

Signaler un spam à Signal-Spam depuis des outils Unix typiques

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 15 Mai 2007. Dernière mise à jour le 5 Novembre 2007

<http://www.bortzmeyer.org/signaler-a-signal-spam.html>

Le service Signal Spam (<http://www.signal-spam.fr/>) permet aux utilisateurs de signaler les spams reçus, dans l'espoir (très ténu) qu'ils soient traités par la police ou la justice. Comment utiliser ce service depuis Unix ?

Signal Spam (<http://www.signal-spam.fr/>) fournit deux greffons, pour les MUA Outlook et Thunderbird. Mais j'utilise mutt voire des scripts semi-automatiques. Signal Spam n'avait pas au début d'API documentée (elle est apparue (http://www.signal-spam.fr/index.php/frontend/extensions/api_de_signalement) par la suite). Mais comme le greffon Thunderbird est livré avec son source Javascript, on peut l'étudier et découvrir qu'il existe une API REST en <https://www.signal-spam.fr/api/signaler>. Celle-ci est appelable avec le programme curl dont on ne dira jamais assez à quel point il permet de faire facilement des clients REST (cf. <http://www.bortzmeyer.org/programmation-rest.html>). Merci beaucoup à Pierre Beyssac (auteur d'un programme équivalent (<http://signal.eu.org/blog/2007/05/15/lutte-contre-le-spam/>) en Python) pour son aide au débogage et à Vincent Lefèvre et Jean-Paul Béconne pour des remarques ultérieures.

Voici donc le programme shell. On utilise OpenSSL pour encoder en Base64 car Signal Spam ne se contente pas de l'encodage « pour cent » des URL (la documentation de l'API explique pourquoi). Le mot de passe figure dans le programme mais curl permet aussi de le mettre dans le fichier `~/.netrc`. Utilisez ce script à votre guise, sous la licence que vous voulez.

```
#!/bin/sh

# Username & password
auth_info=moi:monmotdepasse

# Do not change anything below
# Be careful if you change the curl options. For instance, '--proxy ""'
# is to force a direct access, without going through the proxy (the
# actual value will depend on the rules of your network, ask the
```

```
# system administrator)
if ! which openssl > /dev/null 2>&1; then
    echo "Missing openssl program" >&2
    exit 1
fi
if ! which curl > /dev/null 2>&1; then
    echo "Missing curl program" >&2
    exit 1
fi
curl_version=`curl -V | head -1 | cut -d' ' -f2-`
result=`openssl enc -base64 | curl --silent --proxy "" --request POST \
    --user-agent "signalspam.sh (http://www.bortzmeyer.org/signaler-a-signal-spam.html) curl/$curl_
    --user $auth_info \
    --write-out "%{http_code}" --form message='<-' \
    https://www.signal-spam.fr/api/signaler`
if [ $result != "202" ]; then
    echo "Error $result while reporting the spam"
    exit 1
fi
exit 0
```

Ensuite, je configure mutt pour lancer ce programme lorsque je tape Esc-Z :

```
macro index,pager \ez "<enter-command>set pipe_decode=no\n<pipe-entry>signalspam.sh\n<enter-command>set pipe
```

(Cette syntaxe nécessite mutt $\zeta=1.5$. Avec une version plus ancienne, mettez la macro "index" et la macro "pager" sur deux lignes séparées.)

Vincent Lefèvre me fait remarquer qu'il faudrait mieux sauvegarder l'ancienne valeur de `pipe_decode`, puis la restaurer à la fin comme expliqué dans la documentation de mutt (<http://www.mutt.org/doc/devel/manual.html#set-myvar>). Mais cela semble nécessiter une version très récente de mutt et je ne l'ai pas encore fait fonctionner.

Vérifiant sur mon espace membre Signal Spam, je peux voir que les spams sont bien enregistrés. Il reste à espérer que cela servira à quelque chose. Pour l'instant, je suis très pessimiste. Depuis six mois que le service est ouvert, les « signaleurs » n'ont eu droit à aucune information, ni statistiques (à part que « Depuis l'inauguration de ce service, près de 30 000 internautes se sont inscrits et plus de 4 millions de messages indésirables ont été ainsi transmis à Signal spam. »), ni information individualisée sur les signalements faits et leurs conséquences.