Professionele Bachelor Toegepaste Informatica

Automatische deploy van full mesh VPN-toepassing met Mikrotik routers

Pieter Duchateau

Promotoren:

Andy Geraerts
Cegeka

Gert Van Waeyenberg
Hogeschool PXL Hasselt

Bachelorpaper Academiejaar 2017-2018

Professionele Bachelor Toegepaste Informatica

Automatische deploy van full mesh VPN-toepassing met Mikrotik routers

Pieter Duchateau

Promotoren:

Andy Geraerts — Cegeka
Gert Van Waeyenberg — Hogeschool PXL Hasselt

Bachelorpaper Academiejaar 2017-2018

Dankwoord

Na mijn 3-jarige opleiding in de informatica is het einde in zicht. Bij aanvang van het laatste jaar kregen we de opdracht om een stagebedrijf te zoeken, waarbij ik direct aan Cegeka dacht. Na een stressvol sollicitatiegesprek had ik er toch een goed gevoel bij. Een aantal dagen later kwam het beslissende telefoontje met de bevestiging dat ik mijn stage bij Cegeka mocht starten.

Na een periode van 3 maanden stage is het zover. Ik kan nu mijn eindwerk beëindigen met het schrijven van mijn dankwoord. Het was een zeer leerrijke periode, niet alleen op technisch vlak heb ik veel bijgeleerd, ook als persoon. Daarom wil ik al de personen die dit mogelijk hebben gemaakt uitdrukkelijk bedanken.

Als eerste wil ik graag mijn collega's in het netwerkteam bedanken. Dankzij jullie hulp en steun heb ik deze stageopdracht tot een goed einde kunnen brengen. Jullie stonden altijd klaar om te luisteren naar mijn fouten, mij bij te sturen wanneer nodig en mij te steunen wanneer nodig.

Vervolgens wil ik ook mijn externe promotor Andy Geraerts bedanken. Andy heeft me altijd goed opgevolgd en ondersteund wanneer nodig. Ook al liep het begin van de stage niet op het tempo dat je hoopte, toch zijn we er tijdig geraakt. Dit heb ik allemaal te danken aan de positieve moed die je me in praatte en me op de vingers tikte wanneer ik de verkeerde kant opging.

Daarnaast zou ik ook graag mijn stagebegeleider van PXL bedanken, Gert Van Waeyenberg. Ik weet dat je veel studenten onder je hoede hebt en toch hebben we altijd binnen de dag een antwoord gekregen indien er een vraag was. Je stond altijd klaar om te helpen, ook alle feedback was zeer waardevol. Je hebt je rol als stagebegeleider meer dan goed gedaan.

Ook wil ik mijn medestudenten bedanken voor hun hulp. Ze stonden klaar voor een telefoontje van mij te beantwoorden wanneer ik veel te lang op een configuratie zat te zoeken. Ook mijn medestudenten op Cegeka wil ik hierbij bedanken voor hun meningen wanneer ik niet zeker was wat er in de documentatie vermeld moest worden.

Last but not least, mijn ouders. Altijd stonden ze voor me klaar, ze gaven we wijze raad en fungeerden als een luisterend oor.

Bedankt allemaal!

Abstract

VPN-tunnels worden vaak gebruikt in een bedrijf zoals Cegeka. Vooral wanneer klanten hun omgevingen met elkaar willen verbinden of wanneer ze monitoring willen doen op hun netwerk. Hierdoor is een Mikrotik de juiste oplossing om een site-to-site verbinden te maken. Mikrotiks staan bekend voor hun uitgebreid aanbod aan functionaliteiten tegen een relatief lage prijs.

Het eerste deel van mijn stage bestaat uit het kennis maken met Mikrotiks en hun bijhorende software, RouterOS. Mikrotiks zijn producten die nog niet behandeld waren tijdens mijn opleiding op de PXL waardoor dit product volledig nieuw was voor mij. Eenmaal kennis gemaakt met RouterOS werd er van mij verwacht een labo opstelling te maken met 2 sites waarbij ik de opgedane kennis kom omzetten in de praktijk. Door deze labo opstelling werd ik vertrouwd met RouterOS.

Nadat er kennis is gemaakt met deze software en al de functionaliteiten fatsoenlijk doorgrond zijn kunnen we de VPN-toepassingen van de Mikrotik router bekijken. Dit is het startpunt van mijn onderzoeksopdracht. Hierbij onderzoek ik al de VPN-mogelijkheden die geconfigureerd kunnen worden op een Mikrotik. Dit onderzoek gebeurt aan de hand van gerenommeerde bronnen. Daarbuiten worden ook deze verschillende VPN-mogelijkheden opgesteld in mijn labo. De 2 sites die vooraf geconfigureerd waren worden nu verbonden door verschillende VPN-tunnels. Bij elke opstelling wordt er gebruik gemaakt van Wireshark zodat we elk pakket door de VPN-tunnels uitgebreid kunnen onderzoeken. Door gebruik te maken van WireShark hebben we de mogelijkheid de encryptie van de pakketten te bekijken.

Nadat er een conclusie is getrokken welke VPN-methode het best past in deze opdracht wat betreft functionaliteiten, snelheid en security, kan deze opgesteld worden in het labo. Vanaf hier kan dan onderzocht worden hoe we deze VPN-configuratie het best kunnen automatiseren aan de hand van programmeren.

Door de configuratie voor het grootste deel te automatiseren kan er een VPN-oplossing ontvouwd worden op een Mikrotik met zo min mogelijke tussenkomst van een persoon. Dit gaan we doen aan de hand van een PHP-platform genaamd Symfony. Hierin gaat het TR-069-protocol geprogrammeerd worden zodat we on-site toestellen kunnnen beheren vanop afstand. Het TR-069-protocol geeft ons de mogelijkheid het apparaat te rebooten, bestanden naar het apparaat te pushen, parameters te veranderen en veel meer. Eenmaal de Mikrotik een verbinding heeft gemaakt met het dashboard kan er automatisch configuratie worden gegenereerd om een full-mesh tunnel op te zetten die dan naar het apparaat gepusht kan worden.

Inhoudsopgave

Contents

Lijst van gebruikte figuren

Lijst van gebruikte tabellen

Lijst van gebruikte afkortingen

VPN	Virtual Private Network
RouterOS	Router Operating System
PHP	Hypertext Preprocessor
TR-069	Technical Report 069
IT	Information Technology
POC	Proof Of Concept
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
HTTP	HyperText Transport Protocol
HTTPS	HyperText Transport Protocol Secure
ICT	Information and Communication Technology
CMS	Content Management System
ERP	Enterprice Resource Planning
WAN	Wide Area Network
ACS	Auto Configuration Server
PC	Personal Computer
RB	RouterBoard
CCR	Cloud Core Routers
DHCP	Dynamic Host Configuration Protocol
IP	Internet Protocol
CLI	Command Line Interface
GUI	Graphical User Interface
LAN	Local Area Network
MAC-address	Media Access Control address
CPU	Central Processing Unit
SSH	Secure Shell
OSI-model	Open System Interconnection model
CPE	Customer Premises Equipment
CWMP	CPE Wan Management Protocol
SOAP	Simple Object Access Protocol
XML	Extensible Markup Language
RPC	Remote Procedure Call
PBR	Policy Based Routing
VRF	Virtual Routing and Forwarding
VLAN	Virtual Lan
OSPF	Open Shortest Path First
BGP	Border Gateway Protocol
MPLS	Multi Protocol Label Switching
IPIP	IP-in-IP pakket
GRE	Generic Routing Encapsilation
EoIP	Ethernet over IP
AH	Authentication Header
ESP	Encapsulating Security Payload
ICMP	Internet Control Message Protocol
DES	Data Encryption Standard

AES	Advanced Encryption Standard
OVPN	Open VPN
SSL	Secure Sockets Layer
PPTP	Point To Point Tunneling Protocol
SSTP	Secure Socket Tunneling Protocol
PPP	Point to Point Protocol
TLS	Transport Layer Security
PFS	Perfect Forward Secrecy
DHE	Diffie Hellman Ephemeral
L2TP	Layer 2 Tunnel Protocol
LAC	L2TP Access Concentrator
NUC	Next Unit Computing

Inleiding

Deze stageopdracht draait rond het automatiseren van een VPN-tunnel aan de hand van Mikrotik apparatuur. Hierbij gaan we gebruik maken van verschillende technieken om zo tot een volledig resultaat te komen.

Vooraleer ik kan beginnen aan het automatiseren van een VPN-tunnel moet ik vertrouwd zijn met Mikrotiks. Dit doe ik door labo opstellingen te maken met verschillende Mikrotiks. Zo leer ik de werking en leer ik de functionaliteiten van RouterOS.

Na de functionaliteiten te hebben onderzocht kan mijn onderzoeksopdracht beginnen. Hierbij gaan we verschillende VPN-tunnels opbouwen om de bestaande labo-opstellingen met elkaar te verbinden. Wanneer we een tunnel hebben opgezet gaan we deze onderzoeken. Dit onderzoek bestaat uit het meten van de bandbreedte, onderzoeken van de functionaliteiten en het bekijken van de encryptiemogelijkheden.

De bandbreedte meten we aan de hand van een ingebouwde bandbreedte tester binnenin RouterOS. Bij deze bandbreedte tester hebben we de keuze tussen TCP of UDP en de pakketgrootte of het aantal connecties. Na het meten van de tunnel bekomen we een gemiddelde. Deze waarde gebruiken we om een vergelijking te maken.

De functionaliteiten en de encryptiemogelijkheden worden onderzocht via verschillende gerenommeerde bronnen. De encryptie kunnen we ook bekijken bij het opstellen van een VPN-tunnel en bij het gebruik van Wireshark.

Uiteindelijk wanneer we de onderzoeksopdracht hebben afgerond kunnen we de automatisatie van het script beginnen. Voor het opstellen van deze VPN-tunnels is er informatie nodig. Deze informatie gaan we aan de hand van het TR-069-protocol bemachtigen. Het TR-069-protocol doet op een gekozen interval http(s)-requests waar we data uit kunnen halen. Eenmaal parameters opgevraagd van het apapraat kan er een config file gegenereerd worden wat gepusht wordt naar het device en van daaruit een VPN-tunnel opzet.

I. Stageverslag

1 Bedrijfsvoorstelling

1.1 Cegeka groep NV

Cegeka is een snelgroeiend ICT-bedrijf waarvan de hoofdzetel gelegen is in Hasselt. Het doel van Cegeka is binnen de top 5 van de Benelux te behoren. Als ICT-dienstverlener en solution integrator helpt Cegeka andere bedrijven met de gehele ICT-waardeketen: implementatie en integratie van infrastructuuroplossingen, ontwikkeling en implementatie van software. Ook verzorgt Cegeka het volledige beheer van de ICT-omgeving van andere bedrijven.

De missie van Cegeka is andere ondernemingen helpen om sterk te staan en te groeien in een digitale wereld. Dit wordt mogelijk gemaakt door de geavanceerde IT-oplossingen, een strategische manier van denken en een praktische aanpak.

In 2015 heeft Cegeka een omzet van 369 miljoen euro geboekt. Dit is 31% meer dan in het voorgaande jaar. In 2016 was dit zelf 414 miljoen euro, dit is dus nog eens stijging met 12.4%. Dit is een mooi voorbeeld van de grote groei die Cegeka realiseert. Cegeka is een bedrijf met meer dan 4000 werknemers wereldwijd. Het doet het internationaal ook zeer goed, zo is het actief in 13 Europese landen. Cegeka heeft meer dan 2500 klanten.

Cegeka heeft ook eigen “tier drie” datacenters. Dit wil zeggen dat de beschikbaarheid 99.982% bedraagt.

De aangeboden diensten kunnen worden opgedeeld in vijf categorieën:

- IT-Infrastructure
- Business Solutions
- Markets
- Agile solutions
- IT-consultancy

De stageopdracht die wordt uitgewerkt vindt plaats binnen het onderdeel IT-Infrastructure. Hieronder vallen Cloud Solutions, Managed Services, Outsourcing en Infrastructuurprojecten.

1.2 IT cultuur

Cegeka spitst zich niet specifiek toe op een bepaalde technologie, het past zich aan naar de noden van de klant. Zo hebben ze teams en specialisten voor veel verschillende takken binnen de IT. Ze zijn ook continu op zoek naar nieuwe technologieën om nog beter diensten te kunnen verlenen aan de klanten.

Binnen de tak infrastructuur wordt er ook een zeer uitgebreid gamma aan producten aangeboden. Zo bieden ze allerhande Cloud technologieën aan, Database, IIS, Linux, Lotus Notes Domino, Middleware, MS BizTalk, MS Exchange Server, Networking, Novell, Security, Storage, unix, Version Control System en Virtualisatie.

Binnen de tak Development worden volgende technologiën gebruikt: Mobile, Web, .NET, IDE, Java, JavaScript, Lotus Notes, Mainframe, Microsoft, MS BizTalk, MS SharePoint, PHP, Powerbuilder, Ruby, Scripting, Windev, XML, AS400, C en Delphi.

Cegeka biedt ook uitgebreide Business Solutions. De diensten die Cegeka hiervoor aanbiedt zijn uitgebreide CMS-systemen op maat gemaakt voor de klant. De gebruikte technologieën hier zijn Sitecore, Alfresco, Drupal, Liferay en WordPress. Een andere tak binnen de Business Solutions zijn ERP/CRM-systemen. De gebruikte technologieën hier zijn MS Dynamics Axapta, MS CRM, MS NAV en SAP. De derde aangeboden tak binnen de Business Solutions is Business Intelligence. Hieronder vinden we Business Objects, Cognos, Microsoft BI, Oracle BI, Qlikview en SAS terug. Waar Cegeka zich ook mee bezighoudt is Digital Marketing, ze weten dat het vandaag de dag niet meer volstaat om alleen maar een goed product te leveren om een grote impact te hebben op klanten.

Op vlak van applicatieontwikkeling kiest Cegeka resoluut voor de agile manier van ontwikkelen. Dit laat zich merken door coaching en trainingssessies die Cegeka geeft. Agile ontwikkeling is de manier van ontwikkelen waar een groot project zal opgedeeld worden in kleinere projecten. Het fundeert op een zeer goede samenwerking tussen de ontwikkelaars.

Outsourcing is een zeer grote tak binnen Cegeka. Bij Outsourcing gaat een bedrijf een deel of het geheel van zijn IT overhevelen aan een externa partij, in dit geval dus Cegeka. Enkele voorbeelden van bedrijven die hun IT uitbesteed hebben aan Cegeka zijn Aviapartner, Delta Lloyd, Vandemoortele, Van Gansewinkel, Argenta, Sibelco, …

1.3 Partners

Om nog beter oplossingen te kunnen voorzien voor de klant gaat Cegeka ook Partnerships aan met andere bedrijven.

CISCO – De relatie die CISCO en Cegeka hebben is speciaal op veel verschillende manieren. Zo kunnen Cegeka en CISCO samen hun expertise gebruiken om gezamelijke klanten nog beter te helpen.

IBM – In 2015 hebben Cegeka en IBM samen veel bereikt. Zo hebben ze zeer grote internationale projecten samen tot een goed einde gebracht. Ook heeft IBM alle infrastructuur geleverd voor het datacenter in Nederland.

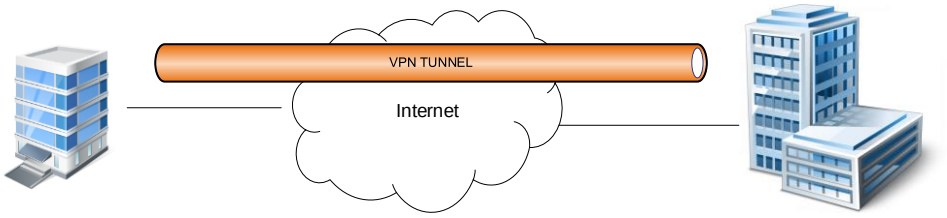
HP – HP is ervan overtuigd dat de globale economie aan het veranderen is naar een “ideeën economie”. Simpele ideeën in combinatie met een agile omgeving kunnen snel veranderen in ideeën die een grote verandering kunnen teweegbrengen. HP gelooft erin dat Cegeka de partner is die dit gedachtengoed ook volgt.

1.4 Motivering

Als laatstejaars student kregen we een ruime keuze aan bedrijven om bij te solliciteren. De keuze voor Cegeka was voor mij snel gemaakt. Een ambitieus bedrijf dat streeft naar constante groei. Dit is zonder enige twijfel de plaats waar ik mijn professionele carrière wil uitbouwen. De ambitie van dit bedrijf is een motivatie om nog harder te werken om het doel te bereiken. Innovatie is ook een van de sterke punten van Cegeka, dit reflecteert zich in de stageopdracht die ik heb mogen uitwerken. Altijd nieuwe manieren zoeken om een bepaald doel te bereiken om de klant nog beter te kunnen dienen.

2 Voorstelling stageopdracht

2.1 Probleemstelling



1 VPN tunnel

Binnenin Cegeka wordt er dagelijks gebruik gemaakt van VPN-tunnels. Deze worden gebruikt voor het monitoren van de verschillende sites, het verbinden van verschillende sites van de klant, en het beheren van de on-site apparatuur. VPN-tunnels zijn goedkope manieren om verbindingen te maken over een WAN met behoudt van vertrouwen doordat deze tunnels geëncrypteerd kunnen worden. Het internet zit vol met hackers of mensen die kunnen meekijken naar de data die verstuurd worden en dat is wat we niet willen. Wanneer we een VPN-tunnel naar een site willen maken moeten we aan beide kanten van de tunnel configuratie toepassen voor een tunnel op te bouwen. Hiervoor moet momenteel elke keer er een tunnel wordt opgezet. Dit kost tijd en geld, wat we willen vermeiden met deze opdracht.

De verschillende probleempunten :

- Implementatietijd
    - Binnen Cegeka wordt er 8 uur gerekend voor op on-site opzetten van een VPN waarbij per klant er meerdere sites kunnen voorkomen. Als we een klant hebben met 20 sites , zitten we al aan 160 uren. Nu dit is alleen nog maar van verschillende sites naar 1 verdeelcentrum. We willen als Cegeka zijnde een robuust netwerk hebben waarbij we zeker zijn dat wanneer 1 verbinding wegvalt we nog altijd verder kunnen gaan. Daarom wordt er gewerkt met Full-Mesh. Dit wil eigenlijk zeggen dat we van iedere site nog eens tunnels gaan bouwen onderling. Dus elke site gaat een tunnel opbouwen naar alle bestaande sites. Eerst zaten we aan 160 uur, maar met Full-Mesh gaan we een stapje hoger gaan en daarvoor is er een formule : x(x-1)/2 = n.Wanneer we de 20 sites op deze formule zouden toepassen bekomen we 20(20-1)/2=378 uur = 47 dagen. Dit zijn een aantal dagen voor 1 klant te voorzien van VPN-tunnels.
- Doorlooptijd
    - Als er een klant VPN-tunnels wilt aanvragen gebeurt dit via een formulier. Dit formulier wordt eerst naar de klant opgestuurd waarbij er informatie moet worden ingevuld worden over de aanvraag. Meestal zijn deze apparaten niet eens van de klant zelf maar van externe firma's. Hierdoor moet de klant nog de informatie aan de externa partij vragen en wordt hier toch wel wat tijd verloren. Eenmaal deze info ingevuld is , wordt dit formulier terug naar Cegeka gestuurd en vult Cegeka hun informatie in. En dan wordt dit zelfs nogmaals terug gestuurdd naar de klant waardoor we veel tijd verliezen.

- Human Error
    - Momenteel wordt de configuratie volledig manueel gedaan waardoor er fouten kunnen optreden. Wanneer er een passwoord verkeerd wordt ingevuld , of er worden punten of comma's ergens fout geplaatst kan het zijn dat deze tunnels niet opkomen. Als een tunnel niet opkomt moet er gezocht worden naar de fouten en verliezen we tijd.

2.2 Doelstelling

De doelstelling van deze opdracht is om automatisch een VPN-tunnel op te zetten, met een zo min mogelijke tussenkomst van een persoon. Dit gaan we doen aan de hand van een protocol genaamd TR-069. Door dit protocol te herprogrammeren binnenin een PHP-framework krijgen we de controle over verschillende sites van klanten zodat we deze vanop afstand kunnen beheren.

Om dit te verwezelijken met zo min mogelijke tussenkomst van een persoon gaan we op elke Mikrotik default config pushen. In deze config is het minimale geconfigureerd zodat er een veilige connectie gemaakt kan worden van de Mikrotik naar een toestel binnenin het Cegeka-netwerk. Deze default config bevat een aantal firewall rules zodat we zeker alleen vanaf Cegeka aan dit apparaat kunnen en de config om het TR-069 gedeelte volledig te laten functioneren, zoals de ACS-link , de authenticatie-gegevens en het interval.

Doordat we werken met default config moet de klant dit apparaat alleen maar voorzien van een internet verbinding en stroom aan de Mikrotik. Eenmaal deze connectie goed verlopen is, kunnen we vanuit Cegeka via een dashboard dit apparaat beheren en aanpassen. Vanaf dat dashboard wordt er dan ook config gegenereerd uniek voor ieder apparaat zodat er een VPN-tunnel opgezet wordt van het desbetreffende apparaat naar een cluster binnen Cegeka en naar al de andere apparaten van de klant. Doordat we een verbinding maken met een cluster in het datacenter en met ieder apparaat van de klant , krijgen we een full-mesh topology.

2.3 Gebruikte producten/technologieën

2.3.1 Mikrotik

2.3.1.1 Wat is Mikrotik

Mikrotik is een bedrijf gevestigd in Latvia dat is opgestart 1996. Mikrotik Levert zowel hardware als software voor internet connectiviteit over de hele wereld. Door middel van standaard pc's en routing systemen hebben ze in 1997 een nieuwe software ontwikkeld genaamd RouterOS. RouterOS is een standalone besturingssysteem gebaseerd op de Linux Kernel. Deze software biedt stabiliteit, controle, en flexibiliteit voor verschillende soorten interfaces en routers.

In 2002 zijn ze begonnen met het creëren van hun eigen hardware, genaamd RouterBOARD. Deze bordjes hadden standaard RouterOS geïnstalleerd en had tal van functionaliteiten. Vooraleer de RouterBoards op de markt waren werd RouterOS gebruikt om computers om te vormen naar een router met al de nodige functionaliteiten, Routing, Firewall, bandwidth, management, wireless access point, hotspot gateway, VPN-server en veel meer.

2.3.1.2 RouterBOARD

Mikrotik maakt verschillende series bordjes onder de naam RouterBOARD waarbij ze dan een implementatie doen van RouterOS. De RouterBOARDS wat in dit project gaan voorkomen zijn de RB750 GR3 (HEX series) en de CCR1009-8-1s-1s+pc.

2.3.1.3 RouterOS

RouterOS is het product waar Mikrotik mee begonnen is. Mikrotik is gebaseerd op de Linux Kernel en wordt geïnstalleerd op de RouterBOARD series of op pc's, het kan ook gebruikt worden als virtuele machine. Als RouterBOARD geïnstalleerd wordt op een pc, wordt deze volledig omgevormd in een netwerkrouter waardoor er verschillende features mogelijk worden : firewall, VPN en veel meer.

2.3.1.4 RouterOS functies

- Firewall/traffic shaping
- Point-to-point tunneling (PPTP, PPPoE, SSTP, OpenVPN)
- DHCP/proxy/Hotspot
- …

2 Cloud Hosted Routing

2.3.1.5 CHR: Cloud Hosted Routing

Cloud Hosted Routing is een aangepaste RouterOS software wat gericht is op virtuele machines zowel lokaal als in de cloud. Deze software is een virtuele image dat de volledig functionaliteit heeft van een origineel mikrotik product. Standaard is er een limitatie op de interfaces van 1Mbit/sec, maar hier zijn licenties voorzien om deze limitatie op te heffen.

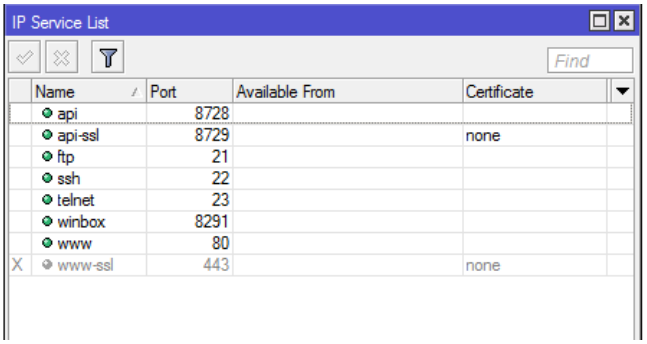
2.3.1.6 Hoe Verbinden met een RouterBoard

Een routerboard heeft verschillende mogelijkheden om een verbinding te maken. Standaard heeft een Routerboard een vast IP-adres ingesteld, namelijk 192.168.88.1. Hierdoor is het makkelijk via een IP-adres te verbinden. Indien we via WinBox connecteren is het ook mogelijk via Mac-adres te verbinden wat dan weer handig is indien het IP-adres verwijderd is.

2.3.1.7 Connectiemogelijkheden

Om met een RouterBOARD interface te connecteren heeft men 3 mogelijkheden:

- Verbinden via Commandline (CLI)
- Verbinden via Web based GUI(webfig)
- Verbinden via WinBox application



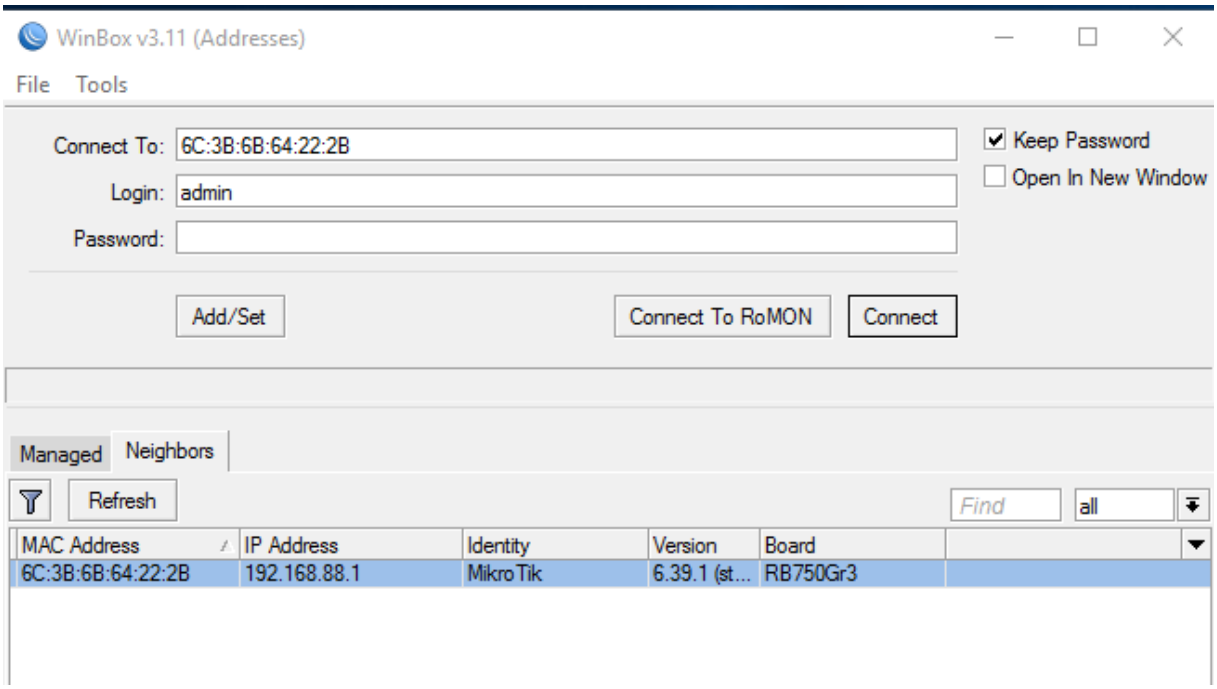
3 IP Service List

2.3.1.8 De setup

Om connectie te krijgen met het RouterBOARD heb ik de ethernet poort van mijn laptop verbonden met de LAN-poort van het RouterBOARD. Na dat deze verbinding actief was kon ik gaan verbinden met een van de connectiemogelijkheden. Doordat de Mikrokit routers standaard DHCP geconfigureerd heeft moest ik geen statisch IP-adres instellen en kreeg ik een IP-adres toegekend van de DHCP-server.

2.3.1.9 Connecteren via Winbox

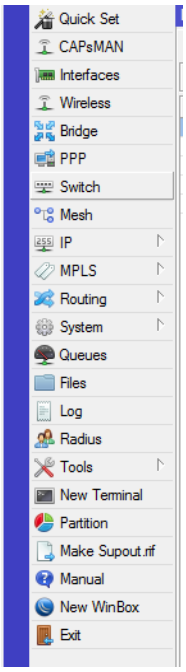
Winbox is een klein programma dat administrators toestaat de GUI van RouterOS makkelijk en snel te gebruiken. Winbox is standaard gemaakt voor op Windows te gebruiken, maar het is ook mogelijk deze op Linux of MacOS te gebruiken door gebruik te maken van Wine.



4 RouterOS Winbox interface

Bij het opstarten van Winbox verschijnt het venster waar verbinding met RouterOS tot stand kan gebracht worden. Bovenaan bevindt zich het veld “Connect to” waar er een IP-adres of een MAC-adres ingevuld kan worden.
Daaronder bevindt zich de login, standaard is deze admin. Eenmaal ingelogd kunnen er altijd meerdere accounts gemaakt worden. Het wachtwoord voor admin is standaard leeg.

Onderaan het venster is er een lijst zichtbaar met alle RouterOS Neigbors.
Hier is het MAC-adres en het IP-adres van de RouterOS zichtbaar.
Bij het klikken op het MAC-adres is het mogelijk te connecteren met het RouterBOARD via Mac-adres. Dit geeft als voordeel dat het RouterBOARD geen IP-adres moet hebben om te kunnen connecteren. Bij het klikken op het IP-adres wordt het IP-adres in de “Connect To” balk ingevuld. Als al de gegevens ingevuld zijn kan er op Connect geduwd worden en zal de RouterOS interface verschijnen.



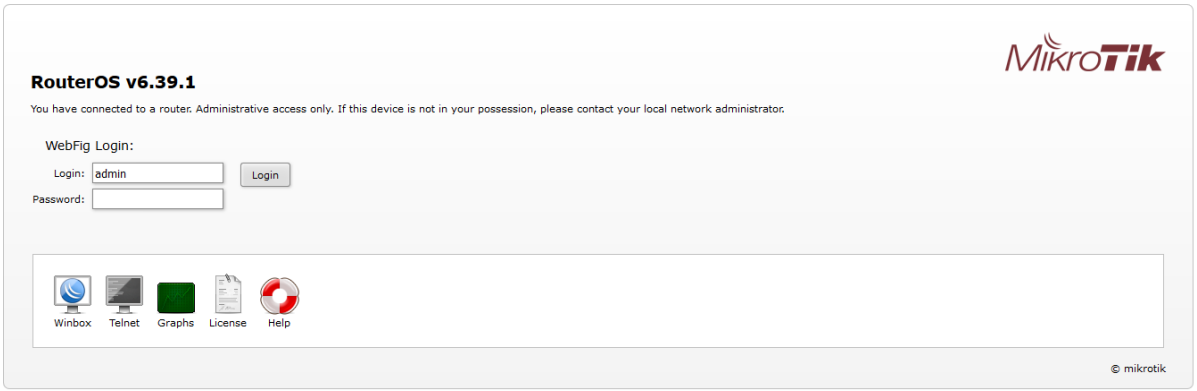
Na het inloggen via Winbox komt het hoofdscherm waar we aan de linkerkant een menu hebben met alle functionaliteiten van RouterOS.

Aan de bovenkant van het venster bevindt zich meer informatie over het toestel zelf Zoals hoelang het toestel al aanstaat (Uptime), Memory, de datum, en het CPU verbruik.

5 Winbox Menu

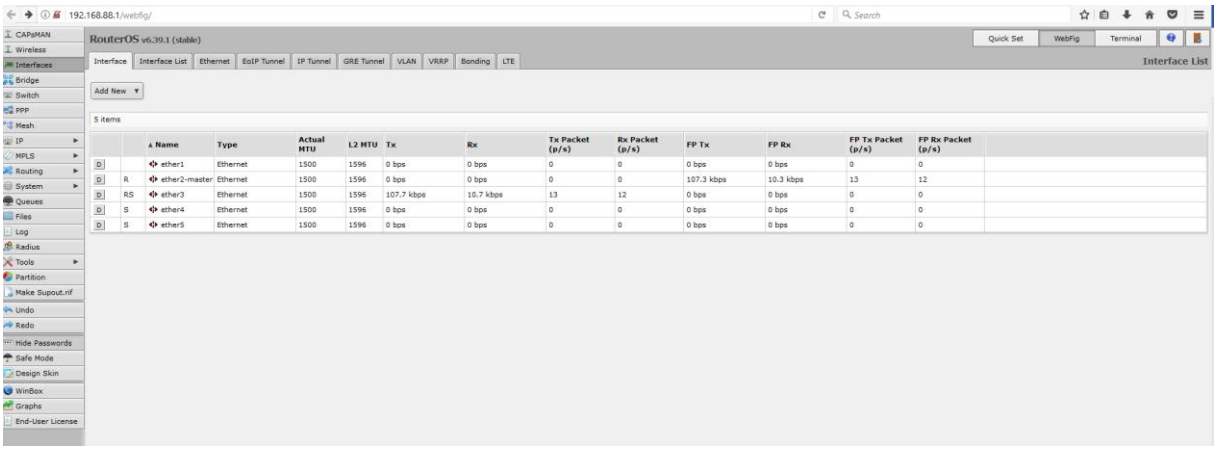
2.3.1.10 Verbinden via browser

Om te connecteren via een browser geeft men het IP-adres in en komen we op de login pagina van RouterOS. Hier kunnen we weer inloggen met Admin als username en geen wachtwoord. Onderaan staan ook de andere mogelijkheden om in te loggen.



6 RouterOS web interface

Eenmaal succesvol ingelogd komen we op de web interface uit van het RouterBOARD.



7 RouterOS web interface logged in

Eenmaal ingelogd op de RB zijn al de interfaces zichtbaar. Elke interface heeft ook een letter voor de interface staan, “R”, “RS” of “S”.

R = Running

RS = Running slave

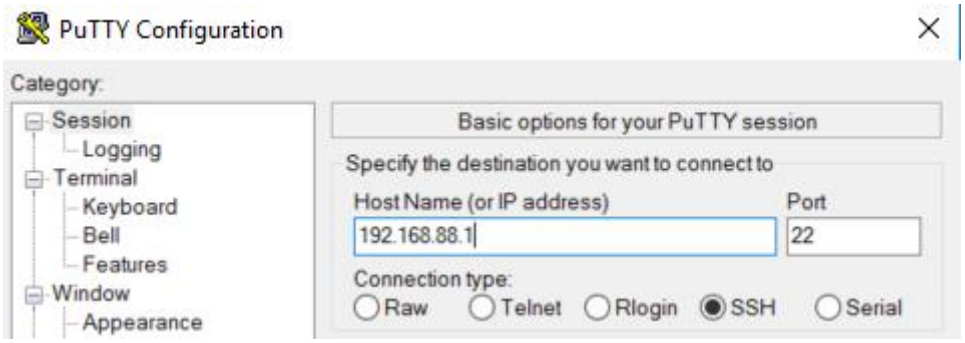
S = Slave

Hieronder staat 1 interface met “Master”. Dit betekendt dat die interface gebruikt is als bridge poort.

Interface 3 – 5 heeft als Master port de ethernet-2-port staan, en zo wordt data tussen deze interfaces verwerkt via een switchchip in plaats van de CPU van het routerboard.

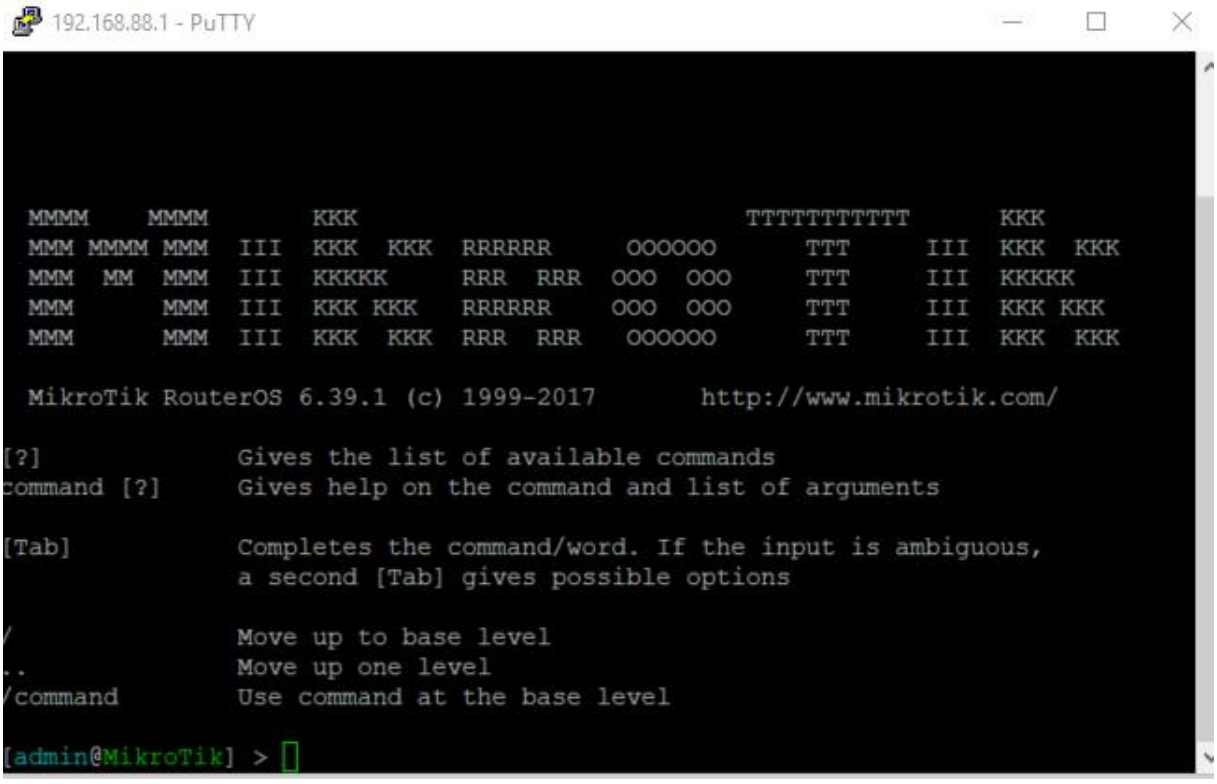
2.3.1.11 Verbinden via SSH

Om te connecteren via SSH kan er gebruik gemaakt worden van Putty. Putty is een kleine SSH/ telnet client.



8 Putty

Na het verbinden met het RouterBOARD komen we op de terminal.

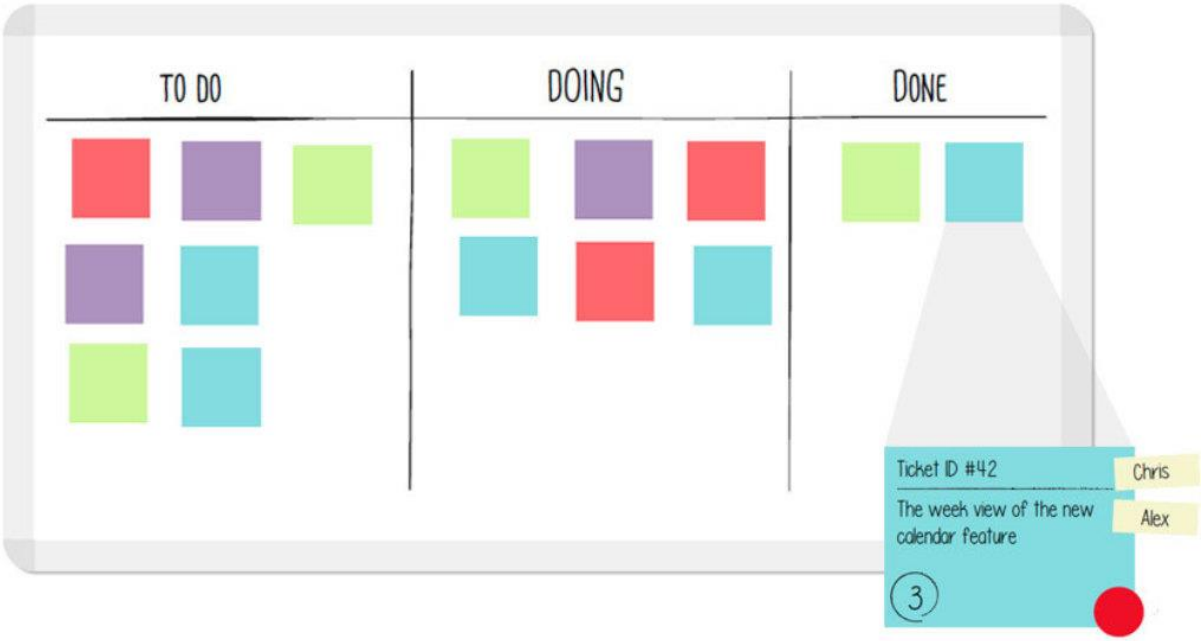


9 RouterOS terminal

2.3.2 Kanban

2.3.2.1 Geschiedenis

Kanban is een concept dat gebruikt wordt in just-in-timeproductie.
Het werd 60 jaar geleden ontwikkeld in Japan door een man genaamd Taiichi Ohno.
In het Japans staat “Kan” voor visueel en “Ban” voor bord wat dan omschrijft wat het bord effectief is. Het Kanbanbord werd ontwikkeld bij het merk Toyota om de productie te laten stijgen. Vanaf 1970 werd het bord bekend in het westen.



10 Kanban bord

2.3.2.2 Werking van een Kanban

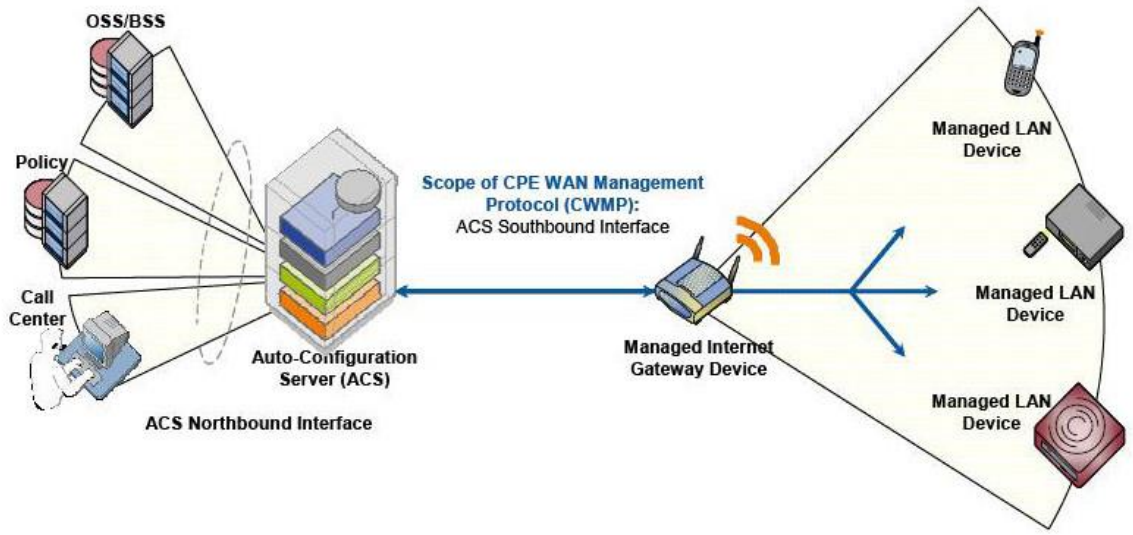
Kanbanborden worden gezien als variaties van de traditionele Kanbankaart. In plaats van een Kanban kaart dat een vraag of een capaciteit weergeeft, wordt op het bord gebruikgemaakt van magneetjes, plastic kaartjes, plakkaartjes, stickers of post-its, die een hoeveelheid werk of een taak vertegenwoordigen. Dit is dan een visueel signaal dat laat zien in welke status het proces zich bevindt. Wanneer er problemen zijn met een bepaald proces zal dit ook zichtbaar zijn op dit bord zodat ieder lid van het team op de hoogte is van het probleem. Doordat dit systeem zo goed werkt, wordt het vaak gebruikt en opgenomen in nieuwe ERP-systemen.

2.3.2.3 Kanban binnen Cegeka

Cegeka is een groot bedrijf met verschillende projecten waar verschillende afdelingen aan werken. Om deze zo goed en zo snel mogelijk af te ronden moet er duidelijkheid zijn, hierdoor speelt een kanbanbord een grote rol.

Cegeka maakt gebruik van een Kanbansysteem genaamd LeanKit. LeanKit is een planningssysteem dat bedrijven helpt in hun productieproces. Het is een overzichtelijk systeem waar duidelijk zichtbaar is welke status welk proces heeft door gebruik te maken van kaartjes, stickers, post-its of magneten. Het Kanbanbord kan aangepaste statussen hebben waarvan Cegeka heeft gekozen voor “Ready”, hier staat het project als het pas aangemaakt is, wachtend totdat iemand start met dit project. “Analyse” kan onderverdeeld worden in “Do it”, wat betekend dat de analyse momenteel bezig is, of “Done”, dat de analyse gedaan is. Na de analyse kan “WIP” of te wel “Work in progress” gestart worden.
Hier is ook weer een onderverdeling tussen “DO IT” en “DOC”, waarbij DOC staat voor “Documentation”. Als laatste stap hebben we “Done”, hier staan al de projecten die volledig afgerond zijn.

2.3.3 TR-069



11 TR-069

2.3.3.1 Wat is TR-069

TR-069 of te wel Technical Report 069 is een specificitie dat zich afspeelt binnenin de application layer van het OSI-model. Deze specificatie wordt gebruik voor het van op afstand bedienen van CPE (Customer-Premises Equipment) dat verbonden is het met het internet. TR-069 werkt tussen een ACS (Auto Configuration Server) en een CPE.TR-069 maakt gebruik van het CWMP(CPE Wan Management Protocol) dat gebruik maakt van SOAP/http(s) communicatie. Wanneer een ACS communicatie start met een CPE zal dit altijd over http of http(s) calls gebeuren. Binnen deze calls wordt er gebruik gemaakt van SOAP(Simple Object Access Protocol) berichten. Via SOAP kunnen er gestructureerde berichten naar elkaar verzonden worden. Deze berichten zijn opgesteld zoals XML-bestanden (Extensible Markup Language).

2.3.3.2 Functies

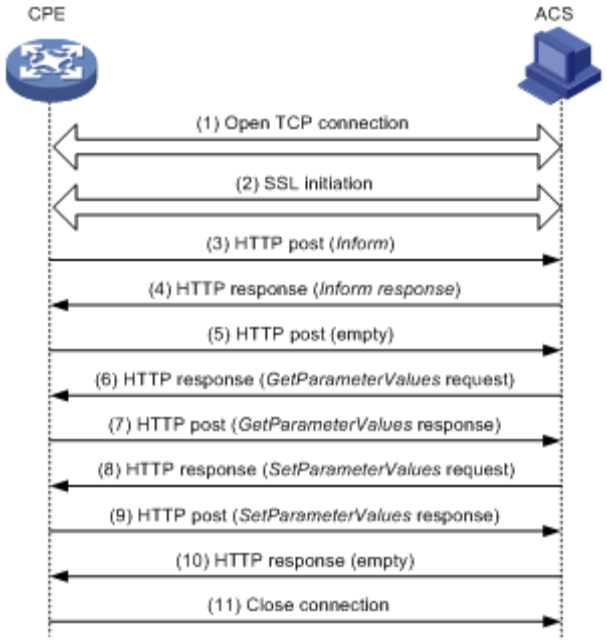
Het CPE Wan Management Protocol beschikt over verschillende functies om een groep van CPE toestellen te beheren.

- Auto-configuration en dynamic service provisioning
- Software/firmware image management
  Het CPE WAN Management Protocol biedt tools voor het beheren van het downloaden van CPE-software / firmware.
- Status en performance monitoring
- Diagnostics

2.3.3.3 Session

Een session is nodig om data uit te wisselen tussen de CPE en de ACS. Een sessie wordt altijd geïnitialiseerd vanuit de CPE. Binnenin deze sessie worden continue SOAP berichten verstuurd naar elkaar waar de nodige informatie in staat om de gedifinieerde acties in het SOAP-bericht uit te voeren.

Hoewel de CPE normaal altijd de sessie start is het ook mogelijk dat de ACS een sessie aanvraagt. Dit kan door het definieren van een Connection Request. Dit zal een request sturen naar de CPE waarin staat dat de ACS een sessie wilt starten. Hierna zal de CPE een sessie starten.



12 TR-069 session

Een sessie verloopt in verschillende stappen :

1. Het opzetten van een TCP connectie.
2. SSL wordt geinitialiseerd.
3. De CPE stuurt een inform request om een CWMP-connectie tot stand te brengen.
   Binnenin een inform request zit informatie waarom de inform gestuurd werd en informatie over het device dat de inform stuurt.
   De device informatie heeft een standaard structuur waar we de fabriekant, OUI, productklasse en het serialnummer vinden.

```
<DeviceId>
  <Manufacturer>Mikrotik</Manufacturer>
  <OUI>999A</OUI>
  <ProductClass>CHR</ProductClass>
  <SerialNumber>0XCAFEBABE</SerialNumber>
</DeviceId>
```

4. De ACS bekijkt de informatie wat verstuurd werd van de CPE en stuurt aan de hand van of deze CPE een connectie mag maken een response. Wanneer er een inform response gestuurd wordt kan de sessie doorgaan. De ACS kan ook een finish response sturen waardoor de connectie beëindigd wordt.
5. Wanneer de CPE een inform response ontvangt van de ACS wordt er een lege http post verstuurd waardoor de ACS weet dat hij vanaf nu acties mag sturen.
6. Vanaf hier kan eender welke actie verstuurd worden naar de CPE, zoals getparameters , setparameters , reboot , download , upload.De eerste 5 stappen kunnen gezien worden als een authenticatie van de CPE. Vanaf dan verschillen de stappen aan de hand van welke actie wordt verstuurd.

2.3.3.4 Events

Wanneer er een inform wordt gedaan naar een acs zit er altijd een reden achter deze inform. Deze reden noemen we een event, waarom wordt er een inform gestuurd? Iedere inform zal altijd een event als parameter meegeven.

- 0 BOOTSTRAP

  Het bootstrap event is het event dat gebruikt wordt wanneer er voor de eerste keer een inform wordt gedaan naar de ACS, of wanneer de ACS url van de CPE verandert. Meestal wanneer dit event wordt verstuurd wordt er initiele configuratie gedaan, zoals het configureren van de basisparameters en het opvragen van de parameters.

- 1 BOOT

  Het BOOT event wordt gebruikt wanneer de CPE een reboot heeft gedaan. Dit event wordt meestal opgeroepen bij een fysieke rebot.

- 2 PERIODIC

  Het periodic event is een event waar de naam zelf al uitlegt wat er zich allemaal afspeelt bij dit event. Bij de configuratie van een CPE kan er een periodic interval meegegeven worden zodat de CPE continue de ACS blijft aanspreken. Dit maakt het makkelijk om te kijken of de het device nog online is.

- 3 SCHEDULED

  Dit event is gebaseerd op het gebruik van scheduleinform RPC.

- 4 VALUE CHANGE

  Ook dit event verklaart zichzelf, bij het veranderen van een waarde, dat gemakeerd is als passief of actief notificeren, wordt er een inform gestuurd naar de ACS met als event dat er een value changed is.

- 6 CONNECTION REQUEST

  Een connection request wordt gebruikt wanneer de ACS een connection request gebruikt om de CPE een sessie te beginnen me de ACS.

- 7 TRANSFER COMPLETE

  Transfer complete wordt gebruikt nadat er een file succesvol geupload of gedownload wordt van of naar de CPE. Door dit event weet de ACS dat de download of upload succesvol verlopen is.

- 8 DIAGNOSTICS COMPLETE

  Dit event zal verschijnen wanneer de CPE één of meerdere diagnose tests afgerond heeft.

- 9 REQUEST DOWNLOAD

  Dit event wordt gebruikt wanneer de CPE een sessie wilt starten waar het de requestdownload RPC wilt aanroepen op de ACS.

- 13 WAKEUP

Wanneer het device in “sleep” mode is of in low-power mode en het komt terug online , zal het wakeup-event gebruikt worden.

2.3.3.5 Method EVENTS

Buiten al deze events zijn er ook nog andere soorten events. M of Method events worden ook nog gebruikt binnenin het event gedeelte. Deze events zijn gelabeld met M dat staat voor methode. Deze methodes zullen altijd samen met de normale events gebruikt worden.

- M Reboot

M Reboot wordt meestal gebruikt samen met 1 BOOT. Dit laat weten dat het device opnieuw is opgestart vanwege de reboot RPC.

- M Scheduled inform

Deze methode wordt altijd samen met 3 SCHEDULED opgeroepen.

- M Download

M Download komt samen met 7 TRANSFER COMPLETE wanneer een download wordt aangeroepen wanneer een download succesvol is.

Dit zijn alle standaard TR-069 events. Buiten deze kunnen fabriekanten zelf ook nog zelf events creëren.

2.3.3.6 Files

Op afstand bedienen van devices is zeer nuttig wanneer het device zich op een langere afstand bevind. Hiervoor maakt TR-069 het makkelijk voor service provides door devices van op afstand te bedienen. Wanneer we kijken naar TR-069 voor wat dit het meest gebruikt wordt is dit wel voor firmware updates te doen.

Deze updates kunnen uitgevoerd worden aan de hand van de Download RPC. Dpw,mpad RPC wordt vaak gebruikt om verschillende soorten bestanden naar de CPE te downloaden , maar wanneer we firmware gaan transporteren naar de CPE komt er een volledig proces bij kijken om deze firmware update succesvol uit te voeren. Dit doordat een firmware update alleen succesvol is wanener de download slaagt en de firmware succesvol wordt toegepast.

Wanneer we een download RPC gaan bekijken bestaat deze uit 13 verschillende argumenten.

De eerste is de key , en als tweede komt het type file als argument mee. We hebben 5 verschillende type files die we kunnen downloaden naar de CPE.

1- Firmware Upgrade Image
2- Web Content
3- Vendor Configuration File
4- Tone file
5- Ringer File

4 en 5 worden specifiek gebruikt bij VOIP bij TR-104.

2.3.3.7 Faultcodes

Wanneer er parameters aangepast worden of downloads uitgevoerd worden willen we altijd feedback krijgen van deze acties , zodat we op de hoogte zijn van wat er gebeurd met onze actie.

TR-069 werkt foutcodes uit via SOAP. SOAP gebruikt zijn eigen methodes om errors te behandelen. Indien er een foutieve actie wordt verstuurd naar de CPE zal er een CWMP:FAULT verstuurd worden vanaf de CPE waarin vermeld staat wat er precies fout ging. Als we een foutcode gaan bekijken bij het aanpassen van een parameter die niet bestaat op de CPE zal deze er ongeveer zo uitzien :

```
<soap:Body>
                    <soap:Fault>
                    <faultcode>Server</faultcode>
                    <faultstring>CWMP fault</faultstring>
                    <detail>
                    <cwmp:Fault>
                    <FaultCode>9005</FaultCode>
                    <FaultString>Parameter not found</FaultString>
                    <SetParameterValuesFault>
                    <ParameterName>InternetGatewayDevice.DeviceInfo.Foo</ParameterName>
                    <FaultCode>9005</FaultCode>
                    <FaultString>Parameter not found</FaultString>
                    </SetParameterValuesFault>
                    <SetParameterValuesFault>
                    <ParameterName>InternetGatewayDevice.DeviceInfo.Bar</ParameterName>
                    <FaultCode>9006</FaultCode>
                    <FaultString>Parameter not writable</FaultString>
                    </SetParameterValuesFault>
                    </cwmp:Fault>
                    </detail>
                    </soap:Fault>
</soap:Body>
```

Hierin is de error meegegeven , namelijk 9005. Deze error staat voor het meegeven van een foutieve naam van een parameter. Deze fout komt meestal voor bij Set/GetParameterValues, GetParameterNames, Set/GetParameterAttributes, AddObject, en DeleteObject.

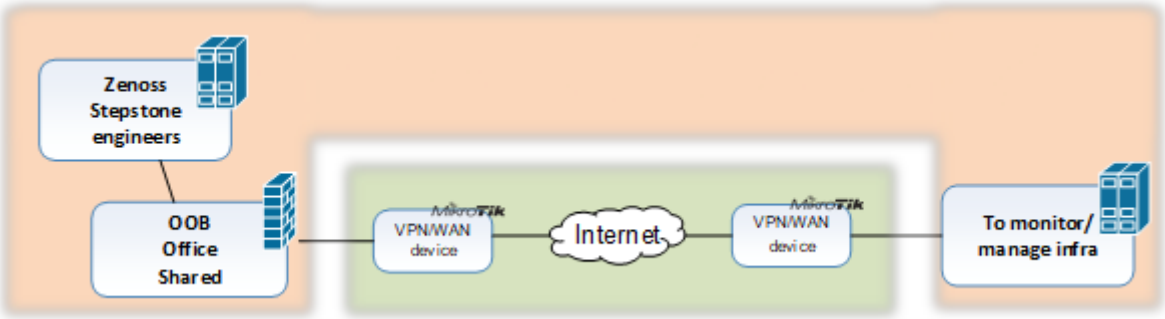
3 Uitwerken stageopdracht

3.1 Gebruikte oplossingen voor automatisatie

3.1.1 High level oplossingen

Om een VPN-tunnel op te bouwen is er veel informatie nodig over beide eindapparaten van een tunnel. Via deze informatie kan er dan configuratie gegenereerd worden wat uiteindelijk naar ieder apparaat gestuurd wordt zodat deze configuratie uitgevoerd kan worden en de tunnel opgebouwd wordt.

Als we de infrastructuur binnenin Cegeka gaan bekijken krijgen we de volgende componenten :



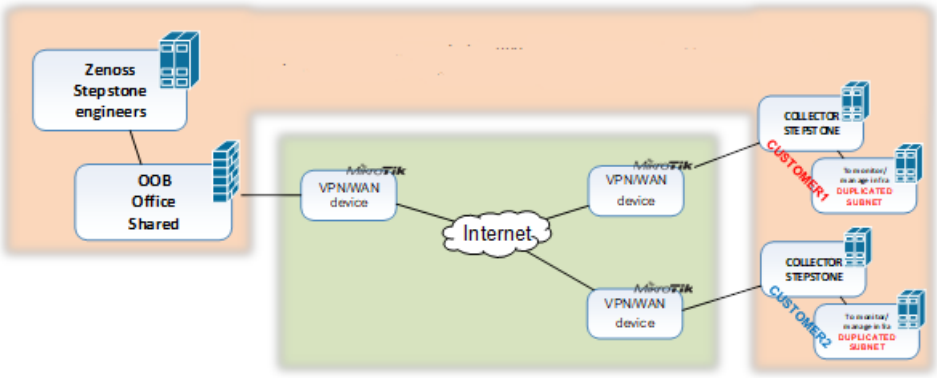
13 Cegeka Infrastructuur

Voor de gedeelde services (Monitoring/management vanaf stepstone) moet er een route over de nioeuwe Mikrotik VPN oplossing gaan, maar als we de klant de keuze geven te kiezen welk subnet hij heeft, en wat gerouteerd moet worden, is de kans groot dat er meerde dezelfde subnets zullen bestaan.

Hiervoor zijn er verschillende oplossingen mogelijk zodat we de uiteindelijke opstelling toch kunnen verwezenlijken zonder problemen tussen verschillende sites te veroorzaken.

De eerste mogelijkheid zou zijn dat we gaan werken met PBR (Policy Based Routing) en VRF(Virtual Routing and forwarding). Via deze methode zullen er geen gedupliceerde netwerken bestaan.

Een andere optie zou kunnen zijn dat we het probleem oplossen aan de kant van de klanten.



Bij deze optie gaan we werken met een dedicated proxy bij iedere site. Hierdoor kunnen er geen overlappende IP-adressen voorkomen en kunnen we routeren binnenin één apparaat/VRF op de Cegeka site.

3.1.2 Mogelijke designs met een Mikrotik

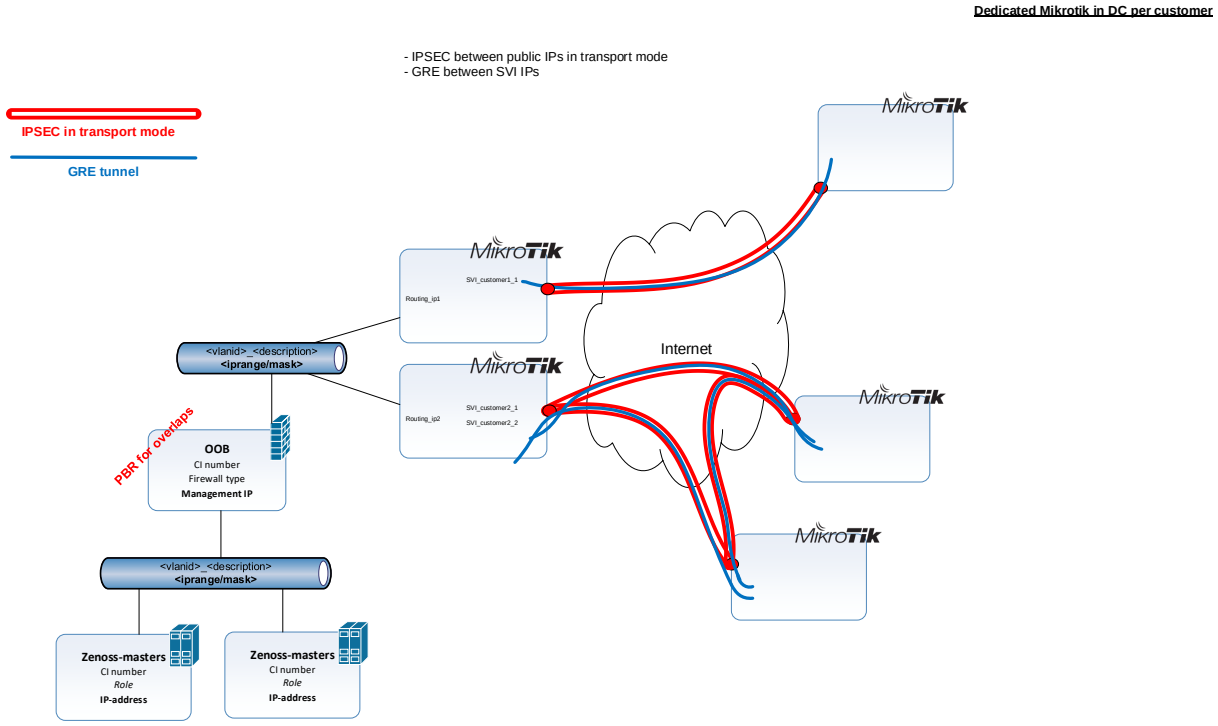
Wanneer we designs gaan bekijken met Mikrotiks in betrokken moeten we opletten met een aantal punten :

- Management IP-adres kan alleen maar in de main routing tabel voorkomen. In de oudere versies van RouterOS was het nog mogelijk het management IP-adres in een VRF te steken. Maar dit is met de nieuwe versies van ROS niet meer mogelijk.
- Doordat we gebruik maken van VRF's is het niet mogelijk om te routeren tussen verschillende routing tables . Het is alleen mogelijk om te routeren van de main routing table naar de verschillende VRF's. Ook kan er maar 1 fysieke interface/VLAN gekoppeld worden aan een VRF.

3.1.3 Verschillende oplossingen

3.1.3.1 Dedicated Mikrotik per klant

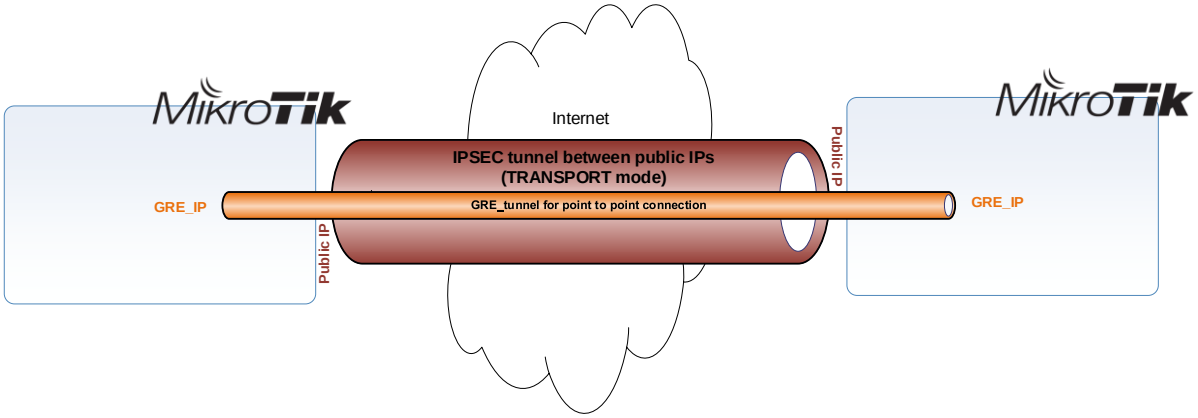
De makkelijks oplossing is om dedicated Mikrotiks in het datacenter te plaatsen voor iedere klant. Hierdoor kan al de routing gebeuren in de main routing tabel op elke Mikrotik apart. Wanneer we overlappende netwerk ranges tegenkomen bij de klant kunnen we dit oplossen door gebruik te maken van Policy Based Routing (PBR).



14 Dedicated Mikrotik per klant schema

Voor een connectie van en naar een remote site moeten we al het verkeer gaan encrypteren dat over het publiek internet wordt gestuurd. Dit kan gedaan worden aan de hand van tunnels. We hebben in het onderzoeksrapport al meer informatie gegeven over de verschillende tunnels dat een Mikrotik aanbied, dus gaan we daar niet verder op ingaan.

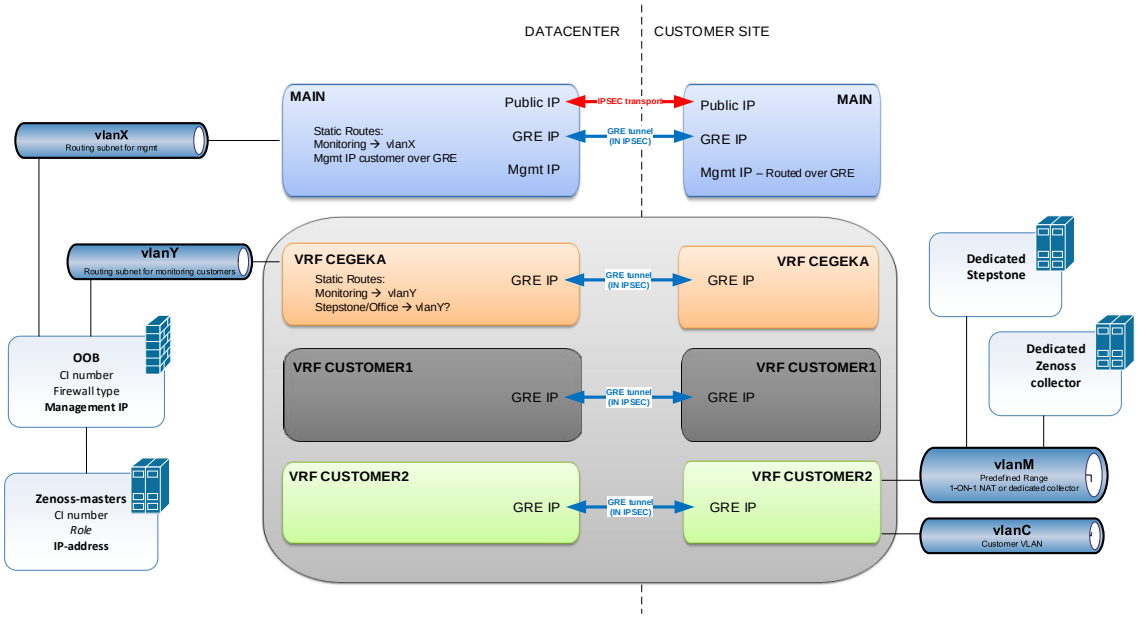
Om al het verkeer te encrypteren gaan we gebruik maken van IPSEC. IPsec gaan we gebruiken in transport mode waardoor al het verkeer van endpoint tot endpoint geëncrypteerd is. Om beide apparaten met elkaar te laten communiceren gaan we gebruik maken van een GRE tunnel om een point-to-point connectie op te zetten. Deze tunnel hebben we uitgebreid getest in de onderzoeksopdracht en daarom gaan we deze tunnel gebruiken voor al oplossingen die beschreven gaan worden.



15 GRE/IPSEC schema

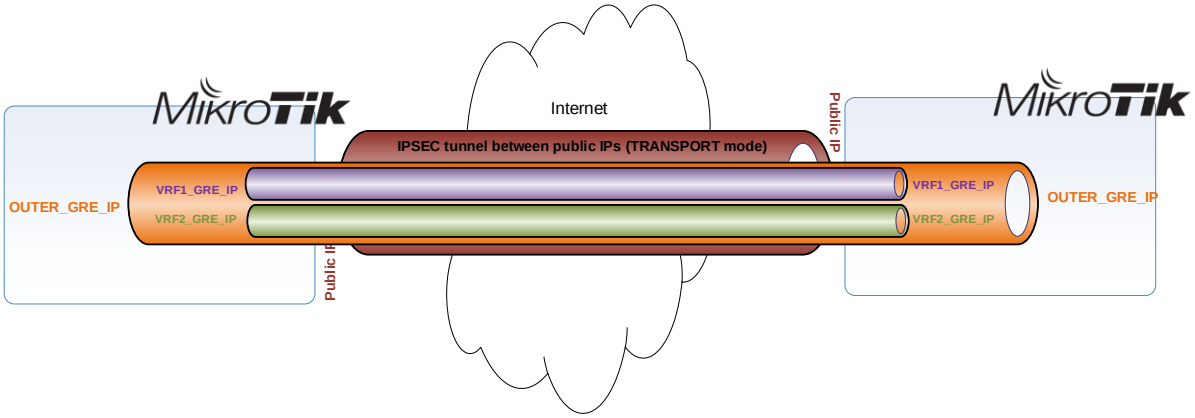
3.1.3.2 GRE/IPSEC met routing protocol (OSPF/BGP)

Momenteel wordt er binnen Cegeka eenzelfde setup gebruikt. Hierin gebruiken we een redundant device in het datenceter en OSPF als routing protocol. VRF's gaan hier gebruikt worden om klanten van elkaar te scheiden , en zelfs verschillende klantomgevingen van elkaar te scheiden.



16 Oplossing met GRE/IPSec en OSPF/BGP

In deze setup gaan we gebruik maken van één VRF per klant waardoomr we voor iedere klant een GRE-tunnel moeten opstellen bovenop de standaard GRE-tunnel. Hierdoor gaan we veel GRE-tunnels binnein een GRE-tunnel krijgen waardoor het moeilijk wordt om deze setup te gaan troubleshooten.



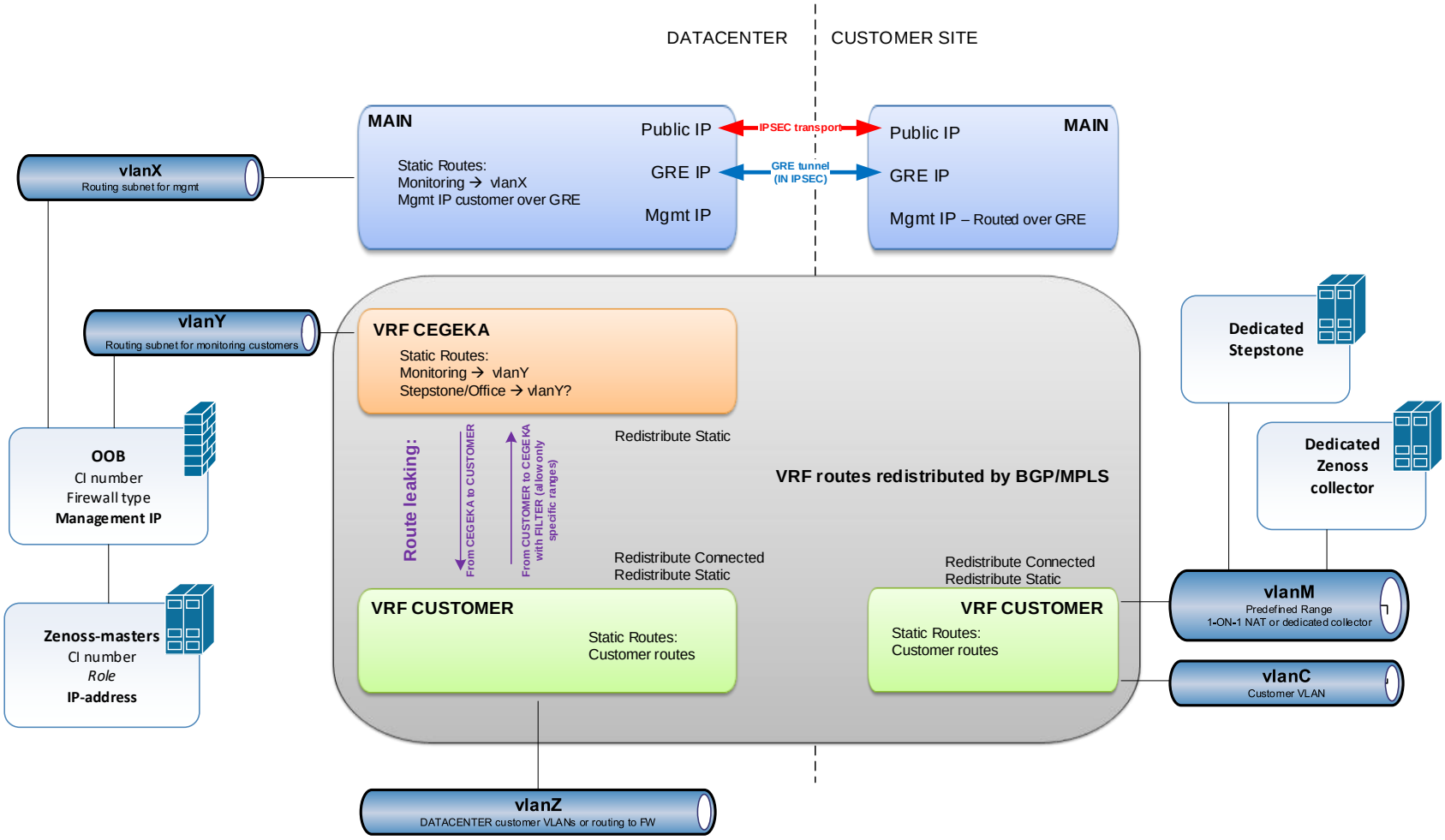
17 GRE/IPSEC met verschillende vrf's

3.1.3.3 GRE/IPSEC + MPLS/BGP + Route leaking

In deze setup is er 1 of een redundant apapraat in het datacenter dat gedeelt wordt voor verschillende klanten. Op de externe site gebruiken we 1 of meerdere apparaten voor redundancy.

Hier gebruiken we , zoals dde andere setups, gebruik van IPSEC in transport mode , dit om al de communicatie te encrypteren van endpoint naar endpoint. Om communicatie door te laten door deze tunnels hebben we een GRE tunnel over deze IPSEC tunnel gebouwd. Door dit de configureren hebben we een point-to-point connectie van het ene apparaat naar het andere.

De enige speciale functies van BGP/MPLS die in deze setup gebruikt worden zijn diegene om VRF-routes te transporteren over de tunnels. Doordat we gebruik maken van MPLS/BGP kunnen we een VRF aanmaken voor een klant op hun apparaat , en al hun routes in desbetreffende VRF zullen gedupliceerd worden naar al de Mikrotiks waar die VRF bestaat.



18 GRE/IPSec + MPLS/BGP + Route leaking

3.1.4 Conclusie

Na heel wat onderzoek en besprekingen over de verschillende mogelijkheden zijn we gegaan voor de MPLS/BGP oplossing met route leaking en PBR aan de andere kant.

Doordat we gebruik maken van deze oplossing kunnen we zeker zijn dat al de klanten de routes van Cegeka krijgt, en en we kunnen beslissen welke routes van een klant we willen binnenhalen in de Cegeka-VRF door mangle rules.

Doordat we deze oplosssing gebruiken hebben we ook volledig de controle over overlappende IP-ranges en wordt het troubleshooten makkelijker.

3.2 De automatisatie

Na het onderzoek van welke VPN-tunnel we effectief gaan implementeren en de verschillende mogelijkheden hebben beken hoe we het best uitkomen wat betreft routering bij verschillende klanten, kunnen we deze informatie allemaal gaan gebruiken en automatiseren zodat de gekozen opstellingen aan de hand van een paar klikken op een knop kunnen opstellen.

3.2.1 De programmeertaal

Bij automatiseren hoort het woord programmeren, we willen iets wat we nu handmatig doen automatisch laten gaan. Hiervoor gaan we gebruikmaken vaan een programmeertaal. Hierbij heb ik eerst wat informatie gevraagd over hoe ze binnen Cegeka verschillende handmatige acties geautomatiseerd hadden. Scripting via python was het eerste waar ik aan dacht en dit bleek een taal te zijn waardoor we al konden starten , zonder na te denken hoe we hierop verder konden bouwen na verloop van tijd. We waren begonnen met het scripten in Python en hadden een callhome-script aangemaakt. Via dit script ging de Mikrotik telkens een callhome doen naar een server zodat we al communicatie hadden. Vooraleer we verder gingen hierin ben ik TR-069 tegengekomen en hebben we dit onderzocht en besloten Python te laten vallen en verder te gaan via TR-069. Dit protocol is dan uiteindelijk via PHP tot stand gekomen. Ikzelf werk graag in PHP waardoor dit zeker een pluspunt was om TR-069 om te vormen naar PHP.

3.3 Het Proces

Het opbouwen van een VPN-tunnel heeft een aantal stappen die overlopen zullen worden bij het opbouwen van een tunnel. Als we kijken naar de stappenkijken die momenteel nodig zijn voor een VPN-tunnel op te bouwen zien we dat er een aantal stappen vermeden kunnen worden. Ook een groot deel van deze stappen zullen geautomatiseerd zijn. Wanneer we het stappenplan bekijken dat we willen realiseren via de automatisatie bekome we dit :

Stap 1 : Default config wordt gepusht op een nieuw apparaat. De default config laat toe dat IP-ranges van Cegeka dit apparaat kunnen bereiken. Ook wordt alles geconfigureerd wat beftreft TR-069. De ACS-link wordt ingesteld , samen met de username en het password vloor de authenticatie.

Stap 2 : Het apparaat wordt vanaf Cegeka verstuurd naar de klant met de default config op gepusht.

Stap 3 : De klant krijgt het apparaat geleverd en hoeft alleen maar een ethernet kabel in te steken op de éérste ethernet poort zodat het apparaat een internetverbinding heeft. Ook moet dit apparaat voor stroom voorzien worden.

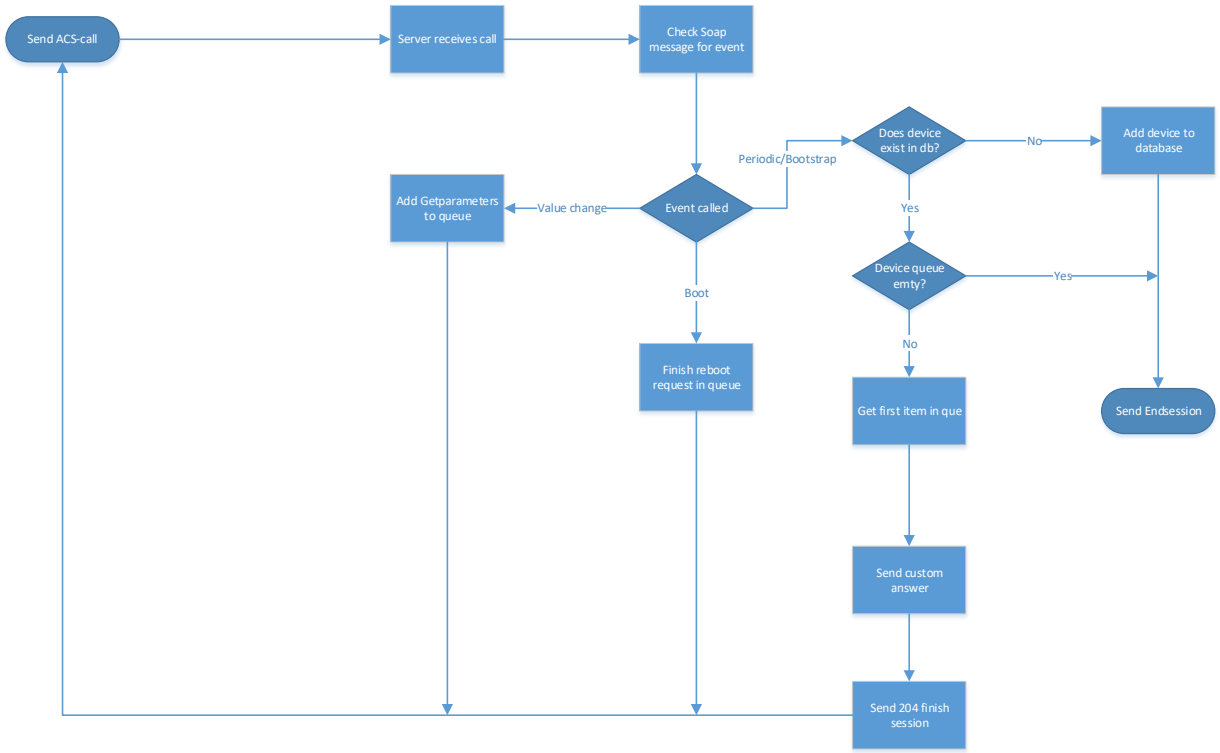
Stap 4 : Het apparaat doet een ACS-call naar de servers binnen Cegeka. Deze call wordt opgevangen en het apparaat wordt bij al de andere apparaten toegevoegd met de status “Waiting”. Nu kan het apapraat geaccepteerd of verwijderd worden door Cegeka.

Stap 5 : Wanneer het apparaat geactiveerd is worden de parameters van het apparaat opgehaald en kan de configuratie gegenereerd worden. De config files worden aangemaakt en in de queue geplaatst. Bij de eerst volgende ACS-call wordt deze file meegegeven naar het apparaat.

Stap 6 : De config is succesvol naar het apparaat on-site gepusht en de tunnel wordt automatisch opgebouwd.

3.4 Stappenplan

3.4.1 Flowchart



19 Flowchart stappenplan ACS-call

3.4.2 Flowchart stap per stap

- Send ACS-call

  Vanaf hier begint het proces altijd. De client die een ACS-call gaat sturen naar de server. De ACS-url link is ingesteld bij het TR-069 protol met de opgegeven authenticatiegegevens.

- Server receives call

  De server krijgt de ACS-call binnen van de client.

- Check SOAP-message for event

  De ACS-call heeft altijd informatie bij zich zodat de server weet waarom de call gedaan werd. Deze informatie bevind zich in een SOAP-message. Deze is opgesteld uit XML-data en hierin vinden we het event waarom deze call gedaan word.

- Event called

  De server gaat de XML-data ontleden zodat we het event terugvinden. Het event bepaald nu hoe het verder verloop is van de connectie.

  - Periodic / Bootstrap
  - Boot
  - Value change
  - Andere

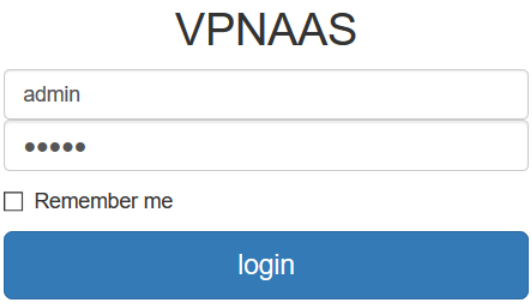
  Eenmaal we bij het event zijn kan dit verschillende paden afleggen doordat er op elk event anders gereageerd moet worden. De meest gebruikte zal periodic zijn en daarom is deze volledig uitgewerkt in de flowchart.

3.5 Het Dashboard

Het dashboard is het front-end van de volledige applicatie. We zouden de configuratie kunnen automatiseren zonder een front-end dashboard te maken maar dan zou dit allemaal via de commandline gedaan moeten worden. Daarom is er via Symfony en Bootstrap een dashboard gemaakt.

3.5.1 De login

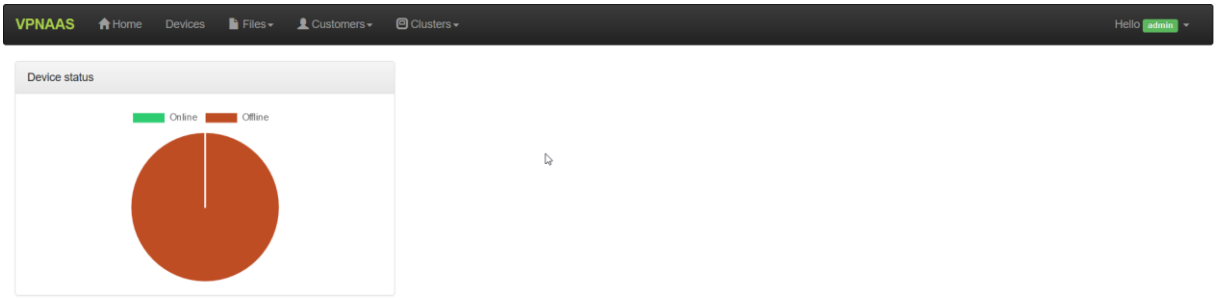
Via dit dashboard gaan er verschillende apparaten beheerd worden en kunnen er verschillende functies uitgevoerd worden op deze apparaten. Ook apparaten van het datacenter zitten in dit dashboard waardoor er toch wel beveiliging op moet zitten. Daarvoor heb ik gebruik gemaakt van sessions binnenin Symfony waardoor er altijd ingelogd moet zijn voordat we eender welke pagina van het dashboard willen bekijken.



20 Login dashboard

3.5.2 De home pagina

Op de homepagina staat er momenteel alleen een cirkeldiagram dat laat zien hoeveel apparaten er online zijn en hoeveel offline. Bij uitbreiding van VPNAAS zouden er nog meer diagrammen kunnen bij geprogrammeerd worden zodat dit meer als een monitor pagina gezien kan worden.

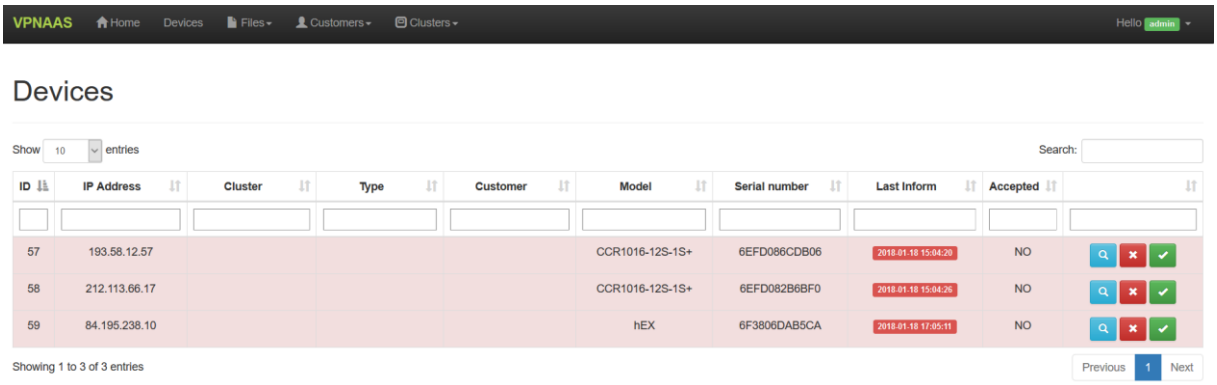


21 Home pagina dashboard

3.5.3 Devices

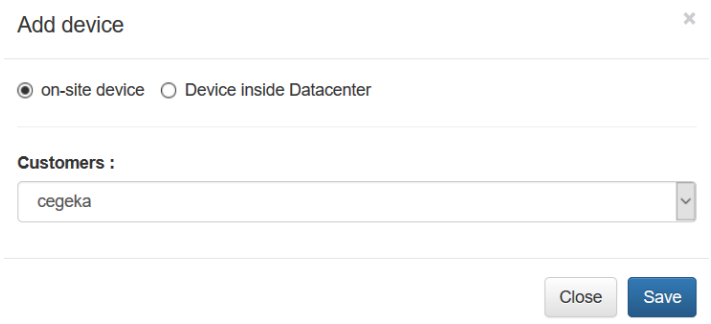
Devices is een van de belangrijkste paginas binnen het dashboard. Hier is een overzicht van al de apparaten die contant hebben gemaakt met de server.

Als de client contact zoekt met de server voor de eerste keer, komt deze in de lijst te staan met een rode achtergrond. Dit laat zien dat dit een nieuw apparaat is en dit nog geaccepteerd moet worden voordat er acties op kunnen gebeuren.



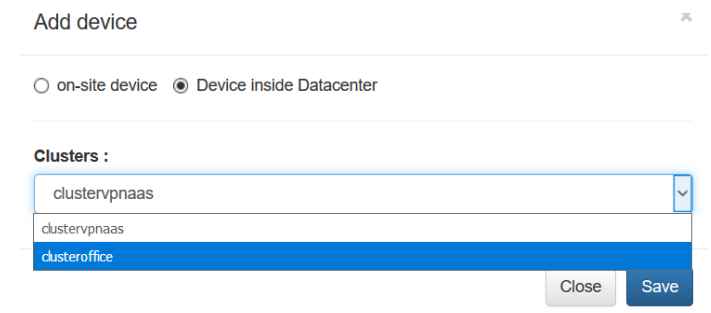
22 Devices list dashboard

Een nieuw apparaat biedt 3 mogelijkheden waarvan het vergrootglas staat voor het bekijken van dat specifiek apparaat. Hier kunnen de gegevens van het apparaat bekenen worden. Acties kunnen nog niet uitgevoerd worden op dit apparaat omdat het nog niet geaccepteerd is. Daarnaast staat de X voor het verwijderen van het apparaat. Indien dit een apparaat is dat we niet kennen of niet willen beheren kan deze uit de lijst verwijdert worden. Als laatste hebben we het vinkje, dit staat voor accepteren. Wanneer we dit apparaat kennen en we dit willen beheren, dan kiezen we voor accepteren waarbij we een keuze venster krijgen. In dit venster is er de keuze wat voor apparaat dit effectief is. Is dit een apparaat dat on-site bij een klant staat, of is dit een apparaat in het datacenter. Bij een on-site apparaat kunnen we dit aan een klant toevoegen. Want elk apparaat is van een klant.



23 on-site device toevoegen

Als we kiezen voor device inside datacenter krijgen we de keuze welk cluster dit apparaat moet zitten. Dit is nodig zodat we weten wanneer we config pushen naar de apparaten dat dit naar ieder apparaat in het cluster gepusht wordt zodat we redundancy hebben.



24 device binnen datacenter toevoegen

3.5.4 Klanten

Bij customers hebben we 2 pagina's. De eerste is een overzicht van al de klanten die Cegeka beheerd. Hierbij zijn ook verschillende opties mogelijk , waarvan het bekijken van de apparaten van de klant, het verwijderen van de klant en het aanpassen van de gegevens van de klant. Daarnaast is de 2de pagina het toevoegen van een klant. Hierbij wordt er gevraagd naar de naam van de klant.

25 Klantenlijst dashboard

Als we de klant gaan bekijken dan bekomen we deze pagina. Hierbij zijn de gegevens van de klant zelf zichtbaar aan de linkerkant en is en een tabel met al de sites van de klant die Cegeka beheren.

26 Klant informatie dashboard

3.5.5 Cluster

Doordat we met redundancy werken worden al de apparaten in het datacenter in een cluster verwerkt. Het cluster gedeelte bestaat ook uit 2 pagina's, waarvan één het toevoegen van een cluster is. Hierbij wordt er gevraagd naar de naam van het cluster. Als 2de pagina krijgen we een lijst van al de bestaande clusters binnen het datacenter.

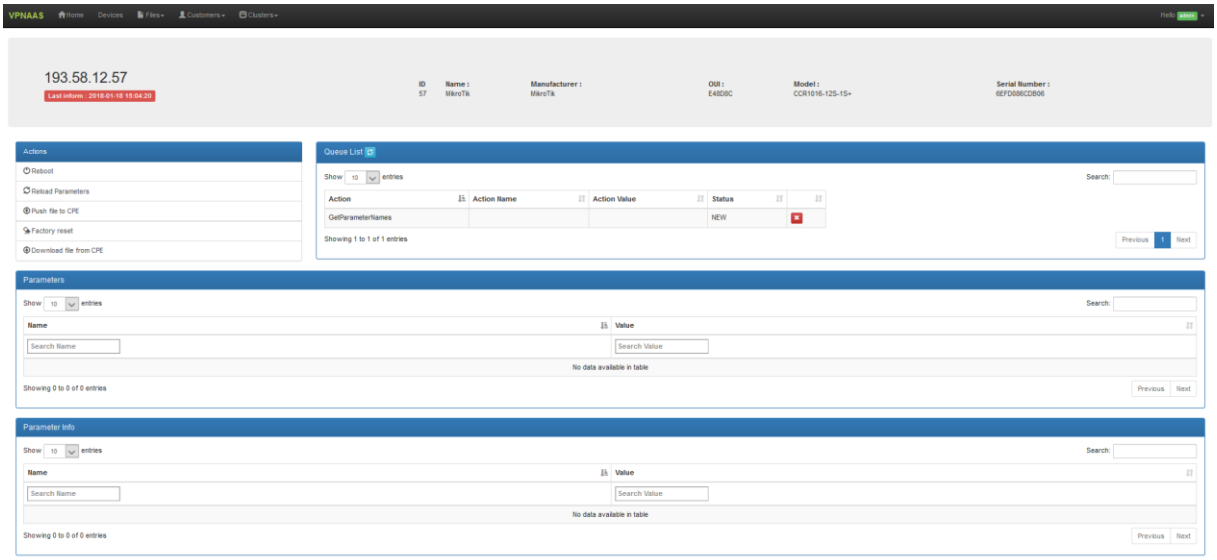
27 Clusterlijst dashboard

Bij het bekijken van een cluster krijgen we ongeveer dezelfde layout als bij de customers. Aan de linkerkant de informatie over het cluster en daarnaast een tabel met al de apparaten binnen het cluster.

28 Cluster informatie dashboard

3.5.6 Device info

De meest belangrijke pagina van het heel dashboard is deze pagina. Hier wordt al de informatie van het apparaat getoont en vanaf hier gebeuren ook al de acties.



29 Device informatie dashbaord

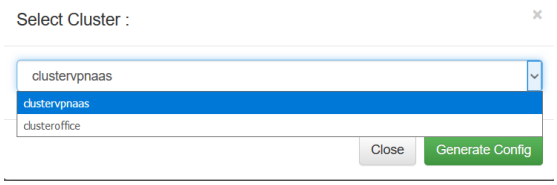
Bovenaan deze pagina bevindt zicht een balk met basis informatie over apparaat met daarbij een groene of rode datum met daarin de laatste keer dat een apparaat een inform heeft gedaan naar de server. Indien deze inform langer dan 1 minuut geleden is dan zal de datum in het rood verschijnen waardoor we weten dat dit apparaat geen connectie meer heeft met de server.

30 Standaard informatie van een apparaat

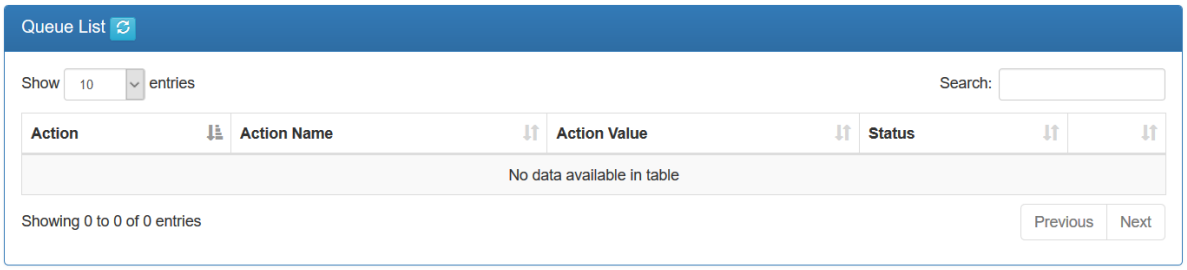
31 Acties die mogelijk zijn op een apparaat

Dit zijn al de acties die mogelijk zijn op het apparaat. Hierin staat reboot voor het opnieuw opstarten van het apparaat, reload parameters voor het vernieuwen van de parameters. Ook kan reload parameters gebruikt worden om de parameters op te vragen als er nog geen parameters zijn. Push file to CPE laat ons bestanden naar het apparaat pushen , hierbij sturen wij een soort van downloadlink naar de CPE waar hij het bestand van kan downloaden. Factory reset zal de default config opnieuw installeren. Dit is de default config die Cegeka er standaard op heeft gezet en dus niet de RouterOS factory config. Download file from CPE zal bestanden van de CPE naar de server downloaden.

Als laatste is er de mogelijkheid de configuratie voor een tunnel op te bouwen te generen. Wanneer we deze actie kiezen komen we op een nieuw venster waar we moeten kiezen naar welk cluster de tunnels opgebouwd moeten worden. Wanneer het cluster gekozen is worden de config files gegenereerd en toegoevoegd aan de queue van ieder apparaat in het cluster en aan het apparaat waarvan deze functie aangeroepen is. Ook worden de files toegevoegd in de file database zodat deze opnieuw naar het apparaat kunnen gepusht worden wanneer dat nodig is.



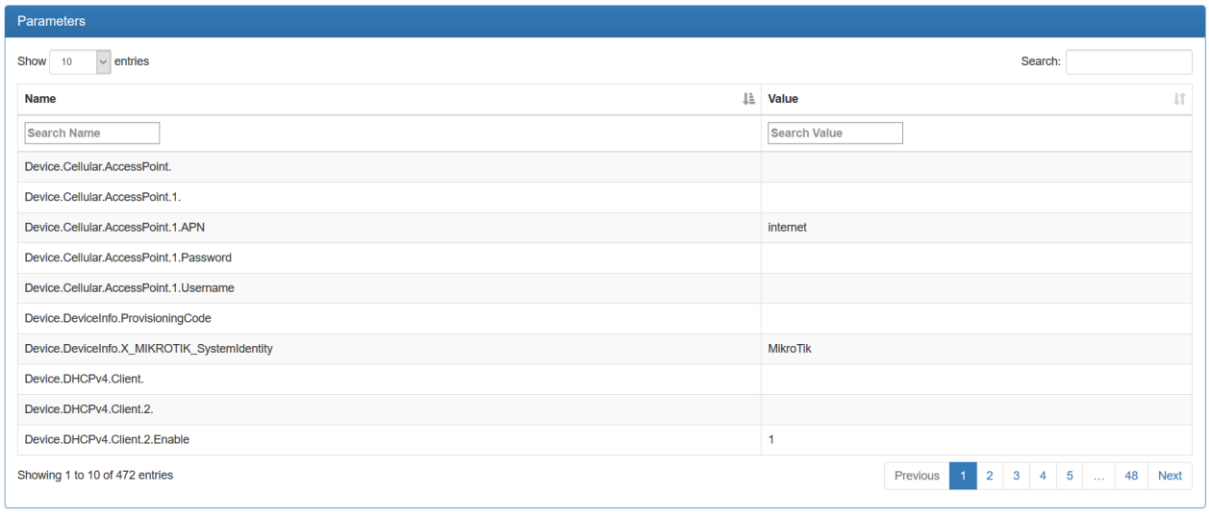
32 Cluster selecteren bij config generate



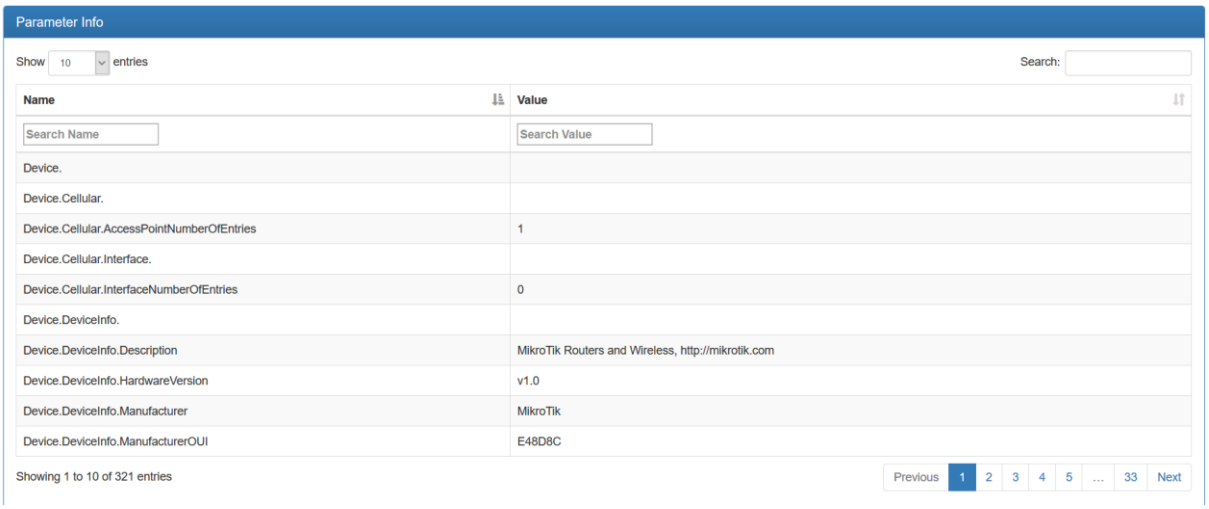
33 Queue list per apparaat

In de queue-list komen al de acties die aangevraagd zijn voor het desbetreffend apparaat. Iedere actie heeft een status, NEW als de actie pas is toegoevoegd en de CPE nog niet aan deze actie zit. SEND als er een inform is gebeurd en het apparaat de actie uit de queue-list heeft opgenomen. FAULT als ern een faultresponse is teruggekomen waardoor de actie niet volledig is uitgevoerd. Wanneer de actie succesvol is uitgevoerd wordt deze uit de queue-list verwijderd.

Als we een apparaat dieper bekijken krijgen we 2 apparte tabellen. Deze tabellen bevatten al de parameters van een apparaat. De eerste tabel heeft al de parameters die aanpasbaar zijn. Door op een parameter te klikken kan de value van deze parameter aangepast worden en wordt er een setparameter naar de queue-list gestuurd zodat deze parameter aangepast kan worden op het apparaat. De tweede tabel bevat al de parameters die beveiligd zijn en dus ook niet aanpasbaar, dit zijn parameters zoals de fabrikant.



34 Aanpasbare parameters van een device

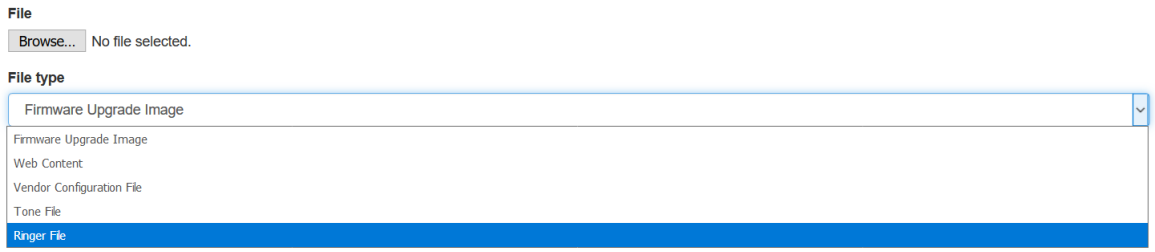


35 Locked parameters van een device

3.5.7 Files

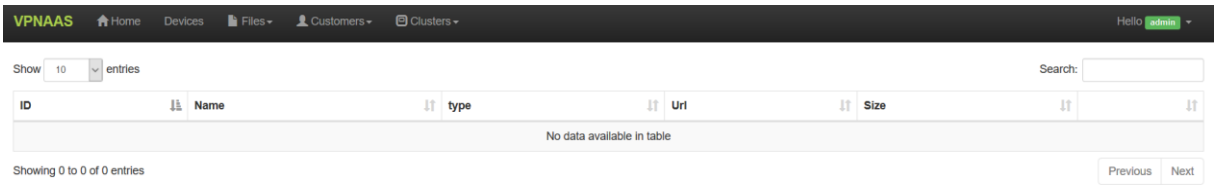
TR-069 maakt het mogelijk bestanden naar de client te pushen. Hiervoor moet er een link meegegeven worden van waar de client deze file moet halen. Via het dashboard kan er een file geüpload worden en wordt deze file op de server bijgehouden.

Bij het files gedeelte op het dasboard bestaat deze uit 2 pagina's. Waarvan de eerste het toevoegen van een bestand is. Hier moet er een bestand meegegeven worden en het type bestand moet geselecteerd worden. Dit is nodig omdat een mikrotik verschillend reageerd op deze file types.



36 Uploaden van een bestand naar de server

Eenmaal de file toegevoegd wordt deze in de lijst van files getoond. In deze lijst is het type, de url en de grootte van het bestand zichtbaar. De link is een publieke link waardoor de client weet dat hij via die link aan het bestand kan.



37 Lijst van verschillende bestanden

II. Onderzoekstopic

1 Vraagstelling

Vergelijking tussen VPN Protocollen: Wat is het beste VPN-protocol wat betreft security en snelheid.

In dit onderzoek zal ik onderzoeken welk van de aangeboden VPN protocollen het best past voor Cegeka tussen 2 Mikrotik Routers.

2 Methode van onderzoek

2.1 Aanpak

VPN of te wel Virtual Private Network zijn verbindingen die binnen Cegeka niet onbekend zijn.

Dagelijks wordt hier gebruik gemaakt van VPN-services. Door gebruik te maken van Mikrotiks komen er nog meer VPN-mogelijkheden bij kijken. En door dat VPN-services zo vaak gebruikt worden binnen Cegeka willen we ervan uitgaan dat we de meest veilige gebruiken. Hiervoor gaan we gebruik maken van gerenommeerde bronnen. Toch zijn gerenommeerde bronnen niet genoeg om volledig overtuigd te zijn wat de beste VPN-oplossing is wat betreft security en snelheid, daarom zal iedere mogelijke VPN-service getest worden. Dit gaan we doen door middel een fysiek labo op te stellen waarbij we 2 sites maken. Hier zal iedere VPN-service opgesteld en geconfigureerd worden.

Eenmaal geconfigureerd zullen hier metingen op gedaan worden zodat we effectieve resultaten hebben om met elkaar te vergelijken. Door deze vergelijking krijgen we een beter beeld over welke VPN-services het meest betrouwbaar zijn op vlak van snelheid en security. De metingen gaan we doen aan de hand van WireShark en een monitoringtool. WireShark gaan we gebruiken om de pakketten te onderscheppen tijdens hun doorstroom in een tunnel. Deze pakketten kunnen we dan gaan ontleden en de effectieve encryptie hiervan bekijken. We zullen nooit precies kunnen zeggen welke encryptie het meest veilige is door pakketten te gaan onderscheppen in WireShark, hiervoor kijken we terug naar de gerenommeerde bronnen. Voor de snelheid gaan we gebruik maken van een standaard geinstalleerd programma genaamd Bandwidth tester. Met dit programma kunnen we metingen uitvoeren over TCP/UDP. Naast de bandbreedte en de encryptie gaan we ook kijken hoe de Mikrotiks reageren op deze doorstroom van pakketten. Door dat er encryptie gaat gebeuren gaan we bekijken hoe de CPU hierop gaat reageren.

Eenmaal we van elk mogelijke VPN een opstelling gemaakt hebben, en hier tal van metingen op hebben gedaan kunnen we deze gaan vergelijken. Met deze effectieve bewijzen kunnen we dan een besluit maken welk van de opgestelde VPN services het best past.

3 Resultaten

3.1 Literatuurstudie

In de Literatuurstudie ga ik een korte omschrijving doen van elke mogelijke VPN-service die geconfigureerd kan worden op Mikrotik routers.

3.1.1 De verschillende VPN oplossingen

3.1.1.1 IPIP (IP-in-IP pakket)

38 IPIP header

Bij IP-in-IP gaan we 1 IP-pakket in een ander IP-pakket steken en dit pakket zo versturen door de tunnel tussen de 2 apparaten. Als het pakket verstuurd wordt, wordt er aan het begin van de tunnel een nieuwe IP-header toegevoegd aan het pakket waardoor de rest als een soort van payload gezien kan worden. Met de nieuwe IP-header weet het pakket waar het einde van de tunnel zich bevindt en waar het pakket naartoe gestuurd moet worden. Eenmaal het pakket aan het einde van de tunnel wordt de payload terug uitgepakt, de buitenste IP-header verwijdert en komt de originele IP-header terug. Het pakket kan nu naar de effectieve bestemming verstuurd worden. IPIP kan ook gebruik maken van IPsec secret, waardoor de data over de IPIP-tunnel geëncrypteerd is.

Als we een ping gaan onderbreken met Wireshark zien we dat als we verkeer door een IPIP-tunnel sturen we een nieuwe IP-laag bovenop onze IP-header krijgen. Hierdoor krijgt ons pakket een nieuwe source en destination.

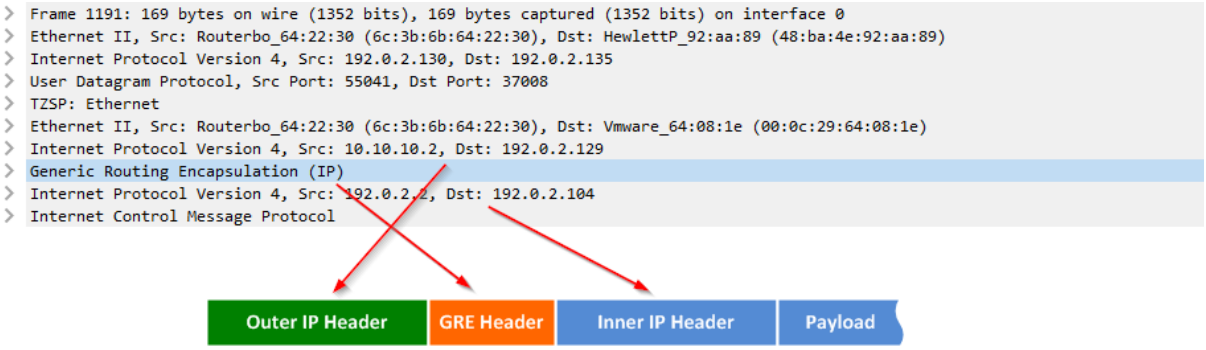
```
> Ethernet II, Src: Routerbo_0a:47:5e (e4:8d:8c:0a:47:5e), Dst: Routerbo_64:22:33 (6c:3b:6b:64:22:33)
> Internet Protocol Version 4, Src: 10.10.10.2, Dst: 192.0.2.129
> Internet Protocol Version 4, Src: 192.0.2.2, Dst: 192.0.2.104
> Internet Control Message Protocol
```

39 IPIP wireshark capture

3.1.1.2 Generic routing encapsulation

Generic routing Encapsulation is een tunnel protocol dat ontworpen is door CISCO en werkt met IP-protocolnummer 47. Het staat bekend voor zijn simpliciteit en eenvoudige configuratie. Ook staat GRE bekend voor het ondersteunen van broadcast en multicast protocollen wat zich onderscheidt van andere VPN-toepassingen. Doordat GRE multicast en broadcast support wordt deze tunnel gezien als eerste keuze als het aankomt op het versturen van Layer 3 protocollen over een tunnel. Buiten het feit dat GRE makkelijk opgezet kan worden heeft het nog andere eigenschappen, zoals de mogelijkheid om meerdere subnets te routeren zonder meerdere tunnels te hebben, ook maakt een GRE-tunnel interfaces aan die routeerbaar zijn waardoor routing protocollen kunnen functioneren over deze tunnel.

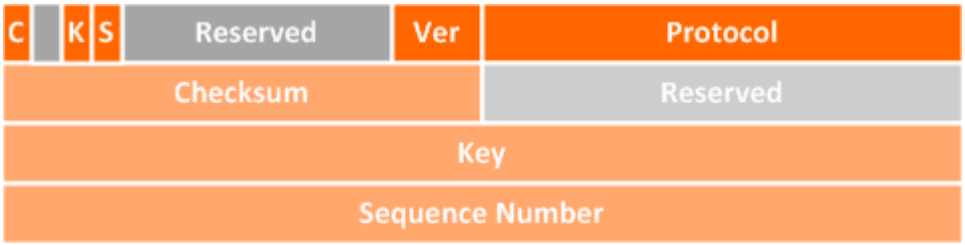
GRE maakt point-to-point connecties zoals dat van een VPN maar het verschil is dat GRE alleen een pakket in een pakket gaat steken terwijl VPN een encryptie gaat toepassen, hoewel hier ook oplossingen voor zijn zodat GRE-pakketten geëncrypteerd zijn.



40 GRE wireshark capture

Als we een GRE-pakket schematisch gaan bekijken zien we dat ons pakket nieuwe lagen heeft bijgekregen. De “Outer IP-header” is de nieuwe header waar we de nieuwe source en destination van het pakketje gaan vinden. Dit zijn IP-adressen in de range van de GRE-tunnel.

Als we dieper ingaan op de “GRE Header” kunnen we zien wat er effectief in de GRE-header zit als informatie.



41 GRE header

- C (Bit 0)
  - C staat voor de checksum of te wel het controlegetal. Als deze bit op 1 staat zal het controlegetal aanwezig zijn en geldige informatie bevatten. De checksum wordt gebruikt om de echtheid van de GRE-header en de payload te garanderen. Het bevat een IP checksum van de GRE-header en het payload pakket.
- K (Bit 2)
  - K staat voor Key, indien de Key bit op 1 staat zal het veld met de sleutel aanwezig zijn in de GRE-header. Het sleutel veld wordt gebruikt voor de authenticatie van de GRE-pakketten. Deze sleutel vermijdt verkeerde configuratie of injecties van pakketten van verkeerde bronnen. De twee uiteindes van de GRE-tunnel accepteren alleen GRE pakektten met de juiste sleutel in de header. Deze sleutel moet manueel geconfigureerd worden op beide uiteindes.

- S (Bit 3)
    - S staat voor Sequence nummer. Beide eindpunten van de tunnel gebruiken de sequence nummer om de volgorde te traceren in hoe pakketten binnekomen en indien nodig pakketten droppen dat niet in de juiste volgorde zijn binnengekomen.
- Ver (Bit 13-15)
    - Ver staat voor versie nummer

3.1.1.2.1 GRE secret

Hoewel GRE beschreven wordt als onveilig doordat er geen encryptie gebeurt, is er sinds ROB v6.30 een nieuwe functie wat het mogelijk maakt te werken met een IPsec secret op een GRE-tunnel. Bij het aanmaken van een GRE-interface kunnen we een secret meegeven waardoor onze data toch geëncrypteerd is.

Bij het ingeven van een IPsec secret aan beide kanten van de tunnel gaat er aan Dynamic IPsec gedaan worden. Dit heeft zowel voordelen als nadelen. Wat de voordelen betreft is dit eenvoudig te configureren sinds er alleen maar een secret moet meegegeven worden en de rest wordt automatisch aangemaakt. Ook zorgt dit voor een bescherming van het verkeer door de GRE tunnel , hoewel deze ook niet volledig veilig is , wat ons bij de nadelen brengt van Dynamic IPsec.

Door dat alles automatisch ingesteld wordt is fase 1, de IPsec peer configuratie, niet maxium beveiligd doordat de fase 1 geconfigureerd wordt met SHA1 – 3des/AES128 algoritmes. Dit kan alleen aangepast worden als er in de plaats van met een IPsec secret te werken, manueel IPsec geconfigureerd wordt. Hoewel fase 1 niet aangepast kan worden , kan bij fase 2 wel gekozen worden welke algoritmes er gebruikt worden voor het encrypteren en authenticeren.

3.1.1.3 EoIP (Ethernet over IP)

Wanneer we over EoIP praten, duiken we direct naar een Layer2. Vaak wordt EoIP gebruikt indien er 2 netwerken zijn waar het layer 2 gedeelte moet verlengd worden tussen beide sites. EoIP wordt ongeveer hetzelfde geconfigureerd als GRE en zal het OSI layer 2 broadcast domain verlengen. Eenmaal deze connectie geconfigureerd is komen de interfaces tevoorschijn en staan deze tussen de interface list alsof het fysieke interfaces zijn. Doordat EoIP werkt met interfaces kan er een bridge geconfigureerd worden tussen de tunnel en eender welke fysieke interface. Daardoor lijkt het alsof er een fysieke kabel tussen de 2 apparaten zit. Een EoIP tunnel kan ook nog geconfigureerd worden met andere tunnels, zoals een PPTP-tunnel, een IPIP-tunnel of eender welke tunnel dat de mogelijkheid heeft om IP te transporteren. Wat veiligheid betreft is een EoIP tunnel standaard niet veilig. Deze is niet geëncrypteerd en al het verkeer wat onderschept wordt kan gelezen worden. Hiervoor kunnen we gebruik maken van het IPsec secret, waardoor er dynamisch IPsec wordt opgebouwd en de tunnel geëncrypteerd is. Indien deze encryptie niet genoeg is kan er nog altijd een IPsec tunnel geconfigureerd worden in plaats van het secret. Waardoor er meer opties mogelijk zijn wat betreft configuratie.

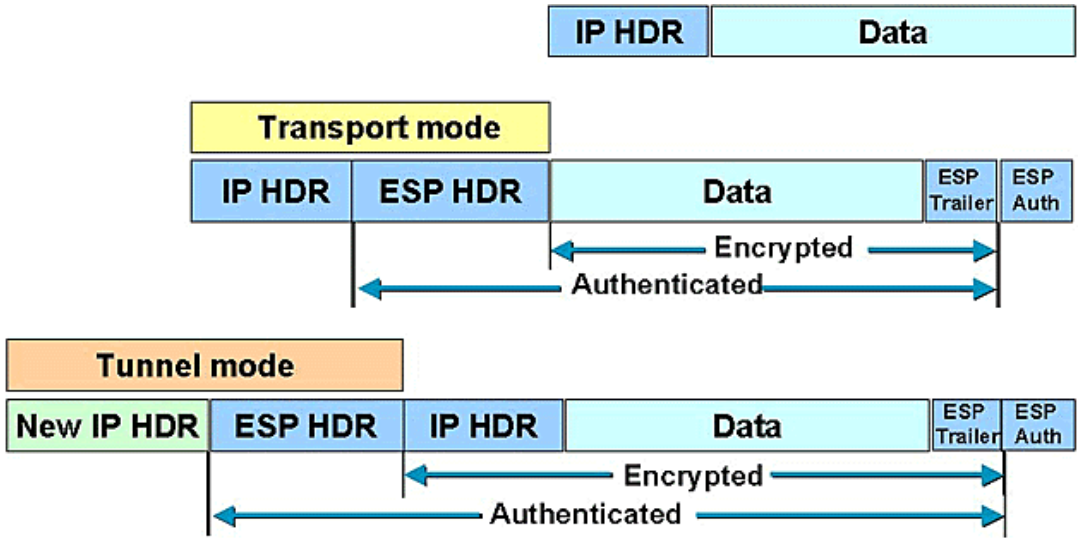
3.1.1.4 IPSEC (IP Security)

IPsec staat voor Internet protocol security of te wel IP security. IPsec is een protocol dat IP-verkeer encrypteerd vooraleer de pakketten getransporteerd zijn van source node naar destination node. IPsec is ook in staat authenticatie te doen tussen de 2 nodes vooraleer de effectieve communicatie plaatsvindt. Het kan geconfigureerd worden om eender welk beschikbaar algoritme te gebruiken voor het encrypteren en decrypteren van netwerkverkeer. Dit zijn de functionaliteiten van IPsec:

- Confidentiality: Door al de data te encrypteren zal het niet mogelijk zijn de verstuurde data te lezen door iemand buiten de zender of de ontvanger.
- Integrity: Dit is nog belangrijker dan Confidentiality. Dit houdt in dat de data die verstuurd wordt niet aangepast kan worden op weg naar zijn bestemming door het berekenen van een hash waarde. Door deze hash waarde kunnen de zender en ontvanger nakijken of er aanpassingen zijn gedaan in het pakket.
- Authentication: De zender en ontvanger zullen authenticatie uitvoeren tussen elkaar zodat ze zeker zijn dat ze praten met het apparaat waar ze effectief verbinding mee wouden maken.
- Anti-replay: Zelfs al is data geëncrypteerd en geauthenticeerd, kan een tussenpersoon deze pakketten opnieuw proberen te verzenden. Anti replay werkt met sequence nummers, indien een nummer te oud is of een nummer niet in de range ligt wordt het pakket afgewezen.

IPsec kan gebruikt worden in 2 modes:

- Transport mode: Bij transport mode gaat alleen de payload van het pakket geëncrypteerd worden.
- Tunnel mode: dit is de default mode. In de tunnel mode is het volledig IP-pakket beveiligd (geëncrypteerd en/of geverifieerd). Er wordt een nieuw IP-header pakket toegevoegd aan het originele pakket met de source, destination en endpoints in.



42 IPsec tunnel mode - Transport mode

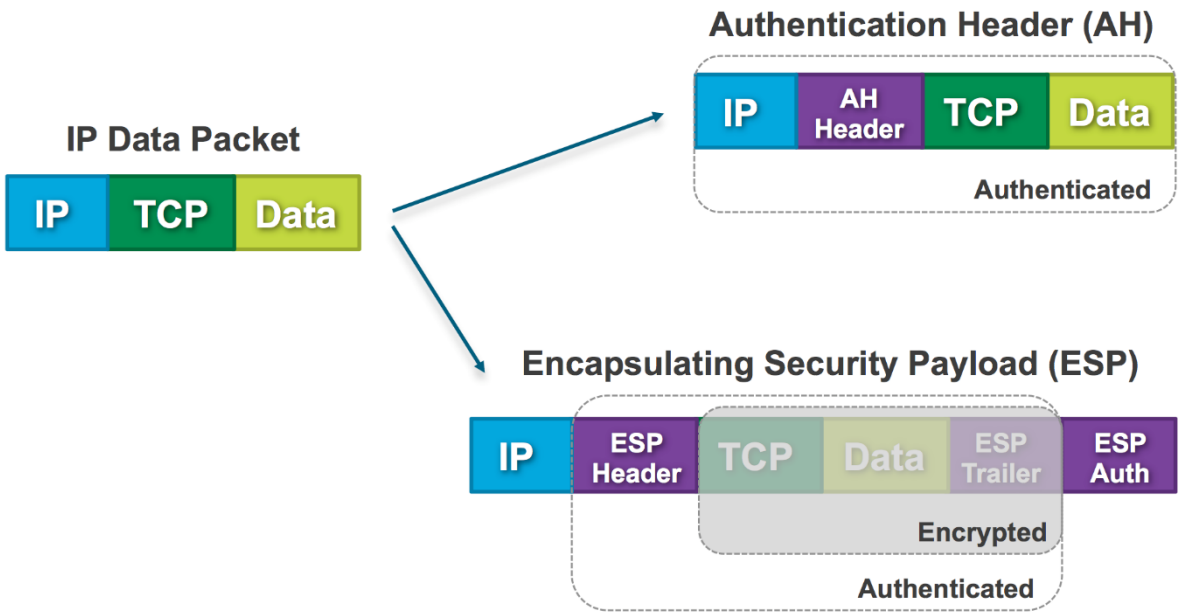
Ipsec protocol kunnen we opdelen in de volgende groepen :

3.1.1.4.1 Authentication Header (AH)

AH is een deel van het IPsec protocol suite. Het biedt authenticatie, echtheid en anti-replay voor het volledige pakket (IP-header en de payload). Er is hier geen sprake van data encryptie. De data is leesbaar, maar beschermd tegen aanpassingen. AH maakt gebruik van HMAC-algoritmes om pakketten te ondertekenen. HMAC staat voor Hash-based message authentication code.

3.1.1.4.2 Encapsulating Security Payload (ESP)

ESP is een transport layer security protocol gemaakt om te functioneren met zowel IPv4 als IPv6 protocollen. Het neemt de vorm aan van een header dat na de IP-header wordt gestoken en voor de volgende laag, zoals TCP, UDP, ICMP. ESP zelf biedt data betrouwbaarheid (encryptie) en authenticatie aan. Het kan gebruikt worden alleen met Encryptie, alleen met authenticatie, of beide. Wanneer ESP-authenticatie functies gebruikt worden gaan deze dezelfde algoritmes gebruiken als AH, maar het verschil tussen de 2 is dat authenticatie via AH het volledige IP-pakket verifieert, en dus ook de buitenste IP-header, terwijl bij ESP-authenticatie alleen het IP-diagram gedeelte van het IP-pakket wordt geverifieerd.



43 IPSec ESP

1.1.1.1.1 Internet Key Exchange (IKE)

IKE is een protocol dat gebruikt wordt bij SAs (Security assosiations) in het IPsec protocol suite. Door middel van IKE kunnen apparaten informatie uitwisselen wat nodig is om beveiligde communicatie te voeren. Hierin wordt gebruik gemaakt van cryptografische sleutels die gebruikt worden bij het versleutelen van informatie tijdens de authenticatie en de payload encryptie.

IKE gaat toestellen toelaten om security associations uit te wisselen en zo hun security association database te vullen. Deze database wordt gebruikt bij de effectieve uitwisseling van datagrammen met de AH en ESP-protocollen.

3.1.1.4.3 Hardware encryption

Hardware encryptie zorgt ervoor dat de encrypties die gedaan worden niet allemaal in de software gebeuren, wat zorgt voor minder belasting van de CPU. Dit is mogelijk doordat er een encryption engine binnenin de CPU zit waardoor bepaalde algoritmes kunnen verwerkt worden binnen de hardware. Momenteel kan er alleen met 128,192,256-bit AES-cbc en SHA1/SHA256 op hardware niveau geëncrypteerd worden.

In mijn labo opstelling beschik ik over Mikrotiks die de mogelijkheid hebben om hardware encryptie te gaan doen. Hoewel deze Mikrotiks niet diegene zijn waar de tunnel mee verbonden is, dus gaat mijn POC niet gebruik maken van hardware encryptie.

3.1.1.4.4 Mikrotik Algoritmes

- Authenticatie algoritmes

  Als we naar authenticatie gaan kijken bij IPsec geeft Mikrotik ons een aantal opties: MD5, SHA1, SHA256, SHA512. MD5 gaan we niet kunnen gebruiken doordat MD5 al gekraakt is sinds 1996 en deze dus als onveilig is verklaard.

  Als we bij de SHA (Secure hash algorithm) familie gaan kijken kunnen we ook SHA1 al van de lijst schrappen sinds SHA1 ook gekraakt is door Google. Bij SHA-256 en SHA-512 is er wat betreft beveiliging niet echt een verschil sinds SHA-256 nog niet gekraakt is en SHA-512 ook niet. Het verschil wat wij zullen merken tussen SHA-256 en SHA-512 is de snelheid. SHA-256 zal sneller werken indien er CPU beperkingen zijn.

7.
- Encryptie algoritmes
8.
    - DES (Data Encryption Standard)
9. Des was een van de eerste symmetrische blok encrypties, het werd gemaakt door IBM en daarna werd het bij de U.S als standaard encryptie algoritme geïmplementeerd.
10. Des is een van de oudere encryptie standaarden waardoor het op de dag van vandaag niet meer gebruikt wordt. Niet alleen omdat dit al een ouder standaard is maar ook omdat het al meer dan 10 jaar geleden meerdere malen effectief gekraakt is geweest. Het kraken van DES was mogelijk doordat DES gebruikt maakt van een korte sleutel die namelijk 56 bits groot is.
11.
    - 3DES
12. 3DES is zoals de naam het zegt 3 keer DES, wat dus ook betekend 3 keer grotere sleutel. De eindsleutel van 3DES is 168 bits groot en er wordt gebruikt gemaakt van meerdere sleutels. 3DES werd oorspronkelijk gemaakt voor efficiënte hardware implementatie, waardoor het minder softwarematig presteert. Het grote nadeel van 3DES is dat het traag is.
13.
    - AES (ADVANCED ENCRYPTION STANDARD)
14. AES wordt gezien als een standaard symmetrisch encryptie algoritme dat gebaseerd is op het Feistel network. AES is een block algoritme dat gebruik maakt van 128 bit blokken en gevarieerde sleutels van 128,192 en 256 bits. Door gebruik te maken van deze gevarieerde sleutels beschikt AES over een sterke beveiliging.
    - BLOWFISH

15. Blowfish is een blok encryptie algoritme gebaseerd op feistel functies die gebruik maken van 64 bit blokken en sleutels tussen de 32-448 bit. Door gebruik te maken van wisselende lengtes van een sleutel is er meer tijd nodig wat betreft CPU.
16.
    - CAMELLIA (cipher)
17. Camellia is een symmetrisch blok algoritme met blok groottes van 128 bits en sleutels van 128,192 of 256 bits. Het was gemaakt door Mitsubishi en wordt vaak vergeleken met AES wat betreft snelheid en security. Toch komt Camellia niet vaak voor in de werkelijkheid.

3.1.1.5 OVPN(OpenVPN)

OpenVPN is een van de nieuwere Technologieën, buiten dit is OpenVPN ook open source. OpenVPN maakt gebruik van de protocollen SSLv3/TLSv1 en de OpenSSL library, samen met een combinatie van andere technologieen. Het protocol is hoogst configureerbaar en werkt via een UDP-poort. Deze poort kan ook aangepast worden zodat OpenVPN via een andere poort communiceert waardoor het moeilijker geblokkeerd kan worden.

Doordat OpenVPN gebruik maakt van OpenSSL zijn er verschillende cryptografische algoritmen wat OpenVPN ondersteunt, 3DES, AES, Camelia, Blowfish, CAST-128 en meer waardoor OVPN als veilig wordt gezien. Wanneer het gaat over versleuteling is AES de nieuwste beschikbare technologie en wordt deze als standaard beschouwd bij OVPN. Dat is simpelweg omdat het geen bekende zwakheden heeft, het is zelfs goedgekeurd door de Amerikaanse overheid en de agentschappen van de VS om ‘gevoelige’ gegevens te beschermen.

3.1.1.6 PPTP (Point-to-point Tunneling protocol)

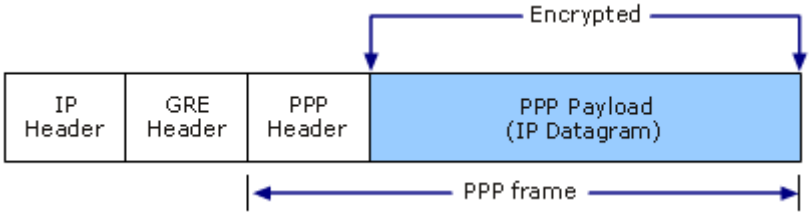
PPTP is een beveiligde tunnel voor het transporteren van IP-verkeer. PPTP is een van de oudste tunneling protocols wat nog gebruikt wordt op de dag van vandaag. Het is ontwikkeld op initiatief van Microsoft om een ander protocol genaamd PPP in te kapselen. Als we kijken naar de authenticatie van PPTP zien we dat deze overeenstemt met de authenticatie van PPP. Hoewel PPTP een van de oudere protocollen is, is het een protocol dat toch nog regelmatig voorkomt, dit omdat dit zeer makkelijk op te zetten is en de snelheid nog altijd zeer hoog ligt. Voor deze reden wordt PPTP vaak bij audio- en videostreaming gebruikt. Maar deze snelheid moet ook een reden hebben, de reden hiervan is dat dit protocol niet veilig is.

PPTP bestaat uit de volgende onderdelen:

- Een deel dat zorgt voor de authenticatie van de gebruiker (gebruikersnaam en wachtwoord), overgenomen van PPP (MS-CHAP of MS-CHAP v2).
- een deel dat zorgt voor het instellen van de Layer 3-protocollen (TCP/IP, NetBIOS), overgenomen van PPP.
- De eigenlijke tunnel. Deze tunnel is een GRE-tunnel.
- Encryptie: Deze encryptie is MPPE, en kan 40, 56 of 128 bit sterk zijn. MPPE is een Microsoft-protocol.
- Compressie: Deze compressie is MPPC.

Als er een PPTP-verbinding wordt opgezet, dan gebeurt dit over TCP-poort 1723. Deze poort wordt gebruikt voor authenticatie, uitwisselen van gegevens voor de encryptie en dergelijke. De tunnel zelf is zoals gezegd een GRE-tunnel. GRE wordt aangegeven als protocol 47.

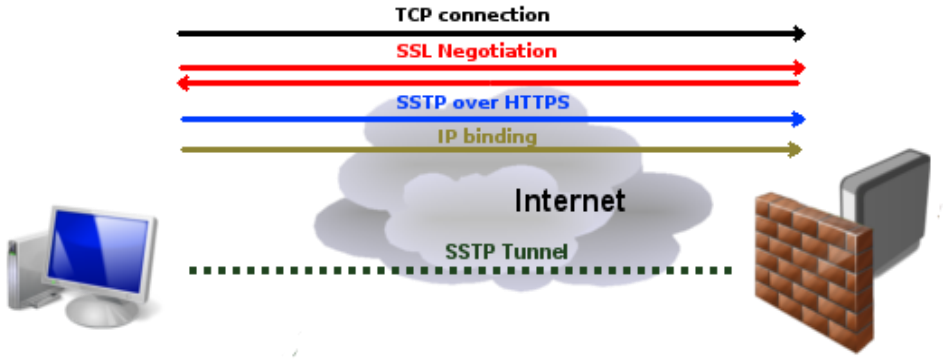
Wat security betreft is PPTP allesbehalve goed beveiligd. PPTP is al vaak het onderwerp geweest van securityonderzoeken omdat er al grote lekken zijn gevonden in dit protocol. Deze lekken waren vooral te linken aan de onderliggende PPP-authenticatie protocollen die gebruikt werden.



44 PPTP header

Wat de encryptie betreft, op de afbeelding is zichtbaar dat de PPP payload encrypted wordt. Dit gebeurt door Microsoft Point-to-point encryptie (MPPE). Dit wordt gedaan door gebruik te maken van encryptie keys die gegenereerd worden van MS-chap v2 of EAP-TLS authenticatie. PPTP maakt gebruik van de onderliggende PPP encrypties en eerdere geëncrypteerde PPP frames.

3.1.1.7 SSTP (Secure socket Tunneling protocol)



45 SSTP

SSTP is een vorm van tunnels dat PPP-verkeer vervoert over een SSL/TLS channel. SSTP biedt transport level security met encryptie en versleuteling Door gebruik te maken van SSL/TLS over TCP-poort 443. Doordat poort 443 gebruikt wordt kan SSTP virtueel door zo goed als elke firewall zonder al te grote problemen. Wanneer we een SSTP-connectie opzetten moet de SSTP-server geverifieerd worden tijdens de SSL/TLS fase. Clients kunnen optioneel geverifieerd worden tijdens de SSL/TLS fase maar moeten geverifieerd worden in de PPP fase.

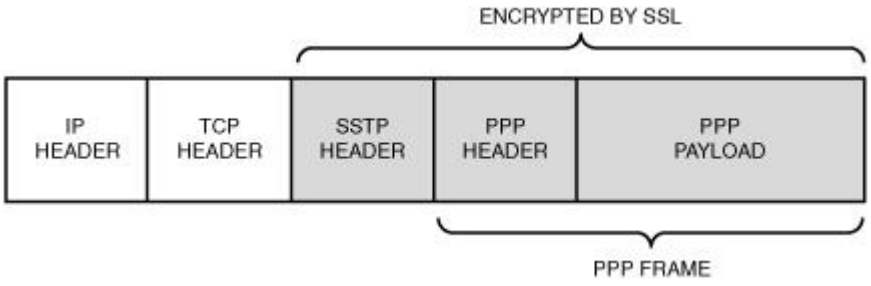
3.1.1.7.1 Wat is SSL/TLS

Transport layer security (TLS) en Secure socket layer(SSL) zijn Encryptie-protocollen die communiceren tussen apparaten. Deze protocollen zijn gebaseerd op public key cryptografie. Hiervan is SSL al een verouderd protocol.

3.1.1.7.2 PFS

PFS of te wel Perfect forward secrecy zorgt ervoor dat bij elk bericht dat er verzonden wordt een andere key wordt gegenereerd. Dit doet hij door middel van DHE (Diffie-Hellman Ephemeral). Elke keer als een nieuwe sessie begint gaat PFS het systeem forceren voor een DHE key exchange. Dit wordt niet gedaan aan de hand van een algoritme, maar in de plaats gaat PFS een volledig willekeurige sleutel aanmaken per sessie. Dit zorgt voor een veilige verbinding tussen 2 apparaten

Hoewel PFS een extra beveiliging laag is komt deze niet vaak voor doordat het ongeveer tussen de 15-27% meer Proccesor power gebruikt.



46 SSTP header

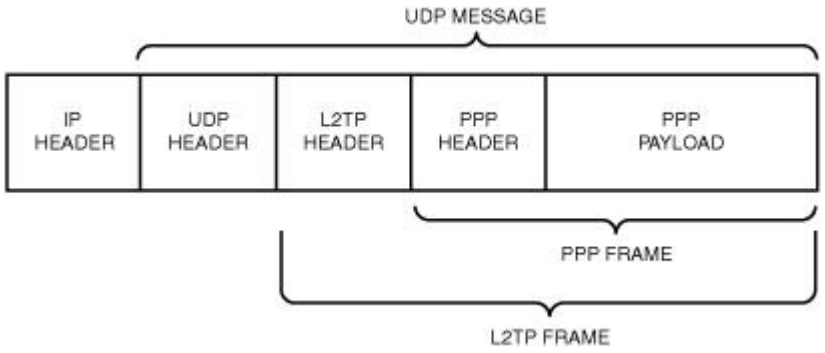
3.1.1.8 L2TP(Layer 2 tunnel protocol)

Layer 2 tunneling protocol is een soort van uitbreiding van Point-to-point tunneling protocol (PPTP). L2TP maakt gebruik van de beste kenmerken van 2 andere tunnels: PPTP van Microsoft en L2F van cisco systems. Het is een tunnel dat IP-verkeer verstuurd door gebruik te maken van PPP. Bij een L2TP tunnel hebben we 2 eindpunten die we LAC (L2TP Access Concentrator) en LNS(L2TP Network Server) genoemd. De L2TP LNS gaat wachten voor een nieuwe tunnel die gemaakt wordt. Eenmaal die tunnel gemaakt is kan er netwerkverkeer tussen de 2 peers stromen.

47 L2TP icon

L2TP kan gebruikt worden zoals eender welk tunneling protocol met of zonder encryptie, hoewel de L2TP standaard zegt dat de veiligste weg om L2TP te gebruiken, de weg is waar we L2TP gaan koppelen met IPsec. IPsec staat bekend voor hun beveiliging en is perfect te gebruiken samen met L2TP waardoor we al de eigenschappen van L2TP samen kunnen gebruiken met IPsec.

L2TP verkeer maakt gebruik, van het UDP protoccol voor zowel controle als datapakketten. Het volledige L2TP pakket, met payload en L2TP header wordt verzonden binnenin een User datagram protocol (UDP). Udp port 1701 is hiervoor gebruikt, dit is voor het opzetten van een link, eenmaal de link is opgezet zal het verkeer eender welke UDP-poort gebruiken om verkeer te versturen, dit kan ook nog op poort 1701 zijn. Dit wilt dus zeggen dat L2TP kan gebruikt worden bij de meeste firewalls en routers door nUDP verkeer te accepteren door de firewall of router.



48 L2TP header

3.2 POC

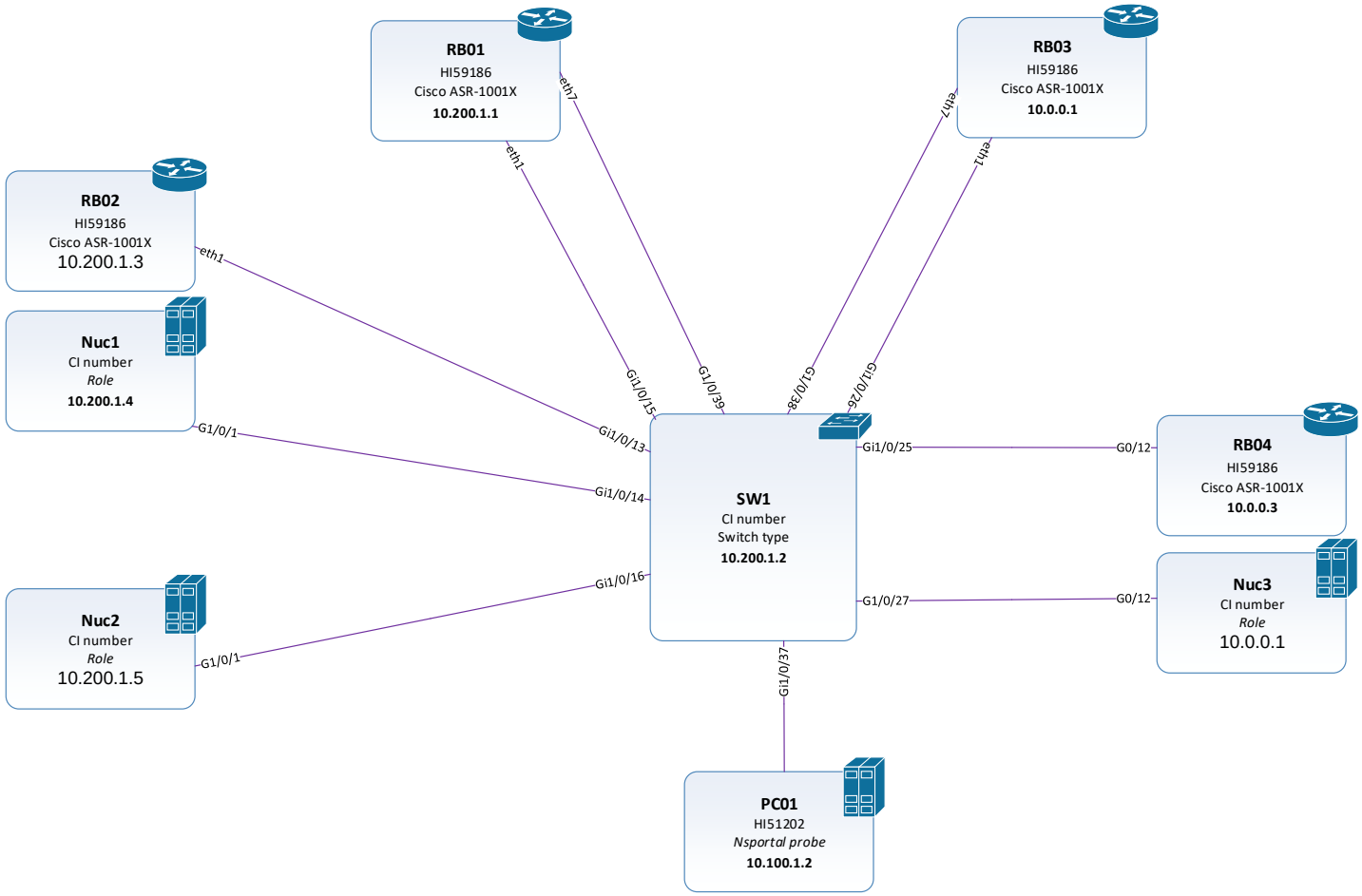
Bij het onderzoek van de verschillende VPN-services kon ik aan de hand van de gerenommeerde bronnen al een zicht krijgen welke VPN-services tegenwoordig veel gebruikt worden, welke niet meer gebruikt worden wegens gebrek aan snelheid en welke al eerder gekraakt zijn geweest en dus niet meer als veilig beschouwd worden. Om deze metingen uit te voeren heb ik de verschillende VPN-tunnels, die ik onderzocht heb, opgebouwd in een lab-opstelling.

Voor dit lab heb ik verschillende apparaten gekregen zodat ik zeker niet te kort kwam voor mijn metingen uit te voeren. Mijn lab bestond uit 4 Cloud Core Routers, 1 Cisco switch en 3 Nuc's.

Op de Nuc's heb ik naderhand Vmware ESXI geïnstalleerd zodat ik met verschillende virtuele machines kon werken. Ook heb ik in het lab gewerkt met VRF's. VRF's zijn verschillende routing tabellen op 1 router. Hierdoor kon ik per klant een routing tabel maken zodat mijn opstelling leek alsof er verschillende klanten aanwezig waren.

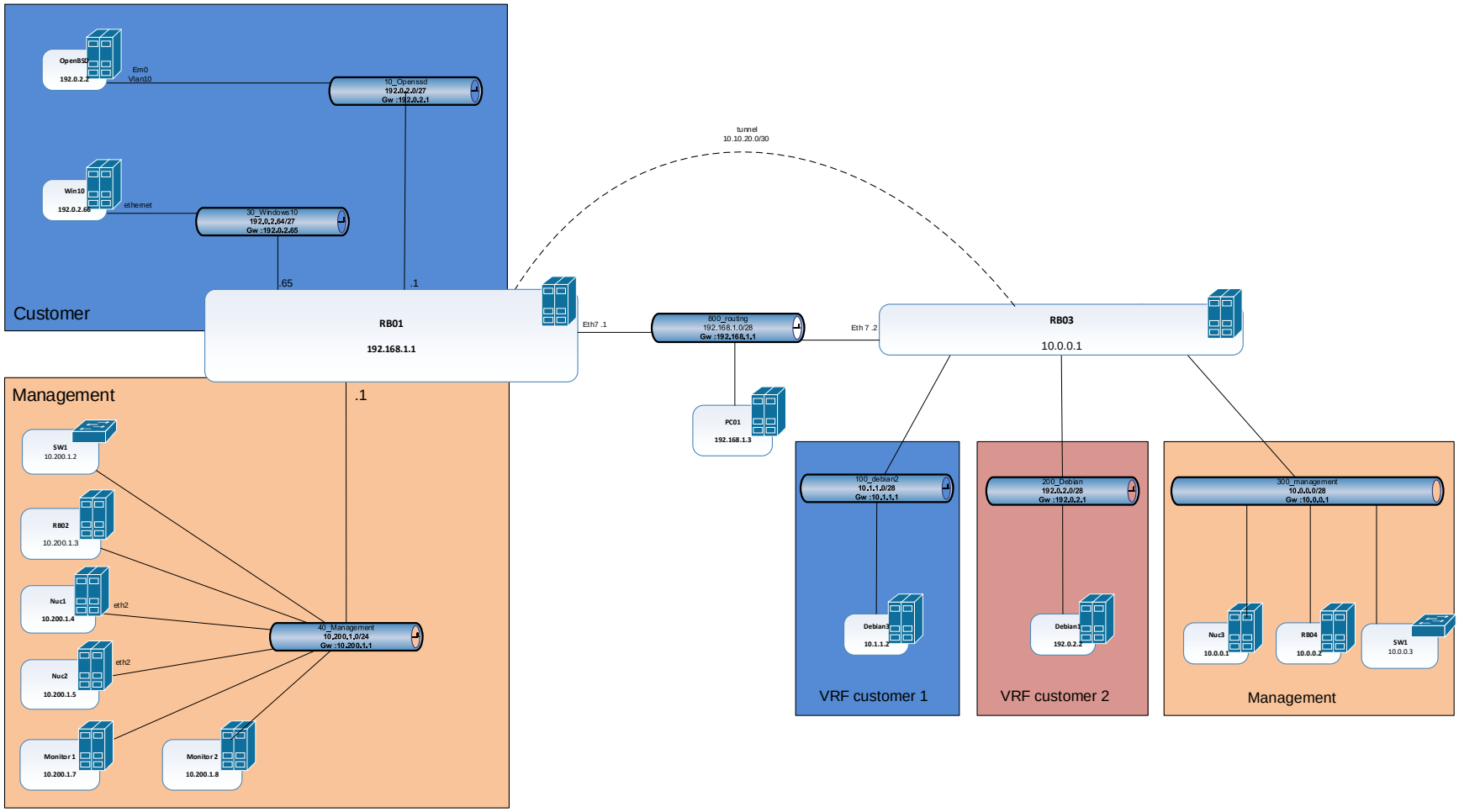
Snelheid hangt af van de hardware apparatuur dat gebruikt wordt in een opstelling, waardoor gerenommeerde bronnen altijd zouden afwijken van de waardes die ik binnen Cegeka zou kunnen verkrijgen. Daarom heb ik iedere VPN-service opgesteld in mijn labo waar ik gebruik maak van hardware apparatuur dat ook binnen Cegeka gebruikt wordt. Bij de opstelling van iedere VPN-service is er een upload en download test gedaan. Als eerste ben ik begonnen met het meten van de bandbreedte zonder tunnel. Zo kreeg ik een zicht van wat de maximumsnelheid is tussen mijn 2 sites en kon ik berekenen hoeveel ik per VPN-tunnel verlies. Deze bandbreedte heb ik gemeten aan de hand van een bandbreedte tester die standaard geïnstalleerd is op RouterOS. Wanneer we deze test zouden gebruiken op het apparaat waar de metingen gingen doen, zouden deze waardes niet eerlijk zijn doordat er data gegenereerd wordt en verstuurd. Dit zou de waardes van de metingen beïnvloeden en dat is wat we niet willen. Daarom heb ik in het lab 4 Cloud Core Routers zitten zodat ik er 2 had voor de tunnel tussen op te bouwen, en achter elke Cloud Core Router nog 1 extra apparaat zodat we vanaf dat apparaat de test konden uitvoeren. Het maakt niet uit wat de CPU doet op dat apparaat , sinds we de metingen zullen doen op de 2 apparaten waar de tunnel tussen geconfigureerd is.

3.2.1 Layer 2 tekening



49 Layer 2 tekening labo opstelling

3.2.2 Layer 3 tekening

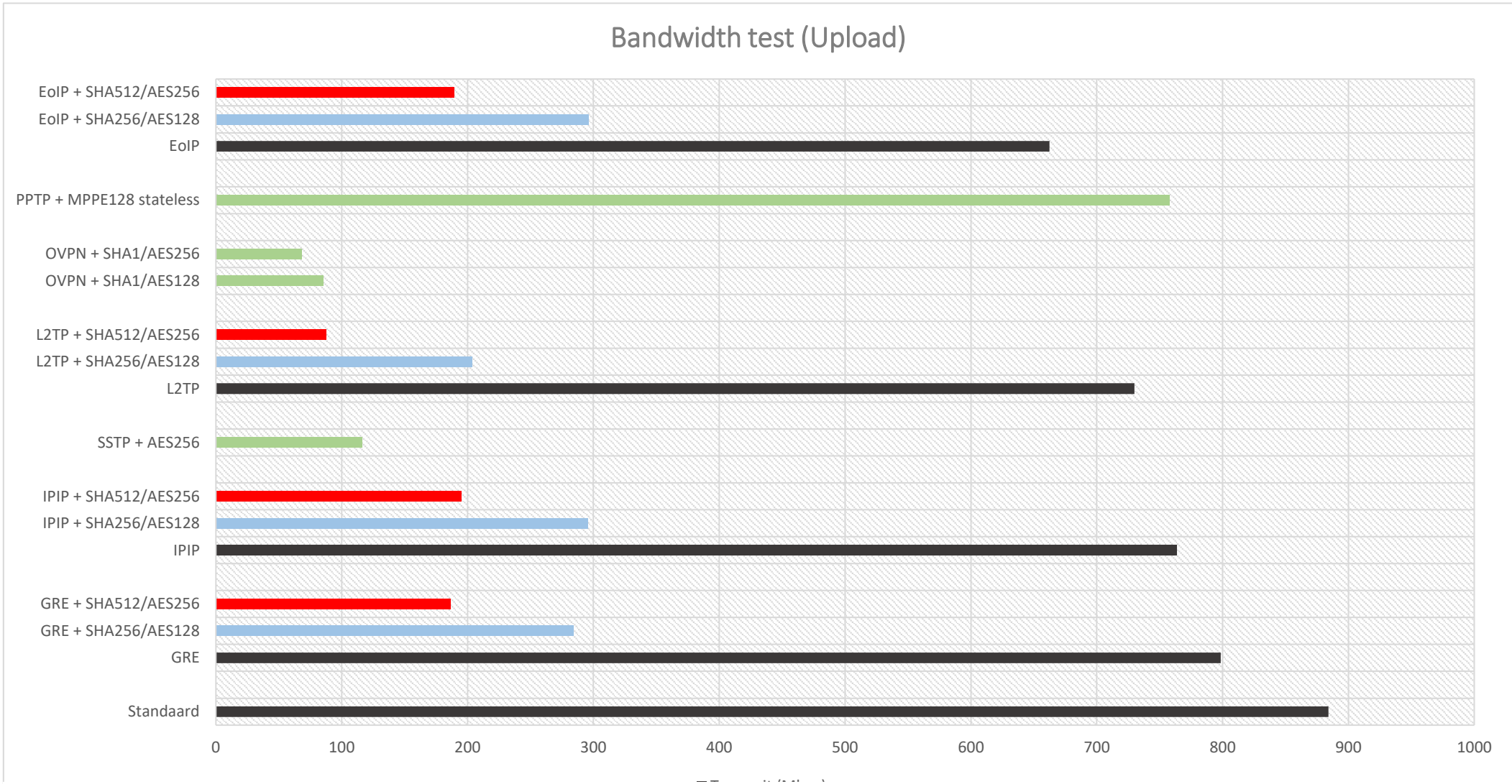


50 Layer 3 labo opstelling

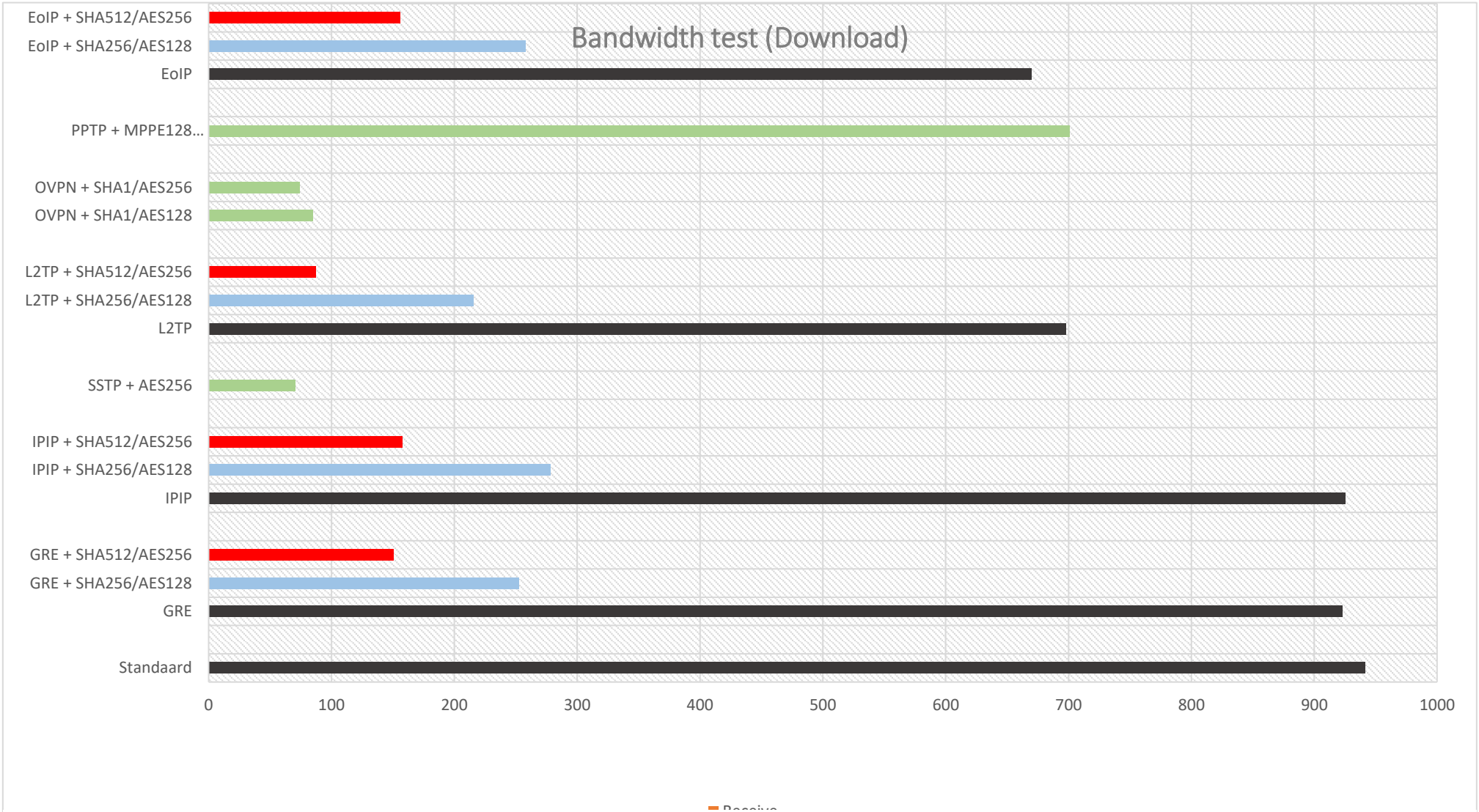
3.2.3 Meetresultaten

	Encryptie	Transmit	% verlies	Receive	% verlies
Standaard	x	884,2	x	941,3	x
GRE	x	798,5	9,69%	922,9	1,95%
	SHA256/AES128	284,4	67,84%	253	73,12%
	SHA512/AES256	186,6	78,90%	150,6	84,00%
IPIP	x	763,8	13,62%	925,6	1,67%
	SHA256/AES128	295,7	66,56%	278,3	70,43%
	SHA2512/AES256	194,9	77,96%	157,9	83,23%
SSTP	AES256	116,1	86,87%	70,4	92,52%
L2TP		730	17,44%	698	25,85%
	SHA256/AES128	203,7	76,96%	215,8	77,07%
	SHA512/AES256	87,5	90,10%	87,3	90,73%
OVPN	SHA1/AES128	85,7	90,31%	85,2	90,95%
	SHA1/AES256	68,1	92,30%	74,5	92,09%
PPTP	MPPE128 stateless	758	14,27%	700,8	25,55%
EoIP	x	662,5	25,07%	670	28,82%
	SHA256/AES128	296,3	66,49%	258	72,59%
	SHA512/AES256	189,2	78,60%	156,1	83,42%

Tabel 1 Meetresultaten onderzoeksopdracht



51 Bandwidth test(Upload)



52 Bandwidth test(Download)

3.3 Vergelijking

3.3.1 Encryptie

Encryptie is een belangrijk onderdeel bij VPN-tunnels. Door deze algoritmes is het mogelijk om data zo te encrypteren dat er niemand anders deze data kan zien buiten de zender en de ontvanger.

Na het onderzoeken van deze verschillende VPN-tunnels aan de hand van gerenommeerde bronnen is er gebleken dat Mikrotik het nog steeds mogelijk maakt tunnels op te bouwen met encrypties waarvan de algoritmes achterhaald zijn. Deze algoritmes zijn jaren geleden al gekraakt geweest of worden als onveilig verklaard.

Bij het opstellen van een IPsec tunnel hebben we de keuze bij de authenticatie en bij de encryptie welk algoritme we hiervoor gaan gebruiken.

De authenticatie kan gebeuren aan de hand van:

- MD5
- SHA1
- SHA256
- SHA512

MD5 is een authenticatie algoritme dat bij iedereen wel bekend is. Het wordt vaak gebruikt bij databases om wachtwoorden om te vormen. Hierbij wordt geen rekening gehouden met hoe veilig MD5 effectief is. MD5 is een snel hashing algoritme, wat het onveilig maakt. Doordat het zo snel is kan een persoon miljoenen passwoorden proberen op één enkele CPU. Hieruit blijkt dat een snel algoritme niet altijd goed is.

SHA1 is de voorganger van SHA256 en SHA512. Als we SHA1 vergelijken met MD5, is SHA1 een veiliger algoritme omdat SHA1 meer rondes doet dan MD5 (80-64) en SHA1 heeft een 160-bit output en MD5 een 128 bit output. Hierdoor is het moeilijker om een SHA1 algoritme te kraken. Ook al is SHA1 sterker dan MD5, toch is dit geen veilig algoritme voor een VPN-tunnel. SHA1 is gekraakt door google in 2017 waardoor we deze niet gaan gebruiken.

SHA256 en SHA512 zijn beide algoritmes dat op de dag van vandaag nog niet gekraakt zijn. Het verschil zit hier bij het aantal rondes beide algoritmes maken. SHA512 doet meer rondes dan SHA256 waardoor het een verhoogde bit sterkte geeft. Wat de snelheid betreft is dit afhankelijk van welk platform hiervoor gebruikt wordt, 64 of 32 bit, en de hoeveelheid data dat er gehasht wordt.

De encryptie kan gebeuren aan de hand van:

- DES/3DES
- Twofish
- Camellia128-256
- AES128-256

18.

Als eerste hebben we DES/3DES. DES is een gekraakt algoritme. Dit gebeurde op één dag. Hierdoor sluiten we DES uit om te gebruiken in dit project. 3DES is een algoritme dat momenteel nog niet gekraakt is. Hoewel dit alleen een kwestie van tijd is. 3DES is een oud algoritme en ook bekend voor zijn lage snelheid.

AES128,192,256 zijn alle 3 nog niet gebroken. AES128 alleen al zou honderden jaren duren eer het gekraakt zou kunnen worden, en kunnen we dit dus als veilig beschouwen. AES192,256 zijn bredere versies van AES128 waardoor deze ook als veilig beschouwd worden. AES is ook een algoritme dat aangeraden wordt momenteel en dat het meest wordt toegepast. Ook een groot voordeel van AES is dat het gebruik kan maken van hardware encryptie waardoor het sneller is dan andere encrypties die softwarematig gebeuren.

Twofish en Camellia zijn beide algoritmes die nog niet gebroken zijn. Deze zijn sterke tegenstanders van AES. Twofish, Camellia en AES zijn zo gelijkaardig dat het niet uitmaakt welk van deze encrypties geïmplementeerd wordt. Ik ga verder met AES omdat dit een encryptie is dat niet is gekraakt is, de snelheid hoog ligt voor zijn encryptie en ondersteunt wordt wanneer er hardware encryptie mogelijk is.

3.3.2 Bandbreedte

Voor een bandbreedte test correct uit te voeren zijn er bepaalde vereiste waar aan voldaan moeten zijn. Als eerste moesten al de metingen die gedaan worden, gebeuren van één apparaat naar één apparaat. De richting mag niet verwisseld worden want zo kunnen er verschillende waardes worden gemeten. Ook moeten deze testen met éénzelfde configuratie gebeuren. Bij de bandbreedte test die ik gebruikte was de enige instelling die veranderd kon worden het aantal TCP-connecties.

Ten slotte moet ook de encryptie gelijk worden gesteld tijdens de metingen. Als we een meting van één VPN-tunnel doen met een hoge encryptie en deze vergelijken met een VPN-tunnel met een lage encryptie zullen de waardes niet correct vergeleken kunnen worden.

Bij iedere tunnel die ik heb geconfigureerd heb ik telkens de laagste en de hoogste encryptie getest. Dit zijn encrypties die nog niet gebroken zijn geweest en nog beschouwd worden als veilig.

Van de grafiek kunnen we afleiden dat er 1 VPN-tunnel met encryptie veel hoger ligt dan al de andere tunnels. PPTP is een VPN-tunnel dat gebruik maakt van MPPE 128 stateless. Helaas is dit protocol niet meer veilig en kunnen we dit dus niet gebruiken.

Naast PPTP komen er 3 VPN-tunnels uit die ongeveer dezelfde waardes hebben. EoIP, GRE en IPIP. Doordat deze alle 3 duidelijk sterker zijn wat betreft bandbreedte gaan we deze vergelijken op hun specificaties.

Na het bekijken van de grafieken zien we dat IPIP, wat betreft bandbreedte, het beste uit de metingen komt. Maar als we naar de specificaties gaan kijken zit er wel wat verschil in de 3 VPN-tunnels.

	GRE	IPIP	EoIP
Encryptie	AES128-256	AES128-256	AES128-256
Add Overhead	24 bytes	20 bytes	42 bytes
Packet sequencing	X	0	X
Packet fragmentation	X	0	x
Encapsulate layer 3 protocols	X	0	X
Point-to-point	X	X	X
Moeilijkheidsgraad installatie	Makkelijk	Makkelijk	Makkelijk
Supported	Meerdere merken	Meerdere merken	Mikrotik <-> mikrotik

Tabel 2 vergelijking vpn-oplossingen onderzoeksopdracht

Op encryptie niveau verschillen deze VPN-tunnels niet van elkaar. Geen van deze VPN-tunnels is standaard geëncrypteerd en dus niet veilig. Hierdoor wordt op de 3 tunnels IPsec geconfigureerd zodat er encryptie gebeurd. Dit verklaart ook de gelijkheid in encryptie.

Overhead is data dat extra wordt toegevoegd aan iedere payload dat verzonden wordt. Dit moet zo laag mogelijk gehouden worden. EoIP heeft hier de hoogste overhead wat logisch is doordat EoIP werkt met GRE. Die 42 bytes komen van 8 byte GRE + 14 byte ethernet + 20 byte IP. IPIP scoort hier laag op doordat bij IPIP alleen een IP-header wordt toegevoegd van 20 bytes en hier niet gebeurt aan pakket fragmentatie.

Packet sequencing zorgt ervoor dat pakketten die verzonden worden, aankomen in diezelfde volgorde. Indien een pakket niet in de juiste volgorde komt, wordt dit pakket gedropt. Dit zorgt voor efficiëntie, consistentie en een extra laag beveiliging.

Packet fragmentatie heeft te maken met MTU (Maximum Transmission Unit). Elk netwerk heeft een MTU, dit is de maximum grootte dat een pakket kan zijn. Elk pakket dat groter is dan deze MTU zal moeten worden gefragmenteerd. Doordat dit gebruikt wordt zal er wel meer CPU nodig zijn. Maar sinds wij werken met multi-core Mikrotiks is dit geen probleem

4. Conclusie en aanbevelingen

4.1 Conclusie

Na een grondige vergelijking tussen de mogelijke VPN-tunnels op een Mikrotik hebben we verschillende data langs elkaar kunnen leggen om uiteindelijk een besluit te maken welke tunnel past binnenin het project. Hieruit is gebleken dat een GRE-tunnel als best passende tunnel uit de vergelijking kwam. Dit doordat de snelheid van een GRE-tunnel aan de hoge kant want vergeleken met de andere VPN-tunnels. GRE was ook een bekend tunnel protocol bij Cegeka waardoor we voor het onderzoek al meer die kant uit keken. Eenmaal de POC afgerond te hebben wist ik zeker dat GRE perfect paste binnenin mijn project.

4.2 Aanbevelingen

Na het afronden van mijn onderzoeksopdracht was er 1 vraag dat voor mij onbeantwoord bleef. Waarom is er geen protocol dat een tunnel opbouwt aan de hand van 2 IP-adressen? Wanneer we nu tunnels opbouwen zijn het altijd verschillende protocollen dat we bij elkaar moeten mengen om 1 deftige tunnel te configureren. Dit kost tijd en veel configuratie. Terwijl we eigenlijk alleen de 2 IP-adressen moeten hebben van de end devices om de pakketten te vertellen waar ze naartoe moeten worden gestuurd.

4.3 Persoonlijke reflectie

Vooraleer ik begon aan mijn onderzoek wist ik nog niet veel van VPN's af. Ik had er wel al opgezet maar daar bleef het dan ook bij. En nu moest Ik informatie opzoeken over VPN's op een apparaat waar ik ook nog maar pas mee in aanraking was gekomen. Ik wist dus al dat ik veel ging bijleren van dit onderzoek.

Het begon bij het opzoeken van alle mogelijke tunnels, hier kwam ik tunnels tegen waarvan ik niet wist dat deze bestonden. Nadat ik al deze tunnels had opgezocht moest ik gaan kijken welke paste in mijn project, omdat er tunnels waren die niet diende voor een site-to-site verbinding te maken moest ik al direct VPN's gaan schrappen. Bij het opzoeken van al deze data ben ik wel wat tijd verloren, de bekende VPN's stonden wel uitgelegd op de officiele pagina van Mikrotik zelf, maar deze informatie was niet genoeg om iedere tunnel uitgebreid uit te kunnen leggen.

Elke VPN is daarna geconfigureerd geweest wat wel wat tijd in beslag nam omdat VPN-tunnels opzetten op een Mikrotik volledig onbekend terrein was voor mij. Nadat ik deze tunnels had opgezet kon ik de pakketten gaan capturen. Wireshark was software waar ik wel al mee in aanraking was gekomen, wat me dan wel hielp met het bekijken van de verschillende pakketten. Hier kon ik elke layer zien van elk pakket dat verstuurd werd, wat me toch wel een beter zicht gaf over wat iedere tunnel precies deed met het pakket.

Buiten VPN-tunnels heb ik ook veel opgezocht over encryptie. Doordat sommige VPN-tunnels standaard niet geëncrypteerd waren moest hier nog met IPsec gewerkt worden. IPsec was een term waar ik wel al van had gehoord, maar na wat documentatie te hebben doorzocht kan ik zeggen dat ik toch meer weet over IPsec.

Als ik terugkijk naar hoe ik dacht over dit onderzoek op het begin en nu, had ik een heel ander gedacht over welke tunnel effectief als “de beste” hieruit ging komen. Zelf dacht ik dat OpenVPN een van de betere VPN-tunnels was en deze gebruikt ging worden. Ook van de snelheden was ik verrast. Bij het meten van de tunnels zonder encryptie had ik het gevoel dat deze waarden ongeveer met 10% ging dalen, maar na het configureren van IPsec daalde deze waardes enorm.

Conclusie

Als ik terugblik naar mijn stage en mijn eerste dag vergelijk met mijn laatste dag, mag ik met een gerust hart zeggen dat dit voor mij zeer goed is verlopen. Ik herinner me nog goed de dag dat ik aankwam op stage, een beetje verloren. Je komt terecht in een groot bedrijf en weet niet waar naartoe te gaan. Toen ik uiteindelijk aankwam in de ruimte waar mijn heel team zat was dit toch wel even wennen. Ik had geen vaste plaats doordat de ruimte al redelijk volzet was met al de system engineers. Niet veel later werd ik aan een bureau gezet met 4, toen nog onbekende, collega's. Al snel liep alles los. Maar het gevoel van onwetendheid bleef zeker overheersen de eerste weken. De eerste dagen zullen mij zeker bijblijven doordat ik iedereen nog moest leren kennen. Ik kreeg te maken met zeer veel nieuwe technologieën, waarmee ik nog niet veel gewerkt had. De eerste weken waren voor mij vooral een kennismaking met deze technologieën. Uiteindelijk was dit voor mij zeer positief, ik heb hier de tijd gekregen om dagen aan een stuk met deze technologieën bezig te zijn, hierdoor zijn mijn vaardigheden in stijl tempo gestegen. Na de kennismakingsperiode kwam de periode waarin alles een beetje duidelijk werd. Ik was nu in de mogelijkheid om vlot met de technologieën te werken. Ik kon nu mijn creativiteit loslaten in deze technologieën.

Eenmaal ik volledig weg was met de technologieën kon ik aan mijn onderzoeksopdracht beginnen. Hierbij zijn er wel wat problemen bij komen kijken , zoals de metingen die ik had uitgevoerd om dan tot de conclusie te komen dat ik zat met een bottleneck. Doordat ik een gratis versie gebruikte bleven mijn poorten op 1 Mbit/sec werken en mijn metingen dus zeer laag waren. Ook waren er nog wat CPU problemen op de kleine mikrotiks bij het genereren van testdata. Daarom zijn de mikrotiks uit de testomgeving gehaald en er grote Cloud Core Routers gebruikt zodat we duidelijke resultaten hadden. Uiteindelijk had ik zelf verwacht veel sneller klaar te zijn met de onderzoeksopdracht, maar door het opbouwen van een volledig labo, en binnen dat labo de verschillende vpn-tunnels nog teens te bouwen, heeft dit toch wel wat tijd in beslag genomen.

Na het afwerken van mijn onderzoeksopdracht was er toch wel wat stres door de hoeveelheid tijd die ik nog had voor een volledige stageopdracht af te werken. Veel tijd om te treuzelen was er dus niet meer en daarom dat ik hier direct in ben gevlogen. Ik moest nog een heel framework leren gebruiken en methodes zoeken om te automatiseren. De laatste 2 weken waren de meest stresvolle weken. Zoals altijd gebeuren er dingen die niet moeten gebeuren en kosten deze problemen ook nog eens veel tijd.

Uiteindelijk de laatste week kon het testen gebeuren. Dit heeft niet veel tijd in beslag genomen en alles verliep vlot, wat een opluchting dat was. Op dit moment stond ik vol vertrouwen achter mijn afgeleverde product en kon ik mijn eindwerk afwerken. Dan komt de laatste fase. Alles is af. Je bent klaar met de opdracht. Je kijkt terug op de periode en merkt dat de 3 maanden echt zijn omgevlogen. Voor de stage begon was ik aan het twijfelen tussen verder studeren of werken. Deze stage heeft mij overtuigd om te gaan werken, ik denk dat dit wel de perfecte uitlegt hoe goed de stage me bevallen is!

Bibliografie

[1] CISCO, „CISCO,” 2 04 2016. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/wireless/asr_5000/20/IPSec/b_20_IPSec/b_20_IPSec_chapter_01.pdf.

[2] Mikrotik, „Manual:Interface/EoIP,” 6 9 2017. [Online]. Available: https://wiki.mikrotik.com/wiki/Manual:Interface/EoIP.

[3] Mikrotik, „Manual:Interface/PPTP,” Mikrotik, 4 1 2017. [Online].

[4] Mikrotik, „Manual:Interface/SSTP,” 15 10 2016. [Online]. Available: https://wiki.mikrotik.com/wiki/Manual:Interface/SSTP.

[5] Mikrotik, „Mikrotik Manual:Interface/Gre,” Mikrotik, 7 12 2015. [Online]. Available: https://wiki.mikrotik.com/wiki/Manual:Interface/Gre.

[6] Mikrotik, „Mikrotik Manual:Interface/IPIP,” Mikrotik, 7 12 2015. [Online]. Available: https://wiki.mikrotik.com/wiki/Manual:Interface/IPIP.

[7] Mikrotik, „Mikrotik Manual:IP/IPsec,” Mikrotik, 13 11 2017. [Online]. Available: https://wiki.mikrotik.com/wiki/Manual:IP/IPsec.

[8] Mikrotik, „Mikrotik OpenVPN,” Mikrotik, 12 5 2015. [Online]. Available: https://wiki.mikrotik.com/wiki/OpenVPN.

[9] Symfony, „Symfony API,” [Online]. Available: http://api.symfony.com/3.4/index.html.

[10] B. Forum, „Broadband Forum,” 11 2013. [Online]. Available: https://www.broadband-forum.org/technical/download/TR-069_Amendment-5.pdf.

[11] qacafe, „TR-069 training,” [Online]. Available: https://www.qacafe.com/tr-069-training/.

[12] Mikrotik, „Manual:TR069-client,” 26 4 2017. [Online]. Available: https://wiki.mikrotik.com/wiki/Manual:TR069-client.

[13] Symfony, „Installing & Setting up the Symfony Framework,” [Online]. Available: https://symfony.com/doc/3.4/setup.html.

[14] B. Guzel, „HTTP Headers for Dummies,” 2 12 2009. [Online]. Available: https://code.tutsplus.com/tutorials/http-headers-for-dummies--net-8039.

[15] Symfony, „Databases and the Doctrine ORM,” [Online]. Available: https://symfony.com/doc/3.4/doctrine.html.

Bijlage