



DoT, DoH : le DNS et le chiffrement

Shaft, 28 juin 2019



Shaft

Internaute auto-radicalisé

Chief Disruption Officer, Shaft Inc.

Mail : john+pses@shaftinc.fr

Mastodon : shaft@mamot.fr

Blog : <https://www.shaftinc.fr/>

GPG : A2C3 885D 0501 EF60



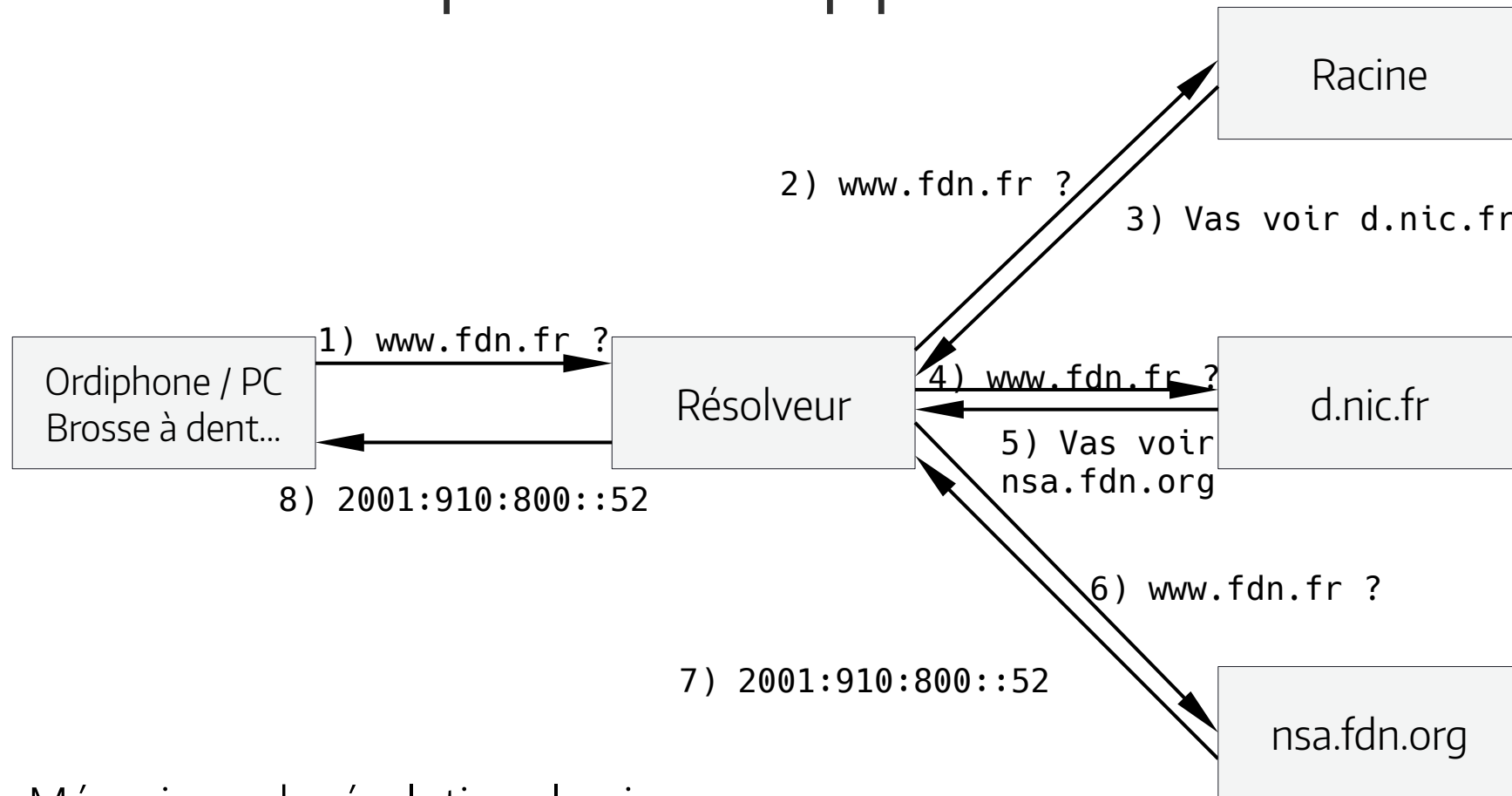
Sommaire

- DNS & Vie privée : rappels utiles
- DNS sur TLS (DoT)
- DNS sur HTTPS (DoH)
- Administrer un résolveur DoT/DoH : quelques éléments
- Conclusion

DNS & Vie privée : rappels

- DNS, conçu dans les années 80
- Pas de considérations pour la vie privée à l'époque
 - ▶ Voyage en clair
 - ▶ En général via UDP
 - ▶ Le résolveur pose la question complète à chaque étape
- Avantage : a grandement simplifié la mise en œuvre

DNS & Vie privée : rappels



Mécanisme de résolution classique

DNS & Vie privée : rappels

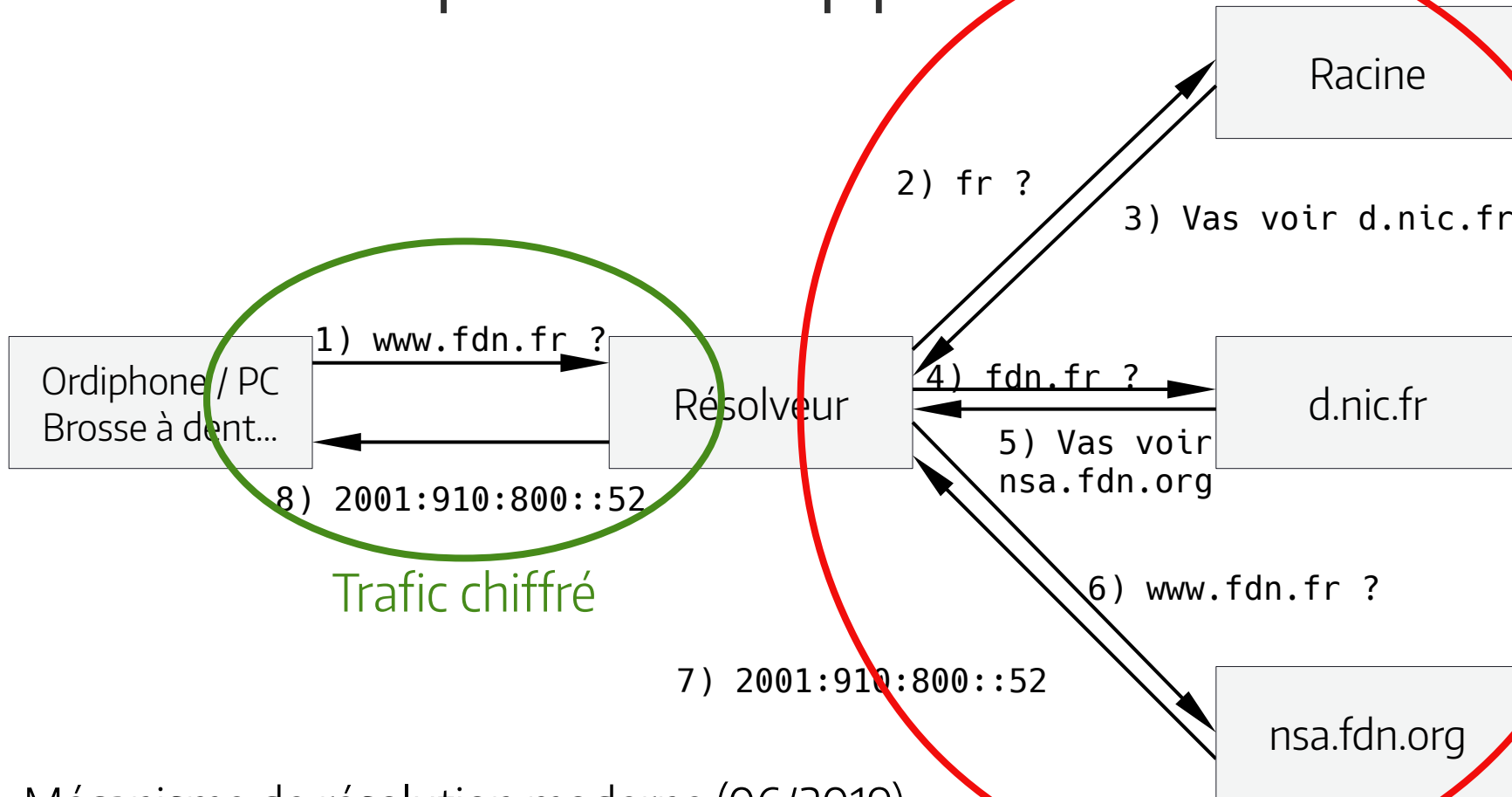
- 2013, un admin-sys dévoile le pot aux roses
 - ▶ Les espions espionnent
 - ▶ Massivement depuis la généralisation d'Internet
- Plus possible d'ignorer le problème
 - ▶ L'écoute passive est une attaque contre le réseau (RFC 7258)
 - ▶ Les protocoles – anciens et nouveaux – doivent intégrer la protection de la vie privée

DNS & Vie privée : rappels

- Dans le cas du DNS
 - ▶ Minimisation de la question posée (RFC 7816)
 - ▶ Chiffrement du trafic
- Chiffrement du DNS, en 2019
 - ▶ Le trafic Résolveur → Serveurs faisant autorité n'est pas chiffré : problème complexe, travaux en cours
 - ▶ Le trafic Client → Résolveur : problème plus simple, déjà mis en œuvre (DoT, DoH...)

DNS & Vie privée : rappels

Trafic en clair




Mécanisme de résolution moderne (06/2019)

DNS sur TLS (DoT)

- Normalisé en mai 2015 (RFC 7858)
- Présent dans les logiciels sérieux
- Principe
 - ▶ Protéger le transport des requêtes entre le résolveur et le PC de la famille Michu
 - ▶ De préférence garantir que l'on parle au bon serveur
 - ▶ Via TLS (protocole bien connu), sur un port dédié (853 par défaut)

DNS sur TLS (DoT)

- Attention ! 
 - ▶ Rappel : DoT ne protège pas le transport entre le résolveur et les serveurs faisant autorités.
 - ▶ DoT ne garanti pas l'intégrité des données (c'est le travail de DNSSEC)
 - ▶ DoT protège contre l'écoute du trafic, **ne protège pas du serveur indélicat**
 - ▶ TLS et ses implémentations ne sont pas exempts de failles de sécurité

DNS sur TLS (DoT)

- Principales difficultés pour le client
 - ▶ Technique : authentifier le serveur DoT auquel on se connecte
 - ▶ Humaine : on va confier la résolution DNS à une personne tierce. Nécessite d'avoir confiance en l'administrateur·trice du service (problème déjà présent sans DoT)

DNS sur TLS (DoT)

- À qui faire confiance ?
 - ▶ Pas de réponses simples
 - ▶ A priori éviter les grosses boîtes commerciales (Google, CloudFlare & co.)
 - ▶ L'administrateur·trice du résolveur devrait : publier une politique de vie privée, publier sa configuration, ne pas logger les requêtes, ne pas faire mentir son résolveur...
 - ▶ ...Mais très dur à vérifier... Des structures ayant une charte (CHATONS, FFDN) sont préférables

DNS sur TLS (DoT)

- Résoudre la difficulté d'authentification
 - ▶ Il existe 2 profils de connexion avec DoT : opportuniste et strict
 - ▶ Le profil opportuniste n'authentifie pas ou continue si l'authentification échoue
 - ▶ Le profil strict renonce à utiliser le service si l'authentification échoue
- Le profil strict est à préférer

DNS sur TLS (DoT)

- Résoudre la difficulté d'authentification (bis)
 - ▶ Le RFC 7858 introduisait une méthode, le RFC 8310 en ajoute
 - ▶ « Épinglage de clé » + adresse IP du résolveur (RFC 7858) : la clé publique (SPKI) est connue à l'avance par le client et on vérifie à la connexion.
 - ▶ ADN (nom de domaine du résolveur DoT) + IP : l'authentification se fait sur le nom de domaine et on vérifie qu'il correspond par le certificat X.509 du résolveur.
- Ces deux méthodes nécessitent de connaître à l'IP voir la clé publique, qui peuvent changer

DNS sur TLS (DoT)

- Résoudre la difficulté d'authentification (ter)
 - ▶ ADN seul : Méthode la plus simple mais fait fuiter des « méta-requêtes » pour récupérer l'IP du résolveur
 - ▶ DANE : Plus fiable que le système des AC, mais fait également fuiter des requêtes en clair. A priori pas d'implémentations connues. Ceci dit, l'administrateur·trice du résolveur a tout intérêt à utiliser DANE pour son service
 - ▶ DHCP, DNSSEC Chain Extension : pas normalisé, pas d'implémentation connues

DNS sur TLS (DoT)

- Principales limites de DoT
 - ▶ Utilise TLS (et donc TCP) : Nécessite plus de ressources et implique une latence plus importante
 - ▶ L'écoute reste possible mais est plus compliquée (TLS ne protège pas la taille des paquets, on peut deviner une question en regardant la taille de la réponse)
 - ▶ Le trafic sort en clair du résolveur : si peu de clients sont connectés au serveur, facile de retrouver qui a envoyé telle ou telle requête
 - ▶ Port 853 peut facilement être bloqué

DNS sur TLS (DoT)

- Solution à ces limites
 - ▶ TCP a des mécanismes pour garder une connexion ouverte et pour accélérer la connexion en elle-même (TCP Fast Open). TLS a un mécanisme de ticket.
 - ▶ Le remplissage (EDNS(0) Padding pour le DNS) brouille les pistes : on remplit de « 0 » questions et réponses afin qu'ils aient une taille fixe
 - ▶ Rien de normalisé. Une piste est l'obfuscation : le résolveur noie son trafic sortant de requêtes bidon
 - ▶ DNS over HTTPS

DNS sur TLS (DoT)

- Quels clients utiliser ?
- 3 principaux logiciels libres
 - ▶ Stubby
 - ▶ Knot Resolver
 - ▶ Unbound
- Chacun a ses avantages et ses défauts
 - ▶ Certains défauts sont contournables, d'autres se corrigeront avec le temps (normalement)

DNS sur TLS (DoT)

- Stubby



DNS sur TLS (DoT)

- Stubby
 - ▶ Résolveur minimum (« Stub resolver »)
 - ▶ Disponible sous forme de paquet dans les distributions sérieuses (Debian 10, Ubuntu, Arch, Manjaro...) et sous macOS (via Homebrew) et Windows.
 - ▶ Utilise OpenSSL pour la partie TLS

DNS sur TLS (DoT)

- Stubby : avantages
 - ▶ À jour en terme de techniques (EDNS(0) Padding, connexion persistantes, bientôt compatible DoH...) et développement actif
 - ▶ Le plus simple à configurer : fonctionne dès la « sortie de boîte » Les principaux serveurs DoT sont préconfigurés (on peut ne pas les utiliser)
 - ▶ Permet une double authentification (IP du résolveur + SPKI et IP + ADN)
 - ▶ Sous macOS, une GUI existe pour le configurer (non testé)

DNS sur TLS (DoT)

- Stubby : désavantages
 - ▶ Résolveur minimum donc pas de cache (problème contournable)
 - ▶ La configuration utilise la syntaxe YAML : attention à l'indentation a priori
 - ▶ Toujours considéré comme en bêta : voir la stabilité dans le temps et certains aspects moins accueillants (peu de logs donc plus dur à déboguer en cas de problème)

DNS sur TLS (DoT)

- Knot Resolver
 - ▶ Résolveur complet avec cache
 - ▶ Disponible sous forme de paquet dans la plupart des distributions Linux (pas sous Manjaro) et sous macOS (via Homebrew). Rien sous Windows
 - ▶ Peut faire office de serveur DoT ou de client
 - ▶ Utilise GnuTLS pour la partie TLS
 - ▶ Résolveur inclus dans le routeur Turris Omnia

DNS sur TLS (DoT)

- Knot Resolver : avantages
 - ▶ Possède un cache : diminue le nombre de requêtes et la latence
 - ▶ À peu près à jour en termes de techniques. Activées par défaut pour celles qui touchent à DoT
 - ▶ Le cache n'est pas lié au service s'occupant des requêtes, redémarrer le résolveur ne vide pas le cache

DNS sur TLS (DoT)

- Knot Resolver : désavantages
 - ▶ Intégration à `systemd` plus complexe (vaut surtout pour le configurer en tant que serveur)
 - ▶ Configuration en LUA, syntaxe plus complexe
 - ▶ Très complet, la documentation est touffue et pas souvent claire pour les moins initié·e·s
 - ▶ Authentification via SPKI + IP ou ADN + IP mais pas les 2 à la fois

DNS sur TLS (DoT)

- Unbound
 - ▶ Résolveur complet avec cache
 - ▶ Disponible sous forme de paquet sous Linux, sous macOS (via Homebrew) et Windows
 - ▶ Peut faire office de serveur DoT ou de client
 - ▶ Utilise OpenSSL pour la partie TLS

DNS sur TLS (DoT)

- Unbound : avantages
 - ▶ Possède un cache : diminue le nombre de requêtes et la latence
 - ▶ Relativement simple à configurer (quelques pièges en fonction de la distribution ceci dit)
 - ▶ Parfaitement documenté

DNS sur TLS (DoT)

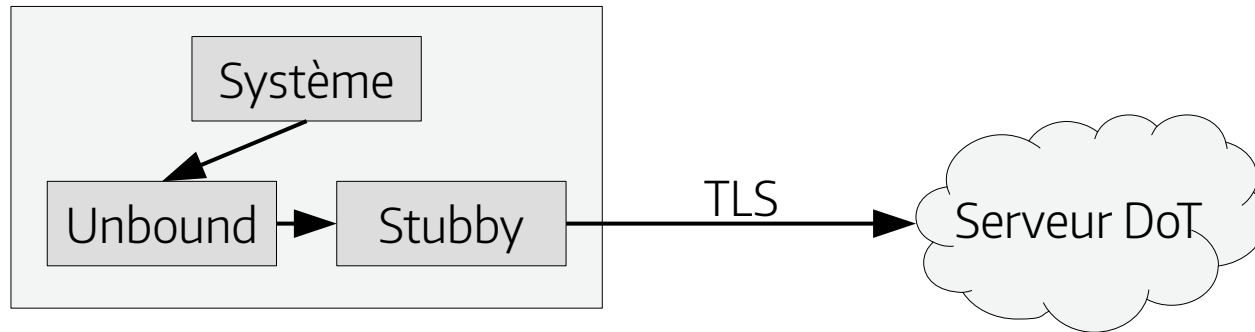
- Unbound : désavantages
 - ▶ Manque des techniques (EDNS(0) Padding, pas de SPKI pour l'authentification notamment)
 - ▶ Redémarrer le service vide le cache
 - ▶ Quand configuré en client DoT, ne suit pas la recommandation de garder la connexion TCP ouverte un certain temps : (très) mauvaises performances & surconsommation de ressources pour le client et le serveur

DNS sur TLS (DoT)

- Unbound : désavantages
 - ▶ Ne pas garder les connexions TCP ouvertes dans ce cas précis rend Unbound difficilement utilisable en tant que client DoT.
 - ▶ Pas résolu dans la version 1.9.2, la plus récente en juin 2019

DNS sur TLS (DoT)

- Unbound + Stubby
 - ▶ Principe : sur la machine locale, Unbound est le résolveur utilisé par le système mais transmet tout à Stubby qui va gérer la partie TLS et discuter avec le serveur DoT



DNS sur TLS (DoT)

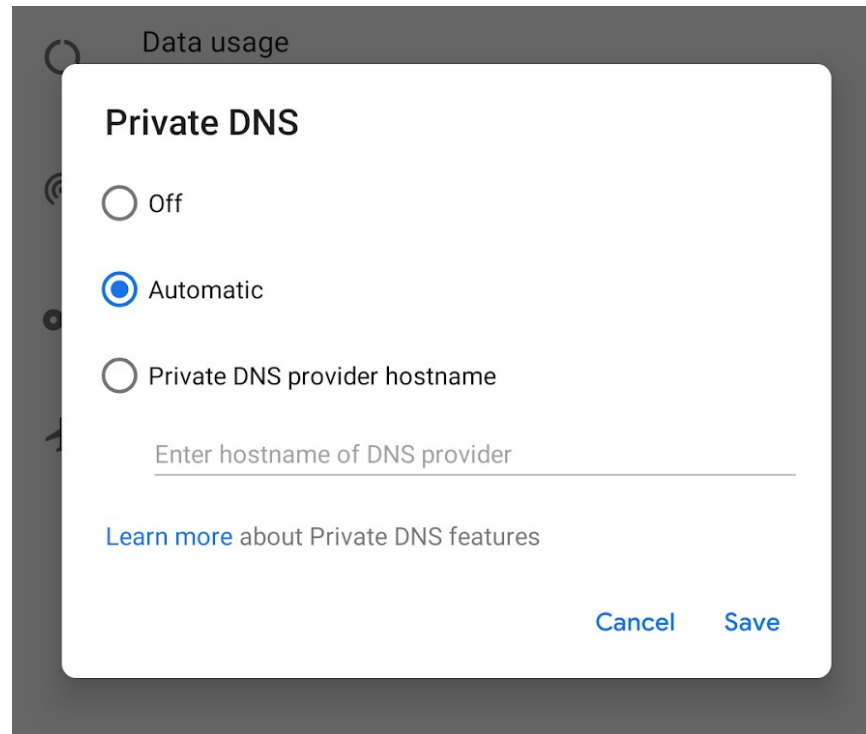
- Unbound + Stubby
 - ▶ Avantages : Le cache et les options d'Unbound, la qualité de la gestion TLS et l'avancement technique de Stubby
 - ▶ Désavantage : un peu plus compliqué à configurer (mais pas beaucoup plus)

DNS sur TLS (DoT)

- Android 9 et dérivés (LineageOS 16.1 notamment)
 - ▶ Premier OS à intégrer DoT nativement
 - ▶ Principe : mode « automatique » (Android teste si le résolveur fourni par le réseau cause DoT et l'utilise si oui) ou strict (on fournit l'ADN d'un serveur DoT)
 - ▶ Pas testé :-)

DNS sur TLS (DoT)

- Android 9 et dérivés



DNS sur HTTPS (DoH)

- DoH dans les grandes lignes
 - ▶ Récent : RFC 8484 (10/2018)
 - ▶ Le port de DoT est trivial à bloquer
 - ▶ Le port 443 l'est rarement
 - ▶ Principe : encapsuler une requête DNS dans une requête HTTP/2 et l'envoyer à un serveur HTTPS qui va faire la résolution et renvoyer la réponse en HTTPS

DNS sur HTTPS (DoH)

- DoH dans les grandes lignes
 - ▶ Récent : RFC 8484 (10/2018)
 - ▶ Le port de DoT est trivial à bloquer
 - ▶ Le port 443 l'est rarement
 - ▶ Principe : encapsuler une requête DNS dans une requête HTTP/2 et l'envoyer à un serveur HTTPS qui va faire la résolution et renvoyer la réponse en HTTPS

DNS sur HTTPS (DoH)

- Autres avantages
 - ▶ HTTPS est connu et maîtrisé
 - ▶ HTTP/2 : streams, padding et connexions persistantes
 - ▶ Permet à des dévs JS d'avoir un mécanisme de résolution complet
- Désavantages
 - ▶ Un peu plus lent (TCP → TLS → HTTPS → DNS)
 - ▶ HTTP trop bavard (User agent,...)
 - ▶ Internet over HTTPS

DNS sur HTTPS (DoH)

- Le protocole dont beaucoup de monde parle... en mal
 - ▶ Attaqué pour son but (cf. les telcos ou Paul Vixie : « My network, my rules »)
 - ▶ Attaqué à tort pour la manière dont les géants (Google, Mozilla) comptent le déployer
 - ▶ Certaines reproches légitimes (Trop bavard, ossification)

DNS sur HTTPS (DoH)

- Ça ressemble à quoi DoH ?

```
# curl -v --doh-url https://doh.powerdns.org/ www.shaftinc.fr
...
* Connected to doh.powerdns.org (2a01:7c8:d002:1ef:5054:ff:fe40:3703) port 443 (#1)
* ALPN, offering h2
[ On se connecte et on propose de parler HTTP/2 ]
...
* SSL connection using TLSv1.3 / TLS_AES_256_GCM_SHA384
* ALPN, server accepted to use h2
* Server certificate:
*  subject: CN=doh.powerdns.org
...
* SSL certificate verify ok.
[ Tout est OK côté TLS, chiffrement fort (TLS 1.3) ]
...
* Using Stream ID: 1 (easy handle 0x55eab6a378f0)
> POST / HTTP/2
Host: doh.powerdns.org
Accept: */*
Content-Type: application/dns-message
Content-Length: 33
[Envoie de la requête]
...
[ Suspense ]
```

DNS sur HTTPS (DoH)

- Ça ressemble à quoi DoH ?

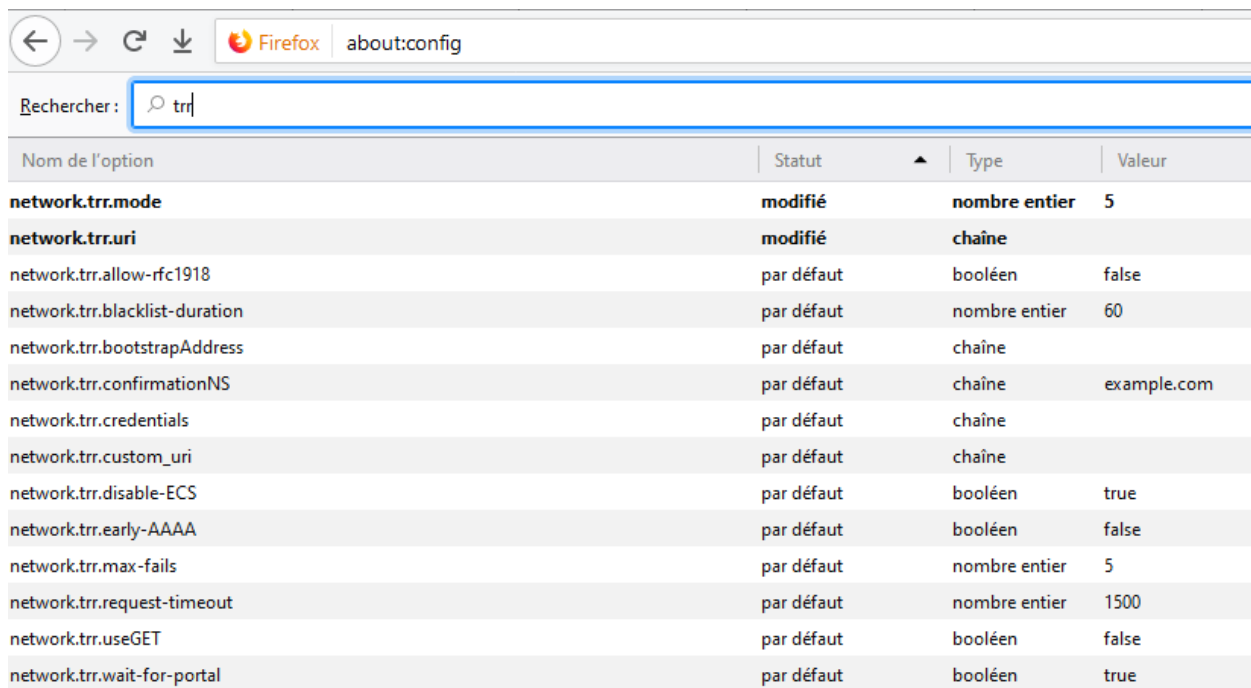
```
< HTTP/2 200
< server: h2o/2.3.0-DEV@6a25801e
< content-type: application/dns-message
< content-length: 49
[ On a la réponse \o/ ]
...
* DOH Host name: www.shaftinc.fr
* TTL: 86215 seconds
* DOH A: 37.187.2.182
* DOH AAAA: 2001:41d0:000a:02b6:0000:0000:0000:0001
[ Réponse décodée – C'est en binaire sinon ]
```

DNS sur HTTPS (DoH)

- Les clients DoH
 - ▶ curl
 - ▶ Navigateurs (Firefox, Chromium. Expérimental)
 - ▶ Quelques applis mobiles (libres ou privatives)
 - ▶ Des trucs pour les dévs (pour PHP, la glibc...)
 - ▶ Stubby, dans une prochaine version

DNS sur HTTPS (DoH)

- Sous Firefox



The screenshot shows the Firefox 'about:config' page with a search bar containing 'trr'. The search results are displayed in a table with the following columns: 'Nom de l'option', 'Statut', 'Type', and 'Valeur'.

Nom de l'option	Statut	Type	Valeur
network.trr.mode	modifié	nombre entier	5
network.trr.uri	modifié	chaîne	
network.trr.allow-rtc1918	par défaut	booléen	false
network.trr.blacklist-duration	par défaut	nombre entier	60
network.trr.bootstrapAddress	par défaut	chaîne	
network.trr.confirmationNS	par défaut	chaîne	example.com
network.trr.credentials	par défaut	chaîne	
network.trr.custom_uri	par défaut	chaîne	
network.trr.disable-ECS	par défaut	booléen	true
network.trr.early-AAAA	par défaut	booléen	false
network.trr.max-fails	par défaut	nombre entier	5
network.trr.request-timeout	par défaut	nombre entier	1500
network.trr.useGET	par défaut	booléen	false
network.trr.wait-for-portal	par défaut	booléen	true

Administrer un résolveur DoT/DoH

- Seulement quelques éléments
 - ▶ Donnés à titre indicatif
 - ▶ Pour lancer des vocations ? :-)
- Principaux logiciels pour DoT
 - ▶ Unbound
 - ▶ Knot Resolver
 - ▶ BIND ne gère pas directement DoT (et utiliser BIND n'est pas forcément une bonne idée)

Administrer un résolveur DoT/DoH

- Compétences nécessaires
 - ▶ Savoir configurer, administrer un résolveur et se débrouiller avec la génération de certificats
- Avant de se lancer il faut
 - ▶ Un certificat X.509 (et donc une clé privée RSA ou ECDSA). Let's Encrypt peut faire l'affaire, mais attention il est préférable de ne pas changer de la clé à chaque renouvellement du certificat (avec `certbot` par exemple)
 - ▶ Calculer le SPKI. Depuis la clé privée par exemple

Administrer un résolveur DoT/DoH

- Configuration
 - ▶ Assez simple sous Unbound et Knot Resolver
 - ▶ Typiquement 4-5 lignes de config
- Superviser !!!
 - ▶ Le paquet `getdns-utils` contient un utilitaire `getdns_server_mon`. Compatible Nagios, Naemon, Icinga, Shinken, Sensu
 - ▶ Outre les tests DNS classiques, tester **l'authentification** et **l'expiration du certificat** !

Administrer un résolveur DoT/DoH

- Et un résolveur DoH ?
 - ▶ Support expérimental dans Knot Resolver depuis la 4.0.0 (04/2019)
 - ▶ Beaucoup de bouts de code à droite à gauche sinon

Administrer un résolveur DoT/DoH

- Bonne pratique de déploiement (selon moi)
 - ▶ Activer DNSSEC (pour rappel :-)), utiliser le Padding (DoT), la réutilisation de connexion (DoT), interdire edns-client-subnet...
 - ▶ Être à jour niveau logiciel et suivre leurs actualités (et celle de l'IETF sur le sujet)
 - ▶ **Publier** : politique de vie privée (pas de logs de requêtes), SPKI, le logiciel & la version utilisée, la configuration (on peut censurer le chemin de la clé privée et du certificat) et faire un petit guide pour configurer un client
 - ▶ Superviser !!!

Conclusion

- Le chiffrement du trafic DNS va être nécessaire
- Il est assez accessible... aux geeks et autres libristes, mais pas encore au plus grand nombre
- Il faut des résolveurs de confiance
- Il ne faut pas laisser les Ogres faire n'importe quoi sur le sujet et imposer leurs solutions de déploiement

Merci !

