

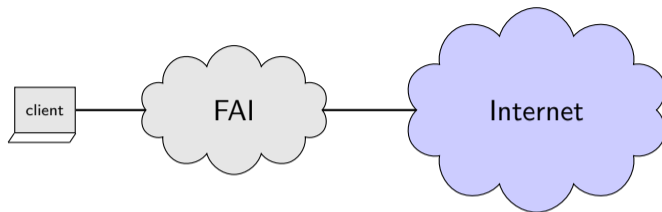
# Éléments d'analyse de DNS-over-HTTPS dans les navigateurs

François Contat (ANSSI) et Olivier Levillain (Telecom Paris)

Suite au projet de Julien Buttin Le Meur, Gregory Benassy et Valentin Penciolelli

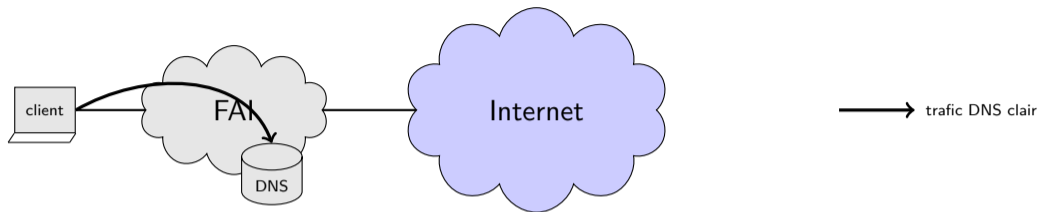
SSTIC 2020

## DNS - court rappel



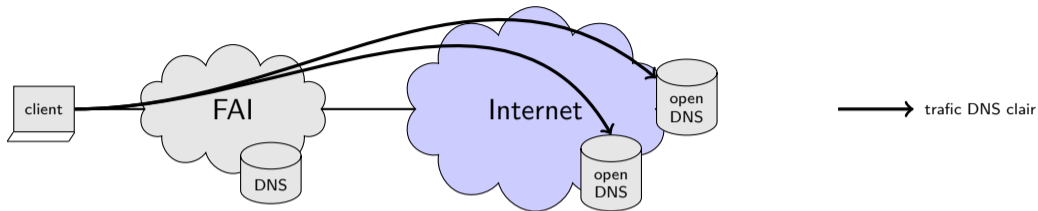
- ▶ usage classique : association nom / IP
- ▶ requêtes formulées auprès d'un DNS cache

## DNS - court rappel



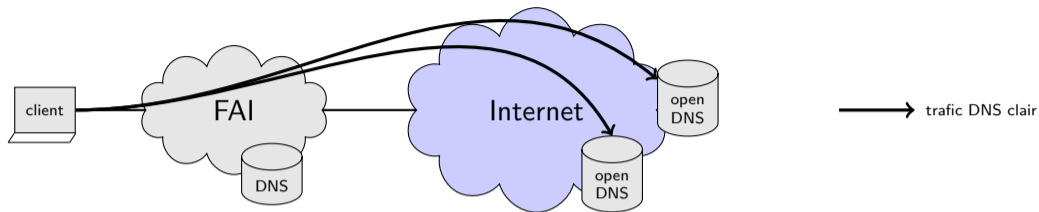
- ▶ usage classique : association nom / IP
- ▶ requêtes formulées auprès d'un DNS cache

## DNS - court rappel



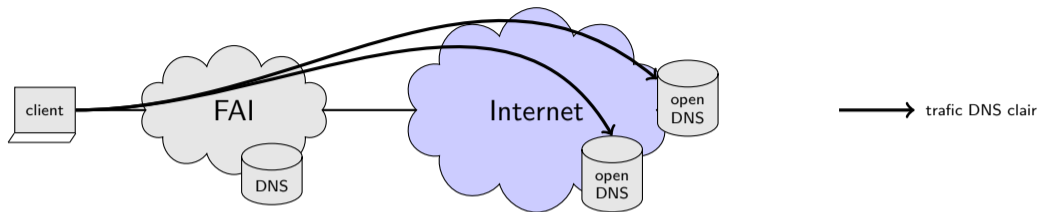
- ▶ usage classique : association nom / IP
- ▶ requêtes formulées auprès d'un DNS cache

# DNS - court rappel



- ▶ usage classique : association nom / IP
- ▶ requêtes formulées auprès d'un DNS cache
- ▶ DNS traditionnellement en clair
- ▶ Enjeux : vie privée / espionnage / censure

## DNS - court rappel



- ▶ usage classique : association nom / IP
- ▶ requêtes formulées auprès d'un DNS cache
- ▶ DNS traditionnellement en clair
- ▶ Enjeux : vie privée / espionnage / censure
- ▶ En l'absence de confiance dans les acteurs sur le chemin (FAI/états), comment protéger DNS ?

# DoT & DoH : DNS over {TLS,HTTPS}

DoT (RFC7858) - port 853 : tunnel TLS

## DoT & DoH : DNS over {TLS,HTTPS}

DoT (RFC7858) - port 853 : tunnel TLS

- + confidentialité des requêtes DNS
- + simple à mettre en œuvre
- + possibilité de réutiliser une connexion pour traiter plusieurs requêtes



## DoT & DoH : DNS over {TLS,HTTPS}

DoT (RFC7858) - port 853 : tunnel TLS

- + confidentialité des requêtes DNS
- + simple à mettre en œuvre
- + possibilité de réutiliser une connexion pour traiter plusieurs requêtes
- nécessite d'utiliser un nouveau port (853)

## DoT & DoH : DNS over {TLS,HTTPS}

DoT (RFC7858) - port 853 : tunnel TLS

- + confidentialité des requêtes DNS
- + simple à mettre en œuvre
- + possibilité de réutiliser une connexion pour traiter plusieurs requêtes
- nécessite d'utiliser un nouveau port (853)

DoH (RFC8484) - port 443 : tunnel TLS / HTTP1.1|2 / DNS|json

## DoT & DoH : DNS over {TLS,HTTPS}

DoT (RFC7858) - port 853 : tunnel TLS

- + confidentialité des requêtes DNS
- + simple à mettre en œuvre
- + possibilité de réutiliser une connexion pour traiter plusieurs requêtes
- nécessite d'utiliser un nouveau port (853)

DoH (RFC8484) - port 443 : tunnel TLS / HTTP1.1|2 / DNS|json

- + confidentialité des requêtes DNS
- + possibilité de réutiliser une connexion pour traiter plusieurs requêtes
- + réutilisation du port passe-partout (HTTPS, 443)

## DoT & DoH : DNS over {TLS,HTTPS}

DoT (RFC7858) - port 853 : tunnel TLS

- + confidentialité des requêtes DNS
- + simple à mettre en œuvre
- + possibilité de réutiliser une connexion pour traiter plusieurs requêtes
- nécessite d'utiliser un nouveau port (853)

DoH (RFC8484) - port 443 : tunnel TLS / HTTP1.1|2 / DNS|json

- + confidentialité des requêtes DNS
- + possibilité de réutiliser une connexion pour traiter plusieurs requêtes
- + réutilisation du port passe-partout (HTTPS, 443)
- complexité plus grande
- contournement par le navigateur de la configuration système
- interactions potentiellement complexes au sein d'un navigateur (cookies, secrets)

## DoT & DoH : DNS over {TLS,HTTPS}

DoT (RFC7858) - port 853 : tunnel TLS

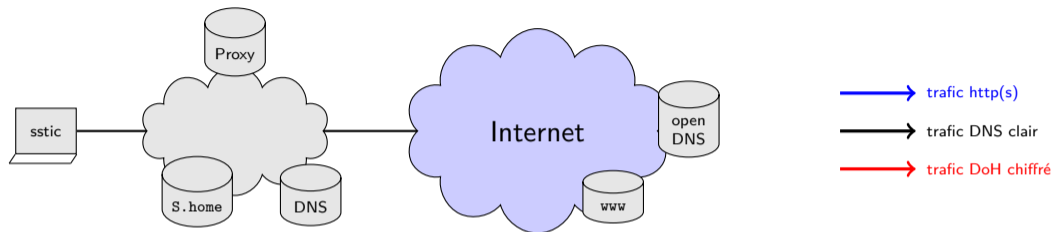
- + confidentialité des requêtes DNS
- + simple à mettre en œuvre
- + possibilité de réutiliser une connexion pour traiter plusieurs requêtes
- nécessite d'utiliser un nouveau port (853)

DoH (RFC8484) - port 443 : tunnel TLS / HTTP1.1|2 / DNS|json

- + confidentialité des requêtes DNS
- + possibilité de réutiliser une connexion pour traiter plusieurs requêtes
- + réutilisation du port passe-partout (HTTPS, 443)
- complexité plus grande
- contournement par le navigateur de la configuration système
- interactions potentiellement complexes au sein d'un navigateur (cookies, secrets)

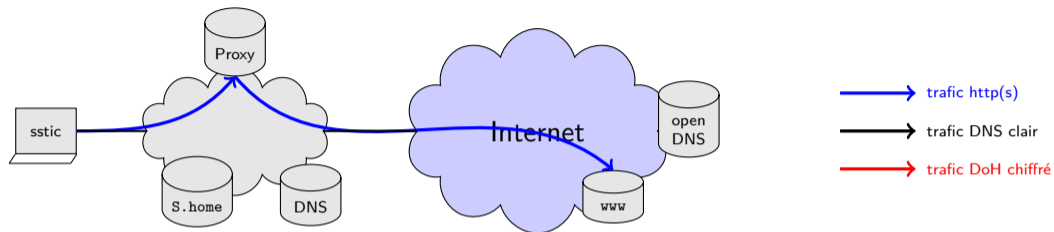
Deux navigateurs populaires ont fait le choix de DoH : Firefox et Chrom{e,ium}

## Exemple de scénario à éviter



Architecture classique avec des services internes

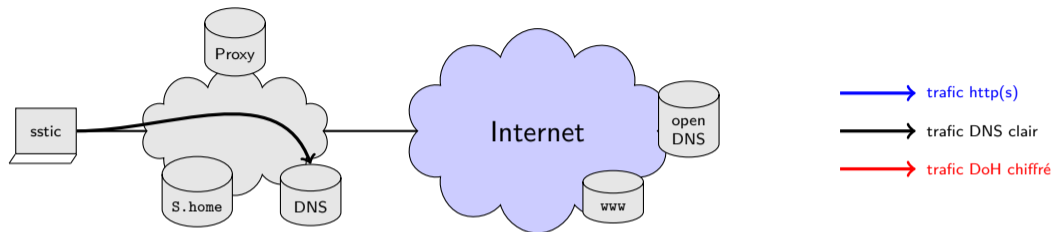
## Exemple de scénario à éviter



Architecture classique avec des services internes

- ▶ le trafic HTTP(S) passe par un proxy

## Exemple de scénario à éviter

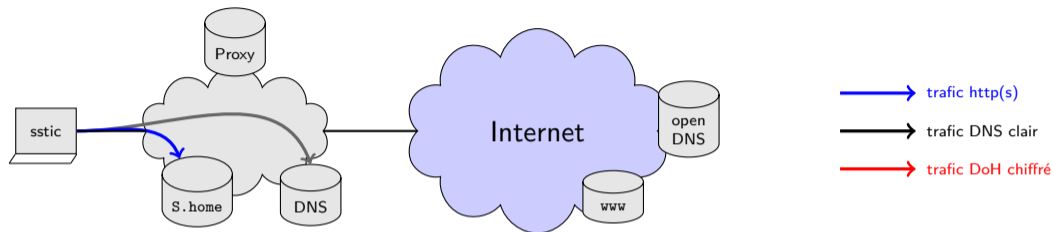


Architecture classique avec des services internes

- ▶ le trafic HTTP(S) passe par un proxy
- ▶ sauf pour certains domaines « locaux » (home)



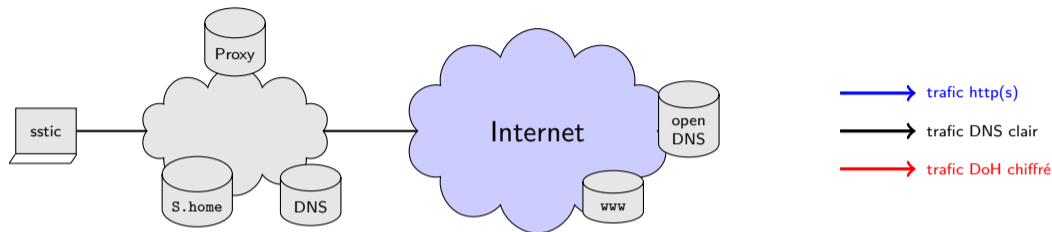
## Exemple de scénario à éviter



Architecture classique avec des services internes

- ▶ le trafic HTTP(S) passe par un proxy
- ▶ sauf pour certains domaines « locaux » (home)

## Exemple de scénario à éviter

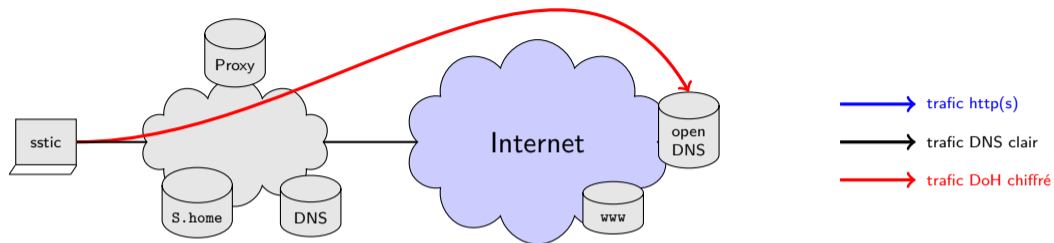


Architecture classique avec des services internes

- ▶ le trafic HTTP(S) passe par un proxy
- ▶ sauf pour certains domaines « locaux » (home)

Que se passe-t-il lorsque DoH est activé par défaut sur les navigateurs ?

## Exemple de scénario à éviter



Architecture classique avec des services internes

- ▶ le trafic HTTP(S) passe par un proxy
- ▶ sauf pour certains domaines « locaux » (home)

Que se passe-t-il lorsque DoH est activé par défaut sur les navigateurs ?

- ▶ imaginons que les requêtes pour `.home` partent en DoH à l'extérieur...

# Mise en place d'une plateforme de tests

## Dockerisation de différentes versions de Firefox sous Debian

- ▶ une version pour chaque numéro majeur entre FF 59 et FF 76
- ▶ compilation de `libnss3` avec `SSLKEYLOGFILE` pour pouvoir inspecter les échanges générés par le navigateur
- ▶ injection de paramètres via `user.js` dans le profil

## Virtualisation de Chromium et Firefox sous Windows

- ▶ une version pour chaque numéro majeur entre Chromium 78 et 81
- ▶ tests de FF 76
- ▶ extraction possible des secrets via une variable d'environnement

## Inspection des connexions dans diverses situations

# Étude de Firefox

## Fonctionnement de DoH dans Firefox

Les paramètres régissant le comportement DoH dans Firefox sont regroupés sous l'appellation Trusted Recursive Resolver (`network.trr.*`)

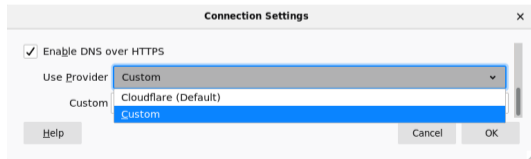
- ▶ `mode` : paramètre principal qui définit le comportement DoH
- ▶ `uri` : URI à utiliser pour envoyer les requêtes
- ▶ `bootstrapAddress` : adresse IP du serveur de l'URI (optionnel)
- ▶ etc.

## Fonctionnement de DoH dans Firefox

Les paramètres régissant le comportement DoH dans Firefox sont regroupés sous l'appellation Trusted Recursive Resolver (`network.trr.*`)

- ▶ `mode` : paramètre principal qui définit le comportement DoH
- ▶ `uri` : URI à utiliser pour envoyer les requêtes
- ▶ `bootstrapAddress` : adresse IP du serveur de l'URI (optionnel)
- ▶ etc.

Dans l'interface graphique, cela se résume à une case à cocher et à la définition de l'URI



## Les valeurs de `network.trr.mode`

network.trr.mode		Requêtes observées		Case cochée
Valeur	Description	DNS	DoH	
0	<i>Off (default)</i>	oui	non	non
1	<i>Reserved / Race mode</i>	oui	non (69-76)	<b>oui</b>
2	<i>First</i>	si DoH échoue	oui	oui
3	<i>Only</i>	non	oui	oui
4	<i>Reserved / Shadow mode</i>	oui	non (69-76)	<b>oui</b>
5	<i>Off (explicit)</i>	oui	non	non



# Contournement de DoH avec FF

Configuration locale (GUI)

- ▶ case à cocher

# Contournement de DoH avec FF

## Configuration locale (GUI)

- ▶ case à cocher

## Configuration locale (via GPO)

- ▶ `network.trr.mode` (tableau précédent)
- ▶ `network.trr.excluded-domains` (débrayage de DoH pour certains domaines)
- ▶ `network.trr.enable_when_{vpn,proxy,nrpt}_detected` (désactivation de DoH sous conditions)
  - ▶ l'objectif est d'éviter le scénario présenté au début
  - ▶ sous Linux, le paramètre est bien positionné, mais inefficace
  - ▶ sous Windows, manque de temps pour réaliser les tests

# Contournement de DoH avec FF

## Configuration locale (GUI)

- ▶ case à cocher

## Configuration locale (via GPO)

- ▶ `network.trr.mode` (tableau précédent)
- ▶ `network.trr.excluded-domains` (débrayage de DoH pour certains domaines)
- ▶ `network.trr.enable_when_{vpn,proxy,nrpt}_detected` (désactivation de DoH sous conditions)
  - ▶ l'objectif est d'éviter le scénario présenté au début
  - ▶ sous Linux, le paramètre est bien positionné, mais inefficace
  - ▶ sous Windows, manque de temps pour réaliser les tests

## Interactions réseau (administrateur)

- ▶ canari DNS `use-application-dns.net`
- ▶ blocage des connexions DoH  $\Rightarrow$  DNS en clair sauf si `mode=3`
- ▶ présentation d'un certificat invalide  $\Rightarrow$  DNS en clair sauf si `mode=3`

## Inquiétudes vis-à-vis de l'implémentation FF de DoH

De notre point de vue, l'implémentation de DoH pose problème dans Firefox

- ▶ contournement possible avec la politique de sécurité locale
- ▶ interface graphique en décalage avec les paramètres réels
- ▶ faux sentiment de sécurité en cas de dysfonctionnement de DoH
- ▶ transmission de données personnelles à un acteur extérieur

# Inquiétudes vis-à-vis de l'implémentation FF de DoH

De notre point de vue, l'implémentation de DoH pose problème dans Firefox

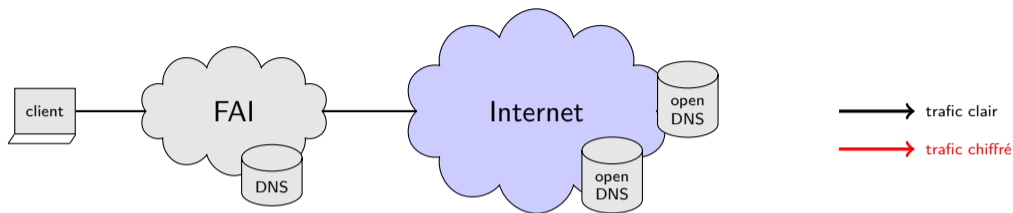
- ▶ contournement possible avec la politique de sécurité locale
- ▶ interface graphique en décalage avec les paramètres réels
- ▶ faux sentiment de sécurité en cas de dysfonctionnement de DoH
- ▶ transmission de données personnelles à un acteur extérieur

Autres points intéressants concernant les données émises

- ▶ possibilité d'ajouter le *User-Agent* et la langue utilisée dans les requêtes DoH
- ▶ sous Windows, chaque requête HTTP déclenche une requête DoH, ce qui amplifie les données envoyées

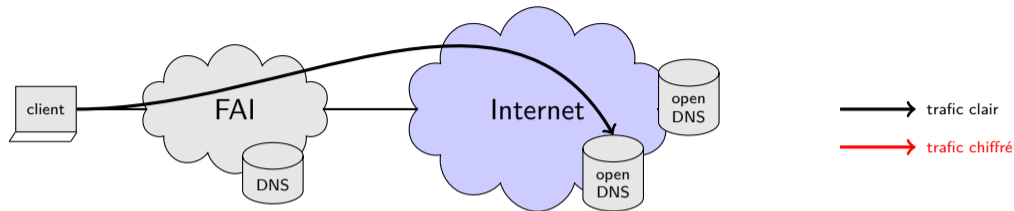
# Étude de Chromium

# Fonctionnement de DoH dans Chromium



- ▶ Liste en dur de 9 couples serveurs DNS/DoH connus :  
<https://www.chromium.org/developers/dns-over-https>
- ▶ Si le serveur DNS système dans la liste, alors Chromium utilise son pendant DoH
  - ▶ DoH intégré en version 79, mais à activer via `chrome flags#dnsoverhttps`
  - ▶ DoH activé par défaut en version 80
  - ▶ DoH devient paramétrable à partir de la version 83 dans les menus

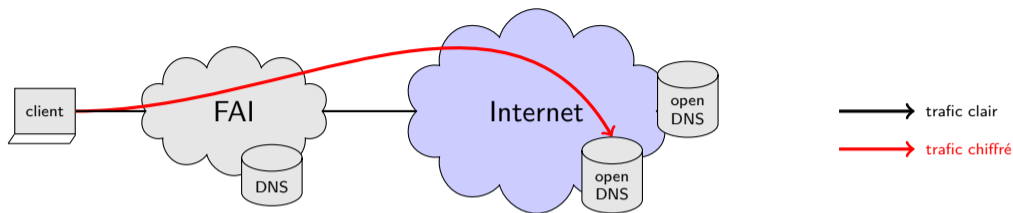
# Fonctionnement de DoH dans Chromium



- ▶ Liste en dur de 9 couples serveurs DNS/DoH connus :  
<https://www.chromium.org/developers/dns-over-https>
- ▶ Si le serveur DNS système dans la liste, alors Chromium utilise son pendant DoH
  - ▶ DoH intégré en version 79, mais à activer via `chrome flags#dnsoverhttps`
  - ▶ DoH activé par défaut en version 80
  - ▶ DoH devient paramétrable à partir de la version 83 dans les menus



# Fonctionnement de DoH dans Chromium



- ▶ Liste en dur de 9 couples serveurs DNS/DoH connus :  
<https://www.chromium.org/developers/dns-over-https>
- ▶ Si le serveur DNS système dans la liste, alors Chromium utilise son pendant DoH
  - ▶ DoH intégré en version 79, mais à activer via `chrome flags#dnsoverhttps`
  - ▶ DoH activé par défaut en version 80
  - ▶ DoH devient paramétrable à partir de la version 83 dans les menus

# Résultats Chromium

Conditions de tests :

1. DNS système positionné à 8.8.8.8
2. Sans action de l'utilisateur

<b>Version</b>	<b>Positionnement du <i>flag</i></b>	<b>Trafic DoH</b>	<b>Comportement face à un certificat invalide</b>
78	Inexistant	non	inapplicable
79	À activer manuellement	non	DNS clair
80	Activé par défaut	oui	DNS clair
81 (actuelle)	Activé par défaut	oui	DNS clair

# Conclusion

## Conclusion

DoT/DoH sont des protocoles d'importance et dont l'utilité ne fait aucun doute.

Toutefois, la philosophie de l'implémentation a son importance.

Volonté de Mozilla d'imposer à tout prix a un coût :

- ▶ Manque de visibilité et transparence sur le bon fonctionnement
- ▶ Transfert de données important vers un tiers mal connu

Sujet d'inquiétude avec les deux implémentations : verbosité des clients à destination des serveurs DoH.

À l'avenir :

- ▶ Support de DoH dans Microsoft Windows
- ▶ Chromium (83) aura-t-il une remontée d'alerte en cas de défaillance de DoH ?
- ▶ Une évolution de l'affichage et de la GUI de Mozilla Firefox sera-t-elle faite ?

Perspectives :

- ▶ Déchiffrer trafic gQUIC chromium
- ▶ Suivre les évolutions de DoH dans les OS et navigateurs

# Si vous voulez faire du DoH, ce que l'on conseille

1. Il existe des serveurs tenus par des organismes à but non lucratif :
  - ▶ <https://doh.42l.fr/dns-query>
  - ▶ <https://ldn-fai.net/dns-query>
  - ▶ <https://odvr.nic.cz/doh>
  - ▶ <https://doh.powerdns.org/>
  - ▶ <https://dns.hostux.net/dns-query>
2. Vous pouvez aussi utiliser votre propre instance
3. La concentration de données, DNS, navigation, usages doit rester au cœur de vos choix
4. Pour Firefox, aligner le contenu des options `network.trr.excluded_domains` et `network.trr.network_no_proxies_on`