RFC 8336 : The ORIGIN HTTP/2 Frame

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 22 mars 2018

Date de publication du RFC : Mars 2018

https://www.bortzmeyer.org/8336.html

Le concept d'**origine** est crucial pour la sécurité de HTTP. L'idée est d'empêcher du contenu actif (code JavaScript, par exemple) d'interagir avec des serveurs autres que ceux de l'origine, de l'endroit où on a chargé ce contenu actif. Dans la version 1 de HTTP, cela ne posait pas (trop) de problèmes. Mais la version 2 de HTTP permet d'avoir, dans une même connexion HTTP vers un serveur donné, accès à des ressources d'origines différentes (par exemple parce qu'hébergées sur des "*Virtual Hosts*" différents). Ce nouveau RFC ajoute donc au protocole HTTP/2 un nouveau type de trame, `ORIGIN`, qui permet de spécifier les origines utilisées dans une connexion.

L'origine est un concept ancien, mais sa description formelle n'est venue au'avec le RFC 6454 [1], dont la lecture est fortement recommandée, avant de lire ce nouveau RFC 8336. Son application à HTTP/2, normalisé dans le RFC 7540, a posé quelques problèmes (sections 9.1.1 et 9.1.2 du RFC 7540). En effet, avec HTTP/2, des origines différentes peuvent coexister sur la même connexion HTTP. Si le serveur ne peut pas produire une réponse, par exemple parce qu'il sépare le traitement des requêtes entre des machines différentes, il va envoyer un code de retour 421, indiquant à un client HTTP de re-tenter, avec une connexion différente. Pour lui faire gagner du temps, notre nouveau RFC 8336 va indiquer préalablement les origines acceptables sur cette connexion. Le client n'aura donc pas à essayer, il saura d'avance si ça marchera ou pas. Cette méthode évite également au client HTTP de se faire des nœuds au cerveau pour déterminer si une requête pour une origine différente a des chances de réussir ou pas, processus compliqué, et qui ne marche pas toujours.

Ce n'est pas clair ? Voici un exemple concret. Le client, un navigateur Web, vient de démarrer et on lui demande de se connecter à `https://www.toto.example/`. Il établit une connexion TCP, puis lance TLS, et enfin fait du HTTP/2. Dans la phase d'établissement de la connexion TLS, il a récupéré un certificat qui liste des noms possibles (`subjectAltName`), `www.toto.example` mais aussi `foobar.example`.
Et, justement, quelques secondes plus tard, l'utilisateur demande à visiter `https://foobar.example/ToU/TLDR/`.

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc6454.txt`

Un point central de HTTP/2 est la réutilisation des connexions, pour diminuer la latence <https://www.bortzmeyer.org/latence.html>, due entre autres à l'établissement de connexion, qui peut être long avec TCP et, surtout TLS. Notre navigateur va donc se dire « chic, je garde la même connexion puisque c'est la même adresse IP et que ce serveur m'a dit qu'il gérait aussi `foobar.example`, c'était dans son certificat » (et la section 9.1.1 du RFC 7540 le lui permet explicitement). Mais patatras, l'adresse IP est en fait celle d'un répartiteur de charge qui envoie les requêtes pour `www.toto.example` et `foobar.example` à des machines différentes. La machine qui gère `foobar.example` va alors renvoyer `421 Misdirected Request` au navigateur qui sera fort marri, et aura perdu du temps pour rien. Alors qu'avec la trame `ORIGIN` de notre RFC 8336, le serveur de `www.toto.example` aurait dès le début envoyé une trame `ORIGIN` disant « sur cette connexion, c'est `www.toto.example` et rien d'autre ». Le navigateur aurait alors été prévenu.

La section 2 du RFC décrit en détail ce nouveau type de trame (RFC 7540, section 4, pour le concept de trame). Le type de la trame est 12 (cf. le registre des types <https://www.iana.org/assignments/http2-parameters/http2-parameters.xml#frame-type>), et elle contient une liste d'origines, chacune sous forme d'un doublet longueur-valeur. Une origine est identifiée par un nom de domaine (RFC 6454, sections 3 et 8). Il n'y a pas de limite de taille à la liste, programmeurs, faites attention aux débordements de tableau. Un nom de la liste ne peut pas inclure de jokers (donc, pas d'origine `*.example.com`, donc attention si vous avez des certificats utilisant des jokers). Ce type de trames doit être envoyée sur le ruisseau HTTP/2 de numéro 0 (celui de contrôle).

Comme toutes les trames d'un type inconnu du récepteur, elles sont ignorées par le destinataire. Donc, en pratique, le serveur peut envoyer ces trames sans inquiétude, le client HTTP trop vieux pour les connaitre les ignorera. Ces trames `ORIGIN` n'ont de sens qu'en cas de liaison directe, les relais doivent les ignorer, et ne pas les transmettre.

Au démarrage, le client HTTP/2 a un jeu d'origines qui est déterminé par les anciennes règles (section 9.1.1 du RFC 7540). S'il reçoit une trame `ORIGIN`, celle-ci remplace complètement ce jeu, sauf pour la première origine vue (le serveur auquel on s'est connecté, identifié par son adresse IP et, si on utilise HTTPS, par le nom indiqué dans l'extension TLS SNI, cf. RFC 6066) qui, elle, reste toujours en place. Ensuite, les éventuelles réponses 421 ("*Misdirected request*") supprimeront des entrées du jeu d'origines.

Notez bien que la trame `ORIGIN` ne fait qu'indiquer qu'on peut utiliser cette connexion HTTP/2 pour cette origine. Elle n'authentifie pas le serveur. Pour cela, il faut toujours compter sur le certificat (cf. section 4 du RFC).

En parlant de sécurité, notez que le RFC 7540, section 9.1.1 obligeait le client HTTP/2 à vérifier le DNS **et** le nom dans le certificat, avant d'ajouter une origine. Notre nouveau RFC est plus laxiste, on ne vérifie que le certificat quand on reçoit une nouvelle origine dans une trame `ORIGIN` envoyée sur HTTPS (cela avait suscité des réactions diverses lors de la discussion à l'IETF). Cela veut dire qu'un méchant qui a pu avoir un certificat valable pour un nom, via une des nombreuses AC du magasin, n'a plus besoin de faire une attaque de l'Homme du Milieu (avec, par exemple, un détournement DNS). Il lui suffit, lorsqu'un autre nom qu'il contrôle est visité, d'envoyer une trame `ORIGIN` et de présenter le certificat. Pour éviter cela, le RFC conseille au client de vérifier le certificat plus soigneusement, par exemple avec les journaux publics du RFC 6962, ou bien avec une réponse OCSP (RFC 6960 montrant que le certificat n'a pas été révoqué, en espérant qu'un certificat « pirate » sera détecté et révoqué…)

Les développeurs regarderont avec intérêt l'annexe B, qui donne des conseils pratiques. Par exemple, si un serveur a une très longue liste d'origines possibles, il n'est pas forcément bon de l'envoyer dès le début de la connexion, au moment où la latence est critique. Il vaut mieux envoyer une liste réduite, et attendre un moment où la connexion est tranquille pour envoyer la liste complète. (La liste des origines, dans une trame `ORIGIN`, ne s'ajoute pas aux origines précédentes, elle les remplace. Pour retirer une

origine, on envoie une nouvelle liste, sans cette origine, ou bien on compte sur les 421. Ce point avait suscité beaucoup de discussions au sein du groupe de travail.)

Pour l'instant, la gestion de ce nouveau type de trames ne semble se trouver que dans Firefox <https://bugzilla.mozilla.org/show_bug.cgi?id=1337791>, et n'est dans aucun serveur, mais des programmeurs ont annoncé qu'ils allaient s'y mettre.