Virtual Services Howto

Brian Ackerman, brian@nycrc.net
Adaptation française Julien Garnault judge@club-internet.fr
v2.1, 15 Aout 1998

Ce document a été écrit pour répondre au nombre grandissant de questions sur la manière de rendre un service virtuel.

Contents

1 Introduction

1.1 Pré-requis

Créer une machine pour des services virtuels n'est pas du tout difficile. Cependant, des connaissances basiques ne sont pas suffisantes. De plus, ce document n'est pas destiné à vous expliquer comment configurer une machine sous Linux.

Afin que vous puissiez comprendre ce HOWTO, nous supposerons que les documents suivants vous sont tout à fait familiers :

- Compiler un noyau Linux et ajouter le support de l'IP aliasing *IP alias mini-HOWTO*
- Installer et configurer des périphériques réseau *NET-3 HOWTO*
- Installer inetd *NET-3 HOWTO*
- Compiler et installer divers paquetages de logiciels en réseau comme *Le site de Sendmail*
  Le site d'Apache
  La FAQ de Wu-Ftpd
- Mettre en place le DNS *DNS HOWTO*

Si vous n'êtes pas certain de la marche à suivre pour effectuer une des actions précédentes, il est FORTEMENT recommandé que vous suiviez les liens proposés pour vous familiariser avec tous ces paquetages. Je ne répondrai à AUCUN mail concernant les points précédents. Veuillez s'il vous plaît rediriger toute question à l'auteur du HOWTO approprié.

1.2 But

Le but des services virtuels est de permettre à une seule machine de reconnaître de multiples adresses IP sans avoir de multiples cartes réseau. L'IP aliasing est une option du noyau qui vous permet d'assigner plus d'une adresse IP à chaque périphérique réseau. Le noyau multiplexe (les échange très rapidement) en tâche de fond et l'utilisateur a l'impression d'avoir plusieurs cartes réseau dans sa machine.

Ce multiplexage permet à de multiples domaines (www.domaine1.com, www.domaine2.com, etc.) d'être logés sur la même machine pour le même coût que pour un seul domaine. Malheureusement, la plupart des services (ftp, web, courrier électronique) n'ont pas été conçus pour gérer de multiples domaines. Afin de les faire fonctionner correctement, vous devrez modifier à la fois les fichiers de configuration et le code source. Ce document décrit comment réaliser ces modifications pour la mise en place d'une machine virtuelle.

Un démon est également nécessaire afin de faire fonctionner les services virtuels. Le code source de ce démon (virtuald) est fourni plus loin dans ce document.

1.3 Commentaires

Ce document va grossir au fur et à mesure que les paquetages seront mis à jour et que le code source ou que les modifications proposées changeront. Si quelque partie de ce document n'est pas claire, envoyez-moi vous questions ou suggestions. Afin que je n'aie pas à chercher dans le HOWTO en entier, assurez vous s'il vous plaît que les commentaires soient aussi spécifiques que possible et incluent la section où se trouve le point discutable. Il est important que tout mail ait un champ Subject: contenant VIRTSERVICES HOWTO. Tout autre mail sera considéré comme du mail personnel, et tous mes amis savent que je ne lis pas tout le temps mon mail personnel et il risque donc d'être effacé avec le leur.

Veuillez également noter que mes exemples ne sont pas autre chose que des exemples, et ne doivent pas être recopiés tels quels. Il se peut que vous ayez à insérer vos propres valeurs. Si vous rencontrez des problèmes, envoyez moi un mail, contenant tous les fichiers de configuration pertinents et les messages d'erreur que vous avez obtenu lors de l'installation. Je vous renverrai mes suggestions.

1.4 Historique des changements

V1.0 Document initial.

V1.1 Correction d'une erreur dans la section sur le web virtuel.

V1.2 Correction de la date.

V2.0

Mise à jour des liens html.

Mise à jour de la section Web.

Nouvelle option pour sendmail.

Nouvelle option pour Qmail.

Mise à jour de la section Syslogd.

Mise à jour de la section FTP.

Option par défaut de Virtuald.

Nouvelle section Samba.

Mise à jour de la FAQ.

V2.1

Tous les paths ont été changés pour /usr/local.

Ajout de l'option de compilation VERBOSELOG.

Correction d'un bug de setuid/setgid dans virtmailfilter.

Correction du bug de execl dans virtmailfilter

Correction du bug de captalization dans virtmailfilter.

Suppression du code mbox de virtmailfilter/virtmaildelivery.

Ajout d'une section tcpserver.init pop pour Qmail

Ajout de la question alias domain name à la FAQ.

Correction de virtmailfilter pour envoyer le répertoire home à virtmaildelivery.

1.5 Copyright/Distribution

Ce document est Copyright (c) 1997 par The Computer Resource Center Inc.

Une copie de ce document peut être reproduite ou distribuée sur n'importe quel support physique ou électronique sans la permission de l'auteur. De la même façon, les traductions sont autorisées sans permission expresse si elle incluent un mot disant qui l'a traduit (NdT : voir le début du document pour mes coordonnées). Une redistribution commerciale est autorisée et encouragée. Dans ce cas cependant, faites-en part à *Computer Resource Center* .

Vous pouvez utiliser des extraits de ce document sans accord de l'auteur, à condition que l'oeuvre dérivée contienne une copie de ce document ou un pointeur vers une copie de ce document.

Vous avez la permission d'effectuer des copies ainsi que de les distribuer à condition que le paragraphe sur le copyright ainsi que cette note soient préservés sur toutes les copies.

En bref, nous désirons promouvoir la dissémination de cette information par quelque moyen que ce soit. Cependant, je désire conserver le copyright sur ce document, et aimerait être tenu au courant de tous les plans de redistribution de ce HOWTO.

2 IP aliasing

L'IP aliasing est une option du noyau qui doit être mise en place afin de pouvoir faire tourner des services virtuels sur une machine. Il existe déjà un mini-HOWTO expliquant l' *IP aliasing* . Référez vous à ce document pour toute question concernant la mise en place de cette option.

3 Virtuald

3.1 Comment ça marche

Toute connexion réseau est composée de deux paires adresse IP/port. L'API (Applications Program Interface, ou Interface de Programmation d'Applications) pour la programmation réseau est nommée l'API Sockets. La socket agit comme un fichier ouvert, et vous pouvez envoyer ou recevoir des données à travers une connexion réseau en lisant ou en écrivant dans la socket. Il existe une fonction, `getsockname`, qui retourne l'adresse IP de la socket locale. Virtuald utilise `getsockname` pour déterminer sur quel adresse IP de la machine locale la connexion a été faite. Virtuald lit un fichier de configuration pour récupérer le répertoire associé à cette adresse IP. Il va utiliser `chroot` sur ce répertoire et prendre en compte la connexion au service. `Chroot` change le répertoire / (le répertoire root) vers un nouveau point, de sorte que tout ce qui est au dessus de ce répertoire devienne inaccessible pour le programme. Ainsi, chaque adresse IP se voit assigné un système virtuel de fichiers. Pour le programme réseau, ceci est transparent, et le programme va se comporter comme si de rien n'était. Virtuald, en conjonction avec un programme comme inetd, peut être utilisé pour virtualiser n'importe quel service.

3.2 inetd

Inetd est un super serveur réseau qui écoute sur de multiples ports et, lorsqu'il reçoit une demande de connexion (par exemple, une requête POP), inetd réalise la connexion et l'envoie au programme spécifié. Cela évite de faire tourner des serveurs pour rien lorsqu'il n'y a aucune demande pour eux

Un fichier `/etc/inetd.conf` standard ressemble à ceci :

```
ftp stream tcp nowait root /usr/sbin/tcpd \
        wu.ftpd -l -a
pop-3 stream tcp nowait root /usr/sbin/tcpd \
        in.qpop -s
```

Un fichier `/etc/inetd.conf` virtualisé ressemble à ceci :

```
ftp stream tcp nowait root /usr/bin/virtuald virtuald /virtual/conf.ftp wu.ftpd -l -a
pop-3 stream tcp nowait root /usr/bin/virtuald virtuald /virtual/conf.pop in.qpop -s
```

3.3 Fichier de Configuration

Chaque service se voit attribué un fichier de configuration qui contrôlera quelles IPs et quels répertoires sont autorisés pour ce service. Vous pouvez avoir un fichier de configuration principal ou de nombreux fichiers de configuration si vous désirez que chaque service se voit attribuer une liste de domaines différents. Un fichier de configuration ressemble à ceci :

```
# C'est un commentaire, comme le sont les lignes blanches

# Format IP "ESPACE" dir "PAS D'ESPACES"
10.10.10.129 /virtual/foo.bar.com
10.10.10.130 /virtual/bar.foo.com
10.10.10.157 /virtual/boo.la.com
```

3.4 Le code source de virtuald

Ceci est un code source en C du programme virtuald. Compilez-le et installez-le dans /usr/local/bin avec les permissions 0755, l'utilisateur root, et le groupe root. La seule option de compilation est VERBOSELOG qui active ou désactive l'option de log.

```
#include <netinet/in.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <stdarg.h>
#include <unistd.h>
#include <string.h>
```

```
#include <syslog.h>
#include <stdio.h>

#define BUFSIZE 8192
int getipaddr(char **ipaddr)
{
        struct sockaddr_in virtual_addr;
        static char ipaddrbuf[BUFSIZE];
        int virtual_len;
        char *ipptr;
        virtual_len=sizeof(virtual_addr);
        if (getsockname(0,(struct sockaddr *)&virtual_addr,&virtual_len)        {
                syslog(LOG_ERR,"getipaddr: getsockname failed: %m");
                return -1;
        }
        if (!(ipptr=inet_ntoa(virtual_addr.sin_addr)))
        {
                syslog(LOG_ERR,"getipaddr: inet_ntoa failed: %m");
                return -1;
        }
        strncpy(ipaddrbuf,ipptr,sizeof(ipaddrbuf)-1);
        *ipaddr=ipaddrbuf;
        return 0;
}

int iptodir(char **dir,char *ipaddr,char *filename)
{
        char buffer[BUFSIZE],*bufptr;
        static char dirbuf[BUFSIZE];
        FILE *fp;

        if (!(fp=fopen(filename,"r")))
        {
                syslog(LOG_ERR,"iptodir: fopen failed: %m");
                return -1;
        }
        *dir=NULL;
        while(fgets(buffer,BUFSIZE,fp))
        {
                buffer[strlen(buffer)-1]=0;
                if (*buffer=='#' || *buffer==0)
                        continue;
                if (!(bufptr=strchr(buffer,' ')))
                {
                        syslog(LOG_ERR,"iptodir: strchr failed");
                        return -1;
                }
                *bufptr++=0;
                if (!strcmp(buffer,ipaddr))
                {
```

```
                        strncpy(dirbuf,bufptr,sizeof(dirbuf)-1);
                        *dir=dirbuf;
                        break;
                }
                if (!strcmp(buffer,"default"))
                {
                        strncpy(dirbuf,bufptr,sizeof(dirbuf)-1);
                        *dir=dirbuf;
                        break;
                }
        }
        if (fclose(fp)==EOF)
        {
                syslog(LOG_ERR,"iptodir: fclose failed: %m");
                return -1;
        }
        if (!*dir)
        {
                syslog(LOG_ERR,"iptodir: ip not found in conf file");
                return -1;
        }
        return 0;
}
int main(int argc,char **argv)
{
        char *ipaddr,*dir;
        openlog("virtuald",LOG_PID,LOG_DAEMON);
#ifdef VERBOSELOG
        syslog(LOG_ERR,"Virtuald Starting: $Revision: 1.1.1.1 $");
#endif
        if (!argv[1])
        {
                syslog(LOG_ERR,"invalid arguments: no conf file");
                exit(0);
        }
        if (!argv[2])
        {
                syslog(LOG_ERR,"invalid arguments: no program to run");
                exit(0);
        }
        if (getipaddr(&ipaddr))
        {
                syslog(LOG_ERR,"getipaddr failed");
                exit(0);
        }
#ifdef VERBOSELOG
        syslog(LOG_ERR,"Incoming ip: %s",ipaddr);
#endif
        if (iptodir(&dir,ipaddr,argv[1]))
        {
```

```
                syslog(LOG_ERR,"iptodir failed");
                exit(0);
        }
        if (chroot(dir)<0)
        {
                syslog(LOG_ERR,"chroot failed: %m");
                exit(0);
        }
#ifdef VERBOSELOG
        syslog(LOG_ERR,"Chroot dir: %s",dir);
#endif
        if (chdir("/")<0)
        {
                syslog(LOG_ERR,"chdir failed: %m");
                exit(0);
        }
        if (execvp(argv[2],argv+2)<0)
        {
                syslog(LOG_ERR,"execvp failed: %m");
                exit(0);
        }

        closelog();

        exit(0);
}
```

4 Scripts shell

4.1 virtfs

Chaque domaine doit avoir une arborescence de répertoires. Puisque vous utilisez `chroot`, vous aurez besoin de copies multiples des librairies, binaires, fichiers de configuration, etc. J'utilise le répertoire `/virtual/domaine1.com` pour chaque domaine que je crée.

Je comprends bien que cela représente du gaspillage d'espace disque, mais l'espace disque est meilleur marché qu'une nouvelle machine ou que des cartes réseau. Si vous désirez réellement sauver de l'espace disque, vous pouvez faire des liens, afin qu'une seule copie de chaque binaire soit présente. Le systeme de fichiers que j'utilise prend un peu plus de 2Mo. Le script essaye de copier tous les fichiers du système de fichiers principal pour que ce soit le plus identique possible.

Voici un fichier virtfs d'exemple :

```
#!/bin/bash

echo '$Revision: 1.1.1.1 $'

echo -n "Saisissez le nom de domaine : "
read domain

if [ "$domain" = "" ]
```

```
then
        echo Vous n’avez rien saisi : on arrête là
        exit 0
fi

leadingdir=/virtual

echo -n "Saisissez le nom du répertoire contenant les domaines (défaut: $leadingdir): "
read ans

if [ "$ans" != "" ]
then
        leadingdir=$ans
fi

newdir=$leadingdir/$domain

if [ -d "$newdir" ]
then
        echo Le répertoire $newdir existe déjà
        exit 0
else
        echo Nouveau répertoire : $newdir
fi

echo Création de $newdir
mkdir -p $newdir

echo Création de bin
cp -pdR /bin $newdir

echo Création de dev
cp -pdR /dev $newdir

echo Création de dev/log
ln -f /virtual/log $newdir/dev/log

echo Création d’etc
mkdir -p $newdir/etc
for i in /etc/*
do
        if [ -d "$i" ]
        then
                continue
        fi
        cp -pd $i $newdir/etc
done

echo Création de etc/skel
mkdir -p $newdir/etc/skel
```

```
echo Création de home
for i in a b c d e f g h i j k l m n o p q r s t u v w x y z
do
        mkdir -p $newdir/home/$i
done

echo Création de home/c/crc
mkdir -p $newdir/home/c/crc
chown crc.users $newdir/home/c/crc

echo Création de lib
mkdir -p $newdir/lib
for i in /lib/*
do
        if [ -d "$i" ]
        then
                continue
        fi
        cp -pd $i $newdir/lib
done

echo Création de proc
mkdir -p $newdir/proc

echo Création de sbin
cp -pdR /sbin $newdir

echo Création de tmp
mkdir -p -m 0777 $newdir/tmp
chmod +t $newdir/tmp

echo Création de usr
mkdir -p $newdir/usr

echo Création de usr/bin
cp -pdR /usr/bin $newdir/usr

echo Création de usr/lib
mkdir -p $newdir/usr/lib

echo Création de usr/lib/locale
cp -pdR /usr/lib/locale $newdir/usr/lib

echo Création de usr/lib/terminfo
cp -pdR /usr/lib/terminfo $newdir/usr/lib

echo Création de usr/lib/zoneinfo
cp -pdR /usr/lib/zoneinfo $newdir/usr/lib
```

```
echo Création de usr/lib/\*.so\*
cp -pdR /usr/lib/*.so* $newdir/usr/lib

echo Création de usr/sbin
cp -pdR /usr/sbin $newdir/usr

echo Lien de usr/tmp vers /tmp
ln -s /tmp $newdir/usr/tmp

echo Création de var
mkdir -p $newdir/var

echo Création de var/lock
cp -pdR /var/lock $newdir/var

echo Création de var/log
mkdir -p $newdir/var/log

echo Création de var/log/wtmp
cp /dev/null $newdir/var/log/wtmp

echo Création de var/run
cp -pdR /var/run $newdir/var

echo Création de var/run/utmp
cp /dev/null $newdir/var/run/utmp

echo Création de var/spool
cp -pdR /var/spool $newdir/var

echo Lien de var/tmp vers /tmp
ln -s /tmp $newdir/var/tmp

echo Création de var/www/html
mkdir -p $newdir/var/www/html
chown webmast.www $newdir/var/www/html
chmod g+s $newdir/var/www/html

echo Création de var/www/master
mkdir -p $newdir/var/www/master
chown webmast.www $newdir/var/www/master

echo Création de var/www/server
mkdir -p $newdir/var/www/server
chown webmast.www $newdir/var/www/server

exit 0
```

4.2 Virtexec

Afin d'exécuter des commandes dans un environnement virtuel, vous devez utiliser `chroot` sur ce répertoire puis lancer la commande. J'ai écrit un script shell nommé virtexec se chargeant de ces opérations, pour n'importe quelle commande :

```
#!/bin/sh

echo '$Revision: 1.1.1.1 $'

BNAME=`basename $0`
FIRST4CHAR=`echo $BNAME | cut -c1-4`
REALBNAME=`echo $BNAME | cut -c5-`

if [ "$BNAME" = "virtexec" ]
then
        echo Vous ne pouvez pas lancer virtexec directement. Il FAUT un lien symbolique
        exit 0
fi

if [ "$FIRST4CHAR" != "virt" ]
then
        echo Le lien ne pointe pas sur une fonction virtuelle
        exit 0
fi

list=""
num=1
for i in /virtual/*
do
        if [ ! -d "$i" ]
        then
                continue
        fi
        if [ "$i" = "/virtual/lost+found" ]
        then
                continue
        fi
        list="$list $i $num"
        num=`expr $num + 1`
done

if [ "$list" = "" ]
then
        echo Je ne trouve pas d'environnement virtuel
        exit 0
fi

dialog --clear --title 'Virtexec' --menu Pick 20 70 12 $list 2> /tmp/menu.$$
if [ "$?" = "0" ]
then
```

```
        newdir=`cat /tmp/menu.$$`
else
        newdir=""
fi
tput clear
rm -f /tmp/menu.$$

echo '$Revision: 1.1.1.1 $'

if [ ! -d "$newdir" ]
then
        echo Le nouveau répertoire $newdir N'EXISTE PAS
        exit 0
else
        echo Nouveau répertoire : $newdir
fi

echo bname: $BNAME

echo realbname: $REALBNAME

if [ "$*" = "" ]
then
        echo arguments: aucun
else
        echo args: $*
fi

echo Changement de répertoire vers $newdir
cd $newdir

echo Lancement de $REALBNAME

chroot $newdir $REALBNAME $*

exit 0
```

Veuillez noter que vous devez disposer du programme `dialog` sur votre système pour que ce script fonctionne. Pour utiliser virtexec, créez un lien symbolique d'un programme vers celui-ci. Par exemple :

```
ln -s /usr/local/bin/virtexec /usr/local/bin/virtpasswd
ln -s /usr/local/bin/virtexec /usr/local/bin/virtvi
ln -s /usr/local/bin/virtexec /usr/local/bin/virtpico
ln -s /usr/local/bin/virtexec /usr/local/bin/virtemacs
ln -s /usr/local/bin/virtexec /usr/local/bin/virtmailq
```

A présent, si vous tapez `virtvi` ou `virtpasswd` ou encore `virtmailq`, cela vous permettra d'éditer un fichier, changer le mot de passe d'un utilisateur, ou vérifier la file d'attente de mail sur votre système virtuel. Vous pouvez créer autant de liens vers virtexec que vous le désirez. Cependant, notez bien que si votre programme nécessite une librairie partagée, celle-ci doit se trouver sur le système de fichiers virtuel, ainsi que les binaires.

4.3 Notes

J'installe tous les scripts dans `/usr/local/bin`. Tout ce que je ne désire pas mettre sur le système de fichiers virtuel, je le place dans `/usr/local`. Le script ne touche à rien dans ce répertoire lors de la copie. Les fichiers ne devant pas chevaucher plusieurs systèmes de fichiers virtuels doivent être supprimés. Par exemple, `ssh` est installé sur mon système, et je n'ai pas voulu que les clefs privées soient disponibles sur tous les systèmes de fichier. J'ai donc supprimé le fichier des systèmes de fichiers virtuels après avoir lancé `virtfs`. Je change également le `resolv.conf` et supprime tout ce qui contient le nom d'un autre domaine, pour des raisons légales. Par exemple, les fichiers `/etc/hosts` et `/etc/HOSTNAME`.

Les programmes pour lesquels je fais un lien symbolique vers virtexec sont :

- virtpasswd – changer le mot de passe d'un utilisateur
- virtadduser – ajouter un utilisateur
- virtdeluser – supprimer un utilisateur
- virtsmbstatus – consulter l'état de samba
- virtvi – éditer un fichier
- virtmailq – vérifier la mailq
- virtnewaliases – reconstruire la table des alias mail

5 DNS

Vous pouvez configurer le DNS normalement. Vous pouvez consulter le *DNS HOWTO* .

6 Syslogd

6.1 Problème

Syslog est l'outil de *logging* couramment utilisé sur les systèmes UNIX. C'est un démon qui ouvre un fichier spécial appelé FIFO. Une FIFO est un fichier spécial, se comportant comme une file d'attente. Tout ce qui y est écrit "ressortira" en lecture. Le démon syslog attend les données en lecture. Il existe des fonctions C pour écrire dans les FIFO. Si vous utilisez ces fonctions C dans vos programmes, la sortie ira vers syslod.

Souvenez vous que vous avez utilisé `chroot` et que la FIFO que syslog lit `/dev/log` ne se trouve pas dans l'environnement virtuel. Cela implique qu'aucun des environnements virtuels ne pourra utiliser `syslog`. Nous ne pouvons pas tout simplement copier le fichier, puisque les programmes utiliseraient `/dev/log` au lieu du nouveau que nous aurions créé.

6.2 Solution

Syslog peut scruter d'autres FIFO si vous le lui dites en ligne de commande. Lancez donc syslog avec l'argument :

```
syslog -p /virtual/log
```

Faites alors un lien de `/dev/log` vers `/virtual/log` (un lien symbolique) :

```
ln -sf /virtual/log /dev/log
```

Puis liez toutes les copies de `/dev/log` vers ce fichier avec la commande (attention, c'est un lien NON symbolique) :

```
ln /virtual/log /virtual/domain.com/dev/log
```

Le script virtfs ci-dessus le fait pour vous. Puisque `/virtual` est un disque entier, et que les `/dev/log` sont liés, ils ont le même numéro d'inode et pointent vers les mêmes données. Le `chroot` ne peut pas empêcher cela, et donc tous vos `/dev/log` virtuels vont à présent fonctionner. Notez également que tous les messages de toutes les machines virtuelles seront écrits dans un même fichier. Cependant, vous pouvez écrire des programmes pour filtrer les données.

6.2.1 Syslogd.init

Cette version du fichier `syslog.init` refait les liens vers les `/dev/log` à chaque fois que vous le lancez. Voici un `syslog.init` modifié :

```
#!/bin/sh

# Source function library.
. /etc/rc.d/init.d/functions

case "$1" in
  start)
        echo -n "Starting dev log: "
        ln -sf /virtual/log /dev/log
        echo done
        echo -n "Starting system loggers: "
        daemon syslogd -p /virtual/log
        daemon klogd
        echo
        echo -n "Starting virtual dev log: "
        for i in /virtual/*
        do
                if [ ! -d "$i" ]
                then
                        continue
                fi
                if [ "$i" = "/virtual/lost+found" ]
                then
                        continue
                fi
                ln -f /virtual/log $i/dev/log
                echo -n "."
        done
        echo " done"
        touch /var/lock/subsys/syslog
        ;;
  stop)
```

```
        echo -n "Shutting down system loggers: "
        killproc syslogd
        killproc klogd
        echo
        rm -f /var/lock/subsys/syslog
        ;;
  *)
        echo "Usage: syslog {start|stop}"
        exit 1
esac

exit 0
```

6.3 Plusieurs syslod

6.3.1 Un par disque

Si vous manquez de place sur un système de fichiers, et que vous devez séparer vos domaines virtuels en plusieurs disques, rappellez-vous que les liens (non symboliques) ne peuvent pas passer à travers plusieurs disques. Ce qui implique de devoir lancer un syslogd pour chaque groupe de domaine par disque. Par exemple, si vous avez 13 domaines sur /virtual1 et 15 sur /virtual2, vous devrez faire un lien pour les 13 domaines sur /virtual1/log et lancer syslogd avec `syslogd -p /virtual1/log` , ainsi qu'un lien pour les 15 domaines sur /virtual2/log et lancer `syslogd -p /virtual2/log` .

6.3.2 Un par domaine

Si vous ne voulez pas centrer les logs sur un seul endroit, vous pouvez aussi lancer un syslogd par domaine. Cela donne des pertes de processus ID, donc je ne le recommande pas, mais c'est facile à mettre en oeuvre. Vous devriez modifier votre fichier syslod.init pour lancer syslogd par `chroot /virtual/domain1.com syslogd` pour chaque domaine. Ceci lancera syslogd dans le `chroot` et les logs se trouveront dans /virtual/domain1.com/var/log au lieu d'être tous rassemblés dans /var/log. N'oubliez pas de lancer syslod normalement `syslod` pour le système principal ainsi qu'un logger pour le noyau `klogd` .

7 FTP virtuel

7.1 Inetd

Wu-ftpd intégre en standard le support des domaines virtuels. Cependant, vous ne pouvez pas utiliser des fichiers de mot de passe différents pour chaque domaine. Par exemple, si `bob@domaine1.com` et `bob@domaine2.com` désirent tous les deux un compte, vous devrez donner à l'un des deux le nom `bob2` ou demander à un des utilisateurs de choisir un autre nom de login. Puisque vous disposez à présent de systèmes de fichiers différents pour chaque domaine, vous disposez de fichiers de mot de passe différents et ce problème disparaît. Vous n'avez qu'à créer un script `virtnewuser` et `virtpasswd` de la façon qui est expliquée ci-dessus, et tout fonctionnera.

Les entrées pour wu-ftp dans `inetd.conf` sont :

```
ftp stream tcp nowait root /usr/local/bin/virtuald \
        virtuald /virtual/conf.ftp wu.ftpd -l -a
```

7.2 Les FTP anonymes

Ils ne sont pas affectés par la presence de virtuald. Pour un utilisateur anonyme, créer l'utilisateur FTP dans /virtual/domain1.com/etc/passwd comme vous le feriez d'habitude.

```
ftp:x:14:50:Anonymous FTP:/var/ftp:/bin/false
```

Puis l'installation du répertoire anonyme du FTP. Vous avez des fichier de mot de passe séparés, donc vous pouvez restreindre n'importe quel domaine à un FTP avec un compte anonyme. Notez que comme le serveur à un `chroot` dans le répertoire /virtual/domain1.com vous n'avez pas a préfixer de chemins avec.

7.3 Utilisateurs de FTP Virtuel

Wu-ftpd supporte quelque chose qui s'appelle un groupe d'invité (guest group). Ça permet de créer différentes zones FTP pour chaque utilisateur. Le serveur FTP fait un `chroot` vers la zone specifiée, donc, l'utilisateur ne peut sortir du répertoire. Si vous créez un utilisateur dans un domaine virtuel, il ne sera pas capable de voir le système de fichiers Système.

Ajouter le group guest au fichier /virtual/domain1.com/etc/ftpaccess.

Créer une entrée dans /virtual/domain1.com/etc/passwd avec `chroot` et le répertoire home de départ séparé par `/./` :

```
guest1:x:8500:51:Guest FTP:/home/g/guest1/./incoming:/bin/false
```

Puis installer le home du guest comme vous feriez pour un FTP anonyme. Vous avez des fichiers de mots de passe separés pour chaque domaine, donc vous pouvez spécifier quel domaine dispose d'un compte guest et les utilisateurs qui sont des utilisateurs guest dans un domaine. Notez que depuis que le serveur FTP est `chrooté` vers le repertoire /virtual/domain1.com vous n'avez pas à préfixer de chemin.

8 Web virtuel

8.1 Lancement avec virtuald

8.1.1 Non recommandé

Apache supporte en standard la gestion des domaines virtuels. C'est d'ailleurs le seul programme pour lequel je recommande d'utiliser le système de gestion des domaines virtuels fourni avec. Lorsque vous lancez un programme par l'intermédiaire d'inetd, il y a un coût supplémentaire, puisque le programme doit démarrer à chaque fois qu'il y a une demande de connexion, et vous obtenez des temps de réponse beaucoup plus longs, inacceptables pour le web. Apache intégre également un mécanisme pour stopper les connexions lorsqu'elles sont trop nombreuses.

Comme il est simplement indiqué ci-dessus, rendre virtuel Apache avec virtuald est une tres mauvaise idée. Le but de virtuald est de combler la lacune des serveurs qui n'ont pas leur propre systeme interne pour faire ce travail. Virtuald n'est pas fait pour remplacer du bon code qui remplit déjà la tâche.

Ce qui suit ne restera pas ici, c'est pour expliquer à ceux qui sont assez idiots pour le faire.

8.1.2 Inetd

Éditez /etc/inetd.conf

```
vi /etc/inetd.conf # Ajouter cette ligne
www stream tcp nowait www /usr/local/bin/virtuald \
        virtuald /virtual/conf.www httpd -f /var/www/conf/httpd.conf
```

8.1.3 Httpd.conf

Éditez /var/www/conf/httpd.conf

```
vi /var/www/conf/httpd.conf # Où l'emplacement des fichiers de configuration d'Apache
Il doit y avoir :
ServerType standalone

Remplacez-le par :
ServerType inetd
```

8.1.4 Configuration

Ensuite, configurez chaque cas du serveur Apache comme si vous n'aviez qu'un seul domaine.

8.1.5 Httpd.init

Un fichier httpd.init n'est pas nécéssaire si le serveur est lancé par inetd.

8.2 Lancer Apache avec VirtualHost

Apache a trois fichiers de configuration `access.conf` , `httpd.conf` et `srm.conf` . De nouvelles versions d'Apache ont rendues les trois fichiers de configuration inutiles. Ainsi, je trouve que séparer en trois sections la configuration la rend plus simple à gérer, donc je garderai ce style dans le HOWTO.

8.2.1 Access.conf

Ce fichier de configuration est utilisé pour contrôler l'accès aux répertoires dans la structure du répertoire web. Voici un exemple de fichier de configuration qui montre comment avoir plusieurs options pour chaque domaine.

```
# /var/www/conf/access.conf: Configuration des accès globaux

# Les options sont heritées du répertoire précedent
# Mettre les options par défaut pour le répertoire principal
<Directory />
AllowOverride None
Options Indexes
</Directory>

# Créer un répertoire protegé par mot de passe pour un domaine
```

```
<Directory /virtual/domain1.com/var/www/html/priv>
AuthUserFile /var/www/passwd/domain1.com-priv
AuthGroupFile /var/www/passwd/domain1.com-priv-g
AuthName PRIVSECTION
AuthType Basic
<Limit GET PUT POST>
require valid-user
</Limit>
</Directory>

# Créer un autre domaine Server Side Include (SSI)
<Directory /virtual/domain2.com/var/www/html>
Options IncludesNOEXEC
</Directory>
```

8.2.2 Httpd.conf

Ce fichier de configuration est utilisé pour contrôler les options principales du serveur Apache. Voici un exemple de fichier de configuration qui montre comment avoir différentes options pour chaque domaine.

```
# /var/www/conf/httpd.conf: Fichier de configuration principal du serveur

# Début: Section principale
ServerType standalone

# Numéro du port
Port 80

# Log des clients avec le nom et l'IP
HostnameLookups on

# Utilisateur qui lance le serveur
User www
Group www

# Emplacement des fichiers de config, erreurs et log
ServerRoot /var/www

# Processus ID du serveur dans ce fichier
PidFile /var/run/httpd.pid

# Informations du processus interne du serveur
ScoreBoardFile /var/www/logs/apache_status

# Les options du Timeout et KeepAlive
Timeout 400
KeepAlive 5
KeepAliveTimeout 15

# Nombre de Serveur à lancer
```

```
MinSpareServers 5
MaxSpareServers 10
StarsServers 5
MaxClients 150
MaxRequestsPerChild 30

# Fin: Section de configuration principale

# Début: Section de l'hébergement virtuel

# Indique au serveur d'accepter les connexions pour IP:Port
# Il y a une ligne pour chaque IP nécessaire donc, vous pouvez ignorer certains
# domaines
Listen 10.10.10.129:80
Listen 10.10.10.130:80

# La commande VirtualHost permet de spécifier un autre domaine virtuel sur le
# serveur. La plupart des options d'Apache peuvent être spécifiées dans cette
# section.
<VirtualHost www.domain1.com>

# E-mail à laquelle sont envoyées les erreurs
ServerAdmin webmaster@domain1.com

# Endroit où sont mis les documents du domaine virtuel
DocumentRoot /virtual/domain1.com/var/www/html

# Nom du serveur
ServerName www.domain1.com

# Fichiers de Log relatifs à l'option ServerRoot
ErrorLog logs/domain1.com-error_log
TransferLog logs/domain1.com-access_log
RefererLog logs/domain1.com-referer_log
AgentLog logs/domain1.com-agent_log

# Utiliser les scripts CGI dans ce domaine
ScriptAlias /cgi-bin/ /var/www/cgi-bin/domain1.com/
AddHandler cgi-script .cgi
AddHandler cgi-script .pl
</VirtualHost>

<VirtualHost www.domain2.com>

# E-mail à laquelle sont envoyées les erreurs
ServerAdmin webmaster@domain2.com

# Endroit où sont mis les documents du domaine virtuel
DocumentRoot /virtual/domain2.com/var/www/html
```

```
# Nom du Serveur
ServerName www.domain2.com

# Fichiers de Log relatifs à l'option ServerRoot
ErrorLog logs/domain2.com-error_log
TransferLog logs/domain2.com-access_log
RefererLog logs/domain2.com-referer_log
AgentLog logs/domain2.com-agent_log

# Pas de script CGI pour ce domaine
</VirtualHost>
# Fin: Section de l'hebergement virtuel
```

8.2.3 Srm.conf

Ce fichier de configuration est utilisé pour contrôler comment sont servies les demandes et comment sont formattés les résultats. Vous n'avez pas besoin d'éditer quoi que ce soit ici pour les domaines virtuels. Le fichier de configuration de base d'Apache doit fonctionner.

8.2.4 Httpd.init

Rien de spécial n'est à faire dans ce fichier. Utilisez la version de base qui est fournie avec Apache.

8.3 Descripteurs de fichiers : limite de capacité

8.3.1 Attention

Cela s'applique seulement à la version standalone du serveur Apache. Un serveur qui se lance au travers d'inetd n'intervient pas avec les autres domaines et à accès à toute la table des descripteurs de fichiers.

Chaque fichier de log qu'ouvre Apache est un autre descripteur de fichier pour le processus. Il y a une limite de 256 descripteurs de fichier par processus au coeur du systeme Linux. Depuis que vous avez plusieurs domaines, vous utilisez plus de descripteurs de fichiers. Si vous avez trop de domaines tournant sur un processus du serveur Apache, vous pouvez engorger cette table. Cela peut impliquer que certains logs ne fonctionneront pas et que certains CGI seront interrompus.

8.3.2 Plusieurs serveur Apache

Si vous prévoyez cinq descripteurs de fichiers par domaine, vous pouvez avoir 50 domaines tournant sur votre serveur Apache sans problèmes. Mais, si votre serveur a des problèmes, vous pouvez créer /var/www1 avec un serveur Apache qui s'occupe des domaines 1 à 25 et /var/www2 avec un autre serveur qui s'occupe des domaines 26 à 50. Ainsi, chaque serveur aura son propre fichier de configuration, d'erreurs et de log. Chaque serveur doit être configuré separement avec ses propres directives de Listen et VirtualHost. Et n'oubliez pas de lancer plusieurs serveurs dans votre fichier httpd.ini.

8.4 Héberger plusieurs serveurs avec une IP

8.4.1 Économiser des IP

La version 1.1 du protocole HTTP (HyperText Transfer Protocol) inclue une fonction qui communique le nom du serveur au client. Ce qui implique que le client n'a pas besoin de rechercher le nom du serveur à partir de son adresse IP. Comme ça, deux serveurs virtuels peuvent avoir la même adresse IP et être deux site Web différents. La configuration d'Apache est la même qu'avant, à part que vous n'avez pas besoin de mettre plusieurs directives Listen comme les deux domaines ont la même IP.

8.4.2 Inconvénient

Le seul problème est que virtuald utilise les adresses IP pour distinguer les domaines. Dans sa forme actuelle Virtuald ne serait pas capable de `chroot` vers un répertoire de mail (spool) pour chaque domaine. Donc, les mails ne peuvent etre reçus que sur une IP et il n'y aurait plus un répertoire spool pour chaque domaine. Tous les clients d'un serveur web partageant une IP devront se partager le meme repertoire spool. Ce qui signifierait que dupliquer les noms d'utilisateurs serait encore une solution. Enfin, c'est le prix à payer pour économiser une IP.

8.5 Plus d'informations

Ce HOWTO montre seulement comment implémenter le support virtuel sur le serveur Web Apache. La plupart des serveur Web utilisent une interface similaire. Pour plus d'informations sur l'hébergement de web virtuel, consultez le *WWW-HOWTO* , la documentation d'Apache sur le *Site d'Apache* , ou la documentation sur *ApacheWeek* .

9 Mail virtuel avec Pop

9.1 Problème

Le support du mail virtuel est une demande toujours grandissante. Sendmail affirme qu'il supporte le mail virtuel. En fait, il se contente d'être à l'écoute de mail pour différents domaines. Vous pouvez alors demander à faire suivre le mail quelque part. Cependant, si vous le faites suivre sur la machine locale et que vous avez du mail pour bob@domaine1.com et bob@domaine2.com, ils vont atteindre la même boîte. C'est un problème puisque les bob sont deux personnes différentes, avec du courrier électronique différent.

9.2 Solution

Vous pouvez vous assurer que chaque nom d'utilisateur est unique en utilisant une numérotation des noms d'utilisateurs : bob1, bob2, etc... Vous pourriez également hacker le mail et le pop pour que ces conversions soient invisibles, mais cela peut devenir désordonné. Le mail sortant à pour domaine domaineprincipal.com et vous désirez que chaque mail envoyé dans chaque sous-domaine ait une adresse From: différente.

J'ai deux solutions. L'une fonctionne avec sendmail et l'autre avec Qmail. La solution avec sendmail devrait fonctionner avec une installation standard de sendmail. Cependant elle partage toutes les limitations établies dans sendmail. Il est nécessaire aussi qu'un sendmail ait été lancé en mode de file d'attente (queue mode) pour chaque domaine. Avoir 50 ou plus processus sendmail en mode de file d'attente qui se réveillent toutes les heures peut ajouter des contraintes sur une machine.

La solution pour Qmail ne requiert pas plusieurs exemplaires de Qmail et peut n'utiliser qu'un seul répertoire de file d'attente. Il a besoin d'un autre programme puisque que Qmail ne se fonde pas sur virtuald. Je crois qu'une procédure similaire peut être faite avec Sendmail. Cependant, Qmail se prête plus aisément à cette solution.

Je ne cautionne aucun des deux programmes en particulier. L'installation de Sendmail est un peu moins complexe mais Qmail est pobablement le plus puissant des deux paquetages de serveur Mail.

9.3 Solution pour Sendmail

9.3.1 Introduction

Chaque système de fichiers virtuel fournit à chaque domaine un fichier `/etc/passwd`. Cela signifie que bob@domaine1.com et bob@domaine2.com sont des utilisateurs différents dans des fichiers `/etc/passwd` différents, donc le mail ne constituera aucun problème. Ils possèdent également chacun un spool de mail, donc les boîtes aux lettres seront des fichiers différents sur des système de fichiers virtuels différents.

9.3.2 Créer un fichier de configuration Sendmail

Créez `/etc/sendmail.cf` comme vous le feriez d'habitude avec m4 :

```
divert(0)
VERSIONID(`tcpproto.mc')
OSTYPE(linux)
FEATURE(redirect)
FEATURE(always_add_domain)
FEATURE(use_cw_file)
FEATURE(local_procmail)
MAILER(local)
MAILER(smtp)
```

9.3.3 Édition du fichier de configuration

Éditez /virtual/domain1.com/etc/sendmail.cf pour l'adapter à votre domaine virtuel :

```
vi /virtual/domain1.com/etc/sendmail.cf
```

Approximativement à la ligne 86 il doit y avoir :

```
#Dj$w.Foo.COM
```

Remplacez-le avec :

```
Djdomain1.com
```

9.3.4 Distribution locale par Sendmail

Éditez /virtual/domain1.com/etc/sendmail.cw

```
vi /virtual/domain1.com/etc/sendmail.cw
mail.domain1.com
domain1.com
domain1
localhost
```

9.3.5 Sendmail et les domaines virtuels : la bidouille (pre8.8.6)

Cependant, sendmail nécessite un changement mineur de son code source. Sendmail utilise un fichier nommé `/etc/sendmail.cw` qui contient tous les noms de machine pour lequel il distribuera le mail localement au lieu de le faire suivre à une autre machine. Sendmail fait une vérification interne de toutes les interfaces réseau de la machine pour initialiser cette liste avec les adresses IP locales. Cela présente un problème si vous envoyez des mails entre deux domaines virtuels de la même machine. Sendmail pensera que l'autre domaine virtuel est une adresse locale et il distribuera le mail localement. Par exemple, bob@domaine1.com envoie un mail à fred@domaine2.com. Puisque le sendmail de domaine1.com pense que domaine2.com est une adresse locale, il va envoyer ce mail à domaine1.com et ne l'enverra jamais à domaine2.com. Vous avez à modifier sendmail (ce que j'ai fait sans problème sur la version 8.8.6) :

```
vi v8.8.5/src/main.c
```

Vers la ligne 494 vous devriez remplacer la ligne :

```
load_if_names();
```

Par :

```
/* load_if_names(); Commenté puisque cela casse les domaines virtuels */
```

Notez que cette modification n'est nécessaire que si vous désirez envoyer du mail entre des domaines virtuels, ce qui est probable, je pense.

Cela corrigera le problème. Cependant, l'adaptateur réseau principal eth0 n'est pas supprimé. Ainsi, si vous envoyez un mail depuis une adresse IP virtuelle vers une adresse sur eth0 de la même machine, il sera délivré localement. Pour cela, j'utilise une adresse IP bidon virtuel1.domaine.com (10.10.10.157). Je n'envoie jamais de mail à cet hôte, les domaines virtuels non plus. C'est aussi l'adresse IP que j'utiliserai pour me connecter sur la machine via ssh, pour vérifier si le système fonctionne.

9.3.6 Sendmail et les domaines virtuels : Nouvelle fonction (POST8.8.6)

Depuis la version 8.8.6 de Sendmail, il existe une option qui désactive le chargement des interfaces réseaux supplémentaires. Ce qui implique la NON nécéssité de toucher au code source. Elle s'appelle `DontProbeInterfaces` .

Editer /virtual/domain1.com/etc/sendmail.cf

```
vi /virtual/domain1.com/etc/sendmail.cf
```

Ajouter la ligne :

```
O DontProbeInterfaces=True
```

9.3.7 Sendmail.init

Sendmail ne peut pas être lancé tel quel, vous allez devoir le lancer à travers inetd. C'est un moyen inefficace qui implique un temps de réponse plus long, mais si vous avez un site tellement occupé pour que la différence soit importante, alors vous devriez utiliser une machine dédiée à ce site. Notez que vous ne de devez PAS utiliser l'option `-bd`. Notez également que vous devez lancer `sendmail -q` pour chaque domaine que vous gérez. Le nouveau fichier `sendmail.init` est le suivant :

```
#!/bin/sh

# Source function library.
. /etc/rc.d/init.d/functions

case "$1" in
  start)
        echo -n "Starting sendmail: "
        daemon sendmail -q1h
        echo
        echo -n "Starting virtual sendmail: "
        for i in /virtual/*
        do
                if [ ! -d "$i" ]
                then
                        continue
                fi
                if [ "$i" = "/virtual/lost+found" ]
                then
                        continue
                fi
                chroot $i sendmail -q1h
                echo -n "."
        done
        echo " done"
        touch /var/lock/subsys/sendmail
        ;;
  stop)
        echo -n "Stopping sendmail: "
        killproc sendmail
        echo
        rm -f /var/lock/subsys/sendmail
        ;;
  *)
        echo "Usage: sendmail {start|stop}"
        exit 1
esac

exit 0
```

9.3.8 Configuration d'inetd

Pop devrait s'installer normalement, sans effort supplémentaire. Vous n'avez qu'à modifier l'entrée pour pop dans le fichier `inetd.conf` pour utiliser le démon virtuald. Pour sendmail et pop, cela donne :

```
pop-3 stream tcp nowait root /usr/bin/virtuald \
        virtuald /virtual/conf.pop in.qpop -s
smtp stream tcp nowait root /usr/bin/virtuald \
        virtuald /virtual/conf.mail sendmail -bs
```

9.4 Solution pour Qmail

9.4.1 Introduction

Cette solution prend la responsabilité de distribution du qmail-local, donc l'utilisation des fichiers .qmail dans les répertoire home des domaines virtuels ne marcheront pas. Cependant, chaque domaine aura toujours un utilisateur maître par domaine qui contrôlera les aliasing du domaine. Deux programmes externes seront utilisés pour le fichier `.qmail-default` des maîtres de domaine. Le mail passera par ces deux programmes afin de distribuer le courrier à chaque domaine.

Deux programmes sont necéssaires, car l'un deux est lancé avec le setuid root. C'est un petit programme qui se change en un utilisateur non administrateur et lance le second programme. Consultez le site le plus proche relatif à la sécurité pour une discussion sur le pourquoi est-ce nécessaire.

Cette solution se passe de virtuald. Qmail est assez flexible pour ne pas avoir besoin d'une configuration virtuald génerale. La conception de Qmail utilise l'enchaînement de programmes pour distribuer les mails. Cette conception facilite l'insertion d'une section virtuelle dans le processus de distribution de Qmail sans altérer une installation standard de Qmail.

Depuis que vous utilisez Qmail, tout nom de domaine non qualifié sera développé en utilisant le serveur principal. C'est dû au fait que vous n'avez pas un Qmail pour chaque domaine. Donc, soyez sûr que vos clients (Eudora, elm, mutt, etc.) puissent developper tous vos noms de domaines non qualifiés.

9.4.2 Installation des domaines virtuels

Qmail doit être configuré de manière à accepter les mails pour chaque domaine que vous désservez. Tapez la commande suivante :

```
echo "domain1.com:domain1" >> /var/qmail/control/virtualdomains
```

9.4.3 Installation de l'utilisateur maître du domaine

Ajouter à votre fichier `/etc/passwd` principal l'utilisateur domain1. Je choisirais le shell /bin/false pour que le maître du domaine ne puisse se connecter. Cet utilisateur sera capable d'ajouter des fichier .qmail et tous les mails passeront par ce compte. Notez que les noms d'utilisateurs ne peuvent faire que 8 caractères et les noms de domaines peuvent être plus long. Les caractères restants seront ignorés. Ce qui implique, que les utilisateurs domain12 et domain123 seraient le même utilisateur et Qmail pourra être perturbé. Donc, attention à votre convention pour nommer les utilisateurs maîtres des domaines.

Créer les fichiers .qmail du maître de domaine avec les commandes suivantes. Ajoutez les autres alias système au même endroit. Par exemple, webmaster ou hostmaster.

```
echo "user@domain1.com" > /home/d/domain1/.qmail-mail-daemon
echo "user@domain1.com" > /home/d/domain1/.qmail-postmaster
echo "user@domain1.com" > /home/d/domain1/.qmail-root
```

Créez le fichier .qmail-default du maître de domaine. Il filtre tous les mails du domaine virtuel.

```
echo "| /usr/local/bin/virtmailfilter" > /home/d/domain1/.qmail-default
```

9.4.4 Tcpserver

Qmail a besoin d'un pop spécial qui supporte le format maildir. Le programme pop doit être rendu virtuel. L'auteur de Qmail recommande d'utiliser `tcpserver` (un remplacement à inetd) avec Qmail, donc mes exemples utilisent tcpserver et NON inetd.

Tcpserver n'a pas besoin de fichier de configuration. Toutes les informations peuvent etre passées par une ligne de commande. Ci-dessous, se trouve le fichier tcpserver.ini que vous devez utiliser pour le démon mail et le serveur pop :

```
#!/bin/sh

. /etc/rc.d/init.d/functions

QMAILDUSER=`grep qmaild /etc/passwd | cut -d: -f3`
QMAILDGROUP=`grep qmaild /etc/passwd | cut -d: -f4`

# Regarder comment nous étions appellés.
case "$1" in
  start)
        echo -n "Starting tcpserver: "
        tcpserver -u 0 -g 0 0 pop-3 /usr/local/bin/virtuald \
                /virtual/conf.pop qmail-popup virt.domain1.com \
                /bin/checkpassword /bin/qmail-pop3d Maildir &
        echo -n "pop "
        tcpserver -u $QMAILDUSER -g $QMAILDGROUP 0 smtp \
                /var/qmail/bin/qmail-smtpd &
        echo -n "qmail "
        echo
        touch /var/lock/subsys/tcpserver
        ;;
  stop)
        echo -n "Stopping tcpserver: "
        killall -TERM tcpserver
        echo -n "killing "
        echo
        rm -f /var/lock/subsys/tcpserver
        ;;
  *)
        echo "Usage: tcpserver {start|stop}"
        exit 1
esac

exit 0
```

9.4.5 Qmail.init

Vous pouvez utiliser le script standard de Qmail.init fourni. Qmail est livré avec une très bonne documentation décrivant comment le mettre en place.

9.4.6 Source

Vous avez besoin de deux autres programmes pour que votre serveur mail virtuel fonctionne avec Qmail. Ce sont virtmailfilter et virtmaildelivery. Ceci est le code source en C de virtmailfilter. Il doit être installé dans /usr/local/bin avec les permissions 4750, l'utilisateur root et le groupe nofiles.

```
#include <sys/wait.h>
#include <unistd.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <ctype.h>
#include <pwd.h>

#define VIRTPRE                 "/virtual"

#define VIRTPWFILE              "etc/passwd"
#define VIRTDELIVERY            "/usr/local/bin/virtmaildelivery"
#define VIRTDELIVERY0           "virtmaildelivery"

#define PERM                    100
#define TEMP                    111
#define BUFSIZE                 8192

int main(int argc,char **argv)
{
        char *username,*usernameptr,*domain,*domainptr,*homedir;
        char virtpath[BUFSIZE];
        struct passwd *p;
        FILE *fppw;
        int status;
        gid_t gid;
        pid_t pid;

        if (!(username=getenv("EXT")))
        {
                fprintf(stdout,"environment variable EXT not set\n");
                exit(TEMP);
        }

        for(usernameptr=username;*usernameptr;usernameptr++)
        {
                *usernameptr=tolower(*usernameptr);
        }
```

```
if (!(domain=getenv("HOST")))
{
        fprintf(stdout,"environment variable HOST not set\n");
        exit(TEMP);
}

for(domainptr=domain;*domainptr;domainptr++)
{
        if (*domainptr=='.' && *(domainptr+1)=='.')
        {
                fprintf(stdout,"environment variable HOST has ..\n");
                exit(TEMP);
        }
        if (*domainptr=='/')
        {
                fprintf(stdout,"environment variable HOST has /\n");
                exit(TEMP);
        }

        *domainptr=tolower(*domainptr);
}

for(domainptr=domain;;)
{
        snprintf(virtpath,BUFSIZE,"%s/%s",VIRTPRE,domainptr);
        if (chdir(virtpath)>=0)
                break;

        if (!(domainptr=strchr(domainptr,'.')))
        {
                fprintf(stdout,"domain failed: %s\n",domain);
                exit(TEMP);
        }

        domainptr++;
}

if (!(fppw=fopen(VIRTPWFILE,"r+")))
{
        fprintf(stdout,"fopen failed: %s\n",VIRTPWFILE);
        exit(TEMP);
}

while((p=fgetpwent(fppw))!=NULL)
{
        if (!strcmp(p->pw_name,username))
                break;
}

if (!p)
```

```
        {
                fprintf(stdout,"user %s: not exist\n",username);
                exit(PERM);
        }

        if (fclose(fppw)==EOF)
        {
                fprintf(stdout,"fclose failed\n");
                exit(TEMP);
        }

        gid=p->pw_gid;
        homedir=p->pw_dir;

        if (setgid(gid)<0 || setuid(p->pw_uid)<0)
        {
                fprintf(stdout,"setuid/setgid failed\n");
                exit(TEMP);
        }

        switch(pid=fork())
        {
                case -1:
                        fprintf(stdout,"fork failed\n");
                        exit(TEMP);
                case 0:
                        if (execl(VIRTDELIVERY,VIRTDELIVERY0,username,homedir,NU
                                fprintf(stdout,"execl failed\n");
                                exit(TEMP);
                        }
                default:
                        if (wait(&status)<0)
                        {
                                fprintf(stdout,"wait failed\n");
                                exit(TEMP);
                        }
                        if (!WIFEXITED(status))
                        {
                                fprintf(stdout,"child did not exit normally\n");
                        break;
        }

        exit(WEXITSTATUS(status));
}
```

9.4.7 Source

Ceci est le code source de virtmaildelivery. Il doit etre installé dans /usr/local/bin avec les permissions 0755, l'utilisateur root et le groupe root.

```
#include <sys/stat.h>
#include <sys/file.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <stdio.h>
#include <errno.h>
#include <time.h>

#define TEMP                    111
#define BUFSIZE                 8192
#define ATTEMPTS                10

int main(int argc,char **argv)
{
        char *user,*homedir,*dtline,*rpline,buffer[BUFSIZE],*p,mail[BUFSIZE];
        char maildir[BUFSIZE],newmaildir[BUFSIZE],host[BUFSIZE];
        int fd,n,nl,i,retval;
        struct stat statp;
        time_t thetime;
        pid_t pid;
        FILE *fp;

        retval=0;

        if (!argv[1])
        {
                fprintf(stdout,"invalid arguments: need username\n");
                exit(TEMP);
        }

        user=argv[1];

        if (!argv[2])
        {
                fprintf(stdout,"invalid arguments: need home directory\n");
                exit(TEMP);
        }

        homedir=argv[2];

        if (!(dtline=getenv("DTLINE")))
        {
                fprintf(stdout,"environment variable DTLINE not set\n");
                exit(TEMP);
        }
        if (!(rpline=getenv("RPLINE")))
        {
                fprintf(stdout,"environment variable RPLINE not set\n");
                exit(TEMP);
```

```
}

while (*homedir=='/')
        homedir++;
snprintf(maildir,BUFSIZE,"%s/Maildir",homedir);
if (chdir(maildir)<0)
{
        fprintf(stdout,"chdir failed: %s\n",maildir);
        exit(TEMP);
}

time(&thetime);
pid=getpid();
if (gethostname(host,BUFSIZE)<0)
{
        fprintf(stdout,"gethostname failed\n");
        exit(TEMP);
}

for(i=0;i<ATTEMPTS;i++)
{
        snprintf(mail,BUFSIZE,"tmp/%u.%d.%s",thetime,pid,host);
        errno=0;
        stat(mail,&statp);
        if (errno==ENOENT)
                break;

        sleep(2);
        time(&thetime);
}
if (i>=ATTEMPTS)
{
        fprintf(stdout,"could not create %s\n",mail);
        exit(TEMP);
}

if (!(fp=fopen(mail,"w+")))
{
        fprintf(stdout,"fopen failed: %s\n",mail);
        retval=TEMP; goto unlinkit;
}

fd=fileno(fp);

if (fprintf(fp,"%s",rpline)<0)
{
        fprintf(stdout,"fprintf failed\n");
        retval=TEMP; goto unlinkit;
}
```

```
if (fprintf(fp,"%s",dtline)<0)
{
        fprintf(stdout,"fprintf failed\n");
        retval=TEMP; goto unlinkit;
}

while(fgets(buffer,BUFSIZE,stdin))
{
        for(p=buffer;*p=='>';p++)
                ;

        if (!strncmp(p,"From ",5))
        {
                if (fputc('>',fp)<0)
                {
                        fprintf(stdout,"fputc failed\n");
                        retval=TEMP; goto unlinkit;
                }
        }

        if (fprintf(fp,"%s",buffer)<0)
        {
                fprintf(stdout,"fprintf failed\n");
                retval=TEMP; goto unlinkit;
        }
}

p=buffer+strlen(buffer);
nl=2;
if (*p=='\n')
        nl=1;

for(n=0;n<nl;n++)
{
        if (fputc('\n',fp)<0)
        {
                fprintf(stdout,"fputc failed\n");
                retval=TEMP; goto unlinkit;
        }
}

if (fsync(fd)<0)
{
        fprintf(stdout,"fsync failed\n");
        retval=TEMP; goto unlinkit;
}

if (fclose(fp)==EOF)
{
        fprintf(stdout,"fclose failed\n");
```

```
                retval=TEMP; goto unlinkit;
        }

        snprintf(newmaildir,BUFSIZE,"new/%u.%d.%s",thetime,pid,host);
        if (link(mail,newmaildir)<0)
        {
                fprintf(stdout,"link failed: %s %s\n",mail,newmaildir);
                retval=TEMP; goto unlinkit;
        }

unlinkit:
        if (unlink(mail)<0)
        {
                fprintf(stdout,"unlink failed: %s\n",mail);
                retval=TEMP;
        }

        exit(retval);
}
```

9.5 Remerciements

Merci à *Vicente Gonzalez (vince@nycrc.net)*

pour son aide qui a rendu possible la solution pour Qmail. Vous pouvez certainement écrire a Vince, pour le remercier, ainsi que lui poser vos questions et commentaires a propos de Qmail, le reste concernant ce HOWTO devant m'être adressé.

10 Samba Virtuel

10.1 Mise en place

L'installation de Samba Virtuel est très simple. Soyez sûr que les fichiers suivants soit installés correctement :

- /virtual/domain1.com/etc/smb.conf FICHIER
- /virtual/domain1.com/var/lock/samba REPERTOIRE
- /virtual/domain1.com/var/log/ REPERTOIRE
- /usr/local/bin/virtsmbstatus SYMLINK /usr/local/bin/virtexec

10.2 Inetd

Éditez /etc/inetd.conf

```
vi /etc/inetd.conf
```

Ajoutez cette ligne :

```
netbios-ssn stream tcp nowait root /usr/local/bin/virtuald \
        virtuald /virtual/conf.smbd smbd
```

10.3 Smb.init

Un fichier smb.init n'est pas nécessaire tant que le serveur est lancé via inetd.

11 Le reste

Tout autre service devrait suivre une procédure similaire :

- Lancer virtfs pour ajouter le binaire et les librairies au système de fichiers virtuel ;
- L'ajouter dans `/etc/inetd.conf` ;
- Créer un fichier `/virtual/conf.service` ;
- Créer tout script virtuel manquant.

12 Conclusion

Voici tout ce dont vous avez besoin. J'espère que cet article répond à vos attentes. Vous pouvez utiliser l'email à *Computer Resource Center* pour tout commentaire (NdT : en anglais bien sûr). Si vous avez une question, ou si vous me proposez une mise à jour, faites-le moi savoir et je l'ajouterai.

Ce document a rencontré un grand intêret. Je remercie toutes les personnes qui m'ont envoyé des questions et qui m'ont aidé à former ce document pour obtenir l'intérêt des utilisateurs. Avant de me poser des questions, je vous recommande de lire la FAQ pour voir si cela n'a pas déjà été demandé. Merci encore. *Brian*

13 FAQ

Q1. J'ai créé un sendmail.init et syslogd.init. Je les ai mis dans /usr/local/bin et essayé de les lancer, mais ils me donnent des erreurs.

R1. Ces fichiers sont appellés scripts d'initialisation. Ils sont lancés par le programme init quand votre ordinateur démarre. Ils ne vont pas avec les binaires de /usr/local. Consultez le Guide de l'Administrateur Système Linux (Linux System Administrators Guide) ou le Guide pour Bien Démarrer avec Linux (Linux Getting Started Guide) pour des informations sur la manière d'utiliser les scripts d'initialisation du système.

Q2. J'ai mis ces lignes dans /etc/sendmail.cf :

```
divert(0)
VERSIONID(‘tcpproto.mc’)
OSTYPE(linux)
FEATURE(redirect)
FEATURE(always_add_domain)
FEATURE(use_cw_file)
FEATURE(local_procmail)
MAILER(local)
MAILER(smtp)
```

Et j'obtiens une sortie vraiment étrange. Pourquoi ?

R2. Vous ne devez pas mettre ces lignes directement dans /etc/sendmail.cf. Le fichier sendmail.cf a été écrit pour que sendmail le comprenne facilement et est difficile à lire aux humains. Ainsi, pour le rendre facile à configurer, nous utilisons un programme appellé `m4` et ses capacités de macros pour créer le fichier `sendmailf.cf` . Les lignes FEATURE sont en fait des macros qui se développent par rapport à la configuration de Sendmail. Lisez la documentation de sendmail pour savoir comment configurer sendmail avec cette méthode. Aussi, notez que vous créez un fichier /etc/sendmail.cf principal et le script virtfs le copie sur /virtual/domain1.com/etc/sendmail.cf. Puis, vous éditez ce sendmail.cf pour l'adapter à votre domaine.

Q3. Ou puis-je trouver virtuald, qu'est-ce donc et comment l'utiliser ?

R3. Virtuald est un programme en C que j'ai écrit pour lancer un service virtuel. Il est inclus dans ce HOWTO. Vous le compilez comme un programme C normal : `make virtuald` . Le binaire résultant est placé dans /usr/local/bin. Ajoutez les lignes nécessaires à /etc/inetd.conf pour utiliser virtuald comme une facade vers un programme serveur.

Q4. Dialog n'est pas installé sur mon système ?

R4. Dialog est un programme qui permet d'avoir des fenêtres dans vos scripts shell. Il est nécéssaire pour faire fonctionner mon script shell virtuel. Vous pouvez trouver une copie de dialog sur *metalab* . Il ne devrait pas y avoir de problèmes pour le compiler et l'installer.

Q5. Comment puis-je savoir si le syslogd virtuel marche ?

R5. Quand virtuald est lancé, il doit envoyer le message suivant à syslogd (/var/log/messages) :

```
Nov 19 17:21:07 virtual virtuald[10223]: Virtuald Starting: $Revision: 1.1.1.1 $
Nov 19 17:21:07 virtual virtuald[10223]: Incoming ip: 204.249.11.136
Nov 19 17:21:07 virtual virtuald[10223]: Chroot dir: /virtual/domain1.com
```

Le message du `chroot` est envoyé par virtuald une fois l'appel système `chroot` effectué. Si ce message apparaît, alors le syslogd virtuel fonctionne. Si le service que vous rendez virtuel logue les messages par syslogd et que vous les voyez, c'est aussi un signe que le syslod virtuel fonctionne correctement.

Noter que si vous n'avez pas mis l'option VERBOSELOG lors de la compilation, Virtuald ne loguera pas du tout. Le seul moyen de savoir si le syslogd virtuel marche dans ce cas là, c'est de voir si un démon qui rend un service virtuel indépendamment, logue quelque chose avec syslogd.

Q6. Comment puis-je installer des quotas à travers un système de fichiers virtuel ?

R6. Vous l'installez comme vous le feriez d'habitude. Aller voir le *Quota mini-HOWTO* . Cependant, vous devez être sûr qu'il n'y ait pas de conflits d'uid entre les domaines. S'il y'a des conflits, les utilisateurs devront partager un quota. Préparez un intervalle d'uid qui auront le quota activé et dites aux domaines qu'ils ne peuvent avoir d'utilisateurs dans cet intervalle, à part ceux qui sont retenus pour avoir un quota.

Q7. Que fait cette notation "\" dans toutes les entrées du inetd.conf ?

R7. C'est juste une méthode pour couper une ligne d'un fichier de configuration en deux lignes. J'ai fait ça pour que les lignes puissent avoir un retour à la ligne de manière à obtenir une meilleure présentation. Vous pouvez ingorer le "\" et les rejoindre en une seule.

Q8. Quand je lance `passwd` ou n'importe quel programme concernant les logins, le système me renvoie `permission denied` . Quand je lance FTP ou `su` le système me renvoié `no modules loaded for service XXX` . Pourquoi ?

R8. Ce sont les messages d'erreur de PAM. J'ai écrit les scripts avant que PAM ne sorte. Mon script virtfs ne copie pas `/etc/pam.d` , `/usr/lib/cracklib_dict.*` , `/lib/security` ou n'importe quel fichier dont PAM a besoin. PAM en a besoin pour fonctionner. Si vous éditez mon script virtfs pour copier ces fichiers, il n'y aura plus de problèmes.

Q9. Est-ce que virtuald peut fonctionner avec les hosts.allow et hosts.deny de tcpd ?

R9. Oui, il peut, mais avec quelques modifications.

D'abord, le code source doit être changé en deux endroits.

Il faut insérer ces lignes là où les arguments sont analysés.

```
        if (!argv[3])
        {
                syslog(LOG_ERR,"invalid arguments: no program to run");
                exit(0);
        }
```

La ligne d'exécution doit remplacer :

```
        if (execvp(argv[2],argv+2)<0)
```

par :

```
        if (execvp(argv[2],argv+3)<0)
```

Deuxièmement, les lignes du fichier `inetd.conf` :

```
ftp stream tcp nowait root /usr/local/bin/virtuald \
        virtuald /virtual/conf.ftp tcpd wu.ftpd -l -a
```

Troisièmement, éditer les fichier `/virtual/domain1.com/etc/hosts.allow` et `/virtual/domain1.com/etc/hosts.deny` pour mettre vos paramètres.

Q10. Est-ce que mon serveur virtuel peut lancer des CGI ?

R10. Bien sûr, mais je vous recommande de mettre le répertoire `/cgi-bin` à un endroit en dehors du `chroot` , où vous seul avez accès. Par exemple, /var/www/cgi-bin/domain1.com. Donner l'accès aux cients à /cgi-bin leur donne la possibilité de lancer des programmes sur votre serveur. C'est un gros trou de sécurité. Soyez prudent. Je ne laisse aucun CGI tourner sur mon système sans que je n'ai pas personnellement cherché d'éventuels bugs.

Q11. Mes fichiers de configuration sont différents de vos exemples. Que dois je faire ?

R11 Il y a deux styles de configuration : System V et BSD. Les exemples fournis dans ce HOWTO sont basés sur les fichiers de configuration System V. Les services virtuels marchent aussi bien sur l'autre système. Pour des informations sur la méthode pour configurer vos fichier de style BSD, consultez l'origine de votre distribution ou le site LDP le plus près.

Q12. Je vous ai envoyé un mail et n'ai pas reçu de réponses ou alors elles ont pris un long moment avant de me parvenir. Pourquoi ?

R12. Vous n'avez sûrement pas mis VIRTSERVICES HOWTO dans le sujet. Sachez que je suis un administrateur réseau, et que parmi mes 20 heures par jour, j'administre mes machines virtuelles et celles de mes clients. Un mail qui est proprement adressé aura toujours une réponse dans les deux ou trois jours suivants. Les mails mal adressés ne seront pas filtrés vers ma boîte aux lettres pour VIRTSERVICES, et peuvent m'être inconnus pendant plusieurs jours, voir semaines.

Q13. Est-ce que virtuald marche avec une connection a 100Mbit ?

R13. La vitesse d'une carte réseau est totalement indépendante du fait que virtuald fonctionne ou pas. Vérifiez que votre serveur tourne sous 10Mbit et que votre carte 100Mbit fonctionne normalement sans un serveur virtuel.

Q14. Est-ce que je dois utiliser la `table virthost` de sendmail ?

R14. Non, c'est une fonctionnalité de Sendmail qui reçoit les informations pour plusieurs domaines. Virtuald donne à chaque Sendmail son propre environnement `chroot` . Installez Virtuald et configurez sendmail comme vous le feriez à l'habitude pour chaque domaine.

Q15. Puis-je installer un telnet virtuel sur ma machine ? Et est-il possible de créer un compte root virtuel, pour que les client puissent administrer leur propre domaine ?

R15. Ces questions reviennent souvent, et pour etre honnête, j'en ai un peu marre de les entendre. La réponse, qui est dans ce document, est que n'importe quel service lancé par inetd peut être rendu virtuel, donc rien ne vous empêche de le faire. Rien, à part le bon sens. Cependant, les bénéfices que vous pouvez avoir sont fortement altérés par le prix de la securité de votre machine virtuelle (ainsi que les sites que vous êtes supposés héberger d'une manière résponsable). Voici quelques exemples :

- Afin de duper une session telnet entrante vous devez modifier le noyau, pour avoir plusieurs proccessus, réinitialiser votre adresse IP source pour les connections sortantes, duper `gethostname` pour qu'il utilise le nom de domaine virtuel et non celui du système, etc. Si vous êtes un utilisateur avancé, hackez le kernel. Pour un débutant, je ne le recommande pas.
- En autorisant les utilisateurs à venir sur votre machine en telnet, vous les autorisez à lancer des programmes dangereux. Et ceux qui savent hacker peuvent prendre les privilèges root et causer des dommages sur votre système.
- Donner un accès root en telnet sur une machine virtuelle est très mauvais. Un utilisateur root virtuel peut quand même lire les fichiers des péripheriques , ce qui annule le `chroot` , peut éteindre le système et tuer les autres processus sur le système.
- Telnet est un service réseau non sécurisé. Des mots de passes sont envoyés en clair par le réseau. Si un utilisateur avec de mauvaises intentions récupère ce mot de passe, il ou elle peut utiliser les attaques mentionnées ci-dessus pour nuire à votre système.
- Votre environnement virtuel devra être plus gros. Vous aurez besoin de plus de librairies, plus de fichiers de configuration, et plus d'exécutables. Un disque de 6Go peut être très vite rempli.

Comme quoi, c'est une très mauvaise idée d'autoriser des connections sur une machine virtuelle. Si vous le permettez, tous les sites hebergés sur cette machine seront en danger. Si vous voulez autoriser un propriétaire de site à administrer ses utilisateurs, vous devez alors écrire (pas de script) le programme nécessaire pour lancer un processus virtuel qui l'autorise à les ajouter, effacer ou modifier en se connectant par ssh. Ceci devra être complètement exécuté par menus, vous ne devrez jamais autoriser de consoles, ou d'accès root. Afin d'accomplir ceci, vous devrez changer le propriétaire des fichiers concernés de root à un autre utilisateur. Si c'est fait de cette maniere, c'est assez securisé pour être incorporé dans une machine virtuelle. Il ne sera jamais acceptable d'autoriser des connections root en telnet ou ssh. Le faire, serait simplement une invitation au désastre. S'il y avait une raison de le faire, le site devrait être hébergé sur une machine dédiée, où le risque serait juste pour lui. Aucun administrateur responsable ne ferait autrement et donc je ne perdrai pas plus de temps sur cette question.

Q16. Y a-t-il un rpm, tar, site web, liste de diffusion, etc. associé à virtuald et au Virtual-Services HOWTO ?

R16. Pour le moment il n'y a rien de tout ceci. Ce HOWTO est la seule source d'information sur tout ce que j'ai fait concernant ce projet. Je trouve ce HOWTO assez informatif, rendant le besoin d'autres renseignements superflu.

Q17. Quand j'essaye de lancer virtexec en tant que simple utilisateur, j'ai `chroot: operation not permitted` . Pourquoi ?

R17. `chroot` est un appl système restreint au root. Seulement le super utilisateur peut l'executre. Le script virtexec lance le programme `chroot` ce qui implique le besoin d'être root pour le lancer.

Q18. J'ai mis en place pop et sendmail, mais la récuperation des mails ne semble pas marcher. D'où cela vient-il ?

R18. Certains programmes pop prennent comme emplacement des fichiers mail `/usr/spool/mail` . Je sais que `qpop` doit etre édité manuellement pour résoudre ce problème. Soit vous recompilez les sources de votre programme, soit vous faites un lien symbolique de `/virtual/domain1.com/usr/spool` vers `/virtual/domain1.com/var/spool` .

Q19. Je n'ai pas utilisé le programme mentionné dans votre HOWTO, j'utilise le programme XXX. Il ne marche pas. Pourquoi ?

R19. J'ai essayé de faire des exemples le plus génerique possible pour chaque serveur. Je sais que certaines personnes ont leur version favorite de chaque serveur. Envoyez-moi le plus d'informations possible, et j'essaierai de trouver une solution à votre problème et je l'incluerai dans la FAQ. L'information la plus importante est de me dire ou trouver la version du programme que vous utilisez (sous la forme ftp://ftp.domain.com/subdir/subdir/file.tgz).

Q20. Quand je lance virtexec il dit `symlink not a virt function` . Qu'est-ce que cela veut dire et comment le réparer ?

R20. Virtexec est programme pour lequel les arguments sont les quatres premiers caractères, et il lance le nom restant dans l'environnement virtuel. Par exemple, `virtpasswd` lance `passwd` . Si les quatres premiers caracteres ne sont pas `virt` , il se plaint et sort un message d'erreur. Virtexec est écrit en script shell et devrait être très simple à porter. Référez vous aux pages de manuels de bash ou du shell que vous utilisez pour vos question sur la programmation de script shell.

Q21. J'ai une question à propos de Qmail, Samba, Apache, etc. qui n'a aucun rapport avec la mise en place de virtuald ou l'interface entre le paquetage et virtuald.

R21. Tous les paquetages décris ici sont pleinement documentés. Certains ont un site web comme `www.nom_du_paquetage.org` qui leur est entièrement dédié. S'il vous plait consultez ces documents à propos de ce genre de questions.

Q22. J'ai plusieurs alias de domaines pointant sur domain1.com mais les mails continuent à être renvoyés aux alias. D'où est-ce que ca vient ?

A22. Virtmaildelivery compte sur les variables d'environnement qui lui sont passées pour déterminer quel répertoire /virtual/domain1.com utiliser pour distribuer le courrier. Il ne fait pas de recherche DNS pour déterminer l'adresse du mail. Puis, si l'adresse est `submail.mail.domain1.com` , virtmaildelivery essayera en premier cette adresse puis `mail.domain1.com` et puis `domain1.com` . Il essaye dans cet ordre, jusqu'à ce qu'une concordance ait lieu où qu'il ne reste plus de noms de domaines.

De toutes facons, si vous avez des alias de domaines qui ne sont pas des sous-domaines d'un autre, vous devez créer des liens symboliques comme :

```
cd /virtual
ln -s domain1.com domain1alias.com
```

De cette manière, virtmaildelivery sera trompé en pensant que ces mêmes répertoires existent même si l'un d'eux est un lien symbolique et le mail pourra être distribué à `user@domain1.com` ou `user@domain1alias.com` . Notez que `virtexec` listera les deux répertoires des domaines dans la boîte de dialogue quand vous le lancerez. Vous pouvez choisir n'importe lequel, mais ce sera le même système de fichier.