Service de Messagerie
ISRI 2021

Présentation de Postfix

1- L'infrastructure Postfix

1.1 Présentation :

Cette présentation ne reprend pas les concepts des protocoles de messagerie.(voir le cours et le TD pour cela).

Vous pouvez consulter la traduction de la documentation postfix sur :

http://postfix.traduc.org/

Le processus principal est le programme « master » qui est lancé par le script de démarrage. Il est résident et se charge d'appeler les différents modules en fonction des traitements et des besoins :

- Réception des messages,
- Analyse des messages,
- Gestion des files d'attente,
- Routage des courriers,
- Distribution des courriers,
- …

Il existe 2 fichiers principaux pour la configuration de Postfix qui sont « ***main.cf*** » et « ***master.cf*** ». L'ensemble des processus composant postfix sont définis dans master.cf. Le fichier main.cf permet de configurer l'ensemble de l'infrastructure à travers les variables de configuration.

Main.cf :

Le fichier de configuration typiquement Linux. Toute la configuration générale du serveur de messagerie. Les entrées sont du type VarX= Valeur ou VarY=$VarX ou multi-valeurs var= val1,val2,val3. Certaines variables peuvent aussi faire référence au contenu d'un fichier. La variable permet alors de définir le type et le nom de fichier de définition associé. Il faut parfois (re)construire le fichier au format indiqué par la commande « postmap » (fichiers indexés, hashés...).

Exemple :

« Transport_maps = hash:/etc/postfix/Nom_Fic » permet de définir un fichier contenant les valeurs des transports dans un format « hash ». La commande Postmap /etc/postfix/Nom_Fic va construire le fichier au format hash.

Les variables de ce fichier sont utilisées par les différents processus composants postfix définit dans master.cf. Néanmoins, pour différentes instances de programmes de même nature (smtpd généralement) on peut faire varier la configuration en redéfinissant la valeur au niveau de la définition du processus par surcharge des valeurs de main.cf (dans master.cf).

Les exercices de la suite du cours vous permettront de mieux saisir cette particularité de la configuration.

Chacune des lignes du fichier master.cf est composée de colonnes permettant de préciser le comportement du processus (instance(s) de MTA, client(s) smtp, agent(s) de distribution…).

Exemple:

clientsmtp unix - - n - 16 smtp

127.0.0.1:2626 inet n - n - 16 smtpd
-o content_filter=spamassassin
-o smtp_send_xforward_command=yes

Dans main.cf on peut trouver des valaurs de variables qui font référence à ces processus:

ex: content_filter=clientsmtp:127.0.0.1:2626

Les différents « processus de traitement » (fils du processus master):

Master est le processus père de tous les autres. Il supervise l'ensemble du système.

- pickup : lit les messages dans la file d'attente locale (file maildrop)
- cleanup : vérifie la cohérence des messages locaux et réseaux
- trivial-rewrite : Traitement des messages (vérifie, modifie, ajoute les entêtes manquantes, convertie les adresses, choisit le bon transport…)
- bouce deferred : Créer un rapport de non-délivrance (notification)
- qmgr est le gestionnaire des files d'attentes

Ces binaire se trouvent ici :
$ ls **/usr/lib/postfix/sbin/**
anvil cleanup dnsblog flush lmtp master oqmgr pipe postfix-tls-script post-install postmulti-script proxymap qmqpd showq smtpd tlsmgr trivial-rewrite virtual bounce discard error fsstone local nqmgr pickup postfix-script postfix-wrapper postlogd postscreen qmgr scache smtp spawn tlsproxy verify

Les agents d'entrés et de distributions principaux :

Entrée :

- smtpd : un serveur smtp (réception des messages du réseau)
- postdrop : processus qui réceptionne les messages locaux (comptes systèmes…)

Distribution :

- smtp, lmtp : se comporte comme un client de messagerie smtp
- - pipe : processus qui attend des infos en entrée pour le faire passer à un script, driver…
- - virtual : se charge des messages pour les boites dîtes « virtual » (compte non système)

Les files d'attente :

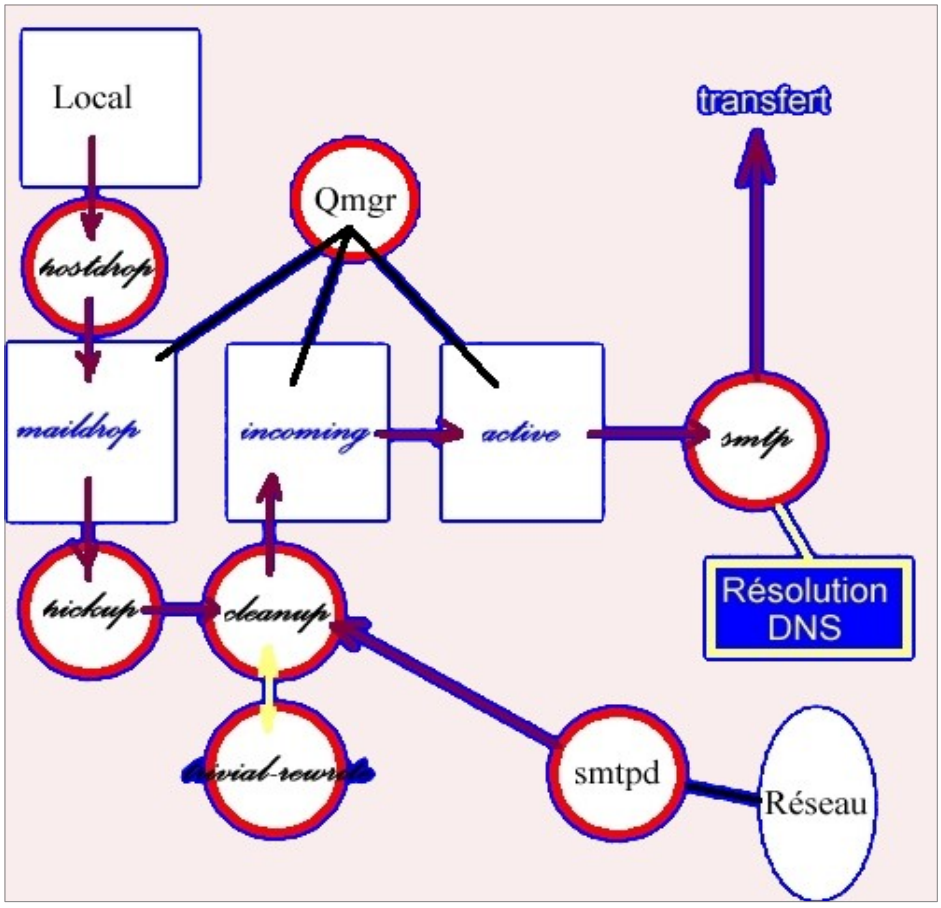
- maildrop : les messages locaux tombent dans cette file d'attente (provenance postdrop)
- incoming : les messages conformes (après analyse par cleanup et trivial-rewrite) transitent dans cette file. Ils sont passés à la file active.
- active : file des messages prêt à être traité par un transport (agent de distribution).
- deffered : file des messages ne pouvant être distribué
- corrupt : file des messages corrompus
- hold : file des messages conservés indéfiniment.

Exemple de la liste des processus (utilisation de la commande linux pstree)

```
├─master(2139)─┬─cleanup(6663)
│              ├─local(6665)
│              ├─pickup(6302)
│              ├─qmgr(4455)
│              ├─trivial-rewrite(6664)
│              └─smtpd(6735)
```

Bien entendu, l'affichage peut varié en fonction du moment où nous tapons la commande. En effet, certains processus sont exécutés à intervalles réguliers ou à l'arrivée d'événements particuliers.

Synoptique simplifié (cas local et réseau):



La notion de domaine :

Domaine Canonique : domaine local, domaine du serveur
Domaine relayé : les domaines a relayer (on ne stocke pas dans une boite, on ne sert que d'intermédiaire (MX de faible valeur pour un domaine ami). Les messages sont en file d'attente.
Domaine virtuel : domaine hébergé mais pas canonique. On distingue les domaines d'alias virtuel (re-routage vers comptes unix) et les boites aux lettres virtuelle (boites ne correspondant pas à un compte système : les messages sont stockés dans des fichiers ou répertoires) ou utilisent un transport particulier pour router les messages (lmtp, pipe...).

Les alias :
Les alias permettent de rediriger une adresse vers une autre, un domaine par un autre. Les alias sont configurés à travers une table des alias. La variable « virtual_alias_maps » est utilisée pour indiquer le format et le nom du fichier de stockage des alias.

Exemple :
Virtual_alias_maps = hash:/etc/postfix/virtual.alias

Dans le fichier alias on a:

Adresse1@doamine adresse@domaine...

Les fichiers « listes » après une modification se reconstruisent via la commande « postmap Nom_Fichier ». La commande produit un ficher au format indiqué et utilisé par le système.
La commande « postmap /etc/postfix/virtual.alias » va produire un fichier .db au format indiqué

Routage par les transports :
Il existe une table de transport pour diriger les messages en fonction de critères définis. La variable transport_maps permet d'indiquer le format et le fichier liste qui permet ce routage.
« Transport_maps = hash:/etc/postfix/transport »

Dans le fichier transport on a :
adresse@domaine Nom_du_transport

Nom_du_transport correspondant à une entrée du fichier master.cf (un pipe, un script...)

Attention! le fichier des alias est consulté avant le fichier des transports.

Dans tous les cas, on passe d'abord par les alias (virtual_alias_maps) donc si on veut qu'une adresse particulière d'un domaine virtuelle par exemple soit redirigée vers une adresse locale on peut le faire en plaçant une entrée dans cette table. On peut également créer un compte fourre tout en plaçant le nom de domaine virtuel sur boite unique (ex: @etab.ac-amiens.fr service). Par contre, si on place cette entrée dans le fichier des alias plus rien n'arrive au domaine virtuel car les alias sont prioritaires sur les domaines virtuels (voir la suite de la présentation).

Réécrire des émetteurs et destinataires d'un mail :

Cette fonction utilise les canonical maps. Celles-ci peuvent être appliqués en émission, en réception ou les deux.

Voyons comment opérer au moyen de 3 exemples simples.

1- Changement d'émetteur :

Admettons que vous avez un utilisateur root@monserveur.lan, nous allons réécrire l'adresse avec une canonical map afin de la faire apparaître en tant que root-serveur@example.org

Ajoutez la ligne suivante dans votre fichier /etc/postfix/main.cf

sender_canonical_maps = hash :/etc/postfix/sender_canonical

Insérez la ligne suivante dans le fichier /etc/postfix/sender_canonical.

root@monserveur.lan root-serveur@example.org

Enfin tapez :

postmap /etc/postfix/sender_canonical

service postfix reload

2- Changement de destinataire

Vous souhaitez que les messages adressés à hostmaster@dns1.example.org soient adressés à dns-master@example.org. Nous allons créer une canonical map pour cela.

Ajoutez la ligne suivante dans /etc/postfix/main.cf

recipient_canonical_maps = hash :/etc/postfix/recipient_canonical

Insérez la ligne suivante dans le fichier /etc/postfix/recipient_canonical :

hostmaster@dns1.example.org dns-master@example.org

Enfin tapez :

postmap /etc/postfix/recipient_canonical

service postfix reload

Changer un nom de domaine

3-Réécrire de toutes les adresses d'un nom de domaine donné et pour cela nous allons créer une canonical map générique (envoi + réception) afin de remplacer domain.lan par example.org.

Ajoutez la ligne suivante dans /etc/postfix/main.cf

canonical_maps = hash :/etc/postfix/canonical

Insérez la ligne suivante dans le fichier /etc/postfix/canonical :

@domain.lan @example.org

Enfin tapez

postmap /etc/postfix/canonical

service postfix reload

Installation du serveur SMTP : Postfix
Par les paquets **binaires ou sources**.

Si par les Sources :

Assurez-vous d'avoir une machine parfaitement configurée: c'est important pour le fonctionnement de postfix ! (réseau, résolution DNS, Domaine, nom de la machine, fichier hosts…),
Vérifier la version correspondant à votre distribution,
Vérifier et Installer les dépendances,
Télécharger et installer cette version à partir des sources (téléchargement des sources par par les dépôts et installation par make, make install sans le .configure),
Le service va fonctionner via UID et GID postfix/postdrop, créer cet utilisateur et ce groupe (adduser, groupadd, addgroup).
Vérifier que postfix appartient au groupe postdrop,
Installer si ce n'est pas le cas le MDA procmail (dans le cadre du TP, c'est lui qui va stocker le message dans le fichier mail de l'utilisateur du système dans /var/spool/mail/user).

apt install build-essential
Valider les dépôts sources (pas fait par défaut sur Ubuntu)
vi /etc/apt/sources.list
apt update
apt source --download postfix
apt build-dep postfix

adduser postfix
groupadd postdrop
adduser postfix postdrop

mkdir /usr/local/postfix
cp postfix_3.4.13.orig.tar.gz /usr/local/postfix/
tar -zxvf postfix_3.4.13.orig.tar.gz
cd postfix-3.4.13/

make
make install

Il faut ensuite répondre aux questions de l'installation (chemin d'installation des composants)

Par le paquet deb c'est plus simple :

apt install postfix

Configuration du serveur SMTP : Postfix

Comme nous l'avons déjà vu en introduction de ce document, les fichiers principaux sont « main.cf » et « master.cf » dans le répertoire de configuration.

Exemple :
Mon_domaine = dom.com
Mon_origine = $Mon_domaine

Les principaux paramètres du fichier main.cf
Les chemins d'accès sont renseignés automatiquement après l'installation :
queue_directory : spécifie le répertoire des files d'attente
command_directory : spécifie le répertoire des commandes administratives
daemon_directory : spécifie le répertoire des démons de postfix
mail_spool_directory : spécifie le répertoire des boites mail
mail_owner : spécifie le compte utilisé par le démon et les queues
myhostname : le non du serveur pleinement qualifié
mydomain : le domaine du serveur
myorigin : utilisé pour renseigner la partie droite d'une adresse mail non renseignée ($myhostname ou $mydomain)
mydestination : permet d'indiquer pour qui on accepte les mails localement ($myorigin, localhost, une liste dans une base…)
alias_maps : indique l'ensemble des bases d'alias d'adresses (correspondances d'adresse)
alias_database : idem pour ce qui se rapporte au monde unix
mailbox_command : permet d'indiquer l'agent de délivrance à utiliser (voir le cours)
relayhost :Pour une délivrance des messages vers un autre MTA

La conf minimale :

En plus des variables pré-renseignées (les chemins de configurations) il faut paramétrer un minimums de variables:

Pour ce TP vous pouvez supprimer la config pour le chiffrement.

Il reste à configurer quelques variables :

mynetworks = le réseau autorisé à se connecter ex :172.30.177.0/24
 # Le nom du domaine
mydomain = le domaine reconnu ex : in.ac-amiens.fr
 # Le nom de la machine pleinement qualifié
myhostname = ex : linux.in.ac-amiens.fr
 # Pour des adresses non pleinement qualifiées ajouter myorigin à droite du @
myorigin = $mydomain
 # Pour identifier les domaines locaux (à delivrer sur cette machine)
mydestination = $myhostname, localhost, $mydomain
 # identifier le MDA pour délivrer un message local
mailbox_command = /usr/bin/procmail

L’agent de délivrance des messages sera /usr/bin/procmail car en effet, dans un premier temps, nous utilisons les comptes systèmes (linux) dans le cadre du TP.

Procmail est un MDA. Le fonctionnement de procmail est très simple : il est appelé avec un mail en entrée standard, et traite le mail en fonction des filtres contenus dans le fichier *.procmailrc* (si il existe) du répertoire home de l'utilisateur. Si ce fichier n’existe pas il le dépose simplement dans la boite locale de l’utilisateur /var/mail/compte. Vous pouvez également utiliser le MDA « virtual » pour gérer des comptes sans avoir à créer un compte sur le système (faire un « man virtual » pour obtenir plus d’infos).
Voir également l’excellent article sur :
http://www.linux-france.org/article/mail/procmail/procmail.html

alias_maps = hash:/chemin/aliases permet d'indiquer le fichier des alias. Il faut construire cette BDD (ici un simple format hash) via la commande postmap (même si le fichier est vide au début de l'installation) et vérifier la création du fichier .db sinon le service ne fonctionne pas correctement.

Pour une config mydestination=myhostname,localhost et mydomain, les comptes valides seront les adresses du type nom@NomMachine, nom@Localhost, nom@domaine . Les logs du démon sont stockés dans /var/log/maillog (Ils sont écrit par le démon syslog).
Le test de la configuration se fait via la commande « postfix check » pour lancer arrêter le service utiliser « start » et « stop » à la place de « check ». Pour avoir la configuration de toutes les variables même celles par défaut la commande est « postconf » et pour avoir les valeurs par défauts il faut utiliser l'option -d.

Travail à réaliser :

Installer votre serveur SMTP postfix et placer les directives minimales dans le fichier main.cf, créer 2 comptes systèmes, démarrer et connectez vous et sur votre service SMTP et tester le fonctionnement en envoyant un message d'un compte vers l'autre (comme vous l'avez fait en TD). Vérifiez au niveau système via la commande mail (en étant connecté via le compte destinataire du message) ou directement en affichant le contenu du fichier mails (/var/spool/mail/Compte_User) et vous vérifiez également dans les traces du fichier de log.

r*oot@test6isri:/etc/postfix#* ***telnet 127.0.0.1 25***
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
220 test6isri.local.isri.fr ESMTP Postfix
helo pv
250 test6isri.local.isri.fr
mail from: tech
250 2.1.0 Ok
rcpt to: tech
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>

test

.
250 2.0.0 Ok: queued as DA1591270E4
quit
221 2.0.0 Bye
Connection closed by foreign host.

Vérification de la réception du message directement dans le fichier local système :

root@test6isri:/etc/postfix# ***cat /var/spool/mail/tech***
From tech@local.isri.fr Fri Mar 26 10:59:14 2021
Return-Path: <tech@local.isri.fr>
X-Original-To: tech
Delivered-To: tech@local.isri.fr
Received: from pv (localhost [127.0.0.1])
by test6isri.local.isri.fr (Postfix) with SMTP id DA1591270E4
for <tech>; Fri, 26 Mar 2021 10:59:00 +0100 (CET)
Message-Id: <20210326095906.DA1591270E4@test6isri.local.isri.fr>
Date: Fri, 26 Mar 2021 10:59:00 +0100 (CET)
From: tech@local.isri.fr

test

Installation d'une solution de serveur pop/imap

1 - Serveur pop/imap de l'université de Washington:

le site http://www.washington.edu
(imap-2007f pour la version disponible)
https://packages.ubuntu.com/fr/source/xenial/uw-imap pour ubuntu
–> Nous installons ce serveur dans la suite de ce document ;

2 - Serveur courier-imap

Utilisation simple à travers les comptes systèmes Linux.

Attention dans ce cas il faut :

Dans main.cf de postfix avoir :

home_mailbox = Maildir/
A la place de
#mailbox_command = /usr/bin/procmail

On utilise plus le répertoire *var*spool/mail mais il faut créer dans chacun du home répertoire des utilisateurs systèmes ~/Maildir
(On peut utiliser le skel pour ça)

Dans le fichier de config de /etc/courier/impad

Si on ne travaille pas en mode chiffré alors diéser les
#IMAP_CAPABILITY_TLS..
Ajouter

IMAP_CAPABILITY="AUTH=PLAIN"
MAILDIRPATH=Maildir

Tester en utilisant un simple telnet sur le port 143 (voir TD)

3 - Serveur dovecot

dovecot, uw-imap, zimbra…
https://fr.qaz.wiki/wiki/Comparison_of_mail_servers

A Réaliser pour ce TP:

- Installer l'un de ces 3 serveurs !

Pour le cas de l'installation du serveur pop/imap de l'université de Washington:

Pour limiter le temps nécessaire à ce TP, nous n'utiliserons pas les versions chiffrées des protocoles (génération des clés, certificats...).

Les sources sont accessibles via le site http://www.washington.edu
(imap-2007f pour la version disponible)
ou par le paquet source de votre distribution (uw-imap pour ubuntu 16.04)

https://packages.ubuntu.com/fr/source/xenial/uw-imap pour ubuntu

wget https://www.mirrorservice.org/sites/ftp.cac.washington.edu/imap/imap-2007f.tar.gz
apt-get build-dep uw-imap
vi Makefile → modifier les variables
make slx

root@PV-Linux:/usr/src/imap-2007f# make slx

make[1]: entrant dans le répertoire « /usr/src/imap-2007f »

++

+ Building in NON-COMPLIANCE with RFC 3501 security requirements:

+ Non-compliant:

++ TLS/SSL encryption is NOT supported

++ Unencrypted plaintext passwords are permitted

+

+ In order to rectify this problem, you MUST build with:

++ SSLTYPE=nopwd

+ You must also have OpenSSL or equivalent installed.

++

Do you want to continue this build anyway? Type y or n please:

Y
…

`cat ../c-client/CCTYPE` -I../c-client `cat ../c-client/CFLAGS` -c -o tquota.o tquota.c

`cat ../c-client/CCTYPE` -I../c-client `cat ../c-client/CFLAGS` -o tmail tmail.o tquota.o ../c-client/c-client.a `cat ../c-client/LDFLAGS`

make[2]: quittant le répertoire « /usr/src/imap-2007f/tmail »

make[1]: quittant le répertoire « /usr/src/imap-2007f »

Les binaires ipopd et imapd sont bien compilés :

root@PV-Linux:/usr/src/imap-2007f# ls -l ipopd

total 6028

-rwxr-xr-x 1 root dip 2942668 déc. 19 09:57 ipop3d

...

root@PV-Linux:/usr/src/imap-2007f# ls -l imapd

total 3628

-rwxr-xr-x 1 root dip 3136709 déc. 19 09:57 imapd

...

On peut les copier dans le répertoire *usr*bin du système :

root@PV-Linux:/usr/src/imap-2007f# cp ./ipopd/ipop3d /usr/bin

root@PV-Linux:/usr/src/imap-2007f# cp ./imapd/imapd /usr/bin

Ce paquetage renferme un serveur pop(s) et un serveur imap(s) (uniquement la version imap sous certaines distributions). La configuration par défaut compile l'application avec la couche SSL. Nous simplifions l'installation, en supprimant la couche SSL, utilisation en ipv4 et sans chiffrement des mots de passe. il est donc nécessaire de modifier les sources (fichier Makefile) option SSLTYPE, IP, et PASSWDTYPE. Installer les dépendances. Ces services fonctionnent au dessous du super démon inetd (inetd, ou xinetd ou openbsd-inetd en fonction des distributions. Faire la compilation en fonction de votre système (linux : make slx), copier le démon dans le répertoire /usr/sbin (imapd et ipopd), vérifier le bit x, ajouter les lignes nécessaires dans /etc/inetd.conf ainsi que dans le fichier /etc/services si besoin. (PASSWDTYPE=std, SSLTYPE=none, IP=4)
N'oublier pas de relancer le service inetd (ou la variante de votre distribution).

Exemple de configuration :

Pour inetd:

```
pop3    stream    tcp    nowait    root    /usr/sbin/ipop3d    ipop3d
```

Pour xinetd :

```
service pop3
{
        socket_type      = stream
        wait             = no
        user             = root
        server           = /usr/bin/ipop3d
        log_on_success  += HOST DURATION
        log_on_failure  += HOST
        instances        = 150
        cps              = 70 30
        disable          = no
}
```

Relancer le service xinetd et tester la connexion :

```
root@PV-Linux:/etc/xinetd.d# telnet 10.1.16.50 110
Trying 10.1.16.50...
Connected to 10.1.16.50.
Escape character is '^]'.
+OK POP3 PV-Linux 2007f.104 server ready
QUIT
+OK Sayonara

Connection closed by foreign host.
```

Test de la configuration (voir TD):

Connectez vous sur le serveur pop et imap avec l'un des deux comptes et vérifiez la connexion. La connexion se fait en mode commande via un telnet sur le port 110 (143 pour imap).

- Vous installez également un client de messagerie (thunderbird, evolution…) pour tester l'envoi et la réception de vos messages.

Nous avons donc maintenant une architecture de messagerie opérationnelle avec un MTA serveur (SMTP) gérant le domaine de l'entreprise et les MAA serveurs (IMAP, POP) pour permettre aux usagers de retirer leurs messages. Il faut maintenant consolider notre infrastructure (configuration, anti-virus, anti-spam...).

Exercice :

- Dans un premier terminal vous exécutez ce script :

while [1]; do sleep 2 ; pstree 32109; done

- Dans un second terminal vous vérifier l'évolution des process postfix en envoyant un message via telnet.

(Il faut bien entendu utiliser votre propre numéro de process « master »)

```
master─┬─pickup
       ├─qmgr
       └─smtp
master─┬─pickup
       ├─qmgr
       └─smtp
...
master─┬─pickup
       ├─proxymap
       ├─qmgr
       ├─smtp
       └─smtpd
...
master─┬─pickup
       ├─proxymap
       ├─qmgr
       ├─smtp
       ├─smtpd
       └─trivial-rewrite

...
master─┬─cleanup
       ├─local
       ├─pickup
       ├─proxymap
       ├─qmgr
       ├─smtp
       ├─2*[smtpd]
       └─trivial-rewrite
```

Un peu plus loin avec Postfix...

Postfix permet de nombreux contrôles paramétrables , sur le contexte de la communication (réseaux, ip, résolution DNS…), au niveau des commandes : Enveloppe : HELO, MAIL FROM…, au niveau des variables d'en-têtes, du corps du message...

voir http://postfix.traduc.org et http://www.postfix.org/uce.html

C'est ce que nous allons voir maintenant...

Un premier exemple :

Interdire les fichiers dangereux…

Il est possible de sécuriser la boite aux lettres de vos utilisateurs en testant les pièces jointes transitant par le serveur.

Ex : refuser les fichiers portant une extension « dangereuse » exe com bat…

Ligne à rajouter dans /etc/postfix/main.cf :

header_checks = regexp:/etc/postfix/header_check

Cela permet d'indiquer de tester les entêtes via un fichier de règles construites via des expressions régulières. Dans le TD nous avons vu le contenu d'un message avec un fichier attaché, ce qui nous permet d'écrire cette règle :

Le contenu du fichier avec la règle :

```
/filename="?.*\.(bat|com|pif|vb|exe|lnk|scr|reg|chm|wsh|js|inf|shs|job|ini|
shb|scp|scf|wsc|sct|dll|msc|msi|ppt|pps|mdb|rar|bmp|wav|mpg|mpeg|wma|wmv)/
REJECT
/^Content-(Disposition|Type):.*(file)?name="?.*\.(bat|com|pif|vb|exe|lnk|
scr|reg|chm|wsh|js|inf| shs|job|ini|shb|scp|scf|wsc|sct|dll|msc|msi|ppt|
pps|mdb|rar|bmp|wav|wma|wmv)/ REJECT Extension de fichier joint refusée par
l administrateur ISRI Desole!
```

Tester via votre client graphique.

Autres exemples pour un début d'anti-spam :

Mise en place des contrôles d'accès au serveur de mail :

Exemples de paramètres de contrôles :

au niveau de la connexion SMTP (directive *smtpd_client_restrictions)*
au niveau de l'adresse donnée dans la phase HELO/EHLO (directive *smtpd_helo_restrictions)*
au niveau de l'adresse donnée dans la phase MAIL FROM: (directive *smtpd_sender_restrictions)*
au niveau du RCPT TP: (directives smtpd_recipient_restrictions= reject_unauth_destinations)

Exemple:

Par défaut Postfix accepte tous les mails de n'importe quel site émetteur (réception d'un message pour un utilisateur du domaine). Pour constituer une liste noire , on peut créer un fichier */etc/postfix/access* dont la syntaxe est la suivante (c'est un moyen rapide pour créer une règle d'anti-spam) :

Site1.com REJECT
Site2.com REJECT
Site3.dom..com OK

Comme expliqué précédemment, certains fichiers seront indexés via la commande « postmap » dbm:*/etc/postfix/access* ou hash:*/etc/postfix/access* selon les types d'indexation supportés (nous utiliserons le format hash).

Ce fichier de liste-noire peut être utilisé dans les variables "*smtpd_sender_restrictions"* et "*smtp_client_restrictions"* pour restreindre les adresses émettrices dont on accepte les mails

smtpd_sender_restrictions = check_sender_access hash:*/etc/postfix/access*

Exercices :

- Un envoi massif de spam de l'adresse contact-info@pub.fr arrive dans les boites de vos utilisateurs, vous réalisez un filtre qui permet de bloquer cette adresse. Vous testez votre filtre.

- Rechercher tous les paramètres postfix du type « reject_* » et tester ces paramètres dans la configuration de votre serveur.

Un exemple de lignes de config :

maps_rbl_domains = bl.spamcop.net
smtpd_client_restrictions = check_client_access hash:/etc/postfix/access, reject_rbl_client
smtpd_sender_restrictions = check_sender_access hash:/etc/postfix/access
mime_header_checks = regexp:/etc/postfix/header_check

Configuration du serveur SMTP en serveur de relayage (dit null client)

Un relais ne sert pas de comptes locaux il transmet uniquement le message vers un autre serveur SMTP après toutefois les avoir stockés ou même **avoir fait des traitements particuliers si nécessaire (spam, anti-virus…).**

Il y a principalement 2 variables pour relayer :

relayhost et content_filter

relayhost = $mydomain
relayhost = [gateway.example.com]
relayhost = [ip_add]

Pour tester cette configuration vous installez 2 serveurs SMTP sur une même machine en utilisant les ports 25 et 2525. Celui sur le port 25 relaye uniquement vers celui sur le port 2525.

Pour réaliser cet exercice vous utilisez la directive « content_filter ».

–> L'intérêt de cette exercice est de comprendre également l'utilisation de **master.cf**

Dans le fichier main.cf vous ajoutez :

content_filter = clismtp:localhost:2525 dans le fichier main.cf

Il faut bien entendu définir clismtp et faire tourner un autre serveur smtp sur le port 2525

Dans le fichier master.cf vous définissez :

```
2525      inet  n       -       y       -       -       smtpd
   -o content_filter= -o myhostname=smtp2.local.isri.fr
clismtp   unix  -       -       y       50      -       smtp
```

clismtp est donc un processus qui va permettre d'envoyer des messages vers un smtpd. smtpd est un processus qui est en écoute de message (dans cette config un sur le port 25 (smtp) et 2525 (écrit en dur). Nous avons redéfinit la variable hostname du second serveur.

Les lignes suivantes sont identiques :

```
smtp      inet  n       -       y       -       -       smtpd
avec
25        inet  n       -       y       -       -       smtpd
```

- Vous comprenez cette configuration
- Vous testez via un envoi par telnet
- vous suivez l'échange via mail.log

Dans la suite de ce TP nous réutiliserons ce mode de fonctionnement pour passer un message à plusieurs serveurs SMTP :

Exemple :

MESSAGE –> SMTP-RECEPTION –> SMTP-ANTI-VIRUS –> SMTP-SPAM –> Stockage

Pour forwarder uniquement un domaine particulier :

Il faut identifier un transport particulier pour le domaine à « forwarder » (ex : ac-amiens.fr forwarde vers smtp.ac-amiens.fr). La ligne « **ac-amiens.fr smtp:[smtp.ac-amiens.fr]** » est a placer dans le fichier transport.

* Il ne faut pas oublier de placer la ligne indiquant le fichier transport dans le fichier main.cf et de reconstruire la BDD via un postmap.

Identification du fichier transport : « transport_maps = hash:/etc/postfix/transport »

Contenu du fichier transport :
ac-amiens.fr smtp:[smtp.ac-amiens.fr]

Construction de la BDD :
postmap transport
relancer le serveur

Exercice :

Vous êtes le MX de secours d'une filiale qui possède son propre domaine et infrastructure de messagerie (beauvtt.fr) . Que faut il faire pour palier à un problème de connexion internet de cette filiale ? Elle se propose également de devenir secours pour votre domaine... que faut il faire ?

Exemple de config pour le FORWARD :

On souhaite forwarder le domaine « ac-amiens.fr » vers smtp.ac-amiens.fr

On va donc accepter les messages pour le domaine ac-amiens.fr (il faudra peut être une règle ACCEPT si vous avez des règles sur le « MAIL FROM : » (voir début du TP)

Dans le fichier nommé transport
ac-amiens.fr smtp:[smtp.ac-amiens.fr]
–> les crochets interdise la resolution MX
Dans main.cf:
transport_maps = hash:/etc/postfix/transport

Mutualiser plusieurs noms de domaine sur un même domaine:

On peut en avoir besoin pour mutualiser un domaine interne et un domaine externe par exemple

Définir le ou les domaines virtuels partagés:
virtual_alias_domains = alias.ac-amiens.fr (c'est un exemple)
Définir la table des alias pour les différents domaines
virtual_alias_maps = hash:/etc/postfix2/virtual_alias

Dans le fichier virtual_alias, on définit les équivalences :
info@domaineexterne.fr util@domaineinterne.fr

Ici, un message pour « info@domaineexterne.fr » tombe dans la boite de « util@domaineinterne.fr »

Pour héberger plusieurs domaines de messagerie dans l'arborescence du serveur :

Dans le cadre du TP, nous avons configurer le serveur pour utiliser les boites systèmes des utilisateurs. Avec cette option, nous allons utiliser la notion de domaine virtuel ce qui va permettre de stocker les messages dans un fichier classique même si l'utilisateur n'a pas de compte système. Il faudra ensuite mettre en place un serveur pop/imap capable d'exploiter ces fichiers de messages.

On utilise pour cela le paramètre « virtual_mailbox_domains »:
"virtual_mailbox_domains = dom1.fr, dom2.fr"
ou via un fichier si on utilise de nombreux domaines virtuels
« virtual_mailbox_domains = /etc/postfix/dom_virtuels »

On utilise ensuite :
virtual_mailbox_base pour identifier la racine du repertoire de stockage
Il reste a identifier la boite (le fichier) pour chacun des comptes via les directives :

virtual_mailbox_domains = etab.fr
virtual_mailbox_base = /var/spool/etab
virtual_mailbox_maps = hash:/etc/postfix/mailbox_etab
virtual_uid_maps = static:113
virtual_gid_maps = static:120
(user 113 postfix et group 120 postfix dans ce cas, c'est certainement différent pour vous !)

root@debian:/etc/postfix# cat mailbox_etab
prof@etab.fr prof
Créer le fichier .db

Envoi du message:

Après envoi d'un message vers prof@etab.fr:

root@debian:/etc/postfix# ls -l /var/spool/etab/
total 4
-rw------- 1 postfix postfix 466 janv. 9 18:03 prof

root@debian:/etc/postfix# cat /var/spool/etab/prof

From pascal@beauvelo.fr Thu Jan 9 18:03:25 2014
Return-Path: <pascal@beauvelo.fr>
X-Original-To: prof@etab.fr
Delivered-To: prof@etab.fr
Received: from pascal (testcal.in.ac-amiens.fr [172.30.177.86])
        by smtp.beauvelo.fr (Postfix) with SMTP id 0F015C5D95
        for <prof@etab.fr>; Thu, 9 Jan 2014 18:02:58 +0100 (CET)
subject: essai
Message-Id: <20140109170309.0F015C5D95@smtp.beauvelo.fr>
Date: Thu, 9 Jan 2014 18:02:58 +0100 (CET)
From: pascal@beauvelo.fr

test

root@debian:/etc/postfix#

Il faut créer le répertoire /var/spool/etab avec les bons droits (uid gid du compte postfix)

Petite précisions :

- encore une fois, attention, dans tous les cas on passe d'abord par les alias (virtual_alias_maps) donc si on veut qu'une adresse particulière du domaine virtuelle soit redirigée vers une adresse locale on peut le faire en plaçant une entrée dans la table des alias.

- Si vous créez un compte fourre tout en plaçant le nom de domaine sur une boite unique

ex: @etab.ac-amiens.fr   service

plus rien n'arrive au domaine virtuel car les alias sont prioritaires sur les domaines virtuels.

Pour envoyer un message vers un script pour un traitement particulier:

Nous allons ici configurer notre serveur pour qu'il route un message vers un script classique Linux. Nous illustrons cette option par un exemple à travers une utilisation via Nagios.

Nagios est application de supervision qui est capable d'envoyer une alerte par mail. Nous allons récupérer ce message d'alerte pour alimenter un fichier xml qui va servir d'entrée à un flux rss et qui va être affiché à travers un serveur web (c'est juste un exemple).

Un mail pour nagios@local.isri.fr va être associé au **transport « traitement »** qui renvoie sur un pipe qui va servir un script en entrée. Le message est placé en entrée de la commande fluxrss.sh.

Dans master.cf:

```
traitement  unix -      n       n       -       -       pipe  user=nobody argv=/usr/local/bin/fluxrss.sh
```

Identifier le fichier transport, insérer une ligne pour ce transport et régénérer le fichier transport

Exemple de fichier fluxrss.sh:

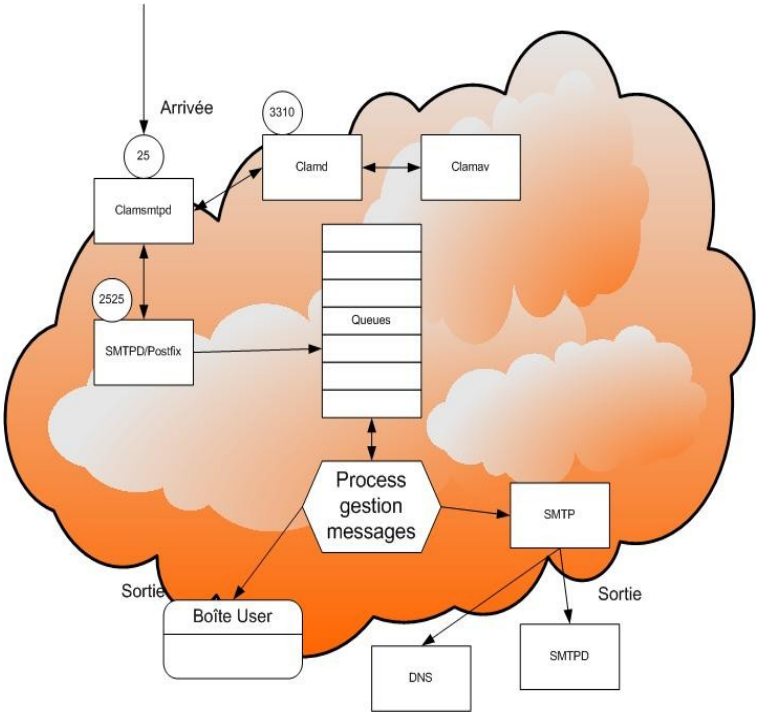
```
# on place le message reçu en entrée dans le fichier « /var/spool/mail/message_nagios »
tee > /var/spool/mail/message_nagios
# ici on va découper le message pour placer les infos dans des variables
Host=`cat /var/spool/mail/message_nagios |grep Host| sed 's/Host: //'`
Address=`cat /var/spool/mail/message_nagios |grep Address| sed 's/Address: //'`
State=`cat /var/spool/mail/message_nagios |grep State| sed 's/State: //'`
Service=`cat /var/spool/mail/message_nagios |grep Service| sed 's/Service: //'`
Nom=`cat /var/spool/mail/message_nagios |grep Nom| sed 's/Nom: //'`
Date=`cat /var/spool/mail/message_nagios |grep Time| sed 's/Date\/Time: //'`
#formatage du fichier rss. On le place dans le repertoire du serveur apache (par exemple)
sed -i '/\/channel/d' /usr/local/httpd/htdocs/fluxrss.xml
sed -i '/\/rss/d' /usr/local/httpd/htdocs/fluxrss.xml
echo " <item>" >> /usr/local/httpd/htdocs/fluxrss.xml
echo "  <title>$Address-$Service</title>" >> /usr/local/httpd/htdocs/fluxrss.xml
echo "    <link>http://nagios.agriates.ac-amiens.fr/nagios/cgi-bin/status.cgi?
host=$Nom#$Date</link>" >> /usr/local/httpd/htdocs/fluxrss.xml
echo "    <guid isPermaLink="False">$Service-$Date</guid>" >>
 ...
```

Exercice :

1- Sur le même principe pour générer une page html consultable via votre navigateur à travers votre serveur Web (Vous avez reçu un message de xxx le contenu est xxx).
2- rechercher, télécharger et installer le script « répondeur » (écrit en Perl) qui permet d'envoyer un mail de réponse à un courrier reçu sur un compte particulier (revoir la notion de transport). Dans ce cas le traitement va permettre de récupérer l'émetteur du message, de construire un message de réponse et de l'envoyer par une commande du type « mail ».

Installation d'un Antivirus sur le MTA (clamav)

Installation d'un anti-virus pour les messages :



Clamav est une application qui permet de scanner un ensemble de fichiers directement via la commande **clamscan**. **Freshclam** permet de télécharger les signatures. **Clamd** est un démon qui permet de passer un flux à scanner à clamav. Il faudra donc transmettre les messages à ce démon pour qu'il les analyse.

Installation via les paquets binaires :

apt-get install clamav
apt-get install clamav-daemon

Via les sources du projet:
wget http://downloads.sourceforge.net/clamav/clamav-X.tar.gz

tar -zxvf clamav-X.tar.gz (ou via un apt-get source)
cd clamav-X

Via les sources de la ditribution Ubuntu:
apt-get build-dep clamav
apt-get –download-only clamav
vi INSTALL
groupadd clamav
useradd -g clamav ... avec les options qui vont bien !
./configure --prefix=/usr/local/clamav –disable-zlib-vcheck --disable-unrar
make
make install
cd /usr/local/clamav/
cd etc
./clamconf
cp clamd.conf clamd.conf.ori
vi clamd.conf
Voir les signatures
ll /usr/share/share/clamav/
vi /usr/local/clamav/etc/freshclam.conf
../bin/clamconf
#Dieser Exemple
chown et chgrp clamav clamd.conf
lire la conf par clamconf (attention au chemin)
#Mise a jour des signatures
../bin/freshclam
chown -R clamav /usr/local/clamav
#Lancer un scan d'un repertoire
../bin/clamscan -r /home
créer un fichier avec la signature Eicar:
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!
$H+H*
#Lancer en mode demon
./sbin/clamd
touch /var/log/clamd.log
chown clamav /var/log/clamd.log
./sbin/clamd
netstat -lnt
vi clamd.log
touch /var/run/clamd.pid
chown clamav /var/run/clamd.pid

Vérifier le fonctionnement (ps, netstat...)

Le fichier clamd.conf pour une écoute sur TCP/IP:

LocalSocket /tmp/clamd.socket

LogFile /usr/local/clamav/var/log/clamd.log

LogFileUnlock yes

LogFileMaxSize 200M

LogTime yes

LogClean yes

LogVerbose yes

PidFile /var/run/clamd.pid

TemporaryDirectory /var/tmp

DatabaseDirectory /usr/local/clamav/share/clamav

TCPSocket 3310

TCPAddr 127.0.0.1

User clamav

ScanMail yes

Clamav tourne sur une socket Unix et sur une socket inet (réseau) sur TCP port 3310.

root@PV-UBUNTU-172:/home/pascal# telnet 127.0.0.1 3310

Trying 127.0.0.1...

Connected to 127.0.0.1.

Escape character is '^]'.

PING

PONG

Connection closed by foreign host.

Fonctionnement :

clamd reconnaît quelques commandes (voir man clamd), on peut lui passer un flux de données à analyser. Ce n'est pas un MTA, il ne peut donc pas dialoguer directement avec un serveur de messagerie. C'est pourquoi, nous allons utiliser un MTA spécialement créer pour dialoguer avec Clamav plutôt que d'utiliser postfix pour faire le lien avec le démon Clamav. Ce démon est clamsmtp, il permet de faire le lien avec clamd, puis de récupérer le message et de le transférer à une autre instance de MTA (postfix en écoute locale par exemple).Clamsmtp va recevoir les messages de l'extérieur sans aucun contrôle, il va les passer à clamd puis clamav.
Installation :
Comme d'habitude il existe différentes possibilités :

- apt-get install clamsmtp
- wget http://ftp.de.debian.org/debian/pool/main/c/clamsmtp/clamsmtp_1.10.orig.tar.gz
- apt-get –download-only clamsmtp (avec un apt-get build-dep clamsmtp)

Dans le cadre d'une installation via les sources:

```
 tar -zxvf clamsmtp-X.tar.gz
 cd clamsmtp-X
./configure --prefix=/usr/local/clamsmtp
make
make install
cd /usr/local/clamsmtp/
mkdir etc
cp /chemin/clamsmtp-1.8/doc/clamsmtpd.conf ./etc
cd etc
# Modifier le fichier de conf
vi clamsmtpd.conf
#création d'un fichier pid vide
 cat > /var/run/clamsmtp.pid
chown clamav.clamav /var/run/clamsmtp.pid
```

Le fichier de clamsmtpd.conf:

Attention clamsmtp tourne sous le compte précisé par la directive User. Il faut donc utiliser les ports hauts si le daemon tourne sous un autre compte que root. Il est préférable d'utiliser le même compte pour clamsmtpd et clamav car les processus vont s'échanger les "fichiers mails".

Pour ne pas compliquer le TP nous allons faire tourner clamsmtp sur le port 25 sous le compte root qui va faire passer à clamd qui va également tourner sur le compte root sous le port 3310 qui va repasser à clamsmtp qui va à son tour passer le mail à une instance de postfix configuré sur le port 2525 sur l'ip loopback (127.0.0.1). Nous n'avons plus besoin d'ouvrir postfix sur le réseau car c'est clamsmtpd qui va se charger de la réception des messages.

Tester la chaîne en envoyant un mail puis tester de nouveau en envoyant un fichier avec un Virus.

Le contenu de clamd.conf:

LocalSocket /tmp/clamd.socket

LogFile /usr/local/clamav/var/log/clamd.log

LogFileUnlock yes

LogFileMaxSize 200M

LogTime yes

LogClean yes

LogVerbose yes

PidFile /var/run/clamd.pid

TemporaryDirectory /var/tmp

DatabaseDirectory /usr/local/clamav/share/clamav

TCPSocket 3310

TCPAddr 127.0.0.1

#User clamav

User root

ScanMail yes

Remplacer la ligne pour écouter sur le port 2525 du master.cf de notre postfix d'origine:
#smtp inet n - - - - smtpd

2525 inet n - - - - smtpd

Faire écouter sur 127.0.0.1 par la ligne « inet_interfaces = 127.0.0.1
» dans main.cf

Même architecture, en utilisant une variante avec un filtre particulier via la directive « content_filter » :

Le but du TP est de configurer l'architecture pour utiliser une instance postfix pour la réception des messages sur le réseau, qui va passer à clamsmtp puis à une autre instance de postfix. C'est « Content_filter » qui permet d'indiquer un filtre qui peut être un script, un client smtp, un serveur smtp... qui va recevoir le message.

Utiliser 2 instances de smtpd (à définir dans master.cf): un smtpd sur le port 25 qui utilisera la directive content_filter pour passer le mail à clamsmtp (port 2525) qui passe à clamav (port clamd 3310) pour le repasser ensuite à la deuxième instance smtpd (2626). Clamsmtpd et clamav utiliseront cette fois le compte clamav. Vous expliquez le fonctionnement de cette configuration.

main.cf:

inet_interfaces = 10.1.24.220
On envoie à clamsmtp
content_filter = clientsmtp:127.0.0.1:2525

master.cf:
utilise par content_filter qui est un client smtp qui envoie le mail à clamsmtp
Pour cette instance on redéfinit les options nécessaires
clientsmtp unix - - n - 16 smtp
-o smtp_send_xforward_command=yes

L'option « smtp_send_xforward_command=yes”
doit permettre de ne pas modifier le nom passé par le client lors de la connexion

Retour du mail (conf clamav) apres antivirus attention a redéfinir enveloppe du mail
Pour cette instance on redéfinit les options nécessaires
127.0.0.1:2626 inet n - n - 16 smtpd
-o content_filter=

content_filter est un transport qui envoie le message vers 127.0.0.1 sur le port 2525 (clamsmtp). clamsmtp renvoie le message après scan sur smtpd en écoute sur 127.0.0.1:2626.

Installation de spamassassin:

apt-get install spamassassin
apt-get install spamc

spamc est une commande qui permet d'envoyer le flux en entrée à spamd.

Tester directement par la commande spamc < fichier_courrier_in

Configuration pour postfix (le fichier main.cf):
# l'instance principale écoute sur le port 25
# le content_filtre permet de transférer le message vers 127.0.0.1:2525 (smtpd clamsmtp)
content_filter = clientsmtp:127.0.0.1:2525
#transport_maps = hash:/etc/postfix/transport
#va permettre d'envoyer un message si déjà tester par spamassassin vers 127.0.0.1:2727
# voir le contenu du fichier filtre un peu plus bas
header_checks = pcre:/etc/postfix/**filtre**

Le fichier master.cf:
#on a besoin du smtpd en ecoute sur le port 25 qui va uniquement utiliser le content_filter du main.cf soit:
clientsmtp:127.0.0.1:2525 (clamsmtp)
25 inet n - - - - smtpd
#
utilise par content_filter: client smtp/lmtp qui envoie a clamsmtp le mail
clientsmtp unix - - n - 16 smtp
-o smtp_send_xforward_command=yes
Retour du mail apres antivirus attention a redéfinir l'enveloppe du mail si besoin : ici c'est clamsmtp qui renvoi sur 2626

127.0.0.1:2626 inet n - n - 16 smtpd
-o content_filter=spamassassin
-o smtp_send_xforward_command=yes
#filtre commande qui va passer le contenu du message au demon spamd pour test spam
#le courrier revient avec des entêtes supplémentaires de valeur de spam
l option -e de spamc permet de passer le retour à une commande ici
sendmail qui ré-injecte le mail dans la queue

spamassassin unix - n n - - pipe
user=clamav argv=/usr/bin/spamc
-f -e /usr/sbin/sendmail -oi -f ${sender} ${recipient}

Ce smtpd va recevoir les mail avec la marque de spamassassin

127.0.0.1:2727 inet n - n - 16 smtpd
-o content_filter=
-o receive_override_options=no_header_body_checks

-o smtp_send_xforward_command=yes

Le fichier filtre permet de préciser le transport à utiliser pour les messages contenant une marque /X-Spam-Level/ ce qui va permettre au message de ne pas boucler indéfiniment...

/X-Spam-Level/ FILTER clientsmtp:127.0.0.1:2727

Configurer, tester, expliquer.

Pour tester simplement spamassassin:

- copier le code source d'un message dans un fichier
- configurer /etc/spamassassin/local.cf avec

required_score -2

rewrite_header Subject *****SPAM*****

- lancer spamd puis la commande spamc < fichier.msg

root@PV-UBUNTU-172:/home/pascal# spamd
janv. 7 13:24:22.587 [11558] info: spamd: server started on port 783/tcp (running version 3.3.2)
janv. 7 13:24:22.587 [11558] info: spamd: server pid: 11558
janv. 7 13:24:22.589 [11558] info: spamd: server successfully spawned child process, pid 11560
janv. 7 13:24:22.590 [11558] info: spamd: server successfully spawned child process, pid 11561
janv. 7 13:24:22.592 [11558] info: prefork: child states: IS
janv. 7 13:24:22.592 [11558] info: prefork: child states: II
janv. 7 13:24:37.117 [11560] info: spamd: connection from localhost [127.0.0.1] at port 40466
janv. 7 13:24:37.120 [11560] info: spamd: setuid to root succeeded
janv. 7 13:24:37.122 [11560] warn: spamd: still running as root: user not specified with -u, not found, or set to root, falling back to nobody
janv. 7 13:24:37.125 [11560] info: spamd: processing message <00000143696fcb4b-4e82c93d-cc43-4c93-aae5-0f5571d5c215-000000@email.amazonses.com> for root:65534
janv. 7 13:24:43.327 [11560] info: spamd: identified spam (1.4/-2.0) for root:65534 in 6.2 seconds, 2339 bytes.
janv. 7 13:24:43.327 [11560] info: spamd: result: Y 1 - NO_DNS_FOR_FROM,SPF_SOFTFAIL scantime=6.2,size=2339,user=root,uid=65534,required_score=-2.0,rhost=localhost,raddr=127.0.0.1,rport=40466,mid=<00000143696fcb4b-4e82c93d-cc43-4c93-aae5-0f5571d5c215-000000@email.amazonses.com>,autolearn=no

janv. 7 13:24:43.379 [11558] info: prefork: child states: II

root@PV-UBUNTU-172:/home/pascal# spamc < /home/pascal/contenu.msg > retour

Dans retour on trouve un message modifié par spamassassin.

Les logs avec la configuration smtp dans mail.log:

Mar 26 16:11:02 ubuntu postfix/smtpd[11331]: connect from pvaniet.in.ac-amiens.fr[172.30.177.85]
Mar 26 16:11:02 ubuntu postfix/smtpd[11331]: 53AE9404458: client=pvaniet.in.ac-amiens.fr[172.30.177.85]
Mar 26 16:11:02 ubuntu postfix/cleanup[11334]: 53AE9404458: message-id=<20080326151102.53AE9404458@ubuntu.in.ac-amiens.fr>
Mar 26 16:11:02 ubuntu postfix/smtpd[11331]: lost connection after QUIT from pvaniet.in.ac-amiens.fr[172.30.177.85]
Mar 26 16:11:02 ubuntu postfix/smtpd[11331]: disconnect from pvaniet.in.ac-amiens.fr[172.30.177.85]
Mar 26 16:11:02 ubuntu postfix/qmgr[11239]: 53AE9404458: from=<pvaniet@in.ac-amiens.fr>, size=407, nrcpt=1 (queue active)
Mar 26 16:11:02 ubuntu clamsmtpd: 1001D1: accepted connection from: 172.30.177.85
Mar 26 16:11:02 ubuntu postfix/smtpd[11337]: connect from localhost[127.0.0.1]
Mar 26 16:11:02 ubuntu postfix/smtpd[11337]: 6BE3340445C: client=localhost[127.0.0.1]
Mar 26 16:11:02 ubuntu postfix/cleanup[11334]: 6BE3340445C: message-id=<20080326151102.53AE9404458@ubuntu.in.ac-amiens.fr>
Mar 26 16:11:02 ubuntu postfix/qmgr[11239]: 6BE3340445C: from=<pvaniet@in.ac-amiens.fr>, size=645, nrcpt=1 (queue active)
Mar 26 16:11:02 ubuntu postfix/smtp[11335]: 53AE9404458: to=<pvaniet@in.ac-amiens.fr>, relay=127.0.0.1[127.0.0.1]:2525, delay=0.15, delays=0.01/0.03/0.06/0.04, dsn=2.0.0, status=sent (250 2.0.0 Ok: queued as 6BE3340445C)
Mar 26 16:11:02 ubuntu postfix/qmgr[11239]: 53AE9404458: removed
Mar 26 16:11:02 ubuntu clamsmtpd: 1001D1: from=pvaniet@in.ac-amiens.fr, to=pvaniet@in.ac-amiens.fr, status=CLEAN
Mar 26 16:11:02 ubuntu postfix/smtpd[11337]: disconnect from localhost[127.0.0.1]
Mar 26 16:11:02 ubuntu spamd[9293]: spamd: connection from localhost [127.0.0.1] at port 56290
Mar 26 16:11:02 ubuntu spamd[9293]: spamd: setuid to clamav succeeded
Mar 26 16:11:02 ubuntu spamd[9293]: spamd: processing message <20080326151102.53AE9404458@ubuntu.in.ac-amiens.fr> for clamav:113
Mar 26 16:11:02 ubuntu spamd[9293]: spamd: clean message (1.4/5.0) for clamav:113 in 0.2 seconds, 630 bytes.
Mar 26 16:11:02 ubuntu spamd[9293]: spamd: result: . 1 - ALL_TRUSTED,AWL,TVD_SPACE_RATIO scantime=0.2,size=630,user=clamav,uid=113,required_score=5.0,rhost=localhost,raddr=127.0.0.1,rport=56290,mid=<20080326151102.53AE9404458@ubuntu.in.ac-amiens.fr>,autolearn=no
Mar 26 16:11:02 ubuntu postfix/pickup[11238]: B070C40445E: uid=113 from=<pvaniet@in.ac-amiens.fr>

Mar 26 16:11:02 ubuntu postfix/cleanup[11334]: B070C40445E: filter: header X-Spam-Level: * from local; from=<pvaniet@in.ac-amiens.fr> to=<pvaniet@in.ac-amiens.fr>: clientsmtp:127.0.0.1:2727
Mar 26 16:11:02 ubuntu postfix/cleanup[11334]: B070C40445E: message-id=<20080326151102.53AE9404458@ubuntu.in.ac-amiens.fr>
Mar 26 16:11:02 ubuntu postfix/pipe[11340]: 6BE3340445C: to=<pvaniet@in.ac-amiens.fr>, relay=spamassassin, delay=0.28, delays=0.04/0/0/0.24, dsn=2.0.0, status=sent (delivered via spamassassin service)
Mar 26 16:11:02 ubuntu postfix/qmgr[11239]: 6BE3340445C: removed
Mar 26 16:11:02 ubuntu postfix/qmgr[11239]: B070C40445E: from=<pvaniet@in.ac-amiens.fr>, size=971, nrcpt=1 (queue active)
Mar 26 16:11:02 ubuntu postfix/smtpd[11344]: connect from pvaniet.in.ac-amiens.fr[172.30.177.85]
Mar 26 16:11:02 ubuntu postfix/smtpd[11344]: B4D7F404458: client=pvaniet.in.ac-amiens.fr[172.30.177.85]
Mar 26 16:11:02 ubuntu postfix/cleanup[11334]: B4D7F404458: message-id=<20080326151102.53AE9404458@ubuntu.in.ac-amiens.fr>
Mar 26 16:11:02 ubuntu postfix/qmgr[11239]: B4D7F404458: from=<pvaniet@in.ac-amiens.fr>, size=1187, nrcpt=1 (queue active)
Mar 26 16:11:02 ubuntu spamd[9288]: prefork: child states: II
Mar 26 16:11:02 ubuntu postfix/smtp[11335]: B070C40445E: to=<pvaniet@in.ac-amiens.fr>, relay=127.0.0.1[127.0.0.1]:2727, delay=0.03, delays=0.02/0/0.01/0.01, dsn=2.0.0, status=sent (250 2.0.0 Ok: queued as B4D7F404458)
Mar 26 16:11:02 ubuntu postfix/qmgr[11239]: B070C40445E: removed
Mar 26 16:11:02 ubuntu postfix/smtpd[11344]: disconnect from pvaniet.in.ac-amiens.fr[172.30.177.85]
Mar 26 16:11:02 ubuntu postfix/local[11345]: B4D7F404458: to=<pvaniet@in.ac-amiens.fr>, relay=local, delay=0.02, delays=0/0/0/0.01, dsn=2.0.0, status=sent (delivered to command: /usr/bin/procmail)
Mar 26 16:11:02 ubuntu postfix/qmgr[11239]: B4D7F404458: removed