Mise en place d'une solution intégrée pour la sécurité du serveur SIGMA

Thomas Gouverneur

Le 23 mai 2006

Remerciements

Je remercie chaleureusement :

Monsieur Jean Marchal, pour m'avoir avoir accueilli dans son service durant la période de stage.

Monsieur Alassane-Ballé Ndiaye de m'avoir offert l'opportunité de m'intégrer à son pool transport et d'avoir pu réaliser un stage dans les meilleures conditions.

Monsieur Denys Mercenier, mon parrain d'école, pour son soutien et ses conseilles avisés.

Monsieur Christophe Morrone, mon maitre de stage, pour sa prise en charge et son soutien tout au long du stage ainsi que son aide précieuse lors de la rédaction de ce mémoire.

La direction et l'ensemble du corps professoral pour leur travail et leur soutien tout au long de l'année.

Enfin, je tiens à remercier tout les gens ayant participé de près ou de loin à ce stage, et particulièrement les personnes de #linuxbe.

Table des matières

Chapitre 1

Introduction

Sommaire

1.1 L'Université de Liège

La fondation de l'Université de Liège en 1817, à l'initiative du roi Guillaume 1er des Pays-bas, est l'aboutissement d'une longue tradition intellectuelle qui remonte aux origines de la Principauté. A partir du XIe siècle, sous l'impulsion des princes-évêques, les écoles liégeoises constituent, en effet, un pôle d'attraction pour les étudiants et les chercheurs qui viennent y conquérir leurs premiers grades ou, comme Pétrarque, exploiter les richesses des bibliothèques.

Si la réputation des écoles médiévales valut à Liège le nom de nouvelle Athènes, que dire de celle du Collège qui s'ouvrit en 1496, à l'emplacement même de l'actuel batiment central de l'Université, place du 20-Août. Les frères de la Vie Commune y furent les promoteurs d'un enseignement rénové : celui des humanités et les professeurs de milliers d'étudiants liégeois et même étrangers, comme le futur pédagogue Jean Sturm, qui fit souffler l'esprit du Collège liégeois à Strasbourg et, de là, dans un grand nombre de gymnases réformés.

A la fin du XVIe siècle, les jésuites remplaceront les frères dans leur propre maison. Celle-ci abritera ensuite, après la suppression de la Compagnie de Jésus, le Grand Collège en Ile, confié au clergé séculier par le prince-évêque Velbruck (prélat éclairé qui réorganisa en Académie anglaise le Collège des jésuites anglais, installé à Liège depuis 1614, et suscita tout un enseignement technique de haut niveau par la création de plusieurs écoles).

Le décret de Napoléon 1er du 17 mars 1808, portant sur organisation d'une Université impériale et désignant Liège comme siège d'une Académie comportant notamment une Faculté des Lettres et une Faculté des Sciences, est la première charte universitaire liégeoise. Ni les écoles médiévales, pourtant si renommées, ni le Collège des bords de Meuse, bien qu'il dispensat deux cours supérieurs, ni les écoles ouvertes sous Velbruck, ni même l'Académie anglaise, en dépit de son nom, ne peuvent être considérés comme des établissements universitaires. Liège doit son université au seul souverain des Pays-Bas dont elle dépendit jamais : Guillaume 1er sut se souvenir du passé prestigieux d'enseignement et de culture de la Cité ardente, quand il décida d'implanter une université d'état en terre wallonne.

Près de 200 ans après, même si elle s'est installée pour partie au Sart Tilman, l'Université de Liège, qui dépend maintenant de la Communauté française de Belgique, est toujours à la même place, au bord de la Meuse, au centre de ce qu'on appela longtemps l'Ile, Quartier latin de Liège depuis l'époque moderne.

L'Université de Liège n'a pas un long passé, mais elle a déjà une histoire et surtout de profondes racines. Est-il meilleur gage pour l'avenir, vers lequel elle est résolument tournée ? La vocation pluraliste de notre Alma Mater, son ouverture sur les réalités politiques, sociales et industrielles d'aujourd'hui et de demain, sa participation active à des programmes internationaux et européens de recherche fondamentale ou appliquée, sa présence dans toutes les actions communautaires en matière de mobilité des étudiants sont autant d'atouts pour que vive Liège, Université d'Europe.

1.1.1 Le Blason de l'Université de Liège

Les armoiries de l'Université de Liège lui ont été octroyées par le Roi Baudouin en 1967, à l'occasion du 150e anniversaire de l'Institution, à la requête de Marcel Dubuisson, Recteur, agissant en qualité de Président du Conseil d'Administration.

Les armoiries veulent rappeler "la participation plus que séculaire de l'Université à la vie de la Cité de Liège" et rattacher l'Institution à l'histoire du Sart Tilman, où l'Université s'est implantée en grande partie.

C'est ainsi qu'on trouvera sur le blason, en 1 et 4, le perron liégeois, entouré du L et du G, du mot LièGe, en or sur fond rouge. En 2 et 3, en rouge sur fond or, se détache un grill rectangulaire à cinq barres, avec une tige annelée, entourée de quatre coquilles Saint-Jacques.

Le domaine du Sart Tilman était jadis la possession de l'Abbaye de Saint-Jacques et de l'Abbaye de Saint-Laurent.Les coquilles de Saint-Jacques, que l'on retrouve dans les écus de l'ancienne Abbaye liégeoise, étaient portées par les pèlerins du Moyen Age, à Saint-Jacques de Compostelle.Quant au grill, il figurait sur une des bornes du domaine de l'Abbaye Saint-Laurent. Il symbolise le supplice du grill que le Saint dut subir en 258.

L'écu est surmonté par la couronne royale. Enfin, l'inscription latine "Universitas leodiensis" (Université de Liège), figure dans un bandeau, en dessous de l'écu.

1.2 Le service de l'ANAST

Architecture Navale

Génie Maritime
Navigation Intérieure et Maritime
Analyse des Systèmes de Transport

L'université est divisée en plusieurs services ; chacun d'entre eux possède un domaine de recherche très précis. De plus, des cours théoriques concernant chaque domaine sont proposés aux étudiants qui ont choisi d'approfondir leurs connaissances dans cette matière.

C'est ce service qui m'a accueilli dans ses locaux afin que je puisse mener à bien ma tâche.

Les différents domaines de recherche de l'ANAST :

- Construction navale.
- Navigation intérieure, navigation maritime et modélisation du trafic fluvio-maritime et intermodal
- Télématique appliquée à la gestion des infrastructures fluviales et du matériel de navigation
- Développement d'un logiciel intégré (CAO-IAO) en construction navale.
- Optimisation des structures navales et flottantes.
- Analyse technico-économique comparative entre modes de transport (+ intermodalité).
- Etablissement d'un plan de transport.
- Modélisation mathématique de prévision et d'affectation des flux de trafic.
- Modélisation des déplacements urbains.
- Techniques d'essais après optimisation en bassin de carènes et hydrodynamique navale.
- Exploitation des moyens de transport
- Analyse de la mobilité (facteurs explicatifs, évolution, etc.).
- Evaluation financière et économique de projets de transport
- Mise au point d¿outils d'aide à la décision basés sur une approche multicritères
- Mise au point de modèles mathématiques de prévision et d'affectation des flux de trafic
- Mise au point de modèles mathématiques d'analyse des comportements de mobilité
- Mise au point d'outils d'évaluation financière et économique de projets de transport
- Modélisation du trafic, gestion des flux de trafic urbain et analyse du transport public
- Gestion du transport

Les différents domaines d'enseignement de l'ANAST :

- Théorie du navire et des propulseurs.
- Construction, technologie et exploitation navales.
- Compléments de construction navale relatifs aux bateaux de petites dimensions
- Conception et installation des équipements électromécaniques de bord.
- Règlements de classification en construction navale
- Organisation des chantiers navals.
- Optimisation économique des navires.

- Méthodes modernes en génie maritime.
- Questions portuaires et côtières.
- Exploitation des moyens de transport.
- Analyse technico-économique des systèmes de transport.
- Etablissement d'un plan de transport.
- Techniques d'analyse des prévisions et affectations des flux de trafic.

Les différents partenaires mondiaux :

- L'Université de Sao-Paulo (USP, Brésil) ;
- L'Université Centrale de Quito (Equateur)
- L'Université Technique de Madrid (Espagne) ;
- L'Académie des Sciences de Pekin (Chine) ;
- L'Université Technique de Dalian (Chine) ;
- Le "Ninth Design and Research Institute" de Shanghai (Chine) ;
- VWS, "Versuchsantalt für Wasserbau und Schiffbau" de Berlin (Allemagne) ;
- Ecole Centrale de Nantes (France) ;
- L'Université "Centro Estadual de Ensino Tecnol. Paulo Souza", Sao-Paulo (Brésil) ;
- EBD, European Development Centre for Inland Navigation, Duisburg (Allemagne) ;
- L'Ecole Nationale Supérieure des Techniques Avancées, ENSTA, Paris (France),
- Université d'Osaka (Japon),
- IFREMER, Institut Français de Recherches pour l¿Exploitation des Mers, (France).
- Université Pattimura de Ambon (Indonésie) ;
- Faculté TIUTI de Curitibas, Parana (Brésil) ;
- Université Minas Gerais de Bello Horizonte (Brésil).

Expert ou conseiller pour les organismes suivants :

- Banque Mondiale.
- Banque Asiatique de Développement.
- FAO (Nations Unies).
- Banque Africaine de Développement.
- Administration Générale de la Coopération Belge au Développement.
- Société Régionale d'Investissement.
- OPI (Office de Promotion Industrielle).
- Commission Européenne (expert pour l'évaluation de propositions de recherche).
- Bureaux d'Etudes : Louis Berger (USA), Sodetec (France), Tractebel, Cadic, Bureau Seco, Girec, Société de Gestion et de Participation, Heacon, Vanachai Group (Thaîlande).
- Companhia Energetica de Sao Paulo (gérant des plus grands complexes hydro-électriques et écluses fluviaux du monde).
- Gouvernements étrangers (Equateur, Bangladesh, Indonésie, Argentine, Brésil, Pays-Bas, France).
- Région Wallonne (Ministère des Travaux Publics (Voies Hydrauliques), Ministère des Transports et de l'Aménagement du Territoire).
- Commission Eurégionale de la Politique Structurelle (transport en commun et transport de marchandises)
- CNRS, France.

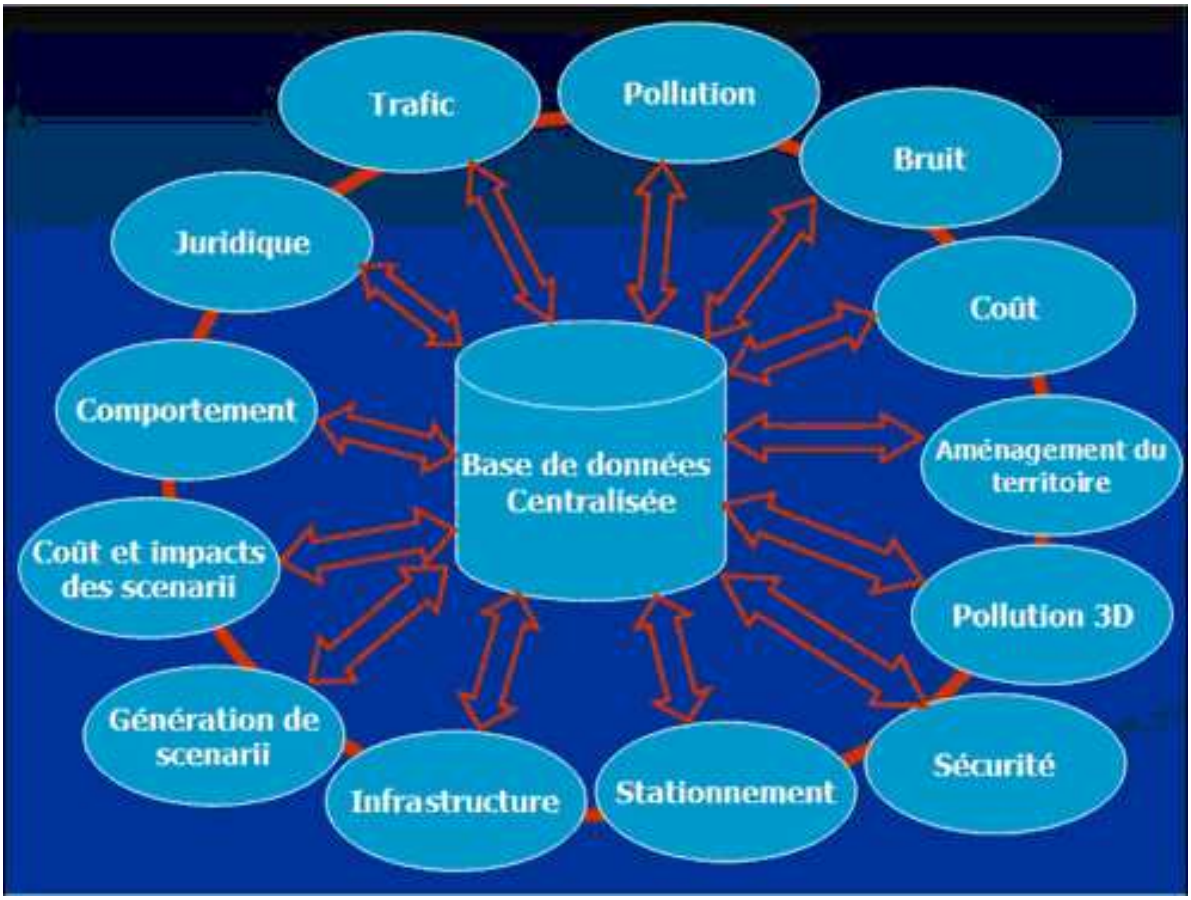
Logiciels développés :

- AIWAT, modélisation du trafic fluvial.
- ESTIMA, modélisation du choix du mode de transport.
- OLEMSE1, optimisation de la localisation des postes de secours.
- OLEMSE2, optimisation du choix de l'itinéraire en cas d'intervention.
- LUNAIS, logiciel intégré complet de CAO en construction navale adapté aux PME.
- Développement d'un module de gestion du trafic au droit d'un complexe d'écluses.
- Développement d'un module de gestion du trafic au droit d'un complexe portuaire.
- Développement d'un logiciel d'évaluation économique 'avantages/coûts'.
- Développement d'un logiciel d'analyse de marché 'attractivité-compétitivité'.
- LBR-4, dimensionnement et calcul des structures flottantes et navales.
- LBR-5, optimisation des structures flottantes et navales.
- Développement d'un logiciel d'aide à la décision multicritères.
- CCT-VEI, logiciel de calcul du coût de transport fluvial.
- Logiciel de calcul des coûts de transport d'une chaîne intermodale.

1.3 Le projet SIGMA

SIGMA est une application qui permet l'intégration et l'interopérabilité des divers modèles nécessaires à la gestion de la mobilité. Celle-ci permet le dialogue entre les modèles, leurs permettant ainsi de s'alléger de leur base de données. La bonne gestion de celles-ci étant prise en charge par SIGMA. Les modèles vont donc graviter autour de SIGMA comme de véritables planètes autour du système solaire. Les modèles garderons néanmoins toute leur autonomie et préserveront la confidentialité de leurs données respectives. SIGMA est également multicouches. L'aspect de 'partage' étant la base du SIGMA (première couche). Viennent également se superposer des outils d¿aides à la décision, des serveurs d'informations GIS et des clients permettant l'élaboration et l'étude de scénarii. SIGMA est un outil complet de gestion de la mobilité parce qu'il orchestre les meilleurs modèles existants de leurs domaines. La réalisation du modèle SIGMA pourra être considéré comme le noyau central dans la réalisation de n'importe quel PDU futur.

En cours de développement en section transport à l¿ANAST, Université de Liège.



Le serveur SIGMA tourne à présent sous une distribution `GNU/Linux` nommée `Gentoo`, et l'objet du stage est d'y intégrer tout un tas de logiciels ayant leur place dans le projet SIGMA, ainsi que de la sécurité approfondie de ce serveur.

Chapitre 2

Introduction à LaTeX

Preface

Le projet SIGMA nécéssite l'écriture d'un certain nombre de tutoriaux, ou documentation décrivant les tâches effectuées lors de l'intégrations de certains logiciels ou concepts sécuritaires. Celui-ci nécéssitait donc l'apprentissage d'un outil d'écriture professionnel tel que LaTeX.

Voici le tutorial ayant servi de base à la rédaction de toutes ces documentations...

Introduction

Ce tutoriel est écrit dans le cadre de mon stage de fin d'étude au sein de l'Université de Liège, département A.N.A.S.T.. Il à pour but d'expliquer pas à pas les opérations à effectuer pour obtenir un environnement de programmation LaTeX, il couvre également les premieres opérations élémentaires à la réalisation d'un document LaTeX et posera certaines bases dans la rédaction quotidienne de ceux-ci, il tentera de se poser en aide-mémoire utile au rédacteur débutant.

Sommaire

2.1 Pré-requis

LaTeX est prévu pour fonctionner avec de nombreux éditeurs, parmis ceux-ci on nottera l'éditeur très complet de la suite KDE, qui ravira les ferrus d'interfaces graphiques. Pour notre part, nous ne détaillerons que l'utilisation de la suite LaTeX avec l'éditeur ViM, d'ailleur utilisé pour la rédaction de ce document. Enfin, l'installation de ces outils sera détaillée pour convenir au système Gentoo GNU/Linux ; Toutefois, il est imaginable d'utiliser ces mêmes outils sur biens d'autres systèmes, les urls de référence seront fournies à titre indicatif et sont considérées comme valides à l'heure où ces lignes sont écrites.

2.1.1 Gentoo

Sous gentoo, nous veillerons à installer l'éditeur ViM et plus particulièrement sa version graphique gViM qui permettra la navigation et la découverte de la suite LaTeX en quelques instants, rendant encore plus aisée son utilisation. Il nous faudra également installer la suite LaTeX déguisée sous le nom charmeur de `latexsuite`.

En bref : `emerge -av vim gvim latexsuite`

Après avoir effectué cette lourde tâche, vous voilà prêt à utiliser LaTeX avec l'environnement gViM, l'utilisation de celui-ci est détaillée ci après.

2.1.2 Les autres...

Pour les autres distributions, la suite latex pour ViM peut être trouvée sur Internet au rayon Google, "latex suite" devrais faire l'affaire. Voici toutefois l'url actuelle : `http://vim-latex.sourceforge.net/`. gViM est quant à lui souvent fourni avec tout système, en package ou compilables (ports, portage, ...). Si celà n'étais pas le cas, il est disponible online, en binaire ou sources.

2.2 Découverte de gViM

L'environnement gViM allie la puissance de l'éditeur ViM, ainsi qu'une interface graphique comportant des menus intuitifs, spécialement fournis avec la suite LaTeX. Commençons par lancer l'éditeur en lui spécifiant le nom d'un nouveau document LaTeX :

```
gvim test01.tex
```

Une nouvelle page vide de gViM ainsi que les menus de la suite devraient maintenant apparaitre. les éléments suivants : `TeX-Suite`, `TeX-Environments`, `TeX-Elements` et `Tex-Math` ; Ces derniers ayants des noms assez éloquents, nous laisserons le lecteur découvrir les fonctions de chacun d'eux.

2.2.1 Les templates.

Les templates sont des débuts de documents communs, elles permettent d'avoir un canevas de document, faisant office de base pour l'écriture de documents d'une même étoffe.

L'exemple suivant illustre la template utilisée pour la rédaction de ce tutoriel.

```
\documentclass[french,12pt]{report}
\usepackage[T1]{fontenc}
\usepackage[english]{babel}
\usepackage{indentfirst}
\usepackage[latin1]{inputenc}
\begin{document}
\selectlanguage{french}
\newcommand{\code}[1]{\foreignlanguage{english}{\texttt{#1}}}
<<< corps du document ici >>>
\end{document}
```

Explications :

- `\documentclass` defini les arguments à passer à tout les packages (langue, taille de font), il spécifie aussi le type de document.
- BABEL est le package de gestion de langues, c'est lui qui traduit par exemple "table of contents" par "table des matières".
- IDENTFIRST permet une identation de la première ligne d'un paragraphe comme c'est le cas courrament lors de documents en français.
- `\usepackage[latin1]{inputenc}` est indispensable à la gestion des caractères accentués dans nos documents LaTeX.
- NEWCOMMAND permet de redéfinir des commandes souvent usitées, ici la balise `\code` permettra d'afficher du code `de cette façon`...

2.2.2 Trucs et astuces dans ViM

Voici quelques petits "trucs et astuces" découverts lors de l'utilisation du package latex-suite, cette liste ne se veut pas exhaustive, mais présente les raccourcis et actions les plus accessibles et utiles lors de la composition d'un document LaTeX.

Macros de ViM

Certaines macro éxistent et aident à la rédaction du document, encore une fois, nous n'avons pas la possibilité de toutes les énumérer, en voiçi quelques-unes.

Symboles

- `\ldots` sera automatiquement inséré lorsque vous entrerez trois points de suite.
- Le couple `‘‘` et `’’` sera automatiquement couplé lors de l'utilisation de la touche guillemets.

Autres

- Lorsque vous devez saisir un bloc (p-ex : `\begin{.. ..}`), vous avez peut être remarqué l'ajout par ViM de `< + + >`. Cet ajout n'est pas anodin, il permet avec la combinaison de touche "`C-c j`"[1] de se rendre au prochain bloc ainsi précédement ajouté, et de supprimer celui-ci. Ainsi, le déplacement en fin de bloc est souvent rendu plus aisé, à condition de s'être bien sûr habitué à de tels "raccourcis"

Réduction/Agrandissement du code

Il est possible de rendre l'édition des documents LaTeX plus lisible et agréable gâce à des petits raccourcis permettant de "compresser" certaines parties du code. Comme un exemple vaut mieux qu'un long discours, placez-vous en mode normal sous ViM/gVim et choisissez un chapitre à "réduire", utilisez la combinaison de touche "`\za`", Vous devriez voir se réduire le chapitre en question, et apparaitre quelque chose comme :

```
+---- 10 lines: \subsection {. . . . .} --------
```

Pour retrouver le paragraphe dans sa forme normale, ré-itérez la commande sur la ligne magiquement apparue...

D'autres raccourcis dont le comportement est similiaire à quelques point près existent, nous n'allons toutefois pas tous les détailler, rien ne vous empêche de tester leur comportement par vous même ; "`\rf`" par exemple.

Compilation des documents LaTeX

La compilation est encore une fois une chose simple et enfantine gâce a deux raccourcis claviers prévu dans ViM et gViM.

- `\ll` permet de compiler le document LaTeX courant. Si la compilation de celui-ci se passe sans encombres, vous ne devriez pas voir de différences, mais c'est bien effectué, pour vous en convaincre, vous pouvez vérifier dans votre répertoire de travail si d'autres fichiers ont été créés (.log, .aux, .dvi, ...).
- `\lv` permet quant à lui la visualisation du même document, pourvu que vous disposiez du visionneur correspondant au format de sortie choisi.

D'autres commandes éxistent pour personnaliser la compilation et la visualisation automatique des documents LaTeX, vous pouvez les découvrir par vous même grâce à gViM et ses menus LaTeX graphiques, rendez vous dans `TeX-Suite -> Target`

[1] Control+C, puis J

`Format` qui vous permettra par exemple de changer le format de destination (pdf, dvi, ...).

2.3 Commandes LaTeX

Les commandes reprises ici seront les plus importantes et les plus nécessaires à l'élaboration d'un document LaTeX simple.

2.3.1 usepackage

Cette commande permet de déclarer les packages que nous utiliserons dans le document, il est également possible de passer des arguments à ces packages.

```
\usepackage[argument(s)]{package(s)}
```

2.3.2 input

`\input{file}` permet l'inclusion d'un fichier .tex au milieu du document, il est recommandé d'utiliser plusieurs fichiers pour l'élaboration d'un document LaTeX, généralement un par chapitre.

2.3.3 Structure du document

Pour structurer un document latex en plusieures parties, il éxiste plusieurs séparations logique :

- `\part`
- `\chapter`
- `\section`
- `\subsection`
- `\subsubsection`
- `\paragraph`
- `\subparagraph`

2.3.4 maketitle

La page comportant le titre du document peut-être générée automatiquement grâce à une commande LaTeX reprennant les informations bibliographiques. Voici un exemple de son utilisation.

```
\title{Mon titre}
\author{Prenom \textsc{Nom}}
\date{Le \today}
```

Pour la déclaration des informations bibliographiques, ensuite, à l'endroit ou doit apparaitre la page de titre :

```
\maketitle
```

2.3.5 Tableaux

Les tableaux sont choses assez aisée sous LaTeX, comme un petit exemple vaut mieux qu'un long discour, voici un exemple simple de tableau :

```
\begin{tabular}{|l|r|r|} \hline
 & Mouhs & Blebles \\ \hline
 Gwreg & 1 & 0 \\ \hline
 Frwan & 1 & 1 \\ \hline
 Wouf & 3 & 1 \\ \hline
\end{tabular}
```

Ce qui donne comme résultat :

	Mouhs	Blebles
Gwreg	2	2
Frwan	1	0
Wouf	3	1

Rien de bien sorcier, on notera que le caractere "&" vise à séparer les cellules horizontales, ainsi que "\\" effectue un retour à la ligne.

2.3.6 verb

La commande `\verb` permet d'échapper des commandes LaTeX, ainsi, tout ce qui se trouve après la commande et entre deux caractères '|'[2], sera affiché "tel quel".

2.3.7 Listes

bleh

2.3.8 Quelques caractères spéciaux...

Symbôle	LaTeX
...	`\ldots`
LaTeX	`\LaTeX{}`

2.3.9 Les images

Pour inclure des images dans nos documents, nous veillerons d'abord à transformer celles-ci en documents PostScript, pour se faire, beaucoup d'utilitaires éxistent, notamment Xv. Enregistrez tout simplement votre image au format `.ps`.

Dans LaTeX, un package viendra à notre secour pour inclure les graphiques :

`\usepackage[dvips]{graphics}`

Ensuite, il ne nous restera plus qu'à utiliser la fonction `\includegraphics{image.ps}` pour inclure l'image ou nous le désirons.

[2] On peut très bien utiliser un autre caractère tant que celui ci ne se retrouve pas dans le texte à échapper.

2.4 Lecture supplémentaire

2.4.1 Ouvrages de référence

Voici quelques uns des ouvrages de référence sur LaTeX :

- Joli Manuel Pour LaTeX (`http://edgard.fdn.fr/`)
- LaTeX Help (`http://www.emerson.emory.edu/services/latex/latex_toc.html`)

Chapitre 3

Hardened Firewall

Preface

Lors de la sécurisation du serveur hébergant le projet SIGMA, il nous faut penser à sa protection contre l'extérieur et le monde "dangereux" d'Internet. Pour ce faire, nous réaliserons une politique de `firewalling` complète, qui permettra au serveur de se protéger contre certaines attaques bien connues, ainsi que contre certaines technique permettant aux personnes interessées de se procurer des informations à propos de notre serveur.

C'est ce que nous voyons dans les pages suivantes...

Introduction

Ce tutoriel est écrit dans le cadre de mon stage de fin d'étude au sein de l'Université de Liège, département A.N.A.S.T.. Il démontre certains principes de sécurisation par le biais du firewall `iptables`. Nous démontrerons notament comment le rendre plus robuste façe aux divers scanport et attaques de type flood, DDoS, Smurfing qui figuraient dans le précédent travail effectué par Patrice Lenaers.

Sommaire

3.1 Rappels et définitions

3.1.1 Pre-requis

Il sera supposé dans ce document que l'administrateur connait déjà la base iptables ainsi que certaines notions des protocoles `TCP/IP`. Ce document ne décrit pas comment réaliser un firewall complet, simplement comment améliorer un firewall éxistant en parant certaines "attaques".

3.1.2 TCP/IP

Voici un bref rappel des flags TCP dont nous parlerons plus tard au cours de ce document.

- `SYN` : Indique la demande d'une nouvelle connection
- `ACK` : Sert d'accusé de réception pour un paquet précédement reçu.
- `FIN` : Indique la fin de la connection en cours.
- `RST` : Indique au destinataire de couper la connection brutalement.
- `URG` : Indique que le paquet est considéré comme urgent, et doit donc être transmit avant les autres.
- `PSH` : Ordonne au recepteur de vider tout ses `buffers`[1] ainsi que d'envoyer immédiatement toute donnée en attente.

3.1.3 Iptables

Iptables est un firewall avancé sous GNU/Linux, son utilisation peut considérablement agir sur l'aspect sécuritaire d'un serveur.

3.1.4 Scanport

Un `scanport` peut souvent être considéré comme la première source d'informations pour l'attaquant, le principe est simple, le programme interroge chacun des ports de la machine l'un après l'autre. C'est une vision simpliste bien sur, les programmes effectuants les `scanport` sont aujourd'hui beaucoup plus évolués et permettent d'obtenir diverses informations sur la machine de destination.

3.1.5 SYN-Flood

Les attaques de types `SYN-Flood` sont assez simple à mettre en oeuvre, leur principe est enfantin, il suffit[2] d'initier un nombre maximum de connexion avec la destination, sans jamais plus acquitter ces connexions par la suite.[3] Chaque paquet envoyé cause l'allocation d'une socket sur la machine destination, et remplis rapidement toutes les sockets[4] disponibles, et rend impossible les "vraies" connexions.

3.1.6 ICMP-Flood

Le protocole ICMP permet aux administrateurs réseau d'effectuer des tests et opérations diverses sur le réseau, comme par exemple la vérification du chemin emprunté

[1] Mémoire temporaire.
[2] Un moyen pratique concret sera abordé plus tard dans ce document.
[3] On envoie donc juste le premier paquet d'initiation de connexion contenant le flag `SYN`
[4] voie de communication.

par un paquet, il est également utile pour le routage en permettant de propager des redirections ou des changements de routes[5]. Son utilité la plus banale, le ping[6] peut également s'avérer dangereux. En effet, il est plus particulierement utile pour saturer la bande passante d'un serveur[7]. Cette technique est d'ailleurs fréquemment utilisée pour saturer les connexions de faible débit descendant[8].

[5] icmp-redirect
[6] consiste à vérifier si une voie de communication vers le client existe.
[7] En envoyant un nombre conséquent de paquets icmp de grande taille
[8] 56k, ADSL, Cable, ...

3.2 Constatation des problèmes.

Nous allons dans ce chapitre mettre en évidence certains problèmes dans la sécurité du firewall d'un serveur GNU/Linux, et nous tenterons d'apporter des solutions à ces problèmes au chapitre suivant.

3.2.1 Scanports

Le scanport est souvent la première démarche d'un pirate, et est donc par conséquent, le premier contact avec la machine. Il est donc évident que plus le scanport est rendu difficile, plus les informations à la disposition du pirate seront moindres.

Voyons tout d'abord le compte rendu d'un scanport sur un serveur dépourvu de `Firewall`.

```
dunno ~ # nmap -sS kriek
Starting nmap 3.83.DC13 ( http://www.insecure.org/nmap/ )
Interesting ports on kriek (WWW.XXX.YYY.1):
(The 1665 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
53/tcp   open  domain
1337/tcp open  waste
MAC Address: 00:0E:2E:XX:XX:XX (Edimax Technology Co.)
Nmap finished: 1 IP address (1 host up) scanned in 0.723 seconds
```

Nous remarquons déjà que ce scanport au risque d'être incomplet, s'effectue en moins d'une seconde...

Un scanport plus évolué peut donner encore plus d'informations :

```
 dunno ~ # nmap -sS kriek -p 1-65535 -sV -O
Starting nmap 3.83.DC13 ( http://www.insecure.org/nmap/ )
Interesting ports on kriek (WWW.XXX.YYY.1):
(The 65531 ports scanned but not shown below are in state: closed)
PORT      STATE    SERVICE VERSION
53/tcp    open     domain?
1337/tcp  open     ssh     OpenSSH 4.2 (protocol 2.0)
6881/tcp  filtered unknown
MAC Address: 00:0E:2E:XX:XX:XX (Edimax Technology Co.)
Device type: general purpose
Running: Linux 2.4.X2.5.X|2.6.X|
OS details: Linux 2.4.0 - 2.5.20, Linux 2.4.18 - 2.6.11
Nmap finished: 1 IP address (1 host up) scanned in 20.422 seconds
```

Après ce scanport plus détaillé, nous pouvons avoir plus d'informations sur le système tournant sur celle-ci, mais également sur les versions des programmes et kernel installés. La durée de ce scanport est encore une fois risible comparée à son éfficacité.

3.2.2 DDoS

Les `DDoS`[9] est l'attaque la plus banale sur internet, au sens ou c'est certainement la plus répandue. Elle peut être due à un bug dans un programme serveur, ou tout simplement essayer de surcharger celui-ci en lui envoyant le plus de demande

[9] Distributed Denial of Service

possible en un délai restreint. Il est évident que la source de ces attaques est multiple (distributed).

Voici une simulation d'attaque DDoS de type SYN-Flood sur le port 80 :

```
greg ~ # hping2 -S -V --rand-source -i u200 -p 80 192.168.0.191
using eth0, addr: 192.168.0.161, MTU: 1500
HPING 192.168.0.191: S set, 40 headers + 0 data bytes
^C
--- 192.168.0.191 hping statistic ---
6216 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

La constatation de cette attaque se retrouvera dans le netstat de la machine attaquée, pour plus de clarté, nous allons nous limiter à compter les connections avant et après l'attaque ainsi que de détailler une entrée de cette table, via le fichier \proc\net\ip_conntrack qui reflète l'état des connections :

```
/proc/net $ grep "dport=80" ip_conntrack | grep -c "dst=192.168.0.191"
40
/proc/net $ grep "dport=80" ip_conntrack | grep -c "dst=192.168.0.191"
5827
```

```
tcp 6 0 CLOSE src=2.1.191.156 dst=192.168.0.191 sport=1936 dport=80 use=1
```

On peut clairement voir l'effet de l'attaque par SYN-Flood sur les sockets allouées par le système. Leur état est bien évidement toujours fermé (CLOSED), mais elles sont belles et bien présentes en mémoire.

3.3 Premières résolutions.

3.3.1 SYN-Flood

Pour parer les attaques de type SYN-Flood, nous mettons en place des règles iptables visant à limiter le nombre de connexions initiées par secondes et par ip/réseau.

Voici les règles iptables et leur explications :

```
iptables -N syn-flood
iptables -A syn-flood -m hashlimit --hashlimit 2/sec
 --hashlimit-burst 4 --hashlimit-mode dstip
 --hashlimit-name synflood -j RETURN
iptables -A syn-flood -j DROP
iptables -A INPUT -m state --state NEW -p tcp --syn -j syn-flood
```

Tout d'abord, nous avons créé une nouvelle chaine, et nous y signalons que tous les paquets la traversant ne peuvent excéder 2 par seconde, avec un `burst` de 4. Cela signifie que nous autorisons en moyenne 2 nouvelles connexions par secondes, et ce nombre sera rechargé de un a chaque fois que cette limite n'est pas atteinte, le maximum étant de 4 (le `burst`).

Ces nombres ont été choisis dans l'optique d'un mini-serveur web, il est évident que ces nombres sont à fixer en gardant en tête qu'ils doivent être supérieur aux nombres de connexions initiées par les clients "normaux" se connectant sur le serveur.

Le mode assigné au module hashlimit est effectivement DSTIP, en effet, celui ci regardera tousles paquets ayant la meme ip de destination, peu importe le port et l'ip de provenance, ce qui englobera bien nos attaques de type `Distributed`.

Ensuite, nous spécifions dans la chaine INPUT, que tous les paquets avec le flag `SYN`, indiquant une nouvelle connexion (`state NEW`), doit absolument passer par la chaine syn-flood pour pouvoir passer au travers de la chaine `INPUT`.

Le nombre de connexions initié par `SYN-Flood` est maintenant clairement limité. Démonstration :

Ré-itérons l'attaque de tout à l'heure, mais cette fois avec notre système mis en place :

```
HPING 192.168.0.191: S set, 40 headers + 0 data bytes
--- 192.168.0.191 hping statistic ---
1056 packets tramitted, 0 packets received, 100% packet loss
```

```
sigmund net # grep "dport=80" ip_conntrack | grep -c "dst=192.168.0.191"
4
sigmund net # grep "dport=80" ip_conntrack | grep -c "dst=192.168.0.191"
17
```

On constate dors et déjà que les connexions ouvertes "pour rien" sont déjà bien moindre que les milliers de tout à l'heure.

3.3.2 Paquets invalides

Beaucoup de systèmes d'exploitation peuvent être reconnus de par leur réaction à certains paquets dits "invalides"[10] ; Voici un tableau récapitulatif de différents paquets possédants des couples de flags invalides.

[10] De ca quelques mois, Windows XP® réagissait assez mal a un paquet ayant la meme addresse pour source et destination.

SYN	ACK	FIN	RST	URG	PSH	ALL	NONE
X	X	X	X	X			
		X		X	X		
						X	
		X					
X	?	?	X	?	?		
X	?	X	?	?	?		
							X

Exemple explicatif : Il est logique de dire qu'une connection ne peut être ouverte et fermée en une opération, et donc qu'un paquet TCP composé du flag `SYN`[11] ainsi que du flag `FIN`[12] doit être considéré comme paquet invalide.

Nous pouvons maintenant convertir ce tableau en règles iptables :

```
iptables -A INPUT -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j DROP
iptables -A INPUT -p tcp --tcp-flags ALL FIN,URG,PSH -j DROP
iptables -A INPUT -p tcp --tcp-flags ALL ALL -j DROP
iptables -A INPUT -p tcp --tcp-flags ALL FIN -j DROP
iptables -A INPUT -p tcp --tcp-flags SYN,RST SYN,RST -j DROP
iptables -A INPUT -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP
iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP
```

D'autres règles éxistent pour identifier d'autres types de paquets invalides, par exmemple, si nous possédons des interfaces réseaux avec ip publiques, il est impensable de retrouver des paquets contenant des ip dites "locales". De tels paquets pourraient effectivement être forgés par un attaquant, pour par exemple outrepasser certaines règles du firewall s'appliquant à des addresses locales[13] ou l'interface n'est pas spécifiée[14].

Egalement les paquets ayant un état noté INVALID peuvent être rejetés, les paquets marqués de la sorte sont en fait des paquets qui ne correspondent à aucune connection en cours, et donc n'ont pas lieu d'être traités.

Nous effectuerons celà au moyen des règles suivantes :

```
iptables -A INPUT -m addrtype --src-type LOCAL -i eth0 -j DROP
iptables -A INPUT -m addrtype --dst-type LOCAL -i eth0 -j DROP
iptables -A INPUT -m state --state INVALID -i eth0 -j DROP
```

Attention toutefois a cette derniere règle, en effet, lors de l'utilisation de `VPN`[15], cette règle doit etre placée après la règle gérant le traffic de celui ci, ou, celle ci ne doit pas s'appliquer sur l'interface de ce `VPN`.

3.3.3 Lock Manager

Le `lock manager`[16], souvent situé sur le port 0 ne doit pas être accessible. Attention toutefois, si vous servez du NFS[17], il est possible de devoir changer le port

[11] `SYN` indique le début d'une connexion
[12] `FIN` indique la terminaison d'une connexion
[13] ex : serveur NFS accessible uniquement par l'ip 10.0.0.1.
[14] ex : `iptables -A INPUT -p tcp 22 -s 10.0.0.1 -j ACCEPT`
[15] Virtual Private Network
[16] Gestionnaire de verrous, en français.
[17] Network File System, partage de fichier via le réseau

de celui-ci, ainsi que de le rendre accessible aux clients NFS.

```
iptables -A INPUT -p tcp --dport 0 -j DROP
iptables -A INPUT -p tcp --sport 0 -j DROP
iptables -A INPUT -p udp --dport 0 -j DROP
iptables -A INPUT -p udp --sport 0 -j DROP
```

3.3.4 Scanport, deuxième essai

Après ces quelques adaptations du firewall, il est temps de relancer un scanport pour vérifier la différence :

```
greg ~ # nmap -sS -sV -O sigmund -p 1-65535
Starting nmap 3.83.DC13 ( http://www.insecure.org/nmap/ )
Insufficient responses for TCP sequencing (2),
OS detection may be less accurate
(The 65532 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 4.2 (protocol 2.0)
80/tcp    open  http    Apache httpd
MAC Address: 00:11:2F:98:4B:EB (Asustek Computer)
Device type: general purpose
Running: Linux 2.4.X|2.6.X
OS details: Linux 2.4.19 - 2.4.20, Linux 2.4.20 - 2.4.28,
Linux 2.4.21 (x86 SuSE), Linux 2.4.6 - 2.4.26 or 2.6.9,
Linux 2.6.10, Linux 2.6.5 - 2.6.11
Nmap finished: 1 IP address (1 host up) scanned in 65812.221 seconds
```

On constate tout d'abord que la durée du scanport est atteinte par les modifications effectuées tout au long de ce document, ce qui est grandement[18] du à la vitesse à laquelle nmap procède aux scanports, en effet, notre protection anti `SYN-Flood` "récolte" beaucoup de tentatives du scanner.

Nmap nous indique aussi qu'a cause d'un manque de réponses aux séquences TCP, que l'`Os-Fingerprinting`[19] sera moins précis.

3.3.5 Limites Icmp

L'Icmp est utile pour les administrateurs réseau et doit donc être la plupart du temps conservé, mais il convient toutefois d'éviter que l'on n'en abuse à des fins de `DoS`[20] ou autres surcharges de bande passante inutile.

Dans les commandes iptables qui vont être citées, les types d'icmp ont étés convertis en chiffres, pour avoir la "traduction" de ceux ci, consultez la commande `iptables -p icmp -h`.

Voici les règles iptables déduites pour la limitation de l'icmp.

[18] 65812 secondes soit 18H de scan.
[19] Detection de l'Os utilisé par la cible.
[20] Denial of Service, attaque devenue classique sur internet

```
iptables -A INPUT -p icmp -m length ! --length 500:65535 \
                  --icmp-type echo-request \
                  -m limit --limit 1/sec --limit-burst 2 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
iptables -A INPUT -p icmp -j DROP
```

Ces dernières veillent à limiter les dégats qu'un potentiel `flooder`[21] pourrait occasionner ; Plus précisément, nous limiterons les paquets `echo-request` à une taille acceptable, sachant qu'un ping standard oscille souvent aux alentours de 84 bytes ; Nous limiterons également la fréquence de ceux-ci. Dans un second temps, nous veillerons à accepter les paquets de retour de requêtes, pour éviter que les ping initié par notre machine ne reviennent jamais.

[21] Personne envoyant un nombre énorme de données en un minimum de temps pour faire tomber la machine de destination.

3.4 Autres chaines

D'autres chaines[22] méritent également quelques sécurisations, mais cela deviens beaucoup plus dépendant de l'implémentation de votre serveur. En effet, ces règles dépendront de votre politique de sécurité ainsi que de l'utilisation de la machine.

3.4.1 OUTPUT

L'output est la chaine souvent oubliée, en effet, beaucoup la laissent vide. Nous ne donnerons pas d'exemple de remplissage de cette chaine, notez toutefois les choses suivantes lors de l'élaboration des règles de celle-ci.

- Vous pouvez filtrer les paquets en fonction de l'utilisateur qui les génères (`-m owner --uid-owner 0`)
- Il serais judicieux d'empecher les connexions des utilisateurs comme nobody, ou des utilisateurs faisant tourner les processus comme Apache.

Exemple : Certains services comportent une partie administrative qui se met en écoute sur un port différent, et souvent sur la boucle locale[23], il conviens qu'uniquement les utilisateurs ayant les droits administratifs puissent se connecter sur ce port local. Pour mettre ce cas en pratique, nous utiliserons `iptables` :

```
iptables -A OUTPUT -p tcp --dport 2260 -o lo -m owner ! --uid-owner 0 -j DROP
```

Ici, on refuse tout les paquets à destination du port 2260, sortant par l'interface locale, et dont l'appartenance n'est pas liée à l'id 0, qui correspond au root[24].

Dans un second point, il serais utile de ne pas autoriser apache[25] à se connecter à l'extérieur, pourquoi ? Imaginons la situation suivante :

- Un pirate s'introduis sur un de vos sites web, et arrive à y éxécuter une commande, cette commande sera éxécutée sur la machine serveur avec les droits d'apache.
- Le pirate voudra certainement télécharger un fichier qu'il pourra par la suite éxécuter, encore une fois avec les droits de l'utilisateur apache.

Il deviens évident alors que si cet utilisateur est restreint quant à ses connections de sortie, les actions du pirate éventuel seront d'autant plus limitées.

Ici, voyons la règles pour empecher apache, ici l'id 213, de se connecter sur un serveur web extérieur.

```
iptables -A OUTPUT -p tcp --dport 80 -o eth0
                   -m owner --uid-owner 213 -j DROP
```

Remarque : Veillez toutefois à faire attention, si des scripts s'éxécutant dans vos pages web requierent des connections à certaines bases de données ou autres ressources réseau externes, il conviendra de les autoriser.

3.4.2 FORWARD

La sécurisation de cette chaine dépend si vous mettez en place une passerelle ou un simple serveur, dans le premier cas, une politique bien définie et réfléchie est à

[22] Une chaine est un ensemble de règles, ex : `INPUT`,`OUTPUT`,`FORWARD`
[23] localhost/127.0.0.1
[24] Administrateur
[25] Serveur web

envisager ; Dans le second, il est courant de définir une politique `DROP`[26], mais de tout de meme veiller à accepter ce qui concerne l'interface locale.

```
iptables -P FORWARD DROP
iptables -A FORWARD -i lo -j ACCEPT
iptables -A FORWARD -o lo -j ACCEPT
```

[26] = Tout sera perdu.

3.5 Conclusion.

Dans ces quelques pages, nous avons appris à ralentir considérablement les `scanport`, ainsi que les attaques de type `SYN-Flood`. Lors de l'élaboration de vos règles de firewall, veillez bien à définir à l'avance quels services devront être accessibles par qui. Les quelques "règles" suivantes sont souvent de bon conseils :

- Veillez à garder vos chaines de règles raisonablement courtes, plus celles-ci seront grandes et plus l'OS mettra du temps à les parcourir lors de l'arrivée d'un paquet...
- N'acceptez pas un service pour tous[27] si seul les personnes de votre réseau local sont susceptibles d'y acceder.
- Ne pas oublier d'accepter le traffic généré par l'interface locale[28], certains programmes utilisent cette interface pour communiquer.

[27] 0.0.0.0
[28] lo

Chapitre 4

X-Window Distant

Preface

Certaines applications composant le projet SIGMA ont besoin d'être installées à distance. En effet, le serveur SIGMA n'est pas forcément situé géographiquement auprès de son administrateur ; Il nous fallais donc prévoir l'éventualité de pouvoir lancer des applications graphiques à distance, et c'est le rôle du tutorial qui va suivre...

Introduction

INTRODUCTION

Ce tutoriel est écrit dans le cadre de mon stage de fin d'étude au sein de l'Université de Liège, département A.N.A.S.T.. Ce document constitue un guide pour l'éxécution d'applications graphique à distance via un serveur `X-Window`.

Sommaire

4.1 Dépendances

Pour éxécuter un programme `X-Window` à distance, il n'est pas nécessaire d'avoir un serveur X hébergé sur la machine ou l'application tourne effectivement. Toutefois, il faudra quand même y installer les librairies et fonts graphiques, pour que les applications puissent s'en servir pour s'éxécuter. Vous pouvez toutefois vous limiter aux librairies X ainsi que les fonts graphiques.

Plus clairement, `emergez`[1] le package gérant le serveur X : `xorg-x11`.

Lors de l'utilisation d'un framework d'affichage comme par exemple `GTK` ou `wxWindows`, il conviendra d'installer les librairies/fonts nécessaires à leur éxécution.

[1] En rapport à la commande, pas au verbe.

4.2 Configuration

Il y a deux cas possible lors de l'utilisation de `X-Window` à distance, tout d'abord, vous pouvez, pour plus de sécurité, encapsuler la connection au serveur X dans une connection `SSH`[2], qui sera donc cryptée et votre authentification gérée par ce dernier.

D'un autre coté, vous pouvez également utiliser directement le protocole de connection X-Window, la configuration de celui-ci sera dans ce cas abordée.

4.2.1 SSHd

Pour l'utilisation de `X-Window` avec `SSH`, il est nécessaire de modifier quelque peu la configuration du service.

Pour ce faire, rendez-vous dans son fichier de configuration, habituellement `/etc/ssh/sshd_config`, et ajoutez/modifiez les variables suivantes :

- X11Forwarding yes
- X11DisplayOffset 10
- X11UseLocalhost yes

Ensuite, pour appliquer ces changements, redémarez le service `SSH` via la commande : `/etc/init.d/sshd restart`.

La configuration coté serveur d'applications est désormais effectuée pour la méthode via `SSH`

4.2.2 Xorg

Dans la méthode utilisant directement le protocole `X-Window`, nous devons veiller à ce que l'instance de X écoute bel et bien sur l'interface communiquant avec le serveur d'applications.

Cette fois, c'est un script système qu'il nous faudra modifier, éditez le fichier `/usr/bin/startx`, et modifiez la ligne suivante :

- defaultserverargs="-nolisten tcp -br"

Par celle-ci :

- defaultserverargs="-br"

Ce qui aura pour effet de faire écouter la prochaine instance du serveur X, sur les interfaces publiques de notre serveur, veillez bien à en restreindre l'accès au moyen par exemple d'un firewall.

La deuxième chose à effectuer lors de l'utilisation de ce protocole, est d'autoriser les ip clientes[3] à se connecter, on effectue ceci grâce à la commande `xhost`.

Exemple :

```
xhost +10.254.254.14
```

Nous autorisons désormais les clients de l'ip 10.254.254.14 à se connecter sur notre serveur `X-Window`.

[2] SecureSHell

[3] clients du serveur X-window

4.3 Execution distante

4.3.1 via SSH

Exemple d'une session complète, le but étant d'executer xclock à distance :

```
wildcat@wildcat ~ $ ssh -XY 10.254.254.42
Password:
Warning: No xauth data; using fake authentication data for X11 forwarding.
Last login: Mon Mar 13 11:02:07 2006 from 10.254.254.3
wildcat@sigmund ~ $ echo $DISPLAY
localhost:11.0
wildcat@sigmund ~ $ xclock
^C
wildcat@sigmund ~ $
```

L'affichage de Xclock s'effectue correctement à travers la connection ssh.

4.3.2 via le protocole X-Window

Exemple d'une session complète, le but étant d'executer xclock à distance :

```
wildcat@wildcat ~ $ ssh 10.254.254.42
Password:
wildcat@sigmund ~ $ export DISPLAY=10.254.254.3:1.0
wildcat@sigmund ~ $ xclock
wildcat@sigmund ~ $
```

Nous constatons l'affichage de xclock sur le serveur X distant.

La connection cette fois n'est pas sécurisée via SSH, elle passe en clair sur le réseau !

Veillez bien à effectuer la partie de configuration vue précédement, et plus particulièrement la partie concernant `xhost`.

4.4 Librairies dépendantes

Il sera certainement nécessaires pour certaines applications plus conséquentes, et si vous ne désirez pas compiler toutes les librairies sur la machine ou devra tourner l'application, de copier tout simplement celles-ci. Bien sûr, celà implique notament que la version du compilateur et des librairies systemes de la machine ou les librairies seront copiées soient les mêmes que celle ou elles auront été extraites.

Pour lister les librairies dont dépendent un programme, une commande système éxiste, il s'agit de `ldd`, bien sur, sa sortie écran étant un peu rébarbative, et la copie des librairies l'étant tout autant, nous assisterons celle-ci au moyen d'un script bash.

```
#!/bin/bash
if [ ``$@'' != ``1'' ]; then
 echo ``$0 <exec name>''
 exit;
fi

mkdir -p -- ``lib-$1''
cd -- lib-$1
PATHVIM=`which $1`

for lib in ` ldd $PATHVIM|cut -f 1 -d' ' ``

 echo -n ``finding $lib. . .''
 PATHLIB=`whereis $lib|cut -f 2 -d':'|cut -f 2 -d' ' ``
 if [ ``$PATHLIB'' ==  ``'' ]; then
  continue;
 fi
 echo ``found at $PATHLIB.''
 echo -n ``copying. . .''
 cp -fLp -- $PATHLIB ./$lib 2>&1
 cp -fLp -- $PATHLIB ./
 echo ``done''
done

cd ..
echo ``end copying all libs, packing. ..''
tar czvf $1-libs.tgz lib-$1
rm -rf lib-$1
```

Ce script copiera dans un répertoire toutes les librairies dépendantes du programme passé en argument, il effectuera ensuite une archive `tgz`[4] de tout son contenu.

L'utilisation de ce tgz est variable, lors du transfert de librairie vers un systeme identique, c'est-à-dire la même version des librairies systèmes, du compilateur ; Aucun problème ne devrais se noter, toutefois, dans le cas contraire, les librairies principales ne devront pas etre utilisées, sous peine de paralyser la console en cours d'utilisation.

Voici la marche à suivre :

[4] Décompressable avec la commande `tar xzvf <fichier>`

- `ssh -XY <host>` // se log en ssh
- `tar xzvf libxxx.tgz` // décompression des librairies
- `export LD_LIBRARY_PATH=‘‘/usr/lib:/lib:./libxxx/’’` // (ajouter a cette variable les path des librairies système
- `./xxxx` // lancez le programme

Chapitre 5

OpenVPN, Intégration

Preface

Les différent “pôles” composant le projet sigma, comme par exemple le traffic et la pollution, peuvent être représentés comme des services à part entière, délocalisé géographiquement, et disposant comme moyen de communication d'un simple accès à Internet. Ces pôles devront être interconnectés pour pouvoir dialoguer avec le serveur SIGMA, et on imagine très mal ces communications transiter dans le vaste et “dangereux” Internet. Pour remédier à celà, nous intégrerons au projet SIGMA une solution complete de VPN, évolutive à souhait, permettant l'interconnection globale de tout les pôles composant le projet, ainsi que pourquoi pas, les sous réseaux qui les composent. Nous verrons au travers de ce tutoriel d'installation, comment effectuer un routage pratique et facile à administrer...

Introduction

Ce tutoriel est écrit dans le cadre de mon stage de fin d'étude au sein de l'Université de Liège, département A.N.A.S.T. Il guide le lecteur pas à pas dans l'intégration du logiciel `OpenVPN` en tant que serveur et client, il abordera également quelques notions de routage basiques ainsi que diverses solutions supplémentaires.

Sommaire

5.1 Le Serveur

5.1.1 Installation

Il va, tout d'abord, être question d'installer les packages nécessaires à la bonne configuration de notre VPN sous Gentoo GNU/Linux. Avant de démarrer la compilation d'openvpn, il va falloir s'assurer qu iproute2 se trouve bien dans la variable USE du make.conf.

```
emerge openvpn
```

Le package qui suit va permettre d'obtenir la commande tunectl, nécessaire la gestion automatique des interface VPN par les scripts systèmes.

```
emerge usermode-utilities
```

5.1.2 Configuration : cléfs

Nous allons créer le répertoire de scripts qui gère les clefs, ainsi quun fichier vars qui contiendra les variables dont les scripts se serviront pour la production des clefs.

```
cd /etc/openvpn
cp R /usr/share/openvn/easy-rsa ersa
cd ersa
vi vars
```

/etc/openvpn/ersa/vars

```
export EASY_RSA=‘‘‘pwd‘’’
export KEY_CONFIG=‘‘$EASY_RSA/openssl.cnf"
export KEY_DIR=’’/etc/openvpn/sigmund_keys‘‘
echo NOTE: If you run ./clean-all, I will be doing a rm -rf on $KEY_DIR
export KEY_SIZE=1024
export CA_EXPIRE=3650
export KEY_EXPIRE=3650
export KEY_COUNTRY=’’BE‘‘
export KEY_PROVINCE=’’LIE‘‘
export KEY_CITY=’’Liege‘‘
export KEY_ORG=’’ANAST‘‘
export MAIL=admin@sigmund.be
```

A présent, nous allons créer un utilisateur vpn et l'ajouter au groupe vpn en ne lui donnant pas d'accès au shell, et dont sa home sera /etc/openvpn.

```
groupadd vpn
useradd -g vpn -s /bin/false -d /etc/openvpn vpn
```

Maintenant, nous allons charger les variables d'environnements gâce à notre fi-

chier vars de tout à l'heure, supprimer toutes les clefs et créer un index grâce à la commande `clean-all`, et générer le `certificat autority` qui va servir à générer toutes les autres clefs. Lors de la génération, les variables contenues dans le fichier vars sont prises en considération dans le shell courant.

```
.   ./vars ( équivalent à source ./var)
./clean-all
./build-ca
```

Une série de question va à présent nous être posée, dont les variables définies précédemment répondent à merveille. Nous nous contenterons donc de taper `ENTER` pour chaque question afin de valider les propositions par défaut.

Ca y est, nous possédons la clef et le certificat pour notre serveur : `ca.crt & ca.key`.

`ca.crt` est la cléfs publique du serveur, elle sera possédée par tout les clients, et `ca.key` est la cléfs privée devant etre possédée uniquement par la machine signant les cléfs des clients.

Nous réalisons à présent la clef publique et la clef privée pour notre serveur.

```
./build-key-serveur sigmund
```

Nous possédons, à présent, le certificat ainsi que les clefs pour notre serveur.

Ces trois fichiers sont propres au serveur et ne seront partagée qu'avec ce dernier.

Générons maintenant les paramètres "Diffie Helman" qui servira lors des échanges de cléfs entre client et serveur.

```
./build-dh
```

Après quelques temps nous récupérons le fichier `dh1048.pem` qui ne sera utile qu'au serveur.

Générons maintenant la cléfs privée propre au serveur, qui sera partagée avec tout les clients, et servant pour crypter l'échange de base avec le serveur lors de l'échange des cléfs.

```
openvpn --secret ta.key --genkey
```

5.1.3 Configuration : openvpn

Réalisons maintenant la configuration propre au vpn, après avoir rassemblé les cléfs précédement générées dans `/etc/openvpn/sigmund_keys`.

/etc/openvpn/sigmund.conf

```
proto udp
port 32458
dev tap0

# Clefs:
ca sigmund_keys/ca.crt
cert sigmund_keys/sigmund.crt
key sigmund_keys/sigmund.key
dh sigmund_keys/dh1024.pem

server 10.254.253.0 255.255.255.0

ifconfig-pool-persist ipp.txt
float

client-config-dir ccd

# permet le traffic entre les clients du vpn
client-to-client

keepalive 10 120
tls-auth sigmund_keys/ta.key 0 # This file is secret
cipher AES-256-CBC

max-clients 20

user vpn
group vpn

persist-key
persist-tun

status openvpn-status.log
status-version 2

log-append openvpn.log
verb 4
mute 20

plugin /usr/lib/openvpn/openvpn-down-root.so ''/etc/openvpn/down.sh''
```

Plusieurs choix s'offrent à vous dans cette configuration, tout d'abord celui d'un `VPN TCP` ou alors `UDP`; Ce choix s'effectue tout simplement par rapport au type de votre connection, l'udp est plus rapide et conviens mieux aux lignes de type `ADSL`, qui ont des déconnections régulières. Le tcp quant à lui conviens aux lignes à ip fixes sans déconnections, comme les lignes dédiées.

Afin de convenir aux deux utilisations, nous choisirons un vpn de type UDP.

- `Float` : Permet de se reconnecter très vite quand l'ip du client change à la volée.
- `Client config-dir ccd` : Dans le répertoire ccd on ira mettre les infos necessaires à chaque client.
- `Client-to-client` : Permet d'autoriser le trafic entre les clients du vpn.
- `Duplicate-cn` : Ne pas mettre ! Signifie que deux clients peuvent se connecter avec le meme certificat !
- `Persist_key` : Evite les problèmes de redémarrage du `vpn`.
- `Persist_tune` : Voir option précédente.
- `Status <fichier>` : Fichier contenant le status des clients et du `VPN`.
- `Dev tap0` : Interface du vpn.
- `Verb 4` : Verbosité des logs.
- `Mute 20` : Ne pas répéter 20 fois le même message...

Le fichier `ipp.txt`[1] témoigne des associations nom/adresse `IP` ayant été attribuées dans le `VPN`, pour pouvoir rendre cette même ip lors de la déconnection temporaire d'un client par exemple.

5.1.4 Configuration : `Gentoo`

Définition du tap0 dans le fichier de définition des interfaces ; Afin que celle de notre `VPN` soit reconnue au démarrage.

/etc/conf.d/net

```
config_tap0=( ‘‘null’’ )
iface_eth0=‘‘1.2.3.4 broadcast 1.2.3.255 netmask 255.255.255.0’’
gateway=‘‘eth0/1.2.3.1’’
```

Ensuite nous créons l'interface de notre `VPN` : `tap0` sur base de `eth0`.

```
ln -s net.eth0 net.tap0
```

Dans le kernel, nous ajoutons le module TUN/TAP device driver, compilons, et chargeons le device.

`Menu kernel: Device Driver / Network device / universal TUN/TAP device driver support`
`modprobe tun`
`ls -al /dev/net/tun` (Afin de vérifier quil existe bel et bien)

Nous définissons maintenant les permissions pour l'utilisateur vpn :vpn sur tout le contenu du répertoire /etc/openvpn, il conviendra lors de l'ajout de nouveau fichiers dans le répertoire de configuration `ccd` d'effectuer ces mêmes changements de permissions.

```
cd /etc/openvpn
mkdir ccd
chown vpn:vpn .  -R
```

[1] IP Pool Persist

```
chmod 700 .  -R
```

Dans la configuration du service openvpn, vous aurez sans doute aperçu un script s'apparentant à un script devant se lancer lorsque le `VPN` s'éteint[2], en effet, il nous faut libérer l'ip de l'interface de celui-ci pour que lors de son redémarage, il puisse lui en rendre une.

Nous effectuons ceci grâce à la commande :

```
ifconfig tap0 0.0.0.0
```

Voici le script complet, guère plus complexe :

/etc/openvpn/down.sh

```
config_tap0=( ‘‘null’’ )
#!/bin/bash
/sbin/ifconfig tap0 0.0.0.0
```

Nous pouvons, à présenter, démarrer notre nouvelle interface `tap0` ainsi que le `VPN`. Il sera utile de créer un `openvpn.sigmund` afin de désigner le `VPN` que nous venons de créer. Il est tout à fait possible d'en créer d'avantage, avec d'autres configurations.

```
/etc/init.d.net.tap0 start
```

```
cd /etc/init.d
ln -s openvpn openvpn.sigmund
/etc/init.d/openvpn.sigmund start
```

[2] `plugin /usr/lib/openvpn/openvpn-down-root.so “/etc/openvpn/down.sh“`

5.2 Le client

Avant d'installer openvpn sur les nouveaux postes, il est obligatoire de générer des cléfs sur le serveur et d'attribuer, au sein du VPN, une IP à ce nouveau poste. C'est par celà que nous allons commencer.

5.2.1 Configuration du client sur le serveur

Sur le serveur, rendez vous dans le répertoire /etc/openvpn et générez les clefs publiques / privées, destinées au nouveau poste, de la manière suivante :

. ./vars (Identique à : source /etc/openvpn/ersa/vars)
./build-key <nomduclient> (Génération des clefs)

Il va à présent falloir réaliser une archive contenant les fichiers qui vont intéresser le nouveau poste ; A savoir, les cléfs et le certificat.

```
cd /etc/openvpn/sigmund-keys
tar czvf nomduclient.tgz nomduclient.* ta.key ca.crt
```

Ensuite, nous allons créer un fichier dans le répertoire ccd, portant le même nom que les cléfs précédement générées. Celui-ci contiendra les paramètres à passer au client, principalement son adresse IP et son masque de réseau.

/etc/openvpn/ccd/nomduclient
ifconfig-push "10.254.253.x 255.255.255.0".x 255.255.255.0"

Il suffit, à présent, d'envoyer l'archive, de manière sécurisée, sur le poste concerné.

5.2.2 Configuration Gentoo GNU/Linux

Installation

Il va, tout d'abord, être question d'installer les packages nécessaires à la bonne configuration de notre VPN sous Gentoo GNU/Linux. Avant de démarrer la compilation d'openvpn, il va falloir s'assurer qu iproute2 se trouve bien dans la variable USE du make.conf.

```
emerge openvpn
```

Le package qui suit va permettre d'obtenir la commande tunectl, automatisant la gestion automatique par les scripts système des interface VPN.

```
emerge usermode-utilities
```

Les cléfs

Installons les clefs générées par le serveur et fournies de manière sécurisée dans le répertoire keys d'OpenVPN.

```
cd /etc/openvpn
mkdir keys
tar xzvf client.tgz
```

Configuration

A présent, nous allons créer la version client du fichier sigmund.conf.

/etc/openvpn/sigmund.conf

```
client
dev tap0
proto udp

remote 1.2.3.4 32458
resolv-retry infinite
nobind

user vpn
group vpn

persist-key
persist-tun

ca keys/ca.crt
cert keys/angel.crt
key keys/angel.key

ns-cert-type server
tls-auth keys/ta.key 1
cipher AES-256-CBC

verb 3
;mute 20

plugin /usr/lib/openvpn/openvpn-down-root.so "/etc/openvpn/down.sh"
```

Définition de l'interface `tap0` dans le fichier de définition des interfaces, afin que celle-ci soit reconnue au démarrage.

/etc/conf.d/net

```
config_tap0=( "null" )
iface_eth0="1.2.4.1 broadcast 1.2.4.255 netmask 255.255.255.0"
gateway="eth0/1.2.4.254"
```

Ensuite nous créons l'interface de notre `VPN` : `tap0` sur base de `eth0`.

```
cd /etc/init.d/
ln -s net.eth0 net.tap0
```

Dans le kernel, ajoutez le module TUN/TAP device driver, compilez, et chargez le device.

```
cd /usr/src/linux
make menuconfig
* Device Driver -> Network device -> universal TUN/TAP device driver
support
make modules_install
modprobe tun
ls -al /dev/net/tun (Afin de vérifier qu'il existe bel et bien)
```

A présent, nous allons créer un utilisateur et un groupe de travail dédié pour le VPN.

```
groupadd vpn
useradd -g vpn -s /bin/false -d /etc/openvpn vpn
```

Nous Créons ensuite le fichier down.sh dans openvpn. Son objectif est le même que pour le serveur.

```
chmod +x down.sh
```

Démarrez l'interface VPN.

```
/etc/init.d/net.tap0 start
```

Le propriétaire du répertoire /etc/openvpn sera l'utilisateur et le groupe précédement attribué au VPN.

```
chown vpn:vpn -R /etc/openvpn
```

Arrangeons nous de la configuration des scripts de démarrage du VPN, et ajoutons le service au démarrage.

```
cd /etc/init.d
ln -s openvpn openvpn.sigmund
/etc/init.d/openvpn.sigmund start
rc-update add openvpn.sigmund default
rc-update add net.tap0 default
```

Consultons maintenant la table de routage, ainsi qu'un ping pour controller le bon fonctionnement

```
route -n
ping 10.254.253.1
```

5.2.3 Configuration Windows

Téléchargons `OpenVPN` sur le site de celui-ci : `http://openvpn.net`.

L'installation de ce dernier ne devrait pas poser de problèmes particulier, elle se trouves être automatisée.

Décompressons le tgz, récupérons-le sur le serveur de manière sécurisée, rendons nous ensuite dans `C:\Program-file\openvpn\config` et créons un fichier sigmund.ovpn dans ce même répertoire.

`C:\Program Files\openvpn\config\sigmund.ovpn`

```
client
dev tap
proto udp
remote 1.2.3.4 32458
resolv-retry infinite
nobind

persist-key
persist-tun
ca ca.crt
cert nomduclient.crt
key nomduclient.key
ns-cert-type server
tls-auth ta.key 1
cipher AES-256-CBC
verb 3
```

Veillons ensuite à configurer notre firewall pour qu'il autorise le traffic du `VPN`, et démarrons le `VPN` en faisant un clic droit sur le fichier ovpn puis "Start openvpn on this config file".

Et hop, nous voici dans le VPN.

5.3 Sortie internet

Lors de la mise en place d'un VPN, il est souvent utile de prévoir une sortie internet sécurisée pour les utilisateurs nomades, en effet, lors du déplacement de ceux-ci en zone non sécurisée, il convient de faire attention aux informations propagées sur un réseau inconnu. Pour palier à ce problème, nous utiliserons le tunnel VPN comme sortie internet. C'est donc une autre machine présente dans le réseau VPN qui effectuera réellement les requetes à destination d'internet, sans aucune connaissance de celles-ci pour le réseau dans lequel le client nomade se trouves.

5.3.1 Configuration iptables

Finalement, la configuration d'iptables pour effectuer cette sortie internet est grandement semblable, voire identique, aux scripts réalisés lors de la mise en place d'une Gateway[3].

Exemple :

- `iptables -t nat -A POSTROUTING -o eth0 \`
  `-s 10.254.254.2 -j MASQUERADE`
- `iptables -A FORWARD -i tap0 -s 10.254.254.2 \`
  `-o eth0 -j ACCEPT`
- `iptables -A FORWARD -i eth0 -o tap0 -d 10.254.254.2 \`
  `-m state ----state ESTABLISHED,RELATED -j ACCEPT`

Ces 3 règles accepent le traffic provenant de l'interface du VPN[4], à destination de l'interface internet[5], et en tenant compte de la source[6]. Egalement, du NAT[7] est mis en oeuvre pour translater les addresses privées provenant du VPN, en adresse publiques, ici, on changera donc l'adresse source de la requete en effectuant du MASQUERADING.

Egalement, si votre machine n'agit pas encore comme passerelle, il conviens d'activer le forwarding, ceci se fait via l'interface /proc/sys/net/ipv4/ip_forward, qui signale au kernel d'effectuer celà.

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Cette valeur peut être ajoutée dans le fichier /etc/sysctl.conf pour être repris à chaque redémarrage de la machine.

5.3.2 Ajouter les routes aux clients

Pour ajouter les routes au clients, deux solutions s'offrent à vous, soit gérée du coté du serveur, soit du coté du client ; Les deux solutions sont abordées ci-dessous, tenant compte des systèmes GNU/Linux ainsi que des systèmes Windows.

[3] Passerelle
[4] Ici, tap0
[5] Ici eth0
[6] Ici, accepter le traffic provenannt de 10.254.254.2
[7] Translation d'adresses dynamique

Coté serveur

Pour ajouter des routes aux clients se connectant au serveur, nous pouvons tout simplement ajouter une directive à la configuration par client dans le repertoire `ccd`[8] du serveur `VPN`.

Par exemple, pour redistribuer la route par défaut, on ajoutera la directive suivante :

– `push "redirect-gateway"`

Qui aura pour effet d'enlever la route par défaut, d'ajouter une route vers le serveur `VPN` passant par l'ancienne route par défaut, pour finalement ajouter une route par défaut contenant l'ip du serveur `VPN`.

Coté client

GNU/Linux L'ajout manuel de routes sous linux est chose aisée, imaginons la situation suivante :

– La gateway par défaut du réseau ou se trouve le client est 192.168.0.1
– La gateway du serveur VPN est 10.254.254.1
– L'ip du serveur VPN sur internet est 1.2.3.4

Les commandes suivantes seront éxécutées pour, dans un premier temps ajouter une route vers le serveur `VPN`, pour pouvoir toujours joindre celui-ci lors de la suppression de la route par défaut ; Ensuite, la suppression de cette dernière ; Et finalement l'ajout de la route par défaut passant par le serveur `VPN`.

– route add -host 1.2.3.4 gw 192.168.0.1
– route del default
– route add default gw 10.254.254.1

On pourrais imaginer dans ce cas d'utiliser un autre serveur se trouvant dans le même réseau `VPN`.

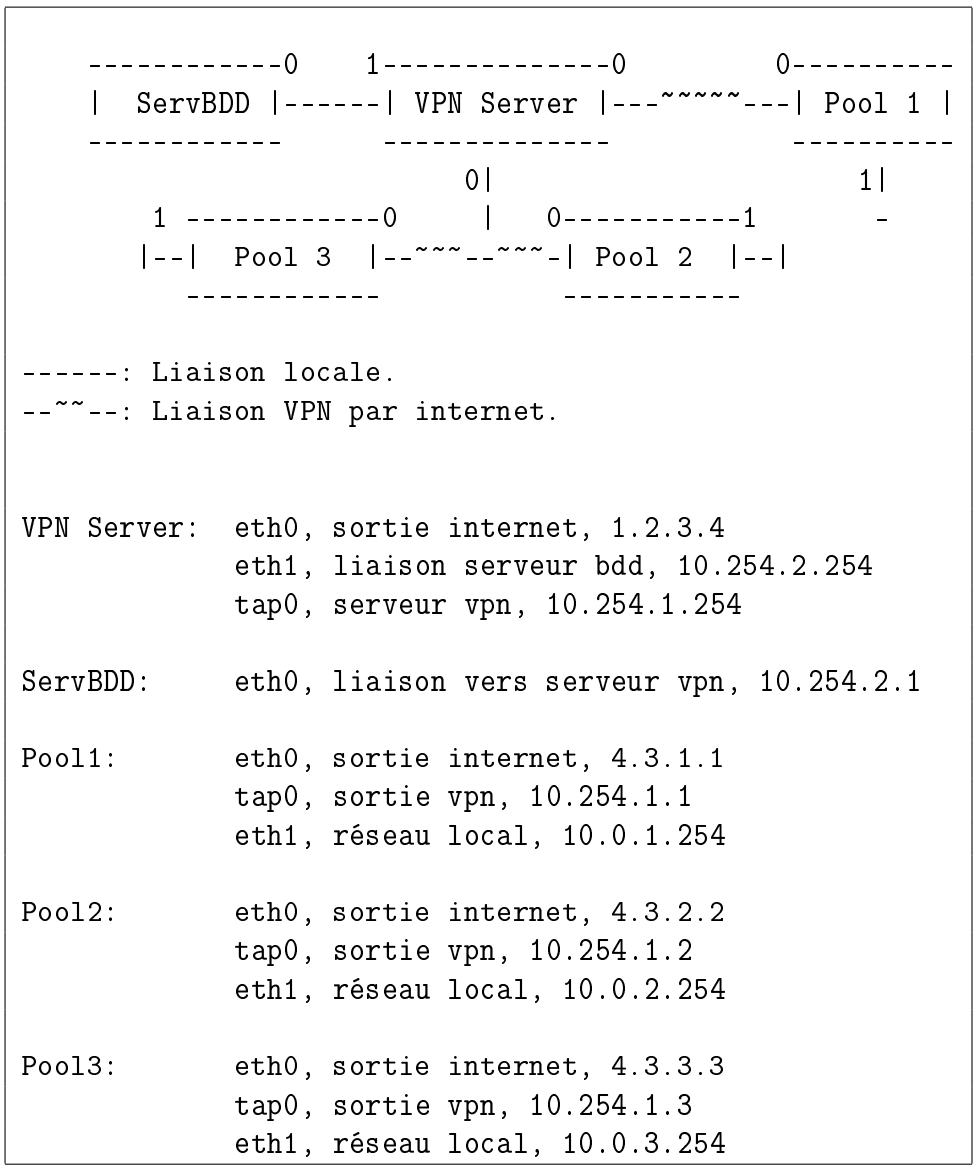
Windows L'équivalence de commandes ci dessus pour les système Windows® se forge comme suit :

– route add 1.2.3.4 MASK 255.255.255.255 192.168.0.1
– route delete 0.0.0.0 MASK 0.0.0.0
– route add 0.0.0.0 MASK 0.0.0.0 10.254.254.1

[8] Client Config Directory

5.4 Routage statique

Nous allons analyser examiner une solution de routage complète pour un cas réel, voici le plan de la topologie réseau utilisée tout au long de ce chapitre, elle met en avant la future solution utilisée par le modèle intégré `SIGMA`, composé d'un serveur central, et de plusieurs pôles "greffés" sur ce même serveur.



On désire mettre en place un routage statique entre les différents pool, ainsi que sur le serveur de base de donnée, nous allons étudier l'exemple de route du serveur `VPN`, puis nous fournirons les commandes pour les serveurs de chaques pool ainsi que du serveur de base de donnée.

Serveur `VPN` :

Ces commandes vont définir le point de passage pour joindre les sous-réseaux de chaque pool, pour ce faire il est évident qu'il faudra utiliser comme point de passage, le sereur de chaque pool ainsi connecté au `VPN`. Ainsi donc, pour joindre le réseau du pool 2, `10.0.2.0/24`, le serveur `VPN` devra utiliser comme point de passage, l'adresse `10.254.1.2`, correspondant au client `VPN` se trouvant sur le serveur du pool 2.

```
route add -net 10.0.1.0/24 gw 10.254.1.1
route add -net 10.0.2.0/24 gw 10.254.1.2
route add -net 10.0.3.0/24 gw 10.254.1.3
```

Serveur Base de donnée :

```
route add -net 10.0.1.0/24 gw 10.254.2.254
route add -net 10.0.2.0/24 gw 10.254.2.254
route add -net 10.0.3.0/24 gw 10.254.2.254
```

Serveur Pool 1 :

```
route add -net 10.0.2.0/24 gw 10.254.1.2
route add -net 10.0.3.0/24 gw 10.254.1.3
route add -net 10.254.2.0/24 gw 10.254.254.254
```

Serveur Pool 2 :

```
route add -net 10.0.1.0/24 gw 10.254.1.1
route add -net 10.0.3.0/24 gw 10.254.1.3
route add -net 10.254.2.0/24 gw 10.254.254.254
```

Serveur Pool 3 :

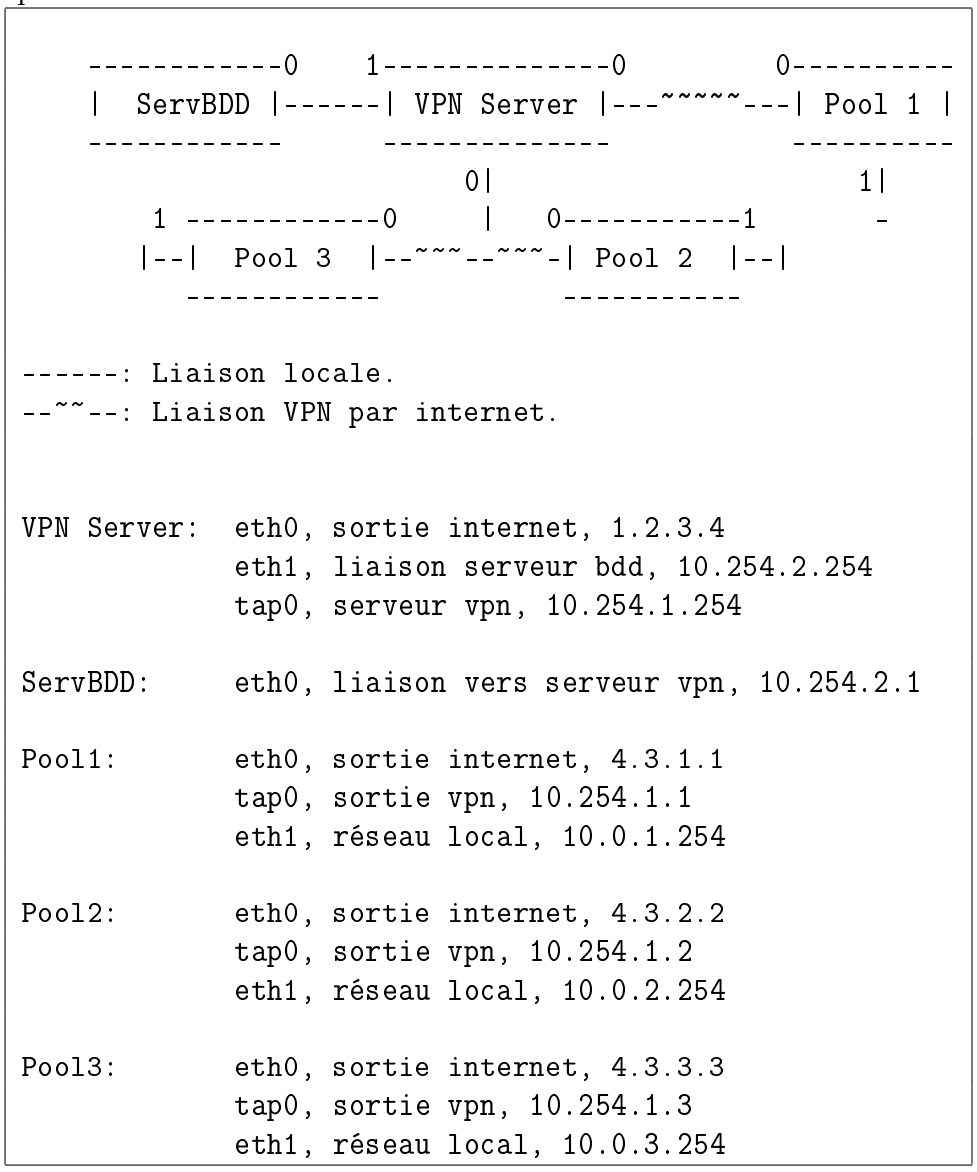
```
route add -net 10.0.1.0/24 gw 10.254.1.1
route add -net 10.0.2.0/24 gw 10.254.1.2
route add -net 10.254.2.0/24 gw 10.254.254.254
```

Une solution tel que celle-ci est évidement fort lourde à maintenir, et l'ajout d'une route sur tout le réseau nécéssite la modification de tout les scripts, sur tout les serveurs.

Toutefois, dans le cas d'un réseau ne nécéssitant jamais de changements topologique, cette méthode sera la plus éfficace...

5.5 Routage dynamique

Le cas étudié dans le cadre du routage dynamique sera le même que précédement pour le routage statique, toutefois, il est plus conséquent à mettre en place mais offre une possibilité d'évolution à moindre frais.



Les protocoles de routage utilisés ici seront `BGP` et `OSPF`, `BGP` est un protocole inter-passerelle qui conviendra très bien entre le serveur `VPN` et ses serveurs "pool". D'autre part, si ces mêmes pool avaient plus d'un réseau derrière eux, nous envisagerions l'utilisation d'`OSPF` derriere ceux-ci, qui supporte très bien l'envoi de ses routes au travers du `BGP`.

OpenVPN ne tournant pas sur le matériel réseau fort bien connu du monte du routage, c'est sous GNU/Linux que nous effectuerons celui-ci, gâce notamment à la suite de routage `quagga`. Cette suite se compose de plusieurs services de routage comme `BGP`[9] ou `OSPF`[10].

[9] Border Gateway Protocol
[10] Open Shortest Path First

5.5.1 Définitions BGP et données de départ

Le serveur principal BGP se trouvera au même endroit que le serveur VPN, celui-ci redistribuera les routes à tous ses peers[11] et acceptera également celles-ci.

Chaque peer BGP se verra attribuer un numéro d'AS[12], qui l'identifiera sur le réseau BGP.

Il n'y a pas de règles particulières pour choisir les numéro autonomes, toutefois, la range de numéro privé se trouve entre 64512 et 65535 et peut être utilisée pour tout réseau qui ne se lie pas vers internet. Attention toutefois, deux numéros ne peuvent se retrouver dans un même réseau BGP

Nom	AS
VPN Server	65139
ServBDD	65140
Pool 1	65141
Pool 2	65142
Pool 3	65143

5.5.2 Installation de Quagga

Cette partie de l'installation sera commune à tous les serveurs du VPN qui sont directement raccordé à celui-ci.

Avant de lancer la compilation de Quagga, qui comprend la suite de services nécessaire à l'utilisation du BGP, nous devons vérifier que les mots cléfs tcp-zebra et fix-connected-rt sont bien dans la variable USE du fichier /etc/make.conf.

```
emerge -av quagga
```

Voyons maintenant du coté des scripts de démarrage Gentoo.

/etc/conf.d/zebra
```
ZEBRA_OPTS="-l 255 -A 127.0.0.1"
```

Nous définissons ici un niveau de log maximal, et désignons également l'adresse locale comme adresse d'écoute pour l'interface d'administration du service zebra.

5.5.3 Configuration de zebra

Maintenant, nous allons effectuer la configuration minimale des services zebra, qui coordonne les communications entre services de routage, et de bgpd qui se chargera du protocole BGP proprement dit.

/etc/quagga/zebra.conf
```
hostname ServeurVPN(zebra)
password monpass

line vty
```

[11] "Clients"
[12] Autonomous System, Système autonomme

/etc/quagga/bgpd.conf

```
hostname ServeurVPN(bgpd)
password monpass

line vty
```

Ici nous n'avons uniquement que le nom de l'instance du service ainsi que le mot de passe pour y acceder. Profitons-en pour donner les permissions à l'utilisateur `quagga` aux fichiers de configuration.

```
chown quagga:quagga /etc/quagga/*
```

5.5.4 Configuration de bgpd sur le serveur

Passons maintenant à la configuration du serveur `BGP` se trouvant sur le Serveur `VPN`, celle-ci s'effectue à l'aide d'une interface `telnet` locale, et copie la syntaxe des routeurs d'un fabriquant fort connu dans ce marché.

Connectons nous à cette interface...

```
wildcat@sigmund ~ $ telnet 127.0.0.1 2605
Trying 127.0.0.1..
Connected to 127.0.0.1.
Escape character is '^]'.

Hello, this is Quagga (version 0.98.5).
Copyright 1996-2005 Kunihiro Ishiguro, et al.


User Access Verification

Password:
```

Entrons le password et passons en mode administrateur, ainsi qu'en mode configuration :

```
ServeurVPN(bgpd)> ena
ServeurVPN(bgpd)# conf t
ServeurVPN(bgpd)(config)#
```

Maintenant, lancons un processus de routage `BGP`, celui correspondant à notre `AS` précédement choisis.

```
ServeurVPN(bgpd)(config)# router bgp 65139
ServeurVPN(bgpd)(config-router)#
```

Maintenant, nous devons définir les réseau qui devront etre propagés par notre serveur, en l'occurence, il s'agira de `10.254.2.0/24` et `10.254.1.0/24`, étant respectivement le réseau partagé par le serveur de base de donnée, et le réseau `VPN` lui même.

```
ServeurVPN(bgpd)(config-router)# network 10.254.2.0/24
ServeurVPN(bgpd)(config-router)# network 10.254.1.0/24
```

L'essentiel étant fait, nous pouvons à présent prévoir une configuration pour chacun des pool devant se connecter à notre serveur BGP, les informations à fournir sont à chaque fois les mêmes ; Le numéro d'AS défini précédment ainsi que l'adresse IP du client BGP.

```
ServeurVPN(bgpd)(config-router)# neighbor 10.254.2.1 remote-as 65140
ServeurVPN(bgpd)(config-router)# neighbor 10.254.1.1 remote-as 65141
ServeurVPN(bgpd)(config-router)# neighbor 10.254.1.2 remote-as 65142
ServeurVPN(bgpd)(config-router)# neighbor 10.254.1.3 remote-as 65143
ServeurVPN(bgpd)(config-router)#
```

Nous finirons par sauvegarder la configuration, et sortirons proprement du mode de configuration.

```
ServeurVPN(bgpd)(config-router)# write
ServeurVPN(bgpd)(config-router)# exit
ServeurVPN(bgpd)(config)# exit
ServeurVPN(bgpd)#
```

Voilà, nous pouvons maintenant passer à la configuration des autres clients BGP, notez toutefois qu'il faudra bien évidement veiller à ouvrir le port 179, correspondant à BGP pour que celui-ci puisse joindre ses clients, et recevoir ceux-ci comme il se doit.

5.5.5 Configuration de bgpd sur le serveur de base de données

Pour les configurations suivante, nous nous contenterons d'afficher la configuration de bgp, celles-ci étant similiaires à celle du serveur.

```
hostname ServeurBDD(bgp)
password monpass

router bgp 65140
 neighbor 10.254.1.254 remote-as 65139


line vty
```

5.5.6 Configuration de bgpd sur les pools

Pool 1 :

```
hostname Pool1(bgp)
password monpass

router bgp 65141
 neighbor 10.254.1.254 remote-as 65139


line vty
```

Pool 2 :

```
hostname Pool2(bgp)
password monpass

router bgp 65142
 neighbor 10.254.1.254 remote-as 65139


line vty
```

Pool 3 :

```
hostname Pool3(bgp)
password monpass

router bgp 65143
 neighbor 10.254.1.254 remote-as 65139


line vty
```

N'oublions pas de sauver chaque configuration dans son routeur grâce a la commande `write`.

5.5.7 Tests et vérifications

Il est également possible via l'interface d'administration d'effectuer certaines vérifications sur l'état des routes.

`show ip bgp summary` ou `sh ip bgp sum`

Cette commande nous montres l'état des clients `BGP` connectés à la machine, ainsi que le nombre de routes recues de ceux-ci.

Si nous sortons de l'interface d'administration, nous pouvons vérifier l'état des routes grâce a deux commandes différentes :

`ip route show` `route -n`

Cette commande faisant partie de la suite `iproute2`, vous devrez peut-être lancer la compilation de celle-ci avant de pouvoir effectuer cette commande. Elle nous montre parfaitemant les routes disponibles sur le systemes, ainsi que leur type. `zebra` étant l'indicateur d'une route obtenue via un protocole de routage.

Enfin, gâce à ces commandes, nous devrions voir apparaitre la carte complete du réseau une fois que tout les clients `BGP` se seront échangés leurs informations.

5.6 Conclusion

Nous avons maintenant acquis les bases du protocole de routage BGP, nous avons également appris à installer correctement le service OpenVPN pour le système Gentoo.

Nous avons également vu les différentes et les implications des systèmes de routage statique et dynamique, et avons mis en place une solution complete de routage dynamique basée sur le protocole BGP.

Chapitre 6

Serveur DNS et Administration PHP

Preface

SIGMA et l'ANAST disposent de leurs noms de domaines respectifs, il faut à présent les administrer et leur permettre de gérer leurs dns simplement et intuitivement. Nous avons allié une solution robuste de DNS grâce au serveur DNS très connu `BIND 9`, et nous utiliserons également une interface de gestion pour ce dernier, correspondant exactement à nos besoins. Le détail de l'installation de ceux-ci se trouves dans les pages à venir...

Introduction

Ce tutoriel est écrit dans le cadre de mon stage de fin d'étude au sein de l'Université de Liège, département A.N.A.S.T. Il décrit l'installation et la configuration du serveur `DNS BIND 9` ainsi qu'un programme d'administration `PHP` pour ce dernier.

Sommaire

6.1 Données de départ

Notre but dans ce document, sera d'élaborer pas à pas, une solution `DNS` complète, tant pour nos domaines locaux, que pour les `DNS` de notre `VPN`.

Il devra bien sûr comporter les sécurisation nécessaire pour ne pas laisser passer d'informations sur l'architecture de ce dernier dans les réseaux publics tels qu'internet, et se contentera de servir ces informations aux clients du `VPN`.

Egalement, il nous est demandé d'installer également un panel pour la gestion des `DNS` des domaines publics, et d'appliquer ce cas au domaine `anast.be`.

La zone factice pour le `DNS` du réseau interne `VPN` sera `vpn.anast.be`.

6.2 Installer BIND9

6.2.1 Packages

Nous allons dès à présent lancer la compilation de `BIND9`, pour se faire, entrez la commande suivante

```
emerge -av bind9
```

Lorsque cette derniere est achevée, l'installeur vous indique que si vous êtes dans le cas d'une utilisation dite `chrootée`, ce qui signifie que le processus du service `DNS`, sera emprisonné dans un environnement qui lui sera propre. Ceci est recommendé pour plus de sécurité, en effet, une faille dans ce serveur compromettrais les données à l'intérieur de cet environnement et aucunement celles à l'extérieur.

```
emerge -config '=net-dns/bind-9.3.2'
```

Ceci aura pour effet de préparer l'environnement `chroot` dans le répertoire par défaut `/chroot/dns`, il est possible de changer ce répertoire via le fichier de configuration `/etc/conf.d/named`, toutefois, cela ne nous est aucunement nécessaire.

6.2.2 Configuration de base

Nous allons à présent effectuer la configuration de base de `BIND9`, définir par exemple les adresses d'écoute du service `DNS`.

/chroot/dns/etc/bind/named.conf

```
options {

  directory "/var/bind";
  listen-on { 127.0.0.1; 10.254.253.1; 1.2.3.4; };
  pid-file "/var/run/named/named.pid";
};

zone "." IN {
 type hint;
 file "named.ca";
};

zone "localhost" IN {
type master;
 file "pri/localhost.zone";
 allow-update { none; };
 notify no;
};

zone "127.in-addr.arpa" IN {
 type master;
 file "pri/127.zone";
 allow-update { none; };
 notify no;
};
```

6.2.3 Démarrage

Nous pouvons maintenant démmarer `BIND9` et effectuer la manipulation pour le lancer à chaque démarrage du système.

```
/etc/init.d/named start
rc-update add named default
```

6.3 Simple Management for Bind

SMBind est une interface PHP de gestion de DNS prévue pour le serveur bind, son principal avantage est, qu'a l'instar des autres panels de gestions dns, il ne nécéssite aucune tache s'éxécutant à intervalles réguliers[1], en effet, celui-ci intègre directement les changements en écrivant les fichiers de zones, et en rechargant la configuration de BIND9 à la volée.

6.3.1 Dépendances

Avant tout, nous devont installer le serveur web apache ainsi que php, les versions choisies seront respectivement apache 2.x et php 5.x. Ce panel nécéssite également l'installation d'un serveur mysql.

Tout d'abord, ajoutons les flags suivant à notre variable USE se trouvant dans le fichier de configuration /etc/make.conf.

```
sqlite session -recode mpm-worker threads dba iconv memlimit xml xsl
zip pear pcre cli hardenedphp apache2 mysql gd
```

Ceux-ci ne sont pas tous nécessaire, mais il conviendra toutefois de veiller à garder les bonnes association pour les packages apache ainsi que php5.

```
emerge -av mysql
```

Il nous faudra effectuer les opérations post-installation pour le packages serveur mysql, ces opérations se résument finalement en la commande emerge -config ... figurant à la fin de l'emerge de celui-ci.

```
emerge -av apache dev-lang/php
```

La configuration d'apache et la sécurisation de php n'entrant pas dans le cadre de ce tutoriel, nous nous limiterons à leur installations par défaut, convenant parfaitement à notre environnement de tests.

Le panel PHP SMBind utilise également un système de template nommé Smarty, il conviens d'installer celui-ci.

```
emerge smarty
```

6.3.2 Extraction et intallation php

Nous allons tout d'abord récupérer l'archive de SMBind, celle-ci ne se trouvant pas dans le portage...Le site officiel répertoriant les sources du projet est hébergé sur le célèbre site sourceforge, rendons-nous donc à l'addresse suivante : http://sourceforge.net/projects/smbind/ et téléchargons-y la dernière version de SMBind.

Une fois récupérée, décompressons-la et copions le contenu du répertoire php/ dans notre répertoire public apache.

```
tar xjvf smbind-0.4.1.tar.bz2
cp -R smbind-0.4.1/php/* /var/www/localhost/htdocs/bind/
```

[1] aka. crontab

```
cd /var/www/localhost/htdocs/bind/
chown apache config.php template_c
chmod 600 config.php
chmod 755 template_c
```

6.3.3 Bug découvert et Patch

Lors de notre utilisation du panel d'administration, nous avons eu l'occasion de constater un bug dans la modification du propriétaire d'une `zone DNS`, ce bug a été corrigé par nos soins et fais l'occasion d'un mini-patch. La version actuellement patchée est la version `0.4.1`.

Nous allons appliquer le patch directement dans le répertoire public de `smbind`.

```
cd /var/www/localhost/htdocs/bind
wget http://wildcat.espix.org/smbind-0.4.1.diff
patch -p0 < smbind-0.4.1.diff
```

Voilà, le petit bug est dès à présent corrigé.

6.3.4 Installation des tables mysql

L'interface nécessite une base de données mysql, créons celle-ci et ajoutons un utilisateur pourvu des droits sur cette dernière.

```
mysqladmin -u dbuser -p create smbind
mysql -p -u dbuser -D smbind < smbind-mysql.sql
mysql> grant ALL on smbind.* to smbind@localhost;
mysql> set password for smbind@localhost=PASSWORD('monpass');
mysql> flush privileges;
mysql> exit;
```

6.3.5 Configuration de BIND9

Nous devons ajouter à présent dans la configuration du service `DNS`, la ligne d'inclusion de toute la configuration qui sera générée par notre interface. A la fin de notre fichier de configuration (`/chroot/dns/etc/bind/named.conf`), ajoutons la ligne suivante :

```
include "/etc/smbind/smbind.conf";
```

Créons par la même occasion ce répertoire et ce fichier de configuration, et donnons-lui les attributs appropriés. Par la même occasion, linkons la version principale de la clef utilisée pour l'administration du serveur via l'outil rndc, lui-même utilisé par notre panel. Ce dernier sert à relancer la configuration de bind sans pour autant avoir des privilèges élevés sur la machine.

```
mkdir /chroot/dns/etc/smbind
touch /chroot/dns/etc/smbind/smbind.conf
chown named:named /chroot/dns -R
chmod 770 /chroot/dns -R
ln -sf /chroot/dns/etc/bind/rndc.key /etc/bind/rndc.key
```

Pour que l'utilisateur apache puisse mettre à jour les fichiers `DNS`, il nous faut éxécuter la commande suivante, visant à ajouter ce dernier dans le groupe `UNIX` du serveur `DNS`.

```
usermod -G named apache
```

6.3.6 Configuration de l'interface

Le fichier `/var/www/localhost/htdocs/bind/config.php` contiens les variables de configuration basiques de l'interface, comme par exemple les paramètres d'accès à la base de données, ainsi que les chemin vers les différentes dépendances de ce package.

`/var/www/localhost/htdocs/bind/config.php`

```
// Include paths.
$_CONF['smarty_path'] = "/usr/lib/php/Smarty";
$_CONF['peardb_path'] = "/usr/share/php";

// Database DSN.
$_CONF['db_type'] = "mysql";
$_CONF['db_user'] = "smbind";
$_CONF['db_pass'] = "monpass";
$_CONF['db_host'] = "localhost";
$_CONF['db_db'] = "smbind";

// Zone data paths (normal).
#$_CONF['path']  = "/var/named/";
#$_CONF['conf'] = "/etc/smbind/smbind.conf";

// Zone data paths (chroot).
$_CONF['path'] = "/chroot/dns/var/bind/";
$_CONF['conf'] = "/chroot/dns/etc/smbind/smbind.conf";

// BIND utilities.
$_CONF['namedcheckconf'] = "/usr/sbin/named-checkconf";
$_CONF['namedcheckzone'] = "/usr/sbin/named-checkzone";
$_CONF['rndc']      = "/usr/sbin/rndc";
```

Enfin, pour vérifier votre configuration, le script "http ://sigmund.anast.be/bind/configtest.php" vous aidera à débuguer votre configuration si besoin étais.

6.3.7 Présentation de l'interface

L'interface `PHP` de gestions des dns est assez intuitive, elle se compose d'un menu vous permettant de choisir vos actions, une gestion d'utilisateurs simple, pour déléguer la gestion de certains domaines.

Lors de l'ajout ou la modification de zones, il convient toujours d'aller actionner l'option *Commit Changes*.

Les différents types d'enregistrements `DNS` sélectionnables via le menu option, ainsi que les paramètres par défaut.

6.3.8 Définition des paramètres par défaut

Les paramètres par défaut de tous les domaines sont bien évidement modifiables (et doivent l'être !), rendez vous pour ce faire dans le menu *Options* et completez y les informations de la maniere suivante :

`Site Hostmaster Address` : Adresse internet de l'administrateur, attention, l'arobase est remplacé par un simple '.'.
`Default Primary NS` : Votre serveur DNS primaire.
`Default Secondary NS` : Votre serveur DNS secondaire.

Validez vos changements.

Les autres options visent à sélectionner les différents types d'enregistrement accessible dans le menu des zones, par défaut, ces paramètres incluent les pricipaux types utilisés. Toutefois, si vous aviez besoin d'un type supplémentaire listé dans cette zone, cochez simplement celui-ci et validez les changements.

6.3.9 Template SIGMA

Pour démontrer la facilité de modification de l'apparence de l'interface d'administration, un thème `SIGMA` à été réalisé pour remplacer le thème par défaut. Les fichiers `.tpl` sont en fait des fichiers `HTML` avec certaines balises spéciales utilisées par `Smarty` pour y afficher le contenu dynamique. Nous pouvons modifier ces fichiers `.tpl` à notre guise.

6.4 Conclusion

Chapitre 7

Sigma Ebuild

Preface

Le projet SIGMA tournant désormais sous le système GENTOO GNU/Linux, nous avions besoin d'une procédure d'installation simple pour tout nos programmes et dépendances SIGMA. Pour ce faire, nous avons développés un petit nombre d'`ebuild`[1] correspondant aux programmes SIGMA et leurs dépendances. Voici dans les quelques pages suivantes, une description pas à pas de la création de ces ebuilds.

[1] système de package de Gentoo.

Introduction

Ce tutoriel est écrit dans le cadre de mon stage de fin d'étude au sein de l'Université de Liège, département A.N.A.S.T. Il guide le lecteur pas à pas dans la découverte du système de package de la distribution GNU/Linux Gentoo, on y découvre comment créer son propre package ainsi que ses dépendances.

Sommaire

7.1 Système d'ebuild

7.1.1 Présentation

Un ebuild est un script bash spécifiquement développé pour le système PORTAGE de `Gentoo GNU/Linux`, celui-ci prend en charge toute la procédure d'installation d'un programme, ainsi que parfois même sa configuration. Une collection d'ebuild de base est fournie lors de l'installation du système `Gentoo`, mais rien ne vous empêche de développer vos propres ebuilds.

7.1.2 Installation d'un portage tiers

Lors du développement ou de l'utilisation d'ebuilds tiers, il est important de ne pas les mélanger avec le `Portage` de `Gentoo`, pour ce faire, nous allons préparer notre système à acceuillir un nouvel arbre de `Portage`, créé par nos soins, ou tout simplement récupéré sur internet.

Dans cet exemple, nous allons supposer l'installation du `Portage` d'ANAST.

```
sigma$ mkdir -p /usr/local/portage
sigma$ echo ‘‘PORTDIR\_OVERLAY=’/usr/local/portage’’’
sigma$ env-update ; . /etc/profile
sigma$ tar xzvf anast-portage-latest.tgz -C /usr/local/
```

Et voilà ! Maintenant, vous pouvez installer l'application SIGMA d'une simple commande, après avoir spécifié les `USE Flags` nécessaires dans votre `make.conf` :

```
sigma$ emerge -av ana-sigma/sigma
```

7.1.3 Classement des ebuilds

Le portage de `Gentoo` suis une certaine norme pour son arborescence, il se divisent en catégories, elles mêmes contenant les différents programmes, pour finalement arriver aux ebuild proposés pour un programme. Il peut y avoir plusieurs ebuild par programme, on les distingueras par leur numéros de version.

Voici l'arborescence utilisée pour le portage ANAST :

ana-sigma		
	sigma	
		sigma-0.2
dev-php		
	php	
		php-4.3.2
	php-cgi	
		php-cgi-4.3.2

7.1.4 Cahier des charges

Voici ce que notre `ebuild sigma` devra accomplir, mais également les autres tâches que nous aurons à réaliser :

- Prendre en charge toutes les dépendances concernant le gestionnaire de scenarii.
- Installer les versions de php et php-cgi prévues par le développeur de ce dernier.
- Initialiser la base de donnée si celle-ci doit l'être.
- Les ebuild pour mapserver étant inéxistant, il nous faudra en faire un également.
- Les ebuild de php et php-cgi n'étant pas complet, il convien de les modifier.

7.2 Ebuild SIGMA

L'`ebuild` développé pour SIGMA se chargera d'installer, entre autre, Scnearii Maker et tout le nécessaire à son éxécution.

7.2.1 Dépendances

Scenarii maker a besoin de beaucoups de dépendances, voici une liste de celles-ci :

- proj, localisée dans sci-libs
- geos, localisée dans sci-libs
- postgis, localisée dans dev-db
- gd, localisée dans media-libs
- gdal, localisée dans sci-libs
- curl, localisée dans net-misc
- tiff, localisée dans media-libs
- freetype, localisée dans media-libs
- mapserver, localisée dans ana-sigma[2]
- apache, localisée dans net-www
- php, localisée dans dev-php[3]
- php-cgi, localisée dans dev-php[4]

Elles seront installées en tant que telles, mais seront toutefois sélectionnées avec soins gâces aux USE flags défini plus tard...

Il y a deux variables gérant les dépendances des ebuild, dans l'ebuild sigma, nous définirons ces deux variables. La premiere symbolise les dépendances directes, nécessaires pour la compilation de l'ebuild sigma, quant à la deuxième variable, elle symbolise les dépendances nécessaire pour l'éxécution du package sigma.

```
   DEPEND="proj?  ( sci-libs/proj ) geos?  ( sci-libs/geos ) dev-db/postgis
mapserver?  ( media-libs/gd sci-libs/gdal net-misc/curl media-libs/tiff
media-libs/freetype )"
   RDEPEND="mapserver?  ( net-www/apache dev-php/php dev-php/php-cgi ana-sigma/mapserver
)"
```

Il est possible, comme vous pouvez le constater, de lier certaines dépendances en fonction d'un certain USE flag, par exemple, si le USE flag `proj` est appliqué, la dépendance envers la librairie `sci-libs/proj` sera nécessaire.

7.2.2 USE flags

Les USE-flags sont définis dans notre ebuild sigma ainsi que dans chaque ebuild dépendant de celui-ci, c'est la variable `IUSE` qui contient ceux-ci.

Voici les USE-flags pour l'ebuild sigma :

```
   IUSE="mapserver proj geos shapedata webdata"
```

[2] Se situe dans notre portage ANAST
[3] Version spécifique ANAST
[4] Version spécifique ANAST

7.2.3 Code source

Comme tout ebuild, SIGMA a besoin d'une source pour installer ses programmes, ici, scenarii-maker. Nous spécifierons cette URL via la variable SRC_URI.

```
SRC_URI="shapedata?  ( http://wildcat.espix.org/pub/shape_files.tgz
) webdata?  ( http://wildcat.espix.org/pub/sigma-0.2.tar.gz )"
```

Comme pour les dépendance, le système de USE flag est toujours valide, nous ne téléchargerons les `shapes files` uniquement si nous en avons besoin, ainsi que le site web de l'application scenarii maker, uniquement si celui-ci nous est demandé.

7.2.4 Configuration de Scenarii Maker

7.3 Conclusion