RFC 5575 : Dissemination of Flow Specification Rules

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 25 mars 2013

Date de publication du RFC : Août 2009

https://www.bortzmeyer.org/5575.html

Lorsqu'on a un grand réseau compliqué, diffuser à tous les routeurs des règles de filtrage, par exemple pour faire face à une attaque par déni de service, peut être complexe. Il existe des logiciels permettant de gérer ses routeurs collectivement mais ne serait-il pas plus simple de réutiliser pour cette tâche les protocoles existants et notamment BGP ? On profiterait ainsi des configurations existantes et de l'expérience disponible. C'est ce que se sont dit les auteurs de ce RFC. « FlowSpec » (nom officieux de cette technique) consiste à diffuser des règles de traitement du trafic en BGP, notamment à des fins de filtrage. Ce RFC a depuis été remplacé par le RFC 8955 [1].

Les routeurs modernes disposent en effet de nombreuses capacités de traitement du trafic. Outre leur tâche de base de faire suivre les paquets, ils peuvent les classifier, limiter leur débit, jeter certains paquets, etc. La décision peut être prise en fonction de critères tels que les adresses IP source et destination ou les ports source et destination. Notre RFC 5575 encode ces critères dans un attribut NLRI ("*Network Layer Reachability Information*" est décrit dans la section 4.3 du RFC 4271) BGP, de manière à ce qu'ils puissent être transportés par BGP jusqu'à tous les routeurs concernés. Sans FlowSpec, il aurait fallu qu'un humain ou un programme se connecte sur tous les routeurs et y rentre l'ACL concernée.

Pour reconnaitre les paquets FlowSpec, ils sont marqués avec le SAFI (concept introduit dans le RFC 4760) 133 pour les règles IPv4 et 134 pour celles des VPN.

La section 4 du RFC donne l'encodage du nouveau NLRI. Un message `UPDATE` de BGP est utilisé, avec les attributs `MP_REACH_NLRI` et `MP_UNREACH_NLRI` du RFC 4760 et un NLRI FlowSpec. Celui-ci compte plusieurs couples {type, valeur} où l'interprétation de la valeur dépend du type (le type est codé sur un octet). Voici les principaux types :

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc8955.txt`

— Type 1 : préfixe de la destination, représenté par un octet pour la longueur, puis le préfixe, encodé comme traditionnellement en BGP (section 4.3 du RFC 4271). Ainsi, le préfixe `10.0.1.0/24` sera encodé (noté en hexadécimal) `01 18 0a 00 01` (type 1, longueur 24 - 18 en hexa - puis le préfixe dont vous noterez que seuls les trois premiers octets sont indiqués, le dernier n'étant pas pertinent ici).
— Type 2 : préfixe de la source (lors d'une attaque DoS, il est essentiel de pouvoir filtrer aussi sur la source).
— Type 3 : protocole (TCP, UDP, etc).
— Types 4, 5 et 6 : port (source ou destination ou les deux). Là encore, c'est un critère très important lors du filtrage d'une attaque DoS. Attention, cela suppose que le routeur soit capable de sauter les en-têtes IP (y compris des choses comme ESP) pour aller décoder TCP ou UDP. Tous les routeurs ne savent pas faire cela (en IPv6, c'est difficile <`https://www.bortzmeyer.org/analyse-pcap-ipv6.html`>).
— Type 9 : options activées de TCP, par exemple uniquement les paquets `RST`.

Tous les types de composants sont enregistrés à l'IANA <`https://www.iana.org/assignments/flow-spec/flow-spec.xml`>.

Une fois qu'on a cet arsenal, à quoi peut-on l'utiliser ? La section 5 détaille le cas du filtrage. Autrefois, les règles de filtrage étaient assez statiques (je me souviens de l'époque où tous les réseaux en France avaient une ACL, installée manuellement, pour filtrer le réseau de l'EPITA). Aujourd'hui, avec les nombreuses DoS qui vont et viennent, il faut un mécanisme bien plus dynamique. La première solution apparue a été de publier via le protocole de routage des préfixes de destination à refuser. Cela permet même à un opérateur de laisser un de ses clients contrôler le filtrage, en envoyant en BGP à l'opérateur les préfixes, marqués d'une communauté qui va déclencher le filtrage (à ma connaissance, aucun opérateur n'a utilisé cette possibilité, en raison du risque qu'une erreur du client ne se propage). De toute façon, c'est très limité en cas de DoS (par exemple, on souhaite plus souvent filtrer sur la source que sur la destination). Au contraire, le mécanisme FlowSpec de ce RFC donne bien plus de critères de filtrage.

Cela peut d'ailleurs s'avérer dangereux : une annonce FlowSpec trop générale et on bloque du trafic légitime. C'est particulièrement vrai si un opérateur accepte du FlowSpec de ses clients : il ne faut pas permettre à un client de filtrer les autres. D'où la procédure suggérée par la section 6, qui demande de n'accepter les NLRI FlowSpec que s'ils contiennent un préfixe de destination, et que ce préfixe de destination est routé vers le même client qui envoie le NLRI. Ainsi, un client ne peut pas déclencher le filtrage d'un autre puisqu'il ne peut influencer que le filtrage des paquets qui lui sont destinés.

Au fait, en section 5, on a juste vu comment indiquer les critères de classification du trafic qu'on voulait filtrer. Mais comment indiquer le traitement qu'on veut voir appliqué aux paquets ainsi classés ? (Ce n'est pas forcément les jeter : on peut vouloir être plus subtil.) FlowSpec utilise les communautés étendues du RFC 4360. La valeur sans doute la plus importante est 0x8006, `traffic-rate`, qui permet de spécifier un débit maximal pour les paquets qui correspondent aux critères mis dans le NLRI. Le débit est en octets/seconde. En mettant zéro, on demande à ce que tous les paquets classés soient jetés. Les autres valeurs possibles permettent des actions comme de modifier les bits DSCP du trafic classé.

Comme toutes les armes, celle-ci peut être dangereuse pour celui qui la manipule. La section 10 est donc la bienvenue, pour avertir de ces risques. Par exemple, comme indiqué plus haut, si on permet aux messages FlowSpec de franchir les frontières entre AS, un AS maladroit ou méchant risque de déclencher un filtrage chez son voisin. D'où l'importance de la validation, n'accepter des règles FlowSpec que pour les préfixes de l'AS qui annonce ces règles.

Ensuite, comme tous les systèmes de commande des routeurs à distance, FlowSpec permet de déclencher un filtrage sur tous les routeurs qui l'accepteront. Si ce filtrage est subtil (par exemple, filtrer tous les paquets plus grands que 900 octets), les problèmes qui en résultent seront difficiles à diagnostiquer.

Et les réalisations concrètes ? Parmi les auteurs du RFC, il y a autant d'employés de Cisco que de Juniper mais ce sont des arrivés récents chez Cisco. Ce RFC a d'abord été mis en œuvre sur les Juniper (on peut consulter leur documentation en ligne <https://www.juniper.net/techpubs/en_US/src/topics/task/configuration/policy-mgm-actions-flowspec.html>), puis chez Alcatel. Il existe aussi une implémentation de FlowSpec dans le logiciel d'injection de routes ExaBGP <http://bgp.exa.org.uk/>. D'autre part, en utilisant FlowSpec, attention aux problèmes de performance <http://mailman.nanog.org/pipermail/nanog/2011-January/030051.html>.

FlowSpec a joué un rôle important dans la panne CloudFlare du 3 mars 2013 <http://seenthis.net/messages/118644>, en propageant des critères incorrects (une taille de paquet supérieure au maximum permis par IP) à tous les routeurs. Ce n'est pas une bogue de FlowSpec : tout mécanisme de diffusion automatique de l'information à N machines différentes a le même problème potentiel. Si l'information était fausse, le mécanisme de diffusion transmet l'erreur à tous... (Dans le monde des serveurs Unix, le même problème peut se produire avec des logiciels comme Chef ou Puppet. Lisez un cas rigolo avec Ansible <http://jpmens.net/2013/02/06/don-t-try-this-at-the-office-etc-sudoers/>.) Comme le prévient notre RFC : « *"When automated systems are used, care should be taken to ensure their correctness as well as to limit the number and advertisement rate of flow routes."* » Toutefois, contrairement à ce que laisse entendre le RFC, il n'y a pas que les processus automatiques qui injectent des erreurs : les humains le font aussi.

Si vous voulez en apprendre plus sur FlowSpec :

— Excellent exposé très détaillé <http://media.frnog.org/FRnOG_18/FRnOG_18-6.pdf> (en anglais) de Frédéric Gabut-Deloraine à une réunion FRnog,
— Excellent exposé général <http://uknof.org.uk/uknof15/Mangin-NakedBGP.pdf> (en anglais) sur le BGP concret, par Thomas Mangin, avec quelques transparents sur FlowSpec à la fin,
— Un article en français sur LinuxFr <https://linuxfr.org//2011/01/27/27810.html>, avec exemples de configuration d'ExaBGP,
— Exposé (en anglais) de Jean-Marc Uzé <http://www.terena.org/activities/tf-ngn/tf-ngn17/uze-flowspec.pdf>, détaillant le cas des attaques par déni de service,
— Un exposé (en anglais) de David Lambert, avec des exemples Juniper <http://resources.nznog.org/2006/Friday-240306/DavidLambert-BGPFlowSpecificationUpdate/Lambert.ppt>,
— Une bonne étude de Flowspec sur les Juniper <http://www.junosandme.net/m/article-119193150.html>,
— Le projet du projet Cymru d'un serveur Flowspec Public <http://www.cymru.com/jtk/misc/community-fs.html> (je ne sais pas ce qu'il est devenu),
— Un exposé sur l'utilisation de Flowspec à Cloudflare <http://www.slideshare.net/junipernetworks/flowspec-bay-area-juniper-user-group-bajug> (écrit avant la panne).

Si vous vous intéressez à l'utilisation de BGP lors d'attaques par déni de service, vous pouvez aussi consulter les RFC 3882 et RFC 5635.