Lutte contre les courriers électroniques non sollicités;

Table des matières

ICourrier électronique; les courriers indésirables.

1Introduction:

Le courrier électronique est probablement l'application la plus utilisée dans le monde de l'Internet. Envoyer un mail est un geste fort répandu chez les internautes. Le Conseil Régional d'Aquitaine est un pionnier, car il dispose de cet outil depuis 1993 pour certains utilisateurs. Désormais, on peut considérer que le courrier électronique est devenu un réflexe pour chacun des 750 utilisateurs depuis 1997.

De ce fait, ils sont devenus très sensibles à une perturbation dans son fonctionnement, ou même la suspicion d'un problème. Il convient donc de maîtriser cet outil, afin d'offrir un haut niveau de qualité de service, ainsi qu'une continuité de service irréprochable. En conséquence de quoi les interruptions de fonctionnement doivent être les plus courtes possibles, et l'on se doit d'essayer de les cantonner en dehors des heures ouvrables.

Parmi les désagréments que subissent les abonnés de messageries, il y en a un nouveau, qui est de la compétence de la délégation TIC:

Le courrier non-sollicité, autrement nommé. SPAM[1]

Les termes suivants sont également employés; UBE pour Unsolicited Bulk E-mail, UCE pour Unsolicited Commercial E-mail, pourriel, polluriel, ...

Certains sites web se font les chantres de la lutte contre ce phénomène; http://www.cauce.org, http://euro.cauce.org comme exemple parmi d'autres.

2Contexte technique:

Ce chapitre décrit le fonctionnement du courrier électronique,

1 L'origine du mot « spam » provient d'une marque de corned-beef; « Spice Pork And Meat » (pâté épicé à base de porc et de viande. Le mot a été repris dans un sketch des célèbres Monthy Python, où, dans un café, une serveuse et des vikings chantent à tue tête le mot spam, afin de solliciter un client à en consommer. Voir `http://halte-spam.com/article.php?id_article=5` pour de plus amples informations.
Par opposition, les courriers qui ne sont pas du spam sont nommés «ham »

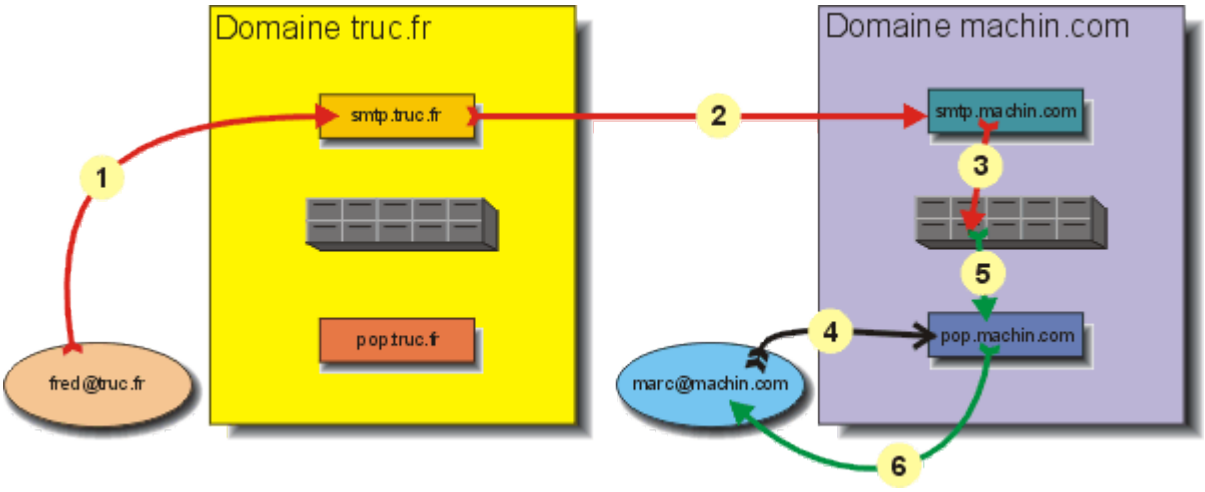
2.1Description par analogie.

Décrivons la vie trépidante d'un courrier électronique. Pour aller d'un expéditeur à un destinataire, il doit suivre un itinéraire. Prenons un exemple simple, et pourtant le plus courant. Soit deux personnes, Fred et Marc. Ils possèdent tous deux une adresse de courrier électronique, chaque adresse est hébergée auprès d'un domaine différent. Supposons que ces adresses soient Fred@truc.fr pour l'expéditeur, et Marc@machin.com pour le destinataire.

Tout comme le courrier papier, on peut distinguer cinq étapes principales.

1. Envoyer le courrier, ce qui correspond à mettre sa lettre dans une enveloppe, et la déposer dans un réceptacle géré par son centre de tri postal. (truc.fr)
2. Acheminer le courrier. Ce centre de tri postal va faire parvenir la lettre au destinataire, à partir des informations que nous avons écrit sur l 'enveloppe. Comme le destinataire n'est pas inscrit dans le même centre de tri, la lettre est envoyée au centre de tri du destinataire. (machin.com).
3. Remettre le courrier. Le centre de tri du destinataire va déposer, via ses facteurs, le courrier dans votre boite aux lettres.
4. Récupération du courrier. Harassé par votre journée de travail, vous rentrez à votre domicile. Vous ouvrez votre boite aux lettres, et vous procédez à la
5. Lecture de votre courrier. Rassurez-vous, nous n'allons pas entrer dans votre vie privée. L'histoire va s'arrêter ici.

Après les jolis contes, voici une bien belle image, comme on aimerait en voir plus souvent.



Pour le courrier électronique, la différence se situe au niveau des points quatre, cinq et six du schéma. En effet, les boites à lettres ne sont pas situées au « domicile » du destinataire, mais au sein du même domaine que son centre de tri. Ainsi, pour pouvoir récupérer votre courrier, il est nécessaire de s'identifier et de s'authentifier auprès du domaine. Nous allons décomposer cette opération en trois phases.

- (point 4) Vous présentez votre identifiant et votre mot de passe au système.
- (point 5) Récupération par ce système de votre courrier dans votre boite aux lettres.
- (point 6) Remise de ce courrier par le système à vous même.

2.2Constitution d'un message.

Un message est constitué de trois parties distinctes.

1. L'enveloppe. On y trouve toutes les informations spécifiques aux moyens de transport, comme l'adresse de l'expéditeur et l'adresse du destinataire. Elle est modifiée par chaque centre de tri relayant le message.
2. L'entête. On y trouve les données nécessaires à la manipulation du message, et qui ne sont pas spécifiques au transport, comme, par exemple, le sujet et la date d'expédition. Ces données sont placées en début du message. Une ou plusieurs lignes vides séparent cet entête du corps du message.
3. Le corps. Ce sont les données utiles, transmises de l'expéditeur au destinataire.

On y trouve les données nécessaires à la manipulation du message, et qui ne sont pas spécifiques au transport, comme, par exemple, le sujet et la date d'expédition. Ces données sont placées en début du message. Une ou plusieurs lignes vides séparent cet en-tête du corps du message.

Voici un message dans son intégralité:

```
From bmayer@xxx  Tue Jul 22 16:47:21 2003
Return-Path: <bmayer@xxx>
Delivered-To: bmayer@localhost.xxx
Received: by xxx (Postfix, from userid 505)
      id 65A194115; Tue, 22 Jul 2003 16:47:21 +0200 (CEST)
Received: from localhost (localhost.localdomain [127.0.0.1])
      by xxx (Postfix) with ESMTP id E47384112
      for <bmayer@localhost>; Tue, 22 Jul 2003 16:47:17 +0200 (CEST)
Delivered-To: mayer@xxx
Received: from xxx [80.118.11.225]
      by localhost with POP3 (fetchmail-5.9.7)
      for bmayer@localhost (single-drop); Tue, 22 Jul 2003 16:47:17 +0200 (CEST)
Received: from istanbul.magic.fr (istanbul.magic.fr [62.210.190.11])
      by xxx (Postfix) with ESMTP id 1FEBD1B4A0
      for <mayer@xxx>; Tue, 22 Jul 2003 16:41:06 +0200 (CEST)
Received: from xxx (xxx [80.118.11.215])
      by amsterdam.magic.fr (8.11.7/8.11.2) with ESMTP id h6MEhsO25425
      for <mayer@xxx>; Tue, 22 Jul 2003 16:43:54 +0200 (CEST)
Received: by xxx (Postfix, from userid 501)
      id 44FE44112; Tue, 22 Jul 2003 16:46:16 +0200 (CEST)
Date: Tue, 22 Jul 2003 16:46:16 +0200
From: Bernard MAYER <mayer@xxx>
To: mayer@xxx
Subject: test avec mutt
Message-ID: <20030722144616.GA7818@xxx>
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Disposition: inline
User-Agent: Mutt/1.3.27i
X-UIDL: K8F!!GhI!!3IG"!#nF!!
```

```
Ceci est le corp du message,
Bonne reception.
```

2.3Description des mécanismes mis en jeu.

Précisons le sens de quelques termes:

- MUA
  L'outil qui permet de rédiger/déposer et de récupérer/lire son courrier est appelé un MUA, pour Mail User Agent. On le trouve également sous l'appellation « client de messagerie » . Les outils les plus courants sont, entre autres; Outlook Express, Mozilla Messenger, Netscape Messenger, Eudora, Mutt, Pegasus, Sylpheed, Kmail, Evolution, ...
- MTA
  Les « centres de tri » sont appelés MTA pour Mail Transfert Agent. On les trouve également sous l'appellation « serveur de messagerie » . Le plus répandus, et un des plus anciens est Sendmail (Le plus complexe aussi . Une légende raconte que l'administrer rend fou ...). Il existe aussi Postfix, Exim, Qmail, Exchange, et bien d'autres.
- MDA
  Le dépôt dans les boites aux lettres est fait par le MDA, pour Mail Delivery Agent. Cette fonction peut être intégré dans le MTA. Certains systèmes sophistiqués utilisent des programmes séparés pour ce faire. Le plus courant s'appelle Procmail.
- SMTP
  Le dépôt et l'acheminement du courrier est réalisé à l'aide du protocole SMTP, pour Simple Mail Transfer Protocol . Il est défini par le RFC 821.
- POP
  Pour la récupération du courrier dans la boite aux lettres, le protocole le plus courant est POP3, pour Post Office Protocol version 3. Il est défini par le RFC 1939.

3Définition du périmètre:

Les compétences de l'équipe « technique réseau » de la délégation TIC au sein du Conseil Régional d'Aquitaine peuvent porter sur deux axes, complémentaires et non concurrents.

- Filtrer les courriers indésirables,
- Ne pas fournir de moyens aux spammeurs.

Définissons ce que l'on entend par ces deux termes.

- Spam:
  Le spam est l'action d'envoyer un courrier à des personnes qui ne l'ont pas sollicité, et dont l'utilité n'est pas avérée. La publicité rentre dans ce cadre. Cela provoque une perte de temps non

négligeable pour l'utilisateur.

- Relayage abusif:
  Le relayage abusif consiste à utiliser les ressources d'un système dont on n'est pas responsable. En d'autres termes, en utilisant les machines d'une entreprise à son insu, par exemple, ses serveurs de messagerie, en usurpant son identité et en « pillant » ses ressources.

4Solutions:

4.1Pour le spam, la partie « contenu »

La difficulté à déterminer si un courrier est ou n'est pas un spam, provient de la nécessité de comprendre la signification du message. Malgré cela, nous sommes aidés par le fait que les émetteurs de spam (les spammeurs) utilisent un ensemble d'astuces bien connues. Par exemple, ils n'utilisent pas certains lecteurs de courrier, tel Mutt. Le message est bien souvent au format HTML. Il est souvent question d'argent, de sexe, de ventes de médicaments, ...

Bref, un ensemble de faits, qui, connus et descriptibles, sont exploitables par un biais logiciel.

Récemment, (2002) notre arsenal a été renforcé par Paul Graham avec un algorithme « génétique » qui introduit une technique de filtrage dite « Bayesian » ou « filtre de Bayes »[2]. Cette méthode s'appuie sur les occurrences et la disposition des mots, et la possibilité de les voir apparaître dans des courriers légitimes, ou dans les courriers non sollicités.

A partir d'un petit outil écrit en Perl par Mark Jeftovic, une équipe de développeurs ont élaboré un outil nommé « SpamAssassin »[3]. Le corpus de ce chapitre va décrire la mise en oeuvre de SpamAssassin.

Toutefois, l'erreur est possible, et le sera toujours. Ainsi, l'outil ne fera que marquer[4] les messages afin qu'ils parviennent quand même aux abonnés du service de messagerie. Ils pourront tout à loisir, configurer leur lecteur de courrier pour que les messages marqués soit mis à l'écart, afin de les soulager d'une corvée de tri, et sauvegarder une éventuelle erreur, d'un courrier normal, qui serait marqué comme spam.. Nous reviendrons sur cette partie importante de l'usage de SpamAssassin à la fin de cette section.

4.2Pour le relayage; la partie « enveloppe »

2 Pour avoir plus d'informations, mais en anglais, voir `http://www.paulgraham.com/spam.html`
3 Voir `http://www.spamassassin.org`
4 Un terme convenu pour désigner ses marques est « tag »

Il existe des solutions simples et efficaces. Elles consistent d'une part à vérifier les éléments constitutifs du message, enveloppe et en-tête. Il est possible de contrôler les parties constitutives obligatoires de cette enveloppe, comme la stricte conformité de la syntaxe de l'enveloppe, l'existence réelle du serveur qui envoie le message (par son inscription dans un DNS[5]), sa présence ou absence de listes de spammeurs connus (« black lists »), voir de moyens encore plus coercitifs, mais impliquant des restrictions quant à l'usage de la messagerie. Citons le cryptage entre serveurs, l'acceptation d'une liste restreinte de correspondants, ...

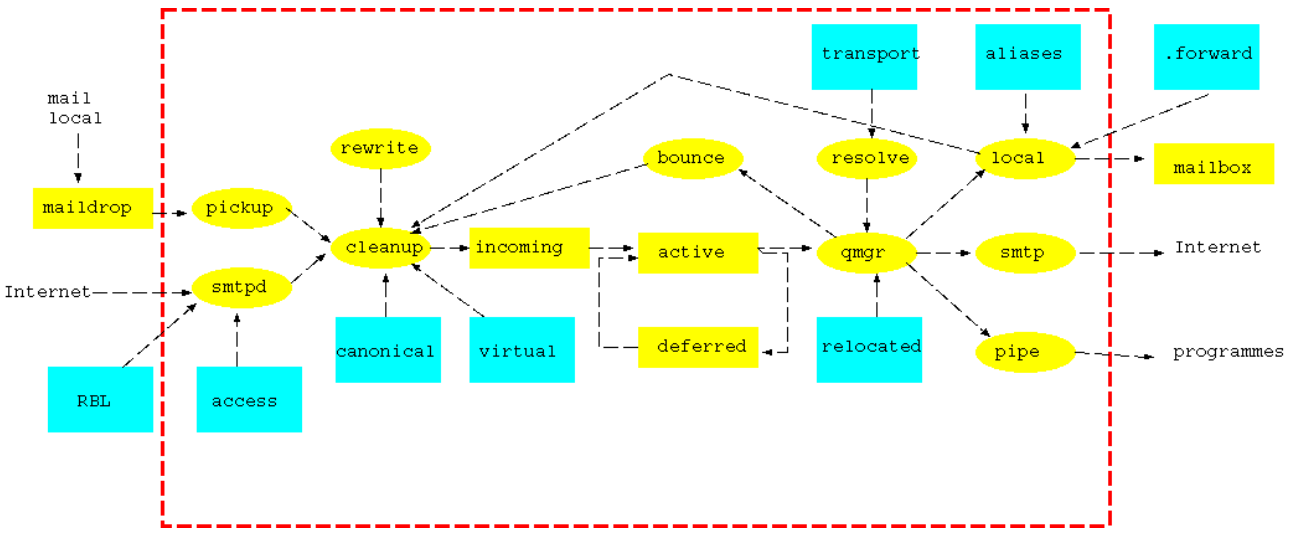
Quoi qu'il en soit, la mise en oeuvre de ses moyens est réalisée par le/les serveurs de messagerie eux-mêmes.

5Cadre de la mise en pratique:

Quel que soit le cas de figure, le point focal des opérations est le coeur du système de courrier électronique. Ici, c'est un serveur hébergeant un MTA[6] Postfix[7]

Ce serveur dispose de toutes les conditions à réunir pour obtenir un fonctionnement optimal et pérenne. Il est très sûr, très efficace, très robuste, rapide et performant. Son architecture modulaire autorise l'adjonction de programmes supplémentaires. Toutefois, une démarche de sécurité cohérente implique des restrictions. Ces restrictions qui ne doivent pas ajouter d'autres contraintes. L'architecture imaginée au Conseil Régional d'Aquitaine répond à ces prescriptions.

Un seul serveur de courrier est connu et accessible depuis l'extérieur. Il possède les outils de filtrage de spam, il en héberge les paramètres et les règles de filtrage. Il est protégé contre le relayage, et lui-même l'interdit. Le dispositif est flexible, car on peut très bien déléguer certaines fonctions à d'autres machines.



5 Domain Name Service, voir rfc 1034
6 Mail Transfert Agent, routeur/serveur de messagerie.
7 Voir `http://www.postfix.org`

Voici le schéma structurel du système de messagerie Postfix.

Ici sont représentés en jaunes les processus de Postfix, et, en bleu, les fichiers dont ces processus dépendant. Le cadre en pointillés rouge limite ce qui est propre du serveur Postfix. On peut très bien distinguer le cheminement du courrier, de gauche à droite.

6Mise en oeuvre:

6.1Protection par Postfix

Le serveur de messagerie Postfix étant pensé à l'origine pour offrir une sécurité maximale, il n'y a que quelques paramètres de configuration à manipuler, dont la plupart sont activés par défaut.

La documentation de Postfix décrit par le menu la marche à suivre.

Tout ce qui nous intéresse est contenu dans le principal fichier de configuration de Postfix; `/etc/postfix/main.cf`.

```
##    Il convient d'abord de signifier à Postfix son identité, et ce qu'est le
domaine a gérer.
myhostname = host.domain.name
mydomain = domain.name
mydestination = $myhostname, mail.$mydomain, smtp.$mydomain
##    et comment completer les noms locaux
myorigin = $myhostname

##    Gestion des boites aux lettres
home_mailbox = Mailbox
mail_spool_directory = /var/spool/mail
mailbox_command = /usr/bin/procmail -a « $extension »

##    Banniere de connexion
smtpd_banner = $myhostname ESMTP $mail_name

##    Restreindre aux réseaux suivants
mynetworks = 10.1.0.0/24, 127.0.0.0/8
##    Brider le relayage au seul domaine géré
relay_domains = $myhostname

##    Ne pas accepter sans resolution de nom inverse, ni ceux contenus dans le
fichiers « access »
smtpd_client_restrictions = reject_unknow_client, hash:/etc/postfix/access

##    Rejeter si domaine inscrit en blacklists, et emplacement des blacklists
smtpd_client_restrictions = reject_maps_rbl
maps_rbl_domains = rbl.maps.vix.com, dul.maps.vix.com

##    Verifications particulieres sur le message, d'apres fichiers header_checks
```

```
et body_checks
header_checks = regexp:/etc/postfix/header_checks
body_checks = regexp:/etc/postfix/body_checks
```

Voici un exemple de fichier contenant les régles de contrôle pour Postfix.

```
##   /etc/postfix/header_checks
/^from: .+@*mechant.com/ REJECT
/^subject: *viagra*/i REJECT
```

6.2Protection par SpamAssassin; installation

SpamAssassin possède plusieurs modes de fonctionnement. Tous commencent par l'installation du produit. Pour ce faire :

- Il faut donc le récupérer sur son site, sous la forme de « tarball », c'est à dire un fichier archive . compressé (.tar.gz). Nous avons utiliser la version 2.55 .
- Désarchiver dans un répertoire sans importance, par exemple `/tmp`. Il y sera créé une arborescence Mail-SpamAssassin-<version>. Pour rappel, la commande qui permet ceci est
  `$ tar –xzvf nom_du_fichier.tar.gz`
- S'y déplacer, et lire attentivement les fichiers README, INSTALL, et USAGE. Si nécessaire, prêter attention au notes concernant Perl 5.8
- SpamAssassin est écrit en Perl. Il y a peut-être besoin de modules absents du système. Le fichier INSTALL les énumère. Pour tester leur présence, ou leur absence, le plus simple est de les utiliser! Utilisons la possibilité d'exécuter directement du code Perl par la ligne de commande.
  `$ perl –e 'use File::Spec;'`
- La présence du module ne fait rien retourner, alors que son absence produit un message explicite. Les modules absents seront installés par l'interface intégré au CPAN, comme indiqué dans le fichier INSTALL
  `# perl –MCPAM –e shell`
  `> o conf prerequisites-policy ask` (facultatif)
  `> install Net::DNS`
  `.../...`(Affichage du déroulement)
  `> install autre-module`
  `> exit`
- Puis, vient la fabrication du programme et les compilations des modules additionnels, que nous allons réaliser dans le répertoire précédemment créé.
  `$ perl Makefile.PL`
  .../...
  `$ make`
  .../...
  Il convient de suivre le déroulement des opérations, car, si un problème survient, le message d'erreur est assez clair, mais indispensable pour déterminer une solution. Il est possible de passer des options à cette étape du processus, afin de changer, par exemple le répertoire dans lequel sera disposé le programme, ou bien l'emplacement des fichiers de configuration.

- L'installation impose d'avoir les droits suffisants
  `# make install`
- Réalisons les premiers tests.
  Le programme est livré avec deux fichiers reproduisant des messages de tests, un figurant un spam, et l'autre non.. Il est également possible d'utiliser de vrais messages. Le test se fait simplement, en invoquant SpamAssassin en ligne de commande.
  ```
  $ spamassassin –t < sample–nonspam.txt > nonspam.out
  $ spamassassin –t < sample–spam.txt > spam.out
  ```
  Vérifier, par un simple affichage, que des informations supplémentaires sont apparus dans les messages, tant au niveau du contenant (l'en-tête), que du contenu (le corps).
- A partir de ce moment, il est possible de configurer SpamAssassin, afin que le résultat du filtrage corresponde aux desiderata. La réalisation en cours étant destinée, non à un utilisateur particulier, mais à l'ensemble des utilisateurs,[8] il est de bon ton que la mise en place s'effectue de façon transparente pour eux. C'est à dire que la configuration de SpamAssassin soit faite de telle sorte que ni le corps ni le sujet du message ne soit modifié. Le marquage sera réalisé uniquement au niveau de l'enveloppe. Le fichier de configuration adéquat est en annexe.

6.3Protection par SpamAssassin; mise en oeuvre

Quelqu' esprit curieux, tordus diront certains, aura remarqué que le programme SpamAssassin est écrit en langage Perl. Ce qui permet souplesse, adaptabilité, et plein d'autres avantages. En contrepartie, Cela signifie également beaucoup de mémoire nécessaire, et durée de chargement non négligeable. Pour la mémoire, la commande `top` nous éclaire sur la conssomation du programme, et le prix actuel des composants fait que nous aurons tôt fait d'en ajouter si besoin est. Par contre, pour la durée de chargement, une bonne dose de ruse d'administrateur système s'impose. L'exploration de l'arborescence des fichiers sources de SpamAssassin nous fait découvrir une branche `/spamd/` où, comme de bien entendu, nous trouvons le fichier `README.spamd`, que nous lisons. Nous apprenons que, dans ce répertoire, existe une version « daemonizée » de SpamAssassin, spécialement développée afin d'améliorer les performances. Et oui, Perl est un langage qui est particulier au sens où il est interprète un fichier texte à chaque exécution, pour le compiler. En cela, c'est une perte de temps énorme.

Nous avons vu qu'il est appelé pour un et un seul message. Ce qui sous-entend que chaque message va provoquer une interprétation/compilation. La solution proposée par spamd permet de ne charger le programme qu'une seule fois, et qu'il attende en mémoire, non qu'on lui passe un flot de données sur l'entrée standard, mais qu'on lui passe via un socket réseau. Le seul temps d'exécution ne sera désormais constitué que du temps de traitement du message.

Pour pouvoir communiquer avec spamd, Postfix appellera un petit programme, écrit en C, optimisé en vitesse. Ce programme s'appelle spamc. L'architecture modulaire de Postfix nous montre que, pour ne traiter que les messages en provenance de l'extérieur, il va falloir intervenir au niveau du daemon smtpd, qui écoute sur le port dédié à smtp (le port 25). Le fichier `/etc/master.cf` pilote les daemons de Postfix. Dans l'arborescence des fichiers sources de Postfix, nous trouvons le fichier FILTER_README, qui nous détaille comment installer un filtre dans l'architecture Postfix.

Deux solutions sont présentées.

8 Ce type d'installation est communément appellée « system-wide »

La première se fait par l'appel d'un programme externe et passage de paramètres, la seconde est réalisée par une communication IP entre Postfix et un autre programme.

Il convient donc d'utiliser la première solution, via un pipe qui lancera un script de lancement de spamc, qui enverra le message à spamd, qui lui, le traitera et le retournera au script. Le script réinjectera le message dans Postfix, comme un mail local. Un script, un programme sont des fichiers, logés sur des disques durs. Aussi performant soit-il, un disque dur sera toujours beaucoup plus lent que de la mémoire. Ne risquons nous pas de d'amoindrir de ce fait les gains escomptés par spamd ? Une voie de contournement se présente par la possibilité de mise en place d'un « ramdisk [9]» . Le ramdisk hébergera donc le script de lancement de spamc, spamc lui même, et les programmes nécessaires (sendmail, sh et cat dans notre cas).

Apportons quand même quelque précisions sur le ramdisk. Comme tout système de fichiers, il doit être monté dans le système. Cette opération est réalisé par le script d'initialisation. Comme tout montage, il possède un point de montage, ici `/etc/postfix/filter`. Nous allons utiliser le fait que le contenu d'un point de montage n'est plus visible, à partir du moment où le montage est réalisé. Ceci va permettre de démarrer le dispositif, même si la création du ramdisk échoue. Nous allons réaliser une fois pour toute, des liens symboliques entre les exécutables qui seront copiés dans le ramdisk, et le point de montage.

```
# ln -s /bin/sh /etc/postfix/filter
# ln -s /bin/cat /etc/postfix/filter
# ln -s /usr/sbin/sendmail /etc/postfix/filter
# ln -s /usr/bin/spamc /etc/postfix/filter
# chown filter.filter /etc/postfix/filter
```

Trêve de bavardage, place à l'action !

Il va donc falloir ajouter au système un utilisateur dédié à cette tâche. Ce sera l'utilisateur `filter`.

SpamAssassin version daemon doit être lancé à chaque démarrage du système, et stoppé à chaque arrêt. Cela implique un script de contrôle , et les liens adéquats dans l'arborescence des fichiers d'initialisation. Le contenu de spamd.sh est en annexe. Les valeurs numérique des scripts K (Kill) et S (Start) n'est qu'indicative, et doit être adaptée à chaque système.

- `/etc/rc.d/init.d/spamd.sh`
- `/etc/rc.d/rc0.d/K37spamd.sh`
- `/etc/rc.d/rc1.d/K37spamd.sh`
- `/etc/rc.d/rc2.d/S69spamd.sh`
- `/etc/rc.d/rc3.d/S69spamd.sh`
- `/etc/rc.d/rc4.d/S69spamd.sh`
- `/etc/rc.d/rc5.d/S69spamd.sh`
- `/etc/rc.d/rc6.d/K37spamd.sh`

Il convient de modifier la configuration de Postfix. Cette fois, c'est le fichier /etc/master.cf que nous allons modifier. Ce fichier contrôle le mode de fonctionnement de chacuns des programmes qui animent Postfix. On y ajuste simplement la ligne suivante.

```
##  le filtre spamc en ramdisk !
filter  unix    -       n       n       -       5       pipe    user=filter
argv=/etc/postfix/filter/spamc.sh -f ${sender} -- ${recipient}
```

(Hormis la ligne de commentaire, celle commençant par filter est d'un seul tenant)

L'arrêt puis la relance de Postfix, s'effectue simplement par:

9 Un ramdisk est un système de fichiers logé en mémoire vive, et non sur disque dur.

```
/etc/init.d/postfix stop
/etc/init.d/spamd.sh start
/etc/init.d/postfix start
```

Voici deux traces du fonctionnement, la première pour les courriers identifiés comme spam, la seconde pour les ceux n'ayant pas atteint le score de 4:

```
Sep  1 12:01:58 mail spamd[6759]: identified spam (6.6/4.0) for filter:1031 in
0.0 seconds, 4312 bytes.
Sep  1 12:02:43 mail spamd[6844]: identified spam (9.9/4.0) for filter:1031 in
1.0 seconds, 5097 bytes.
Sep  1 12:04:07 mail spamd[6955]: identified spam (6.2/4.0) for filter:1031 in
0.0 seconds, 6160 bytes.
Sep  1 12:04:33 mail spamd[6995]: identified spam (14.2/4.0) for filter:1031 in
0.0 seconds, 1583 bytes.
Sep  1 12:04:52 mail spamd[7016]: identified spam (6.4/4.0) for filter:1031 in
0.0 seconds, 3803 bytes.
Sep  1 12:05:16 mail spamd[7127]: identified spam (16.1/4.0) for filter:1031 in
0.0 seconds, 1899 bytes.
Sep  1 12:05:26 mail spamd[7147]: identified spam (10.4/4.0) for filter:1031 in
0.0 seconds, 1683 bytes.
Sep  1 12:06:23 mail spamd[7248]: identified spam (16.2/4.0) for filter:1031 in
0.0 seconds, 2432 bytes.
Sep  1 12:07:50 mail spamd[7382]: identified spam (6.9/4.0) for filter:1031 in
0.0 seconds, 4594 bytes.


Sep  1 12:00:56 mail spamd[6705]: clean message (2.3/4.0) for filter:1031 in 0.0
seconds, 14565 bytes.
Sep  1 12:01:21 mail spamd[6735]: clean message (0.7/4.0) for filter:1031 in 0.0
seconds, 3111 bytes.
Sep  1 12:02:10 mail spamd[6775]: clean message (-3.4/4.0) for filter:1031 in
0.0 seconds, 717 bytes.
Sep  1 12:02:12 mail spamd[6781]: clean message (-3.4/4.0) for filter:1031 in
0.0 seconds, 725 bytes.
Sep  1 12:02:13 mail spamd[6789]: clean message (-2.0/4.0) for filter:1031 in
0.0 seconds, 716 bytes.
Sep  1 12:02:26 mail spamd[6801]: clean message (-3.7/4.0) for filter:1031 in
1.0 seconds, 3102 bytes.
Sep  1 12:02:29 mail spamd[6810]: clean message (-0.7/4.0) for filter:1031 in
0.0 seconds, 685 bytes.
Sep  1 12:02:30 mail spamd[6817]: clean message (-0.6/4.0) for filter:1031 in
0.0 seconds, 795 bytes.
Sep  1 12:02:34 mail spamd[6826]: clean message (-0.5/4.0) for filter:1031 in
1.0 seconds, 3689 bytes.
Sep  1 12:02:35 mail spamd[6833]: clean message (2.0/4.0) for filter:1031 in 0.0
seconds, 14553 bytes.
Sep  1 12:02:56 mail spamd[6873]: clean message (0.4/4.0) for filter:1031 in 1.0
seconds, 20554 bytes.
Sep  1 12:03:12 mail spamd[6893]: clean message (-4.7/4.0) for filter:1031 in
0.0 seconds, 3767 bytes.
Sep  1 12:04:04 mail spamd[6943]: clean message (-1.7/4.0) for filter:1031 in
1.0 seconds, 1293 bytes.
Sep  1 12:04:13 mail spamd[6966]: clean message (0.2/4.0) for filter:1031 in 0.0
seconds, 3026 bytes.
Sep  1 12:04:59 mail spamd[7030]: clean message (-2.6/4.0) for filter:1031 in
1.0 seconds, 6992 bytes.
```

Les modifications apportées aux courriers, respectivement spam et non spam:

```
X-Spam-Status: Yes, hits=5.0 required=4.0
```

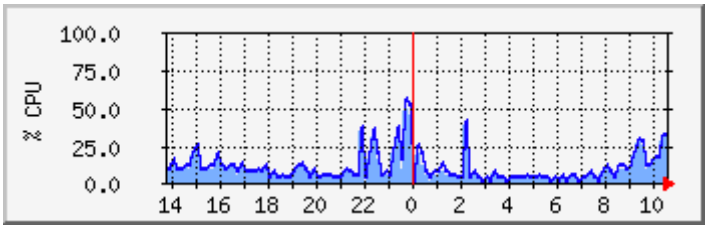
```
        tests=BASE64_ENC_TEXT,BAYES_90,HTML_50_60,MIME_HTML_ONLY
        version=2.55
X-Spam-Level: *****
X-Spam-Checker-Version: SpamAssassin 2.55 (1.174.2.19-2003-05-19-exp)
X-Spam-Report:   ---- Start SpamAssassin results
  5.00 points, 4 required;
  *  3.0 -- BODY: Bayesian classifier says spam probability is 90 to 99%
  [score: 0.9716]
  *  0.1 -- BODY: Message is 50% to 60% HTML
  *  1.8 -- RAW: Message text disguised using base-64 encoding
  *  0.1 -- Message only has text/html MIME parts
  ---- End of SpamAssassin results
X-Spam-Flag: YES
X-Spam-Status: No, hits=0.0 required=4.0
        tests=none
        version=2.55
X-Spam-Level:
X-Spam-Checker-Version: SpamAssassin 2.55 (1.174.2.19-2003-05-19-exp)q
```

Il a été mis à disposition des utilisateurs 2 boites à lettres techniques, nommées spamoui et spamnon. Elles sont destinées à recevoir les courriers mal identifiés. C'est à dire reconnu comme spam alors qu'ils n'en sont pas, et vice versa. Après vérification, on invoque un exécutable de SpamAssassin, sa-learn, en lui passant en paramètre le nom de la boite à lettres (--mbox spamoui) et le type de message qu'elle contient (--spam). On fait la même opération avec la boite à lettres spamnon, mais avec le paramètre nospam. Ainsi, on enrichit la base de connaissance de SpamAssassin, afin qu'il ne fasse plus les mêmes erreurs.

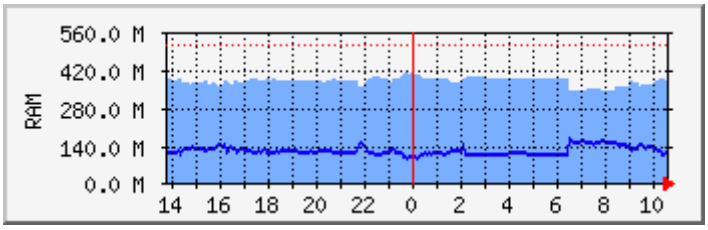
7Conclusion:

Au bout d'une semaine de fonctionnement faisons les comptes. Pendant une semaine de congés d'été, le MTA a traité 57000 messages. 23200 étaient en provenance de l'extérieur. Sur ces 23200, 10200 ont été marqués comme spam, et consécutivement 13000 non. Cela représente 44% de courriers non sollicités.
Après vérification, il est possible d'avancer que le taux de discrimination est supérieur à 95% !

La charge supplémentaire que supporte la machine est négligeable, comme en témoigne les graphiques suivants :



Moyenne sur 1 minute de la charge CPU



Moyenne sur 1 minute de la mémoire utilisée

Le procédé a été mis en route à 19 heures. Ni les relevés faits in-vivo à ce moment, ni les relevés graphiques ci dessus n'indiquent une modification sensible des indicateurs de la machine.

Le bilan est donc on ne peut plus satisfaisant. Je regrette néanmoins de n'avoir eu le temps d'élaborer une solution de rapport, qui aurait été élaborée à partir les fichiers de traces, et présentée par une page web, dans le prolongement des divers outils déjà existants au Conseil Régional.