Janne Niinisaari

DATADIODI JA SEN SOVELLUTUKSET

Opinnäytetyö

Tekniikan ammattikorkeakoulututkinto

Tieto- ja viestintätekniikan koulutus

2022

Kaakkois-Suomen
ammattikorkeakoulu

Kaakkois-Suomen ammattikorkeakoulu

Tutkintonimike	Insinööri (AMK)
Tekijä/Tekijät	Janne Niinisaari
Työn nimi	Datadiodi ja sen sovellutukset
Toimeksiantaja	Xamk
Vuosi	2022
Sivut	40 sivua
Työn ohjaaja(t)	Vesa Kankare

TIIVISTELMÄ

Tämän opinnäytetyön tarkoituksena oli tutustua yksisuuntaisiin yhdyskäytäviin eli datadiodeihin ja niiden käyttökohteisiin, sekä rakentaa omatekemä datadiodi Kaakkois-Suomen ammattikorkeakoulun Kotkan kampuksen Kyberturvallisuuslaboratorioon opetuskäyttöön. Tavoitteena oli luoda mahdollisimman edullinen ja yksinkertainen toteutus datadiodista ja testata sen käytettävyyttä muutamilla erilaisilla tiedonsiirtomenetelmillä.

Opinnäytetyön teoriaosassa käydään läpi erilaisia tapoja toteuttaa datadiodi ja mitä asioita täytyy ottaa huomioon yksisuuntaisen yhteyden luomisessa ja käyttämisessä. Teoriaosassa esitellään myös yleisimpiä käyttökohteita datadiodille sekä työssä testattavat tiedonsiirtomenetelmät.

Työn tutkimusmenetelmänä on konstruktiivinen tutkimus ja pääpaino työssä on datadiodin rakentamisessa. Toteutusvaihe suoritettiin syklimäisesti, jolloin jokaisen datadiodin toteutusmallin jälkeen tulosta analysoitiin ja kehitettiin ja aloitettiin uuden toteutuksen suunnittelu. Toteutusvaiheen aineisto kerättiin havaintojen ja testausten pohjalta.

Lopputuloksena luotiin hyvin pelkistetty ja edullinen datadiodi, jonka hinta-arviota ja käytettävyyttä verrattiin kaupallisiin tuotteisiin muutaman lähteen pohjalta. Opinnäytetyö toimii myös oppaana omatekoisen datadiodin rakentamiseen ja käyttämiseen muutamilla yleisillä tiedonsiirtomenetelmillä. Työn edetessä syntyi myös muutamia jatkokehitysideoita toimeksiantajalle.

Asiasanat: data, diodit, kyberturvallisuus, optiset kuidut

South-Eastern Finland University of Applied Sciences

Degree title	Bachelor of Engineering
Author (authors)	Janne Niinisaari
Thesis title	Data diode and its applications
Commissioned by	Xamk
Time	2022
Pages	40 pages
Supervisor	Vesa Kankare

ABSTRACT

The objective of the thesis was to study unidirectional gateways, i.e., data diodes, and their applications and to build a self-made data diode for educational use in the Cyber Security Laboratory of the Kotka campus of Southeastern Finland University of Applied Sciences. The goal was to create the most affordable and simple implementation of a data diode and test its usability for data transfer with five different methods.

The theory part of the thesis reviews different ways to implement a data diode and issues to be taken into account when creating and using a unidirectional connection. The theory part also introduces the most common uses for a data diode and the methods for data transfer tested in the thesis.

The research method of the thesis is constructive research, and the focus is on the construction of a data diode. The implementation phase was carried out cyclically, whereafter each data diode implementation model the result was analyzed and developed, and the planning for a new implementation was started. The material for the implementation phase was collected based on observation and test results.

The result was a very simplified and inexpensive data diode, whose price estimate and usability were compared to commercial products based in a few different sources. The thesis also serves as a guide for building and using a homemade data diode with a few common methods for data transfer. As the work progressed, some further development ideas for the client also emerged. These ideas mostly focus on using data diode in virtual environments and finding a way to monitor the traffic going through a unidirectional connection

Keywords: data, diodes, cyber security, optical fibres

SISÄLLYS

1 JOHDANTO

Jatkuvasti huimaa vauhtia kehittyvässä verkostoituvassa maailmassa tietoverkkojen turvaaminen kasvaa koko ajan merkittävämmäksi. Erityisesti kriittisen infrastruktuurin vahva suojaaminen kyberhyökkäyksiltä on erityisen tärkeää yhteiskunnan sujuvan toiminnan kannalta. Yleisesti parhaiksi ratkaisuiksi näiden kriittisten verkkojen turvaamiseen on todettu yhteyksien yksinkertaistaminen, vähentäminen ja eristäminen. Verkkojen täysi eristäminen toisistaan kuitenkin vaikeuttaa merkittävästi arvokkaiden tietojen saatavuutta ja täten perinteiset turvallisuuden haasteet ovat olleetkin pääsyn rajaaminen ja riskien minimointi kuitenkin säilyttämällä tietojen saatavuus. (Owl Cyber Defence Solutions 2018, 1.)

Yksisuuntaiset yhdyskäytävät eli datadiodit kehitettiin ratkaisemaan edellä mainitut ongelmat tarjoamalla vahvistettua verkon puolustusta samalla sallien tiedon turvallisen jakamisen. Datadiodin perimmäinen idea on luoda fyysisesti yksisuuntainen yhteys kahden verkon välille käyttäen yleensä yhtä valokuitua, jolloin tietoliikenteen on mahdoton kulkea vastakkaiseen suuntaan. Näin voidaan eristää kriittisiä verkkoja, kuten voimalaitoksia estämällä pääsy sisään, mutta kuitenkin mahdollistamalla esimerkiksi sensoridatan tai lokitietojen jakamisen alemman turvallisuusluokan verkkoon. (Owl Cyber Defence Solutions 2018, 1.)

Esimerkiksi vuonna 2015 Ukrainan voimaverkkoon tehdyssä hyökkäyksessä hakkerit onnistuivat tunkeutumaan yritysverkkoihin, joiden kautta he pääsivät käsiksi työntekijöiden tunnuksiin. Näillä tunnuksilla he saivat etäyhteyden voimaverkon SCADA-verkkoon (Supervisory control and data acquisition eli valvonta ja tiedonkeruu), josta he pystyivät sammuttamaan voimalaitoksia ympäri Ukrainaa. (Zetter 2016.) Tässä on hyvä esimerkki SCADA-verkkojen turvaamisen merkityksestä, ja datadiodit toimivat hyvänä ratkaisuna niiden eristämiseen ulkopuolisilta uhkilta.

Datadiodeja on monia erilaisia ja monesti ne suunnitellaan ja konfiguroidaan käyttökohtaisesti. Kaikkia kuitenkin yhdistää se, että datadiodin toiminta perustuu laitteiston ja ohjelmistojen yhdistettyyn kokonaisuuteen. Valtaosa tietoliikenteessä käytetyistä protokollista vaatii kaksisuuntaista kommunikaatiota

yhteyden muodostamiseksi, joten yksisuuntainen yhteys vaatii päätelaitteilta ohjelmistotasoisia ratkaisuja protokollien muuttamiseen. (Kertysova, Frinking & Gricius 2019, 13.)

Tässä työssä esitellään yleisimpiä tapoja datadiodin toteuttamiselle ja rakennetaan muutama erilainen datadiodi käyttämällä Kaakkois-Suomen ammattikorkeakoulun Kotkan kampukselta löytyvää laitteistoa sekä niin sanottuja hyllystä saatavia osia. Tavoitteena saada datadiodi välittämään liikennettä mahdollisimman monella eri tavalla, kuten tiedostonsiirto, ruudunjako, videon suoratoisto, aikatietojen jakaminen ja lokipalvelin. Työssä tullaan myös testaamaan, kuinka suuriin tiedonsiirtonopeuksiin kyetään ilman, että tiedonvälityksen varmuus kärsii.

2 TUTKIMUSASETELMA

Lähtökohtana tälle työlle on se, ettei Kaakkois-Suomen ammattikorkeakoulun Kotkan kampuksen Kyberturvallisuuslaboratoriossa vielä ole datadiodia opetuskäytössä. Työn tarkoituksena on perehtyä erilaisten datadiodien toimintaperiaatteisiin ja rakentaa yksi tai useampi datadiodi käyttämällä edullisia komponentteja sekä koululta jo löytyviä laitteita. Työssä pyritään vastaamaan seuraaviin kysymyksiin:

- Mikä on datadiodi?
- Mihin datadiodia voidaan hyödyntää?
- Miten datadiodi toteutetaan omatekoisesti mahdollisimman edullisesti ja tehokkaasti?

Työ suoritetaan konstruktiivisena tutkimuksena, joka tutkimusotteena voidaan ajatella laadullisten tutkimusten yhdistelmänä. Konstruktiivisen tutkimuksen tavoitteena on etsiä ratkaisua ongelmaan ja poistamaan ongelma. Ongelma ratkaistaan konstruktiolla, joka tässä työssä on laitteiston kasaaminen ja konfigurointi. Oleellista konstruktiiviselle tutkimukselle on, että tutkija on itse mukana prosessissa ja tutkimus perustuu aikaisempaan teoriaan aiheesta. (Kananen 2017, 14–16.)

Aineisto kerätään pääosin sekundääriaineistona, joka muodostuu verkkodokumenteista, aiemmista tutkimuksista, white papereista ja raporteista. Tutustutaan datadiodeihin käsitteenä ja etsitään aiempia tutkimuksia omatekemistä

datadiodeista. Primääriaineisto muodostuu havainnoista tehdyistä muistiinpanoista sekä testituloksista. Aineistonkeruumenetelmien valinta perustuu valittuun tutkimusotteeseen ja siihen, että opinnäytetyö pohjautuu aiempaan teoriapohjaan ja tutkittuun tietoon.

Työ suoritetaan syklimäisesti eli suunnitellaan, toteutetaan ja havainnoidaan interventiota ja mikäli haluttuun muutokseen ei päästä, aloitetaan sykli alusta (Kananen 2017, 34). Näiden kehittämissyklien pohjalta tuotoksia kehitetään ja arvioidaan jatkuvasti, jolloin syklien edetessä lähestytään kehittämistutkimukselle asetettuja tavoitteita ja mahdollisesti nostetaan esiin uusia tavoitteita (Pernaa 2013). Toteutusvaiheessa tehdään muistiinpanoja kaikista vaiheista. Lopputuloksena luodaan mahdollisimman kattava ja yleistettävä kuvaus datadiodin kasaamisesta ja käyttöönotosta. Toteutusvaiheen lopulla tehdään vielä testejä datadiodin suorituskyvystä eri sovellutuksissa ja verrataan tuloksia aiempiin tutkimuksiin ja kaupallisten versioiden ilmoittamiin tietoihin.

Tutkimuksen luotettavuutta arvioidaan neljän perinteisen laadullisen tutkimuksen kriteerin avulla, jotka ovat uskottavuus, siirrettävyys, luotettavuus ja vahvistettavuus (Pernaa 2013). Nämä kriteerit pyritään täyttämään luomalla kattava dokumentointi suoritetuista kehittämissykleistä ja osoittamalla lopputulos toimivaksi ja onnistuneeksi. Työssä testataan toteutusta eri sovellutuksissa ja luodaan mahdollisimman yleispätevä tuotos, joka on siirrettävissä muihin ympäristöihin.

3 DATADIODI

Yleisesti datadiodi määritellään koostuvan kahdesta komponentista, joista toinen vain lähettää ja toinen vastaanottaa dataa. Näin ollen datadiodi fyysisesti rajoittaa tiedonsiirron vain yhteen suuntaan. Datadiodin tarkoitus on eristää verkkoja toisistaan kuitenkin samalla takaamalla saatavuuden toiseen suuntaan. (Owl Cyber Defence Solutions 2018, 3.)

Pelkkä yksisuuntainen linkki ei kuitenkaan vielä riitä datadiodin toimivuuteen, sillä hyvin monet tietoliikenneprotokollat vaativat kaksisuuntaista kommunikaatiota yhteyden muodostamiseksi. Ohjelmiston liittäminen kokonaisuuteen syn-

nyttääkin suurimmat erot erilaisten datadiodien välillä. Toiset integroivat ohjelmiston suoraan laitteistoon esimerkiksi virtapiiriin, kun taas toiset selvästi erottavat ohjelmiston laitteistosta. Monesti käytetään erillisiä palvelimia yhteyden molemmissa päissä, jotka sallivat kommunikaation vastaavien verkkojen välillä. (Kertysova, Frinking & Gricius 2019, 13.)

3.1 Protokollat

TCP (Transmission Control Protocol) on OSI-mallin kerroksen neljä protokolla ja yleisin IP:n (Internet Protocol) päällä käytetty protokolla tietokoneiden väliseen viestintään. TCP on kuitenkin yhteyteen perustuva protokolla ja se vaatii kaksisuuntaista kommunikaatiota yhteyden muodostamiseksi. (RFC 793: 1981.) Jotta TCP:n kaltaiset yhteyteen perustuvat protokollat voidaan ohjata datadiodin läpi, tarvitaan ohjelmistoja muuttamaan kaksisuuntaiset protokollat yksisuuntaisiksi. (Kertysova, Frinking & Gricius 2019, 13.)

UDP (User Datagram Protocol) on toinen OSI-mallin kerroksen neljä protokolla. Verrattuna TCP:hen, UDP on yhteydetön protokolla eli se ei vaadi jatkuvaa yhteyden muodostusta ja ylläpitoa kommunikoivien laitteiden välillä. (RFC 768: 1980.) Tämän takia UDP on luonnostaan sopiva protokolla yksisuuntaisiin yhteyksiin, vaikkakin monet UDP:ta hyödyntävät protokollat vaativat jonkin tasoista kaksisuuntaista viestintää. UDP:n suurimpana heikkoutena tiedonsiirron luotettavuuden kannalta on, että yksittäiset paketit eivät vaadi vahvistusta vastaanottajalta, mikä datadiodissa ei ole edes mahdollista, joten tiedonsiirto on jokseenkin epäluotettavaa. (Menoher & Mraz 2009, 4.)

3.2 Yksisuuntaiset tiedonsiirtoteknologiat

Owl Cyber Defense Solutions, LLC (2018, 13) luettelee e-kirjassaan neljä erilaista teknologiaa, jotka ovat kehitetty vuosien saatossa yksisuuntaiseen tiedonsiirtoon. Tässä luvussa tutustutaan näihin neljään teknologiaan ja perehdytään tarkemmin niiden rakenteeseen, ominaisuuksiin ja suorituskykyihin. Edellä mainittujen neljän teknologian lisäksi Menoher ja Mraz (2009, 6) mainitsevat kaksi muuta menetelmää (niin sanotun lenkkariverkon eli tiedonsiirron ulkoisilla talletuslaitteilla, sekä monimutkaiset ohjelmistot, kuten käyttöjärjes-

telmien politiikkasäännöt), mutta nämä jätetään tarkastelematta niiden poikkeavan luonteensa vuoksi. Mainittakoon vielä Kangin ym. (1996) esittelemä niin sanottu verkkopumppu tai datapumppu, joka toimii yksisuuntaisena tiedonsiirtosysteeminä, mutta sallii vahvistusviestien vastaanottamisen eikä fyysisesti estä tiedon kulkua toiseen suuntaan.

3.2.1 Kaapelin muokkaaminen yksisuuntaiseksi

Yksinkertaisimmillaan datadiodin voi tehdä muokkaamalla kaksisuuntaista kaapelia irrottamalla tai leikkaamalla toisen suunnan yhteyden pois. Historiallisesti tämä toteutettiin käyttämällä RS-232-sarjakaapelia leikkaamalla paluuliikenteeseen tarkoitettu johto poikki (Menoher & Mraz 2009, 7). Yleisempi tapa on käyttää yhtä valokuitua, jolloin tiedonkulku on erehtymättömästi yksisuuntaista.

Näin toteutettuna yhteyden luominen on hyvin ongelmallista, sillä valtaosa liikenteestä vaatii jonkinlaisia vahvistusviestejä vastakkaiseen suuntaan. Kaksisuuntaiset protokollat rikkoutuvat ja tarvitaan vähintäänkin uudelleenlähetyksiä, jos ylipäätään dataa saadaan liikkumaan. (Owl Cyber Defence Solutions, LLC 2018, 17.)

3.2.2 Palomuurit

Poikkeuksena muihin tässä luvussa listattaviin teknologioihin, palomuurit toimivat ohjelmistotasolla eivätkä fyysisesti estä tiedon kulkua toiseen suuntaan. Ideana palomuurien käytössä yksisuuntaisena tiedonsiirtotapana on käyttää yhtä tai useampaa palomuuria, jotka sallivat liikenteen vain yhteen suuntaan. Palomuurit tarjoavat korkeaa suoritustehoa ja sallivat monenlaisen tiedonsiirron samanaikaisesti, mutta ohjelmistopohjaisen luonteen ja monimutkaisten konfiguraatioiden vuoksi ne ovat melko haavoittuvaisia monille hyökkäyksille. Yksikin virhe tai muutos asetuksissa voi avata suoran reitin hyökkäyksille. (Owl Cyber Defence Solutions, LLC 2018, 14, 16, 20.)

3.2.3 Yksisuuntainen yhdyskäytävä

Yksisuuntainen yhdyskäytävä on laitteistokokonaisuus, joka koostuu lähettävästä sekä vastaanottavasta laitteesta ja niiden lisäksi kahdesta välityspalvelimesta, yksi kummassakin päässä. Nämä välityspalvelimet toimivat yhdyskäytävinä omille verkoilleen ja vastaavat kaksisuuntaisten protokollien muuttamisesta yksisuuntaiseksi ja päinvastoin, jotta yksisuuntaisessa linkissä tapahtuva liikenne on mahdollista. (Owl Cyber Defence Solutions, LLC 2018, 14.) Tämä on yleisin toteutustapa, jolla kotitekoiset datadiodit rakennetaan, ja monesti käsitteillä datadiodi, yksisuuntainen yhdyskäytävä ja yksisuuntainen verkko viitataan tämänkaltaiseen toteutukseen.

Suurin merkitys toteutuksessa on välityspalvelimilla, sillä niiden konfiguraatioiden varassa on käytännössä kaikki yhteyden toimivuuteen liittyvä. Kotitekoisen datadiodin rakentamisen kannalta välityspalvelimet aiheuttavat suurimmat ongelmat. Avoimen lähdekoodin ratkaisuja on vielä melko vähän ja käytettävät protokollat jäävät myös vähäisiksi. (Maatkamp 2015, 4.)

3.2.4 Älykäs datadiodi

Owl Cyber Defence Solutions, LLC (2018, 15) käyttää e-kirjassaan termiä älykäs datadiodi (Intelligent Data Diode) yksittäiselle laitteelle, johon on sisällytetty kaikki edellisessä luvussa esitellyt ominaisuudet ja laitteistot. Näiden lisäksi älykäs datadiodi voi käyttää reitintöntä yhden suunnan protokollaa ja on fundamentaalisesti suunniteltu takaamaan korkea luotettavuus sekä matala viive yksisuuntaiselle tiedonsiirrolle. Älykäs datadiodi on yleensä koottu yhteen suljettuun laitteeseen suojatakseen kytkentöjä peukaloinnilta sekä pienentääkseen tilaa, painoa ja virrankulutusta.

Tämä on toteutustapa, jolla kaupalliset yritykset valmistavat tuotteensa. Markkinoilta löytyy paljon erilaisia laitteita, joilla kaikilla on hieman eri ominaisuuksia, kuten tiedonvälitysprotokollat, tiedonsiirtonopeus, liitännät ja yhteensopivuudet käytettyihin järjestelmiin sekä mahdollisiin lisälaitteisiin. Esimerkiksi OPSWAT tarjoaa laitetta nimeltään MetaDefender Kiosk, joka tarkastaa kaiken datadiodiin siirrettävän datan haittaohjelmien varalta (OPSWAT 2022b).

Owl Cyber Defence Solutions, LLC (2022) käyttää datadiodeissaan tiedonsiirtoon ATM:ään (Asynchronous Transfer Mode) perustuvaa protokollaa. ATM pilkkoo datan kiinteiksi 53:n tavun paketeiksi ja samalla poistaa kaiken reititykseen liittyvän informaation. Tämä lisää tietoturvaa sekä tarjoaa korkeaa kaistaa ja matalaa viivettä.

3.3 Datadiodin käyttökohteet

Toimintaperiaatteensa vuoksi datadiodit ovat ideaaleja eristämään eri turvallisuusluokan tietoverkkoja toisistaan samalla sallien tiedonkulun toiseen haluttuun suuntaan. Valtiollisissa verkoissa, asevoimissa, teollisissa ohjausjärjestelmissä ja kriittisessä infrastruktuurissa datadiodit korvaavat paljon verkon kohtia, jotka on ennen jouduttu eristämään täysin niin sanotulla ilmaraolla (air gap) (Scott 2015, 27).

3.3.1 Pääsynhallintamallit

Datadiodien yhteydessä törmää usein kahteen tietoverkkojen pääsynhallinnan malliin, jotka ovat Bell-LaPadula-malli ja Biba-malli. Lyhyesti Bell-LaPadula-mallin periaate on, että eri turvallisuusluokkien välillä pätee sääntö ”no read up, no write down” eli käyttäjä ei pysty lukemaan itseään korkeamman turvallisuusluokan sisältöä eikä alentamaan oman turvallisuusluokkansa informaatiota. (Bell & LaPadula 1973.) Biba-malli on käytännössä tämän vastakohta eli vastaavasti eri turvallisuusluokkien välillä pätee sääntö ”no read down, no write up” (Biba 1975).

Bell-LaPadula-malli keskittyy tiedon luottamuksellisuuteen ja saatavuuteen. Datadiodin kannalta malli kuvaa tilannetta, jossa tiedon kulku sallitaan vain alemman turvallisuusluokan verkosta ylempään, jolloin pääsy salassa pidettäviin tietoihin on estetty. (Menoher & Mraz 2009, 5.) Tähän tiedon kulun suuntaan datadiodeja on käytetty jo vuosikymmeniä korkean turvallisuusluokan ympäristöissä, kuten sotilaskäytössä ja tiedustelupalveluissa (OPSWAT 2022a). Muita yleisiä tiedonsiirtomenetelmiä alemman turvallisuusluokan verkosta ylempään ovat sähköpostin välittäminen, tiedostojen ja päivitysten jakaminen, verkkosivujen peilaaminen ja videon tai äänen suoratoisto (Petersen 2016.)

Biba-malli puolestaan keskittyy tiedon eheyden säilyttämiseen ja vastaavasti kuvaa tilannetta, jossa sallitaan tiedon kulku vain ylemmän turvallisuusluokan verkosta alempaan. Näin ollen tietoa voidaan lähettää suojatusta verkosta ulos samalla eristäen verkko ulkopuolisilta yhteyksiltä. (Menoher & Mraz 2009, 5.) Merkittävä kyberhyökkäysten kasvu viime vuosina on nostattanut datadiodien mielenkiintoa kriittisen infrastruktuurin parissa, kuten sähkön ja veden jakelussa, voimalaitoksissa, öljynjalostamoissa sekä myös perinteisimmissä IT-ympäristöissä kuten datakeskuksissa (Ribeiro 2022). Myös äänestysjärjestelmiin on suunniteltu datadiodin hyödyntämistä tulosten koskemattomuuden säilyttämiseksi (Jones & Bowersox 2016).

3.3.2 Tiedonsiirtomenetelmät

Lähtökohtaisesti on oleellista, että datadiodin läpi pystytään siirtämään tiedostoja. Tässä työssä käytettiin tiedonsiirtoon UDPcast-nimistä työkalua. UDPcastin voi konfiguroida toimimaan ilman vahvistusviestejä vastaanottajalta ja se sisältää ominaisuuksia, kuten FEC (Forward Error Correction) sekä lähetysnopeuden rajoittamisen. Täten UDPcast on ideaali työkalu tiedonsiirtämiseen datadiodin läpi.

Ruudunjako ja videon suoratoisto ovat hyödyllisiä ominaisuuksia, kun halutaan jakaa kuvaa esimerkiksi jostain teollisuuslaitteen monitorista tai videokamerasta, ja samalla estää tarpeeton pääsy laitteille verkkoyhteyden kautta. Videon suoratoisto toteutettiin käyttämällä VLC media playeriä ja FFmpeg-ohjelmaa. Molemmilla näistä voi myös jakaa ruutua suoratoistamalla. Tässä työssä oli tavoitteena testata näitä ratkaisuja ja verrata niiden soveltuvuutta tarkoitukseen.

Lokitietojen kerääminen on oleellista systeemien sujuvan toiminnan varmistamiseksi sekä vianetsinnässä mahdollisten ongelmien sattuessa. Datadiodin avulla seurattavan laitteen lokitiedot voidaan turvallisesti siirtää halutulle lokipalvelimelle vaarantamatta itse laitetta kyberhyökkäyksiltä. Syslog-tiedot voidaan lähettää käyttämällä Rsyslog-ohjelmaa UDP-yhteydellä. Muiden lokitietojen siirto voidaan toteuttaa kopioimalla tiedosto tai kansio datadiodin yli UDPcastilla ja automatisoimalla lähetys tapahtumaan tietyin väliajoin.

NTP (Network Time Protocol) on protokolla, jolla jaetaan täsmällistä aikatietoa verkkolaitteiden välillä. NTP on UDP-pohjainen protokolla ja pienemmissä verkkokokonaisuuksissa aikatietoja voidaan jakaa myös multicast- ja broadcast-paketeilla. Erityisesti broadcastilla vastaanottaja voidaan konfiguroida kuuntelemaan passiivisesti broadcast-paketteja ilman mitään vahvistusviestejä. (RFC 5905: 2010.) Tätä ominaisuutta käyttämällä NTP:tä voidaan käyttää myös datadiodin yli. Työssä testattiin NTP:tä käyttämällä ohjelmia NTP ja Chrony.

3.4 Toteutusmallit

Kotitekoisen datadiodin toteutukseen on useita mahdollisuuksia. Lähtökohtaisesti tässä työssä keskityttiin testaamaan fyysisesti kahta hieman erilaista toteutusta, joiden pohjana toimii aiemmat tutkimukset, mutta työn edetessä toteutuksia jalostettiin vielä hieman yksinkertaisempaan ja edullisempaan muotoon. Laajemman verkkokokonaisuuden käyttökohteita testattiin Kaakkois-Suomen ammattikorkeakoulun Kotkan kampuksen kyberturvallisuuslaboratoriossa toimivassa virtuaalilaboratoriossa, joka mahdollistaa laitteiden lisäämisen ja verkon laajentamisen vaivattomasti ja lisäämättä kustannuksia.

3.4.1 Kytkimet

Ensimmäinen tässä työssä suoritettu toteutus perustuu Hewesin (2017) projektiin Githubissa, jossa hän esittelee kytkennän ja konfiguraatiot käyttäen kahta Cisco Catalyst 3850 -kytkintä sekä kolmea optista SFP:tä (Small form-factor pluggable). Kytkinten väliin liitetty valokuitu on kytketty toiseen lähettävän kytkimen SFP:n lähettävään porttiin (TX) ja toinen pää on kytketty vastaanottavan kytkimen SFP:n vastaanottavaan porttiin (RX). Kolmas SFP on kytketty lähettävään kytkimeen ja sen TX-portista lähtevä valokuitu on liitetty kyseisen kytkimen edellä mainitun SFP:n RX-porttiin. Tämän kolmannen SFP:n tarkoitus on vain luoda signaali toisen SFP:n RX-porttiin, jotta sovitin aktivoituu.

Kytkinten konfiguraatiot ovat melko yksinkertaiset. Linkin sovittimiin konfiguroidaan IP-osoitteet, molempiin kytkimiin luodaan VLANit kuvaamaan niiden takana olevia verkkoja sekä annetaan staattiset reitit edellä mainittuihin osoitteisiin. Lisäksi määritellään lähettävään kytkimeen staattinen ARP-kartoitus (Address Resolution Protocol) vastaanottavan kytkimen SFP:hen, koska yksisuuntainen yhteys ei salli ARP-kutsuun vastaamista. Kytkimet konfiguroidaan vielä käyttämään LLDP:tä (Link Layer Discovery Protocol) sekä sovittimien välinen neuvottelu estetään linkistä.

Tässä työssä suoritetussa toteutuksessa molempien kytkinten taakse liitettiin tietokoneet, joihin asennettiin Linux Ubuntu 22.04 -käyttöjärjestelmät. Koneisiin asennettiin UDPcast, VLC Media Player, FFmpeg, NTP sekä Chrony ja testattiin tiedoston siirtämistä, ruudunjakoa, videon suoratoistoa, lokitietojen jakoa Rsyslogilla sekä aikatiedon jakamista NTP:llä ja Chronylla. Pohjana tiedoston siirron ja ruudunjaon käyttöönottoon käytettiin Vrolijkin (2022) projektia Githubissa.

3.4.2 Mediamuuntimet

Toisena toteutuksena sovellettiin Stevensin (1999) esittelemää ratkaisua, jossa käytetään kolmea mediamuunninta kahden tietokoneen välillä. Käytetyt mediamuuntimet muuttavat kuitusignaalin kuparisignaaliksi RJ45-liitäntään. Periaatteeltaan kytkentä on sama kuin ensimmäisessä mallissa eli kytketään valokuitu yksisuuntaiseksi linkiksi ja käytetään kolmatta mediamuunninta tuottamaan paluusignaali lähettävään muuntimeen. Poikkeuksena edelliseen jätetään kytkimet pois välistä ja kokeillaan yhteyden muodostusta suoraan tietokoneelta toiseen. Yhteyden muodostuksen jälkeen työssä testattiin samoja tiedonsiirtomenetelmiä kuin ensimmäisessä toteutuksessa.

3.4.3 Kuitujakaja

Työn edetessä syntyi idea voisiko paluusignaalin luoda jakamalla lähtevä signaali kahdeksi ja ohjata toinen niistä takaisin. Kyberturvallisuuslaboratoriosta ei valmiiksi löytynyt kuitujakajia, joten ideaa testattiin ensin käyttämällä multiplekseriä, jossa on lähtevän signaalin monitorointiin tarkoitettu portti. Tästä

monitoriportista ohjattiin signaali takaisin lähettävään SFP:hen ja toteutus voitiin todeta toimivaksi. Tämän jälkeen kyberturvallisuuslaboratorioon tilattiin kuitujakaja, jonka avulla edellisen luvun toteutuksesta saatiin pudotettua yksi mediamuuntimista pois. Tästä toteutuksesta löytyi myös vastaava esimerkki Vrolijkin (2022) projektista Githubissa.

4 TOTEUTUS

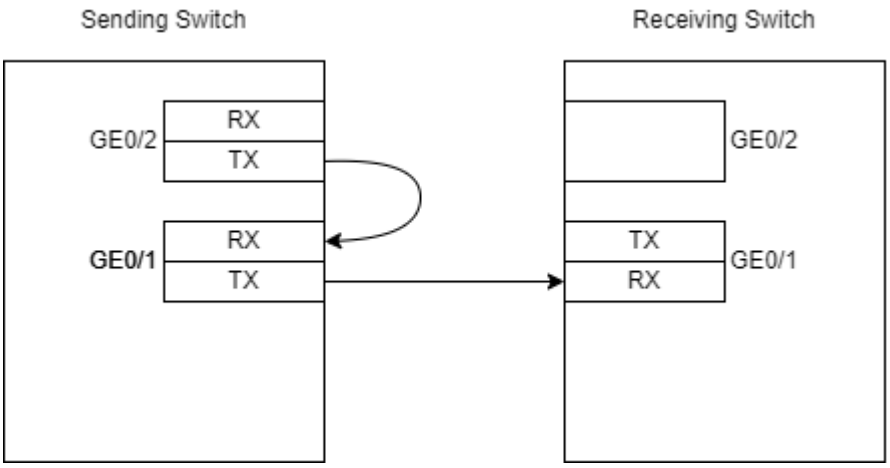
Tässä luvussa käydään yksityiskohtaisesti läpi vaiheet, joilla työssä toteutetut datadiodit rakennettiin. Luvussa esitellään kytkennät, konfiguraatiot yhteyden muodostukselle sekä käytetyt ohjelmistot eri tiedonsiirtomenetelmille. Saatuja tuloksia analysoidaan ja verrataan niitä mahdollisiin vaihtoehtoisiin toteutustapoihin.

4.1 Kytkennät

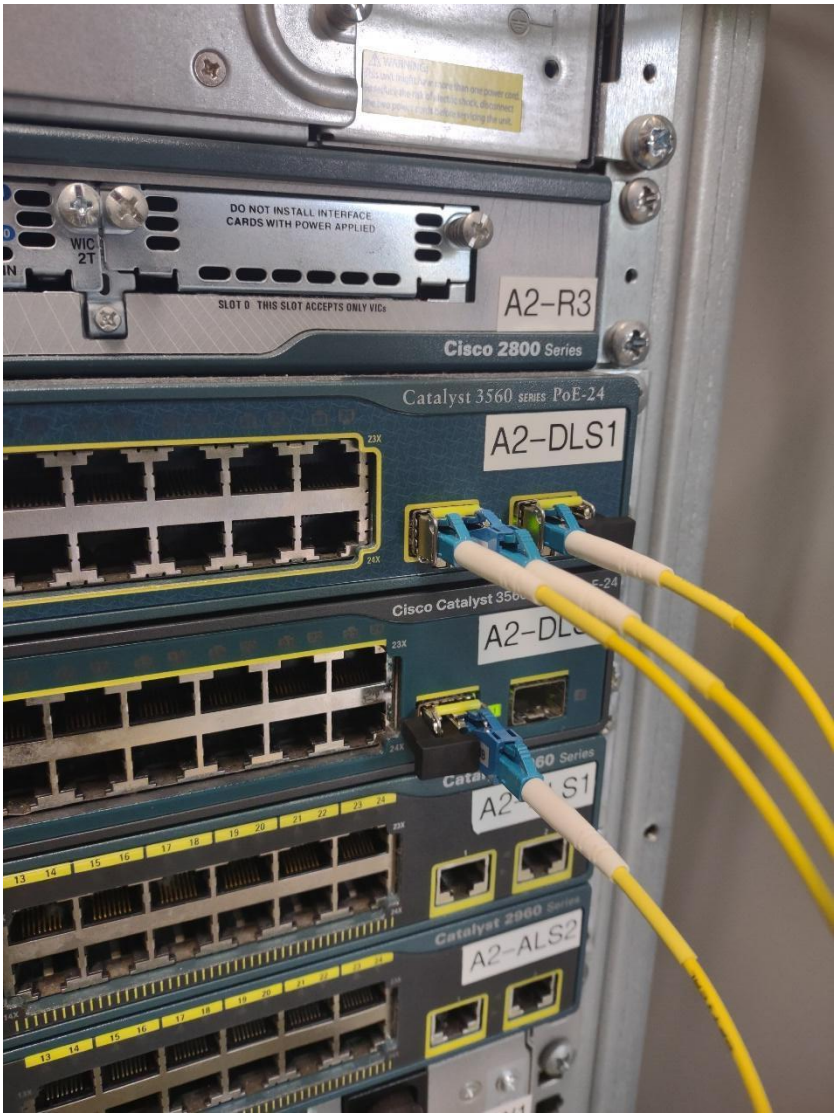
Kotitekoinen datadiodi voidaan rakentaa monella eri tavalla riippuen saatavilla olevista resursseista. Kyberturvallisuuslaboratoriosta löytyi jo valmiiksi valtaosa tarvittavista laitteista ensimmäisiin kokeisiin ja työn edetessä tilattiin lisää laitteita tarpeen vaatiessa. Tässä luvussa esitellään työssä toteutetut kytkentämallit yksisuuntaisen yhteyden muodostamiseksi.

4.1.1 Datadiodi kytkimien avulla

Ensimmäinen toteutus tehtiin käyttämällä Kyberturvallisuuslaboratoriosta löytyvää laitteistoa. Kytkiminä käytettiin Ciscon Catalyst 3560 -sarjan PoE-24-kytkimiä. Yhteys muodostettiin käyttämällä yksimuotoisia CWDM 1.25G 1550 nm -SFP:itä. Lähettäjänä toimivaan kytkimeen kytkettiin kaksi SFP:tä ja vastaanottavaan kytkimeen yksi. Toisesta lähettäjän SFP:stä liitettiin valokuitu lähettävästä portista vastaanottavan kytkimen SFP:n vastaanottavaan porttiin. Toisesta lähettäjän SFP:n lähettävästä portista liitettiin valokuitu lähettäjän toisen SFP:n vastaanottavaan porttiin. Lopuksi lisättiin vielä 6 dB:n vaimentimet molempiin kuituihin rajoittamaan näiden lyhyiden linkkien tehoa. Kuvat 1 ja 2 havainnollistavat kytkentää.



Kuva 1. Kytkennän fyysinen topologia

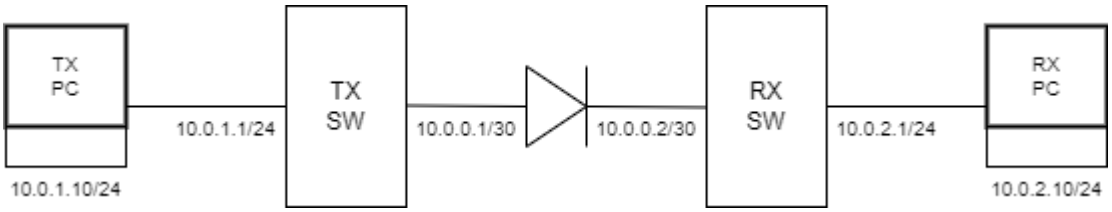


Kuva 2. Kuva kytkennästä

Vaikka lähettävän kytkimen lähetyksestä vastaavaan SFP:hen kytkettiin myös paluusignaali, tämä ei vielä riittänyt sovittimen aktivoimiseen. Molemmat kytki-

met konfiguroitiin reitittäviksi sekä määriteltiin molempien kytkimien GigabitEthernet0/1-portteihin (linkin muodostavat portit) IP-osoitteet. Linkin ominaisuuksien neuvottelu estettiin komennolla speed nonegotiate. Kolmas paluusignaalia simuloiva sovitin jätettiin konfiguroimatta. Nyt sovittimet aktivoituivat, mutta lähettävään kytkimeen täytyi vielä lisätä staattinen ARP-kartoitus vastaanottavaan sovittimeen, koska ARP-pyynnöt eivät saaneet vastauksia. Tämän jälkeen yksisuuntainen yhteys muodostui.

Päätelaitteina käytettiin kahta PC:tä, joihin asennettiin käyttöjärjestelmäksi Linux Ubuntu 22.04. PC:ihin asennettiin myös ohjelmistot Wireshark, VLC Media Player, FFmpeg, UDPcast ja NTP. PC:t liitettiin vastaavien kytkimiensä FastEthernet0/1-portteihin RJ45-kaapeleilla. Kytkimiin konfiguroitiin VLAN 10, näihin VLAN:eihin IP-osoitteet, FE0/1-portit access-porteiksi ja sallittiin portteihin pääsy edellä mainittuun VLAN:iin. Lähettävään kytkimeen lisättiin staattinen reitti vastaanottavan kytkimen VLAN:in verkkoon.



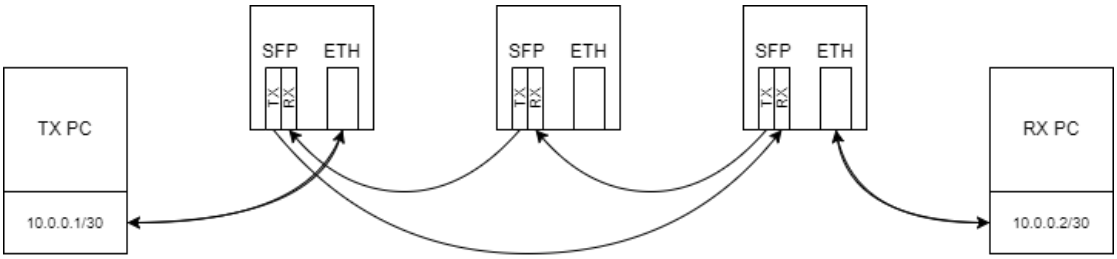
Kuva 3. Verkon looginen topologia

PC:ihin vaihdettiin staattiset IP-osoitteet niiden vastaavien kytkinten VLAN-verkoista kuvan 3 mukaisesti. Tietoliikenne alkoi nyt kulkea lähettävältä PC:ltä vastaanottavalle. Tämä voitiin testata avaamalla Wireshark tietokoneilta ja pingaamalla lähettävältä koneelta. ICMP-paketit menivät perille, mutta vastauksia ei voitu toimittaa takaisin.

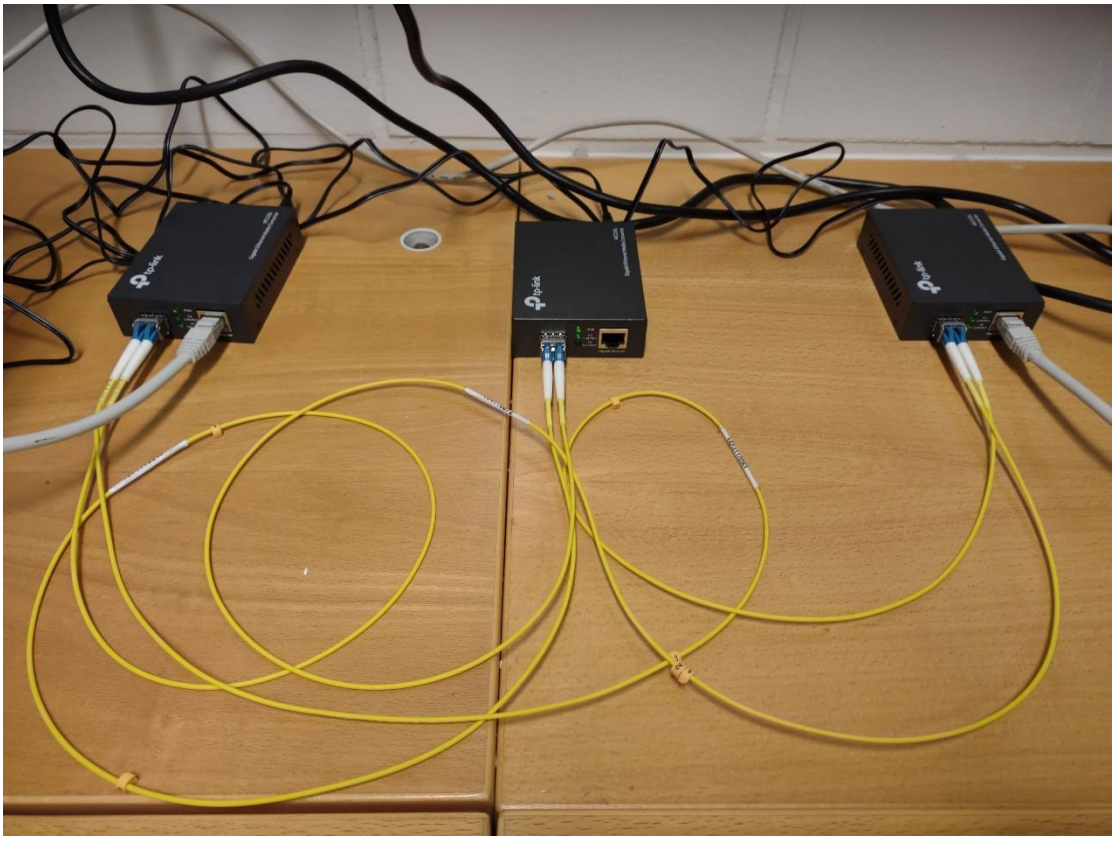
4.1.2 Datadiodi mediamuuntimilla

Tämä versio toteutettiin käyttämällä kolmea TP-Link MC220L -mediamuunninta kytkettyinä suoraan tietokoneisiin. Molemmista tietokoneista kytkettiin yhdet ethernet-kaapelit kahteen eri mediamuuntimeen. Kaikkiin kolmeen mediamuuntimeen asennettiin SFP:t ja kytkettiin lähettävän koneen mediamuuntimen SFP:n lähettävään porttiin valokuitu, jonka toinen pää kytkettiin vastaanottavan koneen mediamuuntimen SFP:n vastaanottavaan porttiin. SFP:t eivät

vielä aktivoituneet, joten kolmannen mediamuuntimen SFP:stä kytkettiin valokuidut kahden muun mediamuuntimen SFP:iden vapaisiin portteihin. Kuvat 4 ja 5 havainnollistavat kytkentöjä.



Kuva 4. Kytkennän fyysinen topologia



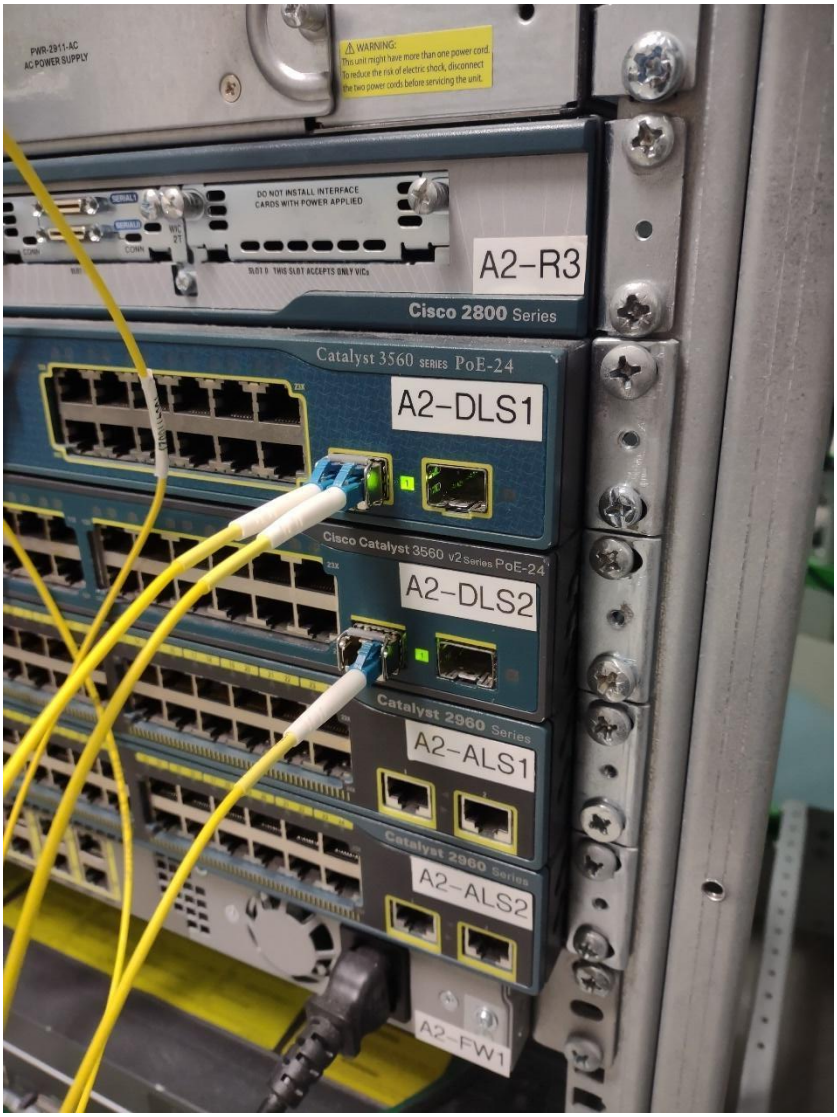
Kuva 5. Kuva kytkennästä mediamuuntimilla

Tietokoneiden IP-osoitteet vaihdettiin samaan aliverkkoon ja lähettävään tietokoneeseen lisätiin staattinen ARP-kirjaus vastaanottavan koneen verkkosovittimeen. Nyt yksisuuntainen yhteys saatiin muodostettua ja paketit alkoivat kulkea datadiodin läpi. Tämä voitiin testata esimerkiksi pingaamalla kuten edellisessä kytkennässä.

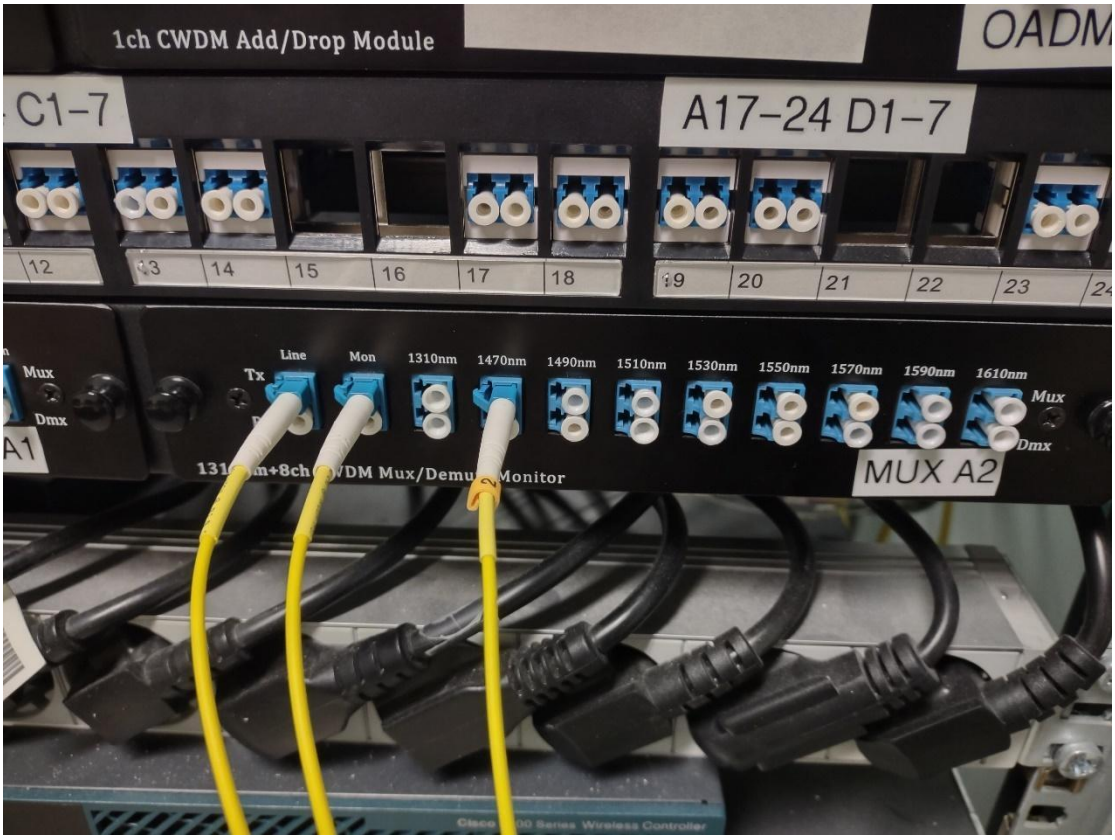
4.1.3 Datadiodi multiplekserin avulla

Työn edetessä syntyi idea voisiko lähettävän sovittimen aktivoivan signaalin saada aikaiseksi jakamalla lähtevä signaali kahdeksi ja ohjaamalla toinen niistä takaisin. Tämä menettely poistaisi kolmannen aktiivilaitteen kytkennästä ja yksinkertaistaisi kytkentää. Parhaiten tämän testaamiseen sopisi yksinkertainen kuitujakaja, joka jakaa yhden valokuidun kahdeksi. Huolena oli lähinnä, syntyisikö kyseisestä silmukasta ongelmia lähettävän sovittimen tai kytkimen toimivuuteen. Ennen edellä mainitun kuitujakajan hankintaa ideaa testattiin käyttämällä hyväksi Kyberturvallisuuslaboratoriosta löytyviä multi-/demultiplekssereitä.

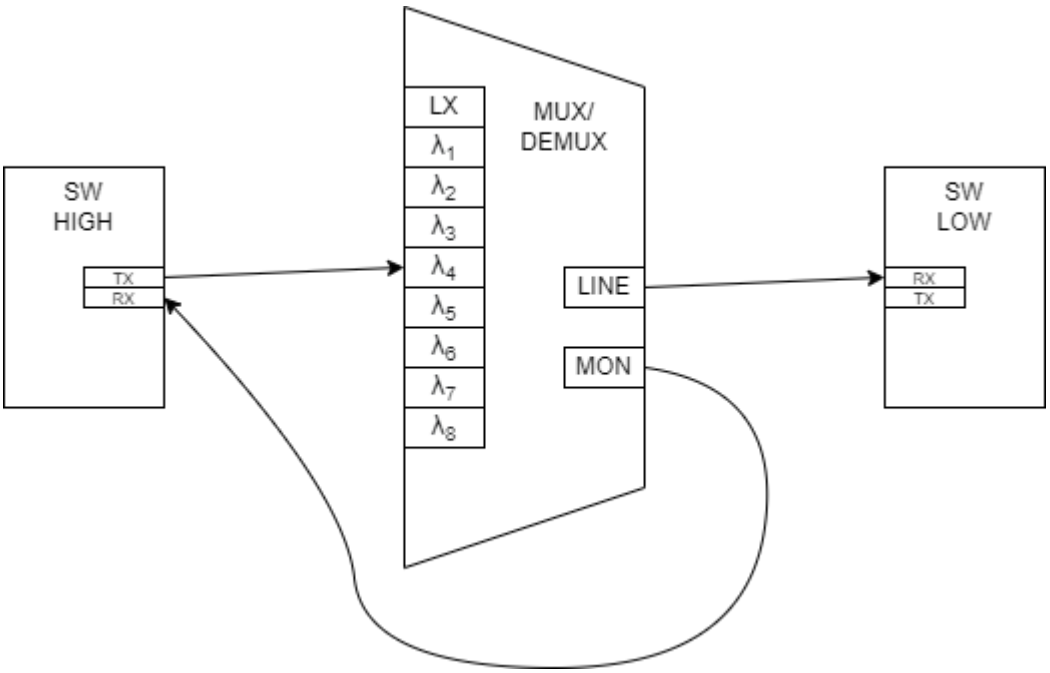
Laitteistona käytettiin kytkimiä kuten luvussa 4.1.1 ja kytkimet konfiguroitiin täysin samoin. Eroavaisuutena aiempaan lähettävän kytkimen toinen SFP jätettiin pois ja linkistä vastaava kuitu ohjattiin multiplekserin läpi. Kyseisessä multiplekserissä on ulostulolinja lähetystehon monitoroinnille, johon jakautuu 3 % lähtevän signaalin tehosta. Tätä linjaa hyödyntäen multiplekseristä lähtevä signaali saatiin jaettua kahdeksi ja monitorilinjasta lähtevä signaali ohjattiin takaisin lähettävän kytkimen SFP:n RX-porttiin. Kuvat 6–8 kuvaavat kytkentää.



Kuva 6. Kytkinten kytkennät



Kuva 7. Multiplekserin kytkennät



Kuva 8. Kytkennän fyysinen topologia

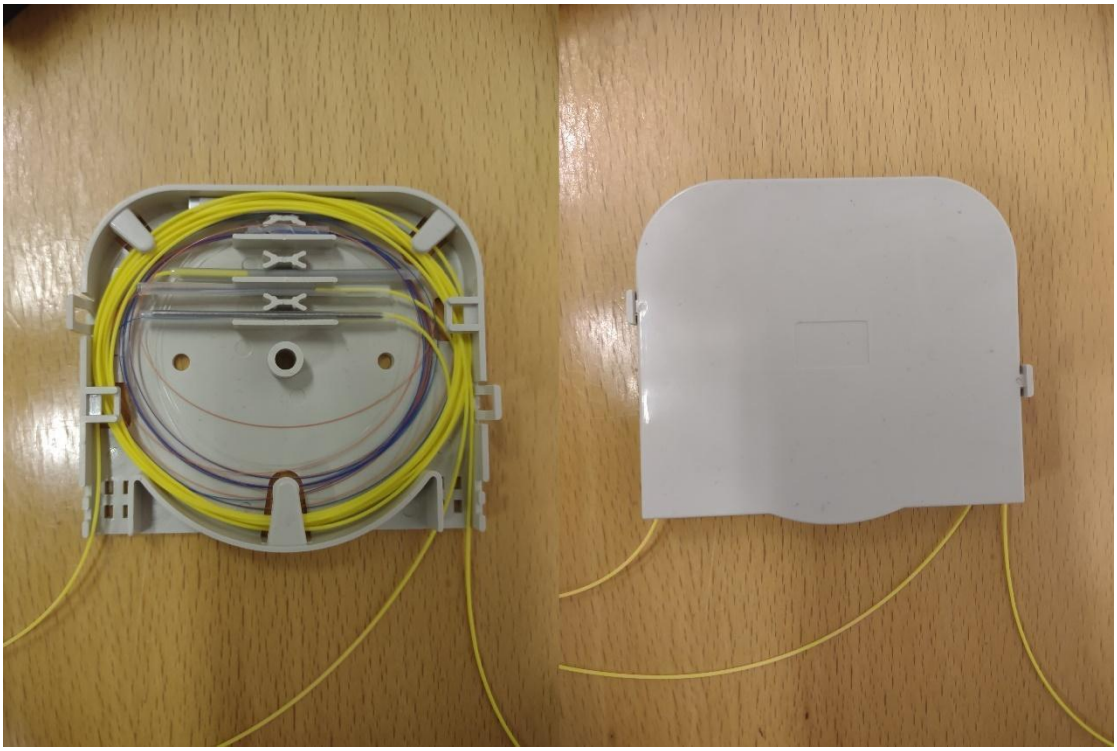
Aluksi kytkentää testattiin käyttämällä LX-aallonpituutta (1310 nm), mutta kyseisessä SFP:ssä ei lähetysteho riittänyt aktivoituakseen 3%:n paluusignaalilla. Kun lähettäväksi SFP:ksi vaihdettiin CWDM-aallonpituutta (1470 nm) käyttävä SFP, niin lähettävä sovitin aktivoitui ja yhteys saatiin muodostettua.

Tästä voitiin todeta, että lähettävä SFP saadaan aktivoitua hyödyntämällä sen omaa lähetyssignaalia ja täten myös kuitujakajan tulisi toimia vastaavaan tarkoitukseen.

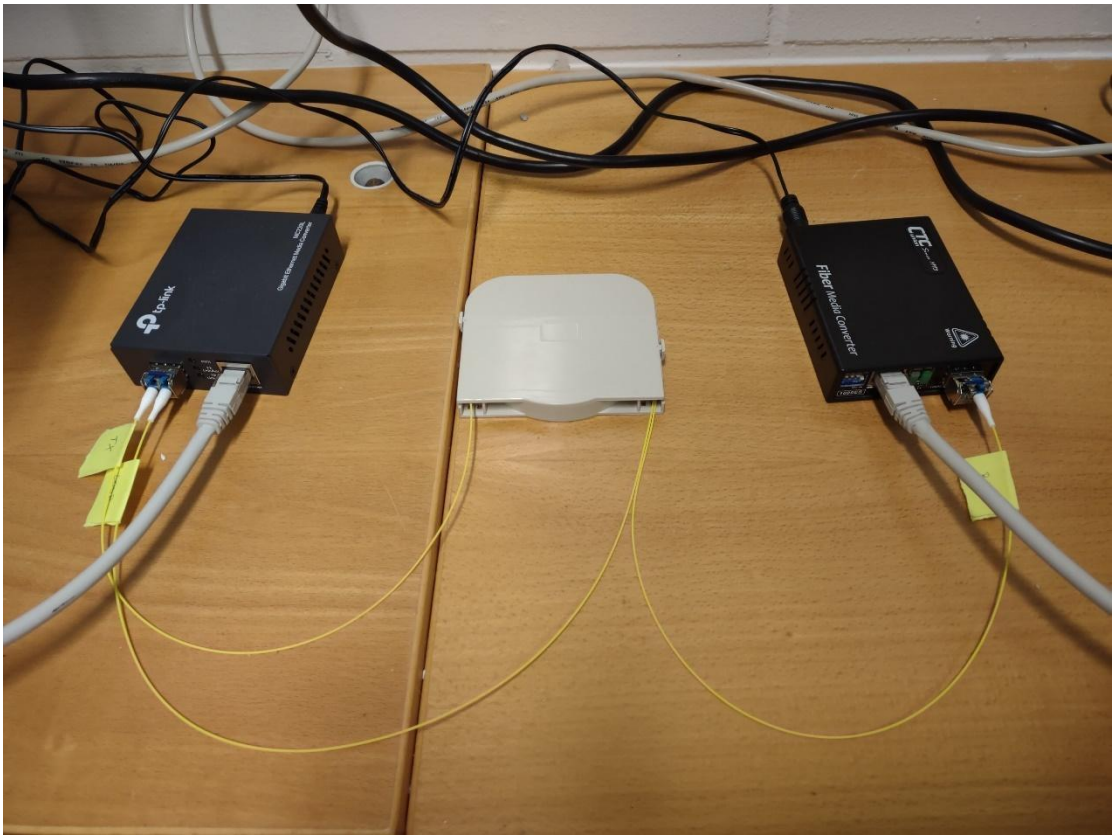
4.1.4 Datadiodi kuitujakajalla

Tämän toteutuksen testaukseen kyberturvallisuuslaboratorioon tilattiin kuitujakajia ja liitäntäkoteloita. Kyseiset kuitujakajat olivat paljasta kuitua ilman liittimiä ja ne jakavat signaalin tasaisesti kahteen kuituun. Kyberturvallisuuslaboratoriosta löytyi valmiiksi kuitujen hitsauslaite sekä LC-liittimellisiä kuidun häntiä. Kuitujakajaan hitsattiin edellä mainitut hännät jokaiseen kolmeen kuidunpäähän. Koska suojaamaton kuitu on melko herkkää, asetettiin ylimääräinen kuitu ja liitäntäkohdat liitäntäkoteloon. Näin kuitujakaja saatiin kätevästi pieneen tilaan ja suojaan ympäristön häiriötekijöiltä. Kuvissa 9 ja 10 esitellään käytetty kuitujakaja sekä lopputulos.

Kuva 9. Kuitujakaja, jonka malli on 1 x 2 PLC Fiber Splitter, Steel Tube, Bare Fiber 250µm, No Connector



Kuva 10. Kuitujakaja liitäntäkotelossa avattuna ja suljettuna



Kuva 11. Datadiodi kuitujakajan avulla

Kyberturvallisuuslaboratoriosta löytyi myös CTC Unionin mediamuunnin Fiber Media Converter 1000ES, joka oli eri mallia kuin luvussa 4.1.2 käytetyt mediamuuntimet. Testauksen jälkeen todettiin, että kyseinen mediamuunnin ei

tarvitse lähetyssignaalia siihen liitetyn SFP:n aktivoimiseen. Tätä hyödyntäen liitäntää saatiin jälleen yksinkertaistettua poistamalla tarve vastaanottavan mediamuuntimen lähetyssignaalille ja lopputulos on esitelty kuvassa 11. Nopean testauksen jälkeen todettiin, että hitsaukset olivat onnistuneita ja liitännät toimivat kokonaisuutena moitteettomasti. Jälkeenpäin huomattiin, että myös TP-Linkin mediamuuntimet toimivat ainakin tässä kytkennässä ilman SFP:n lähetyssignaalia.

4.2 Ohjelmistot

Datadiodin toimintaa testattiin muutamilla yleishyödyllisillä tiedonsiirtomenetelmillä, joita voidaan hyödyntää yksisuuntaisen yhteyden yli. Tässä luvussa käydään yksityiskohtaisesti läpi eri ohjelmistojen käyttöönottoon ja käyttöön vaadittavat toimenpiteet. Kaikki luvussa tehdyt testaukset tehtiin käyttäen luvun 4.1.1 kytkentää, joten tiedonsiirtonopeus on rajattu 100 Mbps:n nopeuteen, mikä johtuu käytettyjen kytkinten FastEthernet-porttien tiedonsiirtonopeudesta. Muilla kytkennöillä tehdyistä testauksista mainitaan tapauskohtaisesti.

4.2.1 UDPcast

Yksisuuntainen yhteys tietokoneiden välille saatiin muodostettua ja ominaisuuksien testaus aloitettiin tiedostonsiirrolla. Aluksi luotiin testitiedosto test.txt, joka sisälsi vain sanan Hello. Lähettävän PC:n verkkosovittimen tunnus oli tässä tapauksessa eno1, joten lähetys valmisteltiin komennolla:

```
udp-sender --interface eno1 --async --fec 8x8/64
--file test.txt --mcast-rdv-address 10.0.2.10
--mcast-data-address 10.0.2.10 --pointopoint
```

Komennon suorittamisen jälkeen ohjelma jäi odottamaan näppäimen painallusta lähetyksen aloittamiseksi. Ennen lähetystä valmisteltiin kuitenkin vastaanottava tietokone valmiuteen tiedoston vastaanottamiseksi. Vastaanottavaan tietokoneeseen syötettiin komento:

```
udp-receiver --interface eno1 --nosync --file
test.txt
```

Komennon suorittamisen jälkeen ohjelma etsi lähettäjän signaalia ja ilmoitti, kun yhteys oli muodostettu. Nyt lähetys voitiin aloittaa ja huomattiin, että tiedosto siirtyi datadiodin läpi.

Yhden sanan sisältävä tekstitiedosto on kuitenkin todella pieni kooltaan, joten sen lähettäminen ei vielä kerro paljoakaan datadiodin tiedonsiirtokyvystä. Suuremman tiedoston lähettämistä testattiin luomalla sattumanvarainen 250 Mb:n tiedosto komennolla:

```
head -c 250M /dev/urandom > testfile.tmp
```

Tiedosto lähetettiin datadiodin yli edellä mainittuja komentoja soveltaen. Tiedonsiirtonopeus oli keskimäärin noin 80 Mbps, koska kytkinten FastEthernet-portit rajoittivat yhteyden 100 Mbps:n tiedonsiirtonopeuteen. Tiedostonsiirto onnistui ja tiedostojen yhteneväisyys voitiin testata vertaamalla niiden SHA-256-tarkisteita.

Kansion lähettäminen toteutettiin pakkaamalla se ensin `tar`-komennon avulla ja lähettämällä pakattu tiedosto datadiodin yli. Lähetys myös automatisoitiin luomalla Python-skripti ja käyttämällä lähettäjän `udp-sender`-komennossa valintaa `--autostart`. Edellä mainitulla valinnalla voitiin määrittää aika sekunneissa, jonka jälkeen lähetys alkoi itsestään. Vastaanottajalle tarvitsi vain luoda silmukka (loop) `udp-receiver`-komennolle, sillä ohjelma käynnistyessään jää odottamaan lähetystä ja lopettaa itsestään, kun lähetys päättyy.

Seuraavaksi testattiin FEC:n vaikutusta lähetyksen luotettavuuteen ja koitettiin löytää optimaalisin asetus, jolla säilytettiin luotettava tiedonsiirto sekä saavutettiin korkein mahdollinen tiedonsiirtonopeus kyseisellä järjestelyllä. Muutettavat valinnat lähetyksessä olivat `--max-bitrate` ja `--fec`. Ensimmäinen määrittää rajan raa'an datan tiedonsiirtonopeudelle ja jälkimmäinen sisältää kolme muutettavaa arvoa, jotka ovat interleave, redundancy ja stripesize.

Taulukko1. FEC-testauksen tulokset

FEC parametrit	Max-bitrate (*Mbps*)	Tietosisällön siirtonopeus (*Mbps*)	Onnistuminen kymmenestä lähetyksestä
-	-	94,2	6/10
-	50	48,5	10/10
-	80	77,7	10/10
-	90	87,4	10/10
-	95	92,2	10/10
-	96	93,3	10/10
-	97	94,2	0/3
8x8/128	-	82,2	10/10
1x1/1	-	47,1	4/6
1x1/2	-	62,8	10/10
2x1/1	-	47,1	10/10
1x1/4	-	75,4	10/10
1x1/8	-	83,7	10/10
1x1/128	-	93,5	10/10
1x2/128	-	92,7	10/10
1x8/128	-	83,7	10/10
2x1/128	-	93,5	10/10
8x1/128	-	93,5	10/10
8x2/128	-	92,7	10/10

Taulukosta 1 huomataan, että ilman FEC:tä ja lähetysnopeuden rajoitusta paketteja katosi paikoitellen ja lähetyksen luotettavuus kärsi. Kuitenkin rajoittamalla raaka lähetysnopeus 96 Mbps:ään saatiin hieman maksiminopeutta pudotettua ja kaikki kymmenen lähetystä onnistui. FEC:n suhteen huomattiin, että korkeimmalla stripesizella saatiin vielä suurempi nopeus kuin ilman FEC:tä. Interleaven kasvatus ei vaikuttanut nopeuteen, mutta redundancyn lisäys hidasti nopeutta hieman. Koska lähetyksissä ei havaittu pitkiä ryppäitä kadonneita paketteja, 8x1/128 pitäisi tässä järjestelyssä riittää hyvin takaamaan lähetyksen luotettavuus sekä korkein tiedonsiirtonopeus.

Työn lopussa tätä hieman testattiin viimeisen toteutuksen kytkennällä, jolloin tiedonsiirtonopeus oli tietokoneiden verkkokorttien mukainen 1 Gbps. Tällöin paketteja katosi kaikilla testatuilla FEC:n arvoilla, mutta maksiminopeuden rajausta ei testattu. Myös UDP-puskurin koon muuttaminen jäi testaamatta näissä kokeissa.

4.2.2 VLC Media Player

Videonjakoa kokeiltiin ensin käyttämällä VLC:n suoratoisto-ominaisuutta. Vastaanottaja valmisteltiin kuuntelemaan porttia 5004 ja odottamaan lähetyksen alkua. Suoratoistolähetys aloitettiin valitsemalla haluttu videotiedosto, suoratoiston tyypiksi RTP / MPEG transport stream, vastaanottajaksi osoite 10.0.2.10 ja portti 5004, lähetyksen nimeksi test ja videon profiiliksi H264 + mp3. Lähetyksen aloituksessa kesti muutama sekunti, mutta lähetys onnistui ja video sekä ääni välittyivät vastaanottajalle. Suoratoisto oli kuitenkin melko epätasaista ja erityisesti kuva pätki jonkin verran.

VLC:llä testattiin myös ruudunjakoa. Lähetettäväksi tyypiksi valittiin Capture Device -välilehdeltä Capture modeksi Desktop ja lähetystä kokeiltiin framerateella 1. Muilta osin asetukset pidettiin samoina yhteyden molemmissa päissä. Vastaanottaja havaitsi lähetyksen, mutta kuva ei välittynyt kunnolla. Muutamalla yrityksellä vastaanottajalle välittyi puolikas kuva lähettäjän kuvaruudusta, mutta muuta ei tapahtunut. Samaa kokeiltiin käyttämällä UDP:tä lähetystyyppinä, mutta mitään merkittävää muutosta ei havaittu. Kun frameratea nostettiin 15:een, suoratoisto alkoi toimimaan paremmin UDP:llä. Lähetyksessä esiintyi silti hieman viivettä ja välillä kuva seisahti useiksi sekunneiksi.

4.2.3 FFmpeg

FFmpeg on ilmainen avoimen lähdekoodin ohjelmisto, jolla voi purkaa, koodata, multipleksata, demultipleksata, suoratoistaa, suodattaa ja toistaa ”lähes kaikkea, mitä ihmiset ja koneet ovat luoneet” (FFmpeg 2022). Ohjelman testaus aloitettiin kokeilemalla ruudunjakoa. Pienen tutustumisen jälkeen päädyttiin käyttämään seuraavia valintoja lähetykseen:

```
ffmpeg -f x11grab -s 1920x1080 -framerate 1
-i :0.0 -async 1 -f mpegts udp://10.0.2.10:1234
```

Vastaanottajalle riitti komento:

```
ffplay udp://10.0.1.10:1234
```

Ruudunjako toimi hienosti, mutta viivettä syntyi noin viisi sekuntia. Nostamalla kuvataajuutta 30:een saatiin viive alle sekuntiin. Myös vastaanottajan valinnat `-analyzeduration 1`, `-fflags nobuffer` ja `-probesize 32` auttoivat hienosäätämään lähetystä vielä sujuvammaksi. Näillä asetuksilla lähetys oli kuvatarkkuudeltaan 480p. Tarkkuutta voitiin parantaa muuttamalla lähetyksen formaatti mpegts:stä h264:ksi lisäämällä vastaanottajalle valinta `-f h264`. Viivettä saatiin pienennettyä valinnoilla `-preset ultrafast` ja `-tune zerolatency`.

Videon lähettäminen onnistui vastaavalla lähettäjän komennolla. Muutamia valintoja muuttamalla komento sai muodon:

```
ffmpeg -re -i Intro.ogg -s 1920x1020 -async 1
-f mpegts udp://10.0.2.10:1234
```

Kyseisessä komennossa Intro.ogg on testauksessa käytetyn videotiedoston nimi.

Formaatin h264 käyttö vaatii lähetykseltä enemmän valmistautumista ja lisää hieman viivettä ennen suoratoiston alkua. Kun vastaanottaja asetettiin valmiuteen ennen lähetystä, sekä video että ruudunjako alkoivat hyvin pienellä viiveellä, mutta kun vastaanottaja aloitti vastaanoton kesken lähetyksen, syntyi viivettä useita sekunteja. Vastaanottajan eri valintojen testausten jälkeen päädyttiin käyttämään videon vastaanottamiseen aiemmin mainitun komennon lisäksi valintaa `-probesize 32` ja ruudunjaolle valintaa `-fflags nobuffer`.

4.2.4 Rsyslog

Rsyslog on lokitietojen etäkäyttöön tarkoitettu palvelu, joka on monessa Linux-distribuutiossa valmiiksi asennettuna. Sillä voidaan lähettää syslog-tietoja automatisoidusti lokipalvelimelle. Rsyslog aktivoitiin muokkaamalla vastaanottavan tietokoneen /etc/rsyslog.conf-tiedostoa poistamalla kommenttimerkit (#) riveiltä

```
#module(load=”imudp”)
```

```
#input(type=”imudp” port=”514”)
```

sekä lisäämällä rivit

```
$template remote-incoming-logs
”/var/log/remote/%HOSTNAME%”.log
*.* ?remote-incoming-logs
```

Lähettävän tietokoneen vastaavaan tiedostoon lisättiin vain rivi

```
*.* @10.0.2.10:514
```

Kun vielä lisättiin käyttäjäoikeudet vastaanottajan kansioon /var/log, niin syslog-viestit alkoivat kulkea.

4.2.5 NTP

NTP:n testaus aloitettiin asentamalla molempiin tietokoneisiin NTP-paketti, joka on oletus-NTP-ohjelma Ubuntulle. Lähettävä tietokone konfiguroitiin NTP-palvelimeksi ja vastaanottava tietokone konfiguroitiin asiakkaaksi. Konfiguraatiot tehtiin oletustiedostoon /etc/ntp.conf, joka sisältää perusasetukset selityksineen. Pitkän testailun ja tiedon etsimisen jälkeen palvelimena toimivan tietokoneen ntp.conf-tiedostoon päädyttiin jättämään vain rivit

```
broadcast 10.0.0.3 minpoll 3 maxpoll 4
disable auth
server 127.127.1.0 prefer
fudge 127.127.1.1 stratum 0
```

Edellä mainituilla palvelimen konfiguraatioilla saatiin palvelin lähettämään säännöllisiä broadcast-paketteja tietokoneiden väliseen linkkiverkkoon. Suurimmat ongelmat aiheutuivatkin asiakkaana toimivan koneen konfiguroinnista. Lähtökohtaisesti kaikki NTP:hen liittyvä kommunikointi ja ajan synkronointi tietokoneiden välillä sisältää ainakin jonkinlaista kaksisuuntaista viestintää. Haasteena olikin löytää oikeat konfiguraatiot, jotta asiakas toimisi täysin passiivisesti ja synkronoisi kellonsa ainoastaan saapuvien broadcast-pakettien avulla.

Ongelman ratkaisua hankaloitti erityisesti se, että internetistä oli varsin vaikea löytää NTP:n broadcast-ominaisuuteen liittyvää tietoa, koska ominaisuus on melko vähän käytetty eikä sen käyttöä suositella normaaleissa kaksisuuntaisissa verkoissa. Lisäksi NTP-ohjelman ohjekirjoja löytyi useita, joissa oli eri

komentoja eikä kaikki toimineet niitä testatessa. Esimerkiksi komennon `broadcastclient` lisävalinnan `novolley` pitäisi ohjeiden mukaan tehdä asiakkaasta täysin passiivisen kuuntelijan siten, ettei se lähettäisi synkronointiviestejä palvelimelle saatuaan ensimmäiset broadcast-paketit. Tämä ei kuitenkaan toiminut työssä käytetyssä NTP:n uusimmassa versiossa.

Lopulta ratkaisu löytyi komennosta `broadcastdelay`, jolla ohjekirjojen mukaan voi säätää manuaalisesti viiveen broadcast-paketeille, mikäli kaksisuuntainen synkronointi ei jostain syystä ole mahdollista. Lisäksi ohjeessa sanotaan, että mikäli komentoa ei käytetä, niin oletusarvo on 4 ms. Tämä johti oletukseen, ettei komentoa tarvitse erikseen käyttää, mutta myöhemmin selvisi, että tämä oli nimenomaan ratkaiseva komento kyseiseen ongelmaan. Näin ollen asiakkaan ntp.conf-tiedostoon riittää rivit

```
disable auth
broadcastclient
broadcastdelay 0.004
interface ignore wildcard
interface listen eno1
```

Tämän jälkeen testattiin vielä ohjelmaa Chrony NTP-palvelimena. Chronya ei ole mahdollista konfiguroida broadcast-asiakkaaksi, joten vastaanottavaan tietokoneeseen jätettiin edellä käytetty NTP-ohjelma konfiguraatioineen. On myös mainitsemisen arvoista, ettei laitteessa voi olla kuin yksi NTP-ohjelma kerrallaan. Pienen Chronyyn tutustumisen jälkeen päädyttiin käyttämään konfiguraatiotiedostossa /etc/chrony/chrony.conf komentoja

```
allow 10.0.0.0/30
broadcast 10 10.0.0.3
local stratum 1
```

Palvelimena toimiva tietokone alkoi lähettää broadcast-paketteja ja vastaanottavan tietokoneen edelliset NTP:n konfiguraatiot toimivat myös Chronyn asiakkaana ja kellot saatiin synkronoitua.

5 TULOKSET JA JOHTOPÄÄTÖKSET

Työn tarkoituksena oli tutustua datadiodiin käsitteenä sekä sen tarkempaan toimintaan. Tarkoituksena oli myös selvittää, miten datadiodia voidaan hyödyntää erilaisissa käyttökohteissa ja -tarkoituksissa. Työn suurin paino keskittyi kuitenkin omatekoisen datadiodin rakentamiseen, testaamiseen ja kehittämiseen mahdollisimman yksinkertaiseksi ja edulliseksi. Kaksi ensimmäistä tutkimuskysymystä ovatkin lähinnä johdattelevia kysymyksiä kolmannelle ja nämä käytiin pääsääntöisesti läpi luvussa 3.

Ensimmäinen toteutus tehtiin käyttäen täysin Kyberturvallisuuslaboratoriosta jo entuudestaan löytyvää laitteistoa, ja tavoitteena oli vain saada luotua yksisuuntainen yhteys huomioimatta käytetyn laitteiston edullisuutta. Kun ensimmäinen kytkentä saatiin toimimaan ja suunnitellut ohjelmistot testattua, tilattiin kolme kappaletta mediamuuntimia, joilla ensimmäisessä toteutuksessa käytetyt kytkimet saatiin korvattua. Pikaisen haun perusteella toteutuksessa käytettyjen Ciscon kytkimien WS-C3560-24TS-E ja WS-C3560-24PS-E listahinnat ovat 4 990 USD ja 5 790 USD, joten mediamuuntimien käyttö laskee kytkennän hintaa noin 10 700 USD. TP-Link MC220L -mediamuuntimen kappalehinta oli 25,99 €. Mediamuuntimilla toteutetun kytkennän kokonaisarvoksi jäi arviolta noin 120 €.

Kolmas toteutus tehtiin vain, jotta voitiin testata lähtevän signaalin jakamista multiplekserin avulla. Tässä kytkennässä käytettiin myös valmiiksi löytyvää laitteistoa ja onnistuneen testauksen tuloksena tilattiin kuitujakaja korvaamaan multiplekseri. Tilatussa kuitujakajassa ei ollut liittimiä, joten kuitujen päihin hitsattiin LC-liittimelliset kuituhännät ja liitäntä sisällytettiin liitäntäkoteloon. Toteutuksen loppusummaan ei ole sisällytetty kuitujen hitsaamiseen tarvittavaa laitteistoa. Myöhemmässä testauksessa huomattiin, että ainakin viimeisessä toteutuksessa myös TP-Linkin mediamuuntimet toimivat pelkällä paluusignaalilla. CTC Unionin mediamuuntimen hinta-arvio on noin 100 €:n luokkaa, joten vaihtamalla se TP-Linkin mediamuuntimeen kokonaiskustannusta saatiin pienennettyä edelleen. Lopullisessa toteutuksessa käytetyt osat on listattu taulukossa 2.

Taulukko 2. Toteutukseen käytetyt osat ja niiden hinnat

Tuote	Kappalehinta (€)	Määrä
TP-Link MC220L	25,99	2
LC-häntä	1,70	3
Liitäntäkotelo	1,10	1
Kuitujakaja	5,90	1
SFP	8,00	2
Cat6 Ethernet-kaapeli (1 m)	2,50	2
Yhteensä	**85,08**	

Taulukon 2 hinnat katsottiin Verkkokaupan ja FS:n verkkosivuilta 24.11.2022. Huomioitavaa on, että taulukon 2 hinta-arvioon sisältyy vain yksisuuntaisen yhteyden fyysiseen kytkentään tarvittavat osat. Toimivan datadiodin rakentamisessa tulee huomioida myös yhteyden päissä olevat päätelaitteet. Tässä työssä käytettiin melko suorituskykyisiä PC:itä, jotka nostaisivat kokonaiskustannuksia huomattavasti. Liikenteen ohjaamiseen yksisuuntaisen linkin yli voitaisiin käyttää myös yksinkertaisia kytkimiä tai reitittimiä, mutta tällöin vaadittavat protokollan muutokset tulisi tehdä jokaisella verkon päätelaitteella erikseen.

Barryn (2012) mukaan yritys- ja SCADA-ympäristöissä kaupalliset datadiodit maksavat yleensä noin 30 000 $ ja teollisuuskäytössä hinta voi nousta jopa yli 150 000 $. Hintaan vaikuttaa pääsääntöisesti laitteen tiedonsiirtonopeus, EAL-luokitus (Evaluation Assurance Level), MTBF-arvo (Mean time between failure) ja datadiodin tyyppi. Hieman tuoreemmassa artikkelissa Castagna (2021)

mainitsee datadiodien tyypilliseksi hinnaksi muutama tuhat dollaria. Hän kuitenkin huomauttaa samoista edellä mainituista ominaisuuksista, joiden taso sekä lisääminen nostaa datadiodin hintaa.

Vaikka päätelaitteet laskettaisiinkin mukaan tämän työn toteutuksen loppusummaan, jäätäisiin silti melkoisesti alle kaupallisten datadiodien myyntihintojen. Tässäkin on huomioitavaa, että mikäli suorituskykyä, tiedonsiirtonopeutta tai mitä tahansa vaadittavaa ominaisuutta käyttötarkoituksesta riippuen haluttaisiin parantaa, tulee kustannukset väistämättä nousemaan. Kuitenkin tämän työn tuloksena voidaan todeta, että yksisuuntaisen yhteyden luominen on suhteellisen edullista ja yksinkertaista huomioiden sen merkityksen tietoturvan kannalta.

OPSWAT (2022a) on luonut kaupallisista datadiodeista vertailulistan, jossa mainitaan jokaisen tuotteen tukemat protokollat. Listassa yleisimmät tuetut protokollat ovat UDP, TCP, Syslog, NTP, FTP/SFTP (SSH File Transfer Protocol), SMTP (Simple Mail Transfer Protocol) ja HTTP/HTTPS (Hypertext Transfer Protocol Secure). Tässä työssä testattiin näistä UDP, Syslog sekä NTP ja näitä käytettiin ilmaisilla ohjelmilla. Kyseiset protokollat ja suunnitellut tiedonsiirtomenetelmät saatiin toimimaan ja tuloksena voidaan todeta, että myös omatekoisen datadiodin käyttö muutamiin yleisimpiin tarpeisiin on melko yksinkertaista ja edullista.

6 POHDINTA

Työn tavoitteena oli luoda yksisuuntainen yhteys ja testata sen toimivuutta useammalla eri tavalla käyttämällä mahdollisimman edullisia osia ja ohjelmia. Tässä mielestäni onnistuttiin varsin hyvin ja työssä on kuvattu eri toteutukset vaiheineen ja kuinka lopulliseen tuotokseen päädyttiin. Lopputulosta pystyttiin vertaamaan muutamiin lähteisiin ja huomattiin selvä ero omatekoisen ja kaupallisten datadiodien hinnoissa.

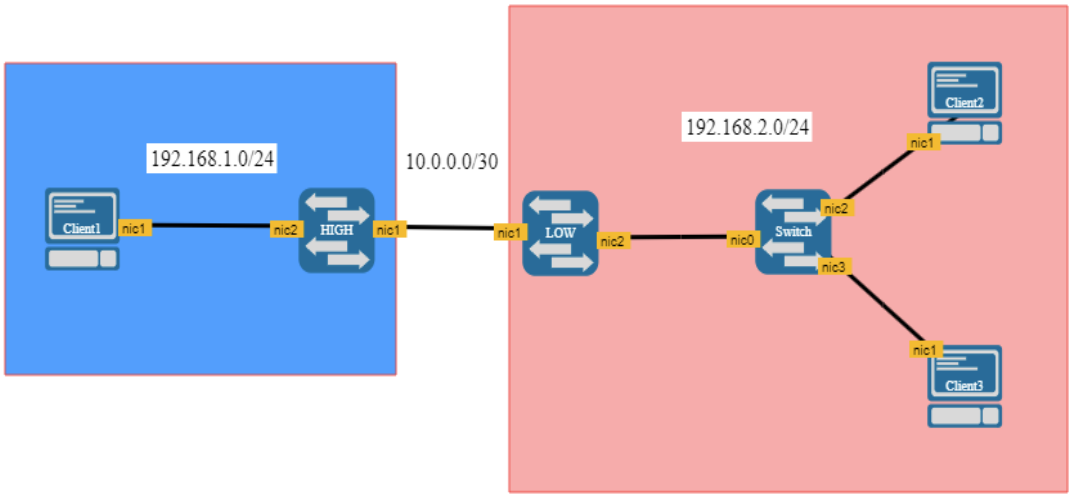
Erilaisia tiedonsiirtotapoja olisi voitu käydä enemmän läpi, kuten SMTP ja HTTP/HTTPS. Myös työssä käytettyjä tiedonsiirtotapoja tuli testattua suhteellisen vähän. Esimerkiksi tiedostonsiirtoa UDPcastilla testattiin suurimmaksi osaksi ensimmäisen toteutuksen kytkennällä, jossa käytettiin Ciscon kytkimiä,

joiden portit rajoittivat tiedonsiirron nopeutta. Työn lopulla tehtiin pienimuotoinen testaus viimeisen toteutuksen kytkennällä, mistä huomattiin, ettei samaa johtopäätöstä FEC:n parametreistä tai tiedonsiirtonopeuden rajauksesta voi tehdä tälle uudelle toteutukselle. Tätä olisi voitu testata enemmän, mutta toisaalta tämä osoittaa myös sen, että testaus tulisi tehdä jokaiselle erilaiselle toteutukselle erikseen käytetystä laitteistosta riippuen.

Työssä on esitelty toteutusten kehityksen kulku vaiheittain ja varsin yksityiskohtaisesti. Lopputulokseen johtava polku on seurattavissa, kehitystyön vaiheet on perusteltu ja tulos on toistettavissa ja siirrettävissä muihin ympäristöihin. Yksisuuntaisen yhteyden luomiseen tarvittavaa laitteistoa voi käyttää missä tahansa verkkoympäristössä, mutta ohjelmistot testattiin vain Linux Ubuntu 22.04 -käyttöjärjestelmällä, joten esimerkiksi Windows-ympäristöissä komennot ja ohjelmistot voivat olla hyvinkin erilaisia.

Lähtökohtaisesti työ perustui aiemmin tehtyyn toteutukseen, jota lähdettiin kehittämään. Myöhemmin huomattiin, että myös lopullisesta toteutuksesta löytyi hyvin vastaava aiempi esimerkki. Vaikka yksisuuntaisen yhteyden luomiseen löytyi aiempia toteutuksia, itse yhteyden käytöstä löytyi melko vähän käytännön esimerkkejä. Tässä työssä on esitelty komentojen tasolla muutaman ohjelman käyttöönotto ja käyttäminen, joiden avulla voidaan datadiodia hyödyntää.

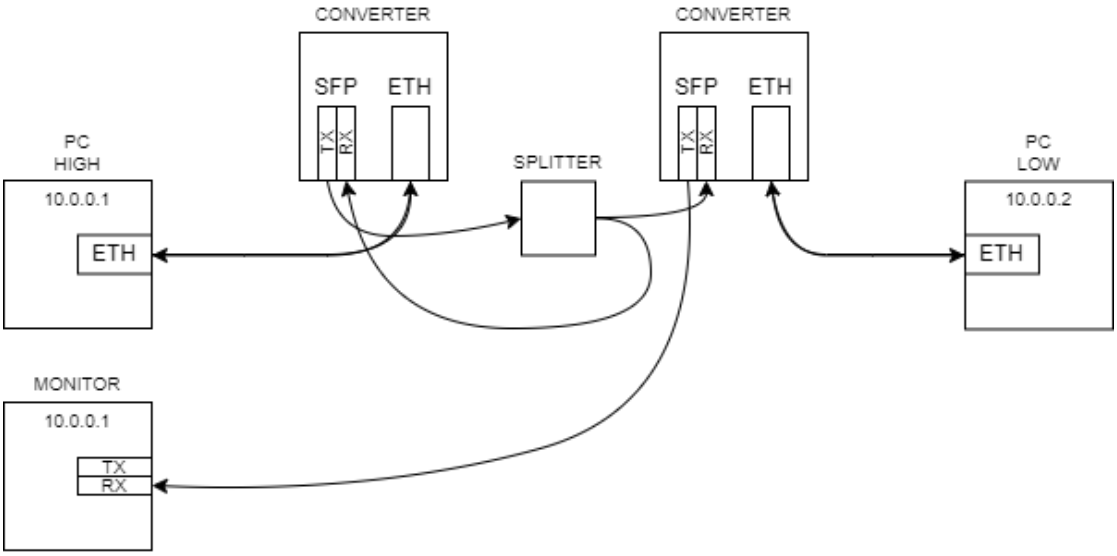
Alun perin työssä oli tarkoitus luoda myös installaatio Kyberturvallisuuslaboratorion Virtuallab-virtuaaliympäristöön ja soveltaa datadiodia suurempiin verkkokokonaisuuksiin. Testaus jäi kuitenkin melko vähäiseksi, mutta Virtuallabiin luotiin topologia, jossa tarkoituksena oli luoda kaksi eri turvallisuusluokan verkkoa ja yksisuuntaisen linkin molempiin päihin laitteet, jotka vastaisivat halutun liikenteen välityksestä, mutta pysyen mahdollisimman huomaamattomina loppukäyttäjän näkökulmasta.



Kuva 12. Virtuallabiin luotu topologia

Kuvassa 12 on esitelty Virtuallabiin luotu topologia. Kaikki laitteet paitsi Switch ovat Linux Ubuntu 22.04 -tietokoneita. HIGH ja LOW toimivat yhdyskäytävinä ja ne ohjaavat Client1:ltä lähtevää liikennettä haluttuun kohteeseen. Testissä käytettiin tässä työssä esiteltyjä menetelmiä tiedonsiirtoon ja Nginxiä liikenteen ohjaamiseen. Tämänkaltaista toteutusta voisi vielä jatkokehittää eri sovelluksiin ja mahdollisesti opetuskäyttöön.

Toinen jatkokehitysidea, joka syntyi työtä tehdessä, oli paluusignaalin monitorointi kolmannella laitteella. Idea syntyi testatessa multiplekseriä lähetyssignaalin jakamiseen, jolloin kokeiltiin myös kytkeä vastaanottavan laitteen lähettävään porttiin kuitu, joka johti kolmanteen laitteeseen. Tämä kolmas laite voisi sijaita lähettävän verkon puolella kuitenkin täysin irrallisena muista laitteista. Vastaanottavaan laitteeseen konfiguroitiin staattinen ARP-kartoitus, jossa lähettävän laitteen IP-osoite yhdistettiin kolmannen laitteen MAC-osoitteeseen. Näin esimerkiksi ping-vastaukset ohjautuivat kolmannelle laitteelle, jolta voitiin nyt monitoroida yhteyden toimivuutta. Kuvassa 13 on esimerkki, miten tämä voitaisiin toteuttaa kuitujakajan ja mediamuuntimien avulla.



Kuva 13. Fyysinen topologia monitorointilaitteella

Monitorointilaite oli tätä testatessa tietokone, jossa seurattiin saapuvia paketteja Wiresharkilla. Testi tehtiin vain pingaamalla, mutta ideana tätä voisi mahdollisesti kehittää pidemmälle. Ajatuksena vain saada jokin varmistus, että paketit menevät perille datadiodin läpi. Yksinkertaisimmillaan monitorointilaite voisi olla vaikka vain pieni laatikko, jossa välkkyy valo, kun paketteja saapuu.

Työn toteutusvaiheessa Kyberturvallisuuslaboratorioon tilattiin muutamia työhön tarvittavia osia, kuten mediamuuntimia, kuitujakaja ja liitäntäkotelo. Näitä osia tilattiin myös ylimääräisiä, joten Kyberturvallisuuslaboratoriosta löytyy tarvittavat osat useamman tässä työssä esitellyn datadiodin kasaamiseen esimerkiksi opetustarkoitukseen tai sovellettaviksi Kyberturvallisuuslaboratoriosta löytyviin laitteisiin ja verkkoihin.

LÄHTEET

Barry, C. 2012. Data Diodes for Cyber Security. TechSurveillance Magazine. PDF-dokumentti. Saatavissa: http://courtneybarry.com/Images/TS_Data_Diodes.pdf [viitattu 3.11.2022].

Bell, D. & LaPadula, L. 1973. Secure Computer Systems: Mathematical Foundations. PDF-dokumentti. Saatavissa: https://web.archive.org/web/20060618092351/http://www.albany.edu/acc/courses/ia/classics/belllapadula1.pdf [viitattu 4.10.2022].

Biba, K. 1975. Integrity Considerations for Secure Computer Systems. PDF-dokumentti. Saatavissa: https://seclab.cs.ucdavis.edu/projects/history/papers/biba75.pdf [viitattu 5.10.2022].

Castagna, R. 2021. Why Data Diodes Bolster IoT Security With One-Way Traffic. IOT World Today. WWW-dokumentti. Saatavissa: https://www.iotworldtoday.com/2021/09/27/why-data-diodes-bolster-iot-security-with-one-way-traffic/ [viitattu 3.11.2022].

FFmpeg. 2022. About FFmpeg. WWW-dokumentti. Saatavissa: https://ffmpeg.org/about.html [viitattu 15.7.2022].

Hewes, M. 2017. Diode-switch-config. Github. WWW-dokumentti. Saatavissa: https://github.com/mitcdh/diode-switch-config [viitattu 6.7.2022].

Jones, D. & Bowersox, T. 2016. Secure Data Export and Auditing using Data Diodes. The University of Iowa. Department of Computer Science. PDF-dokumentti. Saatavissa: https://www.usenix.org/legacy/event/evt06/tech/full_papers/jones/jones.pdf [viitattu 8.11.2022].

Kananen, J. 2017. Kehittämistutkimus interventiotutkimuksen muotona. Jyväskylä: Jyväskylän ammattikorkeakoulu.

Kang, M., Moskowitz, I. & Lee, D. 1996. A network pump. Washington: Naval Research Laboratory. PDF-dokumentti. Saatavissa: https://apps.dtic.mil/sti/pdfs/ADA465278.pdf [viitattu 5.7.2022].

Kertysova, K., Frinking, E. & Gricius, G. 2019. Understanding the strategic and technical significance of technology for security. The case of data diodes for cybersecurity. The Hague Secrity Delta (HSD). PDF-dokumentti. Saatavissa: https://securitydelta.nl/media/com_hsd/report/246/document/HSD-Rapport-Data-Diodes.pdf [viitattu 23.6.2022].

Maatkamp, M. 2015. Unidirectional Secure Information Transfer via RabbitMQ. University College Dublin. School of Computer Science and Informatics. Osa Pro Gradu -tutkielmaa. PDF-dokumentti. Saatavissa: https://arxiv.org/ftp/arxiv/papers/1602/1602.07467.pdf [viitattu 5.10.2022].

Menoher, J. & Mraz, R. 2009. Secure cross border information sharing using one-way data transfer systems. Owl Computing Technologies, Inc. PDF-dokumentti. Saatavissa: https://www.researchgate.net/publication/265232217_Secure_Cross_Border_Information_Sharing_Using_One-way_Data_Transfer_Systems [viitattu 23.6.2022].

OPSWAT. 2022a. Unidirectional Security Gateway and Data Diode Guide. WWW-dokumentti. Saatavissa: https://www.opswat.com/products/unidirectional-security-gateway-guide [viitattu 8.11.2022].

OPSWAT. 2022b. Using Unidirectional Security Gateways with MetaDefender Kiosk and Vault. WWW-dokumentti. Saatavissa: https://www.opswat.com/products/metadefender/using-unidirectional-security-gateways-kiosk-vault [viitattu 30.6.2022].

Owl Cyber Defence Solutions, LLC. 2018. The definite guide to data diode technologies. Owl Cyber Defence Solutions, LLC. E-kirja. Saatavissa: https://studylib.net/doc/25299178/data-diode-technologies [viitattu 23.6.2022].

Owl Cyber Defence Solutions, LLC. 2022. Why Owl data diodes? PDF-dokumentti. Saatavissa: https://owlcyberdefense.com/wp-content/uploads/2019/05/22-OWL-0133-Owl-Advantage-Why-Owl-V8-1.pdf [viitattu 4.7.2022].

Petersen, S. 2016. Why Data Diodes Are Essential for Isolated and Classified Networks. OPSWAT. WWW-dokumentti. Saatavissa: https://www.opswat.com/blog/why-data-diodes-are-essential-isolated-and-classified-networks [viitattu 8.11.2022].

RFC 768. 1980. User Datagram Protocol.

RFC 793. 1981. Transmission Control Protocol.

RFC 5905. 2010. Network Time Protocol Version 4: Protocol and Algorithms Specification.

Ribeiro, A. 2022. Implementation of data diodes can boost cybersecurity architecture at critical infrastructure installations. Industrial Cyber. WWW-dokumentti. Saatavissa: https://industrialcyber.co/analysis/implementation-of-data-diodes-can-boost-cybersecurity-architecture-at-critical-infrastructure-installations/ [viitattu 8.11.2022].

Scott, A. 2015. Tactical data diodes in industrial automation and control systems. SANS Institute. PDF-dokumentti. Saatavissa: https://sansorg.egnyte.com/dl/jcw5vWs4Df [viitattu 4.7.2022].

Stevens, M. 1999. An implementation of an optical data diode. Salisbury: DSTO Electronics and surveillance research laboratory. PDF-dokumentti. Saatavissa: https://apps.dtic.mil/sti/pdfs/ADA365579.pdf [viitattu 8.11.2022].

Vrolijk, R. 2022. OSDD. Github. WWW-dokumentti. Saatavissa: https://github.com/Vrolijk/OSDD [viitattu 6.7.2022].

Zetter, K. 2016. Inside the Cunning, Unprecedented Hack of Ukraine's Power Grid. Wired. WWW-dokumentti. Päivitetty 3.3.2016. Saatavissa: https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/ [viitattu 3.11.2022].