Comment minimiser l'information exposée dans les requêtes DNS ?

Julia Rougier - 3TC - PIR 4

Avril 2021

1 Introduction

Le protocole DNS (Domain Name System), introduit en 1983 est un système informatique distribué et hiérarchique, présent et impliqué dans la majorité des activités sur Internet. En effet, presque toutes les communications sur Internet commencent aujourd'hui avec une recherche par système de noms de domaine. Avant d'accéder à n'importe quelle page, un utilisateur recherche généralement d'abord un système de noms de domaine, qui correspond à un nom de domaine en particulier. Le résolveur, aussi appelé récurseur DNS est le serveur qui reçoit les requêtes DNS des clients, et interagit avec d'autres serveurs afin de retourner une adresse IP : celle du serveur avec lequel l'utilisateur souhaite communiquer. Ce protocole présente quelques failles, et notamment au niveau de la sécurité des données utilisateurs. Les paquets DNS sont transportés via le protocole UDP, sur le port 53. Ce port, aussi appelé Do53, est vulnérable aux écoutes des cybercriminels ou encore aux mauvaises intentions de certains tiers. Il s'avère qu'en DNS, les paquets ne sont pas encryptés. Les demandes et les réponses échangées entre les utilisateurs et les opérateurs DNS sont visibles sur le réseau. Les utilisateurs sont donc contraints d'avoir une confiance sans faille en les opérateurs DNS avec lesquels ils communiquent. Cette contrainte implique notamment que des informations importantes et personnelles à propos des utilisateurs peuvent être révélées. Parmi ces informations, on retrouve notamment les adresses IP personnelles des utilisateurs, les destinations Internet sur lesquelles ils se rendent, et encore bien d'autres. Les opérateurs DNS, pouvant être des ISP (Internet service provider) ou des tiers, reconnaissant les faiblesses et les vulnérabilités en termes de sécurité associées au protocole DNS ont créés et cherchés quelques solutions alternatives.

L'objectif principal de ces services alternatifs au protocole DNS par défaut est bien de minimiser au maximum l'information exposée dans les requêtes DNS, afin de sécuriser davantage les utilisateurs.

2 Etat de l'art

Dans l'objectif d'améliorer les faiblesses du protocole DNS, des réseaux de navigation privée, tel que ToR, pourraient s'avérer être des solutions améliorant la sécurité des informations utilisateurs. ToR utilise un encryptage des couches de son réseau ainsi qu'un mécanisme trois sauts, favorisant l'anonymat des utilisateurs sur le réseau. Cette alternative pourrait être très efficace mais elle connait quelques inconvénients non négligeables. En effet, cela réduit considérablement les performances sur tout le réseau, avec une forte latence du trafic DNS, comme nous pouvons le voir dans l'article [1].

Des solutions visant à minimiser l'information exposée dans les requêtes DNS ont déjà été conçues et implémentées dans le DNS par défaut déjà existant. Parmi ces solutions, on retrouve le DNS-over-HTTPS (DoH), ou encore le DNS-over-TLS (DoT) et le DNS-over-DTLS (DoDT), qui sont des approches dans lesquelles les requêtes DNS sont envoyées sur un canal crypté, empêchant l'opérateur récursif DNS de connaître le contenu de la requête DNS, mais n'empêchant pas cependant l'opérateur de lier la requête DNS et l'adresse IP de l'utilisateur. DNS-over-TLS (DoT) et DNSover-HTTPS (DoH) sont largement pris en charge par les navigateurs et de plus en plus par les systèmes d'exploitation. Dans DoT et DoH, le transport des messages DNS entre le résolveur de l'utilisateur et le résolveur récursif est chiffré en amont. Cependant, DoT et DoH ne sont pris en charge que par un petit nombre de grands fournisseurs DNS (tels que Cloudflare, GoogleDNS). Malgré le fait que ces protocoles cryptés soient accessibles au public, et plus sûrs, ils ne sont pas les plus utilisés encore aujourd'hui. L'utilisation de DoH a cependant augmenté sur le Web en raison des services publics offerts par Cloudfare et NextDNS et leur intégration dans le navigateur Firefox en tant que résolveur récursif de confiance.

Même si DoT et DoH protègent le DNS en cachant au résolveur le contenu des requêtes, ces approches DNS apporte une préoccupation secondaire : tout individu, site Web ou service en ligne peut associer une demande avec un client et son adresse IP ce qui implique que les résolveurs DNS peuvent observer tous les sites Web et services demandés par le client. D'autres approches ont bien sûr été explorées, comme DNSCrypt [1]

ou encore DNSCurve [1], reposant une nouvelle fois sur le cryptage du contenu des requêtes DNS, par différents moyens. Bien que ces différentes approches améliorent au quotidien la sécurisation des données utilisateurs dans le protocole DNS, elles présentent encore trop de failles et d'incertitudes pour que les utilisateurs puissent naviguer de façon sereine.

3 ODNS, ODoH, K-resolver : de nouvelles approches pour l'amélioration de DNS

L'axe d'amélioration le plus important du protocole DNS réside bien dans la sécurisation des données utilisateurs, impliquant donc une sécurisation plus importante de leurs données via les requêtes DNS. C'est pourquoi, bien que les techniques précédemment présentées soient pertinentes et de plus en plus déployées sur le Web, les chercheurs n'en finissent plus d'explorer et de proposer de nouvelles alternatives d'autant plus sécurisées. Parmi ces approches récentes, il semble important de mettre en avant ODNS (Oblivious DNS) [1], ODoH (Oblivious DNS-over-HTTPS) [2], deux approches plutôt similaires et comparables, ainsi que K-resolver [3], une solution qui complète DoH, déjà présenté plus tôt.

3.1 Oblivious DNS

Une première approche, le protocole Oblivious DNS (ODNS) [1] a été proposé par des chercheurs de l'université de Princeton. Le concept est destiné à empêcher le résolveur récursif de connaître à la fois l'identité du client (l'adresse IP du client) et les requêtes DNS correspondantes. Le protocole ODNS se base sur le découplage de ces deux informations en masquant le trafic de requête DNS du résolveur récursif. Ce protocole fonctionne de la manière suivante : le nom de la requête DNS est chiffré à l'aide d'une clé de session. Cette clé de session est chiffrée une nouvelle fois à l'aide de la clé publique du serveur ODNS cible et ajoutée au nom de la première requête chiffrée. Le résolveur ajoute ensuite le nom du domaine de serveur ODNS (ici .ODNS). Ensuite, le stub (un autre serveur DNS) transmet cette requête doublement chiffrée, en tant que requête DNS conventionnelle à son résolveur récursif. Le résolveur récursif ne connaît pas ODNS et traite la requête comme n'importe quelle autre requête DNS basique. Pour obtenir ce nom, le serveur ODNS déchiffrera la clé de session (car il a la clé privée correspondante) et l'utilisera pour déchiffrer le nom de la requête. Il peut ensuite utiliser une procédure de résolution récursive conventionnelle pour résoudre le nom de la requête d'origine. La réponse est chiffrée à l'aide de la clé de session fournie. Le serveur ODNS répondra alors au résolveur récursif avec le nom de la requête chiffrée dans la section de requête et la section de réponse chiffrée qu'il vient de générer. Dès réception de cette réponse, le résolveur récursif la transmettra au résolveur de stub. Enfin, le résolveur de stub utilise sa clé de session pour déchiffrer la réponse, et l'utilisateur obtient donc sa réponse. ODNS est une toute nouvelle approche du problème, en particulier dans son utilisation de l'infrastructure de résolveurs récursifs existante, puisque ce protocole est en effet compatible avec l'infrastructure DNS déjà existante. Cependant, ce n'est pas la seule approche actuellement disponible.

3.2 Oblivious DNS-over-HTTPS

Une approche différente du même problème, a été proposée par Oblivious DoH ou ODoH [2]. Elle est légèrement plus complexe que l'approche ODNS originale car elle utilise également un proxy inconscient en plus d'une cible, enveloppant l'ensemble de la requête DNS dans une enveloppe chiffrée. La conception d'Oblivious DNS-over-HTTPS (ODoH) est légèrement similaire à celle de DNS-over-HTTPS (DoH) avec l'ajout d'un nœud proxy intermédiaire qui effectue une requête au nom d'un client. Cependant, il existe quelques différences entre ODoH et DoH. L'objectif d'ODoH est de chiffrer les requêtes DNS envoyées au résolveur DNS, et d'empêcher ce résolveur d'avoir accès à l'adresse IP du client. Un résolveur de stub ODoH utilise DoH pour transmettre des requêtes à un proxy dit inconscient. Le résolveur de stub prend la requête, génère une clé de session (pour la réponse) et chiffre ces deux objets avec la clé publique de la cible. La cible peut déchiffrer le message DNS d'origine à l'aide de sa clé privée, et elle agira alors comme un résolveur récursif pour résoudre le nom de la requête de manière conventionnelle (comme dans le protocole DNS par défaut). La cible chiffre ensuite la réponse DNS à l'aide de la clé de session symétrique fournie par le client et la renvoie au serveur proxy, qui transmet ensuite la réponse chiffrée au client. Le client peut utiliser la clé de session pour déchiffrer la réponse.

Cette approche ODoH garantit que les requêtes effectuées soient connues uniquement par le résolveur de stub client qui lance la requête et le résolveur cible prévue détenant la clé privée correspondante pour décrypter le message. À tout moment de l'exécution, les clients connaissent la requête qu'ils ont faite, le cible, et le proxy choisi et peuvent vérifier si le retour la réponse a été correctement chiffrée. Le proxy inconscient connaît l'adresse IP du résolveur de stub client et la cible choisie par le client, mais ne peut récupérer aucune information sur

la requête en cours. La cible inconsciente connaît l'adresse IP du proxy transmettant la requête cryptée et peut décrypter le message pour obtenir la question DNS explicitement. Les mesures de performances de ODoH ont permis de voir que les pertes étaient minimales et que cette approche pourrait donc être un remplacement envisageable de DoH.

3.3 K-resolver

Enfin on retrouve K-resolver [3], qui semble pouvoir améliorer les faiblesses de DoH. En effet, comme évoqué plus tôt, DoH et DoT présentent des failles malgré le fait que le contenu des requêtes DNS y soit chiffré. Avec DoH, non seulement il n'y a qu'un seul résolveur qui reçoit les données chiffrées, mais ce dernier peut faire le lien entre l'adresse IP de l'utilisateur et la requête DNS, et ainsi avoir accès à l'historique de navigation de ce dernier. Avec K-resolver, plutôt que d'envoyer toutes les requêtes DNS au même résolveur, l'idée est que ces requêtes sont envoyées à K résolveurs différents : cela rend impossible le fait qu'un unique résolveur puisse accéder à tout l'historique d'un utilisateur, ce qui renforce donc la sécurité des données. En effet, les K résolveurs en question ne peuvent obtenir au maximum qu'un 1/Kème de l'historique des utilisateurs. K-resolver est donc une proposition d'amélioration de DoH, qui présente des failles. Il s'avère que cette approche reste encore instable. En effet, K-resolver et sa dispersion des requêtes DNS sur plusieurs résolveurs ont un impact sur les temps de chargement des pages Internet [3]. Les temps de chargement, plus longs sont dûs au fait qu'il existe aujourd'hui peu de serveurs DoH anycast, qui sont indispensables pour le déploiement de cette solution.

3.4 Comparaison

ODNS et ODoH sont deux solutions qui présentent des similitudes mais gardent chacune leur faiblesse et leurs avantages. Le problème avec ODNS est que c'est une approche un peu limitée du fait qu'elle s'implante directement dans la structure DNS existante. La requête d'origine doit donc être chiffrée selon les méthodes du protocole DNS par défaut. Au contraire, avec ODoH, ce n'est pas la structure DNS de base qui est utilisée, les informations ne circulent pas sous forme de requête DNS, mais utilisent deux agents : le serveur proxy et le serveur cible, qui doivent être gérés par deux entités distinctes pour éviter toutes collisions. Un second problème avec l'approche ODNS est qu'il est évident pour toute personne ayant une vision sur le réseau du serveur stub au serveur récursif que le client effectue une requête ODNS : le type de requête est visible, même si le nom de la requête est masqué. Cela pourrait être facilement résolu si le chemin ODNS du serveur stub au serveur récursif utilisait DoH plutôt que le transport UDP / TCP conventionnel du DNS. Au niveau de la performance, on retrouve une latence très faible, ainsi qu'un impact négligeable sur le trafic au niveau d'ODNS [1]. Pour ce qui est de ODoH, les expériences faites sur le temps de chargement et la latence montrent des pertes de performances minimales en comparaison avec DoH et d'autres protocoles DNS [2].

En ce qui concerne K-resolver, cette solution permettrait donc de compléter DoH tandis qu'ODoH serait un remplacement envisageable à DoH. Entre un scindage des résolveurs DoH dans le but de sécuriser les requêtes DNS et l'utilisation de différents serveurs proxy avec ODoH, ces deux solutions semblent toute deux efficaces. Cependant, DoH étant déjà implanté dans une certaine mesure, il semblerait que la mise en place de K-resolver soit plus simple, malgré le manque de serveurs DoH anycast.

4 Conclusion

Aujourd'hui, le protocole DNS a une place centrale et un rôle primordial dans toutes nos recherches internet. Les failles au niveau des informations des utilisateurs qui sont exposées lors du trafic des requêtes DNS sont très problématiques. Quelques techniques ont été expérimentées et implémentées, en ajoutant un encryptage des données, ce que le DNS par défaut ne faisait pas. Ces approches, comme DoH et DoT ont été déployées assez rapidement et ont suscités la curiosité de nombreux chercheurs. Bien que plus sécurisant que le DNS par défaut, ces techniques restent perfectibles. C'est pourquoi, parmi d'autres, la solution K-resolver a été proposée, permettant de minimiser la vision de l'historique des utilisateurs par les serveurs DoH. Toujours dans cet objectif de minimiser l'exposition des données lors des requêtes DNS, deux autres approches ont été confrontées. Nous avons vu qu'ODNS et ODoH, respectivement implantables dans la structure DNS par défaut ou pour remplacer l'approche DoH, sont deux solutions qui présentent beaucoup d'avantages. Basées toutes deux sur le découplage des adresses IPs des utilisateurs des requêtes DNS associées, en passant ou non par un serveur proxy, ces solutions utilisant des clés d'encryptage sont prometteuses. Laquelle des deux sortira du lot, c'est une question qu'il semble important de se poser. Néanmoins, le point qui semble le plus pertinent reste le fait que ces trois solutions ont pour seul et unique but d'améliorer la sécurité du protocole DNS et de minimiser l'exposition des données utilisateurs lors de requêtes DNS.

5 Références

[1] Paul Schmitt, Anne Edmundson, Allison Mankin, and Nick Feamster. Oblivious dns: Practical privacy for dns queries. Proceedings on Privacy Enhancing Technologies, 2019(2):228–244, 2019.

[2] Sudheesh Singanamalla, Suphanat Chunhapanya, Marek Vavruša, Tanya Verma, Peter Wu, Marwan Fayed, Kurtis Heimerl, Nick Sullivan, Christopher Wood Cloudflare Inc., University of Washington Oblivious DNS over HTTPS (ODoH): A Practical Privacy Enhancement to DNS arXiv:2011.10121v1 [cs.CR] 19 Nov 2020

[3] Nguyen Phong Hoang, Ivan Lin, Seyedhamed Ghavamnia, Michalis Polychronakis Stony Brook University, New York, USA K-resolver: Towards Decentralizing Encrypted DNS Resolution arXiv:2001.08901v2 [cs.NI] 18 Feb 2020

[4] Sandra Siby, Marc Juarez, Narseo Vallina-Rodriguez, Carmela Troncoso EPFL, imec-COSIC KU Leuven, IMDEA Networks Institute DNS Privacy not so private: the traffic analysis perspective

[5] Timm Böttger, Felix Cuadrado, Gianni Antichi, Eder Leão Fernandes, Gareth Tyson, Ignacio Castro and Steve Uhlig. 2019. An Empirical Study of the Cost of DNS-over-HTTPS. In IMC '19: ACM Internet Measurement Conference, October 21–23, 2019, Amsterdam, Netherlands. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3355369.3355575