# Le filtrage de mail sur un serveur de messagerie avec j-chkmail

José Marcio Martins da Cruz
Ecole des Mines de Paris – Centre de Calcul
Jose.Marcio-Martins@ensmp.fr

Gladys Huberman
Ecole des Mines de Paris – Centre de Calcul
Gladys.Huberman@ensmp.fr

## Résumé

*Le spam et les virus sont actuellement les grands ennemis des administrateurs de serveurs de messagerie. Plusieurs solutions existent actuellement. Peu nombreuses sont les solutions françaises de filtrage de mail distribuées sous licence du type GPL.Cet article présente le système de filtrage pour des serveurs de messagerie développé par l'Ecole des Mines de Paris. Le but recherché de ce filtre est son utilisation dans des serveurs dont le niveau de trafic est important. Pour cela, il est peu consommateur de ressources (CPU, mémoire…). Ce filtre développé initialement pour usage interne est mis à la disposition de la communauté sous licence GPL.*

## Mots clefs

Filtrage de mail, virus, filtrage de spam, serveur de messagerie

## 1 Introduction – pourquoi filtrer les messages sur un serveur de messagerie

Il y a encore moins de dix ans, quand Internet était peu accessible au grand public, la messagerie était un cadeau des dieux. Les risques étaient minimes : les vers de messagerie n'existaient presque pas et la lutte contre le *spam*[1] consistait uniquement à s'assurer que son serveur de messagerie n'était pas un relais ouvert[2]. La seule victime potentielle était l'administrateur système. Avec l'apparition des vers de messagerie et du spam, ce «nouveau monde merveilleux » a vite perdu son enchantement.

Les virus sont des programmes qui utilisent la messagerie pour se propager par inondationrécursive : chaque fois qu'une cible est atteinte et que des dégâts sont causés, le virus s'envoie à tous les correspondants de la victime. Une fois déclenchée la propagation, même l'auteur du virus ne peut plus la contrôler.

Le spam, au contraire, a une diffusion ciblée (pas de récursivité) : les destinataires sont choisis par l'émetteur. Les dégâts sont soit d'ordre technique (à cause du volume), soit souvent aussi d'ordre moral, puisque nombreux sont les messages qui concernent des vraies escroqueries ou des sujets portant atteinte à la morale. Parfois, des messages importants peuvent se trouver noyés et oubliés dans une masse de messages inutiles, vue la proportion que prend ce type de communication.

Le serveur de messagerie, point de passage obligé pour tous les messages, est un emplacement privilégié pour la mise en place d'un filtre.

Cette communication est le résultat des travaux menés à l'Ecole des Mines de Paris pour la protection de notre propre serveur de messagerie. Ils ont débouché sur le développement d'un filtre de messagerie (j-chkmail) mis à la disposition des administrateurs de serveurs de messagerie avec une licence du type GPL [1].

L'idée initiale était le développement d'un filtre anti-virus, pour notre serveur de messagerie, utilisant peu de ressources de façon à ce que nous n'ayons pas à changer de machine. En effet, si la consommation de ressources du filtre est marginale par rapport aux besoins du serveur de messagerie lui-même, ce filtre devient aussi très intéressant pour les serveurs dont le trafic est bien plus important.

Le besoin de filtrage de spam s'est ajouté très rapidement au besoin de filtrage antiviral, suite à des attaques du type « dictionnaire », visant à obtenir la liste des utilisateurs valables.

---

[1] SPAM – Il n'y a pas de consensus sur la définition de du terme spam. Disons que c'est de la **publicité sauvage** : des messages non souhaités, souvent anonymes, présentant souvent un caractère publicitaire, sans aucun contrôle de la légalité de l'offre.

[2] Relais ouvert – serveur de messagerie acceptant de remettre (relayer) un message à n'importe quel destinataire, quelle que soit l'origine du message.

## 2 L'intégration d'un filtre sur un serveur de messagerie

La fonction principale d'un MTA (ou Mail Transport Agent, ou logiciel serveur de messagerie) est le routage des messages. Il reçoit des messages en provenance des MUA (Mail User Agent, ou logiciel client de messagerie) locaux ou d'autres MTA et décode la destination. S'il s'agit d'une destination locale, il fait le nécessaire pour que le message soit transmis localement, sinon il retransmet le message vers le MTA du domaine de l'adresse destination.

Des fonctions annexes peuvent s'ajouter au routage des messages : ce sont, pour la plupart, des fonctions de sécurité, telles le filtrage anti-spam ou l'authentification par certificats, le chiffrement…

Ainsi, si l'on veut ajouter un filtre sur le serveur de messagerie, il y a trois possibilités: en entrée, en sortie, ou encore intégré au MTA par une API (interface de programmation).

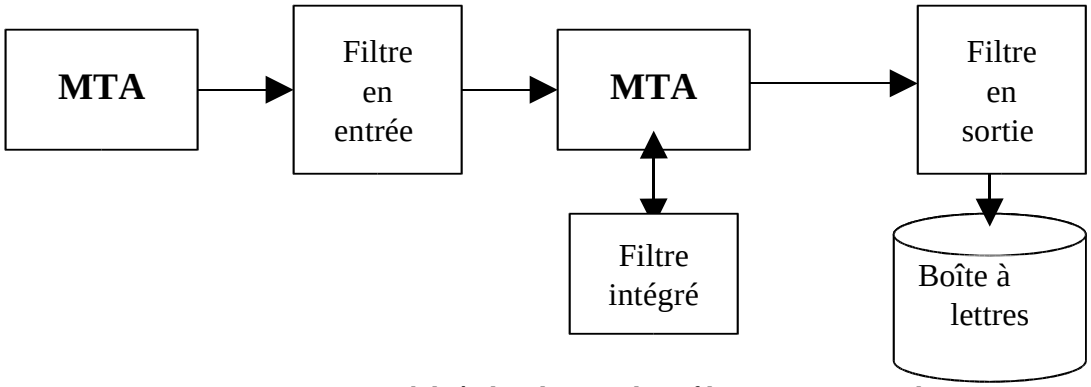


Figure 1 – Les possibilités d'utilisation d'un filtre sur un serveur de messagerie

Les filtres en sortie (après la fonction routage), utilisés généralement directement sur la boîte à lettres du destinataire, sont les plus simples à mettre en œuvre. Un exemple de solution de ce type, est l'utilisation du logiciel *procmail* couplé à un filtre externe. Cette solution permet de personnaliser le filtrage au niveau de l'utilisateur. Ses inconvénients principaux sont le traitement multiple lorsque le même message est envoyé à plusieurs destinataires locaux et le coût des traitements par plusieurs processus.

Le placement en entrée du MTA résout le problème du traitement multiple mais en ajoute d'autres. En effet, les fonctions annexes nécessitent le contact direct entre le MTA et son client, ce qui n'est plus assuré si l'on insère un filtre avant le MTA. Or, ces fonctions n'existent pas d'habitude dans les filtres de contenu (et c'est normal!). Pour contourner cette difficulté, on utilise deux MTA (installés parfois sur deux machines différentes): le premier s'occupant des fonctions annexes et le deuxième de la fonction routage, et on insère le filtre entre les deux. Le dialogue entre les MTA et le filtre se fait selon le protocole SMTP[3].

La troisième solution consiste à utiliser une API de façon que le filtre devienne partie intégrante du MTA. Ce type de filtre peut alors accéder et utiliser toutes les variables internes du MTA pour effectuer un traitement plus fin des messages. Cette solution est beaucoup plus efficace, mais elle exige une interface différente pour chaque MTA (sendmail, postfix…). C'est dans ce type de solution que nous plaçons notre filtre.

Une API, depuis la version 8.12 de Sendmail, permet d'intégrer un filtre externe au fonctionnement du logiciel serveur. C'est l'API *libmilter*. Le filtre s'exécutant dans un processus autre que celui du serveur, la communication entre les deux s'effectue par *socket UNIX*.

Lorsque le serveur reçoit une nouvelle connexion, un nouveau *thread*[4] est créé dans le filtre. A chaque état du protocole SMTP, après avoir effectué le traitement spécifique au serveur, le contrôle passe au filtre qui effectue son propre traitement avant de retourner, à nouveau, le contrôle au logiciel serveur. Le filtre doit donc fournir un point d'entrée pour chaque étape du protocole SMTP.

---

[3] SMTP – Simple Mail Transfer Protocol

[4] Thread – processus léger, POSIX thread – Dans la version actuelle de la bibliothèque milter, chaque connexion correspond à un thread. Il existe une version expérimentale de la libmilter fonctionnant sous le modèle «pool of workers », dont le nombre de threads est limité. Cela permet de traiter plusieurs centaines de connexions en même temps sur des serveurs très importants.

# 3 Caractéristiques du filtre j-chkmail

Nous avons voulu imposer quelques contraintes au filtre de façon à l'adapter à notre organisation et à nos ressources. Ces contraintes sont certainement souhaitées dans d'autres types de structures.

Globalement, outre l'efficacité du filtrage, ces contraintes visent à obtenir une meilleure sécurité et une utilisation optimale des ressources matérielles.

- **Privilégier l'utilisation des fonctionnalités du serveur SMTP** – Si une fonctionnalité souhaitée existe déjà dans le serveur de messagerie, il ne faut pas la recréer dans le filtre.
- **La rapidité du traitement** – Ceci répond à deux problèmes : la sécurité et la bonne utilisation des moyens matériels existants. Plus la charge, en régime normal, du serveur est importante, plus facilement il peut devenir la cible des attaques du type DoS (Déni de Service). D'autre part, l'utilisation d'algorithmes performants ainsi que l'action au bon moment permettent l'utilisation du matériel déjà existant. Le but est d'avoir un filtre dont la consommation marginale de ressources soit du même ordre de grandeur, ou encore mieux – plus petite, que celle du serveur SMTP.
- **Robustesse et tolérance aux pannes** - un filtre anti-viral ou anti-spam est un logiciel de sécurité et, par conséquent, une possible cible pour les attaques. Des spammeurs peuvent, et font, des attaques visant à anéantir le filtre (blocage ou mort du processus), afin de faire passer tout le trafic qui ne passerait pas si le filtre était présent. Le filtre doit donc être résistant aux attaques et être auto régénérateur, dans le sens où, si une attaque a réussi, le filtre doit détecter le fait, se terminer et relancer une nouvelle instance du processus.
- **Observabilité** – Le filtre doit pouvoir être interrogeable, en temps réel, sur son activité et sa performance. Il s'agit de consulter des informations globales du filtre mais aussi l'activité d'un client donné ou alors un type d'événement spécifique. Le filtre doit maintenir des compteurs permettant de présenter, par exemple, un résumé de l'activité pendant les six dernières heures - pour le filtre ou pour une passerelle désignée - ou alors lister les passerelles qui ont envoyé des virus pendant les 3 dernières heures, ou qui ont essayé d'envoyer trop de messages à des utilisateurs inexistants.
- **Journal d'événements** - tout système de filtrage est susceptible de provoquer des fausses détections. Ainsi, il est indispensable d'avoir un journal qui enregistre tout traitement exceptionnel donné à un message (rejet, remplacement…) pouvant être consulté à posteriori et indiquant ce qui a causé ce traitement d'exception. Ce journal d'événements est constitué principalement par les fichiers de log (syslog) qui ne sont effacés que par intervention externe au filtre.
- **Extensibilité** – Le filtrage de spam est un problème très évolutif. Ainsi, il est souhaitable de pouvoir changer la structure du filtrage ou d'ajouter des nouveaux modules sans difficulté, et sans que la structure du filtre soit modifiée à chaque ajout.
- **Respect du contenu** - Le filtrage des messages par leur contenu, même par des moyens automatisés, peut poser des problèmes juridiques. Nous avons choisi de privilégier, autant que possible, parmi les méthodes de filtrage, celles ne prenant pas en compte le contenu. Ainsi, si l'on peut détecter un spam par certaines caractéristiques techniques du message ou de la connexion, on ne regardera pas le contenu du message. C'est ce que nous appelons filtrage comportemental, en opposition à filtrage par le contenu.

Pour assurer la robustesse, la qualité de la programmation doit être très stricte: les raccourcis et astuces de « programmeurs chevronnés » ne sont acceptés que si elles apportent un plus au filtre, et à condition de ne pas nuire à la lisibilité. Nous avons imposé à j-chkmail des règles de programmation et d'organisation similaires à celles de sendmail lui-même.

Pour assurer la tolérance aux pannes, j-chkmail est constitué de deux instances: le filtre et un superviseur. En cas de terminaison ou de fonctionnement anormal du filtre, le superviseur le détecte, provoque la terminaison du filtre (si c'est le cas) effectue les tâches de maintenance nécessaires et lance une nouvelle instance du filtre.

Pour assurer l'observabilité du filtre, le contexte et les résultats de chaque connexion sont enregistrés dans des historiques de taille fixe (auto-nettoyants). Les variables internes du filtre sont accessibles soit par mémoire partagée, soit par enregistrement périodique sur le disque. Ces informations peuvent être consultées soit par un outil en ligne de commande, soit par des scripts de surveillance (MRTG, RRDtool…).

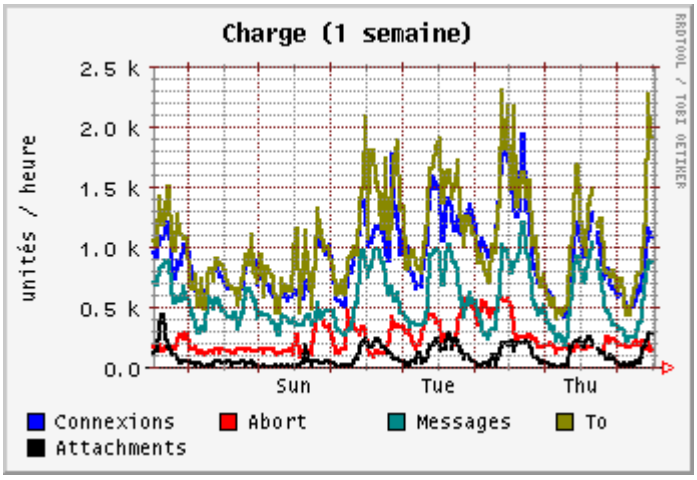


Figure 2 - Visualisation en temps réel du fonctionnement du serveur

Plusieurs principes ont été choisis pour obtenir un traitement rapide des messages.

- Corrélation entre les connexions de même origine – La plupart des filtres, et sendmail lui-même sont des automates sans mémoire. Ils considèrent chaque nouvelle connexion comme étant un événement nouveau. Or, connaître le résultat des messages arrivés par le même chemin est une information en plus, de valeur parfois non négligeable. Cela correspond en fait, à gérer des « listes blanche/noire dynamiques ». Si l'on sait qu'une certaine passerelle « ne se comporte pas bien » et cela de façon habituelle, on peut refuser ses connexions sans perte de temps.
- Détecter, dans la mesure du possible, avant la commande DATA du protocole SMTP, si une connexion doit être refusée. Terminer avant le début de réception de message permet non seulement d'éviter les traitement coûteux de vérification de contenu, mais aussi de protéger le serveur contre certaines attaques de déni de service.
- Classement des clients se connectant sur le serveur selon un critère de confiance ou proximité, selon son adresse IP. j-chkmail définit quatre classes : réseau local, domaine, réseau ami et inconnu (tous ceux qui n'ont pas été classés dans les classes précédentes). Ainsi, si l'on suppose que les machines de notre propre domaine ne nous envoient pas du spam, on ne perdra pas du temps à vérifier le contenu des messages en provenance de ces clients. Ce raisonnement n'est valable que s'il existe une confiance effective à ces machines.
- Programmation en C – c'est le langage le plus adapté au traitement de l'information sur des serveurs performants.

# 4 Le filtrage antiviral

On différencie ici les virus et les vers de messagerie. Les premiers sont constitués d'un code malveillant capable de s'exécuter sur un ordinateur cible et de causer des dégâts pouvant aller de la simple apparition d'un message d'indication de présence jusqu'au formatage du disque dur. Les deuxièmes sont des virus qui se propagent par messagerie, utilisant le carnet d'adresses de l'ordinateur infecté. Sauf quelques rares exceptions, un ver de messagerie est vu par un serveur SMTP comme un message avec un fichier attaché contenant le code exécutable – ces exceptions sont, par exemple, des pages HTML avec des scripts intégrés ou alors des documents bureautiques avec des macros.

Pour détecter un virus, un filtre classique a donc besoin d'extraire les fichiers attachés pour les examiner. Cet examen consiste, dans la plupart des cas, à rechercher l'existence de certaines chaînes d'instructions. Ce sont les signatures des virus. A cela, on peut aussi ajouter d'autres vérifications, comme la taille du fichier ou la signature du fichier évaluée par une fonction de hachage. Ainsi, pour accomplir sa fonction, un antivirus a besoin d'accéder à une base de signatures des virus connus, et il faut que cette base soit mise à jour aussi souvent que possible.

Notre approche est différente. Nous constatons que la grande majorité des virus se propage à l'intérieur de fichiers attachés qui sont exécutés automatiquement par les clients de messagerie sur l'ordinateur cible. Cette propriété est déterminée par le type du fichier, c'est à dire, par son extension. L'idée est donc de détecter les messages dont les fichiers attachés appartiennent à une classe définie par une liste d'extensions. Cette classe de fichiers constitue les «Unsafe Files », tels qu'ils sont définis par Microsoft [2] :

```
ade   adp   bas   bat   bin   btm   chm   cmd   com   cpl   crt   dll
drv   exe   hlp   hta   inf   ini   ins   isp   je    js    jse   lnk
mdb   mde   msc   msi   msp   mst   pcd   pif   reg   scr   sct   shb
```

```
shs   sys   url   vb    vbe   vbs   vxd   wsc   wsf   wsh
```

Cette idée est confortée par le palmarès présenté par Wildlist [3], qui montre que la grande majorité des virus en circulation appartient à la classe des « Unsafe Files ».

En fait, derrière cette apparence empirique se cache une règle élémentaire de bon sens, qui est d'éviter l'exécution de tout programme à l'insu de l'utilisateur. Tous les virus et vers de messagerie sont, sans aucune exception, des programmes qui s'exécutent à l'insu de l'utilisateur. C'est le lien à couper si l'on veut obtenir une protection absolue.

Par rapport à la méthode traditionnelle de blocage après identification des virus, notre méthode présente des avantages et des inconvénients.

Le premier avantage est la rapidité du traitement, puisqu'il s'agit uniquement d'identifier les fichiers attachés dans le message et de vérifier s'ils appartiennent à la classe des «Unsafe Files ». Le décodage et traitement des fichiers attachés, tel qu'il est effectué par les antivirus classiques, devient inutile.

Le deuxième avantage est l'absence de mises à jour fréquentes de la base de signatures, puisque la classe «Unsafe Files » est une classe stable. C'est un avantage aussi bien du point de vue de l'administration du serveur que de l'efficacité, puisque tout nouveau virus est immédiatement détectable.

Cette caractéristique a montré son efficacité à plusieurs reprises cette année: en janvier lors de l'apparition du virus Lirva, ou encore en août lors de l'apparition de Blaster. Lors de l'apparition de ce dernier, le nombre de messages bloqués est passé du chiffre habituel de 100 par jour à un pic de 3000, pour redescendre lentement après. La mise à jour des antivirus par les éditeurs n'était disponible que le lendemain. Le moment de l'apparition de ce virus, pendant les vacances, fait apparaître le problème de l'évaluation des dégâts, puisque malgré les mises à jour automatiques des logiciels antivirus (qui se fait avec retard), l'intervention humaine (absente pendant les vacances) est toujours nécessaire pour restaurer les postes ayant été infectés.

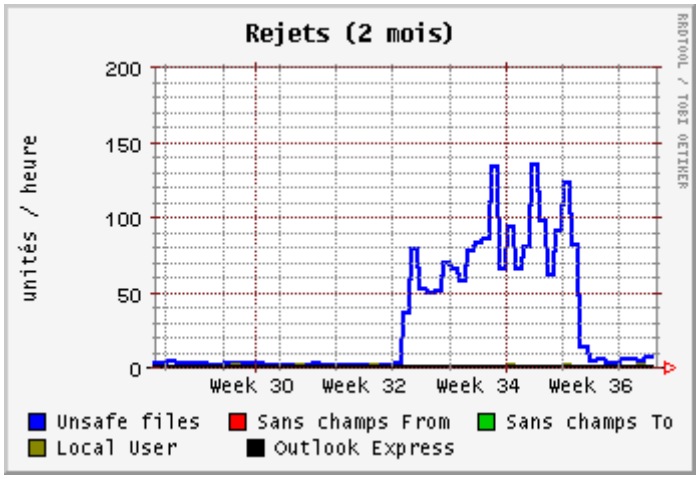


Figure 3 – Activité du virus Blaster en août/septembre 2003

Certains virus peuvent profiter de certaines possibilités prévues par la RFC 2046 [4]: l'envoi d'une référence ou le découpage du message en parties. Dans le premier cas, ce qui est transmis n'est pas le virus lui même, mais l'URL à partir d'où télécharger le virus. Dans le deuxième cas, le virus est envoyé en plusieurs morceaux qui seront assemblés sur l'ordinateur du destinataire. L'antivirus classique, ayant besoin d'accéder au contenu entier du virus pour le détecter, ne pourra pas accomplir sa fonction.

L'inconvénient est que les virus du type macro embarqués dans des documents issus de certains logiciels bureautique ne sont pas détectables, ce sont notamment des virus macro des documents Word et Excel ou des scripts embarqués dans du code HTML.

Le tableau 1 montre le résultat du filtrage d'une semaine à l'époque de l'activité du virus Sobig.F (Blaster). Dans la première partie du tableau nous présentons le nombre de «Unsafe Files » classés par type de fichier. Dans la deuxième partie, ces « Unsafe Files » sont identifiés par un antivirus classique.

D'autre part, certains virus s'attaquent aussi aux serveurs de messagerie, le but étant de tromper le système de filtrage. Les attaques les plus courantes sont des balises MIME incorrectes, qui empêcheront le filtre de déchiffrer correctement les fichiers attachés, mais qui seront interprétées correctement par l'outil de messagerie du destinataire.

Mais, indépendamment de la méthode utilisée, le filtrage antiviral sur le serveur de messagerie ne résout pas complètement le problème. Certains vers utilisent plusieurs méthodes pour se propager, comme par exemple, le partage de fichiers dans le réseau local, le webmail et le web.

On peut aussi citer le cas des ordinateurs portables des utilisateurs itinérants qui, une fois infectés ailleurs, sont branchés sur le réseau local.

Ainsi, le filtrage effectué sur le serveur de messagerie doit être complété par la protection des postes clients. Cela passe par l'utilisation de logiciels antivirus sur ces postes (mis à jour selon une procédure journalière automatisée), et une correcte configuration des logiciels d'accès à internet.

| | 02/09 | 03/09 | 04/09 | 05/09 | 06/09 | 07/09 | 08/09 | TOTAL |
|---|---|---|---|---|---|---|---|---|
| * .bat | 13 | 11 | 14 | 13 | 7 | 2 | 11 | 71 |
| * .exe | 51 | 62 | 43 | 29 | 19 | 30 | 47 | 281 |
| * .mdb | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 3 |
| * .pif | 1277 | 1587 | 2812 | 2003 | 1284 | 1903 | 2420 | 13286 |
| * .scr | 187 | 245 | 346 | 264 | 154 | 227 | 341 | 1764 |
| * .url | 1 | 1 | 3 | 0 | 1 | 0 | 2 | 8 |
| * .zip | 21 | 16 | 16 | 17 | 7 | 6 | 24 | 107 |
| TOTAL | 1550 | 1922 | 3234 | 2327 | 1472 | 2170 | 2845 | 15520 |

Virus en quarantaine...

| | |
|---|---|
| - W32/Sobig.f@MM | 2844 |
| - W32/Klez.eml | 56 |
| - W32/Sobig.dam | 39 |
| - W32/Dumaru.a@MM | 30 |
| - W32/Mimail@MM | 24 |
| - Exploit-MIME.gen.exe | 9 |
| - W32/Bugbear.b.dam | 5 |
| - W32/Bugbear.b@MM | 4 |
| - W32/Dumaru.h@MM | 3 |
| - W32/Hybris.gen@MM | 3 |
| - W32/Yaha.g@MM | 1 |
| - W32/Pate.b | 1 |
| - W32/Bugbear@MM | 1 |

Tableau 1 – Activité du virus Sobig.F - Blaster

L'utilisation de tous ces outils logiciels peut (et doit) être complétée par une bonne configuration des postes clients [5], de façon à éviter qu'un message contenant du code malveillant puisse être exécuté à l'insu de l'utilisateur.

# 5 Le « *spam* »

Le filtrage de spam est un problème complètement différent du filtrage des virus. Le nombre de virus en circulation ainsi que les modes de propagation sont assez réduits et il n'y a pas d'enjeu financier comme c'est le cas pour le spam. Le but du spam est le bénéfice financier résultant d'une campagne publicitaire de coût extrêmement réduit.

Non seulement la diffusion des messages publicitaires utilisant la messagerie électronique coûte beaucoup moins cher que le courrier traditionnel ou télécopie, mais aussi, cela permet d'avoir une étendue géographique bien plus importante avec des délais insignifiants. L'absence d'un quelconque type de censure permet l'utilisation de ce moyen de communication pour faire passer certains messages qui ne passeraient pas par les moyens classiques: la pornographie ou les propositions d'escroquerie, par exemple.

La raison financière explique l'agressivité et l'acharnement des spammeurs pour faire évoluer leurs méthodes: à chaque nouvelle protection mise sur les serveurs, une nouvelle astuce est trouvée pour faire passer les messages.

La guerre contre le spam a commencé vraiment vers 1997, lorsque les serveurs de messagerie étaient peu protégés contre le spam. A cette époque, les spammeurs cherchaient surtout à utiliser les ressources des serveurs publics pour relayer les messages. Ainsi, pour diffuser un message à un million de destinataires, il suffisait d'un seul message envoyé à un relais ouvert qui le distribuait à l'ensemble des destinataires. La première mesure prise par les administrateurs des services de messagerie a été de ne plus laisser relayer les messagesselon la règle suivante : sauf cas particuliers, aucun message venant de l'extérieur ne peut repartir vers l'extérieur. Il existe encore beaucoup de relais ouverts, qui sont souvent utilisés par les spammeurs dans le but d'utiliser les ressources des autres plutôt que les leurs.

C'est à ce moment que sont apparues les listes noires sur DNS. Il s'agit d'utiliser des bases de données publiques contenant des adresses de machines utilisées pour relayer du spam ou alors des sources connues de spam. La maintenance de ces bases de données est actuellement effectuée soit par des entreprises (service payant – e.g. http://www.mail-abuse.org), soit par des groupements libres. Des exemples de groupements libres sonthttp://dsbl.org et http://ordb.org.

Ensuite, la lutte contre le spam s'est divisée en deux types, la vérification de contenu (le fond) et la vérification des caractéristiques des messages/connexions (la forme).

La vérification de contenu peut utiliser des méthodes simples, telles la recherche de certains mots clefs dans les messages, ou alors de méthodes plus complexes, telles les analyses statistiques (méthodes bayesiennes…). Ces dernières cherchent à inférer, à partir de l'évaluation de certains critères pré-définis, l'objet du message, ou alors tout simplement s'il s'agit d'un message publicitaire ou pas. Un score est attribué à chaque test. Lorsque ce score dépasse un certain seuil, on déclare qu'il est fort probable que le message soit un spam.

Ce sont des méthodes mettant en cause le contenu du message.

La vérification des caractéristiques techniques des messages/connexions cherche à identifier si le message a été émis par un « être humain » utilisant un logiciel habituel de messagerie ou par un système automatisé d'envoi en masse, sans forcément s'intéresser à son contenu.

Les trois logiciels de filtrage « open source » vraisemblablement les plus connus actuellement sont SpamAssassin, Bogofilter et SpamOracle [6].

SpamAssassin effectue une série de tests (un peu moins de mille) pour attribuer un score final au message. Ce sont des tests plus ou moins représentatifs. Le poids de chaque test a été établi initialement selon un processus d'optimisation sur une population de spams et de messages légitimes. Les messages arrivant sur l'ordinateur du destinataire pourront être classés dans une boîte aux lettres spéciale, selon le score attribué par SpamAssassin et le seuil défini par l'utilisateur.

SpamOracle [6], conçu par Xavier Leroy (INRIA) vérifie pour chaque mot du message la fréquence avec laquelle il apparaît dans une population de spam/non spam et attribue lui aussi un score. C'est l'utilisateur qui constitue la base de messages qui permettra à SpamOracle d'apprendre à distinguer les spams des messages légitimes. SpamOracle s'intégre très facilement à procmail, est un outil orienté utilisateur, alors que Bogofilter est destiné plutôt à l'utilisation collective.

Des logiciels comme SpamAssassin, SpamOracle et Bogofilter donnent d'excellents résultats sur des petites communautés ou des communautés homogènes, mais présentent quelques inconvénients qui sont dus au principe même d'apprentissage sur un ensemble de messages précis.

- Représentativité spatiale – Si la population d'utilisateurs est importante et hétérogène, non seulement la proportion de spam/non spam n'est pas la même si on prend le tout ou une partie des utilisateurs, mais aussi la valeur attribuée à chaque test. Ainsi, dans un hôpital, la probabilité qu'un message contenant le mot «*viagra* » soit un spam est plus importante dans le corps administratif que dans le corps médical.
- Représentativité temporelle à court terme – Le spam évolue dans le temps pour deux raisons principales. Les messages que les spammeurs ont à écouler un jour ne sont pas forcément les mêmes le lendemain. Ainsi, les paramètres permettant d'avoir une probabilité d'erreur optimale un jour ne sont pas les mêmes le lendemain.
- Faiblesses dues à la connaissance du filtre - Connaissant parfaitement les critères utilisés par les logiciels de filtrage, on voit apparaître des messages qui visent à briser la façon dont la classification est faite. Par exemple, on voit apparaître des messages avec des mots invisibles dans la partie html (couleur du texte égale à la couleur du fond). Ce sont des mots qui ne sont pas tout à fait aléatoires, mais des mots qui n'apparaissent pas, d'habitude, dans les spams. Ceci a pour but, si le nombre de messages de ce type dans la base d'apprentissage est important, d'augmenter l'entropie et, par conséquent, de rendre plus floue la frontière entre spam/non spam. Dans d'autres types de système liés à la sécurité, en particulier les systèmes cryptographiques, la connaissance de son fonctionnement ne doit pas faciliter des éventuels attaques : c'est un pré-requis lors de la conception.

D'autres voies sont en cours d'étude par l'ASRG (Anti-Spam Research Group) à l'IETF [7]. Globalement, on peut dire que ces voies visent, d'une part, à diminuer les messages avec utilisation de fausses adresses d'émission et, d'autre part, à permettre au destinataire final de définir ce qu'il souhaite recevoir ou pas (Consent Framework).

Pour être capable de satisfaire les besoins d'une population importante et hétérogène, l'approche de j-chkmail est, tout en restant dans le domaine des méthodes traditionnelles de lutte contre le spam, de privilégier les vérifications de ce qui est effectivement caractéristique du spam. Nous avons effectivement ajouté un «oracle » dans la dernière phase du filtrage, mais cela n'intervient que sur la dernière tranche des messages non filtrés, donc avec un poids plus faible, comme une sorte de réglage fin du filtrage.

# 6 La détection de SPAM par j-chkmail

j-chkmail intègre plusieurs méthodes de détection de spam. Le but n'est pas de supprimer complètement le spam, mais plutôt de le faire descendre à un niveau acceptable. Ainsi, on bloquera sur le serveur ce qui est manifestement du spam, ou ce qui représente une menace à son fonctionnement. Des messages dont le contenu est douteux sont marqués par l'ajout d'un en-tête. Cela permet à l'utilisateur de les diriger vers une boîte aux lettres où il pourra faire le tri, ou alors les supprimer directement.

Des discussions qui ont eu lieu sur la liste smtp-fr du CRU, nous avons appris que plusieurs utilisateurs utilisant un autre logiciel de filtrage de spam arrivent à recevoir jusqu'à 150 messages marqués par jour dans leurs boîtes à lettres des messages douteux. A eux de les passer en revue dans le but de retrouver les éventuelles erreurs d'appréciation. A notre avis, c'en est trop ! Notre but est d'éliminer tant que possible sur le serveur et de faire en sorte que l'utilisateur final ne reçoive pas plus d'une dizaine de messages marqués par jour. Cela nous semble être la limite acceptable par l'utilisateur, pour que ça ne devienne pas une corvée.

Aussi, il nous a semblé utile de classer les clients SMTP selon leur adresse IP. Cela nous permet d'attribuer des privilèges aux machines locales ou de notre propre domaine: il est à supposer que nos propres machines ne sont pas la source de spam.

Le premier point important de notre filtrage est d'essayer d'arrêter les connexions dès que possible. En effet, les étapes les plus coûteuses du filtrage sont celles concernant le traitement du contenu du message. Ainsi, plus le serveur est chargé, moins il est capable de traiter des cadences de connexions importantes, mais aussi il peut devenir plus vulnérable à des attaques de déni de service.

## 6.1 Mesure de cadence de connexion

Il s'agit de comptabiliser, dans une fenêtre glissante dont la durée est de l'ordre de 10 minutes, le nombre de connexions émises par chaque client et de les refuser si cette cadence dépasse un seuil. L'idée est d'identifier si les connexions proviennent d'un « être humain » (processus de Poisson) ou d'un robot (rafales). Un robot peut envoyer des messages, mais il peut aussi être utilisé pour essayer de récupérer l'ensemble des utilisateurs valables sur le serveur (*harvest attack*). La mesure de la cadence de connexion est une approximation valable, puisque la vérification statistique en temps réel pour chaque client est, dans la pratique, difficile à mettre en œuvre.

j-chkmail vérifie, pour chaque nouvelle connexion, si la cadence de ce client dépasse la limite autorisée, selon sa classe d'appartenance : machine locale, machine amie ou inconnue. Un seuil de 10 à 15 connexions pour les machines inconnues semble suffisant pour la plupart des structures, indépendamment du nombre d'utilisateurs. En effet, généralement si l'on augmente le nombre d'utilisateurs internes, le nombre de correspondants externes croît aussi.

Pour mettre ceci en pratique, il faut placer dans la classe «*réseau ami* » les passerelles externes avec lesquelles notre domaine communique beaucoup : les domaines partenaires et certains serveurs de listes de diffusion. Cette tâche n'est pas aussi difficile qu'on peut le penser à première vue.

Même si les résultats de ce filtrage sont bons, le plus important est la protection apportée contre les attaques de déni de service. En effet, il est facile de réussir une attaque de déni de service contre un serveur de messagerie si on l'attaque avec une cadence de connexion suffisamment élevée et si le serveur n'a pas de protection particulière.

## 6.2 Résolution DNS de la passerelle

Il s'agit de vérifier si le client SMTP est correctement déclaré dans un DNS. En effet, nous avons observé que les spammeurs utilisent souvent des machines secondaires ou des machines non déclarées dans les DNS, pour envoyer du mail. Cela leur permet de passer inaperçus. Généralement, les machines principales (serveurs DNS, mail, web, proxies…) sont correctement déclarées, ce qui n'est parfois pas le cas pour les machines terminales des utilisateurs. Ces dernières machines

sont souvent utilisées pour envoyer du spam leur permettant de dissimuler leur activité. Le fournisseur ne se rendra compte que du volume transféré et, peut-être, du protocole utilisé – il n'aura pas accès aux informations de niveau *application.*

Le problème de cette méthode vient du taux de faux positifs: il existe des serveurs de mail légitimes incorrectement enregistrés dans les DNS. Il s'agit, pour la plupart, de serveurs de petites entités. Pour palier à cela, j-chkmail attribue un petit quota de connexions par jour, à des machines mal déclarées dans le DNS. Il est aussi possible de les déclarer dans une *liste blanche.* Si cette fonctionnalité n'est pas activée, elle est signalée dans la détection de spam par l' « oracle ».

Les résultats de cette vérification sont très bons, notamment pendant les nuits et week-ends.

## 6.3 Détection de connexions vides ou attaques de « dictionnaire »

Il s'agit de détecter les passerelles faisant des connexions sans envoyer de message ou avec un nombre d'erreurs d'adressage trop importants. Cette fonctionnalité a pour but d'arrêter les tentatives de récupération de l'ensemble des adresses valables sur le serveur ou les connexions dont le spammeur choisit au hasard les adresses des destinataires.

## 6.4 Les utilisateurs internes (adresses intranet).

Ce sont les adresses qui ne peuvent être utilisés qu'en interne: des adresses de service ou alors des adresses permettant d'attendre tout le personnel de l'organisation, par exemple. j-chkmail gère une liste d'adresses ne pouvant pas recevoir des messages qu'en provenance d'une passerelle connue. Cette fonctionnalité n'arrête pas beaucoup de spam, mais elle permet de protéger ces adresses souvent critiques.

## 6.5 Recherche d'expressions régulières

Le but est de rechercher, dans le corps du message, l'existence de certaines expressions régulières. Ces expressions peuvent être des expressions caractéristiques, comme par exemple «*Click Here* » ou des expressions confirmées, comme par exemple, « *http://www.freesexontheweb.com* ».

Le premier type constitue un indicateur possible de spam, tandis que le deuxième correspond bien à un spam confirmé déjà détecté. Il s'agit le plus souvent de l'URL d'un site dont le spammeur suggère la visite. Un message ayant l'expression *Click Here* sera marqué par un en-tête (et non pas bloqué), tandis que le deuxième message sera rejeté et n'arrivera pas sur la boîte aux lettres du destinataire.

Des expressions régulières plus complexes permettent de détecter certaines astuces des spammeurs. En voici quelques unes :

- **http://.*freesexontheweb.com** – certains spammeurs changent souvent le nom de leur site web, tout en gardant le même nom de domaine (ajout d'une composante aléatoire dans l'URL). Ex: http://random_part.freesexontheweb.com
- **http://[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}/** - URLs numériques (une adresse IP, au lieu d'un nom de machine) permettant l'installation d'un site web sans avoir un nom de domaine ou de déclaration de nom dans un DNS. Il s'agit très souvent d'un site qui va disparaître très rapidement, ou alors d'une machine piratée. Ex: http://62.136.34.18
- **http://.*@** - URLs avec nom de login – permet d'ajouter une composante aléatoire dans l'URL et de tracer les destinataires pour avoir confirmation de leurs adresses, e.g. http://Joe.Denver@www.freesexontheweb.com
- **http://.*%[0-9a-f]** - Codage hexadécimal des caractères. Ceci permet de dissimuler certaines chaînes de caractères. Ex: http://www.free%73%65%78ontheweb

La constitution d'une liste d'expressions régulières semble être une tâche fastidieuse, alors qu'elle ne l'est pas. Nous avons utilisé les spams reçus dans notre propre boîte aux lettres ainsi que celle de postmaster pour construire notre liste. Au départ, une dizaine d'expressions (URLs) étaient ajoutées chaque jour. Au bout de trois mois, nous n'avons ajouté qu'une ou deux expressions par jour, voire moins. Certaines expressions s'en vont et parfois reviennent. Il s'agit donc, d'utiliser un script de dépouillement (en cours de développement) permettant de détecter la fréquence d'apparition et l'efficacité de chaque expression pour pouvoir gérer la sortie des expressions inutiles. C'est en quelque sorte un processus d'apprentissage *manuel.* Dans notre cas, nous estimons que le nombre d'entrées utiles dans notre table est de l'ordre de 500. Le nombre d'expressions vues la dernière semaine est de l'ordre de la centaine.

## 6.6 L'historique des connexions

Il s'agit de tenir compte de l'historique des connexions passées et plus particulièrement celles qui ont été rejetées. Le but principal est la constitution d'une liste noire dynamique avec l'ensemble des critères précédents.

La plupart des critères précédents, considérés individuellement, étaient déjà dans des listes noires dynamiques. On peut alors se demander pourquoi une de plus et si cette dernière ne remplace pas l'ensemble des précédentes. La réponse est non.

Les listes noires dynamiques fondées sur des critères spécifiques ont un temps de réaction très court et servent surtout à répondre à des événements ponctuels. Aussi, le temps de présence d'une passerelle dans une de ces listes est court: de quelques minutes à quelques heures. La prise en compte de l'historique des connexions permet d'avoir une liste dynamique dont la durée de vie peut-être de quelques jours, puisqu'il s'agit, grosso modo, d'analyser le nombre de fois qu'une certaine passerelle a « récidivé ».

## 6.7 La vérification de la forme ou le « j-oracle »

Il s'agit d'un ensemble de vérifications sur le contenu et sur la forme des messages. C'est une vérification résiduelle, de la « *dernière chance* », puisqu'il s'agit de la dernière vérification destinée à détecter les spams qui ont réussi à passer les tests précédents.

A première vue, ce module peut sembler avoir le même principe de fonctionnement que SpamAssassin, mais les différences sont significatives.

- Il s'agit d'avoir un petit nombre de tests avec une grande pertinence, plutôt qu'un nombre important de tests dont la pertinence sera définie par un processus d'apprentissage sur une population de messages spam/non spam.
- L'attribution du score est faite non pas en fonction d'une fréquence d'apparition des critères dans les populations spam/non spam, mais en fonction de la *gravité* de la déviation comportementale par rapport à un message légitime.

Ces deux points assurent que la méthode dépend le moins possible d'une distribution statistique de la population d'apprentissage comme c'est le cas pour les outils tels SpamAssassin ou Bogofilter.

Au moment de l'écriture de cet article, ce module est en cours de validation, mais les résultats sont déjà très intéressants. Avec une quinzaine de vérifications résiduelles, la probabilité de détection semble être supérieure à 50 % et la probabilité de faux positif inférieure à 10 %. Même si ces chiffres peuvent sembler insuffisants, il faut remarquer qu'il s'agit du résultat sur un trafic déjà largement dégrossi par les vérifications précédentes.

## 6.8 Résultats

Avec cette série de vérifications, nous avons construit un système de filtrage de spam robuste et efficace. Nous mesurons le résultat par le nombre de messages apparaissant sur la boite aux lettres des utilisateurs. Le critère est surtout de savoir si ce nombre est acceptable ou pas. Pour la grande majorité des utilisateurs cela est vrai.

Au moment de la rédaction de cet article, et avec l'option j-oracle expérimentale activée, le nombre de spams arrivant dans la boîte aux lettres de certains de nos utilisateurs les plus exposés est de l'ordre de dix par jour, la plupart dans la boîte aux lettres des messages marqués. Même si cette quantité semble encore importante, il faut tenir compte du fait que le serveur rejète déjà en moyenne un tiers des connexions entrantes, et le fait qu'il s'agit d'un serveur traitant la messagerie d'une population hétérogène d'environ 1500 utilisateurs.

Il existe, bien évidemment, des utilisateurs qui continuent à recevoir beaucoup plus de spam que d'autres. Mais nous avons aussi constaté qu'il s'agit le plus souvent d'utilisateurs qui s'exposent, ou qui sont exposés plus que les autres, soit parce qu'ils fréquentent et s'inscrivent dans des nombreuses listes de diffusion, où les adresses sont récupérées, soit parce que, à cause de l'importance de leurs fonctions, leurs adresses apparaissent dans des documents publics.

D'autre part, certaines situations font qu'il est impossible de supprimer complètement tout message publicitaire sans commettre des erreurs. Comment savoir qu'un message publicitaire, envoyé en masse, par un procédé automatisé, vous proposant les dernières nouveautés musicales vous arrive suite à un achat d'un livre technique que vous avez fait en ligne sur un site très connu ?

La lutte contre le spam semble être une lutte sans fin. Au fur et au mesure que l'on met au point des nouvelles méthodes de détection, les spammeurs trouvent de nouvelles façons de passer outre le filtrage. Il s'agit le plus souvent méthodes de dissimulation : le codage en base 64, envoi d'une image au lieu d'un texte, ajout de commentaires html à l'intérieur d'un mot…

Dans la situation actuelle, ce qui semble être la solution la plus efficace serait être capable d'identifier si l'adresse de l'expéditeur (au moins le nom du domaine), a été usurpée ou pas. C'est une des voies en voies d'étude à l'ASRG. Cela donnera certainement lieu à des modifications soit dans le protocole SMTP, soit dans le contenu des DNS.

# 7 Sécurité – protection du serveur

j-chkmail est, à notre connaissance, le seul filtre de messagerie qui incorpore des fonctionnalités permettant de protéger le serveur contre des attaques de déni de service.

Attaquer un serveur de messagerie n'est pas le but premier des spammeurs. Ils en ont besoin pour effectuer leur « *livraison* ». Néanmoins, il peuvent s'intéresser au filtre : si celui-ci est un obstacle, ils essayent de le « planter » et ainsi pouvoir effectuer leur livraison sans aucun empêchement. Dans ce cas, le comportement du MTA peut être soit de refuser toute connexion tant que le filtre n'est pas présent, soit de tout laisser passer. Les deux solutions sont mauvaises.

Lorsqu'on démarre j-chkmail, c'est un processus superviseur qui est lancé. Celui-ci lance une instance du filtre et la surveille. S'il détecte un fonctionnement anormal ou l'arrêt du filtre, il provoque la terminaison du filtre, si c'est le cas, effectue les tâches de maintenance nécessaires et relance une nouvelle instance propre du filtre.

La mesure de la cadence de connexion a démontré être un outil redoutable contre les attaques de déni de service, même distribuées. j-chkmail comptabilise la cadence de connexion par client smtp, mais aussi pour le serveur. Si l'on ne peut pas vérifier avec certitude que le comportement de chaque passerelle correspond bien à un processus de Poisson, cette même vérification faite au niveau du serveur donne des résultats utilisables, à condition que la cadence de connexions soit d'au moins 15 à 20 connexions par minute.

En avril 2003, notre serveur a été attaqué par une classe C entière (239 passerelles) du domaine rapiddealsbyemail.com: 12000 connexions en 8 minutes, avec un pic de 80 connexions dans la même seconde. L'attaque a été détectée très rapidement et 9000 connexions ont été refusées par ce seul critère. Le plus important est de signaler qu'aucun message légitime n'a été perdu. Cet événement montre l'intérêt de bloquer les connexions dès que l'on sait que la connexion vient d'une passerelle « inintéressante ».

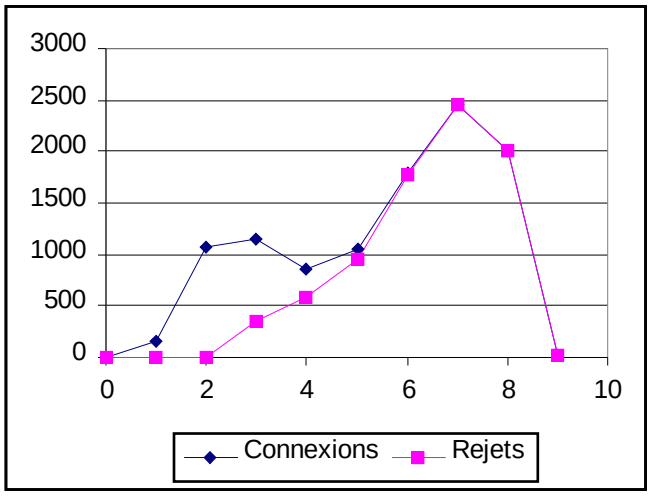


Figure 4 – Comportement du filtre face à une attaque DoS distribuée (nombre de connexions vs temps en minutes)

Une autre protection existante sur le filtre est la limitation du nombre maximal de connexions ouvertes simultanément par chaque client SMTP. En effet, il est possible, dans certains cas, d'épuiser les ressources du serveur de messagerie (mémoire, processus actifs, descripteurs de fichier…), si un client malveillant ouvre simultanément un nombre important de connexions « lentes », c'est à dire, connexions avec un débit très faible, mais avec une durée qui n'est limitée que par le contrôle de dépassement de temps effectué par le serveur (qui est de l'ordre de deux heures). La protection apportée par j-chkmail ainsi que quelques précautions élémentaires permettent de repousser plus loin les limites du serveur. En principe, à chaque fois que l'on pose une borne supérieure à une ressource du serveur, on crée une possibilité de déni de service. C'est le cas, par exemple, lorsqu'on active la limitation globale du nombre de connexions simultanées (variable MaxDaemonChildren).

# 8 Performance

Nous estimons la performance de notre filtre plus que satisfaisante. Notre serveur est un ordinateur Sun E280R, doté de deux processeurs tournant à 900 MHz. Le trafic habituel dans les heures pleines est de l'ordre de 25 à 30 connexions par minute. Le filtre n'utilise que 12 Mo de mémoire RAM et il contribue à environ 1 % de la charge CPU du serveur.

D'habitude, ce serveur traite entre 30000 et 50000 connexions par jour, avec de l'ordre de 20000 messages transmis. Ces chiffres sont en augmentation rapide, non pas à cause de trafic légitime, mais à cause du spam.

Le temps moyen de traitement d'un message par le filtre se situe autour de 50 ms. On pourrait penser, si le filtre était seul sur le serveur, qu'il serait capable de traiter 40 messages par seconde. Il est donc très raisonnable de limiter cela à 10 connexions par seconde, soit 36000 connexions à l'heure.

# 9 Conclusions

Le sujet filtrage de mail sur serveur de messagerie nous étant cher, nous avons développé, en interne, un système de filtrage adapté à nos besoins et satisfaisant les contraintes de notre organisation.

En ce qui concerne le filtrage anti-viral, le nombre de virus passant au travers de notre filtre est comparable à celui des organismes similaires utilisant des vrais antivirus. Mais il faut savoir que la protection contre les virus ne peut pas se situer uniquement au niveau du serveur de messagerie, puisqu'il ne s'agit pas du seul moyen de propagation des virus. Il est aussi important de protéger les postes clients par un antivirus local, mis à jour régulièrement.

En ce qui concerne la lutte anti-spam, nous avons réussi à réduire le nombre de messages sur la boîte aux lettres des utilisateurs à un niveau acceptable.

Mais, au contraire des virus, le spam est un domaine en constante évolution. La première impression que l'on a quand on voit les méthodes utilisées par les spammeurs et les méthodes pour se protéger, c'est qu'il s'agit d'une course entre «le gendarme et le voleur». Et c'est vrai. Pour les administrateurs des serveurs de messagerie, il faut faire évoluer les outils de protection périodiquement. Pour les utilisateurs, il faut accepter que la suppression complète des messages publicitaires soit une idée utopique, et il faut s'habituer à l'idée d'avoir quelques messages dans sa boîte aux lettres électronique et de les supprimer, comme on fait déjà avec ceux qui apparaissent dans la boîte aux lettres physique à l'entrée de l'immeuble…

Les travaux de l'ASRG visant à détecter facilement l'utilisation de fausses adresses permettra certainement de stopper (ou au moins ralentir) la course entre les spammeurs et les concepteurs de systèmes de filtrage. A partir du moment où l'on pourra retrouver facilement la source des spams, on peut espérer que des mesures légales viendront s'ajouter et permettront d'éliminer les abus publicitaires.

# Annexe

- Interrogation du filtre sur son activité les 24 dernières heures:

```
martins@server:~> j-printstats -q -l 1d
First Connection  : Mon Oct 13 12:40:56 2003
 Last Connection   : Tue Oct 14 12:40:44 2003
 Connections       :   33514
 Gateways          :    9856
 Throttle Max      :     475 / 10 min (for the server)
 Throttle Max      :     184 / 10 min (for a single gateway)
 Duration (sec)    :   0.530  51.116 7738.249 173.716 (min mean max std-dev)
 Work (sec)        :   0.000   0.042    2.136   0.094 (min mean max std-dev)
 Mean Throuput     :   0.392 KBytes/sec
Counts
 Messages          :   18257
 Empty Connections :   13034
 Reject            :    5582
 Volume            :  686812 KBytes
 Mean Volume       :   36.74 KBytes/msg
 Recipients        :   25258
 Bad Recipients    :    5725
 Yield             :    0.54 msgs/connection
 Yield             :    0.75 rcpt/connection
 Files             :    2646
 X-Files           :     219
Reject
 DNS resolve       :     136
   FAIL            :     125
   FORGED          :      11
```

```
 Connection Rate   :     690
 Open Connections  :       3
 Empty Connections :    1235
 Bad Recipients    :     263
 Content reject    :    3254
 Rcpt reject       :       0
 Intranet User     :      11
martins@server:~>
```

- Interrogation du filtre sur les passerelles qui nous envoient du spam.

```
martins@serveur:~> j-printstats -q -m c
*** Connections rejected by content checking

. IP ADDRESS           : CONNECT   REJECT : HOSTNAME
. 12.211.233.124       :       1        1 : 12-211-233-124.client.attbi.com
. 12.239.185.34        :       1        1 : 12-239-185-34.client.attbi.com
. 24.166.57.88         :       1        1 : dhcp16657088.neo.rr.com
. 24.166.90.225        :       1        1 : dhcp024-166-090-225.neo.rr.com
. 24.184.160.166       :       1        1 : ool-18b8a0a6.dyn.optonline.net
. 38.113.200.29        :       1        1 : out009.toptx.com
. 61.247.244.199       :       1        1 :
. 64.253.105.22        :       1        1 :
. 66.31.128.223        :       1        1 : h00402b1d5791.ne.client2.attbi.com
. 67.80.15.191         :       1        1 : ool-43500fbf.dyn.optonline.net
. 80.54.120.41         :       1        1 : ya41.neoplus.adsl.tpnet.pl
. 128.242.104.203      :       1        1 : j5.webshots.com
…
*** Records found : 52
martins@serveur:~>
```

- Interrogation du filtre sur les passerelles qui nous envoient des virus.

```
martins@paris:~> j-printstats -q -m x

*** Gateways sending X-Files

. IP ADDRESS     : CONNECT XFILES : HOSTNAME
. 80.14.11.163   :       1      1 : APastourelles-108-1-1-163.w80-14.abo.wanadoo.fr
. 81.22.75.198   :       1      1 :
. 193.252.22.26  :      14      6 : smtp5.wanadoo.fr
. 193.252.22.29  :      12      4 : smtp2.wanadoo.fr
. 193.252.199.51 :      31      1 : AMontsouris-108-2-4-51.w193-252.abo.wanadoo.fr
. 195.130.132.56 :       1      1 : adicia.telenet-ops.be
. 195.202.193.135:       2      1 : smtp.urbanet.ch

*** Records found : 7

martins@paris:~>
```

- Interrogation du filtre sur la cadence de connexions (nombre de connexions les dix dernières minutes)

```
martins@server:~> j-printstats -t
*** THROTTLE TABLE (units each 10 minutes) at Tue Oct 14 12:46:47 2003

*** CONNECTIONS :    252 / 10 min (3567 entries)
    HISTORY     :  12:05:52 (16384/16384 entries)
martins@server:~>
```

# Références


[1] URL serveur web j-chkmail : http://j-chkmail.ensmp.fr

[2] Unsafe Files : http://support.microsoft.com/default.aspx?scid=kb;EN-US;q262631
[3] The WildList Organization International : http://www.wildlist.org/WildList
[4] RFC 2045-9 - MIME (Multipurpose Internet Mail Extensions)
[5] http://www.certa.ssi.gouv.fr/site/CERTA-2000-INF-002/index.html.2.html
[6] SpamOracle – http://cristal.inria.fr/~xleroy/software.html
[7] ASRG – Anti-Spam Research Group – http://www.ietf.org/asrg