HERVÉ SCHAUER CONSULTANTS
Cabinet de Consultants en Sécurité Informatique depuis 1989
Spécialisé sur Unix, Windows, TCP/IP et Internet

Sécuriser sa messagerie d'entreprise

01 SÉMINAIRES

Serveur de messagerie sécurisé et libre

Denis Ducamp
<Denis.Ducamp@hsc.fr>

Introduction

- Aujourd'hui un seul serveur de messagerie ne peut assumer toutes les fonctions :
  - relayage SMTP sécurisé
  - anti-virus
  - anti-spam
- il est donc nécessaire de combiner plusieurs briques.
- Si l'anti-virus permet de protéger les postes clients
  - l'architecture doit elle même être solide (séparation des privilèges...)
  - et fournir certaines fonctionnalités (anti-relayage, smtp/tls...)
  - pour s'adapter aux contraintes de l'Internet.
- Il est enfin possible de n'utiliser que des logiciels libres
  - pour monter une telle architecture.

Plan

- Présentation et installation des logiciels utilisés
  - postfix/tls
  - amavisd-new
  - SpamAssassin
    - DNSBL et bases de spams
    - Razor, Pyzor et DCC
  - clamav
- Administration
  - Côté client
  - Clients et serveurs Windows
  - Optimisations
  - Mises à jour anti-virales et anti-spam
  - Statistiques
- Conclusion

Plan : Présentation

- postfix/tls
  - architecture
  - anti-spam / anti-relayage
  - filtrage de contenu
  - SMTP/TLS
- amavisd-new
- SpamAssassin
  - DNSBL
  - Bases de spams
- clamav

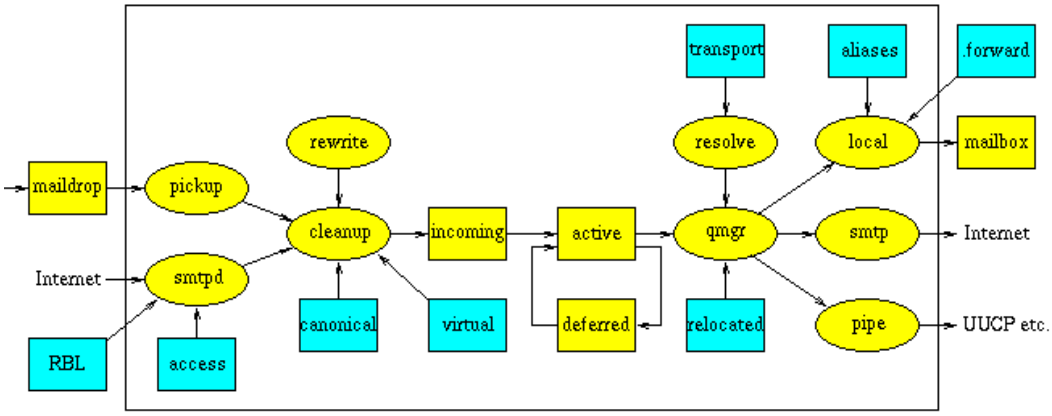
postfix

- Écrit par Wietse Venema (auteur de TCP-Wrapper, Satan et TCT)
  - <http://www.postfix.org/>
- Compatibilité sendmail maximale.
- Écrit avec la sécurité comme principale préoccupation :
  - modulaire :
    - programmes petits et lisibles
    - chaque fonction est isolée
  - chaque module est restreint au maximum :
    - utilisateur postfix
    - exécution dans une cage
  - files d'attente multiples
  - pas de programme SUID
  - l'architecture est difficile à casser

postfix : architecture

```
rond jaune : processus
carré bleu : fichier de configuration
carré jaune : file d'attente
```

(c) Wietse Venema



postfix : anti-spam / anti-relayage

- Mécanismes de sécurité anti-spam
  - Liste noire :
    - Client (adresse IP, RBL <http://www.mail-abuse.org/rbl/>, absence d'enregistrement inverse dans le DNS).
    - `HELO` (format, domaine)
    - `MAIL FROM:` (adresse, domaine, DNS).
  - Autres :
    - Utilisation d'expressions rationnelles dans les entêtes des messages

```
# postconf header_checks

header_checks = regexp:/etc/postfix/headreject

# cat headreject

/^Content-
   (Disposition|Type).*name=\"(document|file|body|data|text|test|doc|message|r
   eadme)(\.exe|\.pif|\.scr|\.zip|\.cmd)/   DISCARD Worm.SCO.A
```

    - et dans le corps des messages.
- Anti-relayage :
  - Vérification de l'adresse IP cliente ou du `RCPT TO:`

postfix : filtrage de contenu

- Possibilités de filtrage via l'interface content_filtering :
  - "Simple content filtering" (script de filtrage réexécutant la commande sendmail pour réinjecter le message "marqué")
  - "Advanced content filtering" (postfix exécute 2 démons smtpd, le marquage est effectué par un relais smtp tiers)
  - voir le fichier `FILTER_README` dans la distribution.
- Il est possible d'utiliser avec le filtrage de contenu avancé :
  - un relais SMTP anti-virus commercial
  - une « glue » qui communique en SMTP et utilise des outils tiers
    - ex : amavisd-new utilisant Mail::SpamAssassin et plusieurs anti-virus dont clamav
  - l'envoie en LMTP vers le relais filtrant permet d'avoir des résultats différents pour chaque destinataire
    - postfix peut alors générer lui même les avis de non distribution nécessaires
    - sinon c'est le relais filtrant qui doit les générer.

postfix : SMTP/TLS (postfix/tls)

- Le chiffrement se fait entre deux serveurs
- SMTP-TLS n'est pas une encapsulation de SMTP dans TLS :
  - Le serveur contacté émet l'annonce STARTTLS
  - Le client envoie la commande STARTTLS
  - Négociation TLS entre les deux parties
  - Session SMTP normale dans le flux TLS
  - Retour à la bannière (EHLO) à la fin de chaque mail
- Grâce à TLS, il est possible :
  - d'authentifier un utilisateur à partir d'un certificat client
  - d'imposer un certificat valide dans un réseau privé
  - de permettre le relayage depuis et vers l'Internet si le certificat client est valide
- Voir <http://www.hsc.fr/ressources/breves/postfix-tls.html>
  - comment patcher postfix et configurer la partie TLS.

amavisd-new

- <http://www.ijs.si/software/amavisd/>
- Démon en perl permettant d'appliquer un filtrage de contenu, anti-virus et anti-spam, à un flux SMTP (LMTP en entrée possible)
- Utilise le module perl Mail::SpamAssassin pour détecter les spams
- Sait utiliser de nombreux anti-virus, démons ou ligne de commande
- Développé pour :
  - limiter les risques de perte de mails
    - il ne prend jamais la responsabilité d'un mail
    - il ne modifie pas les messages :
      - ajoute un entête, met en quarantaine, rejette ou émet un avis de non délivrance.
  - optimiser les flux
    - peut traiter plusieurs messages simultanément
    - garde un cache des derniers résultats pour ne pas retraiter le même mail.

SpamAssassin

- Filtre en perl permettant de détecter les spams à partir d'un système de notations :
  - utilise de nombreux tests de types différents, chacun possédant un certain score
  - les scores sont calculés pour maximiser le taux de détection (>>95%) tout en minimisant les risques de faux positifs (<<0,1%)
- Logiciel libre de qualité professionnelle :
  - McAfee SpamKiller Technology « Powered by McAfee SpamAssassin »
- Peut être utilisé
  - par un utilisateur final depuis procmail ou via un relais pop3
  - dans un relais SMTP via le module Mail::SpamAssassin
- ATTENTION : le filtrage anti-spam prend beaucoup plus de ressources que le filtrage anti-virus :
  - mémoire, temps réel et temps utilisateur.

DNSBL

- Une DNSBL est une Black List DNS :
  - Toutes les adresses IP que le serveur connaît sont suspectes
- Les natures des adresses IP peuvent différer :
  - Adresses IP d'où des spams ont été envoyés
  - Adresses IP de serveurs relais ouverts sur Internet
  - Adresses IP de clients de FAI sur des plages d'adresses dynamiques
  - Etc.
- Les politiques de gestion de ces listes diffèrent :
  - Modalités d'entrée/sortie, réactivité, etc.
- SpamAssassin et postfix peuvent tous les deux utiliser des DNSBL.
- Site de test : http://www.dnsstuff.com/
- Listes de DNSBL : http://www.moensted.dk/spam/ et http://www.declude.com/junkmail/support/ip4r.htm

Bases de spam

- Des bases de spams sont confectionnées de façon collaborative
  - Grâce à la collaboration de ses utilisateurs.
- Chaque (partie de) message reçu(e) est comparé(e) à une base centrale
  - Entre le « client » et le « serveur central » seuls des « hashs » sont échangés.
- Certains systèmes permettent de détecter des spams « mutants ».
- Certains systèmes utilisent un « niveau de confiance/spammicité ».
- De telles bases sont Razor, Pyzor et DCC.
  - SpamAssassin sait les utiliser toutes les trois.

clamav

- <http://www.clamav.net/>
- Anti-virus libre destiné à filtrer les messages électroniques
- Peut aussi être utilisé en ligne de commande pour scanner une arborescence
  - Sous linux un module noyau permet de scanner tout fichier lors de son ouverture et d'en interdire l'accès s'il est infecté.
- Un démon clamd permet d'optimiser les performances en n'initialisant le moteur qu'une seule fois.
- La priorité est portée sur la mise à jour des virus au fur et à mesure des nouvelles apparitions aidés par des ISP pour les détecter
  - mais la base est aussi complétée avec les anciens virus qui ne sont plus (ou peu) en activité.

Plan : Installation

- postfix/tls
- amavisd-new
- SpamAssassin
  - Razor
  - Pyzor
  - DCC
- clamav

postfix/tls (1/4)

- L'installation de postfix est aisée
- Le plus compliqué est d'appliquer le patch postfix_tls et de régénérer les fichiers Makefile :
  - créer les groupes `postdrop` et `postfix`
  - créer l'utilisateur `postfix` avec `postfix` comme groupe principal
    - lui associer ni shell ni répertoire principal

```
$ patch -p1 < ../pfixtls-<version>/pfixtls.diff
```

```
make makefiles CCARGS="-DUSE_SSL -I/usr/local/ssl/include" AUXLIBS="-
  L/usr/local /ssl/lib -lssl -lcrypto"
```

```
$ make
```

`# make install` ou `# make upgrade` en cas de mise à jour.

postfix/tls (2/4)

- /etc/postfix/master.cf : définition des paramètres des programmes.

```
# service type  private unpriv  chroot  wakeup  maxproc command + args
smtp      inet  n       -       y       -       -       smtpd
pickup    fifo  n       -       y       60      1       pickup
cleanup   unix  n       -       y       -       0       cleanup
qmgr      fifo  n       -       y       300     1       qmgr
rewrite   unix  -       -       y       -       -       trivial-rewrite
bounce    unix  -       -       y       -       0       bounce
defer     unix  -       -       y       -       0       bounce
flush     unix  n       -       y       1000?   0       flush
proxymap  unix  -       -       y       -       -       proxymap
smtp      unix  -       -       y       -       -       smtp
relay     unix  -       -       y       -       -       smtp
showq     unix  n       -       y       -       -       showq
error     unix  -       -       y       -       -       error
local     unix  -       n       n       -       -       local
virtual   unix  -       n       y       -       -       virtual
lmtp      unix  -       -       y       -       -       lmtp
tlsmgr    fifo  -       -       y       300     1       tlsmgr
```


postfix/tls (3/4)

- /etc/postfix/main.cf : configuration globale de Postfix

```
mydomain = hsc.fr
myhostname = groar.tlse.hsc.fr
mynetworks_style = host
relayhost = [itesec.hsc.fr]
smtpd_banner = $myhostname ESMTP $mail_name ($mail_version)
```

- Parties tls cliente et serveur

```
smtp_tls_CAfile = /etc/postfix/ca.crt
smtp_tls_cert_file = /etc/postfix/groar.crt
smtp_tls_key_file = /etc/postfix/groar.key
smtp_tls_loglevel = 1
smtp_tls_per_site = hash:/etc/postfix/tls_per_site
smtp_use_tls = yes
smtpd_tls_CAfile = /etc/postfix/ca.crt
smtpd_tls_cert_file = /etc/postfix/groar.crt
smtpd_tls_key_file = /etc/postfix/groar.key
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
smtpd_use_tls = yes
```

postfix/tls (4/4)

- Extension à la partie tls serveur
  - autorisation de relayage via vérification d'un certificat client

```
relay_clientcerts = hash:/etc/postfix/relay_clientcerts
smtpd_recipient_restrictions = ..., permit_tls_clientcerts, ...
smtpd_tls_ask_ccert = yes
```

  - `relay_clientcerts` permet de lister les empreintes des certificats autorisés

```
31:E7:60:6C:72:44:26:4F:72:2E:BA:0B:6C:5C:73:52 groar.tlse.hsc.fr
```

  - `tls_per_site` va permettre de forcer ou de désactiver tls pour un système

```
groar.tlse.hsc.fr       MUST
broken.domain.fr        NONE
```

- Mise en cage
  - voir les scripts dans `postfix-<version>/examples/chroot-setup/`
  - ATTENTION : la mise en cage ne sécurise pas
    - elle permet de minimiser les conséquences en cas de vulnérabilité
    - mais ne permet pas de se passer de la mise à jour en cas de vulnérabilité.

amavisd-new (1/7)

- créer un utilisateur `amavis` et son groupe principal `amavis`
- créer son répertoire principal `/var/amavis`
  - son répertoire temporaire `/var/amavis/tmp` et sa quarantaine `/var/virusmails`
- recopier le fichier `amavisd` dans `/usr/local/sbin`
  - et le fichier `amavisd.conf` dans `/etc`
- créer les alias `virusalert` et `spamalert`
- installer les modules perl dont dépend amavisd-new
  - utiliser CPAN :

```
# perl -MCPAN -e shell
cpan> o conf prerequisites_policy ask
cpan> i <module>
cpan> install <module>
```

amavisd-new (2/7)

- Modules perl à installer :

```
Archive::Tar    (Archive-Tar-x.xx)
Archive::Zip    (Archive-Zip-x.xx) (1.09 or later is recommended!)
Compress::Zlib (Compress-Zlib-x.xx)
Convert::TNEF   (Convert-TNEF-x.xx)
Convert::UUlib (Convert-UUlib-x.xxx)
MIME::Base64    (MIME-Base64-x.xx)
MIME::Parser    (MIME-Tools-x.xxxx)
Mail::Internet (MailTools-1.58 or later have workarounds for Perl 5.8.0 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) (use 1.49 or later, some older cause
   problems)
Unix::Syslog    (Unix-Syslog-x.xxx)
```

- Il est conseillé de régulièrement mettre à jour ces modules perl
  - de façon hebdomadaire !

amavisd-new (3/7)

- Programmes à installer :

```
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/
rar, unrar: http://www.rarsoft.com/, ftp://ftp.kiarchive.ru/pub/unix/arcers/
zoo:         ftp://ftp.kiarchive.ru/pub/unix/arcers/
cpio:        ftp://ftp.gnu.org/pub/gnu/cpio
lzop:        http://www.lzop.org/download/
freeze:      ftp://ftp.warwick.ac.uk/pub/compression/
```

- Il est conseillé de régulièrement mettre à jour ces programmes
  - de façon hebdomadaire !

amavisd-new (4/7)

- Configurer `/etc/amavisd.conf` :

```
$mydomain = 'hsc.fr';             # (no useful default)
$daemon_user  = 'amavis';         # (no default;  customary: vscan or amavis)
$daemon_group = 'amavis';         # (no default;  customary: vscan or amavis)
$TEMPBASE = "$MYHOME/tmp";        # prefer to keep home dir /var/amavis clean?
$inet_socket_port = 10024;        # accept SMTP on this local TCP port
$forward_method = 'smtp:127.0.0.1:10025';  # where to forward checked mail
$final_virus_destiny      = D_PASS;  # (defaults to D_BOUNCE)
$final_banned_destiny     = D_PASS;  # (defaults to D_BOUNCE)
$final_spam_destiny       = D_PASS;  # (defaults to D_REJECT)
$final_bad_header_destiny = D_PASS;  # (defaults to D_PASS), D_BOUNCE
   suggested
$QUARANTINEDIR = '';
$virus_lovers{lc('.')}= 1;
$spam_lovers{lc('.')} = 1;
```


amavisd-new (5/7)

- Configurer `/etc/amavisd.conf` :

```
$sa_local_tests_only = 0;             # (default: false)
$sa_tag_level_deflt  = -999.9; # 3.0; # add spam info headers if at, or above
   that level
$sa_tag2_level_deflt =    5.0; # 6.3; # add 'spam detected' headers at that
   level
$sa_kill_level_deflt =  999.9; # $sa_tag2_level_deflt; # triggers spam evasive
   actions
$sa_spam_subject_tag = '***SPAM*** '; # (defaults to undef, disables)
$sa_spam_modifies_subj = 1;    # may be a ref to a lookup table, default is
   true
### http://clamav.elektrapro.com/
['Clam Antivirus-clamd',
  \&ask_daemon, ["CONTSCAN {}\n", '/var/amavis/clamd'],
  qr/\bOK$/, qr/\bFOUND$/,
  qr/^.*?: (?!Infected Archive)(.*) FOUND$/ ],
# NOTE: run clamd under the same user as amavisd,
# match the socket name in clamav.conf to the socket name in this entry
```

amavisd-new (6/7)

- Intégrer amavisd-new dans postfix ainsi :
  - ajouter les lignes suivantes à `/etc/postfix/master.cf` :

```
smtp-amavis unix -      -       y/n     -       2  smtp
    -o smtp_data_done_timeout=1200
    -o disable_dns_lookups=yes

127.0.0.1:10025 inet n  -       y/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
    -o strict_rfc821_envelopes=yes
```


amavisd-new (7/7)

  - puis entrer les deux commandes suivantes :

```
# postconf -e 'content_filter = smtp-amavis:[127.0.0.1]:10024'
# postsuper -r ALL; postfix reload
```

- Procéder ensuite aux tests comme indiqué dans le fichier `README.postfix` inclus dans les sources d'amavisd-new.

- Possibilité d'utiliser LMTP de postfix à amavisd-new
  - pour cela changer la définition de smtp-amavis par :

```
smtp-amavis unix -      -       y     -       2  lmtp
    -o smtp_data_done_timeout=1200
```

- Durant tous ces tests, il est conseillé de positionner la variable `soft_bounce` à `yes` dans le fichier `/etc/postfix/main.cf`,
  - aucun message n'étant alors bouncé ou rejeté définitivement.

SpamAssassin (1/2)

- Installer les modules perl dont dépend SpamAssassin
  - utiliser CPAN

```
ExtUtils::MakeMaker >= 5.45 (from CPAN, or included in Perl 5.6 and higher)
File::Spec >= 0.8    (from CPAN, or included in Perl 5.6 and higher)
Pod::Usage >= 1.10   (from CPAN, or included in Perl 5.6 and higher)
HTML::Parser >= 3.24 (from CPAN)
Sys::Syslog          (from CPAN)
DB_File              (from CPAN)
Net::DNS             (from CPAN)
Mail::Audit, Mail::Internet, Net::SMTP      (from CPAN)
Digest::SHA1         (from CPAN)
Net::Ident           (from CPAN)
IO::Socket::SSL      (from CPAN)
```

- Il est conseillé de régulièrement mettre à jour ces modules perl
  - de façon hebdomadaire !

SpamAssassin (2/2)

- Installer le module SpamAssassin
  - également via CPAN

```
# perl -MCPAN -e shell
cpan> o conf prerequisites_policy ask
cpan> install Mail::SpamAssassin
cpan> quit
```

- Configuration de `/etc/mail/spamassassin/local.cf`
  - (dés)activations : dns, bayes, dcc, razor et pyzor

```
dns_available       yes
rewrite_subject     1
report_safe         0
bayes_auto_learn    0
use_bayes           0
rbl_timeout         5
use_dcc             0
use_razor2          0
use_pyzor           0
```

Razor

- Razor est une base de spams
- Son utilisation par SpamAssassin réclame l'installation du module agent : razor-agents-2.36
  - ATTENTION : pour qu'il soit utilisable par SpamAssassin, il faut lui appliquer après installation un patch livré avec SpamAssassin : `Razor2.patch`

```
cd /usr/{lib,share}/perl5/.../Razor2
patch -p0 < .../Razor2.patch
```

- La version 1 de razor, de moindre confiance dans son mode de fonctionnement, n'est plus supporté par SpamAssassin
- Pour utiliser razor, il suffit de
  - positionner la variable `use_razor2` à `1` dans le fichier `/etc/mail/spamassassin/local.cf`
  - et de redémarrer SpamAssassin.

Pyzor

- Pyzor est une implémentation OpenSource de Razor
- Il possède malgré tout son propre protocole client/serveur.
- Il est possible de s'installer son propre serveur
  - mais l'implémentation du dialogue entre serveurs n'est pas encore réalisée.
- Le mode de fonctionnement de Pyzor correspond à celui utilisé dans la version 1 de Razor.
- Pour utiliser pyzor, il suffit de
  - positionner la variable `use_pyzor` à `1` dans le fichier `/etc/mail/spamassassin/local.cf`
  - et de redémarrer SpamAssassin.

DCC

- DCC est un système de comptabilisation d'occurrences de sommes de contrôles des mails reçus
  - plus le nombre de destinataires est grand et plus les présomptions sont fortes.
- Les sommes de contrôle utilisées permettent de détecter les spams personnalisés / « mutants ».
- Il est possible de s'installer son propre serveur
  - pour l'utiliser de façon isolée
  - pour l'utiliser comme cache interrogeant les serveurs officiels
    - méthode conseillée au delà de 100.000 messages par jour.
- Pour utiliser dcc avec SpamAssassin il est préférable
  - de configurer et lancer le démon `/var/dcc/libexec/dccifd`
  - puis de rajouter ces deux lignes dans `/etc/mail/spamassassin/local.cf`

```
dcc_home /var/dcc
dcc_dccifd_path /var/dcc/dccifd
```


Clamav

- Compiler et installer clamav :

```
$ ./configure --sysconfdir=/etc --with-user=amavis --with --group=amavis
$ make
# make install
```

- Configurer `/etc/clamav.conf` :

```
#Example
LogFile /var/amavis/clamd.log
LogTime
LogSyslog
PidFile /var/amavis/clamd.pid
LocalSocket /var/amavis/clamd
FixStaleSocket
User amavis
ScanMail
```

- Mettre à jour la base de signatures en lançant la commande `freshclam`.

Démarrage / arrêt

- Les démarrages et arrêts doivent s'effectuer dans cet ordre :
  - Démarrage

```
clamd
amavisd start
postfix start
```

  - Arrêt

```
postfix stop
amavisd stop
kill -TERM /var/amavis/clamd.pid
```


Plan : Administration

- Côté client
- Clients et serveurs Windows
- Optimisations
- Mises à jour antivi-rales et anti-spam
- Statistiques

Côté client

- Voici un exemple de fichier `~/.procmailrc` :

```
:0:
* ^X-Amavis-Alert: INFECTED
Mail/infected

:0:
* ^X-Amavis-Alert: BANNED FILENAME
Mail/banned

:0:
* ^X-Spam-Level: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\
Mail/spam-almost-certainly

:0:
* ^X-Spam-Status: Yes
Mail/spam-probably
```


Clients et serveurs Windows

- Pour les clients Windows il est possible d'y utiliser un relais
  - POP3 : http://mcd.perlmonk.org/pop3proxy/
  - IMAP : http://sourceforge.net/projects/imapassassin
- mais requière d'installer SpamAssassin sous Windows :
  - http://www.openhandhome.com/howtosa.html
- Il existe aussi des programmes et plugins payants :
  - Bloomba et SAproxyPro : http://www.statalabs.com/
  - Plugin eudora : http://www.spamnix.com/
  - SpamKiller : http://www.nai.com/

Optimisations

- SpamAssassin
  - trusted_networks (exempté de DNSBL) (voir le journal de bind) :

```
trusted_networks  127/8 192.168/16 192.70.106/24 205.206.231.19
  205.206.231.27 205.206.231.26
#19.231.206.205.in-addr.arpa domain name pointer
  lists.securityfocus.com.
#27.231.206.205.in-addr.arpa domain name pointer
  outgoing3.securityfocus.com.
#26.231.206.205.in-addr.arpa domain name pointer
  outgoing2.securityfocus.com.
```

- amavisd-new
  - whitelist_sender (exempté de Mail::SpamAssassin) :

```
map { $whitelist_sender{lc($_)}=1 } (qw(
  cert-advisory-owner@cert.org
  [...]
  nobody@cert.org
  securityfocus.com
  ));
```


Mises à jour anti-virales

- Les deux systèmes de filtrage de contenus installés doivent être mis à jour régulièrement pour s'adapter aux nouveautés.
- Pour clamav, lancer la commande `freshclam`
  - soit en mode démon :

```
/usr/local/bin/freshclam --quiet -l /var/log/clam-update.log --http-
  proxy=10.12.14.16:8080 --daemon-notify -d -c 8
```

  - soit depuis la crontab de l'utilisateur amavis :

```
47 1,4,7,10,13,16,19,22 * * * /usr/local/bin/freshclam --quiet -l
  /var/log/clam-update.log --http-proxy=10.12.14.16:8080 --daemon-notify
```

Mises à jour anti-spam

- Pour SpamAssassin l'utilisation des DNSBL et des bases de spams permet de faire contrepoids aux bases de règles statiques.
- Il est tout de même possible d'utiliser des bases de règles générées par d'autres personnes
  - le script `my_rules_du_jour` permet de télécharger plusieurs listes de ce type :

    http://www.exit0.us/index.php/RulesDuJour
- ATTENTION : il est important de vérifier que des jeux de règles tiers utilisés ne génèrent pas de faux positifs
  - pour cela il faut essayer ces règles sur deux corpus de hams et spams, représentatifs des mails reçus et datant de moins de 6 mois
  - par exemples `chickenpox.cf` et `tripwire.cf` peuvent être la cause de faux positifs sur de longs mails en français
  - les raisons des faux positifs avec l'utilisation de nouvelles règles sont :
    - que les scores du nouvel ensemble de tests n'ont pas été recalculés en incluant les nouvelles règles
    - les corpus de hams utilisés pour tester les nouvelles règles sont souvent non significatifs des hams reçus par des tiers.

Statistiques

- Les statistiques générées à partir des journaux permettent souvent de détecter des problèmes ou de voir les conséquences de modifications.
- Plusieurs outils existent, tous utilisant rrdtool

  http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/ :
  - amavis-stats : http://rekudos.net/amavis-stats/
    - comptabilise pour 1 jour / semaine / mois / an les messages passés, infectés et les spams
  - mailgraph : http://people.ee.ethz.ch/~dws/software/mailgraph/
    - comptabilise pour 1 jour / semaine / mois / an les messages envoyés et reçus, les rejets, les `bounce`, les virus et les spams
  - queuegraph : http://www.stahl.bau.tu-bs.de/~hildeb/postfix/queuegraph/
    - comptabilise pour 1 jour / semaine / mois / an les messages dans la queue `deferred` et dans les autres queue.

Conclusion

- L'architecture ainsi construite permet d'effectuer un filtrage de contenu sur les mails reçus.
- Ainsi installé amavisd-new :
  - ajoute un marquage sur tous les messages avec un virus
    - et envoie un avis à l'administrateur
  - ajoute un marquage à tous les spams destinés au domaine local
- Il est fait confiance à l'utilisateur pour filtrer ses messages
  - suivant les marques ajoutées par amavisd-new
- Il sera intéressant lorsque le système sera testé et optimisé :
  - de mettre en quarantaine les virus
  - de rajouter l'envoie d'un avis de mise en quarantaine aux destinataires dans le domaine local
  - de mettre en quarantaine les « gros » spams

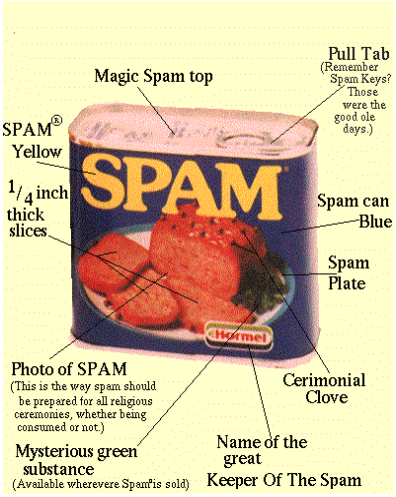
Références

- Car rien ne remplacera une personne bien informée :
- `http://www.admi.net/cgi-bin/wiki?Lutte_Contre_Le_Spam`
  - avec des actualités,
  - des liens et ressources,
  - des initiatives professionnelles et
  - des organismes de lutte contre le spamming
- `http://caspam.org/`

  CASPAM - Collectif Anti Spam
  - avec « 6 choses a faire ou ne pas faire pour lutter contre le spam !! »
- `http://www.cnil.fr/thematic/internet/spam/spam_sommaire.htm`

  CNIL - Halte au spam
  - avec « comment se prémunir ? »
- `http://www.hoaxbuster.com/`

  Première ressource francophone sur les hoax
  - les derniers canulars circulant sur le réseau.



`http://www.physics.upenn.edu/~pcn/spam.gif`

Merci de votre écoute

N'hésitez pas à poser vos questions...

et à venir visiter notre site WEB : http://www.hsc.fr/

pour y lire nos autres présentations sur le même sujet :

- Éléments de réflexion sur le spam

  http://www.hsc.fr/ressources/presentations/ddmspam/

- Spamassassin

  http://www.hsc.fr/ressources/presentations/spamassassin/

et sur bien d'autres sujets de sécurité...