Réflexions sur DNS

Olivier Levillain

ANSSI/CyberEdu

Journée CyberEdu à Nantes
19 avril 2018

Table des matières

Table des matières

Qu'est-ce que le *Domain Name System* ?

Le DNS est une base de données hiérarchique et décentralisée

- les clés sont des « noms de domaine »
- les enregistrements sont de types variés

Qu'est-ce que le *Domain Name System* ?

Le DNS est une base de données hiérarchique et décentralisée

- les clés sont des « noms de domaine »
- les enregistrements sont de types variés
- la hiérarchie est définie par des zones d'autorité
  - la racine, notée `.`
  - les *Top-Level Domains* (TLD), tels que `.fr`
  - les *Second-Level Domains* (SLD), attribués à une entité, tels que `cyberedu.fr`
  - au sein de ces entités, d'autres noms peuvent être alloués librement (`www.cyberedu.fr`)

Qu'est-ce que le *Domain Name System* ?

Le DNS est une base de données hiérarchique et décentralisée

- les clés sont des « noms de domaine »
- les enregistrements sont de types variés
- la hiérarchie est définie par des zones d'autorité
  - la racine, notée `.`
  - les *Top-Level Domains* (TLD), tels que `.fr`
  - les *Second-Level Domains* (SLD), attribués à une entité, tels que `cyberedu.fr`
  - au sein de ces entités, d'autres noms peuvent être alloués librement (`www.cyberedu.fr`)

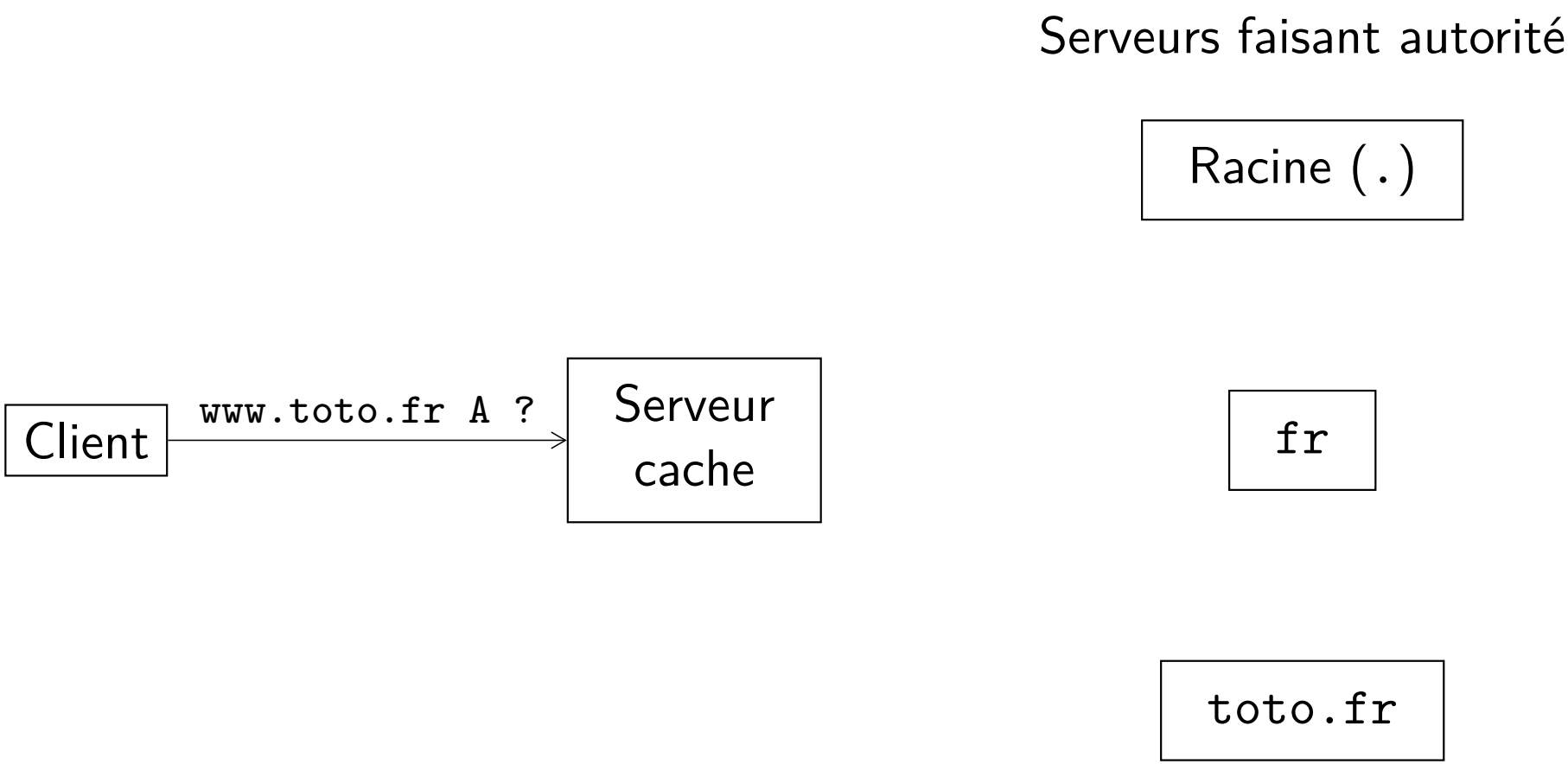
  - parfois les SLD ressemblent à des TLD (`.gouv.fr` ou `co.uk`)
  - cela pose des soucis avec la *Same Origin Policy*
  - il existe une liste d'*effective TLD* (cette liste est maintenue par Mozilla)

Contenu du DNS

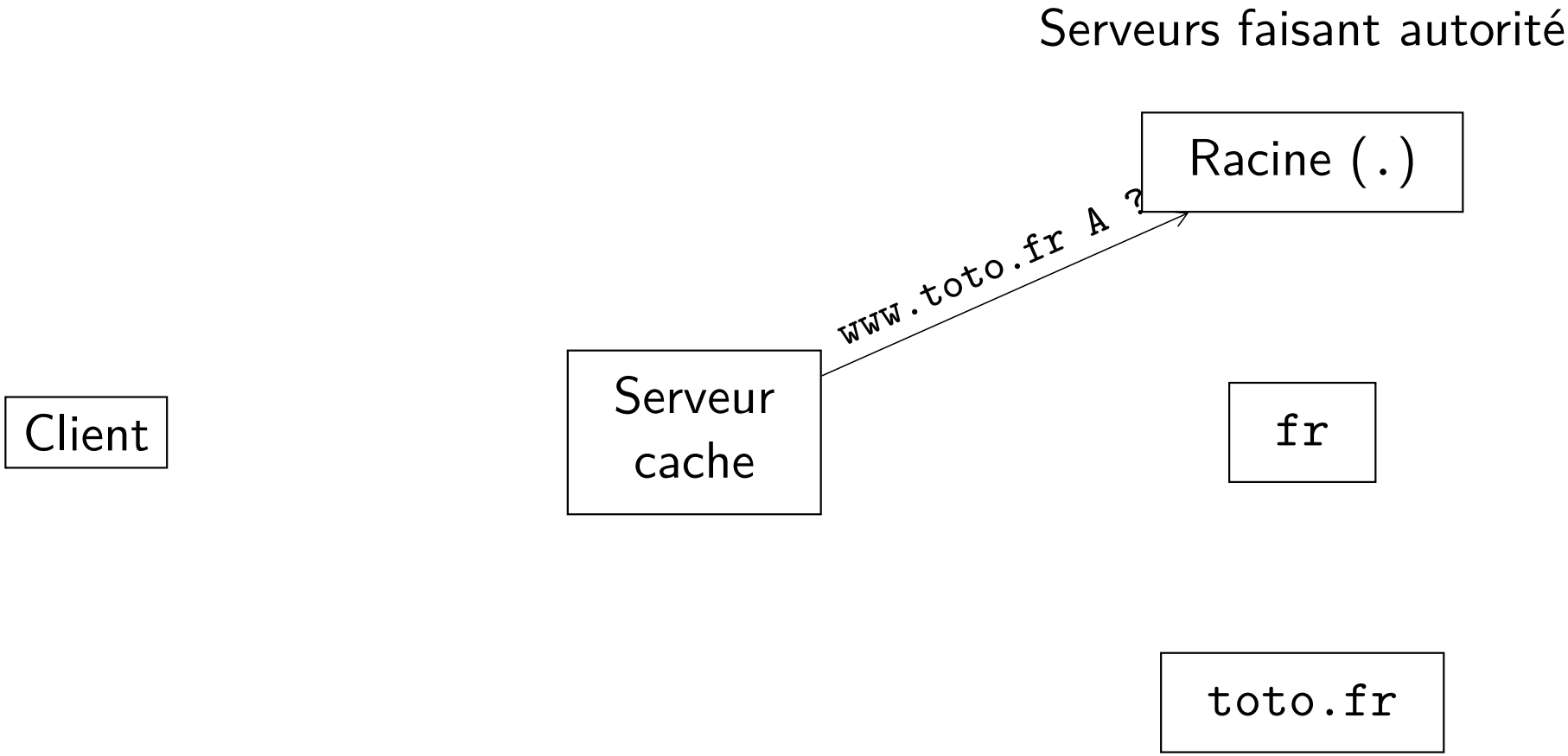
Voici quelques types de *ressource records* :

- adresses IP (`A` et `AAAA`)
- alias (p.ex. : `CNAME`)
- noms de machines assurant un service (`NS`, `MX`, `SRV`)
- enregistrements liés à DNSSEC (`DNSKEY`, `DS`, `NSEC`, `RRSIG`)
- autres enregistrements liés à un mécanisme de sécurité (p.ex. : `DKIM`, `SPF`, `TLSA`)

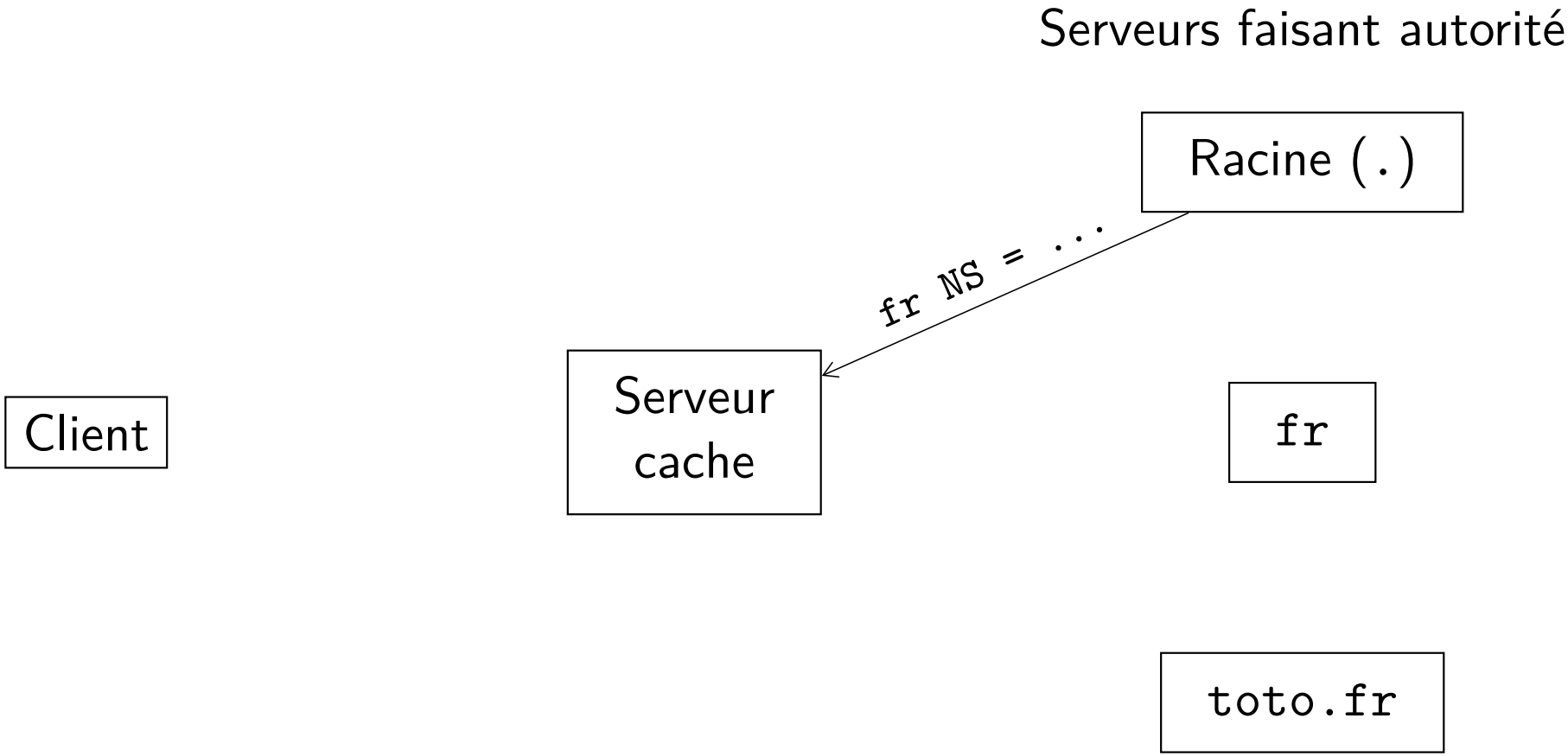
Acteurs du DNS



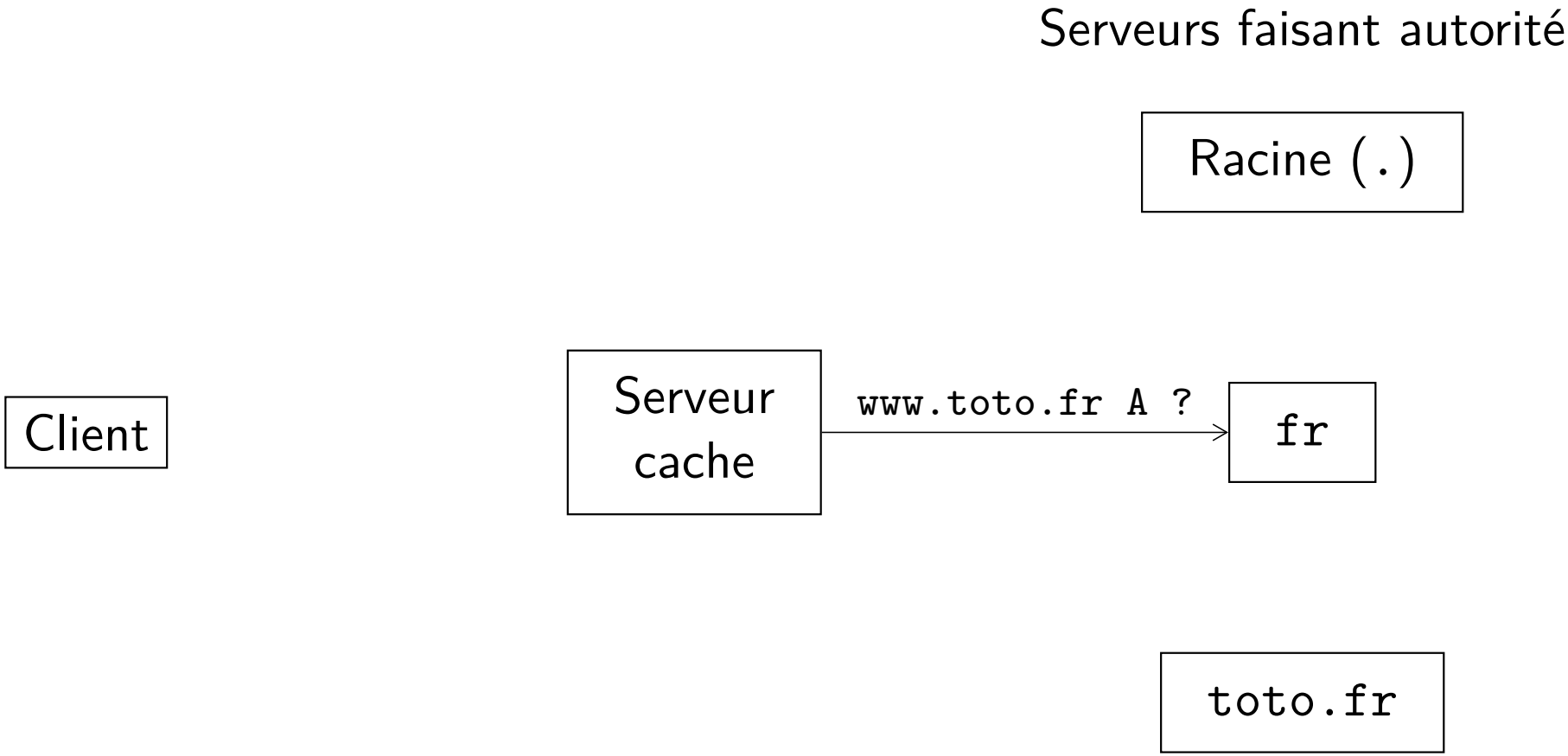
Acteurs du DNS



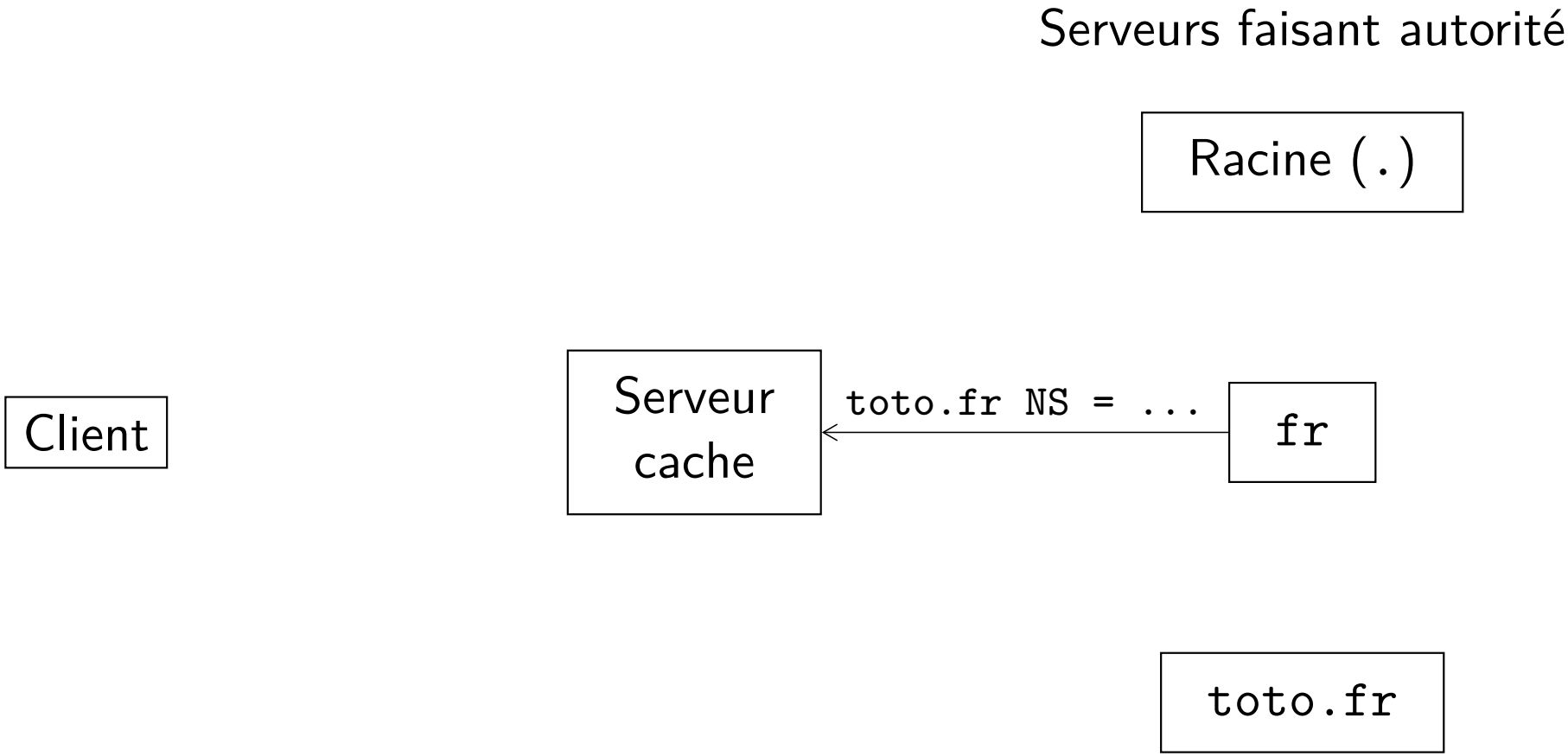
Acteurs du DNS



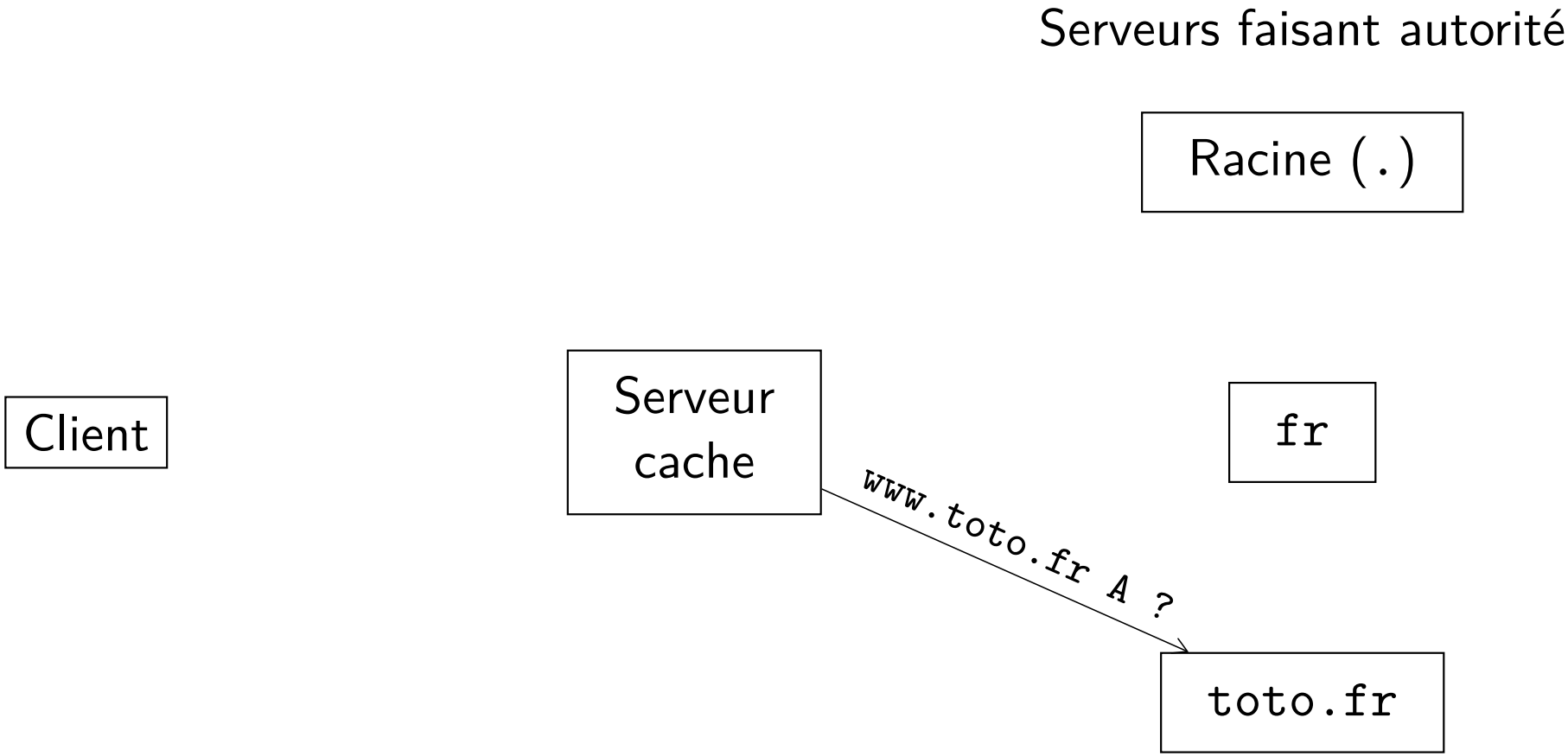
Acteurs du DNS



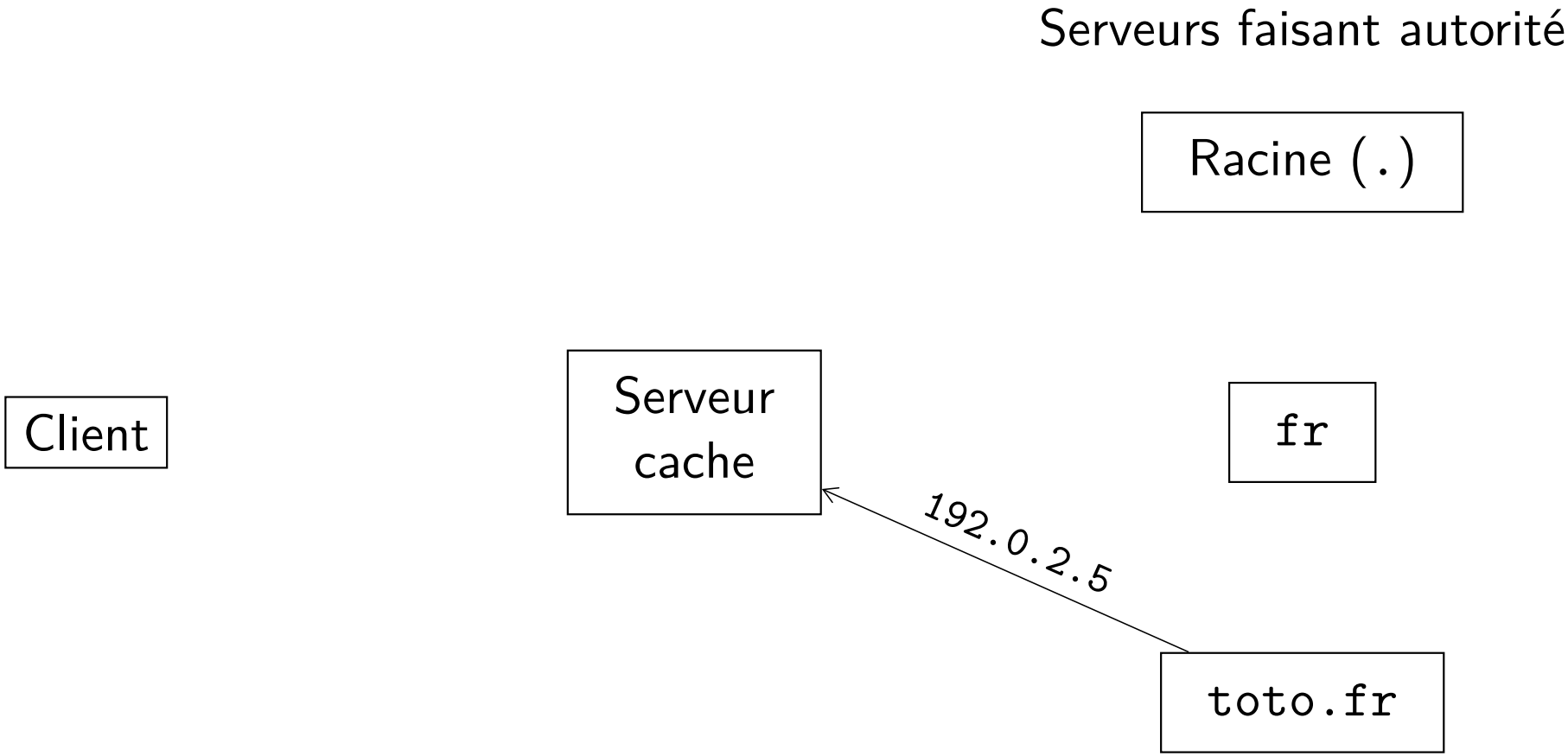
Acteurs du DNS



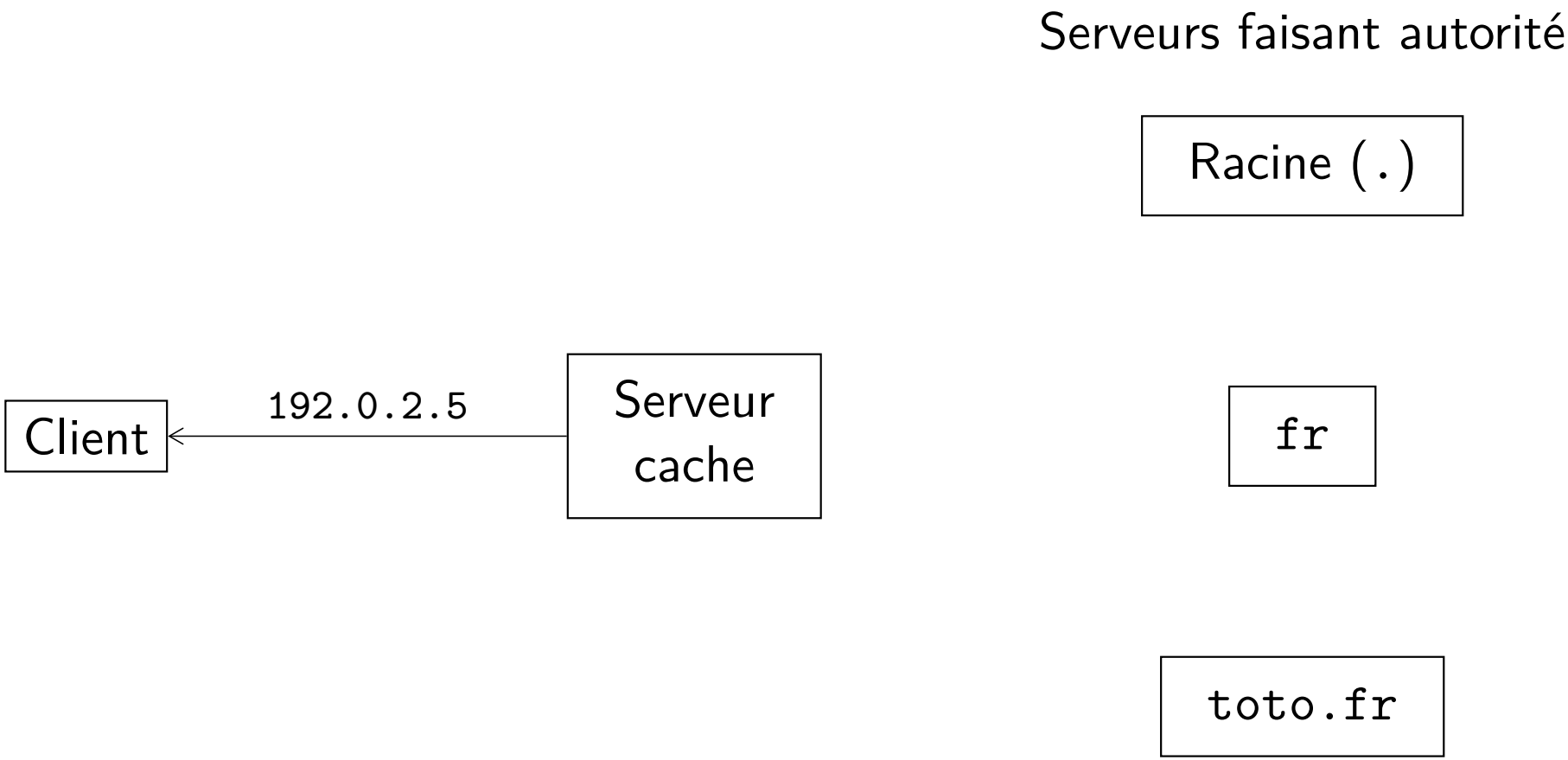
Acteurs du DNS



Acteurs du DNS



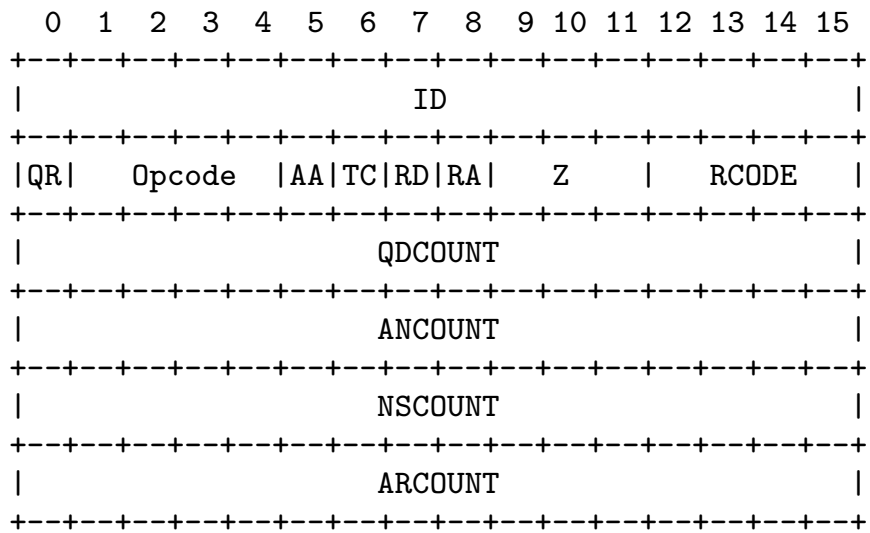
Acteurs du DNS



Protocole (1/2)

Un paquet DNS (requête ou réponse) contient

- un identifiant, le `qid` (16 bits)
- un champ `qr` indiquant si le paquet est une requête ou une réponse
- un code de retour (pertinent si `qr=1`)
- des champs complémentaires
- une ou plusieurs requêtes
- une ou plusieurs réponses (pertinent si `qr=1`)



Protocole (2/2)

Le fonctionnement classique est le suivant :

Protocole (2/2)

Le fonctionnement classique est le suivant :

- envoi d'un paquet UDP de type requête (`qr=0`)
  - `qid` est choisi par le client
  - seule la section `query` est renseignée et contient une requête

Protocole (2/2)

Le fonctionnement classique est le suivant :

- envoi d'un paquet UDP de type requête (`qr=0`)
  - `qid` est choisi par le client
  - seule la section `query` est renseignée et contient une requête
- le serveur répond avec un paquet réponse (`qr=1`)
  - le `qid` est recopié
  - le code de retour est renseigné
  - la section `query` est recopiée
  - si tout s'est bien passé, les sections de réponse sont remplies

Protocole (2/2)

Le fonctionnement classique est le suivant :

- envoi d'un paquet UDP de type requête (`qr=0`)
  - `qid` est choisi par le client
  - seule la section `query` est renseignée et contient une requête
- le serveur répond avec un paquet réponse (`qr=1`)
  - le `qid` est recopié
  - le code de retour est renseigné
  - la section `query` est recopiée
  - si tout s'est bien passé, les sections de réponse sont remplies

- DNS fonctionne de manière similaire sur TCP
- il est théoriquement possible d'envoyer plusieurs requête dans un même message, ou dans une même connexion
- il existe des extensions

Table des matières

Attaque sur l'intégrité des réponses

Plusieurs méthodes possibles

- attaquant en coupure *(man in the middle)*
- attaquant en aveugle sur les paquets UDP
  - il doit répondre avant le serveur légitime
  - il doit *deviner* certains champs

Attaque sur l'intégrité des réponses

Plusieurs méthodes possibles

- attaquant en coupure *(man in the middle)*
- attaquant en aveugle sur les paquets UDP
  - il doit répondre avant le serveur légitime
  - il doit *deviner* certains champs

Note : il est facile de déclencher une requête DNS/HTTP vers un serveur arbitraire pour un attaquant actif

Attaque sur l'intégrité des réponses

Plusieurs méthodes possibles

- attaquant en coupure *(man in the middle)*
- attaquant en aveugle sur les paquets UDP
  - il doit répondre avant le serveur légitime
  - il doit *deviner* certains champs

Note : il est facile de déclencher une requête DNS/HTTP vers un serveur arbitraire pour un attaquant actif

Plusiers cibles

- le client final (*stub resolver*)
  - la réponse obtenue est incorrecte
- le serveur cache (récursif)
  - on parle alors d'empoisonnement de cache
  - l'impact s'étend à l'ensemble des clients utilisant ce serveur récursif

Astuce de Kaminsky

En cas d'échec de l'empoisonnement, il faut attendre que l'enregistrement obtenu expire

- dans certains cas, c'est immédiat
- parfois, cela dépend de la durée de vie d'un programme
- pour un serveur cache, cela peut durer plusieurs jours !

Astuce de Kaminsky

En cas d'échec de l'empoisonnement, il faut attendre que l'enregistrement obtenu expire

- dans certains cas, c'est immédiat
- parfois, cela dépend de la durée de vie d'un programme
- pour un serveur cache, cela peut durer plusieurs jours !

Il existe une astuce pour l'attaquant en aveugle

- au lieu de contrefaire la réponse à une requête `A` vers `www.cyberedu.fr`
- on peut déclencher une requête vers un nom de domaine aléatoire `fdghusbes.cyberedu.fr` et tenter de répondre avant le serveur...
- ... en incluant des informations sur le nom de domaine `cyberedu.fr`

Astuce de Kaminsky

En cas d'échec de l'empoisonnement, il faut attendre que l'enregistrement obtenu expire

- dans certains cas, c'est immédiat
- parfois, cela dépend de la durée de vie d'un programme
- pour un serveur cache, cela peut durer plusieurs jours !

Il existe une astuce pour l'attaquant en aveugle

- au lieu de contrefaire la réponse à une requête `A` vers `www.cyberedu.fr`
- on peut déclencher une requête vers un nom de domaine aléatoire `fdghusbes.cyberedu.fr` et tenter de répondre avant le serveur...
- ... en incluant des informations sur le nom de domaine `cyberedu.fr`
- l'attaquant bénéficie de plusieurs essais

Astuce de Kaminsky

En cas d'échec de l'empoisonnement, il faut attendre que l'enregistrement obtenu expire

- dans certains cas, c'est immédiat
- parfois, cela dépend de la durée de vie d'un programme
- pour un serveur cache, cela peut durer plusieurs jours !

Il existe une astuce pour l'attaquant en aveugle

- au lieu de contrefaire la réponse à une requête `A` vers `www.cyberedu.fr`
- on peut déclencher une requête vers un nom de domaine aléatoire `fdghusbes.cyberedu.fr` et tenter de répondre avant le serveur...
- ... en incluant des informations sur le nom de domaine `cyberedu.fr`
- l'attaquant bénéficie de plusieurs essais
- une fois l'attaque réussie, l'attaquant maîtrise la zone complète

Contremesures (1/2)

Face à un attaquant qui répond en aveugle

Contremesures (1/2)

Face à un attaquant qui répond en aveugle

- il faut randomiser le `qid`
  - on obtient alors 16 bits d'entropie
  - attention : de vieilles implémentations utilisaient un compteur...

Contremesures (1/2)

Face à un attaquant qui répond en aveugle

- il faut randomiser le `qid`
  - on obtient alors 16 bits d'entropie
  - attention : de vieilles implémentations utilisaient un compteur...
- il faut randomiser le port source UDP
  - comportement généralisé après la présentation de Kaminsky en 2008
  - on peut obtenir jusqu'à 16 bits supplémentaires

Contremesures (1/2)

Face à un attaquant qui répond en aveugle

- il faut randomiser le `qid`
  - on obtient alors 16 bits d'entropie
  - attention : de vieilles implémentations utilisaient un compteur...
- il faut randomiser le port source UDP
  - comportement généralisé après la présentation de Kaminsky en 2008
  - on peut obtenir jusqu'à 16 bits supplémentaires
- certaines implémentations jouent sur la casse
  - cela n'a normalement pas d'impact sur la réponse
  - mais le serveur doit répéter la question à l'identique
  - en théorie, on obtient un bit d'entropie supplémentaire par lettre
  - en pratique, des incompatibilités existent

Contremesures (1/2)

Face à un attaquant qui répond en aveugle

- il faut randomiser le `qid`
  - on obtient alors 16 bits d'entropie
  - attention : de vieilles implémentations utilisaient un compteur...
- il faut randomiser le port source UDP
  - comportement généralisé après la présentation de Kaminsky en 2008
  - on peut obtenir jusqu'à 16 bits supplémentaires
- certaines implémentations jouent sur la casse
  - cela n'a normalement pas d'impact sur la réponse
  - mais le serveur doit répéter la question à l'identique
  - en théorie, on obtient un bit d'entropie supplémentaire par lettre
  - en pratique, des incompatibilités existent
- forcer l'utilisation TCP
  - l'attaquant doit alors deviner les numéros de séquence sur 32 bits
  - en pratique, DNS sur TCP est parfois bloqué

Contremesures (2/2)

Que faire

- quand l'entropie n'est pas suffisante ?
- quand l'attaquant est en coupure ?

Contremesures (2/2)

Que faire

- quand l'entropie n'est pas suffisante ?
- quand l'attaquant est en coupure ?

Plusieurs propositions existent

- TSIG (motif d'intégrité symétrique)
- DNSCurve (mécanisme ad-hoc reposant sur Curve25519)
- utiliser DTLS ou TLS (permet d'assurer aussi la confidentialité)
- utiliser DNSSEC (signature des enregistrements)

Contremesures (2/2)

Que faire

- quand l'entropie n'est pas suffisante ?
- quand l'attaquant est en coupure ?

Plusieurs propositions existent

- TSIG (motif d'intégrité symétrique)
- DNSCurve (mécanisme ad-hoc reposant sur Curve25519)
- utiliser DTLS ou TLS (permet d'assurer aussi la confidentialité)
- utiliser DNSSEC (signature des enregistrements)

Dans certains cas, des contrôles de cohérences peuvent aussi être effectués par le client pour détecter des anomalies

DNSSEC

Fonctionnement de DNSSEC

- les réponses sont signées (`RRSIG`)

DNSSEC

Fonctionnement de DNSSEC

- les réponses sont signées (`RRSIG`)
- les serveurs publient les clés permettant de vérifier les signatures (`DNSKEY`)

DNSSEC

Fonctionnement de DNSSEC

- les réponses sont signées (`RRSIG`)
- les serveurs publient les clés permettant de vérifier les signatures (`DNSKEY`)
- afin de garantir le chaînage, l'empreinte des clés d'une zone est stockée dans la zone parente (`DS`)

DNSSEC

Fonctionnement de DNSSEC

- les réponses sont signées (`RRSIG`)
- les serveurs publient les clés permettant de vérifier les signatures (`DNSKEY`)
- afin de garantir le chaînage, l'empreinte des clés d'une zone est stockée dans la zone parente (`DS`)
- les serveurs récursifs validant connaissent les clés de la racine

DNSSEC

Fonctionnement de DNSSEC

- les réponses sont signées (`RRSIG`)
- les serveurs publient les clés permettant de vérifier les signatures (`DNSKEY`)
- afin de garantir le chaînage, l'empreinte des clés d'une zone est stockée dans la zone parente (`DS`)
- les serveurs récursifs validant connaissent les clés de la racine

- il existe un nouveau type d'enregistrement pour indiquer de manière intègre une réponse négative (`NSEC`)

DNSSEC

Fonctionnement de DNSSEC

- les réponses sont signées (`RRSIG`)
- les serveurs publient les clés permettant de vérifier les signatures (`DNSKEY`)
- afin de garantir le chaînage, l'empreinte des clés d'une zone est stockée dans la zone parente (`DS`)
- les serveurs récursifs validant connaissent les clés de la racine

- il existe un nouveau type d'enregistrement pour indiquer de manière intègre une réponse négative (`NSEC`)
- `NSEC3`... `NSEC5`...

Table des matières

DDoS : *Distributed Denial of Service*

Déni de service distribués

- un attaquant provoque l'envoi de nombreux paquets réseau
- vers une même cible
- depuis un ensemble important de sources

DDoS : *Distributed Denial of Service*

Déni de service distribués

- un attaquant provoque l'envoi de nombreux paquets réseau
- vers une même cible
- depuis un ensemble important de sources

Intérêt pour l'attaquant

- force de frappe démultipliée
- attaque difficile à bloquer

DDoS : *Distributed Denial of Service*

Déni de service distribués

- un attaquant provoque l'envoi de nombreux paquets réseau
- vers une même cible
- depuis un ensemble important de sources

Intérêt pour l'attaquant

- force de frappe démultipliée
- attaque difficile à bloquer

Exemples

- un *botnet*
- attaques par réflexion/amplification

Attaques par réflexion

Principe

- un attaquant envoie un paquet UDP en usurpant l'adresse source de sa victime
- le serveur qui reçoit la requête répond à la victime

Intérêt

- l'attaquant réel est difficile à identifier
- en jouant sur plusieurs serveurs rebonds, il est possible de concentrer les flux réseau

Attaques par amplification

Dans certains cas, la réponse du serveur est plus grande que la requête : on parle d'amplification

Protocole	**Facteur**	**Commentaire**
DNS	30-50	Recopie de la question, réponses longues
NTP	500	Modes 6 *(control)* et 7 *(private)*
CharGEN	350	Génération de contenu
Quake	64	...
Memcached	10k-50k	Serveurs de cache d'objets

Origine du problème

Il existe deux grandes raisons pour laquelle l'attaque marche

Origine du problème

Il existe deux grandes raisons pour laquelle l'attaque marche

- l'usurpation d'IP source est possible
  - on en revient à la mise en œuvre de BCP 38
  - avec UDP, seul un filtrage à la source peut aider à résoudre le problème

Origine du problème

Il existe deux grandes raisons pour laquelle l'attaque marche

- l'usurpation d'IP source est possible
  - on en revient à la mise en œuvre de BCP 38
  - avec UDP, seul un filtrage à la source peut aider à résoudre le problème
- l'existence de protocoles (et de serveurs) bavards
  - un serveur ne devrait pas a priori produire (beaucoup) plus de trafic que celui qu'il a reçu dans le paquet initial
  - parfois on peut authentifier le client
  - on peut aussi forcer un aller-retour ou une *preuve de travail*

Réponse dans le cas de DNS

Modifications possibles côté serveur

Réponse dans le cas de DNS

Modifications possibles côté serveur

- introduire des limitations dans les réponses *(rate limiting)*

Réponse dans le cas de DNS

Modifications possibles côté serveur

- introduire des limitations dans les réponses *(rate limiting)*
- réduire la taille des enregistrements
  - par exemple, en ne donnant pas toutes les informations additionnelles

Réponse dans le cas de DNS

Modifications possibles côté serveur

- introduire des limitations dans les réponses *(rate limiting)*
- réduire la taille des enregistrements
  - par exemple, en ne donnant pas toutes les informations additionnelles
- forcer l'utilisation de TCP
  - envoi d'une réponse tronquée (éventuellement vide)
  - attention aux problèmes de compatibilité

Table des matières

Retour sur DNSSEC

DNSSEC garantit l'intégrité des enregistrements, mais

Retour sur DNSSEC

DNSSEC garantit l'intégrité des enregistrements, mais

- cela ne concerne en général pas le dernier lien entre le *stub resolver* et le serveur cache

Retour sur DNSSEC

DNSSEC garantit l'intégrité des enregistrements, mais

- cela ne concerne en général pas le dernier lien entre le *stub resolver* et le serveur cache
- le déploiement est encore loin d'être universel

Retour sur DNSSEC

DNSSEC garantit l'intégrité des enregistrements, mais

- cela ne concerne en général pas le dernier lien entre le *stub resolver* et le serveur cache
- le déploiement est encore loin d'être universel

- DNSSEC produit naturellement des réponses énormes
  - ajout des clés et des signatures
  - explosion des réponses lors du roulement de clés
  - ainsi, corriger l'empoisonnement de cache favorise l'amplification...

Retour sur le *rate limiting*

Ne pas répondre à toutes les requêtes

Retour sur le *rate limiting*

Ne pas répondre à toutes les requêtes

- réduit naturellement la bande passante dans une attaque par amplification...

Retour sur le *rate limiting*

Ne pas répondre à toutes les requêtes

- réduit naturellement la bande passante dans une attaque par amplification...
- ... mais introduit un délai pendant lequel l'attaquant peut empoisonner le cache !

Retour sur le *rate limiting*

Ne pas répondre à toutes les requêtes

- réduit naturellement la bande passante dans une attaque par amplification...
- ... mais introduit un délai pendant lequel l'attaquant peut empoisonner le cache !

Atténuer les effets d'amplification favorise l'empoisonnement de cache...

Trouver l'équilibre

Sécuriser les protocoles est un équilibre à trouver entre

- la sécurité (et parfois plusieurs attaques différentes)
- la performance
- la compatibilité

Trouver l'équilibre

Sécuriser les protocoles est un équilibre à trouver entre

- la sécurité (et parfois plusieurs attaques différentes)
- la performance
- la compatibilité

La réponse à apporter est souvent complexe

Trouver l'équilibre

Sécuriser les protocoles est un équilibre à trouver entre

- la sécurité (et parfois plusieurs attaques différentes)
- la performance
- la compatibilité

La réponse à apporter est souvent complexe

Pour DNS, un plan d'action pourrait être

- poursuivre et favoriser le déploiement de DNSSEC
- mettre en place des moyens de détection d'attaques DDoS
- tronquer les réponses et forcer TCP en cas d'attaques

Trouver l'équilibre

Sécuriser les protocoles est un équilibre à trouver entre

- la sécurité (et parfois plusieurs attaques différentes)
- la performance
- la compatibilité

La réponse à apporter est souvent complexe

Pour DNS, un plan d'action pourrait être

- poursuivre et favoriser le déploiement de DNSSEC
- mettre en place des moyens de détection d'attaques DDoS
- tronquer les réponses et forcer TCP en cas d'attaques
- utiliser des protocoles sécurisés au-dessus (pour qu'une information invalide soit tout de même détectée plus haut)

Table des matières

Résolution de noms de domaine avec DNS (1/2)

Problèmes de sécurité liés à DNS

- intégrité des réponses (empoisonnement de cache)
- source de dénis de service distribués

Causes

- usurpation d'IP source facilitée par l'utilisation d'UDP
- requêtes DNS en grande partie prédictible

Résolution de noms de domaine avec DNS (2/2)

DNSSEC

- \+ protection en intégrité des réponses (sauf le dernier lien)
- − complexité de mise en oeuvre
- − réponses plus longues (amplification des DDoS)

Résolution de noms de domaine avec DNS (2/2)

DNSSEC

- \+ protection en intégrité des réponses (sauf le dernier lien)
- − complexité de mise en oeuvre
- − réponses plus longues (amplification des DDoS)

Limiter les réponses d'un serveur DNS

- \+ atténuation des effets d'un DDoS
- − certains empoisonnements de cache deviennent plus facile
- − dégradation possible du service légitime

Résolution de noms de domaine avec DNS (2/2)

DNSSEC

- \+ protection en intégrité des réponses (sauf le dernier lien)
- − complexité de mise en oeuvre
- − réponses plus longues (amplification des DDoS)

Limiter les réponses d'un serveur DNS

- \+ atténuation des effets d'un DDoS
- − certains empoisonnements de cache deviennent plus facile
- − dégradation possible du service légitime

TCP (usurpation d'IP source plus difficile)

- \+ protection (légère) en intégrité
- \+ atténuation des effets d'un DDoS
- − nombreux *firewalls* ou serveurs mal configurés
- − charge plus importante sur les serveurs

Résolution de noms de domaine avec DNS (2/2)

DNSSEC

- \+ protection en intégrité des réponses (sauf le dernier lien)
- − complexité de mise en oeuvre
- − réponses plus longues (amplification des DDoS)

Limiter les réponses d'un serveur DNS

- \+ atténuation des effets d'un DDoS
- − certains empoisonnements de cache deviennent plus facile
- − dégradation possible du service légitime

TCP (usurpation d'IP source plus difficile)

- \+ protection (légère) en intégrité
- \+ atténuation des effets d'un DDoS
- − nombreux *firewalls* ou serveurs mal configurés
- − charge plus importante sur les serveurs

Réparer le DNS est un problème d'ingénierie complexe !

Questions ?

Merci de votre attention.