



- [Accueil](#)
- [A propos](#)
- [Nuage de Tags](#)
- [Contribuer](#)
- [Who's who](#)

Récoltez l'actu UNIX et cultivez vos connaissances de l'Open Source

06 sept 2008

Clamav, l'antivirus qui vient du froid

Catégorie : [Sécurité](#) Tags : [misc](#)



~~Retrouvez cet article dans :~~ [Misc 17](#)

Quiconque, utilisateur ou administrateur, ignorerait encore la capacité de nuisance des virus et vers informatiques s'expose à de graves et douloureuses déconvenues 1.

Les pays de l'Est avaient plutôt la réputation, jusqu'à une époque encore récente, de fournir des virus plutôt que les armes pour les combattre. Le projet ClamAV, initié et dirigé par Tomasz Kojm (Pologne) depuis 2002, est une bonne exception à cette « règle ». Il rejoint aussi la très petite famille des antivirus développés et distribués sous licence GPL.

Cet article a pour objectif de présenter ClamAV sous un angle pratique. Nous renvoyons le lecteur intéressé par les fondements théoriques de la virologie et de la lutte antivirus aux - excellents - ouvrages 2 et 3 ainsi qu'aux articles parus dans MISC 4.

1. Présentation

1.1 Généralités

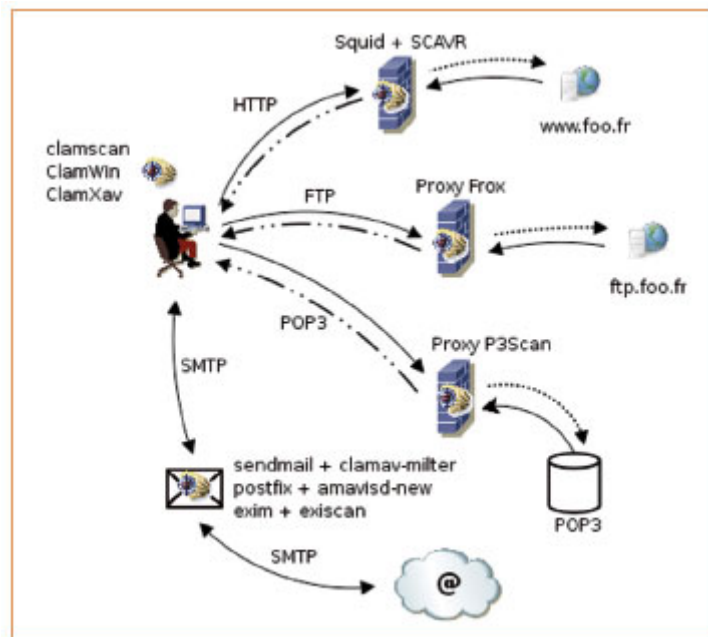
ClamAV est une boîte à outils pour construire une solution antivirus. Le code source disponible sur le site du projet⁵ permet de compiler un moteur

d'analyse, un démon et quelques programmes annexes dont un utilitaire de gestion des mises à jour. Cela peut sembler peu de choses, mais c'est suffisant pour protéger efficacement son poste de travail, sa messagerie électronique ou même sa connexion Internet, comme nous le verrons au travers de quelques exemples. (cf. schéma 1 ci-contre)

Avant cela, attardons-nous un bref instant sur les fonctionnalités offertes par ClamAV actuellement dans sa version 0.80.

1.2 Fonctionnalités

Le moteur fourni par ClamAV fonctionne par recherche d'empreintes ou signatures dans les fichiers analysés. C'est une méthode certes ancienne, mais qui a fait ses preuves et reste encore la plus couramment utilisée. Qui dit « empreintes » et « signatures » dit également « mises à jour ». ClamAV fournit donc un utilitaire de gestion des bases de signatures :



Intégration de ClamAV dans les flux réseau

Freshclam. La notion de boîte à outils évoquée précédemment se traduit par la mise à disposition d'une librairie associée à une API qui permet de « clamaviser » des applications tierces.

Des mécanismes de protection internes ont été ajoutés au fil des ans pour éviter les désagréments liés, par exemple, à l'analyse d'archives piégées (archive bombs).

1.2.1 Un petit mot sur les mises à jour

Les mises à jour des signatures virales sont cruciales et présentent deux dangers :

- absence de signature d'un virus ;
- corruption - volontaire ou non - des bases de signatures.

Freshclam est l'utilitaire de mise à jour des bases ClamAV. Utilisé en ligne de commande ou lancé en tâche de fond sous la forme d'un démon, il interroge régulièrement les serveurs de signatures ClamAV afin de maintenir à jour les bases. L'intégrité des bases est validée à l'aide de mécanismes cryptographiques (en gros à l'aide de signatures numériques). Cette dernière fonctionnalité n'est possible que si Freshclam a été compilé avec les bibliothèques MP 3, mais la mise à jour reste possible même en leur absence.

La procédure de mise à jour est fondée sur l'interrogation du DNS : la version de la base courante est extraite d'un enregistrement TXT et le serveur sur lequel Freshclam doit se connecter pour la télécharger est choisi par round-robin parmi les miroirs associés à l'enregistrement ~~database.clamav.net~~. En matière de détection de nouveaux virus, ClamAV se défend plutôt bien et a récemment été le premier à fournir les signatures pour des virus comme SoBig (variante I).

1.3 Installation

Avant de se lancer tête baissée dans l'installation en tant que telle de ClamAV, il est fortement recommandé de créer un compte système dédié, en général intitulé ~~clamav~~, ainsi qu'un groupe éponyme. ClamAV a tendance à légèrement couiner si ces deux éléments sont absents ; cela oblige à exécuter et faire tourner sous l'identité du super-utilisateur ~~root~~ les programmes ClamAV, pratique mal vue dans le monde de la sécurité.

Autre point important à valider avant la compilation : la présence de quelques programmes de décompression. Les virus ont en effet une fâcheuse tendance à se cacher dans des archives. La bibliothèque LibClamAV prend en charge les formats de compression et d'archivage les plus populaires : zip, RAR, gzip, bzip2 pour ne citer que les principaux. Encore faut-il que les binaires et bibliothèques associées zlib et bzip2 soient présents lors de la compilation. Une fois ces conditions remplies, ClamAV est capable d'aller chercher un virus dans des archives récursives (fichier compressé plusieurs fois) même si plusieurs formats de compression ont été utilisés pour cacher la bestiole (par exemple un fichier ZIP recompressé par BZIP, le tout dans une archive TAR).

Dernier pré-requis optionnel mais fortement recommandé : GNU MP 3.6. Cette bibliothèque est utilisée lors des mises à jour des signatures pour en vérifier

l'authenticité. Notez que les mises à jour sont possibles sans MP, mais génèrent un message d'avertissement à chaque chargement, ce qui s'avère très vite pénible.

L'installation de ClamAV obéit ensuite à la sacro-sainte trinité :

```
configure && make && make install
```

Sauf avis contraire (option `--prefix`), les fichiers sont installés sous ~~/usr/local~~. ClamAV est également disponible dans les ports FreeBSD ou en package Debian.

1.3 Paquetages binaires

Pour les adeptes du « 100% graphique », une version binaire existe pour les environnements MS Windows 7 et Mac OS X 8. À noter concernant Mac OS X que ClamAV s'y compile aussi à partir des sources.

1.4 Configuration

Les paramètres de configuration du démon Clamd sont stockés dans le fichier ~~clamd.conf~~. L'édition de ce fichier est nécessaire, ne serait-ce que pour y commenter le mot ~~Example~~ qui se trouve dans les toutes premières lignes du fichier. Le démon ne peut fonctionner sans cela (ne me demandez pas pourquoi...).

Que les impatients se rassurent, les autres paramètres peuvent conserver - pour le moment - leurs valeurs par défaut.

1.5 ClamAV en action

Dernier point : par défaut, ClamAV ne fait qu'informer l'utilisateur qu'un fichier analysé est porteur d'un virus. Quand un virus est détecté par un des utilitaires ClamAV, trois actions sont possibles :

1. par défaut, ClamAV informe l'utilisateur de la présence du virus et en fournit le nom ;
2. le fichier peut être mis en quarantaine ;
3. le fichier peut être tout bonnement détruit.

La désinfection n'est pas une fonctionnalité offerte par ClamAV, qui ne sait mettre en œuvre que l'option « Nettoyage par le vide ». La mise en quarantaine permet alors d'isoler les fichiers infectés pour décider ensuite de la nécessité ou de la possibilité de les traiter. Encore faut-il que l'utilisateur ait

ou prenne le temps de consulter le contenu du répertoire de quarantaine. Le choix plus nihiliste d'effacer tout fichier infecté est peut-être celui d'une certaine prudence radicale.

2. ClamAV dans la pratique

Passons maintenant aux choses sérieuses et voyons quelques emplois possibles de ClamAV, seul ou associé à d'autres applications.

2.1. La protection du poste de travail

Première utilisation de ClamAV après sa compilation, assurer la protection du poste de travail.

Deux modes d'utilisation sont alors envisageables :

- l'analyse a posteriori, à la demande ou à fréquences fixes de fichiers ;
- l'analyse à la volée (on-access).

2.1.1 Analyse a posteriori

C'est la méthode la plus simple(tte) d'utilisation de ClamAV. Elle consiste à lancer la commande ~~clamscan~~ à la main ou par l'intermédiaire d'un ordonnanceur comme ~~cron~~ en lui passant en paramètre un répertoire ou un fichier.

Lancée sans autre paramètre que celui du point de départ de l'analyse, cette commande se contente d'afficher à l'écran le résultat de l'analyse.

Si on désire analyser toute une arborescence (son ~~home directory~~ par exemple), l'option ~~-r~~ doit être utilisée.

```
$ clamscan -r /home/yom
../bacula.rtf: OK
../cr-sstic-04.ppt: OK
../Price-2.exe: Worm.Bagle.AT FOUND
../Price.exe: Worm.Bagle.AT FOUND
../Price.tar.bz2: Worm.Bagle.AT FOUND

----- SCAN SUMMARY -----
Known viruses: 28151
Scanned directories: 1
Scanned files: 5
Infected files: 3
Data scanned: 0.13 MB
I/O buffer size: 131072 bytes
Time: 4.810 sec (0 m 4 s)
```

Pour une utilisation sous ~~erontab~~ ou en tâche de fond, l'affichage peut être moins volumineux et le résultat des analyses journalisé par l'intermédiaire de ~~syslog~~ ou dans un fichier à part. La mise en quarantaine des fichiers est déclenchée par l'option ~~--move~~, leur destruction par ~~--remove~~.

2.1.2 De l'intérêt du démon

Le « tout à la main » a ses avantages, mais aussi (et surtout ?) des inconvénients :

- lorsque la commande ~~clamscan~~ est lancée, les bases de signatures sont chargées en mémoire avant l'analyse à proprement parler. Tout comme le refroidissement du canon, cela peut prendre un certain temps.
- les bases de signatures utilisées sont celles présentes sur disque au lancement de Clamscan. Si la dernière mise à jour remonte à quelques jours, elles peuvent être obsolètes.

L'utilisation des démons ~~Clamd~~ et Freshclam remédie à ces deux défauts. Lancé par la commande clamd, le démon ClamAV charge en mémoire les signatures présentes sur le disque. Le démon Freshclam, quant à lui, s'occupe de leur mise à jour à intervalles réguliers.

L'utilitaire Clamscan permet de tester le fonctionnement du démon, sa syntaxe est très proche de celle du programme Clamscan. Ainsi, si l'on reprend l'exemple précédent :

```
$ clamscan -r .
../bacula.rtf: OK
../cr-sstic-04.ppt: OK
../Price-2.exe: Worm.Bagle.AT FOUND
../Price.exe: Worm.Bagle.AT FOUND
../Price.tar.bz2: Worm.Bagle.AT FOUND

----- SCAN SUMMARY -----
Known viruses: 28151
Scanned directories: 1
Scanned files: 5
Infected files: 3
Data scanned: 0.13 MB
I/O buffer size: 131072 bytes
Time: 3.123 sec (0 m 3 s)
```

On note une substantielle économie de temps d'analyse. Certes, dans cet exemple, le gain est d'une seconde, ce qui peut paraître peu, mais si l'on se place dans un contexte différent, disons l'analyse de courriels, cette petite seconde est à multiplier par le nombre de courriers reçus ou envoyés quotidiennement...

2.1.3 Analyse à la volée (on access scanning)

Clamd n'analyse les fichiers qu'après leur écriture sur disque. L'analyse à la volée permet la recherche de virus avant leur stockage. Sous Linux, cette fonctionnalité dépend d'un module du noyau : Dazuko 10.

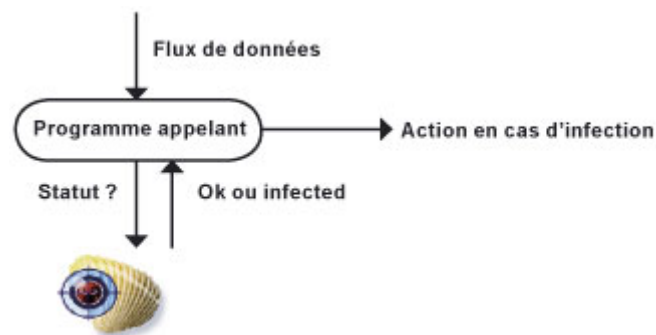
Une fois ce module compilé et installé, il autorise les actions d'analyse à l'ouverture (`ON_OPEN`), à la fermeture (`ON_CLOSE`) et à l'exécution (`ON_EXEC`). L'installation du module Dazuko sous Linux et FreeBSD est décrite en 11.

Le comportement du module est conditionné aux paramètres `Clamuko` du fichier `clamd.conf`. Son action sur un fichier infecté se traduit par un accès impossible à son contenu.

3. Construire une passerelle antivirus avec ClamAV

Au-delà de la protection égocentrique du poste personnel, ClamAV est utilisé pour construire des passerelles antivirus. Il s'agit de serveurs dédiés à la protection antivirus et agissant un peu comme des serveurs mandataires (proxy) ou bien d'instances de ClamAV installées sur des serveurs existants et leur apportant les fonctionnalités décrites précédemment.

Les cas les plus couramment rencontrés concernent les serveurs de messagerie (SMTP mais aussi POP3) et les mandataires HTTP.



Appel à ClamAV par un programme externe

D'une manière générale, l'appel au moteur ClamAV par des programmes externes suit le schéma 2 ci-contre. Noter que c'est au programme appelant qu'il revient de prendre une décision en cas de fichier infecté.

3.1. Antivirus SMTP

Le courrier électronique reste encore le principal vecteur de virus et autres bestioles affiliées. D'une manière générale, quand ClamAV est utilisé avec un serveur SMTP, il se contente d'indiquer à celui-ci le statut d'une pièce jointe analysée : infectée ou non. C'est ensuite au MTA de décider ce qu'il fait du « bébé ». C'est également à lui qu'il revient d'informer expéditeurs et destinataires. Une règle de bonne conduite comme de bon sens veut que l'on n'informe que le destinataire et non l'expéditeur. De toute façon, dans 99% des cas, l'adresse de l'expéditeur est usurpée ou tout simplement bidon.

3.1.1 ClamAV et Sendmail

À tout seigneur tout honneur, commençons par voir comme ClamAV peut agir en tant que filtre (milter) pour Sendmail. Il faut pour cela que ce dernier ait été compilé avec l'option idoïne.

Lors de la compilation de ClamAV, il est nécessaire de positionner l'option ~~--enable-milter~~ lors de la phase configure. On peut également désactiver le support de Clamuko pour éviter toute éventuelle interaction :

```
$ configure --enable-milter --disable-clamuko
```

Après compilation, vous obtiendrez un nouveau démon nommé Clamav-milter. Il fera l'interface entre Sendmail et le moteur d'analyse ClamAV. Pour cela, il vous faut informer Sendmail de la présence de ce nouveau filtre. Si vous utilisez les macros m4, ajoutez à votre ~~sendmail.mc~~ les lignes suivantes :

```
INPUT_MAIL_FILTER(`clmilter',`S=local:/var/run/clmilter.sock, F=,T=S:4m;R:4m')dnl
define(`confINPUT_MAIL_FILTERS',`clmilter')
```

Vérifiez ensuite que le fichier de configuration clamd.conf comporte :

```
LocalSocket /var/run/clamd.sock
```

Lancez Clamav-milter :

```
/usr/local/sbin/clamav-milter -l -o -q var/run/clmilter.sock
```

Les options ~~-e~~ et ~~-l~~ activent l'analyse des courriers entrants et sortants ainsi que ceux émis depuis le réseau local. Pour la tranquillité des administrateurs, l'option ~~-q~~ désactive l'envoi de notifications à Postmaster... Il ne vous reste plus alors qu'à relancer Sendmail et le tour est joué.

3.1.2 ClamAV, Amavisd-new et Postfix

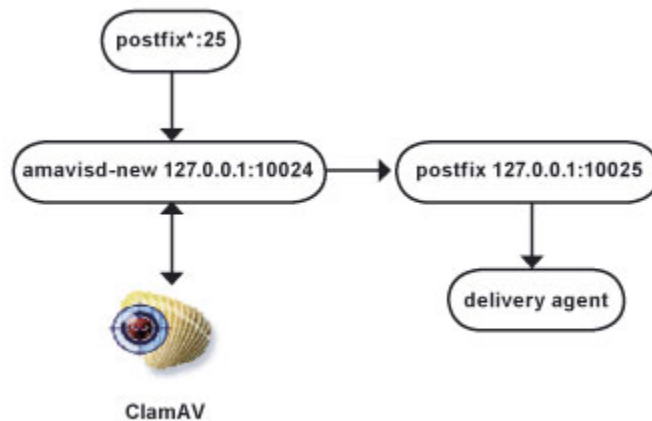
Un MTA sous PostFix peut être adossé à ClamAV par l'intermédiaire d'Amavisd-new, qui est une interface écrite en PERL pour permettre à un MTA d'effectuer des contrôles antispams et antivirus sur le contenu des courriers à

l'aide d'applications externes : typiquement SpamAssassin et ClamAV.

Le schéma 3 ci-après illustre les interactions entre les trois composantes PostFix, Amavisd-new et ClamAV.

Vous n'avez pas à toucher au fichier de configuration de ClamAV : pour rappel, ClamAV va se contenter de répondre ~~OK~~ ou ~~INFECTED~~ aux analyses initiées par Amavisd-new. Il reste néanmoins à configurer PostFix et Amavisd-new.

Si ces composants sont installés sur la même machine, les courriers entrants et sortants sont analysés. S'il est nécessaire de les traiter différemment, il faut lancer deux instances différentes de PostFix en utilisant des alias IP ou utiliser deux serveurs distincts, l'un pour le courrier entrant, l'autre pour le courrier sortant.



PostFix, Amavisd-new et ClamAV

3.1.2.1 PostFix

Le fichier ~~master.cf~~ doit comporter les lignes suivantes :

```

smtp-amavis unix -      n      -      3      smtp
  -o smtp_data_done_timeout=1200
  -o disable_dns_lookups=yes
127.0.0.1:10025 inet 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

```

et le fichier ~~main.cf~~ celle-ci :

```
content_filter = smtp-amavis:[127.0.0.1]:10024
```

Relancer PostFix pour prendre en compte ces paramètres.

3.1.2.2 Amavisd-new

Le fichier ~~amavisd.conf~~ fourni dans l'archive d'Amavisd-new contient plusieurs exemples d'appels à des logiciels antivirus. La section VII leur est en quelque sorte réservée. Vous y trouverez plusieurs pavés proposant des configurations prêtes à l'emploi (ou presque) pour différents moteurs d'analyse, parmi lesquels ClamAV :

```
# ### 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
```

Comme le dit judicieusement la note, arrangez-vous pour qu'Amavisd-new et ClamAV s'exécutent sous la même identité et tout ira pour le mieux. Les trois « briques » de notre Lego(c) sont maintenant assemblées, la solution est fonctionnelle.

3.1.3 ClamAV, Exim et Exiscan

Autre alternative à Sendmail et PostFix : le MTA Exim peut lui aussi tirer parti de la présence de ClamAV. Il vous faudra pour cela patcher les sources ou bien récupérer une version d'Exim modifiée¹². Une fois cette condition remplie, l'interfaçage avec ClamAV se fait en ajoutant les lignes suivantes dans le fichier de configuration d'Exim (~~exim.conf~~ ou ~~exim4.conf~~ selon le mode d'installation) :

```
# MAIN CONFIGURATION SETTINGS #
av_scanner = clamd:/var/run/clamav/clamd

....

check_message:
require verify = header_sender
##### EXISCAN ACL #####
deny message = Ce message est invalide ($demime_reason)
demime = *
condition = ${if >{$demime_errorlevel}{2}{1}{0}}
deny message = Ce message contient un virus ($malware_name)
demime = *
malware = *
deny message = Ce message contient un fichier illicite (.$found_extension)
```

```
demime = bat:com:pif:prf:scr:vbs
warn message = X-Antivirus-Scanner: Clean mail though you should still
use an Antivirus
##### EXISCAN ACL #####
accept
```

Relancer Exim rend la solution opérationnelle.

Concernant les flux SMTP analysés, le principe de fonctionnement est le même que celui de PostFix : s'il est nécessaire de distinguer les courriers sortants des courriers entrants, il faut lancer deux instances d'Exim ou utiliser deux serveurs.

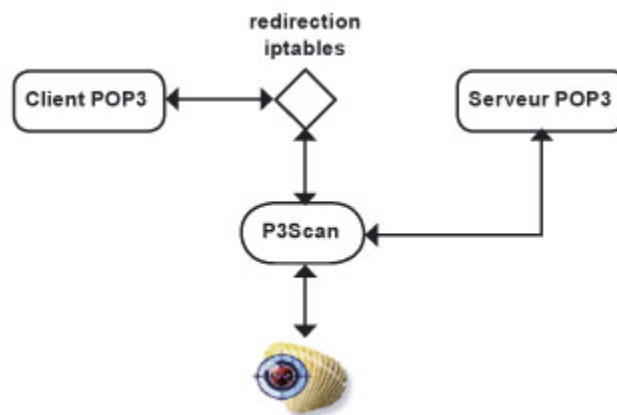
3.2. Antivirus POP3

Dans le monde de la sécurité, on a coutume de dire que seuls les paranoïaques survivront. Si vous avez des doutes sur votre passerelle antivirus SMTP, rien ne vous interdit de la renforcer par une solution de filtrage POP3. On ne sait jamais : une mise à jour des signatures peut avoir eu lieu entre le dépôt d'un courrier dans votre boîte aux lettres et le moment où vous en relèverez le contenu, et un nouveau virus peut être passé à travers les mailles du filtre SMTP.

P3Scan est un mandataire POP3 transparent. Sa mise en œuvre repose sur une redirection via ~~iptables~~ des flux POP3 vers le démon P3Scan qui fait appel à ClamAV pour déterminer le statut des fichiers analysés. La redirection des flux POP3 vers le port d'écoute de P3Scan se fait à l'aide de la commande (version allégée) suivante :

```
# iptables -t nat -A PREROUTING -p tcp -i eth0 --dport pop3 -j REDIRECT --to 8110
```

P3Scan s'appuie sur la fonction ScanMail de ClamAV. Le fichier ~~p3scan.conf~~ contient un bloc d'instructions relatives aux appels à ClamAV ainsi qu'aux actions à entreprendre en cas de détection d'un fichier infecté.



Principe de fonctionnement du proxy P3scan

3.3. Antivirus HTTP

À première vue, il peut sembler inutile de rechercher des virus dans les pages Web. A première vue seulement. Des cas récents ont montré que le protocole HTTP devient un vecteur de choix pour les infections virales et l'utilisation croissante de webmail rend nécessaire l'adjonction d'un antivirus pour flux HTTP aux protections plus classiques des flux SMTP.

3.3.1 SCAVR

Notre solution de filtrage HTTP s'appuie sur un serveur Squid et l'utilitaire SCAVR (SquidClamAVRedirector). SCAVR 13 est un script Python (je sais, je sais...) configuré comme « redirecteur » pour Squid. L'installation de Pyclamav 14, qui permet d'appeler la librairie LibClamAV depuis des scripts Python, est nécessaire. La mise en œuvre de SCAVR est alors simple. Elle repose sur les lignes suivantes à ajouter au fichier de configuration du serveur Squid:

```
redirect_program /usr/local/bin/SquidClamAV_Rdirector.py -c /etc/squid/SquidClamAV_Rdirector.py
redirect_children 5
```

Le fichier SquidClamAV_Rdirector.conf conditionne le fonctionnement de SCAVR :

```
[SquidClamAV]
# URL affichée en cas d'infection
virusurl = http://mon.serveur.com/GetInfected.cgi
cleancache = 300
ForceProtocol = http
MaxRequestsize = 2Mb
log_priority = LOG_INFO
log_facility = LOG_LOCAL6
acceptredirects = 300 301 302 303

[Extensions]
# Quels fichiers doit-on analyser ?
pattern = all .jpg .exe .zip .rar .ar .com .bzip .gz
[Proxy]
http = http://proxy.serveur.com:3128/
http2 =http://proxy2.serveur.com:3128/

[Whitelist]#
```

Y a t-il des sites à qui l'on peut faire confiance ?

3.3.2 mod_clamav

~~mod_clamav~~ est une alternative à SCAVR. Dans ce cas, c'est un serveur Apache qui est utilisé comme serveur mandataire. ~~mod_clamav~~ est un module développé pour la branche 2 d'Apache.

Une fois installé, la mise en œuvre de mod_clamav est relativement simple et passe par l'ajout de ces lignes au fichier de configuration d'Apache :

```
ClamavTmpdir    /tmp/clamav
```

```
ClamavDbdir     /usr/local/share/clamav
```

```
# Ligne originale
```

```
#ClamavSafetypes image/gif image/jpeg image/png
```

```
# Ligne d'après JPEG Exploit
```

```
ClamavSafetypes image/gif image/png
```

```
<Proxy *>
```

```
SetOutputFilter CLAMAV
```

Seules les images GIF et PNG seront épargnées par l'analyse... ce qui n'est pas forcément une bonne idée, mais c'est pour illustrer.

3.4. Antivirus FTP

Comme dirait l'autre, « on remet ça » avec le protocole FTP. Le principe ressemble comme deux gouttes d'eau à celui vu précédemment pour le protocole POP3 : on utilise un serveur mandataire transparent, Frox 15 pour ne pas le nommer, vers lequel on redirige les flux FTP.

Pour compiler Frox dans l'optique de construire une passerelle antivirus FTP, il est nécessaire de passer l'option ~~--enable-virus-scan~~ lors de la phase initiale de l'installation :

```
$ ./configure --enable-virus-scan
```

Une fois Frox compilé, il faut ajouter la règle (allégée) ~~iptables~~ suivante pour rediriger vers Frox les flux FTP :

```
iptables -t nat -A PREROUTING -p tcp --dport 21 -j REDIRECT --to 2121
```

étant entendu que Frox est à l'écoute sur le port 2121. Les paramètres de configuration de Frox sont stockés dans le fichier ~~frox.conf~~.

Pour activer l'analyse des flux FTP, le fichier ~~frox.conf~~ doit comporter les lignes suivantes :

```
VirusScanner '/usr/local/bin/clamscan' "%s"
```

```
VSOK 0
```

```
VSProgressMsgs 30
```

Le paramètre VirusScanner indique quel utilitaire Frox doit appeler pour analyser les fichiers. Noter que Clamscan peut être utilisé si le démon Clamd ne tourne pas et qu'à l'inverse, ce dernier doit tourner si c'est le client

Clamscan qui est appelé par Frox. Le paramètre %s désigne le fichier à analyser.

VSOK désigne le code retourné par le moteur d'analyse quand un fichier n'est pas infecté, en l'occurrence 0. VSPProgressMsgs, enfin, provoque l'envoi de données vers le client toutes les n secondes (30 ici) afin d'éviter, lors du chargement de gros fichiers, les désagréments liés au dépassement des temps d'attente (timeout).

Frox connaît quelques limitations :

- seuls les fichiers téléchargés par les clients FTP sont analysés ;
- les fichiers infectés ne sont jamais envoyés au client, même si l'antivirus utilisé est capable de les nettoyer (pour rappel, ce n'est pas le cas avec ClamAV)
- côté utilisateur, il y aura un certain délai entre le lancement du chargement et sa fin, dû à l'appel au moteur antivirus.

4. En guise de conclusion

Nous n'avons abordé dans cet article qu'un nombre restreint des applications sachant tirer profit de ClamAV. Une liste presque exhaustive est disponible sur le site du projet, et gageons qu'elle est destinée à s'allonger.

Pour illustrer les performances que l'on peut obtenir d'une passerelle antivirus SMTP, citons deux exemples :

- un serveur mono-processeur de type PIII 800 MHz (autant dire une vieille brouette...) et 512 Mo de RAM sauront traiter sans s'essouffler quelques milliers de messages par jour avec PostFix et Amavisd-new ;
- chez un ISP, avec une architecture plus complexe, 9 serveurs sous Sendmail traitent quotidiennement 4 millions de messages pour 330.000 virus détectés. L'analyse est réalisée par Clamav-milter et le démon Clamd. En heure de pointe, 150 instances de Clamav-milter et 26 démons Clamd s'exécutent sur chaque serveur.
- SourceForge utilise ClamAV et SpamAssassin. Durant l'affaire SoBig, les volumes traités ont connu un pic de 20.000 messages infectés toutes les 10 minutes.

D'autres exemples sont disponibles sur le site du projet¹⁶.

Depuis 2003, ClamAV semble avoir acquis ses lettres de noblesse en termes de stabilité et de qualité. Pour preuve, quelques offres grand public de filtrage antivirus pour messagerie de certains fournisseurs d'accès Internet de premier plan sont construites au-dessus de serveurs ClamAV. Dans d'autres cas, ils sont utilisés en secours de solutions plus sérieuses (comprenez commerciales).

Quoi qu'il en soit, ClamAV comble un certain manque : les antivirus sous GPL


ne courent en effet pas les rues.

Liens:

- ClamAV : www.clamav.net

Retrouvez cet article dans : [Misc 17](#)

Posté par ([La rédaction](#)) | Signature : Guillaume Arcas, Stéphane Clodic |

Article paru dans 

Laissez une réponse

Vous devez avoir ouvert une [session](#) pour écrire un commentaire.

« [Précédent](#) [Aller au contenu](#) »

[Identifiez-vous](#)

[Inscription](#)

[S'abonner à UNIX Garden](#)

• Articles de 1ère page

- [Sortez du software !](#)
- [Les dénis de service](#)
- [La mort annoncée du WEP](#)
- [nikto Tests de serveurs HTTP](#)
- [duplicity Sauvegarde chiffrée](#)
- [knl](#)
- [ffmpeg](#)
- [tor](#)
- [Eyes candy, Debian aussi !](#)
- [Développez vos pilotes de périphériques USB](#)



[Actuellement en kiosque :](#)

• Il y a actuellement

- **769** articles/billets en ligne.

• Catégories

- - [Administration réseau](#)
 - [Administration système](#)
 - [Agenda-Interview](#)
 - [Audio-vidéo](#)
 - [Bureautique](#)
 - [Comprendre](#)
 - [Distribution](#)
 - [Embarqué](#)
 - [Environnement de bureau](#)
 - [Graphisme](#)
 - [Jeux](#)
 - [Matériel](#)
 - [News](#)
 - [Programmation](#)
 - [Réfléchir](#)
 - [Sécurité](#)
 - [Utilitaires](#)
 - [Web](#)

• Archives

- [septembre 2008](#)
- [août 2008](#)
- [juillet 2008](#)
- [juin 2008](#)
- [mai 2008](#)
- [avril 2008](#)
- [mars 2008](#)
- [février 2008](#)
- [janvier 2008](#)
- [décembre 2007](#)
- [novembre 2007](#)
- [février 2007](#)

• [GNU/Linux Magazine](#)

- [GLME, partenaire de l'évènement "Paris, capitale du Libre"](#)
- [GNU/Linux Magazine 108 - Septembre 2008 - Chez votre marchand de journaux](#)
- [Edito : GNU/Linux Magazine 108](#)
- [GNU/Linux Magazine HS 38 - Septembre/Octobre 2008 - Chez votre marchand de journaux](#)
- [Edito : GNU/Linux Magazine HS 38](#)

• [GNU/Linux Pratique](#)

- [Linux Pratique soutient la journée mondiale contre les brevets logiciels](#)
- [Linux Pratique, partenaire de l'évènement "Paris, capitale du Libre"](#)
- [Linux Pratique N°49 - Septembre/Octobre 2008 - Chez votre marchand de journaux](#)
- [Edito : Linux Pratique N°49](#)
- [À télécharger : Les fichiers du Cahier Web de Linux Pratique n°49](#)

• [MISC Magazine](#)

- [Misc 39 : Fuzzing - Injectez des données et trouvez les failles cachées - Septembre/Octobre 2008 - Chez votre marchand de journaux](#)
- [Edito : Misc 39](#)
- [MISC 39 - Communiqué de presse](#)

- [Salon Infosecurity & Storage expo - 19 et 20 novembre 2008.](#)
- [Misc 38 : Codes Malicieux, quoi de neuf ? - Juillet/Août 2008 - Chez votre marchand de journaux](#)

© 2007 - 2008 [UNIX Garden](#). Tous droits réservés .