https://clx.asso.fr/spip/?Lutter-efficacement-contre-les

Lutter efficacement contre les contenus SMTP dangereux (1)

- Documentations - Technique -

Date de mise en ligne : mercredi 25 février 2004

La vitesse de propagation des nouveaux virus (*Dumaru*, *MyDoom*, etc.) et l'augmentation de la quantité de pourriel [1] envoyée et reçue sur Internet rendent le filtrage de contenu SMTP de plus en plus critique. Cependant, les techniques qui étaient utilisées jusqu'à maintenant montrent leurs limites. Cet article étudie et explique comment contourner l'une d'elles, celle que j'ai nommée « la patate chaude ».

Le problème

Le problème des émails qui véhiculent des virus, du pourriel ou du contenu suspect est comparable à celui dit de « la patate chaude ». Une fois arrivé dans la file d'un serveur de messagerie et identifié comme indésirable, vous ne savez plus trop qu'en faire. Vous aimeriez bien vous en débarrasser mais ne savez pas qui prévenir dans ce cas :
– L'expéditeur ? En général, soit son adresse est falsifiée, soit elle n'existe pas. Dans le premier cas la pauvre victime dont l'adresse a été usurpée se trouve inondée de messages. Dans le second cas, la situation empire car votre MTA [2] doit prendre en charge un *double bounce* [3].
– Les destinataires ? Pour les mêmes raisons, cela est souvent inutile. Les virus récents se propagent seuls et non plus au sein de documents qu'un correspondant aurait pu vous envoyer et que vous attendiez.
– L'administrateur ? Pitié, non, le pauvre va crouler sous les dizaines (centaines ?) de notifications journalières !

Cet état peut conduire à des files de messages énormes et écrouler certains serveurs sous la charge (cela s'est vu lors de l'épidémie *Dumaru*).

Alors quoi ? Renvoyer le message au grand réceptacle universel des données inutiles et autre *junk* (`/dev/null`) sans prévenir personne ? Sûrement, si vous choisissez cette voie, l'éthique liée au transfert des messages sur Internet vous empêchera de dormir la nuit. Un message ne devrait pas être supprimé sans que personne ne soit au courant.

Donc, point de salut ?

La solution

Quel était le problème déjà ? Ah oui « Que faire des messages malveillants une fois qu'ils ont été identifiés comme tel par mon serveur de messagerie ? ». Et si la question n'était pas la bonne ? Essayons celle-ci : « Comment faire pour refuser de prendre en charge la responsabilité de ces messages ? ». C'est-à-dire, laissons donc dans l'embarras les MTA qui n'ont pas su filtrer ces messages (ou qui n'ont pas voulu le faire sciemment). Oui, laissons-les gérer les notifications, les files de messages saturées par les *Dumaru*, les *MyDoom* et leurs prochains petits frères. Nos ressources sont précieuses, économisons-les.

Mais comment faire ? Afin de rejeter le message (et donc la responsabilité de le transmettre), le contenu de celui-ci devra être contrôlé **pendant** la connexion avec le MTA qui nous le fait parvenir. Cependant la manière dont fonctionnent les logiciels qui sont utilisés pour vérifier le contenu des émail (par exemple Amavis [4] ou MailScanner) ne permet pas cela. Ni, d'ailleurs, les MTA. En effet, il faudrait pouvoir intervenir avant même que le message ne soit accepté par ce dernier.

Je ne dis pas qu'*Amavis* et *MailScanner* sont de mauvais outils (il fut un temps où je les ai encensés, les ayant déployés chacun d'eux plusieurs fois), je dis qu'ils ne sont plus adaptés. Je ne dis pas que *Postfix* est moins bien que *Sendmail* mais ce dernier va nous permettre de résoudre de manière élégante notre problème grâce à une nouvelle API appelée *Milter*.

Milter (pour « *Mail Filter* ») est une API spécifique à Sendmail qui permet de lui connecter des logiciels externes. Ceux-ci vont permettre d'agir sur le comportement de Sendmail lors des différentes étapes de la conversation SMTP (HELO, MAIL FROM, RCPT TO, DATA). Les réponses à ces commandes SMTP pouront être dictées par ces logiciels.

Cette API supporte également la modification des messages par ces programmes externes.

Ainsi l'idée est de prendre le temps de repérer le contenu malveillant pendant la communication SMTP et, le cas échéant, signifier au MTA d'origine que nous refusons de prendre en charge son message. Dans la logique du protocole SMTP cela signifie que c'est à lui qu'incombe la tâche embarrassante de choisir le devenir du message incriminé.

Il y a un autre argument qui va dans le sens de cette technique (je le précise car il m'a déjà été reproché que, finalement, avec cette configuration je relègue le travail de mon MTA aux autres). En effet, la manière dont est géré Internet en général montre qu'il est plus souvent efficace que les problèmes soient pris en charge au plus tôt et, si possible, par le serveur le plus proche de l'utilisateur. Ce qui sera bien le cas ici.

Vite, un exemple pour illustrer cette théorie

Vraiment, vraiment, je suis navré pour tous les fans de *Postix*, *Exim*, etc. mais, à ce jour et à ma connaissance, seul *Sendmail* dispose d'un tel mécanisme [5] (enfin, bon, ceux qui me connaissent ne seront pas dupes).

Voici donc un exemple d'utilisation d'un *milter* [6] fourni par le projet ClamAV. Cet exemple est basé sur la distribution *Debian* mais il devrait être immédiat à « porter » sous *RedHat*, pardon *Fedora*, ou autre.

1) Ajoutez les lignes suivantes au fichier `/etc/apt/sources.list` qui contient la liste des dépôts de logiciels *Debian* :

```
 # ClamAV
deb http://clamav.catt.com/ debian stable main
```

2) Exécutez les commandes :

```
 # dselect update
# apt-get install clamav clamav-daemon clamav-freshclam clamav-milter
```

pour mettre à jour la liste des logiciels connus et installer ceux qui nous sont nécessaires.

3) Ajoutez les lignes suivantes dans le fichier `/etc/mail/sendmail.mc`. Elles permettent de déclarer un filtre *milter* à Sendmail.

```
INPUT_MAIL_FILTER(clamav',S=local:/var/run/clamav-milter/clamav-milter.ctl, F=, T=S:4m;R:4m')dnl
define(confINPUT_MAIL_FILTERS',clamav')
```

4) L'un des défauts du paquetage *clamav-milter* est qu'il ne fournit pas de script de démarrage. Copiez donc le contenu du cadre qui suit dans le fichier `/etc/init.d/clamav-milter`.

```
#! /bin/sh

PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
DAEMON=/usr/sbin/clamav-milter
NAME=clamav-milter
DESC="ClamAV Milter"
OPTIONS="-q -f /var/run/$NAME/$NAME.ctl"

test -x $DAEMON || exit 0

set -e

case "$1" in
  start)
        echo -n "Starting $DESC: $NAME"
        start-stop-daemon --start --quiet --pidfile /var/run/$NAME/$NAME.pid \
                --chuid clamav --make-pidfile --exec $DAEMON -- $OPTIONS \
                2> /dev/null
        echo "."
        ;;
  stop)
        echo -n "Stopping $DESC: $NAME"
        killall $NAME
        rm -f /var/run/$NAME/$NAME.pid
        echo "."
        ;;
  restart|force-reload)
        echo -n "Restarting $DESC: $NAME"
        $0 stop
        sleep 1
        $0 start
        echo "."
        ;;
  *)
        N=/etc/init.d/$NAME
        echo "Usage: $N {start|stop|restart|force-reload}" >&2
        exit 1
        ;;
esac

exit 0
```

Puis exécutez les lignes de commande suivantes pour activer le service lors du démarrage du système (la seconde réalisera pour vous les liens dans les répertoires `/etc/rc?.d`) :

```
 # chmod 755 /etc/init.d/clamav-milter
# update-rc.d clamav-milter defaults 19 21
```

(« 19 » car le *milter* doit être démarré avant et arrêter après Sendmail.)

5) Ajoutez la ligne suivante dans le fichier `/etc/clamav/clamav.conf` afin d'activer la vérification des fichiers au format émail :

```
 ScanMail
```

6) Un second défaut du paquetage *clamav-milter* nous impose de créer nous-même le répertoire qui contiendra le tube nommé servant aux communications entre *Sendmail* et le *milter*. Pour cela, exécutez ces commandes :

```
 # mkdir /var/run/clamav-milter
# chown clamav.clamav /var/run/clamav-milter
# chmod 750 /var/run/clamav-milter
# adduser smmsp clamav
```

7) Enfin, les commandes suivantes permettent de régénérer le fichier de configuration de *Sendmail* (`sendmail.cf`) à partir du fichier de macros *m4* `sendmail.mc` et de démarrer Sendmail et le *milter* :

```
 # cd /etc/mail
# make
# /etc/init.d/sendmail stop
# /etc/init.d/clamav-milter start
# /etc/init.d/sendmail start
```

(Il faut également que le démon clamd soit démarré mais le paquetage clamav-daemon s'en est occupé.)

8) Testez que tout fonctionne correctement. Par exemple :

```
 seb@puck:~$ telnet gaia.anet.fr smtp
Trying 80.118.2.10...
Connected to gaia.anet.fr.
Escape character is '^]'.
220 gaia.anet.fr ESMTP Sendmail 8.12.3/8.12.3/Debian-6.6; Sat, 21 Feb 2004 18:36:25 +0100; (No UCE/UBE)
logging access from:
Mix-Rennes209-1-162.w193-250.abo.wanadoo.fr(OK)-Mix-Rennes209-1-162.w193-250.abo.wanadoo.fr
[193.250.30.162]
HELO puck.anet.fr
250 gaia.anet.fr Hello Mix-Rennes209-1-162.w193-250.abo.wanadoo.fr [193.250.30.162], pleased to meet you
MAIL FROM:<seb@anet.fr> 250 2.1.0 <seb@anet.fr>... Sender ok
RCPT TO:<seb@gaia.anet.fr> 250 2.1.5 <seb@gaia.anet.fr>... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
Subject: Test
From: badguy@verybadnet.org
To: poorguy@gaia.anet.fr

X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
.
550 5.7.1 Virus detected by ClamAV - http://clamav.elektrapro.com
QUIT
221 2.0.0 gaia.anet.fr closing connection
Connection closed by foreign host.
```

Le code de retour 550 retourné par notre MTA à la fin de la transmission du contenu du message [7] est un code d'erreur permanent qui indique que le message n'a pas été pris en charge par notre serveur. Ainsi il n'a aucune chance de venir encombrer la file des messages de celui-ci.

Un dernier point, n'hésitez pas à augmenter le nombre de tests journaliers réalisés par le démon *freshclam* qui vérifie l'existence des nouvelles définitions de virus. Cela peut faire la différence lors d'épidémies fulgurantes comme celles auxquelles nous avons pu assister fin janvier. En effet, à ce moment, *ClamAV* s'est montré plus efficace à les arrêter tout simplement parce que sa configuration par défaut lui a permis d'obtenir la définition des virus avant bien d'autres anti-virus.

Il faut pour cela augmenter un paramètre dans le fichier `/etc/clamav/clamav-freshclam-handledaemon.conf` (n'utilisez pas non plus une valeur trop élevée afin de ne pas surcharger les serveurs du projet *ClamAV*. Par exemple, avec la valeur suivante, une vérification sera réalisée toutes les trois heures environ (huit fois par jour) :

```
FRESHCLAM_CHECK_FREQUENCY=8
```

Le prochain article

La prochaine fois, je vous expliquerais comment étendre ce principe au filtrage des Spams ou des contenus jugés belliqueux.

Ressources

Une introduction à Milter (en anglais)

Une liste de projets contenant le mot « *milter* » sur Freshmeat

Post-scriptum :

Vous l'aurez deviné, cette technique est utilisée sur Gaia. Cela protÚge les listes de diffusion et apporte plus de fiabilité au service.

---

[1] Spam

[2] Mail Transfer Agent. Le démon chargé de faire transiter le courriel sur Internet, NDLR.

[3] Un *double bounce* est un message d'erreur qui ne peut être envoyé à son destinataire et qui revient lui-même en erreur au *postmaster*.

[4] Oui, bon, en fait l'un des rejetons d'*Amavis* appelé *amavisd-new* permet cela mais nous n'en parlerons pas ici car cela viendrait en porte-à-faux sur notre brillante démonstration.

[5] *Postifx* permet de refuser des messages lors de la connexion SMTP mais sur des critères très simples (expressions régulières, en particulier)

et certainement pas à partir de décisions que pourraient prendre des logiciels externes.

[6] Par extension, on nomme « *milter* » les programmes écrit pour s'interfacer avec Sendmail *via* l'API Milter.

[7] La chaîne des 68 caractères est issue du projet *EICAR - Anti-Virus test file* dont le but est de fournir une signature inoffensive reconnue par tous les anti-virus afin de réaliser des tests.