RFC 6235 : IP Flow Anonymization Support

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 14 mai 2011

Date de publication du RFC : Mai 2011

https://www.bortzmeyer.org/6235.html

La question de la protection de la vie privée sur l'Internet est, à juste titre, une question très sensible et qui a déjà agité beaucoup d'électrons sur les blogs et autres sites Web, d'autant plus que de grosses et puissantes entreprises, comme Facebook et Google sont à l'œuvre pour réduire cette vie privée à pas grand'chose. Parmi toutes les problèmes pratiques que pose la protection de la vie privée, ce RFC se limite à celui des fichiers de trafic réseau, contenant des données sensibles et qu'on souhaite anonymiser. Le RFC se concentre sur le format IPFIX mais la plupart des problèmes exposés et des recommandations faites peuvent s'appliquer à d'autres formats comme les journaux de connexion ou comme les fichiers pcap. Un RFC utile, donc, pour tous les techniciens qui se soucient de la vie privée de leurs utilisateurs (et qui n'ont donc pas été embauchés par Facebook).

« Anonymiser » n'est pas un bon terme (mais je reprends celui du RFC). Tout le problème est qu'en effet les données ne sont pas réellement anonymes après cette opération : il est devenu simplement plus difficile de les relier à une personne ou une organisation donnée. Mais ce n'est pas impossible : de nombreux travaux de recherche ont montré qu'on pouvait reconstituer une bonne part des données manquantes dans un fichier « anonymisé ». Un exemple est l'excellent article de Bruce Schneier : « "*Why 'Anonymous' Data Sometimes Isn't*" <http://www.wired.com/politics/security/commentary/securitymatters/2007/12/securitymatters_1213> » qui montre très bien à quel niveau de perfectionnement les outils de « désanonymisation » sont parvenus.

Avant d'attaquer le RFC lui-même, commençons par un exemple simple. Supposons qu'on ait enregistré le trafic réseau en un endroit donné et qu'on récupère donc un fichier pcap. On peut le dire avec divers outils comme tcpdump et voici un résultat typique :

```
% tcpdump -n -v -r MYFILE.pcap
...
15:00:24.451549 IP (tos 0x0, ttl 64, id 17621, offset 0, flags [DF], proto TCP (6), length 1059)
    192.168.2.1.41949 > 193.159.160.51.80: Flags [P.], cksum 0xd268 (correct), seq 1029:2036, ack 247, win 108,
```

On peut y voir que 192.168.2.1 interrogeait le serveur HTTP 193.159.160.51 (un des serveurs d'Akamai, qui héberge www.wired.com, le site cité précedemment). On sait que c'était de l'HTTP par le port de destination (80) mais on aurait aussi pu le déduire en examinant les données (que tcpdump n'affiche pas, mais qui sont bien dans le fichier pcap et qu'un outil comme Wireshark peut afficher de manière très lisible ; ces mêmes données auraient pu nous dire qu'elle était la page Web exacte qui avait été consultée). Autre exemple, le journal d'un serveur HTTP où on trouvera des choses comme :

```
2001:db8:33::dead:beef:1 - - [10/May/2011:19:49:57 +0200] "GET /aventures-jules.html HTTP/1.1" 200 5306 "-"
```

qui nous indique que la machine d'adresse IP 2001:db8:33::dead:beef:1 a consulté la page aventures-jules.html de ce serveur. Ces informations sont-elles sensibles ? D'abord et avant tout sont-elles **nominatives**, ce qui, dans la loi Informatique & Libertés ne se limite pas aux informations comportant le nom d'une personne physique, mais inclut celles qui permettent de retrouver facilement cette personne (comme le « numéro de Sécu »). Ici, l'adresse IP source n'est pas la vraie (les données ci-dessus ont été anonymisées) mais, si elle l'était, elle permettrait assez facilement de retrouver l'auteur de cette visite à Wired et donc il est tout à fait justifié de considérer l'adresse IP comme nominative <http://www.cnil.fr/dossiers/internet-telecoms/fiches-pratiques/accessible/oui/article/ladresse-ip-est-une-donnee-a-caractere-personnel-pour-lensemble-des-cnil-europeennes/>.

Et c'est embêtant pour les chercheurs et les analystes. Car de telles données de connexion sont très utiles et de nombreux articles scientifiques intéressants ont été produits à partir de données comme celles-ci, collectées dans des grandes opérations comme le DITL <http://www.caida.org/projects/ditl/>. De même, les opérateurs réseaux ont souvent besoin d'analyser ces données pour mieux comprendre ce qui se passe dans leur réseau et pour l'améliorer. Enfin, l'étudiant, le curieux, le programmeur qui met au point un nouvel outil d'analyse ont besoin de telles données (on trouve par exemple plein de fichiers pcap sur pcapr.net <https://www.bortzmeyer.org/pcapr.html>). Mais transmettre ces données, les rendre publiques, les envoyer dans un pays qui n'a aucune législation de protection de la vie privée, comme les États-Unis, pose des problèmes éthiques et légaux considérables.

Une façon courante de concilier ces deux demandes est d'**anonymiser** les données. Par exemple, si on supprime l'adresse IP, on a enlevé la donnée la plus sensible, tout en permettant encore plusieurs analyses (par exemple sur la taille des paquets, sur les ports - donc, souvent, les services - les plus fréquents, etc. Mais en faisant cela, on a interdit certaines analyses, comme par exemple la répartition du trafic (tout le monde consomme t-il pareil, ou bien certains gros consommateurs sont-ils « responsables » de la majorité du trafic ?) Bref, l'anonymisation sera toujours un **compromis** : plus on retire de données et mieux on a anonymisé. Et plus les données perdent de leur intérêt... Il faudra donc prendre une décision en fonction des risques et des bénéfices. L'IETF n'a évidemment pas le pouvoir de prendre une telle décision mais ce RFC pose au moins les bases techniques nécessaires pour comprendre les choix. Il insiste notamment sur les techniques de désanonymisation, qui font que certaines techniques d'anonymisation ne protègent guère la vie privée.

Pour prendre l'exemple des requêtes DNS, si on peut croire que l'adresse IP source (qui est celle du résolveur, par celle du client final) n'est pas une information bien sensible, il ne faut pas oublier que le **contenu** de la requête (le QNAME, pour "*Query NAME*") peut en dire beaucoup. Par exemple, certains réseaux sociaux utilisent un sous-domaine qui a le nom de l'utilisateur.

Passons au RFC maintenant. Il a deux parts, une première générale sur les techniques d'anonymisation, leurs forces et leurs faiblesses et une seconde spécifique au format IPFIX (RFC 7011 [1]). La première

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc7011.txt

partie intéresse tous les programmeurs et récolteurs de données. Le fait d'avoir un format standard, IPFIX, pour les échanges d'information sur les flots de données (pcap - un standard de fait - est limité aux paquets individuels, une information de plus bas niveau) est évidemment un avantage pour tous ceux qui font circuler de l'information mais cela augmente aussi les risques de divulgation. D'où l'importance particulière de ce problème pour IPFIX.

Commençons par une définition rigoureuse de l'anonymisation (qu'il faudrait normalement écrire entre guillemets à chaque fois, tellement on est loin du vrai anonymat). La section 1 la définit comme la suppression ou la transformation d'une information qui, sans cette suppression ou cette transformation, pourrait permettre d'identifier la personne ou de l'organisation à l'origine d'une communication. Il ne faut pas utiliser l'anonymisation seule (elle a des tas de limites, notamment en raison de la puissance des techniques de désanonymisation) mais elle peut être un outil utile dans un arsenal de mesures de protection de la vie privée.

La section 1.4 rappelle d'ailleurs que ce RFC a le statut « Expérimental » car il considère que toutes les techniques d'anonymisation sont encore peu fiables (voir Burkhart, M., Schatzmann, D., Trammell, B., et E. Boschi, « "*The Role of Network Trace Anonymization Under Attack*" <ftp://www.snm.ethz.ch/pub/people/martibur/13v40n1b-burkhart.pdf> »).

La première partie de ce RFC revient sur ces techniques, d'une manière indépendante du format des données (pcap, IPFIX, etc). D'abord, la section 3 classifie les techniques d'anonymisation. Toutes ont en commun de chercher à empêcher la traçabilité, tout en préservant les informations utiles à l'analyse. Elles fonctionnent par mise en correspondance de l'espace des vrais valeurs vers un autre espace, « anonyme », suivant une fonction bien définie. La mise en correspondance est **récupérable** si on peut inverser cette fonction (ce qu'on ne souhaite évidemment pas, pour l'anonymisation) et **commensurable** (ou mesurable) si les deux espaces ont la même taille (préservant ainsi toutes les analyses qui cherchent à compter, par exemple le nombre d'adresses IP distinctes). Ainsi, une fonction de **généralisation** qui ne garderait que les N premiers bits de l'adresse IP (pour empêcher l'identification d'un individu) ne serait pas commensurable (plusieurs adresses IP seraient mises en correspondance avec un préfixe unique). Lorsque la fonction est commensurable, la technique d'anonymisation est appelée une **substitution** (je change chaque adresse IP par une autre). Notez que la substitution est vulnérable aux attaques par force brute, même si l'attaquant ignore la fonction utilisée, s'il peut injecter le trafic qu'il veut et observer les résultats (une variante des attaques « par dictionnaire »).

L'anonymisation la plus radicale est de retirer complètement un champ des données (méthode du feutre noir, le "*black-marker*", dit le RFC, du nom de l'outil des censeurs du courrier d'autrefois). Ni récupérable, ni commensurable, elle offre le maximum de sécurité mais peut faire perdre des informations précieuses.

Voyons maintenant chaque champ un à un. La section 4, qui détaille les techniques adaptées à chaque champ, commence évidemment par les adresses IP car c'est le champ le plus « nominatif ». Lorsqu'on parle d'anonymisation, on pense tout de suite à celle des adresses IP, et à juste titre (section 4.1). Le problème de leur anonymisation n'est pas simple. Par exemple, une adresse IP est structurée. Même si on supprime les N derniers bits, ceux qui restent, le préfixe, identifient encore un FAI ou une entreprise et donne donc des informations. Il y a quatre moyens d'anonymiser une adresse IP (sans compter évidemment le feutre noir) :

— La **troncation**, qui supprime les N derniers bits de l'adresse (pour le programmeur, la troncation est un `AND` avec un nombre de M-N uns et N zéros). C'est ce que permet un logiciel comme Squid avec sa directive `client_netmask`, qui permet de préserver la vie privée des utilisateurs du cache en ne gardant dans le journal que le préfixe de leur adresse IP. La troncation est une généralisation (elle ne conserve pas le nombre d'adresses IP.)

— La **troncation inverse** consiste au contraire à supprimer les N premiers bits. Ainsi, l'organisme (FAI ou entreprise) est masqué. C'est aussi une généralisation et deux machines sans lien entre elles (par exemple `192.0.2.57` et `203.0.113.57`) vont être confondues (ici, si on supprime les 24 premiers bits). Attention, si le réseau sur lequel a été fait la collecte est connu, il n'est pas difficile de retrouver le préfixe des adresses de ce réseau.
— La **permutation** consiste à remplacer chaque adresse IP par une autre. Elle conserve donc le nombre d'adresses IP (mais pas la structure des adresses : deux machines du même réseau local peuvent se retrouver très « loin » l'une de l'autre). La façon la plus fréquente de la réaliser est via une fonction de hachage de l'adresse IP. Attention, si cette fonction est connue, une attaque par force brute (tester toutes les adresses) devient possible, surtout en IPv4. Il est donc recommandé d'introduire un élément secret (par exemple un mot de passe concaténé à chaque adresse avant le hachage).
— La **pseudonymisation qui préserve le préfixe** est le quatrième moyen. Elle garde les N premiers bits et fait une permutation sur les M derniers. Cela permet de garder l'information sur la « proximité » de deux adresses. Comme à chaque fois, garder de l'information permet de faire certaines études... mais augmente le risque de désanonymisation.

Les adresses MAC identifient également une machine. Elles ne sont pas toujours présentes dans les données collectées (si le point de capture est sur le réseau d'un gros opérateur, les adresses MAC de la source et de la destination ne sont pas disponibles) mais on peut les trouver dans des captures faites localement. Elles peuvent aussi apparaître comme partie d'un identificateur de couches hautes, par exemple dans certaines adresses IPv6 (cf. section 2.5.1 du RFC 4291, et RFC 8981 pour une solution). Donc, il faut également anonymiser ces adresses (section 4.2). Comme les adresses IP, elles ont une structure, mais plus simple (les 24 ou 40 premiers bits indiquent le constructeur). Comme pour les adresses IP, on peut utiliser la troncation (on ne garde typiquement que les bits qui identifient le constructeur), la troncation inverse, la permutation et la pseudonymisation structurée.

Les numéros de port et de protocole de couche 4 n'identifient pas une entité du monde réel (comme une personne) mais peuvent servir à classer les machines (par exemple, le choix des ports sources permet souvent d'identifier le système d'exploitation, cf. RFC 6056). Ils méritent donc parfois une anonymisation (section 4.5) par exemple par permutation. Le RFC ne le mentionne pas, mais il n'est pas sûr qu'une technique d'anonymisation, qu'elle qu'elle soit, puisse fonctionner avec la distribution typique des protocoles : le plus répandu sera forcément TCP (numéro 6).

Les traces de trafic réseau contiennent souvent des nombres (collectivement appelés compteurs) qui peuvent révéler des choses sur leurs émetteurs (section 4.4). On peut donc les anonymiser en diminuant leur précision ou en les quantifiant, c'est-à-dire en les répartissant dans un petit nombre d'intervalles (les corbeilles ou "*bin*" dans le texte original du RFC) et en ne gardant qu'une valeur par intervalle.

Plus étonnant, les estampilles temporelles que contiennent les traces (date de passage du paquet, dates de début et de fin du flot) peuvent être également indiscrètes (voir Murdoch, S. et P. Zielinski, « "*Sampled Traffic Analysis by Internet-Exchange-Level Adversaries*" <`http://www.cl.cam.ac.uk/~sjm217/talks/pet07ixanalysis.pdf`> »). Par exemple, le séquencement des paquets peut permettre d'identifier une activité (une communication téléphonique n'aura pas du tout la même répartition temporelle des paquets qu'un téléchargement donc, même sans DPI, on pourra identifier le type de trafic). On peut donc essayer d'anonymiser les indications temporelles mais la plupart des méthodes disponibles, à part bien sûr le feutre noir, sont assez faibles contre la désanonymisation (section 4.3). On peut diminuer la résolution des temps (comme on a fait pour les compteurs, par exemple arrondir à la seconde la plus proche), on peut utiliser l'énumération (méthode qui remplace chaque estampille temporelle par un chiffre croissant : elle préserve l'ordre des événements, qui peut être important mais pas les écarts, ce qui interdit de mesurer des grandeurs comme la gigue), etc.

Une fois qu'on a anonymisé les données, il faut indiquer au destinataire comment on l'a fait, pour qu'il puisse faire ses analyses (section 5). Par exemple, si les adresses IP ont été tronquées, le destinataire

ne doit pas tenter d'étudier des distributions de trafic par machine, puisque plusieurs machines sont désormais regroupées dans le même préfixe. Il faudra donc indiquer si l'anonymisation était stable ou pas (la stabilité est la propriété de mise en correspondance d'une valeur de départ avec la même valeur d'arrivée ; une anonymisation instable, au contraire, peut diriger vers une autre valeur d'arrivée après un certain temps). Typiquement, un jeu de données est stable, sauf indication contraire, mais deux jeux de données peuvent ne pas l'être entre eux, donc il vaut toujours mieux l'indiquer explicitement.

S'il y a eu troncation, il faut évidemment indiquer la longueur qui a été conservée. S'il y a eu quantification, il faut donner la liste des valeurs et leurs intervalles. S'il y a eu permutation, il ne faut **pas** donner la fonction utilisée (ou alors, si une clé a été ajoutée, donner la fonction mais sans la clé), afin d'éviter les attaques par dictionnaire (traduire des valeurs et regarder le résultat obtenu).

L'anonymisation soulève d'autres questions de sécurité, décrites en section 9. Elle enfonce le clou que l'anonymisation n'est jamais parfaite (sauf à effacer toutes les données). Elle rappelle que l'anonymisation ne remplace pas la confidentialité. Si des données sensibles sont exportées vers un tiers de confiance, le chiffrement sur le trajet reste nécessaire, même si les données sont anonymisées.

Ici commencent maintenant les questions spécifiques à IPFIX. Si vous ne pratiquez pas ce format, vous pouvez décider d'arrêter là. Sinon, la section 1.1 offre un utile rappel des RFC 5470, RFC 5101 et RFC 7012. IPFIX permet d'envoyer des tuples TLV dont la sémantique est décrite dans un **gabarit** ("*Template*"). Plusieurs gabarits standards figurent dans le registre décrit par le RFC 7012.

La section 6 décrit comment représenter les données anonymisées et les méta-informations (celles de la section 5 sur le mécanisme d'anonymisation) dans un flow IPFIX. Les données effacées par le feutre noir représentent le cas le plus simple : on ne les met pas du tout dans le flot. Pour les autres, il faut ajouter au flot un "*Anonymization Options Template*" qui décrira les fonctions d'anonymisation. Les détails de ce gabarit figurent en section 6.1 et 6.2. Par exemple, un champ `anonymizationTechnique` va indiquer l'absence d'anonymisation s'il vaut 1, la troncation (ou la dégradation de la précision, qui est à peu près la même chose) s'il vaut 2, l'énumération s'il vaut 4, etc (des exemples plus complets figurent en section 8).

Ces considérations d'encodage ont été inscrites dans les registres IANA d'IPFIX <`https://www.iana.org/assignments/ipfix/ipfix.xml`>.`anonymizationTechnique`, par exemple, est identifié par le nombre 286.

L'anonymisation est typiquement effectuée dans un relais IPFIX ("*Mediator*"), en général situé à la frontière administrative du domaine (on n'anonymise pas si les données restent dans l'organisation d'origine). Attention : outre les données proprement dites, le flot contient souvent des méta-informations sur le processus de collecte qui peuvent si on ne prend pas soin de les modifier, casser en partie ou totalement l'anonymisation. Ainsi (section 7.2.3), le temps auquel a été fait l'observation peut permettre de retrouver les estampilles temporelles, même brouillées. L'identité du réseau où a été fait la mesure permet de retrouver une partie de l'adresse IP, même en cas de troncation inverse (cette identité a la même taille qu'une adresse IPv4 et il peut donc être tentant d'utiliser une adresse du réseau local comme identité).

Plus rigolo, la section 7.2.5 recommande de se méfier des adresses IP « spéciales » (RFC 6890). Ces adresses sont en effet souvent utilisées à des fins bien connues, qui font qu'on peut les retrouver, même anonymisées, en examinant leur activité. Cela fournit alors un point de départ au travail du désanonymisateur. Le RFC recommande donc d'exclure ces adresses des données exportées.

Les mises en œuvre d'IPFIX permettent-elles aujourd'hui cette anonymisation ? Je ne crois pas mais, en dehors du monde IPFIX, il existe un grand nombre d'outils d'anonymisation. Citons :

- Celles indiquées dans mon article sur pcapr <`https://www.bortzmeyer.org/pcapr.html`>, pour les fichiers pcap,
- mod_removeip <`https://we.riseup.net/debian/apache`> pour Apache,
- Une méthode pour les journaux du serveur Apache <`http://bluwiki.com/go/How_to_remove_IP_addresses`>.