Nancy-Université
Université Nancy 2

Année universitaire 2008-2009

Mise en œuvre de routeurs et intégration de fonctionnalités de QoS

iut Nancy-Charlemagne

PLANET NETWORK-SERVICES

Auteur :
Gatien Gaspard

28 juin 2009

Table des matières

Rapport de stage :
Mise en œuvre de routeurs
et
intégration de fonctionnalités de QoS

Plan-Net S.A.
Université Nancy 2 - IUT Nancy Charlemagne
Année universitaire 2008-2009

Auteur : Gatien Gaspard

Coordonnées de l'entreprise :
Plan-NET S.A.
8 rue des Martyrs
L-3739
Rumelange, Luxembourg

Coordonnées du tuteur :
M. Brent Frère
tel : +352 26 56 02 22
mail : brf@plan-net.lu

Coordonnées du parrain :
M. Abdessamad Imine
tel : +33 3 54 95 85 35
mail : abdessamad.imine@loria.fr

Chapitre 1

Remerciements

Je tiens à remercier l'ensemble du personnel de Plan-Net pour son accueil, sa sympathie et son apport précieux durant l'ensemble du stage, aussi bien sur le plan professionnel que personnel.

Je remercie ainsi Laurent BOHNERT, avec qui j'ai eu la chance de mettre en place des routeurs chez des clients, Arnaud qui nous a permis, à moi comme aux deux autres stagiaires de la promotion, Yilmaz Ilhan et Luis Dominguez Lopez, d'apprendre à utiliser un ERP, et ainsi de gérer plus efficacement notre temps de travail.

Je me dois de citer Raymond MASSARD, pour m'avoir permis d'intégrer cette entreprise pendant ce stage ainsi que Sylvie GEIS, Noël BUANNIC et George ATAMA-GAMA pour leur accueil et leur aide.

Et je ne peux oublier mes collègues stagiaires, Luis Alonso DOMINGUEZ LOPEZ et Yilmaz ILHAN de licence ASRALL et Aurélien OUYESSAD en BTS Comptabilité à Metz.

Je remercie bien évidement M. Brent FRÈRE, mon tuteur dans l'entreprise, qui a su me confier des tâches intéressantes, et M. Abdessamad IMINE, mon parrain de stage pour l'IUT.

Et pour finir, je tiens à remercier le personnel de l'IUT et notamment les professeurs et intervenant en licence professionnelle ASRALL ainsi que toutes les personnes qui prendront la peine de lire ce rapport.

Chapitre 2

Introduction

2.1 Présentation

2.2 Le cadre du stage

J'effectue un stage de 12 semaines du 6 avril au 26 juin 2009 au sein de l'entreprise Plan-Net à Rumelange dans le cadre de ma formation actuelle, la licence professionnelle ASRALL, qui est une licence tournée vers l'administration réseau et l'utilisation de logiciels libres.

Cette formation m'a permis d'apprendre énormement de chose et m'a conforté dans l'utilisation de logiciels et de systèmes d'exploitation libres.

Étant donné l'orientation réseau de cette licence, j'ai décidé de chercher un stage plutôt orienté réseau tout en utilisant des logiciels libres. C'est donc en recherchant des sociétés travaillant dans ce domaine que j'ai découvert, grâce à Internet, la société Plan-Net S.A. situé à Rumelange dans le Grand Duché du Luxembourg.

Ce qui m'a attiré dans la présentation de cette entreprise, sur leur site Internet, http ://plan-net.lu, c'est la diversité technique affichée et la spécialisation autour du libre tout en travaillant avec tout type de clients, des clients ayant des infrastructures propriétaires existantes à ceux dont l'infrastructure doit être composée, Plan-Net les conseille dans le but de réduire au maximum les coûts pour la société tout en garantissant un service maximum.

2.2.1 Les outils

Durant ce stage, je travaille essentiellement sur des machines DELL équipées de deux cartes réseaux et utilisées en tant que routeurs.

Coté logiciel, je travaille essentiellement dans la configuration de parefeux avec FWBuilder[1] et d'autres outils que j'ai découvert et qui seront présentés par la suite. Je dispose également d'un logiciel de gestion de planning, OpenERP dont le déploiement est en cours au sein de la société, je participe donc, en l'utilisant, au test de cette application en tant qu'utilisateur final et celà me permet de suivre au jour le jour l'avancement de chaque tâches de mon projet.

Afin de faciliter la communication et de rendre accessible les informations concernant mon travail qui peuvent être utiles à d'autres membres de la société, je dispose également d'un

[1] Logiciel de gestion de parefeu multi-plateforme

accès au forum interne de l'entreprise sur lequel je poste régulièrement des informations sur mon travail.

2.2.2 Le sujet du stage

Pour ce stage, mon thème principal est la mise en œuvre de qualité de service au sein de *Firewalls*. Ce sujet comporte différents projets dont la recherche d'informations, le test et la mise en œuvre de qualité de service sur les machines présentées précedemment, soit des machines DELL transformées en routeurs.

La qualité de service *QoS* est l'ensemble des règles permettant la gestion de la bande passante selon des règles de filtrage précises. Ces règles permettent ainsi de garantir une bande passante minimale pour certains services et/ou clients tout en offrant la possibilité d'utiliser un surplus de bande passante lorsque celle-ci est disponible.

Tout l'intérêt de mon projet réside donc dans la découverte, le test et la documentation des divers outils intéressants dans ce domaine. L'ensemble de cette documentation est disponible en interne sur un forum dédié de la société.

Chapitre 3

Présentation de l'entreprise

Ce stage étant effectué au sein de l'entreprise Plan-Net S.A, voici une présentation de cette société.

3.1 Activités de l'entreprise

Plan-Net est une société luxembourgeoise de services et de conseil dans le secteur informatique. Sa particularité avec la majeure partie des sociétés actuelles est son implication dans le monde du logiciel libre. En effet, bien que les techniciens soient parfois amenés à gérer les outils propriétaires chez les clients, ceux-ci leurs préconise d'utiliser au maximum des logiciels libres pour leurs propre pérénité lorsqu'un équivalent libre existe et permet au minimum de faire la même chose que les outils propriétaires installés.

Le but de ces conseils est de ne pas rendre le client captif d'une solution. Ainsi, les solutions proposées sont autant que possible des outils opensource permettant au client de poursuivre son activité dans le cas où cet outil ne serait plus maintenu puisque les sources sont disponibles et qu'il sera donc toujours possible d'adapter l'outil aux besoins du client.

Plan-Net étant avant tout une société de conseil, celle-ci intervient dans divers domaines de l'informatique allant du simple conseil au développement en passant par le dépannage.

Le principal avantage de cette société est un temps de réaction relativement rapide et des solutions permettant la gestion de l'infrastructure de la plupart des clients depuis le réseau Intranet de la société. Cela permet souvent d'économiser de l'argent pour les clients et du temps pour les techniciens qui peuvent intervenir à distance.

3.1.1 Des compétences diverses

Voici une brève présentation du personnel de Plan-Net.
- Le patron, il gère l'entreprise, il a des compétences en développement et en gestion financière.
- Une assistante de direction, qui s'occupe notamment de la comptabilité et de la facturation.
- Un commercial, il cherche des appels d'offres afin d'y répondre, et gère les relations clients.

- Un spécialiste linux, spécialiste réseau sur workstation et serveurs linux. Il a une double compétence due à ses deux diplômes d'ingénieur : informatique et électronique. Il fait surtout de l'assistance réseau, de l'assistance linux et intervient sur tous types de problèmes.
- Un spécialiste réseau linux et windows.
- Un spécialiste réseau Microsoft Certified et Bob.
- Quatres stagiaires, dont 3 en licence ASRALL et en 1 BTS comptabilité qui travaille sur la compréhension d'OpenERP.

Une équipe composée de 5 personnes travaille en permanence pour le développement à l'armée luxembourgeoise. Cette équipe

- Un chef de projet Java J2EE, Workflow, SQL.
- 4 développeurs Java J2EE.

3.2 Aspect juridique et économique

Plan-net est une société luxembourgeoise de services et de conseils en technologies de l'information et en télécommunications crée en février de l'an 2000 sous la forme d'une S.A.R.L.

Son siège est situé à Rumelange (Luxembourg), dans les locaux du plus vieux cinéma encore en activité au Luxembourg.
Plan-net emploie actuellement 14 personnes dans des domaines de compétences très variés mais tous tournés vers le logiciel libre.
Voici les cinq domaines de compétences principaux de Plan-net :

- Dépannage informatique et télécommunications
- Développements (applications, sites web, ...)
- Conseils en technologies de l'information et en télécoms
- Publications électroniques (vidéo texte, télévision câblée, ...)

- Videosurveillance, mise en place de systèmes multimédia, ...

3.2.1 Aspect Juridique

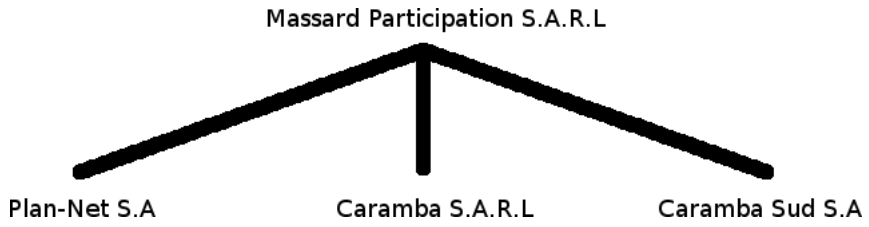

FIG. 3.1 – Organigramme des sociétés gérées par Massard Participation

En 2000, la société Plan-Net a été crée sous la forme d'une S.A.R.L avec un capital d'environ 12000 euros réparti en 50 part entre 5 personnes. Au fur à mesure du développement de la société, ces parts ont finis par se regrouper jusqu'à ne plus appartenir qu'à 2 personnes puis en 2006, la personne ayant le plus de parts les a cédé à la société Massard Participation. Depuis lors, Massard Participation a fait monter le capital à 112 500 euros réparti en 450 parts dont

435 appartenaient à Massard Participation. Une fois cette augmentation de capital effective, la société Plan-Net a changé de forme juridique pour devenir une S.A.
Massard Participation possède la majorité de la société Plan Net. Elle possède aussi la société *Caramba* et la majeure partie de *Caramba Sud*, des sociétés luxembourgeoises d'exploitation des salles de cinéma.

3.2.2 Aspect Financier

- Chiffre d'affaire 2008 : 1,3 millions d'euros
- Bénefice : 80000 euros
- Nombre d'employés : 14 (+4 stagiaires)
- Nombre de clients : 30

3.3 L'éthique

Plan-Net met un point d'honneur à respecter le client afin de tout faire dans son intérêt. En offrant des solutions libres, le client n'est pas prisonnier d'une technologie car le libre offre le plus souvent des solutions pour l'interopérabilité et le respect des standards.
Plan-Net refuse tout accord d'exclusivité ou tout accord le liant avec un fournisseur de logiciels, de matériel ou de services télécoms. Plan-Net refuse toute forme d'intéressement lorsqu'il recommande tel ou tel fournisseur à ses clients (marges arrières) afin que ses conseils ne puissent en aucun cas être biaisés par un intérêt parasitaire.

Plan-Net refuse de rendre un client captif. Pour ce faire, le dossier technique constitué par Plan-Net au contact d'un client lui est restitué sur simple demande, les solutions logicielles développées spécifiquement pour un client lui sont communiquées sous forme de code source et les solutions préconisées sont prioritairement prises dans le monde du logiciel libre ou Open-Source, afin que le client puisse faire appel à d'autres sociétés de service pour leur maintenance et puisse choisir à l'avenir leur fournisseurs de mises à jour.

Plan-Net met en œuvre une politique de gestion des ressources intelligente. L'assistance à distance réduit les frais de déplacement et facilite le dépannage. Le recyclage des anciens ordinateurs de bureaux en firewalls, routeurs ou serveurs de faxes ou d'impression permet d'économiser l'énergie et les moyens.

FIG. 3.2 – Logo officiel de Plan Net S.A.

Chapitre 4

Projet

4.1 Présentation du projet

Mon projet consiste en la mise en œuvre de qualité de service au sein de parefeux. Je vais pour ce faire, rechercher et étudier divers outils permettant de faciliter la mise en place de ce type de configuration.

Les routeurs sont installés depuis une image de Xubuntu 8.04 qui est un version LTS [1] avec des logiciels très pratiques déjà installés tels que Webmin ou FWBuilder.

Voici donc ci-dessous l'expression des besoins de début de projet en ce qui concerne la qualité de service :

- Il doit être possible de limiter la bande passante, par service et/ou par client.
- Si plus de bande-passante est disponible, celle-ci doit être dispatchée équitablement entre les clients connectés.
- Cette solution de gestion de QoS doit être simple à mettre en œuvre et tenir compte des logiciels installés.
- Simplifier la procédure d'installation des routeurs.

Ce cahier des charges à par la suite impliqué d'autres tâches comprenant notamment la migration des parefeux de la société de SuseFirewall2 vers FWBuilder. J'ai donc essentiellement travaillé sur l'installation et la configuration de routeurs ainsi que les divers éléments que ceux-ci peuvent comporter, DNS, DHCP, tunnels OpenVPN, ...

[1] LTS : Long Time Support, Support technique à long terme.

4.2 Mise en œuvre de qualité de service

Mon sujet principal pendant ce stage consiste à mettre en œuvre de la qualité de service sur les parefeux de la société. Pour ce faire, je vais devoir étudier les différentes méthodes existantes afin de mettre en place des règles telles que la gestion de bande-passante ou la prioritisation de traffic sur les routeurs.

Avant de commencer, voici une petite explication de ce qu'est la qualité de service (QoS). La QoS pour un routeur, cela inclu la gestion de la bande-passante, la gestion des priorités et éventuellement de l'équilibrage de charge.

Le cahier des charges du projet est le suivant :

- Pouvoir limiter la bande passante en fonction de la source et/ou du protocole.
- Pouvoir utiliser un surplus de bande passante lorsque celui-ci est disponible.
- Pouvoir rendre des services, des réseaux, des postes plus prioritaires.
- Utiliser un outil de gestion de bande passante graphique et simple à utiliser.
- Pouvoir visualiser cette gestion de bande passante à fin de développement.
- Rendre la démarche plus facile en essayant de la conjuguer avec FWBuilder.
- Rédiger une notice explicative sur cette mise en œuvre.

Afin de répondre à ces besoins, je commence par rechercher diverses solutions logicielles. Voici donc une liste d'outils existants plus ou moins évolués permettant de gérer des règles de qualité de services :

1. Webmin-Htb :
   - Webmin est un outil qui permet d'administrer une machine depuis une interface web
   - Webmin-Htb est un module pour "Webmin" permettant une mise en œuvre efficace de règles de QoS
   - Licence : GPL
   - Dernière version : 0.1.1 Alpha - 2005
   - Site officiel : `http ://www.sehier.fr/webmin-htb/`

2. RCC :
   - RCC est un logiciel de gestion de QoS pour Linux
   - Licence : GPL
   - Dernière version : 0.8 - Mars 2005
   - Site officiel : `http ://sourceforge.net/projects/rcc`

3. Wonder Shaper :
   - Wonder Shaper est un outil en ligne de commande qui simplifie la gestion de la commande TC
   - Licence : GPL
   - Dernière version : 1.1a - avril 2002
   - Site officiel : `http ://lartc.org/wondershaper/`

4. Trickle :
   - Trickle est un outil qui permet de limiter la bande-passante en ligne de commande pour des applications définies

- Ex : trickle -d 1 wget http ://plan-net.lu/ permet de télécharger la page demandée à la vitesse de 1ko/s
- Licence : BSD
- Dernière version : 1.07 - 2002

5. Mastershaper :
   - Interface web permettant la gestion de règles de QoS
   - Licence :
   - Dernière version : 2007
   - Site officiel : `http ://www.mastershaper.org/index.php/Main_Page`

6. QtMonitor :
   - Outil de monitoring permettant de surveiller le respect des règles de QoS facilement
   - Licence : GPL v2
   - Dernière version : 1.0 - août 2007
   - Site officiel : `http ://rocher.daniel.free.fr/wiki/wakka.php?wiki=QtMonitor`

Pour réaliser la mise en place de règles de QoS, l'utilisation de Webmin-htb et de QtMonitor en parallèle permet un développement efficace. En effet, Webmin étant un outil très utilisé pour l'administration de machines, l'ajout d'un module afin de gérer la qualité de service semble être une bonne idée d'autant que l'outil Webmin est déjà en production sur les différents routeurs du réseau. D'après les élements présentés ci-dessus, les outils les plus intéressants sont le module Webmin-Htb et Master Shaper pour la gestion des règles de QoS et QtMonitor pour surveiller le traffic effectif sur le réseau d'après les classes qui auront été définies.

Master Shaper est un logiciel bien plus complexe que le module Webmin-Htb qui permet de définir en profondeur des règles de qualité de service en choisissant le type de file d'attente utilisée, Htb[2], HFSC[3] ou encore CBQ [4]. Cependant, étant donné la demande, ce logiciel semble trop complexe ainsi, pour la suite, nous utiliserons le module Webmin-Htb.

4.2.1 Mise en œuvre de Webmin-Htb

Webmin-Htb étant un module pour "Webmin", il faut tout d'abord disposer de celui-ci sur le routeur(pare-feu) à configurer. Nous allons donc commencer par installer Webmin en passant par les dépôts d'Ubuntu. Le paquet nécessaire n'étant pas maintenu dans les dépôts officiels des dernières versions, il faut ajouter le dépôt source de *Debian Sarge* pour l'installer avec le gestionnaire de paquets *APT*.

Installation de Webmin

- Ajouter la clé gpg du repository à ajouter au système :

```
cd /root
sudo wget http://www.webmin.com/jcameron-key.asc
sudo apt-key add jcameron-key.asc
```

[2]Htb : Hierarchy Token Bucket
[3]HFSC : Hierarchy ...
[4]CBQ :

- Ajouter la ligne suivante à /etc/apt/sources.list :
  `deb http://download.webmin.com/download/repository sarge contrib`
- Installer Webmin :
  `sudo apt-get update && sudo apt-get install webmin`
- Modifier le mot de passe pour Webmin :
  `sudo /usr/share/webmin/changepass.pl /etc/webmin root votre\_mot\_de\_passe`
- Webmin est désormais accessible via :
  `https://ip\_du\_routeur\_ou\_webmin\_est\_installé:10000/`

Nous avons maintenant un poste avec Webmin d'installé, nous pouvons donc passer à la suite, l'installation du module Webmin-Htb.

Installation du module Webmin-Htb

- Copier le script htb.init dans /etc/init.d/
  `http://sourceforge.net/projects/htbinit/`
- Installer la bibliothèque Perl nécessaire :
  `cpan -i Tree::DAG\_Node`
- Installer Webmin-Htb via le gestionnaire de modules de Webmin :
  `cd /usr/share/webmin/`
  `wget http://www.sehier.fr/webmin-htb/webmin-htb.tar.gz`
  `webmin -> webmin configuration -> webmin modules`
  `indiquer l'emplacement du module et affecter les droits à un utilisateur`
- Le module est maintenant accessible sous :
  `Networking -> Hierarchy Token Bucket Queuing`

Ce module étant en version alpha, les labels sur les boutons de l'interface ne sont par défaut visibles que si l'interface Webmin est configurée en français. Pour palier à ce problème et permettre l'utilisation de ce module dans les autres langues, il suffit de créer un fichier correspondant à la langue désirée à partir du fichier fr dans :

`/usr/share/webmin/htb/lang`

Ainsi, pour pouvoir utiliser ce module convenablement dans la langue de son choix, il suffit de copier ce fichier, de le traduire (10 lignes) et de le renommer selon le pays désiré.

Configuration du module Webmin-Htb

Maintenant que Webmin et le module Webmin-Htb sont installés, nous allons commencer la configuration de notre module. Avant tout, ce module sauvegarde ses données dans /etc/sysconfig/htb. Il faut donc créer ce répertoire :

`mkdir /etc/sysconfig/htb/`

Nous pouvons maintenant commencer la configuration de Webmin-Htb, que nous allons découvrir progressivement au travers de son interface web. Nous commençons donc par ouvrir l'url suivante sur *Firefox* :

`https://ip_du_routeur:10000/`

puis dans le menu de gauche :

`Network -> Hierarchy Token Bucket`

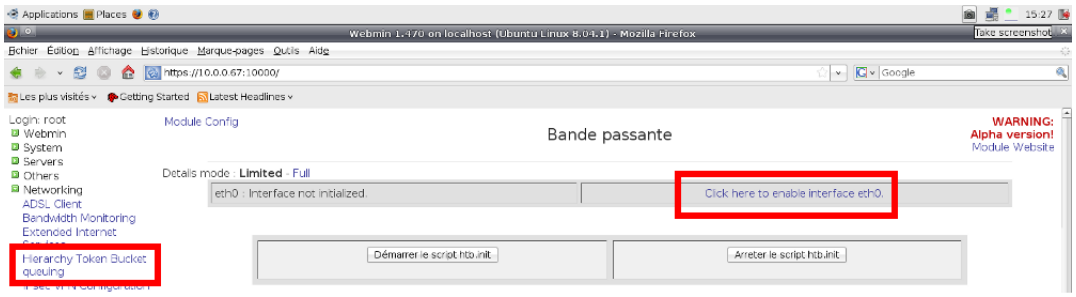

FIG. 4.1 – Interface du module Webmin-Htb

Nous allons maintenant créer une structure en arbre représentant les différentes connexions intéressantes :

Par exemple pour 3 clients se partageant une connexion 2Mbit avec possibilité d'utiliser le surplus de bande passante inutilisé par les autres clients :

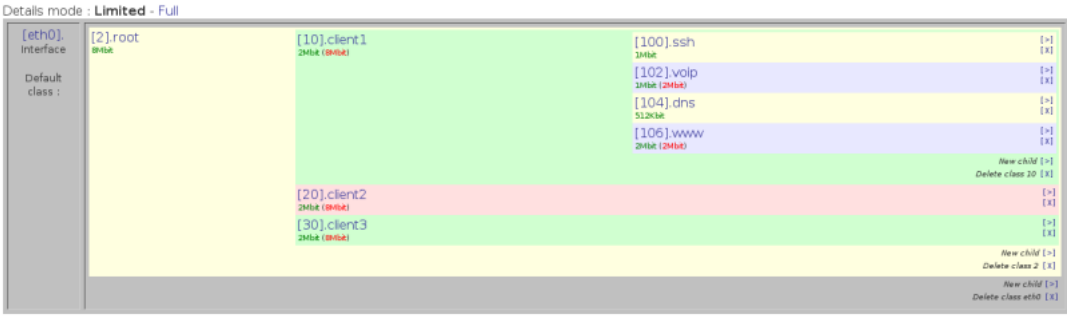

FIG. 4.2 – Configuration de test sur Webmin-Htb

Pour créer une nouvelle branche dans la structure actuelle, il suffit de cliquer sur la flèche située en bout de ligne.

Le rate indique la vitesse de connexion par défaut (celle réservée au client)
`ex : 2Mbit`
Le ceil indique la bande passante supplémentaire utilisable si disponible
`ex : 8Mbit`
La priorité indique l'ordre de priorité d'attribution de la bande passante
`ex : ssh prio=1, voip prio=2, dns prio=3, www prio=4`

Une fois la structure crée, il ne reste plus qu'à lancer le script en cliquant sur le bouton de gauche "Démarrer le script htb.init"

Automatiser le lancement du script (au démarrage du système)

`sudo update-rc.d -f htb.init defaults`

Les règles régissant la bande-passante sont maintenant crées et opérationnelles, cependant, il faut vérifier leur bon fonctionnement. C'est là qu'intervient QtMonitor puisque celui-ci va nous permettre de vérifier le traffic de chaque classe crée.

Visualisation des règles de QoS

Avant même de commencer à utiliser QtMonitor afin de visualiser le traffic en cours dans chaque classe crée, commençons par vérifier si nos classes ont bien été prises en compte par le système en utilisant directement la commande TC :

`tc -d class show dev ethXX` (ou ethXX est l'interface définie)

Si les classes sont bien définies, la réponse doit être une liste de `class htb ...` et nous pouvons visualiser le contenu de ces classes via QtMonitor que nous allons installer dès à présent.

4.2.2 Installation de QtMonitor

QtMonitor fonctionne en mode client-serveur, il faut donc installer le serveur sur la machine à surveiller et le client sur un autre poste (ou le même si l'on travaille en local).

1. Les dépendances nécessaires côté client et serveur :
   - libpam0g-dev
   - g++
   - libqt4-dev

2. Compilation et installation du serveur :
   - cd server
   - qmake
   - make
   - sudo make install

3. Compilation et installation du client :
   - cd client
   - qmake
   - make
   - sudo make install

4. Génération des clés (ssl) sur le serveur :
   - cd /etc/qtmonitord/
   - openssl genrsa -des3 -out privkey.pem 2048 (clé privé - droit 640)
   - saisir une passphrase
   - Certificat autosigné avec la clé privkey.pem :

– openssl req -new -x509 -key privkey.pem -out server.pem -days 10000

5. Ajouter les utilisateurs autorisés à se connecter
   – Les ajouter dans /etc/qtmonitord/qtmonitord.users

6. Démarrage du serveur :
   – /etc/init.d/qtmonitord start

QtMonitor nous permet de voir, sur le client, les différentes classes définies ainsi que leur application sous forme de graphiques mais il permet également, en phase de développement de déterminer les noms des classes htb crées si l'on a du mal à comprendre leur numérotation depuis l'interface Webmin-Htb qui ne correspond pas à un numéro de classe htb mais simplement au nom du fichier de config de cette branche.

Voici un exemple de visualisation de traffic :

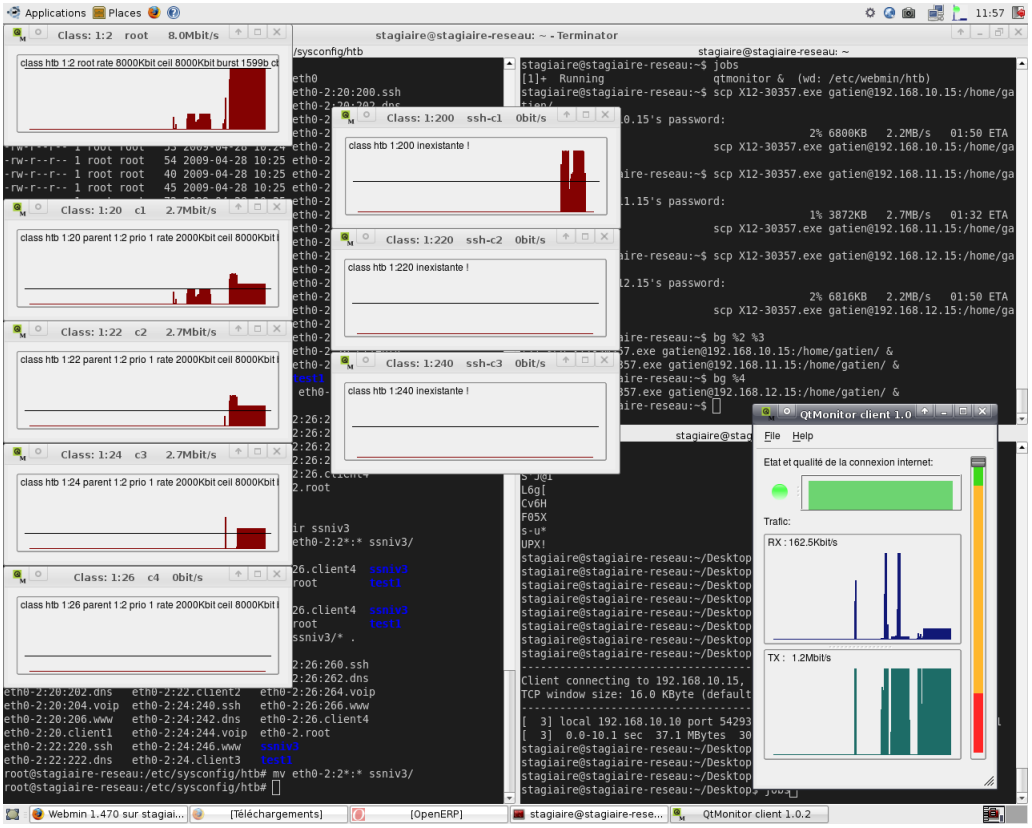

FIG. 4.3 – Aperçu du traffic dans les différentes classes définies

4.3 Installation d'un routeur pour A3F

Configuration d'un routeur pour a3f avec la méthode actuelle et l'aide de Laurent afin de me familiariser avec cette façon de faire.

Pour ce faire, j'ai commencé par installer une machine dans le labo sur laquelle j'ai effectué les tâches suivantes :

- Installation d'une 2ème carte réseaux
- Partitionnement du disque (/ et swap)
- Copie de l'image routeur sur la machine depuis un serveur NFS (Romulux2)
- Personnalisation de la machine (interfaces, partitions, nom, bootloader, ...)
- Mise à jour de l'outil d'administration Webmin
- Configuration des services nécessaires :
  - DHCP[5]
  - DNS[6] forwarding vers la passerelle internet
  - Tunnels OpenVPN[7]
  - Parefeu

Une fois cette machine opérationnelle, je me suis rendu dans les locaux de cette entreprise avec Laurent afin de mettre en place et en production ce routeur. Nous avons donc installé cette machine chez le client, nous avons terminés la configuration de ce routeur par rapport à leur structure réseau puis nous avons restructuré une partie de leur réseau afin de faciliter la maintenance. Ce routeur servant également de serveur DHCP en interne, il a également fallu renégocier l'adresse IP de chaque poste afin qu'ils soient reconnus par le nouveau serveur mais également afin de vérifier la validité des paramètres fournis par ce DHCP.

Une fois la configuration sur place terminée, nous vérifions que ce poste est bien accessible depuis notre Intranet pour pouvoir terminer la mise en place des règles de parefeu à distance.

[5] DHCP : Protocole de configuration dynamique des hôtes
[6] DNS : Serveur de nom de domaine
[7] OpenVPN : Logiciel de paramétrage de réseaux privés virtuels

4.4 Création d'un script d'installation automatisé de routeur

Afin de faciliter la mise en œuvre de routeur qui s'avère relativement compliquée au premier abord étant donné le nombre d'éléments à modifier et l'ordre dans lequel ceux-ci doivent l'être, j'ai crée un script permettant d'automatiser au maximum cette installation.

L'installation d'un routeur se fait en 3 étapes à partir d'une image disponible sur l'un des serveurs de la société.

Il faut tout d'abord partitionner le disque dur du routeur de sorte à avoir une partition / marqué comme bootable et une partition de swap. Celà se fait à l'aide d'un live-cd et de la commande fdisk.

Une fois le partitionnement effectué et la machine redémarrée, passons aux choses sérieuses. Il faut maintenant rapatrier l'image disponible sur le serveur vers le disque local. Celà se fait via une commande rsync une fois le système démarré sur un live-cd et la partition / montée dans l'arborescence.

ex : `rsync -vax romulux2::bu/Installations_type/routerx_8.04/ /a`

Une fois la copie terminée, il faut personnaliser les paramètres du routeur. Il faut donc commencer par copier les devices vers le /dev de la partition montée puis se *chrooter* dans cette partition afin d'y effectuer les modifications nécessaires.

Parmi ces modifications, la correction de grub, la suppression des interfaces réseaux (qui seront régénérées), la mise à jour de fstab et mtab et éventuellement la création d'utilisateurs supplémentaires.

C'est donc pour faciliter cette opération que j'ai écrit un script qui doit être exécuté depuis un live-cd après partitionnement et qui permet de faire toutes les opérations depuis le montage de la partition jusqu'à la création d'un utilisateur en passant par toutes les étapes intermédiaires.

Après 2 heures de tests et de modification, ce script s'avère fonctionnel et permet donc l'installation d'un routeur depuis l'image disponible de manière bien plus efficace.

Ce fameux script est disponible en annexe 7.1.

4.5 Création de tunnels VPN

Dans la configuration des routeurs intervient également la configuration des interfaces réseaux et notamment des tunnels. Leur mise en place se fait via *OpenVPN* lorsqu'il s'agit de créer des tunnels permanents entre différents sites ou avec les clients mais il est également possible, dans le cas où un tunnel temporaire peut permettre de faciliter un dépannage, de créer des tunnels rapidement via *SSH*[8] et son option `-w` qui permet la création de VPN par ssh depuis la version 4.3.
Cependant, il faut modifier la configuration du serveur, il faut modifier le fichier /etc/ssh/sshd_config et y définir :

```
PermitRootLogin yes
PermitTunnel point-to-point
```

Il faut ensuite redémarrer le service ssh :

```
/etc/init.d/ssh restart
```

Exemple :

ssh -w 0 :0 root@toto.plan-net.lu

Cette simple commande permet de créer un tunnel entre la machine locale et toto.plan-net.lu

Cependant cela ne suffit pas, si l'on veut vraiment créer un tunnel exploitable en IP, il faut lancer 2 commandes :

```
ssh -w 0:0 root@toto.plan-net.lu ifconfig tun0 192.168.200.1 \
pointopoint 192.168.200.2
```

```
ifconfig tun0 192.168.200.2 pointopoint 192.168.200.1
```

Ces 2 commandes permettent de créer un tunnel sécurisé et d'attribuer les paramètres IP nécessaires à chacune des interfaces de celui-ci.

Lorsque l'on met en place un tunnel, notamment avec OpenVPN, il faut bien prendre en compte les divers éléments à traverser et notamment s'il y a des règles de NAT. En effet, un tunnel VPN peut être, une fois établi, soit en UDP soit en TCP. Dans la majeur partie des cas, ceux-ci seront en UDP afin de pouvoir utiliser des programmes fonctionnant en UDP ou en TCP dans ce tunnel. Dans le cas ou de la NAT est en place, afin de permettre au informations de circuler, le tunnel doit être établi en TCP. Cela pose enormément de problèmes lorsque du traffic TCP passe dans celui-ci puisqu'en cas de perte de paquets, ceux du tunnel et ceux de l'application seront renvoyés jusqu'à ce que le destinataire les reçoivent, ce qui consomme donc énormement de bande-passante inutilement.

Un tunnel OpenVPN permet d'établir une connexion directe entre 2 postes à travers Internet. Cela permet d'établir des réseaux Intranet en utilisant les connexions Internet de chaque site. Le principe d'établissement de la connexion OpenVPN se fait sur un échange de type clé partagée.

Pour mettre en place un tunnel avec OpenVPN, il faut créer un fichier de configuration (dans /etc/openvpn) puis créer une clé partagée via la commande :

[8] SSH : Shell sécurisé permettant de se connecter à distance à un poste

openvpn –genkey –secret nom_de_la_cle.pem

Il faut ensuite créer le fichier de configuration correspondant sur l'autre poste et y copier la clé partagée. Il ne reste plus qu'à éditer la configuration du parefeu afin d'autoriser cette connexion dans le(s) sens souhaité(s).

Une fois le tunnel en place, il est possible d'atteindre directement la machine et/ou le réseau distant via son adresse à travers le tunnel.

4.6 Installation d'un routeur pour une clinique

J'ai également participé à la mise en place d'un routeur dans une clinique à Malmedy en Belgique. J'ai tout d'abord préparé ce routeur en utilisant mon script d'automatisation de l'installation qui après quelques retouches, permet le déploiement rapide de l'image sur le routeur en y apportant toutes les modifications nécessaires pour obtenir un système stable et prêt à être utilisé.

Ce routeur sert à mettre en place une liaison directe entre deux établissements via un tunnel VPN que nous avons établis. Nous avons également configuré le parefeu de ce routeur à l'aide de FWBuilder afin de n'autoriser que ce traffic à travers le routeur. Cela pourra être modifié à tout moment à distance en fonction des demandes du client puisqu'un VPN avec Plan-Net à également été mis en place. Cela permet d'éviter les 2 heures de route afin d'effectuer des opérations de maintenance.

4.7 Configuration d'un second accès Internet à Belval

Mise en place d'un deuxième accès internet au cinéma de Belval avec maintient/rétablissement des connexions lors du basculement entre les 2 accès.

Dans ce cinéma, environ 60 monolithes (pc sans ventilateur et de taille réduite) gèrent des écrans affichant des bandes annonces, des pubs ou encore des affiches de film. Ces monolithes maintiennent une connexion ssh constante avec un serveur basé à Londres afin de permettre leur gestion via un reverse ssh mais également afin que ceux-ci récupèrent automatiquement, chaque nuit, la playlist à jouer à l'écran jusqu'à la prochaine mise à jour.

L'accès internet redondant étant nécessaire dans cette infrastructure, j'ai participé à sa mise en place avec Brent. Nous avons donc configuré une connexion pppoe comme route principale, permettant ainsi d'utiliser l'autre route directement lorsque la connexion pppoe *tombe* puisque l'interface virtuelle qui lui est associée disparaît ainsi que la route correspondante. L'autre route ayant un metric plus élevé, dans le cas où la connexion pppoe reparaît, c'est à nouveau cette route qui sera utilisée.

Le problème, c'est que la mise en place de cette connexion ne suffit pas. Il faut également modifier les règles du parefeu associé et effectuer des modifications sur les monolites et le serveur de Londres afin de maintenir les connexions ssh en cas de basculement.

Nous avons donc ajouter les lignes suivantes à la configuration du serveur et des monolites :

- `TCPKeepAlive yes      # autoriser les connexions persistantes`

- `ServerAliveInterval 15   # vérifier l'état de la connexion`

La mise en place de ces lignes sur le serveur de Londres n'a pas posé de difficultés puisque celui-ci était accessible directement via ssh. Pour la configuration des monolites, en revanche, étant donné que le deuxième accés a été mis en place avant la découverte du problème, toutes les connexions ssh ont été perdues or les monolites sont identifiés d'après leur adresse MAC sur le serveur de Londres.

Étant donné l'impossibilité d'accéder à ceux-ci depuis ce serveur, nous avons scanné la plage ip correspondant aux monolites, via nmap, afin de tous les lister et de les mettre à jour via un script.

Nous avons donc écrit ce petit script qui demande à tous les monolites d'aller se mettre à jour à partir d'un partage prévu pour eux sur le serveur londonien.

Une fois la mise à jour effectuée, les monolites ont été redémarrés de la même maniere puis ils se sont reconnectés au serveur londonien et acceptent désormais les changements d'accès internet en récupérant leur connexion ssh au plus tard 15 secondes après le basculement.

Un autre problème est survenu plus tard, n'ayant aucun lien avec notre modification mais le problème étant très facilement reproductible, je trouve intéressant de l'expliquer ici. En fait, en écoutant le traffic sur l'interface externe via un TCPdump, nous avons découvert des paquets qui ne sont pas transférés correctement. Le problème via en fait du fait que ces paquets transitent sur un tunnel VPN à travers une interface ppp0 elle même basée sur eth1 or lorsque l'on essaie d'effectuer un ping avec un paquet supérieur à 1300 bytes, ce paquet est rejeté.

Il faut regarder du côté du paramétrage de ces interfaces. En effet, si l'on regarde la valeur MTU[9] nous pouvons observer :
eth1 : 1500 (valeur par défaut d'une connexion ethernet)
ppp0 : 1408 (1500 - données de l'interface eth1 - données paquets ppp - valeur recommandée)
tun0 : 1500

Comme nous pouvons le constater immédiatement, il y a un problème puisque le MTU du VPN est plus important que celui de la connexion sur laquelle celui-ci s'établit. Il suffit donc de rajouter dans le fichier de configuration d'OpenVPN, l'option `tun-MTU 1300` afin de résoudre ce problème.

4.8 Équipement spécialisé utilisé en tant que routeur

Bien que non prévu au départ, la demande d'un client d'un équipement spécifique résistant à des conditions extrêmes de température, m'a permis de découvrir et tester un système embarqué, un IA-240-LX du constructeur Moxa qui résiste dans sa version T à des températures variant de -40 ° C à +85 ° C pour un prix d'environ 500€. Ce client voudrait utiliser ce type d'équipement en tant que routeur placé à l'extérieur d'un bâtiment.

Il s'agit d'un boitier comportant diverses entrées/sorties spécialement conçues pour l'industrie mais également de 2 port Ethernet, d'un port USB et s'un slot SD. Ce matériel est

[9] MTU : Maximum Transmission Unit, nombre de bit maximum par paquet

basé sur une architecture RISC[10] à base de processeur ARM.

Nous avons donc commandé une de ces machines afin de savoir exactement quelles sont ses possibilités. J'ai ainsi établi, en plus des éléments fournis dans la notice constructeur, une documentation sur les possibilités de cette machine. Étant donné la version relativement ancienne du kernel et le fait que l'on ne peut mettre à jour ce système qu'à travers une mise à jour du firmware, celui-ci regroupant le BIOS et le système d'exploitation et n'étant fourni par le constructeur qu'en fichier binaire, nous avons contacté le fabricant afin de savoir comment créer notre propre firmware. Nous savons donc après cette communication qu'il est bien possible de créer un firmware personnalisé cependant, n'ayant pas reçu tous les éléments nécessaires dans un délai restreint de la part du constructeur, je n'ai pas pu m'essayer à la création d'un firmware pour cet équipement.

J'ai cependant rédigé sur le forum de la société, une liste de caractéristiques concernant cette machine comprenant notamment les versions des logiciels installés, les capacités maximales de cartes SD[11] utilisables pour ce modèle soit 1Go ainsi que les points de montages des périphériques et la configuration réseau originale de l'appareil.

Voici l'équipement en question :

FIG. 4.4 – Équipement routeur : Moxa IA-240-LX

Après plusieurs contacts avec le revendeur afin d'obtenir plus d'information sur la possibilité de créer notre propre firmware, nous avons obtenu une réponse qui ne nous convient pas puisque Moxa ne fournit que le code source du *kernel* afin de permettre le développement de drivers pour cet équipement mais ne distribue pas la méthode permettant de générer un firmware pour cet appareil avec un kernel modifié car celui-ci est propriétaire et ils ne souhaitent pas le rendre publique ou même le vendre.
Nous avons donc du nous résoudre à ne pas utiliser cet équipement.

[10] RISC : Reduce Instruction Set Computer, ordinateur à nombre d'instructions réduites
[11] SD : format de carte mémoire flash

4.9 Transfert de la configuration des routeurs vers FWBuilder

Bien que l'on m'ai donné comme sujet principal, l'intégration de fonctionnalités de QoS au sein de routeurs, une fois ma documentation sur la manière de mettre en œuvre cette qualité de service, j'ai appris par la suite que pour l'instant seuls les parefeux des routeurs des clients sont crées sur FWBuilder or comme il m'a été demandé de baser au maximum la gestion de la QoS sur FWBuilder après que j'en ai expliqué les possibilités de la dernière version, j'ai donc du réecrire les parefeux des routeurs de la société afin de permettre par la suite la mise en place de QoS sur ceux-ci.

Le fait de devoir réecrire ces règles de parefeux de SuseFirewall2 [12] vers FWBuilder m'a pris beaucoup de temps mais m'a aussi permis de comprendre plus profondément le fonctionnement de FWBuilder qui même s'il semble facile à utiliser au premier abord se révèle relativement complexe lorsqu'il s'agit de transférer une configuration d'un parefeu comportant une trentaine de tunnels OpenVPN ainsi que plusieurs interfaces réseaux avec la gestion d'une DMZ et parfois de plusieurs réseaux distant derrière le même tunnel.

J'en ai ainsi appris beaucoup sur les règles de parefeux, la façon de les définir et l'importance de l'ordre dans lequel ces règles sont définies.

La configuration d'un parefeu SuseFirewall2 se fait dans un fichier à éditer directement avec un éditeur de texte. Ce fichier comporte des règles qui ne sont pas toujours très bien documentées, comportant même des erreurs dans les explications originales présentes au dessous des règles (notamment concernant la NAT).
La configuration d'un parefeu sous FirewallBuilder en revanche se fait en mode graphique et offre plus de possibilités que les fichiers de SuseFirewall2. Cependant, bien que l'interface soit assez agréable, elle manque encore d'ergonomie notamment au niveau du *drag&drop* pour placer les éléments sur la grille de règles et l'impossibilité d'utiliser des raccourcis clavier pour copier - coller des éléments d'une règle à une autre ou encore de sélectionner plusieurs éléments dans les règles en place.

Ce travail m'a donc pris beaucoup de temps, ce qui ne m'en a pas laissé suffisament pour mettre en œuvre réellement de la QoS sur ces routeurs.
J'ai cependant laissé une notice explicative sur la façon de mettre en œuvre cette QoS sur le forum de la société.

Voici un exemple de règles crées avec FWBuilder :

[12] SuseFirewall2 : Logiciel de configuration de parefeu de la distribution OpenSuse

FIG. 4.5 – Exemple de règles crées sur FWBuilder

L'énorme avantage de FWBuilder ne réside pas réellement, selon moi, dans l'interface graphique mais essentiellement dans la possibilité de gérer un ensemble de parefeu depuis un seul fichier généré par FWBuilder et contenant l'ensemble des réseaux et machines utilisées dans les règles des différents parefeux.

Les réseaux, routeurs, postes, ... sont ainsi crées dans une bibliothèque qui permet leur utilisation au sein de plusieurs parefeu qui peuvent être de nature différente (Systèmes d'exploitation, architectures, ... différents).

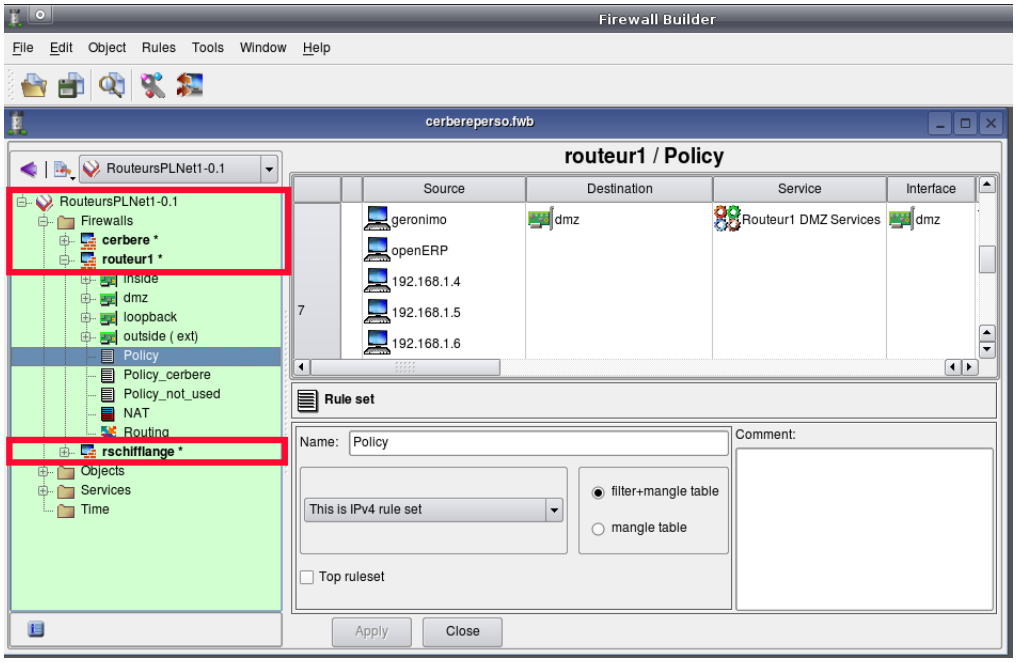

FIG. 4.6 – Exemple de fichier FWBuilder contenant plusieurs parefeux

Nous pouvons voir sur cette capture plusieurs parefeux configurés dans le même fichier objet. De plus le fichier créer par FWBuilder, avant la compilation qui crée un script éxécutable, est un fichier *XML* [13] respectant une *DTD*[14], fwbuilder.dtd ce qui permet en cas de besoin de créer un autre programme pour traiter des cas particuliers ou tout simplement pour s'abstraire à l'interface graphique de FWBuilder.

4.9.1 Travail effectué avec FWBuilder

J'ai crée via FWBuilder, un fichier d'objets comprenant la configuration de plusieurs des parefeux de la société ainsi que la définition des hôtes, réseaux et services accessibles depuis la société. Depuis cette configuration, il est ainsi possible de mettre à jour les différents parefeux du site.

Lors de la première tentative de migration, il y a eu un problème, le parefeu bloquait tout même les communications en local (ping localhost) et renvoyait *Operation not permited*. Ce problème étant bloquant et n'ayant pas trouvé de solution évidente, nous avons provisoirement rétablis l'ancien routeur. Après plusieurs heures passés à revérifier la configuration dans FWBuilder, à passer en revue les règles et toutes les options et après de veines recherches sur Internet, j'ai fini par trouver la solution, il s'agissait de cocher la case *top ruleset* associé à l'objet règles du parefeu, sans cette option activée, celui-ci refuse tout avant même d'ajouter les règles. Les règles sont donc présentes mais ignorées.

4.9.2 Migration effective du routeur principal

Lors de la deuxième tentative de migration, tout s'est déroulé sans problème, les deux ou trois éléments manquants ont très vite été mis en place. Il s'agissait en fait de quelques ex-

[13] XML : Extensible Markup Language, language de balisage extensible
[14] DTD : Document Type Définition, fichier donnant la structure à suivre d'un fichier XML

ceptions qui ont été enlevées durant la mise au point du parefeu puisque la plupart étaient devenues inutiles.

Une fois en place, j'ai surveillé les logs à la recherche d'éventuel traffic bloqué alors qu'il devrait être autorisé. Cela m'a permis de découvrir quelques légers problèmes qui ont été résolus très rapidement en modifiant les règles de filtrage ou de NAT correspondantes. La migration s'est donc effectuée en douceur puisque le routeur a été remplacé à 7h30, les accés Internet, les communications internes et les VPNs ainsi que la DMZ ont fonctionnés immédiatement. Les problèmes apparus au cours de la matinée ont tous été résolus en quelques minutes.

4.10 La détection d'intrusion

En fin de stage, je me suis également intéressé à la détection d'intrusion, puisqu'en matière de sécurité, le parefeu n'est pas suffisant.

Dans ce domaine, plusieurs types d'outils existent : Les NIDS et les HIDS.

Les NIDS (Network Based Intrusion Detection System) surveillent l'état de la sécurité au niveau du réseau et les HIPS (HostBased Intrusion Detection System) surveillent l'état de la sécurité au niveau des hôtes. Il existe également des IDS hybrides mais ceux-ci présentent peu d'intérêt.

Étant donné la structure réseau de l'entreprise et les orientations de ces différents types d'IDS, nous allons nous orienté vers un NIDS.

Voici les NIDS les plus utilisés :
- Snort :
  Snort est un NIDS sous licence GNU GPL qui permet de générer des alertes lorsqu'une intrusion est détectée.
  Celui-ci peut être couplé à BASE (Basic Analysis and Security Engine) qui est une interface web permettant de gérer des alertes générées par Snort en les organisant et en produisant des diagrammes.
- Bro :
  Bro est un NIDS conçu et maintenu par des centres de recherches, celui-ci se base sur l'analyse du flux réseau, ce qui permet de concevoir une cartographie du réseau et d'en générer un modèle qui sera comparé en temps réel au flux afin de lever une alerte lorsque la déviance entre les deux est trop importante.
  Cependant Bro à un gros problème, celui-ci ne dispose pas d'outils graphiques pour le paramétrer ce qui explique son utilisation uniquement dans les milieux universitaires.

4.10.1 Les techniques de détection

- Vérification de la pile protocolaire :
  Cela permet de détecter un grand nombre d'intrusions basées sur des violations des protocoles IP, TCP, UDP et ICMP.

- Vérification des protocoles applicatifs :
  Cela permet de détecter des intrusions basées sur des violations de protocoles applicatifs tels que NetBios (exemple : WinNuke qui utilise des données NetBios invalide).

- Reconnaissance par *Pattern Matching*
  Cette méthode se base sur la recherche de signatures d'attaques connues.

4.10.2 Les différentes actions réalisées par des IDS

- Reconfiguration d'équipement afin de bloquer l'intrusion
- Envoi d'une alerte (trap SNMP, envoi de mail, ...)
- Journalisation de l'attaque
- Sauvegarde des paquets suspects
- Lancement d'une application
- Envoi d'une fin de connexion (s'il s'agit d'une attaque sur TCP)

– Notification de l'alerte dans une console

Un outil tel que Snort est très puissant, cependant, son paramétrage n'est pas forcément évident mais ne doit pas poser énormément de difficulté au vu de la documentation existante sur cet outil. En revanche, le principal problème des NIDS est une énorme consommation en ressources systèmes. Les routeurs installés par Plan-Net étant des machines avec de faibles ressources (256Mo de RAM), il n'est pas envisageable de mettre en œuvre Snort sur ces machines sans quoi leur rôle premier serait rendu bien plus difficile.

Il est également possible de placer un NIDS sur une autre machine dans le réseau, cependant, dans le cas de l'installation chez un client, il n'est pas possible de proposer un routeur à 100 euros et un autre poste beaucoup plus cher pour la détection d'intrusion.

4.10.3 Avantages et inconvénients

Avantages :
– Alerte en cas d'intrusion
Inconvénients :
– Nécessite beaucoup de ressources, une machine puissante
– Coût de la machine » 100€

4.11 Tâches en rapport indirect

Durant ce stage, j'ai également eu la chance de découvrir de nombreuses autres choses. J'ai notamment assisté à diverses interventions chez les clients au cours desquelles j'ai énormement appris.

Voici un récapitulatif de quelques interventions auquels j'ai participé :

– Création d'un raid5 sur 3 disques pour le serveur principal et sa réplique sur 3 autres disques pour le backup puis rapatriement des données depuis l'ancien serveur de backup pour un client.
– Résolution d'un problème de synchronisation d'une ligne adsl.
– Mise au point de tunnels VPN entre clients.
– Vérification de la fiabilité d'une connexion ADSL.

Dans le cadre de mon stage, j'ai également participé, en tant qu'utilisateur, au test de l'application de gestion open source *OpenERP*. J'ai utilisé cette application afin de gérer mon temps de travail plus efficacement.

Chapitre 5

Conclusion

J'ai trouvé ce stage extrêmement intéressant dans la mesure où l'on m'a confié des tâches dans mon domaine de formation, c'est à dire des tâches d'administrateur système utilisant des logiciels libres tels que FWBuilder, Webmin et bien d'autres sur un système d'exploitation libre (Ubuntu). J'ai également apprécié dans ce stage la communication au sein de l'entreprise qui m'a permis d'élever mon niveau de connaissance via la transmission d'informations sur les problèmes rencontrés chez différents clients ainsi que la façons d'effectuer un diagnostique où encore les solutions mise en œuvre pour palier aux divers problèmes rencontrés.

Les différentes tâches qui m'ont été confiées, la recherche sur la QoS, la migration des routeurs, l'automatisation de leur installation, l'expérimentation de matériel industriel ou encore quelques interventions chez les clients en compagnie de Laurent ou Brent m'ont réellement intéressés. L'étendue des tâches bien que centrées essentiellement sur les routeurs, m'a permis de mettre en œuvre et d'acquérir de nombreuses compétences.

De même, le fait de tenir à jour une liste de tâches et de projets via OpenERP permet d'améliorer son organisation au niveau du travail et de ne pas oublier certaines tâches réalisées ou à réaliser afin de pouvoir faire un rapport convenable et optimiser son temps de travail.

En ce sens, j'ai trouvé ce stage très enrichissant au niveau compétences comme au niveau humain. J'ai ainsi pu découvrir que la communication est l'élément fondamental au sein d'une entreprise afin d'assurer sa pérénité.

Chapitre 6

Bibliographie

6.1 Généralités

- doc.ubuntu.com
- wikipedia.org
- Les pages de "man"
- Les ressources internes à l'entreprise (forum, ...)

6.2 La QoS

- QoS IPtables : http://www.coredump.fr.to/qos-linux-iptables-limite-debit/2/
- QoS : http://ferry.eof.eu.org/lesjournaux/pg/public_html/x6624.html
- Webmin-Htb : http://www.sehier.fr/webmin-htb/
- Webmin : http://www.webmin.com/

6.3 Les parefeux

- FWBuilder : http://www.fwbuilder.org/
- Netfilter, IPtables : http://www.netfilter.org/

6.4 Trucs & astuces

- Utilisation de Vim : http://ceyquem.free.fr/www/articles/misc_vim/vim.htm
- Utilisation de Sed : http://www.commentcamarche.net/faq/478-sed

Rapport en LaTeXintégralement rédigé sous Vim par Gatien GASPARD en juin 2009

Chapitre 7

Annexe

7.1 Script d'installation de routeurs

```
#!/bin/bash
echo Installation d'un routeur Plan-Net automatisée (en root sur live-cd):
echo Installation avec seulement une partition / (sda1) et une de swap (sda2):
echo Les 2 partitions doivent déjà avoir été crées.
#
echo "Script d'installation de routeur automatisé"
echo "appuyer sur une touche pour continuer (ou CTRL+C pour annuler)"
# attente de vérification
read
#
echo "Désactivation du swap"
swapoff -a
#
#echo "Création des systèmes de fichiers"
#mkfs.ext3 /dev/sda1
#mkswap /dev/sda2
#
echo "Montage de la partition /"
mkdir /a
mount /dev/sda1 /a
#
echo "Copie de l'image depuis romulux2"
rsync -vax romulux2::bu/Installations_type/routerx_8.04/ /a
#
echo "Copie des périphériques dans le point de montage"
cp -a /dev/* /a/dev
#
#### echo Chroot dans la partition /
#### chroot /a

echo "
echo Configuration de grub
grub-install hd0 -y
# Equivalent
#grub
```

```
#commandes à taper dans ce shell
  #root (hd0,0)
  #setup (hd0)
  #quit
#
echo Mise à jour des identifiants du système avec ses périphériques
blkid
#
echo Mise à jour des informations de grub:
# editer grub (# kotp=root=/dev/sda1)
sed -i '/^# kopt/ s:.*:# kopt=root=/dev/sda1:g' /boot/grub/menu.lst
sed -i '/hd0,./ s:hd0,.:hd0,0:g' /boot/grub/menu.lst
sed -i '/root=/ s:root=.* :root=/dev/sda1 :g' /boot/grub/menu.lst
update-grub #(choisir la version du responsable du paquet)
#
echo Suppression des interfaces réseau existantes #(elles seront régénérées)
# vim /etc/udev/rules.d/70-persistent-net.rules #( supprimer les interfaces )
sed -i '3,$ d' /etc/udev/rules.d/70-persistent-net.rules
#
echo Mise à jour de mtab et fstab
# editer mtab et fstab
sed -i 's:/dev/.d..:/dev/sda1:g' /etc/mtab
sed -i '/ \/ / s:/dev/.d..:/dev/sda1:g' /etc/fstab
sed -i '/ swap  / s:/dev/.d..:/dev/sda1:g' /etc/fstab
update-initramfs -u
#
### Ne pas oublier de creer des users si nécessaire ###
# PlanNet est présent par défaut.
# echo Création de l'utilisateur stagiaire:
# adduser stagiaire (uniquemement pour le stage)
#
#echo Redémarrage de la machine
echo Vous pouvez redémarrer la machine
#reboot
" > /a/tmp/install_routeur.sh

# Rendre le script executable
chmod u+x /a/tmp/install_routeur.sh

echo "Chroot dans la partition /"
# Exécuter la suite des commandes dans un chroot
chroot /a /tmp/install_routeur.sh

echo Installation terminée
```