RFC 8386 : Privacy Considerations for Protocols Relying on IP Broadcast or Multicast

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 29 novembre 2018

Date de publication du RFC : Mai 2018

https://www.bortzmeyer.org/8386.html

Plusieurs protocoles applicatifs utilisent la diffusion, par exemple pour la découverte d'un service, et envoient donc des messages qui vont toucher toutes les machines du réseau local. Cela a des conséquences pour la vie privée : un observateur, même purement passif, peut apprendre plein de choses en écoutant. Il est donc important lorsqu'on conçoit des protocoles applicatifs de veiller à ne pas être trop bavard.

La diffusion, c'est envoyer à tout le monde. Comme il n'existe pas (heureusement !) de mécanisme fiable pour envoyer à tout l'Internet, en pratique, la diffusion se limite au réseau local. Mais c'est déjà beaucoup ! Connecté dans un café ou dans un autre endroit à WiFi, les messages diffusés arrivent à un groupe inconnu : un nombre potentiellement grand de machines. (L'utilisation d'un commutateur ne protège pas, si c'est de la diffusion.) La diffusion est très importante pour certaines fonctions (auto-configuration lorsqu'on ne connait pas sa propre adresse IP, ou bien résolution locale de noms ou d'adresses). La diffusion est tellement pratique (cf. RFC 919[1] et RFC 3819) qu'elle est utilisée par beaucoup d'applications.

Mais la diffusion est dangereuse ; à la conférence TRAC 2016 <http://conferences.imt-atlantique.fr/trac2016/>, les auteurs du RFC avaient, dans un excellent exposé, publié un premier résultat de leurs travaux sur la question (Faath, M., Weisshaar, F., et R. Winter, "*How Broadcast Data Reveals Your Identity and Social Graph*", 7th International Workshop on TRaffic Analysis and Characterization IEEE TRAC 2016, September 2016). En une journée à écouter le trafic diffusé sur leur université, ils avaient récolté 215 Mo de données. Les protocoles les plus bavards : 1) mDNS 2) SSDP 3) LLMNR 4) NetBIOS 5) Dropbox. Le seul client Dropbox diffuse à la cantonade l'ID du client, et celui des "*shares*" où il se connecte. Il est facile de faire un graphe des utilisateurs en mettant ensemble ceux qui se connectent au même "*share*". Les mêmes auteurs avaient mené une expérience à grande échelle en écoutant le trafic diffusé lors

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc919.txt

de la réunion IETF 93 <https://www.ietf.org/how/meetings/93/> à Prague, et cela avait suscité bien des débats, notamment juridico-légaux <https://mailarchive.ietf.org/arch/msg/93attendees/88H9vnO2MNzhMJiwBtaxj6PnKtU> (a-t-on le droit d'écouter du trafic qui est diffusé à tous ?) Comme en médecine, la science ne justifie pas tout et il est nécessaire de se pencher sur les conséquences de ses expériences.

Bien sûr, du moment qu'on envoie des données sur un réseau, elles peuvent être écoutées par des indiscrets. Mais la diffusion aggrave le problème de deux façons :

— Un éventuel indiscret n'a même pas besoin de techniques particulières pour écouter, il lui suffit d'ouvrir ses oreilles,
— La solution évidente à l'écoute, le chiffrement, marche mal avec la diffusion : il n'est pas facile de chiffrer lorsqu'on ne connait pas les destinataires.

Il est donc justifié de se préoccuper de près des conséquences de la diffusion sur la confidentialité (RFC 6973).

Pour certains protocoles conçus à l'IETF, il y a déjà eu des réflexions sur les problèmes de vie privée liés à leur usage de la diffusion. C'est évidemment le cas pour DHCP, dont les RFC 7819 et RFC 7824 ont pointé la grande indiscrétion. C'est aussi le cas des mécanismes de génération des adresses IPv6, expliqué dans le RFC 7721. Mais il y a également beaucoup de protocoles non-IETF qui utilisent imprudemment la diffusion, comme celui de Dropbox, présenté à la conférence TRAC. Ces protocoles privés sont en général peu étudiés, et la préservation de la vie privée est située très bas sur l'échelle des préoccupations de leurs auteurs. Et ils sont souvent non documentés, ce qui rend difficile toute analyse.

La section 1.1 de notre RFC résume les différents types de diffusion qui existent dans le monde IP. IPv4 a la diffusion générale (on écrit à `255.255.255.255`, cf. section 5.3.5.1 du RFC 1812) et la diffusion dirigée (on écrit à une adresse qui est celle du préfixe du réseau local, avec tous les bits « machine » à 1, cf. section 5.3.5.2 du même RFC). IPv6, officiellement, n'a pas de diffusion mais uniquement du "*multicast*" mais c'est jouer avec les mots : il a les mêmes possibilités qu'IPv4 et les mêmes problèmes de confidentialité. Si une machine IPv6 écrit à `ff02::1`, cela donnera le même résultat que si une machine IPv4 écrit à `255.255.255.255`. Parmi les protocoles IETF qui utilisent ces adresses de diffusion, on trouve mDNS (RFC 6762), LLMNR (RFC 4795), DHCP pour IPv4 (RFC 2131), DHCP pour IPv6 (RFC 8415), etc.

La section 2 détaille les problèmes de vie privée que l'envoi de messages en diffusion peut entrainer. D'abord, le seul envoi des messages, même sans analyser ceux-ci, permet de surveiller les activités d'un utilisateur : quand est-ce qu'il est éveillé, par exemple. Plus les messages sont fréquents, meilleure sera la résolution temporelle de la surveillance. Notre RFC conseille donc de ne pas envoyer trop souvent des messages périodiques.

Mais un problème bien plus sérieux est celui des identificateurs stables. Bien des protocoles incluent un tel identificateur dans leurs messages, par exemple un UUID. Même si la machine change de temps en temps d'adresse IP et d'adresse MAC (par exemple avec macchanger <https://www.gnu.org/software/macchanger/>), ces identificateurs stables permettront de la suivre à la trace. Et si l'identificateur stable est lié à la machine et pas à une de ses interfaces réseau, même un changement de WiFi à Ethernet ne suffira pas à échapper à la surveillance. C'était le cas par exemple du protocole de Dropbox qui incluait dans les messages diffusés un identificateur unique, choisi à l'installation et jamais changé ensuite. D'une manière générale, les identificateurs stables sont mauvais pour la vie privée, et devraient être utilisés avec prudence, surtout quand ils sont diffusés.

Ces identificateurs stables ne sont pas forcément reliés à l'identité étatique de la personne. Si on ne connait pas la sécurité, et qu'on ne sait pas la différence entre anonymat et pseudonymat, on peut penser que diffuser partout qu'on est `88cb0252-3c97-4bb6-9f74-c4c570809432` n'est pas très

révélateur. Mais outre que d'avoir un lien entre différentes activités est déjà un danger, certains protocoles font qu'en plus ce pseudonyme peut être corrélé avec des informations du monde extérieur. Par exemple, les iPhone diffusent fièrement à tout le réseau local « je suis l'iPhone de Jean-Louis » (cf. RFC 8117). Beaucoup d'utilisateurs donnent à leur machine leur nom officiel, ou leur prénom, ou une autre caractéristique personnelle. (C'est parfois fait automatiquement à l'installation, où un programme demande « comment vous appelez-vous ? » et nomme ensuite la machine avec ce nom. L'utilisateur n'est alors pas conscient d'avoir baptisé sa machine.) Et des protocoles diffusent cette information.

En outre, cette information est parfois accompagnés de détails sur le type de la machine, le système d'exploitation utilisé. Ces informations peuvent permettre de monter des attaques ciblées, par exemple si on connait une vulnérabilité visant tel système d'exploitation, on peut sélectionner facilement toutes les machines du réseau local ayant ce système. Bref, le RFC conseille de ne pas diffuser aveuglément des données souvent personnelles.

Comme souvent, il faut aussi se méfier de la corrélation. Si une machine diffuse des messages avec un identificateur stable mais non parlant (qui peut donc être ce `700a2a3e-4cda-46df-ad6e-2f062840d1e3` ?), un seul message donnant une autre information (par exemple nom et prénom) est suffisant pour faire la corrélation et savoir désormais à qui se réfère cet identificateur stable (`700a2a3e-4cda-46df-ad6e-2f062840d1e3`, c'est Jean-Louis). Lors de l'expérience à Prague citée plus haut, il avait été ainsi possible aux chercheurs de récolter beaucoup d'informations personnelles, et même d'en déduire une partie du graphe social (la machine de Jean-Louis demande souvent en mDNS celle de Marie-Laure, il doit y avoir un lien entre eux).

La plupart des systèmes d'exploitation n'offrent pas la possibilité de faire la différence entre un réseau supposé sûr, où les machines peuvent diffuser sans crainte car diverses mesures de sécurité font que tout le monde n'a pas accès à ce réseau, et un réseau public complètement ouvert, genre le WiFi du McDo, où tout est possible. Il serait intéressant, affirme le RFC, de généraliser ce genre de service et d'être moins bavard sur les réseaux qui n'ont pas été marqués comme sûrs.

La section 3 du RFC note que certains points d'accès WiFi permettent de ne pas passer systématiquement la diffusion d'une machine à l'autre, et de ne le faire que pour des protocoles connus et supposés indispensables. Ainsi, les requêtes DHCP, terriblement indiscrètes, pourraient ne pas être transmises à tous, puisque seul le point d'accès en a besoin. Évidemment, cela ne marche que pour des protocoles connus du point d'accès, et cela pourrait donc casser des protocoles nouveaux (si on bloque par défaut) ou laisser l'utilisateur vulnérable (si, par défaut, on laisse passer).

En résumé (section 4 du RFC), les conseils suivants sont donnés aux concepteurs de protocoles et d'applications :

— Autant que possible, utiliser des protocoles normalisés, pour lesquelles une analyse de sécurité a été faite, et des mécanismes de limitation des risques développées (comme ceux du RFC 7844 pour DHCP),
— Essayer de ne pas mettre des informations venant de l'utilisateur (comme son prénom et son nom) dans les messages diffusés,
— Tâcher de ne pas utiliser d'identificateurs stables, puisqu'ils permettent de surveiller une machine pendant de longues périodes,
— Ne pas envoyer les messages trop souvent.