



Figuur 1 (Wärtsilä, 2020)

# OPEN SOURCE DATADIODE (OSDD)

*Hoe kan een open source datadiode bijdragen aan een hogere cybersecurity binnen de lager gerubriceerde domeinen van overheidsinstanties?*

Auteur(s)	Studenten Fontys ICT Semester 3: Kiggen, T.P.T.   Ligtenberg, J.M.   Oosten, W.E.J.   Ruijter, de, M.   Selst, van, D.   Verschuren, R.A.M.
Datum	16-1-2022
Versie	1.1
Status	Definitief
Opdrachtgever	Cyber Innovation Hub (Ministerie van Defensie)

## Samenvatting

Hoe kan een open source datadiode bijdragen aan een hogere cybersecurity binnen de lager gerubriceerde domeinen van overheidsinstanties? Om antwoord te geven op deze vraag zijn er een tweetal use-cases beschreven. Binnen defensie is het noodzakelijk dat locaties worden voorzien van een correcte tijd met een GPS-tijd module. Vanwege securitybeleid mogen de verschillende netwerken niet gekoppeld zijn aan elkaar op wat voor manier dan ook. Een datadiode kan een uitkomst bieden. Het is bij een datadiode enkel mogelijk om verkeer één kant op te laten stromen. Het is fysiek onmogelijk om de andere kant op te gaan met de gegevensstroom. Dit zorgt ervoor dat in het geval van de NTP-tijd use-case dat er meerdere netwerken potentieel gekoppeld kunnen worden aan dezelfde GPS-tijd module.

De tweede use-case is een onderzoek naar de werking van een datadiode in een Certificate Authority omgeving. Normaliter is het zo dat een Certificate Authority geen verbinding met het internet kan maken, om zo het uitlekken van de private key te kunnen voorkomen. Met het plaatsen van een datadiode in deze omgeving is het mogelijk om het gebruik van de Certificate Authority gebruiksvriendelijker en veiliger te maken.

Om een antwoord te kunnen geven op de hoofdvraag heeft de onderzoeks groep een workshop bijgewoond om kennis op te doen over wat een datadiode nu exact is. Door de opgedane kennis was het mogelijk om use-cases uit te werken in een lab. De resultaten hiervan zijn voorgelegd aan de opdrachtgever.

De resultaten bestaan uit een tweetal use-cases van het gebruik van een datadiode. Ten eerste de NTP-tijd use-case welke door middel van een NTP-broadcast kan worden verstrekt aan het netwerk achter een datadiode. Met de Certificate Authority use-case wordt het mogelijk om een certificaat aan te vragen op een veilige manier met een netwerkconnectie. Zonder het risico dat de Certificate Authority overgenomen wordt door kwaadwillende.

De conclusie van het onderzoek is dat een datadiode kan bijdragen aan een hoger niveau van cybersecurity binnen de lager gerubriceerde domeinen van overheidsinstanties. Dit kan door op locaties waar verschillende netwerken onderling niet verbonden met elkaar mogen zijn, te werken met een datadiode. Er dient rekening te worden gehouden met de mogelijkheden en onmogelijkheden van uni-directioneel verkeer. Hier is omheen te werken door per gebruikte techniek een oplossing te implementeren.

# Inhoud

Samenvatting .....	2
Inhoud .....	3
1. Inleiding.....	5
2. Opdracht .....	6
2.1      Opdrachtgever.....	6
2.2      Probleemstelling .....	6
2.3      Opdrachtomschrijving.....	6
3. Literatuuronderzoek .....	7
3.1      Wat is een datadiode? .....	7
3.2      Hoe werkt een datadiode?.....	7
3.3      Veiligheid.....	8
3.4      Inzetbaarheid .....	8
4. Onderzoeks methode .....	9
5. Use-Cases .....	10
5.1      Beveiligen Certificate Authority .....	10
5.1.1 <i>Haalbaarheidsonderzoek</i> .....	12
5.1.2 <i>Risicoanalyse</i> .....	13
5.1.3 <i>Security analyses</i> .....	16
5.1.4 <i>Kosten en baten</i> .....	19
5.1.5 <i>Advies</i> .....	26
5.2      Distribueren GPS-tijd atoomklok (NTP).....	28
5.2.1 <i>Haalbaarheidsonderzoek</i> .....	29
5.2.2 <i>RFC5905</i> .....	34
5.2.3 <i>Onderzoek met fysieke apparatuur</i> .....	35
5.2.4 <i>Advies</i> .....	37
6. Resultaten .....	39
6.1      Gerubriceerde domeinen.....	39
6.1.1 <i>Rubriceringen</i> .....	39
6.1.2 <i>Merkingen</i> .....	39
6.2      Voor- en nadelen.....	40

6.3	Beperkingen mitigeren .....	41
6.4	Risico's afdekken .....	42
6.5	Open source vs. commercieel .....	43
6.5.1	<i>Algemeen</i> .....	43
6.5.2	<i>Documentatie</i> .....	43
6.5.3	<i>Klantenservice</i> .....	43
6.5.4	<i>Security</i> .....	43
6.5.5	<i>Prijzen</i> .....	44
6.5.6	<i>Wie kiezen er het vaakst voor open source?</i> .....	44
6.5.7	<i>Model van Tracy en Wiersema</i> .....	44
7.	Conclusie .....	46
	Bibliografie .....	47
	Afkortingen .....	49
	Bijlage 1: Details Beveiligen Certificate Authority .....	50
	Bijlage 2: Details distribueren GPS-tijd atoomklok (NTP) .....	55
	Bijlage 3: Handleiding GNS3WIN-OSDD .....	63
	Bijlage 4: OSDD Test platform manual .....	79

## 1. Inleiding

Hedendaagse ICT-netwerken moeten beschikken over een hoge mate van cybersecurity. Vitale netwerken van ministeries, infrastructuur, banken en andere vitale overheidsprocessen vormen ideale doelwitten van hackers voor een cyberaanval.

Vaak stelt men de vraag of zij gehackt zullen worden, maar de vraag zou moeten zijn wanneer zij gehackt worden. Juist bij die vraag kan een **datadiode** uitkomst bieden en de cybersecurity drastisch verhogen.

Dit rapport is bedoeld als onderzoeksrapport en bevat praktische informatie over een (open source) datadiode. Er zijn een tweetal use-cases waarin de Open Source Data Diode (OSDD) gebruikt wordt en dienen als voorbeeld voor een bepaalde toepassing. Als eerste wordt er een literatuuronderzoek gedaan naar de werking van een datadiode. Hiervoor worden diverse artikelen/bronnen gebruikt welke te vinden zijn op internet. Bij het raadplegen is men selectief, er is gekeken naar betrouwbaarheid en gezocht naar overeenkomsten. Om het geheel goed te ondersteunen en betrouwbaar te maken is er gebruikt gemaakt van de Open Source Data Diode (OSDD) demonstrator van het Ministerie van Defensie (MoD) die de diode heeft ontwikkeld in samenwerking met Technolotion en de Haagse Security Delta. In de use-cases wordt er dieper ingegaan op de technische werking van een datadiode in combinatie met de toepassing en staan er adviezen beschreven die mogelijk kunnen helpen. Verderop in het onderzoek wordt behandeld wat gerubriceerde domeinen zijn en hoe deze te onderscheiden zijn. Hierna wordt in kaart gebracht wat voor- en nadelen zijn van een datadiode, wordt de impact van een open source vs. commerciële datadiode beschreven, welke beperkingen er zijn met het gebruik van een datadiode en hoe deze te mitigeren zijn en welk risico er afgedekt wordt door het gebruik ervan.

Het onderzoeksrapport is als volgt opgebouwd: hoofdstuk twee bevat informatie over de opdracht. Hierin zijn bijvoorbeeld de probleemstelling en de onderzoeks vragen te vinden. Hoofdstuk drie bestaat uit een literatuuronderzoek waarin bestaande theorieën en onderzoeken worden beschreven die zeer informatief zijn en achtergrondinformatie geeft over een datadiode. In hoofdstuk vier staat beschreven welke onderzoeks methodes er gehanteerd zijn voor het onderzoek. Hoofdstuk vijf beschrijft twee use-cases die zorgvuldig zijn uitgewerkt. Hoofdstuk zes bestaat uit de resultaten die antwoord geven op de onderzoeks vragen (deelvragen) en sommige daarvan hebben een verband met de use-cases. De uiteindelijke conclusie wordt vastgelegd in hoofdstuk zeven.

## 2. Opdracht

### 2.1 Opdrachtgever

De opdrachtgever voor deze challenge is de Cyber Innovation Hub van het Ministerie van Defensie. Om het projectteam te ondersteunen zijn er een tweetal coaches toegevoegd aan de challenge. De eerste coach (Fontys): dhr. L.R. van Bokhorst en tweede coach (Fontys): dhr. P.R.M.L. Dauphin.

### 2.2 Probleemstelling

De opdrachtgever heeft een Open Source Data Diode ontwikkeld maar constateert dat er in het publieke domein weinig kennis aanwezig is hoe deze technologie in te zetten. Om deze reden verzoekt de opdrachtgever om een aantal scenario's of use-cases uit te werken waaruit blijkt waar een datadiode toepasbaar is. Mogelijke uitkomsten kunnen zijn; projectplan en resultaatrapport en/of beschrijving van use-cases.

### 2.3 Opdrachtomschrijving

Er is een opdracht neergelegd met betrekking tot een datadiode. Wanneer een datadiode in een netwerk is geplaatst, gaat het verkeer nog maar één kant op. Data kan door de datadiode heen maar niet terug. Dit levert tal van mogelijkheden op om netwerken te beveiligen, maar ook uitdagingen.

Onderstaande onderzoeks vragen kunnen het onderzoek ondersteunen:

**Hoe kan een open source datadiode bijdragen aan een hogere cybersecurity binnen de lager gerubriceerde domeinen van overheidsinstanties?**

- Hoe werkt een datadiode?
- Wat zijn gerubriceerde domeinen en hoe zijn deze te onderscheiden?
- Welke voor- en nadelen zijn er voor het gebruik van een datadiode?
- Wat zijn de beperkingen van een datadiode en hoe worden deze gemitigeerd?
- Welk risico wordt er afgedekt door het gebruik van een datadiode?
- Wat is de impact van een open source vs. een commerciële datadiode?

Eerst worden er twee use-cases behandeld waarin toepassingen beschreven staan voor het gebruik van een datadiode. Daarna volgt een hoofdstuk met de antwoorden op de deelvragen. In dit onderzoek wordt gebruik gemaakt van het Model van Tracy en Wiersema voor waarde strategie. Sterktes, zwaktes, kansen en bedreigingen worden opgesomd in een SWOT-analyse en STRIDE voor om IT-security te identificeren. Na het uitgebreide onderzoek met bijpassend advies staat er een slot conclusie beschreven.

### 3. Literatuuronderzoek

In dit hoofdstuk staat informatie beschreven over een datadiode. Deze informatie is verkregen via diverse bronnen en is daarmee op basis van bestaande literatuur.

#### 3.1 Wat is een datadiode?

Een datadiode is een netwerkapparaat met als doel een veilige dataverbinding op te zetten tussen twee netwerk segmenten, waarbij data wel van A naar B kan stromen maar niet van B naar A. Dit zorgt voor een niet-routeerbare, volledig afgesloten eenrichtingsweg tussen netwerken (Crum, 2018). Bij het gebruik van een hardware matige datadiode wordt dit fysiek afgedwongen, het is dan onder geen enkele situatie mogelijk om bi-directioneel dataverkeer te krijgen. Dit werkt anders als bij voorbeeld een firewall, omdat deze softwarematig te manipuleren is en waar er door middel van o.a. regels en scanning (respectievelijk classic en next-gen firewalls) verkeer wordt toegestaan dan wel geblokkeerd.

Hier is het mogelijk dat er een softwarefout in de vorm van bijvoorbeeld een lek, zero-day, etc. toegang verschafft aan kwaadwillende ongeacht de correcte configuratie. Configuratie daarentegen is ook belangrijk. Een beheerder kan geen fouten maken met de configuratie van een datadiode. Daarnaast kan een insider ook niet bewust een achterdeur toevoegen. Dit met uitzondering van het fysiek verkeerd aansluiten van de datadiode. Een datadiode heeft deze zwakte niet aangezien er geen fysieke manier is om meer dan één kant verkeer op te sturen.

#### 3.2 Hoe werkt een datadiode?

Een standaard apparaat heeft ingebouwde systemen die het correct functioneren van een netwerk waarborgen. Een datadiode maakt dit moeilijker aangezien deze al direct op laag 1 van het OSI-model problemen oplevert, er is immers maar één kant op signalen (enkel RX of enkel TX). Een standaard netwerk apparaat detecteert dit als foutief en schakelt de relevante netwerkpoort uit.

Nu zou het natuurlijk mogelijk kunnen zijn om een nieuwe standaard te maken en te proberen om deze op zoveel mogelijk client devices te krijgen, maar in de praktijk is dit echter zeer lastig aangezien een datadiode slechts zeer selecte toepassingen heeft. Het verspreiden van een nieuwe standaard tot een niveau dat deze 'plug-and-play' is, is daarmee niet pragmatisch.

Daarom wordt deze aanpassing standaard afgehandeld op de datadiode zelf en verbind een client device via een conventionele bi-directionele verbinding (UTP, fiber). De client communiceert dan met een proxy op de datadiode die op een standaard gedefinieerde manier reageert. De proxy stuurt dan verkeer door naar de 'andere kant' of ontvangt verkeer van de 'andere kant' over het netwerk pad. Alle verwerking en non-standard zaken worden buiten de clients afgehandeld. De controle protocollen die bij dataverkeer worden gebruikt zorgen nog voor een verder probleem bij het verzenden via een datadiode, aangezien deze heel vaak (soms onverwachts) bi-directionele functionaliteit verwachten kan dit niet via een datadiode worden verzonden. Het gebruik van 'Protocol Breaks' lost dit op, een protocol break script het controle protocol van het netwerkverkeer en bouwt een geschikt controle protocol op aan de andere kant (The Hague Security Delta, 2020).

### 3.3 Veiligheid

Een netwerkverbinding waar een datadiode voor wordt gebruikt zal voornamelijk de link tussen een 'secure' en 'less secure' netwerk zijn. Deze discrepantie aan veiligheid en integriteit maakt het noodzakelijk dat er geen bi-directionele communicatie mogelijk is. Een voorbeeld hierin is het ontvangen van sensordata van een kritiek systeem, dit systeem staat in een streng beveiligd netwerk waar je een standaard 'gebruikers' netwerk geen volledige toegang aan wilt verschaffen. Wanneer er een verzendende datadiode aan de kant van het kritieke systeem staat is het volledig veilig de gegevens van de sensor uit te lezen op het minder veilige netwerk.

Een ander voorbeeld is het veilig opslaan van geheime documenten, wanneer er een verzendende datadiode in het niet veilige bron-netwerk staat waar het document is opgemaakt kan dit document veilig worden verzonden naar de 'high-security' opslag. Het is daarna niet meer mogelijk om opgeslagen documenten te filteren. Beide voorbeelden zijn natuurlijk uni-directioneel, dit zorgt er inherent voor dat protocollen zoals TCP niet functioneren en programma's en systemen die gebruik maken van sessies aangepast moeten worden, als dit überhaupt mogelijk is. Op dit moment zijn kosten een probleem die een datadiode in veel situaties onbruikbaar maakt, de gemiddelde prijs van €20.000 is niet in alle use-cases haalbaar en/of rendabel. (The Hague Security Delta, 2020).

### 3.4 Inzetbaarheid

Op dit moment bestaan er datadiodes die zich vooral focussen op hogere segment. Datadiodes zijn duur en worden mede daardoor alleen gebruikt in netwerken waarbij informatiebeveiliging een toprioriteit is (The Hague Security Delta, 2020). Er zijn op dit moment wel low-end en low-cost datadiodes maar deze zijn vaak ietwat houtje-touwtje en de vraag is hierbij of de veiligheid wel gewaarborgd is (Soullié, 2020).

## 4. Onderzoeksmethode

### Onderzoeksstrategieën

#### *Bieb (Library):*

- Best practices van fabrikanten
- Beschikbare literatuur (internet)
- Forumonderzoek

#### *Werkplaats (Workshop):*

- Brainstorm met medestudenten
- Brainstrom met collega's
- Online reviews
- Vergelijken

#### *Lab:*

- Oplossing bouwen
- Systeemtest
- Gebruikstest
- Beveiligingstest

### Methoden

Omdat er binnen het team nog weinig kennis was over het onderwerp; datadiode, is er eerst een kwalitatieve literatuurstudie gedaan. Hierbij is gelet op de kwaliteit van de bron die gebruikt is. Er is door andere partijen al wel onderzoek gedaan naar de datadiode en hier is veel waardevolle informatie uit gehaald. Hoewel de technologie nog in opkomst is, zijn er al wel artikelen te vinden met use-cases en potentiële toepassingen.

Door te experimenteren met de verschillende ideeën die binnen de groep speelde is er snel veel kennis vergaard. Deze kennis kon direct worden toegepast doordat

Door middel van een workshop gegeven door de opdrachtgever te Den Haag (Cyber Innovation Lab), hebben we meer informatie ontvangen over de datadiode. In deze workshop kwam naar voren; hoe een datadiode werkt, wat de mogelijk- en onmogelijkheden zijn en een aantal mogelijke toepassingen van de datadiode. Tijdens deze workshop heeft het onderzoeksteam instructies gekregen om een virtuele datadiode te bouwen op basis van Ubuntu. Op deze manier kon het team op een efficiënte manier use-cases onderzoeken en de theorie snel controleren in de praktijk.

### Afweging

De resultaten die zijn gekomen uit het literatuuronderzoek, zijn getest in de praktijk waar haalbaar. Hierdoor is het onderzoek gevalideerd door het team en zijn er niet zomaar zaken voor waarheid aangenomen. De uitkomsten van dit onderzoek en de staving aan de realiteit zijn helder beschreven in dit document.

## 5. Use-Cases

### 5.1 Beveiligen Certificate Authority

Het wereldwijde internet is ontzettend afhankelijk van het gebruik van SSL-certificaten, elke keer dat iemand een website bezoekt wordt er gecontroleerd of deze website legitiem is en of er geen sprake is van valse zaken. Om dit te doen wordt er gebruik gemaakt van SSL-certificaten, ook wel X.509 certificaten. Deze certificaten kunnen moeten worden ondertekend door een vertrouwde autoriteit, een zogenoemde 'Certificate Authority'.

De beveiliging van een Certificate Authority is ontzettend belangrijk, er hoeft maar één instantie te zijn waar kwaadwillende na binnen te komen om de integriteit van het gehele https-verkeer in gevaar te brengen. In Nederland is de meest bekende situatie waar dit verkeerd is gegaan de hack van DigiNotar in 2011 (Wolff, 2016) (Rhysider, 2017). Hackers wisten hierbinnen te dringen tot op de CA (Certificate Authority) servers die certificaten uit kunnen geven. Er zijn verschillende redenen waarom dit heeft kunnen gebeuren, o.a.:

- Out-of-date software.
- Configuratie fouten op de security appliances.
- Permanente plaatsing van fysieke security keys die enkel bij het uitgeven van certificaten geplaatst moesten worden.

Naast problemen met onderhoud van software en fouten in de configuratie legt de hack van DigiNotar een belangrijk aspect van beveiliging bloot; gemak is belangrijk. Het systeem van DigiNotar was 'fool-proof', de CA-server die certificaten voor klanten uit moest geven kon dit enkel doen wanneer er een fysieke sleutel in een beveiligde ruimte werd geplaatst, hierdoor kunnen kwaadwillende op afstand geen misbruik maken. Zo lang deze procedure wordt gevolgd.

Omdat de beveiligingsmethode van DigiNotar het inherent onmogelijk maakte om zeer snel nieuwe certificaten uit te geven kwam het steeds vaker voor dat de fysieke sleutel simpelweg in de server bleef zitten, in combinatie met de eerdergenoemde problemen zorgde dit uiteindelijk voor de kwetsbaarheid die de hack mogelijk maakte.

De situatie bij DigiNotar werd dus veroorzaakt door een mismatch in gebruikersgemak en beveiliging. Er is dus een nieuwe beveiligingsmethode nodig die het mogelijk maakt om een veilig certificate request uit te sturen en de resultaten te ontvangen en dit op een veilige manier te doen.

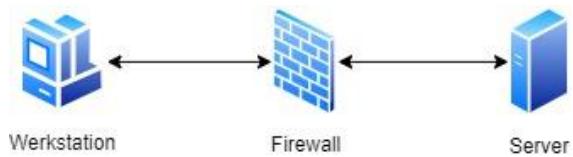
Er zijn 3 beveiligingsmethodes die we hier gaan behandelen:

- Firewall only
- Datadiode only
- Firewall en datadiode

- Beveiliging met een firewall

In deze mogelijkheid wordt er gebruik gemaakt van 1 firewall die het dataverkeer toelaat op basis van ingestelde regels. De firewall controleert het dataverkeer op laag 3 in het OSI-model en kijkt of het dataverkeer overeenkomt met de ingestelde regels. Wanneer dit het geval is wordt de data door gerouteerd naar de andere kant van de firewall, om zo bij de bestemde server te komen.

Het grote voordeel van een firewall is dat TCP-verkeer mogelijk is aangezien opgezette TCP-verbindingen terug de firewall uit mogen om de client te voorzien van de opgevraagde data. De data kan dus (indien toegestaan in de firewall rules) beide kanten op. Daarom zijn bijna alle protocollen te beheren doormiddel van een firewall. De beheerbaarheid van de firewall kan dus ook voor risico's zorgen.



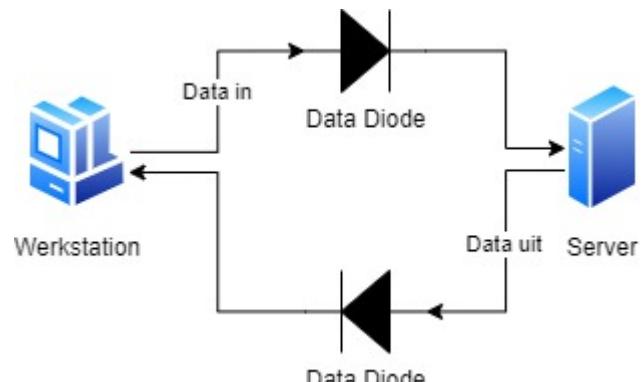
Figuur 2 Beveiliging met firewall

- Beveiliging met een datadiode

Wanneer je het netwerk gaat beveiligen met een datadiode wordt het al snel wat gecompliceerder.

Dit omdat je bij een verzoek om data minimaal 2 datadiodes nodig hebt omdat het verkeer alleen maar 1 kant op mag in de datadiode. Ook moet je de data op een andere manier gaan behandelen dan wanneer je de data door een firewall stuurt. Dit neemt mogelijk risico's met zich mee.

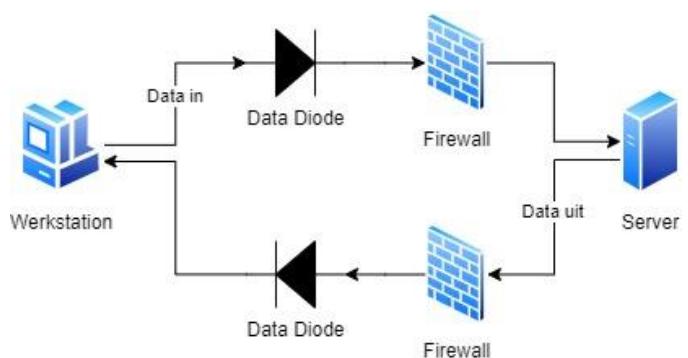
De inhoudelijke vraagstukken over het versturen van de data zullen later in een ander document verder toegelicht worden.



Figuur 3 Beveiliging met datadiode

- Beveiliging met een datadiode en een firewall

Bovenstaande mogelijkheden kun je ook combineren. Hierdoor krijg je het beste van beide mogelijkheden. Ook neem je eventuele risico's van de ene oplossing weg, omdat de andere een bepaald risico niet mogelijk maakt. Een groot nadeel hiervan is dat het snel complex en prijzig wordt aangezien je meer fysieke appliances nodig gaat hebben.



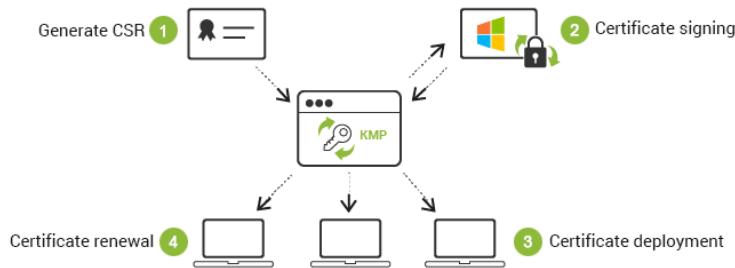
Figuur 4 Beveiliging met datadiode en firewall

### 5.1.1 Haalbaarheidsonderzoek

Bij de klassieke situatie loopt het dataverkeer enkel via een firewall, hierdoor kan het aanvragen van een certificaat op een traditionele wijze geconfigureerd worden door middel van bijvoorbeeld een web portaal.

Wanneer de gebruiker inlogt op het web portaal dan kan hij/zij hierop een CSR (Certificate Signing Request) sturen naar de Certificate Authority. Deze zal het CSR analyseren en daarna terugsturen.

Het certificaat is daarna te downloaden vanaf het web portaal.



**Figuur 5 CSR-verzoek en certificaat uitgifte (Manage Engine, Z.D.)**

Aangezien er bij de andere CA oplossingen gebruik gemaakt gaat worden van zowel een firewall en datadiode is het belangrijk dat de manier van het aanvragen van certificaten op dezelfde manier gebeurt. Aanvragen moeten via UDP verzonden worden naar de Certificate Authority server aangezien TCP-verkeer niet werkt via een datadiode. De aanvragen zullen verstuurd worden via UDP-SENDER en UDP-RECEIVER welke onderdeel zijn van de applicatie UDPCAST. De certificaten zullen uitgegeven worden door OpenSSL.

Er wordt onderzocht hoe er CSR-verzoeken via het SENDER segment verstuurd worden naar de Certificate Authority server. Door middel van bijvoorbeeld een cronjob op de server kan er vervolgens een certificaat worden gegenereerd en het resultaat worden teruggestuurd naar de requester of een andere relevante machine. Dit proces wordt ten alle tijden via UDP gestuurd vanwege de sessie loze aard van de datadiode.

Bij de laatste use-case wordt zowel een datadiode als firewall gebruikt om de data te routeren.

Het grote voordeel hierbij is dat het verkeer gefilterd en geanalyseerd kan worden voor dat het de server bereikt. Zo kan mogelijke kwaadaardige software/code de server niet bereiken.

Doormiddel van bijvoorbeeld IPS wordt bekende kwaadaardige software tegenhouden en kan het niet bij de server komen. Ook kunnen poorten door de firewall geblokkeerd worden om zo verkeer beperkt toe te staan, iets wat met alleen een datadiode alleen niet kan.

Verder worden eventuele backdoors op de firewall onklaar gemaakt door een datadiode. Aangezien de kwaadwillende geen verbinding terug kan opzetten om zo beheer te krijgen over de firewall. Ook bij deze use-case wordt het alleen mogelijk om gebruik te maken van UDP-SENDER aangezien, het verkeer door de OSDD alleen UDP-verkeer kan zijn.

### 5.1.2 Risicoanalyse

Bij de risicoanalyse wordt er gebruik gemaakt van onderstaande rating tabel:

Rating	Ernst
A	Zware schade
B	Aanzienlijke schade
C	Lichte schade
D	Verwaarloosbaar

Tabel 1 Risicoanalyse ernst overzicht

#### Beveiliging met een firewall

Risico	Rating	Mogelijk(e) gevaar(en)	Mogelijke oplossing(en)
Een kwaadwillende kan toegang verkrijgen tot de managementinterface van de firewall. (door middel van backdoors)	A	De kwaadwillende kan firewall rules/routes aanpassen zodat hij/zij toegang kan verkrijgen tot systemen achter de firewall.	Managementfuncties beperken en niet beschikbaar stellen via andere netwerken. De firewall mag alleen via een directe verbinding benaderd worden.
TCP hijacking op verbindingen die via de firewall laptop.	B	Een kwaadwillende kan onversleuteld TCP-verkeer hijacken zodat de informatie, die is opgevraagd onderschept kan worden.	Zo veel mogelijk gebruikmaken van versleutelde TCP-verbindingen.
Misconfiguratie van de firewall.	C	De beheerder van de firewall kan fouten maken in de configuratie van de firewall en hierdoor kunnen bepaalde onderdelen stoppen met werken.	<ul style="list-style-type: none"> <li>- Peer reviews van de geplande configuratie.</li> <li>- Configuratie testen in een acceptatieomgeving.</li> </ul>
Firmware updates voor beveiliging/functie updates.	D	De firewall moet om de zoveel tijd voorzien worden van de nieuwste firmware versie. Dit kan downtime veroorzaken in het netwerk.	De firewall updaten buiten gebruikstijden.
Analyse van verkeer door middel van een Intrusion Prevention System (IPS) in de firewall.	B	Wanneer bepaald verkeer gemarkerd wordt als (mogelijk) kwaadwillend verkeer door de IPS-applicatie kunnen bepaalde applicaties niet meer naar behoren werken.	<ul style="list-style-type: none"> <li>- Door middel van IPS om te zetten naar Intrusion Detection System (IDS), om zo verkeer alleen te monitoren.</li> <li>- De IPS-regels finetunen en meerdere keren testen voor ingebruikname.</li> </ul>
Fysieke toegang tot de appliance.	B	Als een kwaadwillende fysiek toegang krijgt tot de firewall dan kan hij/zij toegang krijgen tot de systemen achter de firewall.	De firewall kan in een beveiligde ruimte geplaatst worden zodat alleen bevoegde fysiek toegang hebben tot de firewall.
Een kwaadwillende kan toegang krijgen tot de server via SSH of RDP waardoor het mogelijk wordt om de private key te downloaden van de CA.	A	De kwaadwillende kan door middel van de private key informatie die versleuteld is met een ondertekend certificaat vanuit de CA, ontsleutelen en zo de versleutelde informatie lezen.	Op de server en firewall dienen alle remote managementpoorten geblokkeerd te zijn, zodat het niet mogelijk is om via het netwerk op de server in te loggen.

Tabel 2 Risicoanalyse gebruik firewall

### Beveiliging met een datadiode

Risico	Rating	Mogelijk(e) gevaar(en)	Mogelijke oplossing(en)
Verlies van data bij toevoer van grote hoeveelheden data door de datadiode.	B	Data kan verloren raken bij grote hoeveelheden data, omdat er geen check kan plaats vinden of de data ook ontvangen is.	- Snelheid reduceren - Forward error control
Het ontbreken van een statisch ARP-record na een reboot.	A	De data kan de dienst achter de datadiode niet vinden aangezien de dienst achter de datadiode geen ARP-verzoek terug kans sturen.	Het vast leggen van het statische ARP-record in het /etc/ethers bestand. Als er gebruik wordt gemaakt van unicast verkeer. Ook kan er gekozen worden om het verkeer te broadcasten over het netwerk. Dan hoeft er geen ARP-record aanwezig te zijn.
Fysieke toegang tot de appliance.	B	Als een kwaadwillende fysiek toegang krijgt tot de datadiode dan kan hij/zij toegang krijgen tot de systemen achter de datadiode. Ook is het mogelijk om de netwerkabels direct door te verbinden waardoor er bi-directioneel verkeer mogelijk is.	De datadiode kan in een beveiligde ruimte geplaatst worden zodat alleen bevoegde fysiek toegang hebben tot de datadiode.
Verkeer wordt niet gemonitord en daardoor kan al het verkeer naar de servers achter de datadiode	B	Het verkeer dat door de datadiode gaat wordt niet geanalyseerd op eventuele kwaadwillende software/code. Hierdoor kunnen systemen die achter de datadiode hangen mogelijk besmet worden met virus/malware.	Het plaatsen van 1 of 2 firewalls na de datadiodes zodat verkeer gemonitord kan worden en eventueel geblokkeerd.
Meerdere punten die kunnen uitvallen in het netwerk	B	Wanneer een van de diodes uitvallen is kan het verkeer niet meer naar de server gestuurd worden of ontvangen worden op de client. Hierdoor zal het aanvragen van een certificaat niet mogelijk zijn.	Het netwerk kan redundant uitgevoerd worden zodat het niet mogelijk is om data te verliezen tijdens het zenden of ontvangen.

Tabel 3 Risicoanalyse gebruik datadiode

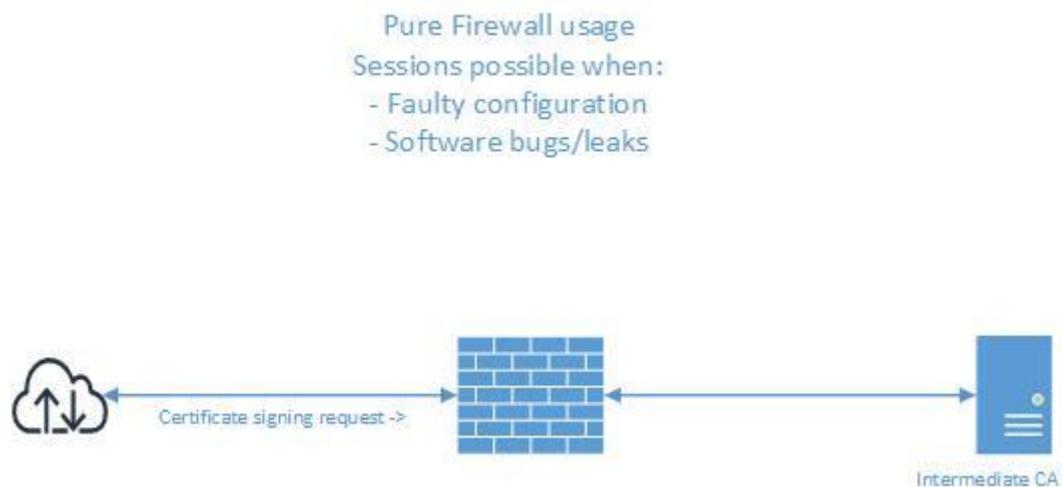
### Beveiliging met een firewall en datadiode

Risico	Rating	Mogelijk(e) gevaar(en)	Mogelijke oplossing(en)
Misconfiguratie van de firewall.	C	De beheerder van de firewall kan fouten maken in de configuratie van de firewall en hierdoor kunnen bepaalde onderdelen stoppen met werken.	<ul style="list-style-type: none"> <li>- Peer reviews van de geplande configuratie.</li> <li>- Configuratie testen in een acceptatieomgeving.</li> </ul>
Firmware updates voor beveiliging/functie updates.	D	De firewall moet om de zoveel tijd voorzien worden van de nieuwste firmware versie. Dit kan downtime veroorzaken in het netwerk.	De firewall updaten buiten gebruikstijden.
Analyse van verkeer door middel van een IPS in de firewall.	B	Wanneer bepaald verkeer gemarkerd wordt als (mogelijk) kwaadwillend verkeer door de IPS-applicatie kunnen bepaalde applicaties niet meer naar behoren werken.	<ul style="list-style-type: none"> <li>- Door middel van IPS om te zetten naar IDS, om zo verkeer alleen te monitoren.</li> <li>- De IPS-regels minder streng te zetten.</li> </ul>
Verlies van data bij toevoer van grote hoeveelheden data door de datadiode.	B	Data kan verloren raken bij grote hoeveelheden data, omdat er geen check kan plaats vinden of de data ook ontvangen is.	<ul style="list-style-type: none"> <li>- Data comprimeren</li> <li>- Optellende checks meesturen met de data.</li> </ul>
Het ontbreken van een statisch ARP-record na een reboot.	A	De data kan de dienst achter de datadiode niet vinden aangezien de dienst achter de datadiode geen ARP-verzoek terug kan sturen.	Het vast leggen van het statische ARP-record in het /etc/ethers bestand.
Meerdere punten die kunnen uitvallen in het netwerk	C	Wanneer een van de appliances uitvalt werkt het netwerk niet meer. Hierdoor kunnen systemen die achter de firewall en datadiode hangen niet meer werken.	Systemen redundant uitvoeren in het netwerk.
Fysieke toegang tot de appliances	B	Als een kwaadwillende fysiek toegang krijgt tot de firewall dan kan hij/zij toegang krijgen tot de systemen achter de firewall.	De firewall kan in een beveiligde ruimte geplaatst worden zodat alleen bevoegde fysiek toegang hebben tot de firewall.

Tabel 4 Risicoanalyse gebruik firewall en datadiode

### 5.1.3 Security analyses

#### Firewall-only (oude situatie)



Figuur 6 CSR door firewall-only oplossing

#### STRIDE

Spoofing	Laag risico, NG-firewalls hebben hier securitymaatregelen tegen.
Tampering	Bij correcte configuratie is dit enkel mogelijk bij software fouten en zero-days
Repudiation	Menselijk risico, security config op de firewall zelf wordt gewaarborgd d.m.v. logging, verkeer erdoorheen wordt niet aan gebruikers gekoppeld.
Information disclosure	Menselijk risico, een kwaadwillende in de organisatie kan configuratie lekken.
Denial of Service	Laag risico, Anti-DDoS functionaliteit op de firewall kan dit afschermen.
Elevation of privilege	Correcte toegang en software is vereist, zonder correcte configuratie en bij aanwezigheid van zero-days kan dit een probleem zijn.

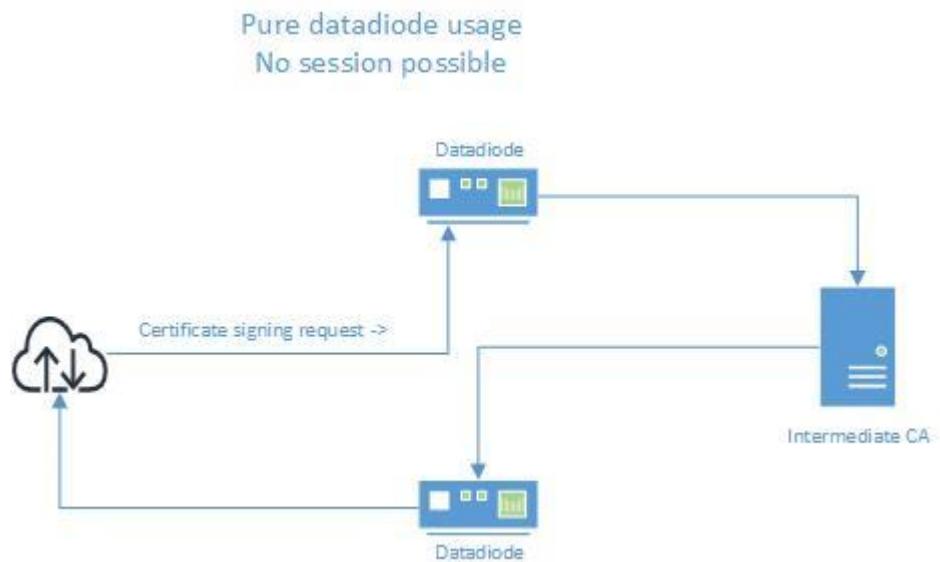
Tabel 5 Stride analyse firewall-only oplossing

#### SWOT

	Positief	Negatief
Extern	<b>Kansen:</b> <ul style="list-style-type: none"> <li>- Continue ontwikkelingen op security</li> </ul>	<b>Bedreigingen:</b> <ul style="list-style-type: none"> <li>- Kostbaar</li> <li>- Escalation of privileges</li> </ul>
Intern	<b>Krachten:</b> <ul style="list-style-type: none"> <li>- Gemakkelijk in opzet</li> <li>- Bekend en stabiel in gebruik</li> </ul>	<b>Zwakten:</b> <ul style="list-style-type: none"> <li>- Software bugs</li> <li>- Configuratie gevoelig</li> <li>- Niet standaard applicaties complex om te configureren</li> </ul>

Tabel 6 SWOT analyse firewall-only oplossing

## Datadiode-only



Figuur 7 CSR door datadiode-only oplossing

### STRIDE

Spoofing	Spoofing is mogelijk wanneer er kennis is van beide kanten, kwetsbaarheid afhankelijk van de server.
Tampering	Niet op netwerkbasis mogelijk, wel als er een kwetsbaarheid van de achterliggende server bekend is.
Repudiation	Niet inzichtelijk op de datadiode, enkel op de server die ervoor en achter zit.
Information disclosure	Laag risico, er is een fout/bug/lek in het request systeem nodig om data exfiltratie mogelijk te maken.
Denial of Service	Dit is mogelijk, er wordt niet tegen DDoS beschermd.
Elevation of privilege	Niet mogelijk, geen sessies mogelijk.

Tabel 7 Stride analyse datadiode-only oplossing

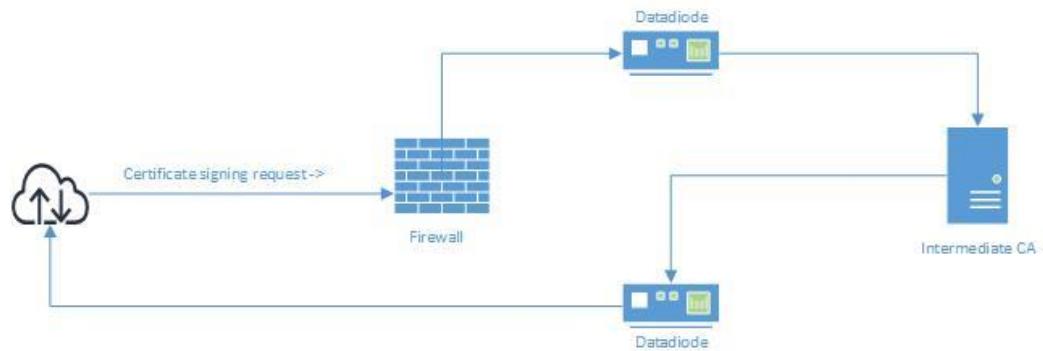
### SWOT

	Positief	Negatief
Extern	<b>Kansen:</b> <ul style="list-style-type: none"> <li>- Geen bugs</li> <li>- Niet gevoelig voor marktontwikkelingen</li> </ul>	<b>Bedreigingen:</b> <ul style="list-style-type: none"> <li>- Wanneer hoger gelegen systemen een lek hebben neemt de bescherming van de datadiode af</li> </ul>
Intern	<b>Krachten:</b> <ul style="list-style-type: none"> <li>- Geen escalation of privileges</li> <li>- Goedkoop (OSDD)</li> <li>- Uni-directioneel verkeer</li> </ul>	<b>Zwakten:</b> <ul style="list-style-type: none"> <li>- Geen actieve scanning en beveiliging</li> <li>- Complex voor opzet</li> </ul>

Tabel 8 SWOT-analyse datadiode-only oplossing

## Firewall en datadiode

Both Datadiode and Firewall  
 - No session possible  
 - Usage of rules for security



Figuur 8 CSR door firewall en datadiode combinatie

## STRIDE

Spoofing	Laag risico, NG-firewalls hebben hier securitymaatregelen tegen, daarnaast moet er na de spoof-poging rekening worden gehouden met de asynchrone aard van het systeem.
Tampering	Zeer lastig, op netwerkbasis niet mogelijk, enkel op de achterliggende server waar ook nog extra beveiliging door de firewall zit.
Repudiation	Menselijk risico, configuratie op de firewall wordt gewaarborgd d.m.v. logging, verkeer erdoorheen niet, een datadiode doet hier ook niets mee.
Information disclosure	Data exfiltratie niet mogelijk.
Denial of Service	Laag risico, Anti-DDoS functionaliteit op de firewall kan dit afschermen.
Elevation of privilege	Niet mogelijk, geen sessie mogelijk.

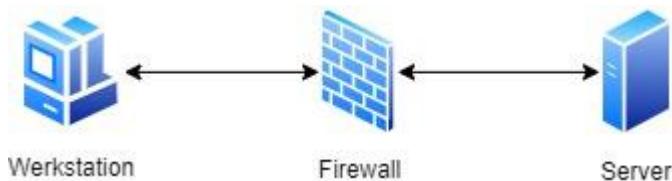
Tabel 9 STRIDE-analyse firewall en datadiode combinatie

## SWOT

	Positief	Negatief
Extern	<b>Kansen:</b> <ul style="list-style-type: none"> <li>Ook bij nieuwe marktontwikkelingen neemt de beveiliging niet snel af</li> </ul>	<b>Bedreigingen:</b> <ul style="list-style-type: none"> <li>Kostbaar</li> <li>Complex design</li> </ul>
Intern	<b>Krachten:</b> <ul style="list-style-type: none"> <li>Goede actieve en passieve beveiliging</li> <li>Continue ontwikkelingen op security</li> </ul>	<b>Zwakten:</b> <ul style="list-style-type: none"> <li>Configuratie gevoelig</li> <li>Sommige bugs</li> <li>Complex</li> </ul>

Tabel 10 SWOT Analyse firewall en datadiode combinatie

#### 5.1.4 Kosten en baten



**Figuur 9 Kosten-baten Firewall**

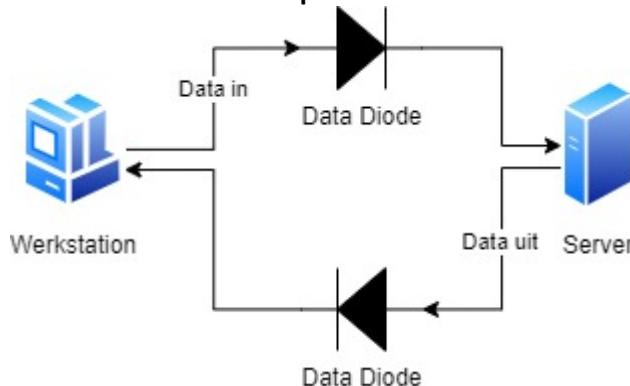
Rekening houden kosten Firewall:

- Aanschaf hardware
  - o Firewall
    - Voor de deze use-case is het aantal devices niet erg van belang, aangezien er achter de firewall alleen maar 1 server zit die de certificaten uitgeeft. Hierdoor is een mid-range firewall voldoende voor deze use-case. De gemiddelde prijs hiervan zit ongeveer rond de €600 à €800 euro. Eventuele licentie nog niet meegerekend (Bron: KommaGo).
  - o Server
    - Aangezien de server alleen certificaten hoeft uit te leveren is een eenvoudige server met minimale specificaties voldoende. Hierdoor kan de prijs beperkt blijven tot ongeveer €900 a €1100 euro. (Bron: Dell)
  - o Client
    - Om de certificaten aan te vragen is er een Client systeem nodig. Een handige toevoeging is een werkstation voor de certificaten. Dit kost tussen de €400 à €500 euro. (Bron: Dell) Hierbij heb je ook nog een werkstation nodig die alleen de certificaten doorstuurt. Dit client systeem hoeft ook niet meer te doen dan alleen certificaat verzoeken door te sturen naar de Certificate Authority server en uit eindelijk weer te ontvangen. Hierdoor kan dit systeem ook beperkt zijn in specificaties. En werkstation dat voldoet aan de specificaties kost ongeveer tussen de €400 a €500 euro. (Bron: Dell)
- Personeel
  - o Investering kennis
    - Voor het juist configureren van de firewall is het belangrijk dat het personeel kundig is. Hierdoor is het mogelijk nodig om personeel extra op te leiden. De kosten voor een training in firewalls configureren zitten op ongeveer €1200 euro per persoon. (Bron: ictivitytraining.nl)
  - o Urloon
    - Het configureren van het netwerk kost tijd het gemiddeld urloon van een netwerkbeheerder zit zo tussen de €20 a €25 euro per uur. (Bron: ictergezocht.nl)
- Servicecontract/Licentiekosten
  - o Om de continuïteit van de firewall te kunnen waarborgen is het belangrijk dat er een service contract zit op de hardware. Hierdoor ben je verzekerd van goedwerkende hardware. Deze contracten worden vaak per jaar gerekend. In dit geval is het aan te raden om voor alle hardware in het netwerk een servicecontract af te sluiten.

- Voor de firewall zijn de kosten voor zowel de licenties voor de ingebouwde tools als de hardware ondersteuning in 1 pakket gebundeld. De kosten hiervan zitten op €340 euro per jaar. (Bron: KommaGo) Dit is exclusief inbraakdetectiesysteem (IDS) licentie. Als je een IDS licentie erbij wilt hebben dan kost dat €420 per jaar inclusief IDS.
- Voor de server dient er ook een service contract afgesloten te worden. Met dit contract komt de fabrikant binnen 4 uur langs om het systeem te repareren/te vervangen. De kosten voor 4 jaar zitten op €200 euro per jaar. (Bron: Dell)
- Het werkstation is minder curiaal dit omdat deze intern snel vervangen kan worden. Maar ook hier is het verstandig om een extra service contract af te sluiten. Deze zit op €130 euro per 4 jaar met Next Business day ondersteuning. (Bron: Dell)
- Beveiliging serverruimte

Om de netwerk/serverruimte goed te beveiligen is het belangrijk de toegang tot de ruimte goed geregeld is. Doormiddel van cijferslot deurbeslag is de ruimte goed te beveiligen. Iedereen die toegang mag hebben tot de ruimte krijgt een eigen code. De kosten van dit deurbeslag zit op €335 euro. (Bron: stedaparts.nl)

#### Kosten-baten van een opensource en commerciële datadiode



Figuur 10 datadiode voor in- en uitgaand verkeer

#### Kosten commerciële datadiode

Rekening houden kosten datadiode:

- Aanschaf hardware
- Datadiode 2x
  - Voor deze use-case heb je twee datadiodes nodig om de data goed te laten functioneren. Er zijn nu twee verschillende datadiodes die je commercieel kan kopen bij een commercieel bedrijf. Dat zijn FOX-IT en Technolution. Bij FOX-IT is een datadiode met full package ongeveer €20.000 euro. Hier zitten dan ook licentiekosten en het servicecontract bij. Bij Fox-IT kunnen ze de prijs van alleen de hardware van de datadiode niet vertellen omdat dit commercieel vertrouwelijk is. Bij Technolution is een datadiode €2975 euro. (Bron: Technolution) Dit is wel voor alleen de hardware van de datadiode.

- Server
  - Aangezien de server alleen certificaten hoeft uit te leveren is een eenvoudige server met minimale specificaties voldoende. Hierdoor kan de prijs beperkt blijven tot ongeveer €900 a €1100 euro. (Bron: Dell)
- Client
  - Om de certificaten aan te vragen is er een Client systeem nodig. Een handige toevoeging is een werkstation voor de certificaten. Dit kost tussen de €400 à €500 euro. (Bron: Dell) Hierbij heb je ook nog een werkstation nodig die alleen de certificaten doorstuurt
  - Dit client systeem hoeft ook niet meer te doen dan alleen certificaat verzoeken door te sturen naar de Certificate Authority server en uit eindelijk weer te ontvangen.
  - Hierdoor kan dit systeem ook beperkt zijn in specificaties.
  - En werkstation dat voldoet aan de specificaties kost ongeveer tussen de €400 à €500 euro. (Bron: Dell)
- Personeel
- Investering kennis
  - De juiste kennis van een datadiode is exceptioneel voor het beheren van een datadiode. Hiervoor heeft iemand je juiste kennis van nodig. Omdat een datadiode nog vrij nieuw is kan het zo zijn dat de opleidingskosten hoger liggen dan bijvoorbeeld bij een firewall cursus/opleiding. Een firewall cursus is ongeveer €1200 euro per persoon. (Bron: ictivitytraining.nl)
- Uurloon
  - Het configureren van het netwerk kost tijd het gemiddeld uurloon van een netwerkbeheerder zit zo tussen de €20 a €25 euro per uur. (Bron: ictergezocht.nl) Dit is ongeveer een indicatie van verschillende factoren. In de werkelijkheid zal dit bedrag hoger liggen.
  - Ook is het handig om een beheerder/ontwikkelaar te hebben die de proxy's onderhoud van een datadiode omdat dit vrij specifieke handelingen zijn. Ook hier zal het salaris tussen de €20 a €25 euro per uur zijn.
- Servicecontract/Licentiekosten
- Bij een commerciële datadiode van FOX-IT zit alles inbegrepen in de totale kosten. Na nader onderzoek heb ik niet de losse licentiekosten of servicecontracten kunnen vinden bij FOX-IT. Je kan wel een indicatie maken door het verschil met Technolution te vergelijken. De kosten van een datadiode pakket bij FOX-IT is €20.000 euro.
- Beveiliging serverruimte
- Om de netwerk/serverruimte goed te beveiligen is het belangrijk de toegang tot de ruimte goed geregeld is. Doormiddel van cijferslot deurbeslag is de ruimte goed te beveiligen. Iedereen die toegang mag hebben tot de ruimte krijgt een eigen code. De kosten van dit deurbeslag zit op €335 euro. (Bron: stedaparts.nl)

### Kosten opensource datadiode

Rekening houden kosten datadiode:

- Aanschaf hardware
- o Datadiode 2x
  - De OSDD is een stuk goedkoper dan een commerciële datadiode. Dit komt vooral omdat je opensource software gebruikt of helemaal gratis is. De hardware van een datadiode goed verkrijgbaar. Op GitHub staan verschillende mogelijkheden hoe je een datadiode makkelijk in elkaar zet. Bijvoorbeeld met 2 Raspberry pi's, 2 TP-link switches & development board. (Zie afbeelding 11). Dit kan je gebruiken als een opensource datadiode. Hoe een daadwerkelijke datadiode eruit komt te zien is nog niet helemaal duidelijk. Wel kunnen we ervan uitgaan dat een opensource datadiode niet meer gaat kosten dan €300 euro voor alleen de hardware. Bij Defensie heeft men als test opstelling van een opensource datadiode 2 laptops en een datadiode demonstrator staan. De OSDD (figuur 12) bestaat alleen maar uit een FPGA-development board met 2 ethernet aansluitingen waarbij de FPGA een minimale instructieset heeft om verkeer door te sturen.



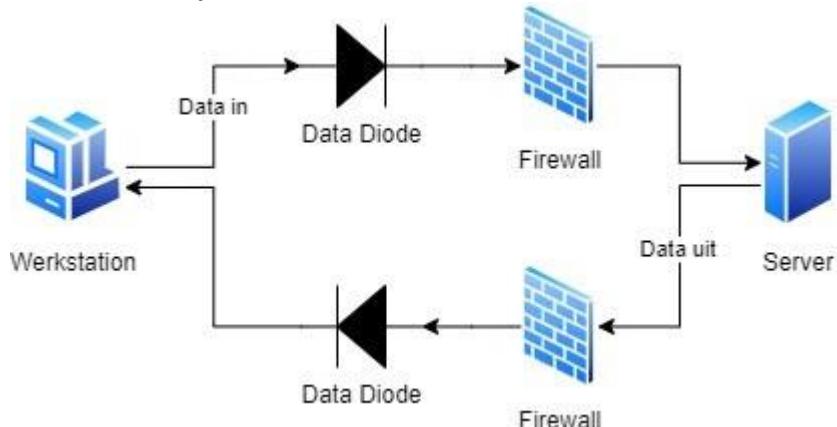
Figuur 12 "DIY"-datadiode



Figuur 11 OSDD

- Servicekosten
- o Voor de OSDD zijn er nu nog geen leveranciers van service contracten. Dus de kosten hiervan zijn ook nog niet bekend. De Security Delta (HSD) heeft geholpen met het ontwikkelen van de demonstrator van de OSDD. Wanneer de OSDD gereleaset wordt vanuit het Cyber Innovation Hub van defensie zal deze voor verschillende bedrijven de mogelijkheid geven hier hun eigen support op te gaan bieden, aangezien het een opensource product is.

### Kosten-baten opensource datadiode & Firewall



Figuur 13 Datadiode in combinatie met firewall

Rekening houden kosten datadiode & Firewall:

- Aanschaf hardware

- Datadiode 2x
  - Voor deze use-case heb je twee datadiodes nodig om de data goed te laten functioneren. Zoals je in bovenstaand hoofdstuk hebt gezien kun je voor ongeveer €300 euro een datadiode maken. Voor deze use-case wordt dat dan €600 euro in totaal voor de hardware van de datadiode.
- Firewall
  - Aangezien het verkeer twee kanten op moet gaan zijn er ook twee firewalls nodig. Dit omdat de zowel het binnenkomende als uitgaande verkeer geanalyseerd dient te worden. Het verkeer dat door het netwerk gaat is erg minimaal qua hoeveelheid. Hierdoor is een mid-range firewall voldoende voor deze use-case. De gemiddelde prijs per stuk hiervan zit ongeveer rond de €600 à €800 euro. Eventuele licentie nog niet meegerekend (Bron: KommaGo). Je kunt er ook voor kiezen om maar één firewall te gebruiken omdat een firewall meestal vier of meer interfaces heeft.
- Server
  - Aangezien de server alleen certificaten hoeft uit te leveren is een eenvoudige server met minimale specificaties voldoende. Hierdoor kan de prijs beperkt blijven tot ongeveer €900 à €1100 euro. (Bron: Dell)
- Client
  - Om de certificaten aan te vragen is er een Client systeem nodig. Een handige toevoeging is een werkstation voor de certificaten. Dit kost tussen de €400 a €500 euro. (Bron: Dell) Hierbij heb je ook nog een werkstation nodig die alleen de certificaten doorstuurt.  
Dit client systeem hoeft ook niet meer te doen dan alleen certificaat verzoeken door te sturen naar de Certificate Authority server en uit eindelijk weer te ontvangen.  
Hierdoor kan dit systeem ook beperkt zijn in specificaties.  
En werkstation dat voldoet aan de specificaties kost ongeveer tussen de €400 à €500 euro. (Bron: Dell)
- Personeel
- Investering kennis
  - Voor het juist configureren van de firewall is het belangrijk dat het personeel kundig is. Hierdoor is het mogelijk nodig om personeel extra op te leiden. De kosten voor een training in firewalls configureren zitten op ongeveer €1200 euro per persoon. (Bron: ictivitytraining.nl)  
Een opleiding/training voor de OSDD is er momenteel niet. De kosten voor het opdoen van de kennis zijn dus gelijk aan het uurloon van de beheerders.
- Uurloon
  - Het configureren van het netwerk kost tijd het gemiddeld uurloon van een netwerkbeheerder zit zo tussen de €20 à €25 euro per uur. (Bron: ictergezocht.nl)
  - Ook is het handig om een beheerder/ontwikkelaar te hebben die de proxy's onderhoud van een datadiode, omdat dit vrij specifieke handelingen zijn. Ook hier zal het salaris tussen de €20 à €25 euro per uur zijn.
- Servicecontract/Licentiekosten
- Om de continuïteit van de firewall te kunnen waarborgen is het belangrijk dat er een service contract zit op de hardware. Hierdoor ben je verzekerd van goedwerkende hardware.  
Deze contracten worden vaak per jaar gerekend.

In dit geval is het aan te raden om voor alle hardware in het netwerk een servicecontract af te sluiten.

Voor de firewall zijn de kosten voor zowel de licenties voor de ingebouwde tools als de hardware ondersteuning in 1 pakket gebundeld. De kosten hiervan zitten op €340 euro per jaar. (Bron: KommaGo) Dit is exclusief inbraakdetectiesysteem (IDS) licentie. Als je een IDS licentie erbij wilt hebben dan kost dat €420 per jaar inclusief IDS.

Voor de server dient er ook een service contract afgesloten te worden. Met dit contract komt de fabrikant binnen 4 uur langs om het systeem te repareren/te vervangen. De kosten voor 4 jaar zitten op €200 euro per jaar. (Bron: Dell)

Het workstation is minder curiaal dit omdat deze intern snel vervangen kan worden. Maar ook hier is het verstandig om een extra service contract af te sluiten. Deze zit op €130 euro per 4 jaar met Next Business day ondersteuning. (Bron: Dell)

Voor de OSDD zijn er nu nog geen leveranciers van service contracten. Dus de kosten hiervan zijn ook nog niet bekend. Wanneer de OSDD vanuit The Hague Security Delta beschikbaar gesteld zal worden zullen er verschillende bedrijven hier support op gaan aanbieden, aangezien het een opensource product is.

- Beveiliging serverruimte
- o Om de netwerk/serverruimte goed te beveiligen is het belangrijk de toegang tot de ruimte goed geregeld is. Doormiddel van cijferslot deurbeslag is de ruimte goed te beveiligen. Iedereen die toegang mag hebben tot de ruimte krijgt een eigen code. De kosten van dit deurbeslag zit op €335 euro. (Bron: stedaparts.nl)

### Conclusie kosten en baten

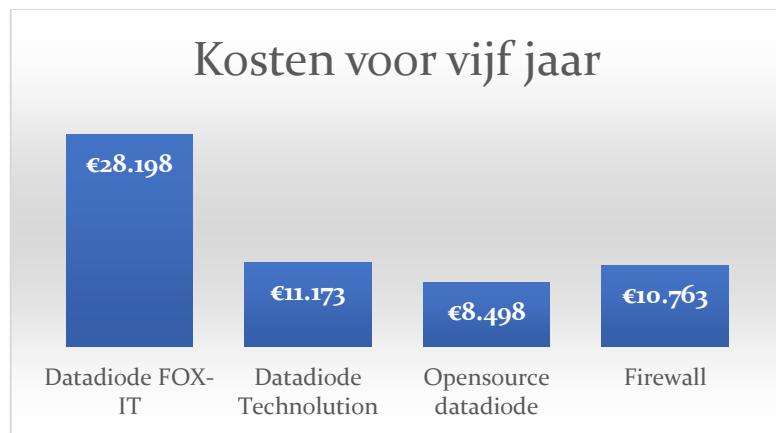
Zoals hier in onderstaand tabel kan zien met alle kosten bij elkaar die voor deze use-cases belangrijk zijn. Deze zijn per beveiligings use-case uitgewerkt. Je kunt hier zien dat de OSDD in zijn totaliteit een stuk goedkoper is dan de andere oplossingen.

#### Indicatie van de kosten per use-case

Product	Eenmalige kosten	Jaarlijkse kosten	Personelekosten
Datadiode FOX-IT	<ul style="list-style-type: none"> <li>• <b>Aanschaf datadiode:</b> €20.000 euro met proxy's.</li> <li>• <b>Aanschaf fysieke beveiliging:</b> €335 euro +/-</li> </ul>	Is niet beschikbaar (Commercieel vertrouwelijk)	<ul style="list-style-type: none"> <li>• <b>Opleiding:</b> Hoger dan €1200 per persoon</li> <li>• <b>Uurloon (40 uur werk week):</b> €20 á €25 euro per uur</li> </ul>
Datadiode Technolution	<ul style="list-style-type: none"> <li>• <b>Aanschaf datadiode:</b> €2975 euro zonder proxy's</li> <li>• <b>Aanschaf fysieke beveiliging:</b> €335 euro +/-</li> </ul>	Is niet beschikbaar	<ul style="list-style-type: none"> <li>• <b>Opleiding:</b> Hoger dan €1200 per persoon</li> <li>• <b>Uurloon (40 uur werk week):</b> €20 á €25 euro per uur</li> </ul>

Opensource datadiode	<ul style="list-style-type: none"> <li><b>Aanschaf datadiode:</b> €300 euro +/-</li> <li><b>Aanschaf fysieke beveiliging:</b> €335 euro +/-</li> </ul>	De opensource datadiode is nog volop in ontwikkeling. Hier zit ook nog geen support op vanuit bedrijven.	<ul style="list-style-type: none"> <li><b>Opleiding:</b> Hoger dan €1200 per persoon</li> <li><b>Uurloon (40 uur werk week):</b> €20 á €25 euro per uur</li> </ul>
Firewall	<ul style="list-style-type: none"> <li><b>Aanschaf firewall:</b> €600 á €800 euro</li> </ul>	<ul style="list-style-type: none"> <li><b>Licentie firewall incl. IDS:</b> €420 euro per jaar</li> </ul>	<ul style="list-style-type: none"> <li><b>Opleiding:</b> €1200 per persoon +/-</li> <li><b>Uurloon (40 uur werk week):</b> €20 á €25 euro per uur</li> </ul>
Server	<ul style="list-style-type: none"> <li><b>Aanschaf OpenSSL Server:</b> €900 á €1100 euro</li> </ul>	<ul style="list-style-type: none"> <li><b>OpenSSL Server licentie:</b> €200 per jaar</li> </ul>	
Client	<ul style="list-style-type: none"> <li><b>Aanschaf Client:</b> €400 á €500 euro</li> </ul>	<ul style="list-style-type: none"> <li><b>Client licentie:</b> €130 euro per 4 jaar</li> </ul>	

Tabel 11 Indicatie van de kosten per use-case



Tabel 1122 grafiek Indicatie van de kosten voor vijf jaar

#### Verschil kosten commerciële datadiode vs. opensource datadiode

Het verschil in kosten van een commerciële datadiode en een opensource datadiode kun je in bovenstaand tabel zien. Zoals je kunt zien betaal je ongeveer voor de opensource datadiode €8.498 euro per vijf jaar. Hier zit nog geen jaarlijkse kosten aan vast omdat de opensource datadiode nog volop in ontwikkeling is. De software/licenties zijn hiervoor ook gratis omdat deze opensource zijn.

De commerciële datadiode van FOX-IT komt op het duurst uit. Deze bedraagt €28.198 euro per vijf jaar. Dit heeft te maken met de proxy's die bij de datadiode zitten en de maatregelen die genomen zijn om in hoog gerubriceerde omgevingen te mogen gebruiken. Hierover is contact geweest met FOX-IT, maar is helaas is er geen

prijsopgave verkregen van alleen de hardware (datadiode) los, omdat dit commercieel vertrouwelijke informatie is.

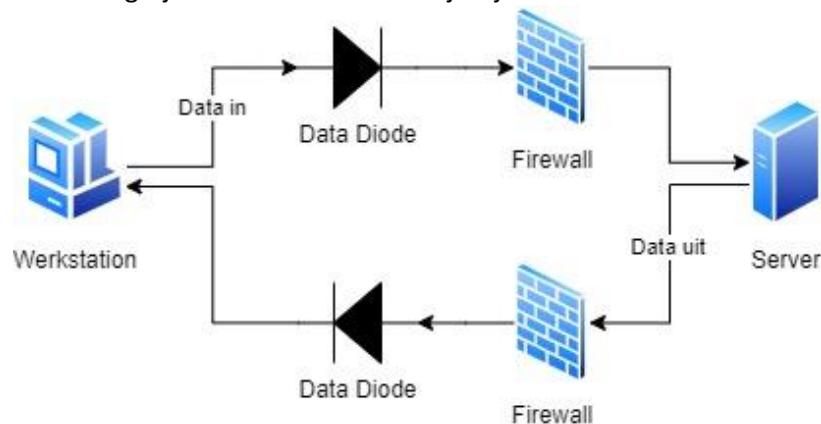
De commerciële datadiode van Technolution is redelijk te vergelijken met de opensource datadiode. De totaalprijs van de datadiode van Technolution is €11.173 euro per vijf jaar. Hier moet men wel rekening mee houden dat hier de datadiode zonder proxy's is.

Je ziet dat de kosten niet veel van elkaar verschillen opensource of commercieel. Dit komt ook mede omdat men niet precies de kosten kunnen berekenen wat het in de werkelijkheid is. Wel zie je dat de datadiode los (alleen hardware) de opensource een stuk goedkoper is dan een commerciële datadiode.

### 5.1.5 Advies

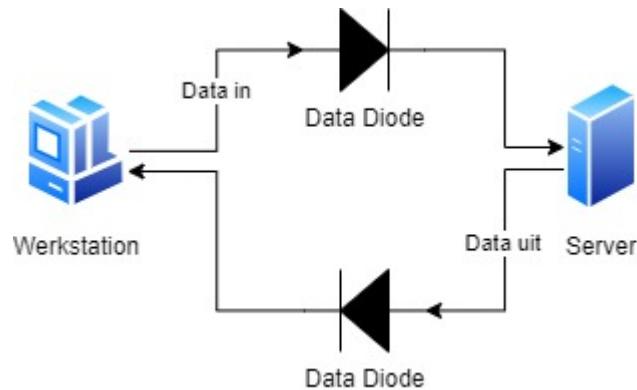
Gebaseerd op de analyses die zijn uitgevoerd op de situatie blijkt het, zoals helaas vaker, dat de verstandigste implementatie erg afhankelijk is van de situatie. De meest aantrekkelijke keuze is voornamelijk afhankelijk van het risico waarvan er sprake is.

- Grote partij. Wanneer er een implementatie wordt ontworpen voor een grote partij zoals een publieke certificate authority is het volledig inzetten van zowel een (Next-Gen) firewall en een datadiode de verstandigste keuze. In een dergelijke situatie zijn er veel middelen beschikbaar waardoor veel van de nadelen zoals kosten en implementatie complexiteit niet of verminderd van belang zijn. De extra kosten en tijd zijn de reductie van risico waard.



Figuur 14 Datadiode in combinatie met firewall

- Kleine partij of interne CA. Bij veelal simpelere omgevingen zoals een interne CA is extreem complexe beveiliging vaak minde van belang, dikwijls zijn er in de eerste plaats weinig tot geen network-level beveiligingsimplementaties aanwezig. Een dergelijke situatie is daarop weer vrij goed te beveiligen met enkel een datadiode, zoals in de bijlage uiteen is gezet is het volstrekt mogelijk om een implementatie effectief in te zetten.



**Figuur 15 datadiode voor in- en uitgaand verkeer**

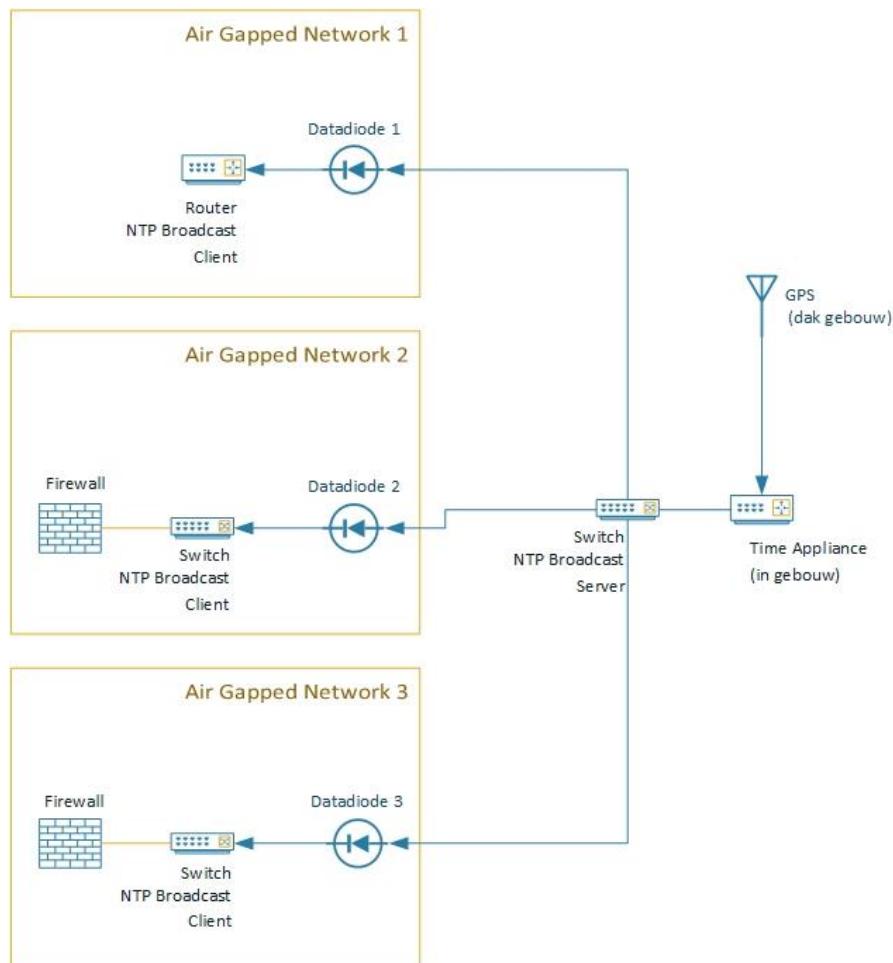
Het principe (en daarmee ook de ‘Best practices’) van een datadiode is simpel, bij een firewall is er meer sprake van “overhead” in mogelijkheden, daarom is het van belang om goed in te zien met wat voor situatie er wordt gewerkt alvorens de beslissing van product(combinatie) te maken.

## 5.2 Distribueren GPS-tijd atoomklok (NTP)

Eén van de twee use-case die in dit onderzoek wordt behandeld is het distribueren van een gps-tijd (NTP) naar meerdere air-gapped netwerken. Het gebruik van een GPS-klok met NTP geeft een aantal voordelen:

- Elk systeem binnen het netwerk synchroniseert met dezelfde klok;
- Integriteit van timestamps van bijvoorbeeld logfiles wordt gegarandeerd;
- IoT-apparatuur binnen hetzelfde netwerk synchroniseert met dezelfde klok wat fouten voorkomt;
- Synchroniseren van IT-functionaliteiten verloopt beter en sommige applicaties vereisen een atoomklok voor correcte werking;
- Netwerkbeveiliging is erg afhankelijk van tijd;
- Debuggen of beheer van systemen is beter beheersbaar.

Belangrijk om te weten is dat voor deze voorbeeld use-case netwerken zich in een gebouw bevinden waar op het dak van het gebouw maar één ingang is voor een GPS-signalen. De verschillende netwerken mogen niet aan elkaar gekoppeld worden en daarvoor biedt een datadiode uitkomst. Om een beeld te krijgen hoe dit er ongeveer uit ziet staat hieronder (figuur 16) het functioneel ontwerp weergegeven.



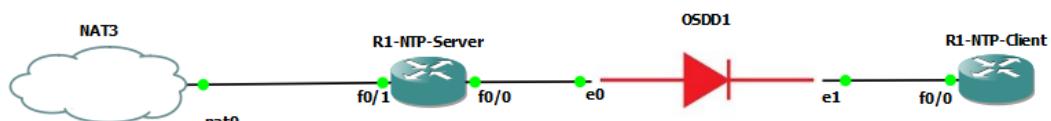
Figuur 16 Functioneel ontwerp voorbeeld

### 5.2.1 Haalbaarheidsonderzoek

Om te kijken of het geheel haalbaar is en uiteindelijk kan gaan werken is e.e.a. uitgezocht door opstellingen virtueel te bouwen. Om gps-tijd te distribueren wordt er gebruik gemaakt van het 'Network Time Protocol' (NTP). Dit protocol is een server/client applicatie en als deze op de normale manier werkt dan vraagt een client de tijd aan de server. Met het gebruik van een datadiode in een netwerk wordt éénrichtingsverkeer (uni-directioneel) gegarandeerd en daardoor kan een client onmogelijk de tijd opvragen bij de server. Om dit probleem te tackelen is er getest met broadcasting. Broadcast- en multicast mode zijn speciaal, omdat bij deze modus de NTP-server de synchronisatie informatie stuurt naar die clients. Broadcasting wil zeggen dat een datapakket wordt gestuurd naar elk aangesloten apparaat op hetzelfde netwerk segment. Multicast vereist dat clients multicast functionaliteiten geconfigureerd hebben en daardoor is het meer selectief. Een multicast pakket kan worden ontvangen door een ontvanger die het pakket wil hebben en dat is dus feitelijk anders dan bij een broadcast waar elk apparaat sowieso het pakket (binnen het segment) ontvangt. Echter werkt NTP-multicast (multicast verkeer gaat WEL door de OSDD) niet in combinatie met de OSDD, omdat dit hetzelfde werkt als traditioneel NTP-verkeer en valt daarmee buiten de scope.

Voor het testen is er gebruik gemaakt van twee verschillende netwerksoftware emulators: EVE-NG en GNS3. Aangezien EVE-NG niet helemaal soepel liep en de mogelijkheden in GNS3 voor het project sneller te vinden zijn (bijv. tracen verbindingen m.b.v. Wireshark) en er meer documentatie beschikbaar is, is er besloten de virtuele omgeving te beperken tot GNS3. Naast het gebruik van deze software is er een test uitgevoerd op Windows met een versie van NTPd (Network Time Protocol daemon) die geschikt is voor dit platform. De resultaten van de testen staan beschreven in de volgende alinea's. Details van toegepaste configuraties en captures van verbindingen zijn te vinden in: 'Bijlage 2: Details distribueren GPS-tijd atoomklok (NTP)'.

#### Test GNS3



Figuur 17 Voorbeeld testopstelling GNS3

Het onderzoek is begonnen met het testen van tijdsynchronisatie (NTP) in GNS3 (zie figuur 17). In eerste instantie is het geprobeerd zonder OSDD, om te kijken of het überhaupt virtueel werkend te krijgen is. Er is begonnen met het testen op IOU-images. Dit zijn Cisco images die draaien op Unix. Aanvankelijk werkte NTP niet op deze images, maar uiteindelijk is het wel gelukt door een interface direct te koppelen met internet en zo een echte NTP-server toe te wijzen. Zonder de koppeling naar internet is er getest met verschillende images, maar telkens met zonder resultaat. Hieruit is de conclusie getrokken dat het om een bug moet gaan, omdat het op fysieke apparatuur direct werkend is gekregen. Daardoor is er vervolgens getest met 'Dynamips' (emulator images geschreven voor het emuleren van Cisco routers) en m.b.v. deze software werkt NTP direct, maar nog zonder tussenkomst van de OSDD. Vervolgens is er besloten om de OSDD (virtueel) ertussen te plaatsen en te kijken of de client (broadcastclient) de tijd kan ontvangen van de server (broadcastserver). Met de tussenkomst van de OSDD is het gelukt om het aan de praat te krijgen met IOS 12.4, maar met IOS 15.1 (en hoger) werkt het niet, omdat vanaf deze versie (IOS

15.1) wil gaan communiceren met de NTP-server (als normale NTP-client) als er NTP-broadcast pakketten binnenkomen. Dit komt omdat het NTP-mechanisme (de werking) in deze versie(s) veranderd is. Er wordt een zogenaamde 'volley' naar de NTP-broadcastserver gestuurd en daardoor veranderd de NTP-broadcastclient heel even naar een klassieke client en dat werkt niet met het gebruik van een datadiode of ander uni-directioneel apparaat. Hierbij is nog een getracht om een statische ARP (Adres Resolutie Protocol) injectie te doen op de client met het IP-adres en MAC-adres van de server, maar helaas zonder resultaat. RFC5905 (Network Time Protocol Version 4: Protocol and Algorithms Specification) is gebruikt als bron en referentie voor de werking van NTPv4 en details die hiervan gebruikt zijn om het onderzoek te ondersteunen zijn te vinden in hoofdstuk 5.1.2.

### Test Windows

Ook is de werking van het NTP-protocol over broadcast getest onder Windows. Onder Windows wordt het ontvangen van broadcast packets met NTP-data genegeerd door de W32time service. Er zijn pakketten die hier wel mee overweg kunnen zoals; Meinberg NTP for Windows (Burnicki, 2016). Dit pakket is eigenlijk het NTPd van Linux maar dan geschikt gemaakt voor Windows. In de test kwam naar voren dat zowel het verzenden van broadcasts als het ontvangen ervan niet ging omdat de applicatie niet aan de interface kon binden.

Level	Date	Source	Category	File
Error	25/11/2021 19:27:07	NTP	1	None
Information	25/11/2021 19:27:07	NTP	3	None
Information	25/11/2021 19:27:07	NTP	3	None
Information	25/11/2021 19:27:07	NTP	3	None
Information	25/11/2021 19:27:07	NTP	3	None
Information	25/11/2021 19:27:07	NTP	3	None
Information	25/11/2021 19:27:07	NTP	3	None
Information	25/11/2021 19:27:07	NTP	3	None

Event 1, NTP

General Details

bind(496) AF\_INET 192.168.1.255#123 flags 0x419 failed: Can't assign requested address

Figuur 18 Eventviewer melding NTP-broadcast

Het maakte hierbij niet uit welke configuratie we hierbij gebruikte. Het lukte niet om deze twee machines aan elkaar te knopen.

Na verschillende tests op bovenstaande manier is er gekeken om de Windowsapplicatie te laten luisteren als broadcastclient naar een fysieke time appliance welke geconfigureerd staat als broadcastserver. Middels deze manier is het gelukt om de tijd te laten synchroniseren. Vervolgens is de configuratie van de Windowsapplicatie aangepast, zodat deze zowel als broadcastclient en broadcastserver werkt. Met een handige techniek (fysieke netwerk adapter koppelen aan de virtualisatie laag) is de server zijde aan GNS3 gekoppeld waarbij het gelukt is om de Linux machine te laten synchroniseren. Als laatste is deze opstelling getest met de OSDD ertussen, maar dit ging helaas niet werken vanwege dezelfde reden als de test met GNS3.

### K9 en Tardis

Tijdens het onderzoek is er ook een andere applicatie gebruikt voor Windows om de haalbaarheid onder Windows aan te tonen. Met de K9 en Tardis 2000 is het ook mogelijk om NTP-broadcasts uit te zenden en te ontvangen onder Windows. Houdt

rekening met mogelijke licentiekosten in een productieomgeving. Een daadwerkelijke berekening van licentiekosten valt buiten scope van dit onderzoek.

K9 is een kleine applicatie die is gemaakt om de klok van de PC te synchroniseren met andere op hetzelfde LAN. De applicatie K964.exe luistert naar broadcasts op het LAN, pakt de data die in het broadcast pakketje zit en voedt deze aan de W32time service. K9 heeft geen user interface omdat er niets te configureren is. De werking van het programma is namelijk simpel. Met een debug optie kan de gebruiker zien of er data binnenkomt en met de install parameter kan van K9 een service worden gemaakt.

Als er nog geen NTP-broadcastserver in het netwerk hangt kan er door middel van Tardis 2000 vanuit een ander netwerk de tijd worden opgehaald en deze worden doorgestuurd naar het interne LAN door middel van broadcast pakketten.

In fase 1 is gebruik gemaakt van bovenstaande applicaties. In de eerste fase van de test zijn twee Windows VM's gebouwd. Een van de machines was de NTP-broadcastserver en de ander slechts de client. Deze test was direct succesvol. NTP-verkeer ging over het netwerk en de applicatie K9 pakte dit op en synchroniseerde de tijd in Windows. Ook als deze handmatig op een andere tijd werd gezet.

```
C:\Users\LAB\Desktop>k964.exe debug
Listening for time broadcasts
NTP message received from 192.168.1.11 mode 5
Delta is 0.000 seconds
Slew clock
Poll NTP server
NTP Time broadcast received from 192.168.1.11
NTP message received from 192.168.1.11 mode 3
NTP message received from 192.168.1.11 mode 4
Offset is 0.000 seconds
NTP message received from 192.168.1.11 mode 5
Delta is -0.001 seconds
Slew clock
NTP Time broadcast received from 192.168.1.11
NTP message received from 192.168.1.11 mode 5
Best guess of frequency is 156249
Delta is -0.001 seconds
Slew clock
NTP Time broadcast received from 192.168.1.11
NTP message received from 192.168.1.11 mode 5
Delta is 0.000 seconds
Slew clock
NTP Time broadcast received from 192.168.1.11
```

Figuur 19 NTP-broadcast geeft correcte tijd door aan de Windows machine

In fase twee is de (virtuele) OSDD ertussen gezet om te testen of de pakketten nog steeds van VM1 naar VM2 liepen, en dat de tijd nog steeds goed werd gezet in overeenstemming met verwachting. Wat direct opviel was dat de tijd niet meer synchroniseerde. In de debug meldingen van K964.exe kwam geen duidelijke oorzaak naar voren, waarom de tijd niet meer gesynchroniseerd werd. Een Wireshark packet capture weergegeven in figuur 21 laat zien wat er gebeurt.

Op het moment dat het NTP-pakket binnenkomt probeert K9/NTP protocol een ARP request uit te zenden voor het IP-adres waarvan het NTP-pakket is binnengekomen. Deze zal nooit antwoord krijgen vanwege de OSDD die tussen de twee machines is geplaatst.

De volgende stap was het handmatig toevoegen van een ARP-entry in de ARP-tabel van de client machine. Op deze manier zou er nooit een ARP-verzoek hoeven te worden gestuurd.

```
Opdrachtprompt - k964.exe debug
Delta is 1229.627 seconds
Failed to set the time
Poll NTP server
NTP Time broadcast received from 10.0.0.1
NTP message received from 10.0.0.1 mode 5
Delta is 1229.527 seconds
Failed to set the time
Poll NTP server
NTP Time broadcast received from 10.0.0.1
NTP message received from 10.0.0.1 mode 5
Delta is 1229.474 seconds
Failed to set the time
Poll NTP server
NTP Time broadcast received from 10.0.0.1
NTP message received from 10.0.0.1 mode 5
Delta is 1229.905 seconds
Failed to set the time
Poll NTP server
NTP Time broadcast received from 10.0.0.1
NTP message received from 10.0.0.1 mode 5
Delta is 1229.801 seconds
Failed to set the time
Poll NTP server
NTP Time broadcast received from 10.0.0.1
NTP message received from 10.0.0.1 mode 5
Delta is 1229.743 seconds
Failed to set the time
Poll NTP server
NTP Time broadcast received from 10.0.0.1
```

Figuur 20 Tijd kan niet ingesteld worden door broadcast

No.	Time	Source	Destination	Protocol	Length	Info
332	197.506139	VMware_15:ea:43	Broadcast	ARP	42	Who has 10.0.0.1? Tell 10.0.0.2
333	199.858130	10.0.0.1	10.0.0.255	NTP	96	NTP Version 3, broadcast
334	199.859597	VMware_15:ea:43	Broadcast	ARP	42	Who has 10.0.0.1? Tell 10.0.0.2
335	200.506644	VMware_15:ea:43	Broadcast	ARP	42	Who has 10.0.0.1? Tell 10.0.0.2
336	201.506483	VMware_15:ea:43	Broadcast	ARP	42	Who has 10.0.0.1? Tell 10.0.0.2
337	203.954405	10.0.0.1	10.0.0.255	NTP	96	NTP Version 3, broadcast
338	203.955917	VMware_15:ea:43	Broadcast	ARP	42	Who has 10.0.0.1? Tell 10.0.0.2
339	204.497312	VMware_15:ea:43	Broadcast	ARP	42	Who has 10.0.0.1? Tell 10.0.0.2
340	204.515051	10.0.0.2	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
341	205.497719	VMware_15:ea:43	Broadcast	ARP	42	Who has 10.0.0.1? Tell 10.0.0.2
342	205.545574	10.0.0.2	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1

> Frame 333: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface \Device\NPF\_{957F368D-04EB-43AE-9C46-69E8DC865F12}, id 0

> Ethernet II, Src: VMware\_d3:86:c2 (00:0c:29:d3:86:c2), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

> Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.0.0.255

> User Datagram Protocol, Src Port: 123, Dst Port: 123

> Network Time Protocol (NTP Version 3, broadcast)

Figuur 21 Er wordt een ARP request gedaan

Nadat de ARP entry was gemaakt op de client machine, resulteerde dit in een directe werking van de tijd synchronisatie in K9.

```
PS C:\Windows\system32> New-NetNeighbor -InterfaceIndex 5 -IPAddress '10.0.0.1' -LinkLayerAddress '000c29d386c2' -State Permanent
```

Figuur 22 ARP-Injection onder Windows

PS C:\Windows\system32> Get-NetNeighbor		LinkLayerAddress	State	PolicyStore
ifIndex	IPAddress			
1	ff02::16		Permanent	ActiveStore
1	ff02::c		Permanent	ActiveStore
8	239.255.255.250	01-00-5E-7F-FF-FA	Permanent	ActiveStore
8	224.0.1.1	01-00-5E-00-01-01	Permanent	ActiveStore
8	224.0.0.251	01-00-5E-00-00-FB	Permanent	ActiveStore
8	224.0.0.22	01-00-5E-00-00-16	Permanent	ActiveStore
8	10.0.0.255	FF-FF-FF-FF-FF-FF	Permanent	ActiveStore
8	10.0.0.1	00-0C-29-D3-86-C2	Permanent	ActiveStore
1	239.255.255.250		Permanent	ActiveStore
1	224.0.1.1		Permanent	ActiveStore
1	224.0.0.22		Permanent	ActiveStore

Figuur 23 ARP-tabel van de Windows machine

Een Wireshark packet capture laat dan ook zien dat er geen ARP-verzoeken meer over het netwerk worden gestuurd.

No.	Time	Source	Destination	Protocol	Length	Info
186	124.414923	10.0.0.2	10.0.0.1	NTP	90	NTP Version 3, client
187	126.973237	0.0.0.0	255.255.255.255	DHCP	328	DHCP Discover - Transaction ID 0x27396c06
188	127.997428	10.0.0.1	10.0.0.255	NTP	90	NTP Version 3, broadcast
189	127.998822	10.0.0.2	10.0.0.1	NTP	90	NTP Version 3, client
190	128.460934	0.0.0.0	255.255.255.255	DHCP	328	DHCP Discover - Transaction ID 0xbe7aa133
191	132.094058	10.0.0.1	10.0.0.255	NTP	90	NTP Version 3, broadcast
192	132.095376	10.0.0.2	10.0.0.1	NTP	90	NTP Version 3, client
193	136.189667	10.0.0.1	10.0.0.255	NTP	90	NTP Version 3, broadcast
194	136.191624	10.0.0.2	10.0.0.1	NTP	90	NTP Version 3, client
195	140.285242	10.0.0.1	10.0.0.255	NTP	90	NTP Version 3, broadcast
196	140.288462	10.0.0.2	10.0.0.1	NTP	90	NTP Version 3, client

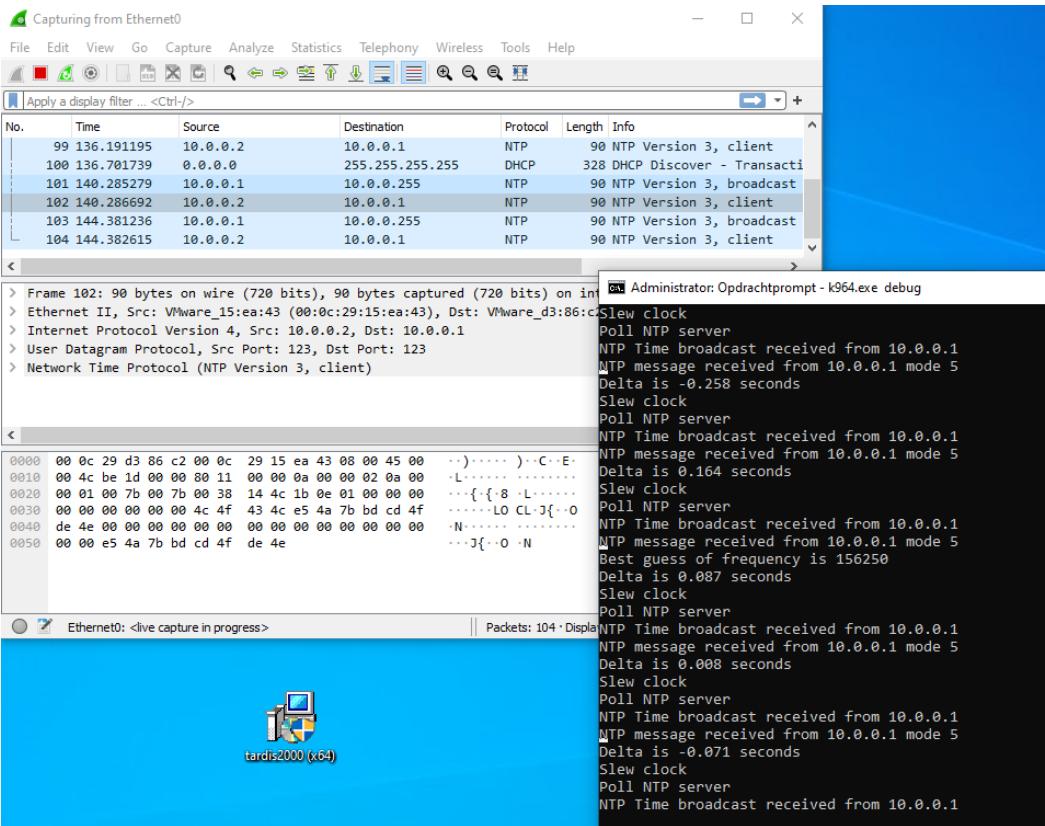
> Frame 192: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface \Device\NPF_{957F3680-04EB-43AE-9C46-69E8DC865F12}, id 0
> Ethernet II, Src: VMware_15:ea:43 (00:0c:29:15:ea:43), Dst: VMware_d3:86:c2 (00:0c:29:d3:86:c2)
> Internet Protocol Version 4, Src: 10.0.0.2, Dst: 10.0.0.1
> User Datagram Protocol, Src Port: 123, Dst Port: 123
> Network Time Protocol (NTP Version 3, client)

0000  00 0c 29 d3 86 c2 00 0c 29 15 ea 43 08 00 45 00  ..).... )..C..E
0010  00 4c 06 9d 00 00 80 11 00 00 0a 00 00 02 0a 00  ..L.....
0020  00 01 00 7b 00 38 14 4c 1b 0e 01 00 00 00 00 00  ..{. 8 L.....
0030  00 00 00 00 00 00 4c 4f 43 4c e5 4a 75 96 f0 e5  ..LO CL.Ju...
0040  5f 2b 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .._+.....
0050  00 00 e5 4a 75 96 f0 e5 5f 2b  ..Ju..._+

Figuur 24 Er wordt geen ARP-request meer gedaan

Het eindresultaat was een werkende NTP-broadcast client en server met een OSDD ertussen. Uit nader onderzoek blijkt dat de reden waarom dit werkt, is omdat Tardis2000 een oudere versie van het NTP-protocol gebruikt. Namelijk NTPv3. In versie vier van hetzelfde protocol wordt er namelijk als beveiligingsmaatregel gebruik gemaakt van een zogenaamd “volley”-mechanisme. Dit om de beveiliging te verbeteren. In het onderzoek naar RFC5905 die later in dit document wordt uitgediept gaan we hier verder op in.



Figuur 25 De tijd wordt correct gezet in combinatie met een datadiode

### 5.2.2 RFC5905

RFC5909 (Network Time Protocol Version 4: Protocol and Algorithms Specification) is gebruikt als bron en referentie om op te helderen wat de verschillende modi zijn van de protocollen en hoe het NTP-protocol werkt in broadcast mode. Protocol modi worden als volgt onderscheiden:

“There are three NTP protocol variants: symmetric, client/server, and broadcast. Each is associated with an association mode (a description of the relationship between two NTP speakers) as shown in Figure 1. In addition, persistent associations are mobilized upon startup and are never demobilized. Ephemeral associations are mobilized upon the arrival of a packet and are demobilized upon error or timeout.

Association Mode	Assoc. Mode Value	Packet Mode Value
Symmetric Active	1	1 or 2
Symmetric Passive	2	1
Client	3	4
Server	4	3
Broadcast Server	5	5
Broadcast Client	6	N/A

Tabel 13 (RFC5905, NTPv4, 2010, p. 6-7)

Over NTP-broadcast staat het volgende beschreven:

“In the broadcast variant, a persistent broadcast server association sends periodic broadcast server (mode 5) packets that can be received by multiple clients. Upon reception of a broadcast server packet without a matching association, an

ephemeral broadcast client (mode 6) association is mobilized and persists until error or timeout. It is useful to provide an initial volley where the client operating in client mode exchanges several packets with the server, so as to calibrate the propagation delay and to run the Autokey security protocol, after which the client reverts to broadcast client mode. A broadcast server pushes synchronization to clients and other servers.” (RFC5905, NTPv4, 2010, p. 6-7)

Uit bovenstaand citaat is op te maken dat de zogenaamde ‘volley’ geen vereiste is, maar wel handig is om te gebruiken als er in de broadcast modus gewerkt wordt. Als de ‘volley’ **wel** gebruikt wordt veranderd de broadcastclient heel even naar een klassieke NTP-client die packets stuurt naar de broadcastserver om de propagatie vertraging te kalibreren en het autokey securityprotocol uit te voeren. Als de ‘volley’ **niet** gebruikt wordt blijft de broadcastclient zoals deze is en hierdoor is dus te verklaren wanneer het wel of niet werkt. Ook is hiermee duidelijk geworden dat bijvoorbeeld Cisco IOS deze functionaliteit met de komst van IOS 15.1 (Releasedatum 10-nov-2010) heeft ingebouwd en daardoor NTP-broadcast niet meer werkt als de OSDD ertussen zit. Vanzelfsprekend geldt dit niet alleen voor Cisco-software, want naar alle waarschijnlijkheid hebben meer fabrikanten deze functie ingebouwd. Dit verklaart en bewijst daarmee de tests die zijn uitgevoerd waarbij het niet lukte om client te laten synchroniseren met de server.

Kortom: als de OSDD wordt gebruikt in combinatie met NTP-broadcast en de software maakt gebruik van de volley, dan gaat de synchronisatie niet lukken. Hiermee is de use-case niet nutteloos, want... verderop in het hoofdstuk meer.

### 5.2.3

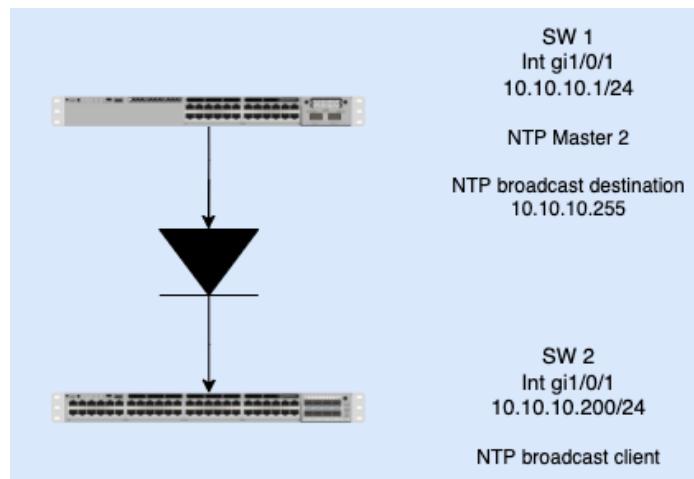
#### Onderzoek met fysieke apparatuur

Na de virtuele test die zijn uitgevoerd, waarbij sommige tests wel werkte en sommige niet, zijn er ook nog test uitgevoerd met fysieke apparatuur. Dit is gedaan om zeker te weten of er mogelijk met fysieke apparatuur NTP-broadcast wel, niet of anders reageert. Tijdens deze test er is gebruik gemaakt van de volgende apparatuur:

- Cisco routers
- Cisco switches
- NTP-timeserver (Elroma NTS-3000)
- Windows PC

### Fysieke test 1

Testopstelling (zie figuur 26) gemaakt met 2x Cisco Catalyst 9300, 1x Cisco Catalyst 9300-24U-A & 1x Cisco Catalyst 9300-48U-A. Beide switches met softwareversie IOS-XE 17.3.4 (recommended release op het moment van testen.)



Figuur 26 Testopstelling test 1

Helaas was er op het moment van fysiek testen geen datadiode aanwezig. Als workaround is er een access-list toegepast tussen switch 1 (NTP-server) en switch 2 (NTP-client) die al het NTP-verkeer blokkeert tussen switch 2 en switch 1 op de uitgaande interface van switch 2. Dit simuleert het meeste wat er zou gebeuren als er een datadiode tussen zou zitten, behalve dat in dit geval elk ander verkeer wel kan doorkomen. Er wordt immers alleen NTP (UDP 123) geblokkeerd.

SW1 (NTP-Server) SW2 (NTP-Client)

Op beide switches zijn de poorten als routed gezet door op de interfaces het commando te gebruiken no switchport. Dit is ook te testen met switchpoorten en interface vlan's. Uitkomst blijft hetzelfde.

```
conf t
int gi1/0/1
no switchport
```

Configuratie op de interfaces zijn hetzelfde als bij de virtuele test. Alleen zijn hier de interfaces Gi1/0/1 i.p.v. Fa1/1. Behalve op het interface Gi1/0/1 op switch 2 zit nog de access-list die NTP uitgaand verkeer blokkeert.

De resultaten zijn helaas exact hetzelfde als op de virtuele omgeving vanaf IOS 15.1 nadat broadcast berichten zijn ontvangen wil de NTP-client meteen een bevraging gaan doen bij de NTP-server, omdat in dit geval dit wordt geblokkeerd d.m.v. een access list lukt dit niet. We zien exact hetzelfde gedrag als op de virtuele omgeving.

### Fysieke test 2: Windows K964

In fysieke test 2 is geprobeerd om op Windows het NTP-signalen op te vangen door de OSDD zonder dat de NTP-client een volley doet om de bron te verifiëren. In de virtuele opstelling was dit succesvol met het programma K964 en dit bewijst dat het mogelijk is onder Windows en met NTPv4.

De opstelling bestaat uit de volgende apparatuur (figuur 27):

- NTP Tijdserver NTPv4 (ELPROMA NTS-3000)
- Open Source Data Diode
- Cisco Router 819F (switch poorten van de router)
- Laptop met ethernetpoort



Figuur 27 Testopstellingen fysieke apparatuur

Op een laptop is de applicatie K964 gestart. Deze vangt het NTP-verkeer op en veranderd de tijd op de Windows laptop zodra het signaal is ontvangen. In theorie is het ook mogelijk om de lokale tijd dan weer te gebruiken om dit verder uit te zenden over het interne netwerk.

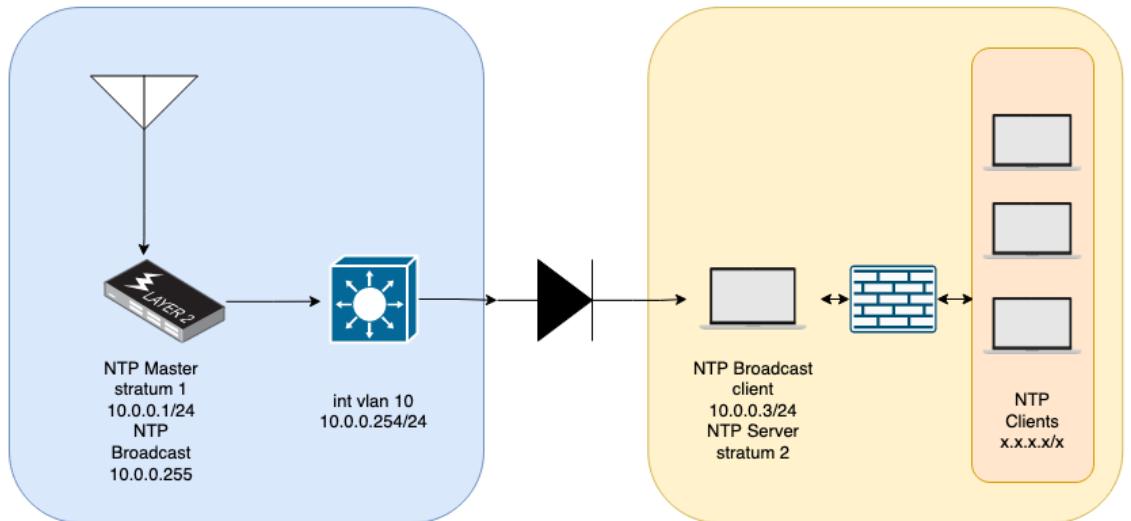
Door het commando “*k964.exe debug*” uit te voeren kan men volgen waar de NTP-pakketten vandaan komen en hoe deze worden geïnterpreteerd. Na verloop van tijd en naarmate er meer pakketten worden ontvangen zal de tijd accurater worden (delta). De test was succesvol, met de Windowsapplicatie k964 is het mogelijk om de lokale tijd te synchroniseren met de gps-tijd door de OSDD heen. In de bijlage (bijlage 2, figuur 49) staat een Wireshark capture weergegeven waarin te zien is dat er NTP-broadcast v4 pakketten worden verstuurd en dat er geen traditioneel NTP-verkeer teruggaat.

#### 5.2.4 Advies

Op basis van alle verzamelde informatie, analyses, ervaringen en het onderzoek d.m.v. virtuele en fysieke tests dat gedaan is naar het distribueren van een GPS-tijd signaal met behulp van NTP door een datadiode, worden er twee aanbevelingen gedaan om deze specifieke use-case te gaan gebruiken.

- Voor een variant welke men direct zou willen implementeren in verschillende air gapped netwerken is het advies om tenminste een opstelling neer te zetten welke bestaat uit: GPS + time-server (broadcastserver) → switch → OSDD → Mini-PC dual NIC (broadcastclient + traditionele NTP-server). Per netwerk een OSDD en Mini-PC. De keuze voor een Mini-PC (bare metal met OS) met twee ethernet kaarten is helder uit te leggen, omdat het de kosten behoorlijk drukt t.o.v. bijvoorbeeld server en is ook een heel stuk kleiner en neemt daarmee minder ruimte in. Daarnaast is het zo dat er qua resources niet veel gevraagd wordt van deze PC en daarmee wordt een server die voor zware taken is bedoeld niet licht ingezet. Het systeem moet worden voorzien van tenminste Windows 10, Windows Server 2016 of hoger, de K9 software die wordt ingezet als broadcastclient en Tardis of Meinberg NTP (NTPd voor Windows) om te fungeren als traditionele NTP-server om het tijdsignaal te distribueren over het air gapped netwerk.

Om het bovenstaande advies nog beter toe te lichten staat hieronder een mogelijk technisch ontwerp (figuur 28) weergegeven. IP-adressen etc. zijn vanzelfsprekend naar eigen smaak in te richten en te kiezen.



Figuur 28 Technisch ontwerp “Distribueren GPS-tijd (NTP)” voorbeeld

- Het tweede advies is het ontwerpen en bouwen van een soort timeproxy. Dit zou bijvoorbeeld kunnen d.m.v. scripting (python etc.). Deze proxy zal de functie van de NTP Broadcast client overnemen. Hiermee wordt de stabiliteit verhoogd en de complexiteit en kosten verlaagd. Om deze manier goed uit te zoeken is vervolgonderzoek nodig en is niet meegenomen in dit onderzoek, omdat het buiten scope van dit project valt.

Verder moet rekening gehouden worden om de nodige securitymaatregelen te implementeren. Op de switch die gebruikt gaat worden is het bijvoorbeeld handig om poorten die niet gebruikt worden op ‘shutdown’ te zetten. Daarnaast is het aan te bevelen dat er port-security, accesslists etc. worden geconfigureerd. Op Windows moet gedacht worden aan de firewall regels. In principe hoeft er alleen maar NTP-verkeer vanuit een specifieke host toegestaan te worden. Verder is het erg belangrijk dat er wordt gedacht een afgesloten ruimte of 19 inch kuis met speciale toegang waar de time-server en switch in gaan om zo controle te houden en bijvoorbeeld te zorgen dat een aanvaller geen malicious pakketten vanuit de time-server kan versturen. NTP-authenticatie is een mogelijke optie om te gebruiken, maar is niet meegenomen in dit onderzoek. Om precies vast te stellen wat nodig is qua securitymaatregelen, is het aan te raden om in een vervolgonderzoek te overleggen en af te stemmen met de mensen die gaan over fysieke- en informatiebeveiliging.

*Als laatste is het belangrijk om te vermelden dat het niet gebruiken van de NTP-volley in deze inzet geen negatieve gevolgen heeft aangezien de fysieke beveiliging rondom de datadiode volledig onder controle is. Een aanvaller is dan ook niet in staat om bij een goed geconfigureerde opstelling maliciouse pakketten naar de broadcast client te sturen.*

## 6. Resultaten

Dit hoofdstuk beschrijft de onderzoeksresultaten van de deelvragen die het onderzoek ondersteunen.

### 6.1 Gerubriceerde domeinen

Het onderzoek dat is besproken in dit document bespreekt voornamelijk de Open Source Data Diode (OSDD). Deze datadiode, hoewel effectief, kan niet op alle rubriceringen ingezet worden. Op dit moment stelt het NBV dat bij een evaluatie maximaal het niveau departementaal vertrouwelijk haalbaar is, maar wat zijn rubriceringen eigenlijk? Wat betekent het voor de inzetbaarheid van een OSDD dat deze niet voor alle rubriceringen inzetbaar is?

De OSDD heeft niet als bedoeling om ingezet te worden voor de bescherming van staatsgeheimen, hiervoor zijn momenteel al de bestaande commerciële oplossingen beschikbaar. (Ministerie van Defensie, 2017)

#### 6.1.1 Rubriceringen

Verschillende rubriceringen bestaan om de hoeveelheid potentiële schade aan te geven die kan optreden wanneer de gerubriceerde informatie wordt ingezien door onbevoegden.

*Restricted – Departementaal vertrouwelijk*

Deze rubricering betreft geen staatsgeheim en kan bij lekken impact hebben op het belang van Nederlandse Ministeries.

*Confidential – Staatsgeheim Confidentieel*

De laagste vorm van staatsgeheim, wanneer dit lekt kan dit schade toebrengen aan Nederland en haar bondgenoten.

*Secret – Staatsgeheim Geheim*

De overtreffende trap van ‘Confidentieel’, bij lekken kan dit zware schade toebrengen aan Nederland en haar bondgenoten.

*Top Secret – Staatsgeheim Zeer Geheim*

De zwaarste rubricering, bij lekken kan dit zeer zware schade toebrengen aan Nederland en haar bondgenoten.

#### 6.1.2 Merkingen

Bij alle rubriceringen kan een ‘merking’ worden aangebracht om verder aan te geven hoe er met de informatie om moet worden gegaan.

*NLD Ongerubriceerd*

Hoewel de informatie niet is gerubriceerd is deze niet voor het brede publiek bedoeld. Er geldt nog steeds een ‘need-to-know’ principe.

*Personenvertrouwelijk*

Dit is persoonlijke informatie, enkel bedoelt voor een individu, het verstrekken van de informatie aan derden kan de persoon schade toebrengen.

#### *Commercieel vertrouwelijk*

Informatie van een bedrijf, naast algemene interne informatie kan dit ook specifieke productietechnieken omvatten. Het verstrekken van deze informatie aan derden kan naast het genoemde bedrijf ook Nederland schaden.

#### *Medisch geheim*

Deze informatie omvat alle medische (lichamelijk en geestelijk) informatie van een persoon. Het lekken van deze gegevens kunnen vooral de belangen van de persoon schaden, een persoon kan immers anders behandeld worden wanneer deze bijvoorbeeld een chronische ziekte heeft.

#### *NLD-eyes-only*

Word samen met een rubricering gebruikt om informatie puur voor Nederlanders te houden. Wanneer buitenlanders deze informatie inzien kan dit schade voor defensie opleveren.

#### *Releasable to*

Dit merk geeft aan welke o.a. organisatie de informatie mag inzien. Verstrekken aan onbevoegden kan de organisatie waar de informatie naar refereert beschadigen.

#### *Crypto*

Dit merk wordt geplaatst op informatie die gaat over cryptografie, enkel cryptodeelnemers mogen dit inzien.

#### *Crypto-Security*

Vergelijkbaar met het 'Crypto' merk, deze informatie kan er bij lekken echter voor zorgen dat niet-geautoriseerde versleutelde informatie kunnen ontcijferen.

#### *Comsec*

Dit merk wordt aangebracht op informatie die niet als crypto of crypto-security zijn gemerkt maar waar wel speciale regels voor verbindingsbeveiliging geldt.

## 6.2 Voor- en nadelen

Welke voor- en nadelen heeft een opensource datadiode nou eigenlijk? Tijdens dit onderzoek is er veel informatie ondervonden over de OSDD. Hier zitten natuurlijk ook voor- en nadelen aan. Een datadiode is een apparaat die alleen verkeer van A -> B mag sturen maar niet meer terug van B -> A. Door het onderzoek naar de OSDD zijn we tot verschillende voor- en nadelen gekomen die kunnen helpen je systeem te beveiligen of toch (niet) goed werkbaar zijn.

De volgende voorbeelden komen uit het onderzoek van de use-case Certificate Authority en de use-case van Distribueren GPS-tijd atoomklok (NTP):

#### **Voordelen van een datadiode:**

- Dataverkeer kan altijd maar één kant op (Uni-directioneel).
- De software achter de datadiode is niet makkelijk te manipuleren. Men moet echt fysiek inbreken om erbij te komen omdat het een fysiek hardware apparaat is.
- Men zit niet vast aan één leverancier bij de opensource datadiode. Het is vrij om te bepalen om een eigen leverancier te kiezen waar men het beste voordeel uit kan halen.
- Een datadiode heeft geen toegang tot het internet omdat er geen logica aanwezig is in de datadiode.
- Kosten voor een opensource datadiode zijn betaalbaar voor ieder bedrijf die een datadiode zou willen gebruiken.

- Multicast verkeer gaat door de datadiode.
- UDP-cast gaat ook goed in combinatie met de datadiode.

#### Nadelen van een datadiode:

- Bijna alle netwerkprotocollen zijn gemaakt met bi-directionaliteit als validatie. Voor goed gebruik van een datadiode moet er een alternatief worden gevonden voor het betrouwbaar afleveren van IP-pakketten en data.
- NTP versie 4 vereiste een volley die terugkwam vanuit de server op de virtuele Cisco Switch.
- Kosten voor een commerciële datadiode zijn nog redelijk aan de hoge kant. Dit ligt aan de proxy's die meestal bijgeleverd worden door een leverancier.
- Met een datadiode is er geen actieve scanning en beveiliging. Dit zou men apart moeten leveren.
- Met de kennis van nu is een opensource datadiode nog te complex voor opzet.

### 6.3 Beperkingen mitigeren

Uit de twee onderzoeken van de use-cases is gebleken dat er flink wat beperkingen zitten aan het gebruik van de OSDD. Vooral bij de NTP use-cases kwam de volley naar voren als probleem. NTP-versie 4 vereiste een volley die terugkwam vanuit de server op de virtuele Cisco switch, waarop het getest is. Desondanks NTP een UDP-protocol is. Dit probleem zou eventueel opgelost kunnen worden door een NTP-proxy die voor en na de datadiode staat. Alleen zitten hier weer andere risico's en mogelijke problemen aan verbonden zoals de mogelijke vertraging tussen de proxy's.

Proxy's kunnen ook bij veel andere problemen waar normaliter TCP-verkeer bij nodig is ingezet worden. Zo kunnen de server en client met het TCP- en/of UDP-protocol communiceren met de proxy.

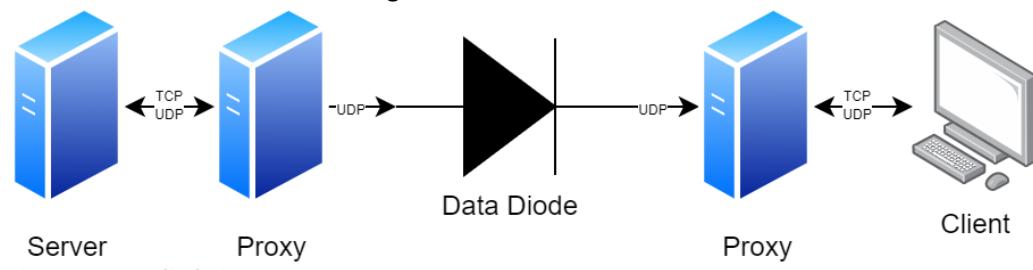
In de NTP use-case zou dat zo werken. De server broadcast een NTP-pakket over het netwerk die de proxy ontvangt.

De proxy zet het NTP-pakket om naar platte tekst en stuurt het over UDP door naar de andere proxy. De proxy aan de client kant zet de platte tekst weer om naar een NTP-pakket en stuurt deze als een broadcast weer door naar de client.

De proxy aan de client kant zal dan ook reageren op de volley die de client als extra check stuurt.

Deze manier van communiceren kan ook toegepast worden op ander TCP-verkeer dat in principe alleen maar één kant op hoeft. Denk hierbij aan filetransfers, video streamen, mail verzenden, etc. Toepassingen die twee kanten op moeten werken natuurlijk niet, zoals DHCP, HTTP(S), SSH.

Bij commerciële datadiodes zijn daarom vaak de proxy's prijziger dan de OSDD zelf. Dit komt omdat deze vaak speciaal ontwikkeld moeten worden voor de betreffende use-case. Maar door middel van de proxy's wordt het mogelijk om andere toepassingen ook door de datadiode te laten gaan.



Figuur 29 Datadiode i.c.m. proxy

Wat een ander probleem kan zijn is de snelheid en de mogelijke packet loss van de OSDD. Bij file transfers waar veel data op hoge snelheid door de OSDD moet, kan het zijn dat er pakketjes verloren gaan bij het verzenden. Dit omdat er geen check gedaan kan worden tussen de zendende en ontvangende kant.

Om dit te voorkomen dient de snelheid beperkt te worden tot een lagere snelheid dan de snelheid van de fysieke poort op de datadiode.

Met forward error correction is het mogelijk om extra bits met informatie toe te voegen aan de data flow, zodat de missende informatie weer hersteld kan worden aan de andere kant mocht er data verloren zijn. Het nadeel hiervan is dat de data flow groter wordt door deze extra informatie en daardoor zal de snelheid nog meer afnemen.

Een andere optie die nog zou kunnen is het toevoegen van volgnummers aan de pakketten die vanaf de zendende proxy naar de ontvangende proxy gestuurd worden. Hiermee kun je valideren dat de ontvangen informatie correct en compleet is.

Mocht er een volgnummer ontbreken dan wordt de ontvangen informatie gemarkeerd als incorrect en niet compleet en wordt de informatie verder verzonden.

#### 6.4 Risico's afdekken

Welke risico's kan je afdekken met het gebruik van een datadiode?

Datadiodes zijn in principe ontwikkeld om bi-directioneel verkeer tussen 2 netwerken niet mogelijk te maken, ze zijn ontwikkeld om alleen verkeer van netwerk A -> B mogelijk te maken of van B -> A, dit noem je ook wel uni-directioneel verkeer, verkeer kan altijd maar één richting uit. Met een datadiode kan je dus garanderen dat verkeer altijd maar één kant op kan. Hiermee kan je meerdere risico's afdekken.

In de volgende voorbeelden gebruiken we als upstream network A en als downstream network B

##### **Informatie van het netwerk B mag niet uitgelekt worden naar de buitenwereld.**

Met een datadiode kan je garanderen dat er geen informatie vanuit netwerk B uitgelekt kan worden via netwerk A omdat er geen verkeersstroom mogelijk is, het is wel mogelijk om informatie vanuit netwerk A te delen met netwerk B. (In dit geval dien je wel rekening te houden met andere risico's, denk bijvoorbeeld aan USB-poorten die geblokkeerd moeten worden en/of het blokkeren van printen op netwerk B enz.)

##### **Command and control verkeer wordt niet mogelijk.**

Als netwerk B geïnfecteerd wordt via netwerk A en er wordt contact gezocht met een command and control server is dit niet mogelijk. Je beperkt dus het risico als het netwerk geïnfecteerd raakt. Er is geen verkeer mogelijk met het internet.

Ook als je de OSDD andersom plaatst beperk je dit risico (B -> A) als het netwerk geïnfecteerd raakt (bijvoorbeeld via een USB-stick), kun je wel communiceren met een command and control server echter kan deze niet met terug communiceren. (Op TCP kan je dan ook geen sessie opzetten, je krijgt dan namelijk geen SYN-ACK terug.)

##### **Vanuit netwerk A mag er niet gecommuniceerd worden met netwerk B of andersom**

Met een datadiode kan je ervoor zorgen dat geen communicatie mogelijk is vanuit het ene netwerk naar het andere. Je kan vanuit netwerk B wel communiceren met netwerk A maar niet andersom. Dit kan nuttig zijn als er vanuit netwerk B, informatie opgehaald moet worden aan netwerk A, maar vanuit netwerk A geen informatie opgehaald mag worden aan netwerk B. (Fox-IT)

## 6.5 Open source vs. commercieel

Wat zijn de verschillen tussen open source en commercieel? Het antwoord op die vraag wordt beschreven in deze paragraaf.

### 6.5.1 **Algemeen**

Als men praat over open source software (OSS) wordt er vaak gedacht aan gratis software en bij commercieel speelt de gedachte dat men moet betalen. Alleen is gratis bij open source niet altijd waar. Open source betekent in de basis dat de broncode open is en dat geeft men de mogelijkheid om deze code aan te passen, door te ontwikkelen en mogelijk kunnen er extra's worden toegevoegd. Iets wat eigenlijk ook de intentie is van open source. Bij commerciële software of ook wel closed source software (CSS) genoemd, is de broncode niet vrij beschikbaar en daardoor kan men ook geen aanpassingen doen. Dit komt omdat de broncode bijvoorbeeld is ontwikkeld door een bedrijf dat geld wil verdienen aan het verkopen van hun product. Met dit type software kan het dus zomaar zijn dat een bepaalde functionaliteit niet in het pakket zit waarmee je afhankelijk bent van de ontwikkelaar. (B. Holzer, 2021)

### 6.5.2 **Documentatie**

Kijkend naar de beschikbare documentatie voor open source software dan valt op dat verreweg het meeste hiervan zijn gemaakt voor gekwalificeerd ICT-personeel en ontwikkelaars waarin de inhoud zich voornamelijk richt op wat voor manier de code gebruikt kan worden. Dit betekent dat men zelf aan de slag moet om zorg te dragen dat de software een compleet geheel wordt. Als dit vergeleken wordt met de commerciële kant dan is er een duidelijk verschil. Deze software wordt over het algemeen geleverd als compleet pakket en daarbij zitten uitgebreide handleidingen en andere documentatie. Dit zijn bijvoorbeeld installatiehandleidingen en gebruiker handleidingen. (B. Holzer, 2021)

### 6.5.3 **Klantenservice**

Voor wat betreft klantenservice en ondersteuning moge het duidelijk zijn dat de commerciële of closed source de overhand heeft. Bijna alle leveranciers en/of bedrijven in deze hoek hebben een supportafdeling of helpdesk ingericht waar men terecht zou kunnen met vragen of problemen. Binnen dit soort afdelingen zijn vaak verschillende niveaus van ondersteuning. Uiteraard betaal je als klant een bedrag voor deze ondersteuning. Bij open source zijn dit soort opties beperkter, omdat er geen grote spelers zijn die deze ondersteuning leveren, maar het is wel in opkomst doordat er wat te verdienen valt met het leveren van support en het verbeteren van de software zelf. Over het algemeen genomen zoekt men eerst op internet naar mogelijke oplossingen zoals bijvoorbeeld forums. (B. Holzer, 2021)

### 6.5.4 **Security**

Over security in het hedendaagse IT-landschap zijn de meningen bijzonder verdeeld en dat is natuurlijk niet zomaar. In principe is het zo dat er op IT-security gebied in zowel open source als commercieel voor- en nadelen zitten. Bij open source software is het heel fijn dat de broncode open is, maar uiteraard zit hier ook een groot risico aan vast. Zo kunnen bijvoorbeeld hackers de code helemaal binnenstebuiten keren om zo een mogelijk zwakke plek te vinden in de code en hier een stuk malicious code in plaatsen om vervolgens snel informatie te kunnen onttrekken. Dit is dus een duidelijk risico. De keerzijde hiervan is dat mogelijke zwakke code ook snel gerepareerd kan worden waarna er een nieuwe release kan

worden uitgebracht. Bij commerciële of closed source software is de broncode niet beschikbaar dus vaak denkt men dat dit een veiligere optie is. Vanzelfsprekend is dit anno 2022 een naïeve gedachte, omdat ook deze software wordt afgezocht naar zwakke plekken en als er zwakke plekken zijn dan is het te hopen de ontwikkelaars van de software de onveiligheden snel herstellen. (B. Holzer, 2021)

#### 6.5.5 Prijzen

Open source software is over het algemeen gratis, maar is lang niet altijd het geval. Het is bijvoorbeeld mogelijk dat er licentiekosten zijn als men deze software zou willen gebruiken binnen een productieomgeving. Hiervoor moet dan een zogenaamde subscriptie afgesloten worden. Dit betekent simpelweg dat de software gebruikt mag worden, maar daarnaast kan men de beschikking krijgen over updates en/of support. Voor commerciële of closed source software moet vaak betaald worden al is dat niet altijd het geval, zoals met bijvoorbeeld Adobe Reader, Virtualbox etc. Hoeveel er betaald moet worden ligt aan het type software en op hoeveel systemen het geïnstalleerd gaat worden. Hierbij kan het ook zijn dat er per CPU of per core betaald moet worden. Wel is het mogelijk om voor een bepaalde tijd of met een beperkt aan resources van de software een gratis testversie te krijgen. (B. Holzer, 2021)

#### 6.5.6 Wie kiezen er het vaakst voor open source?

Uit onderzoek blijft dat overhedsinstanties, kleine bedrijven en bedrijven met een technologie insteek het vaakst kiezen voor open source software. Deze hebben allemaal een eigen reden om daarvoor te kiezen.

Het blijkt dat overhedsinstanties het vaakst hiervoor kiezen. Door open source software te gebruiken heeft men veel vrijheid en kan ervoor gezorgd worden dat de ontwikkeling ervan snel gaat. Daarnaast werken deze instanties vaak op eenzelfde manier en hierdoor kan software overgenomen of gedeeld worden. Door deze uniformiteit kan tijd bespaard worden.

Kleine bedrijven kiezen om een heel andere reden voor open source. Deze bedrijven hebben vaak niet heel veel geld voor het aankopen van dure software en maken daardoor vaker gebruik van de grote keuze op het gebied van open source software. Hierdoor kunnen deze bedrijven besparen op kosten, omdat ze geen contracten en/of andere services afsluiten. Wel moet er rekening gehouden worden met mogelijke licentie- en supportkosten.

Technologiebedrijven kiezen ook redelijk vaak voor open source. Dit is omdat deze bedrijven invloed kunnen uitoefenen over de ontwikkeling van de software en zo kan men er sturing aan geven. (ICT Portal, 2021)

#### 6.5.7 Model van Tracy en Wiersema

Het model van Tracy en Wiersema wordt gebruikt om waarde strategieën in kaart te brengen. Het model bestaat uit die waardes waarop een bedrijf kan uitblinken:

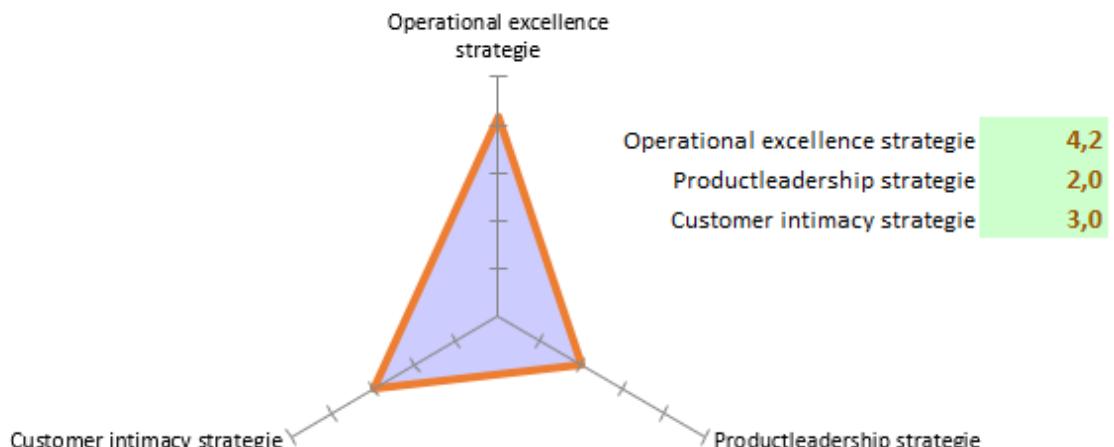
- Productleiderschap (Product Leadership)
  - Continue nieuwe producten
  - Creatief
  - Innovatief
  - Beter dan de concurrent
- Klantenpartnerschap (Customer Intimacy)
  - Goede klantenservice
  - Klantrelatie
  - Producten afstemmen op wensen van de klant

- Kostenleiderschap (Operational Excellence)
  - Kosten zo laag mogelijk
  - Effectiviteit
  - Prijs/kwaliteit verhouding

Elk bedrijf doet iets in de drie waarde strategieën, maar blinkt uit op één onderdeel om zo te proberen onderscheidend te zijn ten opzichte van andere bedrijven. Defensie is een organisatie die in principe niks verkoopt en heeft geen commercieel belang, maar werkt samen met Technolution om mogelijk de OSDD op de markt te krijgen. De broncode zou dan vrij beschikbaar moeten zijn, maar er zijn bijvoorbeeld wel kosten voor de hardware en licentiekosten. Om deze reden is getracht in kaart te brengen op welke waarde men mogelijk gaat uitblinken. Gezien de broncode open source is en deze beschikbaar gesteld moet gaan worden aan klanten moet het duidelijk zijn dat men gaat uitblinken in kostenleiderschap. (Marketingscriptie, 2021) De reden hiervoor kenmerkt zich door de volgende punten:

- Kosten voor het ontwikkelen van de broncode zijn erg laag, omdat dit een samenwerking is tussen de Haagse Security Delta en het Ministerie van Defensie welke beide organisaties zijn zonder winstogmerk;
- Technolution bouwt de fysieke behuizing met daarin de hardware waarbij het de prioriteit heeft dat het zo klein en betaalbaar mogelijk wordt;
- De OSDD heeft geen management interface etc. waardoor effectiviteit gegarandeerd wordt;
- De totale kosten van de OSDD zijn erg laag, maar deze is in overheidstermen wel geschikt te krijgen tot en met departementaal vertrouwelijk;
- Er is een duidelijk verschil met de huidige markt, omdat er geen open source datadiode op markt is.

Hieronder (figuur 30) staat de verhouding weergegeven tussen de drie waarde strategieën.



Figuur 30 - Model van Tracy en Wiersema toegepast

(Managementmodellensite, 2018)

## 7. Conclusie

In dit onderzoek is onderzocht hoe een open source datadiode kan bijdragen aan een hogere cybersecurity binnen de lager gerubriceerde domeinen van overheidsinstanties. Hiervoor zijn 2 scenario's (use-cases) gebruikt en uitgewerkt.

Het gebruik van een datadiode is niet nieuw, het product wordt reeds ingezet bij o.a. defensie, echter zitten hier vanwege voornamelijk ondersteunende hardware hoge kosten aan verbonden. De inzetbaarheid van een lager geprijsde (en open-source) variant is, zoals uit dit onderzoek is gebleken, zeer reëel.

Hoewel het gebruik van de Open Source Data Diode haalbaar is, zijn er wel limitaties en werkwijzen om rekening mee te houden. Op dit moment limiteert de aard van het apparaat haar inzetbaarheid tot non-staatsgeheime rubriceringen, dit maakt het product nuttig voor de commerciële sector, maar gelimiteerd binnen defensie.

Praktisch moet er rekening gehouden worden met de technische consequenties van het gebruik van een datadiode pur sang, uit het onderzoek is duidelijk gebleken dat er bijna altijd ingrijpende aanpassingen in de te beveiligen systemen gemaakt moeten worden. Er bestaan scenario's waar actief om o.a. een securityfeature heen gewerkt moet worden ten behoeve van een datadiode. Bij het gebruik van een closed source (commerciële) datadiode zit dit werk inbegrepen in de (hogere) prijs. Daarnaast zitten er wél extra kosten verbonden aan het onderhouden van capabele medewerkers die met een dergelijk product om kunnen gaan, iets wat minder het geval is bij een commerciële oplossing tenzij deze dienst door een commerciële partij uit handen wordt genomen.

Bij implementaties is gebleken dat er, alvorens een datadiode in te zetten, een goede analyse en PoC nodig is voor het gebruik van een datadiode. Een datadiode breekt inherent netwerkprotocollen, dit heeft verwachte en onverwachtse consequenties voor het functioneren van systemen en protocollen, dit heeft het NTP-scenario zeer duidelijk aangegeven.

Daarentegen is ook gebleken dat een datadiode in een uitgebreidere setup zoals met meerdere datadiode's een extra laag aan beveiliging kan toevoegen zonder externe systemen in huidige vorm inherent onbruikbaar te maken. Zoals beschreven in het onderzoek is de technische opzet van het Certificate Authority scenario niet uniek maar zelfs breed toepasbaar.

Uit dit onderzoek is gebleken dat de datadiode een plek heeft in het IT-landschap en dat een open-source variant het gebruik ervan gaat vergroten. Vanwege enerzijds kosten en anderzijds leeftijd van de bestaande producten is het concept nog relatief niche en zijn veel bestaande protocollen en systemen er nog niet op aangepast. De bredere beschikbaarheid van een betaalbaar product gaat echter de kans op laagdrempelige implementatie enorm verhogen en de beveiliging van vele instanties en bedrijven verbeteren.

## Bibliografie

- B. Holzer. (2021, 07 23). *Open source versus closed source software: wat is het verschil?* Opgehaald van Open Circle Solutions: <https://www.opencirclesolutions.nl/open-source-versus-closed-source-software-wat-is-het-verschil/>
- Burnicki, M. (2016, 09 06). *NTP Broadcast Mode*. Opgehaald van Meinberg Knowledge Base: [https://kb.meinbergglobal.com/kb/time\\_sync/ntp/configuration/ntp\\_broadcast\\_mode](https://kb.meinbergglobal.com/kb/time_sync/ntp/configuration/ntp_broadcast_mode)
- Crum, D. (2018, 06 25). *What is a Data Diode & How Do Data Diodes Work?* Opgehaald van OWL Cyber Defense: <https://owlcyberdefense.com/blog/what-is-data-diode-technology-how-does-it-work/>
- Fox-IT. (sd). *The DataDiode explained in 5 simple steps*.
- Cyber Innovation Hub. (2021). *Setup OSDD virtuele ontwikkel omgeving*.
- Cyber Innovation Hub. (2021). *Whitepaper draft: high volume file transfer using a data diode*.
- ICT Portal. (2021, 10 19). *Welke mogelijkheden biedt Open Source software voor bedrijven?* Opgehaald van ICT Portal: <https://www.ictportal.nl/ict-lexicon/open-source>
- Manage Engine. (sd). Opgehaald van <https://www.manageengine.com/key-manager/microsoft-ca-certificate-signing.html>
- Managementmodellensite. (2018, 09 19). *Scan waardedisciplines Treacy & Wiersema*. Opgehaald van Managementmodellensite: [https://managementmodellensite.nl/scan-waardedisciplines-treacy-wiersema/#.YeKqv\\_7MJGM](https://managementmodellensite.nl/scan-waardedisciplines-treacy-wiersema/#.YeKqv_7MJGM)
- Marketingscriptie. (2021, 03 04). *Waardestrategieën van Treacy en Wiersema invullen / Marketingscriptie.nl*. Opgehaald van Marketingscriptie.nl: <https://www.marketingscriptie.nl/waardestrategieen-treacy-en-wiersema/>
- Ministerie van Defensie. (2017). *ABDO*. Opgehaald van [https://www.defensie.nl/binaries/defensie/documenten/beleidsnotas/2017/06/13/abdo-2017/ABDO+2017\\_web.pdf](https://www.defensie.nl/binaries/defensie/documenten/beleidsnotas/2017/06/13/abdo-2017/ABDO+2017_web.pdf)
- Networklessons.com (Regisseur). (2017). *OpenSSL Certification Authority (CA) on Ubuntu Server (YouTube)* [Film].
- RFC5905, NTPv4. (2010, p. 6-7, June). Opgehaald van Datatracker IETF: <https://datatracker.ietf.org/doc/html/rfc5905>
- Rhysider, J. (2017). Opgehaald van Darknet Diaries: <https://darknetdiaries.com/episode/3/>
- Soullié, A. (2020, 01 28). *Github*. Opgehaald van DYODE : Do Your Own Diode: <https://github.com/wavestone-cdt/dyode>

The Hague Security Delta. (2020, 07 29). *Open Source Data Diode*. Opgehaald van HSD:

[https://securitydelta.nl/images/Factsheet\\_Data\\_Diode.pdf](https://securitydelta.nl/images/Factsheet_Data_Diode.pdf)

Wärtsilä. (2020, 02 27). *Cybersecurity looks to the cloud to protect data at sea*. Opgehaald van Wartsila.Com: <https://www.wartsila.com/insights/article/cybersecurity-looks-to-the-cloud-to-protect-data-at-sea>

Wolff, J. (2016). Opgehaald van Slate: <https://slate.com/technology/2016/12/how-the-2011-hack-of-diginotar-changed-the-internets-infrastructure.html>

## Afkortingen

CA	Certificate Authority
CSR	Certificate Signing Request
CSS	Closed Source Software
DHCP	Dynamic Host Configuration Protocol
FEC	Forward Error Control
GPS	Global Positioning system
HSD	The Hague Security Delta
HTTP(S)	HyperText Transfer Protocol (Secure)
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
MoD	Ministerie van Defensie
NBV	Nationaal Bureau voor Verbindingsbeveiliging
NG-Firewall	Next-generation Firewall
NTP	Network Time Protocol
OSDD	Open Source Data Diode
OSS	Open Source Software
SSH	Secure Shell
STRIDE	Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege.
SWOT	Strengths, Weaknesses, Opportunities, Threats
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

## Bijlage 1: Details Beveiligen Certificate Authority

### Het versturen van het CSR-verzoek

Als eerste is er begonnen met het opzetten van het versturen van het CSR-verzoek.

Om dit werkend te krijgen heb zijn er 2 Ubuntu virtuele machines opgezet met 2 losse netwerkverbindingen.

Dit omdat straks het verkeer via 1 netwerkverbinding uit de machine moet en via een andere verbinding weer terug moet komen.

Hierop zijn 2 verschillende subnets ingesteld 10.100.0.0/24 voor de zendende kant (E0) en 10.101.0.0/24 voor de ontvangende kant (E1). Deze 2 interfaces worden ook gebruikt in de configuratie van UDPcast met de tools udp-sender en udp-receiver.

Na aanleiding van de whitepaper (Hub, 2021) vanuit de stakeholder is er gekozen om het commando wat hij gebruikte in zijn testen aan te passen naar de configuratie binnen use-case van de groep.

De volgende commando's zijn eerst los getest om een bestand te versturen.

```
udp-sender --interface ens3 --async --fec 8x8/128 --max-bitrate 20Mbps --file request.csr --broadcast
```

*Commando op de client*

```
udp-receiver --nosync --interface ens4  
-file request-in.csr
```

*Commando op de server*

De commando's uit de whitepaper vanuit de stakeholder werkte niet direct naar behoren, na wat research op de site van UDPcast zijn de problemen verholpen.

Hiermee was het mogelijk om het bestand request.csr te versturen vanuit de client naar de server.

Dit werkte na wat finetunen helemaal naar behoren.

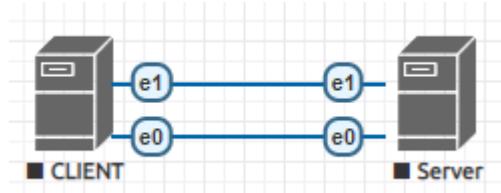
### Het automatisch versturen en ontvangen van het CSR-verzoek

Nu het versturen vanuit de client naar de server werkt, moet het verzoek ook teruggestuurd kunnen worden naar de client. Hiervoor is het nodig om meerdere handelingen achter elkaar uit te voeren, binnen Linux kan dit met Bash. Het grote voordeel van udp-receiver is dat deze tool als deze gestart wordt blijft luisteren op de interface tot dat er een bestand verzonden is. Hierdoor kan de tool in de achtergrond draaien en wordt het eenvoudiger om een bashscript goed te laten werken. De commando's uit de eerdere tests zijn samengevoegd in een Bash script.

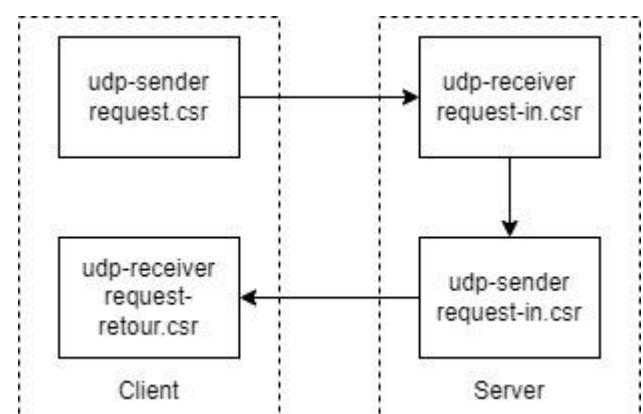
Dit werkte na het opzetten van het script direct.

Alleen moest er wel telkens op enter gedrukt worden om het versturen te starten. Dit is ook verholpen na het raadplegen van de website van UDPcast.

Het bestand werd vanaf de client naar de server verstuurd en daarna weer teruggestuurd naar



Figuur 31 CSR verzoek in testopstelling



Figuur 32 Route van de informatie

de client. De udp-receiver op de client kant past de naam aan om de 2 bestanden op de server te kunnen onderscheiden. Uiteindelijk kwamen de volgende scripts eruit.

```
#!/bin/bash
udp-sender --interface ens3 --async --fec 8x8/128 --max-bitrate
20Mbps --file request.csr --broadcast
udp-receiver --nosync --interface ens4 --file request-retour.csr
```

*Bashscript op de client*

```
#!/bin/bash
udp-receiver --nosync --interface ens3 --file request-in.csr
udp-sender --interface ens4 --async --fec 8x8/128 --max-bitrate 20Mbps
--file request-in.csr --broadcast
```

*Bashscript op de server*

### Het aanmaken van een CA in OpenSSL

Nadat het versturen en ontvangen van de CSR-informatie werkt, moet de Certificate Authority opgezet worden. Hiervoor is gebruik gemaakt voor de tool OpenSSL, deze tool zit standaard in Ubuntu. De informatie over hoe de CA aangemaakt kan worden is opgedaan uit een YouTube Video (Networklessons.com, 2017). OpenSSL maakt gebruik van standaard mappen in Ubuntu, deze kunnen aangemaakt worden in /root. Hierin dienen de volgende mappen/bestanden aanwezig te zijn voor een juiste werking.

```
mkdir /root/ca/private/
mkdir /root/ca/newcerts/
echo '1234' > /root/ca/serial
```

*Commando's voor het aanmaken van de juisten mappen/bestanden.*

Eerst moet het root certificaat aangemaakt worden, waarmee de certificaten ondertekend zullen worden. Het certificaat bestaat uit 2 onderdelen het certificaat (public key) en de private key.

Voor het certificaat moeten de gegevens van de uitgevende instantie ingevoerd worden.

En voor de private key moet er een wachtwoord opgegeven worden.

Deze wordt ingegeven bij de parameter -passout pass:"wachtwoord".

Het certificaat en private key kunnen aangemaakt worden doormiddel van de volgende commando's.

```
openssl genrsa -aes256 -out /root/ca/private/cakey.pem -passout
pass:test123 4096
```

*Aanmaken van de private key*

```
openssl req -new -x509 -key /root/ca/private/cakey.pem -out
/root/ca/cacert.pem -days 3650
```

*Aanmaken van het root certificaat.*

Wanneer de private key en het certificaat aangemaakt zijn moet er in de configuratie 1 parameter aangepast worden. Deze parameter zorgt ervoor dat de ondertekende certificaten in de juiste map komen te staan op de server. Het configuratie bestand is te benaderen via het volgende commando:

```
nano
/usr/lib/ssl/openssl.cnf
```

Onder het kopje [ CA\_default ] dient de locatie van het certificaat en de private key opgegeven

worden. Bij de waarde dir moet de map /root/ca komen te staan. Zo kan OpenSSL de benodigde bestanden vinden om het certificaat aan te maken. Hierna kunnen de certificaten aangemaakt worden via de CA server.

Nadat de CA server naar behoren werkt, is het mogelijk om CSR-verzoeken te ondertekenen met het Root certificaat. Om een juiste werking van de CA server te kunnen garanderen is het belangrijk dat er enkele parameters toegevoegd worden aan het commando. -batch is noodzakelijk voor het definitieve script, anders is het noodzakelijk voor de gebruiker om extra handelingen uit te voeren.

Dit is straks niet mogelijk bij het definitieve server script. Ook dien de parameter -passin toegevoegd te worden, anders vraagt OpenSSL om het wachtwoord van de private key. Door deze parameter is het mogelijk om het wachtwoord van tevoren te definiëren. Doormiddel van het volgende commando is het mogelijk om het CSR-verzoek te laten ondertekenen door het Root certificaat.

```
openssl ca -batch -passin pass:test123 -in request-in.csr -out signedcert-out.crt
```

*Ondertekenen van een CSR-verzoek en het aanmaken van het certificaat.*

### **Het opzetten van het script**

Nu beide onderdelen werken, kunnen deze samengevoegd worden tot 1 script.

Hier voor zijn er 2 bestanden aangemaakt op het client en server systeem.

De werking van de UDPcast en OpenSSL commando's zijn eerder in dit document beschreven.

#### **Client script**

Op het client systeem is het volgende script aangemaakt:

```
#!/bin/bash
udp-sender --interface ens3 --async --fec 8x8/128 --max-bitrate
20Mbps --file request.csr --broadcast --autostart 1
NOW=$( date '+%F-%H-%M-%S' )
udp-receiver --nosync --interface ens4 --
file ./certs/signedcert-$NOW.crt
```

*Script op het client systeem*

In dit script wordt eerst het request.csr bestand via udp-sender verzonden. Daarna wordt de variabele NOW aangemaakt, hiermee wordt de huidige tijd opgenomen voor de naam van het terugkomende certificaat. De waarde die hierin komt te staan is als volgt: 2021-12-6-19-54-23 of te wel jaar-maand-dag-uur-minuut-seconde.

Om het ondertekende certificaat terug te krijgen moet er via een separate netwerkverbinding het certificaat teruggestuurd te worden. Hier voor is er gekozen de werking van het UDPcast om te draaien, de CA server stuurt het certificaat terug via udp-sender en de client wacht weer op het certificaat via udp-receiver. Het certificaat wat ontvangen is wordt geplaatst in de map "certs" waar het script in opgeslagen is.

## Server script

Op de CA server is het volgende script aangemaakt:

```
#!/bin/bash
while true
do
    rm signedcert-out.crt
    cp error-message.crt signedcert-out.crt
    udp-receiver --nosync --interface ens3 --file request-
in.csr
    openssl ca -batch -passin pass:test123 -in request-in.csr
    -out signedcert-out.crt
    udp-sender --interface ens4 --async --fec 8x8/128 --
    max-bitrate 20Mbps --file signedcert-out.crt --broadcast --
    autostart 1
done
```

### *Script op de CA server*

Om het script in de achtergrond onbeperkt te kunnen laten draaien is het belangrijk dat het script na het voltooien hiervan opnieuw gestart wordt, dit gebeurt met de while true loop. Het script blijft telkens opnieuw lopen totdat het script geforceerd gestopt wordt met ctrl + c. Wanneer de while loop begint wordt eerst het oude certificaat verwijderd en een error melding gekopieerd. Hiermee krijgt de client een bestand terug met een foutmelding, mocht het certificaat niet aangemaakt kunnen worden. Het signedcert-out bestand heeft dan de inhoud "No cert signed".

Daarna wordt het udp-receiver commando gestart die onbeperkt blijft luisteren voor het bestand vanuit de client dat met udp-sender verzonden wordt. Het ontvangen CSR-bestand wordt dan in het bestand request-in.csr geplaatst.

Wanneer het CSR-bestand op de CA server staat, wordt via OpenSSL het request-in.csr verzoek ondertekend met het Root certificaat. OpenSSL exporteert het certificaat daarna in het bestand signedcert-out.crt. Dit certificaat dient weer teruggestuurd te worden naar de client.

Het ondertekende certificaat moet nu weer teruggestuurd worden naar de client. Hiervoor is er gekozen om het certificaat terug te sturen met udp-sender. Dit is hetzelfde commando als wat gebruikt wordt op de client om het CSR-verzoek te vesturen. Het certificaat wordt nu teruggestuurd naar de client.

Wanneer het script op het einde is wordt het script opnieuw gestart. udp-receiver zal weer gaan luisteren op de data die udp-sender vanaf de client opnieuw gaat versturen. En zo wordt het proces weer opnieuw gestart.

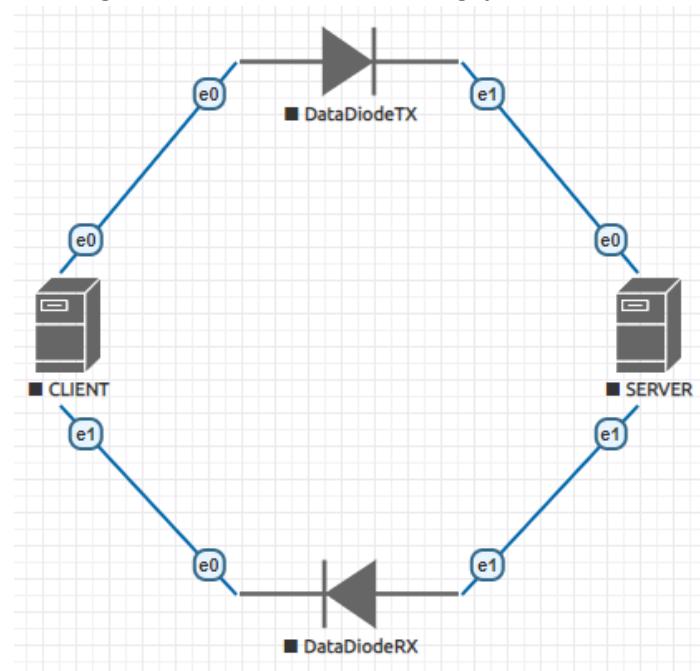
## Het toevoegen van de datadiode tussen de netwerken

In de vorige hoofdstukken is er nog geen gebruik gemaakt van een datadiode.

Dit omdat de werking van de client en server eerst getest moet worden voor dat de datadiode ertussen komt te staan.

Aangezien er zowel data verzonden als ontvangen moet worden is het belangrijk dat er 2 datadiodes in de opstelling gebruikt worden. De ene datadiode gaat gebruikt worden voor het verzenden van de data van de client naar de server. En de andere datadiode wordt gebruikt om de data van de server naar de client te sturen.

Binnen de lab tool EVE-NG heb moeten de huidige verbindingen los gemaakt worden en verbonden worden met 2 datadiodes. De virtuele datadiodes zijn opgezet volgens de handleiding van onze stakeholder (Hub, Setup OSDD virtuele ontwikkel omgeving, 2021). De werking van de scripts wordt hierdoor niet beïnvloed, maar beide netwerken zijn nu wel beveiligd door de werking van de datadiode.



Figuur 33 Datadiode in EVE-NG netwerk

## Bijlage 2: Details distribueren GPS-tijd atoomklok (NTP)

### Details virtuele test GNS3

#### Configuratie switches

R1 – NTP Server Cisco IOS 12.4

R2 – NTP Client Cisco IOS 12.4

R3 – NTP Server Cisco IOS 15.2

R4 – NTP Client Cisco IOS 15.2

```
R1#show run
Current configuration : 123 bytes
!
interface FastEthernet0/0
  ip address 10.10.10.1 255.255.255.0
  ntp broadcast destination 10.10.10.255
!
ntp master 2
!
end

R2#show run
!
interface FastEthernet0/0
  ip address 10.10.10.200 255.255.255.0
  ntp broadcast client
!
End
```

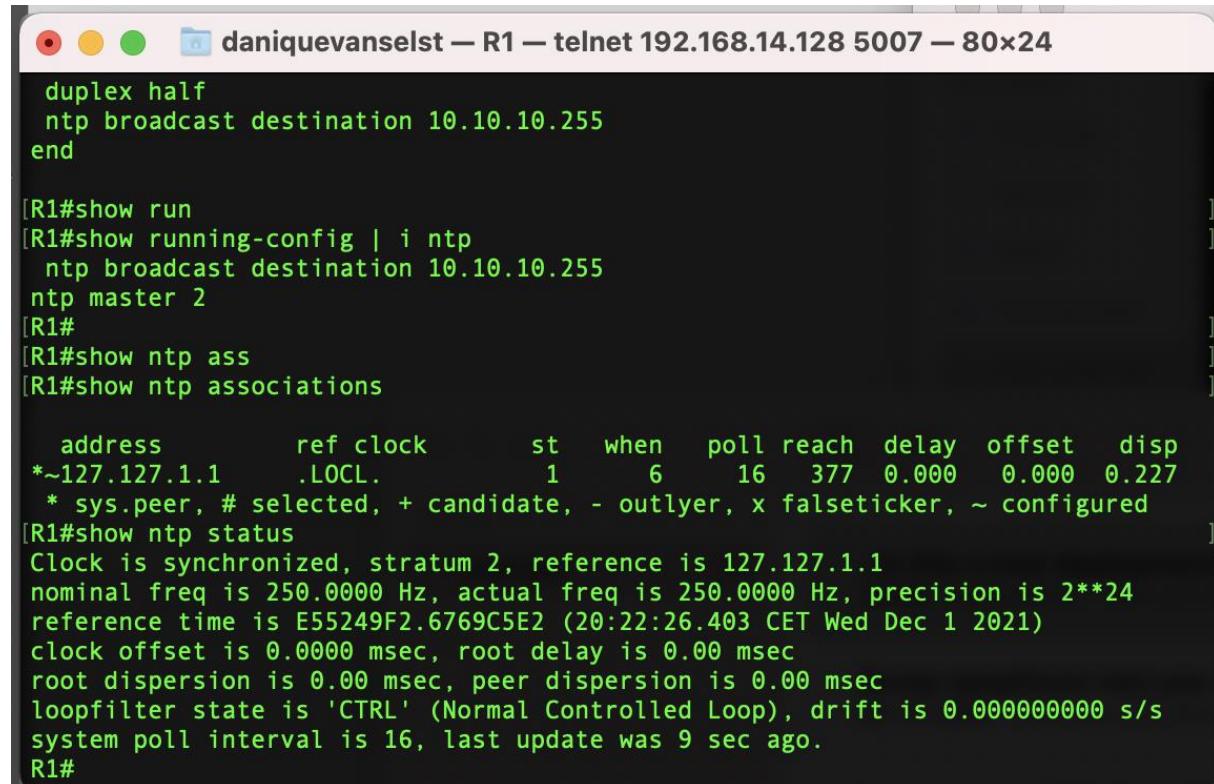
```
R3#show run
Current configuration : 123 bytes
!
interface FastEthernet0/0
  ip address 10.10.10.1 255.255.255.0
  ntp broadcast destination 10.10.10.255
!
ntp master 2
!
End
```

```
R4#show run
!
interface FastEthernet0/0
  ip address 10.10.10.200 255.255.255.0
  ntp broadcast client
!
End
```

Hier is te zien er voor beide opstellingen dezelfde configuratie is gebruikt, alleen andere softwareversies zijn gebruikt.

## Screenshots GNS3

R1 – R2 werkend met datadiode Cisco IOS 12.4



```

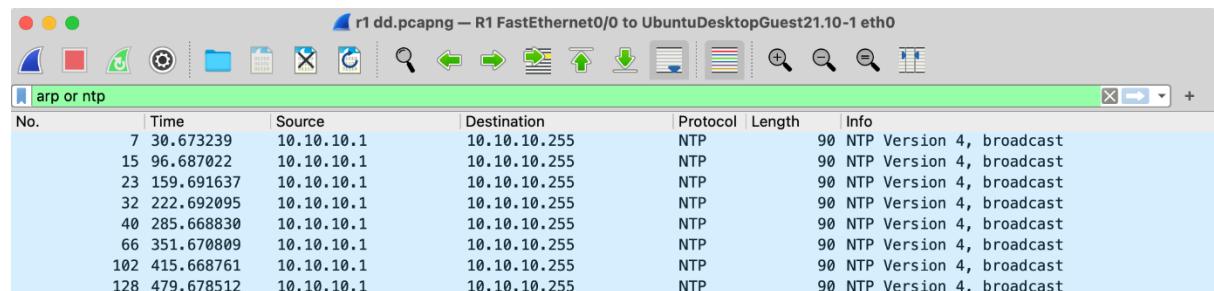
duplex half
ntp broadcast destination 10.10.10.255
end

[R1#show run
[R1#show running-config | i ntp
  ntp broadcast destination 10.10.10.255
  ntp master 2
[R1#
[R1#show ntp ass
[R1#show ntp associations

  address      ref clock      st  when  poll reach  delay  offset  disp
*~127.127.1.1      .LOCL.        1    6    16   377  0.000  0.000  0.227
  * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
[R1#show ntp status
Clock is synchronized, stratum 2, reference is 127.127.1.1
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**24
reference time is E55249F2.6769C5E2 (20:22:26.403 CET Wed Dec 1 2021)
clock offset is 0.0000 msec, root delay is 0.00 msec
root dispersion is 0.00 msec, peer dispersion is 0.00 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000000000 s/s
system poll interval is 16, last update was 9 sec ago.
R1#

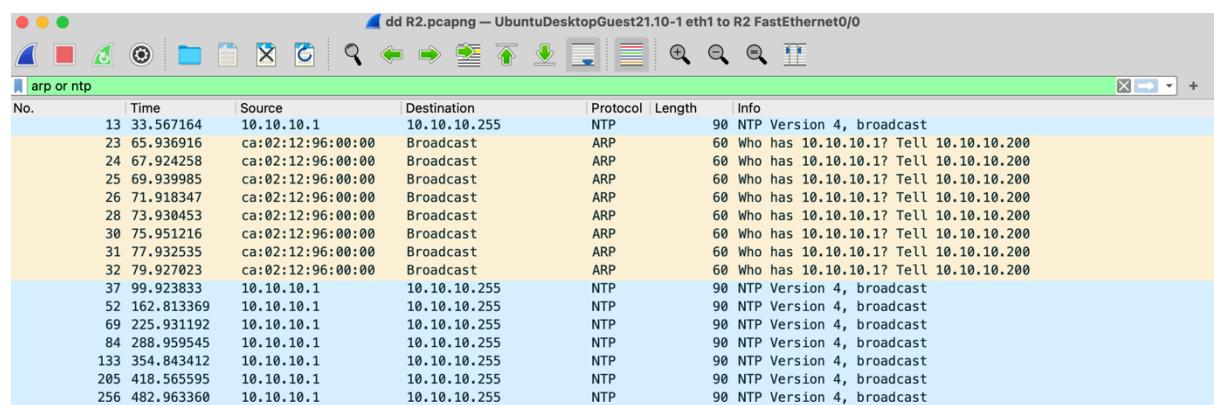
```

Figuur 34 R1 eigen clock gesynchroniseerd.



No.	Time	Source	Destination	Protocol	Length	Info
7	30.673239	10.10.10.1	10.10.10.255	NTP	90	NTP Version 4, broadcast
15	96.687022	10.10.10.1	10.10.10.255	NTP	90	NTP Version 4, broadcast
23	159.691637	10.10.10.1	10.10.10.255	NTP	90	NTP Version 4, broadcast
32	222.692095	10.10.10.1	10.10.10.255	NTP	90	NTP Version 4, broadcast
40	285.668830	10.10.10.1	10.10.10.255	NTP	90	NTP Version 4, broadcast
66	351.670809	10.10.10.1	10.10.10.255	NTP	90	NTP Version 4, broadcast
102	415.668761	10.10.10.1	10.10.10.255	NTP	90	NTP Version 4, broadcast
128	479.678512	10.10.10.1	10.10.10.255	NTP	90	NTP Version 4, broadcast

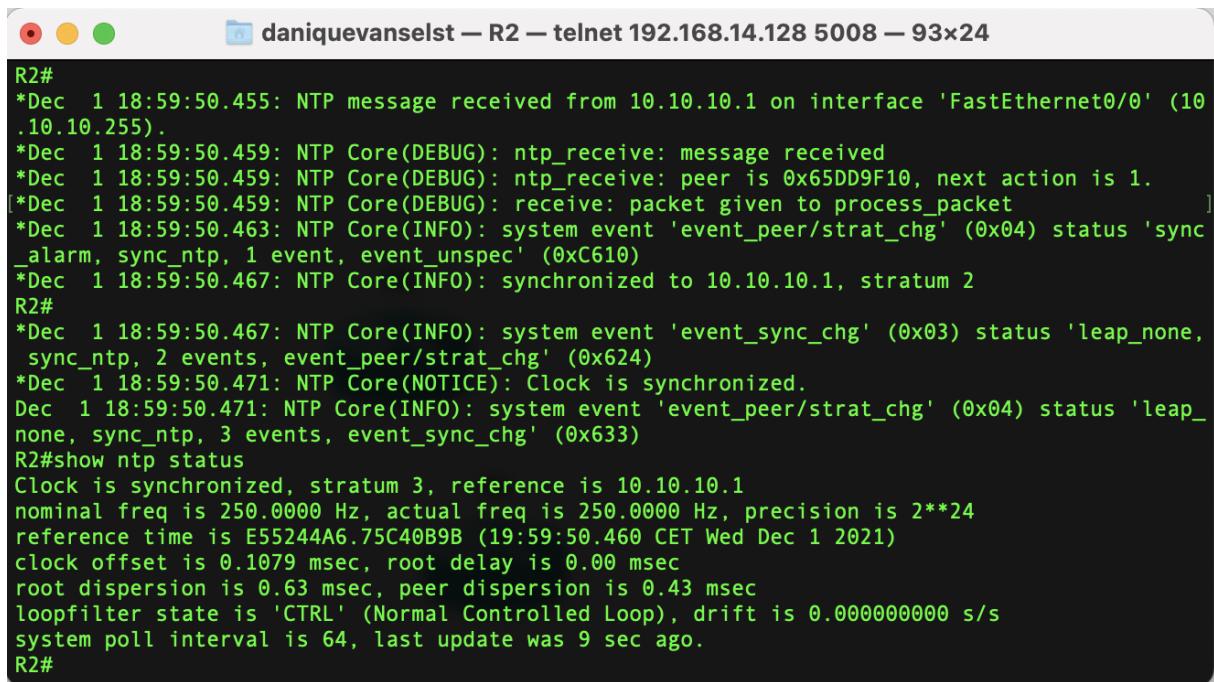
Figuur 35 R1 Wireshark waar de NTP-broadcast messages worden verstuurd.



No.	Time	Source	Destination	Protocol	Length	Info
13	33.567164	10.10.10.1	10.10.10.255	NTP	90	NTP Version 4, broadcast
23	65.936916	ca:02:12:96:00:00	Broadcast	ARP	60	Who has 10.10.10.1? Tell 10.10.10.200
24	67.924258	ca:02:12:96:00:00	Broadcast	ARP	60	Who has 10.10.10.1? Tell 10.10.10.200
25	69.939985	ca:02:12:96:00:00	Broadcast	ARP	60	Who has 10.10.10.1? Tell 10.10.10.200
26	71.918347	ca:02:12:96:00:00	Broadcast	ARP	60	Who has 10.10.10.1? Tell 10.10.10.200
28	73.930453	ca:02:12:96:00:00	Broadcast	ARP	60	Who has 10.10.10.1? Tell 10.10.10.200
30	75.951216	ca:02:12:96:00:00	Broadcast	ARP	60	Who has 10.10.10.1? Tell 10.10.10.200
31	77.932535	ca:02:12:96:00:00	Broadcast	ARP	60	Who has 10.10.10.1? Tell 10.10.10.200
32	79.927023	ca:02:12:96:00:00	Broadcast	ARP	60	Who has 10.10.10.1? Tell 10.10.10.200
37	99.923833	10.10.10.1	10.10.10.255	NTP	90	NTP Version 4, broadcast
52	162.813369	10.10.10.1	10.10.10.255	NTP	90	NTP Version 4, broadcast
69	225.931192	10.10.10.1	10.10.10.255	NTP	90	NTP Version 4, broadcast
84	288.959545	10.10.10.1	10.10.10.255	NTP	90	NTP Version 4, broadcast
133	354.843412	10.10.10.1	10.10.10.255	NTP	90	NTP Version 4, broadcast
205	418.565595	10.10.10.1	10.10.10.255	NTP	90	NTP Version 4, broadcast
256	482.963360	10.10.10.1	10.10.10.255	NTP	90	NTP Version 4, broadcast

Figuur 36 R2 Wireshark na de eerste NTP-broadcast message

R2 Wireshark na de eerste NTP-broadcast message wordt er geprobeerd de NTP-server te benaderen, dit lukt niet want er komt geen ARP-reply terug. Hierna worden er geen ARP-requesten meer gestuurd en worden de NTP-broadcast messages geaccepteerd en daarmee de clock gesynchroniseerd.



```

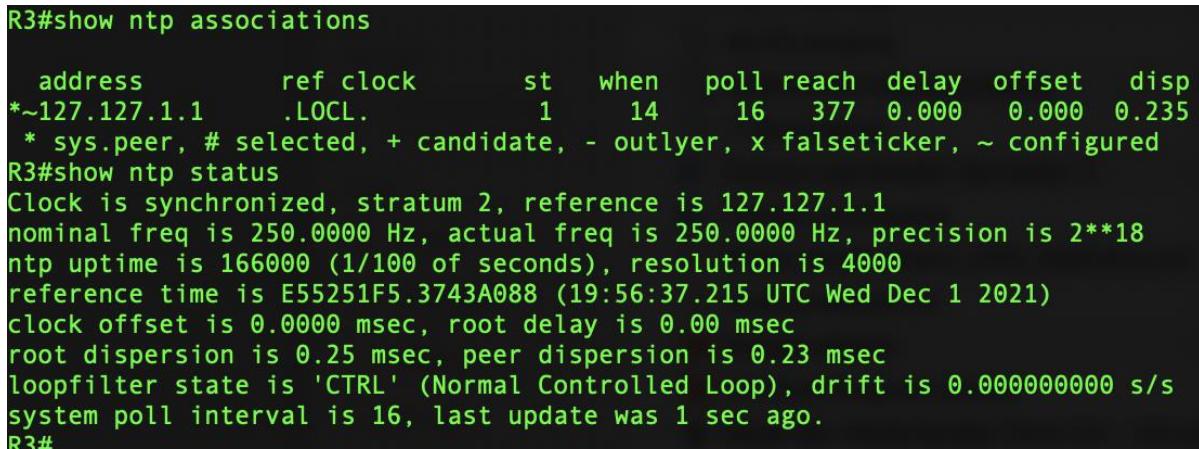
R2#
*Dec 1 18:59:50.455: NTP message received from 10.10.10.1 on interface 'FastEthernet0/0' (10.10.10.255).
*Dec 1 18:59:50.459: NTP Core(DEBUG): ntp_receive: message received
*Dec 1 18:59:50.459: NTP Core(DEBUG): ntp_receive: peer is 0x65DD9F10, next action is 1.
*Dec 1 18:59:50.459: NTP Core(DEBUG): receive: packet given to process_packet
*Dec 1 18:59:50.463: NTP Core(INFO): system event 'event_peer/strat_chg' (0x04) status 'sync_alarm, sync_ntp, 1 event, event_unspec' (0xC610)
*Dec 1 18:59:50.467: NTP Core(INFO): synchronized to 10.10.10.1, stratum 2
R2#
*Dec 1 18:59:50.467: NTP Core(INFO): system event 'event_sync_chg' (0x03) status 'leap_none, sync_ntp, 2 events, event_peer/strat_chg' (0x624)
*Dec 1 18:59:50.471: NTP Core(NOTICE): Clock is synchronized.
Dec 1 18:59:50.471: NTP Core(INFO): system event 'event_peer/strat_chg' (0x04) status 'leap_none, sync_ntp, 3 events, event_sync_chg' (0x633)
R2#show ntp status
Clock is synchronized, stratum 3, reference is 10.10.10.1
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**24
reference time is E55244A6.75C40B9B (19:59:50.460 CET Wed Dec 1 2021)
clock offset is 0.1079 msec, root delay is 0.00 msec
root dispersion is 0.63 msec, peer dispersion is 0.43 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000000000 s/s
system poll interval is 64, last update was 9 sec ago.
R2#

```

Figuur 37 R2 clock is synchronized

R2 clock is synchronized, met de debugs waar te zien is dat alles werkt.

R3—R3 niet werkend met datadiode Cisco IOS 15.2



```

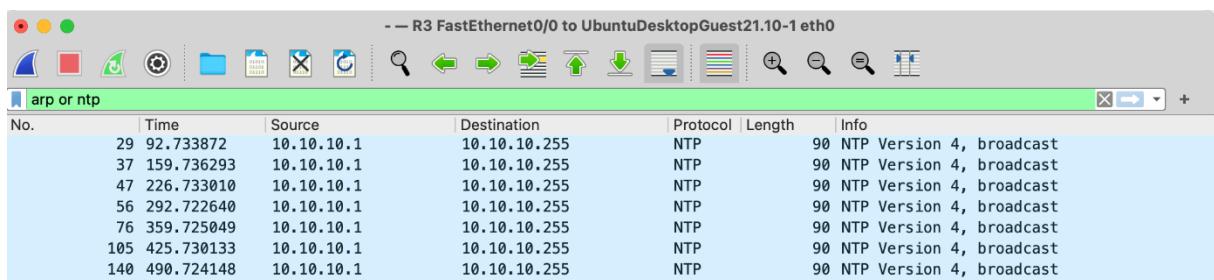
R3#show ntp associations

  address      ref clock      st  when  poll reach  delay  offset  disp
*~127.127.1.1      .LOCL.      1    14     16   377  0.000  0.000  0.235
  * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured

R3#show ntp status
Clock is synchronized, stratum 2, reference is 127.127.1.1
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**18
ntp uptime is 166000 (1/100 of seconds), resolution is 4000
reference time is E55251F5.3743A088 (19:56:37.215 UTC Wed Dec 1 2021)
clock offset is 0.0000 msec, root delay is 0.00 msec
root dispersion is 0.25 msec, peer dispersion is 0.23 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000000000 s/s
system poll interval is 16, last update was 1 sec ago.
R3#

```

Figuur 38 R3 eigen clock synchronized



Figuur 39 R3 Wireshark waar de NTP-broadcast messages worden getoond.

No.	Time	Source	Destination	Protocol	Length	Info
54	91.337012	10.10.10.1	10.10.10.255	NTP	90	NTP Version 4, broadcast
56	92.231751	ca:04:3e:49:00:00	Broadcast	ARP	60	Who has 10.10.10.17 Tell 10.10.10.200
71	157.228580	ca:04:3e:49:00:00	Broadcast	ARP	60	Who has 10.10.10.17 Tell 10.10.10.200
72	158.312133	10.10.10.1	10.10.10.255	NTP	90	NTP Version 4, broadcast
73	159.223729	ca:04:3e:49:00:00	Broadcast	ARP	60	Who has 10.10.10.17 Tell 10.10.10.200
92	223.224651	ca:04:3e:49:00:00	Broadcast	ARP	60	Who has 10.10.10.17 Tell 10.10.10.200
93	225.383035	10.10.10.1	10.10.10.255	NTP	90	NTP Version 4, broadcast
94	227.226959	ca:04:3e:49:00:00	Broadcast	ARP	60	Who has 10.10.10.17 Tell 10.10.10.200
109	288.220449	ca:04:3e:49:00:00	Broadcast	ARP	60	Who has 10.10.10.17 Tell 10.10.10.200
111	291.032311	10.10.10.1	10.10.10.255	NTP	90	NTP Version 4, broadcast
113	292.222806	ca:04:3e:49:00:00	Broadcast	ARP	60	Who has 10.10.10.17 Tell 10.10.10.200
147	352.224489	ca:04:3e:49:00:00	Broadcast	ARP	60	Who has 10.10.10.17 Tell 10.10.10.200
149	354.218900	ca:04:3e:49:00:00	Broadcast	ARP	60	Who has 10.10.10.17 Tell 10.10.10.200
153	356.228166	ca:04:3e:49:00:00	Broadcast	ARP	60	Who has 10.10.10.17 Tell 10.10.10.200
155	358.460762	10.10.10.1	10.10.10.255	NTP	90	NTP Version 4, broadcast
156	359.219166	ca:04:3e:49:00:00	Broadcast	ARP	60	Who has 10.10.10.17 Tell 10.10.10.200
207	418.226607	ca:04:3e:49:00:00	Broadcast	ARP	60	Who has 10.10.10.17 Tell 10.10.10.200
209	421.224286	ca:04:3e:49:00:00	Broadcast	ARP	60	Who has 10.10.10.17 Tell 10.10.10.200
214	424.226315	ca:04:3e:49:00:00	Broadcast	ARP	60	Who has 10.10.10.17 Tell 10.10.10.200
215	424.257306	10.10.10.1	10.10.10.255	NTP	90	NTP Version 4, broadcast
217	427.218774	ca:04:3e:49:00:00	Broadcast	ARP	60	Who has 10.10.10.17 Tell 10.10.10.200
218	429.223379	ca:04:3e:49:00:00	Broadcast	ARP	60	Who has 10.10.10.17 Tell 10.10.10.200
266	483.226137	ca:04:3e:49:00:00	Broadcast	ARP	60	Who has 10.10.10.17 Tell 10.10.10.200
271	485.222097	ca:04:3e:49:00:00	Broadcast	ARP	60	Who has 10.10.10.17 Tell 10.10.10.200
287	489.151859	10.10.10.1	10.10.10.255	NTP	90	NTP Version 4, broadcast
288	489.219656	ca:04:3e:49:00:00	Broadcast	ARP	60	Who has 10.10.10.17 Tell 10.10.10.200
295	493.227443	ca:04:3e:49:00:00	Broadcast	ARP	60	Who has 10.10.10.17 Tell 10.10.10.200

Figuur 40 R4 Wireshark na iedere NTP-broadcast

R4 Wireshark na iedere NTP-broadcast messages wordt geprobeerd de server te benaderen.

```
*Dec  1 19:36:42.031: NTP Core(DEBUG): ntp_receive: message received
*Dec  1 19:36:42.031: NTP Core(DEBUG): ntp_receive: peer is 0x00000000, next action is 6.
*Dec  1 19:36:42.099: NTP message sent to 10.10.10.1, from interface 'FastEthernet0/0' (10.10.
10.200).
R4#
*Dec  1 19:36:43.099: NTP message sent to 10.10.10.1, from interface 'FastEthernet0/0' (10.10.
10.200).
R4#
*Dec  1 19:36:46.099: NTP message sent to 10.10.10.1, from interface 'FastEthernet0/0' (10.10.
10.200).
*Dec  1 19:36:46.099: NTP message sent to 10.10.10.1, from interface 'FastEthernet0/0' (10.10.
10.200).
```

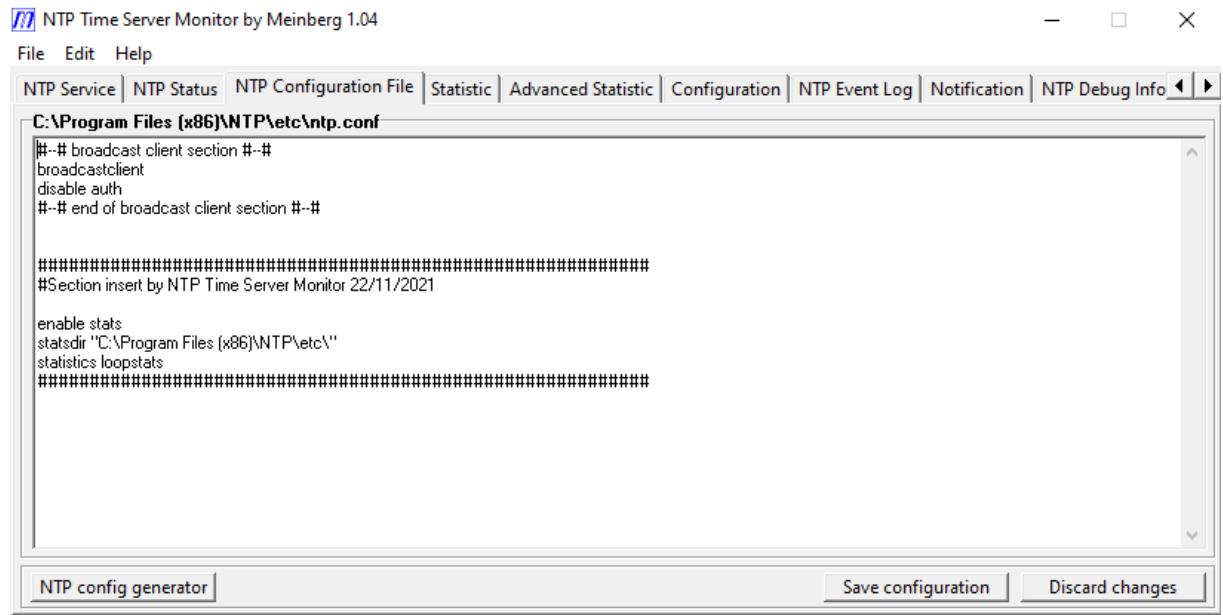
Figuur 41 R4 NTP debug

Figuur 42 R4 unsynchronized, NTP associations staan op INIT

R4 unsynchronized, NTP associations staan op INIT. Bij iedere broadcast message wordt een extra server address toegevoegd. Dit komt door de eerde genoemde problemen met de 'volley' waarvoor bi-directioneel verkeer benodigd is.

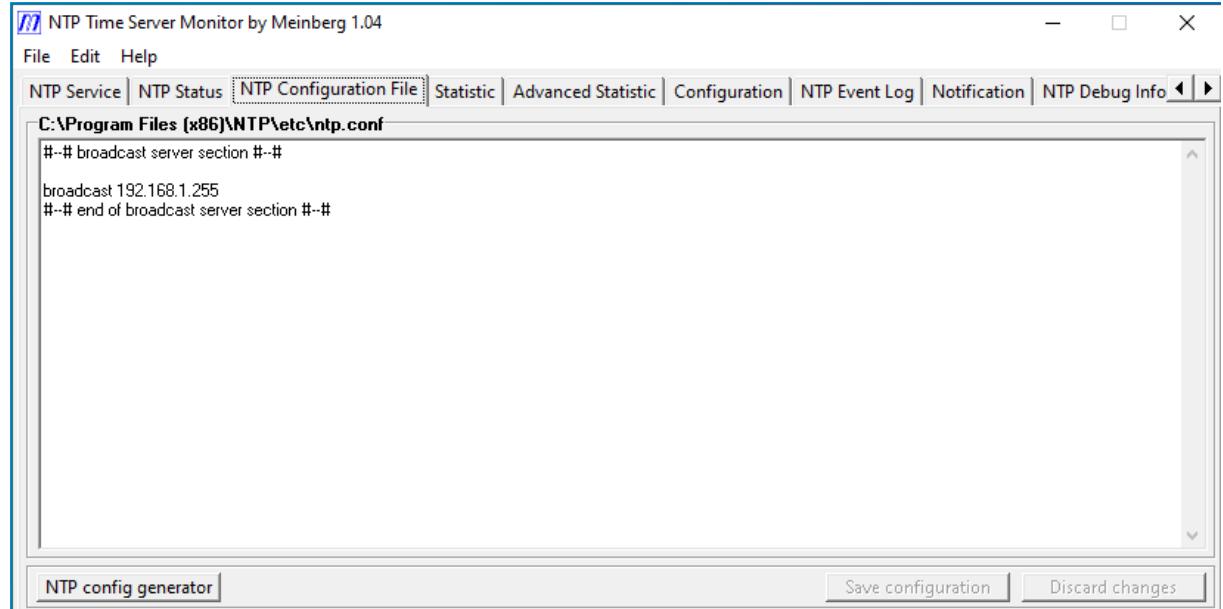
## Details virtuele test Windows NTPd (Meinberg) voor Windows

Voor de client hebben is de volgende configuratie gebruikt.



Figuur 43 Meinberg NTP Time server

Aan de server kant is onderstaande configuratie in het programma gezet.

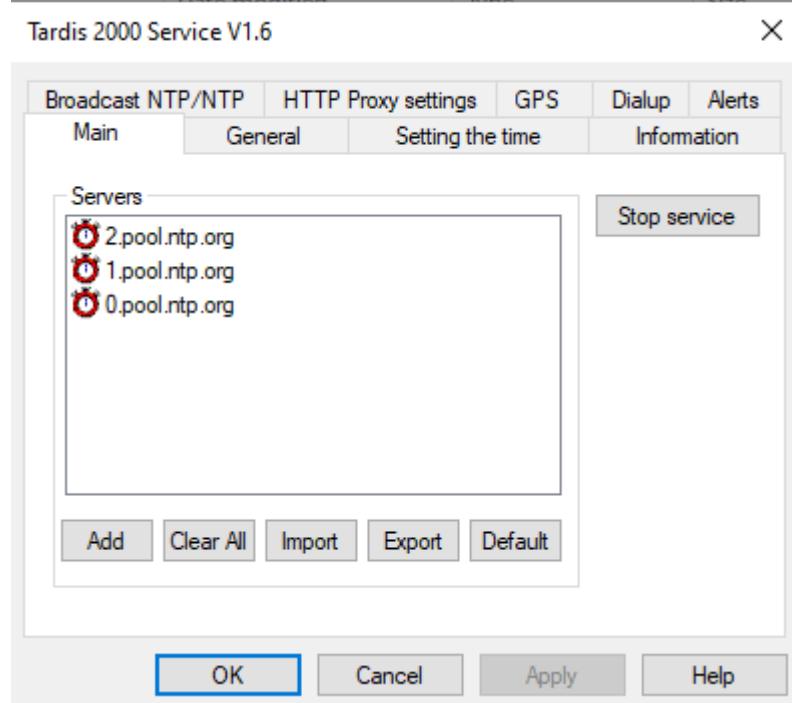


Figuur 44 Parameters Meinberg NTP Time server

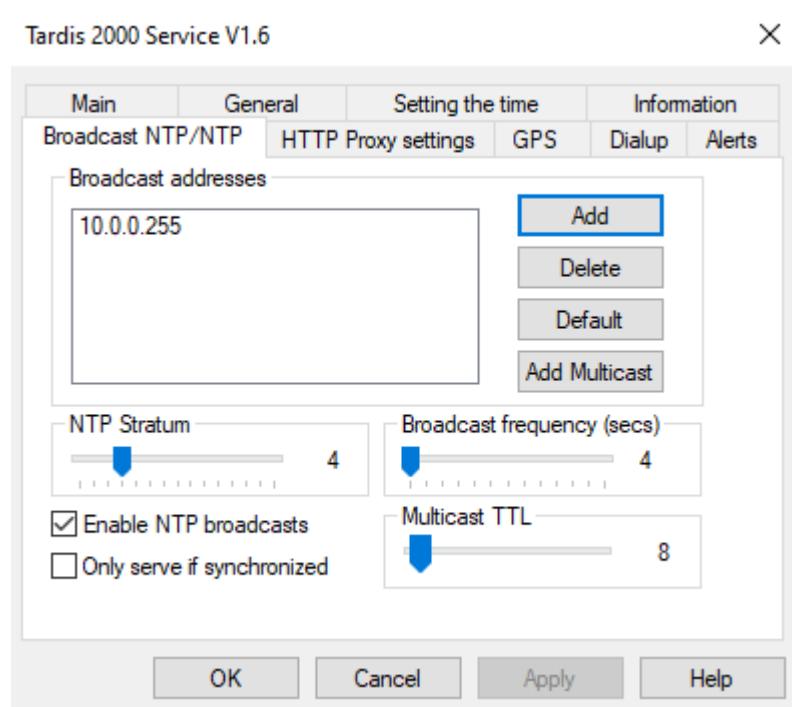
Hierbij is ook de firewall van Windows uitgezet om zeker te weten dat er voor de test geen problemen konden ontstaan.

## Instellingen Tardis2000 NTP Broadcast

Instellingen server 10.0.0.1



Figuur 45 Instellingen server 10.0.0.1



Figuur 46 Instellingen server 10.0.0.1

## Details fysieke test

### Instellingen Elproma NTS-3000 time-server

**NTS-3000 Configuration Panel**

[www.clepsydratime.com](http://www.clepsydratime.com)

**Interfaces**

- LAN1
- LAN2
- Services
- SYSLOG
- SNMP
- NTP
- Date/time
- Manual set
- Time Zone
- Authentication
- Password
- NTP MD5 Keys
- SSH Key
- SSL Key
- RADIUS
- Miscellaneous
- DNS
- Antenna direction
- System
- GPS status data
- Save settings
- Logout

**Service enable**  **Enabled**  
Set this option to enable communication over LAN1 interface

**IP Address** 10.0.0.1 **OK**  
Enter IP address here

**Mask** 255.255.255.0 **OK**  
Enter mask here

**Gateway** **Error**  
Enter gateway address here

**IPv6 Address** **Error**  
Enter canonical IPv6 address here

**Prefixlength (Bits)** 64 **OK**  
Enter prefixlength here

**NTP Broadcast**  **Enabled**  
If this option is set, NTP will work in broadcast mode over LAN1  
10.0.0.255 **OK**  
Enter broadcast range here  
 **Enabled**  
Set this option to enable encrypting for broadcast

**Antenna socket direction** Input NMEA GPS/GLOANSS  
Select antenna A socket direction  
Disabled  
Select antenna B socket direction

Figuur 47 Instellingen Elproma NTS-3000 time-server

**NTS-3000 Configuration Panel**

[www.clepsydratime.com](http://www.clepsydratime.com)

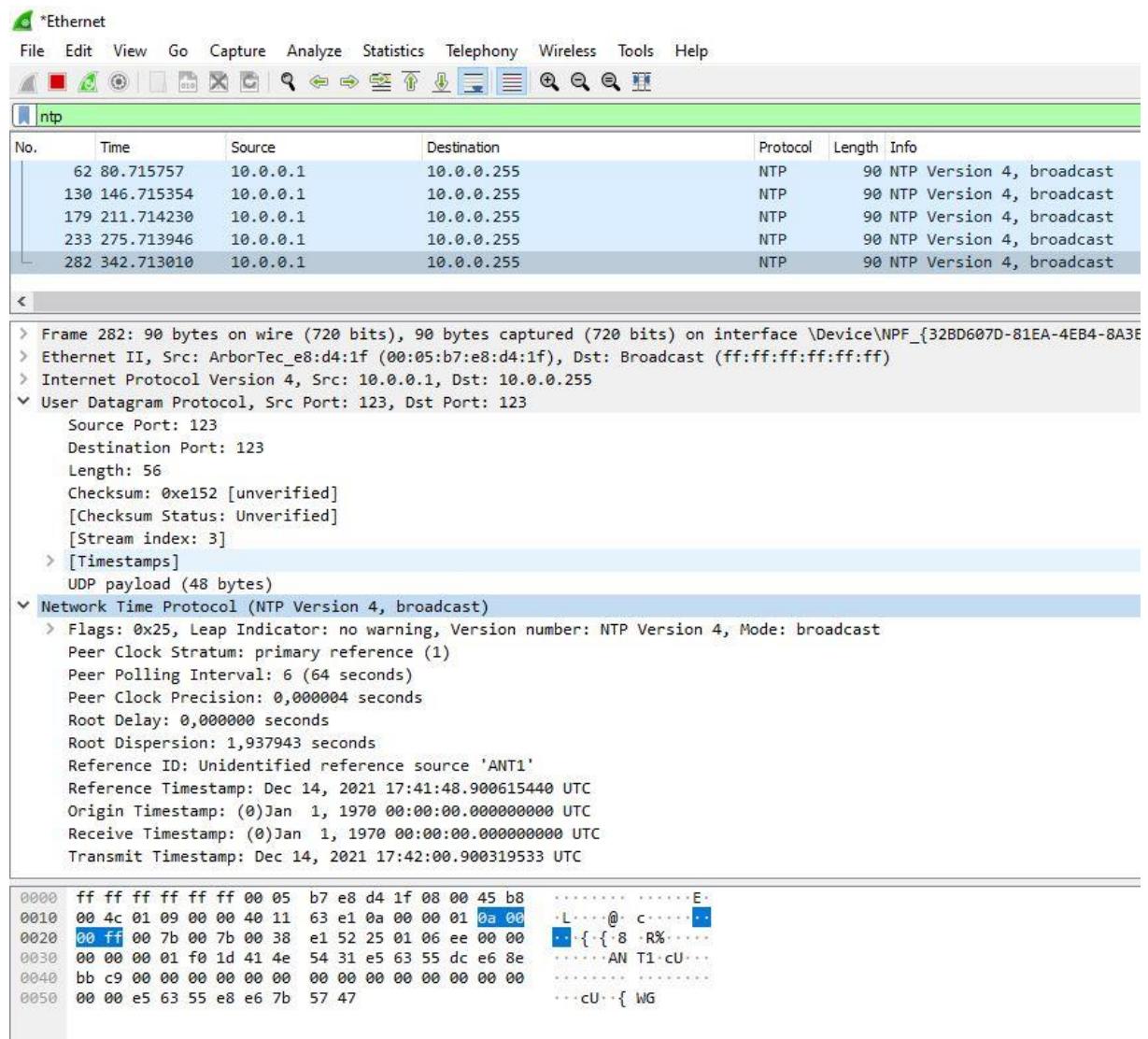
**Interfaces**

- LAN1
- LAN2
- Services
- SYSLOG
- SNMP
- NTP
- Date/time
- Manual set
- Time Zone
- Authentication
- Password
- NTP MD5 Keys
- SSH Key
- SSL Key
- RADIUS
- Miscellaneous
- DNS
- Antenna direction
- System
- GPS status data
- Save settings
- Logout

**Antenna socket direction** Input NMEA GPS/GLOANSS  
Select antenna A socket direction  
Disabled  
Select antenna B socket direction

Figuur 48 Instellingen Elproma NTS-3000 time-server

## Wireshark capture fysieke test



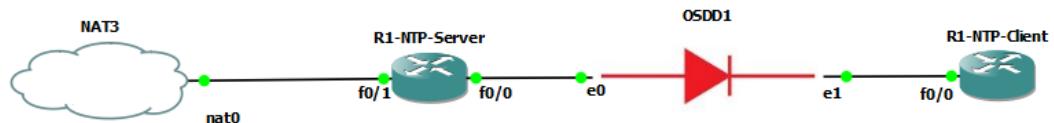
Figuur 49 Wireshark capture fysieke test

Tijdens de test met fysieke apparatuur is het gelukt NTP-broadcast te laten werken met de K9 applicatie op het Windows platform. Uit figuur 49 blijkt dat er alleen nog maar NTP-broadcastpakketten over het netwerk worden verstuurd en dat er geen traditioneel NTP-verkeer meer is. Belangrijk om te vermelden is dat het om NTPv4 gaat.

## Bijlage 3: Handleiding GNS3WIN-OSDD

### Inleiding

Deze handleiding is bedoeld om de virtuele variant van de open source datadiode op te zetten in GNS3 en hierbij is enige netwerkkennis vereist om bepaalde configuraties te bouwen. Het doel is dat door deze handleiding het opzetten een stuk eenvoudiger wordt. De opstelling die gemaakt wordt middels deze handleiding is te zien in figuur 50.



Figuur 50 De te bouwen opstelling in GNS3

GNS3 is netwerkemulatiesoftware (soort virtualisatiesoftware) waarbij allerlei soorten en merken netwerkapparatuur gebruikt kan worden voor bijvoorbeeld testdoeleinden. Daarnaast wordt tijdens de installatie van GNS3 ook Wireshark geïnstalleerd (als dit pakket nog niet geïnstalleerd is) en hiermee kunnen packet captures worden gedaan. Het advies is dan ook om GNS3 te gebruiken om de open source datadiode virtueel te testen met verschillende soorten netwerkapparatuur.

Er zijn systeemeisen voor de installatie en het gebruik van GNS3 op Windows. Houdt hier rekening mee en zorg ervoor dat het systeem geschikt is en over voldoende resources beschikt om het werk te kunnen doen. In tabel 14 staan de systeemeisen weergegeven.

Item	Minimum req.	Recommended req.	Optimal req.
<b>Operating System</b>	Windows 7 (64bit) or later	Windows 7 (64bit) or later	Windows 7 (64bit) or later
<b>Processor</b>	2 or more logical cores	4 or more logical cores – AMD-V/RVI or Intel VT-X/EPT	8 or more logical cores – Core i7 or i9 Intel / R7 or R9 AMD - AMD-V/RVI or Intel VT-X/EPT
<b>Virtualization</b>	Virtualization extensions required	Virtualization extensions required	Virtualization extensions required
<b>Memory</b>	4 GB RAM	16 GB RAM	32 GB RAM
<b>Storage</b>	1GB available (Windows install is <200 MB)	SSD with 35 GB available	SSD with 80 GB available

Tabel 14 Systeemeisen Windows GNS3 (bron: <https://docs.gns3.com/docs/getting-started/installation/windows>)

## Installatie GNS3 op Windows

Om met de installatie te kunnen beginnen moeten er een aantal bestanden worden gedownload. Hiervoor moet eenmalig geregistreerd worden. De volgende pakketten dienen gedownload te worden:

- GNS3 Windows (<https://gns3.com/software/download>)
- GNS3 VM for VMware Workstation (<https://gns3.com/software/download-vm>) of
- GNS3 VM for VMware ESXi (<https://gns3.com/software/download-vm>) of
- GNS3 VM for Microsoft Hyper-V (<https://gns3.com/software/download-vm>) of
- GNS3 VM for Virtualbox

Voor optimale werking is het belangrijk dat ook de VM van GNS3 wordt meegenomen. Voor welke software of platform is per situatie verschillend en dient de gebruiker zelf te selecteren. Als er wordt gekozen voor een GNS3 VM onder Windows dan kan dit tijdens de setup van GNS3 (zie link). Deze VM moet vervolgens geïmporteerd worden in het virtualisatieplatform dat gebruikt wordt.

## Setup GNS3

Aan het einde van de installatie zijn de volgende pakketten geïnstalleerd:

- GNS3
- WinPcap
- Npcap
- Wireshark

Stappen:

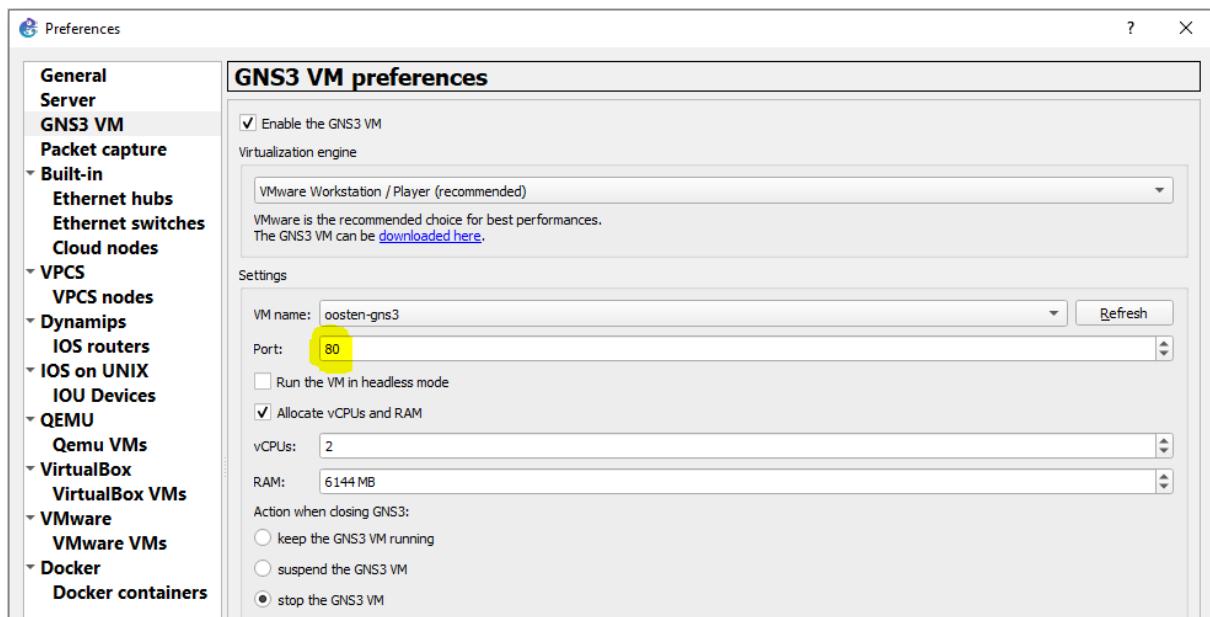
1. Start “GNS3-x.x.x.-all-in-one-regular.exe”.
2. Klik next tot u komt bij ‘Choose Components’.
3. Vink GNS3 Webclient en GNS3 VM.
4. Kies de aangemaakte VM en zet settings naar keuze.
5. Vervolg de installatie totdat deze volledig is voltooid.

## Opstarten GNS3 en configureren

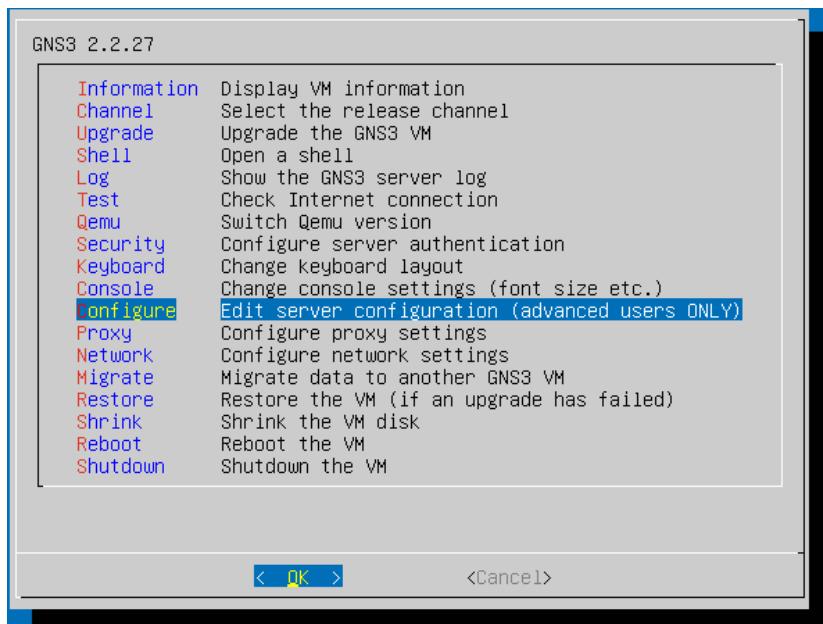
Als de installatie gelukt is kan GNS3 worden opgestart en af geconfigureerd. Bij een goede installatie start de VM automatisch zodra het programma opgestart wordt. De eerste dat GNS3 wordt opgestart wordt er een 'Setup Wizard' gestart en het is belangrijk dat deze voltooid wordt.

Als eerste moet er bepaald worden op welk systeem simulaties gestart moeten worden. Hierbij is het handig om te kiezen dat draaien in de GNS3 VM. Als tweede worden er instellingen gezet m.b.t. de local server configuration. Hier hoeft niets aan veranderd te worden. Als derde en laatste optie kunt u, mocht dat niet zijn gedaan, de GNS3 VM selecteren en instellen hoeveel vCPU cores en hoeveel geheugen u deze wilt toewijzen. (Tip: wees niet te zuinig, maar let op de resources van de host)

Als de GNS3 VM op dezelfde host draait als de software is het handig om poort 80 toe te wijzen aan de VM, omdat hiermee mogelijke firewall issues (lokaal) worden voorkomen. Dit moet gedaan worden op twee plekken: in de applicatie en op de VM zelf. Figuur 51, 52 en 53 geven weer waar en hoe dit gedaan moet worden.



Figuur 51 Application GNS3 VM preferences change port

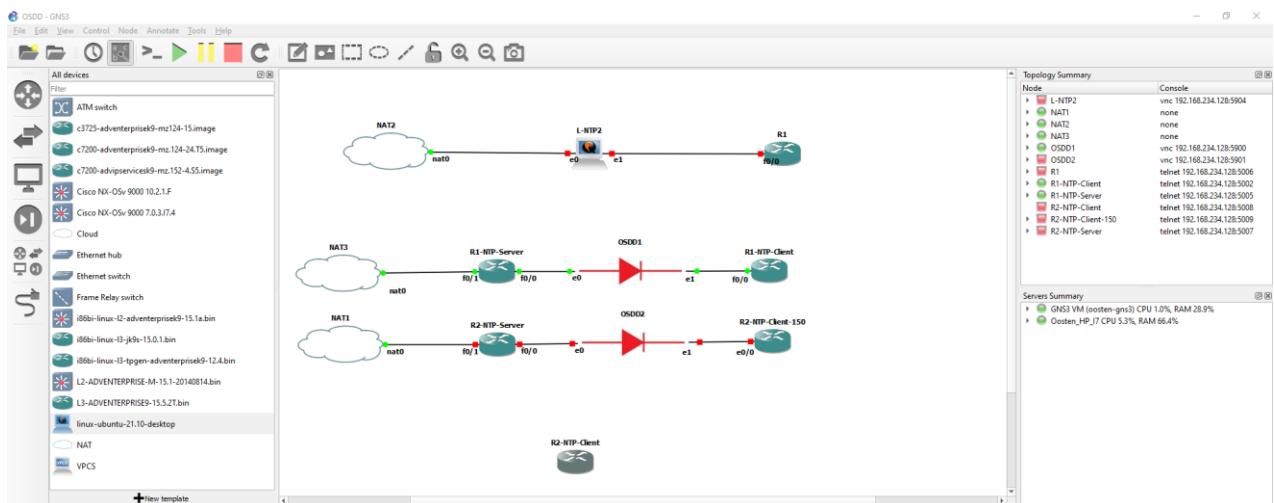


Figuur 52 GNS3 VM Menu

```
GNU nano 4.8
[Server]
host = 0.0.0.0
port = 80
images_path = /opt/gns3/images
projects_path = /opt/gns3/projects
report_errors = True
auth = False
```

Figuur 53 GNS3 VM gns3\_server.conf for changing the port

Als de poort nummers zijn gewijzigd is het verstandig om de GNS3 applicatie en VM opnieuw op te starten voor een correcte werking. Als dit gelukt is, is de configuratie voltooid en kan er begonnen worden met het toevoegen van templates (apparaten) en opzetten van de open source datadiode in GNS3. Figuur 54 geeft een project in GNS3 weer waarbij ook templates zijn toegevoegd.



Figuur 54 GNS3 project

## Templates toevoegen aan GNS3

Zonder templates is het opzetten van een project een beetje karig en daarom is het handig om templates toe te voegen aan GNS3. Sowieso is de applicatie standaard niet voorzien van bijvoorbeeld een Linux template en daarom wordt deze handmatig toegevoegd.

De lijst met templates die toegevoegd kunnen worden is ontzettend groot. Hierbij moet gedacht worden aan firewalls, guest OS (Linux, Windows etc.), routers en switches. Deze handleiding beperkt zich tot het toevoegen van Linux (Ubuntu) en Dynamips emulator (Cisco IOS router templates), welke beschikbaar zijn op internet.

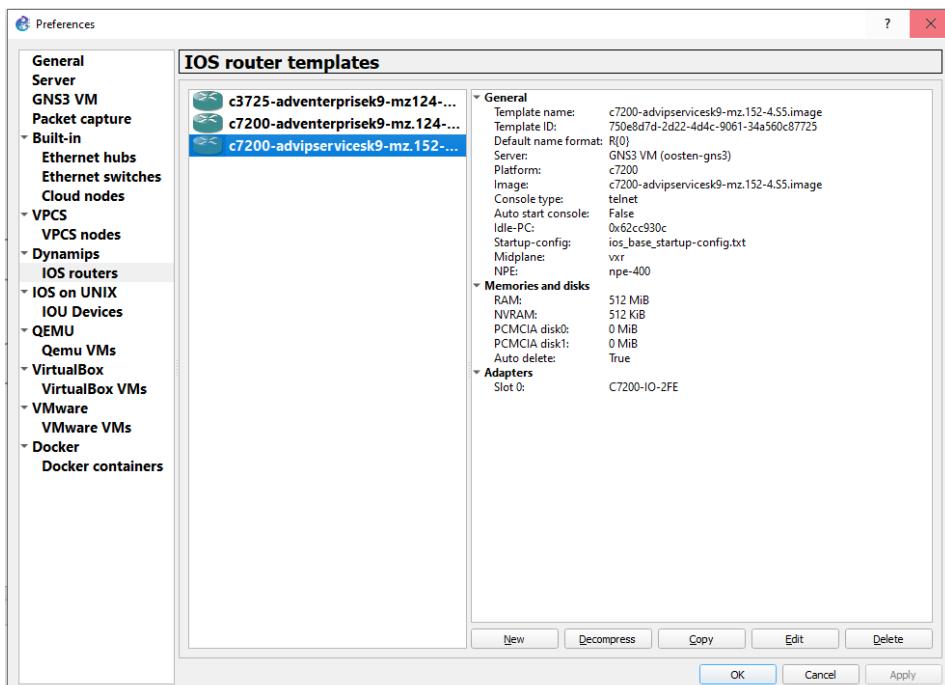
### Dynamips

Dynamips zijn IOS-router templates van Cisco, welke zijn gemaakt voor netwerkemulatiesoftware en deze zijn te downloaden op internet. In dit voorbeeld wordt een variant van de C7200 router toegevoegd. Dit gebeurt in het 'preferences' menu waar eerder al instellingen zijn aangepast.

Stappenplan toevoegen Dynamips template:

1. Ga naar Dynamips.
2. IOS-routers en klik op new.
3. Selecteer 'Run this IOS router on the GNS3 VM' en klik next.
4. Selecteer 'New Image' vervolgens op browse en zoek de locatie van het template dat gebruikt gaat worden en klik next.
5. Geef het template een naam en kies het platform en eventueel chassis en klik next.
6. Laat het RAM-geheugen staan en klik next.
7. Het aantal sloten kan blijven staan maar eventueel naar wens worden veranderd en klik next.
8. Klik op 'Idle-PC finder' en vervolgens op finish.

Met het doorlopen van deze stappen is het template toegevoegd aan GNS3 en deze kan gebruikt worden in een project. De instellingen die zijn gemaakt tijdens de wizard kunnen altijd worden aangepast. In figuur 55 is te zien dat er in dit geval 3 templates zijn toegevoegd.



Figuur 55 Dynamips IOS router templates

## QEMU

QEMU VMs is een open source hypervisor. Het emuleert de processor van een machine door de dynamische binaire vertaling. Het werkt als Kernel-based Virtual Machine (KVM) om virtuele machines op native snelheid te laten draaien. Het kan ook emulatie uitvoeren voor processen op gebruikersniveau, waardoor toepassingen die voor een bepaalde architectuur zijn gecompileerd op een andere kunnen worden uitgevoerd. Hierdoor is het mogelijk om in GNS3 meerdere types toe te voegen. De meeste gebruikte extensie zijn de zogenaamde QCOW2 files welke van QEMU zelf zijn. Ook is het mogelijk om bijvoorbeeld bestanden te gebruiken met de VMDK-extensie. Hierbij is het aan te raden om een VMDK te converteren naar QCOW2, omdat deze extensie in meerdere netwerkemulatiesoftware wordt gebruikt. In de Linux wereld wordt hiervoor het programma 'qemu-img' gebruikt en deze software is ook beschikbaar voor Windows. De meeste Linux distributies hebben al de beschikking over de software. Mocht dit niet het geval zijn dan kan dit simpel met het volgende commando: 'sudo apt-get install qemu-img'. De Windows variant van deze software is te downloaden op de volgende website: <https://www.qemu.org/download/#windows>

De virtuele OSDD draait op Ubuntu en in onderstaand stappenplan staat beschreven hoe dit op te zetten is:

1. Download de Ubuntu ISO (<https://ubuntu.com>).
2. Maak een nieuwe VM aan met VMware Workstation, Hyper-V of een ander virtualisatieplatform. Zorg ervoor dat de virtual harddisk opgeslagen wordt als één bestand en geef deze 20GB, voeg twee netwerkadapters toe en installeer Ubuntu.
3. Als de installatie van Ubuntu is voltooid, installeer dan alvast de volgende packages (sudo apt-get install):
  - a. Net-tools
  - b. Pv
  - c. Daemonlogger
  - d. Udpcast
4. Pas de instellingen van de netwerkadapters aan: sudo nano /etc/network/interfaces

```
auto TX-veranderen
iface TX-veranderen inet manual
auto RX-veranderen
iface RX-veranderen inet manual
```

Voorbeeld:

```
auto ens3
iface ens3 inet manual
auto ens4
iface ens4 inet manual
```

5. Maak een nieuw file aan en kopieer daarin het virtuele OSDD script:

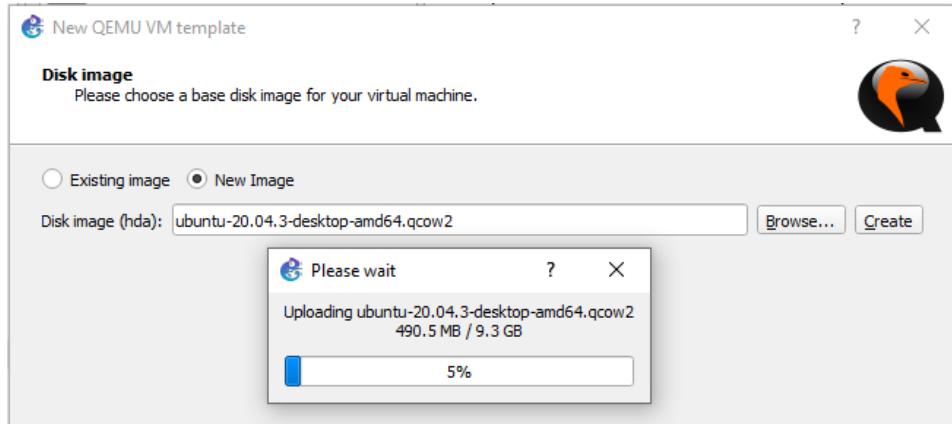
```

●      sudo bash -c 'cat> /etc/system/systemd/osdd.service' <<'EOS'
●      [Unit]
●      Description=Configure network for OSDD function
●      After=network.target
●
●      [Service]
●      Type=oneshot
●      ExecStart=/usr/sbin/ifconfig TX-veranderen up
●      ExecStart=/usr/sbin/ifconfig RX-veranderen up
●      ExecStart=/usr/sbin/ifconfig TX-veranderen -multicast
●      ExecStart=/usr/sbin/ifconfig RX-veranderen -multicast
●      ExecStart=/usr/sbin/ip -6 addr flush TX-veranderen
●      ExecStart=/usr/sbin/ip -6 addr flush RX-veranderen
●      ExecStart=/usr/sbin/ifconfig TX-veranderen promisc
●      ExecStart=/usr/bin/daemonlogger -i TX-veranderen -o
RX-veranderen -d
●      TimeoutStartSec=0
●      RemainAfterExit=yes
●
●      [Install]
●      WantedBy=multi-user.target
●      EOS
●      sudo systemctl enable osdd

```

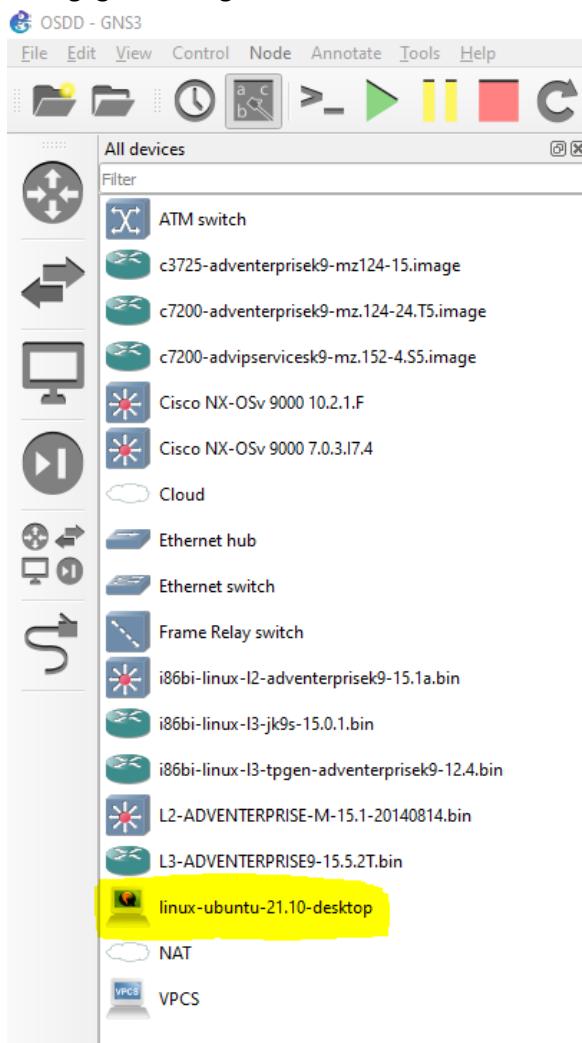
6. Maak het script executable door het volgende: 'chmod +x bestandnaam' en installeer het script door er dubbel op te klikken. Hiermee wordt het OSDD-script geïnstalleerd en dit kan gecontroleerd worden door in de terminal het volgende commando op te geven: 'sudo systemctl status/start/stop osdd'
7. Als dit allemaal gelukt is moet de VM worden afgesloten om geïmporteerd te worden in GNS3. (Indien de virtual harddisk een '.qcow2' extensie moet krijgen dient deze eerst nog te worden omgezet.)
  - Om een virtual harddisk om te zetten naar '.qcow2' moet het volgende worden gedaan:
    - Go to C:\Program Files\qemu en open PowerShell as Admin
    - .\qemu-img.exe convert -f vmdk "source path.vmdk" -O qcow2 "destination path.qcow2"
8. Ga in het preferences menu van GNS3 naar Qemu VMs en klik 'New'.
9. Selecteer 'Run this Qemu VM on the GNS3 VM'.
10. Geef een naam op voor de nieuwe Qemu VM.
11. Laat de binary staan en pas het geheugen aan naar bijvoorbeeld 1024 MB.
12. Laat telnet staan als console type.

13. Kies bij disk image voor een 'New Image', zoals te zien is in figuur 56 en zoek het te gebruiken file op en selecteer deze (de tijd van het uploaden is afhankelijk van de grootte van het bestand en kan dus even duren)



Figuur 56 New QEMU VM template new disk image

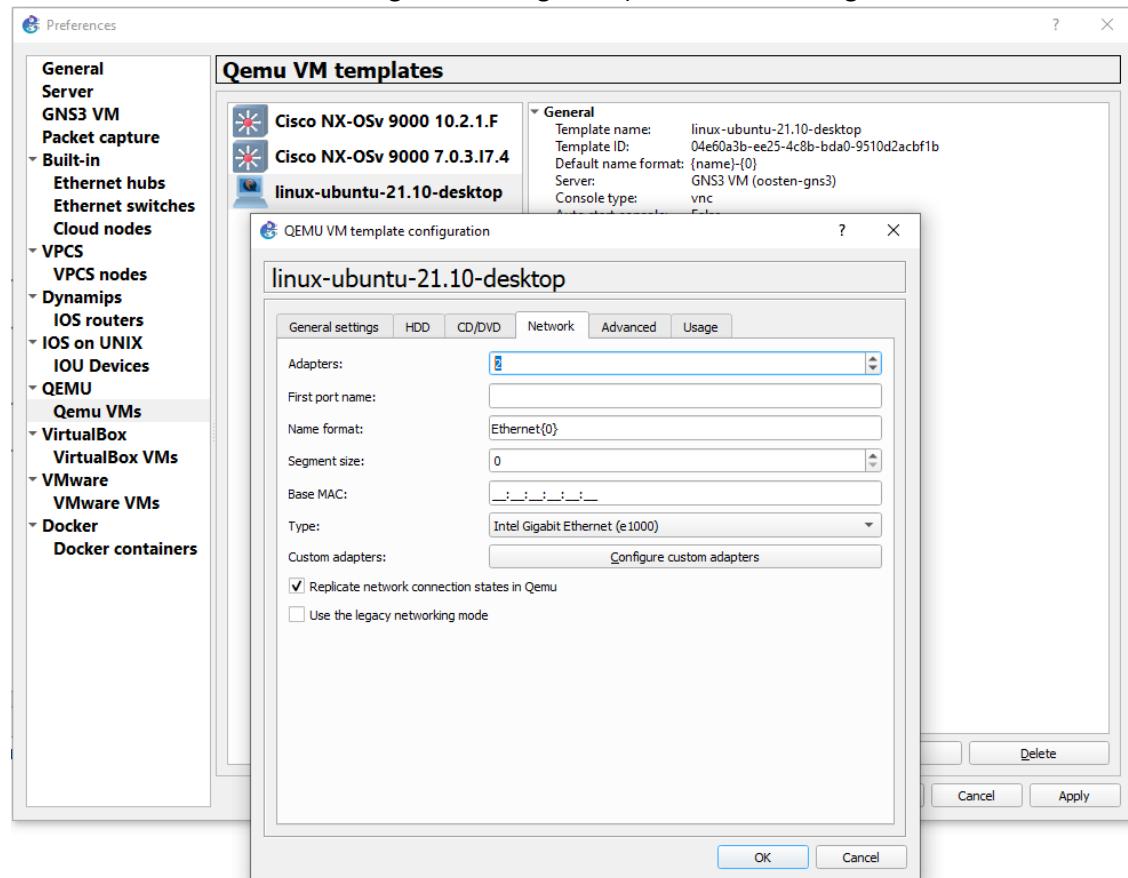
14. Als het uploaden klaar is kan de wizard voltooid worden door op finish te klikken en kan de VM worden gebruikt voor projecten in GNS3 en is te vinden bij de devices zoals weergegeven in figuur 57.



Figuur 57 All devices list in GNS3 with the new Ubuntu VM template

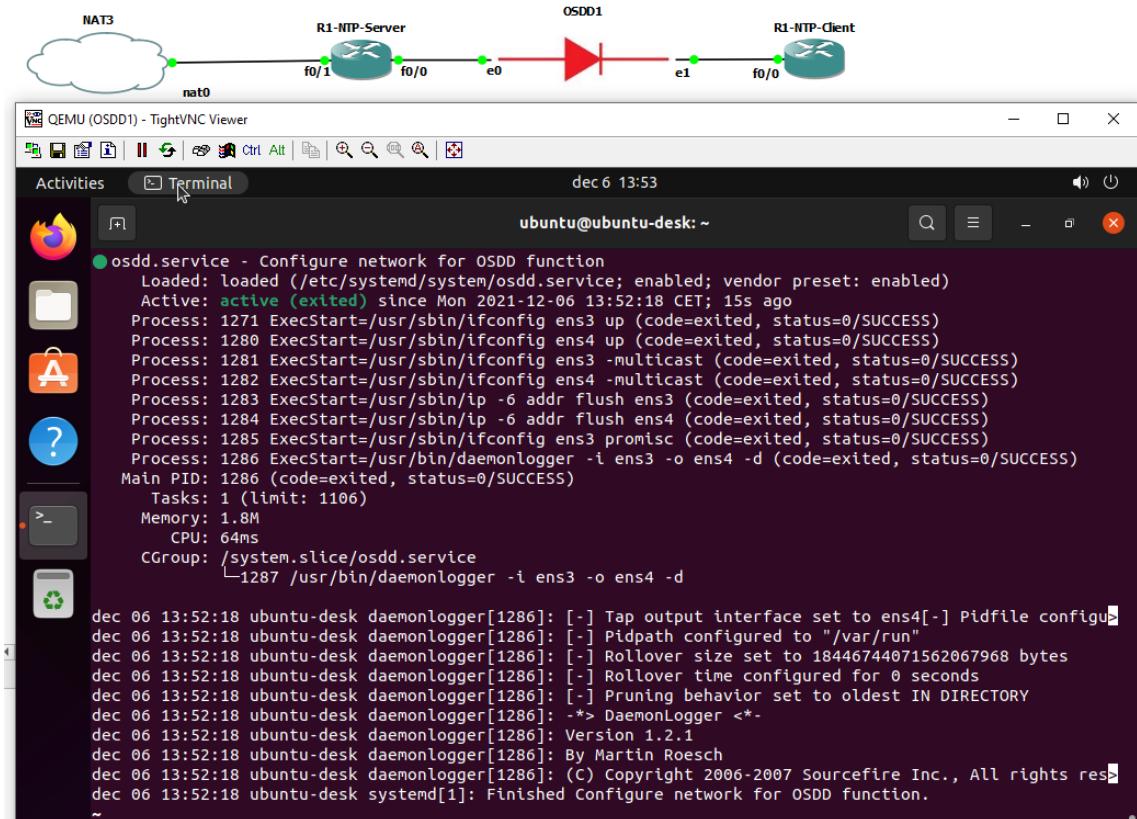
15. Het VM-template is voorzien van één netwerkadapter dus dit moet nog wel aangepast worden. Dit kan op twee manieren:

- i. Als de template is toegevoegd aan een project door erop te klikken met rechtermuis → Configure → Network → Adapters
- ii. De VM-template aanpassen, zodat elke keer als het template gebruikt wordt de machine gelijk twee netwerkadapters heeft (hier kunnen ook andere instellingen worden gedaan). Dit is te zien in figuur 58.



Figuur 58 Qemu VM template configuration change network adapters

16. Als het template is toegevoegd in een project, kan deze gestart worden. De settings voor de OSDD zijn in een eerder stadium al gedaan en daarom hoeft alleen gecontroleerd te worden of de service draait (sudo systemctl status osdd). Hoe dit er exact uitziet staat duidelijk in figuur 59.



Figuur 59 OSDD service status in GNS3 project

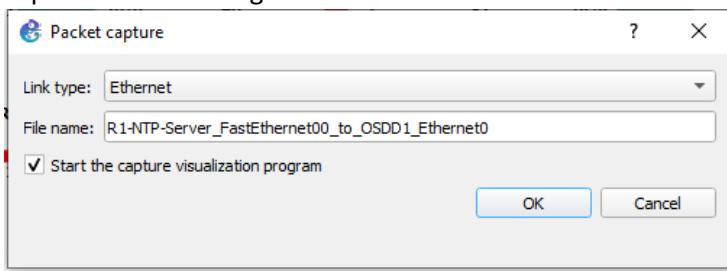
## Packet capture m.b.v. Wireshark

In GNS3 is het mogelijk om een link (netwerkconnectie tussen twee apparaten) te monitoren met behulp van het programma Wireshark. Deze software is een ‘protocol analyzer’ dat elk ‘gecaptured’ pakket weergeeft tot het diepste detail als mogelijk. Hiermee word je in staat gesteld om goede en betrouwbare analyses te doen over het netwerkverkeer. Het lijkt enigszins op de ‘tcpdump’ die Linux biedt, maar Wireshark is veel gedetailleerder in de uitvoering.

Een packet capture aanzetten in GNS3 is niet moeilijk en in een paar simpele stappen uit te voeren. Als de packet capture is opgezet wordt Wireshark automatisch gestart. Om te begrijpen wat het programma laat zien is netwerkkenis (OSI-model etc.) een vereiste.

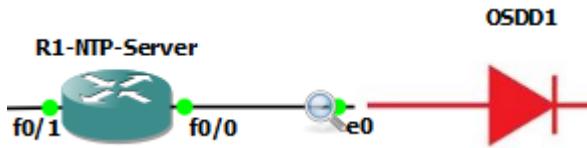
Met de volgende stappen wordt een packet capture opgezet:

1. Ga met de muis naar link die gemonitord moet gaan worden → rechter muis → Start capture en klik ok. Figuur 60 illustreert hier een voorbeeld van.



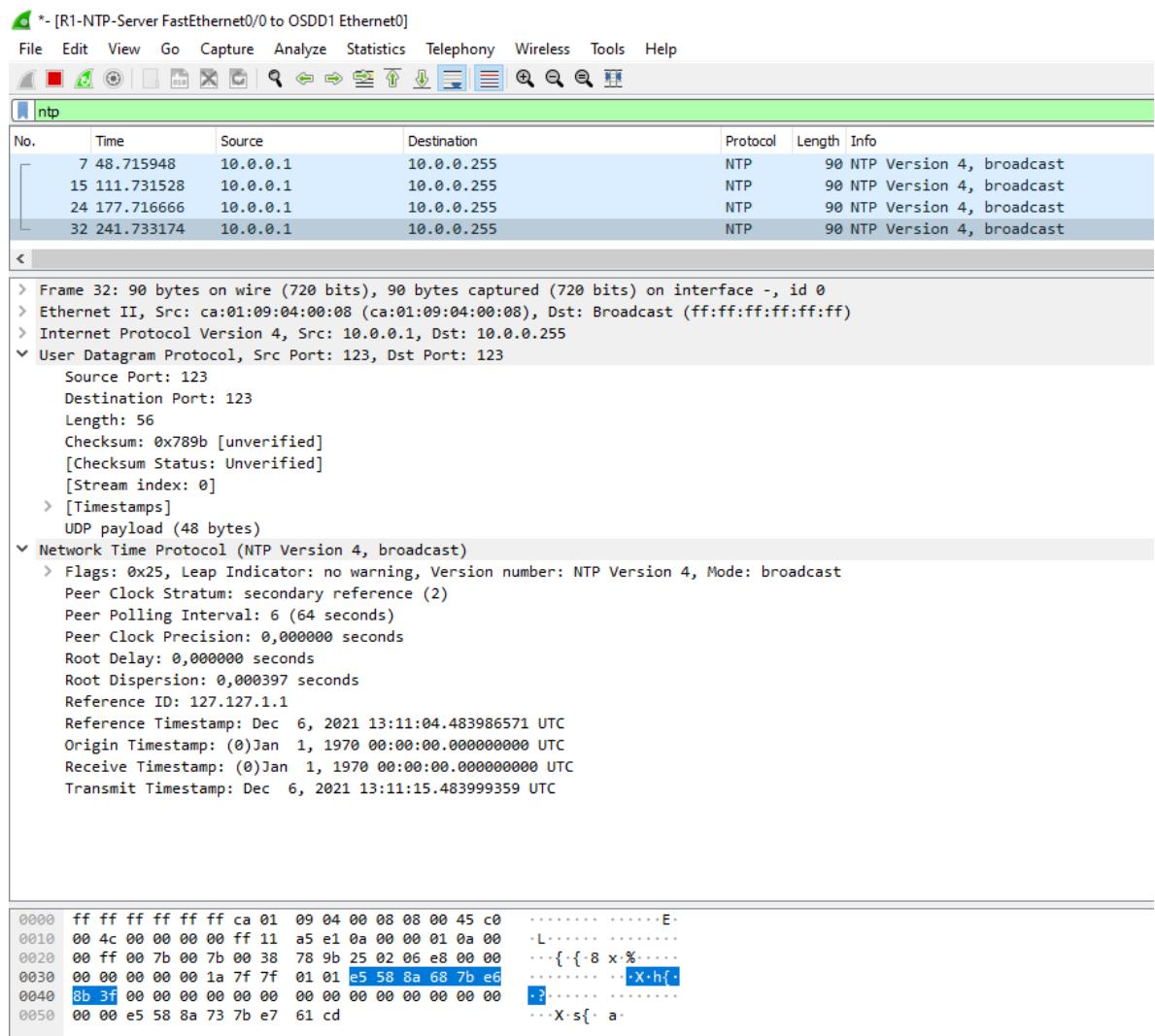
Figuur 60 Start packet capture in GNS3

2. Vervolgens wordt Wireshark gestart en is er een vergrootglas te zien in figuur 61 op de lijn die gemonitord wordt.



Figuur 61 Packet capture started on link

3. Met Wireshark kunnen nu pakketten worden geïnspecteerd (zie figuur 62). Ook kan hier gefilterd worden op bijvoorbeeld NTP-pakketten zoals hieronder weergegeven. Dit is naar wens aan te passen en daardoor kan er heel precies gezocht worden naar bepaalde pakketten.



Figuur 62 Wireshark filter for NTP packets

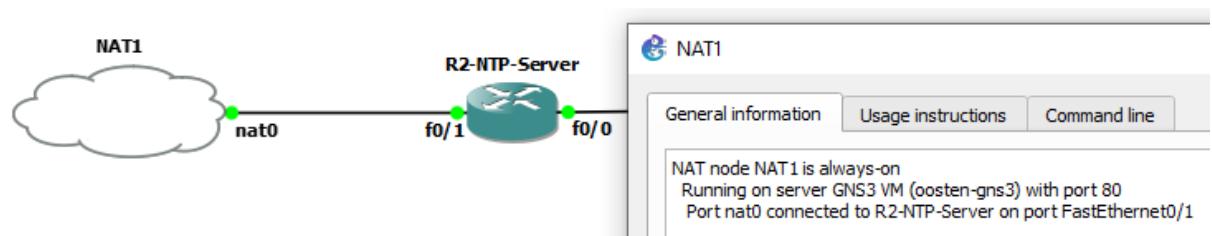
4. Met tcpdump kunnen pakketten worden geïnspecteerd op de virtuele OSDD. Dit is te zien in figuur 63.

```
ubuntu@ubuntu-desk:~$ sudo tcpdump -i ens3
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
14:24:20.542206 CDPv2, ttl: 180s, Device-ID 'R1-NTP-Server'
    0x0000: 4369 7363 6f20 494f 5320 536f 6674 7761
    0x0010: 7265 2c20 3732 3030 2053 6f66 7477 6172
    0x0020: 6520 2843 3732 3030 2d41 4456 454e 5445
    0x0030: 5250 5249 5345 4b39 2d4d 292c 2056 6572
    0x0040: 7369 6f6e 2031 322e 3428 3234 2954 352c
    0x0050: 2052 454c 4541 5345 2053 4f46 5457 4152
    0x0060: 4520 2866 6333 290a 5465 6368 6e69 6361
    0x0070: 6c20 5375 7070 6f72 743a 2068 7474 703a
    0x0080: 2f2f 7777 772e 6369 7363 6f2e 636f 6d2f
    0x0090: 7465 6368 7375 7070 6f72 740a 436f 7079
    0x00a0: 7269 6768 7420 2863 2920 3139 3836 2d32
    0x00b0: 3031 3120 6279 2043 6973 636f 2053 7973
    0x00c0: 7465 6d73 2c20 496e 632e 0a43 6f6d 7069
    0x00d0: 6c65 6420 4672 6920 3034 2d4d 6172 2d31
    0x00e0: 3120 3036 3a34 3920 6279 2070 726f 645f
    0x00f0: 7265 6c5f 7465 616d
    0x0000: 4369 7363 6f20 3732 3036 5658 52
    0x0000: 0000 0001 0101 cc00 040a 0000 01
    0x0000: 4661 7374 4574 6865 726e 6574 302f 30
    0x0000: 0000 0001
    0x0000: c0a8 4c00 18
    0x0000: 01, length 348
14:24:26.916901 Loopback, skipCount 0, Reply, receipt number 0, data (40 octets)
```

Figuur 63 Tcpdump in Ubuntu

### Internet koppeling GNS3

Het is mogelijk om apparaten binnen GNS3 te koppelen aan internet. Om dit te doen moet er een “NAT” wolk worden toegevoegd aan het project (zie figuur 64). Bij het toevoegen zijn er twee opties om te kiezen: de host pc en de GNS3 VM. Kies hier voor de GNS3 VM. De wolk kan nu gekoppeld worden aan een device dat een koppeling moet krijgen met internet. In het onderstaande voorbeeld heeft een Cisco router een koppeling met internet om zo NTP te laten synchroniseren met een atoomklok. Vanzelfsprekend is het ook mogelijk om andere apparaten te koppelen.



Figuur 64 NAT (internet) connected to Cisco router + NAT1 information

Router instellingen voor internet connectie en NTP-synchronisatie:

```
•      !
•      R2-NTP-Server# conf t
•      R2-NTP-Server(config)# service timestamps debug datetime
localtime
•      R2-NTP-Server(config)# service timestamps log datetime
localtime
•      !
•      R2-NTP-Server(config)# clock timezone CET +1
•      !
•      R2-NTP-Server(config)# ip name-server 1.1.1.1
•      R2-NTP-Server(config)# ip name-server 8.8.8.8
•      !
•      R2-NTP-Server(config)# interface FastEthernet0/1
•      R2-NTP-Server(config-if)# ip address dhcp
•      R2-NTP-Server(config-if)# no shutdown
•      R2-NTP-Server(config-if)# exit
•      !
•      R2-NTP-Server(config)# ntp update-calendar
•      R2-NTP-Server(config)# ntp server pool.ntp.org
•      R2-NTP-Server(config)# ntp server nl.pool.ntp.org
•      !
```

Het testen van de internetverbinding op de router kan getest worden zoals in figuur 65.

```
R2-NTP-Server#ping google.nl

Translating "google.nl"...domain server (1.1.1.1) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 216.58.214.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/18/24 ms
R2-NTP-Server#
```

Figuur 65 Ping test to DNS-name 'google.nl' from R2-NTP-Server

In figuur 66 is te zien welke referentie klokken gebruikt worden op de router.

```
R2-NTP-Server#sh ntp associations

  address      ref clock      st  when   poll  reach  delay  offset  disp
+~69.164.213.136  200.98.196.212  2    129    128   177 116.09  -5.403  1.740
*~213.239.154.12  193.79.237.14   2    226    128   372 20.131   0.877  2.490
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
R2-NTP-Server#sh ntp stat
R2-NTP-Server#sh ntp status
Clock is synchronized, stratum 3, reference is 213.239.154.12
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**24
reference time is E559A6C2.9F0DAC1A (10:24:18.621 CET Tue Dec 7 2021)
clock offset is 0.0008 msec, root delay is 0.02 msec
root dispersion is 0.01 msec, peer dispersion is 0.00 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.00000003 s/s
system poll interval is 128, last update was 361 sec ago.
R2-NTP-Server#
```

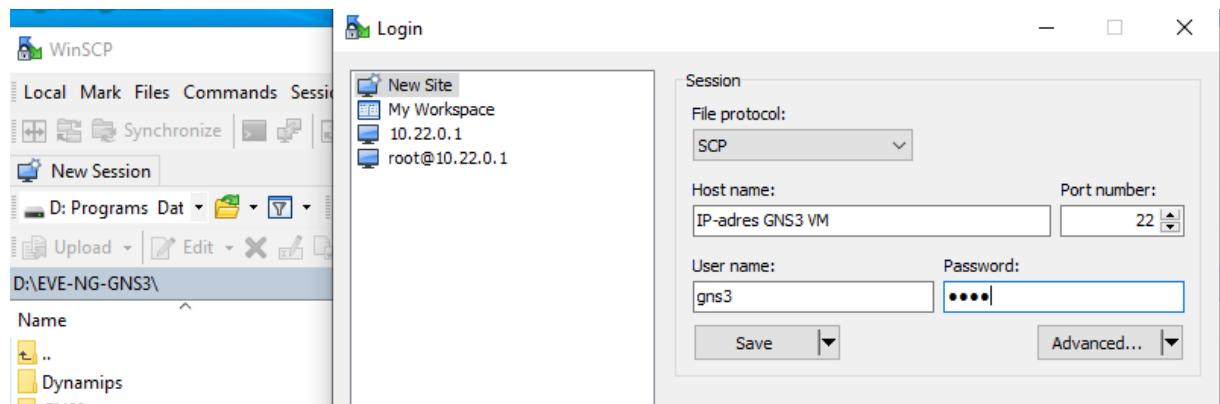
Figuur 66 R2-NTP-Server NTP associations and NTP status with the clock synchronized

## Handig GNS3 web client

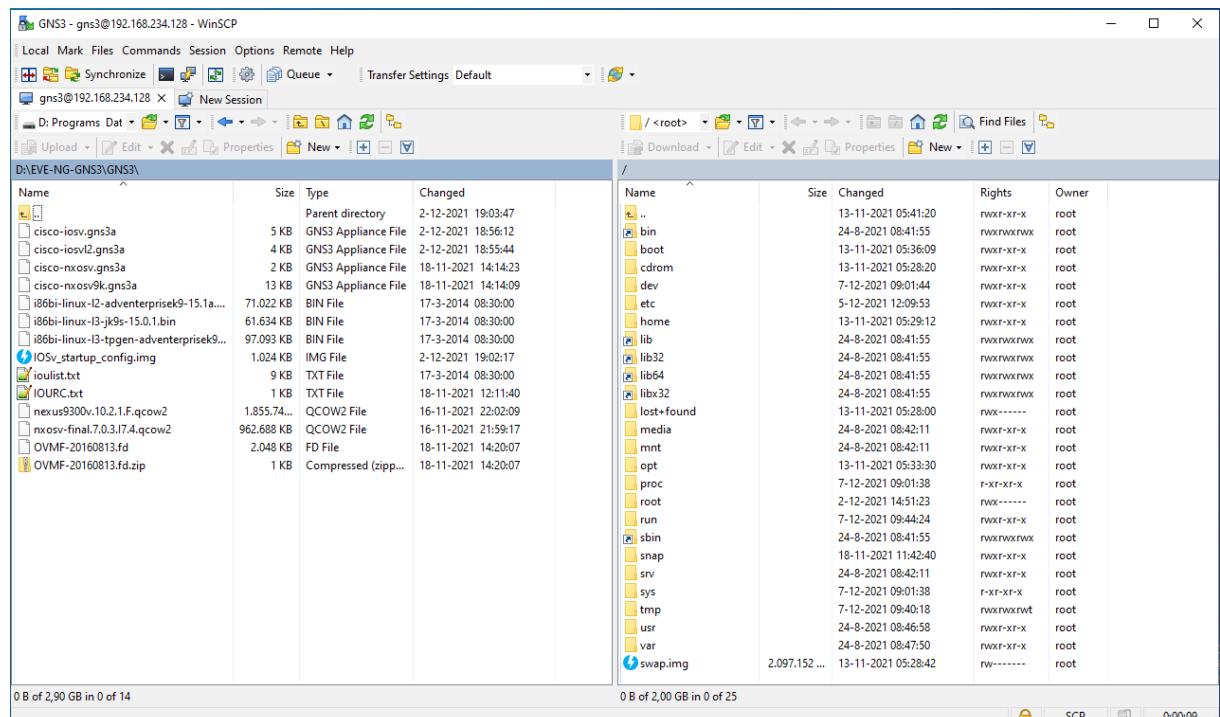
Als er geen gebruik wordt gemaakt van de software, beschikt GNS3 ook over een web client en kan simpel benaderd worden door het IP-adres van de GNS3 VM in te tikken in de browser.

## WinSCP

Als er snel bestanden moeten worden toegevoegd of worden afgehaald van de GNS3 VM dan is dit mogelijk middels het programma WinSCP (<https://winscp.net/eng/download.php>). Dit is een open source ftp-client. In figuur 67 en 68 is weergegeven hoe dit er uit ziet.



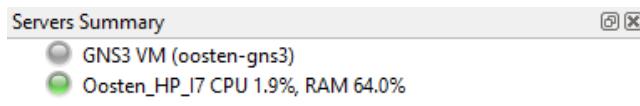
Figuur 67 WinSCP login to GNS3 VM



Figuur 68 WinSCP logged in at the GNS3 VM

### GNS3 VM 'greyed out' issue:

Het kan voorkomen dat de GNS3 VM 'greyed out' (zie figuur 69) is zodra het programma wordt opgestart en daarmee heeft het programma niet de beschikking over alle functionalisten.



Figuur 69 GNS3 VM 'greyed out'

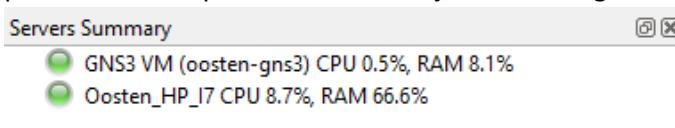
Met een paar simpele stappen is dit probleem snel op te lossen:

1. Sluit GNS3 af en wacht ook tot de GNS3 VM is afgesloten.
2. Ga naar de instellingen van de netwerkadapters en lokaliseer de VMware Virtual Ethernet (weergegeven in figuur 70) Adapters → Rechtermuis disable → Rechtermuis enable



Figuur 70 VMware Virtual Ethernet Adapters

3. Start GNS3 opnieuw op en wacht ook totdat de GNS3 VM is opgestart. Hiermee is het probleem verholpen en dit is duidelijk te zien in figuur 71.



Figuur 71 GNS3 VM 'green' and working

### Verander symbol in GNS3

Apparaten in GNS3 zijn standaard voorzien van een symbool en deze is naar wens aan te passen. Aangezien er in deze handleiding een virtuele datadiode wordt gemaakt is dat symbool, wat eigenlijk een desktop was, veranderd in het symbool van een diode. Het te gebruiken symbool moet hiervoor op de host machine staan. Vervolgens wordt er simpelweg met rechtermuis geklikt op het apparaat waarvan het symbool veranderd moet worden en wordt er gekozen voor 'Change symbol' waarna vervolgens een aangepast symbool opgezocht kan worden. In figuur 72 is te zien hoe het symbool van een standaard linux desktop is veranderd in het symbool van een diode.



Figuur 72 Symbol Linux desktop changed

### Interface labels

In GNS3 is het erg handig om de zogenaamde 'interface labels' te kunnen zien. Standaard staat dit uit maar is simpel aan te zetten door in de applicatie te gaan naar: View → Show/Hide interface labels.

## Bijlage 4: OSDD Test platform manual

### Introduction

In the modern IT world, it seems we see news articles about security breaches, data leaks and successful hacking basically every week. IT departments the world over, do their best to protect the company infrastructure from intrusions using different kinds of firewalls, proper system hardening, proper IAM, etc.

In the last few years there has been a new development called a Data Diode, as the name suggests, this device permits traffic in only one direction. The product has already been deployed by commercial parties in some locations like the government, however the costs often prove to be prohibitive to implement this technology. Commercial products incorporate a proprietary proxy solution in order to overcome the inherent restrictions one-way data traffic impose. This low adoption rate and high costs have also prevented the technology to be accounted for in commonly used network protocols, from file transfers to NTP implementations.

A number of institutions have since started to develop an open-source Data diode to provide a more affordable solution and improve product adoption, it is still in fairly early stages, but it is now possible for interested parties to build lab environments in which they can test and develop for an OSDD (Open-Source Data Diode). In this document a guide is provided to create such a lab environment and start developing for the solution. Keep in mind that a virtual data diode is inherently not as secure as a physical solution. A physical data diode is incapable of doing anything other than passing along one-way traffic whereas a virtual appliance can be compromised.

## Building test environment

The OSDD (Open Source Data Diode) can be demonstrated in a virtual environment, you can use all common hypervisors to do this;

- ESXi, often used in businesses, industry standard.
- KVM, Linux hypervisor, for example in Proxmox.
- VirtualBox, popular free hypervisor.
- VMware Workstation (Player), popular enterprise hypervisor on desktop.
- QEMU, commonly used in labbing environments like GNS3 and EVE-NG.
  - o Use the E1000 or derivative network adapter only, VIRTIO adapters will cause problems.

To build a basic lab you will need to create 3 virtual machines;

- VM1 – TX, transmitting machine. This is the sending proxy
- VM2 – OSDD, the datadiode
- VM3 – RX, the receiving machine. This is the receiving proxy

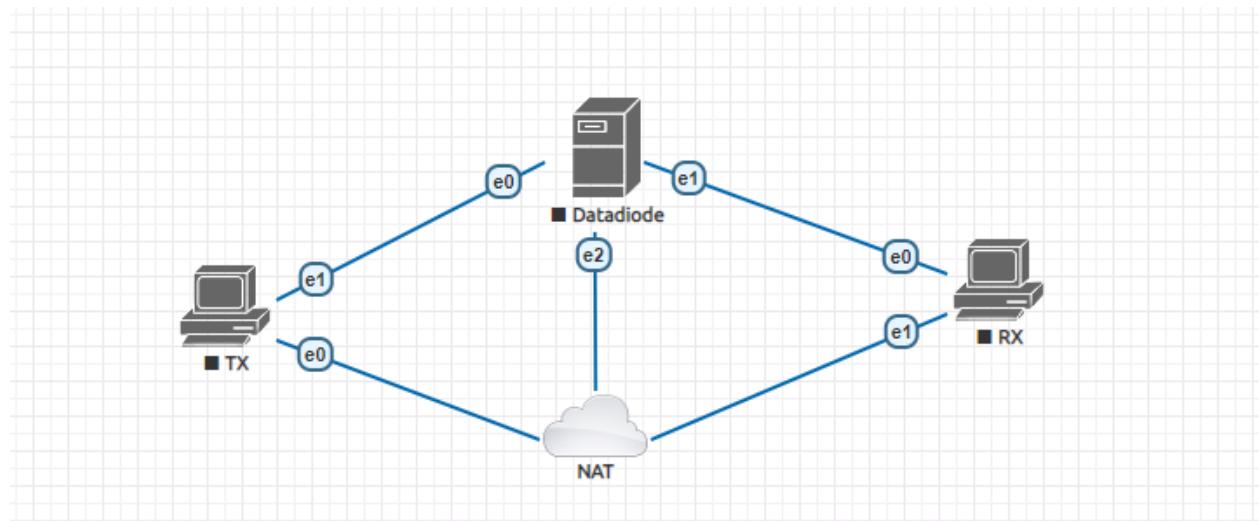
You also need to create 2 separate networks:

- SEND – This network is used between the TX machine and the OSDD.
- RECEIVE – This network is used between the OSDD and the RX machine.

To keep things simple and compatible we will be using Ubuntu Desktop on VirtualBox, the GUI is needed when tests with video streaming are executed.

## Building the requisite VMs

We are going to build the below this topology:



Figuur 73 Topologie icm datadiode

Create the below VM's:

**VM1 (TX)**

<b>Name</b>	TX
<b>OS</b>	Ubuntu 20 or newer
<b>RAM</b>	2GB
<b>CPU</b>	2 cores
<b>Storage</b>	100GB
<b>NIC1</b>	Attached to NAT
<b>NIC2</b>	Attached to 'SEND' network IP: 10.0.0.1 Subnet mask: 255.255.255.0
<b>Required packages</b>	net-tools pv udpcast

**VM2 (OSDD)**

<b>Name</b>	OSDD
<b>OS</b>	Ubuntu 20 or newer
<b>RAM</b>	2GB
<b>CPU</b>	2 cores
<b>Storage</b>	100GB
<b>NIC1</b>	Attached to 'SEND' network No IP configuration
<b>NIC2</b>	Attached to 'RECEIVE' network No IP configuration
<b>NIC3</b>	Attached to NAT
<b>Required packages</b>	net-tools daemonlogger

**VM3 (RX)**

<b>Name</b>	RX
<b>OS</b>	Ubuntu 20 or newer
<b>RAM</b>	2GB
<b>CPU</b>	2 cores
<b>Storage</b>	100GB
<b>NIC1</b>	Attached to 'RECEIVE' network IP: 10.0.0.2 Subnet mask: 255.255.255.0
<b>NIC2</b>	Attached to NAT
<b>Required packages</b>	net-tools pv udpcast

## Configuration

A datadiode inherently breaks the basic principles of networking, therefore it is needed to use a proxy to allow normal devices to send data across the diode. In the testing environment it will do to inject an entry into the ARP table of the sending machine.

On the virtual OSDD you'll need to create a service that acts as a datadiode. Essentially, all it does is copy all incoming packets from interface A over to interface B. Daemonlogger is used as a SPAN port to accomplish this.

### VM1 (TX)

In this document we are going to assume the 'SEND' network is attached to the 'enp0s8' interface of the TX machine. Below we are telling the arp table that the IP address '10.0.0.2' is mapped to the MAC address 'ff:ff:ff:ff:ff:ff', which can be found behind interface 'enp0s8'.

```
sudo arp -s 10.0.0.2 ff:ff:ff:ff:ff:ff -i enp0s8
```

### VM2 (OSDD)

The virtual datadiode is a bit more involved, we'll have to create a SPAN port as shown below.

Create a new file and paste the below config into it, make it executable using 'chmod +x' and run it, you now have a working virtual datadiode. Don't forget to check your interface labels.

In this service a couple of things happen;

- the ingress and egress interfaces are set to 'up'.
- multicast is enabled on both interfaces.
- All IPv6 config is removed (e.g. flushed).
- The ingress interface is put into promiscuous mode, this makes sure all traffic on the interface is handled directly by the CPU (punted) instead of any other (virtualized) hardware ASIC.
- Daemonlogger is configured as a deamon to act as a SPAN port.

```
sudo bash -c 'cat > /etc/systemd/system/osdd.service' << 'EOS'
[Unit]
Description=Configure network for OSDD function
After=network.target

[Service]
Type=oneshot
ExecStart=/usr/sbin/ifconfig enp0s3 up
ExecStart=/usr/sbin/ifconfig enp0s8 up
ExecStart=/usr/sbin/ifconfig enp0s3 -multicast
ExecStart=/usr/sbin/ifconfig enp0s8 -multicast
ExecStart=/usr/sbin/ip -6 addr flush enp0s8
ExecStart=/usr/sbin/ip -6 addr flush enp0s8
ExecStart=/usr/sbin/ifconfig enp0s3 promisc
ExecStart=/usr/bin/daemonlogger -i enp0s3 -o enp0s8 -d
TimeoutStartSec=0
RemainAfterExit=yes

[Install]
```

```
WantedBy=multi-user.target
EOS
sudo systemctl enable osdd
```

## VM3 (RX)

We don't need to configure anything here.

### Using the test environment

#### Transferring data using netcat

Netcat is a simple tool to send data over udp or tcp, it is very well suited to test a datadiode.

To start, we'll start nc (netcat) on the RX (listening) machine to listen to UDP datagrams on port 1234:

```
nc -u -l 1234
```

Nc will now show us any data that it receives on UDP port 1234

We can start with sending a simple message from the TX (sending) machine, first enter the below command:

```
nc -u 10.0.0.2 1234
```

You can now send any text message entered into the program, for example:

```
test
```

If you were to use real physical hardware for this scenario, you would get the same results.

Of course, you can also demonstrate more useful applications. One of the use-cases of a datadiode is to create a one-way syslog connection, devices can log their events to the syslog server but once the data has been sent, it cannot be exfiltrated.

You can send the result of a follow tail using nc, this means we can use the below line to automatically send any syslog messages from the TX machine to the RX machine:

```
tail -f /var/log/syslog | nc -u 10.0.0.2 1234
```

Instead of the 'Test' message from earlier, you'll now see all syslog messages generated by the RX machine, for example opening the Firefox browser.

#### TX sending syslog data

```
ubuntu@tx:~$ tail -f /var/log/syslog | nc -u 10.0.0.2 1234
```

Figuur 74 TX sending syslog data

## OSDD sending along TX' syslog data

```
ubuntu@osdd:~$ systemctl status osdd
● osdd.service - Configure network for OSDD function
  Loaded: loaded (/etc/systemd/system/osdd.service; enabled; vendor preset: enabled)
  Active: active (exited) since Sun 2021-12-19 13:17:04 CET; 5min ago
    Main PID: 604 (code=exited, status=0/SUCCESS)
      Tasks: 1 (limit: 2313)
     Memory: 1.8M
        CPU: 86ms
       CGroup: /system.slice/osdd.service
           └─607 /usr/bin/daemonlogger -i ens3 -o ens4 -d

dec 19 13:17:04 osdd daemonlogger[604]: [-] Pidpath configured to "/var/run"
dec 19 13:17:04 osdd daemonlogger[604]: [-] Rollover size set to 18446744071562067968 bytes
dec 19 13:17:04 osdd daemonlogger[604]: [-] Rollover time configured for 0 seconds
dec 19 13:17:04 osdd daemonlogger[604]: [-] Pruning behavior set to oldest IN DIRECTORY
dec 19 13:17:04 osdd daemonlogger[604]: -> DaemonLogger <-
dec 19 13:17:04 osdd daemonlogger[604]: Version 1.2.1
dec 19 13:17:04 osdd daemonlogger[604]: By Martin Roesch
dec 19 13:17:04 osdd daemonlogger[604]: (C) Copyright 2006-2007 Sourcefire Inc., All rights reserved
dec 19 13:17:04 osdd systemd[1]: Finished Configure network for OSDD function.
dec 19 13:17:04 osdd daemonlogger[607]: start sniffing() device ens3 network lookup: ens3: no IPv4 address assigned
```

Figuur 75 OSDD sending along TX' syslog data

## RX receiving TX' syslog data

```
ubuntu@rx: $ nc -u -l 1234
Dec 19 13:21:10 rx systemd[1]: apt-daily-upgrade.service: Deactivated successfully.
Dec 19 13:21:10 rx systemd[1]: Finished Daily apt upgrade and clean activities.
Dec 19 13:21:10 rx systemd[1]: apt-daily-upgrade.service: Consumed 24.152s CPU time.
Dec 19 13:22:03 rx anacron[478]: Job `cron.daily' started
Dec 19 13:22:03 rx anacron[5833]: Updated timestamp for job `cron.daily' to 2021-12-19
Dec 19 13:22:03 rx systemd[1]: Starting Download data for packages that failed at package install time...
Dec 19 13:22:03 rx systemd[1]: update-notifier-download.service: Deactivated successfully.
Dec 19 13:22:03 rx systemd[1]: Finished Download data for packages that failed at package install time.
Dec 19 13:22:03 rx cracklib: no dictionary update necessary.
Dec 19 13:22:03 rx anacron[478]: Job `cron.daily' terminated
Dec 19 13:24:08 rx systemd[660]: Started Application launched by gnome-shell.
Dec 19 13:24:08 rx systemd[660]: Started snap.firefox.firefox.eb67a85d-3506-49a7-b7b3-4a49b3338719.scope.
Dec 19 13:24:08 rx systemd[660]: app-gnome-firefox-firefox-5935.scope: Deactivated successfully.
Dec 19 13:24:08 rx kernel: [ 431.907267] audit: type=1400 audit(1639916648.957:59): apparmor="DENIED" operation="capable" profile="/usr/lib/snapd/snap-confine" pid=5933 comm="snap-confine" capability=39 capname="bpf"
Dec 19 13:24:08 rx kernel: [ 431.907820] audit: type=1400 audit(1639916648.957:60): apparmor="DENIED" operation="capable" profile="/usr/lib/snapd/snap-confine" pid=5935 comm="snap-confine" capability=38 capname="perfmon"
Dec 19 13:24:10 rx kernel: [ 433.605818] audit: type=1326 audit(1639916650.653:61): uid=1000 gid=1000 ses=2 subj=snap.firefox.firefox pid=5935 comm="GeckoMain" exe="/snap/firefox/767/usr/lib/firefox/firefox" sig=0 arch=c000003e syscall=314 compat=0 ip=0x7ff8ff853089d code=0x50000
Dec 19 13:24:23 rx kernel: [ 446.663443] audit: type=1326 audit(1639916663.713:62): uid=1000 gid=1000 ses=2 subj=snap.firefox.firefox pid=6181 comm="GeckoMain" exe="/snap/firefox/767/usr/lib/firefox/firefox" sig=0 arch=c000003e syscall=314 compat=0 ip=0x7f6da091289d code=0x50000
Dec 19 13:24:24 rx rtkit-daemon[715]: Supervising 6 threads of 3 processes of 1 users.
Dec 19 13:24:24 rx rtkit-daemon[715]: Supervising 6 threads of 3 processes of 1 users.
Dec 19 13:24:24 rx rtkit-daemon[715]: [ session uid=1000 pid=713] Activating service name='io.snapcraft.Settings' requested by ':1.108' (uid=1000 pid=6207 comm="dbus-send -print-reply=literal --session --dest=i" label="snap.firefox.firefox (enforce)")
Dec 19 13:24:24 rx rtkit-daemon[715]: [ session uid=1000 pid=713] Successfully activated service 'io.snapcraft.Settings'
Dec 19 13:24:24 rx io.snapcraft.Settings[6210]: userd.go:93: Starting snap userd
Dec 19 13:24:24 rx rtkit-daemon[715]: Supervising 6 threads of 3 processes of 1 users.
Dec 19 13:24:24 rx rtkit-daemon[715]: message repeated 3 times: [ Supervising 6 threads of 3 processes of 1 users.]
Dec 19 13:24:24 rx rtkit-daemon[715]: Successfully made thread 632c of process 5935 owned by '1000' RT at priority 10.
Dec 19 13:24:24 rx rtkit-daemon[715]: Supervising 7 threads of 4 processes of 1 users.
Dec 19 13:24:24 rx kernel: [ 447.693480] audit: type=1326 audit(1639916664.741:63): uid=1000 gid=1000 ses=2 subj=snap.firefox.firefox pid=6341 comm="GeckoMain" exe="/snap/firefox/767/usr/lib/firefox/firefox" sig=0 arch=c000003e syscall=314 compat=0 ip=0x7ff34402b89d code=0x50000
Dec 19 13:24:24 rx kernel: [ 450.383477] audit: type=1326 audit(1639916667.433:64): uid=1000 gid=1000 ses=2 subj=snap.firefox.firefox pid=6708 comm="GeckoMain" exe="/snap/firefox/767/usr/lib/firefox/firefox" sig=0 arch=c000003e syscall=314 compat=0 ip=0x7f5767c7689d code=0x50000
```

Figuur 76 RX receiving TX' syslog data

## Transferring data using UDPcast

A different way to send data is the use of UDPcast, this method is able to operate on higher data rates and does not need an ARP injection on the sending machine.

If you don't have a large file to send you can create one with random data using the below command:

```
head -c 1024M /dev/urandom > 1gb-testfile.tmp
```

Now we'll set the receiving machine (RX) to write any receiving data to disk, make sure to set the receiving interface appropriate to your situation:

```
udp-receiver -nosync -interface [receiving interface] -file 1gb-testfile.tmp
```

Finally, we can send the actual file to the RX machine from the TX machine, forward error correction is used to increase reliability:

```
udp-sender -interface [sending interface] -async -fec 8x8/128 -max-bitrate 300Mbps -file 1gb-testfile.tmp -broadcast
```