RFC 6590 : Redaction of Potentially Sensitive Data from Mail Abuse Reports

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 8 avril 2012

Date de publication du RFC : Avril 2012

https://www.bortzmeyer.org/6590.html

Il est fréquent qu'un message électronique contienne de l'information sensible ou personnelle, ne serait-ce que le nom des parties qui communiquent. Si ce message doit être transmis, comme faisant partie d'un rapport de problème, cette information doit être protégée. Ce RFC décrit un cadre général pour la **retouche** des messages. Le terme de « retouche » désigne l'opération d'occultation des parties confidentielles (notez que le "*redaction*" de l'original en anglais est un faux-ami, il ne signifie pas « rédaction »).

Le format ARF est normalisé dans le RFC 5965 [1]. Ce format standard permet de transmettre un rapport structuré (donc analysable automatiquement, par un programme) indiquant du spam, du hameçonnage ou tout autre abus. ARF permet également d'inclure dans le rapport le message qui a déclenché le processus de plainte. Mais ce message peut contenir des informations confidentielles, qu'on ne souhaite pas partager avec un tiers (sans compter les obligations légales comme, en France, la loi Informatique & Libertés). Le message est donc parfois retouché ("*redacted*" dans la langue de Katy Perry) avant d'être transmis. La section 8.5 du RFC 5965 décourage plutôt cette pratique (l'information occultée était peut-être nécessaire à la compréhension du rapport) mais cette approche « au diable la vie privée, la fin justifie les moyens » est très contestée et ce RFC 6590 adopte une vue plus raisonnable. L'occultation est désormais considérée comme justifiée.

L'important est donc plutôt de donner des conseils pratiques. Il y a des bonnes et des mauvaises façons d'occulter. Ainsi, remplacer toutes les parties locales des adresses (`stephane+blog` dans mon adresse `stephane+blog@bortzmeyer.org`) par la même chaîne de caractères (mettons `xxxxx@bortzmeyer.org`) fait perdre beaucoup d'information : on ne sait plus si deux rapports concernent le même utilisateur.

La section 3 de notre RFC conseille donc plutôt :

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc5965.txt`

— D'utiliser une transformation cohérente : deux chaînes identiques doivent donner le même résultat après occultation, et deux chaînes différentes doivent donner deux résultats différents.
— De décider clairement quelles parties du message sont privées (par exemple, uniquement les parties locales des adresses, comme dans mon premier exemple) et de n'appliquer les transformations qu'à celles-ci.
— De respecter la syntaxe des élements de protocole : si la transformation d'une partie locale d'une adresse donne une chaîne comportant un @ (caractère illégal dans une partie locale d'adresse), il faut l'encoder pour que l'adresse reste syntaxiquement légale, évitant ainsi de planter l'analyseur.

En appliquant ces règles, on a partiellement anonymisé le rapport, tout en permettant l'identification de tendances (par exemple, que le spam est plus souvent envoyé à certains utilisateurs).

Mais quelle opération de transformation utiliser ? Après la section 3 qui posait les principes, la section 4 s'occupe de technique. Ce RFC ne normalise pas **une** opération de transformation particulière. Si ROT13, qui est réversible, ne devrait pas être utilisé, les méthodes possibles incluent un hachage cryptographique (comme dans le RFC 2104) ou le remplacement des noms par un identifiant interne (numéro de client, par exemple).

Voici l'exemple qui figure dans l'annexe A du RFC. Le message de spam originel était :

```
From: alice@example.com
To: bob@example.net
Subject: Make money fast!
Message-ID: <123456789@mailer.example.com>
Date: Thu, 17 Nov 2011 22:19:40 -0500

Want to make a lot of money really fast?  Check it out!
http://www.example.com/scam/0xd0d0cafe
```

Ici, le récepteur, le FAI `example.net` est furieux et veut transmettre un rapport à `example.com` pour lui demander de faire cesser ces spams. `example.net` va occulter le nom du destinataire (son client), avec SHA-1 et la clé `potatoes`, qui sera concaténé au nom avant hachage, le résultat étant encodé en Base64. Cela donnera :

```
% echo -n potatoesbob | openssl sha1 -binary | openssl base64 -e
rZ8cqXWGiKHzhz1MsFRGTysHia4=
```

On va pouvoir alors construire un rapport ARF incluant :

```
From: alice@example.com
To: rZ8cqXWGiKHzhz1MsFRGTysHia4=@example.net
Subject: Make money fast!
Message-ID: <123456789@mailer.example.com>
Date: Thu, 17 Nov 2011 22:19:40 -0500

Want to make a lot of money really fast?  Check it out!
http://www.example.com/scam/0xd0d0cafe
```

Attention, en pratique, il existe pas mal de pièges. Par exemple, comme le note la section 5.3, l'information confidentielle peut se trouver aussi à d'autres endroits et des techniques de corrélation peuvent permettre de retrouver l'information occultée. Globalement, les messages retouchés selon ce RFC ne fourniront **pas** une forte confidentialité. Ainsi, le champ `Message-ID:` peut permettre, en examinant le journal du serveur de messagerie (celui de `example.com` dans l'exemple précédent), de retrouver émetteur et destinataire. C'est pour cette raison que le RFC n'impose pas l'usage de la cryptographie : elle n'apporterait pas grand'chose en sécurité.

Même chose pour les informations non structurées, par exemple le texte du message : il peut contenir des indications permettant de remplir les cases occultées (section 6).

D'une manière générale, il faut garder en mémoire qu'il existe de puissantes techniques de **désanonymisation** comme illustré par exemple par les articles « "*A Practical Attack to De-Anonymize Social Network Users*" `<http://www.iseclab.org/papers/sonda-TR.pdf>` » de Gilbert Wondracek, Thorsten Holz, Engin Kirda et Christopher Kruegel ou bien « "*Robust De-anonymization of Large Sparse Datasets*" `<http://www.cs.utexas.edu/~shmat/shmat_oak08netflix.pdf>` » de Arvind Narayanan et Vitaly Shmatikov.