SPAM ASSASSIN

Mathieu KERN - Aymeric HINDERCHIETTE

Décembre 2014

Table des matières

Première partie
Présentation de SpamAssassin

1 Problématique

Le mail (ou courriel) est aujourd'hui le moyen privilégié de communication à travers le monde. Massivement utilisé, d'une certaine fiabilité et éprouvé par des décennies d'utilisation il reste le moyen le plus répandu pour les communications entres les personnes. Malheureusement, mail est également aujourd'hui synonyme de spam, ces messages indésirables qui s'entassent dans nos boites mails. C'est ici qu'entre en jeu SpamAssassin.

1.1 Le SPAM

Avant de poursuivre sur SpamAssassin, rappelons concrètement ce qu'est le SPAM et ce qu'il implique.

Comment reconnaître un SPAM :

- De par sa nature, un SPAM n'est pas désiré par l'utilisateur qui le reçoit.
- La réception d'un SPAM résulte d'un envoi massif : une machine (souvent un bot) envoie le même message à plusieurs destinataires sans aucun discernement. Cela s'oppose aux messages ciblés par exemple de commerçant, qui n'envoie que à leurs prospects (à but publicitaire ou informatif).
- Son contenu n'est pas destiné spécifiquement à l'utilisateur (chaque personne reçoit le même contenu).
- Une importante liste de destinataires.
- Entête des messages souvent corrompues ou ne respectant pas les normes.

Statut légal La loi pour la confiance dans l'économie numérique du 21 juin 2004 contient une transposition de la directive européenne du 12 juillet 2002 [1] relative à la protection de la vie privée dans le secteur des communications électroniques :

> Est interdite la prospection directe au moyen d'un automate d'appel, d'un télécopieur ou d'un courrier électronique utilisant, sous quelque forme que ce soit, les coordonnées d'une personne physique qui n'a pas exprimé son consentement préalable à recevoir des prospections directes par ce moyen.

1. Le principe introduit figurera également à l'article L.34.5 du code des postes et des communications électroniques

Les SPAM sont donc connus du droit français et encadrés par des textes spécifiques.

2 Le projet

2.1 Informations

Développeur	Apache Software Foundation	
Langage	Perl Dernière version	3.4.0 (11 février 2014) [+/-]
Environnements	Multiplate-forme	
Type	Anti-spam	
Licence	Licence Apache 2.0	

SpamAsassin est donc aujourd'hui sous le giron de la Apache Software Foundation, organisation à but non lucratif qui s'occupe également du serveur Apache, Logiciel de distribution de contenu WEB le plus utilisés au monde. Elle gère également 150 autres projets. EN outre tout ses projets sont distribués sous sa propre Licence, la licence Apache(actuellement en 2.0) , qui est compatible GPL v3. Cette licence met l'accent sur le copyright tout en restant bien sur une licence libre. Les objectifs principaux de la Fondation sont de protéger juridiquement le travail des contributeurs et d'empêcher que la marque Apache soit utilisée illégalement.

Le projet SpamAssassin est actif depuis plus d'une décennie et est constamment en développement pour s'adapter aux développements des méthodes qu'utilisent les spammeurs. C'est en outre le programme anti-spam le plus utilisé à cause de son efficacité.

2.2 Développement

SpamAssasin contient environ 300 000 lignes de codes ce qui en fait un très gros projet(Graphique 1). Le projet est à maturité et il ne grossit plus depuis plusieurs années, les développeurs se concentrant sur l'optimisation du code existant. Il y actuellement 23 développeurs pricnipaux, avec une répartition des lignes codes assez inégales, notamment deux développeurs qui ont fait la majorité du code(Tableau 1). Mais vu que c'est un projet libre et open source, chacun est libre de contribuer et forker le projet. Cela concerne aussi bien les particuliers que les entreprises (En respectant bien sur les restirction de la licence Apache).

2.3 Qu'est ce que SpamAssassin

SpamAssassin est un programme écrit en PERL dont le but est de filtrer activement les Emails en se basant sur des mécanismes internes. SpamAs-

Author Id	Changes	Lines of Code	Lines per Change
Totals	26092 (100.0%)	1403447 (100.0%)	53.7
jm	8136 (31.2%)	721593 (51.4%)	88.6
spamassassin role	7997 (30.6%)	463448 (33.0%)	57.9
axb	741 (2.8%)	54525 (3.9%)	73.5
mmartinec	1779 (6.8%)	32348 (2.3%)	18.1
felicity	1625 (6.2%)	29134 (2.1%)	17.9
kmcgrail	605 (2.3%)	21294 (1.5%)	35.1
quinlan	1100 (4.2%)	19583 (1.4%)	17.8
parker	309 (1.2%)	10407 (0.7%)	33.6
khopesh	1369 (5.2%)	10296 (0.7%)	7.5
dos	427 (1.6%)	8427 (0.6%)	19.7
jhardin	1021 (3.9)	6845 (0.5%)	6.7
wtogami	137 (0.5%)	6470 (0.5%)	47.2
hstern	63 (0.2%)	6029 (0.4%)	95.6
sidney	275 (1.1%)	3552 (0.3%)	12.9
jquinn	28 (0.1%)	2181 (0.2%)	77.8
mss	133 (0.5%)	2072 (0.1%)	15.5
hege	95 (0.4%)	1931 (0.1%)	20.3
duncf	109 (0.4%)	1892 (0.1%)	17.3
jgmyers	66 (0.3%)	508 (0.0%)	7.6
smf	35 (0.1%)	499 (0.0%)	14.2
maddoc	11 (0.0%)	319 (0.0%)	29.0
fanf	12 (0.0%)	49 (0.0%)	4.0
kb	19 (0.1%)	45 (0.0%)	2.3

TABLE 1 – Statistique des développeurs du projets
Source [2]

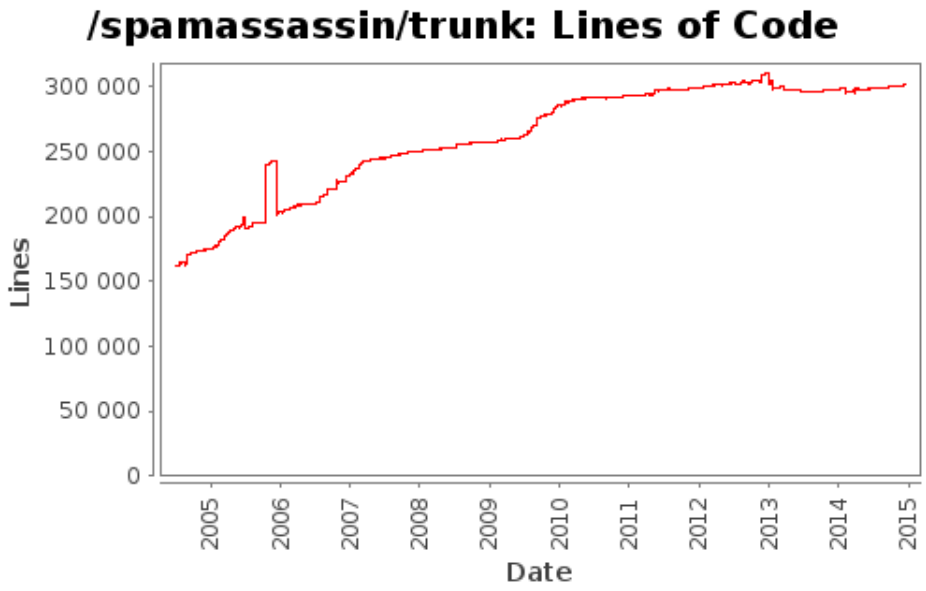

FIGURE 1 – Évolution du nombre de ligne de codes

sassin n'effectue aucune action envers les mails, il ajoute seulement des informations personnalisés qui peuvent être utilisée par d'autres programmes pour effectuer des actions sur les mails (les ranger des dossiers distincts, les supprimer, les bloquer, ...)

Il peut être utilisé de plusieurs manières :

— En mode client, lancé à chaque fois que l'on fait appel à lui
— En mode demon grâce à *spamd* , les appels au demon étant fait avec l'utilitaire *spamc*.
— Comme une interface de programation : des programmes qui nécessitent des fonctionnalités de filtrage de SPAM peuvent s'interfacer avec SpamAssin pour construire des solutions utilisant ses fonctionnalités

Deuxième partie
Ses fonctionnalités

3 Comment il filtre

SpamAssassin reçoit des mails que lui redirigent d'autres programmes, y effectue des tests pour déterminer si ce sont des SPAMS, puis renvoie les mails testés au programme qui les as envoyés.

3.1 Les test de SpamAssassin

Champs d'entête En se basant sur la forme des entêtes et en les comparant avec des schéma connus par SpamAssasin. En effet on peut se base sur la façon dont certains systèmes de SPAMs construisent leurs messages pour les filtrer.

Corps du message Bien sur SpamAssasin permet de filtrer les mails suivant les mots et expressions qu'ils contiennent. “ceci n'est pas un SPAM”, “Bonjour je suis une princesse d'un royaume africain”, “Venez chechez votre lot”, . . .sont des expressions typiques pour des SPAMs.

Filtre bayésien(détail plus loin 4) Filtrer les entêtes et le corps d'un message résultera toujours en de multiples faux positifs. C'est ici que le filtrage bayésien se révèle intéressant car il va prendre en considération ce que l'on considère comme SPAM et non SPAM soit des “bon mails” (“HAM” en anglais). Il va ensuite utiliser les répertoires de SPAMs connu et de “HAM“ connus, pour y identifier les mots et phrases (Définits comme ”Tokens“ en anglais) qui n'apparaissent que dans les SPAMs et que dans les ”HAMs“. Un token SPAM trouvé résultant d'une hausse du score (voir 4) SPAM, un token résultant en une baisse de ce niveau. Ce filtrage permet d'être plus précis et d'éviter les faux positifs, en ne se basant pas juste sur un mot ou une phrase mais des ensembles.

List noire/blanche automatique SpamAssassin garde automatiquement une liste blanches des expéditeurs des mails. Pour chaque nouveau mail le programme compare le mail précédent provenant de la même adresse mail et adresse IP. Comme précédemment si une adresse email a envoyé un SPAM, ce nouveau mail verra son score abaissé. A l'inverse si c'était un bon mail son score se verra baisser.

Liste noire/blanche manuelle Il est tout à fait possible de définir ses propres listes, en autorisant ou interdisant des mails de certaines adresses

Signalements En utilisant des signatures établis à partir de mails signalés par les utilisateurs. Il y a notamment les projets DCC, Pyzor, et Razor2 qui possèdent des bases de données de mails signalés comme SPAM. SpamAssassin va ainsi demander à ces bases si les mails qu'il reçoit sont présents dans leurs donnée.

DNS blocklists Ce sont des des bases de données contenant des adresses IP signalées comme expédiant du SPAM ou mal configurée(Par exemple en étant un relais ouvert). Également sont pris en compte les IP de particuliers (considérant qu'il y a peu de chance qu'un particulier envois directement des mails sans passer par son FAI. Ces signalements vont être pris en compte par SpamAsassin pour le score des mails. Il intègre nativement quelque une de ces listes.

Caractères et langues on peut spécifier des caractères et langues comme SPAM.

C'est ces ensembles de règles qui fonctionnant conjointement permettent à SpamAssassin de garantir un haut niveau de fiabilité de détection des SPAM, un test pouvant ne pas fonctionner mais sera contrebalancé par les autres.

Spam Assassin effectue sur chaque mail qui lui est donné à traiter une série de test, qui vont ensuite donner lieu à un score, qui sera indiqué dans un entête si il est considéré comme SPAM. Ce résultat sera ensuite utilisé par d'autres programmes pour déterminer des actions à entreprendre.

4 Le score

C'est la base du signalement des SPAM de SpamAsassin. Un mail après avoir subi des test différents se voit attribuer une note. Cette note permet ensuite de définir des actions à effectue. Quand un mail est considéré comme SPAM, SpamAssassin ajoute ses propres entêtes (Exemple **??**) :

— "-Spam-Level : " : Renseigne le score (ici 9)

— -Spam-Status : Yes, score=9.0 required=5.0 tests=BAYES_99,FROM_EXCESS_BASE64,FR_H
... : Indique des informations relatives aux test effectués et la configuration)

. L'utilisateur peut paramétrer ce score pour définir une marge de définitions des SPAMs(valeurs "require"). Une fois ce score ateint le mail est considéré comme SPAM. C'est ensuite aux autres programmes d'utiliser ces données. Par exemple on pourrait configurer un MUA pour qu'il sépare les SPAM par score suivant des filtres définit.

5 Le filtre bayésien

Une fonctionnalité qui est une des plus importantes caractéristiques de SpamAssassin est sans doute son filtre bayésien [3]. Il va permettre d'établir une connaissance des éléments qui constituent un SPAM et ceux qui n'en sont pas.

Utilisation par SpamAssassin Comme annoncé plus haut, ce type de filtre fait partis de son arsenal de test anti-spam. Il va effectuer des tests en utilisant les algorithmes de Bayes pour tester des mots et expressions(des"Tokens"), et baisser le score globale du mail si des tokens SPAM sont trouvés. A l'inverse si des des tokens "HAM" (devant se trouver dans de bon mail) sont trouvés il va abaisser le score du mail, ce qui peut résulter en des score négatifs.

Sa learn [4] Mais ce que permet surtout ce filtre c'est d'apprendre. En effet il va s'adapter aux utilisateurs. Les utilisateurs peuvent déclarer un courrier comme SPAM ou "HAM" et il sera alors pris en compte par le filtre bayésien de SpamAssasin. Le programme s'utilise de cette maniéré :

- sa-learn –spam /Chemin/du dossier
- sa-learn –ham /Chemin/dudossier

Il faut pour garantir l'efficacité et coller aux besoins des utilisateurs que les mails soient ceux de l'utilisateur et représentatif de ce qu'il reçoit régulièrement. En outre pour garantir le meilleur taux de réussite dans la découverte de SPAM, il faut lui faire apprendre plusieurs milliers de mails. Ainsi C'est aux utilisateurs de définir ce qu'ils considèrent comme SPAM et non spam. Par défaut les tokens sont stockées dans le répertoire des utilisateurs. Pour définir le filtre pour ensembles utilisateurs il faut ajouter au *local.cf* :

```
bayes\_path /var/spamassassin/bayes\_db/bayes
bayes\_file\_mode 0777}
```

Il faudra veiller à ce que sa-learn puisse régulièrement apprendre les mails des utilisateurs (en l'intégrant par exemple dans un script lancé régulièrement). Les tokens que génère sa-learn sont stockés dans des bases de données, qui sont par défaut distincte pour chaque utilisateur. Ces Dernière pour éviter de devenir trop volumineuse intègre des mécanismes qui régulièrement vont éliminer des tokens inutiles. On peut également en enlever manuellement. La directive *bayes_expiry_max_db_size* peut être ajoutée au fichier de configuration générale *local.cf* pour définir une limite du nombre de tokens gardés dans la base(par défaut limité à 150 000tokens).

3. Théorème de Bayes
4. Documentation de sa learn

Sa-learn nous permet egalement un export de ces bases (avec l'option –backup du programme sa-learn, en redirigeant la sortie standard),. La restauration se fait avec l'option –restore.

6 Articulation du programme

6.1 Configuration

La configuration de Spamassassin se fait principalement à travers le fichier *local.cf* (exemple 2), qui se trouve dans le répertoire '*/etc/spamassassin*. Les utilisateurs UNIX peuvent également avoir leurs propres configurations grâce au fichier *user_prefs* qui se trouve dans le répertoire *.spamassassin* de leurs home respectifs. Par défaut, un certain nombre d'options sont prédéfinies.

```
# This is the right place to customize your installation of SpamAssassin.
#
# See 'perldoc Mail::SpamAssassin::Conf' for details of what can be
# tweaked.
#
###########################################################################

# How high a score is considered spam?
required_hits 5

# How should spam reports be inserted into the message?
report_safe 1

# Should we tag the subject of spam messages?
rewrite_subject 1

# By default, SpamAssassin will run RBL checks.  If your ISP already
# does this, set this to 1.
skip_rbl_checks 0
```

FIGURE 2 – Exemple de configuration basique de SpamAssassin

6.1.1 Options de configuration

Spam assassin possèdent plusieurs options dont certaine non activés par défaut. Il suffit de les ajouter aux fichiers de configuration. Les options les plus importantes sont les suivantes :

Désactiver les préférences utilisateurs Dans */etc/default/spamassassin*, il faut changer la ligne d'options pour désactiver les préférences par utilisateur (qui serait normalement stockées dans leur home directory) et utiliser les préférences globales à la place.

Options de score required_score n.nn (default :5) Définit le score par défaut considérant un message comme SPAM. la valeur peut être un réel ou un entier

score SYMBOLIC_TEST_NAME n.nn [n.nn n.nn n.nn] Assigne les scores(voir 4) à un test donné.

6.2 Les règles

Il est possibles de définir des règles personnalisés.

7 Utilisation

SpamAssassin peut être utilisé avec des Mail Transfert Agent comme Postfix. On peut par exemple contrôler les messages avec SpamAssassin juste après leurs réception par le demon SMTPD. Mais la façon la plus répandue est d'utiliser le MDA qui se charge de filtrer statiquement les mails et de les déposer dans leurs boites mails respectives.

7.1 MDA

Souvent couplé avec *Postfix* (Le serveur de messagerie le plus utilisé dans le monde), procmail est le Mail Delivery Agent(Egalement connu comme Local Delivery Agent, sont des programmes qui sont responsable de la délivrance des mails dans les boites des utilisateurs) que l'on trouve par défaut sur beaucoup de systèmes UNIX. L'utilisation avec procmail est assez simple. Il faut modifier ou créer le fichier procmail.c qui se trouve dans le répertoire */etc.*. Ainsi tout les messages passeront par SpamAssassin avant d'être délivré par procmail.
Un exemple de procmailrc minimale pour utiliser SPAM assassin :

```
DROPPRIVS=yes

LOGFILE=/var/log/procmail.log
VERBOSE=ON

# appel du deamon SpamAssassin
| /usr/bin/spamc -f

 :0e
{
   EXITCODE=$?
}
```

On peut tout à fait remplacer *spamc* par *spamassassin*. La seul différence étant au niveau des performances, chaque appel à la commande *spamassassin*créant un processus distinct, alors que *spamc* fait appel au demon *spamd*. A partir de là tout va passer à travers SpamAssassin.

7.2 SMTP

Filtrer les mail à l'entré est aussi possible.

Troisième partie
Autour de SpamAssasin

8 Plugins non offciels

Par défaut SpamAssassin intègre plusieurs plugins (modules perl). Mais du fait que c'est un logiciel libre et que son code soit accessible plusieurs plugins non officiels ont vu le jour. Cela inclut également les plugins commerciaux(Plugins utilisant des licences non-libres). Parmi ces plugins on peut citer :

Plugin gratuit

DSPAM Quand DSPAM (antispam léger qui travaille directement sur les connexion SMTP, fait partis du projet open BSD) est utilisé en conjonction avec amavisd-new (Qui fait office d'interface entre un MTA et plusieurs gestionnaire de contenus)ce dernier a DSPAM qui calcule automatiquement la probabilité que un message est un HAM/SPAM, et de positionner des entêtes. Ce plugin permet de faire prendre en compte ces entêtes pour le score des mails.

Bayes OCR Plugin Même méthode que le filtre Bayésien intégré mais qui cette fois s'effectue sur des images en effectuant une reconnaissance optique de caractères (ROC ou OCR en anglais). cela sert à detecter les SPAM envoyer sous forme d'images.

Image cerberus Même but que le précédent en analysant les images et en utilisant des techniques de reconnaissances de pattern

SaveHits Stocke une copie d'un message dans un répertoire daté quand des règles spécifiques sont atteintes, il créé ensuite un repertoire pour chaque règle contenant un lien symbolique du message. Cela permet de rapidement trouvé un message par date correspondants à certaines règles. Utile pour le développement de règles.

DecodeShortURLs Ce plugin décode les URL raccourcis en effectuant une requête *HTTP HEAD* au service de raccourcissement d'URL. puis l'ajoute à la liste des URL extraites par SpamAssassin , pour que d'autres plugins puissent y avoir accès.

DNSWL spam reporting Ajoute le service *DNSWL* aux services qui reçoivent des signalements de SPAM grâce à la commande "spamassassin –report" .

SAGrey Ce plugin est un outil de greylist, qui s'execute en deux phases.Son but est de permettre de repérer les SPAM uniques. Il effectue deux donc deux vérifications :
- Il vérifie si le score SPAM du mail dépasse le seuil définie
- Si c'est le cas il va également vérifié si l'expéditeur est présent dans la liste banche automatique(AWL) Si le message dépasse le seuil et n'est pas dans la liste blanche, il va le considérer comme SPAM unique et augmenté le score du mail. Il peut également placer ses propres entes. Les avantages de ce module sont de réduire les mails à temporiser dans le cas de greylisting, et de ne pas augmenter la taille des bases regroupant les expéditeurs connus.

MTX Une liste blanche de DNS distribuées.

Plugin commerciaux

9 Comparatif avec des logiciels similaires à SpamAssassin

Naturellement, il existe d'autres logiciels qui ont la même fonctionnalité que SpamAssassin, avec leur propres avantages et inconvénients, en voici quelques uns :

Spampal
- Avantages : Peut être couplé à SpamAssassin, fonctionne de manière transparente en tant que proxy POP ou IMAP, utilise beaucoup de blacklists piochées sur le net pour filtrer le spam.
- Inconvénients : SpamPal n'est pas totalement opérationnel au niveau du blocage par blacklist, et l'utilisation des blacklists DNS ralentissent et utilisent beaucoup de bande passante.

K9
- Avantages : Facile à mettre en place, léger.
- Inconvénients : Ne gère pas le protocole IMAP ni le cryptage SSL.

POPFile
- Avantages : Très efficace une fois mis en place
- Inconvénients : Compliqué à installer, ne gère pas les boites Hotmail

Spamilihator
- Avantages : Facile à installer, interface user-friendly
- Inconvénients : Doit apprendre de beaucoup de Spams reçus avant d'être efficace

Bogofilter
- Avantages : Analyse très rapide des mails, temps d'apprentissage des Spams beaucoup plus court, très bon complement à SpamAssassin.

— Inconvénients : Pas de réels inconvénients, Bogofilter reste le principal concurrent de SpamAssassin

10 Quelques logiciels pouvant travailler avec SpamAssassin

Les principaux logiciels avec lesquels SpamAssassin doit être couplé pour tirer profit du maximum de ses capacités sont évidemment les clients/serveurs de messagerie, comme par exemple :

— Procmail
— qmail
— sendmail
— Exim

— Thunderbird
— kmail
— Evolution

Mais SpamAssassin peut aussi se coupler a d'autres logiciels anti-spam pour obtenir un meilleur résultat de filtrage. On peux notamment l'associer à SpamPal, mais le logiciel anti-spam auquel il est le plus souvent couplé est Bogofilter car il reste le concurrent le plus performant sur ce plan. Bien configurés, l'alliance des deux logiciels permet un filtrage quasi-optimal.