuuta 2024. https://doi.org/10.1109/EIECS53707.2021.9588135.

Tian, D., A. Bates ja K. Butler. 2015. “Defending against malicious USB firmware with GoodUSB”, nide 7-11-December-2015, 261–270. 34 citations (Crossref) [2024-02-21], ACM International Conference Proceeding Series. ISBN: 978-1-4503-3682-6. https://doi.org/10.1145/2818000.2818040.

Tian, Dave, Nolen Scaife, Adam Bates, Kevin Butler ja Patrick Traynor. 2016. “Making USB Great Again with USBFILTER”. Teoksessa *25th USENIX Security Symposium (USENIX Security 16),* 415–430. Austin, TX: USENIX Association, elokuu. ISBN: 978-1-931971-32-4. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/tian.

Tian, Jing, Nolen Scaife, Deepak Kumar, Michael Bailey, Adam Bates ja Kevin Butler. 2018. “SoK: "Plug & Pray"Today – Understanding USB Insecurity in Versions 1 Through C”. Teoksessa *2018 IEEE Symposium on Security and Privacy (SP),* 1032–1047. ISSN: 2375-1207, 2018 IEEE Symposium on Security and Privacy (SP). Toukokuu. Viitattu 7. maaliskuuta 2024. https://doi.org/10.1109/SP.2018.00037.

“USB Device Enumeration”. 2009. Viitattu 28. maaliskuuta 2024. https://ftdichip.com/wp-content/uploads/2020/08/TN_113_Simplified-Description-of-USB-Device-Enumeration.pdf.

“usbkill.com”. 2023. Viitattu 12. huhtikuuta 2024. https://usbkill.com/products/usbkill-v4.

Al-Zarouni, Marwan. 2006. “The Reality of Risks from Consented use of USB Devices”. Medium: PDF Publisher: [object Object], *Proceedings of 4th Australian Information Security Management Conference* Edith Cowan University:2006. Viitattu 4. maaliskuuta 2024. https://doi.org/10.4225/75/57B6543434762.