CVO Antwerpen-Zuid

dag- en avondonderwijs voor volwassenen

Open Source Virtualisering bij een Kleine VZW

Schooljaar 2007 – 2008

Projectwerk tot het behalen van
het diploma Hoger Beroepsonderwijs
Studiegebied Handelswetenschappen en bedrijfskunde
Afdeling Informatica
Optie Netwerkbeheer

Cursist : Peter Veltmans

Docent : Marc Rousseau en Vaya Willemen

CVO Antwerpen-Zuid

dag- en avondonderwijs voor volwassenen

Open Source Virtualisering bij een Kleine VZW

Schooljaar 2007 – 2008

Projectwerk tot het behalen van
het diploma Hoger Beroepsonderwijs
Studiegebied Handelswetenschappen en bedrijfskunde
Afdeling Informatica
Optie Netwerkbeheer

Cursist : Peter Veltmans

Docent : Marc Rousseau en Vaya Willemen

Abstract

Een kleine vzw, gevestigd te Brussel in België, beschikt over een klein computernetwerk, bestaande uit drie personal computers en een aantal printers. Voor het werk van alledag worden deze machines, die voorzien zijn van propriëtaire software, op een feitelijk onbeheerde manier gebruikt. Dit leidt, naast heel wat dataduplicatie en -proliferatie, tot een chaotische, ongeorganiseerde werkwijze inzake het installeren van verschillende softwareproducten. Het zou verkieslijk zijn het bestaande netwerk zodanig te reorganiseren dat data en applicatiesoftware gecentraliseerd kunnen worden en tegelijk beschikbaar worden voor lokaal gebruik en voor toegang vanop afstand. Hiervoor lijkt een server-georiënteerde aanpak het beste. Dit blijkt echter onhaalbaar, daar de organisatie beperkt is in haar financiële mogelijkheden. Tenzij er adequaat gebruik gemaakt wordt van virtualisatietechnieken en van software onder open bron licentie. Dit werk heeft de ambitie een aantal van deze technieken en softwarepakketten op hun beloftevolle mogelijkheden te onderzoeken.

Une petite a.s.b.l., située à Bruxelles en Belgique, dispose d'un réseau informatique limité qui ce compose de trois ordinateurs et de quelques 'imprimantes. Pour le travail quotidien, ces machines, qui dispose d'un logiciel propriétaire, sont utilisées de manière non-planifiée. Cela conduit, outre la duplications et la prolifération inutile des données, à un processus de travail chaotique et mal organisée en matière d'installation des différents logiciels. Il serait préférable de réorganiser le réseau de façon à ce que les données et les logiciels d'application puissent être centralisés et disponibles, simultanément pour usage local que pour l'accès à distance. Une approche orientée sur le serveur semble la meilleure option. Les moyens financiers de l'organisation étant restreints, ceci semble irréalisables à premier vue, à moins d'utiliser de façon adéquate des techniques de virtualisation et des logiciels sous licence à source ouverte. Cet ouvrage s'est fixé pour but l'étude d'un certain nombre de ces techniques et de leurs possibilités prometteurs.

A small nonprofit organization, situated in Brussels, Belgium, is in possession of a small computer network, consisting of three personal computers and a bunch of printers. For day to day work, these machines, which are run with proprietary software, are used in an essentially unmanaged way. This leads to a lot of data duplication and proliferation, combined with a chaotic, unorganized way of installing various software products. It would be preferable to reorganize the existing network in such a way that data and applications can be centralized and made available for local as well as remote access. For this a server oriented approach would seem to be the best solution. However, as the organization is limited by financial constraints, this does not seem to be feasible. Unless adequate use is made of the possibilities provided for by virtualization techniques and of open source licensed software. The present work has the ambition to investigate the promising possibilities of some of these techniques and software packages.

Conventies

- In de tekst werd zoveel mogelijk geprobeerd om Nederlandstalige termen te gebruiken. Helaas is dat echter niet altijd mogelijk. Computertechnologie is in hoge mate Angelsaksisch geörienteerd. Soms is het gebruik van Engelstalige termen dan ook niet te vermijden. Zoveel als mogelijk werden die Engelse termen gecursiveerd weergegeven. In een aantal gevallen ook hebben we ons verplicht gevoeld om gebruik te maken van Anglicismen, dit wil zeggen dat we gebruik maakten van 'vernederlandste' Engelse woorden.
- Wanneer er is in de tekst regels voorkomen in vet, beginnend met een prompt-aanduiding, in een niet-proportioneel lettertype, dan gaat het om code die rechtstreeks werd ingetikt in een Linux-console. Bijvoorbeeld :

```
[root@testmachien ~]# service xend start
```

- Wanneer dergelijke codevoorbeelden starten met een **#** (na de prompt), dan wijst dit erop dat dit commando dient te worden gegeven als 'root' (het Linux-equivalent van de 'administrator' onder Windows).
- Wanneer echter het teken **$** staat na de prompt, dan betekent dit dat het commando dient te worden gegeven door een gewone gebruiker.
- Soms kan een commando niet worden weergegeven op één enkele regel (terwijl het in de console wel als één regel dient te worden ingevoerd). In dat geval zal die ene regel opgesplitst worden over meerdere regels, waarbij elk regeleinde wordt weergegeven als een backslash (\), bijvoorbeeld als volgt :

```
[root@testmachien ~]# echo -n 0000:05:00.0 > \
/sys/bus/pci/drivers/r8169/unbind
```

Dankwoord

Elk werk van enige omvang kan slechts tot een goed einde worden gebracht mits de welwillende steun van anderen. Dat is hier niet anders. Ik wil dan ook mijn moeder, broers en zussen bedanken voor het geduld dat ze hebben opgebracht voor mijn maandenlange, quasi volledige afwezigheid. Maandenlang ook hebben Marcel, Dirk, Guy, Peter, Jurgen, net zoals Frieda, Petra, Eva, Mia en ongetwijfeld nog een heleboel andere vrienden en vriendinnen, moeten verdragen dat elk gesprek met mij al gauw uitdraaide op technische, computer-gerelateerde kwesties. Ik vrees dat ik hun geduld danig op de proef heb gesteld. Dat ze het toch – soms zelfs geamuseerd – ondergaan hebben, daarvoor bedank ik hen van harte. Mijn collega's Ronny, Hugo en Gerda hebben dan weer regelmatig de volledige inzet van mijn arbeidskracht moeten missen. Ik dank hen voor hun begrip. Ook mijn medecursisten wil ik bedanken. Hun ideeën en praktische voorbeelden waren vaak een grote inspiratiebron. Ook Linus Torvalds en Richard Stallman ben ik grote dank verschuldigd. Zonder de door hen op gang gebrachte beweging rond vrije of open source software, zou ik niet eens een onderwerp hebben gehad om over te schrijven. Hetzelfde geldt natuurlijk voor de gemeenschappen rond Xen, OpenVZ en KVM/Qemu. Anne Bausart en Monique Nagielkopf hielpen met de Franse vertaling van de tekst der abstract, waarvoor mijn dank. Tenslotte wil ik ook Marc Rousseau en Vaya Willemen danken. Ondanks hun vele werk als docent waren ze steeds beschikbaar voor commentaar en goede raad. Zonder hen zou dit werk niet zijn wat het uiteindelijk geworden is.

Peter Veltmans, 2 juni 2008.

Toelating tot bruikleen

Inhoudstafel

Abstract 3
Conventies 5
Dankwoord 7
Toelating tot bruikleen 7
Inhoudstafel 9

1. Inleiding 15

2. Functionele Analyse 17
2.1. Inleiding 17
2.2. Formele kijk op organisatiestructuur van de vzw 17
2.2.1. Formele organisatiestructuur van de vzw 17
2.2.2. Formele taakverdeling in de vzw 18
2.3. Praktische kijk op de organisatie van de vzw 19
2.3.1. Praktische organisatiestructuur van de vzw 20
2.3.2. Praktische taakverdeling in de vzw 21
2.3.2.1. Intern gerichte processen 22
2.3.2.1.1. Workflow ‘briefwisseling’ 22
2.3.2.1.2. Workflow ‘rekeningbeheer’ 24
2.3.2.1.3. Workflow ‘infrastructuur’ 24
2.3.2.1.4. Workflow ‘subsidiedossiers’ 25
2.3.2.1.5. Workflow ‘ledenbeweging’ 26
2.3.2.1.6. Workflow ‘werkingsthema’ 26
2.3.2.2. Extern gerichte processen 27
2.3.2.2.1. Workflow ‘tijdschriften’ 27
2.3.2.2.2. Workflow ‘website’ 28
2.3.2.2.3. Workflow ‘promotiemateriaal’ 28
2.3.2.2.4. Workflow ‘audiovisueel’ 29
2.3.2.2.5. Workflow ‘uitgave didactisch materiaal’ 29
2.4. Conclusies van de functionele analyse 30

3. Technische Analyse 33
3.1. Inleiding 33
3.2. Plattegronden 33
3.2.1. Algemene indeling 33
3.2.2. Detail van de secretariaat-ruimte 35
3.3. Beschrijving van het bestaande netwerk 36
3.3.1. Inleidend overzicht 36
3.3.2. De aanwezige router 37
3.3.3. Beschrijving van de aanwezige computers 37
3.3.3.1. Computer ‘A’ 37
3.3.3.2. Computer ‘B’ 38
3.3.3.3. Computer ‘C’ 39
3.3.4. Beschrijving van de aanwezige printers 40
3.3.5. Beschrijving van de aanwezige software 41
3.3.6. Beschrijving van het omgaan met data 41
3.4. Voorlopige conclusies 41
3.5. Open Source als mogelijke oplossing ? 43
3.5.1. Wat is Open Source ? 43
3.5.2. Voordelen van Open Source 44
3.5.3. Nadelen van Open Source 44
3.6. Een mini-datacenter als oplossing ? 44
3.7. Virtualisering als oplossing ? 47
3.7.1. Overzicht van virtualiseringsmogelijkheden 47
3.7.2. Desktop- en Enterprise-virtualisering 48
3.7.3. Relatie virtualisering en besturingssysteem 48

4. Opzetten van een testomgeving 51
4.1. Inleiding 51
4.2. Overzicht van de testcomputer 51
4.2.1. Het moederbord 51
4.2.2. De processor 51
4.2.3. RAM-Geheugen 52
4.2.4. Grafische kaart 52

4.2.5. Harde schijven 52
4.2.6. RAID-Controller 52
4.2.7. Besturingssysteem 53
4.2.8. Indeling der harde schijven 54
4.2.9. RAID-configuratie 54
4.2.10. Partities en Logical Volumes 55
4.2.11. Initiële netwerk-configuratie 56
4.2.12. Een lokale *repository* aanmaken 57
4.2.13. Bedenkingen omtrent beveiliging 61
4.2.13.1. BIOS afschermen 61
4.2.13.2. GrUB en het uitschakelen van single user mode 62
4.2.13.3. Encryptie van de harde schijven 62
4.2.13.4. SELinux, iptables en Intrusion Detection 63
4.2.13.5. Blue Pill rootkits versus certificaten 63
4.2.13.6. Beveiligen der virtuele machines zelf 64
4.2.13.7. Versleutelde verbindingen 64
4.2.13.8. Brute force versus Single Packet Authorization 64

5. Oplossingen 67

5.1. Oplossing 1 : Xen 67
5.1.1. Verschillende versies van Xen 68
5.1.2. Netwerken onder Xen 69
5.1.2.1. *Bridged networking* 69
5.1.2.2. *Routed networking* 71
5.1.2.3. VLAN met NAT 72
5.1.3. Xen als binair bestand installeren 74
5.1.4. Soorten virtuele harde schijven onder Xen 77
5.1.5. Paravirtualisatie of volledige virtualisatie 78
5.1.6. Virtuele machines onder Xen 3.1 78
5.1.7. Virtuele machines en beveiliging onder Xen 86
5.1.8. Xen vanuit broncode compileren en installeren 95
5.1.9. Virtuele machines onder Xen 3.2 97

5.1.10. Programma's voor het beheer van een Xen-cluster 101
5.1.10.1. xen-tools 101
5.1.10.2. XenMan en ConVirt 102
5.1.10.3. openQRM 104
5.1.10.4. Enomalism 105
5.1.11. Xen - Besluit 106

5.2. Oplossing 2 : OpenVZ 107
5.2.1. Inleiding 107
5.2.2. Installatie van OpenVZ 108
5.2.3. Netwerken onder OpenVZ 110
5.2.3.1. Gebruik van een *virtual ethernet device* 111
5.2.3.2. Gebruik van een *virtual network device* 111
5.2.4. OpenVZ en beveiliging 112
5.2.5. Hulpprogramma's voor OpenVZ 113
5.2.6. OpenVZ *templates* 114
5.2.7. Een eerste OpenVZ-container installeren 114
5.2.8. Parameters voor OpenVZ-containers 118
5.2.9. Programma's voor het beheer van OpenVZ-containers 122
5.2.10. OpenVZ - Besluit 124

5.3. Oplossing 3 – KVM/Qemu 125

5.3.1. Inleiding 125
5.3.2. Installatie van KVM/Qemu 126
5.3.3. Netwerken onder KVM/Qemu 127
5.3.3.1. *Root networking* (Tuntap) 129
5.3.3.2. *User mode networking* (Slirp) 129
5.3.3.3. *Host only networking* 130
5.3.3.4. *Public bridged networking* 131
5.3.3.5. *Virtual distributed ethernet networking* (VDE) 133
5.3.4. Virtuele harde schijven onder KVM/Qemu 134
5.3.5. Virtuele machines installeren onder KVM/Qemu 136

5.3.6. Beveiliging onder KVM/Qemu 137
5.3.7. Beheer onder KVM/Qemu 137
5.3.8. KVM/Qemu – Besluit 139

6. Samenvatting, aanbeveling en besluit 141

Lijst van afbeeldingen 145
Lijst van tabellen 146
Bibliografie 147
Appendix A : Oorspronkelijke probleemomschrijving 153
A.1. Situatieschets 153
A.2. Doelstellingen 153
Appendix B : Akkoordverklaring der vzw 155
Appendix C : Verslag vergadering met vzw 157
Appendix D : Overzicht software-installatie bij de vzw 161
Appendix E : Kickstart-script voor Xen VM 165
Appendix F : Script voor een veilige Xen VM 169
Appendix G : Simpel script voor OpenVZ-containers 171
Appendix H : Implementatieplan 177

1. Inleiding

Dit projectwerk is gestart vanuit de verzuchtingen van een kleine vzw. Die vzw is op zoek naar een haalbare manier om met behulp van informatica haar interne organisatie drastisch te verbeteren. Het resultaat moet de leden van de vzw uiteindelijk toestaan om het geheel van hun werkzaamheden af te handelen via één (of meer) centrale computer(s). Ook dienen de leden van de vzw deze centrale computer(s) te kunnen benaderen vanop afstand. Tegelijk mag dit alles niet te veel kosten.

Tijdens de analysefase bekeek ik de behoeften van de vzw zo goed mogelijk. Al snel leidde die analyse tot de slotsom dat de vzw inderdaad behoefte had aan een centraal computersysteem. Tegelijk echter diende dat centraal computersysteem in te staan voor meerdere, verschillende diensten. Er was met andere woorden nood aan meerdere serversystemen, bij voorkeur onder een *open source* licentie. Dit laatste om dure licentievoorwaarden te kunnen ontlopen.

Aangezien de vzw ietwat armlastig is, bleek daarenboven al gauw – om niet te zeggen direct – dat meerdere serversystemen op aparte servercomputers een onhaalbare kaart was. Virtualisering bleek hier echter een oplossing voor te kunnen bieden. Anders gezegd, dankzij virtualisering zou het mogelijk worden om op één centrale computer meerdere virtuele serversystemen geïsoleerd van elkaar te laten draaien. Daarvoor moest het natuurlijk wel een 'gespierde' computer zijn.

Op dat ogenblik had ik echter geen concrete ervaring met open source virtualiseringsoplossingen. Ik had het naïeve vermoeden dat het allemaal ietwat zou lijken op VMWare Server of op Microsoft's Virtual PC. Ik ging er dan ook vanuit dat ik het meeste werk zou hebben, niet met de virtualiseringsoplossingen, maar wel met het concreet configureren van de diverse serversystemen. Dat draaide helemaal anders uit.

Uiteindelijk bleken de *open source* virtualiseringsoplossingen bijzonder rijk aan mogelijkheden te zijn, maar tegelijk ook heel verschillend in de manier waarop die mogelijkheden worden waargemaakt. Ook de moeilijkheidsgraad bleek niet te onderschatten (al waren niet alle oplossingen even moeilijk).

Daar kwam nog bij dat ik weliswaar enige kennis bezit van het Linux-besturingssysteem, maar dat ik toch werkelijk niet kan beweren dat ik er echt grondig van op de hoogte ben. Dat bleek dan ook een handicap in het werken met de hier behandelde virtualiseringsoplossingen.

Ik heb dan ook een dubbele leercurve moeten doorlopen tijdens het werken aan dit project. Enerzijds het optimaler leren werken met Linux en anderzijds het leren werken met drie verschillende virtualiseringsoplossingen onder Linux.

Het resultaat toont naar mijn mening aan dat *open source* virtualisering inderdaad gebruikt kan worden om op één centrale computer meerdere virtuele serversystemen te creëren, die dan bruikbaar kunnen zijn voor, bijvoorbeeld, een kleine organisatie, zoals een vzw.

1. Functionele Analyse

2.1. Inleiding

Het doel van een functionele analyse bestaat eruit een organigram te maken van de manier waarop het zogenaamde *business proces* verloopt in de organisatie die we bestuderen. We kijken dus naar de werkelijkheid van de organisatie zoals die er vandaag uitziet (de *as is situation*, in managementjargon). Concreet wil dat zeggen dat we enerzijds de bestaande structuur van de organisatie beschrijven en anderzijds uiteenzetten hoe de vervulling van de verschillende zakelijke processen in de organisatie precies verlopen en welke concrete rollen of functies daarmee samenhangen.

We stoten daarbij echter al meteen op een probleem. De literatuur over dit onderwerp handelt immers voornamelijk – zoniet uitsluitend – over ondernemingen. Ons studieobject is echter geen zuivere onderneming, maar een vereniging zonder winstoogmerk. Dat lijkt misschien een klein verschil, maar is het niet. Bij een onderneming kunnen we veralgemeend stellen dat die slechts één doel heeft : het maken van winst[1]. Daaruit volgt quasi direct dat zowel de structuur als de verschillende zakelijke processen (en de rollen of functies die ermee samenhangen), bekeken kunnen worden vanuit dit ene oogmerk. Bij een vzw is dat niet helemaal hetzelfde. *"Wie een vzw opricht, moet voor ogen houden dat de vereniging een menslievend, ideëel en/of onbaatzuchtig doel moet hebben en dat de "commerciële daden" steeds bijkomstig moeten zijn om het hoofddoel te realiseren."*[2]

Bovendien worden zowel de structuur als de zakelijke processen en rollen niet alleen bepaald door de wettelijk omschreven verplichtingen, noch door de zelfgekozen doelstelling van de vzw, maar ook door – zelfopgelegde – ideologische beperkingen en/of wensen. Dit zal duidelijk worden in de hieronder volgende beschrijving.

2.2. Formele kijk op de organisatiestructuur van de vzw

2.2.1. Formele organisatiestructuur van de vzw

De formele structuur van een vzw is niet iets dat zomaar vrij gekozen kan worden (net zomin als die van een onderneming overigens). Concreet legt de vigerende wetgeving regels op waaraan elke vzw dient te beantwoorden.[3]

1 *"Ieder mens tracht zijn kapitaal zodanig te laten werken dat het de grootste waarde oplevert. Men heeft over het algemeen noch de intentie om het algemene belang te dienen, noch weet men in welke mate men dat belang dient. Men richt zich slechts op zijn eigen veiligheid, zijn eigenbelang. En daarin wordt de mens geleid door een onzichtbare hand om een doel te die-nen dat geen onderdeel van zijn intentie vormt."* in SMITH, Adam, *'The Wealth of Nations*', 1776, London, Penguin Classics.

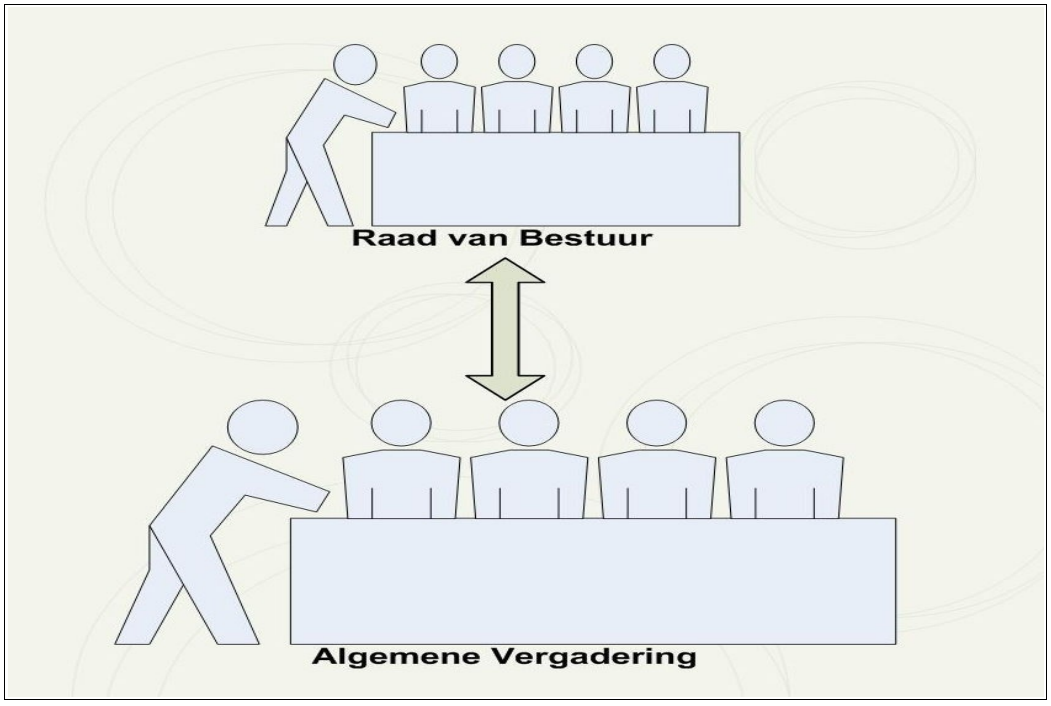
2 Citaat afkomstig van de website van het 'Vlaams Studie- en Documentatiecentrum voor V.Z.W.'s' : http://www.vsdc.be/content/frame.htm

3 Zie hiervoor dezelfde hoger genoemde website van het 'Vlaams Studie- en Documentatiecentrum voor V.Z.W.'s' (http://www.vsdc.be/content/frame.htm), alsook deze meer op notarissen gerichte website (http://www.notare.be/vzw.htm) en deze zeer informatieve, maar meer algemene website (http://www.devzw.be/publicaties.php).

We gaan hier niet dieper in op tal van verplichtingen die te maken hebben met de statuten, waar die neergelegd dienen te worden en dergelijke zaken meer. Wel is het duidelijk dat alle v.z.w.'s een wettelijk opgelegde structuur hebben.[4]

Deze verplichte structuur bestaat uit een Algemene Vergadering en een Raad van Bestuur. In het algemeen is het zo dat de Algemene Vergadering de verzameling is van alle leden van de vzw. Deze Algemene Vergadering is ook het hoogste gezagsorgaan in de vzw. De Raad van Bestuur is het uitvoerend orgaan, dat de beslissingen van de Algemene Vergadering uitvoert en dus belast is met de uitvoering van het beleid.

Schematisch kunnen we dit als volgt voorstellen :



Afbeelding 1 : Eenvoudig overzicht der vzw-structuur

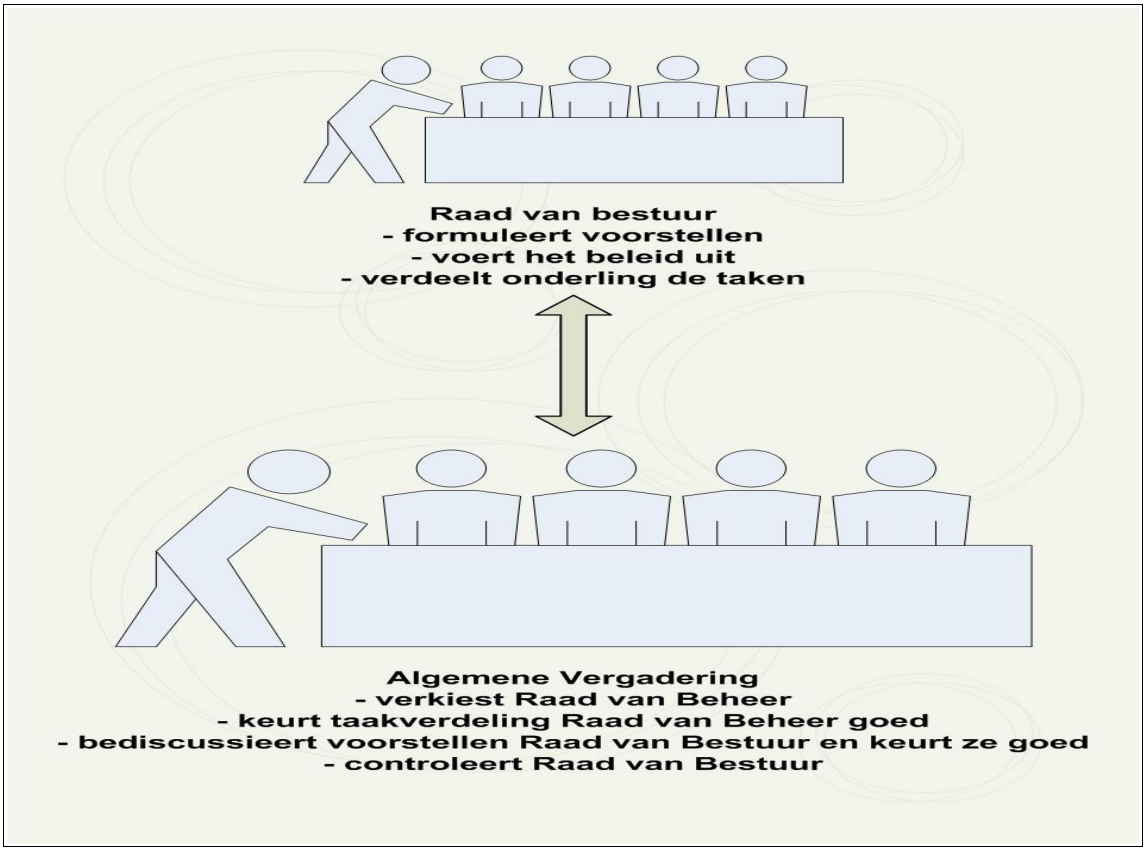
De wetgeving legt onder meer op dat in de statuten dient vermeld te worden hoe men lid kan worden van de vzw en hoe de leden van de Raad van Bestuur worden aangesteld en welke bevoegdheden ze precies hebben.[5]

2.2.2. Formele taakverdeling in de vzw

Een blik op de statuten leert ons dan ook direct welke 'rollen' er precies in de vzw aanwezig zijn. Daaruit blijkt dat deze vzw een onderscheid maakt tussen volgende verantwoordelijke functies (of 'rollen') : een voorzitter, een secretaris en een schatbewaarder. In het 'huishoudelijk reglement' van de vzw worden dan weer de regels vastgelegd waaraan de leden van de Algemene Vergadering (en à fortiori ook de le-

4 Wet van 27 juni 1921, zoals gewijzigd door de wet van 2 mei 2002.
5 Zie Art. 2 van de wet van 27 juni 1921, zoals gewijzigd door de wet van 2 mei 2002.

den van de Raad van Bestuur, aangezien zij ook lid zijn van de Algemene Vergadering) onderworpen zijn. Ook daaruit kunnen we functies of rollen binnen de vzw afleiden, die wezenlijk alle van praktische aard zijn en direct te maken hebben met de naar de buitenwereld gerichte werking van de vzw. Samengevat kunnen we het hier boven geschetste schema dan ook iets verder verfijnen en invullen :



Afbeelding 2 : Schema formele structuur der vzw

Aangestipt dient te worden dat de vzw niet beschikt over voldoende financiële middelen om ook permanent betaalde medewerkers in dienst te nemen. Wel betaald men soms individuen om één of andere dringende of ietwat gespecialiseerde taak snel tot een goed einde te brengen. Soms worden er ook subsidies aangevraagd voor een concreet project, waarbij er voor de duur van dat project meestal ook voorzien wordt in een tijdelijke betaalde kracht.

2.3. Praktische kijk op de organisatie van de vzw

Het mag duidelijk zijn dat de wettelijk opgelegde, formele structuur met daaraan verbonden de formele invulling van de taakverdeling nogal abstract is en niet direct beantwoordt aan de dikwijls complexe problemen waarmee ook een vzw geconfronteerd wordt.

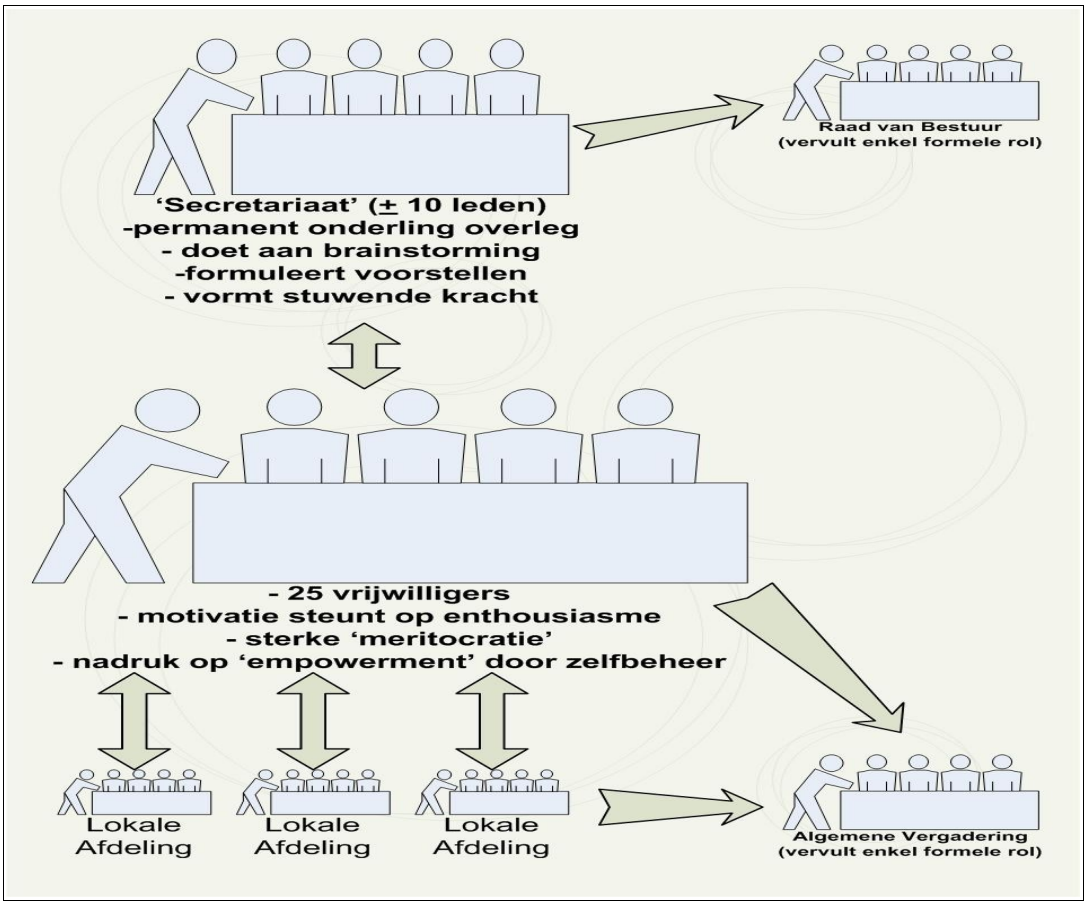
Bovendien huldigt de vzw intern ook een aantal stilzwijgende conventies, dewelke eerder van ideologische aard zijn.[6] In het geval van deze vzw betekent dat vooral een sterke nadruk op egalitarisme, zelfbeheer en (ecologische) duurzaamheid.

In zeer algemene termen komt deze ideologie neer op een vrij anarchistische manier van werken. Concreet staat dit voor een taakgerichte aanpak. Eerder dan te kiezen voor een formele opsomming van alle taken die voortvloeien uit de doelstellingen van de vzw (en uit de stilzwijgende, ideologische conventies) en die daarna dan toe te wijzen aan 'vast benoemde' concrete personen, gaat men ervan uit dat diegene die iets belangrijks vindt, het ook zelf wel zal uitvoeren. De uitvoerder wordt dan – als het ware uit zichzelf – ook de verantwoordelijke.

2.3.1. Praktische organisatiestructuur van de vzw

In de praktijk betekent dit voornamelijk dat de Raad van Bestuur enkel een formeel leven leidt (nodig voor het beantwoorden aan de wettelijke vereisten). De vzw functioneert dan ook feitelijk als een soort permanente Algemene Vergadering. Wel dient daarbij aangestipt te worden dat niet alle leden permanent even intensief deelnemen aan deze manier van werken. Een relatief kleine groep van zo'n vijfentwintig mensen neemt wekelijks deel aan één of andere taak binnen de v.z.w. en dit op basis van vrijwilligheid. Binnen deze groep van vijfentwintig vrijwilligers zijn er een tiental die voortdurend met elkaar in contact staan en dus feitelijk fungeren als een leidende groep. Zij noemen zichzelf 'het secretariaat'. Daarnaast is er ook een relatief sterke lokale werking. In een tiental steden verspreid over heel België organiseren kleine groepjes vrijwilligers regelmatig activiteiten. Ook zij worden zoveel als mogelijk betrokken bij het interne bestuur van de v.z.w. Het werkelijke organigram van de v.z.w. ziet er dan schematisch als volgt uit :

6 Volgens Van Dale's Hedendaags Nederlands is ideologie "het geheel van ideeën dat ten grondslag ligt aan een wijsgerig stelsel, m.b.t. hun maatschappelijke of politieke strekking" – cf. http://www.vandale.nl/opzoeken/woordenboek/?zoekwoord=ideologie.



Afbeelding 3 : Schema praktische structuur der vzw

2.3.2. Praktische taakverdeling in de vzw

Als we dan ook een organigram willen maken van de taakverdeling zoals ze er in de praktijk uitziet, dan kunnen we niet anders dan vertrekken van deze werkelijkheid. Daarin kunnen we eerst en vooral volgende taken onderscheiden :

- algemene secretariaatstaken (papieren briefwisseling, e-mail en dergelijke)
- beheer van de financiële rekening, betaling van facturen, inning van de leden-bijdragen en andere inkomsten
- beheer van de (occasionele) subsidiedossiers
- beheer van de infrastructuur
- beheer van de leden en contacten
- ontwerpen en invullen van jaarlijks werkingsthema
- opmaak van een tijdschrift (Nederlands- en Franstalig)
- beheer van de websites (Nederlands- en Franstalig)

- opmaak van promotiemateriaal (Nederlands- en Franstalig)
- maken en monteren van het audiovisueel materiaal
- uitgeven van didactisch materiaal (boeken, brochures, presentaties, DVD's, enz.)

Opvallend is dat in dit lijstje geen sprake is van het bijhouden van een boekhouding. Dat komt omdat de boekhouding uitbesteed wordt.

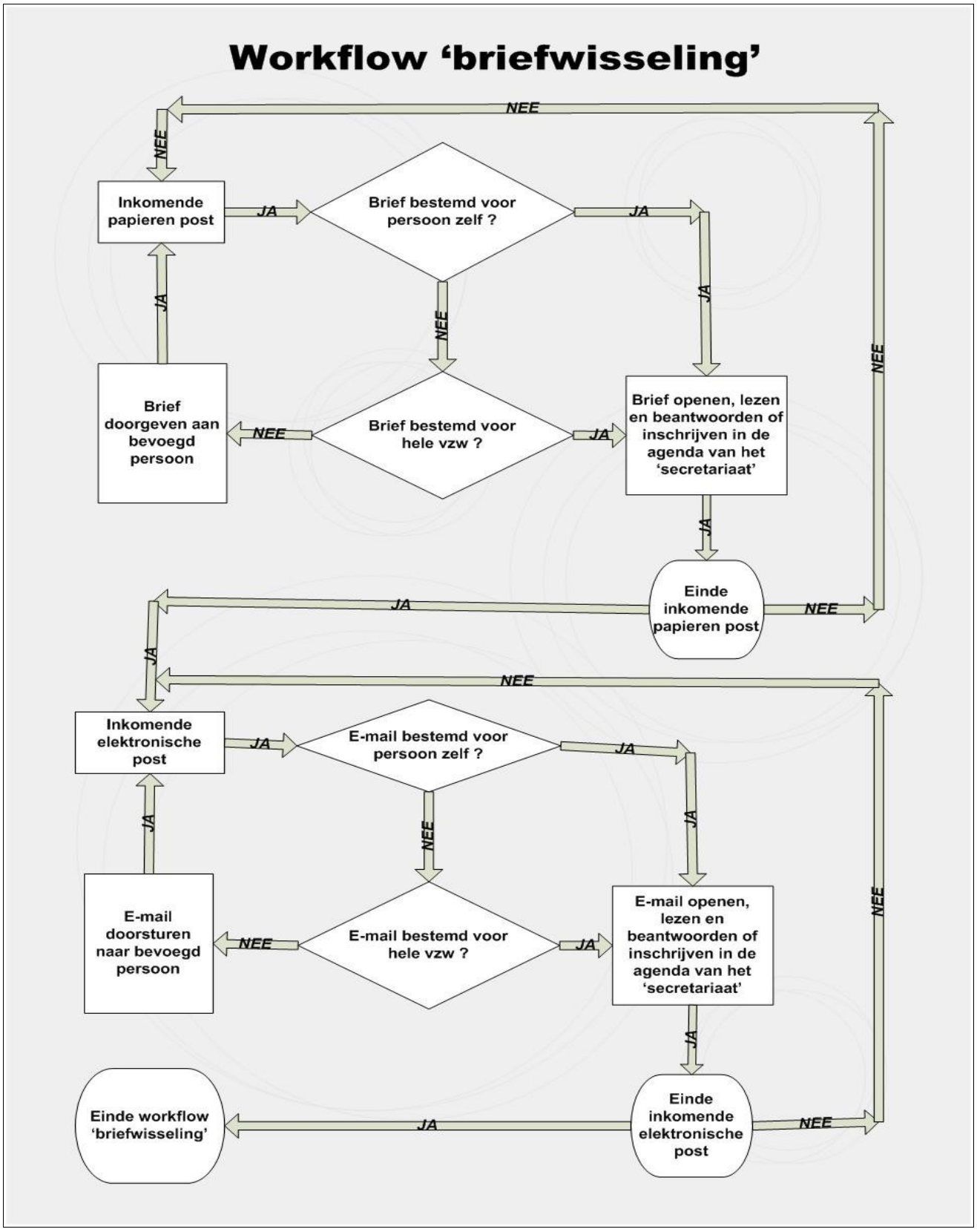
Verder bordurend op de hiervoor gemaakte opsomming kunnen we nu proberen een aantal processen (ook wel 'workflows' genoemd) in beeld te brengen. We maken daarbij een onderscheid tussen intern en extern gerichte processen :

2.3.2.1. Intern gerichte processen

2.3.2.1.1. Workflow 'briefwisseling'

De workflow 'briefwisseling' omvat relatief eenvoudige taken, die zowel slaan op de inkomende papieren post als op de elektronische post (i.e. de verwerking van e-mails). De eerste persoon die op een dag de lokalen van de vzw betreedt, neemt de papieren post uit de brievenbus en kijkt één voor één na of het gaat om brieven die aan hem persoonlijk gericht zijn. Is dat het geval, dan opent hij de brief, leest hem en beantwoordt hem indien nodig. Is de brief niet ter zijner attentie geadresseerd, dan kijkt hij of het gaat om een brief gericht aan de gehele vzw. Is dat het geval, dan opent hij de brief, leest hem en neemt een gepaste beslissing (in casu onmiddellijk beantwoorden, doorgeven aan een meer competent persoon of klaarleggen voor de eerstvolgende vergadering van het secretariaat, waar dan collectief een beslissing kan genomen worden). Gaat het echter om een brief die specifiek gericht is ter attentie van één der medewerkers, dan opent hij de brief uiteraard niet, maar bezorgd hem rechtstreeks aan de betrokkene.

Schematisch kunnen we deze workflow als volgt weergeven :


Workflow 'briefwisseling'
Inkomende papieren post
Brief bestemd voor persoon zelf ?
Brief bestemd voor hele vzw ?
Brief doorgeven aan bevoegd persoon
Brief openen, lezen en beantwoorden of inschrijven in de agenda van het 'secretariaat'
Einde inkomende papieren post
Inkomende elektronische post
E-mail bestemd voor persoon zelf ?
E-mail bestemd voor hele vzw ?
E-mail doorsturen naar bevoegd persoon
E-mail openen, lezen en beantwoorden of inschrijven in de agenda van het 'secretariaat'
Einde inkomende elektronische post
Einde workflow 'briefwisseling'
JA
NEE

Afbeelding 4 : Schema van de workflow ‘briefwisseling’

2.3.2.1.2. Workflow ‘rekeningbeheer’

Het beheer van de rekeningen verloopt iets ingewikkelder. Binnen de v.z.w. is er slechts één persoon die verantwoordelijk is voor het geheel van de financiën. Dat is ook logisch, want anders zou niemand zich er ook echt voor verantwoordelijk voelen. Ook is het zo dat de controle van slechts één financiële verantwoordelijke veel transparanter en makkelijker is. De workflow ‘rekeningbeheer’ is dan ook de workflow van slechts één persoon. Concreet bestaat deze workflow eruit dat de financiële verantwoordelijke tenminste één keer per week de voor hem bestemde papieren post doorneemt. Hij filtert daaruit de rekeninguittreksels en de te betalen facturen. Vervolgens start hij een computer op (tenzij er al één opgestart is, natuurlijk). Dan kijkt hij nogmaals de rekeningen oppervlakkig na middels elektronisch bankieren. Daarna verricht hij elektronisch de betaling der facturen. Vervolgens start hij een Excel-template op voor het inbrengen van een overzicht der inkomsten. Hiertoe gebruikt hij de rekeninguittreksels, waarbij hij de inkomsten groepeert in rubrieken zoals ‘subsidies’, ‘lidgelden’ (per afdeling), ‘steun’ (per afdeling) en ‘diversen’ (voor al wat niet onder een andere categorie valt). In hetzelfde template brengt hij eveneens de uitgaven in. Dit gebeurt gedetailleerd, wat nodig is voor de controle door de subsidiërende instanties. In de laatste week van de maand stuurt hij dit ingevulde template via e-mail door naar de boekhouder. Van dit doorgemailde template maakt hij ook een geprinte hardcopy.

2.3.2.1.3. Workflow ‘infrastructuur’

Net zoals voor de financiën is er ook voor het beheer van de infrastructuur slechts één eindverantwoordelijke. Het voordeel hiervan is vooral dat iedereen weet wie aan te spreken ingeval van klachten of verzuchtingen met betrekking tot de infrastructuur. Het is niet zo dat de infrastructuurverantwoordelijke een regelmatig werkritme hanteert. Gedurende heel het jaar ontvangt hij voortdurend klachten, opmerkingen, verzuchtingen, ideeën en voorstellen van alle gebruikers van de infrastructuur der vzw. Zijn werkterrein is dan ook zeer uitgebreid. Het beslaat zowel het uitzicht (bijvoorbeeld het plannen en (laten) uitvoeren van schilderwerken), het structureel onderhoud (bijvoorbeeld het (laten) herstellen van het dak), als de uitrusting van de lokalen. Die uitrusting kan heel breed opgevat worden. Er valt zowel het voorzien in kantoormeubilair onder, het beheer van het computerpark en alle randapparatuur, de archiefbenodigdheden, vergadertafels en –stoelen, didactisch materiaal (bijvoorbeeld *whiteboards*, *beamer*, enz.) en meer doordeweekse zaken – die meer op comfort gericht zijn – zoals bijvoorbeeld het aankopen van een nieuwe koffiezetmachine.

Het is moeilijk een dergelijk, uiterst gevarieerd takenpakket samen te vatten in een schema, temeer daar er geen tijdslimieten aan gebonden zijn. Concreet wordt het aan de verantwoordelijke overgelaten om zelf in te schatten wanneer het aantal klachten, opmerkingen, verzuchtingen, ideeën en voorstellen een zogenaamde kritische massa bereikt hebben.[7] Wanneer dat punt bereikt is, gaat de infrastructuurverantwoordelij-

7 In het jargon van de vzw spreekt men – verwijzend naar de Duitse filosoof Hegel - over *“het omslaan van de kwantiteit (der klachten, opmerkingen, verzuchtingen, ideeën en voorstellen), in kwaliteit”.*

ke over tot daden. Eerst overlegd hij met de financiële verantwoordelijke om te bekijken wat de financiële marge is. Op grond daarvan filtert hij uit de (soms grote) hoeveelheid klachten, opmerkingen, verzuchtingen, ideeën en voorstellen datgene uit wat hem haalbaar en noodwendig lijkt. Vervolgens doet hij aan prospectie. Er zijn daarvoor geen vooropgestelde regels. Hij kiest dus zelf of hij prospecteert via internet, door bezoeken aan winkels of door her en der raad te vragen. Wel kan hij niet anders dan zich beperken tot officiële verkopers. De vzw is immers gebonden aan allerlei toezicht- en controlemechanismen en kan zich niet permitteren betrapt te worden op het gebruik van schemerige *deals*. Na de prospectie werkt hij een voorstel uit dat hij indient bij het secretariaat. Indien zijn voorstel daar wordt goedgekeurd – eventueel na aanpassingen – wordt het voorgelegd aan de vrijwilligersgroep. Indien deze akkoord gaat (wat veelal een formaliteit is), plaatst de infrastructuurverantwoordelijke de nodige bestellingen en maakt hij de nodige afspraken. Dit gebeurt al naar het geval telefonisch, per brief of per e-mail. De factuur wordt vervolgens opgestuurd naar de vzw, waarna de financiële verantwoordelijke instaat voor de betaling ervan. In sommige gevallen echter gaat de infrastructuurverantwoordelijke persoonlijk naar de leverancier om daar het desbetreffende goed op te halen. Hij betaalt dan zelf onmiddellijk de rekening en brengt de factuur binnen bij de financiële verantwoordelijke, die hem dan terugbetaald.

2.3.2.1.4. Workflow ‘subsidiedossiers’

De workflow subsidiedossiers begint steevast met een brainstorming in het secretariaat op grond van het jaarlijkse werkingsthema. In functie van dat werkingsthema wordt het nut van een eventuele subsidie afgewogen aan de extra werklast, die het aanvragen van subsidies nu eenmaal met zich meebrengt. Vervolgens wordt een rudimentair schema van de subsidieaanvraag opgesteld (wie dat doet wordt in de schoot van het secretariaat in onderling overleg bepaald). Daarna tracht men de goedkeuring te verkrijgen van de groep vrijwilligers (met raadpleging van de lokale afdelingen). Daarna wordt door de eerder aangestelde tijdelijke verantwoordelijke een formeel besluit voor de Algemene Vergadering opgemaakt. Met dat voorstel tot besluit worden alle leden van de Algemene Vergadering op de eerstvolgende vrijwilligersvergadering uitgenodigd. Op die vrijwilligersvergadering (formeel omgevormd tot Algemene Vergadering) wordt het voorstel bediscussieerd en goedgekeurd (eventueel na aanpassingen). Het voorstel wordt dan (binnen de wettelijke tijdslimiet) verzonden naar de subsidiërende overheid (in het geval van deze vzw is dat steevast de Franse Gemeenschap). De tijdelijke verantwoordelijke, samen met de financiële verantwoordelijke en eventueel bijkomende geinteresseerden wonen dan de nodige toelichtings- en andere vergaderingen met de subsidiërende overheid bij. Eventueel wordt het voorstel aangepast aan bijkomende eisen van de subsidiërende overheid. Uiteindelijk wordt dan het definitief voorstel ingediend. Dit gebeurt door de tijdelijke verantwoordelijke per aangetekend schrijven. Indien er een positief antwoord komt van de subsidiërende overheid, dan wordt er een krediet aangevraagd bij de bank (dit in afwachting van de ontvangst der subsidies, die steeds ongeveer een jaar later gestort worden). Daarna worden de beoogde uitgaven gedaan. Indien er echter een negatief antwoord wordt gegeven, dan wordt de planning voor het uitwerken van het werkingsthema volledig herzien in functie van de veel beperkter voorhanden zijnde financiële middelen.

Een speciale situatie doet zich voor wanneer het gaat om het subsidiëren van een (tijdelijke) arbeidskracht. Dan dienen alle nodige stappen gezet te worden om dit volgens de wettelijke regels te kunnen doen. Dit wil zggen dat er door de tijdelijke verantwoordelijke contact wordt gelegd met het sociaal secretariaat, waarna er formeel een vacature uitgeschreven wordt. Meestal is er intern wel een kandidaat voorhanden, die dan ook een schriftelijk contract voorgelegd krijgt, waarna hij formeel wordt in dienst genomen.

2.3.2.1.5. Workflow ‘ledenbeweging’

Hier is enige toelichting noodzakelijk. De leden worden immers aangebracht door en in de lokale afdelingen. Zij geven wel door aan het nationale secretariaat wie er precies lid is en sinds wanneer. De leden zelf betalen een vrije bijdrage – waarbij nadruk wordt gelegd op de ‘ideologische’ keuze dat de sterkste schouders de zwaarste lasten dragen. Beter gegoede leden betalen dan ook (soms aanzienlijk) meer lidgeld dan meer armlastige leden. Het lidgeld wordt niet rechtstreeks gestort in de nationale kas, maar wel in de kas der lokale afdeling. Elke afdeling stort 70 % van de ontvangen lidgelden door naar ‘nationaal’. De financiële verantwoordelijke heeft dan ook het beste zicht op de evolutie van het ledenaantal en van de inkomsten die daaruit voortspruiten.

Indien de financiële verantwoordelijke vaststelt dat er een negatieve evolutie is in de globale ontvangsten uit ledenbijdragen, dan gaat hij eerst na hoe dat komt (bijvoorbeeld herziening naar beneden van hun bijdrage door een deel der leden; afhaken van een deel der leden;...).

Als blijkt dat het ledenaantal zelf daalt, legt de financiële verantwoordelijke een voorstel voor aan het secretariaat om daar iets aan te doen. Dit voorstel komt er steevast op neer de afdelingen aan te sporen nieuwe leden te maken en oude, afgehaakte leden opnieuw te doen aansluiten. Deze oproep wordt vervolgens voorgelegd aan de vrijwilligersgroep’, waarna de oproep ook nog eens per e-mail aan alle afdelingen wordt bezorgd. Ook alle individuele leden ontvangen deze oproep. Tot nu toe leidde dit naar verluidt steeds tot positieve resultaten.

Als echter het ledenaantal stabiel blijft, maar de inkomsten uit lidgelden dalen of geen gelijke tred houden met de stijging der levensduurte, dan legt de financieel verantwoordelijke een voorstel voor aan het secretariaat voor een zogenaamde financiële campagne. Afhankelijk van de ernst der toestand kan dit voorstel verschillende vormen aannemen, die ook gecombineerd kunnen worden. In eerste instantie kan het gaan om een overtuigingscampagne naar individuele leden toe om hun persoonlijke bijdrage te verhogen. Deze campagne wordt gevoerd door middel van een e-mail aan alle leden. Soms wordt er ook wel eens overgegaan tot overreding via telefonische en/of persoonlijke contactname. In dit laatste geval worden niet alle leden gecontacteerd, maar enkel diegene die over hogere inkomsten beschikken. In de regel zijn dat de al wat oudere leden. Als dat niet voldoende is, kunnen de afdelingen verzocht worden om – al dan niet tijdelijk – meer af te dragen aan ‘nationaal’ dan de normale 70 % van de ledenbijdragen. Is ook dat onvoldoende, dan dringt zich een algemene financiële campagne op (bijvoorbeeld onder de vorm van de verkoop van steunkaarten). Is het probleem erg acuut, dan worden de afdelingen gevraagd om vanuit hun lokale

kas bij te springen. Dit verzoek wordt gedaan per e-mail. In elk geval wordt dergelijke financiële campagne slechts gelanceerd na goedkeuring ervan door de stuurgroep.

2.3.2.1.6. Workflow ‘werkingsthema’

Ook deze workflow start met brainstormen in de schoot van het secretariaat. De resultaten van deze brainstorming worden op papier gezet door de verslaggever van de vergadering (de rol van verslaggever zelf wordt vervuld door middel van een beurtrol). Daarna tracht men goedkeuring te verkrijgen van de groep vrijwilligers (met raadpleging lokale afdelingen). Vervolgens wordt een formeel besluit opgemaakt voor de Algemene Vergadering. Daarna worden alle leden van de Algemene Vergadering per e-mail op de eerstvolgende vrijwilligersvergadering uitgenodigd. Op die vrijwilligersvergadering (formeel omgevormd tot Algemene Vergadering) wordt het voorstel bediscussieerd en – na eventuele aanpassingen – goedgekeurd. Het secretariaat zoekt dan gepaste inleiders voor de vormingsavonden en/of –weekends. Als de beschikbaarheid van deze inleiders in orde is, dan zoekt het ‘secretariaat’ locaties voor de vormingsavonden en/of –weekends. Tenslotte wordt het het geheel overgemaakt aan de workflow promotie.

2.3.2.2. Extern gerichte processen

2.3.2.2.1. Workflow ‘tijdschriften’

De vzw zorgt voor de publicatie van twee tijdschriften : een Nederlands- en een Franstalig, die qua inhoud niet identiek zijn aan elkaar. De frequentie van deze tijdschriften is driemaandelijks.

Voor het maken van deze tijdschriften bestaat er een maandelijkse (open) redactievergadering, die is samengesteld uit geïnteresseerden afkomstig uit de groep vrijwilligers, eventueel aangevuld met vrijwilligers uit de lokale afdelingen. Deze redactievergadering bepaalt de inhoud van het volgende nummer en verdeelt het aantal weerhouden artikels over het benodigde aantal vrijwillige auteurs en/of vertalers. Niet alle artikels worden immers zelf geschreven, sommige worden overgenomen uit bevriende en/of gelijkgezinde, buitenlandse publicaties en moeten dan vertaald worden.

Op de eerstvolgende vergadering controleert de redactievergadering het aantal reëel binnengekomen artikels en neemt gepaste maatregelen (bijvoorbeeld : te weinig artikels, dus meer nood aan vertalingen, dus meer vertalers nodig). Eén maand voor publicatie worden alle binnengekomen artikels gebundeld en overgemaakt aan de layout-verantwoordelijke. Zij zet alle artikels om van het tekstverwerkingsformaat waarin ze zijn opgesteld naar een zuiver tekstformaat. Vervolgens verzorgt ze de opmaak van het nieuwe nummer. Ze gebruikt daartoe een door een bevriende, professionele firma gemaakt template, hetgeen haar werk fel vergemakkelijkt. Daarna wordt de proefversie via e-mail doorgestuurd naar een aantal vrijwilligers die instaan voor de *proofreading*. Na het verbeteren van de fouten wordt een definitieve versie gemaakt die via e-mail wordt doorgestuurd naar de (externe, commerciële) drukkerij. Na het drukken wordt de voorraad opgehaald bij de drukkerij. De voltallige vrijwilligersploeg staat dan in voor de verzending.

Hiertoe drukken zij eerst de etiketten met de adressen der abonnees. Daartoe moeten ze echter eerst de voor etiketten voorziene printer aansluiten op een computer, waarbij ze de normaal aan deze computer hangende printer eerst moeten losmaken. Deze procedure wordt gevolgd, omdat alle andere aanwezige printers te warm worden, waardoor de etiketten loskomen van hun papieren drager en zich hechten aan het printermechanisme. Daarna worden de etiketten op grote enveloppen gekleefd, waarna de tijdschriften vervolgens in die enveloppen worden gestoken. Dan wordt het verzendingsformulier voor de post opgemaakt. Hierdoor vermijdt men het één voor één moeten kleven van postzegels of elke afzonderlijke envelop en bespaart men bovendien op de verzendingskosten. Vervolgens wordt de verzameling enveloppen naar de post gebracht, die dan instaat voor de bezorging ervan. Tenslotte wordt de rekening van de post ontvangen en vereffend door de financiële verantwoordelijke.

2.3.2.2.2. Workflow ‘website’

Naast de twee tijdschriften staat de vzw ook in voor de inhoud en de vormgeving van twee websites (één Nederlands- en één Franstalige). De vormgeving daarvan brengt weinig werk met zich mee, aangezien de vzw in het verleden veel tijd gestoken heeft in het op punt zetten van een zogenaamd *content management system*. Over het algemeen is men zeer tevreden over het resultaat daarvan, dus wijzigingen dringen zich daar niet op. De vzw kan zich dan ook concentreren op de inhoud van de website zelf.

De workflow zelf hangt nauw samen met de die van de workflow tijdschriften. Het gaat immers om quasi dezelfde inhoud. De door de redactie gemaakte of vertaalde artikels, die door de layout-verantwoordelijke werden omgezet naar een gewoon tekstformaat, worden door de web-verantwoordelijke ingelezen in het *content management system* en daar waar nodig voorzien van de benodigde opmaak. Vervolgens wordt de site gepubliceerd (hetgeen een onderdeel is van het *content management system*.

2.3.2.2.3. Workflow ‘promotiemateriaal’

De activiteiten van de vzw volgen een jaarlijks wederkerend stramien (enkel de inhoud ervan wijzigt in functie van het jaarlijkse werkingsthema). Dit stramien ziet er als volgt uit : tijdens het eerste weekend van september is er een zogenaamde ‘zomerschool’, die duurt van vrijdagavond tot zondagnamiddag. In februari, tijdens het weekend dat valt in de krokusvakantie, is er een zogenaamde winterschool. Het hele jaar door worden er ook vormingsreeksen georganiseerd. De inhoud daarvan wordt nationaal vastgelegd, net zoals de sprekers, maar de avonden zelf gaan door in de lokale afdelingen. Daarnaast kunnen de lokale afdelingen ook zelf nog eens bijkomende activiteiten organiseren, die zowel een vormend als een meer ontspannend karakter kunnen aannemen (of een combinatie van beide). Soms organiseert de vzw ook wel eens studiedagen of colloquia (meestal worden daarvoor subsidies aangevraagd). Ten behoeve van jongere leden ondersteunt de vzw ook een tijdens de zomer gehouden internationaal jongerenkamp.

Voor dit alles is er natuurlijk promotiemateriaal nodig. Het maken van dit materiaal volgt voor een deel hetzelfde stramien als het maken van tijdschriften en website. Na-

dat het desbetreffende onderdeel (bijvoorbeeld de vormingsavonden) werd uitgewerkt in het secretariaat en goedgekeurd door de vrijwilligersgroep, wordt een verantwoordelijke aangesteld. Die zorgt – al dan niet een beroep doende op anderen om hem te helpen – voor de concrete invulling van de desbetreffende activiteit. Dit wil zeggen dat hij de benodigde sprekers contacteert en vastlegt. Daarna maakt hij een rudimentaire affiche en een promotiefoldertje. Eventueel zoekt hij er ook illustraties bij. De platte tekst hiervan bezorgt hij aan de layout-verantwoordelijke die er iets moois van maakt. Dit gaat vervolgens via e-mail naar de drukker die het drukt. Na afhaling wordt het materiaal verspreid over de afdelingen, die instaan voor de lokale verspreiding ervan. De financiële verantwoordelijke staat in voor de betaling der factuur.

2.3.2.2.4. Workflow ‘audiovisueel’

De vzw heeft het geluk in de rangen van secretariaat een medewerker te hebben die uitgebreide ervaring heeft met het maken en monteren van audiovisueel materiaal. Beroepshalve was deze medewerker werkzaam als journalist, waarbij hij zelf moest instaan voor het (doen) filmen van reportages, die hij vervolgens ook zelf diende te monteren tot een bruikbare uitzending. De vzw maakt dankbaar gebruik van deze expertise. Zo werd onder meer, met behulp van subsidies, reeds een dubbel-DVD geproduceerd over leven en werk van professor Ernest Mandel[8]. De workflow van dit proces hangt uiteraard nauw samen met de aard van het te maken audiovisueel materiaal (zo is het soms noodzakelijk dat er daarvoor buitenlandse reizen gemaakt worden). Eén en ander hangt dan ook sterk af van de voorhanden zijnde financiële middelen. Zonder subsidies is het meestal onmogelijk om dit proces tot een goed einde te brengen. Concreet start dit proces steeds met een voorstel door de filmverantwoordelijke (i.e. de bewuste medewerker). Dit voorstel behelst niet alleen een inhoudelijke beschrijving, maar ook een (rudimentaire) raming van de voorziene kostprijs. Daarna laat de financiële verantwoordelijke er zijn licht over schijnen. Vervolgens doorloopt het voorstel de workflow van het proces subsidiedossiers (zie hoger). Als dat allemaal goed afloopt, dan gaat de filmverantwoordelijke aan het werk. Hij filmt daar waar nodig, neemt (gefilmde) interviews af, en dergelijke meer. Als al het benodigde materiaal ‘ingeblikt’ is, zet hij het (digitaal) gefilmde materiaal over op de harde schijf van de meest krachtige computer in het nationaal lokaal. Op die computer doet hij ook de montage en de ondertiteling. Uiteindelijk leidt dit tot het maken van twee master-versies : één voor de Nederlandstaligen en één voor de Franstaligen. Die master-versies worden vervolgens gebrand op een DVD, die dan bij een externe firma vermenigvuldigd wordt. Daarna worden de DVD’s verspreid over de afdelingen en meegenomen naar alle publieke activiteiten van de vzw waar ze verkocht worden. In de tijdschriften en op de websites wordt er uiteraard ook reclame voor gemaakt.

2.3.2.2.5. Workflow ‘uitgave didactisch materiaal’

In de loop der jaren heeft de vzw een aanzienlijke hoeveelheid gedrukt materiaal uitgegeven. Het grootste deel hiervan is niet in digitale vorm voorhanden. Het gaat on-

8 Ernest Mandel was onder andere professor aan de Vrije Universiteit Brussel. Voor een over-zicht van de betekenis van zijn leven en werk, zie http://nl.wikipedia.org/wiki/Ernest_Mandel.

der meer om artikelen, brochures en boeken. Eerder dan elk jaar opnieuw het warme water te willen uitvinden, kiest de vzw ervoor om dit 'oude' materiaal zoveel mogelijk te reproduceren (weliswaar aangevuld met een gepast, geactualiseerd voor- of nawoord). Zo werden de afgelopen jaren bijvoorbeeld twee stevige brochures samengesteld met teksten enerzijds omtrent het nationaliteitenvraagstuk in België en anderzijds omtrent het Israëlisch-Palestijns conflict. Ook enkele boeken (onder andere van Ernest Mandel) werden in een facsimile herdruk heruitgegeven. Het gaat daarbij om een relatief makkelijke en goedkope manier om ouder materiaal toch niet verloren te laten gaan en nuttig te (her)gebruiken. Bijkomend (theoretisch) voordeel is ook dat de aldus gereproduceerde publicaties meteen ook in een digitale versie voorhanden zijn, wat bijvoorbeeld nuttig kan zijn indien de noodzaak aan een herdruk zich zou voordoen.

Deze workflow bestaat er eerst en vooral uit dat een selectie moet worden gemaakt uit het uitgebreide papieren archief van de vzw. Een specifieke verantwoordelijke is er niet voor deze workflow. Dat hangt immers mee af van het onderwerp. Dat onderwerp zelf hangt samen met het jaarlijks vastgestelde werkingsthema. Voelt iemand onder de vrijwilligers zich geroepen, dan zoekt hij zelf in het archief naar geschikt materiaal. Vindt hij dat, dan maakt hij de nodige scans. Daartoe maakt hij gebruik van de scanner die onderdeel uitmaakt van het *multifunctional device* dat zich in het lokaal bevindt. Daarna print en verknipt hij de aldus bekomen scans tot een nieuwe layout. Die scant hij opnieuw in, waarna het resultaat wordt opgeslagen in een voor de drukkerij bruikbaar formaat. Dit bestand print hij enkele keren uit, waarna hij zijn voorstel voorlegt aan het secretariaat. Als dat zijn fiat geeft – na raadpleging van de financiële verantwoordelijke –wordt het resultaat per e-mail doorgestuurd naar de drukker. De aldus bekomen gedrukte exemplaren worden afgehaald en maken vanaf dan deel uit van de stock der vzw. Ze worden deels verspreid over de afdelingen en deels meegenomen naar alle publieke activiteiten van de vzw waar ze verkocht worden. In de tijdschriften en op de websites wordt er uiteraard ook reclame voor gemaakt.

2.4. Conclusies van de functionele analyse

De organisatiestructuur en de daarmee samenhangende manier van werken van de vzw wordt gekenmerkt door een hoge mate van informaliteit. Sterke nadruk wordt dan ook gelegd op het belang van het individueel engagement en de daaruit voortvloeiende zelf opgenomen verantwoordelijkheid. Dit model wordt ook weleens omschreven als steunend op het beginsel van *empowerment*[9].

9 Mogelijke definities van *empowerment* zijn :

- *"A series of actions designed to give employees greater control over their working lives. Businesses give employees empowerment to motivate them according to the theories of Abraham Maslow and Fredrick Herzberg.*
- *To invest with power or give authority to complete. To empower employees.*
- *Being allowed to make decisions and take actions on your own, apart from management.*
- *A contract that involves the delegation of authority and commitment to an individual to act or authorize actions to be taken, in exchange for the acceptance of responsibility and accountability to fulfill a defined objective. Used to increase an organizations responsiveness, effectiveness and efficiency without increasing the budget."*

De verschillende processen die het werk van de vzw mee vorm geven, zijn in hoge mate historisch gegroeid. Ze zijn als het ware organisch tot stand gekomen, dit wil zeggen dat ze voortvloeiden uit de onmiddellijke noodwendigheden, zonder dat er sprake was van veel planmatig inzicht of ingrijpen.

In de loop der tijden werd weliswaar een zekere hoeveelheid informaticamateriaal toegevoegd aan het patrimonium van de vzw. We kunnen echter stellen dat de nieuwe mogelijkheden die daarmee samenhangen slechts tot op zekere hoogte gebruikt worden. Eigenlijk is het zo dat de mogelijkheden der informatica werden *toegevoegd* aan de bestaande werkmethoden, zonder dat dit leidde tot een optimale rationalisering van die werkmethoden.

Voor een groot deel beperkt het gebruik van informatica zich tot e-mail (ter vervanging van de vroegere papieren post), zonder dat daarrond echter klare, duidelijke afspraken bestaan. In twee gevallen zorgde de informatica ervoor dat er nieuwe processen bijkwamen. Het gaat daarbij meer bepaald om de processen 'audiovisueel materiaal' en 'didactisch materiaal' (respectievelijk het maken van films en het (her)uitgeven van boeken en brochures).

Veel werkzaamheden gebeuren ook nog altijd in de lokalen der vzw, terwijl de informatica het nochtans mogelijk maakt dit te vermijden via toegang op afstand (*remote access*), wat in een aantal gevallen ook nuttig zou kunnen zijn.

Tenslotte doet de als noodzakelijk ervaren aanwezigheid van de financiële verantwoordelijke – zelfs in de planningfase ! – bij zo goed als elk werkingsproces, vermoeden dat de financiële marges van de vzw erg beperkt zijn.

Bron : http://www.bpmenterprise.com/dictionary/Empowerment-225.htm

Zie ook : GABOR, Andrea, *'The Capitalist Philosophers : The Geniuses of Modern Business – Their Lives, Times and Ideas'*, Random House Time Business Books, 2000. Meer bepaald pagina 222 : *"True empowerment begins with commitment."*

2. Technische Analyse

3.1. Inleiding

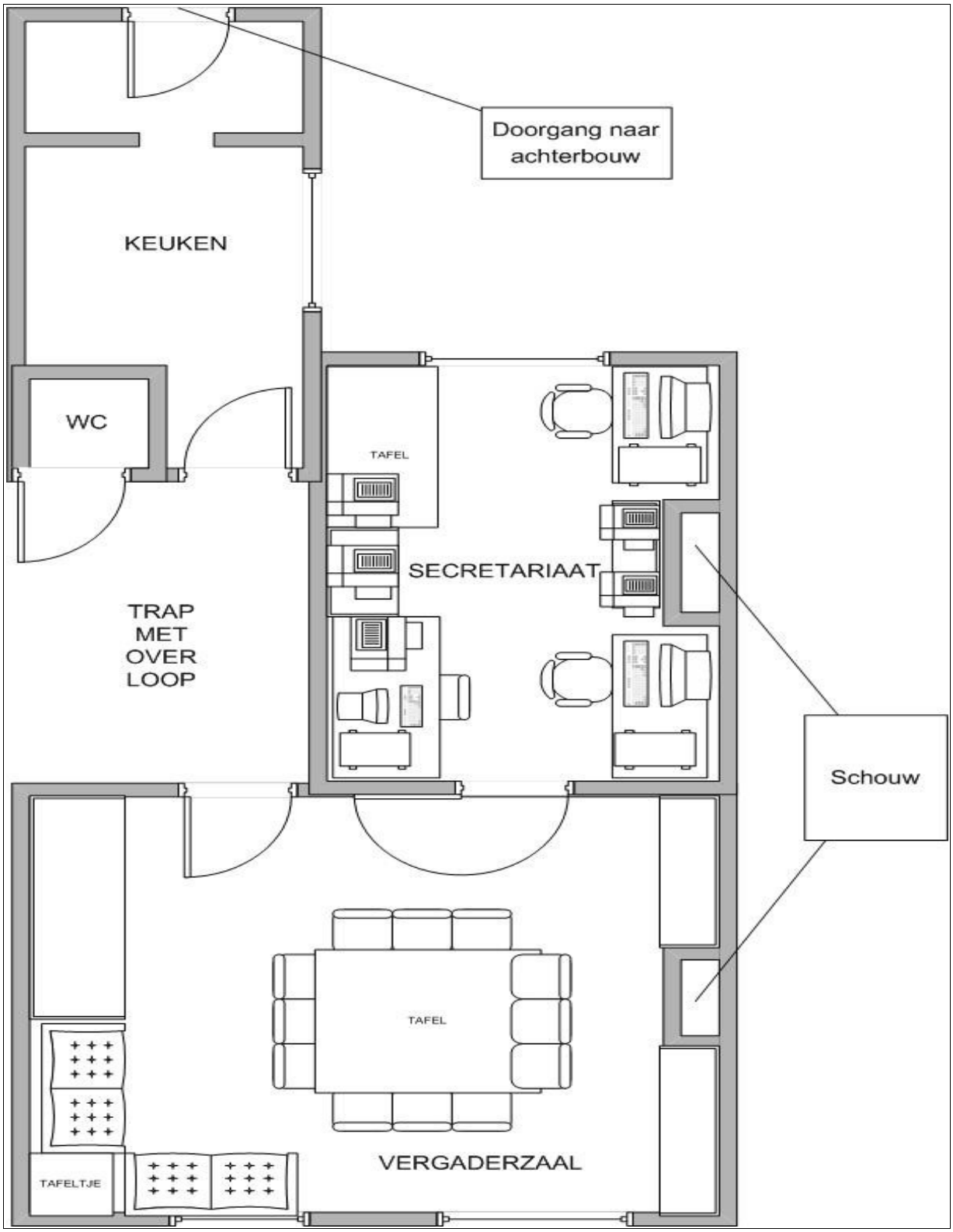
Nadat we in de functionele analyse hebben bekeken *wat* de verschillende activiteiten van de vzw precies behelzen en welke processen daarmee samenhangen, bekijken we in de technische analyse *hoe* dit alles precies in zijn werk gaat. We kijken daarbij eerst naar de aanwezige infrastructuur om ons daarna meer in detail te buigen over de aanwezige informaticamiddelen en hoe deze aangewend worden. Tenslotte schetsen we een – in onze ogen – meer optimale hypothese.

3.2. Plattegronden

De vzw is gevestigd te Brussel. Het gebouw bestaat uit een gelijkvloers appartement, een appartement op de eerste en tweede verdieping, een zolder en een bijkomende achterbouw van twee verdiepingen. Zelf huurt de vzw de eerste verdieping en de zolder. De zolder gebruikt ze als archiefruimte. We zullen ons dan ook beperken tot de lokalen op de eerste verdieping.

3.2.1. Algemene indeling

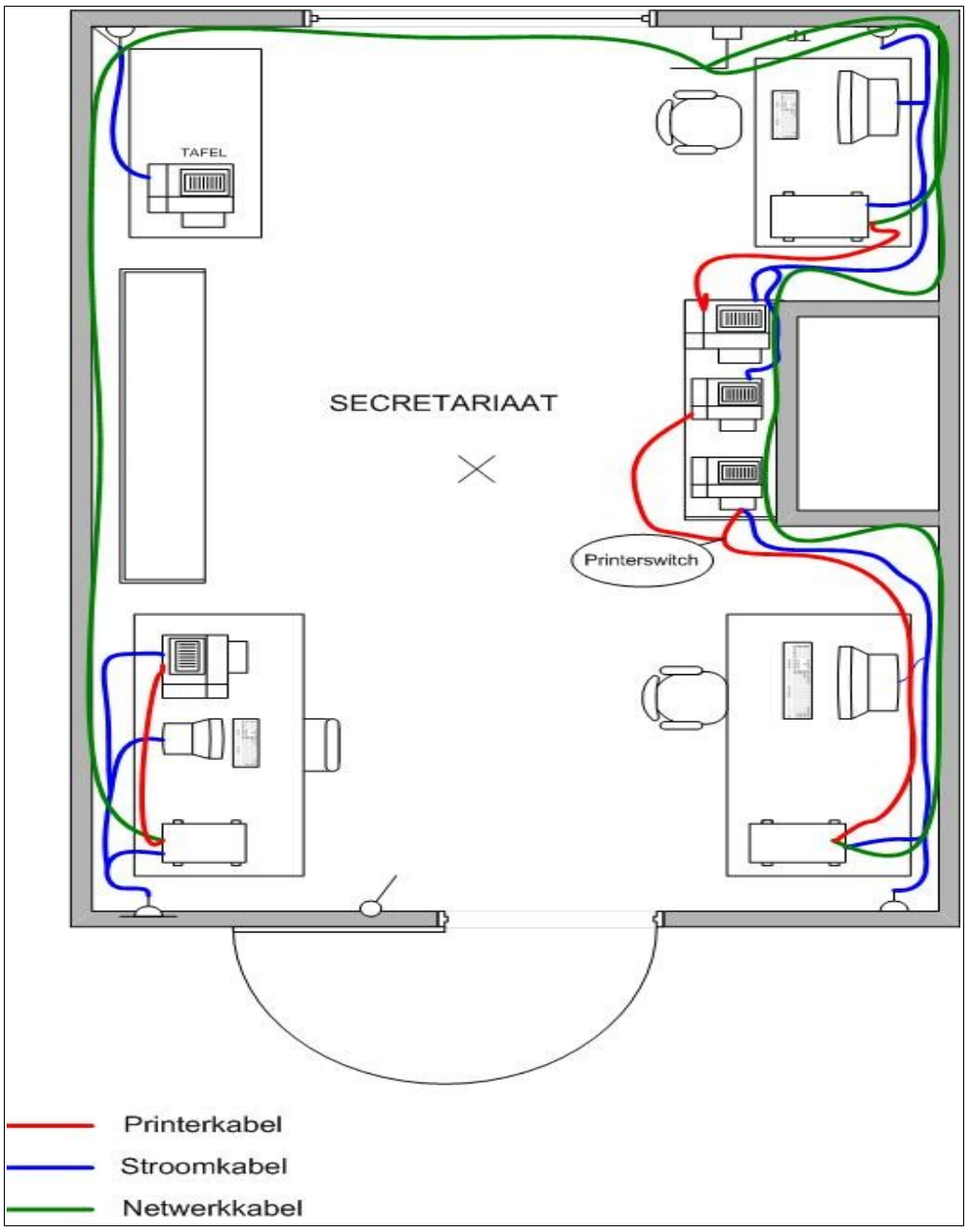
Deze eerste verdieping beslaat concreet vier ruimten, waarvan er feitelijk slechts drie gebruikt worden. Het gaat om een keukenruimte, die ook dienst doet als doorgang naar de eerste verdieping van de achterbouw. Het is deze ruimte die niet of slechts zeer sporadisch gebruikt wordt. Daarnaast is er het sanitair, dat uiteraard wel gebruikt wordt. Verder nog een ruime vergaderplaats en tenslotte het feitelijke secretariaat, waar ook de computers staan. Hieronder zien we een schema van de algemene onderverdeling :



Afbeelding 5 : Inrichting van de lokalen der vzw

3.2.2. Detail van de secretariaatsruimte

Over de keuken, het sanitair en de vergaderruimte gaan we ons niet verder buigen. Wel bekijken we in detail de secretariaatsruimte. Op onderstaand schema worden de (losliggende) kabelverbindingen aangegeven.

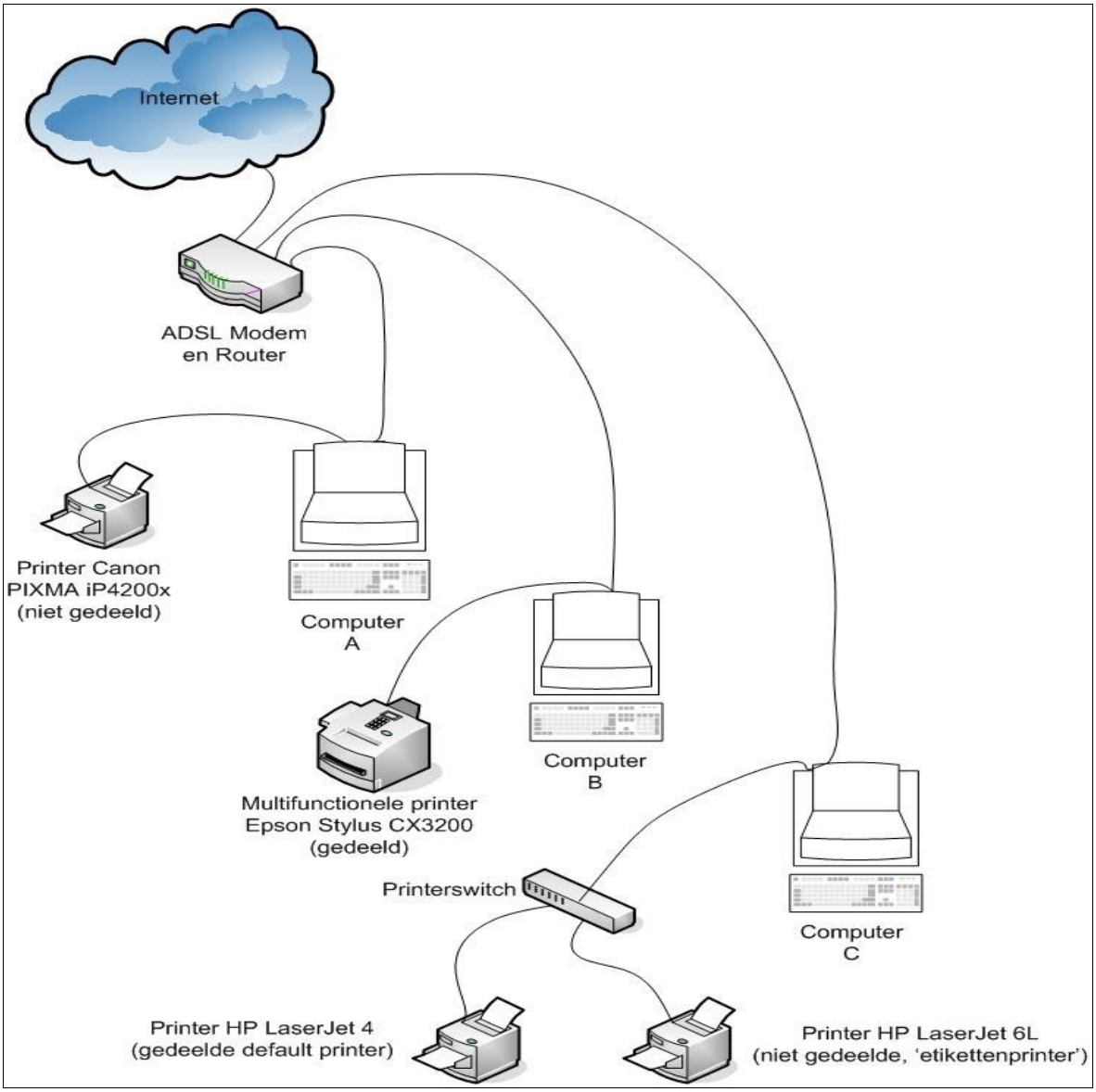


Afbeelding 6 : Kabels in de secretariaatsruimte

3.3. Beschrijving van het bestaande netwerk

3.3.1. Inleidend overzicht

We bekijken nu het aanwezige computernetwerk. Concreet bestaat dit uit 1 ADSL-modem annex router met wireless-functionaliteit, drie computers en vijf printers (waarvan één zogenaamde 'multi-functional'). Schematisch kunnen we dit als volgt voorstellen :



Afbeelding 7 : Schema van het aanwezige netwerk

Op dit fysieke netwerk worden alle computers opgenomen in één werkgroep. Er is dus geen Active Directory en bijgevolg ook geen gecentraliseerd beleid van welke aard ook. De naam van de werkgroep is simpelweg 'WORKGROUP'.

3.3.2. De aanwezige router

Tot voor kort was er geen router aanwezig. Verbinding met het Internet werd toen gemaakt door slechts één computer via een eenvoudige ADSL-modem. Dit stelde uiteraard problemen, omdat de andere twee computers nooit op het Internet konden en evenmin printers konden gebruiken die niet fysiek aangekoppeld waren aan de eigen parallelle of USB-poort. . Bovendien konden bezoekers met een laptop evenmin gebruik maken van netwerkfunctionaliteit. Als oplossing voor deze problemen werd recent een ADSL-modem/router geïnstalleerd.

De ADSL-modem/router is van het merk US Robotics, met als naam 'Wireless MAXg ADSL Gateway'[10]. Deze router is verbonden met Belgacom/Skynet als Internet Service Provider en beschikt over 4 netwerkinterfaces voor computers en over één printserver via een USB-poort voor het gebeurlijk aansluiten van een USB-printer, die dan gebruikt kan worden als netwerkprinter. Deze USB-poort wordt echter niet gebruikt. Daarnaast is de router ook uitgerust met wireless-functionaliteit. Intern bevat de router alle noodzakelijke netwerkservices, zoals daar zijn een DNS-server, een firewall en een DHCP-server. Bijkomende kan een VPN-server geïnstalleerd worden, wat echter niet gebeurd is. De router is slechts rudimentair geconfigureerd. Het wireless-verkeer verloopt volstrekt onbeveiligd. De WEP-encryptie, noch de WPA-encryptie werden ingesteld. De DHCP-server werd louter ingeschakeld, maar niet verder geconfigureerd, hetgeen betekent dat deze DHCP-server quasi onbeperkt IP-adressen uitdeelt aan elke computer die zich aanmeldt. MAC-adresbeperkingen werden niet ingeschakeld. Toegang tot de configuratie-interface van de router werd niet beveiligd.

3.3.3. Beschrijving van de aanwezige computers

Er zijn zoals gezegd drie computers aanwezig in het netwerk. Het gaat om computers van een onbestemd merk, die alle werden gemaakt door een kleine, zelfstandige computerbouwer. Ze zijn ook alle drie ongelijk van configuratie. Voor het gemak spreken we hierna over computer A, B en C.

3.3.3.1. Computer 'A'

De computernaam van deze computer is COMP-A. Deze computer beschikt over een Intel Pentium 4 processor met een snelheid van 3,20 GHz. De aanwezige hoeveelheid RAM-geheugen beslaat 512 MB. Qua harde schijfruimte beschikt deze computer over 202 GB. Met deze specificaties gaat het om de best uitgeruste van de drie computers. Hij wordt dan ook gebruikt voor de meest reken- en opslagintensieve werkzaamheden, zoals editering en mastering van DVD-films.

De harde schijf van COMP-A werd niet gepartitioneerd. Er is dus enkel een C-partitie, waarop zich zowel systeemsoftware, applicatiesoftware als databestanden bevinden. Geen enkele directory of folder wordt gedeeld over het netwerk (er zijn dus geen 'shares' ingesteld). Wie dus met de data van deze computer aan de slag wil, dient ook daadwerkelijk fysiek van deze computer gebruik te maken.

COMP-A beschikt over een op het moederbord ingebouwde netwerkkaart van 100Mbps, die zijn netwerkadres krijgt van de DHCP-server op de router. De Ether-

10 Zie http://www.usr-emea.com/products/pbroadbandproduct.asp?prod=bb9108a&loc=bene

netkabel tussen COMP-A en de router ligt los op de vloer, langsheen de muur. Deze kabel is maar net lang genoeg en komt soms los, zodat hij weer manueel moet worden aangesloten.

Aan deze computer is, via een USB-poort,, één printer verbonden. Deze printer is een kleureninktjet van het merk Canon, meer bepaald type ‘Canon PIXMA iP4200x’. Deze printer beschikt over twee manieren voor papierinvoer (onderaan een cassette en boven een 'rechte' invoer), wat het werken met dikker papier makkelijk maakt. Er zijn aparte cartridges voor de vier verschillende kleuren, met één losse, door alle kleuren te gebruiken printkop. Ook kunnen met deze printer CD’s (of DVD’s) bedrukt worden. Deze printer wordt niet gedeeld over het netwerk, dit om te verhinderen dat de dure kleureninkt te snel verbruikt zou worden.

Daarnaast kan deze computer ook gebruik maken van over het netwerk gedeelde printers, met name de HP LaserJet 4 op COMPC (zie verder) en de multifunctionele Epson Stylus CX3200 op COMPB (zie verder).

De monitor van deze computer is een ‘oude’ 15 inch CTX-monitor van het merk Targa. De grafische kaart is van het merk NVidia, met 128 MB videogeheugen aan boord.

De stroomkabel, net zoals de kabels naar de monitor, printer en router, vormen achter de bureautafel één zogenaamde kabel-spaghetti. De verschillende kabels dragen ook geen *tags*.

Op deze computer werd slechts één gebruikers-account aangemaakt, met de naam *user*. Deze beschikt over alle administrator-bevoegdheden. Aanloggen gebeurt automatisch, dus zonder paswoord. Van toegang-beveiliging is dan ook geen sprake.

Het besturingssysteem van COMP-A is Windows XP Home Edition met Service Pack 2, in Nederlandstalige versie. De automatische update-mogelijkheid werd ingeschakeld en werkt probleemloos, wat erop wijst dat het hier gaat om een legale versie van Windows.

3.3.3.2. Computer ‘B’

De computernaam van deze computer is COMP-B. Deze computer beschikt over een Intel Celeron processor met een snelheid van 2,40 GHz. De aanwezige hoeveelheid RAM-geheugen beslaat 224 MB. Qua harde schijfruimte beschikt deze computer over 80 GB. Het opstarten van deze computer gaat gepaard met veel, rammelend lawaai, wat doet vermoeden dat de harde schijf niet goed werd vastgemaakt aan het chassis. Deze computer wordt gebruikt voor algemene secretariaatstaken en voor het maken van lay-outs.

De harde schijf van COMP-B werd evenmin onderverdeeld in meerdere partities. Er is dus eveneens enkel een C-partitie, waarop zich zowel systeemsoftware, applicatiesoftware als databestanden bevinden. Evenmin worden directories of folders gedeeld over het netwerk. Wie dus met de data van deze computer aan de slag wil, dient eveneens daadwerkelijk fysiek van deze computer gebruikt te maken.

COMP-B beschikt over een op het moederbord ingebouwde netwerkkaart van 100Mbps, die zijn netwerkadres krijgt van de DHCP-server op de router. De Ethernet-kabel tussen COMP-B en de router ligt ook los op de vloer, langsheen de muur.

Aan deze computer is, via een USB-poort, één printer verbonden. Dit is de 'Epson Stylus CX3200', een zogenaamde *multifunctional*, bestaande uit een kleureninkjet-printer en een vlakbedscanner van A3-formaat. De printercomponent van de *multi-functional* wordt gedeeld over het netwerk (de scanner echter niet).

Daarnaast kan deze computer ook gebruik maken van de andere over het netwerk gedeelde printer, met name de HP LaserJet 4 op COMP-C (zie verder).

De monitor van deze computer is eveneens een 'oude' 15 inch CTX-monitor van het merk Targa. De grafische kaart is een S3 Graphics Pro Savage, aangesloten op de AGP-poort, met 64 MB videogeheugen aan boord.

De stroomkabel, net zoals de kabels naar de monitor, printer en router, vormen ook hier achter het verplaatsbare, op wieltjes gemonteerde computermeubel één kabelspaghetti. De verschillende kabels dragen evenmin *tags*.

Op deze computer werd ook slechts één gebruikersaccount aangemaakt, met de naam 'BELGATECH'. Deze beschikt eveneens over alle administrator-bevoegdheden. Aanloggen gebeurt automatisch, dus zonder paswoord. Van beveiligde toegang is ook hier geen sprake.

Het besturingssysteem van COMP-B is eveneens Windows XP Home Edition met Service Pack 2, deze keer in Franstalige versie. De automatische update-mogelijkheid werd ook hier ingeschakeld en werkt probleemloos, wat erop wijst dat het ook hier gaat om een legaal exemplaar van Windows.

3.3.3.3. Computer 'C'

De computernaam van deze computer is COMP-C. Deze computer beschikt over een Intel Pentium 4 processor met een snelheid van 2,40 GHz. De aanwezige hoeveelheid RAM-geheugen beslaat 256 MB. Qua harde schijfruimte beschikt deze computer eveneens over 80 GB. Ook deze computer wordt gebruikt voor algemene secretariaatstaken en voor het maken van lay-outs.

De harde schijf van COMP-C werd evenmin gepartitioneerd. Er is dus ook hier enkel een C-partitie, waarop zich zowel systeemsoftware, applicatie-software als databestanden bevinden. Evenmin worden directories of folders gedeeld over het netwerk. Wie dus met de data van deze computer aan de slag wil, dient eveneens daadwerkelijk fysiek van deze computer gebruikt te maken.

COMP-C beschikt over een netwerkkaart van het type Fast Ethernet SiS 900 met een snelheid van 100Mbps, aangesloten via een PCI-sleuf. Deze kaart krijgt haar netwerkadres eveneens van de DHCP-server op de router. De Ethernet-kabel tussen COMP-C en de router ligt eveneens los op de vloer, langsheen de muur.

Aan deze computer is, via de parallelle poort, een eenvoudige *printerswitch* verbonden. Aan deze schakelaar hangen twee printers, met name de eerder genoemde (en gedeelde) HP LaserJet 4, een zwart-wit laserprinter. De andere printer die aan deze schakelaar verbonden is, is een HP LaserJet 6L. Deze printer staat meestal niet aan en wordt enkel gebruikt wanneer er zelfklevende etiketten moeten worden afgedrukt (de HP LaserJet 4 blijkt voor die taak niet meer opgewassen, omdat hij te snel te warm wordt, waardoor de etiketten loskomen van hun drager en de printer zelf verstopt raakt met etiketten).

De monitor van deze computer is weer een ‘oude’ 15 inch CTX-monitor van het merk Targa. De grafische kaart is een NVidia GeForce 4 MX 440, aangesloten op de AGP-poort, met 64 MB videogeheugen aan boord.

De stroomkabel, net zoals de kabels naar de monitor, printer en router, vormen achter het verplaatsbare, op wieltjes gemonteerde computermeubel alweer één zogenaamde kabelspaghetti. De verschillende kabels dragen ook hier geen *tags*.

Op deze computer werd ook slechts één gebruikersaccount aangemaakt, met de naam *user*. Deze beschikt eveneens over alle administrator-bevoegdheden. Aanloggen gebeurt ook hier weer automatisch, dus zonder paswoord. Van beveiligde toegang is dan ook weer geen sprake.

Het besturingssysteem van COMP-C is Windows XP Professional met Service Pack 2, in Franstalige versie. De automatische updatemogelijkheid werd wel ingeschakeld, maar werkt slechts in beperkte mate, omdat Windows Genuine Advantage aangeeft dat het hier mogelijk gaat om een niet-legale versie.

3.3.4. Beschrijving van de aanwezige printers

Zoals gezegd beschikt de vzw over vijf printers, waarvan er echter slechts drie permanent ingeschakeld zijn. Deze printers zijn de volgende :

PRINTERNAAM	SOORT	POORT	GEBRUIK	GEDEELD ?
Canon i350	Eenvoudige ‘bubble jet’ kleureninkt-jetprinter	Parallel	Niet gebruikt	Niet gedeeld
Canon PIXMA iP4200x	- Hoogwaardige kleureninktjetprinter - Kan papier en CD/DVD’s bedrukken - beschikt over vier aparte kleur-cartridges met één losse, gedeelde printkop	USB	Gebruikt bij de productie van DVD’s en hun hoezen	Niet gedeeld
Epson Stylus CX3200 Multi-functional	- A3 kleuren vlakbedscanner - Hoogwaardige kleureninktjetprinter - beschikt over vier aparte kleur-cartridges met één losse, gedeelde printkop	USB	Enkel de scanner wordt veel gebruikt (bij het maken van lay-outs). De inkt-cartridges zijn niet allemaal aangevuld.	Gedeeld (enkel de printercomponent)
HP LaserJet 4	Zwart-wit laser-printer	Parallel (via printer-schakelaar)	Standaardprinter voor alle computers. Wordt dan ook voor bijna alles gebruikt.	Gedeeld (default printer)

PRINTERNAAM	SOORT	POORT	GEBRUIK	GEDEELD ?
HP LaserJet 6L	Zwart-wit laser-printer. Enkel gebruikt voor drukken van etiket-ten.	Parallel (via printer-schake-laar)	Enkel voor het drukken van eti-ketten.	Niet gedeeld

Tabel 1 : Soorten printers en hun eigenschappen

3.3.5. Beschrijving van de aanwezige software

Dat er geen enkel, al dan niet gecentraliseerd, beleid bestaat inzake het gebruik en het onderhoud van de computers blijkt overduidelijk uit de software, zoals die op de verschillende computers aanwezig is. In Appendix D geven we daarvan in tabelvorm een overzicht weer (tabel 2).

Uit die tabel blijkt vooral hoe rommelig het wel is. Het feit dat er niemand werkelijk verantwoordelijk is voor één (of meerdere) PC’s, leidt er blijkbaar toe dat iedere gebruiker de software van zijn of haar voorkeur installeert en soms ook andere programma’s de-installeert. Blijkbaar gebeurt dat laatste ook niet altijd met veel kennis van zaken, want er staat op alle drie de computers nog wel wat halvelings verwijderde software. Met de licentievoorwaarden wordt ook al erg lichtvaardig omgesprongen.

3.3.6. Beschrijving van het omgaan met data

Hetzelfde kan ook gezegd worden over het omspringen met databestanden. Aangezien er geen enkele map gedeeld wordt over de verschillende computers, worden alle databestanden opgeslagen op elke computer afzonderlijk. Dit leidt onvermijdelijk tot een heleboel dubbele bestanden (elk uiteraard qua omvang en inhoud verschillend van elkaar). Zo vonden we niet minder dan drie van elkaar verschillende ledenlijsten, tal van dubbele brieven en nota's, verschillende financiële rapporten, enz. Er is dan ook sprake van een zeer hoge mate van data-duplicatie en -proliferatie.

3.4. Voorlopige conclusies

Het mag duidelijk zijn dat er aan de bestaande situatie heel wat verbeterd kan worden.

Zo zouden we om te beginnen willen voorstellen om de diverse verbindingskabels tussen computers, printers en router te voorzien van *tags*. In geval van problemen kan daar dan makkelijker en sneller aan verholpen worden.

Vervolgens zouden we de netwerkkabel tussen computer A en de router willen vervangen door een iets langer exemplaar, zodat het uitvallen van die kabel vermeden kan worden.

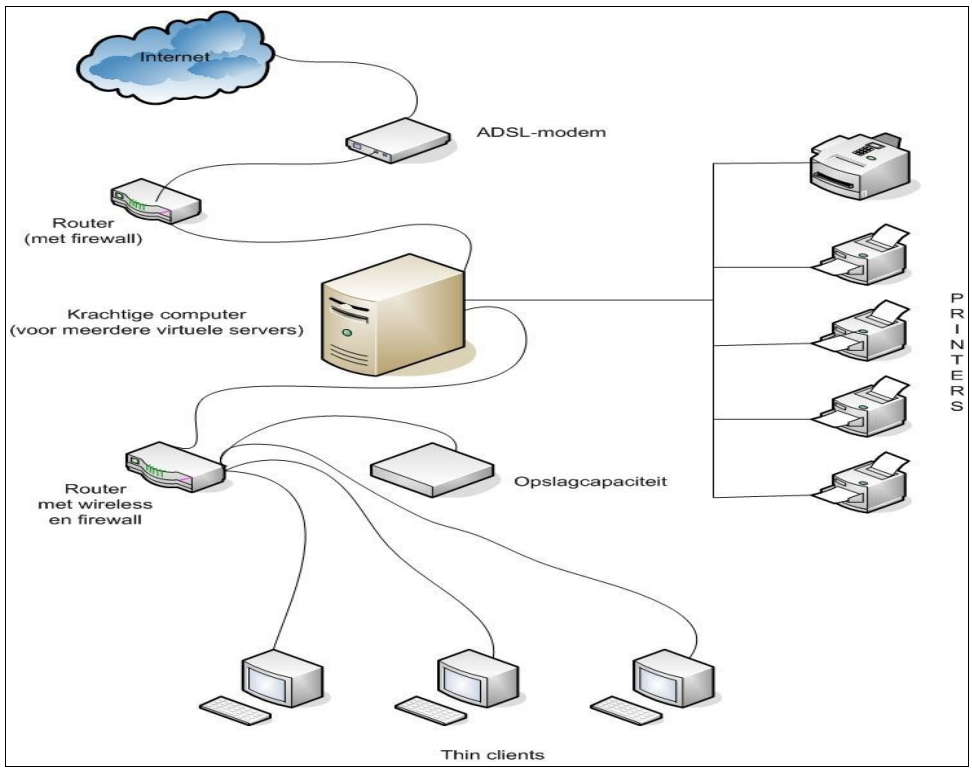
Ook zouden we durven voorstellen dat er kabelbanen worden aangebracht langsheen de muren, ter hoogte van het werkblad van de bureaus of computermeubels. In die

kabelbanen kunnen we dan alle verbindingskabels wegwerken. Het voordeel hiervan is niet alleen dat alles er wat overzichtelijker uitziet. Het is vooral handiger bij het schoonmaken der lokalen. Ook het gebeurlijk per ongeluk uitrukken van een kabel zal dan minder frequent voorvallen.

Hoe nuttig deze maatregelen echter ook zijn, ze kunnen weinig verhelpen aan de fundamentele zwakten van het bestaande netwerk.

In feite is dit netwerk slechts een rommelige verzameling van objecten, die zonder veel doorzicht hier en daar aan elkaar gekoppeld zijn. Ook de beveiliging ontbreekt totaal. De functionaliteit van de aanwezige software is al even rommelig en onaangepast. Bovendien heerst er een totale chaos op het vlak van de licentiepolitiek. Dit laatste is niet zomaar een detail, aangezien de vzw zich zo blootstelt aan eventuele juridische vervolging, met de daarbij horende (hoge) financiële boetes. Gezien de erg beperkte financiële middelen van de vzw is dat toch iets dat hen zorgen zou moeten baren.

We stellen dan ook voor om in de plaats van het bestaande netwerk een heel nieuwe configuratie op te bouwen. Daarbij zullen we wel zoveel mogelijk uitgaan van de reeds aanwezige apparatuur, die we echter optimaler zullen integreren met elkaar. Dat nieuwe netwerk zou er schematisch zo kunnen uitzien :



Afbeelding 8 : Het voorstel van een nieuw netwerk

Voor wat de software betreft stellen we voor om volledig over te stappen op *open source* software, zowel wat betreft besturingssysteem als wat betreft applicaties.

3.5. Open Source als mogelijke oplossing ?

3.5.1. Wat is Open Source ?

Open source is een beetje een containerbegrip, waardoor niet altijd even duidelijk is wat er precies mee bedoeld wordt. Oorspronkelijk werd alleen gesproken over een verschil tussen vrije en propriëtaire software. Vrije software is eenvoudig gezegd dan ook tegengesteld aan onvrije software. De vrijheid waarvan hier sprake is, slaat niet alleen op het feit dat er voor software al dan niet betaald hoeft te worden. Dat wordt meestal beschouwd als een secundaire (maar wel aantrekkelijke) kwestie. Het echte verschil slaat er op dat de gebruiker van vrije software er ook daadwerkelijk vrij mee kan omspringen, zelfs om de software zelf te veranderen. De enige voorwaarde die daaraan verbonden wordt, is dat het nieuwe, veranderde programma ook voor de volgende gebruiker weer even vrij dient te zijn. Vrije software is dan ook eigenlijk niks anders dan software waarvan de broncode vrijelijk beschikbaar is. Deze principes staan uitgebreid beschreven in de zogenaamde GNU General Public Licence van de Free Software Foundation[11].

De term *open source* is van recentere datum. Ze werd voor het eerst gemeengoed met Eric Raymond's essay *'The Cathedral and the Bazaar'*. De aanleiding voor dat essay was het feit dat vrije software weliswaar erg succesvol was (in de zin dat het daadwerkelijk werkende software was), maar dat het slechts mondjesmaat lukte om dergelijke software te doen aanvaarden in het (Amerikaanse) bedrijfsleven. Raymond's (onuitgesproken) analyse was blijkbaar dat de vrijheid waar de Free Software Foundation zo graag over spreekt, in de ogen van het bedrijfsleven al te zeer als een ietwat socialistisch-ideologisch concept overkomt. Met zijn essay trachtte Raymond dan ook de voordelen van vrije software in overeenstemming te brengen met de ethiek en ideologie van het Amerikaanse bedrijfsleven. Daartoe benadrukte hij dat het bij vrije of *open source* software veeleer zou gaan om een model voor softwareontwikkeling, eerder dan om een bedrijfsmodel. Vandaar dat hij pleitte voor het gebruik van de term *open source* in plaats van *free software*. De Free Software Foundation blijft het met die redenering fundamenteel oneens. Op het Internet woeden daarover dan ook nog steeds virulente discussies[12].

De meeste mensen maken geen wezenlijk onderscheid tussen beide termen. Zij gebruiken de termen ofwel door elkaar, ofwel gebruiken ze de term *open source* om er 'gratis' software mee aan te duiden. Dit mag dan theoretisch onjuist zijn[13], in de praktijk is het zo dat software waar je vrijelijk veranderingen aan kunt aanbrengen, meestal ook gratis verkrijgbaar is. Wij zullen in dit werk de term *open source* gebruiken in die zin.

Dikwijls wordt ook gedacht dat *open source* software gelijk staat met software voor het Linux-platform. Dat is echter een misvatting. Er bestaat *open source* software voor elk denkbaar platform, van AmigaOS tot en met zOS en alles wat daar maar kan tussenvallen. Wel is het zo dat het Linux-platform zelf volledig steunt op het concept

11 Zie http://www.gnu.org/licenses/gpl.html en http://www.gnu.org/philosophy/free-sw.html
12 Zie onder andere : http://www.gnu.org/philosophy/open-source-misses-the-point.html
13 Er bestaat software waarvoor wel degelijk betaald moet worden, maar waarbij je toch ook de beschikking krijgt over de volledige broncode, mét de toelating deze zo nodig naar eigen inzicht te veranderen. Je mag die gewijzigde versie echter niet verder verspreiden.

van de vrije en/of *open source* software. Bijgevolg is er voor dit platform heel wat meer vrije en/of *open source* software voorhanden.

In dit werk zullen we eveneens kiezen voor het Linux-platform, hoewel we in het implementatieplan een overgangsperiode zullen inlassen, waarbij *open source* software voor het Windows-platform zal aangewend worden als brug naar Linux.

3.5.2. Voordelen van *Open Source*

Voor onze vzw heeft *open source* software onmiskenbare voordelen. Deze kunnen als volgt samengevat worden :

- geen (financiële) licentie-verplichtingen
- geen afhankelijkheid van een specifieke leverancier
- onnoemlijk veel keuze inzake kwaliteitsvolle softwarepakketten
- enorm veel online documentatie

3.5.3. Nadelen van *Open Source*

Open source software heeft echter ook nadelen. Deze kunnen we als volgt samenvatten:

- geen formele ondersteuning (*no support*)
- geen gegarandeerde verhelping van fouten (*no guaranty for bugfixes*)
- dikwijls wordt slechts rudimentaire documentatie meegeleverd

3.6. Een mini-datacenter als oplossing ?

Als we teruggrijpen naar de functionele analyse hierboven (hoofdstuk 2), dan weten we dat onze vzw een aantal workflows of processen dient waar te maken. In de technische analyse hebben we dan weer kunnen vaststellen dat de huidige manier waarop deze processen met behulp van informatica worden aangepakt, leidt tot versnippering van software én data, met alle gevolgen van dien op het vlak van (gebrek aan) beheer. Een mogelijke oplossing hiervoor bestaat erin af te stappen van het paradigma der individuele computers (elk met hun eigen besturingssysteem annex software), ten voordele van het paradigma van een datacenter. Hierbij worden de benodigde applicaties geplaatst in een netwerkstructuur van servers. Voor de gebruikers verandert er daarbij wezenlijk niet zo heel veel. Ze blijven voor een 'eigen' scherm zitten en kunnen elk op zich werken met de toepassingen. Alleen, van die toepassingen is er telkens maar één exemplaar (dat dan uiteraard wel geschikt moet zijn voor gebruik door meerdere gebruikers tegelijkertijd). Tegelijk worden ook de bestanden van de gebruikers centraal opgeslagen, zodat er een einde komt aan de data-proliferatie.

Willen we voor onze vzw zo'n datacenter uitbouwen, dan dienen we eerst vast te stellen welke applicaties er precies nodig zijn. Wij stellen voor om daarvoor uit te gaan van het volgende[14] :

14 We baseerden ons voor de hier voorgestelde, mogelijke serverapplicaties op de website van de firma Optaros : http://www.eosdirectory.com/directory/searchprojectbycateg/ Deze web-

Workflow	Applicatie
Briefwisseling	Groupware-server (bijvoorbeeld SocialTextOpen[15] of MediaWiki[16] of Scalix[17] of eGroupware[18] of Open-Xchange[19] of Zimbra[20])
Rekeningbeheer	CRM-server (bijvoorbeeld SugarCRM[21] of *phpBB*[22])
Infrastructuur	ERP[23]-server (bijvoorbeeld Alfresco[24] en/of *OCS Inventory NG*[25])
Subsidiedossiers	CRM[26]-server (bijvoorbeeld Alfresco en/of SugarCRM)
Ledenbeweging	CRM-server (bijvoorbeeld Alfresco en/of SugarCRM)
Werkingsthema	Groupware-server (bijvoorbeeld SocialTextOpen of MediaWiki of Scalix of eGroupware of Open-Xchange of Zimbra)
Tijdschriften	Groupware-server, CRM-server, Virtual Desktop Server[27]
Website	Groupware-server, CRM-server, Virtual Desktop Server
Promotie	Groupware-server, CRM-server, Virtual Desktop Server
Audiovisueel	Groupware-server, CRM-server, Virtual Desktop Server
Didactisch materiaal	Groupware-server, CRM-server, Virtual Desktop Server

Tabel 3 : Overzicht workflows en benodigde servers

De generieke omschrijving van de benodigde applicaties dienen we natuurlijk ook nog te concretiseren naar welbepaalde softwarepakketten. Dat is niet eenvoudig, want de keuzemogelijkheden zijn erg groot. We willen de vzw ook niet direct onze persoonlijke voorkeur opleggen. In de plaats daarvan zullen we hen meerdere mogelijke softwarepakketten voorstellen. Tijdens de overgangsperiode kunnen de medewerkers van de vzw de verschillende pakketten uitproberen, zodat ze uiteindelijk een keuze zullen kunnen maken, die steunt op enige kennis van zaken. Concreet wil dat

site beschikt over handige zoekmogelijkheden.

15 Zie : http://www.socialtext.net/open/index.cgi (een wiki-systeem voor het uitwisselen van en/of samenwerken aan teksten)

16 Zie : http://www.mediawiki.org/ (eveneens een wiki-systeem voor het uitwisselen van en/of samenwerken aan teksten)

17 Zie : http://www.scalix.com/ (een platform voor messaging en samenwerking)

18 Zie : http://www.egroupware.org/ (een groupware systeem voor e-mail, kalender, content management, forums en nog veel meer).

19 Zie : http://www.open-xchange.org/ (groupware, e-mail, kalender)

20 Zie : http://www.zimbra.com/ (groupware, e-mail, kalender, bestandsdeling)

21 Zie : http://www.sugarcrm.com/ (voor het beheren van zowat alles wat te maken heeft met het beheer van klanten- en medewerkersgegevens).

22 Zie : http://www.phpbb.com/ (een forum voor het uitwisselen van berichten en het voeren van discussies).

23 ERP : Enterprise Resource Planning (software voor het beheer van de infrastructuur van een onderneming).

24 Zie : http://www.alfresco.com/ (een systeem om documenten en klanten- en medewerkersgegevens mee te beheren)

25 Zie : http://www.ocsinventory-ng.org/ (voor het beheren van een inventaris van computer hard- en software in een lokaal netwerk).

26 CRM : Customer Relationship Management (software voor het beheer van de klantengegevens van een onderneming. Ook geschikt voor het beheer van de gegevens der medewerkers).

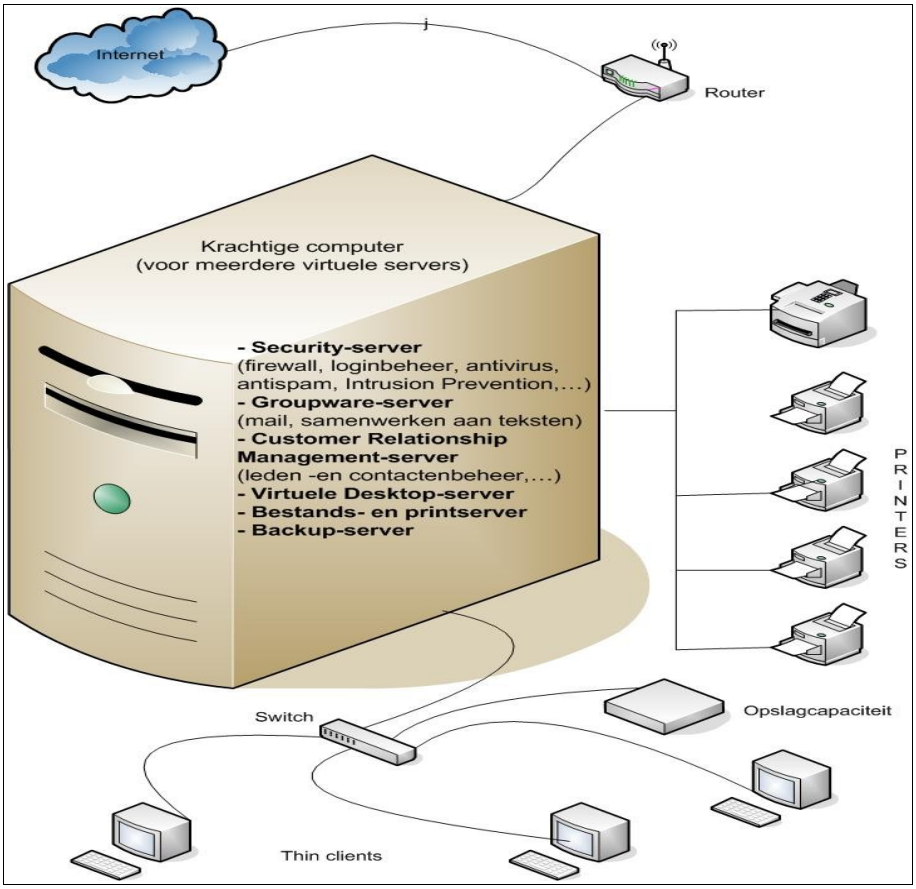
27 Virtual Desktop Server : een server die desktopomgevingen uitdeelt aan thin clients.

ook zeggen dat we voor deze applicaties pakketten zullen voorstellen die zowel op het Windows- als op het Linux-platform kunnen werken.

We zouden dit soort mini-datacenter perfect kunnen bouwen op één enkele (stevige) computer[28]. Daar zijn echter nadelen aan verbonden. Het combineren van verschillende server-applicaties op één enkele fysieke servermachine maakt immers komaf met het grote voordeel van een datacenter : het isoleren van de verschillende server-applicaties. Dat zorgt ervoor dat het delicate evenwicht tussen de verschillende applicaties in het gedrang komt. Als er iets fout loopt in de configuratie van één der applicaties, dreigt meteen het hele datacenter te kapseizen. Bovendien kan de load van één der applicaties die der andere negatief beïnvloeden.

In een datacenter voor een groot bedrijf lost men dit op door deze applicaties telkens op een aparte server te installeren,. Dat is voor de vzw (al was het alleen maar om financiële redenen) geen optie. We moeten daar dus iets anders op vinden. Dat 'iets anders' is niks anders dan virtualisering, wat we in het volgende onderdeel van naderbij bekijken.

We kunnen ons nu een voorstelling maken van het vernieuwde netwerk voor de vzw :



Afbeelding 9 : Het vernieuwde netwerk voor de vzw

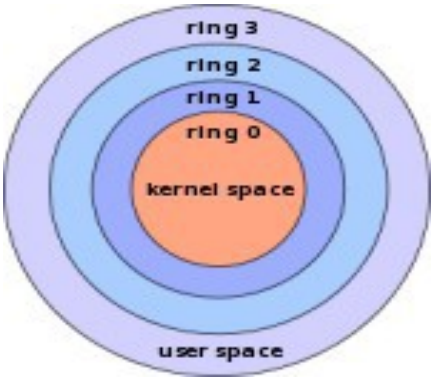
28 Waarbij de verschillende serverapplicaties dan allemaal draaien bovenop één enkel besturingssysteem.

3.7. Virtualisering als oplossing ?

Het is dus onze bedoeling om meerdere servers in gebruik te nemen, zonder dat de kostprijs de pan uitswingt en zonder dat de ene server de andere in de weg komt te zitten. Daarom gaan we die servers installeren in virtuele machines. Er zijn echter verschillende virtualiseringsmethoden.

3.7.1. Overzicht van virtualiseringsmogelijkheden

Virtualisering is niet iets dat recent werd uitgevonden. Reeds in de jaren '60 ontwikkelde IBM virtualiseringssoftware voor haar toenmalige mainframe-computers[29]. Recentelijk is echter de mogelijkheid tot het virtualiseren van de Intel x86-computer in een stroomversnelling terecht gekomen. De pionier hierbij is ongetwijfeld het bedrijf VMWare geweest. VMWare is ondertussen echter lang niet meer alleen.



Afbeelding 10 : Het ring-model van de Intel x86-computer[30]

Virtualisering maakt gebruik van het onderscheid dat gemaakt wordt tussen verschillende 'ringen' in een computer. Elke ring stelt een bepaalde mate van privilegiering voor. Ring 0 (de eerste ring) staat het dichtst bij de hardware van de computer. Toegang tot deze ring is alleen bedoeld voor diensten die rechtstreeks toegang dienen te hebben tot de hardware (denk bijvoorbeeld aan driversoftware). Gewone gebruikersapplicaties mogen in principe nooit rechtstreeks toegang hebben tot ring 0. Die gebruikersapplicaties vinden we dan weer wel terug in Ring 3 of 4[31]. In het Unix-kamp noemt men Ring 3 ook wel *user space*.

We kunnen nu op verschillende wijze een onderscheid maken tussen de beschikbare virtualiseringsmethoden. Zo is er bijvoorbeeld een onderscheid tussen desktop-virtualisering en enterprise-virtualisering[32].

29 Voor een zo goed als volledig en fascinerend verslag van het pionierswerk dat hiermee gepaard ging, zie : VARIAN, Melinda, *VM and the VM Community : Past, Present and Future*, August 1997, paper presented at SHARE 89, Sessions 9559-9061 (te vinden op : http://www.princeton.edu/~melinda/25paper.pdf).

30 Tekening afkomstig van volgende website : http://www.nodemaster.de/xen-basics/

31 Unix- (en Linux-)adepten spreken over Ring 3, Microsoft gebruikt de term Ring 4, omdat Microsoft begint te tellen vanaf 1, Unix/Linux vanaf 0.

32 Dit onderscheid ligt mee aan de basis van een uitstekend overzichtswerk over virtualisering : WOLF, Chris, HALTER, Erick M., *'Virtualization, From the Desktop to the Enterpri-*

3.7.2. Desktop- en Enterprise-virtualisering

Met desktop-virtualisering bedoeld men dat de virtualiseringsapplicatie draait bovenop het persoonlijke bureaublad. Concreet wil dat zeggen dat de virtualiseringsapplicatie draait in de zogenaamde user space (ring 3) van je computer. Bij enterprise-virtualisering is dat niet zo. Daar draait de virtualiseringsapplicatie op een laag tussen de hardware en het eigenlijke besturingssysteem. Men noemt dit ook wel *bare metal* of Ring -1.

Onder de desktop-virtualisering kunnen we pakketten noemen als VMWare Workstation[33], Microsoft Virtual PC, Parallels Desktop en VirtualBox. Onder de pakketten voor enterprise-virtualisering kunnen we VMWare ESX Server, VMWare GSX Server, Microsoft Virtual Server, Microsoft Hyper-V, SWSoft Virtuozzo en Virtual Iron noemen. Al deze pakketten behoren tot de zogenaamd propriëtaire software. Weliswaar worden er soms gratis pakketten beschikbaar gesteld door deze bedrijven. De broncode ervan is echter in principe niet vrij[34].

Naast deze propriëtaire pakketten zijn er ook meerdere *open source* pakketten voor (enterprise-)virtualisering beschikbaar. De meeste hiervan worden ontwikkeld voor (en voornamelijk gebruikt op) het Linux-platform. Enkele voorbeelden van deze pakketten zijn : Xen, OpenVZ[35], Linux V-Server, lGuest en KVM. Sinds kort werken ook IBM en Sun Microsystems aan een eigen implementatie van Xen, net zoals Oracle.

3.7.3. Relatie virtualisering en besturingssysteem

Een andere manier om een onderscheid te maken tussen de verschillende virtualiseringsmethoden bestaat eruit te kijken naar hoe de virtualiseringsmethode omspringt met het besturingssysteem. Hier kunnen we drie verschillende manieren van aanpak onderscheiden.

Ten eerste die van de pure *bare metal* installatie. Hierbij wordt er bovenop de computer (het zogenaamde *bare metal*) een soort abstractie-laag gelegd. Bovenop dit abstractieniveau kunnen er dan virtuele machines met elk hun eigen besturingssysteem geïnstalleerd worden. De virtuele machines zijn zich er niet bewust van dat ze de fysieke hardware delen met andere virtuele machines. Tot voor kort kon deze methode enkel toegepast worden, indien het besturingssysteem geschikt gemaakt werd voor deze vorm van virtualisering. Door het besturingssysteem aan te passen (te *patchen*) kon het besturingssysteem de abstractie-laag herkennen en aanspreken alsof het een echte, fysieke computer was. Deze vorm van virtualisering noemt men paravirtualisatie. Sinds kort is dit niet meer strikt noodzakelijk. De producenten van Intel-compatibele processoren hebben in hun chips immers hardware-matige virtualiseringsextensies ingebouwd. Hierdoor kunnen nu ook niet-aangepaste besturingssystemen wor-

se', 2005, Apress, New York.

33 Merkwaardig genoeg hoort ook het voor persoonlijk gebruik gratis verkrijgbare pakket VMWare *Server* tot de categorie der *desktop*-virtualisering.

34 Een uitzondering hierop vormt VirtualBox. Dit is beschikbaar onder een zogenaamde dual licence. Er bestaat een meer uitgewerkte versie (vrij voor persoonlijk gebruik) onder een commerciële licentie en zonder broncode, naast een minder uitgewerkte open source versie. De toekomst van dit project is ietwat onduidelijk, nu Innotek Systemberätung Gmbh (eigenaar van VirtualBox) werd overgenomen door Sun Microsystems.

35 OpenVZ is de open source versie van het propriëtaire Virtuozzo.

den geïnstalleerd bovenop de abstractie-laag. Open source voorbeelden van dit soort virtualisering zijn Sun's xVM[36], Oracle's OVM[37], User Mode Linux[38] (ULM) en Xen[39].

Een andere aanpak bestaat eruit helemaal geen virtualisatie-laag te installeren, maar binnenin (of zo u wil : bovenop) het besturingssysteem wel een soort isolatielaag te installeren. Hierdoor wordt het mogelijk bovenop een besturingssysteem zogenaamde containers te installeren. Deze containers zijn er zich niet bewust van dat zij bestaan naast andere containers. Ze draaien dus alle geïsoleerd van elkaar. Het voordeel van deze aanpak is dat het onderliggende besturingssysteem instaat voor alle diensten. Heeft een container bijvoorbeeld toegang nodig tot de harde schijf, dan wordt dit eenvoudigweg 'doorgelust' naar het onderliggende besturingssysteem. Dankzij deze aanpak kan het container-besturingssysteem heel slank blijven. Containers verbruiken dus minder machinebronnen, wat de performantie zeer ten goede komt. *Open source* voorbeelden van deze vorm van virtualisering zijn Linux V-Server[40] en OpenVZ[41].

Nog een andere aanpak kunnen we omschrijven als een hybride vorm. Hierbij worden de voordelen van de *bare metal* installatie gecombineerd met de containeraanpak. Dit doet men door de virtualiseringsmogelijkheid als het ware in te bouwen in de Linux-kernel, zodat deze de hardwarematige virtualiseringsextensies van de Intel-compatibele processoren rechtstreeks kan aanspreken. De Linux-kernel wordt daarmee feitelijk zelf een hypervisor, die daarna de containeraanpak gebruikt om bovenop zichzelf diverse, van elkaar geïsoleerde instanties te creëren. Het enige voorbeeld (en ook *open source*) hiervan is de zogenaamde Kernel-Based Virtual machine manager (KVM)[42].

Omdat we kiezen voor *open source* oplossingen, hebben we dan ook een ruime keuze. We zullen meer bepaald drie van deze methoden van naderbij bekijken : Xen, OpenVZ en KVM.

36 Zie : http://sun.com/xvm (Sun's xVM steunt overigens op Xen).
37 Zie : http://www.oracle.com/technologies/virtualization/index.html (Oracle's OVM steunt eveneens op Xen).
38 Zie : http://user-mode-linux.sourceforge.net/
39 Zie : http://www.cl.cam.ac.uk/research/srg/netos/xen, http://www.xen.org/ en/of http://www.xensource.com
40 Zie : http://linux-vserver.org/
41 Zie : http://openvz.org
42 Zie : http://kvm.qumranet.com

4. Opzetten van een testomgeving

4.1. Inleiding

Voor we hiertoe kunnen overgaan, moeten we echter eerst een aantal algemene, voorafgaande vereisten in orde brengen. Zo hebben we uiteraard een testsysteem nodig. Dat testsysteem dient te beantwoorden aan de vereisten die nodig zijn om de verschillende virtualiseringsmethoden te kunnen gebruiken. Daarnaast dienen we ook een aantal voorafgaande beslissingen te nemen, die te maken hebben met het verwerkelijken van een configuratie die nuttig kan zijn voor ons project.

4.2. Overzicht van de testcomputer

We kiezen voor een testsysteem met de volgende kenmerken :

- Moederbord : Gigabyte 965P-DS3
- Processor : Intel Core2Duo 6400 2.13 Ghz
- Harde schijven : 2x 500GB SATA Samsung HD501LJ, 7.200 rpm, 16MB cache
- DVD/RW : DVD-RAM GSA-H30N
- RAM : 6 GB
- Grafische Kaart : ATI Radeon RV530 X1650 Pro, 256MB
- On-board RAID-Controller jMicron BIOS version 1.06.59. ID : GRAID (= *fake-RAID* !)
- Netwerkkaart 1 : D-Link System Inc DGE-528T Gigabit Ethernet Adapter
- Netwerkkaart 2 : US Robotics USR997902 10/100/1000 Mbps PCI Network Card (Ethernet)
- Geluidskaart : ingebouwd, soundblaster compatible.

Hierbij is wel een woordje uitleg nodig.

4.2.1. Het moederbord

Zo is het moederbord er één dat kan werken met de huidige generatie processoren (zoals de Core2Duo met twee 64-bits processor-kernen), maar evengoed met *quadcore* of 4 processorkernen. Het is om zo te zeggen een *future-proof* moederbord.

4.2.2. De processor

De geïnstalleerde processor heeft als belangrijke kenmerk dat hij beschikt over hardware-matige virtualiseringsextensies, hetgeen hem uitermate geschikt maakt voor het gebruik dat wij ervan willen maken. Hij beschikt ook over *Physical Address Extension* ((PAE), waardoor hij in staat is – zelfs met gebruik van een 32-bits besturingssysteem – om meer dan 4 GB RAM-geheugen te adresseren. Handig, want we beschikken over 6 GB RAM-geheugen.

4.2.3. RAM-geheugen

Die hoeveelheid RAM-geheugen lijkt wellicht wat overdreven, maar het brengt wel met zich mee dat we wat meer speelruimte hebben bij het toewijzen van geheugen aan onze virtuele machines. Bovendien is de prijs voor RAM-modules momenteel zo laag dat er weinig tot geen bezwaar is om meer RAM te plaatsen.

4.2.4. Grafische kaart

De grafische kaart is aan de zware kant voor een servermachine. De meeste servers (en uiteindelijk ook deze) draaien immers in zogenaamde *headless mode* (i.e. zonder monitor). Waarom dan zo'n zware grafische kaart ? De reden daarvoor heeft weer te maken met het kiezen voor een *future-proof* machine. Momenteel is het weliswaar zo dat virtuele machines (zoals wij er willen installeren) en *thin clients* (die we ook willen plaatsen in het netwerk) geen of slechts in beperkte mate gebruik kunnen maken van de mogelijkheden van de grafische kaart op de server. Het ziet er echter naar uit dat dat niet erg lang meer zo zal blijven. Zo is het nu al mogelijk (hoewel slechts experimenteel) om onder de Xen-virtualiseringsmethode de grafische kaart toe te wijzen aan één virtuele machine (en dus niet meer aan de onderliggende *hypervisor*. Die virtuele machine kan dan volledig gebruik maken van de mogelijkheden der grafische kaart. Dat kan bijvoorbeeld nuttig zijn om een dergelijke virtuele machine te gebruiken voor het bewerken van multimediabestanden.

Daarnaast is het zo dat virtualisering iets is waar heel druk aan gewerkt wordt. Voortdurend komen er dan ook nieuwe ideeën en toepassingen beschikbaar. Niet toevallig heeft bijvoorbeeld Microsoft onlangs een bedrijfje opgekocht[43] dat bezig is een systeem te ontwikkelen om het gebruik van grafische hardware door meerdere virtuele machines en thin clients mogelijk te maken. Ons testsysteem is dus voorbereid op die toekomstige vernieuwingen.

4.2.5. Harde schijven

Tenslotte zijn de harde schijven aan de grote kant (inderdaad : het systeem heeft in totaal een opslagcapaciteit van 1 terabyte !). We willen op dit systeem echter ook een vorm van RAID gaan toepassen. En tegelijk zijn we zo in één keer verlost van vele potentiële opslagproblemen. Voor de prijs moeten we het ook niet laten, want ondanks deze specificaties is dit testsysteem toch niet duur. Het werd op maat gemaakt voor de ronde prijs van € 1200.

4.2.6. RAID-controller

Toch is er ook één negatief aspect aan dit systeem en dat is de ingebouwde RAID-controller. Deze werkt niet met het door ons gekozen besturingssysteem. We probeerden achtereenvolgens de Linux-distributies CentOS 5.1 (64-bit), Fedora 8 (32- en 64-bit) en Debian 4 (32- en 64-bit). Allemaal gaven ze eenzelfde soort foutmelding :

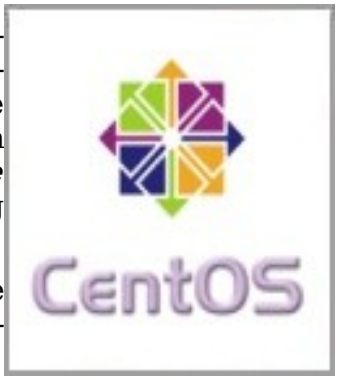
43 Zie http://blogs.technet.com/virtualization/archive/2008/03/12/Kidaro-to-be-added-to-Microsoft_2700_s-desktop-virtualization-products.aspx. In dit artikel is er sprake van meerdere overnames, onder andere van het bedrijfje Calista Technologies, dat naar eigen zeggen in staat is (of zal zijn ?) om 3D *graphics*, DirectX en audio weer te geven in *thin clients*.

“* glibc detected *** /usr/bin/python: malloc(): memory corruption: 0x0000000009f98ab0 ***”.**

Ook met verschillende opstartopties of *boot options* bereikten we geen succes.[44] Op verschillende websites kregen we ook negatieve meningen te lezen over zogenaamde *fake raid* (ofwel valse raid).[45]

Uiteindelijk kozen we ervoor om de jMicron RAID-controller niet te activeren. We zullen in plaats daarvan de software-RAID oplossing van Linux zelf gebruiken.

4.2.7. Besturingssysteem



Het besturingssysteem dat we uiteindelijk kozen voor ons testsysteem is CentOS 5.1 in de 64-bits uitvoering. Dit is een besturingssysteem dat volledig identiek is aan Red Hat Enterprise Linux, maar zonder de daaraan verbonden kostprijs.[46] We kozen hiervoor omdat dit een zogenaamd *testproven, robust enterprise system* is. Het is dus uiterst stabiel en kan gegarandeerd overweg met onze hardware.

Daarenboven komt dit besturingssysteem ook als beste uit de hoek voor wat betreft het werken met de Xen-virtualiseringsmethode.[47]

Bij de installatie van CentOS kozen we voor het Engels als taal, met een Belgisch toetsenbord. We installeerden slechts een minimaal systeem, met uitschakeling van alle software die overbodig is voor onze hardware (zoals ondersteuning voor *smart cards*, infrarood, PCMCIA-kaarten en dergelijke meer), maar wel voor de Gnome-desktop (een grafische omgeving), omdat het daarmede toch makkelijker werken is. Bovendien is het zo dat deze configuratie uiteindelijk gebruikt zal worden door mensen die minder of geen ervaring hebben met *command driven* software-installaties. Een grafische desktop is voor hen beter geschikt. Op dit moment installeren we ook nog geen ondersteuning voor virtualisatie. We willen immers eerst een stabiel basissysteem opzetten. Later zullen we de virtualiseringsmogelijkheid toevoegen.

De installatie van dit basissysteem zonder virtualisering is ook nuttig mocht er iets fout gaan met de (later te installeren) virtualisatiesystemen. We kunnen dan immers steeds via een herstart of *reboot* terugkeren naar dit basissysteem.

---

44 Weliswaar wordt door de firma jMicron beweerd dat *“JMB bios 1.06.53 has Linux support with kernel 2.6.18 OR higher”* en *“kernel already built-in parameter "all-generic-ide=1" to support newer controller. Ex: "boot:linux all-generic-ide=1".* Cf. http://www.jmicron.com/Support_ FAQ.html. Onze tests wezen echter uit dat dit (nog ?) niet klopte.

45 Zie bijvoorbeeld http://linux-ata.org/faq-sata-raid.html

46 CentOS (*Community Enterprise Operating System*) wordt gemaakt door de broncode te nemen van Red Hat Enterprise Linux, daaruit alle logo's van Red Hat te verwijderen, waarna de broncode opnieuw gecompileerd wordt. Helemaal in de geest van *open source* dus.

47 In de mailinglist voor Xen-gebruikers vonden we meerdere berichten ten voordele van CentOS in vergelijking met andere Linux-distributies.

4.2.8. Indeling der harde schijven

We gaan niet voor een automatische partitionering van de twee harde schijven. In plaats daarvan willen we een (noodgedwongen softwarematig) RAID-systeem opzetten, met daar bovenop een partitieconfiguratie die we makkelijk kunnen wijzigen (aanvullen, vergroten, verkleinen, enz.). Bovenop de RAID-1-configuratie willen we daartoe een LVM-systeem opzetten (LVM = *Logical Volume Management*). Dergelijke configuratie moeten we manueel opzetten.

De hiërarchie van het indelen der harde schijf gebeurt dan als volgt :

1. RAID-1 opzetten
2. Logical Volume Groups definiëren
3. Logical Volumes aanmaken in de Logical Volume Groups (dit zijn de eigenlijke partities).

4.2.9. RAID-configuratie

We kiezen voor RAID-1, omdat we zo redundantie verkrijgen – als 1 disk niet meer werkt, werkt het softwaresysteem verder tot we een vervangende schijf kunnen installeren. Na de installatie van de vervangende schijf, zal de RAID-array terug opgebouwd kunnen worden.

Onze partitie-indeling met RAID-1 en LVM ziet er als volgt uit :

RAID-*DEVICE*	***MOUNT-POINT***	**TYPE**	**GROOTTE**
/dev/md0[48]	/boot	ext3	100 MB
/dev/md1	VG_SYSTEM[49]	(LVM PV)[50]	51.200 MB
/dev/md2	VG_VMS[51]	(LVM PV)	410.260 MB
		Vrije ruimte	51.200 MB

Tabel 4 : Partitionering met RAID-1 en Logical Volume Management

We laten met opzet wat vrije ruimte over op de harde schijven. Het is immers zo dat we vandaag niet zeker kunnen weten of we in de toekomst nog een harde schijf kunnen vinden van exact dezelfde grootte als diegene die we nu gebruiken. Mocht er onverhoopt iets mis gaan met één van onze schijven, dan zouden we in zo'n geval problemen kunnen hebben met het heropbouwen van onze RAID-*array*. Door wat speling over te laten, verkleinen we dat risico.

De RAID-*device* '/dev/md0' voorzien we van het ext3-bestandssysteem, waarop we de startconfiguratiebestanden van Linux plaatsen. De RAID waarvoor we kozen is RAID-1 (= mirroring). Dat zou overigens ook niet anders gaan bij een configuratie met

48 In Unix-achtige systemen (zoals Linux) begint men te tellen bij 0.
49 VG_SYSTEM = *Volume Group System* : op deze volume group wordt het basissysteem geïnstalleerd (de naam is overigens volstrect arbitrair in te stellen).
50 LVM PV = *Logical Volume Management Physical Volume* : een abstractielaag waarop *logical volumes* met een eigen bestandssysteem kunnen geïnstalleerd worden.
51 VG_VMS = *Volume Group Virtual Machine Systems* : op deze volume group zullen we de virtuele machines installeren, elk in hun eigen logical volume.

slechts twee harde schijven. Weliswaar zouden we ook kunnen kiezen voor een configuratie met RAID-0 (=striping). We verliezen dan echter het redundantie-voordeel. Bovendien zouden we ook dan toch minstens van de *boot*-partitie een RAID-1 moeten maken, omdat Linux enkel kan opstarten of booten vanop een RAID-1 device.

4.2.10. Partities en Logical Volumes

Op de RAID-*device* '/dev/md1' creëren we dan een Logical Volume Group van 51.200 MB. In deze Volume Group met de naam VG_SYSTEM kunnen we verschillende *Logical Volumes* aanmaken, telkens één *logical volume* per benodigde partitie. Dat ziet er dan als volgt uit :

RAID-*device*	**Naam**	***Mount-point***	**Grootte**
/dev/md1	LV_root	/	5 GB
	LV_usr	/usr	3 GB
	LV_tmp	/tmp	2 GB
	LV_var	/var	5 GB
	LV_home	/home	1 GB
	LV_opt	/opt	3 GB
	LV_isos	/isos	2 GB
	LV_repo	/repo	15 GB
	LV_swap	-	2 GB[52]
	Vrije ruimte	-	12 GB

Tabel 5 : De indeling met logical volumes van VG_SYSTEM

Hoewel we slechts een minimaal systeem installeren, kiezen we toch voor een vrije grote omvang (ongeveer 50 GB), omdat we van plan zijn op dit filesysteem niet alleen ons basissysteem te installeren, maar ook een volledige *repository* voor alle software die voor CentOS beschikbaar is. Daardoor zullen we niet telkens opnieuw alle pakketten van over het Internet moeten downloaden. Bovendien kunnen we zo op een veilige manier ons systeem installeren en configureren vóór we verbinding gaan maken met het Internet. We maken in de *volume group* VG_SYSTEM dan ook nog bijkomende *logical volumes* aan, met name LV_isos en LV_repo. Op LV_isos plaatsen we de zogenaamde ISO-bestanden die nuttig kunnen zijn om onze virtuele servers te kunnen installeren. We beperken deze logical volume in grootte tot 2 GB omdat we voor onze virtuele servers toch slechts één image zullen gebruiken. LV_repo dient dan weer

52 Over de precieze omvang van swap-partities worden op het Internet hele (figuurlijke) veldslagen uitgevochten. Ons leek de mening dat je best wat meer ruimte voor swap voorziet redelijk te zijn. Groter dan 2 GB is voor één swap-volume onder Linux op het Intel-platform overbodig, omdat Linux slechts volumes tot 2 GB blijkt aan te spreken (hoewel we ook dan nog bijkomende swap-volumes of -files kunnen aanmaken). Tijdens het werken hebben we wel nooit meer dan 250 MB swap-gebruik kunnen vaststellen. Mocht er onverhoopt (bijvoorbeeld door een *infinite loop*) toch meer swap-ruimte nodig zijn, dan kunnen we die altijd nog aanmaken. Zie voor deze en andere interessante meningen over swap ook : http://www.ibm.com/developerworks/linux/library/l-swaptip2.html

groot genoeg te zijn om de volledige omvang van alle beschikbare pakketten voor ons CentOS-systeem te kunnen bevatten.[53]

Op het derde RAID-device (‘/dev/md2’) plaatsen we de *Logical Volume Group* VG_VMS. Hier zullen we onze virtuele machines gaan plaatsen, waarbij we per virtuele server de nodige *Logical Volumes* zullen aanmaken. Het voordeel van LV’s is onder andere dat de grootte ervan aanpasbaar is. Bovendien blijken virtuele machines in logical volumes ook performanter te zijn. Tenslotte zijn *logical volumes* ook heel handig voor het maken van backups via *snapshotting*.

De redenen waarom we kiezen voor deze vrij ingewikkelde indeling van onze harde schijven, heeft niet alleen te maken met ons voornemen tot virtualisering. Minder en kleiner beslag op de harde schijven zorgt bijvoorbeeld voor een flexibeler systeem (dat makkelijker uitbreidbaar is), maakt backups makkelijker en vermindert de mogelijkheden tot schijffragmentatie.[54]

4.2.11. Initiële netwerk-configuratie

Daarna configureren we onze testmachine in het netwerk en wel als volgt :

IP eth0 : 192.168.1.011/24 (statisch IP-adres – naar Internet)

IP eth1 : 192.168.1.111/24 (statisch IP-adres – naar intern netwerk)

Hostname : testmachien.testdomein.org

Gateway : 192.168.1.1

Primary DNS : 192.168.1.1

Secondary DNS : 127.0.0.1

TimeZone : Europe/Brussels

Na de installatie en een reboot, doen we de post-installatie. Daarbij zetten we de firewall (voorlopig) uit, en stellen we de security enhancements van SELinux in op *permissive*. We doen dit omdat het straks anders aartsmoeilijk wordt om virtualisatiepakketten te installeren. We schakelen ook de zogenaamde dump-omgeving voor kernel-fouten uit (die zou anders een deel van het RAM-geheugen reserveren en dat hebben we niet nodig). We controleren de tijd en maken van de gelegenheid gebruik om ook een automatische synchronisatie via de *ntpd-service*, ofwel de *network time protocol daemon*. Dit is handig, omdat we zo op ons systeem de exacte *Coordinated Universal Time* (UTC)[55] kunnen bijhouden. De *ntpd-service* controleert regelmatig een *timeserver* en past zo nodig de systeemklok aan. Dit is nuttig bij het werken met virtuele machines. Een nadeel van virtualisering is immers dat de interne systeemklok van de gevirtualiseerde servers niet noodzakelijk synchroon blijft lopen. Via de *ntpd-ser-*

53 De volledige omvang van alle beschikbare pakketten is veel groter dan wat er op de installatie-DVD te vinden is.

54 Zie : http://www.ibm.com/developerworks/linux/library/l-partitiontip.html

55 De afkorting UTC combineert Engels (Universally Coordinated Time) met Frans (Temps Universel Coordonné) en vervangt het vroegere *Greenwich Mean Time*.

vice worden afwijkingen dan rechtgezet. We creëren géén gewone gebruiker voor ons CentOS-systeem, zodat enkel de zogenaamde root-user er toegang toe kan hebben. Het geluid activeren we niet, aangezien er aan een server toch geen geluidsboxen hangen.

Na een nieuwe herstart kijken we nu nog eens na of er niet nog een paar overbodige services (of *daemons* in het Linux-jargon) op de achtergrond blijven draaien. We schakelen de (automatisch geïnstalleerde) *daemons* voor *smartcards*, PCMCIA-kaarten en infrarood-*interfaces* uit. We kijken ook nog na of er geen andere optimalisaties zijn die we kunnen toepassen, ondermeer inzake het tmp-systeem voor tijdelijke bestanden.[56]

4.2.12. Een lokale *repository* aanmaken

We gaan bij het installeren van onze verschillende (virtuele) servers meerdere keren gebruik moeten maken van verschillende pakketten die behoren tot de CentOS-distributie. Deze pakketten kunnen we downloaden van een zogenaamde *repository*[57] op het Internet. Om bandbreedte en tijd te sparen, is het natuurlijk beter dat we een lokale *mirror* of spiegelserver op onze eigen harde schijf aanmaken. Dan hoeven we alleen af en toe wat updates op te halen en verder kunnen we alles rechtstreeks vanaf onze eigen harde schijf installeren. Daarnaast is het ook zo dat we straks, wanneer we effectief gaan werken met virtuele machines, moeten zorgen voor optimale beveiliging. Ook daar zal die lokale repository zijn nut kunnen bewijzen (zie ook verder). Met andere woorden, we gaan een eigen, *local repository* aanmaken.[58] Dat doen we als volgt :

1. Kopieer de inhoud van de folder /Packages op de installatie-DVD naar de folder /repo[59] :

```
[root@testmachien ~]# cp /media/dvd/CentOS/*  /repo/base

[root@testmachien ~]# umount /media/dvd
```

2. Installeer de webserver *apache* (om er later programma-pakketten mee aan te kunnen bieden) :

```
[root@testmachien ~]# yum install httpd
```

56 Zie : CILIENDO E, TUNIMASA T., '*Linux Performance and Tuning Guidelines'*, IBM Redbooks, IBM International Technical Support Organisation, 2007, free download from http://www.redbooks.ibm. com/abstracts/redp4285.html - registratie nodig ! Hier werd vooral gebruik gemaakt van suggesties in '*Chapter 4 : Tuning the operating system'.*

57 Een *repository* is de verzameling van alle programma-pakketten die deel uitmaken van een Linux-distributie (i.e. véél meer dan er op de installatie-DVD te vinden zijn). Dergelijke *repository* is te vinden op een server op het Internet. Van dergelijke servers bestaan er gelukkig ook tal van dichtbijgelegen *mirrors* of spiegelservers, zoals bijvoorbeeld ftp.belnet.be/centos/.

58 Voor deze procedure hebben we gebruik gemaakt van de richtlijnen op http://www.howtoforge.com /setting-up-a-local-yum-repository-fedora8. We hebben deze richtlijnen echter wel wat aangepast (o.a. voor gebruik met CentOS én met een apart *Logical Volume,* in casu /VG_SYSTEM/LV_repo).

59 Waarbij /repo een *mount point* is voor de werkelijke partitie /VG_SYSTEM/LV_repo

3. Zorg ervoor dat de apache-webserver bij de opstart van het systeem automatisch mee opstart :

```
[root@testmachien ~]# chkconfig --levels 235 httpd on
```

4. Start de apache-webserver :

```
[root@testmachien ~]# /etc/init.d/httpd start
```

5. Installeer het pakket *createrepo* :

```
[root@testmachien ~]# yum install createrepo
```

6. Maak de folders aan die de apache-webserver nodig heeft om de software te kunnen aanbieden via een webbrowser :

```
[root@testmachien ~]# mkdir -p /var/www/html/CentOS/base/5/x86_64

[root@testmachien ~]# mkdir -p /var/www/html/CentOS/updates/5/x84_64
```

7. Maak een snelkoppeling of *symbolic link* aan, die vanuit deze folders doorverwijst, respectievelijk naar de folders /repo/base en /repo/updates :

```
[root@testmachien ~]# ln -s /repo/base \ /var/www/html/Cen-
tOS/base/5/x86_64

[root@testmachien ~]# ln -s /repo/updates \ /var/www/html/CentOS/up-
dates/5/x86_64
```

8. Synchroniseer[60] nu de lokale *repository* met een CentOS-mirror - bijvoorbeeld die van ftp.belnet.be/centos/. Opgelet : dit vraagt heel wat tijd. Doe het bij voorkeur 's nachts !

```
[root@testmachien ~]# rsync -avrt \
rsync://ftp.belnet.be/centos/5/os/x86_64/CentOS/ /repo/base
```

9. Creëer nu de lokale *base-repository* met het commando *createrepo* :

```
[root@testmachien ~]# createrepo /repo/base
```

10. We controleren nu (via de snelkoppeling ofwel *symbolic link*) de inhoud van de nieuwe folder /repo/base/repodata en we krijgen het volgende resultaat :

```
[root@testmachien ~]# ls -l \ /var/www/html/Cen-
tOS/base/5/x86_64/base/repodata/

total 28648

-rw-r--r-- 1 root root  8920266 2008-03-11 20:16 filelists.xml.gz

-rw-r--r-- 1 root root 16550919 2008-03-11 20:16 other.xml.gz
```

60 We gebruiken hiervoor het commando *rsync*, ofwel *remote synchronization*. Dit commando vergelijkt bit per bit de inhoud van een *remote* folder met een lokale en schrijft de bits die op de *remote* folder verschillend zijn ten opzichte van de lokale folder, weg naar de lokale folder. Erg efficiënt.

```
-rw-r--r-- 1 root root  3814775 2008-03-11 20:16 primary.xml.gz

-rw-r--r-- 1 root root      951 2008-03-11 20:16 repomd.xml

[root@testmachien ~]#
```

11. Met hetzelfde, eerder gebruikte commando *rsync* gaan we nu de folder /repo/updates vullen :

```
[root@testmachien ~]# rsync -avrt rsync://ftp.belnet.be/centos/5/up-
dates/x86_64/ --exclude=debug/ \ /repo/updates
```

12. Nu gaan we er voor zorgen dat de beide folders (/repo/base en repo/updates) regelmatig *up to date* worden gebracht. We stellen dit zo in dat elke 15de dag van de maand, telkens om 1 minuut na twee uur 's nachts de folder /repo/updates wordt gesynchroniseerd met de spiegelserver. Hiervoor gebruiken we het zogenaamde *crontab*-commando, dat een zogenaamde *cronjob* (dit wil zeggen een taak die automatisch uitgevoerd wordt) zal wegschrijven :

```
[root@testmachien ~]# crontab -e

01 02 15 * * /usr/bin/rsync -avrt \
rsync://ftp.belnet.be/centos/5/updates/x86_64/ --exclude=debug/ \
/repo/updates

:wq
```

13. Tenslotte gaan we ervoor zorgen dat ook ons eigen CentOS-systeem vanaf nu enkel gebruik zal maken van onze eigen *repository*. Hiervoor moeten we het bestand /etc/yum.conf aanpassen. Dat doen we als volgt :

```
[root@testmachien ~]# vi /etc/yum.conf
```

14. Ga naar de laatste regels, waar er staat :

```
# PUT YOUR REPOS HERE OR IN separate files named file.repo

# in /etc/yum.repos.d
```

15. Voeg daar nu onderstaande tekst toe :

```
[base-local]

name=CentOS $releasever - $basearch

failovermethod=priority

baseurl=http://192.168.1.11/CentOS/base/$releasever/$basearch/base/

#mirrorlist=http://mirrors.CentOSproject.org/mirrorlist? \

repo=CentOS-$releasever&arch=$basearch

enabled=1
```

```
gpgcheck=0


[updates-local]

name=CentOS $releasever - $basearch - Updates failovermethod=priority

baseurl=http://192.168.1.11/CentOS/updates/ \

$releasever/$basearch/updates/

#mirrorlist=http://mirrors.CentOSproject.org/mirrorlist? \

repo=updates-released-f$releasever&arch=$basearch

enabled=1

gpgcheck=0
```

Hiermee is onze lokale *repository* ingesteld. Nu moeten we er nog voor zorgen dat we deze lokale *repository* niet alleen kunnen gebruiken voor ons eigen basissysteem, maar ook voor onze eerste virtuele machine[61]. We hebben daartoe al de beschikking over de Apache webserver (httpd), die de bestanden uit de *repository* kan aanbieden aan de virtuele machine. We hebben echter meer nodig. Met name willen we dat de virtuele machine via DHCP een IP-adres toebedeeld krijgt en een hostnaam. Ook willen we dat DNS zodanig geconfigureerd is dat de virtuele machine kan behoren tot een apart subnet, dat toch toegang heeft tot het Internet. Naast de Apache webserver installeren we dan ook nog de onder Linux veel gebruikte ISC DHCP-server. We passen het configuratiebestand[62] voor de DHCP-service zodanig aan dat de virtuele machine een IP-adres en een hostnaam kan krijgen en dat minimale routering wordt opgezet[63] :

```
[root@testmachien ~]# vi /etc/dhcpd.conf

option domain-name “testdomein.org”;

option domain-name-servers 192.168.1.11;

option routers 192.168.1.11:

max-lease-time 172800;

default-lease-time 172800;

option subnetmask 255.255.255.0;
```

61 We hebben voor deze procedures rijkelijk geput uit een artikel van Faye GIBBINS uit Sys Admin Magazine. Zie http://www.samag.com/documents/s=10112/sam0702e/0702e.htm

62 Het gaat om een lichte aanpassing van dit dhcpd.conf-bestand : http://www.samag.com/documents/s=10112/sam0702e/0702e_l1.htm

63 We zouden dit ook kunnen configureren via een DNS-server als Bind, maar dat is een beetje *overkill* voor wat we hier nodig hebben.

```
option broadcast-address 192.168.1.255;

ddns-update-style ad-hoc;

subnet 10.0.10.0 netmask 255.255.255.0 {

    range 10.0.10.200 10.0.10.210;

}

group{

    host krb5.testdomein.org { hardware ethernet 00:16:3e:00:00:11; \

     fixed-address 10.0.10.10; option host-name \

     “krb5.testdomein.org”; always-broadcast on;}

}
```

4.2.13. Bedenkingen omtrent beveiliging

Op zichzelf behoort Linux tot de vrij veilige besturingssystemen, onder andere omdat Linux profiteert van de lange geschiedenis van Unix. Ook het *open source* karakter van Linux draagt bij tot de relatief grote veiligheid van het besturingssysteem. Fouten en tekortkomingen worden immers door een bijzonder grote gemeenschap van ontwikkelaars bekeken. Het duurt meestal niet lang voor er *patches* en andere bijsturingen beschikbaar worden gesteld. Toch moeten we nadenken over beveiliging en zullen we niet kunnen ontsnappen aan het nemen van een paar maatregelen. Zonder in detail te willen gaan[64], zullen we hieronder dan ook trachten een minimale beveiligingsstrategie te beschrijven. Daarbij volgen we de logische opstart-volgorde van het testsysteem, van het *Basis Input Output System* (BIOS), over het besturingssysteem tot de verschillende virtuele machines én het netwerk.

4.2.13.1. BIOS afschermen

Beveiliging van een computersysteem begint bij het afschermen van de fysieke toegang tot het systeem. Enkel die personen die moeten werken met dergelijk systeem, mogen er ook daadwerkelijk toegang toe hebben. We kunnen daarbij denken aan afsluitmogelijkheden voor het lokaal waar de computer zich in bevindt. Dat is echter onvoldoende. Want ook een persoon die wel met de computer mag werken, maar die zich niet mag bezighouden met de administratie van het systeem, moet die toegang ontzegt kunnen worden. Niets is immers zo makkelijk als een compleet beveiligd computersysteem opnieuw onderuit te halen door op te starten vanop een zogenaamde *bootable* CD om dan te doen wat hij of zij wil doen. De eerste beveiliging bestaat er in dit geval dan ook uit dat de opstart-volgorde voor de computer, zoals die ingesteld wordt in het BIOS, vertrekt bij de harde schijf. Het wordt dan onmogelijk om de computer op te starten vanaf een CD-Rom of floppy.

64 Het thema beveiliging is een onderwerp op zich, een specifiek eindwerk waardig.

Ook dat is echter niet genoeg. De malicieuse gebruiker kan dan immers nog steeds de instellingen van het BIOS zelf weer wijzigen. Daarom dient de toegang tot het BIOS ook beschermd te worden, wat mogelijk is door er een paswoord op te plaatsen dat enkel gekend is door de administrator(s).

4.2.13.2. GrUB en het uitschakelen van *single user mode*

Ook dit is echter onvoldoende. Een Linux-systeem kan immers ook opgestart worden in zogenaamde *single user* modus. Hoewel dit niet zo maar één-twee-drie te realiseren is met een Red Hat Enterprise Linux-derivaat (zoals CentOS), is het toch nog altijd mogelijk[65]. Daarbij dient de automatische opstart-sequentie van de GrUB-*bootloader* onderbroken en gewijzigd te worden. Om ervoor te zorgen dat enkel de administrator over deze mogelijkheid kan beschikken, dienen we de mogelijkheid tot het veranderen van de GrUB-*bootloader* dus ook ontoegankelijk te maken voor gewone gebruikers. Ook hier zullen we een paswoord moeten instellen.

4.2.13.3. Encryptie van de harde schijven

Dit instellen van paswoorden ter voorkoming van manipulatie van het BIOS en de bootloader zijn allemaal stappen in de goede richting. Naast het feit dat paswoorden weleens vergeten worden, hebben ze ook één groot nadeel : ze kunnen gekraakt worden[66]. Het blijft dus mogelijk dat een onbevoegd persoon alsnog toegang zou kunnen krijgen tot de harde schijf van het systeem. Een mogelijke oplossing hiervoor kan encryptie-software zijn[67]. Hierbij wordt de inhoud van de gehele harde schijf (of één of meer partities) geëncrypteerd. Om na het opstarten van het systeem ook daadwerkelijk het besturingssysteem van start te kunnen laten gaan, dient dan eerst een zogenaamde *passphrase* te worden ingegeven. Dit is een langer paswoord, dat bijgevolg ook veel moeilijker te ontcijferen valt. Na het ingeven van de correcte *passphrase* wordt de inhoud van de harde schijf ontsleuteld, waarna het besturingssysteem daadwerkelijk van start kan gaan. Uiteraard zal dit de opstarttijd van het systeem gevoelig verlengen[68]. Aangezien ons systeem echter bedoelt is om permanent actief te zijn, vormt dit geen probleem. Wel maken we het zo véél moeilijker voor een onbevoegde om ons systeem te compromitteren doorheen een eenvoudige herstart.

Toegang van buitenaf tot ons systeem hoeft echter niet enkel te gebeuren via het fysiek manipuleren van de machine in het lokaal waar die machine zich bevindt. Ook toegang via het netwerk behoort immers tot de mogelijkheden. Om ons ook hiertegen te beschermen, kunnen we verschillende zaken doen.

65 Zie : http://www.centos.org/docs/2/rhl-cg-en-7.2/rescuemode.html

66 Voor het paswoord dat de toegang tot het BIOS moet afschermen, zal dat zelfs relatief makkelijk gaan, omdat de lengte ervan sowieso vrij beperkt is.

67 Zoals bijvoorbeeld met het pakket *Linux Unified Key Setup* (LUKS), zie http://luks.endorphin.org/

68 Eventueel kunnen we ons beperken tot het encrypteren van de (kleinere) opstart-partitie, zonder dewelke het systeem sowieso niet kan opstarten.

4.2.13.4. SELinux, iptables en Intrusion Detection

Zo kunnen we eerst en vooral gebruik maken van *Security Enhanced Linux* (SELinux[69]), een systeem dat toestemmingen verleent op grond van zogenaamde *Mandatory Access Control* beperkingen. Standaard wordt CentOS geïnstalleerd met inbegrip van SELinux. Het installeren van virtualiseringsoplossingen op serverniveau terwijl SELinux volledig actief is, is echter zo goed als onmogelijk. Vandaar dat we bij onze installatie SELinux hebben ingesteld op *permissive*. Hierdoor kunnen we via de *logs* wel in het oog houden of er iets gebeurd dat volgens de beperkingen van de *Mandatory Access Limits* eigenlijk niet toegestaan is. Wanneer de virtualiseringsoplossing dan geïnstalleerd en geconfigureerd is en in zoverre er geen alarmerende of onverklaarbare meldingen te vinden zijn in de *logs*, dienen we uiteraard het niveau van bescherming door SELinux opnieuw op volledig (*enabled*) in te stellen. Hetzelfde dienen we uiteraard ook te doen met de firewall van Linux.

Daarnaast kunnen we ook een zogenaamd *Intrusion Detection System* installeren. Onder Linux zijn er zo verschillende, zoals het met CentOS meegeleverde *Aide*[70] en het meer bekende *Tripwire*[71]. Dankzij dergelijke software kunnen we in het oog houden of en verwittigd worden dat er met onze bestanden iets ongeoorloofd is gebeurt.

Voor wat betreft de beveiliging van ons gehele netwerk kunnen we de snel aan bekendheid winnende software *PacketFence*[72] installeren. Dergelijk systeem is een zogenaamd *Network Access Control system*, waarmee we onder meer het al dan niet inzetten van apparaten op of in ons netwerk kunnen controleren of beperken.

Virtualisering op serverniveau brengt eigen, specifieke beveiligingsproblemen met zich mee. In de eerste plaats is daar de moeilijkheid dat dergelijke oplossingen bijna niet geïnstalleerd kunnen worden wanneer er een firewall en een SELinux-configuratie actief zijn. Vandaar ook dat we een eigen, lokale repository hebben gecreëerd. We kunnen dan de oplossingen installeren met uitschakeling van firewall en SELinux, zonder dat ons testsysteem verbonden moet zijn met een actieve verbinding met het Internet. Na afloop kunnen we SELinux en de IPtables-firewall dan terug actief maken (na configuratie, uiteraard).

4.2.13.5. *Blue Pill rootkits* versus certificaten

Daarmee is de kous echter niet af. Theoretisch is het immers mogelijk dat een malicieuse derde partij een door hemzelf gecreëerde virtuele machine binnensmokkelt op ons systeem. We zouden in dat geval te maken hebben met een zogenaamde *blue pill attack*[73]. Hiermee worden we feitelijk opgezadeld met een zo goed als onzichtbare *rootkit*[74]. Ons systeem zou dan ook volledig gecompromitteerd zijn. Om dergelijk scenario zo veel als mogelijk te vermijden, gebruiken de meeste virtualiseringsoplossingen op serverniveau een certificatiesysteem. Virtuele machines kunnen dan enkel actief gemaakt worden (i.e. opgestart worden) als ze voorzien zijn van een goedge-

69 Een beveiligingsarchitectuur, ontworpen door het Amerikaanse *National Security Agency*. Zie: http://www.nsa.gov/selinux/
70 Zie : http://www.bofh-hunter.com/2008/04/10/centos-5-and-aide/
71 Zie : http://sourceforge.net/projects/tripwire
72 Zie : http://www.packetfence.org/english/home.html
73 Zie : http://theinvisiblethings.blogspot.com/2006/06/introducing-blue-pill.html
74 Zie : http://en.wikipedia.org/wiki/Rootkit

keurd, geldig certificaat. Elke virtualiseringsoplossing heeft hiertoe zijn eigen methode ontwikkeld (hoewel deze wel op elkaar lijken). Hierbij is echter grote voorzichtigheid geboden, getuige de recent ontdekte onvolkomenheid van het certificatiesysteem in de Debian-versie van *OpenSSL*[75], waardoor het genereren van willekeurige nummers voorspelbaar en dus ontcijferbaar werd. De betrouwbaarheid van dergelijke certificaten staat of valt ook met de betrouwbaarheid van de zogenaamde *certification authority*. Voor onze kleine vzw is dit wellicht minder een probleem, maar voor grote productieomgevingen is het feit dat de meeste virtualiseringsoplossingen werken met zelf gegenereerde certificaten toch wel een zwak punt. Naast verplichte certificering voor virtuele machines, kunnen we ook best de toegang tot ons hostsysteem zelf volledig beperken tot de *root-user*.

4.2.13.6. Beveiligen der virtuele machines zelf

Zelfs indien we zeker zijn dat elke virtuele machine op ons systeem een degelijk gecertificeerd systeem is, dan nog is het beveiligingsverhaal niet af. Elke virtuele machine op zich is immers ook een volwaardige computer, die bijgevolg ook zelf volwaardig beveiligd dient te worden. Met andere woorden, alles wat geldt voor de beveiliging van het hostsysteem, geldt evenzeer voor de verschillende gastssystemen. Ook op hen is dus de volledige beveiligingsstrategie van toepassing, van hun virtueel BIOS, over hun virtuele *bootloader*, de toegang tot hun virtuele harde schijf en hun virtueel netwerk, enz. Uiteraard betekent dit enorm veel werk inzake installatie, configuratie, monitoring én beheer door de adminstrator[76].

4.2.13.7. Versleutelde verbindingen

Ook het verkeer tussen de diverse virtuele machines enerzijds én het verkeer over het netwerk in het geheel dient zoveel mogelijk beveiligd te worden, zoniet lopen we het risico op een zogenaamde *man in the middle attack* (i.e. een aanval waarbij de aanvaller zich tussen twee elkaar vertrouwende partijen voordoet als één van deze betrouwbare anderen) . Die beveiliging kunnen we verkrijgen door dat verkeer altijd geëncrypteerd te laten verlopen via de zogenaamde *Secure Shell* (SSH) en zo mogelijk ook via een versleuteld virtueel netwerk of Virtual Private Networking (VPN).

4.2.13.8. *Brute force* versus Single Packet Authorization

Helaas is het zo dat zelf dergelijke versleutelde verbindingen niet volledig onkwetsbaar zijn. Veel hangt af van de manier waarop gebruikers omgaan met hun paswoorden. Vele gebruikers kiezen al te makkelijke paswoorden, die dan snel ontcijferd kunnen worden door middel van een zogenaamde *brute force attack* (i.e. het achterhalen van het paswoord door herhaaldelijk proberen). Om dat probleem te omzeilen, kunnen we in onze firewall beperkingen inbouwen inzake het maximaal aantal keren dat een foutief paswoord ingegeven kan worden. Als de gebruikers echter het belang van een veilig paswoord sowieso al niet inzien, dan zullen ze in geval van een geweigerde

75 Zie : http://www.itnews.com.au/News/76080,openssl-bug-found-in-debian-linux.aspx en http://www.linux.com/feature/135270

76 Vandaar dat voor grote productie-eenheden aangeraden wordt het beheer van de verschillende virtuele machines te delegeren aan telkens een andere administrator.

verbinding bij het meermaals ingeven van een fout paswoord al snel bijzonder geërgerd raken.

Een mogelijkheid om hieraan te verhelpen is het voorzien in een configuratie voor *port knocking*[77]. Daarbij worden alle (maar dan ook alle !) poorten van de server doorheen de configuratie van de firewall gesloten, óók die voor de *ssh-daemon*. Voor een buitenstaander lijkt het netwerk daardoor onbereikbaar. Onze gebruikers kunnen vanop afstand echter wel toegang verkrijgen, door in een bepaalde volgorde een aantal vooraf bepaalde, niet-toegankelijke poorten aan te spreken. Op de server worden deze (uiteraard mislukte) pogingen gelogd. Als de pogingen beantwoorden aan de vooraf vastgelegde volgorde, dan wordt de SSH-poort door de server opengesteld voor deze gebruiker, die dan met zijn of haar paswoord kan inloggen. Deze toegang-in-schuifjes-aanpak verhindert een eenvoudige *brute force attack*. Volledig veilig is het echter nog steeds niet. Derden kunnen de sequentie van het op de deur kloppen immers achterhalen om ze vervolgens te imiteren. Daar kan *Single Packet Authorization* een stokje voor steken. Hierbij wordt één enkel, geëncrypteerd pakket aangeboden aan één enkele, vooraf in de firewall ingestelde poort. Komt dit specifieke pakket, met die specifieke encryptie aan bij die specifieke poort, dan zal de firewall de poort voor de *ssh-daemon* openstellen voor die gebruiker, die dan via zijn paswoord toegang kan verkrijgen. Hierdoor wordt de mogelijk gemonitorde sequentie van het traditionele *port knocking* onmogelijk gemaakt. Bovendien zorgt dergelijke aanpak ook voor veel minder netwerkverkeer. Voor de gebruiker zal één en ander ook sneller verlopen.

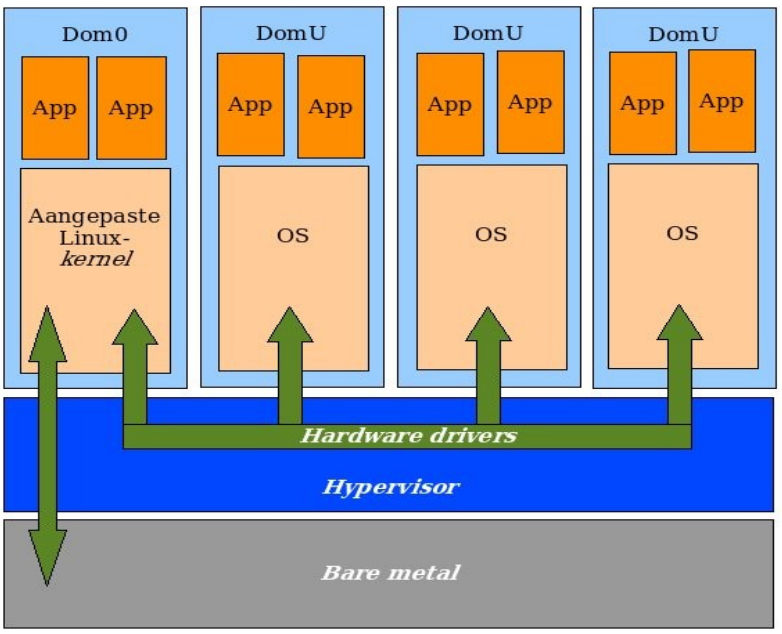
Na al deze voorbereidingen kunnen we dan nu eindelijk de verschillende virtualiseringsoplossingen van naderbij gaan bekijken.

77 Zie KEMP J., 'Who's There ? - Remote access security with single-packet port knocking', in Linux Magazine, June 2008, pagina's 24-28 (Britse uitgave)

5. Oplossingen

5.1. Oplossing 1 : Xen

Onder de verschillende virtualiseringsoplossingen behoort Xen tot de categorie der *hypervisors*. Dit wil zeggen dat er geen besturingssysteem is dat de hardware van de computer volledig bestuurt. In plaats daarvan draait er bovenop het naakte metaal – in ring -1 – een speciale softwarelaag : de *hypervisor*. Bovenop de *hypervisor* draaien virtuele machines, die *domains* worden genoemd. Er is altijd minstens één *domain* actief, het zogenaamde *domain*-0 (*dom0*). Dit *dom0* is ook weer een speciaal, want geprivilegieerd domein. Het gaat om een aangepaste (patched) Linux-kernel. Deze aangepaste Linux-kernel draait in Ring-3 (user space) en staat als enig domein via zijn drivers in voor het aansturen van de hardware. Hardware-aanroepen vanuit de niet-geprivilegieerde *domains* worden omgeleid naar het geprivilegieerde dom0, dat dan de werkelijke aanroeping van de hardware verzorgt via zijn drivers. Hiertoe beschikt dom0 over zogenaamde *back-end device drivers*. In de andere *domains* zijn dan weer zogenaamde *front-end device drivers* aanwezig, die optreden als aanspreekbare *interfaces* voor het besturingssysteem in de virtuele machine. De *hypervisor* zelf staat in voor het toewijzen van het deel van het geheugen waarover de domeinen mogen beschikken en voor het verdelen van de computerbronnen (bijvoorbeeld de processortijd) over de verschillende virtuele machines. Met andere woorden : de *hypervisor* staat in voor de isolatie van de verschillende virtuele machines of domains. Vereenvoudigd kunnen we dit als volgt voorstellen :



Afbeelding 11 : Virtualisering met een hypervisor als Xen

5.1.1. Verschillende versies van Xen

Er bestaan verschillende uitvoeringen van het virtualisatiesysteem Xen. Zo is er een door Citrix[78] verspreide Xen-versie (versie 4.1), die onder drie benamingen beschikbaar wordt gesteld onder een commerciële licentie. Het gaat meer bepaald om XenExpress, XenStandard en XenEnterprise. Geen van deze uitvoeringen is *open source*.[79] Verder gebruiken ook bedrijven zoals Sun Microsystems en Oracle Xen als basis voor hun eigen virtualiseringsoplossingen[80]. Daarnaast zijn er ook verschillende uitvoeringen van de *open source* versies van Xen beschikbaar. Dit is nogal een ingewikkelde zaak.

Zo heeft elke grote Linux-distributie een eigen, reeds gecompileerde (*binary*) versie van Xen in haar repository opgenomen. Deze versies verschillen echter in meer of mindere mate van elkaar[81], net zoals Linux-distributies in het algemeen ook van elkaar verschillen. Deze distributie-versies zijn gebaseerd op de stabiele tak van de Xen-ontwikkelboom. De nummering die de Linux-distributeurs eraan geven, loopt weleens uiteen. In het algemeen kan je echter stellen dat het hier momenteel gaat om versie 3.1 van Xen, waarvan telkens 32- en 64bits-versies bestaan.

Naast de distributie-versie(s) stelt XenSource[82] zelf ook verschillende Xen-versies ter beschikking. Het gaat daarbij om de huidige testing-tak van de Xen-ontwikkelboom (Xen 3.2). Hiervan worden zowel gecompileerde (*binary*) bestanden als broncodebestanden[83] aangeboden. De gecompileerde of binaire versies van XenSource (die beschikbaar zijn voor Linux-versies van Red Hat, Suse en Debian) zijn 32bits-versies. De versies in broncode kunnen ook gecompileerd worden op en voor 64bits-machines.

Tenslotte is er ook nog de *cutting edge*-versie. Het gaat daarbij om de Xen-versie die momenteel in ontwikkeling is.

Aangestipt dient te worden dat Xen-3.2 (64-bit) momenteel ter beschikking staat als zogenaamde *release candidate*. Die versie zal dan ook relatief snel worden opgenomen in de verschillende Linux-distributies.

Alle versies van Xen kunnen overweg met PAE[84] en met de virtualiseringsextensies van Intel-compatibele processoren[85].

78 Citrix nam onlangs XenSource (het bedrijf dat Xen ontwikkeld) over. Zie : http://domain-b.-com/companies/companies_c/citrix_systems/20070821_acquires.htm

79 XenExpress is wel gratis voor persoonlijk gebruik, met maximaal 4 virtuele servers en lijkt daarmee op het verspreidingsmodel van VMWare Server. XenStandard en XenEnterprise zijn te betalen.

80 Sun Microsystems stelt xVM ter beschikking (http://sun.com/xvm) onder een zogenaamde *dual licence* (d.w.z. zowel *open source* voor Linux en OpenSolaris als – betalend – propriëtair voor SunSolaris). Oracle heeft dan weer het *open source* OVM in de aanbieding (http://www.oracle.com/technologies/virtualization/index.html) onder Linux.

81 Handleidingen voor het werken met bijvoorbeeld een Debian- of Gentoo-gebaseerde versie van Linux kunnen dan ook niet zomaar worden toegepast op een installatie van Xen onder CentOS, Suse of Red Hat, net zomin als omgekeerd.

82 http://www.xensource.com

83 Een binaire versie of *binary* is een vooraf reeds gecompileerde versie. *Non-binaries* zijn broncodebestanden die je zelf dient te compileren.

84 PAE : *Physical Address Extension*. Een techniek die het 32bits-machines mogelijk maakt om meer dan 4 GB RAM-geheugen te adresseren.

85 De recente versies van de Intel-processoren zijn uitgerust met zogenaamde VMX-extensies

Concreet zullen wij niet alle versies met elkaar vergelijken. We zullen ons beperken tot enerzijds de distributie-versie (Xen 3.1) en anderzijds de broncode-versie van Xen 3.2.

5.1.2. Netwerken onder Xen[86]

Xen kent verschillen manieren om om te gaan met computernetwerken. We onderscheiden achtereenvolgens bridged networking, routed networking en virtual local networking met NAT.

5.1.2.1 *Bridged Networking*

Standaard wordt Xen ingesteld voor bridged networking. Deze techniek wordt gebruikt om verschillende netwerk-segmenten met elkaar te verbinden. Daartoe creëert Xen zelf de benodigde bruggen *(bridges)*, waaraan de virtuele netwerkkaarten van de virtuele machines verbonden worden. De brug (of bruggen) zelf verbinden twee (of meer) *local area networks* (LAN's) met elkaar. Daarbij worden *frames* doorgestuurd op basis van het MAC-adres[87]. Dat betekent dus dat de communicatie gebeurt op het niveau van de *datalink*-laag van het OSI-model[88]. Het Internet protocol (IP) komt er dan ook niet aan te pas. Adressering gebeurt op basis van het MAC-adres en gebruikt daarvoor broadcasting om niet gekende netwerkapparatuur te vinden. Als het apparaat gevonden wordt, wordt het MAC-adres opgeslagen in de bridge table.

(voorheen bekend onder de codenaam Vanderpool), terwijl de AMD-processoren gebruik maken van de zogenaamde VMD-extensies (codenaam Pacifica). Aangestipt dient te worden dat beide technieken niet volledig identiek zijn, waarbij AMD's (iets langer bestaande) methode volgens de mailing-list van Xen-gebruikers iets beter uitgebalanceerd lijkt.

86 Zie hiervoor ook hoofdstuk 5 in CHAGANTI P., '*Xen Virtualization – A Fast and Practical Guide to Supporting Multiple Operating Systems with the Xen Hypervisor*', Packt Publishing Ltd., 2007. Dit boek is zeer verdienstelijk, maar bevat helaas ook nogal wat fouten, zowel grammaticaal als in de code-voorbeelden. Het is ook reeds enigszins verouderd, want gebaseerd op Xen 3.0. De bespreking van Xen's netwerkmogelijkheden in dit boek steunt op het TCP/IP-model, terwijl wij hier uitgaan van het OSI-model.

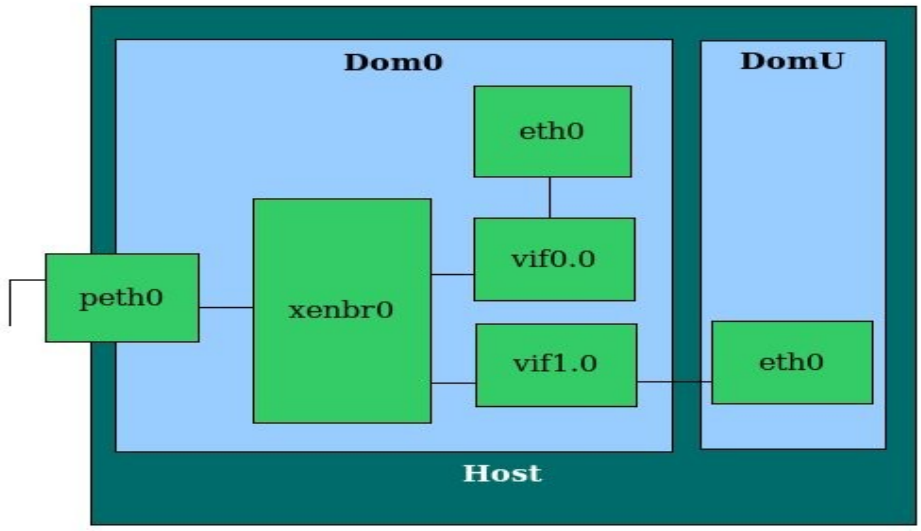
87 Een virtuele netwerkkaart kan natuurlijk niet anders dan een virtueel MAC-adres hebben. Zo'n MAC-adressen kunnen *at random* of willekeurig worden gecreëerd door Xen, wat echter het nut van een MAC-adres grotendeels, zoniet volledig teniet doet, aangezien er bij het herstarten van de virtuele machine telkens een ander MAC-adres zal worden aangemaakt. Wij zullen zelf MAC-adressen genereren, die we dan zullen 'vastmaken' aan de desbetreffende virtuele machine. Dit genereren doen we met behulp van het volgende eenvoudige python-script :

```
#!/bin/bash

#Dit script maakt random MAC-adressen aan voor Xen

python -c 'import random; r=random.randint; print "00:16:3E:%02X:%02X:%02X" % (r(0, 0x7f),
r(0, 0xff), r(0, 0xff))' >> /root/Desktop/MAC.txt
```

De beginaanduiding 00:16:3E: van de Xen-MAC-adressen is een OUI (*Organizationally Unique Identifier*), die toegekend werd aan Xensource, Inc. De OUI-lijst is raadpleegbaar op http://standards.ieee.org/regauth/oui/oui.txt

88 Zie http://nl.wikipedia.org/wiki/OSI-model

Eenvoudig gesteld betekent dit dat er van de fysieke netwerkkaart een (onzichtbaar) bruggetje wordt gelegd naar de virtuele netwerkkaart van de virtuele machine. Bekeken vanuit het standpunt van de virtuele machine (en onze huidige Linux-omgeving is reeds zo'n virtuele machine !) is er maar één netwerkkaart, met name eth0. De werkelijke netwerkkaart wordt omgedoopt tot peth0 (physical ethernet 0). Deze manier is zeer eenvoudig en staat toe om snel aan de slag te gaan. Schematisch kunnen we dit zo voorstellen :



Afbeelding 12 : *Bridged* netwerk onder Xen

Om Xen in te stellen voor bridged networking moeten we de configuratie van Xen aanpassen. Daartoe moeten we eerst de xen-service (xend) stoppen :

```
[root@testmachien ~]# service xend stop
```

Vervolgens openen we met een editor het bestand /etc/xen/xend-config.sxp :

```
[root@testmachien ~]# vi /etc/xen/xend-config.sxp
```

We halen de commentaaraanduiding (#) weg bij deze twee regels :

```
(network-script network-bridged)

(vif-script vif-bridge)
```

Door in het algemene configuratiebestand van Xen (xend-config.sxp) deze twee regels actief te maken, kunnen de nodige scripts voor bridged networking worden aangeroepen en uitgevoerd.

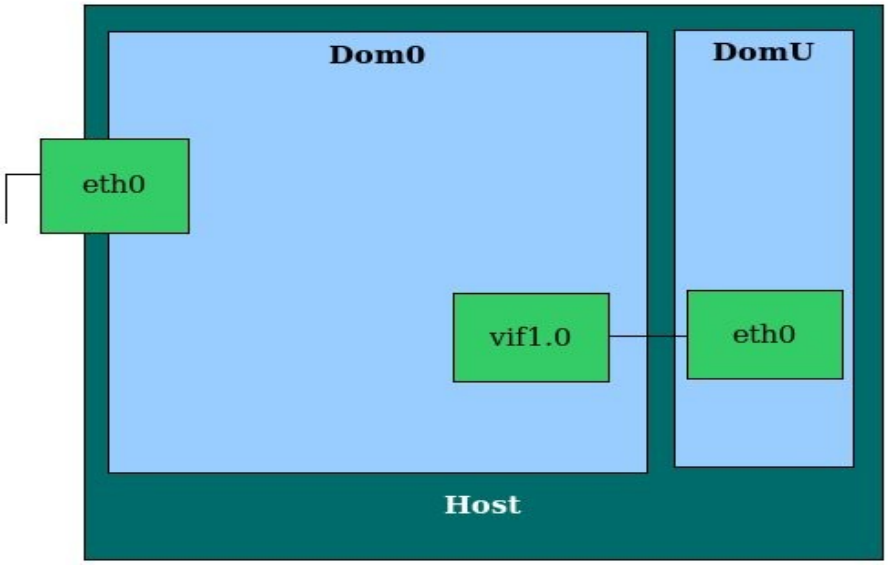
Daarna starten we de xen-service weer op :

```
[root@testmachien ~]# service xend start
```

In de configuratiebestanden voor elk van onze virtuele machines (zie verder) dienen we nu aan te geven welk MAC-adres (of adressen) die bepaalde virtuele machine dient te gebruiken.

5.1.2.2. *Routed Networking*

Bij *routed networking* zitten we niet meer op het niveau van de datalinklaag, maar wel op dat van de netwerklaag. Hier wordt IP *forwarding* gebruikt om het ene netwerksegment te doen communiceren met het andere. Het geprivilegieerde Domain-0 treedt daarbij zelf op als router tussen de verschillende virtuele machines onderling en tussen deze machines en het netwerk waarin de Xen-machine opereert. Domain-0 onderhoudt dan ook een eigen *routing table*. In plaats van via *broadcasting*, wordt hierbij gebruik gemaakt van *unicasting*. Schematisch kunnen we dit zo voorstellen :



Afbeelding 13 : Gerouteerd netwerk onder Xen

Als we gebruik willen maken van *routed networking*, dan moeten we de configuratie van Xen aanpassen. Daartoe moeten we eerst de xen-service (xend) stoppen :

```
[root@testmachien ~]# service xend stop
```

Vervolgens dienen we het configuratiebestand van Xen weer aan te passen :

```
[root@testmachien ~]# vi /etc/xen/xend-config.sxp
```

We commentariëren de regels omtrent bridged networking weer weg door er een hekje voor te plaatsen en we halen het hekje weg bij de volgende regels :

```
(network-script network-route)

(vif-script vif-route)
```

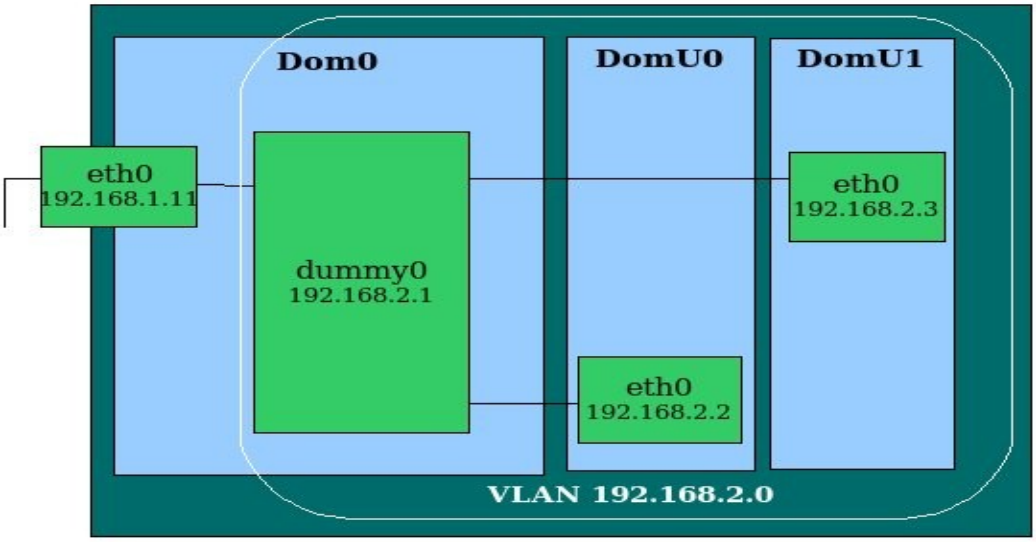
Daarna starten we de xen-service weer op :

```
[root@testmachien ~]# service xend start
```

In de configuratiebestanden voor elk van onze virtuele machines (zie verder) dienen we nu aan te geven welk IP-adres (of adressen) die bepaalde virtuele machine dient te gebruiken.

5.1.2.3. VLAN met NAT

Onder *routed networking* heeft elke virtuele netwerkkaart van elke virtuele machine een eigen IP-adres. Wanneer we VLAN met NAT toepassen verandert dat. Enkel Domain-0 heeft een publiek IP-adres. De virtuele machines hebben ook IP-adressen, die echter niet toegankelijk zijn van op het 'normale' netwerk. Als er bij Domain-0 een verzoek binnenkomt met het IP-adres van Domain-0, maar voor een welbepaalde poort, dan zal er via *address translation* voor gezorgd worden dat dat verzoek wordt gestuurd naar de virtuele machine die ingesteld staat voor het gebruik van die poort. Stel bijvoorbeeld dat we een virtuele machine geconfigureerd hebben als webserver, dan zullen de verzoeken voor toegang tot poort 80 van domain-0 toegezonden worden aan de virtuele machine waarop de webserver draait. Door in de configuratie der virtuele machines bedachtzaam om te springen met de toewijzing van poorten, kunnen we zo aan adresvertaling doen. Schematisch kan dit er zo uit zien :



Afbeelding 14 : VLAN met NAT onder Xen

Om gebruik te maken van VLAN met NAT, dienen we allereerst een nieuwe network interface voor ons VLAN aan te maken. Die kunnen we bijvoorbeeld dummy0 noemen. We maken daartoe een nieuw bestand met volgende inhoud aan :

```
[root@testmachien ~]# vi /etc/sysconfig/network-scripts/ifcfg-dummy0

A DEVICE=dummy0

BOOTPROTO=none

ONBOOT=yes

USERCTL=no

IPV6INIT=no

PEERDNS=yes

TYPE=Ethernet

NETMASK=255.255.255.0

IPADDR=x.x.x.x

ARP=yes
```

Nu moeten we aan dit 'device' een IP-adres verbinden en dit toevoegen aan ons bestand met network devices :

```
[root@testmachien ~]# vi /etc/sysconfig/networking/devices

auto dummy0

iface dummy0 inet static

       address 192.168.2.1

       netmask 255.255.255.0
```

Vervolgens passen we weer het xen configuratiebestand aan, waardoor we de bridge binden aan de nieuwe virtuele netwerk interface dummy0 door er de volgende regel aan toe te voegen :

```
[root@testmachien ~]# vi /etc/xen/xend-config.sxp

(network-script 'network-bridge netdev=dummy0')
```

We zorgen ervoor dat IP-forwarding geactiveerd is :

```
[root@testmachien ~]# echo “1” > /proc/sys/net/ipv4/ip_forward
```

Dan herstarten we sysctl zodat alles actief wordt :

```
[root@testmachien ~]# /sbin/sysctl -p
```

Tenslotte herstarten we de xend-service :

```
[root@testmachien ~]# service xend restart
```

5.1.3. Xen als binair bestand installeren

We kunnen Xen installeren als een binair bestand, afkomstig van de *repository* van CentOS. Dat heeft echter het nadeel dat we dan een relatief oude versie (versie 3.0) zullen moeten installeren. Die versie kan niet overweg met *logical volumes* (zie ook verder) en dat is toch wel een grote beperking. CentOS beschikt echter ook over een zogenaamde 'plus' *repository*, waarin zich een meer recente Xen-versie bevindt. We gaan dan ook eerst deze plus-*repository* activeren. Dat doen we door handmatig het script CentOS-Base.repo in de directory /etc/yum.repos.d aan te passen. We veranderen daar onder de hoofding [centosplus] de waarde bij *enabled* van 0 in 1 en slaan het bestand weer op[89].

Niks is nu nog eenvoudiger dan het installeren van Xen als binair bestand. We geven daartoe het volgende commando :

```
[root@testmachien ~]# yum install kernel-xen xen virt-manager
```

Wie liever werkt via een grafische *interface*, kan dat ook. Klik dan in de Gnome Desktop op *Applications*, *Add/Remove Software* en vervolgens op het tabblad *Search*. Typ dan xen in de zoekbalk en selecteer vervolgens de eerder genoemde pakketten kernel-xen, xen en virt-manager. Kies wel de meest recente versienummers.

Dit commando installeert inderdaad de laatste stabiele versie van Xen (versie 3.1), samen met de aan Xen aangepaste Linux-kernel (die daarna dienst zal doen als geprivilegieerd domein - het zogenaamde Dom0). Daarnaast installeert dit commando in één keer ook alle software waarvan beide xen-pakketten afhankelijk zijn (de zogenaamde *dependencies* of afhankelijkheden).

We installeren verder dus ook de virt-manager, een grafisch hulpprogramma voor het creëren en beheren van virtuele machines, waarmee het makkelijk werken is. Dit programma is heel gebruiksvriendelijk, maar als we straks willen vergelijken hoe deze vooraf gecompileerde versie van Xen het ervan afbrengt tegenover de zelf gecompileerde (en meer recentere) versie, dan hebben we een probleem. De met de distributies meegeleverde virt-manager blijkt immers niet compatibel met de nieuwste Xen-versie. Daarom zullen we de virt-manager straks weer moeten verwijderen[90].

Xen wordt nu samen met zijn *dependencies* geïnstalleerd. Het installatieprogramma heeft er ook voor gezorgd dat het opstartmenu van de GrUB-bootmanager[91] werd uitgebreid met een nieuwe optie, met name die voor de Xen-kernel. Dat gebeurt echter op een zogenaamde *failsafe* manier. De standaardwaarde blijft immers ingesteld staan op de 'gewone' Linux-kernel. We kunnen dan ook maar beter de GrUB-configuratie aanpassen. Dat doen we als volgt :

89 De lezer zal hierbij opmerken dat het toch wat vreemd is dat we nu pas deze plus-*repository* activeren. Zo heeft het eerder aanmaken van onze lokale repository blijkbaar toch niet veel zin gehad ? De kwestie is dat we na het installeren van de Xen-kernel deze plus-repository weer ontoegankelijk zullen maken. We willen immers niet dat bij elke (overigens veel voorkomende) update van die Xen-kernel ons systeem opnieuw moeten opgestart. In plaats daarvan zullen we gebeurlijke updates eerst testen en daarna pas – handmatig - installeren.

90 Virt-manager wordt naast en apart van Xen ontwikkeld door Red Hat. Het is dan ook perfect mogelijk de broncode van virt-manager te (her)compileren voor een recentere Xen-versie.

91 GrUB staat voor ***Gr**and **U**nified **B**oot manager*.

```
[root@testmachien ~]# vi /boot/grub/grub.conf
```

Na de aanpassingen, ziet het resultaat er als volgt uit (onze wijzigingen hebben we gecursiveerd, onderlijnd en een grijze achtergrond gegeven – ook hebben we noodgedwongen de layout wat moeten aanpassen):

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /boot/, eg.
#          root (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/VolGroup00/LV_root
#          initrd /initrd-version.img
#boot=/dev/md0
default=0
timeout=10
splashimage=(hd0,0)/grub/splash.xpm.gz
#hiddenmenu
title Xen (2.6.18-53.1.14.el5xen)
        root (hd0,0)
        kernel /xen.gz-2.6.18-53.1.14.el5 dom0_mem=512M
        module /vmlinuz-2.6.18-53.1.14.el5xen ro
                        root=/dev/VolGroup00/LV_root rhgb quiet
        module /initrd-2.6.18-53.1.14.el5xen.img
title CentOS (2.6.18-53.1.14.el5)
        root (hd0,0)
        kernel /vmlinuz-2.6.18-53.1.14.el5 ro
        root=/dev/VolGroup00/LV_root rhgb quiet
        initrd /initrd-2.6.18-53.1.14.el5.img
```

```
title CentOS (2.6.18-53.el5)
        root (hd0,0)
        kernel /vmlinuz-2.6.18-53.el5 ro root=/dev/VolGroup00/LV_root \
                rhgb quiet
        initrd /initrd-2.6.18-53.el5.img
~

"/boot/grub/grub.conf" 26L, 1041C
:wq
```

Xen zal bij het herstarten van de computer na 10 seconden automatisch worden opgestart door de GrUB-bootmanager. Dat geeft ons wat tijd om eventueel toch de 'gewone' Linux-kernel te kiezen, mochten we dat verkiezen of nodig hebben. We hebben ook het geheugengebruik van de Xen-enabled Linux-kernel beperkt tot 512 MB, zodat de rest van ons RAM-geheugen beschikbaar is voor de virtuele machines[92].

We moeten nu alleen nog de computer herstarten om met Xen te kunnen werken. Dat doen we dan ook :

```
[root@testmachien ~]# reboot
```

Na de herstart (waarbij we weer inloggen als 'root'), controleren we eerst of Xen wel degelijk gestart is en bijgevolg of we nu aan het werken zijn in onze eerste virtuele machine (de Linux die we nu voor ogen hebben, is inderdaad zelf niks anders dan een virtuele machine, zij het dan één met grote privilegies).

```
[root@testmachien ~]# uname -r

2.6.18-53.1.14.el5.centos.plusxen
```

We zien dan inderdaad dat de naam van de kernel eindigt op xen. Het is dus de kernel met xen-extensies. Dat het daarbij effectief gaat om een virtuele machine, blijkt als we het volgende commando ingeven :

```
[root@testmachien ~]# xm list
Name                                        ID Mem(MiB) VCPUs State   Time(s)
Domain-0                                     0      512     2 r-----    913.5
```

We zien dat er één Domain-0 actief is, met ID 0 en dat dit domein over 512 MB RAM-geheugen beschikt en van alle beschikbare processoren (twee, het is tenslotte een

92 We zouden het geheugen voor Xen nog veel verder naar beneden kunnen brengen, indien we geen grafische desktopomgeving zouden gebruiken.

dual core) gebruik maakt. Met het volgende commando zien we voor welke runlevels xen actief is :

```
[root@testmachien ~]# chkconfig --list | grep 3:on | grep xen

xend            0:off   1:off   2:on    3:on    4:on    5:on    6:off

xendomains      0:off   1:off   2:off   3:on    4:on    5:on    6:off
```

Hoewel we nu dus echt van start lijken te kunnen gaan met Xen, is dit toch niet helemaal waar. We beschikken immers nog niet over netwerkfunctionaliteit. Daarvoor moeten we eerst nog het algemene configuratiebestand van Xen aanpassen en daarin een netwerkconfiguratie kiezen[93]. We kiezen voor bridged networking. Hierboven hebben we reeds uitgelegd hoe dit in zijn werk gaat. Nadat we het configuratiescript hebben aangepast, herstarten we de xend-service :

```
[root@testmachien ~]# service xend restart
```

5.1.4. Soorten virtuele harde schijven onder Xen

We kunnen nu beginnen met het aanmaken van virtuele machines. Toch moeten we eerst nog enkele beslissingen nemen. In de eerste plaats moeten we beslissen hoe we virtuele harde schijven gaan creëren. Xen kan op drie manieren omgaan met virtuele harde schijven.

Ofwel gebruikt Xen een eenvoudig bestand als virtuele harde schijf. Dit is ook de manier waarop VMWare Server of Microsoft Virtual PC dit aanpakken. Aan deze manier kleeft echter een stevig nadeel : het is niet erg performant. Bovendien is het ook zo dat de omvang van dergelijke virtuele harde schijf onder Xen later niet meer makkelijk verandert kan worden.

Een andere mogelijkheid is om virtuele harde schijven te installeren als zogenaamde *raw partitions*. In dat geval is de virtuele harde schijf feitelijk een echte partitie van een echte harde schijf. Dit is de meest performante methode. Helaas is echter ook de grootte van dergelijke (niet zo virtuele) harde schijf later niet meer aanpasbaar.

De derde mogelijkheid is die waarbij Xen *logical volumes* zal gebruiken als virtuele harde schijf. Dit is ietwat minder performant dan het gebruik van een rauwe partitie, maar de grootte van dergelijke logical volumes is heel eenvoudig te veranderen, hetzij via de command line, hetzij via de grafische *Logical Volume Manager*. Deze methode geniet onze voorkeur. Bovendien hebben we ons testsysteem zo ingericht dat er reeds een *volume group* aangemaakt werd waarbinnen we erg makkelijk de nodige *logical volumes* kunnen aanmaken.

93 Na het herstarten van de machine kan het zijn dat er helemaal geen netwerkfunctionaliteit meer is. Dat heeft er blijkbaar mee te maken dat de 'verwisseling' van de fysieke netwerkkaart met de virtuele niet helemaal perfect verloopt. Er is verder echter niks aan de hand. We moeten dan ook enkel de netwerkinterfaces (her)starten, bijvoorbeeld als volgt :

```
[root@testmachien ~]# ifup eth0
```

Tot voor kort kon deze methode enkel gebruikt worden met nieuwere Xen-versies (d.w.z. met Xen-versies die we eigenhandig dienen te compileren)[94]. De Xen-versie die beschikbaar is in de CentOSplus-*repository* kan echter probleemloos om met deze aanpak.

5.1.5. Paravirtualisatie of volledige virtualisatie

Xen kan op twee verschillende manieren aan virtualisatie doen : paravirtualisatie of volledige virtualisatie. Het verschil tussen een *paravirtualized* systeem en een *fully virtualized* systeem bestaat eruit dat paravirtualisatie geen beroep doet op de hardwarematige virtualiseringsextensies van de processor en *full virtualization* wel. Tot voor kort was paravirtualisatie dan ook de enige mogelijkheid (processoren met virtualiseringsextensies bestonden immers niet)[95]. Onder paravirtualisatie wordt het geïnstalleerde virtueel besturingssysteem gepatched om te werken onder de Xen-hypervisor. Het gaat dus om een aan Xen aangepast systeem. De aanpassing waarvan sprake strekt er om zo te zeggen toe dat het gevirtualiseerd systeem bewust gemaakt wordt van het feit dat het gevirtualiseerd is. Dit kan uiteraard slechts met systemen waarvan de broncode beschikbaar is. Vandaar dat Windows niet kan geïnstalleerd worden als een geparavirtualiseerd systeem.

Paravirtualisatie wordt in de literatuur omschreven als performanter dan volledige virtualisering[96]. Met de nieuwe virtualiseringsextensies in de x86-processoren is het verschil echter nauwelijks waarneembaar geworden.[97]

We zullen zowel paravirtualisatie als hardwarematige (volledige) virtualisatie benutten.

5.1.6. Virtuele machines onder Xen 3.1

We kunnen virtuele machines aanmaken op meerdere manieren. Om te beginnen zullen we de grafische methode gebruiken[98]. Voor we daar echter mee van start gaan,

94 Zowel in het boek van CHAGANTI als in dat van BUYTAERT et. al. wordt beschreven dat je hiervoor eerst zelf een Xen-kernel moet compileren. Wij hebben echter ondervonden dat het ook kan met de laatste kernel van de CentOSplus-*repository* (Xen versie 3.1). In het algemeen is het zo dat de meeste boeken over Xen (die allemaal dateren uit 2007) nu al verouderd aandoen, aangezien ze allemaal steunen op Xen versie 3.0. Dat maakt die boeken daarom niet overbodig, maar het is soms wel uitkijken geblazen.

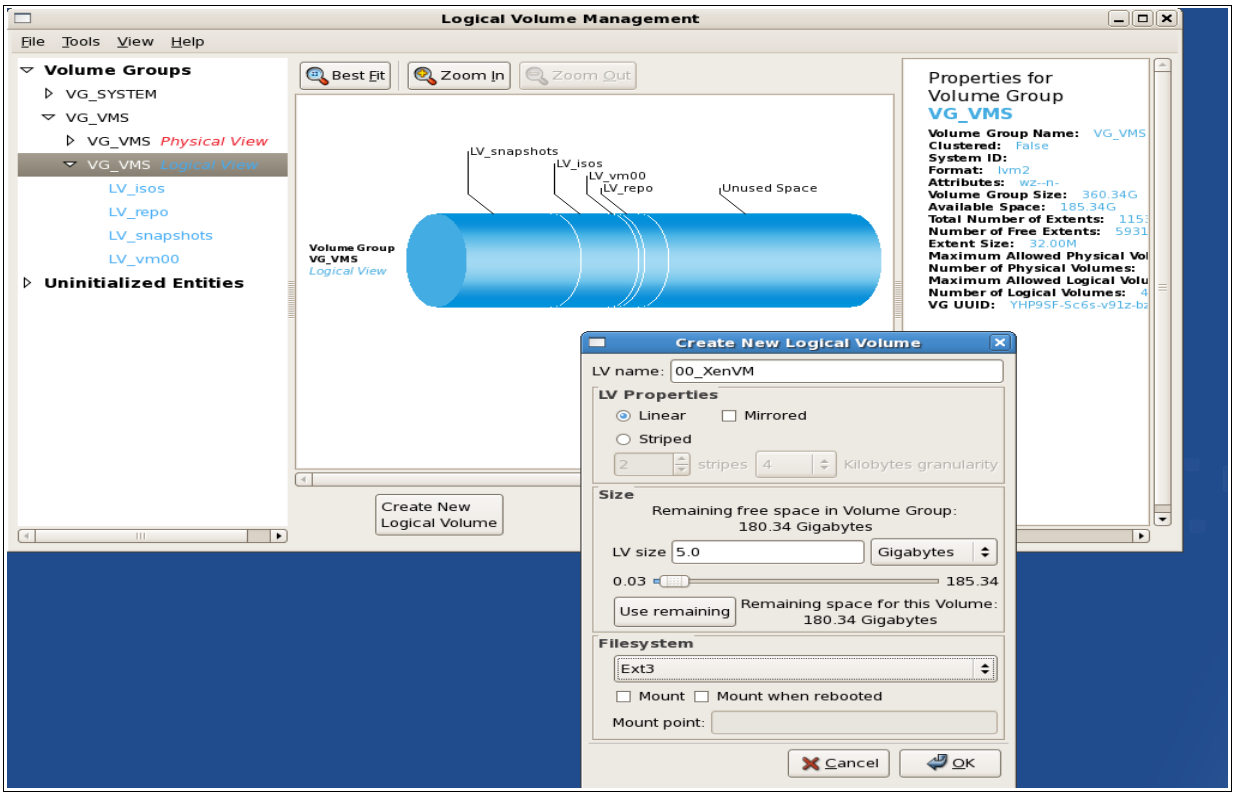
95 In vele beschrijvingen van Xen die we kunnen vinden op het Internet wordt Xen, samen met UML (i.e. User Mode Linux), als systeem tegenover alle andere gesteld, precies omdat het in staat is om te paravirtualiseren. Zie bijvoorbeeld dit vergelijkend overzicht van virtualiseringsmogelijkheden onder Linux : http://www-128.ibm.com/developerworks/library/l-linuxvirt/index.html

96 Zie http://www.linux-mag.com/id/1769

97 In de mailinglist van Xen-gebruikers wordt er regelmatig op gewezen dat het performantievoordeel van paravirtualisatie op recente hardware *"overroepen"* is.

98 De *command line* methode gaat eigenlijk makkelijker, maar omdat we later ook zullen werken met de zelf uit broncode te compileren Xen versie 3.2, waarbij de grafische virt-manager niet meer beschikbaar is, kunnen we beter op dat ogenblik de command line methode illustreren.

gebruiken we eerst de grafische *Logical Volume Manager* om een *logical volume* aan te maken, waarop we onze eerste virtuele machine zullen installeren[99].



Afbeelding 15 : Een nieuw logical volume aanmaken

Na enige tijd is het logical volume aangemaakt[100] en kunnen we van start gaan met de (grafische) virt-manager. Voor we echter aan de slag gaan met de virt-manager, moeten we eerst nog een probleem oplossen. Er zit namelijk een hardnekkige *bug* in Xen, die ervoor zorgt dat de *tx checksums* van UDP-pakketten gecorrumpeerd raken. We moeten dus, voor we beginnen met het installeren van virtuele machines, eerst die *tx checksumming* uitschakelen en dit zowel voor wat betreft de fysieke netwerkkaart, de virtuele netwerkkaart in domain-0 en de virtuele netwerkkaart die verbinding maakt met de bridge. Dit doen we als volgt :

```
[root@testmachien ~]# ethtool -K peth0 tx off

[root@testmachien ~]# ethtool -K eth0 tx off

[root@testmachien ~]# ethtool -K vif0.0 tx off
```

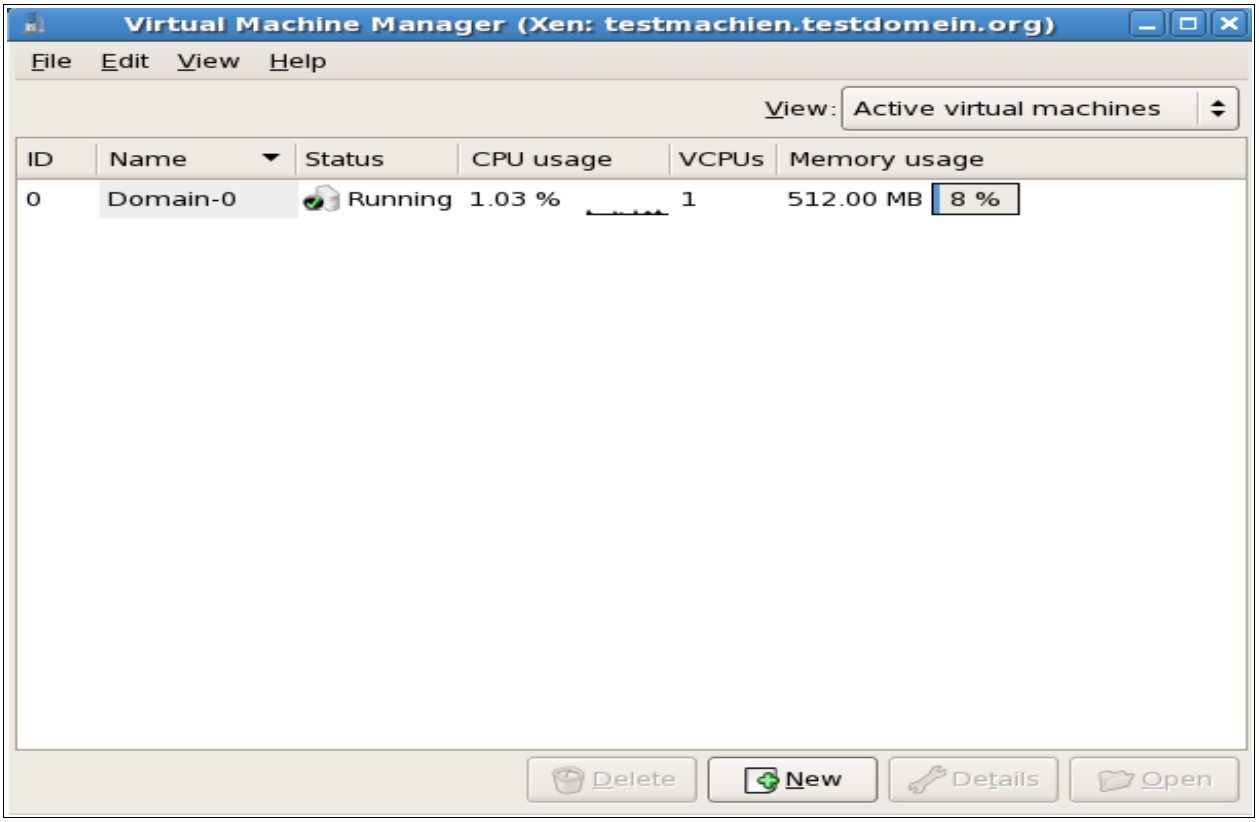
Daarna moeten we de DHCP-service herstarten :

99 We zouden dit ook via de command line kunnen doen (met het commando **`lvcreate -L 5G -n 00_XenVM VG_VMS`**), maar we willen ook de grafische *Logical Volume Manager* illustreren (die overigens zéér makkelijk werkt).

100 Het aanmaken van logical volumes via de command line gaat wel aanzienlijk sneller.

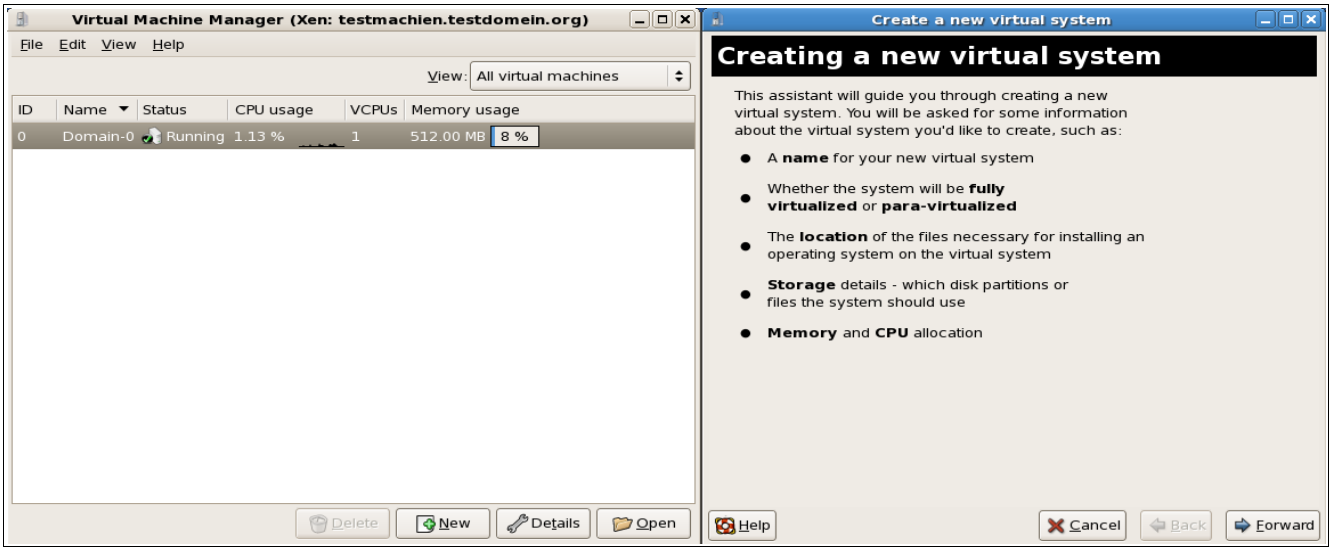
```
[root@testmachien ~]# service dhcpd restart
```

Nu kunnen we eindelijk de virt-manager opstarten. We klikken links bovenaan in het Gnome-panel op *Applications*, vervolgens op *System Tools* en tenslotte op *Virtual Machine Manager*. We kunnen overigens ook in een shell het commando ./**virt-manager** geven. De virt-manager-applicatie start op en toont ons in eerste instantie enkel het lopende dom0.



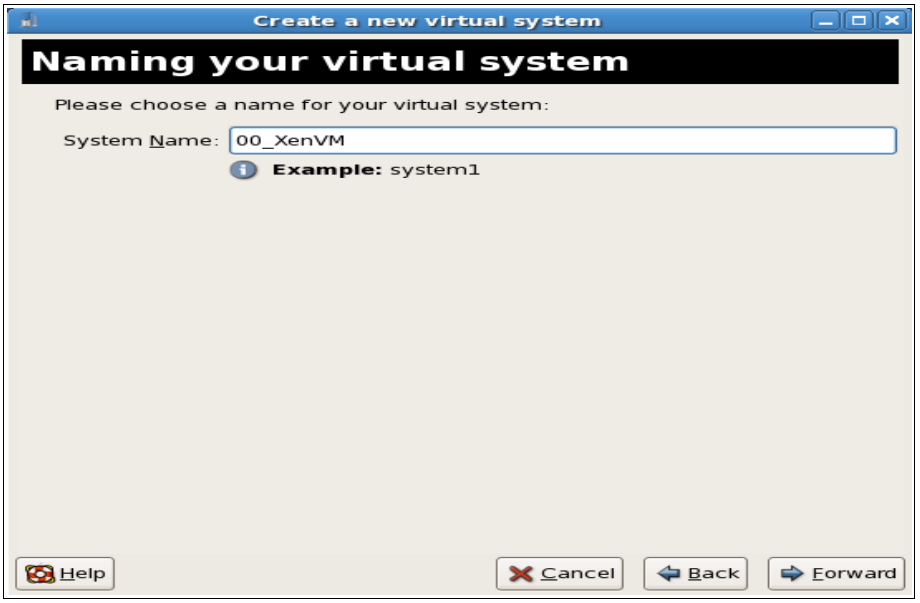
Afbeelding 16 : De virt-manager

We klikken onderaan in het scherm op *New*. Via een wizard-interface kunnen we nu de diverse parameters meegeven waarmee we ons eerste virtuele machine creëren.



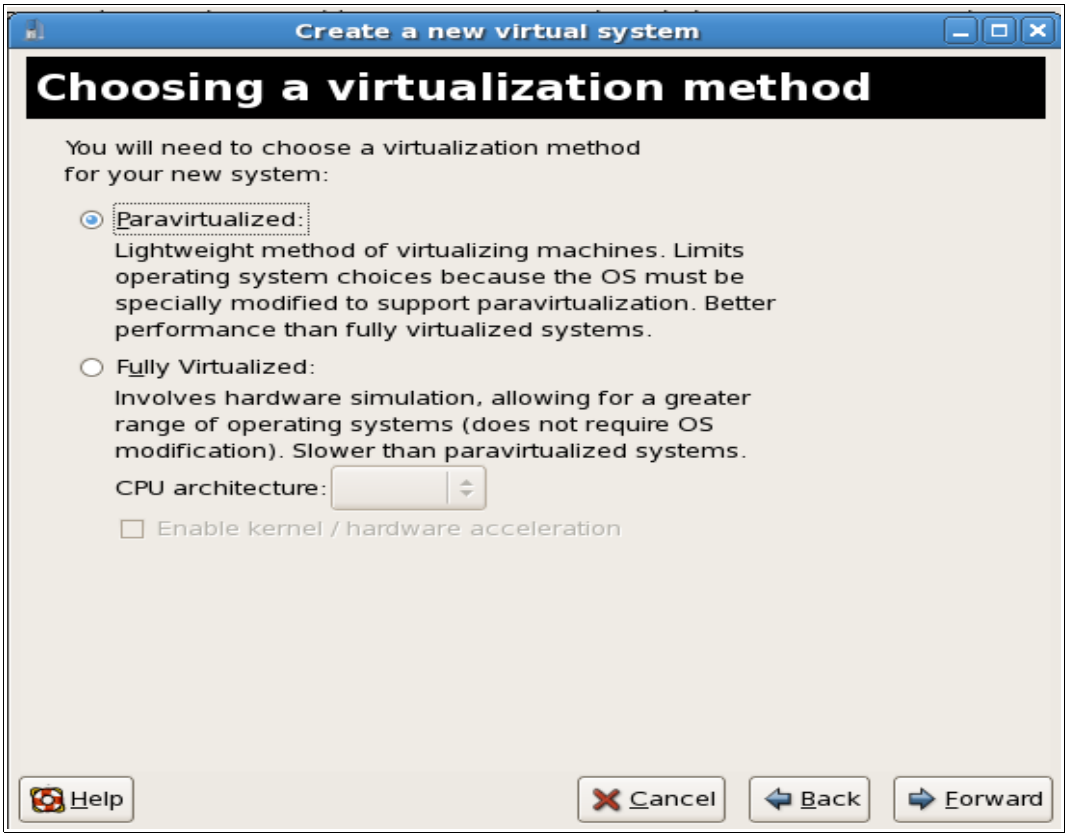
Afbeelding 17 : Een virtuele machine maken met virt-manager

Na het introductiescherm kunnen we nu een naam opgeven voor onze virtuele machine. We kiezen voor een naam die begint met het getal 00. Dit zal ons later van pas komen, wanneer we de verschillende virtuele machines zullen laten opstarten samen met het opstarten van de hostmachine. We kunnen dan namelijk een timing gebruiken, zodat niet alle virtuele machines gelijktijdig opstarten, maar wel de één na de ander. Dit is nuttig, omdat sommige virtuele machines andere moeten kunnen aanspreken (bijvoorbeeld : een DHCP-server). Dat kan niet als ze allemaal gelijktijdig opstarten. Vandaar deze nummeringsmethode. Ook geven we al onze virtuele machines een naam waarin Xen voorkomt, om ze straks (bij oplossingen 2 en 3) te kunnen onderscheiden van de virtuele machines onder OpenVZ en KVM.



Afbeelding 18 : Een (genummerde) naam voor de virtuele machine

Vervolgens geven we aan welke virtualiseringsmethode we verkiezen. We kiezen voor paravirtualisatie :

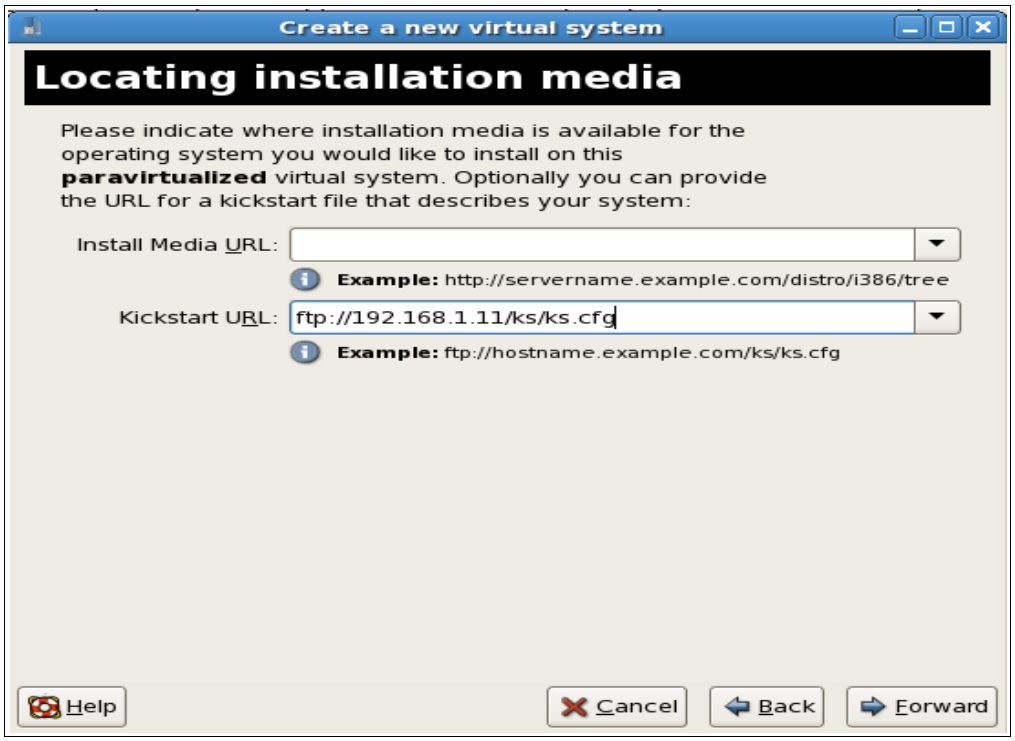


Afbeelding 19 : Een virtualisatiemethode kiezen

In het volgende scherm moeten we kiezen wat we als installatiebron willen gebruiken. We kunnen kiezen uit twee mogelijkheden. Ofwel gebruiken we een zogenaamd ISO-bestand, dan wel onze volledige eigen lokale *repository*[101] als installatiebron, ofwel gebruiken we een *kickstart*-script. Omdat CentOS een derivaat is van Red Hat Enterprise Linux, kunnen we gebruik maken van dergelijk kickstart-bestand om het creëren van virtuele machines te automatiseren. Voor onze allereerste virtuele machine zullen we gebruik maken van het *kickstart*-bestand *xen-build-script.sh*[102]. Dit script installeert volautomatisch, vanuit onze eigen, lokale *repository*, een volledig werkende, optimaal beveiligde, eerste virtuele machine, die dienst zal doen als authenticatieserver voor gebruikers en voor alle andere virtuele machines.

101 Naar keuze beschikbaar te stellen via HTTP, FTP of als NFS-share.

102 Het gaat om een licht aangepaste versie van het script van Faye GIBBINS (http://www.samag.com/documents/s=10112/sam0702e/0702e_l2.htm). Voor ons volledige script, zie Appendix X : Kickstart-script voor Xen VM.

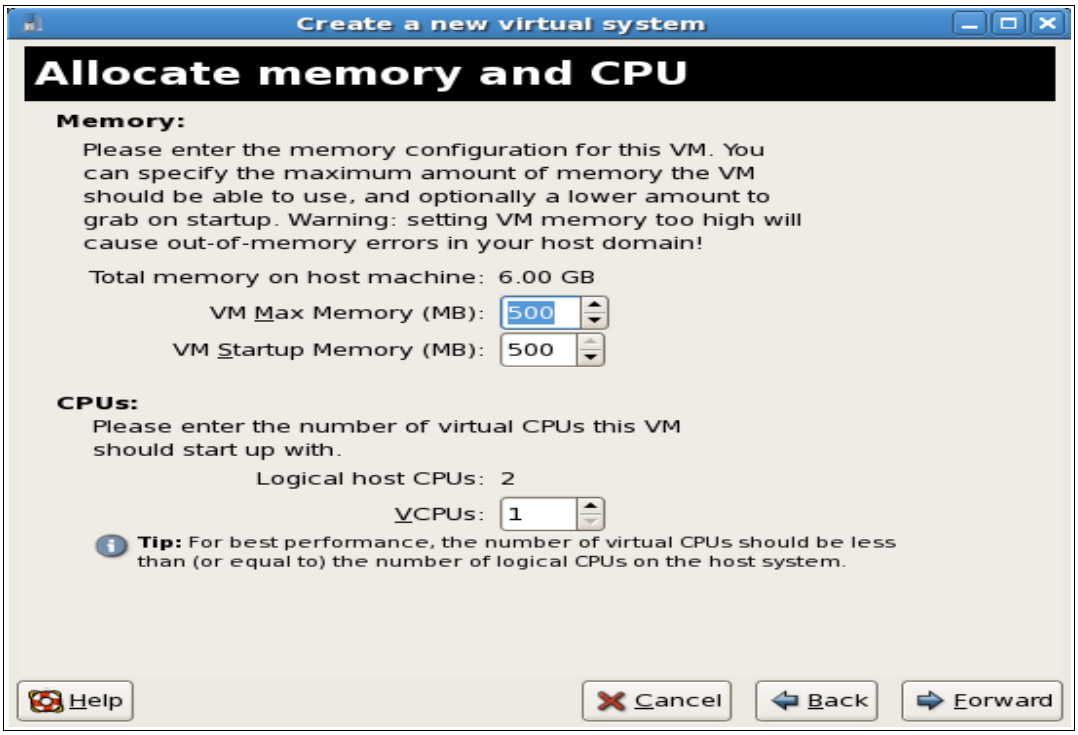


Afbeelding 20 : De installatiemedia kiezen

Dan moeten we opgeven waar de opslagruimte (i.e. de virtuele harde schijf) dient gecreeerd te worden. Die opslagruimte hebben we eerder reeds aangemaakt met de *Logical Volume Manager*. Zoniet, dan zouden we nu dat *logical volume* eerst nog moeten maken.

Afbeelding 21 : Een virtuele harde schijf aanmaken in een logical volume

Voor het netwerk behouden we de standaardwaarde, met name een *bridged* netwerk. Daarna wijzen we nog 512 MB RAM-geheugen toe en één virtuele processor.



Afbeelding 22 : Geheugen en processor toewijzen

Tenslotte bekijken we de samenvatting van onze instellingen en geven we door op *Finish* te klikken opdracht om tot installatie over te gaan.

Virt-manager vraagt ons nu om een paswoord. Dat geven we natuurlijk in.

Afbeelding 23 : Een paswoord opgeven voor de keyring-manager

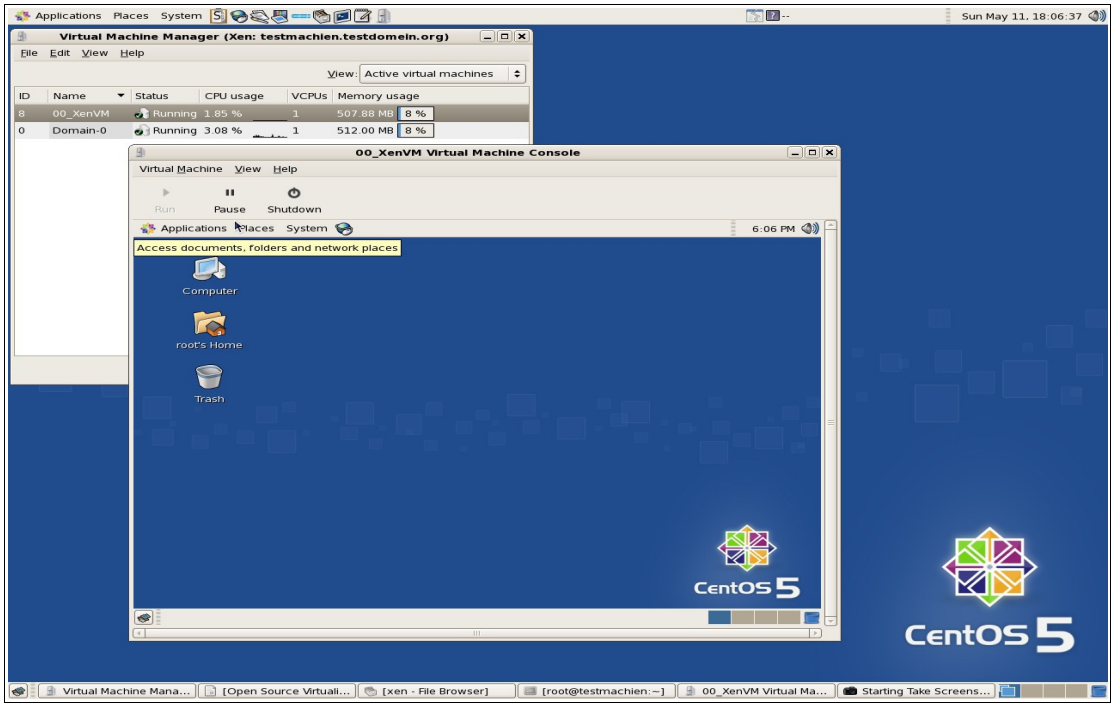
Daarmee is onze eerste virtuele machine klaar voor om geïnstalleerd te worden. We starten de installatie dan ook op, door in de virt-manager rechts te klikken op onze nieuwe virtuele machine en dan *Run* te kiezen. Als we nog eens rechtsklikken en dan *Open* kiezen, zien we ook daadwerkelijk in een venster dat onze machine draait. En

bezig is CentOS volautomatisch te installeren. Na afloop kunnen we de virtuele machine niet zomaar herstarten. Als we dat wel zouden doen, dan zou immers opnieuw het zelfde configuratiescript worden gebruikt en in dit script werd door virt-manager vastgelegd dat bij het opstarten gebruik gemaakt moet worden van het eerder genoemde *kickstart*-script. Zo zouden we dus in een zogenaamde *infinite loop* terecht komen. We moeten dus eerst het configuratiescript zodanig aanpassen dat het *kickstart*-script niet meer gebruikt kan worden. Dat doen we door de desbetreffende regel in het configuratie-script weg te commentariëren :

```
[root@testmachien ~]# vi /etc/xen/00_XenVM

name = "00_XenVM"

uuid = "9bdafa26-f44c-c05e-9016-e51eb4f71f90"

maxmem = 512

memory = 512

vcpus = 1

kernel = "/boot/xen.gz-2.6.18-53.1.14.el5.centos.plus"

ramdisk = "/boot/initrd-2.6.18-53.1.14.el5.centos.plusxen.img"

boot = "c"

pae = 1

acpi = 1

apic = 1

on_poweroff = "destroy"

on_reboot = "restart"

on_crash = "restart"

root = "/dev/xvda1"

#extra = "ks=ftp://192.168.1.11/ks/ks.cfg"

hostname = "krb5.testdomein.org"

sdl = 0

vnc = 1

vncunused = 1

disk = [ "phy:/dev/VG_VMS/00_XenVM,xvda1,w", “phy:/dev/VG_VMS/00_XenVM,xvdb,w" ]
```

```
pci = [ '05,00,0', '05,01,0' ]

vif = [ ''mac=00:16:3e:00:00:13,bridge=xenbr0'' ]

serial = "pty"

keymap 'fr-be'
```

En ja hoor, na de herstart lukt het en we kunnen probleemloos inloggen.



Afbeelding 24 : Een eerste virtuele machine

5.1.7. Virtuele machines en beveiliging onder Xen

We zouden op deze zelfde manier, na het telkens weer aanpassen van het kickstart-script, al onze benodigde virtuele servers kunnen installeren om ze daarna één voor één te configureren voor hun specifieke rol. Onze vzw zou dan heel snel aan de slag kunnen gaan. Dat zou echter ook een beetje gevaarlijk zijn. Daarom moeten we ons nu eerst verder gaan bezighouden met de beveiliging van ons systeem[103].

Wanneer we het management van de hele Xen-omgeving overlaten aan het geprivilegieerde domain-0, dan betekent dit ook dat dit domein toegankelijk is van buitenaf. Dat levert dan ook een behoorlijk veiligheidsrisico op. Als dom0 gecompromitteerd

103We hebben ons zoals eerder reeds gezegd rijkelijk laten inspireren door het artikel van Faye GIBBINS in Sys Admin Magazine. Inzake beveiliging helpt dat artikel ons al een hele stap verder. We gaan er nu nog het verbergen voor het geprivilegieerde domain-0 van de fysieke netwerkkaarten aan toevoegen.

raakt, dan is dat niet alleen vervelend voor dat domein. In één klap komt onze hele verzameling virtuele machines in het gedrang. We willen dan ook liefst dat niemand rechtstreeks toegang kan krijgen tot dom0 (met uitzondering van de administrator, uiteraard). Om dat te realiseren, moeten we eerst een aantal 'kunstgrepen' uitvoeren.

Standaard is Xen zo ingesteld dat de netwerkkaarten van de *hostcomputer* worden gebonden aan de Xen-*bridge*. Ze worden daarbij 'ombenoemd' naar fysieke netwerk-*interfaces* (bijvoorbeeld : eth0 wordt peth0). In de plaats daarvan komt er een virtuele netwerkinterface die de naam en de instellingen van de fysieke netwerk-*interface* overneemt (de virtuele netwerk-*interface* heet dan... eth0). Dit alles gebeurt in dom0. We kunnen echter de werkelijke, fysieke netwerk-*interfaces* verbergen voor dom0[104]. Dat kunnen we doen door de *boot*-optie "`pciback.hide='(0000:xx:xx.x)`"mee te geven in het configuratiebestand van de GrUB-*bootmanager*.[105] De parameter (0000:xx:xx.x)[106] staat daarbij voor het PCI-adres van de netwerkinterface. Dat PCI-adres moeten we natuurlijk wel eerst achterhalen. Daartoe gebruiken we de opdracht lspci :

```
[root@testmachien ~]# lspci

00:00.0 Host bridge: Intel Corporation 82P965/G965 Memory Controller Hub
(rev 02)

...

05:00.0 Ethernet controller: D-Link System Inc DGE-528T Gigabit Ethernet
Adapter (rev 10)

05:01.0 Ethernet controller: U.S. Robotics USR997902 10/100/1000 Mbps PCI
Network Card (rev 10)
```

We krijgen dan een lijst te zien (hierboven ingekort weergegeven) van al onze PCI-*devices*. Helemaal onderaan vinden we de PCI-adressen van onze netwerk-*interfaces*. We passen nu het GrUB-configuratiebestand (/boot/grub/grub.conf) aan. Het resultaat ziet er als volgt uit (de wijziging hebben we onderstreept) :

```
title Xen 3.1 / CentOS (2.6.18-53.1.14.el5.centos.plusxen)

root (hd0,0)

kernel /xen.gz-2.6.18-53.1.14.el5.centos.plus dom0_mem=512M \
pciback.hide='(05:00:0)(05:01:0)'
```

104Dit klinkt wellicht bijna te mooi om waar te zijn. Toch is het blijkbaar mogelijk. Wel is het zo dat het erg moeilijk is. Getuige Xen-gebruiker Phillip Bennett, die in een bijdrage aan de Xen-gebruikers-*mailinglist* schrijft *"I have been trying to get PCI passthrough working for **MONTHS** now..."* (zie Xen-users Digest, Vol 38, Issue 107). Hopelijk gaat het bij ons iets sneller.

105In diverse handleidingen op het Internet wordt soms ook gesproken over de boot-optie "physdev_dom0_hide". Dit is blijkbaar een overblijfsel uit een eerdere versie van Xen. Bij ons werkte het, na meerdere tests, in elk geval hoegenaamd niet.

106Deze parameter mogen we ook verkort schrijven (xx:xx.x). Concreet staat de volledige parameter voor (domain:bus:slot.function).

```
module /vmlinuz-2.6.18-53.1.14.el5.centos.plusxen ro \
root=/dev/VG_SYSTEM/LV_root rhgb quiet

module /initrd-2.6.18-53.1.14.el5.centos.plusxen.img
```

Als we de machine nu direct zouden herstarten, dan zou er géén netwerkfunctionaliteit meer mogen zijn. De netwerkkaarten zouden als het ware in het luchtledige moeten bestaan. Ze zouden wel functioneren, maar de data die ze zouden ontvangen, zouden ze nergens heen mogen kunnen sturen.

Helaas, op onze testmachine bleek dit niet te werken. De netwerk-*interfaces* werden nog steeds herkend door het besturingssysteem van dom0. Dit bleek echter geen tekortkoming van Xen te zijn, maar wel van het besturingssysteem voor dom0 (met andere woorden : het is de schuld van CentOS[107]). Na lang zoeken, vonden we uiteindelijk een mogelijke remedie.[108]

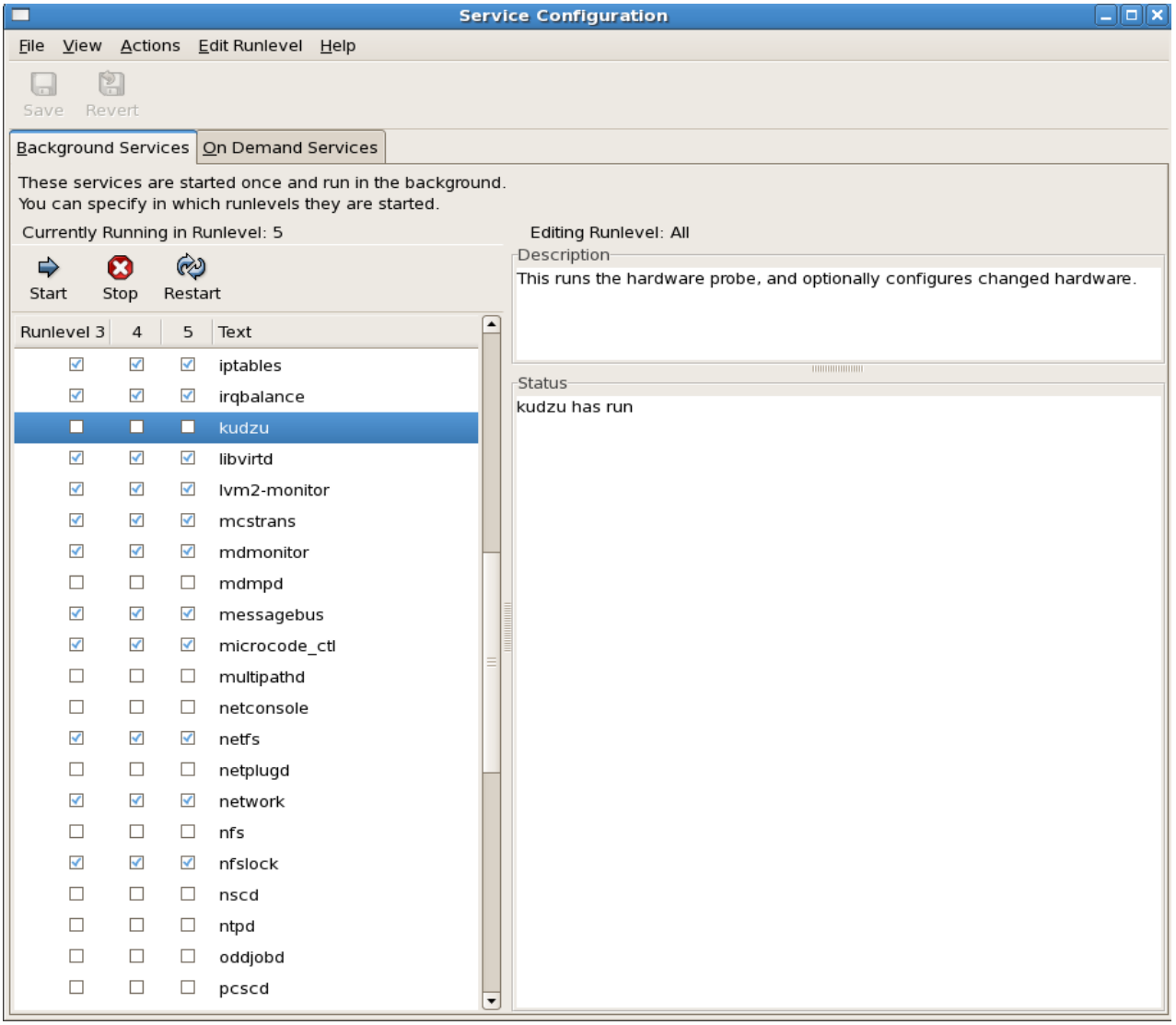
Die komt erop neer dat we enkele regels moeten toevoegen aan het bestand /etc/modprobe.conf. Na aanpassing ziet dat bestand er nu als volgt uit :

```
alias eth0 r8169

alias eth1 r8169

alias scsi_hostadapter ahci

alias scsi_hostadapter1 usb-storage

alias snd-card-0 snd-hda-intel

options snd-card-0 index=0

options snd-hda-intel index=0

remove snd-hda-intel { /usr/sbin/alsactl store 0 >/dev/null 2>&1 || : ; };
/sbin/modprobe -r --ignore-remove snd-hda-intel

#hide networkinterfaces

options pciback hide=(0000:05:00.0)(0000:05:01.0)

install r8169 /sbin/modprobe pciback ; /sbin/modprobe --first-time \

--ignore-install r8169
```

Het komt erop neer dat we de netwerk-*interfaces* niet alleen verbergen, maar dat we ook verhinderen dat er een *driver* voor wordt geïnstalleerd. Na een herstart van de machine, zien we nu inderdaad dat de installatie van de netwerk-*interfaces* eth0 en eth1 niet lukt. We kunnen nu onze virtuele machine die (fysieke) netwerk-*interfaces* laten 'grijpen', zodat enkel die virtuele machine toegang heeft tot het netwerk. Tenminste, dat denken we. In de praktijk blijkt echter dat CentOS (op dom0) na de op-

107En dus eigenlijk van Red Hat, de provider van de broncode van dit besturingssysteem.
108Zie : http://wiki.xensource.com/xenwiki/Assign_hardware_to_DomU_with_PCIBack_as_module

start van het systeem nog steeds de drivers toewijst aan de zogezegd 'verborgen' netwerk-*interfaces*. We vermoeden dat dit te maken heeft met de service 'kudzu' (verantwoordelijk voor het ontdekken van nieuwe hardware). Kudzu 'ziet' na de start van het systeem dat er nog twee netwerk-*interfaces* zijn, die in zijn ogen blijkbaar niet 'gezien' werden door het besturingssysteem. Kudzu installeert vervolgens de drivers. Dat willen we dus verhinderen. Wat we kunnen door de service te stoppen en ook de automatische herstart ervan te verhinderen. We gebruiken daarvoor de grafische services-manager, waar we voor alle runlevels de kudzu-service uitschakelen, vervolgens de instellingen opslaan en... de machine (alweer) herstarten.



Afbeelding 25 : De grafische services-manager

Helaas. Ook nu weer blijkt één en ander niet te lukken. We beginnen zo stilaan te vrezen dat dit inderdaad te mooi is om waar te zijn. Toch zoeken we nog wat verder op het Internet. Uiteindelijk vonden we daar ook deze oplossing[109].

109Zie : http://www.novell.com/coolsolutions/feature/17605.html In die bijdrage worden door

Eerst moeten we al onze vorige stappen weer ongedaan maken. Vervolgens moeten we ervoor zorgen dat de specifieke *kernel-module* die hiervoor nodig is, wordt geladen. Dat doen we met volgend commando :

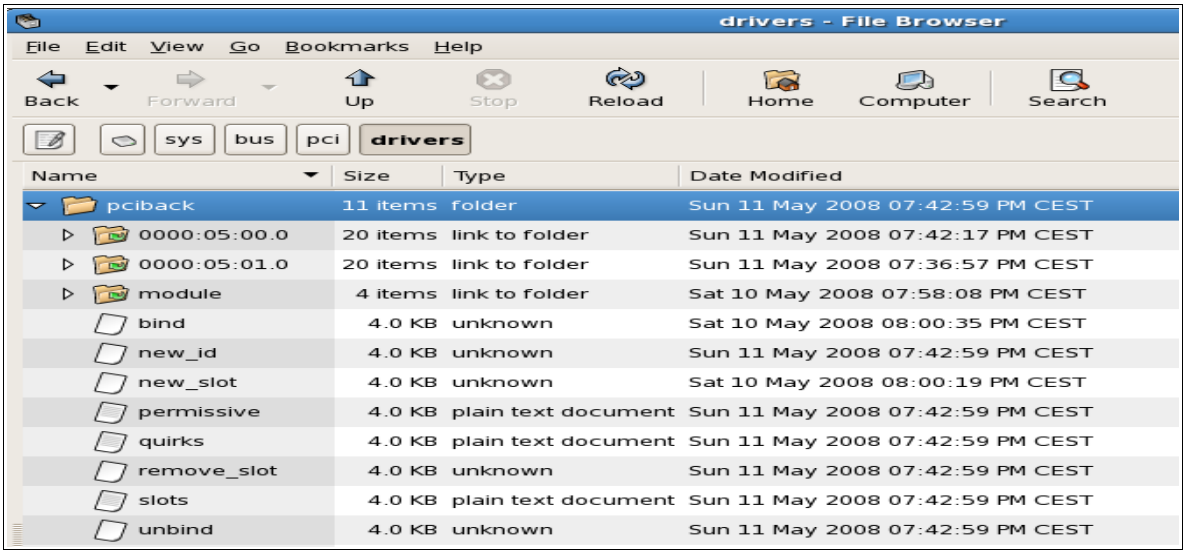
```
[root@testmachien ~]# modprobe pciback
```

We hebben eerder reeds uitgezocht wat het PCI-ID van onze netwerkkaarten precies is (0000:05:00.0 en 0000:05:01.0 om juist te zijn). Om verder te kunnen, moeten we nu te weten komen aan welke driver deze twee netwerkkaarten gebonden zijn. Daartoe bekijken we de directory /sys/bus/pci/drivers/. We zien daar een folder met de naam r8169. Als we in die folder kijken, dan zien we daar de symbolic links voor onze netwerkkaarten staan, welke verwijzen naar hun PCI-ID's. We moeten er nu voor zorgen dat die netwerkkaarten worden 'losgemaakt' van r8169 en in plaats daarvan worden toegewezen aan de *pciback-module*. Dat doen we door ze eerst los te maken (*unbind*), vervolgens nieuwe virtuele *slots* aan te maken (*new_slot*) en ze dan te binden aan dat nieuwe, virtuele *slot*, als volgt :

```
[root@testmachien ~]# echo -n 0000:05:00.0 > \

/sys/bus/pci/drivers/r8169/unbind

[root@testmachien ~]# echo -n 0000:05:01.0 > \

/sys/bus/pci/drivers/r8169/unbind

[root@testmachien ~]# echo -n 0000:05:00.0 > \ /sys/bus/pci/drivers/pci-
back/new_slot

[root@testmachien ~]# echo -n 0000:05:01.0 > \ /sys/bus/pci/drivers/pci-
back/new_slot

[root@testmachien ~]# echo -n 0000:05:00.0 > \ /sys/bus/pci/drivers/pci-
back/bind

[root@testmachien ~]# echo -n 0000:05:01.0 > \ /sys/bus/pci/drivers/pci-
back/bind
```

Als we nu gaan kijken naar de directory /sys/bus/pci/drivers/pciback/, dan zien we dat dit inderdaad gelukt is.

Glen DAVIS verschillende voorbeelden voor netwerkconfiguraties onder Xen gegeven. Zijn derde voorbeeld handelt over het configureren van een virtuele machine voor toegang tot een fysieke netwerkkaart.



Afbeelding 26 : de *pciback-module* met de aan haar gebonden netwerk-*drivers*

Dit lijkt allemaal heel goed te gaan, maar... als we onze machine ooit zouden herstarten, dan zouden al deze configuratieopties meteen komen te vervallen. We moeten dus op de één of andere manier zorgen dat bij het opnieuw opstarten, deze commando's allemaal opnieuw automatisch worden uitgevoerd. Daartoe maken we een opstart-script[110] aan.

```
[root@testmachien ~]# vi /init.d/pcihide

#!/bin/bash

# Bootup script to hide networkinterfaces from dom0

# Thanks too Glen Davis

# (see http://www.novell.com/coolsolutions/feature/17605.html)

# First, set pci-id's to variables pci1 and pci2

pci1=0000:05:00.0

pci2=0000:05:01.0

# Then, Set the driver name of NIC's to variables driver1 and driver2

driver1=r8169

driver2=r8169

# Enable pciback kernelmodule
```

110We pasten de commentaarregels in het script van de eerder genoemde Glen Davis een weinig aan om ze ietwat duidelijker te maken.

```
modprobe pciback

# Now hide the devices from dom0 and give that some time

echo -n $pci1 > /sys/bus/pci/drivers/$driver1/unbind

echo -n $pci2 > /sys/bus/pci/drivers/$driver2/unbind

sleep 1

# Give the devices to pciback, assign them to a new slot,

# then bind them to it

echo -n $pci1 > /sys/bus/pci/drivers/pciback/new_slot

echo -n $pci2 > /sys/bus/pci/drivers/pciback/new_slot

sleep 1

echo -n $pci1 > /sys/bus/pci/drivers/pciback/bind

echo -n $pci2 > /sys/bus/pci/drivers/pciback/bind

sleep 1
```

We maken dit script ook *executable* en daarna moeten we de netwerkkaarten ook nog 'vastmaken' aan onze virtuele machine. Daartoe dienen we het configuratiebestand van die virtuele machine aan te passen.

Concreet ziet het volledige configuratiebestand van onze eerste virtuele machine er dan als volgt uit (de relevante aanpassingen hebben we onderlijnd en met grijze achtergrond weergegeven) :

```
[root@testmachien ~]# vi /etc/xen/00_XenVM

name = "00_XenVM"

uuid = "9bdafa26-f44c-c05e-9016-e51eb4f71f90"

maxmem = 512

memory = 512

vcpus = 1

kernel = "/boot/xen.gz-2.6.18-53.1.14.el5.centos.plus"

ramdisk = "/boot/initrd-2.6.18-53.1.14.el5.centos.plusxen.img"

boot = "c"

pae = 1
```

```
acpi = 1

apic = 1

on_poweroff = "destroy"

on_reboot = "restart"

on_crash = "restart"

root = "/dev/xvda1"

#extra = "ks=ftp://192.168.1.11/ks/ks.cfg"

hostname = "krb5.testdomein.org"

sdl = 0

vnc = 1

vncunused = 1

disk = [ "phy:/dev/VG_VMS/00_XenVM,xvda,w", \

“phy:/dev/VG_VMS/00_XenVM,xvdb,w" ]

pci = [ '05,00,0', '05,01,0' ]

vif = [ 'mac=00:16:3e:00:00:13,bridge=xenbr0' ]

serial = "pty"

keymap 'fr-be'
```

Onze eerste virtuele machine is nu aangemaakt. Om ze ook werkelijk te starten kunnen we weer de virt-manager gebruiken. Vanaf nu kunnen we ervoor zorgen dat het deze eerste virtuele machine is die als bron gaat dienen voor alle andere aan te maken virtuele machines. Daartoe schakelen we nu op het geprivilegieerde domein-0 de Apache-server én de DHCP-server uit. Ook *dismounten* we het *logical volume* voor onze lokale repository. Daarna zorgen we ervoor dat dit *logical volume* gemount wordt in onze eerste virtuele machine. Dat doen we weer door het configuratiebestand ervan aan te passen. We voegen eerst het *repository logical volume* toe aan de disk-parameter :

```
disk = [ “phy:/dev/VG_VMS/00_XenVM,xvda,w”, \

“phy:/dev/VG_VMS/00_XenVM,xvdb,w”, \

“phy:/dev/VG_SYSTEM/LV_repo,xvdc,w”]
```

Daarna starten we de virtuele machine en in die machine passen we de file system tabel aan door er een regel aan toe te voegen :

```
[root@testmachien ~]# vi /etc/fstab

/dev/hda1    /var/repo   ext3    defaults,acl    0 0
```

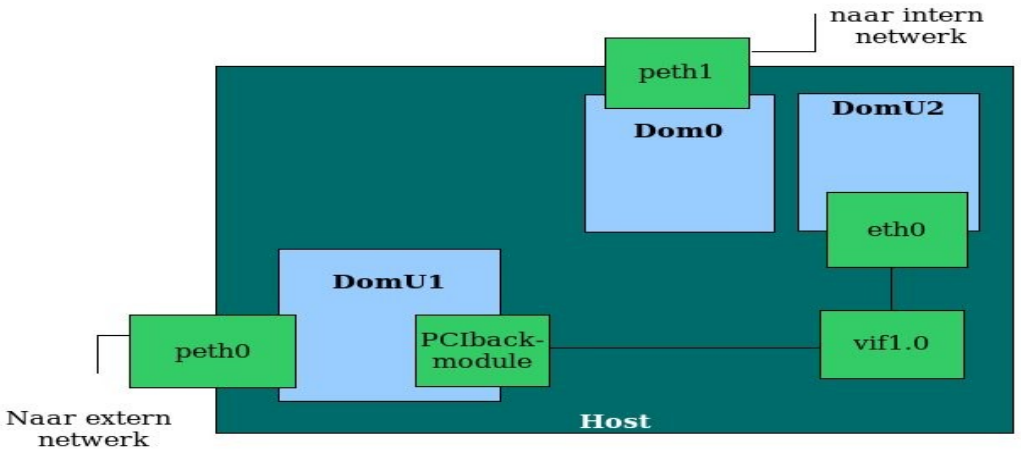
Hierdoor wordt dus onze lokale *repository* gemount binnen in de eerste virtuele machine, zodat deze nu die repository kan gebruiken om nieuwe virtuele machines aan te maken. In de virtuele machine kunnen we nu ook de LDAP-database vullen met de namen en paswoord-configuraties voor onze gebruikers, zodat deze machine vanaf nu kan optreden als authenticatie-server, dit wil zeggen als 'toegangspoort' tot ons virtuele netwerk. De eerste virtuele machine is immers, via het *kickstart*-script, geïnstalleerd met volledige SELinux-beveiliging en kan optreden als KDC- en LDAP-server voor onze virtuele cluster. Alleen de firewall is nog niet in orde. Daarom sluiten we de eerste virtuele machine weer af en geven we in domain-0 volgende commando's :

```
[root@testmachien ~]# ebtables -F INPUT

[root@testmachien ~]# ebtables -F FORWARD

[root@testmachien ~]# ebtables -F OUTPUT

[root@testmachien ~]# ebtables -A FORWARD --out-interface peth0 \

--protocol ipv4 --ip-protocol udp --ip-destination-port 67:68 -j DROP

[root@testmachien ~]# ebtables -A FORWARD --in-interface peth0 \

--protocol ipv4 service dhcpd restart
```

Deze commando's zorgen ervoor dat de DHCP-server in domain-0 geen DHCP-informatie meer uitzendt langs de fysieke netwerkkaart. Het staat echter wel DHCP-verkeer toe tussen de virtuele, bridged netwerkkaarten in de virtuele machines. Ook maakt dit het uitzenden van DHCP-verkeer vanuit de virtuele machines naar buiten onze virtuele cluster onmogelijk. Hierdoor zal DHCP-verkeer zich dus enkel afspelen binnen in de virtuele cluster. De buitenwereld zal er dus nooit last van hebben. Tenslotte wordt, met het voorlaatste commando de DHCP-server onzichtbaar voor het bredere netwerk buiten.

Al deze configuratiestappen kunnen ook in één keer worden uitgevoerd, met behulp van een script. In Appendix F hebben we zo'n script opgenomen. Vanaf nu kunnen we nieuwe virtuele machines maken, zonder dat domain-0 daar nog aan te pas hoeft te komen. De administratie van de gehele virtuele cluster zal nu gebeuren vanuit de eerste virtuele machine. Domain-0 is ook niet meer toegankelijk van buitenaf en is dus optimaal beveiligd. Omdat de eerste virtuele machine ook beveiligd is, is de kans klein dat daar nog iets mee mis zou kunnen gaan. Mocht dat onverhoopt toch gebeuren, dan kan de hele boel relatief snel heropgebouwd worden.

Schematisch kan onze configuratie nu zo worden voorgesteld :



Afbeelding 27 : Eén verborgen netwerkkaart toegewezen aan DomU1

2.1. 5.1.8. Xen vanuit broncode compileren en installeren

We gaan nu de nieuwste versie van Xen (3.2) proberen te installeren. Daartoe moeten we eerst de reeds aanwezige hulpmiddelen voor Xen 3.1 verwijderen, aangezien deze niet volledig compatibel zijn met de nieuwe versie. Om geen onnodige problemen te veroorzaken, verwijderen we in één keer alle bestanden die we eerder installeerden voor Xen 3.1. Daartoe herstarten we de machine eerst, waarbij we deze keer kiezen voor de 'gewone' Linux-kernel (dus zonder Xen. Vervolgens geven we onderstaand commando :

```
[root@testmachien ~]# yum remove kernel-xen xen virt-manager
```

We gaan nu de broncode downloaden om ze daarna te compileren. Voor we dat kunnen doen, moeten we eerst het *revision control system* Mercurial installeren. Dat kan niet zomaar via de grafische interface van 'Add/Remove Programs' of via een eenvoudig yum-commando. CentOS biedt blijkbaar Mercurial niet aan in zijn *repositories*. Daarom gaan we tijdelijk een *repository* toevoegen, met name die van Dag Wiëers. Deze in Linux-kringen bekende Belg heeft een eigen up-to-date RPM-*repository* opgezet dat veel uitgebreider is dan dat van CentOS. Op zijn website (http://dag.wieers.-com/rpm/) vind je het commando om zijn *repository* toe te voegen. Voor onze versie van CentOS is dat :

```
rpm -Uhv http://apt.sw.be/packages/rpmforge-release/ \

rpmforge-release-0.3.6-1.el5.rf.i386.rpm
```

Nu kunnen we eenvoudigweg de nieuwste versie van het Mercurial revision control system installeren :

```
[root@testmachien ~]# yum install mercurial
```

We maken nu eerst een gepaste folder aan en plaatsen onszelf daarin :

```
[root@testmachien ~]# mkdir ~/xen-source

[root@testmachien ~]# cd ~/xen-source
```

Nu gaan we de broncode van Xen 3.2 downloaden en klaarmaken voor compilatie. Dat doen we met behulp van de mercurial-client hg :

```
[root@testmachien ~]# hg clone http://xenbits.xensource.com/ \

xen-3.2-testing.hg
```

Na verloop van tijd bevindt er zich nu in de folder ~/xen-source een nieuwe folder /xen-3.2-testing.hg. Voor we echter de zich hier bevindende broncode kunnen compileren, moeten we eerst nog wat andere programma's installeren, met name gcc (de GNU Compiler Collection), verschillende libraries en header-files en dergelijke meer. We geven volgend yum-commando :

```
[root@testmachien ~]# yum install gcc glibc-devel xen-devel libgomp \ gli-
bx-headers ncurses-devel openssl-devel zlib-devel xorg-X11-proto-devel py-
thon-devel tetex-latex transfig graphviz pciutils-devel[111]
```

Na afloop heeft yum alles geïnstalleerd samen met een hoop *dependencies*. Nu kunnen we de broncode voorbereiden op compilatie door ze te configureren. Dat doen we met het commando make :

```
[root@testmachien ~]# make linux-2.6-xen0-config
```

Met dit commando gaan we in feite een nieuwe linux-kernel configureren voor gebruik met Xen. Er opent zich dan ook een scherm met daarin het hulpprogramma voor de *Linux Kernel Configuration*. Als we willen kunnen we de kernel nu helemaal op maat maken. We doen dat echter niet en accepteren dus alle *default* instellingen. We verlaten het dialoogvenster en slaan de wijzigingen op als ons daarom gevraagd wordt. Daarna gaan we over tot het eigenlijke compileren :

```
[root@testmachien ~]# make linux-2.6-xen0-build
```

De nieuwste versie van Xen is nu gecompileerd en klaar voor gebruik. We moeten ze alleen nog installeren :

```
[root@testmachien ~]# make linux-2.6-xen0-install
```

Tenslotte moeten we de GrUB-*bootmanager* nog aanpassen, zodat we de nieuwe Xen ook daadwerkelijk kunnen opstarten. Hiertoe passen we de bestaande verwijzing naar Xen (versie 3.1) aan de nieuwe Xen-versie aan. Vervolgens herstarten we het systeem, kiezen Xen en klaar is kees.

111 In diverse handleidingen wordt niet gesproken over de bestanden transfig, graphviz en pciutils-devel. Zonder deze bestanden compileert xen-3.2 ook wel, maar dan wel zonder documentatiebestanden en vooral zonder ondersteuning van geavanceerde functies, zoals het kunnen verbergen van zekere pci-hardware.

5.1.9. Virtuele machines onder Xen 3.2

Het aanmaken van virtuele machines met behulp van de virt-manager applicatie gaat nu niet meer, aangezien we deze (wegens incompatibiliteit) hebben verwijderd[112]. We beschikken echter nog steeds over de reeds eerder aangemaakte virtuele machine. Die kunnen we dan ook handmatig terug opstarten :

```
[root@testmachien ~]# xm create -c 00_XenVM
```

De '-c' in dit commando staat voor 'console' en zorgt ervoor dat we ook daadwerkelijk kunnen zien wat de virtuele machine doet.

We gaan nu ter illustratie opnieuw een virtuele machine aanmaken. Dat zouden we nu manueel kunnen doen, dit wil zeggen via de command line methode. Daartoe moeten we dan eerst een geschikt configuratiebestand aanmaken. In de directory /etc/xen/ bevinden zich ook enkele voorbeeld-bestanden. Die kunnen we gebruiken een aanpassen aan onze noden. Er zijn voor het aanmaken van virtuele machines echter ook andere *tools*[113], zoals bijvoorbeeld xenguest-install.py. Dit is een Python-script dat geschreven werd voor gebruik met de Fedora Linux-distributie. Aangezien Fedora een testplatform is voor Red Hat Enterprise Linux (en aangezien onze CentOS niks anders is dan een zo goed als identieke versie van Red Hat Enterprise Linux), kunnen we dit script probleemloos gebruiken. Eerst downloaden we het script[114] naar /usr/sbin/, daarna maken we het met behulp van het commando **chmod 755 /usr/bin/xenguest-install.py** uitvoerbaar.

Het script helpt ons interactief om onze virtuele machine te installeren, waarbij er gebruik gemaakt wordt van onze eerder opgezette lokale *repository*. Het script stelt daartoe vragen naar de naam van onze virtuele machine, de hoeveelheid toegewezen RAM en dergelijke meer.[115] Het script is voornamelijk interessant omdat het een heel makkelijke methode is om virtuele machines aan te maken met een *logical volume* als opslagmedium[116]. We voeren het script nu uit in de interactieve modus (d.w.z. zonder *command line* opties) :

```
[root@testmachien ~]# xenguest-install.py

What is the name of your virtual machine? 01_XenVM

 ERROR: Installs currently require 256 megs of RAM.

How much RAM should be allocated (in megabyte)? 512

 What would you like to use as the disk (path)? /dev/VG_VMS/01_XenVM

 What is the install location? ftp://192.168.1.10/pub/
```

112 Het is ook mogelijk, indien gewenst, om de broncode van virt-manager zelf te downloaden en te hercompileren voor onze nieuwe Xen-omgeving. Virt-manager is immers een apart project met vrij beschikbare broncode.

113 Feitelijk zijn het niet zozeer *tools*, dan wel in Python geschreven scripts.

114 Zie : http://people.redhat.com/~katzj/xenguest-install.py

115 De interactieve werkwijze kan ook omzeild worden door alle nodige opties in één keer mee te geven op de command line.

116 Dat logical volume moeten we wel eerst zelf aanmaken !

Dan maken we een *directory* aan waarbinnen we ons *logical volume* kunnen *mounten*. Daarna *mounten* we het volume ook.

```
[root@testmachien ~]# mkdir -p /vms/01_XenVM

[root@testmachien ~]# mount /dev/VG_VMS/01_XenVM /vms/01_XenVM
```

Nu kunnen we in het *logical volume* 01_XenVM deze virtuele machine gaan installeren. We gebruiken daarvoor onze eerder aangemaakte lokale repository van CentOS.

```
[root@testmachien ~]# yum –installroot=/vms/01_XenVM/ -y groupinstall base
```

Na verloop van tijd is de virtuele machine geïnstalleerd. We moeten nu nog enkele maatregelen nemen om er goed mee te kunnen werken. Zo willen we bijvoorbeeld een console kunnen openen om te kunnen inloggen. We geven volgende achtereenvolgende commando's (in een terminal op de hostmachine) :

```
[root@testmachien ~]# MAKEDEV -d /dev -x console

[root@testmachien ~]# MAKEDEV -d /dev -x null

[root@testmachien ~]# MAKEDEV -d /dev -x zero
```

We willen natuurlijk ook dat deze virtuele machine over het netwerk kan aangesproken worden. Daartoe is uiteraard netwerkconnectiviteit noodzakelijk. We moeten dus zorgen voor de nodige configuratiebestanden. Ook een werkend shadow- en password-bestand zijn nodig. We moeten ook de *kernel-modules* kopiëren van domain-0 naar het domein 01_XenVM. En natuurlijk moeten we ook de juiste Xen-*kernel* installeren in de virtuele machine, zoniet zouden de *kernel-modules* er niet mee kunnen werken. Eerst zorgen we dat we in de virtuele machine zijn :

```
[root@testmachien ~]# cd /vms/01_XenVM
```

Vervolgens zorgen we voor een nieuw root-paswoord :

```
[root@testmachien 01_XenVM]# passwd
```

Daarna zorgen we ervoor dat de virtuele machine de netwerkkaarten kan aanspreken. Hiertoe kopiëren we eenvoudigweg de netwerkconfiguratiebestanden van domain-0 naar onze virtuele machine (zie ook verder) :

```
[root@testmachien 01_XenVM]# cd

[root@testmachien ~]# cp /etc/sysconfig/network-scripts/ifcfg-eth0 \

/vms/01_XenVM/etc/sysconfig/network-scripts/ifcfg-eth0

[root@testmachien ~]# cp /etc/sysconfig/network-scripts/ifcfg-eth1 \

/vms/01_XenVM/etc/sysconfig/network-scripts/ifcfg-eth1
```

En we zorgen er ook voor dat er sowieso netwerkfunctionaliteit is op de virtuele machine. Daartoe kopiëren we eerst het bestand /etc/sysconfig/network naar de virtuele

machine, waarna we vervolgens de hostname veranderen in 01_XenVM.testdomein.org :

```
[root@testmachien ~]# cp /etc/sysconfig/network \

/vms/01_XenVM/etc/sysconfig/network

[root@testmachien ~]# vi /vms/01_XenVM/etc/sysconfig/network

NETWORKING=yes

NETWORKING_IPV6=no

HOSTNAME=01_XenVM.testdomein.org

GATEWAY=192.168.1.1

~

:q
```

Vervolgens installeren we de Xen-kernel met de nodige kernel-modules naar de virtuele machine :

```
[root@testmachien ~]# yum –installroot=/vms/01_XenVM/ -y install kernel-xen
```

Nu moeten we (tenslotte) nog een configuratiebestand aanmaken voor onze virtuele machine. Dat doen we als volgt :

```
[root@testmachien ~]# vi /etc/xen/01_XenVM

kernel = "/boot/vmlinuz-xen"

#initrd = "boot/initrd-2.6.18-53.1.14.el5.centos.plusxen"

name = "01_XenVM"

maxmem = 512

memory = 512

vcpus = 1

boot = "c"

pae = 1

acpi = 1

apic = 1

on_poweroff = "destroy"
```

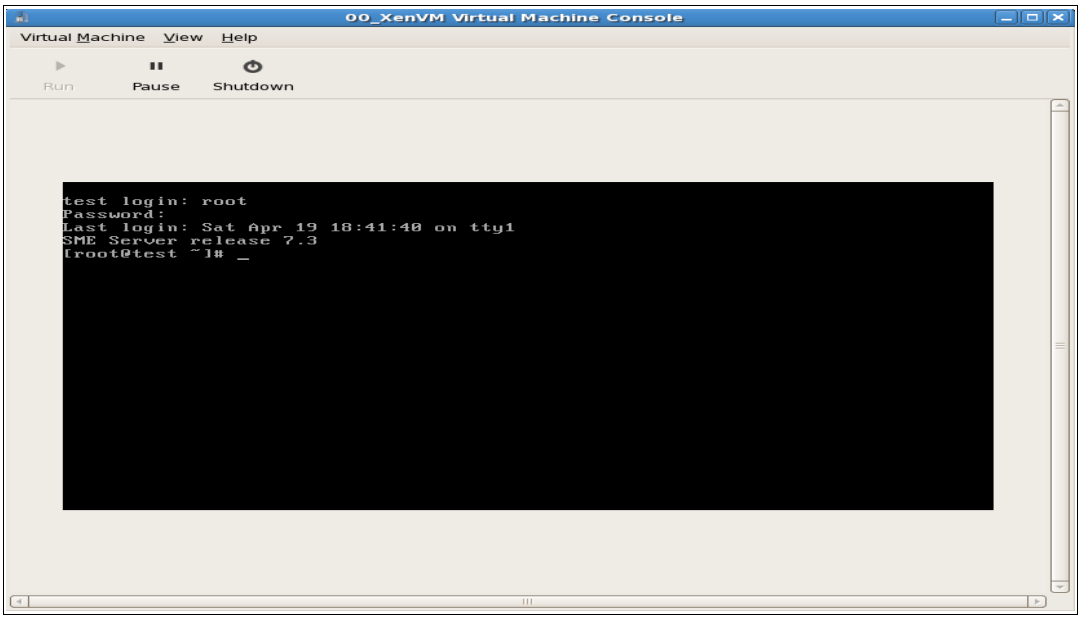
```
on_reboot = "restart"

on_crash = "restart"

sdl = 0

vnc = 1

vncunused = 1

disk = [ 'phy:/dev/VG_VMS/01_XenVM,sda1,w' ]

root = "/dev/sda1 rw"

vif = [ 'mac=00:16:3e:06:3b:80, bridge=xenbr0', ]

serial = "pty"

keymap = “fr-be”

:wq!
```

De meeste opties in dit bestand spreken voor zich. Toch een paar opmerkingen. De zogenaamde *initial ram-disk* (die ervoor dient om bepaalde drivers te laden voor het besturingssysteem opgestart is), hebben we inactief gemaakt door er een #-teken voor te zetten, omdat het niet zeker is dat we hem nodig hebben. Mocht dat toch het geval zijn, dan kunnen we hem alsnog terug actief maken.

We kunnen onze virtuele machine nu opstarten. Eerst moeten we daartoe het *logical volume* 01_XenVM unmounten. Daarna kunnen we onze virtuele machine opstarten. Dat doen we met volgende commando's (waarbij in het laatste commando de optie '-c' ervoor zorgt dat we via een console kunnen zien wat er gebeurt) :

```
[root@testmachien ~]# umount /vms/01_XenVM

[root@testmachien ~]# xm create /etc/xen/01_XenVM -c
```

En ja hoor, onze virtuele machine start :



Afbeelding 28 : Een virtuele machine

We zouden nu kunnen verdergaan met het configureren van deze virtuele machine en vervolgens met het installeren van alle andere virtuele machines en het configureren ervan. Dat doen we echter niet, want deze (relatief) snel aangemaakte virtuele machine diende immers enkel tot illustratie. Veel beter is het echter om voor elke virtuele machine dien we nodig hebben een gespecialiseerd script te maken, zodat ze in één keer kunnen geïnstalleerd worden. We kunnen dit zelfs verder doordrijven en de verschillende scripts samenvoegen tot één groot script voor de installatie en het opstarten alle virtuele machines samen. Er zijn echter ook nog andere opties. Die zullen we nu bekijken.

5.1.10. Programma's voor het beheer van een Xen-cluster

We hebben al kunnen vaststellen dat Xen niet bepaald eenvoudig is om te installeren. Toch zijn er ook hulpmiddelen om dat én het beheer van de virtuele machines in een Xen-cluster te vereenvoudigen. We geven hiervan een kort overzicht.

Eerst en vooral zijn er natuurlijk de *tools* van Xen zelf. We zagen hierboven reeds de grafische virt-manager. Voor de *command line* kennen we de Xen manager xm. We spraken ook reeds over de Python-*tools* die gemaakt werden voor Fedora Linux. Los daarvan werden er ook zogenaamde *third party tools* ontwikkeld.

5.1.10.1. xen-tools

Dit zijn een aantal scripts die geschreven werden door een enthousiaste aanhanger van de Debian-tak der Linux-distributies[117]. Om Debian-distributies te kunnen installeren onder Xen dient gebruik gemaakt te worden van het programma *debootstrap*[118].

117Zie : http://www.xen-tools.org/software/xen-tools/
118Zie : http://packages.debian.org/stable/admin/debootstrap

Dit programma is niet beschikbaar voor de Red Hat Linux-derivaten. Als we het willen gebruiken moeten we dus eerst het bestand downloaden en het daarna omzetten van een .deb-bestand naar een .rpm-bestand. Dat kan met behulp van het programma *alien*[119]. Eens we dat allemaal gedaan hebben, kunnen we de xen-tools downloaden. Feitelijk zijn deze tools niks anders dan een verzameling perl-scripts. Met die scripts kan relatief makkelijk een virtuele machine aangemaakt worden. Omdat de tools zo populair zijn, werd de functionaliteit ervan ondertussen uitgebreid tot de Red Hat Linux-derivaten. In plaats van debootstrap wordt dan gebruik gemaakt van het vergelijkbare *rpmstrap*[120].

Het installeren van nieuwe virtuele machines is met behulp van de xen-tools zeer eenvoudig. Het volgende commando installeert bijvoorbeeld een debian virtuele machine in een logical volume :

```
xen-create-image --hostname=test.my.flat \
  --ip=192.168.10.10 \
  --lvm=vol1 \
  --dist=sarge
```

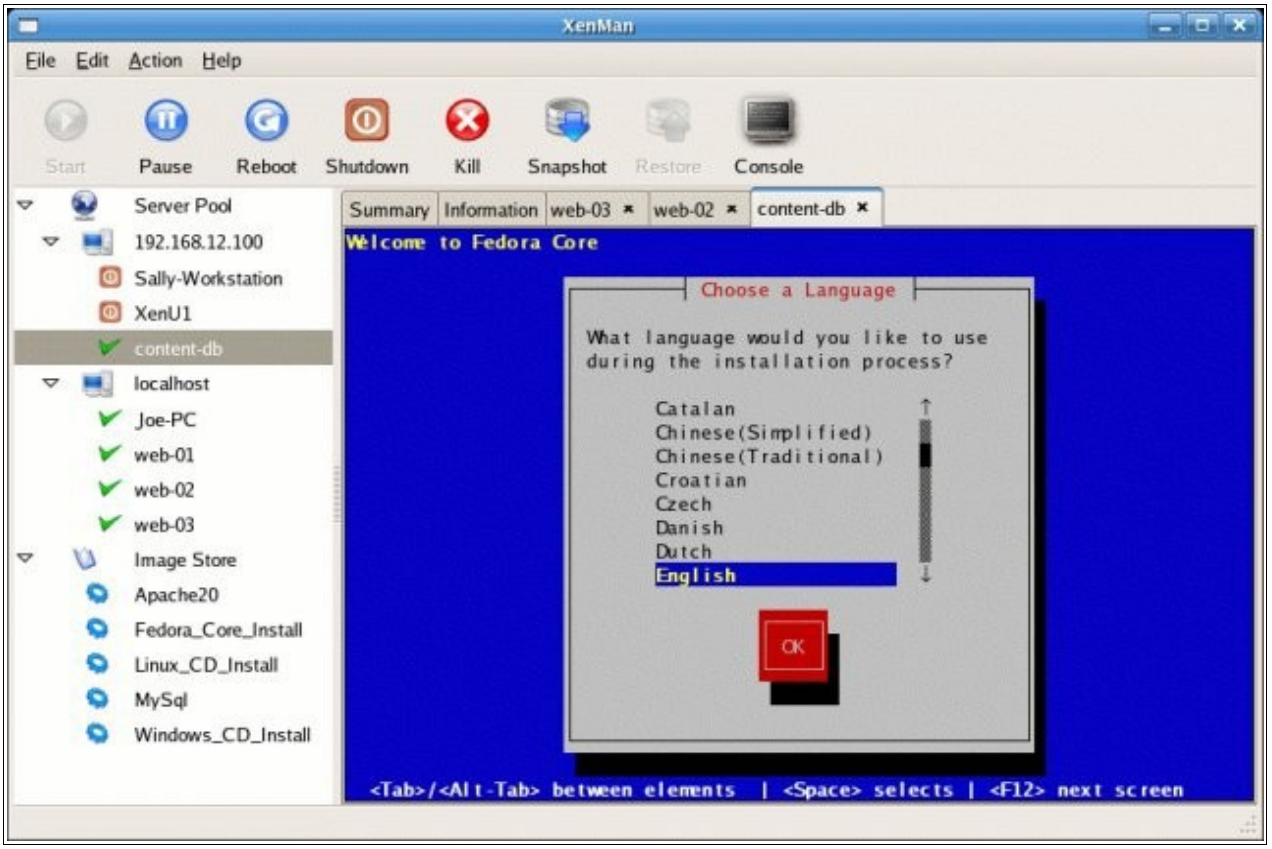
Het *rinse*-script[121] doet hetzelfde voor Red Hat Linux-derivaten. Met het *xen-shell*-script[122] kunnen reeds geïnstalleerde virtuele machines onder Xen worden beheerd. Nog niet helemaal op punt staat ook het *argos*-script[123] dat ertoe strekt uit te groeien tot een dashboard voor gedistribueerde Xen-hostsystemen.

De xen-tools vereisen wel stevig wat kennis en inzicht in en van de Xen-configuratiebestanden. Voor we ze kunnen gebruiken, moeten we de scripts eerst handmatig aanpassen. De documentatie is echter vrij volledig en uiteindelijk lukt één en ander dan ook wel[124].

5.1.10.2. XenMan en Convirt

Scripts zijn allemaal goed en wel, maar zoals gezegd vereisen ze toch wel wat voorafgaande kennis. Grafische interfaces dragen de belofte in zich dat dat niet (of toch niet zo veel) meer nodig is. Naast de virt-manager was één van de eerste grafische tools Xen Man[125]. Een voorbeeld zien we hier :

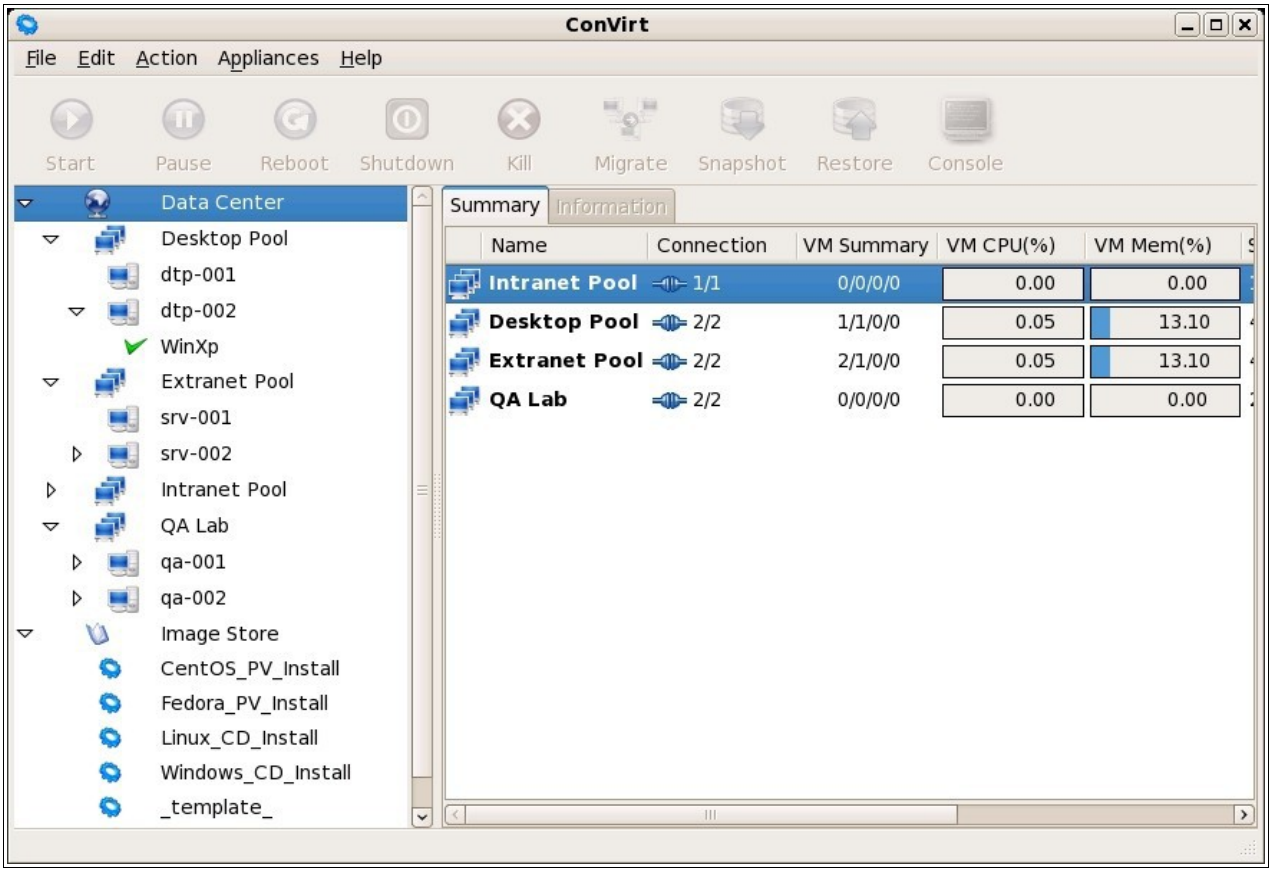
119Zie : http://kitenet.net/~joey/code/alien/ Voor een meer gedetailleerde beschrijving van het gebruik van *alien* en *debootstrap*, zie CHAGANTI P. ibid., pagina 33 en volgende.
120Zie : http://rpmstrap.pimpscript.net/
121Zie : http://www.xen-tools.org/software/rinse/faq.html
122Zie : http://www.xen-tools.org/software/xen-shell/
123Zie : http://www.xen-tools.org/software/argos/
124Een andere *tool*, die eveneens bestaat uit een verzameling scripts, is MLN (Manage Large Networks), ontwikkeld in het kader van een master-thesis aan de Universiteit van Oslo (http://mln.sourceforge.net/).
125Zie : http://xenman.sourceforge.net/doc.html en ook CHAGANTI P., ibid., pagina 64 en volgende



Afbeelding 29 : XenMan in actie[126]

Op basis van XenMan werd ondertussen een nieuw, meer geavanceerd programma gemaakt, met name ConVirt[127]. Het meer geavanceerde schuilt hem vooral in de veel meer uitgebreide aandacht voor beveiliging (via certificaten) van de virtuele machines. Dit is niet onbelangrijk, omdat hypervisors en virtuele machines ook kunnen dienen als verborgen *root kit*[128]. Met de certificatie-aanpak kan een virtuele machine op een Xen-host echter niet opstarten zonder het vereiste certificaat. Het aanmaken en uitgeven van dergelijke certificaten wordt met ConVirt bijna kinderspel. Hieronder een afbeelding van de gebruikersinterface van ConVirt :

126Schermafdruk afkomstig van : https://sourceforge.net/project/screenshots.php?group_id=168929&ssid=50261
127Zie : http://xenman.sourceforge.net/index.html
128Voor een bespreking hiervan, zie : http://theinvisiblethings.blogspot.com/2006/06/introducing-blue-pill.html



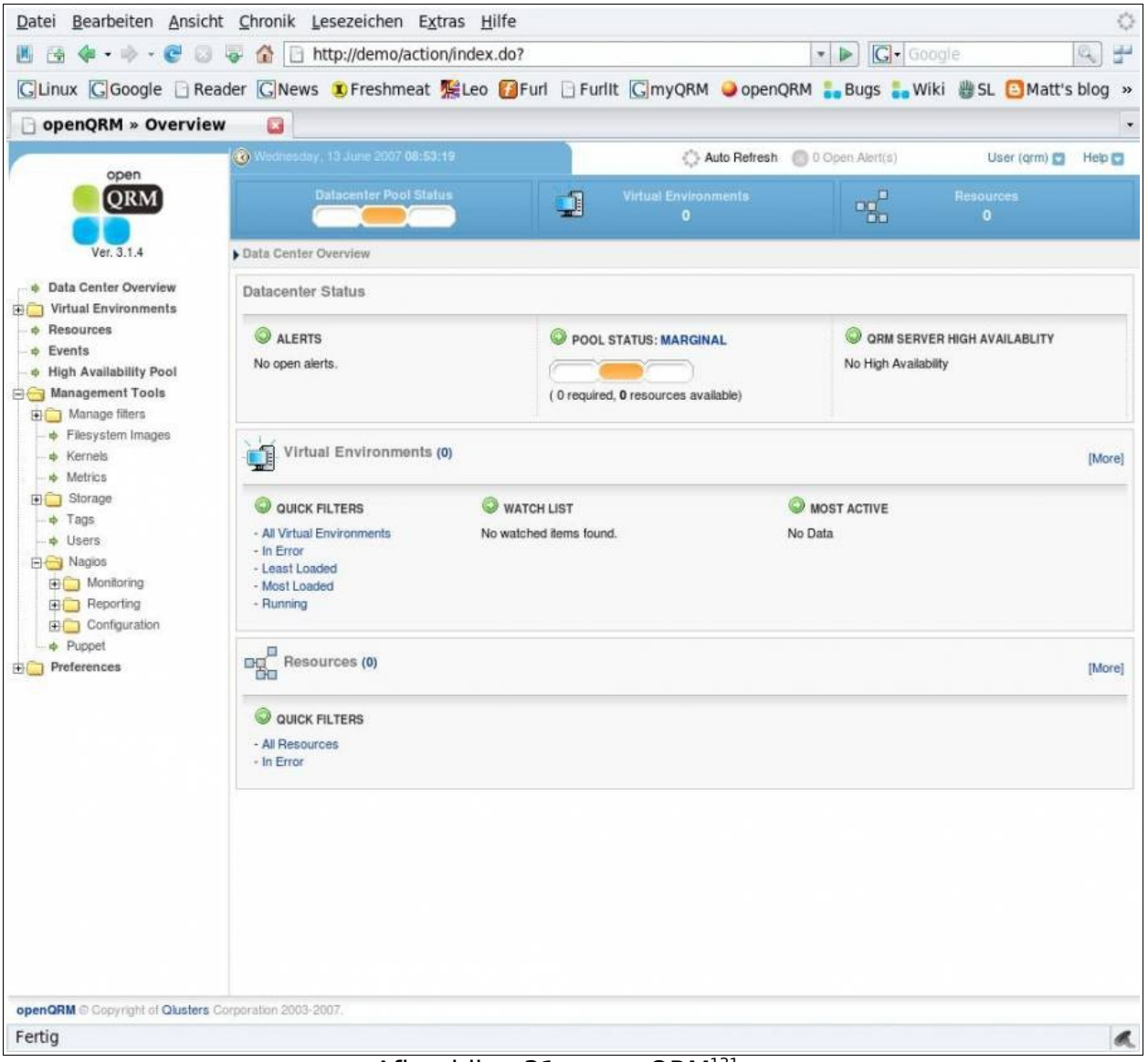
Afbeelding 30 : Het dashboard van ConVirt

Op te merken valt ook dat ConVirt niet alleen enkelvoudige hosts kan beheren, maar ook hele pools van Xen-hosts. Daarnaast wordt het met ConVirt ook mogelijk om te werken met vooraf geconfigureerde installatie-templates. Hiervoor maakt ConVirt gebruik van de mogelijkheden van *rPath*[129], waarmee zogenaamde *software appliances* kunnen worden aangemaakt (in dit geval dus : installatie-*templates* van verschillende Linux-servers).

5.1.10.3. openQRM

Een ander voorbeeld van een grafische interface is openQRM[130]. Tot voor enige tijd was dit een propriëtair programma. Sinds het echter *open source* geworden is, mag het zich verheugen in een vrij snelle ontwikkeling. Het interessante aan dit programma is dat het eigenlijk niet speciaal voor het beheer van virtuele machines gemaakt werd. Het is eigenlijk een 'gewone' *server management tool*, die echter via *plugins* uitgebreid werd met alle mogelijkheden op het vlak van het beheer van virtuele machines. Het kan daarbij niet alleen voor Xen-beheer gebruikt worden, maar ook voor andere virtualiseringsplatformen. Een voorbeeld van deze interface :

129Zie : http://www.rpath.com/corp/
130Zie : http://www.openqrm.org/



Afbeelding 31 : openQRM[131]

5.1.10.4. Enomalism

Tenslotte is er ook nog het snel evoluerende Enomalism[132]. Dit is een programma dat eigenlijk ontwikkeld werd voor het zogenaamde *Cloud Computing Platform*[133], dat onder andere gebruikt wordt door IBM, Amazon en Google. Daarnaast kan Enomalism echter ook overweg met virtuele machines en hun beheer. Doel is om dat te kunnen voor elk mogelijk denkbaar virtualiseringsplatform. Momenteel worden reeds Xen en

131Afbeelding afkomstig van : http://wiki.linuxfellaz.net/doku.php?id=openqrm:webfarm
132Zie : http://www.enomalism.com/
133Zie http://en.wikipedia.org/wiki/Cloud_computing en
http://clusters.wallonie.be/tic/en/news/2008-02-18-reservoir-cetic.html

KVM ondersteunt, terwijl OpenVZ binnenkort zou volgen. Enomalism is momenteel wel nog in het beta-stadium.

Afbeelding 32 : Enomalism[134]

5.1.11. Xen – Besluit

Zonder volledig te kunnen zijn, hebben we heel lang stilgestaan bij Xen. Daarbij hebben we kunnen vaststellen dat het niet de meest simpele virtualisatiemethode is. Het vereist toch wel een grote kennis van Linux. Ook is het een zogenaamd *fast moving project*, waarmee bedoeld wordt dat er nog wel eens wat wil veranderen aan commando's en parameters. De documentatie (vooral voor wat betreft de meer geavanceerde eigenschappen van Xen) loopt ook weleens achter op de ontwikkeling van Xen, wat overigens (helaas) typisch is voor *open source* projecten. Het is wel één van de meest robuuste methodes. Ook qua performantie laat Xen nauwelijks iets te wensen over.

134Afbeelding afkomstig van : http://linux.softpedia.com/progScreenshots/Enomalism-Virtualized-Management-Console-Screenshot-12352.html

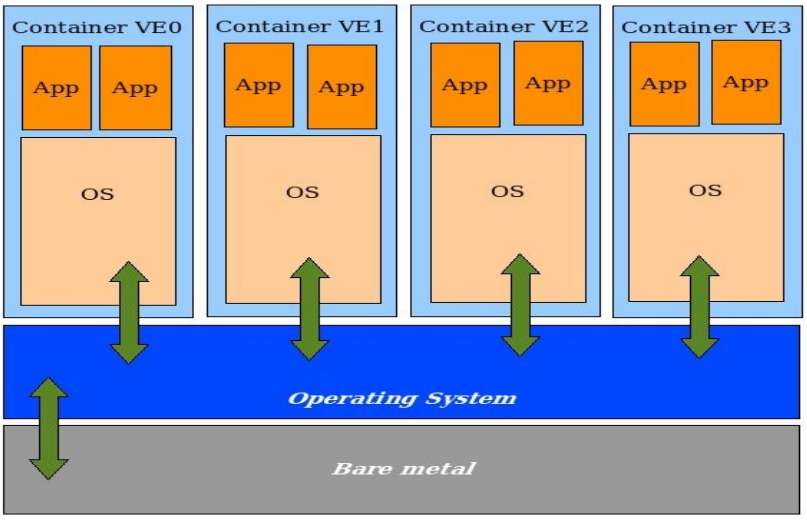
5.2. Oplossing 2 : OpenVZ

5.2.1. Inleiding

OpenVZ[135] is de open source ontwikkeltak van het propriëtaire Parallels Virtuozzo Containers[136]. De firma Parallels Inc. is voornamelijk bekend vanwege de programma's Parallels Desktop en Parallels Workstation (virtualiseringsoplossingen voor de individuele desktop, die heel wat goedkoper uitvallen dan het concurrerende VMWare Workstation).

Hoewel de installatiemethodes van OpenVZ in veel opzichten sterk doen denken aan die van Xen, houdt de vergelijking daar ook meteen op. Xen virtualiseert de hardware van de computer immers als hypervisor, iets waar OpenVZ volledig van afziet.

Wat OpenVZ wel doet is toestaan dat meerdere instanties van het Linux-besturingssysteem tegelijkertijd actief zijn in user space (ring3). Bovendien zijn al die instanties – die containers, *virtual private servers* of *virtual environments* worden genoemd – volstrekt geïsoleerd van elkaar. Dat wil zeggen dat elke container op zich alleen maar het aan hem toegewezen deel van het RAM-geheugen kan gebruiken. De rest van het RAM-geheugen is voor die container dan ook niet zichtbaar. Het beheren van de verschillende containers, gebeurt door het onderliggende besturingssysteem, dat in feite niks anders is dan een aangepaste versie van de Linux-kernel : de *openvz-kernel*. Het is die *openvz-kernel* die, samen met de *openvz-tools*, instaat voor de *scheduling* en het *paging mechanisme* van de verschillende containerprocessen. OpenVZ is dan ook een voorbeeld van *operating system virtualisering*[137]. Hieronder een vereenvoudigd schema van hoe dat er uit ziet :



Afbeelding 33 : *Operating system virtualization*

135Zie : http://wiki.openvz.org/Main_Page en http://en.wikipedia.org/wiki/OpenVZ
136Zie : http://www.parallels.com/en/products/virtuozzo/
137Een ander voorbeeld hiervan is Linux V-Server. Zie daarover : http://linux-vserver.org/Welcome_to_Linux-VServer.org

Op die manier wordt het dus mogelijk om verschillende servers op één enkele fysiek machine gelijktijdig naast elkaar te draaien. Zo zijn die servers toch weer geïsoleerd van elkaar. Bovendien is het zo dat de containers weliswaar zelf een volledig geïnstalleerd Linux-systeem nodig hebben, maar dat ze daar alleen gebruik van maken als het echt nodig is. Met andere woorden, voor tal van processen zullen de containers de taken laten beheren door de onderliggende *openvz-kernel*. Hierdoor wordt een hoge mate van rationalisering bereikt, wat de performantie van het systeem zeer ten goede zou moeten komen[138].

Een nadeel van dit concept is wel dat er in de containers alleen gewerkt kan worden met Linux, al is het wel zo dat het mogelijk is te werken met meerdere, verschillende Linux-distributies[139].

In tegenstelling tot Xen zijn er rond OpenVZ geen gedrukte boeken voorhanden[140]. Ook bestaat er op het Internet geen enorm verspreid aanbod aan handleidingen. Dit lijkt misschien een tekortkoming, maar daar staat tegenover dat er dan ook weinig tot geen elkaar tegensprekende richtlijnen zijn, wat met Xen wel nogal eens het geval is. Omdat OpenVZ gesponsord wordt door Parallels Inc. is er ook een nogal stevige band met dit bedrijf, wat er mee voor zorgt dat de verschillende Linux-distributies niet op grote schaal (en in een hoog tempo) wijzigingen aanbrengen in de code van OpenVZ. Dit leidt er toe dat die code beter eenvormig blijft en dus stabieler is (ook al zal de snelheid van de ontwikkeling daar wel wat onder lijden).

5.2.2. Installatie van OpenVZ

De basisinstallatie van OpenVZ is al bij al vrij eenvoudig[141]. Eerst moeten we verifiëren of onze testmachine voldoet aan de minimale systeemvereisten en dit zowel voor wat betreft de hardware als voor wat betreft de indeling der harde schijven. Inzake de hardware zijn er geen problemen (onze testmachine is veel krachtiger dan de minimale vereisten voorzien[142]). We hebben wel een probleem inzake de configuratie van ons CentOS-systeem. We wezen oorspronkelijk slechts een swap-partitie van minimale omvang aan. OpenVZ vraagt expliciet om 2 keer de totale hoeveelheid geïnstalleerd RAM. Dat betekent dus dat we een swap-partitie nodig zouden hebben van 12 GB. Dit lijkt ons een onvolkomenheid in de configuratierichtlijnen van OpenVZ, aangezien Linux op het Intel-platform slechts een swappartitie met een maximale grootte van 2 GB kan adresseren[143].

138 Op de OpenVZ-website wordt gesteld dat er ten opzichte van een standaardinstallatie (i.e. Niet gevisualiseerd) van Linux slechts een performantieverlies is van 1 tot 3 %.

139 Het propriëtaire Virtuozzo kan op dezelfde manier ook Windows aanpassen voor gebruik met meerdere Windows-containers. Voor elke Windows-container is er dan wel een aparte Windows-licentie nodig. In beide gevallen – OpenVZ en Virtuozzo – kunnen de besturingssystemen niet gemixt worden.

140 Althans, ik heb er geen gevonden. Onlangs verscheen er wel een boek in Duitsland (dat ik echter niet gelezen heb) : BAUER, T., *OpenVZ, Das Kleine Handbuch*, Books On Demand, 2008, ISBN 978-3-83701-384-9. Zie voor een bespreking : ZELLER, Thomas in *LinuxLife, ein Sonderheft der PC Magazin DVD-Ausgabe*, nr. 3/2008, pagina 16.

141 Zie : http://wiki.openvz.org/Quick_installation

142 Zie : http://www.parallels.com/en/products/virtuozzo/hcl/

143 Zie voetnoot 46 hierboven. Overigens kan Linux wel meerdere swappartities en zelfs bijkomende swap-bestanden adresseren. Mocht er zich onverhoopt toch een probleem voordoen, dan kunnen we bijgevolg steeds extra swap-ruimte toewijzen.

Standaard verwacht OpenVZ bij de installatie een apart local volume aan te treffen met de naam /vz. Het is dat local volume dat OpenVZ gaat gebruiken om er zijn containers in te plaatsen. Dat local volume hebben we bij onze installatie echter niet voorzien. We moeten het dan ook eerst aanmaken :

```
[root@testmachien ~]#  lvcreate -L 10G -n LV_vz VG_VMS
```

OpenVZ steunt op Red Hat Enterprise Linux. Voor ons een goede zaak, want ons CentOS-systeem is een zo goed als identieke *clone* van Red Hat Enterprise Linux. Voor we tot de feitelijke installatie kunnen overgaan, moeten we wel de software-repository van OpenVZ toevoegen aan onze lijst van repositories. Dat gaat als volgt :

```
[root@testmachien ~]# cd /etc/yum.repos.d

[root@testmachien yum.repos.d]# wget http://download.openvz.org/openvz.repo

[root@testmachien yum.repos.d]# rpm –import http://download.openvz.org/ \

RPM-GPG-KEY-OpenVZ
```

Er zijn vier verschillende kernels van OpenVZ beschikbaar[144]. Elk van die kernels beantwoordt aan de noden van de specifieke processor in een systeem. Ten eerste is er de kernel UP die staat voor een uniprocessor welke maximaal 4 GB RAM kan aanspreken. Ten tweede is er de kernel SMP, die staat voor een multiprocessor die eveneens maximaal 4 GB RAM kan aanspreken. Ten derde is er entnosplit/PAE-kernel, die eveneens staat voor een multiprocessor maar deze keer één (of meer) die dankzij PAE tot 64 GB RAM kan aanspreken. Tenslotte is er nog de enterprise/ent-kernel die evenzeer staat voor een multiprocessor die via PAE tot 64 GB RAM kan aanspreken, maar die dat RAM opsplitst in delen van 4 GB.

Het nut van deze indeling in vier verschillende kernels is wat ons betreft wel wat achterhaald. Het is immers zo dat een 64-bits-processor met een Linux-systeem van het 64-bits-type sowieso tot 64 GB RAM kan adresseren. We hoeven dan ook niet lang na te denken : aangezien we in onze testmachine beschikken over een dual core 64-bits-processor en onze CentOS-installatie er één is van het 64-bits-type, kiezen we voor de SMP-kernel. We geven dan ook volgend commando :

```
[root@testmachien ~]# yum install ovzkernel smp
```

Na verloop van enige tijd is onze aangepaste openvz-kernel geïnstalleerd, samen met een geëncrypteerde key en alle dependencies. Daarmee is de basisinstallatie feitelijk voltooid.

We controleren nu nog even onze GrUB-configuratie. In tegenstelling tot Xen dienen we nu echter niks te veranderen. De openvz-kernel staat bovenaan in ons GrUB-bootlijstje en de *default* startwaarde verwijst daar ook naar.

144Zie : http://wiki.openvz.org/Different_kernel_flavors_%28UP%2C_SMP%2C_ENTERPRISE%2C_ENTNOSPLIT%29

5.2.3. Netwerken onder OpenVZ

Toch kunnen we onze machine nu niet zomaar direct herstarten om met de openvz-kernel aan de slag te gaan. We moeten eerst de netwerkconfiguratie aanpassen (de wijzigingen hebben we aangeduid met grijze achtergrond en onderstreept[145]) :

```
[root@testmachien ~]# vi /etc/sysctl.conf

# Kernel sysctl configuration file for Red Hat Linux

# For binary values, 0 is disabled, 1 is enabled.  See sysctl(8) and

# sysctl.conf(5) for more details.

# Controls IP packet forwarding

net.ipv4.ip_forward = 1

net.ipv4.conf.default.proxy_arp=0

# Controls source route verification

net.ipv4.conf.default.rp_filter = 1

# Do not accept source routing

net.ipv4.conf.default.accept_source_route = 0

# Controls the System Request debugging functionality of the kernel

kernel.sysrq = 0

# Controls whether core dumps will append the PID to the core filename

# Useful for debugging multi-threaded applications

kernel.core_uses_pid = 1

# Controls the use of TCP syncookies

net.ipv4.tcp_syncookies = 1

# Controls the maximum size of a message, in bytes

kernel.msgmnb = 65536
```

145We maakten hier gebruik van de richtlijnen van een zekere Dustin, die enkel nog te vinden zijn via Google's cache : http://64.233.183.104/search?q=cache:1gY-RkCzOuYJ:the1337-geek.com/%3Fm%3D200804+how+to+install+and+run+Openvz+on+CentOS+5.1&hl=nl&ct=clnk&cd=2&gl=be&client=firefox-a Deze richtlijnen wijken lichtjes af van die van de installatiegids op de wiki van OpenVZ (de afwijkingen hebben te maken met de specificiteit van CentOS).

```
# Controls the default maxmimum size of a mesage queue

kernel.msgmax = 65536

# Controls the maximum shared segment size, in bytes

kernel.shmmax = 42949667295 # in plaats van 68719476736

# Controls the maximum number of shared memory segments, in pages

kernel.shmall = 268435456 # in plaats van 4294967296

net.ipv4.tcp_syncookies = 1

net.ipv4.tcp_synack_retries = 2

# We do not want all our interfaces to send redirects

net.ipv4.conf.default.send_redirects = 1

net.ipv4.conf.all.send_redirects = 0
```

Met deze basisinstellingen hebben we feitelijk enkel een onderliggend framework voor het netwerken van virtuele machines of containers ingesteld, steunend op *IP forwarding*. De verdere netwerkconfiguratie dient te gebeuren doorheen de installatie van die containers zelf. We kunnen daarbij een keuze maken tussen het gebruiken van de netwerkkaarten als *virtual ethernet device (veth)* óf als *virtual network device (venet)*.

5.2.3.1. Gebruik van een *virtual ethernet device*

Kiezen we voor gebruik van *veth*[146], dan komt dat feitelijk neer op *bridged networking.* Binnenin het *host operating system* wordt een *bridge* gecreëerd. Daarnaast dienen binnenin de virtuele machines of containers virtuele *ethernet devices* (*veth*) aangemaakt te worden. De configuratie van de *veth-devices* dient te gebeuren in de containers zelf, via de normale Linux-commando's. Dit lijkt erg flexibel, maar het is ook een nadeel. Het toewijzen van IP-adressen, gateways en routes is dan immers de verantwoordelijkheid van de administrator van de virtuele machine zelf. Als er maar één administrator is voor zowel de hostmachine als de virtuele machines, dan is dat geen probleem (alleen maar veel werk). Zijn er echter verschillende administrators voor meerdere virtuele machines, dan is goed overleg wel noodzakelijk, al was het alleen maar om dubbele toewijzing van IP-adressen te vermijden. Deze vorm van netwerken wordt dan ook niet aangeraden voor productieomgeving. Er is gewoon teveel overhead en dat brengt gevaren met zich mee op het vlak van beveiliging.

146Zie : http://wiki.openvz.org/Veth

5.2.3.2. Gebruik van een *virtual network device*

De andere manier die OpenVZ kent om met netwerken om te gaan is ook de standaard manier : *venet*[147]. Deze manier van netwerken lijkt nog het meest op *peer-to-peer networking* tussen de virtuele machines en de host. Hierbij wordt aan *packet switching* gedaan, steunend op de IP-headers van de pakketten. Hét grote voordeel aan *venet* is dat er bijna geen configuratie mee gemoeid is. Er is zo goed als geen *overhead* (tenminste : als OpenVZ bij installatie goed geconfigureerd is). Het toewijzen in de containers van IP-adressen, gateways en routes gebeurt automatisch via de scripts van vzctl[148]. Met deze methode van netwerken wordt er feitelijk gebruikt gemaakt van een volledige router binnenin het gevirtualiseerde systeem. Dit gebeurt door het laden van de kernel-module vznetdevice. De methode venet is dan ook niks anders dan een vorm van *routed networking*.

OpenVZ springt tegelijk vrij eenvoudig én toch ook heel flexibel om met netwerken. In feite zijn er nauwelijks beperkingen. Omdat we met OpenVZ eigenlijk de mogelijkheden van één enkel besturingssysteem vermenigvuldigen, zonder dat er daarvoor ingrijpende veranderingen nodig zijn, kunnen we die verveelvoudigde mogelijkheden ook helemaal benutten. Vanuit het standpunt van OpenVZ bekeken, zijn alle virtuele machines of containers niet meer dan volledig afzonderlijke, volwaardige computers. We kunnen tussen deze computers dan ook alle mogelijke netwerkconfiguraties opzetten. Op de wiki-pagina's van OpenVZ kunnen we bijvoorbeeld praktische handleidingen vinden voor het opzetten van firewalls binnenin containers, het opzetten van gerouteerde netwerken tussen de virtuele machines, het creëren van VLAN's, enz.

5.2.4. OpenVZ en beveiliging

Ook voor de beveiliging gelden de voordelen van het werken met Linux als basis voor virtualisatie. Alle normale opties voor de beveiliging van een gewoon Linux-systeem zijn ook gewoon van toepassing. Wel zijn er een paar dingen die we in het oog moeten houden.

Zo wordt ons in de OpenVZ-handleidingen aangeraden om SELinux uit te zetten omdat het problemen zou kunnen veroorzaken. In plaats daarvan stellen we SELinux liever in op *permissive*. Dan kunnen we nog steeds via de logbestanden nagaan of er ergens problemen zijn. We zorgen er ook voor dat de firewall op *enabled* staat ingesteld. Net zoals bij de netwerkconfiguratie, profiteren we inzake beveiliging weer van het feit dat OpenVZ ons toestaat zowel onze host als onze virtuele machines of containers te behandelen als volwaardige, 'gewone' Linux-computers.

Het is ook zo dat we met OpenVZ gebruik kunnen maken van *netfilter connection tracking*, waardoor het beveiligingsniveau van de firewall nog kan worden opgeschroefd. Eigenlijk gaat het om twee verschillende mogelijkheden die geboden worden door de firewall, enerzijds *netfilter* en anderzijds *connection tracking*.

“Netfilter is meer dan alleen een firewall faciliteit. Het voorziet in een abstracte structuur om netwerkverkeer in de kernel te controleren. Filteren is slechts één van

147 Zie : http://wiki.openvz.org/Venet

148 Bij het aanmaken van virtuele machines worden als parameter de nodige netwerkgegevens wel meegegeven, maar ze worden daarna, door de vzctl-scripts als het ware 'onthouden'. Je moet ze bij een herstart dus niet meer opnieuw opgeven.

de toepassingen van deze structuur. Diverse delen van de kernel kunnen communiceren met netfilter en krijgen daardoor toegang tot de verkeersstroom door de protocolstack zonder zelf deel te hoeven uitmaken van de protocolstack. Netfilter voorziet ook in communicatie met processen in userspace, zodat complexere bewerkingen buiten de kernel kunnen worden uitgevoerd. Dit is een duidelijke verbetering ten opzichte van ipchains waarbij de mogelijkheden beperkt werden doordat alle bewerkingen in kernelspace moesten plaatsvinden. Door de heldere structuur zijn functionaliteiten als NAT, transparante proxy's en connection tracking relatief makkelijk te implementeren."

"Bij connection tracking wordt de status van iedere connectie bijgehouden. Op basis van deze status worden beslissingen genomen over het al dan niet doorlaten van netwerkpakketjes. Alleen pakketjes die behoren tot een geregistreerde connectie worden doorgelaten."[149]

Om *netfilter connection tracking* te activeren moeten we wel nog iets aan onze configuratie veranderen. Met name dien we de regel **options ip_conntrack ip_conntrack_enable_ve0=1** toe te voegen aan het bestand /etc/modprobe.conf. Dat doen we dan ook.

Dan kunnen we nu onze testmachine herstarten en gebruik gaan maken van OpenVZ.

5.2.5. Hulpprogramma's voor OpenVZ

OpenVZ is nu wel actief (wat we kunnen controleren met het commando **uname -r**), maar verder kunnen we er weinig mee doen. De gebruikerstools voor OpenVZ ontbreken immers nog. Concreet gaat het om beheertools (vzctl) en quotatools (vzquota). Vzctl staat in voor het creëren of vernietigen, het starten en stoppen van de virtuele containers en dergelijke meer, terwijl vzquota instaat voor het beheren van quota's van diezelfde containers. We installeren beide tools heel makkelijk :

```
[root@testmachien ~]# yum install vzctl vzquota
```

OpenVZ werkt met een cache voor templates van besturingssystemen. Om die cache te kunnen aanmaken, hebben we het programma **vzpkg** nodig. Daarnaast gebruikt OpenVZ aangepaste versies van yum en rpm om de besturingssystemen in de containers te installeren en up to date te houden. Het commando om deze aangepaste versies te installeren levert echter problemen op :

```
[root@testmachien ~]# yum install vzpkg vzyum vzrpm43-python vzrpm44-python

...

Error: Missing Dependency: cElementTree.so is needed by package vzyum
```

Dit stukje software maakt nochtans deel uit van onze CentOS-installatie. Het probleem blijkt te maken te hebben met het feit dat de ontwikkeling van OpenVZ voor 64-bit-systemen ietwat achterloopt. Er zit dan ook niets anders op dan onze toevlucht te nemen tot een kunstgreep[150], waardoor vzyum en vzrpm daarna gebruik zullen maken van de 'gewone' yum- en rpm-pakketten. Ook dienen een aantal paden aangepast

149Citaat afkomstig van Bart Jan Kelter (http://www.kelter.nl/artikel-firewall.html).
150Zie : http://wiki.openvz.org/Install_OpenVZ_on_a_x86_64_system_Centos-Fedora

te worden voor gebruik onder een 64-bits-systeem[151]. We kunnen dit alles verwezenlijken door een script te downloaden[152], het uit te pakken en het vervolgens uit te voeren. Dat doen we als volgt :

```
[root@testmachien ~]# wget \

http://linux.carreira.com.pt/ovzutils/setx86_64-0.3.tar.gz

[root@testmachien ~]# tar xzvf setx86_64-0.3.tar.gz

[root@testmachien ~]# sh setx86_64
```

Dit script test eerst of de machine wel goed genoeg is geconfigureerd om het script te kunnen uitvoeren. Vervolgens vraagt het ons of het goed is om door te gaan. Daarna worden in een paar seconden alle nodige patches uitgevoerd.

5.2.6. OpenVZ *templates*

OpenVZ werkt met zogenaamde templates. Dit zijn vooraf geconfigureerde installatiesjablonen. Er zijn verschillende van deze templates aanwezig op de website van OpenVZ. Een aantal daarvan zijn officieel (dit wil zeggen dat ze gecertificeerd zijn voor gebruik onder OpenVZ). Daarnaast zijn er nog een aantal niet-officiële, die werden aangeleverd vanuit de gebruikersgemeenschap.[153] Om te kunnen werken met deze templates hebben we naast het eerder geïnstalleerde (en gepatchedste) programma vzpkg ook nog zogenaamde template metadata nodig.

Deze template metadata bevat informatie over een specifiek template voor een bepaald besturingssysteem. Het bevat een lijst van alle pakketten die het template bevat, de locatie van de repositories, scripts die eigen zijn aan de eigenlijke Linux-distributie en publieke encryptiesleutels die gebruikt worden om de handtekeningen van de pakketten te kunnen controleren. De metadata voor de verschillende templates worden opgeslagen in de folder voor het template waaraan ze gekoppeld zijn. Voor CentOS-5 bijvoorbeeld in /vz/template/centos/5/config/.

Met de informatie die vervat zit in de metadata kan een template voor een besturingssysteem worden aangemaakt. Daarbij worden de nodige pakketten gedownload vanuit de in de metadata opgegeven repository en vervolgens geïnstalleerd in een zogenaamde gzipped tarball. Ze worden dus gearchiveerd en gecomprimeerd. Dit noemt men een template cache. Deze cache-bestanden worden opgeslagen in de folder /vz/template/cache/. Vanuit deze cache kan razendsnel een container worden opgestart. En met razendsnel bedoelen we dus echt wel razendsnel : het duurt niet langer dan enkele seconden.[154]

151 Zie : http://wiki.openvz.org/Install_OpenVZ_on_a_x86_64_system_Centos-Fedora
152 Zie : http://linux.carreira.com.pt/ovzutils/setx86_64-0.3.tar.gz
153 Zie : http://wiki.openvz.org/Download/template/precreated
154 Het opstarten van een container is eigenlijk niks anders dan het uitpakken van tar-bestand en het actief maken ervan (i.e. Het laden in de toegewezen geheugenruimte). Vandaar de snelheid.

5.2.7. Een eerste OpenVZ-container installeren

Voor we een eerste container kunnen installeren, dienen we eerst onze netwerkconfiguratie na te kijken. Om te beginnen dient onze hostmachine[155] zichtbaar en bruikbaar is op ons netwerk. Dat kunnen we controleren door te trachten het Internet te gebruiken (bijvoorbeeld door te pingen naar www.google.be) en/of door vanaf een andere machine op ons netwerk te pingen naar het IP-adres van onze eerste netwerkkaart (192.168.1.11). Beide test zijn succesvol, dus dat is in orde. Vervolgens dienen we voor onze eerste container hetzij een netwerkadres te voorzien dat behoort tot hetzelfde subnet, hetzij op onze hostmachine routering op te zetten naar het adres in een ander subnet dat we willen toewijzen aan onze eerste container. Omdat we willen dat onze servers (i.e. onze containers) behoren tot een ander subnet dan onze hostmachine én tot een ander subnet dan onze clients, dienen we dus routering op te zetten. Onder Linux kunnen we dat op verschillende manieren doen. Zo zouden we dit bijvoorbeeld via DHCP en/of DNS kunnen doen. Omdat we echter maar zeven servers of containers een adres moeten toewijzen, lijkt dit ons een beetje overdreven. We kunnen dan beter het bestand /etc/hosts aanpassen. Dat aanpassen kunnen we manueel doen, of via de grafische network-applicatie. We gebruiken deze laatste mogelijkheid :

Afbeelding 24 : Aanpassen van de netwerkconfiguratie

Na het vervolledigen en opslaan van alle gegevens (i.e. IP-adressen, hostnames en aliassen), herstarten we het netwerk :

```
[root@testmachien ~]# service network restart
Shutting down interface eth0:                              [  OK  ]
Shutting down interface venet0:                            [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Disabling IPv4 packet forwarding:  net.ipv4.ip_forward = 0
                                                           [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface eth0:                                [  OK  ]
Bringing up interface venet0:                              [  OK  ]
```

We bekijken nu onze routeringstabel :

```
[root@testmachien ~]# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use
```

155In het OpenVZ-jargon noemt men dit een HN (Hardware Node).

```
Iface

192.168.1.0     *               255.255.255.0   U     0      0        0
eth0

10.0.10.0       *               255.255.255.0   U     0      0        0 ve-
net0

169.254.0.0     *               255.255.0.0     U     0      0        0 ve-
net0

default         192.168.1.1     0.0.0.0         UG    0      0        0
eth0
```

We hebben nu niet alleen de IP-adressen voor onze (nog aan te maken) containers geconfigureerd. Ook beschikken we vanaf nu over drie verschillende subnetten. Eén waar onze hostmachine toe behoort (netwerk-ID 192.168.1.0), één voor onze containers (network-ID 10.0.10.0) en één voor onze client-machines (169.254.0.0).

Nu kunnen we dan ook van start gaan met het opzetten van onze containers. Daarbij moeten we volgende zaken in acht nemen :

- elke container moet een eigen identificatienummer krijgen
- elke container dient een template met een besturingssysteem toegewezen te worden
- de container zelf dient gecreëerd te worden

In het toewijzen van het uniek identificatienummer zijn we vrij. Wel moeten we er rekening mee houden dat ID 0 gereserveerd is voor de host (gelijkaardig aan Xen, dus). Ook de nummers 1 tot en met 100 zijn gereserveerd voor eventueel toekomstig gebruik door OpenVZ zelf. Wij kiezen ervoor om onze containers te nummeren van nummer 200. Dit heeft als bijkomend voordeel dat de container-ID's overeenstemmen met het laatste getal in de IP-adressen van onze containers.

Vervolgens bekijken we over welke templates van 64-bits-besturingssystemenen we beschikken op onze host :

```
[root@testmachien ~]# vzpkgls | grep x86_64

fedora-core-3-x86_64-minimal

fedora-core-3-x86_64-default

fedora-core-5-x86_64-minimal

fedora-core-5-x86_64-default

fedora-core-4-x86_64-minimal

fedora-core-4-x86_64-default

fedora-core-6-x86_64-minimal
```

```
fedora-core-6-x86_64-default

fedora-7-x86_64-small

fedora-7-x86_64-minimal

fedora-7-x86_64-default

centos-5-x86_64-minimal

centos-5-x86_64-default

centos-4-x86_64-minimal

centos-4-x86_64-default
```

Dat zijn er dus veel meer dan we nodig hebben[156]. Voor de installatie van onze containers zullen we telkens vertrekken van centos-5-x86_64-default als template.

Om containers te creëren maken we gebruik van het commando vzctl create. Dit commando kan gebruikt worden om een container aan te maken, enkel met een identificatienummer en het te gebruiken template. Als we dat echter doen, dan moeten we na afloop handmatig allerlei parameters instellen en/of vervolledigen. Dat is nogal bewerkelijk en kan gemakkelijk leiden tot fouten. OpenVZ levert ons echter de mogelijkheid om voorbeeld-configuratiebestanden te gebruiken. Deze zijn te vinden in de folder /etc/sysconfig/vz-scripts/. In die folder vinden we het bestand ve-vps.basic.conf-sample, een veelgebruikt voorbeeldbestand. We zouden nu een eerste container kunnen creëren met het commando **vzctl create 200 –ostemplate centos-5-x86_64-default –config vps.basic**. Omdat we echter steeds hetzelfde template met steeds hetzelfde voorbeeld-bestand gaan gebruiken, is het beter deze parameters ineens in te voeren in het centrale configuratiebestand van OpenVZ. Dat doen we dan ook door in het bestand /etc/sysconfig/vz de volgende regels als volgt aan te passen : “CONFIGFILE="vps.basic"” en “DEF_OSTEMPLATE="centos-5-x86_64-default"”.

Nu gaan we daadwerkelijk een eerste cache template aanmaken (dit kan wel wat tijd vragen) :

```
[root@testmachien ~]# vzpkgcache -f centos-5-x86_64-default
```

Aan de hand van de metadata worden nu alle pakketten gedownload van onze repository die nodig zijn voor een 'verse' installatie van CentOS. Omdat we eerder reeds een lokale repository installeerden, gaat dit vrij snel (zoniet, dan moet alles van over het Internet gedownload worden, hetgeen uiteraard meer tijd vraagt). Na afloop controleren we of onze cache werd aangemaakt :

```
[root@testmachien ~]# vzpkgls -c

centos-5-x86_64-default
```

Nu creëren we onze eerste container :

156Deze verzameling templates werd, samen met nog een hele reeks 32-bits-templates, geïnstalleerd toen we daarstraks via het gedownloade script onze installatie van de OpenVZ-tools aanpasten voor gebruik onder 64-bits-Linux.

```
[root@testmachien ~]# vzctl create 200 --ostemplate centos-5-x86_64-default

Creating VE private area (centos-5-x86_64-default)

Performing postcreate actions

VE private area was created
```

We kunnen nu onze eerste container daadwerkelijk opstarten :

```
[root@testmachien ~]# vzctl start 200

Starting VE ...

VE is mounted

Setting CPU units: 1000

Configure meminfo: 65536

VE start in progress...
```

We testen nu of onze container goed en wel actief is. Dat doen we door te proberen de editor nano te installeren met behulp van de OpenVZ-tool vzyum :

```
[root@testmachien ~]# vzyum 200 install nano

...

Installed: nano.x86_64 0:1.3.12-1.1

Complete!
```

Dat werkt dus naar behoren. Onze eerste container is nu geïnstalleerd en actief. Nu moeten we hem alleen nog maar configureren. Om dat te kunnen doen, stoppen we eerst de container :

```
[root@testmachien ~]# vzctl stop 200

Stopping VE ...

VE was stopped

VE is unmounted
```

Met behulp van het commando vzctl set gaan we nu volgende zaken aanpassen :

- de opstart-parameters van de container
- de netwerk-parameters van de container
- de paswoorden voor de gebruikers van de container
- de parameters inzake Quality of Service voor de container.

5.2.8. Parameters voor OpenVZ-containers

Allereerst stellen we de opstart-parameters voor de container in. Het gaat er meer bepaald over of de container automatisch dient te worden gestart wanneer we de hostmachine rebooten. Dat is uiteraard het geval. Daarom geven we het volgende commando :

```
[root@testmachien ~]# vzctl set 200 --onboot yes --save

Saved parameters for VE 200
```

Nu gaan we de netwerkparameters toewijzen. Het gaat daarbij om het IP-adres, de hostname en het adres van de DNS-server (i.e. de router). Eerst stellen we de hostname in :

```
[root@testmachien ~]# vzctl set 200 --hostname krb.testdomein.org --save

Saved parameters for VE 200
```

Vervolgens het IP-adres :

```
[root@testmachien ~]# vzctl set 200 --ipadd 10.0.10.200 --save

Saved parameters for VE 200
```

En tenslotte de DNS-server :

```
[root@testmachien ~]# vzctl set 200 --nameserver 192.168.1.11 --save

Saved parameters for VE 200
```

Deze commando's kunnen overigens ook worden ingegeven terwijl de container actief is. Als we dan bovendien de parameter --save weglaten uit het commando, dan zullen de wijzigingen niet worden opgeslagen (ze zullen dus niet persistent zijn). Dat kan nuttig zijn om tests uit te voeren.

We starten nu opnieuw onze eerste container en kijken na of de secure shell (SSH) actief is. Daarbij maken we gebruik van de commando-optie exec, waarmee we commando's kunnen uitvoeren binnenin onze container :

```
[root@testmachien ~]# vzctl start 200

Starting VE ...

VE is mounted

Adding IP address(es): 10.0.10.200

Setting CPU units: 1000

Configure meminfo: 65536

Set hostname: krb.testdomein.org
```

```
File resolv.conf was modified

VE start in progress...

[root@testmachien ~]# vzctl exec 200 service sshd status

sshd (pid 9484) is running...
```

En inderdaad, de SSH-service is actief binnenin onze eerste container. Nu moeten we het paswoord voor de root-gebruiker instellen :

```
[root@testmachien ~]# vzctl set 200 --userpasswd root:2Bn2b

Changing password for user root.

passwd: all authentication tokens updated successfully.
```

Merk op dat we nu niet de expliciete opdracht geven om deze informatie op te slaan (i.e. geen parameter –save). Ook is het zo dat dit paswoord niet wordt opgeslagen in de configuratie-bestanden van de container. Wel onder geëncrypteerde vorm in het bestand /etc/shadow. Het aanmaken van gewone gebruikers-accounts en -paswoorden kan op dezelfde manier gebeuren, of binnenin de container zelf.

We kunnen nu één voor één al onze containers op dezelfde manier creëren en activeren. Of we kunnen daarvoor één, globaal script maken. Dat doen we dan ook (zie Appendix Z). Nadat we het script hebben uitgevoerd, zijn alle zes onze containers (of servers) actief :

```
[root@testmachien ~]# vzlist -a

      VEID      NPROC STATUS  IP_ADDR         HOSTNAME

       200         16 running 10.0.10.200     krb.testdomein.org

       201         16 running 10.0.10.201     groupware.testdomein.org

       202         16 running 10.0.10.202     crm.testdomein.org

       203         16 running 10.0.10.203     erp.testdomein.org

       204         16 running 10.0.10.204     vdi.testdomein.org

       205         16 running 10.0.10.205     backup.testdomein.org

       206         16 running 10.0.10.206     print.testdomein.org
```

We controleren nogmaals of de ssh-service wel werkt op al deze servers. Dat kunnen we doen door voor elke server afzonderlijk het commando **vzctl exec 201 service sshd status** uit te voeren. We kunnen echter ook met één enkel commando hetzelfde bereiken :

```
[root@testmachien ~]# for i in `vzlist -o veid -H`; do \

echo “VPS $i”; vzctl exec $i service sshd status; done
```

```
“VPS 200”

sshd (pid 9484) is running...

“VPS 201”

sshd (pid 11304) is running...

“VPS 202”

sshd (pid 11817) is running...

“VPS 203”

sshd (pid 13385) is running...

“VPS 204”

sshd (pid 13899) is running...

“VPS 205”

sshd (pid 15435) is running...

“VPS 206”

sshd (pid 15951) is running...
```

Er doen zich daarbij duidelijk geen problemen voor, zodat we nu op een veilige manier toegang kunnen krijgen tot onze servers.

Natuurlijk willen we niet dat één der virtuele machines (i.e. containers) al het beschikbare RAM-geheugen naar zich toetrekt. Evenmin willen we dat één (of alle) container(s) te weinig RAM-geheugen zou toebedeeld krijgen. Daarom gaar we nu de beschikbare resources splitsen over onze 6 containers én onze hostmachine. We gaan onszelf daarbij wat extra ruimte geven, zodat we later nog enkele containers bij kunnen maken, mocht dat nodig zijn. We gaan dus een configuratie creëren voor 10 containers die alle gelijktijdig actief kunnen zijn op de hostmachine. Dit doen we als volgt :

```
[root@testmachien ~]# cd /etc/sysconfig/vz-scripts

[root@testmachien ~]# vzsplit -n 10 -f vps.mytest

Config /etc/vz/conf/ve-vps.mytest.conf-sample was created
```

Voor we deze instellingen permanent actief maken, controleren we eerst of alles wel in orde is. Daartoe gebruiken we het validatiecommando **vzcfgvalidate** :

```
[root@testmachien ~]# vzcfgvalidate ve-vps.mytest.conf-sample

Validation completed: success
```

Dit commando maakt een optimale test-configuratie voor ons aan. Vervolgens kunnen we nakijken of onze werkelijke configuratiebestanden wel geoptimaliseerd zijn :

```
[root@testmachien vz-scripts]# vzcfgvalidate /etc/sysconfig/vz-
scripts/200.conf

Error: limit should be = 9223372036854775807 for vmguarpages (currently,
2147483647)

Error: limit should be = 9223372036854775807 for oomguarpages (currently,
2147483647)

Error: limit should be = 9223372036854775807 for physpages (currently,
2147483647)
```

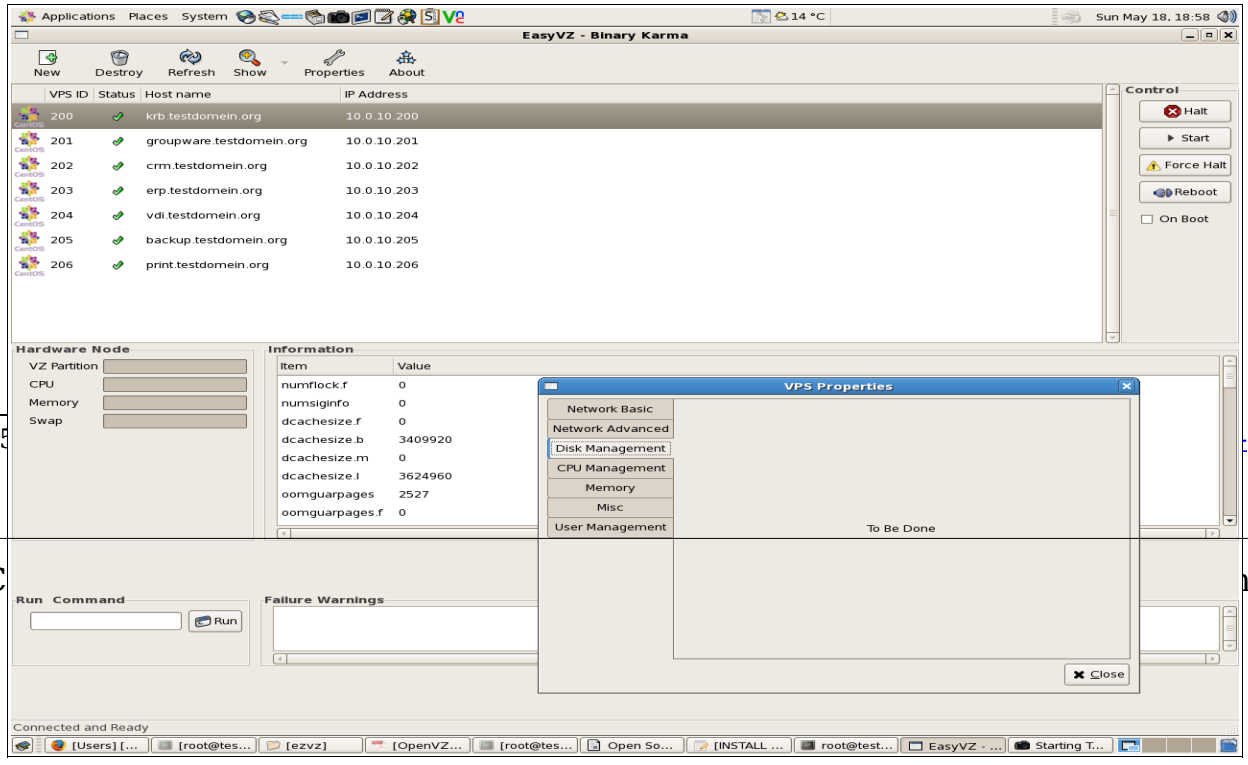
Dat blijkt dus niet het geval te zijn. We moeten dan ook (manueel) in elk configuratie-bestand de drie genoemde parameters veranderen. Dat doen we dan ook. Daarna herhalen we voor al onze containers het validatiecommando (hier geven we slechts één voorbeeld weer :

```
[root@testmachien vz-scripts]# vzcfgvalidate \

/etc/sysconfig/vz-scripts/206.conf

Validation completed: success
```

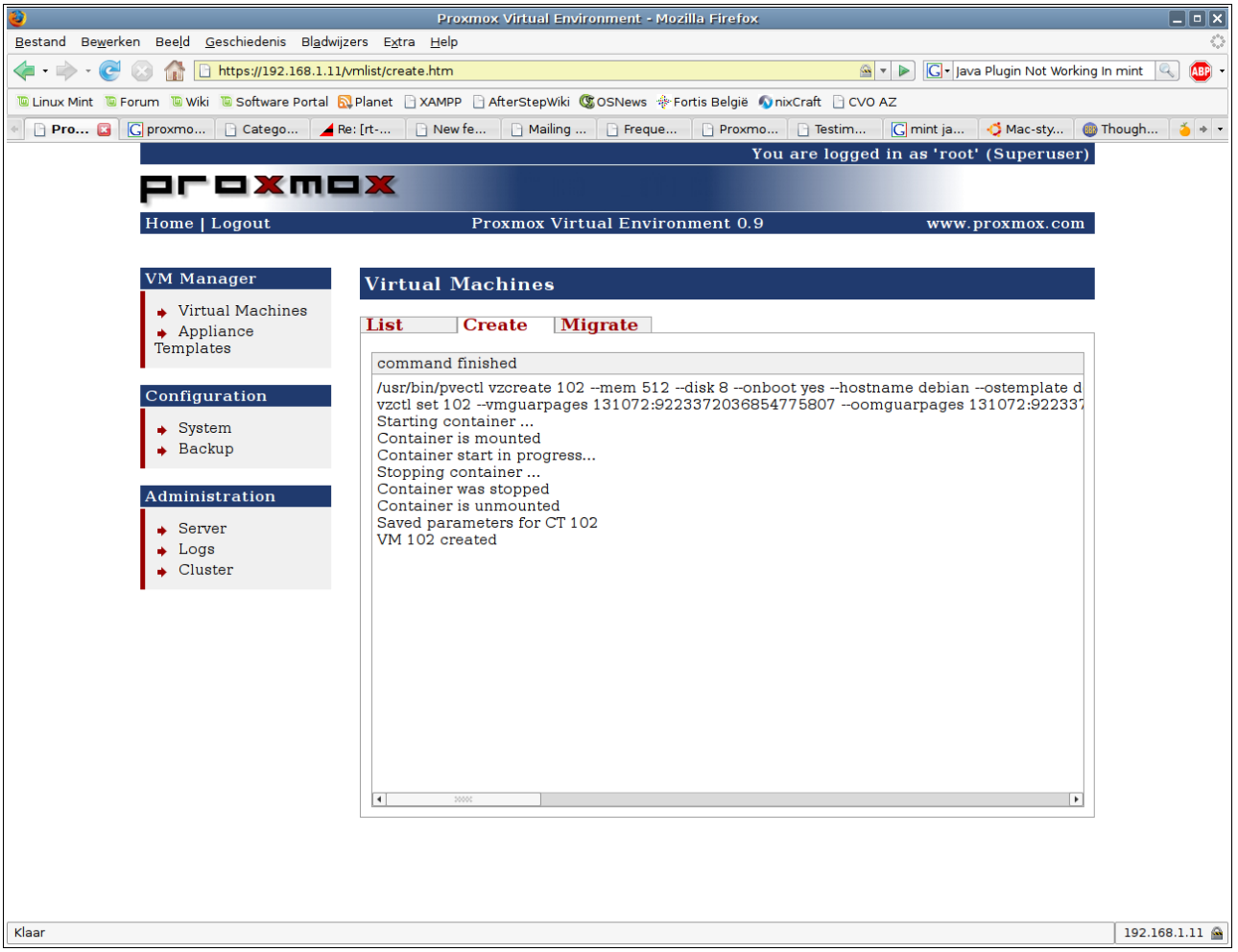
Onze containers zijn nu volledig optimaal geïnstalleerd. We kunnen ze nu één voor één gaan configureren voor gebruik door de vzw. We kijken echter ook nog even naar andere managementtools die het werken met OpenVZ kunnen vergemakkelijken.

5.2.9. Programma's voor het beheer van OpenVZ-containers

OpenVZ wordt zo goed als altijd enkel beheerd met de meegeleverde command line management tools. Toch zijn er ook twee grafische programma's beschikbaar. Het eerste daarvan is EasyVZ[157]. Het gaat wel om zogenaamde alpha-software, die dus nog lang niet over alle functionaliteit beschikt, zoals blijkt uit de schermafdruk op de volgende pagina (let op de melding *to be done*) :



Toch kan er verder al wel wat mee gedaan worden. De software is echter niet erg veilig, aangezien er geen authenticatie aan gekoppeld is. Daarnaast is er ook nog ProxMox Virtual Environment[158]. Dat is eigenlijk geen tool op zich, maar wel een hele installatiemethode. Het gaat om een downloadbaar CD-image (een zogenaamd ISO-bestand, dus). Als je dat brand op een CD en daarna je computer opstart vanaop deze CD, dan wist die je hele bestaande configuratie (!) en installeert in de plaats daarvan op een harde schijf naar keuze een hele omgeving die in één klap geschikt is voor gebruik met OpenVZ en dat met een vrij goede grafische gebruikersinterface (geïmplementeerd als een webpagina). Het werkt allemaal erg makkelijk en transparant, getuige volgende schermafdruk :



Afbeelding 35 : Een virtuele machine aanmaken met ProxMox VE

Toch is deze tool niet ideaal. Om te beginnen maakt hij je hele installatie ongedaan. Bovendien is de automatische installatie waarvoor gezorgd wordt niet geoptimaliseerd. Zo wordt geen rekening gehouden met RAID-vereisten. Daar moet je dan zelf voor instaan, wat enkel kan met een hardware-RAID-controller. Ook is de gebruikte installatie nadien enkel geschikt voor gebruik met 32-bits-besturingssystemen. Ook

158Zie : http://www.proxmox.com/

het onderliggende besturingssysteem van de host (Debian Linux 4) is er enkel in een 32-bits-variant. Wel is het zo dat met deze tool je hele basisinstallatie is afgehandeld in minder dan vijf minuten, wat toch wel indrukwekkend is. Natuurlijk moet je daarna nog wel al je virtuele machines (of containers) installeren. De grafische omgeving maakt dat wel overzichtelijk en makkelijk, maar met ons zelfgemaakte script gaat het toch wel veel sneller. Een laatste pluspunt van ProxMox VE is wel dat dit systeem ook kan omgaan met virtuele machines voor KVM. Maar dat bekijken we meer in detail in het volgende hoofdstuk.

5.2.9. OpenVZ – Besluit

Van alle drie de virtualiseringsmethodes die we in dit werk bekijken, is OpenVZ ontegenzeglijk de makkelijkste. Wie met Linux overweg kan, kan OpenVZ opzetten aan de hand van eenvoudige richtlijnen, die zo toepasbaar zijn, zonder moeizame configuratie. Ook in de verschillende containers of virtuele machines zelf, is er geen extra competentie nodig. De kennis van Linux volstaat.

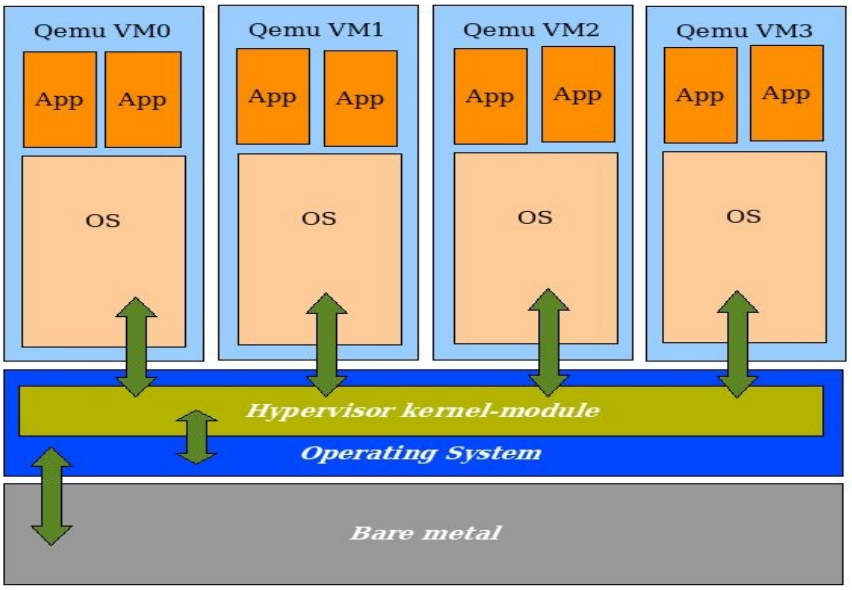
Als daarbij gekozen wordt voor de standaardinstelling van het netwerk (i.e. met *virtual network devices)*, dan gaat het installeren van containers of virtuele machines haast automatisch. Wie liever zelf meer controle heeft en niet opziet tegen een ietwat moeilijker configuratie, kan dat ook.

OpenVZ is zonder enige twijfel ook bijzonder performant. Dat hoeft uiteraard niet te verwonderen. Er is zo goed als geen overhead, aangezien de virtualisatie gebeurt op het niveau van het besturingssysteem. De omweg van een *hypervisor* dient dan ook niet gemaakt te worden.

5.3. KVM/Qemu

5.3.1. Inleiding

Als laatste mogelijke oplossing voor ons virtualisatieprobleem behandelen we nu de Kernel-based Virtual Machine ofwel KVM[159]. Het eerste optreden van KVM als virtualisatiemethode dateert van februari 2007 met versie 2.6.20 van de Linux-kernel. Daar waar Xen als *hypervisor* in de plaats treedt van de Linux-kernel en daar waar OpenVZ een *hypervisor* overbodig maakt door die kernel te patchen, vormt KVM de Linux-kernel feitelijk om tot *hypervisor*. Momenteel gebeurt dat door KVM te laden als een kernel-module, die expliciet geladen moet worden door de gebruiker en die de mogelijkheden van de kernel uitbreidt. Het is echter voorzien dat in toekomstige Linux-versies KVM standaard aanwezig zal zijn in de kernel, waardoor de KVM-interface dan ook direct aanspreekbaar zal zijn, zonder dat tussenkomst van de gebruiker nodig is. Vereenvoudigd ziet KVM-virtualisatie er als volgt uit :



Afbeelding 36 : *Kernel-based Virtualization*

Om KVM te kunnen gebruiken, dient de hostcomputer wel te beschikken over een processor met hardwarematige virtualiseringsextensies[160]. KVM zelf doet eigenlijk niet zoveel. Het is niet meer dan een interface voor een ander programma dat moet

159Zie : http://en.wikipedia.org/wiki/Kernel-based_Virtual_Machine en http://kvm.qumranet.-com/kvmwiki

160Er is wel een begin gemaakt met het mogelijk maken van paravirtualisatie doorheen het schrijven van specifieke driversoftware voor het netwerk en de harde schijftoegang. Dit zou dan leiden tot veel betere performantie. Maar, zoals de auteurs van deze drivers zelf zeggen *"We (Qumranet) have Linux paravirtual network support working, but not ready for general consumption yet"* (http://article.gmane.org/gmane.comp.emulators.kvm.devel/2276).

draaien in de zogenaamde user space (ring 3). Het is dat user space programma dat het mogelijk maakt voor virtuele machines om een afgeschermd deel van het RAM-geheugen te gebruiken, gebruik te maken van virtuele harde schijven en van het scherm op de hostcomputer. Momenteel is het user space programma dat gebruik wordt QEMU[161].

QEMU is een vrij oud programma. Oorspronkelijk was het enkel een zogenaamde emulator. Dit wil zeggen dat het met QEMU mogelijk was (en is) op de hostcomputer (bijvoorbeeld een host van het Intel-type) een andere stadsarchitectuur als gast te laten draaien (bijvoorbeeld een gast voor een ARM-processor). Omdat in zo'n geval de systeemaanroepen voor de ARM-processor dienen te worden vertaald naar die voor de Intel-processor, zorgt dit uiteraard voor veel performantieverlies. Door het gebruik van *dynamic translation* wordt daaraan in zekere mate verholpen. Met *dynamic translation* wordt de vertaalslag als het ware *just in time* uitgevoerd en tegelijk opgeslagen in een cache in het RAM-geheugen. Als de vertaalslag opnieuw nodig is wordt de desbetreffende instructie uit de cache opgehaald. Aangezien de meeste benodigde 'vertalingen' vrij dikwijls terugkomen, leidt dit tot een aanzienlijke performantieverbetering.

Naast deze emulatiekant van QEMU is er dankzij KVM nu ook de virtualiseringsmogelijkheid. Om die te kunnen gebruiken, is er een zogenaamde versneller nodig. Dat is een hulpprogramma dat ervoor zorgt dat de code van de virtuele machine rechtstreeks kan worden uitgevoerd op de processor van de host. Met andere woorden : dit hulpprogramma draait in ring -1, net zoals een hypervisor als bijvoorbeeld Xen.

Voor februari 2007 gebruikte Qemu als hulpprogramma KQEMU. De ontwikkelaars van Qumranet uit Israël hebben met KVM echter een nieuw hulpprogramma voor QEMU geschreven[162].

5.3.2. Installatie van KVM/Qemu

Zoals meestal in de open source wereld kunnen we KVM installeren als een binair (i.e. reeds gecompileerd) bestand, zowel als vanuit de broncode. Wel is het zo dat ons door de ontwikkelaar zelf wordt afgeraden om de installatie vanuit de broncode uit te voeren voor een zogenaamd productiesysteem[163], gezien het gevaar voor bugs tengevolge van de ontwikkelingssnelheid der code. We zullen ons dan ook beperken tot de installatie van het binair bestand afkomstig van de CentOS-repository.

Het installeren van KVM is vrij eenvoudig[164]. Onder CentOS (of andere Red Hat Linux derivaten) gaat dat als volgt :

Eerst installeren we QEMU en daarna de de kernelmodule met de dependencies :

161 Zie : http://en.wikipedia.org/wiki/QEMU en http://fabrice.bellard.free.fr/qemu/

162 De website van Fabrice Bellard, de ontwikkelaar van QEMU, maakt zelfs nog geen melding van KVM.

163 Zie punt 2.2. van de KVM FAQ (Is KVM Stable ?) : http://kvm.qumranet.com/kvmwiki/FAQ#head-4a97776c4810df6b00037b39d88374fb97317112

164 Er bestaat een overzichtelijke, Nederlandstalige handleiding voor het installeren van KVM op een Ubuntu Server : http://wiki.nedlinux.nl/index.php?page=+KVM%2FQEMU+op+je+server

```
[root@testmachien ~]# yum install qemu

[root@testmachien ~]# yum install kvm-kmod kvm
```

Vervolgens herstarten we de machine en daarmee is wat installatie betreft de kous af. We moeten natuurlijk wel nog wat configureren.

5.3.3. Netwerken onder KVM/Qemu

We beginnen bij het netwerk. De netwerkconfiguratie onder KVM is in feite identiek aan deze onder Qemu[165]. Daarbij wordt gebruik gemaakt van het zogenaamde tuntap device.

Tuntap[166] zelf heeft strikt genomen niks te maken met Qemu. Tun en Tap zijn beide virtuele netwerkdrivers voor de kernel van het besturingssysteem van de hostcomputer. Door gebruik te maken van een tuntap device kan het besturingssysteem pakketten naar een programma in user space sturen (bijvoorbeeld naar een virtuele machine, wat wezenlijk niks anders is dan een programma in user space) en omgekeerd. Feitelijk komt het erop neer dat doorheen het tuntap device een netwerk op de host wordt geëmuleerd. Tun en Tap werken elk op een ander niveau (en bijgevolg ook op een andere wijze).

Tun werkt op het niveau van de netwerklaag van het OSI-model[167] (Layer 3). Dat wil zeggen dat het omgaat met pakketten die geschikt zijn voor Layer 3, bijvoorbeeld IP-pakketten. Tun wordt dan ook gebruikt bij het routeren van pakketten.

Tap daarentegen werkt op het niveau van de datalinklaag van het OSI-model (Layer 2). Het simuleert een Ethernet-device en gaat dan ook om met Ethernet frames. Tap wordt dan ook gebruikt voor zogenaamd *bridged networking*, waarbij frames worden doorgestuurd naar een specifiek MAC-adres, welks achterhaald wordt via broadcasts.

De tuntap-driver waarvan Qemu gebruikt maakt, ondersteunt dan ook twee manieren van werken : tun en tap. Qemu gebruikt echter enkel de tap-faciliteit van de driver.

Stel dat we één enkele virtuele machine aanmaken en die netwerkmogelijkheden willen geven middels één enkele geëmuleerde netwerkkaart. In dat geval zullen we te maken krijgen met drie verschillende netwerkkaarten. Eerst en vooral is er de werkelijke, fysieke netwerkkaart (bijvoorbeeld eth0). Deze kaart heeft een eigen MAC-adres en een eigen IP-adres. In de virtuele machine bevindt zich eveneens een (virtuele) netwerkkaart. Standaard geeft Qemu deze virtuele kaart een in Qemu ingebouwd vast MAC-adres mee. Het is dan ook verkieslijk om voor elke virtuele machine zelf een MAC-adres te definiëren[168] en dat mee te geven in de opstartopties van die virtuele machine (zie verder). Via diezelfde opstartopties geven we ook een IP-adres mee aan de virtuele netwerkkaart. Tenslotte is er nog een derde netwerkkaart, met name tap0. Bij gebruik van *root networking* zal Qemu voor elke virtuele machine, naast de virtuele netwerkkaart in de virtuele machine, een nieuw tap-device aanmaken (bij-

165 Qemu is een emulatiesysteem, dat ontwikkelt werd zonder al te veel acht te slaan op zaken als beveiliging. De netwerkmogelijkheden die voor Qemu ontwikkelt werden, houden in veel gevallen dan ook geen rekening met toegangsbeperkingen.

166 Zie : http://en.wikipedia.org/wiki/TUN/TAP

167 Zie : http://en.wikipedia.org/wiki/Osi_layers

168 Bijvoorbeeld door gebruik te maken van het scriptje in voetnoot 87 hierboven.

voorbeeld tap0, tap1, enz.). Aangezien de tap-devices bestaan op de host, kan de virtuele machine via tap0 nu netwerken met die host. Daarbij wordt géén gebruik gemaakt van de fysieke netwerkkaart. Als er vanuit de virtuele machine verkeer dient te gaan naar het netwerk buiten de host (bijvoorbeeld naar het Internet) of omgekeerd, dan moet er gebruikt gemaakt worden van de normale Linux-gereedschappen[169] om tap0 te verbinden met de fysieke netwerkkaart (bijvoorbeeld eth0).

Door gebruik te maken van deze mogelijkheden, kunnen we de virtuele machines plaatsen in één, of meer subnetten die verschillend zijn van het subnet waarin de fysieke netwerkkaart zich bevindt. Als het gebruik van meerdere subnetten wenselijk is, kan elke virtuele machine in een apart subnet geplaatst worden, door telkens een nieuw tap-device aan te maken.

Als we van deze mogelijkheid geen gebruik zouden willen maken, dan moeten we gebruik maken van *kernel bridging*. Hiertoe moeten we, naast de al eerder genoemde drie netwerkkaarten, nog een vierde aanmaken, met name br0. We verbinden daarna eth0 en tap0 met br0 en gebruiken br0 als de netwerkkaart van de host. Om één en ander overzichtelijk te houden, wordt overigens aangeraden br0 te hernoemen naar eth0 en eth0 naar peth0 (overigens precies datgene wat er gebeurt bij *bridged networking* onder Xen). Er zijn nu vier netwerkkaarten, maar we gebruiken feitelijk enkel de virtuele netwerkkaart in de virtuele machine en de bridge (br0 omgenoemd tot eth0). De twee andere kaarten (peth0 en tap0) werken verder op de achtergrond, zonder dat wij er veel van merken.

Door deze aanpak, worden we verlost van het schrijven van een specifieke driver, die anders nodig zou zijn om Ethernet frames te versturen tussen de host en de virtuele machines.

Als we nu meerdere virtuele machines aanmaken, die zich elk in een eigen subnet bevinden, dan kunnen we deze ook onder elkaar laten netwerken door onder Linux gebruik te maken van iptables en routing. Hierdoor kunnen we het doorsturen van pakketten door de host tussen de virtuele machines regelen. Bevinden alle virtuele machines zich samen met de host in één enkel subnet, dan moeten we de verschillende tap-devices enkel verbinden met de bridge.

Omdat Qemu al vrij lang bestaat en er dus al veel mee geëxperimenteerd werd, zijn er zeer veel scenario's inzake netwerkmogelijkheden[170]. De benamingen die in diverse online-handleidingen worden meegegeven, durven nogal eens uiteenlopen, wat het niet altijd makkelijk maakt om één en ander van elkaar te onderscheiden. Wij onderscheiden ondermeer :

- user mode networking of Slirp
- root networking of Tuntap
- host only networking
- bridged networking
- virtual distributed ethernet networking of VDE

We bekijken één en ander wat meer in detail.

169 Bijvoorbeeld /sbin/ip of /sbin/route).
170Zie : http://calamari.reverse-dns.net:980/cgi-bin/moin.cgi/QemuNetwork

5.3.3.1. *Root networking (Tuntap)*

Dit is de oorspronkelijke (en in het begin enige) methode van Qemu om te netwerken. Deze methode veronderstelt zogenaamde *elevated rights* om toegang te kunnen krijgen tot het netwerk. Met andere woorden, daar waar men onder Linux meestal over administrator-rechten dient te beschikken om het netwerk te configureren, was het oorspronkelijk onder Qemu zo dat dergelijke rechten ook nodig waren om het netwerk te kunnen gebruiken. De reden hiervoor was dat het gebruik van een netwerk destijds enkel kon gebeuren doorheen het creëren in het besturingssysteem van de host van een virtuele ethernet-netwerkkaart (met name een zogenaamd *tuntap device*). Dergelijke aanpak brengt dan ook een veiligheidsrisico met zich mee[171]. Qemu is ondertussen flink geëvolueerd en kan op meerdere andere manieren met netwerken omgaan.

5.3.3.2. *User mode networking (Slirp)*[172]

Root networking vereist zoals gezegd zogenaamde *elevated rights*. Met andere woorden, om gebruik te kunnen maken van *root networking* dient telkens een root-paswoord te worden opgegeven. Dit is natuurlijk niet ideaal. Gewone gebruikers zouden ofwel niet kunnen netwerken, ofwel zouden ze moeten beschikken over het root-paswoord, wat niet verkieslijk is (om het zacht uit te drukken). Natuurlijk zouden we hen ook beperkte elevated rights kunnen geven (bijvoorbeeld via het sudoers-script), maar daar heeft de administrator dan weer extra, manueel werk mee.

Om dit probleem op te lossen, werd *user networking* (of slirp[173]) bedacht. Slirp is niks anders dan het softwarepakket dat gebruikt wordt om *user networking* te kunnen toepassen. Voor er sprake was van Internet Service Providers gebruikten (voornamelijk) studenten een inbelmogelijkheid om toegang te krijgen tot een universiteitscomputer. Veel meer dan een simpele shell verkregen ze daarmee niet (vergelijkbaar met telnet). Surfen op het web was er niet bij. Ook eigen netwerkprogramma's konden niet worden gebruikt (enkel deze die op de servermachine aanwezig waren en waarop men gebruiksrechten kon doen gelden). Om deze beperkingen te omzeilen, werd slirp gemaakt.

Slirp verschalkt als het ware een zogenaamde SLIP-verbinding, waarbij SLIP een voorganger is van het nu veel gebruikte PPP-inbelprotocol. Dit werkt als volgt : De gebruiker belt in, geeft zijn gebruikersnaam en paswoord op en start daarna het slirp-programma, waarvoor enkel gewone gebruikersrechten vereist zijn. Dan start de gebruiker vanop zijn thuiscomputer een SLIP-connectie naar het Internet (eigenlijk naar slirp). Van dan af heeft de thuiscomputer toegang het Internet doorheen de inbelconnectie. Aan de andere zijde van de inbelconnectie is het het slirp-programma dat de SLIP-pakketten ontvangt. Slirp is echter een gewoon gebruikersprogramma en kan dan ook niet zelf SLIP-connecties tot stand brengen. Wat slirp doet is feitelijk het converteren van SLIP-pakketten naar zogenaamde *socket calls* en omgekeerd.

171 Als eender welk programma zomaar toegang kon krijgen tot het netwerk, dan zou ook eender welk programma *tuntap devices* kunnen creëren om er eender wat mee te doen. Om dat te vermijden, werd de procedure van de *elevated rights* toegepast.

172 Ook wel *NAT networking* genoemd.

173 De benaming slirp is afkomstig van het softwarepakket dat gebruikt wordt om *user networking* te kunnen toepassen.

Socket calls gebeuren op het niveau van Layer 7 van het OSI-model (de application layer), waarvoor geen *elevated rights* nodig zijn.

Onder Qemu wordt de slirp-code niet gebruikt om SLIP-pakketten naar socket calls en omgekeerd te converteren, maar wel om Ethernet frames om te zetten naar socket calls en omgekeerd. De reden hiervoor is dat Qemu geen modem emuleert (die omgaat met SLIP-pakketten), maar wel een netwerkkaart (die omgaat met Ethernet frames).

Door voor user mode networking gebruik te maken van de slirp-code in Qemu kunnen we dan ook feitelijk aan root networking doen zonder daadwerkelijk root-rechten te bezitten. User mode networking is overigens de default instelling in Qemu. Bijkomend voordeel is dat de huidige implementatie van de slirp-code in Qemu ervoor zorgt dat we feitelijk aan Network Address Translation (NAT) kunnen doen. In de praktijk komt dit erop neer dat alle virtuele machines zich in het subnet 10.0.2.0 zullen bevinden (dit wil zeggen afgescheiden van het subnet waartoe de hostmachine behoort), vanwaaruit zij echter ook het host-subnet kunnen bereiken en daardoor heen ook het Internet.

User mode networking heeft overigens nogal wat nadelen. Zo is er allereerst de overhead die gepaard gaat met het converteren van pakketten naar *socket calls.* Dit zorgt voor een snel oplopend performatieprobleem : het netwerk wordt steeds trager.

Ook is user mode networking beperkt tot gebruik door één enkele Qemu-instantie en dus feitelijk één enkele virtuele machine[174].

Bovendien staat deze vorm van netwerken enkel toe dat TCP-connecties worden gemaakt. Gebruik van ping wordt niet ondersteund, net zomin als UDP-connecties.

Daar staat tegenover dat user mode networking héél makkelijk is om op te zetten én in het gebruik (ook al omdat het Qemu's default is) . Zo bevat user mode networking een ingebouwde DHCP-server die virtuele machines automatisch voorziet van de nodige configuratiegegevens. Hoewel zoals gezegd UDP niet ondersteund wordt, is er toch een geëmuleerde en gepatchde DNS-server aanwezig. Ook kan een geëmuleerde TFTP-server geactiveerd worden. Het hoger genoemde snel optredende performantieverlies doet deze voordelen echter snel teniet (tenzij het netwerk enkel gebruikt zou worden voor websurfen en e-mail lezen).

5.3.3.3. *Host only networking*

Als we geen toegang tot het netwerk buiten de host willen hebben (noch omgekeerd), dan kunnen we ook gebruik maken van host only networking. Waarom zouden we dat echter willen ?

Als we onder Qemu vanuit een virtuele machine een bestand willen sturen naar de hostmachine, dan kunnen we gebruik maken van de netwerkcapaciteiten van Qemu. Daarmee maken we echter als het ware een omweg. We sturen het bestand via de virtuele netwerkkaart in de virtuele machine (veth0), over het tap-device (tap0) naar de bridge (br0) en vandaar via de fysieke netwerkkaart (eth0) naar de host. Omgekeerd dient dezelfde weg gevolgd te worden. Ondertussen kan de fysieke netwerk-

174 Het SLIP-protocol is immers een zogenaamd point-to-point-protocol (feitelijk is SLIP een minder gesofisticeerde, goedkopere versie van het nu wijdverbreide PPP-protocol).

kaart ook ander verkeer te verwerken hebben. Dit alles vertraagt uiteraard de snelheid en vermindert de bandbreedte. Door gebruik te maken van *host only networking* kunnen we deze omweg vermijden.

Om *host only networking* te kunnen gebruiken, moeten we een bijkomend tap-device aanmaken én een bijkomende bridge. Deze dienen we op de gebruikelijke Linux-manier (i.e. door het aanpassen van de desbetreffende configuratie-scripts) te voorzien van de nodige specifieke MAC- en IP-adressen. We maken via bridged networking als het ware een bijkomende, exclusieve verbinding tussen de hostmachine en de virtuele machine, waardoor we rechtstreeks verbinding kunnen leggen tussen beide, zonder gebruik te maken van de omweg langs de fysieke netwerkkaart op de host. Dit wordt soms ook een *private virtual bridge* genoemd. Het is met deze methode ook mogelijk verschillende virtuele machines met elkaar te verbinden, zonder dat hun onderling netwerk zichtbaar is voor derden, noch voor de host.

5.3.3.4. *Public bridged networking*[175]

Met deze methode heeft elke virtuele machine een virtuele netwerkkaart met een eigen MAC- en IP-adres. Daarnaast bevindt er zich op de host, naast tap0, nog een ander virtueel netwerkapparaat, met name een bridge (br0). Deze bridge[176] wordt geconfigureerd tot een virtuele switch. Het IP-adres van die switch is hetzelfde als dat van de fysieke netwerkkaart. De fysieke netwerkkaart (peth0) zelf heeft geen IP-adres meer en treedt in feite op als een ondergeschikte aan de bridge (met andere woorden, de bridge wordt de *master* en de fysieke netwerkkaart de *slave*). De configuratie van dit alles dient te gebeuren door de administrator en om een aantal noodzakelijke bestanden te kunnen laten werken, dienen gebruikers er rechten op te kunnen laten gelden. Er is dus sprake van enige overhead op beheersniveau.

In de praktijk zorgt deze methode ervoor dat elke virtuele machine optreedt als een aparte host op het lokale network, die ook zichtbaar is voor alle andere hosts op dat netwerk (alle hosts behoren tot hetzelfde subnet). De Qemu-handleidingen die op het Internet te vinden zijn, zijn helaas meestal niet meer up-to-date, aangezien één en ander veranderd werd in kernel-versie 2.6.18[177].

Om dit aan de praat te kunnen krijgen, dienen alleszins de bridge-utils geïnstalleerd te zijn. Dit controleren we eerst :

```
[root@testmachien ~]# yum list bridge-utils

Loading "installonlyn" plugin

Setting up repositories

base                      100% |=========================| 1.1 kB    00:00

centosplus                100% |=========================|  951 B    00:00
```

175Dit wordt ook weleens *bridged tap networking* genoemd.

176Een *bridge* wordt geïnstalleerd en geconfigureerd via de *bridge-utils* van Linux en maakt dus geen deel uit van Qemu.

177Deze informatie haalden we uit een recente how-to (van april 2008). Zie http://wiki.centos.org/HowTos/KVM

```
updates                   100% |=========================|  951 B    00:00

extras                    100% |=========================| 1.1 kB    00:00

Reading repository metadata in from local files

Installed Packages

bridge-utils.x86_64                        1.1-2                   installed
```

Vervolgens dienen we het pakket tunctl te installeren. Helaas is dit niet beschikbaar in de repositories van CentOS. We kunnen echter een versie voor Fedora 9 gebruiken. Deze downloaden en installeren we dan ook :

```
[root@testmachien ~]# wget \

http://download.fedora.redhat.com/ \

pub/fedora/linux/development/x86_64/os/Packages/tunctl-1.4-2.fc9.x86_64.rpm

[root@testmachien ~]# rpm -Uvh tunctl-1.4-2.fc9.x86_64 .rpm
```

We moeten nu ook nog het bestand /etc/udev/rules.d/90-kvm-rules aanpassen. Na de regel *KERNEL=="kvm", NAME="%k", GROUP="kvm", MODE="0660"* voegen we de regel *KERNEL=="tun", NAME="%k", GROUP="kvm", MODE="0660"* toe.

Nu moeten we een aantal zaken configureren. Als we willen dat ook gebruikers met gewone rechten de mogelijkheid hebben om virtuele machines te gebruiken, dan moeten we ervoor zorgen dat zij (of hun scripts) toegang hebben tot de nodige commando's. Deze bevinden zich allemaal in de directories /usr/sbin en /sbin. Deze directories voegen we dan ook toe aan het pad.

```
[root@testmachien ~]# PATH=$PATH:/usr/sbin:/sbin
```

Daarna maken we een bridge aan (br0) en wel zodanig dat de bridge als het ware in de plaats treedt van de fysieke netwerkkaart. We maken het IP-adres van de fysieke netwerkkaart dan ook leeg[178].

```
[root@testmachien ~]# brctl addbr br0

[root@testmachien ~]# ifconfig eth0 0.0.0.0

[root@testmachien ~]# brctl addif br0 eth0
```

Vervolgens wijzen we het oorspronkelijke IP-adres van de fysieke netwerkkaart (eth0) toe aan de bridge (br0).

```
[root@testmachien ~]# ifconfig br0 192.168.1.11 netmask 255.255.255.0 up

[root@testmachien ~]# route add -net 192.168.1.0 netmask 255.255.255.0 br0

[root@testmachien ~]# route add default gw 192.168.1.1 br0
```

178Hierdoor verliezen we natuurlijk wel onmiddellijk onze netwerkconnectiviteit !

Met behulp van tunctl voegen we een tap-device toe (tap0) en geven we gebruiker peter permissies voor gebruik daarvan. Dan maken we tap0 actief (zonder toewijzing van een IP-adres) en verbinden we tap0 met br0.

```
[root@testmachien ~]# tunctl -b -u peter

[root@testmachien ~]# ifconfig tap0 up

[root@testmachien ~]# brctl addif br0 tap0
```

De problemen van user mode networking zijn met deze configuratie uit de wereld. Een groot nadeel van deze manier van netwerken onder KVM is echter uiteraard dat de verschillende virtuele machines niet in één of meerdere afgescheiden subnets geplaatst kunnen worden. De volgende methode verhelpt daaraan.

5.3.3.5. *Virtual distributed ethernet networking (VDE)*

Deze methode is oorspronkelijk ontworpen voor gebruik met User Mode Linux, een ander methode om meerdere Linux-systemen tegelijk in user space te laten draaien. VDE moet worden geconfigureerd door de administrator, maar daarna hebben gebruikers die ermee willen werken geen geprivilegieerde rechten meer nodig. VDE zorgt voor volledige netwerkconnectiviteit tussen virtuele machines onderling, met de host én met andere netwerken, zonder ingewikkelde regels voor de firewall en zonder ingewikkelde schema's voor bridging.

VDE is een programma dat vanuit de broncode geïnstalleerd dient te worden[179]. Dat wil dus zeggen dat het eerst gecompileerd dient te worden. Nadat we het programma gedownload hebben naar /tmp/vde/, geven we dan ook volgend commando :

```
[root@testmachien ~]# cd /tmp/vde/

[root@testmachien tmp]# make & make install
```

In de directory /usr/local/bin/ dient zich nu vde_switch te bevinden, wat we controleren[180] :

```
[root@testmachien ~]# ls /usr/local/bin/vde_switch

vde_switch
```

VDE is nu geïnstalleerd, maar Qemu kan daar verder niks mee. Om daaraan te verhelpen, hebben we een hulpprogramma nodig, een zogenaamde *utility*. Die utility vinden we in de subdirectory /tmp/vde/qemu/. We gaan in deze directory en geven het commando **make**. Hierdoor wordt **vdeq** gecreëerd, het hulpprogramma dat we nodig hebben, wat we vervolgens manueel kopiëren naar /usr/bin/[181]. Wat **vdeq** doet, is eigenlijk het opzetten en configureren van VDE tot een speciaal tun-device, dat Qemu kan gebruiken.

179Zie : http://sourceforge.net/projects/vde/

180Volgens de documentatie van VDE zou vde_switch zich moeten bevinden in de directory /usr/bin/, maar in onze setup klopt dat niet.

181Er is dus geen make install commando nodig.

Nu dit alles gebeurt is, kunnen we VDE gebruiken als interface om te komen tot een verbeterde vorm van user mode networking (i.e zonder de reeds genoemde nadelen van user mode networking). We zorgen er eerst en vooral als root voor dat vde_switch in de achtergrond draait (als daemon of service, dus) en gebruikt maakt van het tap-device tap0 :

```
[root@testmachien ~]# vde_switch -tap tap0 -daemon
```

Vervolgens geven we tap0 een IP-adres, dat door de virtuele machines gebruikt gaat worden als default gateway :

```
[root@testmachien ~]# ifconfig tap0 192.168.254.254 netmask 255.255.255.0
```

De virtuele machines zullen in dit geval allemaal moeten behoren tot het subnet 192.168.254.0 met als subnetmask 255.255.255.0. Het tap-device tap0 treedt overigens op als router tussen de virtuele machines en de fysieke netwerkkaart op de host (eth0) en is dus inderdaad een echte default gateway.

Deze handelingen dienen te gebeuren voor de firewall wordt gestart (anders zou het niet lukken). Zonder firewall blijven werken is natuurlijk niet aan te raden. We gaan dan nu ook de firewall configureren.

In concreto dient masquerading mogelijk gemaakt te worden tussen tap0 en het lokale netwerk waartoe de fysieke netwerkkaart behoort (eth0). Dat doen we als volgt :

```
[root@testmachien ~]# echo '1' > /proc/sys/net/ipv4/ip_forward

[root@testmachien ~]# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

We hebben nu het tap-device tap0 opgezet als router die gebruikt maakt van Network Address Translation (NAT), waardoor de verschillende virtuele machines kunnen beschikken over volledige netwerkfunctionaliteit (TCP, UDP, ping, enz.). Tevens kan elke gebruiker die netwerkfunctionaliteit gebruiken, zonder beroep te moeten doen op geprivilegieerde rechten. Ook kunnen alle virtuele machines elkaar zien op het netwerk, zonder ingewikkelde beslissingstabellen nodig te hebben.

Gebruik van VDE maakt dan ook komaf met de nadelen van zogoed als alle andere netwerkmogelijkheden onder Qemu. We zullen in dit werk dan ook van VDE gebruik maken.

5.3.4. Virtuele harde schijven onder KVM/Qemu

Net zoals dat geldt voor de netwerkfunctionaliteit, profiteert KVM van de reeds met Qemu opgedane ervaring op het vlak van virtual block devices, waaronder het omgaan met virtuele harde schijven valt. Qemu ondersteunt verschillende formaten van virtuele harde schijven.

Zo is er eerst en vooral de default : het raw-formaat. Dergelijke rauwe schijf is eenvoudig aan te maken en kan gebruik maken van de mogelijkheden van het onderliggende bestandssysteem om ruimte te besparen[182]. Het is ook relatief makkelijk te exporteren naar andere emulators.

182 Bijvoorbeeld op bestandssystemen als ext2 en ext3 onder Linux en NTFS onder Windows, die zogenaamde 'gaten' of *holes* ondersteunen.

Vervolgens is er het zogenaamde quick copy on write (qcow) formaat. Dit bestaat in een oudere versie (die nauwelijks nog gebruikt wordt, maar om reden van *backward compatibility* nog ondersteunt wordt) en het daaruit gegroeide qcow2-formaat. Dit formaat kan ook worden gebruikt op bestandssystemen die de functionaliteit van ext2/3 of NTFS ontberen (zoals bijvoorbeeld FAT). Virtuele harde schijven in qcow2-formaat zijn ook kleiner, ze kunnen geëncrypteerd en gecompresseerd worden en er kunnen makkelijk snapshots van genomen worden.

Daarnaast kan Qemu ook omgaan met het vmdk-formaat van VMWare. Hiermee wordt dan ook voorzien in een belangrijke mate van interoperabiliteit.

Verder kent Qemu nog het cow-formaat (copy on write) dat afkomstig is van User Mode Linux en dat vroeger het enige formaat was dat kon voorzien in groeiende virtuele schijven. Dit wordt enkel ondersteund om reden backward compatibility.

Tenslotte is er ook nog het cloop-formaat, dat gebruikt kan worden voor gebruik met gecompresseerde CD-ROM's (zoals bijvoorbeeld de bekende Knoppix-CD's).

Om virtuele harde schijven aan te maken, gebruiken we bijvoorbeeld volgend commando :

```
[root@testmachien ~]# qemu-img create -f qcow2 /vdisks/vdisk.img 10G
```

Hiermee wordt de virtuele schijf vdisk.img met een grootte van 10 gigabyte aangemaakt in het formaat *quick-copy-on-write-2*. Andere optionele parameters zijn onder andere -c (om te compresseren) en -e (om te encrypteren). Overigens, stel dat we KVM willen gebruiken zonder Qemu, dan kunnen we dergelijke virtuele harde schijf ook creëren met het gewone Linux-commando **dd** en wel als volgt :

```
[root@testmachien ~]# dd if=/dev/zero \

of=/vdisks/vdisk.img bs=1G count=10
```

Gebruik van Qemu levert nogal eens problemen op bij gebruik van een muis. Omdat te voorkomen, geven we het volgende commando :

```
[root@testmachien ~]# export SDL_VIDEO_X11_DGAMOUSE=0
```

Om een virtuele machine aan te maken, kunnen we een commando met volgende syntax gebruiken :

```
[root@testmachien ~]# qemu-kvm \

-hda vdisk.img -cdrom distro.iso -m 512 -boot d
```

De hier meegegeven parameters betekenen het volgende :

- hda : het soort harddisk dat we willen emuleren (in dit geval een IDE-schijf)
- vdisk.img : de (arbitraire) naam die we meegeven aan onze virtuele harddisk
- cdrom : het feit dat we gebruik willen maken van een (geëmuleerde) cdrom
- distro.iso : het ISO-bestand dat we gebruiken als installatiemedium
- m 1024 : de te gebruiken hoeveelheid geheugen in megabytes

– boot d : het opstartmedium (c=harddisk, d=cdrom)

Als we dit commando uitvoeren start onze virtuele machine op vanaf de geëmuleerde CD-ROM en kunnen we onze Linux-distributie installeren op onze virtuele harde schijf. Merk op dat het qemu-kvm óók gebruikt moet worden, indien we Qemu niet zouden gebruiken. Als de installatie voltooid is, kunnen we ons virtueel systeem opstarten met volgend commando :

```
[root@testmachien ~]# qemu-kvm -hda vdisk.img
```

Naast het omgaan met virtual block devices (zoals virtuele harde schijven), kan Qemu ook overweg met zogenaamde host drives. Dat betekent dat Qemu ook rechtstreeks kan werken met de bestandsnamen van hardware-devices. Zo kan Qemu onder Linux bijvoorbeeld werken met de CD-ROM als /dev/cdrom of /dev/scd0. Hetzelfde geldt voor het gebruik van de reële harde schijf. Voor ons is dit interessant omdat we ons hostsysteem hebben ingericht met LVM. We kunnen Qemu dus rechtstreeks logical volumes laten aanspreken, waarbinnen we onze virtuele machines zullen opzetten. Net zoals onder Xen en OpenVZ kunnen we zo de voordelen der virtualisering combineren met de flexibiliteit van het Logical Volume Management.

5.3.5. Virtuele machines installeren onder KVM/Qemu

We weten nu genoeg om te kunnen overgaan tot het aanmaken van virtuele machines onder KVM/Qemu. Tenminste, dat denken we. Helaas, zoals wel meer gebeurt, spelen de subtiele verschillen tussen de Linux-distributies ons behoorlijk wat parten. Nadat we in een terminal de nodige commando's ingaven, kregen we volgende foutmelding :

```
/etc/qemu-ifup: could not launch network script

Could not initialize device 'tap'
```

Na lang zoeken vonden we op het Internet een nuttige how-to[183]. Daarin lezen we dat het script qemu-ifup nodig is, maar om één of andere reden niet werkt. We kunnen kvm-qemu echter de opdracht geven dit script niet te gebruiken. Dat doen we door in het opstartbestand van de virtuele machine de opdracht script=no toe te voegen. Laten we nu een eerste virtuele machine aanmaken :

```
[root@testmachien ~]$ kvm-qemu \

-net vde,vlan=0 \

-net nic,vlan=0,macaddr= 52:54:00:00:EE:02 \

-hda file=/vms/vdisk.img \

-cdrom file=/dev/scd0 \

-boot d \
```

183Zie : http://wiki.centos.org/HowTos/KVM

```
-m 1024 \

-pidfile 00VM.pid \

-name 00VM \

-k nl-be
```

Deze virtuele machine heeft als naam 00VM, gebruikt het logical volume met dezelfde naam (LV_00VM) als virtuele harde schijf en de cdrom-speler van de host vanwaarop hij ook gaat opstarten, werkt met 1 gigabyte als werkgeheugen, schrijft zijn program identifier weg in het bestand 00VM.pid, gebruikt een Belgisch toetsenbord en VDE-netwerkfaciliteiten[184].

5.3.6. Beveiliging onder KVM/Qemu

De hoofdontwikkelaar van KVM (i.e. Avi Kivity) is van mening dat beveiliging eigenlijk niet de zaak van KVM is. Deze verontrustende stelling steunt op zijn overtuiging dat het de besturingssystemen van de virtuele machines zijn (en hun administrators) die dienen in te staan voor de beveiliging. Bovendien vindt hij dat KVM verder bouwt op Linux en bijgevolg dat eenvoudigweg gebruik gemaakt kan worden van de beveiligingsmechanismen die nu reeds in Linux ingebouwd zijn. Niet iedereen is het daar echter mee eens. Zo stelde Hadi Nahari van het bedrijf Montavista (gespecialiseerd in *embedded* toepassingen van Linux) op het KVM Forum in 2007 dat KVM mechanismen voor *mandatory access control* en voor het veilig isoleren van virtuele machines mist[185]. Het is bovendien ook zo dat élke *hypervisor* (dus ook degene die als een kernel-module wordt geïmplementeerd) op zichzelf een mogelijk doelwit is voor zogenaamde *exploits*. Bijgevolg is daarvoor ook een specifieke beveiligingsaanpak wenselijk. Het laatste woord is hier duidelijk nog niet over gezegd. Toch is het evenzeer duidelijk dat KVM ondertussen toch wel zwak staat op het punt van beveiliging.

5.3.7. Beheer onder KVM

KVM kan uiteraard volledig beheert worden via de command-line. Omdat KVM feitelijk slechts een module is voor de Linux-kernel kunnen daarvoor, naast de specifieke kvm-qemu commando's, ook de gewone Linux-commando's goed van pas komen. Er is uiteraard wel wat kennis van die commando's nodig.

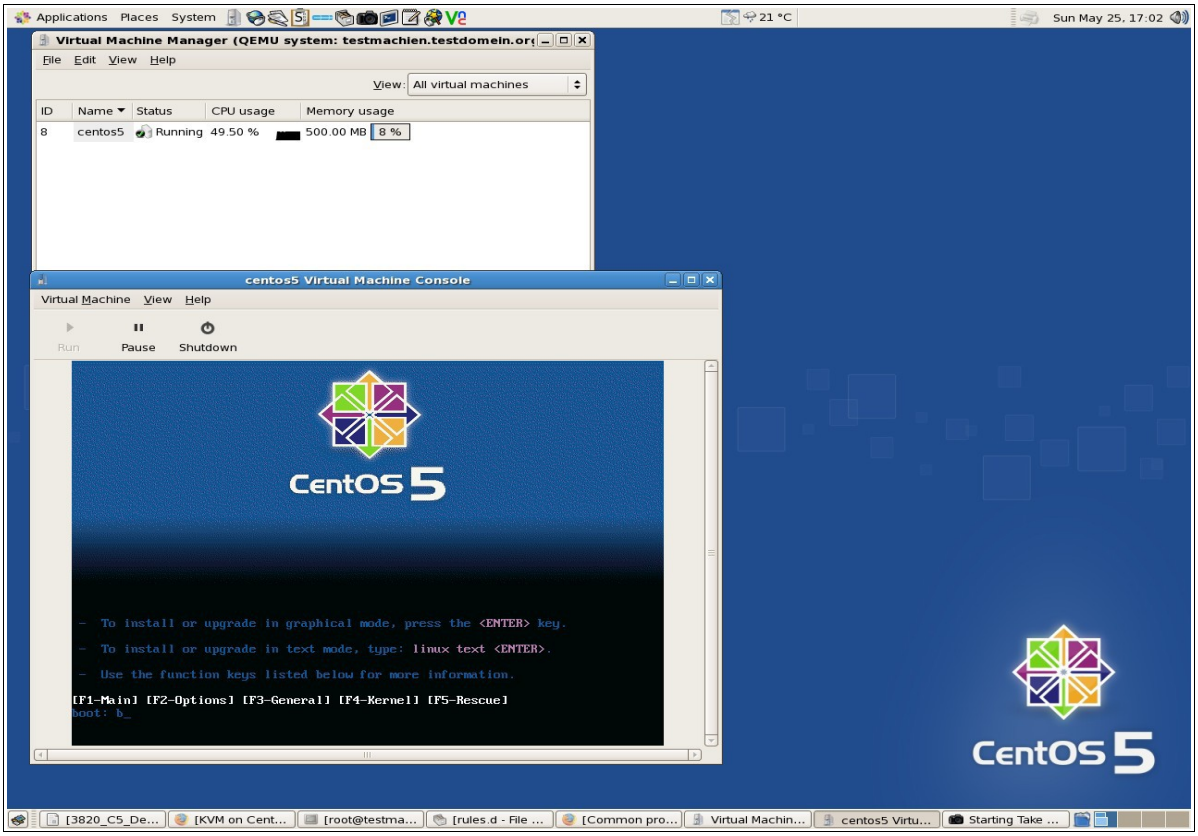
Naast de command line tools zijn er grafische interfaces beschikbaar. De meeste daarvan bevinden zich echter in een onafgewerkte staat (dit wil zeggen : ze zijn nog volop in ontwikkeling). De reden waarom deze ontwikkeling in volle gang is, heeft te maken met de virtualiseringsbibliotheek libvirt. Eerder dan dat elke programmeur zelf een volledige, functionele interface zou gaan schrijven, wordt er gewerkt aan een bibliotheek die in staat is een soort 'deur' te zijn naar alle virtualiseringsoplossingen toe. Eens deze bibliotheek voldoende gerijpt is, kunnen allerlei grafische tools de

184 Waarbij uiteraard verondersteld wordt dat het logical volume, evenzogoed als de configuratie van VDE reeds eerder door de administrator werden geregeld.

185 Zie : http://www.osadl.org/Single-View.111+M5382caa6fee.0.html

'deur' aanspreken en zo binnenstappen in de werelden van Xen, OpenVZ, KVM én in die van VMWare, Parallels, enz.

Momenteel is alleen virt-manager (dat we nog kennen van gebruik met Xen) voldoende vergevorderd om er effectief onder KVM mee aan de slag te kunnen, getuige onderstaande schermafdruk :



Afbeelding 37 : virt-manager onder KVM/Qemu

De meeste andere grafische omgevingen om virtuele machines mee te beheren hebben ofwel ondersteuning voor KVM in experimentele vorm, ofwel kondigen ze deze ondersteuning aan voor de nabije toekomst. Het gaat dan om projecten zoals openQRM en Enomalism.

Enomalism bijvoorbeeld is beschikbaar als zogenaamde alpha-release en veronderstelt daarbij de installatie van verschillende pakketten die reeds aanwezige pakketten op ons CentOS-systeem moeten vervangen. Het valt te vrezen dat er daarbij heel wat mis kan gaan, dus we laten het voorlopig maar zo[186].

OpenQRM is een web-gebaseerd administratiesysteem voor virtuele omgevingen (net zoals Enomalism overigens), waarbij plugins gebruikt worden voor verschillende, meer specifieke taken. Er is ook zo'n plugin verkrijgbaar voor Qemu, maar die is al

186Wie toch geïnteresseerd genoeg is om het onmiddellijk uit te proberen, kan terecht bij volgende installatiegids : http://trac.enomalism.com/enomalism/wiki/enomalism-install-rh

meer dan een jaar uit en dient eigenlijk voor oudere versies van de Red Hat Linux-derivaten. We wachten ook hier liever op een meer recente versie.

5.3.7. KVM/Qemu – Besluit

KVM/Qemu is een zogenaamd *light weight* virtualiseringssysteem. Het is relatief eenvoudig om op te zetten en je kunt er snel mee aan de slag. Er zijn – behoudens het opzetten van een geschikte netwerkomgeving – geen moeilijke vragen (zoals al dan niet paravirtualiseren) om op te lossen. Naar onze mening is het daarom ook niet verwonderlijk dat KVM/Qemu nogal populair is als virtualiseringsoplossing voor de *desktop*. Het staat inderdaad toe om snel verschillende besturingssystemen uit te testen.

Een nadeel van KVM/Qemu is dat de documentatie ervoor nogal verspreid is en elkaar helaas ook dikwijls tegenspreekt. Dat komt voornamelijk omdat KVM soms al te makkelijk leunt op de eerder bijeengeschreven documentatie voor Qemu, die echter in een aantal gevallen flink voorbijgestreefd is. Het gevolg hiervan is dat gebruik van KVM/Qemu in een productieomgeving nogal wat extra werk voor de administrator met zich mee brengt. Daarbij moeten we dan niet zozeer denken aan de installatie ervan, dan wel aan het onderhoud. Troubleshooting kan onder KVM/Qemu erg tijdrovend zijn.

Tenslotte vormt beveiliging bij KVM/Qemu een uitermate zwak punt. Omdat de ontwikkelaars er weinig tot geen belang aan hechten (onder het mom dat beveiliging nu eenmaal de verantwoordelijkheid is van de administrator en niet van de ontwikkelaar), zijn er ook geen ingebouwde beveiligingsmechanismen. Zo is toegang tot de hypervizor KVM niet beperkt tot de rout user en kan de beperking tot de root-user voor toegang tot de netwerkconfiguratie van Qemu eenvoudigweg worden omzeild. Dat lijken ons voor virtualisering op serverniveau toch wel erg bedenkelijke zaken.

6. Samenvatting, aanbeveling en besluit

We hebben nu de drie mogelijke oplossingen voor het virtualiseringsprobleem van onze vzw uitgebreid onderzocht. Het is daarbij niet zo dat we alle hoekjes en kantjes van deze drie virtualiseringsoplossingen van naderbij hebben bekeken. Zo hebben we bijvoorbeeld het migreren van fysieke naar virtuele systemen niet behandeld, net zomin als het migreren van virtuele machine van de ene hostserver naar een andere. We hebben echter wel die zaken behandeld die direct nuttig kunnen zijn vanuit het standpunt van onze vzw.

We kunnen nu de voor- en nadelen van de drie virtualiseringsoplossingen met elkaar gaan vergelijken. Daartoe hebben we een puntensysteem ontworpen. Dit is uiteraard geen wetenschappelijke methode, aangezien ze enkel gebaseerd is op onze eigen ervaringen. Toch kan het nuttig zijn om zo een overzicht te krijgen. Concreet beoordelen we de drie virtualiseringssystemen op een aantal onderdelen, waarbij elk onderdeel een score van 1 tot 5 krijgt toebedeeld. Het getal 1 staat daarbij voor een zeer slechte score, 5 voor een zeer goede. Dat geeft dan volgende resultaten :

Criterium	Xen	OpenVZ	KVM
Beheertools	3	2	2
Beveiligingsmechanismen	3	3	1
Documentatie	3	3	2
Linux-kennis vereist bij installatie	1	3	2
Moeilijkheidsgraad	1	3	2
Netwerkmogelijkheden	4	2	5
Performantie	4	5	4
Stabiliteit	5	5	3
Veelzijdigheid	5	1	2
Linux-kennis vereist na installatie	1	4	1
Totaal	30	31	24

Tabel 6 : Vergelijkend arbitrair overzicht voor- en nadelen virtualiseringsmethodes

Een beetje toelichting bij de gekozen criteria lijkt wel nodig.

Onder 'Beheertools' geven we een score voor het al dan niet beschikbaar zijn van dergelijke tools. We hebben het dan wel over grafische tools, aangezien de mensen van onze vzw niet gewend zijn om te werken met de command line. Echt hoge scores zijn er niet. Dat heeft er natuurlijk mee te maken dat deze grafische tools in volle ontwikkeling zijn. Aangezien echter de virtualiseringsmethodes zelf ook in volle ontwikkeling zijn, treedt er daarbij extra vertraging op.

Onder 'Beveiligingsmechanismen' geven we een beoordeling over de met de betrokken virtualisatiemethode meegeleverde óf ervoor beschikbare beveiligingsmechanismen. Xen scoort hier gemiddeld, niet omdat er onvoldoende beveiligingsmechanis-

men met Xen meegeleverd worden, maar omdat de implementatie ervan niet direct transparant is. De officiële handleiding van Xen is omtrent dit belangrijke onderwerp vrij rudimentair en de op het Internet beschikbare beschrijvingen blinken niet direct uit in duidelijkheid. OpenVZ lijdt aan hetzelfde euvel. De handleidingen zijn ook hier beperkt in omvang en erg rudimentair. Bij KVM/Qemu is de score erg laag, omdat er gewoon zo goed als geen aandacht is voor beveiliging. Eigenlijk wordt er bij alle drie de virtualisatiemethodes vanuit gegaan dat het de Linux-administrator is die moet zorgen voor de beveiliging. Die moet er dan maar zijn plan mee trekken.

Onder 'Documentatie' beoordelen we het al dan niet aanwezig zijn van goede, zo volledig mogelijke en betrouwbare documentatie. Ook daar geen hoge scores. De reden daarvoor is dezelfde als voor de 'Beheertools' : snelle ontwikkeling, met achterlopende documentering.

Onder 'Lenix-kennis vereist bij installatie' kijken we naar de moeilijkheidsgraad op het ogenblik van installatie van de virtualiseringsoplossingen. Hier scoren alle oplossingen ronduit slecht. Ze vereisen allemaal een hoge mate van kennis van het Linux-besturingssysteem. Die noodzaak wordt nog onderstreept door de in onvoldoende mate ontwikkelde documentatie. Dat OpenVZ hier een klein beetje beter scoort dan de beide andere heeft te maken met het feit dat de ontwikkelgemeenschap rond OpenVZ gewoon veel kleiner is, zodat er minder uiteenlopende patches voor voorhanden zijn. Xen doet het hier uitzonderlijk slecht, omdat de Xen-ontwikkelaars er alles aan doen om hun methode opgenomen te zien worden in de Linux-kernel zelf. Dat is voorlopig echter nog niet gebeurt. Als het ooit toch zover zou komen, dan zou dat wellicht een heleboel kunnen vergemakkelijken aan moeilijkheden bij de installatie van Xen. KVM scoort iets minder slecht, wat te wijten is aan het grote talent van de hoofdontwikkelaar van KVM, die op een uitzonderlijk slimme manier weet in te spelen op de technische, maar ook de sociaal-menselijke kenmerken van de Linux-kernel-ontwikkelaars. Daardoor werd zijn oplossing wel reeds opgenomen in de Linux-kernel, wat de installatie ervan inderdaad vergemakkelijkt.

Onder 'Moeilijkheidsgraad' beoordelen we dan de mate waarin het werken met een virtualiseringsoplossing al dan niet moeilijk is (1 is zeer moeilijk, 5 is relatief makkelijk). Geen van de bekeken systemen is echt makkelijk te noemen (ook niet relatief). Dat heeft voornamelijk te maken met het gebrek aan (grafische) beheertools. Omdat OpenVZ van alle drie de systemen het meest transparant (want weinig ingewikkeld) is, haalt dit systeem hier de hoogste score.

Onder 'Netwerkmogelijkheden' bekijken we vooral de keuzemogelijkheden inzake netwerkfunctionaliteit. Hier speelt het feit dat KVM sterk leunt op de opgedane ervaring met het Qemu-project duidelijk in haar voordeel. Ook Xen scoort niet slecht. Moest de mogelijkheid om een netwerk te verbergen voor een hostsysteem stabieler en vooral makkelijker te realiseren zijn, dan zou Xen hier nog hoger kunnen scoren. OpenVZ beschikt dan weer precies over de netwerkfunctionaliteit die nodig is voor deze vorm van virtualisering.

Onder 'Performantie' beoordelen we de snelheid van de virtuele machines. Daar hebben we eigenlijk in geen van alle drie de gevallen echt klachten over. Onder alle drie de virtualiseringssystemen is de performantie van de virtuele machines acceptabel. De enige echte *bottleneck* die kan optreden, heeft niets te zien met het al dan niet virtualiseren. Het gaat met name om de doorgangssnelheid van het netwerk. De snel-

heid daarvan is afhankelijk van allerlei zaken, zoals de capaciteit van de netwerkkaarten, de kabels, de switches en/of routers, enz.

Onder 'Stabiliteit' kijken we naar het gebeurlijk uitvallen of crashen van virtuele machines én van het hostsysteem. Wat dat betreft zitten alle oplossingen goed. Het hostssysteem crashte niet één keer en de virtuele machines enkel wanneer ze (door ons) slecht geconfigureerd werden.

Onder 'Veelzijdigheid' vatten we de prestaties samen inzake de mogelijkheid om om te gaan met meerdere, verschillende besturingssystemen. Hier scoort OpenVZ zeer slecht, wat niet meer dan logisch is : het virtualiseren van het besturingssysteem (in plaats van het werken met of als *hypervisor*) brengt nu eenmaal met zich dat ook enkel besturingssystemen van hetzelfde type als het hostssysteem kunnen gevirtualiseerd worden. KVM loopt hier achter op Xen, omdat KVM vooralsnog niet kan omgaan met geparavirtualiseerde systemen. Daar wordt echter stevig aan gewerkt.

Onder 'Linux-kennis vereist na installatie' vragen we ons af of er met een geïnstalleerde én geconfigureerde virtualiseringsoplossing gewerkt kan worden zonder grote kennis van Linux. Hier blijft Xen erg slecht scoren, terwijl OpenVZ het erg goed doet.

In totaal bekeken liggen de drie virtualiseringsoplossingen erg dicht bij elkaar. Wat de ene als voordeel heeft, wordt dan ook weer gecompenseerd door de voordelen van de andere op een ander vlak.

Toch moeten we een keuze maken. Daarbij dienen we uiteraard uit te gaan van de noden van onze vzw. We hebben hierboven, onder punt 4, gesproken over de wenselijkheid een mini-datacentrum te creëren voor onze vzw. Hiertoe hebben we (virtuele) servers nodig, die kunnen worden aangesproken, hetzij door *thin clients* op het lokale netwerk, hetzij via *remote access*. De meest transparante manier om *remote access* te organiseren, lijkt ons het voorzien in webtoegankelijkheid van onze virtuele servers. Via hun browsers (en met gebruik van het SSH-protocol) kunnen de leden van de vzw toegang krijgen tot de *groupmail*-server (bijvoorbeeld om hun mail of kalender te raadplegen). Hetzelfde geldt voor de vzw-leden die zich bezighouden met het ledenbeheer. Zij kunnen via hun webbrowser met een beveiligde verbinding de CRM-server raadplegen en, daar waar nodig, wijzigingen aanbrengen. Idem voor wat betreft diegenen die instaan voor de infrastructuur van de vzw. Zij kunnen dit vanop afstand perfect beheren door de ERP-server te benaderen via een https-verbinding. Zelfs de backup-server kan op deze manier door de adminstrators benadert worden.

Natuurlijk zou een dergelijk serversysteem ook opgezet kunnen worden door gebruik te maken van een hostingbedrijf (waarbij overigens ook gewerkt wordt met virtualisering). De vzw houdt de zaken echter liever thuis. Naar mijn mening is OpenVZ het meest geschikt om deze aanpak te ondersteunen. Het is trouwens zo dat OpenVZ ook nu al veel gebruikt wordt door dergelijke hostingbedrijven (net als Xen).

Men kan zich echter de vraag stellen of dit wel zo'n goed idee is in het licht van de noden van de print- en VDI-server. Naar mijn gevoel is dit geen probleem. OpenVZ is niet alleen goed in het voorzien in servers voor webtoegang. Een OpenVZ-container kan net zo goed gebruikt worden als TFTP-server om te voorzien in virtuele desktops of als samba-server om bestanden beschikbaar te stellen aan Windows- én Linux-gebruikers.

Onze voorkeur gaat dan ook uit naar OpenVZ. Wel is het slechts een lichte voorkeur. Toekomstige ontwikkelingen kunnen de balans snel doen doorslaan ten voordele van een andere oplossing. Maar voorlopig blijven we toch bij OpenVZ.

Tenslotte is het ook zo dat we een slag om de arm kunnen houden. Gezien KVM immers vooralsnog niks anders is dan een module voor gebruik met *een* Linux-kernel, betekent dit ook dat KVM kan geïnstalleerd worden als module voor een OpenVZ-kernel. Met andere woorden, indien nodig, kan KVM alsnog gecombineerd worden met OpenVZ. Wel moet er dan zorgvuldig gekeken worden naar de netwerkconfiguratie voor de virtuele machines onder KVM.

Lijst van afbeeldingen

Afbeelding 1 : Eenvoudig overzicht der vzw-structuur 18
Afbeelding 2 : Schema formele structuur der vzw 19
Afbeelding 3 : Schema praktische structuur der vzw 21
Afbeelding 4 : Schema van de workflow 'briefwisseling' 23
Afbeelding 5 : Inrichting van de lokalen der vzw 34
Afbeelding 6 : Kabels in de secretariaatsruimte 35
Afbeelding 7 : Schema van het aanwezige netwerk 36
Afbeelding 8 : Het voorstel van een nieuw netwerk 42
Afbeelding 9 : Het vernieuwde netwerk voor de vzw 46
Afbeelding 10 : Het ring-model van de Intel x86-computer 47
Afbeelding 11 : Virtualisering met een hypervisor als Xen 67
Afbeelding 12 : Bridged netwerk onder Xen 70
Afbeelding 13 : Gerouteerd netwerk onder Xen 71
Afbeelding 14 : VLAN met NAT onder Xen 72
Afbeelding 15 : Een nieuw logical volume aanmaken 79
Afbeelding 16 : De virt-manager 80
Afbeelding 17 : Een virtuele machine maken met virt-manager 80
Afbeelding 18 : Een (genummerde) naam voor de virtuele machine 81
Afbeelding 19 : Een virtualisatiemethode kiezen 82
Afbeelding 20 : De installatiemedia kiezen 83
Afbeelding 21 : Een virtuele harde schijf aanmaken in een logical volume 83
Afbeelding 22 : Geheugen en processor toewijzen 84
Afbeelding 23 : Een paswoord opgeven voor de keyring-manager 84
Afbeelding 24 : Een eerste virtuele machine 86
Afbeelding 25 : De grafische services-manager 89
Afbeelding 26 : de pciback-module met de aan haar gebonden netwerk-drivers 91
Afbeelding 27 : Eén verborgen netwerkkaart toegewezen aan DomU1 95
Afbeelding 28 : Een virtuele machine 101
Afbeelding 29 : XenMan in actie 103
Afbeelding 30 : Het dashboard van ConVirt 104
Afbeelding 31 : openQRM 105

Afbeelding 32 : Enomalism 106
Afbeelding 33 : Operating system virtualization 107
Afbeelding 34 : EasyVZ als werk in opbouw 122
Afbeelding 35 : Een virtuele machine aanmaken met ProxMox VE 123
Afbeelding 36 : Kernel-based Virtualization 125
Afbeelding 37 : virt-manager onder KVM/Qemu 138

Lijst van tabellen

Tabel 1 : Soorten printers en hun eigenschappen 40
Tabel 2 : Overzicht software-installatie bij de vzw 161
Tabel 3 : Overzicht workflows en benodigde servers 45
Tabel 4 : Partitionering met RAID-1 en Logical Volume Management 54
Tabel 5 : De indeling met logical volumes van VG_SYSTEM 55
Tabel 6 : Vergelijkend arbitrair overzicht virtualiseringsmethodes 141
Tabel 7 : Implementatieplan 177

Bibliografie

Boeken

[1] GABOR, Andrea, *The Capitalist Philosophers : The Geniuses of Modern Business – Their Lives, Times and Ideas*, 2000, Random House Time Business Books, New York/Toronto.

[2] SMITH, Adam, 1776, *The Wealth of Nations*, Penguin Classics, London, United Kingdom.

[3] RAYMOND, Eric S., *The Cathedral and the Bazaar – Musings on Linux and Open Source by an Accidental Revolutionary*, 2001, O'Reilley Books, Sebastopol, California.

[4] WOLF Chris, HALTER Erick M., *Virtualization, From the Desktop to the Enterprise*, 2005 , Apress, New York.

[5] CHAGANTI Prabakhar, *Xen Virtualization – A Fast and Practical Guide to Supporting Multiple Operating Systems with the Xen Hypervisor*, 2007 , Packt Publishing Ltd.

[6] BUYTAERT Kris et al., *Best Damn Server Virtualization Book Period*, 2007, Syngress Publishing, Burlington, Massachusetts.

[7] BALL, Bill, DUFF Hoyt, *Red Hat Linux and Fedora Unleashed*, 2004, Sams Publishing, Indianapolis, Indiana.

[8] FOX Tammy, *Red Hat Enterprise Linux Administration Unleashed*, 2007, Sams Publishing, Indianapolis, Indiana.

[9] STUTZ Michael, *The Linux Cookbook – Tips and Techniques for Everyday Use*, 2001, No Starch Press, San Francisco, California.

[10] VAN VUGT Sander, *Een Linux-server Inrichten*, 2007, Van Duuren Media, Culemborg, Nederland.

[11] SMITH Roderick W., *Linux Administrator Street Smarts – A Real World Guide to Linux Certification Skills*, 2007, Wiley Publishing, Indianapolis, Indiana.

Artikelen

[12] VARIAN, Melinda, *'VM and the VM Community : Past, Present and Future'*, August 1997, paper presented at SHARE 89, Sessions 9559-9061(te vinden op : http://www.princeton.edu/~melinda/25paper.pdf).

[13] CILIENDO E., TUNIMASA T., *'Linux Performance and Tuning Guidelines'*, 2007, IBM Redbooks, IBM International Technical Support Organisation, free download from http://www.redbooks.ibm. com/abstracts/redp4285.html - registratie is wel nodig !

[14] KEMP J., *'Who's There ? - Remote access security with single-packet port knock ing'*, in Linux Magazine, June 2008, British Edition.

[15] ZELLER, Thomas in *LinuxLife, ein Sonderheft der PC Magazin DVD-Ausgabe*, nr. 3/2008, pagina 16 : een bespreking van BAUER, T., *'OpenVZ, Das Kleine Hand buch'*, 2008, Books On Demand, Duitsland.

[16] LOSHWITZ Martin, FEILNER Markus, *Xensational – Getting Started with Xen Virtualization*, in Linux Magazine, May 2008, British Edition.

[17] DOLLE Wilhelm, WEGENER Christoph, *Virtual Malware – Virtualizing Rootkits and the Future of System Security*, in Linux Magazine, May 2008, British Edi tion.

[18] SCHERF Thorsten, *No Access ! - Mandatory Access Control (MAC) with SELinux*, Linux Magazine, June 2008, British Edition.

[19] SHAH Amit, *Deep Virtue – Kernel-based virtualization with KVM*, in Linux Magazine, January 2008, British Edition.

[20] TUINDER, Olaf, *Logical Volume Management – Logisch Schuiven met disk-space*, in Linux Magazine, Juli 2006, Nederlandse uitgave.

[21] VOS, Jos, *Xen – Open Source Virtuele Machines, in Linux Magazine*, Juli 2006, Nederlandse uitgave.

[22] HUDSON, Paul, *Virtual Smackdown (on hardware virtualization)*, in Linux Format, October 2006.

[23] SCHÜRMANN, Tim, *Die Puppe in der Puppe – Virtualisierung und Emulation*, in LinuxUser, juni 2007, Duitsland.

[24] HABIB, Irfan, *Xen*, in Linux Journal, May 2006, US Edition.

[25] BARTHOLOMEW, Daniel, QEMU – a Multihost, Multitarget Emulator, in Linux Journal, May 2006, US Edition.

[26] HOSKINS, Matthew E., User-Mode Linux, in in Linux Journal, May 2006, US Edi tion.

Websites

[27] Procesmanagement voor architectuurstudenten : http://www.lib.umd.edu

[28] Definities inzake business proces management : http://www.bpmenterprise.com

[29] Definities inzake empowerment :

http://www.bpmenterprise.com/dictionary/Empowerment-225.htm

[30] Managementterminologie :
http://www.bpmenterprise.com/dictionary/glossary.asp

[31] Website van het 'Vlaams Studie- en Documentatiecentrum voor V.Z.W.'s' :

http://www.vsdc.be

[32] Op notarissen gerichte website over vzw's : http://www.notare.be/vzw.htm

[33] Meer algemene website over V.Z.W.'s : http://www.devzw.be

[34] Betekenis van het begrip ideologie : http://www.vandale.nl/opzoeken/woordenboek/?zoekwoord=ideologie

[35] Omtrent de figuur van prof. Ernest Mandel : http://nl.wikipedia.org/wiki/Ernest_Mandel

[36] Beschrijving van de Wireless MAXg ADSL Gateway van US Robotics : http://www.usr-emea.com/products/p-broadband-product.asp?prod=bb-9108a&loc=bene

[37] De GNU General Public License : http://www.gnu.org/licenses/gpl.html

[38] Definitie van vrije software : http://www.gnu.org/philosophy/free-sw.html

[39] Vrije versie van Eric Raymond's boek The Cathedral and the Bazaar : http://catb.org/~esr/writings/cathedral-bazaar/

[40] Artikel van Richard Stallman (stichter van de Free Software Foundation) over het verschil tussen vrije en open source software : http://www.gnu.org/philosophy/open-source-misses-the-point.html

[41] Zoekmachine rond open source software : http://www.eosdirectory.com/directory/searchprojectbycateg/

[42] Software - SocialTextOpen : http://www.socialtext.net/open/index.cgi

[43] Software - MediaWiki : http://www.mediawiki.org/

[44] Software - Scalix : http://www.scalix.com/

[45] Software - eGroupware : http://www.egroupware.org/

[46] Software - Open-Xchange : http://www.open-xchange.org/

[47] Software - Zimbra : http://www.zimbra.com/

[48] Software - SugarCRM : http://www.sugarcrm.com/

[49] Software - phpBB : http://www.phpbb.com/

[50] Software - Alfresco : http://www.alfresco.com/

[51] Software - OCS Inventory NG : http://www.ocsinventory-ng.org/

[52] Het ring-model van de Intel x86-computer : http://www.nodemaster.de/xen-basics/

[53] xVM, virtualiseringsoplossing van SUN Microsystems : http://sun.com/xvm

[54] OVM, virtualiseringsoplossing van Oracle : http://www.oracle.com/technologies/virtualization/index.html

[55] User Mode Linux virtualiseringsoplossing : http://user-mode-linux.sourceforge.net/

[56] Xen virtualiseringsoplossing : http://www.cl.cam.ac.uk/research/srg/netos/xen, http://www.xen.org/ en http://www.xensource.com

[57] Linux Vserver virtualiseringsoplossing : http://linux-vserver.org/

[58] OpenVZ virtualiseringsoplossing : http://openvz.org

[59] Kernelbased Virtual Machine : http://kvm.qumranet.com

[60] Jmicron ingebouwde RAID-controller :

http://www.jmicron.com/Support_ FAQ.html

[61] RAID onder Linux : http://linux-ata.org/faq-sata-raid.html

[62] Evaluatie van het wisselgeheugen (swap) onder Linux :

http://www.ibm.com/developerworks/linux/library/l-swaptip2.html

[63] Evaluatie van partitionering der harde schijf onder Linux :

http://www.ibm.com/developerworks/linux/library/l-partitiontip.html

[64] CentOS mirror in België : ftp.belnet.be/centos/

[65] Voorbeeld voor het opzetten van een eigen, lokale repository onder een Red Hat Enterprise Linux-derivaat :

http://www.howtoforge.com /setting-up-a-local-yum-repository-fedora8

[66] Artikel van Faye GIBBINS in SysAdmin Magazine (online-editie) over het opzet ten van een veilige Xen-omgeving : http://www.samag.com/documents/s=10112/sam0702e/0702e.htm

[67] Opstartmodi van CentOS :

http://www.centos.org/docs/2/rhl-cg-en-7.2/rescuemode.html

[68] Encryptie van de harde schijf met Linux Unified Key Setup (LUKS) :

http://luks.endorphin.org/

[69] SELinux : http://www.nsa.gov/selinux/

[70] Intrusion Detection Software Aide :

http://www.bofh-hunter.com/2008/04/10/centos-5-and-aide/

[71] Intrusion Detection Software Tripwire : http://sourceforge.net/projects/tripwire

[72] Network Security Software PacketFence :

http://www.packetfence.org/english/home.html

[73] *Blue Pill rootkit* :

http://theinvisiblethings.blogspot.com/2006/06/introducing-blue-pill.html

[74] Wat rootkits zijn : http://en.wikipedia.org/wiki/Rootkit

[75] OpenSSL-bug in Debian Linux :

http://www.itnews.com.au/News/76080,openssl-bug-found-in-debian-linux.aspx

[76] Overname XenSource Inc. Door Citrix Systems : http://domain-b.com/companies/companies_c/citrix_systems/20070821_acquires.htm

[77] Lijst van Organizationally Unique Identifier voor hardware devices (gebruikt voor bijvoorbeeld MAC-adressen van netwerkkaarten) :

http://standards.ieee.org/regauth/oui/oui.txt

[78] Overzicht van het OSI-model voor beschrijving van netwerktechnologie : http://nl.wikipedia.org/wiki/OSI-model

[79] Algemeen overzicht van (een groot aantal, maar niet alle) virtualiseringsoplossingen onder Linux :

http://www-128.ibm.com/developerworks/library/l-linuxvirt/index.html

[80] Overzicht van de virtualiseringsmodi van Xen (o.a. omtrent het onderscheid tussen paravirtualisatie en volledige virtualisatie) :

http://www.linux-mag.com/id/1769

[81] Het verbergen van de netwerkkaart voor de hypervisor Xen : http://wiki.xensource.com/xenwiki/Assign_hardware_to_DomU_with_PCIBack_as_module

[82] Uitgebreide (niet-officiële) repository met softwarepakketten van de Belgische CentOS-medewerker Dag Wiëers : http://dag.wieers.com/rpm/

[83] Pythonscript voor het interactief installeren van virtuele machines onder Xen : http://people.redhat.com/~katzj/xenguest-install.py

[84] Xen management met xen-tools : http://www.xen-tools.org/software/xen-tools/

[85] Xen Debian virtuele machines installeren met behulp van debootstrap :

http://packages.debian.org/stable/admin/debootstrap

[86] Equivalent van debootstrap voor Red Hat Enterprise Linux-derivaten (rpmstrap) : http://rpmstrap.pimpscript.net/

[87] Installatie van Debian-pakketten onder Red Hat Enterprise Linux-derivaten met behulp van *alien* : http://kitenet.net/~joey/code/alien/

[88] Xen management met XenMan en ConVirt :

http://xenman.sourceforge.net/doc.html

[89] Het aanmaken van software appliances ofwel vooraf geconfigureerde virtuele machines voor gebruik onder Xen met behulp van rPath : http://www.rpath.com/corp/

[90] Xen management met openQRM : http://www.openqrm.org/

[91] Xen management met Enomalism : http://www.enomalism.com/

[92] Achtergrondinformatie omtrent Cloud Computing : http://en.wikipedia.org/wiki/Cloud_computing en http://clusters.wallonie.be/tic/en/news/2008-02-18-reservoir-cetic.html

[93] Compatibele hardware voor gebruik met OpenVZ : http://www.parallels.com/en/products/virtuozzo/hcl/

[94] Verschillende kernels voor gebruik met OpenVZ : http://wiki.openvz.org/Different_kernel_flavors_%28UP%2C_SMP%2C_ENTERPRISE%2C_ENTNOSPLIT%29

[95] Installatiegids voor OpenVZ onder CentOS van een zekere Dustin (enkel nog raadpleegbaar via Google's cache) : http://64.233.183.104/search?q=cache:1gY-RkCzOuYJ:the1337geek.com/%3Fm%3D200804+how+to+install+and+run+Openvz+on+CentOS+5.1&hl=nl&ct=clnk&cd=2&gl=be&client=firefox-a

[96] Bespreking van de mogelijkheden van de firewall iptables m.b.t. Netfilter en connection tracking : http://www.kelter.nl/artikel-firewall.html

[97] OpenVZ op een 64-bitsysteem : http://wiki.openvz.org/Install_OpenVZ_on_a_x86_64_system_Centos-Fedora

[98] Officieuze, door de gemeenschap aangeleverde, templates van virtuele containersvoor gebruik met OpenVZ :

http://linux.carreira.com.pt/ovzutils/setx86_64-0.3.tar.gz

[99] Officiële templates van virtuele containers voor gebruik met openVZ : http://wiki.openvz.org/Download/template/precreated

[100] Een grafisch programma voor het beheer van een OpenVZ-systeem :

http://sourceforge.net/projects/easyvz

[101] Beheer van OpenVZ en KVM-systemen met behulp van het webgebaseerde ProxMox : http://www.proxmox.com/

[102] Overzicht van KVM : http://en.wikipedia.org/wiki/Kernel-based_Virtual_Machine

[103] Experimentele paravirtualisatie van drivers voor netwerkkaarten onder KVM : http://article.gmane.org/gmane.comp.emulators.kvm.devel/2276

[104] Overzicht van Qemu : http://en.wikipedia.org/wiki/QEMU en

http://fabrice.bellard.free.fr/qemu/

[105] KVM-bron,code niet aangeraden voor productieomgevingen : http://kvm.qumranet.com/kvmwiki/FAQ#head-4a97776c4810df6b00037b39d88374fb97317112

[106] Nederlandstalige handleiding voor installatie van KVM als serversysteem onder Ubuntu : http://wiki.nedlinux.nl/index.php?page=+KVM%2FQEMU+op+je+server

[107] Bespreking van Tuntap : http://en.wikipedia.org/wiki/TUN/TAP

[108] Overzicht van de netwerkmogelijkheden van Qemu :

http://calamari.reverse-dns.net:980/cgi-bin/moin.cgi/QemuNetwork

[109] Qemu en CentOS : http://wiki.centos.org/HowTos/KVM

[110] Virtual Distributed Ethernet Networking (VDE) : http://sourceforge.net/projects/vde/

[111] Installatie van Enomalism onder Red Hat Enterprise Linux (en derivaten) : http://trac.enomalism.com/enomalism/wiki/enomalism-install-rh

Appendix A : Oorspronkelijke probleemomschrijving

Het ‘zenuwcentrum’ van de vzw Vorming Leon Lesoil bestaat uit losse elementen, die niet met elkaar verbonden zijn, waar zo goed als geen enkele voorziening tot beveiliging aanwezig is en dat niet toegankelijk is voor gebruikers van buitenaf, dit alles in een context van beperkte financiële middelen. Het creëren van een netwerk, met beveiligde toegangsmogelijkheden voor gebruikers op het hoofdkantoor én via ‘remote access’ dringt zich op, waarbij rekening dient gehouden te worden met het lage kennisniveau op gebied van informatica van de wisselende medewerkers.

A.1. Situatieschets

De vzw Vorming Leon Lesoil is een kleine organisatie die zich bezighoudt met politiserend vormingswerk. Deze vzw werkt zo goed als volledig met vrijwilligers uit alle delen van het land (Nederlands- en Franstalig, heel soms ook Duits- of Engelstalig). Voor sommige, specifieke taken wordt er wel eens een tijdelijke medewerker aangeworven. De middelen van deze vzw zijn erg beperkt en komen voort uit lidgelden, inkomsten van activiteiten (inschrijvingsgeld bijvoorbeeld) en een heel kleine overheidssubsidie (uiteraard gelieerd aan de ‘core business’ van de vzw).

Deze vzw beschikt momenteel over twee computers (enkele jaren oud), elk met een printer (waarvan één multi-functional) en elk met internet-access, maar *niet* over een netwerk. Beide PC’s draaien Windows XP Professional. Er wordt veel gewerkt met MS Office. Deze software is legaal (want inbegrepen bij de aankoop van de PC’s). Voor eventueel bijkomende licenties heeft de vzw geen geld.

Allerlei vrijwillige medewerkers gebruiken de beide computers, telkens met hetzelfde, algemeen bekende user-ID (van het type ‘administrator’ !) en paswoord. De meeste medewerkers zijn slechts rudimentair onderlegd in informaticasystemen (ze gebruiken de computer vooral als veredelde typemachine annex mailclient).

Daarnaast beschikt de vzw ook over een voor de buitenwereld toegankelijke website (van recente datum, gemaakt met het content management system ‘Joomla’). Over deze website is de vzw zéér tevreden. Daar dringt zich dan ook geen verandering op.

Een bijkomend probleem is dat de medewerkers die zich niet naar het hoofdkantoor te Brussel kunnen begeven, momenteel enkel in contact kunnen komen met de vzw en met elkaar via de publiek toegankelijke website. Vooral voor beheerstaken, maar ook voor onderlinge discussies, is dit niet optimaal (om het zacht uit te drukken).

Daarnaast is de situatie op het hoofdkantoor ook niet zoals het zou moeten (en kunnen) zijn, want onveilig, met een groot risico op zo ongeveer alles wat er zou kunnen mislopen en bovendien erg onpraktisch.

Tegelijk kan de vzw zich geen dure oplossingen veroorloven.

A.2. Doelstellingen

(Weergegeven volgens prioriteit van behandeling)

1. Het van bij het begin opzetten van een documentatiesysteem, dat alle ondernomen stappen documenteert en dat alle nodige informatie bevat die nodig is voor het werken met en beheren van het op te zetten systeem
2. Het uitwerken van een ‘open source’ serversysteem van het type Server Based Computing (SBC), steunend op een Linux-installatie met zogenaamde Kernel-based Virtual Machine (KVM) en ‘thin clients’
3. Als alternatieve oplossing, het uitwerken van een Virtual Desktop Infrastructure (VDI), steunend op een geïnstalleerde Xen-server met virtuele desktops, waarbij uitgezocht moet worden welke desktops er precies nodig/mogelijk zijn (bvb. Windows, Linux, MacOS).
4. Ook de beveiliging en de mailvoorziening dient met open source systemen te gebeuren.
5. Het aanschaffen van de hardware nodig voor het aanleggen van een beveiligd netwerk op het hoofdkantoor, met alles erop en eraan (server(s), clients, wireless en remote access), zo mogelijk met ingebouwde redundantie. De hardware dient zo goedkoop mogelijk te zijn (bijvoorbeeld door gebruik te maken van gerecycleerde (‘refurbished’) apparatuur).
6. Het configureren van dit netwerk, zodanig dat iedereen die toegang mag hebben ook toegang kan krijgen, zonder dat de beveiliging er onder lijdt.
7. Het opleiden van een kleine groep mensen tot administrator, zodanig dat zij dit netwerk zelfstandig kunnen beheren en onderhouden.

Appendix B : Akkoordverklaring vzw

AKKOORDVERKLARING

Hiermee verklaar ik, ondergetekende,

David DESSERS
Schoolbergenstraat 20
3010 Leuven
GSM 0486/69.69.27
daviddessers@hotmail.com

verantwoordelijke voor de ***vzw Vorming Leon Lesoil***, mij ermee akkoord dat

Peter VELTMANS
Osystraat 20, 2060 Antwerpen
GSM 0486/44.76.26
peter.veltmans@skynet.be,

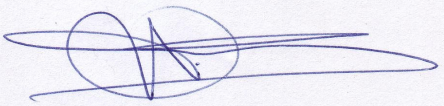
in het kader van zijn projectwerk als cursist bij het Centrum voor Volwassenenonderwijs (CVO) Antwerpen-Zuid, twee voorstellen uitwerkt voor een zo optimaal mogelijke informatisering van het hoofdkantoor van de

vzw Vorming Leon Lesoil, Plantinstraat 20, 1070 Brussel

De concrete implementering van één dezer voorstellen behoort tot de mogelijkheden, doch is afhankelijk enerzijds van de goedkeuring door de Algemene Vergadering van de vzw en anderzijds van het voorhanden zijn van voldoende financiële middelen.

Gedaan te Brussel, 25 oktober 2007 (in drievoud)

Voor de vzw Vorming Leon Lesoil,
David DESSERS, verantwoordelijke



Appendix C : Verslag vergadering met vzw

Aanwezig : David Dessers, Chris Den Hond (verantwoordelijken vzw)
Peter Veltmans (cursist)

A. Dagorde

1. Beknopte situering van projectplan
2. Beknopt overzicht van het aanwezige materiaal
3. Beknopt overzicht van de diverse aandachtspunten der vzw
4. Beknopt overzicht van de wensen der vzw
5. Beknopt overzicht van de taken der vzw
6. Afspraken omtrent verdere werkwijze

B. Verloop van de vergadering zelf

1. Beknopte situering van projectplan

   Peter Veltmans licht kort toe waaruit zijn projectplan precies bestaat en wat de werkafspraken precies inhouden (zoals bijvoorbeeld de nood aan een contact-persoon en aan een akkoordverklaring). Ook de timing wordt toegelicht.

2. Beknopt overzicht van het aanwezige materiaal

   David Dessers en Chris Den Hond overlopen – zonder in detail te treden – het aanwezige computermateriaal (een detaillering zal later opgemaakt worden). Dit komt neer op het volgende :

   - één vierkanaalsrouter (zonder draadloze voorziening)
   - aangesloten op een ADSL-modem van Belgacom
   - drie PC's van verschillend type, die elk via de router internet-access hebben, maar die niet geconfigureerd zijn in een netwerk (er is dus ook geen beveili ging ingesteld)
   - vier printers (1 multifunctional, merk HP; 1 kleuren inktjetprinter, merk Ep son; 2 laserjets, beiden merk HP). Opgemerkt dient te worden dat de multi functional, de kleurenprinter en één laserprinter elk aangesloten zijn op één PC en dat ze niet geshared worden. Dit levert uiteraard praktische problemen op, bijvoorbeeld wanneer er een kleurenprint gemaakt moet worden, terwijl het bestand bewerkt werd op een andere PC dan die waar de kleurenprinter aan gekoppeld is. De tweede laserprinter dient enkel om er etiketten (voor verzendingen) mee te printen. Als dat moet gebeuren, koppelt men de andere laserprinter los om die printer te kunnen aansluiten. Reden waarom er op de andere laserprinter geen etiketten kunnen geprint worden, is blijkbaar de

leeftijd van die printer (hij verwarmt het papier met de zelfklevende etiketten te zeer, zodat deze loskomen...).

- de opslagcapaciteit bestaat louter uit de harde schijven van de respectie velijke PC's. Die schijven zijn niet gepartitioneerd. Evenmin werden er g edeelde schijven gemaakt ('shares')

3. Beknopt overzicht van de diverse aandachtspunten der vzw

   David en Chris lichten toe dat de meeste leden der vzw thuis toegang hebben tot het Internet en dat er dus veel meer 'genetwerkt' zou kunnen worden, als er een netwerk zou zijn op het secretariaat. Dit zou ook kunnen helpen het aantal verplaatsingen van vrijwilligers naar het secretariaat te beperken.

   Wel zijn er een (beperkt) aantal vrijwilligers die wat ouder zijn en dus niet zo vertrouwd met computer en Internet. Het gaat echter over zeer weinig mensen (minder dan vijf), zodat dat geen belemmering mag zijn.

   Het is ook belangrijk dat de potentiële 'remote users' ook daadwerkelijk warm gemaakt kunnen worden voorn zo'n nieuwe werkwijze. Temeer daar gedegen kennis van informatica niet als aanwezig mag worden beschouwd.

4. Beknopt overzicht van de wensen der vzw

   David en Chris lichten toe wat zij als belangrijke zaken beschouwen :

   - het centraliseren van de opslagcapaciteit. Meer concreet wensen ze dat ver slagen, database-bestanden (zoals het adressenbestand), het foto- en filmar chief en eventuele 'groupware'-toepassingen toegankelijk gemaakt worden voor al diegenen die er 'recht' op hebben. Dit veronderstelt natuurlijk ook het instellen van een policy (wie mag er toegang hebben en in hoeverre).
   - Het moet mogelijk zijn/blijven dat losse devices (zoals USB-sticks, DVD's, CD's,...) kunnen aangesloten/geraadpleegd/afgespeeld worden op de secreta riaats-PC's.
   - De printers zouden makkelijker toegankelijk moeten zijn vanaf elke PC, met dien verstande dat er voor de kleurenprinter ook een policy nodig is die de gebruiksrechten erop inperkt (aangezien de inkt voor deze printer zeer duur is).

5. Beknopt overzicht van de taken der vzw

   David en Chris overlopen de taken die dagdagelijks op het secretariaat verricht worden, waarbij hier en daar ook de gebruikte software wordt toegelicht.

   - Montage van films, met Adobe Premiere.
   - DVD mastering, met Canon EasyPrint Toolbox.
   - Lay-out van twee tweemaandelijkse tijdschriften (één in het Nederlands, één in het Frans), met Quark Express en Adobe Indesign.
   - Lay-out van promotiemateriaal, met Quark Express en Adobe Indesign of met MS Publisher.
   - Beheer van het adressenbestand, met MS Access.

- Beantwoorden van briefwisseling, met MS Word.
- Bijhouden van een controlebestand voor de boekhouding, met MS Excel. De boekhouding zelf werd uitbesteed.

6. Afspraken omtrent verdere werkwijze

   Peter licht de verdere werkwijze toe. Hij zal de komende weken regelmatig op het secretariaat vertoeven om :

   - een gedetailleerde plaatsbeschrijving op te maken

   - een gedetailleerde inventaris op te maken

   - een gedetailleerd overzicht te maken van de zogenaamde 'workflow' op het secreatriaat.

   Vervolgens zal er een zogenaamde 'technische analyse' worden opgemaakt (die meer gericht is op het materiaal). Daarna zullen er twee praktische voorstellen op papier worden gezet. En dan is het aan de vzw om een keuze te maken.

   David en Chris merken wel op dat de implementatie van één der voorstellen niet alleen afhangt van de wenselijkheid ervan, maar ook van de goedkeuring door de Algemene Vergadering van de vzw (dit is een statutaire verplichting) en uiteraard van de budgettaire mogelijkheden.

   Peter verzekert dat die voorwaarden zullen worden opgenomen in de nog op te stellen 'akkoordverklaring'.

   Afgesproken wordt ook dat er – naast de informele contacten tijdens Peter's werkbezoeken – ook regelmatig follow-up vergaderingen zullen (kunnen) zijn.

   Peter zal van deze vergadering ook een formeel verslag opmaken.

   Einde van de vergadering.

2.2. Appendix D : Overzicht software-installatie bij de vzw

(Tabel 2 : Overzicht software-installatie bij de vzw)

COMPUTER	GEINSTALLEERDE SOFTWARE	LICENTIE ?	GEDEELTELIJK VERWIJDERDE SOFTWARE
Computer A	Adobe Illustrator 8.0 beta 26	Nee	Kopint-Datorg MultiSignoVerify
	Adobe Photoshop CS	Nee	MS Visual Studio
	Adobe Premiere Pro	Nee	Panda Software
	Adobe Reader 8.0	Freeware	Pinnacle
	Adobe Help Viewer	Nee	Trend Micro Internet Security
	Adobe ImageReady	Nee	Xerox
	Ahead : Nero 6.3.1.17	Nee	Yahoo
	Canon Foto-, film- en printsoftware	Ja	
	Cyberlink PowerDVD	Ja	
	Deamon Tools	Ja	
	DX-Ball : game	Freeware	
	FileZilla	Neen	
	CuteFTP	Open source	
	Google Toolbar	Open source	
	Grisoft AntiVirusGuard 7	Freeware	
	HitManPro	Freeware	
	HP iPAQ-drivers	Freeware	
	ITE Raid Manager	Ja	
	Lavasoft : AdAware SE Personal	Ja	
	MS Messenger	Ja	
	MS Frontpage 3.0 (!)	Nee	
	MS Office : Excel, Access, Word, Powerpoint, Outlook, Publisher	Ja	
	Quark Express	Nee	
	Photoshop installatiebestand	Nee	
	Winzip installatiebestand	Nee	
	Registry Mechanic	Nee	
		Nee	
		Freeware	

COMPUTER	GEINSTALLEERDE SOFTWARE	LICENTIE ?	GEDEELTELIJK VERWIJDERDE SOFTWARE
Computer A (vervolg)	Snapshot Viewer Softwin BitDefender 8 Spybot Search and Destroy Spyware Doctor Spyware Blaster SuperSoft Software Repair ? Webroot SpySweeper WinSCP3 + PUTTY	Freeware Freeware Nee Freeware Open source	
Computer B	Adobe Photoshop 5 Adobe Reader 5 én 6 Ahead : Nero 6.3.1.17 ArcSoft PhotoImpression4 AstonSoft DeepBurner Burnatonce Canon : Foto-, film- en printsoftware CDBurner XP Pro 3 Chikka TXT Messenger Cyberlink PowerDVD DigitalDesign Metric Convertor DivX Epson driver voor multifunctional Frozen Bubble : game Google Toolbar + Desktop Search Grisoft : AntiVirusGuard 7 Java 1.4.0_03 Java WebStart Lavasoft : AdAware 6 MS Messenger MS Frontpage 3.0 (!) MS Image Composer MS Office : Excel, Access,	Nee Freeware Nee Nee Shareware Freeware Ja Nee Nee Ja Shareware Open source Ja Nee Freeware Freeware Open source Open source Freeware Ja Nee Nee Ja	Alkwil Software Avast4 ComPlus Applications MS Visual Studio Norton AntiVirus Systran 4 Xerox XoftSpy

COMPUTER	GEINSTALLEERDE SOFTWARE	LICENTIE ?	GEDEELTELIJK VERWIJDERDE SOFTWARE
Computer B (vervolg)	Word, Powerpoint, Outlook, Publisher MS Movie Maker Netscape 7 Quark Express 4.0.1 RealPlayer ScanSoft OmniPagePro 12.0 + PDFCreate 3 Softwin BitDefender 8 Viewpoint Media Player WinAmp Yahoo toolbar	Ja Open source Nee Freeware Ja Ja Nee Nee Freeware Freeware	
Computer C	Adobe Photoshop CS 2 Adobe Photoshop 7 (!) Adobe InDesign CS Adobe Reader 8 CMedia 3D Audio Driver FileMaker Pro 4.1 FileZilla Client Grisoft AntiVirusGuard 7 Le Petit Robert MS Messenger MS Frontpage 3.0 (!) MS Image Composer MS Office : Excel, Access, Word, Powerpoint, Outlook, Publisher MS Movie Maker Quark Express 6 SmartObjects IsoBuster	Nee Nee Nee Freeware Nee Nee Open source Freeware Nee Nee Nee Nee Nee Nee Nee Nee	ComPlus Applications MS Visual Studio Xerox

Appendix E : Kickstart-script voor Xen VM

```
reboot

install

text

url --url http://192.168.1.11/CentOS/base/

lang en_US.UTF-8

keyboard be-latin1

rootpw --iscrypted $1$P6P/WdnN$xmkGZxEcFNw4JrJFH.EF31

firewall --enabled --port=22:tcp

authconfig --enableshadow --enablemd5

selinux --enforcing

timezone Europe/Brussels

bootloader --location=mbr --driveorder=xvda --append="quiet"

zerombr yes

auth  --useshadow  --enablemd5 --enableldap --enableldapauth --ldapserver \
  192.168.1.10 --ldapbasedn dc=testdomein,dc=org --enablekrb5 --krb5realm \
  TESTDOMEIN.ORG --krb5kdc 192.168.1.10 --krb5adminserver 192.168.1.10

network --bootproto=static --ip=192.168.1.10 --netmask=255.255.255.0 \
  --gateway=192.168.1.1 --nameserver=192.168.1.11 \
  --hostname=krb5.testdomein.org

skipx

clearpart --all

part / --fstype ext3 --size=1 --grow --ondisk=xvda --asprimary

part swap --size=1024 --ondisk=xvdb --asprimary

%packages

@admin-tools
```

```
@base
@base-x
@core
@editors
@text-internet
comps-extras
cracklib-dicts
rmt
tzdata
screen
openssl
openldap-servers
emacs
krb5-server
bind
-gnome-power-manager
-esc
-gnome-screensaver
-gnome-pilot
-pcmciautils
-irda-utils
-bluez-utils
-synaptics
-linuxwacom
%post
echo "preyum ">/dev/tty1
```

```
mv /etc/yum.repos.d{,-orig}
mv /etc/rc3.d/S27ldap /etc/rc3.d/S12ldap
echo "192.168.1.11   testmachien.testdomein.org testmachien" >> /etc/hosts
cat >>/etc/yum.conf <<EOF
[base]
name=CentOS \$releasever - \$basearch - Released Updates
baseurl=http://192.168.1.11/CentOS/base/
enabled=1
gpgcheck=1
gpgkey=http://192.168.1.11/fc5/core/RPM-GPG-KEY-fedora
[updates-released]
name=CentOS \$releasever - \$basearch - Released Updates
baseurl=http://192.168.1.11/CentOS/updates/
enabled=1
gpgcheck=1
gpgkey=http://192.168.1.11/CentOS/base/RPM-GPG-KEY-fedora
EOF

/usr/bin/yum -y update >/dev/tty1
for i in anacron autofs hidd avahi-daemon smartd cups \
  cups-config-daemon; do
        /sbin/chkconfig --level 35 $i off
done
for i in named httpd ldap; do
        /sbin/chkconfig --level 35 $i on
done
mv /etc/rc3.d/S27ldap /etc/rc3.d/S13ldap
```

```
cat >> /etc/rc.local <<EOF
if ! iptables -nL RH-Firewall-1-INPUT | grep "dpt:80" | grep "dpt:80"; then
IP=/sbin/iptables

# delete unwanted rules
        \${IP} -D RH-Firewall-1-INPUT -p tcp -m tcp --dport 631 -j ACCEPT
        \${IP} -D RH-Firewall-1-INPUT -p udp -m udp --dport 631 -j ACCEPT
# add new rules for services we use.
        UDP="-I RH-Firewall-1-INPUT 7 -m udp -p udp"
        TCP="-I RH-Firewall-1-INPUT 7 -m state --state NEW -m tcp -p tcp"
        \${IP} \${TCP} --dport 25 -j ACCEPT
        \${IP} \${TCP} --dport 53 -j ACCEPT
        \${IP} \${UDP} --dport 53 -j ACCEPT
        \${IP} \${TCP} --dport 80 -j ACCEPT
        \${IP} \${TCP} --dport 636 -j ACCEPT
        \${IP} \${UDP} --dport 636 -j ACCEPT
        \${IP} \${TCP} --dport 5432 -j ACCEPT
        \${IP} \${UDP} --dport 5432 -j ACCEPT
        service iptables save
fi
setsebool -P httpd_tty_comm 1
EOF
setsebool -P httpd_tty_comm 1
mkdir -p /var/repo
echo -e "/dev/hda1\t/var/repo\text3\tdefaults,acl\t0 0" >> /etc/fstab
```

Appendix F : Script voor een veilige Xen VM

```
#!/bin/bash
ethtool -K vif0.0 tx off
ethtool -K eth0 tx off
ethtool -K peth0 tx off
ebtables -F INPUT
ebtables -F FORWARD
ebtables -F OUTPUT
ebtables -A FORWARD --out-interface peth0 --protocol ipv4 \
  --ip-protocol udp --ip-destination-port 67:68 -j DROP
ebtables -A FORWARD --in-interface peth0 --protocol ipv4
service dhcpd restart
lvcreate -L10G -n 00_XenVM_root VG_VMS
lvcreate -L1024M -n 00_XenVM_swap VG_VMS
wget -O/tmp/k http://192.168.1.11/CentOS/base/images/xen/vmlinuz
wget -O/tmp/i http://192.168.1.11/CentOS/base/images/xen/initrd.img
cp listing02.txt /var/repo/CentOS/ks.cfg # = het kickstart-script
chmod 755 /var/repo/CentOS/ks.cfg
cp listing03.txt /etc/xen/00_XenVM # = configuratiebestand voor installatie
xm create -c 00_XenVM
sleep 5
service httpd stop
service dhcpd stop
umount /dev/mapper/LV_repo
cp listing04.txt /etc/xen/00_XenVM # = configuratiebestand gewoon gebruik
xm create -c 00_XenVM
```

Appendix G : Simpel script voor OpenVZ-containers

```
#!/bin/bash
############################################################
# SCRIPT VOOR HET AANMAKEN VAN MEERDERE OPENVZ-CONTAINERS #
############################################################
# Maak container 201 - groupware-server
############################################################
vzctl create 201 --ostemplate centos-5-x86_64-default
sleep 1
vzctl set 201 --onboot yes --save
sleep 1
vzctl set 201 --hostname groupware.testdomein.org --save
sleep 1
vzctl set 201 --ipadd 10.0.10.201 --save
sleep 1
vzctl set 201 --nameserver 192.168.1.11 --save
sleep 1
vzctl start 201
sleep 1
vzctl set 201 --userpasswd root:2Bn2b
sleep 1
############################################################
# Maak container 202 - crm-server
############################################################
vzctl create 202 --ostemplate centos-5-x86_64-default
sleep 1
```

```
vzctl set 202 --onboot yes --save
sleep 1
vzctl set 202 --hostname crm.testdomein.org --save
sleep 1
vzctl set 202 --ipadd 10.0.10.202 --save
sleep 1
vzctl set 202 --nameserver 192.168.1.11 --save
sleep 1
vzctl start 202
sleep 1
vzctl set 202 --userpasswd root:2Bn2b
sleep 1
############################################################
# Maak container 203 - erp-server
############################################################
vzctl create 203 --ostemplate centos-5-x86_64-default
sleep 1
vzctl set 203 --onboot yes --save
sleep 1
vzctl set 203 --hostname erp.testdomein.org --save
sleep 1
vzctl set 203 --ipadd 10.0.10.203 --save
sleep 1
vzctl set 203 --nameserver 192.168.1.11 --save
sleep 1
vzctl start 203
sleep 1
```

```
vzctl set 203 --userpasswd root:2Bn2b
sleep 1
############################################################
# Maak container 204 - vdi-server
############################################################
vzctl create 204 --ostemplate centos-5-x86_64-default
sleep 1
vzctl set 204 --onboot yes --save
sleep 1
vzctl set 204 --hostname vdi.testdomein.org --save
sleep 1
vzctl set 204 --ipadd 10.0.10.204 --save
sleep 1
vzctl set 204 --nameserver 192.168.1.11 --save
sleep 1
vzctl start 204
sleep 1
vzctl set 204 --userpasswd root:2Bn2b
sleep 1
############################################################
# Maak container 205 - backup-server
############################################################
vzctl create 205 --ostemplate centos-5-x86_64-default
sleep 1
vzctl set 205 --onboot yes --save
sleep 1
vzctl set 205 --hostname backup.testdomein.org --save
```

```
sleep 1
vzctl set 205 --ipadd 10.0.10.205 --save
sleep 1
vzctl set 205 --nameserver 192.168.1.11 --save
sleep 1
vzctl start 205
sleep 1
vzctl set 205 --userpasswd root:2Bn2b
sleep 1
############################################################
# Maak container 206 - print-server
############################################################
vzctl create 206 --ostemplate centos-5-x86_64-default
sleep 1
vzctl set 206 --onboot yes --save
sleep 1
vzctl set 206 --hostname print.testdomein.org --save
sleep 1
vzctl set 206 --ipadd 10.0.10.206 --save
sleep 1
vzctl set 206 --nameserver 192.168.1.11 --save
sleep 1
vzctl start 206
sleep 1
vzctl set 206 --userpasswd root:2Bn2b
sleep 1
############################################################
```

```
# EINDE SCRIPT

############################################################

exit
```

Appendix H : Implementatieplan

Hieronder geven we een ruwe schets van wat een mogelijk plan zou kunnen zijn voor de implementatie door de vzw van een mini-datacenter met behulp van virtuele machines onder OpenVZ. Het is evident dat een concreet plan enkel tot stand kan komen, na overleg met alle betrokkenen. Vandaar dat dit slechts een schets of een eerste voorstel kan zijn. De totale kostprijs voor dit alles blijft feitelijk beperkt tot de aankoop van de nieuwe, krachtige computer (ca. € 1.200,00) en het duurdere ADSL-abonnement met vast IP-adres (€ 535,72 eerste jaar[187], daarna € 175,27 per jaar). Het praktische werk gebeurt immers allemaal met vrijwilligers.

Fase	Periode	Doelstelling
1	Eerste helft juli 2008	- wegwerken kabel in kabelgoten + vervangen te korte kabels, vastzetten los zittende harde schijf e.d.m. - overstappen van goedkoop ADSL-abonnement met variabel IP-adres naar duurder pakket met vast IP-adres (zodat netwerk klaar is voor toegang van buitenaf)
2	Tweede helft juli 2008	- Backup maken van alle aanwezige data-bestanden - Herinstalleren computers A en B (legale versies van Windows XP Pro, gelicentieerde softwarepakketten + vervangen MS Office door OpenOffice.org, enz.) - Configureren van voorlopig netwerkje, toegankelijk maar ook veilig - Sorteren van databestanden en terugplaatsen onder één, gedeelde map
3	Eerste helft augustus 2008	- Installeren op computer C van een gebruiksvriendelijke Linux-versie (bijvoorbeeld Ubuntu), ter vervanging van de niet-legale Windows-versie. - Installeren van diverse serverapplicaties op deze computer. Deze computer voorlopig dus gebruiken als een hybride machine (server + werkstation).
4	Van augustus tot december 2008	Opleiden van de medewerkers (inleiding tot Linux, systeemadministratie en gebruik van de server-applicaties).
5	Van augustus 2008 tot juni 2009	Proefdraaien tot het gros van de medewerkers vlot overweg kan met de nieuwe manier van werken.
6	Zomer 2009	Aankoop van de gewenste servermachine + configuratie ervan met virtuele machines onder OpenVZ
7	Zomer 2009 tot jaareinde	Dubbel draaien met bestaand netwerkje en nieuwe machine, tot alles op punt staat
8	Januari 2010	Overzetten van de datagegevens naar een container onder OpenVZ; ombouwen van de bestaande, oudere computers tot thin clients (gebruik makend van 2X als connection broker).

187Ten gevolge van de eenmalige installatiekost van € 297,89 + BTW.