

Mise en oeuvre d'une messagerie avec postfix

Cas d'un serveur de messagerie d'un Laboratoire

*Centre d'Océanologie de Marseille
UMS 2196 CNRS*

Maurice.Libes@com.univ-mrs.fr

JT SIARS – Janvier 2005

Introduction

- Aujourd'hui un seul serveur de messagerie ne peut assumer toutes les fonctions nécessaires :
 - Relayage maîtrisé
 - Anti-virus
 - Anti-spam
- Il est donc nécessaire de combiner plusieurs briques
- Une solution à base de logiciels libres
 - Postfix-2.1.4 couplé avec
 - Amavisd-new +
 - antivirus Sophos + sophie + ClamAV
 - SpamAssassin
 - Greylisting (postgrey)

Postfix au Centre d'Océanologie de Marseille

- Un laboratoire CNRS - environ 250 personnes sur 2 sites
- Un mailhost central (MX) qui redistribue sur un site « feuille »
 - Gestion des boites par POPs + IMAPs
 - Postfix-2.1.4 couplé avec
 - Amavisd-new +
 - antivirus Sophos + sophie + ClamAV
 - SpamAssassin
 - Greylisting (postgrey)
- Environ 15000 mails /jour en moyenne (entrées + sorties)
- Pentium IV 1Ghz : 1 Go RAM, 2 disques RAID 30Go
 - Charge CPU assez faible (90% idle)

Introduction

- Anatomie du système : les binaires, les queues
- Configuration, administration de Postfix
 - Exemple de config : un serveur, un « null » client
 - Sécurité :
 - Limiter le Relayage,
 - Divers contrôles anti spam : filtrage d'adresses
 - mise en cage (chroot)
 - limiter les dénis de service : contrôles de charge
 - couplage avec un scanneur de Mail et un antivirus et analyseur de contenu (SA)

Introduction : Pourquoi Postfix?

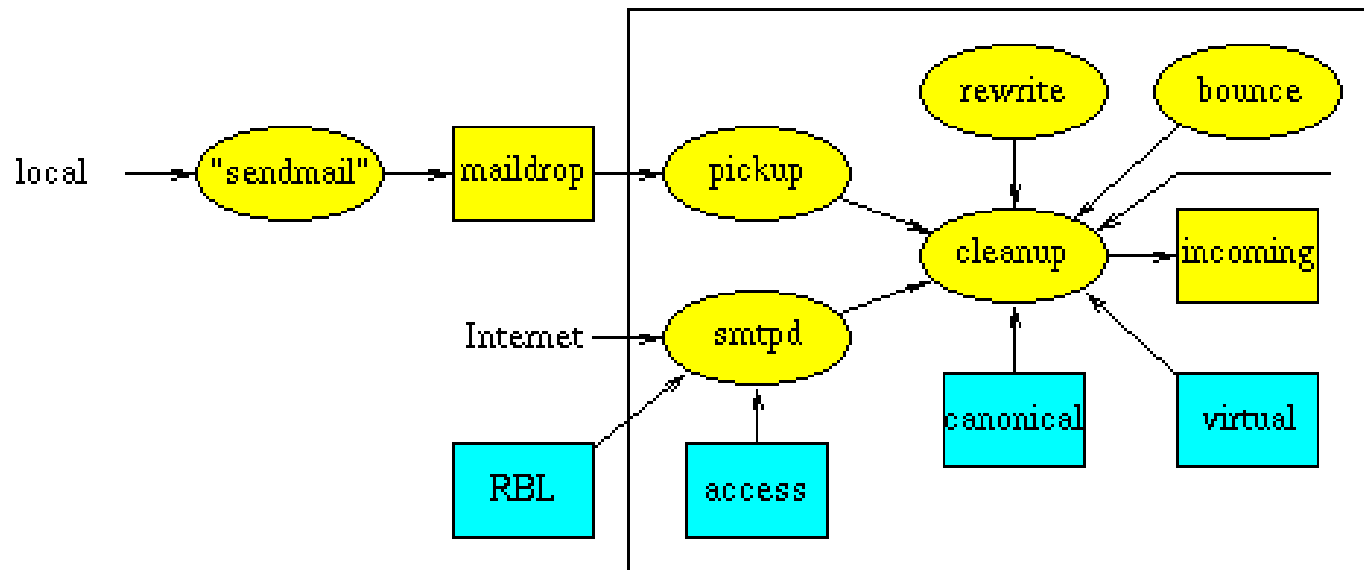
- écrit et maintenu par wietse Venema (tctp-wrapper, satan...)
 - “Postfix attempts to be fast, easy to administer, and hopefully secure, while at the same time being sendmail compatible enough to not upset your users.”
- Se présente comme une alternative à sendmail (*canal historique*) apportant nativement des fonctionnalités intéressantes
- Conçu pour
 - apporter une bonne compatibilité avec sendmail : au niveau de la ligne de commande, des fichiers de conf (alias, .forward)
 - Apporter une simplicité d’administration par une configuration facile à comprendre : *un fichier de conf, des variables « parlantes »*
 - Être rapide et sûr, par une architecture modulaire : *plusieurs process résidents en mémoire, et différentes queues de gestion des mails*

Introduction : Pourquoi Postfix?

- Ecrit avec la sécurité comme principale préoccupation :
 - Architecture non monolithique, extrêmement modulaire, programme petits ayant chacun une fonction précise
 - Nombreuses files d'attentes pour différentes phases de la gestion du mail
 - Pas de programme SUID
 - Peut s'exécuter en environnement "chrooté"
 - Architecture difficile à casser

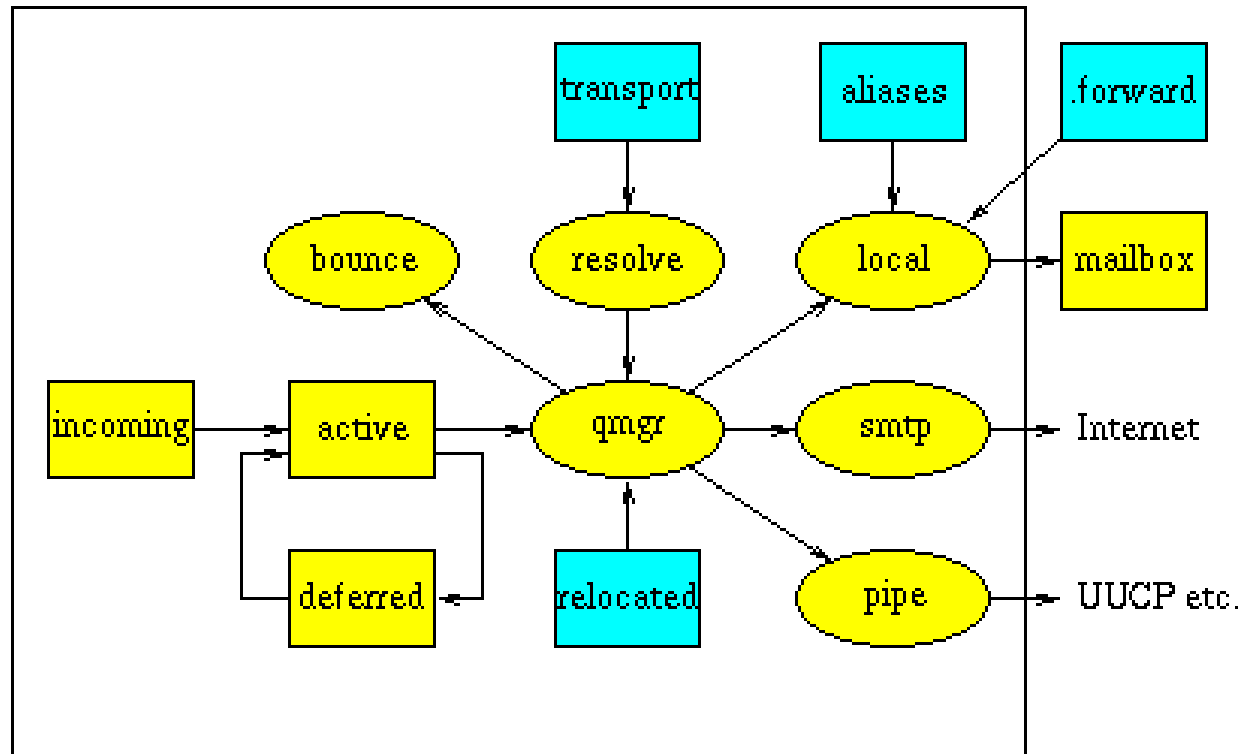
Postfix :architecture du système

- Pas d'architecture monolithique... Postfix est composé d'une dizaine de process assurant chacun une tâche et communiquant entre eux par socket ou queues
- Réception de mail



Postfix : architecture du système

- Délivrance et envoi de mail



Postfix : phase 1 Réception/Insertion locale

- Le daemon « **master** » contrôle le lancement de tous les autres daemons

- **Sendmail / postdrop** : créé pour l'homonymie et compatibilité de l'interface avec *sendmail* « *CH* » :
 - recoit les messages des utilisateurs locaux
 - Dépose le message dans la queue « *maildrop* » via l'intermédiaire du daemon « *postdrop* »
 - Quelques compatibilités avec sendmail:
 - Sendmail -bp ⇔ /usr/bin/mailq -> ../sbin/sendmail
 - Sendmail -I ⇔ /usr/bin/newaliases -> ../sbin/sendmail

- **Postdrop** : dépose un mail dans la queue « *maildrop* » pour ne pas que */var/spool/postfix/maildrop* soit « *world-writable* »

Postfix : phase 1 Réception/Insertion distante

- **Smtpd** : réception de mail d'hôtes distants. « *smtpd* » recoit les connexions réseau (port 25) et engage la transaction smtp. Chaque message reçu est envoyé au daemon « **cleanup** » et le message entrant est déposé dans la queue « **incoming** »

- Réalise les premiers contrôles (directives de « main.cf »)
 - ❑ Contrôle sur les phases *HELO*, *Mail From:*, *RCPT to:* en fonction de fichiers indexés (*liste noire par exemple*)
 - ❑ **content_filter** : possibilité de passer le message à un process qui va analyser le contenu (**amavis**) et rejeter le message ou le réinjecter dans la chaine postfix.
 - ❑ *message_size_limit* : taille max des messages
 - ❑ *smtpd_recipient_limit* : nombre max de destinataires dans un message

Postfix : phase 1 Réception/Insertion

- ***pickup***: simple dépilage de la queue « *maildrop* », réceptionne les messages et les envoie au daemon « *cleanup* »
- Tourne sous root, mais sans aucune interaction avec l'extérieur. Peut être « *chrooté* » dans *master.cf*
- Prend en charge également la directive *content_filter* de « *main.cf* » qui permet de passer le mail à un programme extérieur pour analyse (*amavis*, *spamassin*) avant de le réinjecter dans le circuit, ou le jeter!

Postfix : Phase 2 formatage des entêtes

- **Cleanup + trivial_rewrite** : vérification des entêtes smtp avant de le déposer localement. Examen de Conformité au RFC822. Canonicalise les messages reçus avant de les déposer dans la queue « *incoming* », puis avertit le gestionnaire de queue *qmgr* si tout est correct.
 - ❑ Insère headers manquants : (Resent-) From: Message-Id: Date:
 - ❑ Extrait les adresses de destination de l'enveloppe To: Cc: Bcc:
 - ❑ Accès aux tables des aliases et reverse aliases
 - ❑ Élimine les doublons dans les adresses de destination
 - ❑ Traite les directives de « main.cf » d'examen des headers et du body et de masquerade d'adresses

- **Trivial-rewrite** : appelé optionnellement si les adresses ne sont pas FQDN. destiné à réécrire les adresses au format FQDN
[nom@hote.domaine](#)

Postfix : Phase 3 stockage – livraison - envoi

- **Qmgr** : dépile les messages de « *incoming* » et les remet « intelligemment » (*round robin*) à un agent de livraison en fonction de l'adresse de destination
 - « *local* » | « *procmal* » délivrance locale */var/spool/mail*
 - Gestion du *.forward*
 - *smtp* : résolution DNS de la destination et envoi vers l'extérieur par *smtp*
 - *Pipe* : envoi des messages vers un autre programme externe (*amavis*)
 - *Bounce* : gestion des messages non délivrables : empilement dans la queue *bounce*, et envoi de message à l'émetteur.

Postfix : Les queues de mail

- Répartition intelligente des mails selon leur état de progression dans /
var/spool/postfix/
- → Reprise après crash sans aucun problème
 - *Maildrop* : dépôt des messages émis localement
 - *Incoming* : dépôt des messages entrants émis localement + extérieur smtp
 - *Active* : mail “propres”...en cours de délivrance locale par *qmgr*, taux de dépôt contrôlé et limité
 - *Deferred* : mail ne pouvant pas être délivrés tempo-rairement (*mailq* affiche le status inscrit dans la queue “*defer*”)
 - *qshape.pl deferred*
 - Bounce : messages d’erreur, livraison impossible
- Gestion intelligente des files pour préserver les ressources de la machine: *leaky bucket, fairness, exponential backoff, slow start ...*

Postfix : les fichiers de configuration

- 2 fichiers de configuration principaux:
 - ***/etc/postfix/main.cf*** et ***/etc/postfix/master.cf***

- Des fichiers indexés (*lookup table*) pour rechercher des correspondances (*pas de langage de réécriture d'adresses*)
 - ***aliases*** *ml: libes@com.univ-mrs.fr*
 - ***sender_canonical_maps*** (=reverse aliases)
 - ***access*** (=liste noire)
 - ***protected*** : les listes internes à protéger (*all@labo.univ-xxx.fr*)
 - ***insiders*** : les copains extérieurs autorisés à utiliser nos listes
 - ***header.regexp*, *body.regexp*** : recherche de motifs dans les header ou le body
 - ***mime_headers_checks*** : recherche de motifs dans les entêtes MIME
 - ***relocated*** : liste d'utilisateurs ayant changé d'adresse

Postfix : Les utilitaires d'administration

- *Postfix* [start | stop | reload | check | flush]
 - Postfix reload (*après chaque modification de main.cf*)
- *Postmap* : création des fichiers indexés (.db .dbm)
- *Postconf* [-d || -n || -l || -m] : utilitaire d'affichage ou configuration de *main.cf*
- *Postalias* : maintenance des tables d'alias (compatible sendmail)
- *Postcat* : affichage lisible des mails dans les queues "deferred"
- *Postsuper* : gestion et maintenance des queues de mail (purge après crash)
 - mailq | tail +2 | awk 'BEGIN {RS = ""} /bidon.com/ {print \$1}' | tr -d '!' | postsuper -d -
- *Postlog* : envoyer un message à syslog

Postfix : configuration de base : « main.cf »

■ Indiquer d'où viennent les mails (masquerade From:)

- myhostname = mailhost.labo.univ-mrs.fr (nom FQDN)
- mydomain = labo.univ-mrs.fr
- myorigin = \$myhostname | \$mydomain (*souvent plus adapté*)
- masquerade_domains = \$mydomain

■ Indiquer les messages conservés localement

- (*prévoir le cas où le serveur a plusieurs CNAME ou A record*)
- *Ne pas oublier "localhost" à cause de amavis*
- mydestination = \$myhostname localhost.\$mydomain
www.\$mydomain ftp.\$mydomain

Postfix : configuration de base : « main.cf »

■ Le relayage :

❑ qui autorise t-on à relayer les mails?

❑ Liste spécifique dans “mynetworks”

• mynetworks = 139.124.128.0/22, 127.0.0.0/8

❑ **ou bien**, on laisse postfix régler ça par

• mynetwork_style = host | subnet | class (*subnet par défaut*)

❑ Vers où autorise-t'on à relayer?

❑ Par défaut postfix relaye les mails des sites « étrangers » (clients hors des réseaux autorisés) uniquement vers des réseaux autorisés (les nôtres). Les destinations autorisées de l'extérieur sont celles indiquées dans « **relay_domains** ». Par défaut postfix autorise tous les domaines listés dans « **mydestination** »

■ relay_domains = \$mydestination, 139.124.128.0/22,
139.124.232.0/24, 127.0.0.0/8, localhost, localhost.localdomain

Postfix : configuration de base

■ Un « NULL » client :

Postfix d'une machine qui ne conserve rien... se contente de tout renvoyer vers le mailhost officiel MX du site.

- ❑ myhostname = pcml.com.univ-mrs.fr
- ❑ mydomain = com.univ-mrs.fr
- ❑ myorigin = \$mydomain
- ❑ **relayhost** = mailhost.com.univ-mrs.fr
- ❑ mynetworks = 127.0.0.0/8 139.124.2.0/24 (les réseaux de confiance)
- ❑ mydestination = (rien)

Postfix : Sécurité

- Aucun binaire tournant avec le bit suid, Pas d'exécution sous l'uid root
- **Prémices du filtrage AntiSPAM : filtrage d'adresses au niveau de chaque phase de la transaction SMTP**
 - Contrôle lors de la connexion du *client* sur le port 25
 - Contrôle sur HELO
 - Contrôle sur le *sender* « Mail From »
 - Contrôles sur les différents Headers par regexp (Subject)
 - Contrôles sur le destinataire (recipient)
- Possibilité de mise en cage (chroot) de Postfix
- limitation des dénis de service par contrôle de charge

Postfix : premiers contrôles antispam

1. Contrôle sur le client SMTP (*IP qui initie la connexion*)

smtpd_**client**_restrictions =

permit_mynetworks,	[on autorise nos IP a émettre]
hash:/etc/postfix/access,	[rejet si dans liste noire locale]
reject_maps_rbl,	[rejet si dans \$maps_rbl_domains (désuet en v2.1.x)]
reject_rbl_client list.dsbl.org	[rejet si dans DNSBL list.dsbl.org]
(<i>reject_unknown_client</i>)	[rejet si client sans enregistrement PTR dans le DNS i.e pas de reverse address]

- **maps_rbl_domains** = list.dsbl.org, sbl.spamhaus.org, relays.ordb.org, blackholes.mail-abuse.org, dialups.mail-abuse.org, relays.mail-abuse.org
 - **[ne pas en mettre trop!!!]**

Postfix : LES DNSBL

Une DNSBL est une Liste Noire sous forme de DNS

Les raisons de l'inclusion d'adresses IP recensées dans ces DNSBL peut varier

- Sites qui ont envoyé des spams
- Adresses IP de serveurs de mail agissant en relai ouvert
 - Listes de DNSBL <http://www.moensted.dk/spam>
 - <http://www.declude.com/junkmal/support/ip4r.html>

Postfix : premiers contrôles antispam

2. Contrôles sur la phase HELO

smtpd_helo_required = **yes** *[no par défaut]*

smtpd_helo_restriction =

reject_invalid_hostname, *[adresse malformée]*

reject_non_fqdn_hostname *[adresse non FQDN]*

reject_unknown_hostname, *[adresse sans A ou MX] (sévère)*

Postfix : premiers contrôles antispam

3. Contrôles sur le « sender » Mail from:

(tout permis par défaut)

```
smtpd_sender_restrictions =
```

```
hash:/etc/postfix/access,
```

```
reject_unknown_sender_domain, ## adresse sans MX ou A records  
dans le DNS
```

```
reject_non_fqdn_sender, ##adresse non FQDN
```

```
check_sender_access, hash:/etc/postfix/access, ## rejet si présence  
en liste noire locale
```

```
check_sender_mx_access hash:/etc/postfix/mx_access, ## rejet si  
MX bidons
```

```
reject_maps_rbl
```

Postfix : contrôles sur mx_sender

- `grep « RFC 1918 » /var/log/mail/info`

```
Jan 22 15:45:29 com1 postfix/smtpd[22894]: NOQUEUE: reject: RCPT from
unknown[211.187.59.94]: 554 <QVPEXX@huhmail.com>: Sender
address rejected: mail server in RFC 1918 private network;
from=<QVPEXX@huhmail.com> to=<gilbert@com.univ-mrs.fr>
proto=SMTP helo=<139.124.2.100>
```

```
$ host -t mx huhmail.com
```

```
huhmail.com mail is handled by 0 mail.huhmail.com.
```

```
$ host mail.huhmail.com
```

```
mail.huhmail.com has address 192.168.255.255
```

Postfix : contrôles sur mx_sender

- **more /opt/postfix/etc/mx_access**
- 64.94.110.11 reject mail server in verisign wild-card domain
- 127 reject mail server in loopback network
- 0 reject mail server in broadcast network
- 10 reject mail server in RFC 1918 private network
- 169.254 reject mail server in link local network
- 172.16 reject mail server in RFC 1918 private network
- 192.0.2 reject mail server in TEST-NET network
- 192.168 reject mail server in RFC 1918 private network
- 69.6.61 bad MX record -- spammer
- 209.133.120 bad MX record -- spammer

Postfix : premiers contrôles antispam

4. Contrôles sur le destinataire rcpt to

relay_domains = \$mydestination, 139.124.128.0/22, 139.124.232.0/24,
127.0.0.0/8, localhost, localhost.localdomain

smtpd_**recipient**_restrictions =

permit_mynetworks, (accepte si le rcpt to est dans mynetworks)

hash:/etc/postfix/protected, (protection des listes internes)

reject_unauth_destination, (rejette le mail sauf si le destinataire est dans
relay_domains ou mydestination)

check_policy_service inet:127.0.0.1:10023 (pour le GreyListing)

smtpd_restriction_classes = insiders_only [protection listes internes]

insiders_only = check_sender_access, hash:/etc/postfix/insiders, reject

Postfix : examen de contenus

5. Filtrage sur des motifs (regexp) des header SMTP

- **header_checks** = regexp:/etc/postfix/header.regexp

```
/^Content-Type: multipart.*"----[A-F0-9]
+_Outlook_Express_boundary"/i REJECT
```

```
/^Subject:.*\[0-9]+\.*$/ REJECT
```

```
/^Subject:.*\DVD.*$/ REJECT
```

```
/^Subject:.*\Viagra.*$/ REJECT
```

Postfix : premiers examen de contenus

6. Filtrage sur des motifs (regexp) dans les entêtes MIME

Une directive spéciale pour ne vérifier que les entêtes des attachments MIME situés à l'intérieur du message.. **Utile pour les extensions des pièces jointes**

- ❑ **Mime_header_checks =**
regexp:/etc/postfix/mime_header_checks.regexp

```
/^\s*Content-(Disposition|Type).*name\s* =\s*"?(.+\. (cp|lnk|asd|hlp|ocx|  
reg|bat|c[ho]m|cmd|dll|vxd|pif|wab|scr|hta|jse?|sh[mbs]|vb[esx]|ws  
[fh]|wav|mov|xl))"? \s*$/
```

REJECT piece jointe non autorisee. Le fichier "\$2" du type "\$3" est refuse.

Postfix : Directives de contrôle de charge

Des paramètres pour réguler la charge et lutter contre les DoS

- ❑ limitation du nombre des process lancés
 - *default_process_limit* (défaut: 50) : contrôle le taux d'entrées sorties de mails via le nombre de process concurrents de postfix (smtp, smtpd)
- ❑ régulation du nombre de connexions simultanées
 - *local_destination_concurrency_limit* : max de délivrance simultanées locale identique
 - *default_destination_concurrency_limit* : nombre max de mail simultanés vers un site distant (protection des sites extérieurs)
- ❑ limitation du nombre de destinataires par mail
 - *smtpd_recipient_limit* : nombre maximal de destinataires dans un mail
- ❑ gestion des tentatives de réémission vers sites inaccessibles
 - *maximal_queue_lifetime* (5 jours) : temps de rétention avant que le message soit déclaré non délivrable
 - *queue_run_delay* (1000s) : fréquence des retentatives

Postfix : contrôle de ressources mémoire

Postfix possède des directives de limite d'utilisation de la mémoire afin de limiter les dénis de services.

« *The idea is to keep running under conditions of stress, without making the problem worse.* »

- ❑ `message_size_limit = 7000000`
- ❑ `Header_size_limit = 2048`
- ❑ `line_length_limit = 102400`
- ❑ `bounce_size_limit = 50000`

Postfix : Directives de réécriture d'adresses

Pas de langage complexe de réécriture. Tout se fait à travers de simples fichiers indexés (lookup table) au format dbm ou hash)

- réécriture en prénom.nom
 - `sender_canonical_maps = /etc/postfix/revaliases`

- masquage d'adresse
 - `masquerade_domains = com.univ-mrs.fr`

- redirection d'adresses virtuelles
 - `virtual_alias_maps = hash:/etc/postfix/virtual`

Postfix Protection des listes de mail

■ Protection des listes de mail internes

```
smtpd_recipient_restrictions = hash:/etc/postfix/protected,  
smtpd_restriction_classes = insiders_only
```

```
insiders_only = check_sender_access, hash:/etc/postfix/insiders,  
reject
```

□ Dans /etc/postfix/insiders

mlibes@wanadoo.fr OK

Gerard.manvussa@lset.univ-ersite.fr OK

□ Dans /etc/postfix/protected

info-soc@com.univ-mrs.fr

insiders_only

liste-ita@com.univ-mrs.fr

insiders_only

Postfix : environnement chrooté (dans *master.cf*)

- ❑ `mkdir /var/spool/postfix/etc /var/spool/postfix/lib`
- ❑ `mkdir -p /var/spool/postfix/usr/lib/zoneinfo`
- ❑ `cp /etc/localtime /etc/services /etc/resolv.conf /
etc/nsswitch.conf ~etc`
- ❑ `ln -s /etc/localtime ~usr/lib/zoneinfo`
- ❑ `cp /lib/libnss_* ~lib`

```
# =====  
# service type private unpriv chroot wakeup maxproc(50) command arg  
# =====  
smtp      inet  n       -       y       -       -       smtpd  
pickup    fifo  n       -       y        60      1       pickup  
cleanup   unix  n       -       y        -       0       cleanup  
qmgr      fifo  n       -       y        300     1       nqmgr  
rewrite   unix  -       -       y        -       -       trivial-rewrite  
smtp      unix  -       -       y       -       -       smtp
```

Conclusions : Pourquoi Postfix?

- Simplicité d'administration (*majorité d'options par défaut!*)
- configuration facile à comprendre : *un fichier de conf, des variables « parlantes »*
- fonctionnalités intéressantes (filtrages, contrôles de charge, collaboration avec autres programmes externes)
- *Excellente tenue en charge en cas de déni de service*
- Architecture modulaire : *plusieurs binaires , et queues de gestion des mails*
- Sécurité prise en compte nativement : *relayage, mise en cage, contrôle antispam basique, liste noire, pas de binaires suid, ...*

Conclusions : Pourquoi Postfix?

- bonne compatibilité avec les fichiers de conf de sendmail, (alias, .forward)
- Excellente portabilité sur plusieurs plateformes
- Pas de mauvaises surprises en exploitation, ça fait ce que ça dit...
- Exploitation sans problème au COM depuis 5ans : ça marche tout seul
- Large communauté d'utilisateurs et de développeurs : Beaucoup de logiciels complémentaires

→ <http://www.postfix.org>

AMAVISD-NEW

<http://www.ij.s.si/software/amavisd/>

- Amavisd-new est un scanneur de mail pour détecter virus et/ou spams
- C'est un daemon en Perl interfacé avec postfix par un *socket*. Amavisd-new est une amélioration des versions précédentes, car il est « daemonisé » (résident en mémoire et pré-forké i.e plusieurs daemon en attente)
 - Il permet d'appliquer un filtrage de contenu par un antivirus ou un antispam
 - Il reconnaît l'installation d'un grand nombre d'antivirus (dont sophos et clamav)
 - Il lance le module perl Mail::SpamAssassin pour détecter les spams
 - On l'utilise généralement au niveau de la réception de mails sur le serveur de mails
 - il ajoute un entête, peut mettre en quarantaine, rejeter, émettre un avis de non délivrance

AMAVISD-NEW : installation

- Bien lire le fichier INSTALL
 - ❑ Créer un compte et groupe “amavis”
 - ❑ Adduser amavis –s /bin/false
 - ❑ mkdir /var/amavis
 - ❑ mkdir /var/amavis/tmp /var/amavis/var /var/amavis/db
 - ❑ chown -R amavis:amavis /var/amavis
 - ❑ chmod -R 750 /var/amavis
 - ❑ cp amavisd.conf /etc/ ; chown root /etc/amavisd.conf
 - ❑ chmod 644 /etc/amavisd.conf
 - ❑ mkdir /var/virusmails ; chown amavis:amavis /var/virusmails
 - ❑ chmod 750 /var/virusmails

- Récupérer les modules Perl
 - ❑ perl –MCPAN –e shell

AMAVISD-NEW : *modules perl nécessaires*

- **Archive::Tar** (Archive-Tar-x.xx)
- **Archive::Zip** (Archive-Zip-x.xx) (1.14 or later should be used!)
- **Compress::Zlib** (Compress-Zlib-x.xx)
- **Convert::TNEF** (Convert-TNEF-x.xx)
- **Convert::UUlib** (Convert-UUlib-x.xxx) (stick to the new versions!)
- **MIME::Base64** (MIME-Base64-x.xx)
- **MIME::Parser** (MIME-Tools-x.xxxx) (latest version from CPAN)
- **Mail::Internet** (MailTools-1.58 or later have workarounds for Perl bugs)
- **Net::Server** (Net-Server-x.xx)
- **Net::SMTP** (libnet-x.xx) (use libnet-1.16 or latter for performance)
- **Digest::MD5** (Digest-MD5-x.xx)
- **IO::Stringy** (IO-stringy-x.xxx)
- **Time::HiRes** (Time-HiRes-x.xx) (must use 1.49 or later,)
- **Unix::Syslog** (Unix-Syslog-x.xxx)
- **BerkeleyDB** with bdb library 3.2 or later (4.2 or later preferred)

AMAVISD-NEW : progs nécessaires

file: <ftp://ftp.astron.com/pub/file/>
compress: <ftp://ftp.warwick.ac.uk/pub/compression/>
gzip: <http://www.gzip.org/>
bzip2: <http://sources.redhat.com/bzip2/>
nomarch: <http://rus.members.beeb.net/nomarch.html>
arc: <ftp://ftp.kiarchive.ru/pub/unix/arcers/>
lha: <http://www2m.biglobe.ne.jp/~dolphin/lha/prog/>
unarj: <ftp://ftp.kiarchive.ru/pub/unix/arcers/>
arj: <http://testcase.newmail.ru/files/> (arj is preferable to unarj)
rar, unrar: <http://www.rarsoft.com/>, <ftp://ftp.kiarchive.ru/pub/unix/arcers/>
zoo: <ftp://ftp.kiarchive.ru/pub/unix/arcers/>
lzop: <http://www.lzop.org/download/>
freeze: <ftp://ftp.warwick.ac.uk/pub/compression/>
ripOLE: <http://www.pldaniels.com/ripole/>
pax: <http://www.gnu.org/software/paxutils/>
cpio: <http://www.gnu.org/software/cpio/>
ClamAV: <http://clamav.elektrapro.com/> (open source virus scanner)
~~SAVI: <http://www.csupomona.edu/~henson/www/projects/SAVI-Perl/dist/>~~
dspam: <http://www.nuclearelephant.com/projects/dspam/>

AMAVISD-NEW: interfaçage avec postfix

- Dans main.cf
 - **content_filter** = smtp-amavis:[127.0.0.1]:10024

- Dans master.cf

```
smtp-amavis  unix  -      -      n      -      2      smtp
```

```
-o smtp_data_done_timeout=1200  
-o smtp_send_xforward_command=yes  
-o disable_dns_lookups=yes
```

```
127.0.0.1:10025 inet  n      -      n      -      -      smtpd
```

```
-o content_filter=  
-o local_recipient_maps=  
-o relay_recipient_maps= -o smtpd_restriction_classes=  
-o smtpd_client_restrictions= -o smtpd_helo_restrictions=  
-o smtpd_sender_restrictions=  
-o smtpd_recipient_restrictions=permit_mynetworks,reject  
-o mynetworks=127.0.0.0/8
```

AMAVISD-NEW : configuration

- Editer /etc/amavisd.conf (*beurk*)
- `$max_servers = 3;` # number of pre-forked children
- `$daemon_user = 'vscan';` # (no default; customary: vscan or amavis)
- `$daemon_group = 'vscan';` # (no default; customary: vscan or amavis)

- `$mydomain = 'com.univ-mrs.fr';` # a convenient default for other settings

- `$MYHOME = '/var/amavis';` # a convenient default for other settings
- `$TEMPBASE = "$MYHOME/tmp";` # directory, needs to be created
- `$QUARANTINEDIR = '/var/virusmails';`

AMAVISD-NEW : configuration

- `$inet_socket_port = 10024; # daemon écoute sur ce port TCP`
- `$sa_spam_subject_tag = '***SPAM***'; #tag mis dans le Subject si...`

- `$sa_tag_level_deflt = 2.0; # ajoute entetes spam si >= niveau`
- `$sa_tag2_level_deflt = 5.0; # ajoute 'spam detected' headers`
- `$sa_kill_level_deflt = 15.0; # declenche destruction de spam (quarantaine)`
- `$sa_dsn_cutoff_level = 10; # niveau en dessous duquel DSN n'est pas envoyé`
- `$sa_mail_body_size_limit = 200*1024; #taille en dessous de laquelle SA examine le mail`

- `$final_virus_destiny = D_BOUNCE || D_DISCARD || D_PASS || D_REJECT;`
- `$final_banned_destiny = D_BOUNCE;`
- `$final_spam_destiny = D_PASS ;`
- `$final_bad_header_destiny = D_PASS;`

AMAVISD-NEW : configuration

- **D_REJECT** : mail non délivré, un avis de non remise est renvoyé par postfix si possible
- **D_BOUNCE** : idem, avis envoyé par amavisd-new, SAUF si un virus fait partie de la liste ***\$viruses_that_fake_sender_re***
(reject et bounce sont quadi identiques, la différence provient de qui envoie l'avis)
- **D_DISCARD**: mail rejeté, aucun avis envoyé à l'extérieur.. Mail perdu, mais mis en quarantaine
- **D_PASS** : le mail passe et est délivré même en dépit d'un mauvais contenu (spam ou virus)
- Liste des virus qui usurpent les adresses:
 - ***\$viruses_that_fake_sender_re*** = new_RE(
qr'nimda|hybris|klez|bugbear|yaha|braid|sobig|fizzer|palyh|peido|holar'i,

AMAVISD-NEW : configuration

- Décommenter les lignes correspondant aux antivirus présents

- **@av_scanners = (**

- **['ClamAV-clamd',**

```
\&ask_daemon, ["CONTSCAN {\n", "/var/run/clamav/clamd"],  
qr/\bOK$/, qr/\bFOUND$/,  
qr/^.*?: (?!Infected Archive)(.*) FOUND$/ ],
```

- **['Sophie',**

```
\&ask_daemon, [{"}/\n", '/var/run/sophie'],  
qr/(?x)^ 0+ ( : | [\000\r\n]* $)/, qr/(?x)^ 1 ( : | [\000\r\n]* $)/,  
qr/(?x)^ [-+]? \d+ : (.*) [\000\r\n]* $/ ],
```

AMAVISD-NEW : lancement

- Avoir installé un antivirus et spamassassin au préalable (ils sont reconnus au lancement)
- Il suffit de lancer le daemon
 - `/usr/local/bin/amavisd [debug]`
- En écoute sur le port 10024, il récupère les mails que lui envoie Postfix.. Les traite et les remet dans la chaîne de traitement sur le port 10025
- Bien regarder les logs lors du lancement
- En période de tests, placer la variable de postfix
 - ***soft_bounce=yes***
 - En cas de problème aucun message n'est rejeté ou mis en bounce

L'antivirus CLAMAV : <http://www.clamav.net>

- Clamav est un Antivirus openSource très efficace (100% vs Sophos), découverte et mise à jour parfois avant les AV commerciaux.
- Il est bon d'avoir 2 antivirus qui se complètent mutuellement
 - En cas de défaut de mise à jour de l'un
 - En cas d'erreurs sur le fonctionnement d'un des 2
 - Ex:Jan 23 15:42:14 com1 amavis[26466]: (26466-02) ClamAV-clamd: Can't connect to UNIX socket /var/run/clamav/clamd: Permission denied, retrying (2)
- Installation:
- `./configure --sysconfdir=/etc && make && make install`

L'antivirus CLAMAV : Configuration

■ Dans /etc/clamav.conf

- LogFile /var/log/clamd.log
- LogTime
- LogSyslog
- PidFile /var/run/clamd.pid
- DatabaseDirectory /usr/local/share/clamav
- LocalSocket /tmp/clamd**
- ## attention avec le socket déclaré dans /etc/amavisd.conf**
 - ['ClamAV-clamd',
 - # \&ask_daemon, ["CONTSCAN {\n", **"/var/run/clamav/clamd"**],
 - \&ask_daemon, ["CONTSCAN {\n", **"/tmp/clamd"**],
 - qr/\bOK\$/ , qr/\bFOUND\$/ ,
 - qr/^..*?: (?!Infected Archive)(.*) FOUND\$/],

- User clamav

L'antivirus CLAMAV : *Lancement*

■ Lancement du daemon de scan clamd:

- ❑ `/bin/chown clamav /var/log/clam*`
- ❑ `chmod 600 /var/log/clam-update.log /var/log/clamd.log`
- ❑ `/usr/local/sbin/clamd`

■ Lancement du daemon de mise à jour

- ❑ `/usr/local/bin/freshclam -d -c 6 -l /var/log/clam-update.log`

■ Lister les signatures de virus

- ❑ `sigtool --list-sigs`

■ Lancer clamav avant amavisd, Amavisd détecte automatiquement la présence de clamav

- Jan 23 15:58:22 com1 amavis[26925]: Using internal av scanner code for (primary) ClamAV-clamd
- Jan 23 15:58:22 com1 amavis[26925]: Using internal av scanner code for (primary) Sophie
- Jan 23 15:58:22 com1 amavis[26925]: Found secondary av scanner ClamAV-clamscan at /usr/local/bin/clamscan

Sophie : interface entre Amavisd et Sophos

- Sophie est un daemon qui utilise la librairie 'libsavi' de l'antivirus Sophos (*il est donc nécessaire d'avoir sophos AV + les définitions de virus sur disque*)
- Au lancement, Sophie :
 - utilise l'interface de Sophos Anti-Virus (libsavi),
 - charge les définitions de virus en mémoire,
 - établit un socket et attend une connexion et un mail a scanner
- Sophie et les définitions de virus sont chargées en RAM, le scan des mails est donc beaucoup plus rapide

Sophie : configuration

- Configure `--with-savilib=[path] --enable-net && make`

- `/etc/sophie.cfg` : configuration du daemon sophie
 - Majorité d'optiins par défaut
 - `pidfile: /var/run/sophie.pid`
 - `socketfile: /var/run/sophie`
 - `user: vscan`

- `/etc/sophie.savi` : configuration de SAVI

Sophie : Lancement

- `/usr/local/sbin/sophie -D`
- **Attention**: si les définitions d'antivirus de Sophos changent, il FAUT RELANCER sophie (commande à intégrer dans le script de maj de sophos)
 - Rechargement des signatures d'antivirus
 - `Killall sophie -HUP`

Le Greylisting

- Méthode récente (mi-2003) pour limiter le taux de spams reçus
- Implémenté uniquement au niveau du MTA
 - Endroit le plus nocif pour les spammeurs
- Caractéristiques:
 - Simple à mettre en place et maintenir
 - Impact minime pour les utilisateurs
 - Pas d'analyse de contenu de message
- Tiens la charge (testé sur serveurs 10^6 mails)

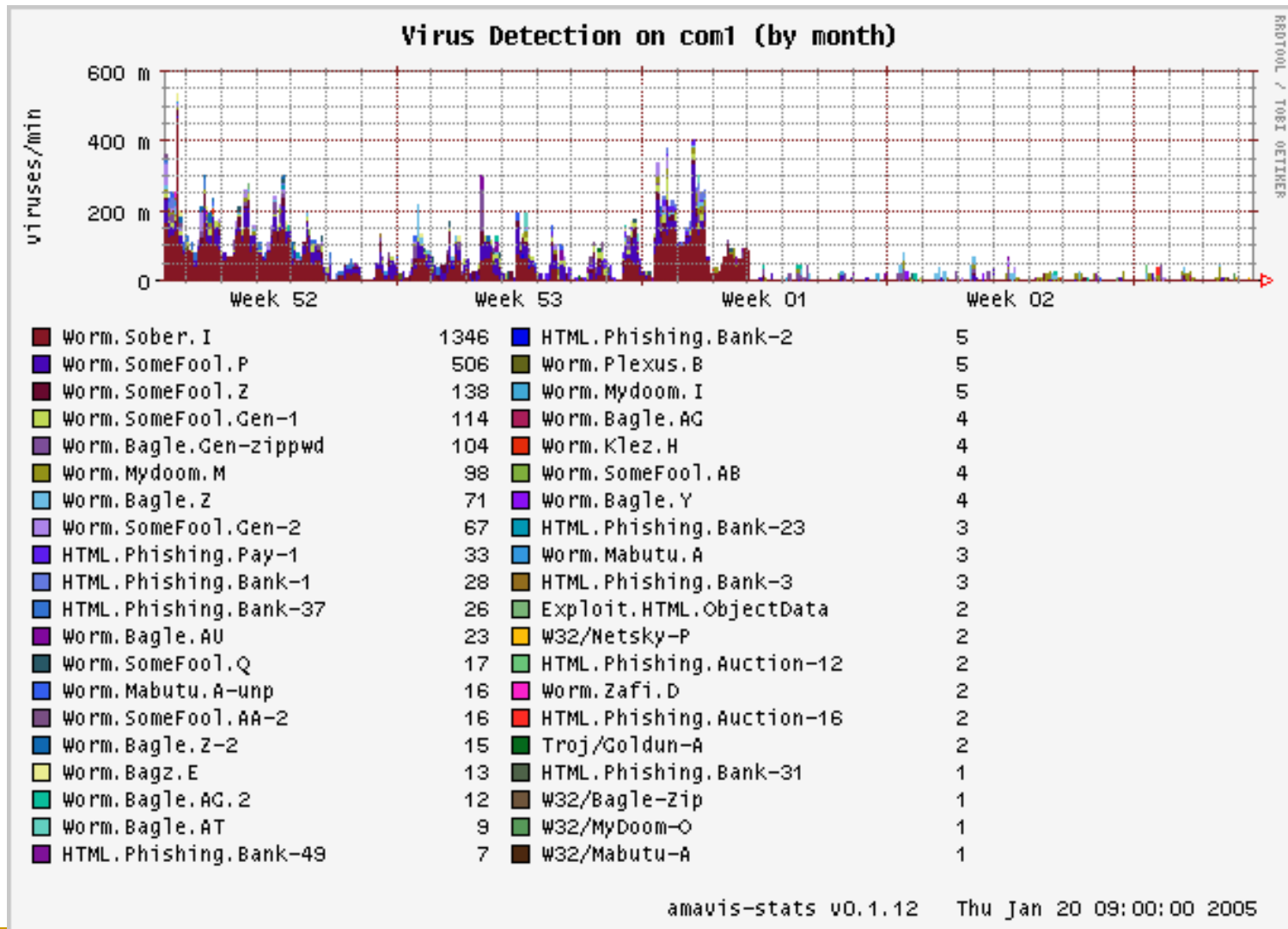
Le Greylisting : principe de base

- Part de l'observation selon laquelle les spams sont envoyés depuis des applications spéciales pour spammer (envoi de milliers de messages à la chaîne)
- Ces applications font du « *fire and forget* »
- Elles envoient leur spam vers des MX de domaines, mais ont une durée de vie temporaire.
- En cas de problème et de refus temporaire des MX, elles ne rémettent pas leur message comme le veut SMTP
- Il suffit donc de refuser tous les mails entrants et d'attendre que le serveur émetteur, ré-émette pour pouvoir penser que ce n'est pas un spammeur

Le Greylisting : principe de base

- Le programme travaille sur 3 informations (triplet) :
 - L'adresse IP de la machine cliente (client)
 - L'adresse Email de l'émetteur (sender)
 - L'adresse Email du destinataire (recipient)
- Ce triplet est logué dans une base locale (fichier .db) à chaque apparition
 - Quand le triplet se présente la 1ere fois (absent dans la base), il est refusé pendant un certain « delay ». Le MX local renvoi un code d'erreur 450 « *temporary failure* » indiquant que l'émetteur doit ré-essayer plus tard
 - Si le serveur distant ré-emet, le triplet se représente, il est déjà connu, on accepte la connexion SMTP
 - Sinon ? Le mail est perdu

Le Greylisting : les résultats



Le Greylisting : inconvénients

- Inconvénients liés délai de refus de notre site + durée de rétention avant réémission (stocké dans le spool « deferred » chez l'émetteur, avant réémission)
- Problème d'urgence dans les mails?
 - 10 mn avant un séminaire: « *tiens je te passe les transparents que je viens de finir* »

Le Greylisting : inconvénients

■ Inconvénients liés à la durée de réémission

./time-greylst.pl /var/mail/libes

==> traitement du fichier /var/mail/libes sur com1

X-Greylst: delayed 1274 seconds by postgrey-1.17 at com1; Wed, 05 Jan 2005 00:16:53 CET
ligne 1 : time= 1274

X-Greylst: delayed 1217 seconds by postgrey-1.17 at com1; Wed, 05 Jan 2005 17:42:22 CET
ligne 2 : time= 1217

X-Greylst: delayed 1544 seconds by postgrey-1.17 at com1; Thu, 06 Jan 2005 13:13:01 CET
ligne 3 : time= 1544

X-Greylst: delayed 1022 seconds by postgrey-1.17 at com1; Mon, 10 Jan 2005 18:18:12 CET
ligne 4 : time= 1022

X-Greylst: delayed 1700 seconds by postgrey-1.17 at com1; Mon, 17 Jan 2005 16:30:38 CET
ligne 5 : time= 1700

X-Greylst: delayed 89898 seconds by postgrey-1.17 at com1; Wed, 19 Jan 2005 18:32:49 CET
ligne 6 : time= 89898

temps total de retention = 96655 pour 6 messages

moyenne de retention = 16109.1666666667

Le Greylisting : inconvénients

■ Inconvénients liés à la durée de réémission

```
./time-greylis.pl /var/mail/thyssen
```

```
==> traitement du fichier /var/mail/thyssen sur com1
```

```
X-Greylis: delayed 815 seconds by postgrey-1.17 at com1; Thu, 06 Jan 2005 17:28:10 CET
```

```
ligne 1 : time= 815
```

```
X-Greylis: delayed 1454 seconds by postgrey-1.17 at com1; Tue, 11 Jan 2005 12:15:28 CET
```

```
ligne 2 : time= 1454
```

```
X-Greylis: delayed 320 seconds by postgrey-1.17 at com1; Thu, 13 Jan 2005 11:38:27 CET
```

```
ligne 3 : time= 320
```

```
X-Greylis: delayed 3341 seconds by postgrey-1.17 at com1; Sun, 16 Jan 2005 23:13:18 CET
```

```
ligne 4 : time= 3341
```

```
X-Greylis: delayed 1607 seconds by postgrey-1.17 at com1; Mon, 17 Jan 2005 08:08:36 CET
```

```
ligne 5 : time= 1607
```

```
X-Greylis: delayed 2727 seconds by postgrey-1.17 at com1; Mon, 17 Jan 2005 15:47:33 CET
```

```
ligne 6 : time= 2727
```

```
X-Greylis: delayed 599 seconds by postgrey-1.17 at com1; Mon, 17 Jan
```

```
ligne 7 : time= 599
```

```
temps total de retention = 10863 pour 7 messages
```

```
moyenne de retention = 1551.85714285714
```

Le Greylisting : la solution=liste blanche

- Passer outre ce problème de refus des mails en 1ère approximation. Eviter le refus et la rétention pour certains sites amis
- Possibilité de Mise en liste blanche
 - **De certains sites émetteurs**
 - ***/opt/postfix/etc/postgrey_whitelist_clients***
 - univ-mrs.fr ou /\.*\.\univ.*\.\fr\$/
 - /\.*\.\cnrs.*\.\fr\$/
 - cnrs-mrs.fr
 - u-3mrs.fr
 - egim-mrs.fr
 - **De certains destinataires (recipient) locaux (les rôleurs)**
 - ***/opt/postfix/etc/postgrey_whitelist_recipients***
 - postmaster@
 - abuse@
 - libes@

Greylisting : détails

- Refus immédiat dès la connexion smtp, pas de ressources CPU consommées
- La base conserve:
 - La date où le triplet se présente la 1ere fois
 - La durée du blocage
 - La date où l'enregistrement du triplet expire (nettoyage des vieux enregistrements)
 - Le nombre de tentatives bloquées du triplet
 - Le nombre de mails du triplet qui sont passés

Le Greylisting : Postgrey

- Postgrey est 1 implémentation de Greylisting en perl couplée avec Postfix
 - rajouter une directive dans « *main.cf* »

```
smtpd_recipient_restrictions = hash:/opt/postfix/etc/protected,  
    permit_mynetworks,  
    reject_unauth_destination,  
    check_policy_service inet:127.0.0.1:10023
```
 - <http://isg.ee.ethz.ch/tools/postgrey/pub>
 - S'installe en 30', nécessite quelques modules Perl
 - - Perl >= 5.6.0
 - - Net::Server (Perl Module)
 - - IO::Multiplex (Perl Module)
 - - BerkeleyDB (Perl Module)
 - - Berkeley DB >= 4.1 (Library)

Le Greylisting : lancement

- `mkdir /var/spool/postfix/postgrey`
- `adduser postgrey && chown postgrey /var/spool/postfix/postgrey`
- Lancer le daemon
- `/usr/local/bin/postgrey -d -v`
 - `--inet=10023`
 - `--dbdir=/var/spool/postfix/postgrey`
 - `--whitelist-recipients=/opt/postfix/etc/postgrey/whitelist_recipients`
 - `--whitelist-clients=/opt/postfix/etc/postgrey_whitelist_clients`
 - `--delay=300` *## durée de refus temporaire 451 depuis la 1ere apparition*
 - `--max-age=N` *## destruction des triplets plus anciens que N jours*

Mail rejeté avec erreur temporaire 451 lors de la 1ere apparition du triplet, ou bien s'il se représente dans les 5minutes (delay) qui suivent

SpamAssassin : Principe de base

- Amavisd-new travaille avec le module Perl de SpamAssassin (SA)
- SA travaille sur des heuristiques : base de règles empiriques permettant de reconnaître des occurrences de spam dans les mail
- Quand le motif d'une règle est reconnu dans un mail, un score est affecté au mail
 - body PAY_SITE \bpay[-]?sites?\b/i
 - describe PAY_SITE Possible porn - Pay Site
 - score PAY_SITE 2.699 2.599 0 2.230
- Chaque règle donne un score qui est additionné au score total
- Plus le score total est élevé, plus le mail a une probabilité forte d'être un Spam...

SpamAssassin : fichiers de conf

- Installation:
 - perl -MCPAN -e shell
 - Cpan>install Mail::SpamAssassin

- La distribution fournit des fichiers de configuration à ne PAS modifier
 - /usr/share/spamassassin/*.cf
 - Règles de base, scores de base

- Personnalisation des règles de reconnaissance dans
 - /etc/mail/spamassassin/local.cf
(attention le fichier est écrasé en cas de mise à jour de SA)

SpamAssassin : heuristiques

- Reconnaissance de motifs dans les mails grâce à la puissance des «expressions régulières»

- Les règles s'appliquent sur le corps du message (BODY) ou les entêtes (Subject)

body FR_REMISE *Abprofiter (?:d'une|de (la|notre)) (?:offre|remise|promo(tion)?)
s?\b/i*

describe FR_REMISE Parle de promotion, en Français
score 2.5

header ML_hd_order *Subject =~ /.*(order|g[eo]t){1,10}meds*/i*
describe ML_hd_order *Subject contains order or get got medecines*
score ML_hd_order 3.50 # 1689s/25h of 58857 corpus

- Balise “**meta**” pour combiner plusieurs règles de base

meta ML_offer_save ML_offer && ML_saveup
describe ML_offer_save *Site offer and saveup*
score ML_offer_save 6.0

SpamAssassin : heuristiques

- Règles portant sur des URLs citées dans les mails

```
uri URI_ML_10 /*online.*net/  
score URI_ML_10 4.0
```

```
uri URI_ML_11 /saf.gruj.com\rep\sales/  
score URI_ML_11 5.0
```

- Pour tester les règles sur un mail

- Spamassassin -D < ./mail-bizarre.txt
- HOWTO pour écrire des règles
- <http://mywebpages.comcast.net/mkettler/sa/SA-rules-howto.txt>