Attaques et sécurisation des couches OSI
DNS et DNSSEC

Carlos Aguilar
carlos.aguilar-melchor@isae-supaero.fr
Benoît Morgan
benoit.morgan@enseeiht.fr

IRIT-IRT

Remerciements

Merci à :

- Carlos Aguilar-Melchor
- Cédric Blancher (EADS)
- Dan Boneh de Stanford University
- Pierre-François Bonnefoi du Master CRYPTIS à Limoges
- Céline Boyer (Canal+)
- Julien Cartigny du Master CRYPTIS à Limoges
- Ron Rivest du M.I.T.
- Matthieu Herrb - ingénieur de recherche du LAAS-CNRS

Sources

[1] http://fr.wikipedia.org/wiki/Domain_Name_System

[2] http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html

[3] http://homepages.laas.fr/matthieu/talks/ttnn-dns.pdf

[4] https://www.xudongz.com/blog/2017/idn-phishing/

[5] https://tools.ietf.org/html/rfc3490

Plan

1. Introduction
   - Service DNS et organisation
   - Rappels techniques
   - Attaques
2. DNSSEC
3. Fin

Domain Name System : présentation

Service

- Nommage de machine dans les internets
  - Nom de machine $\rightarrow$ adresse IP
  - Gère aussi d'autres informations (échangeurs de mails, . . .)

[3] Mathieu Herrb

Domain Name System : présentation

Service

- Nommage de machine dans les internets
  - Nom de machine $\rightarrow$ adresse IP
  - Gère aussi d'autres informations (échangeurs de mails, . . .)
- Espace de nommage hiérarchique
  - Racine : .
  - Domaine de premier niveau : .fr
  - Domaine de second niveau : enseeiht.fr
  - Sous-domaine de premier niveau : perso.enseeiht.fr

[3] Mathieu Herrb

Domain Name System : présentation

Service

- Nommage de machine dans les internets
  - Nom de machine $\rightarrow$ adresse IP
  - Gère aussi d'autres informations (échangeurs de mails, . . .)
- Espace de nommage hiérarchique
  - Racine : .
  - Domaine de premier niveau : .fr
  - Domaine de second niveau : enseeiht.fr
  - Sous-domaine de premier niveau : perso.enseeiht.fr

$\Rightarrow$ Infrastructure quasi-indispensable

[3] Mathieu Herrb

Domain Name System : présentation

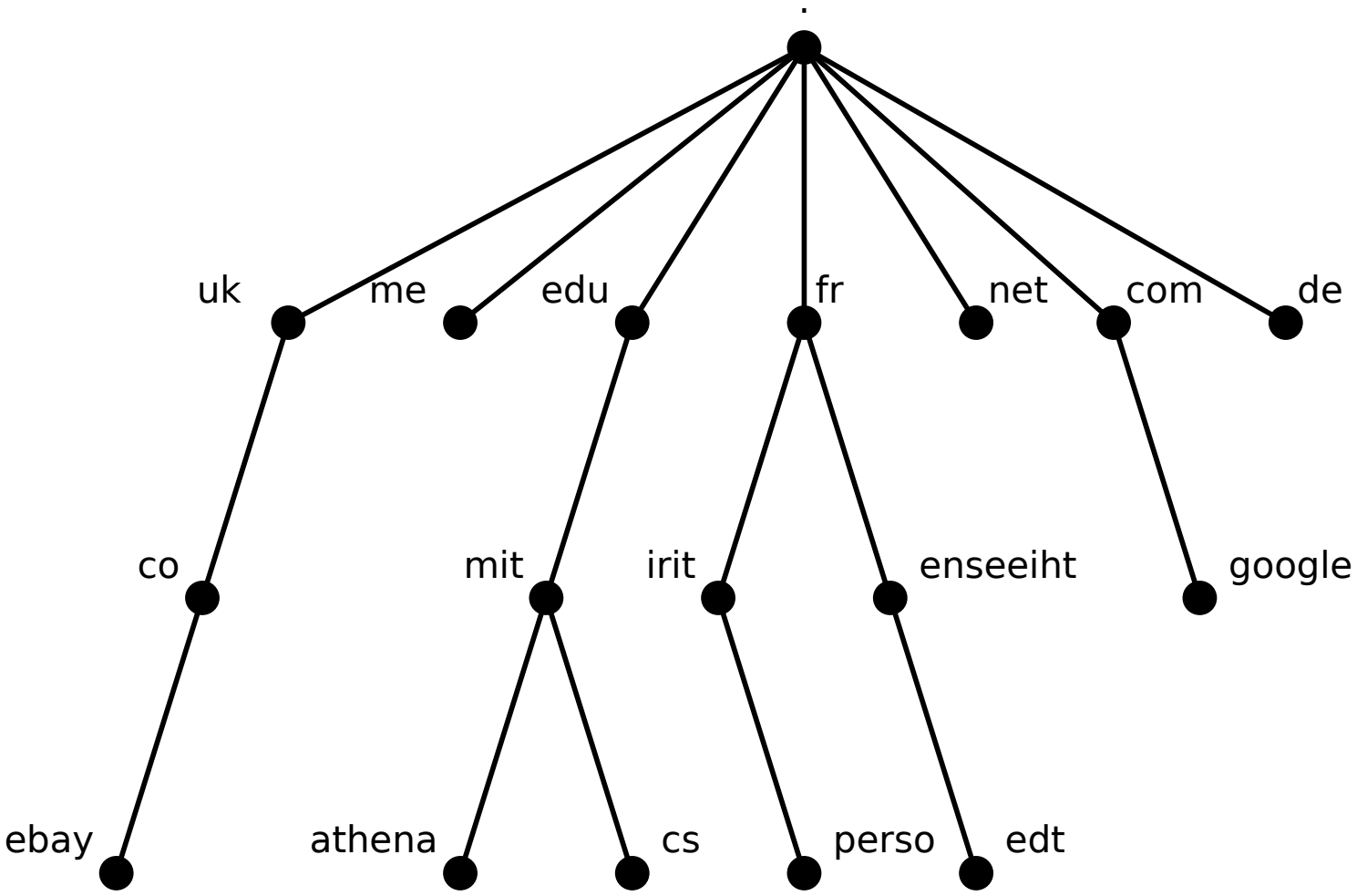
Service

- Nommage de machine dans les internets
  - Nom de machine $\rightarrow$ adresse IP
  - Gère aussi d'autres informations (échangeurs de mails, . . .)
- Espace de nommage hiérarchique
  - Racine : .
  - Domaine de premier niveau : .fr
  - Domaine de second niveau : enseeiht.fr
  - Sous-domaine de premier niveau : perso.enseeiht.fr

$\Rightarrow$ Infrastructure quasi-indispensable. Internet sans DNS ?

[3] Mathieu Herrb

DNS : nommage hiérarchique



DNS : nommage hiérarchique

Domaines de premier niveau

- Historiquement il existe un nombre limité de domaines de premier niveau
- Ils sont classé en différents catégories : national (.fr), générique (.net, .edu), réservé (.example, .local), . . .
- Nouveau domaines de premier niveau (2012) : .pizza, .microsoft, .cafe

`http://data.iana.org/TLD/tlds-alpha-by-domain.txt`

DNS : nommage hiérarchique

Domaines de premier niveau

- Historiquement il existe un nombre limité de domaines de premier niveau
- Ils sont classé en différents catégories : national (.fr), générique (.net, .edu), réservé (.example, .local), ...
- Nouveau domaines de premier niveau (2012) : .pizza, .microsoft, .cafe

`http://data.iana.org/TLD/tlds-alpha-by-domain.txt`

Domaines et sous domaines

- Définis de manière arbitraire par les utilisateurs
- Contraintes techniques sur le format de la chaîne
- Enregistrés à la demande des utilisateurs

Attribution et gestion administrative

- Domaines de premier niveau : *Internet Assigned Numbers Authority* (IANA)
  - Définis les domaines de premier niveau
  - Délègue leur gestion administrative
  - Gestion technique de la racine
- Domaines : registres ou *Network Information Center* (NIC).
  - Maintient la base de données de domaines
  - Délègue parfois l'enregistrement administratif
  - .fr : AFNIC, .org : Public Interest Registry, .com : Verisign
  - Non responsable de la gestion technique
- Bureaux d'enregistrement désigné les NIC
  - *registrars*
  - Interface administrative avec les NIC
  - Gestion techniques des domaines enregistrés
  - Service commercial

Détails du standard DNS

Détails du standard DNS

1. Protocoles de résolution de noms de domaine
   - Protocole de communication applicatif entre entités DNS
   - Interrogation de la base de données répartie
   - Support de la redondance
2. Hébergement de la base de donnée répartie
   - Langage de description de zones DNS
   - Mise en œuvre technique de la base de données hiérarchique répartie
3. Clients du service DNS

Détails du standard DNS

Détails du standard DNS

1. Protocoles de résolution de noms de domaine
   - Protocole de communication applicatif entre entités DNS
   - Interrogation de la base de données répartie
   - Support de la redondance
2. Hébergement de la base de donnée répartie
   - Langage de description de zones DNS
   - Mise en œuvre technique de la base de données hiérarchique répartie
3. Clients du service DNS

Types de serveurs DNS

1. Hébergemene
   - bind9, NSD
2. Résolution de noms (récursifs)
   - bind9, unbound, knot

Détails du standard DNS

Détails du standard DNS

1. Protocoles de résolution de noms de domaine
   - Protocole de communication applicatif entre entités DNS
   - Interrogation de la base de données répartie
   - Support de la redondance
2. Hébergement de la base de donnée répartie
   - Langage de description de zones DNS
   - Mise en œuvre technique de la base de données hiérarchique répartie
3. Clients du service DNS

Types de serveurs DNS

1. Hébergemene
   - bind9, NSD
2. Résolution de noms (récursifs)
   - bind9, unbound, knot
3. Types de clients (*stub resolvers*)
   - `dig`
   - `host`
   - `nslookup`
   - Bib. C `gehostbyname()`

Langage de description des zones

Caractéristiques

- Langage textuel simple (rationnel)

nom	TTL	classe	type	données

- Types d'enregistrement
  - **A** : adresse IPv4
  - **AAAA** : adresse IPv6
  - **NS** : délégation de zone
  - **MX** : adresse d'un échangeur de mail pour cette zone
  - **SOA** : adresse du DNS primare de la zone et mail admin.

Exemple de zone DNS bêta

Il était une fois une base de données centralisée.

```
; nom                    TTL     classe  type    données
enseeiht.fr.             86400   IN      MX      10 n7smtp.enseeiht.fr.
enseeiht.fr.             86400   IN      A       193.48.203.34
albator.enseeiht.fr.     86400   IN      A
zigoto.enseeiht.fr.      86400   IN      A       147.127.128.68
bd.enseeiht.fr.          86400   IN      CNAME   gailuron.enseeiht.fr.
gailuron.enseeiht.fr.    86400   IN      A       147.127.128.43
gala.enseeiht.fr.        86400   IN      CNAME   www.bde.enseeiht.fr.
; et tous les autres domaines des internets...
```

Exemple de zone DNS bêta

Il était une fois une base de données centralisée.

```
; nom                    TTL     classe  type    données
enseeiht.fr.             86400   IN      MX      10 n7smtp.enseeiht.fr.
enseeiht.fr.             86400   IN      A       193.48.203.34
albator.enseeiht.fr.     86400   IN      A
zigoto.enseeiht.fr.      86400   IN      A       147.127.128.68
bd.enseeiht.fr.          86400   IN      CNAME   gailuron.enseeiht.fr.
gailuron.enseeiht.fr.    86400   IN      A       147.127.128.43
gala.enseeiht.fr.        86400   IN      CNAME   www.bde.enseeiht.fr.
; et tous les autres domaines des internets...
```

- Réaliste ?

Exemple de zone DNS bêta

Il était une fois une base de données centralisée.

```
; nom                    TTL     classe  type    données
enseeiht.fr.             86400   IN      MX      10 n7smtp.enseeiht.fr.
enseeiht.fr.             86400   IN      A       193.48.203.34
albator.enseeiht.fr.     86400   IN      A
zigoto.enseeiht.fr.      86400   IN      A       147.127.128.68
bd.enseeiht.fr.          86400   IN      CNAME   gailuron.enseeiht.fr.
gailuron.enseeiht.fr.    86400   IN      A       147.127.128.43
gala.enseeiht.fr.        86400   IN      CNAME   www.bde.enseeiht.fr.
; et tous les autres domaines des internets...
```

- Réaliste ?

⇒ Délégation de zones !

Exemple de zone DNS bêta

Il était une fois une base de données centralisée.

```
; nom                    TTL     classe  type    données
enseeiht.fr.             86400   IN      MX      10 n7smtp.enseeiht.fr.
enseeiht.fr.             86400   IN      A       193.48.203.34
albator.enseeiht.fr.     86400   IN      A
zigoto.enseeiht.fr.      86400   IN      A       147.127.128.68
bd.enseeiht.fr.          86400   IN      CNAME   gailuron.enseeiht.fr.
gailuron.enseeiht.fr.    86400   IN      A       147.127.128.43
gala.enseeiht.fr.        86400   IN      CNAME   www.bde.enseeiht.fr.
; et tous les autres domaines des internets...
```

- Réaliste ?

⇒ Délégation de zones !

⇒ Bénéfice organisationnel

⇒ Bénéfice technique

Exemple de zone DNS bêta

Il était une fois une base de données centralisée.

```
; nom                    TTL     classe  type    données
enseeiht.fr.             86400   IN      MX      10 n7smtp.enseeiht.fr.
enseeiht.fr.             86400   IN      A       193.48.203.34
albator.enseeiht.fr.     86400   IN      A
zigoto.enseeiht.fr.      86400   IN      A       147.127.128.68
bd.enseeiht.fr.          86400   IN      CNAME   gailuron.enseeiht.fr.
gailuron.enseeiht.fr.    86400   IN      A       147.127.128.43
gala.enseeiht.fr.        86400   IN      CNAME   www.bde.enseeiht.fr.
; et tous les autres domaines des internets...
```

- Réaliste ?

⇒ Délégation de zones !

⇒ Bénéfice organisationnel

⇒ Bénéfice technique

- Serveurs faisant autorité
- Types d'enregistrement impliqués
  - *Start Of Authority* (**SOA**)
    Indispensable pour les transferts (serial number)
  - *Name Server* (**NS**)
    Zone parente délègue en pointant un serveur de zone

Délégation de zone : base de données répartie 1/3

Quelque chose de plus réaliste

```
; nom   TTL      classe   type   données
.       86400    IN       SOA    a.root-servers.net. nstld.verisign-grs.com.
.       518400   IN       NS     a.root-servers.net.
.       518400   IN       NS     b.root-servers.net.
.       518400   IN       NS     c.root-servers.net.
;...
fr.     172800   IN       NS     d.ext.nic.fr.
fr.     172800   IN       NS     d.nic.fr.
fr.     172800   IN       NS     e.ext.nic.fr.
fr.     172800   IN       NS     f.ext.nic.fr.
fr.     172800   IN       NS     g.ext.nic.fr.
;...
```

Hébergé sur un des serveurs racines.

Délégation de zone : base de données répartie 2/3

Quelque chose de plus réaliste

```
; nom          TTL     classe  type  données
fr.            5400    IN      SOA   nsmaster.nic.fr. hostmaster.nic.fr.
;...
enseeiht.fr.   172800  IN      NS    sivuca.leei.enseeiht.fr.
enseeiht.fr.   172800  IN      NS    ns1.enseeiht.fr.
enseeiht.fr.   172800  IN      NS    ns2.nic.fr.
;...
irit.fr.       172800  IN      NS    ns1.irit.fr.
irit.fr.       172800  IN      NS    dnsadv.univ-toulouse.fr.
```

Hébergé sur un des serveurs DNS du domaine de premier niveau fr.

Délégation de zone : base de données répartie 3/3

Quelque chose de plus réaliste

; nom	TTL	classe	type	données
enseeiht.fr.	86400	IN	SOA	enseeiht.fr. hm.enseeiht.fr.
;...				
enseeiht.fr.	86400	IN	MX	10 n7smtp.enseeiht.fr.
enseeiht.fr.	86400	IN	A	193.48.203.34
albator.enseeiht.fr.	86400	IN	A	
zigoto.enseeiht.fr.	86400	IN	A	147.127.128.68
bd.enseeiht.fr.	86400	IN	CNAME	gailuron.enseeiht.fr.
gailuron.enseeiht.fr.	86400	IN	A	147.127.128.43
gala.enseeiht.fr.	86400	IN	CNAME	www.bde.enseeiht.fr.

Hébergé sur un des serveurs DNS de l'enseeiht responsables du domaine enseeiht.fr.

Résolution de la racine ?

Pour pouvoir résoudre
`albator.enseeiht.fr.`,

Résolution de la racine ?

Pour pouvoir résoudre `albator.enseeiht.fr.`, il me faut résoudre l'adresse du serveur DNS de la zone `enseeiht.fr.`

Résolution de la racine ?

Pour pouvoir résoudre `albator.enseeiht.fr.`, il me faut résoudre l'adresse du serveur DNS de la zone `enseeiht.fr.` $\rightarrow$ `fr.`

Résolution de la racine ?

Pour pouvoir résoudre `albator.enseeiht.fr.`, il me faut résoudre l'adresse du serveur DNS de la zone `enseeiht.fr.` → `fr.` → `.`

Résolution de la racine ?

Pour pouvoir résoudre `albator.enseeiht.fr.`, il me faut résoudre l'adresse du serveur DNS de la zone `enseeiht.fr.` → `fr.` → `.` → ?

Résolution de la racine ?

Pour pouvoir résoudre `albator.enseeiht.fr.`, il me faut résoudre l'adresse du serveur DNS de la zone `enseeiht.fr.` → `fr.` → `.` →?

- À qui dois-je demander pour `.` ?

Résolution de la racine ?

Pour pouvoir résoudre `albator.enseeiht.fr.`, il me faut résoudre l'adresse du serveur DNS de la zone `enseeiht.fr.` → `fr.` → `.` → ?

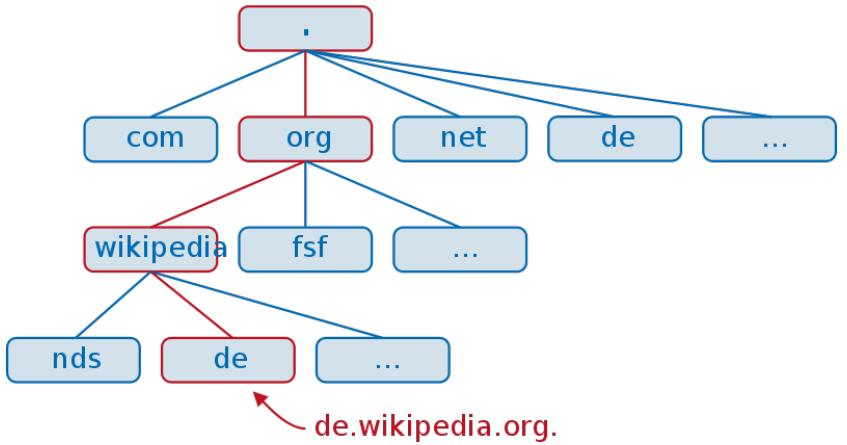
- À qui dois-je demander pour `.` ?

⇒ Résolution statique implantée dans les serveur DNS de résolution.

Messages à retenir

Une structure hiérarchique

- Une racine unique "."
  - Accessible sur de nombreux serveurs répliqués `[a-m].root-servers.net`
  - Chacun répliqué (max, x63 !) par anycast
  - Contient le *root zone file* (200KB) : NS pour les top level domains (.com, .fr, etc.)
- Les Domain Name Registrars actualisent NS des top level domains
- Le reste de la hierarchie est fait par les possesseurs des domaines



Clients (*resolvers*)

Ne connaissent a priori rien sur la hiérarchie des serveurs DNS
Utilisent un serveur DNS *récursif* fourni par configuration statique ou DHCP

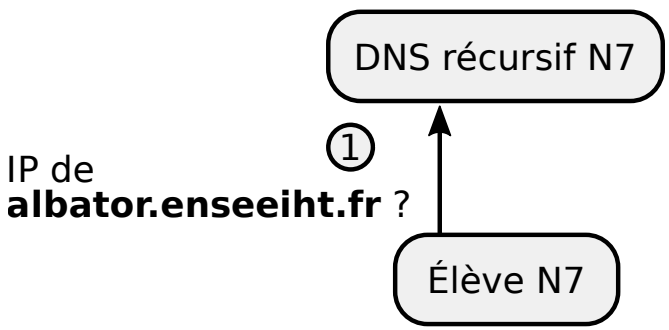
Résolution d'un nom de domaine : démo

- `$ dig enseeiht.fr`
- `$ dig enseeiht.fr NS`
- `$ dig fr`
- `$ dig com`
- `$ dig fr SOA`
- `$ dig albator.enseeiht.fr +trace`
- `$ dig enseeiht.fr +trace`
- Analyse de trace réseau de résolution...

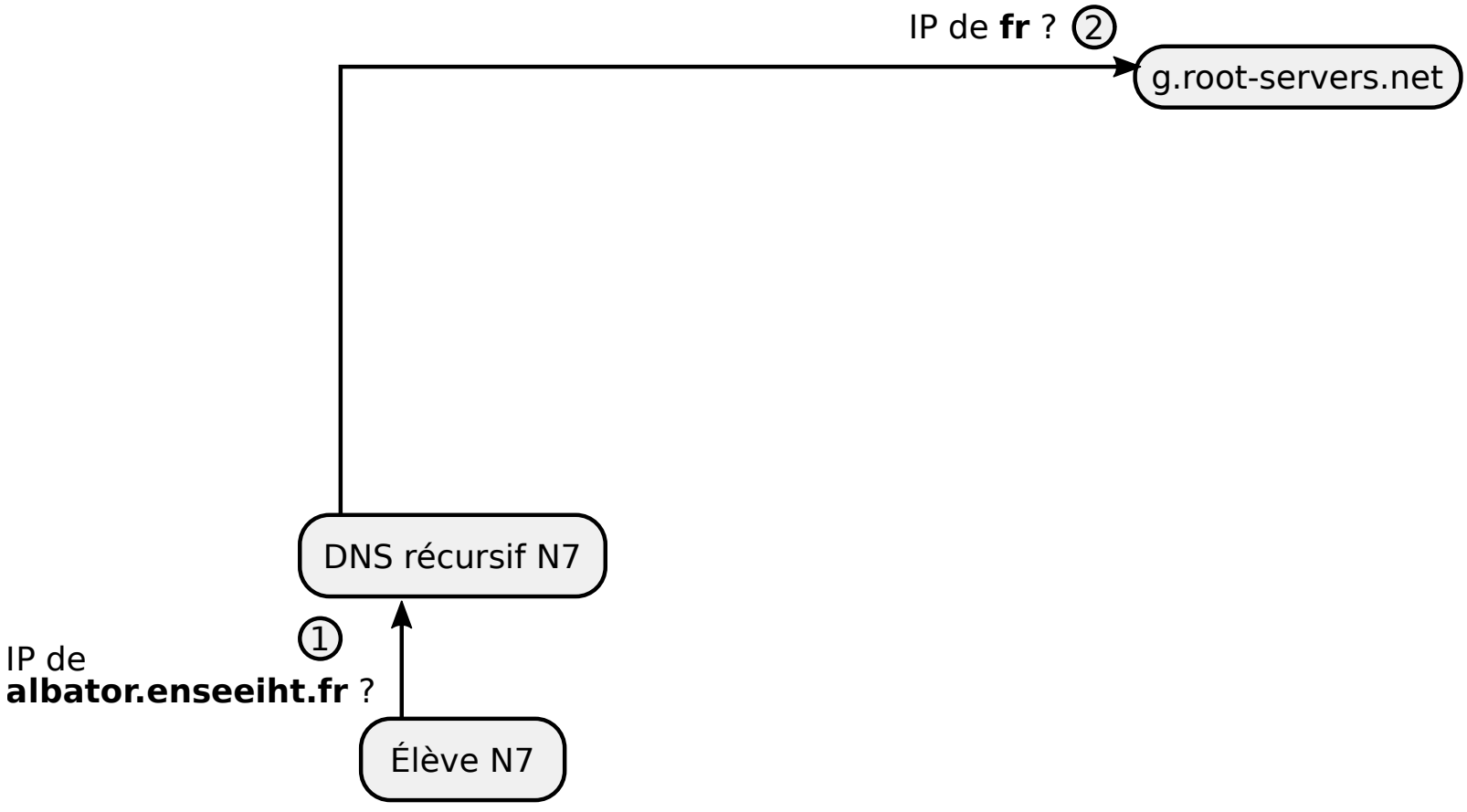
Résolution d'un nom de domaine : concept

Élève N7

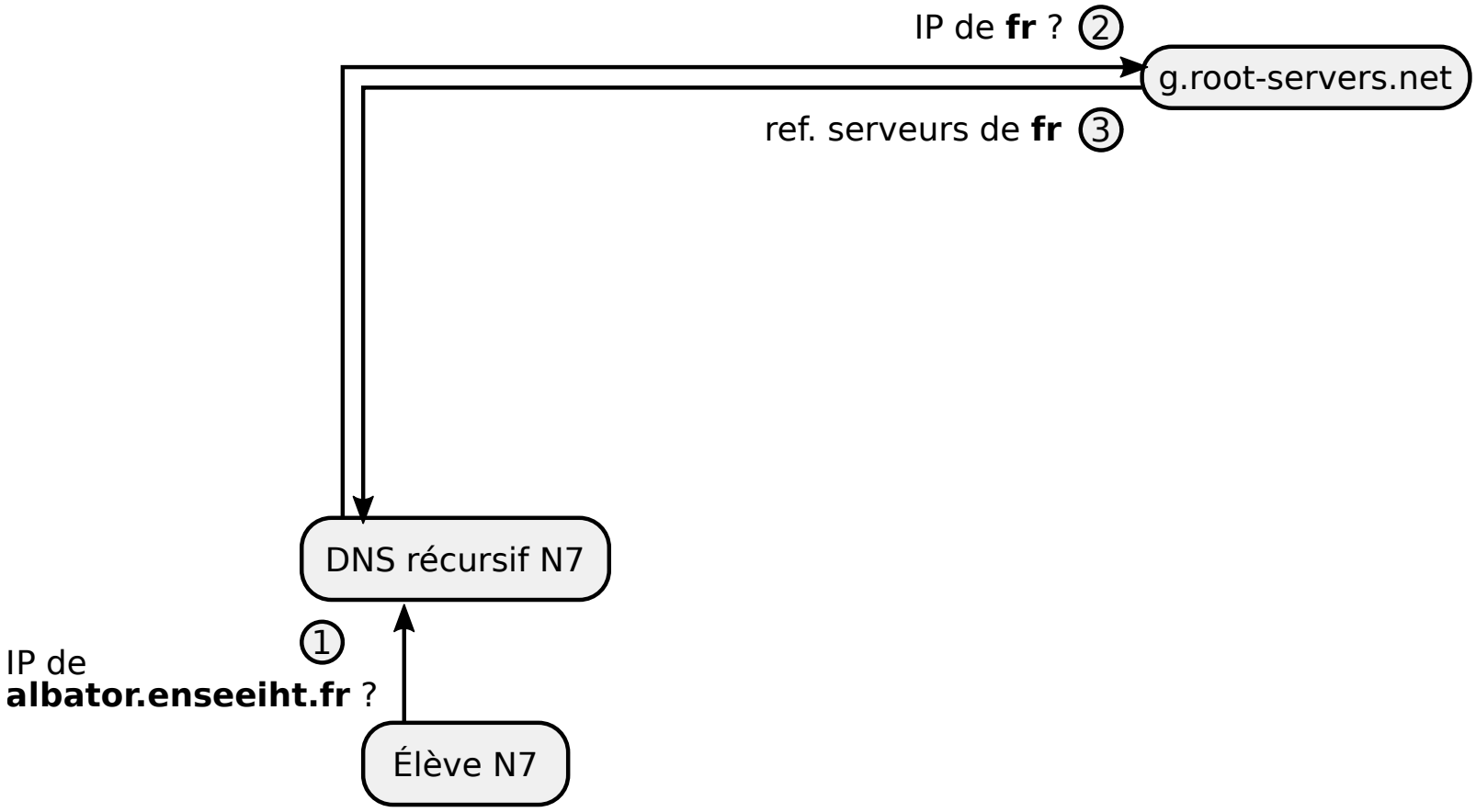
Résolution d'un nom de domaine : concept



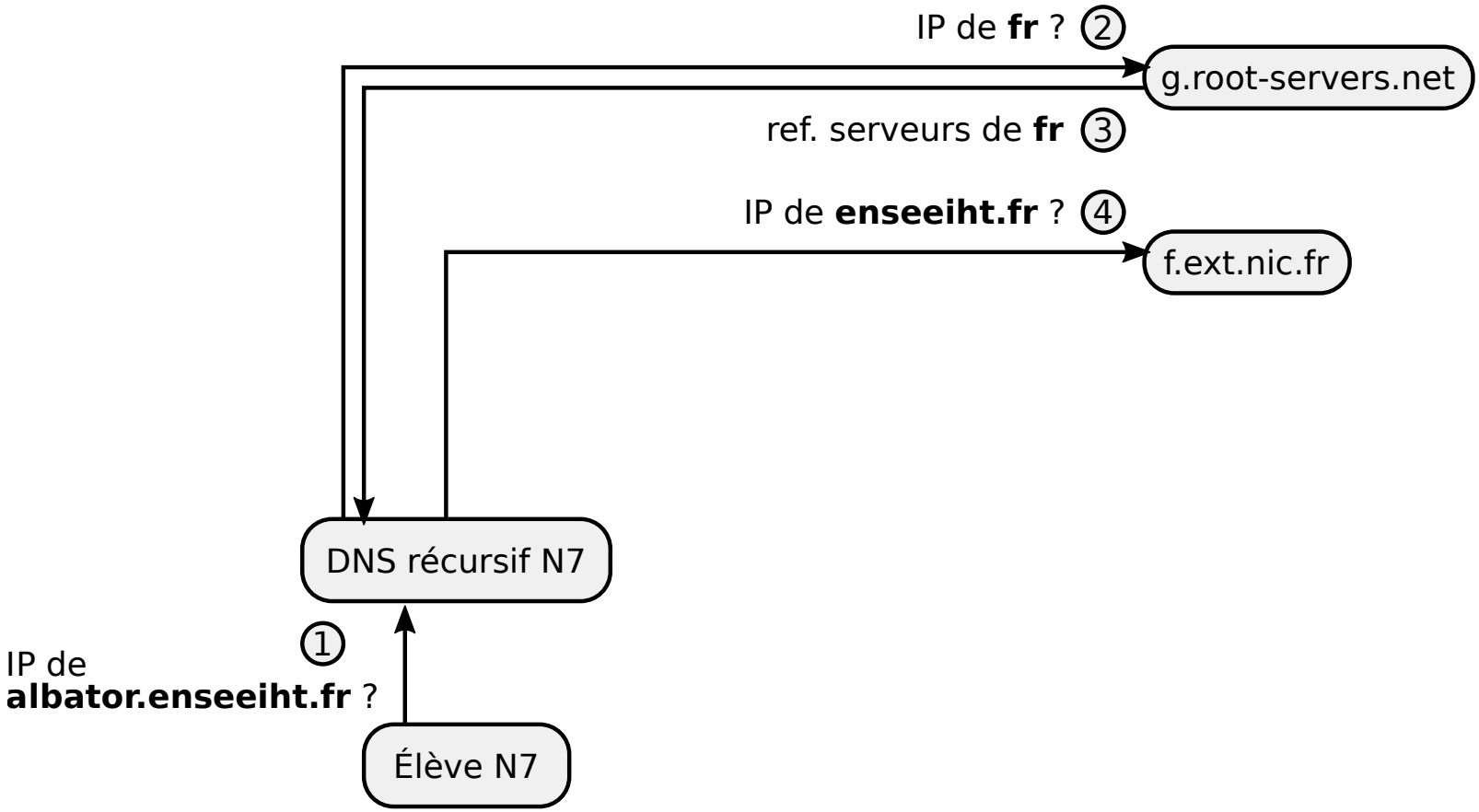
Résolution d'un nom de domaine : concept



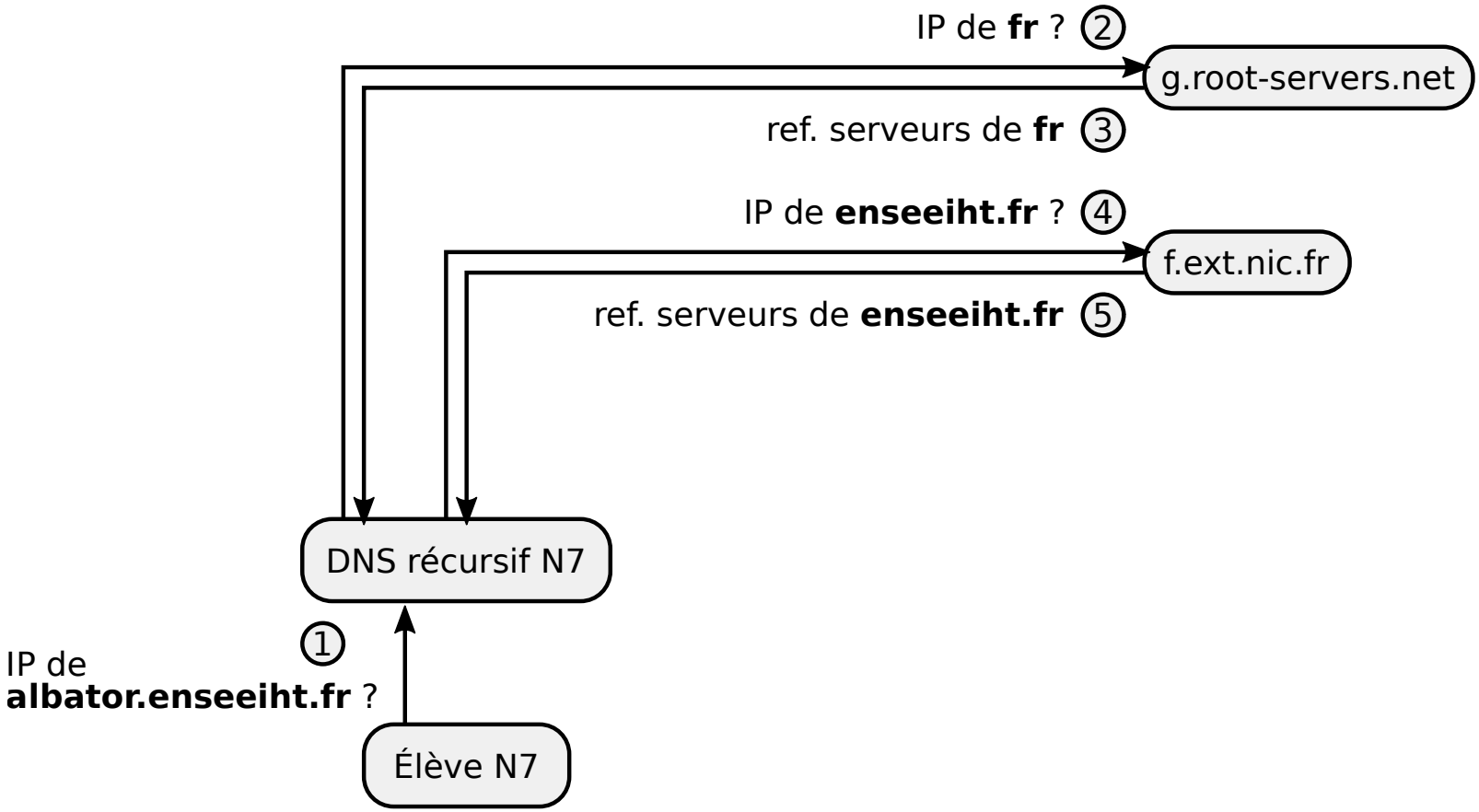
Résolution d'un nom de domaine : concept



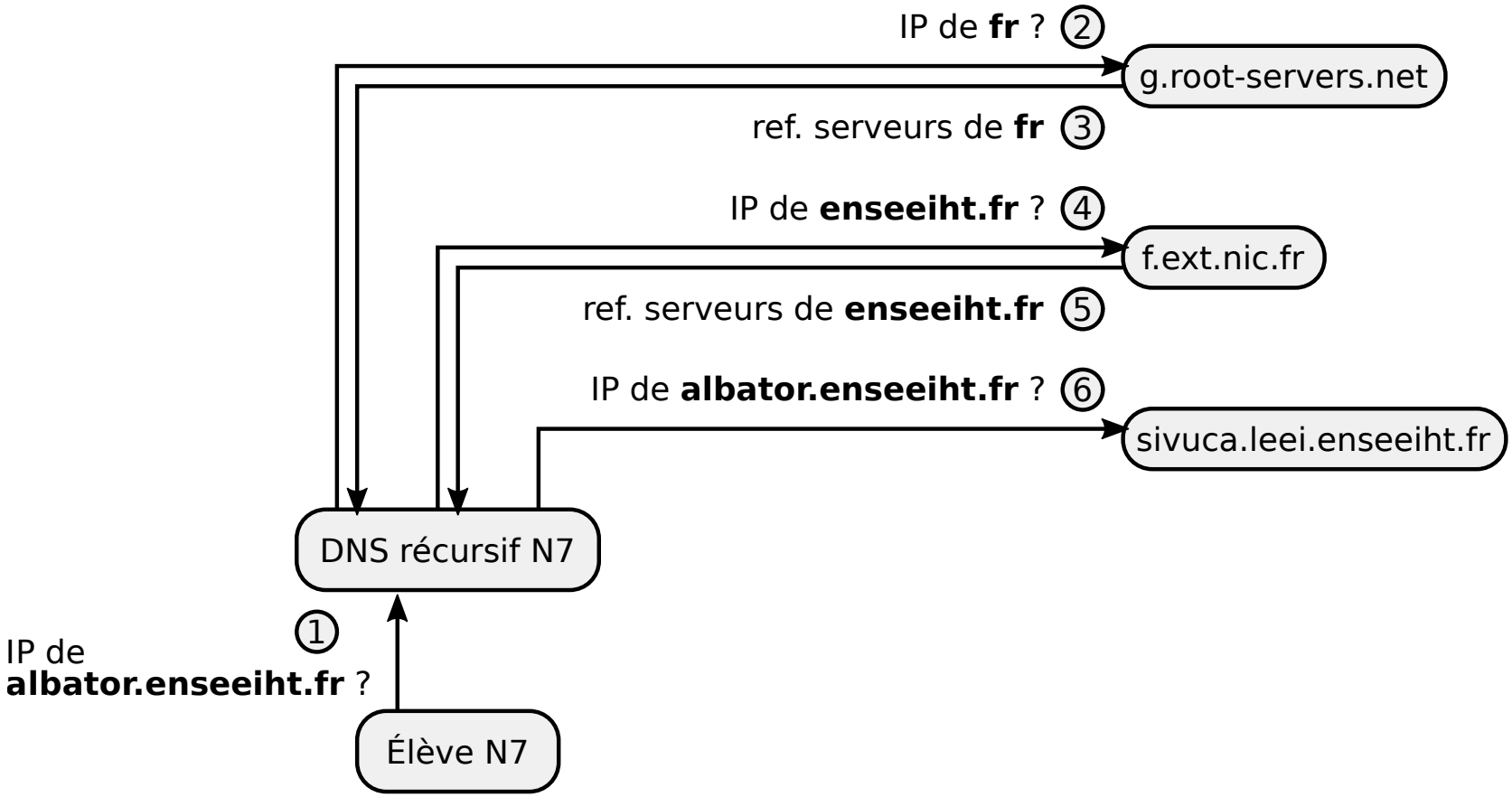
Résolution d'un nom de domaine : concept



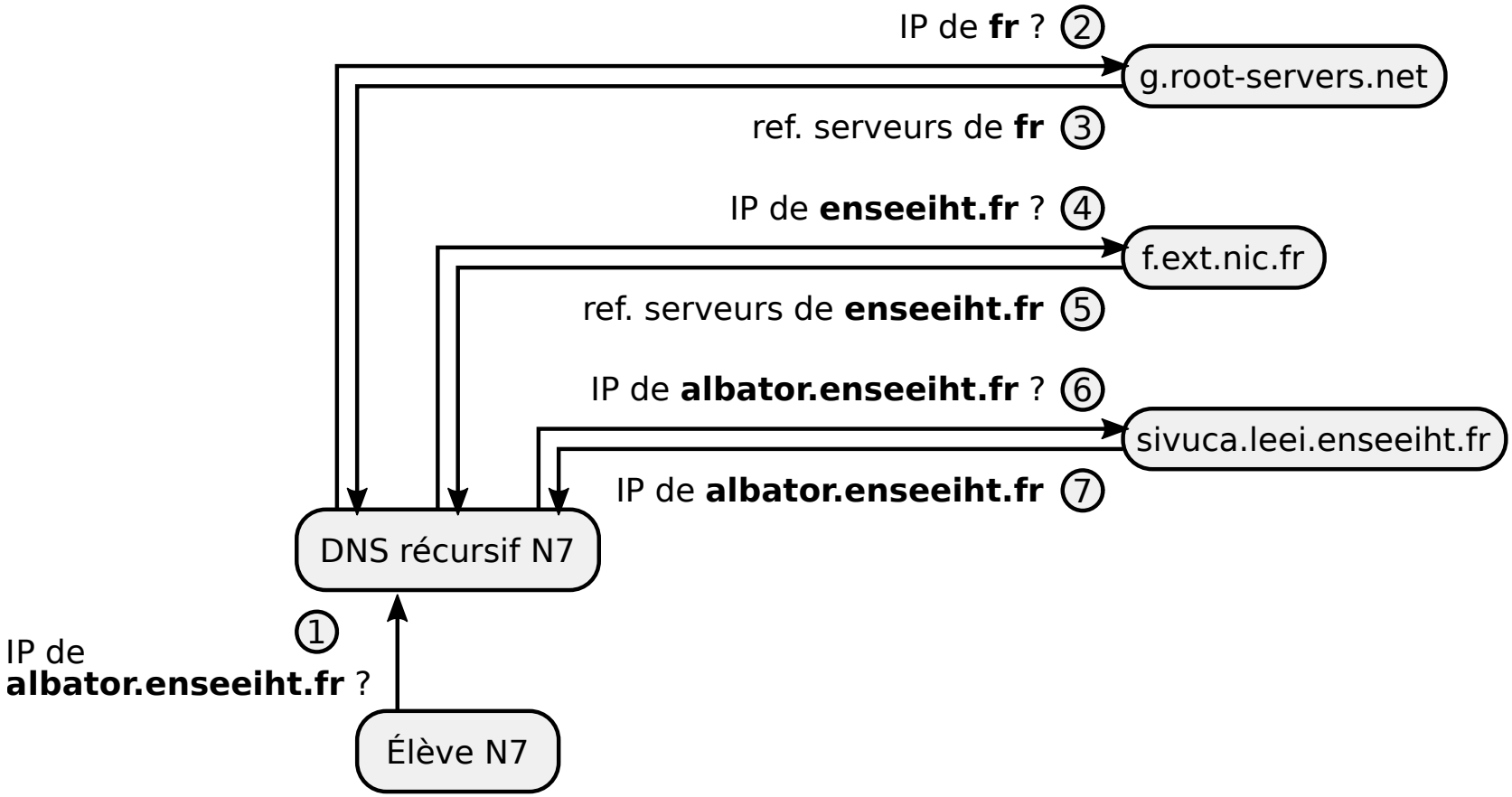
Résolution d'un nom de domaine : concept



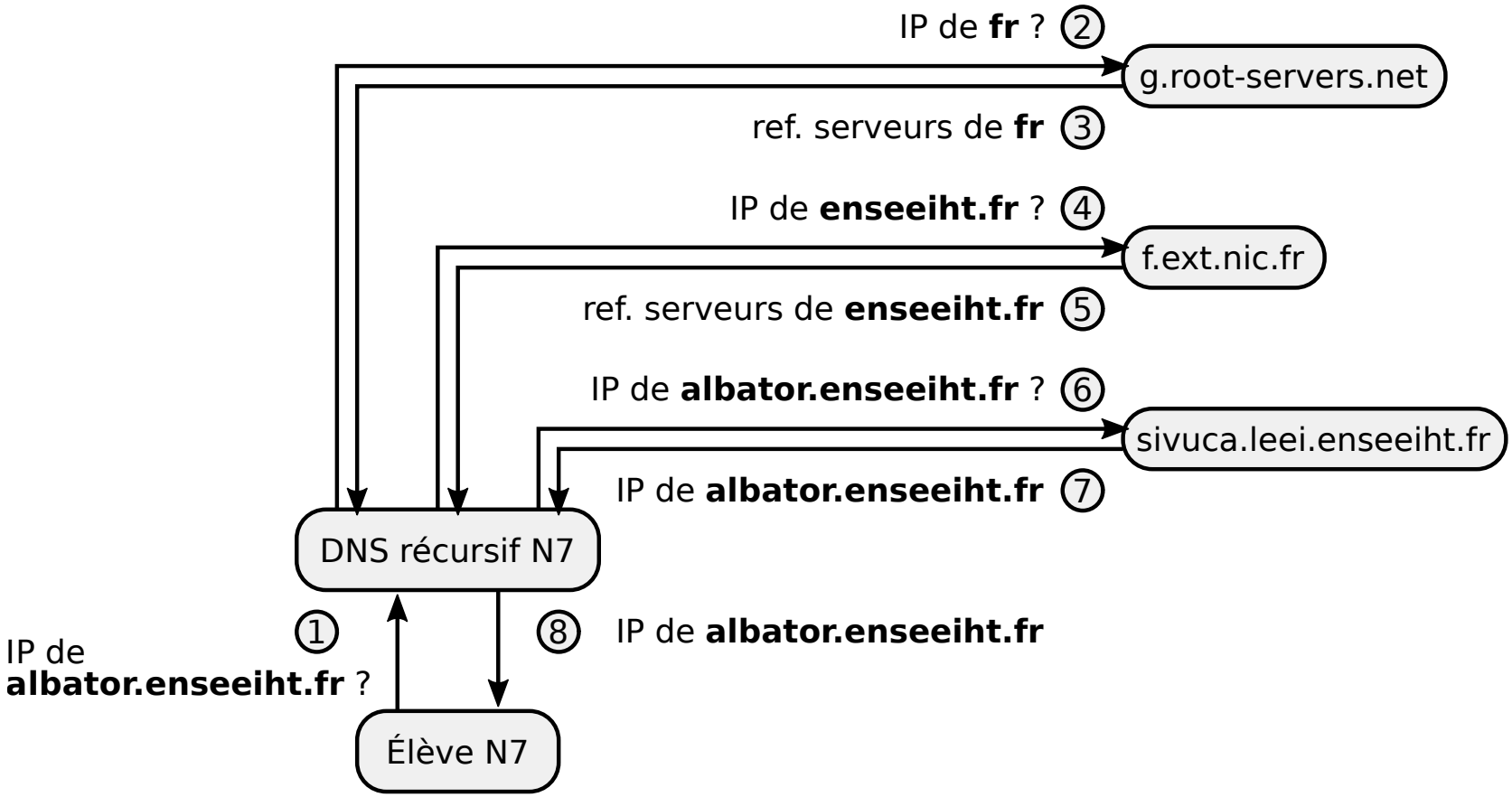
Résolution d'un nom de domaine : concept



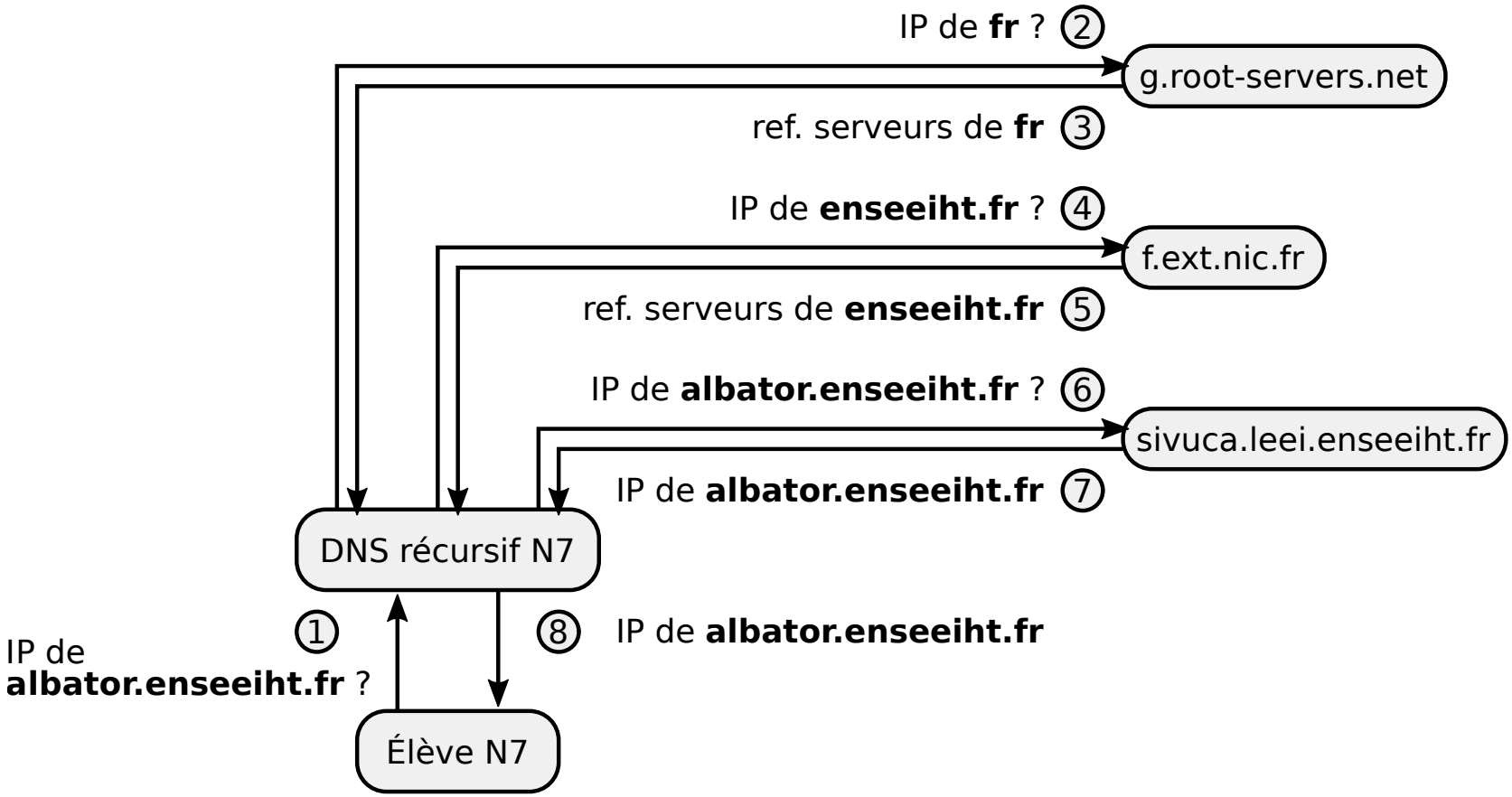
Résolution d'un nom de domaine : concept



Résolution d'un nom de domaine : concept

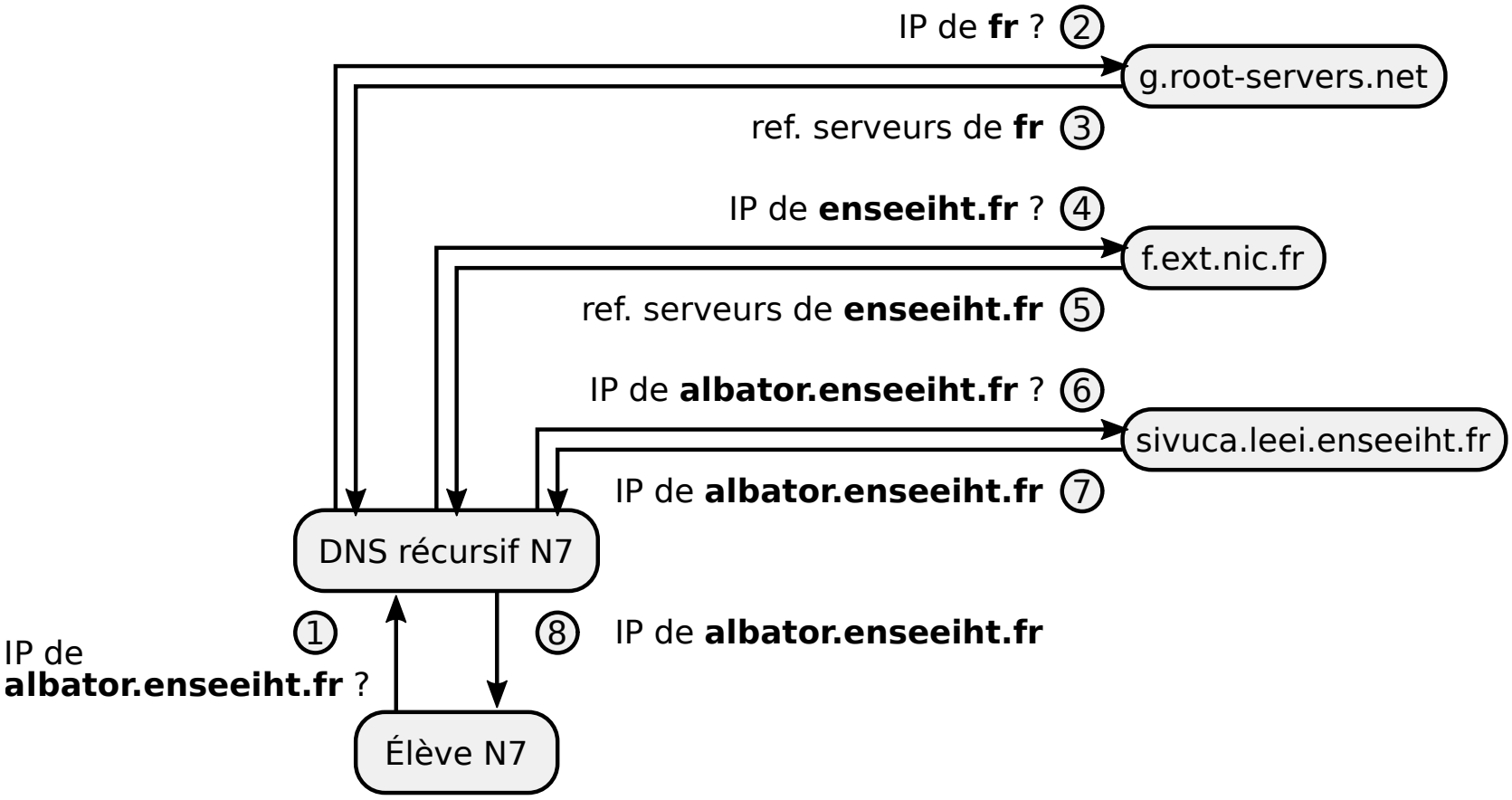


Résolution d'un nom de domaine : concept



- Pourquoi on ne résout pas les noms des DNS ? ? ? (enregistrement A)

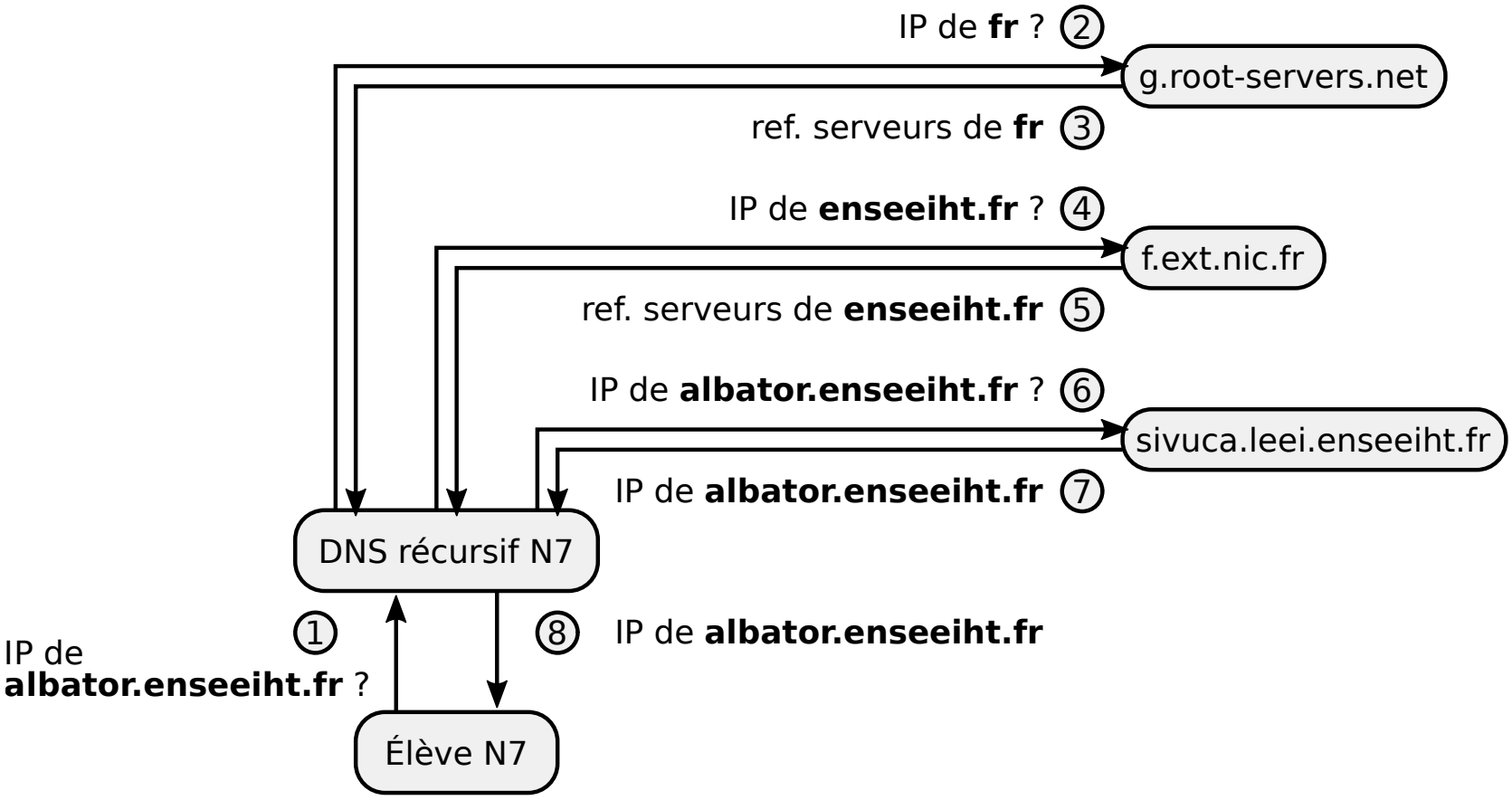
Résolution d'un nom de domaine : concept



- Pourquoi on ne résout pas les noms des DNS ? ? ? (enregistrement A)

$\Rightarrow$ utilisation de colles DNS

Résolution d'un nom de domaine : concept



- Pourquoi on ne résout pas les noms des DNS ? ? ? (enregistrement A)

$\Rightarrow$ utilisation de *glue* DNS

Les *glue records*

- `$ dig albator.enseeiht.fr A +trace +additional +all`
- Techniquement nécessaire pour la résolution
- Réponse non autoritaire
- Doit être dans l'absolu confirmée par un enregistrement **A** ou **AAAA** dans la ayant autorité pour cette déclaration

Les *glue records*

- `$ dig albator.enseeiht.fr A +trace +additional +all`
- Techniquement nécessaire pour la résolution
- Réponse non autoritaire
- Doit être dans l'absolu confirmée par un enregistrement **A** ou **AAAA** dans la ayant autorité pour cette déclaration

Le retour !

Utilisation d'un cache

Cache des réponses (positives)

Mise en cache des NS, glues, et IPs réponses
Permet de

- Accélérer le temps de réponse significativement
- Décharger les serveurs DNS "hauts" dans la hiérarchie

Pourcentage variable mais très significatif ($>$ 75%) est dans le cache

Cache négatif

Requêtes pouvant monter très haut (e.g. `www.google.fr`)
$\rightarrow$ Mises en cache

Durée de vie

Les données ont un TTL défini par le serveur autoritaire (le propriétaire)
TTL envoyé avec chaque enregistrement

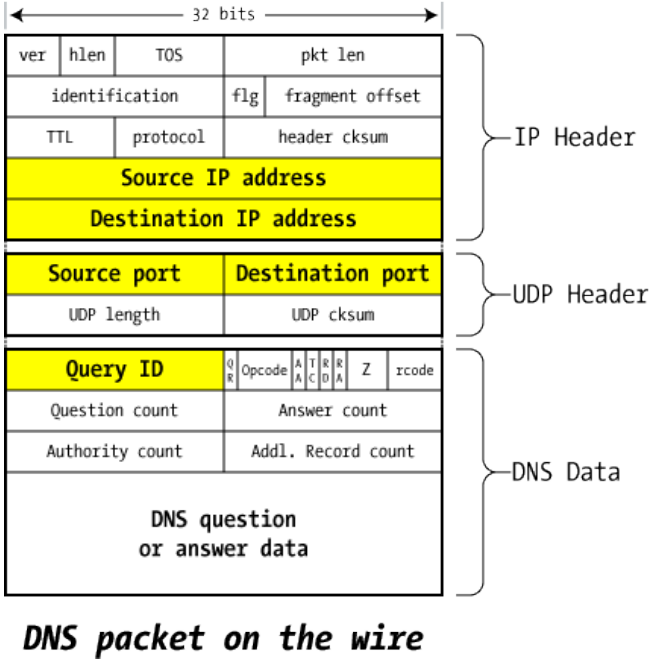
Utilisation d'un cache

- Exemple : serveur récursif unbound
- `$ dig albator.enseeiht.fr`
- `$ dig albatros.enseeiht.fr`

Résolution d'un nom de domaine : protocole

Serveurs récursifs

Message générique



DNS packet on the wire

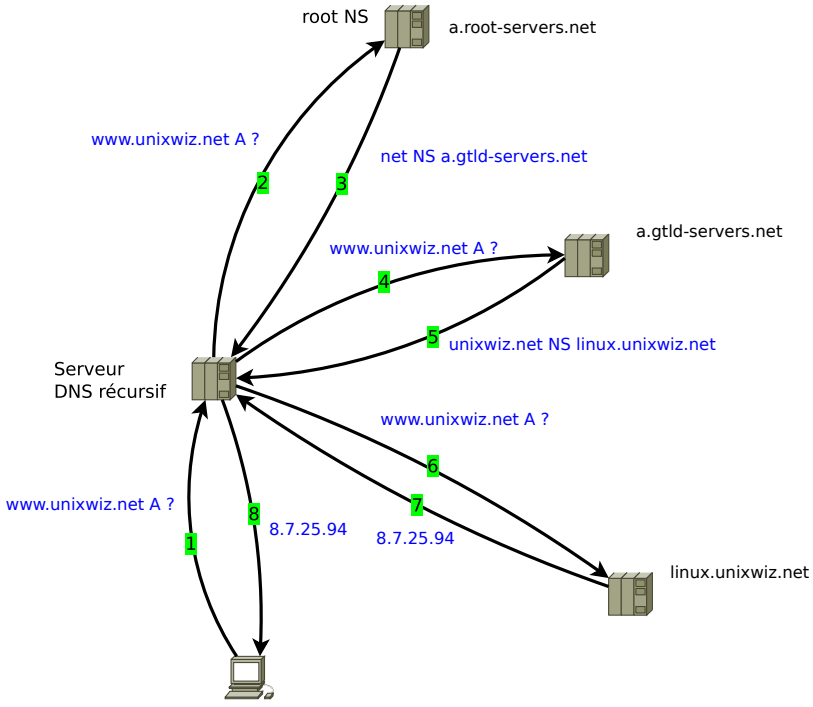


Image tirée de http://fr.wikipedia.org/wiki/Domain_Name_System puis modifiée

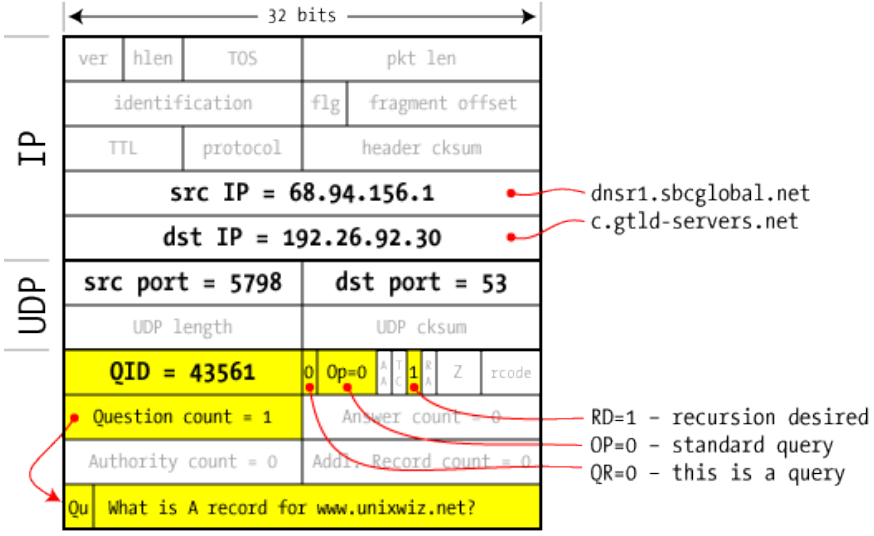
Résolution d'un nom de domaine : protocole

Serveurs récursifs

Message 1 question initiale

Messages 2 et 3 rarement nécessaires

Message 4



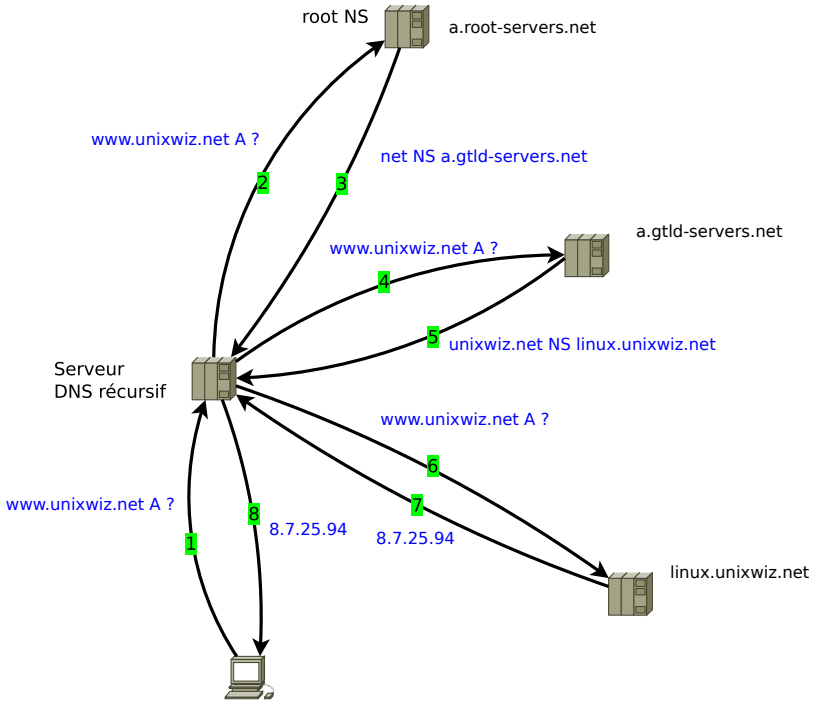
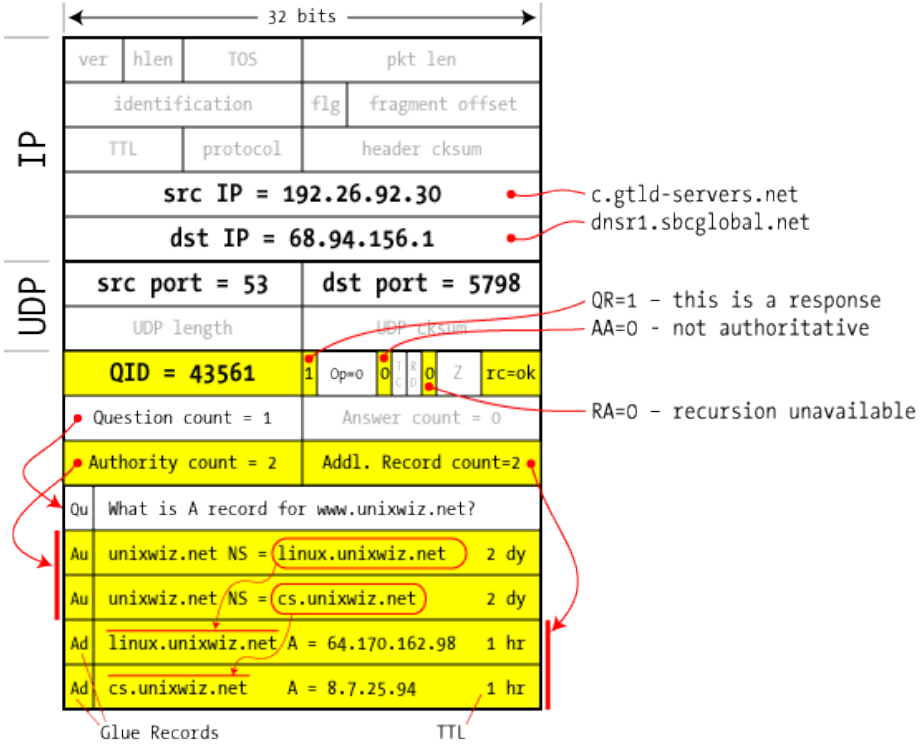


Image tirée de http://fr.wikipedia.org/wiki/Domain_Name_System puis modifiée

Résolution d'un nom de domaine : protocole

Serveurs récursifs

Message 5



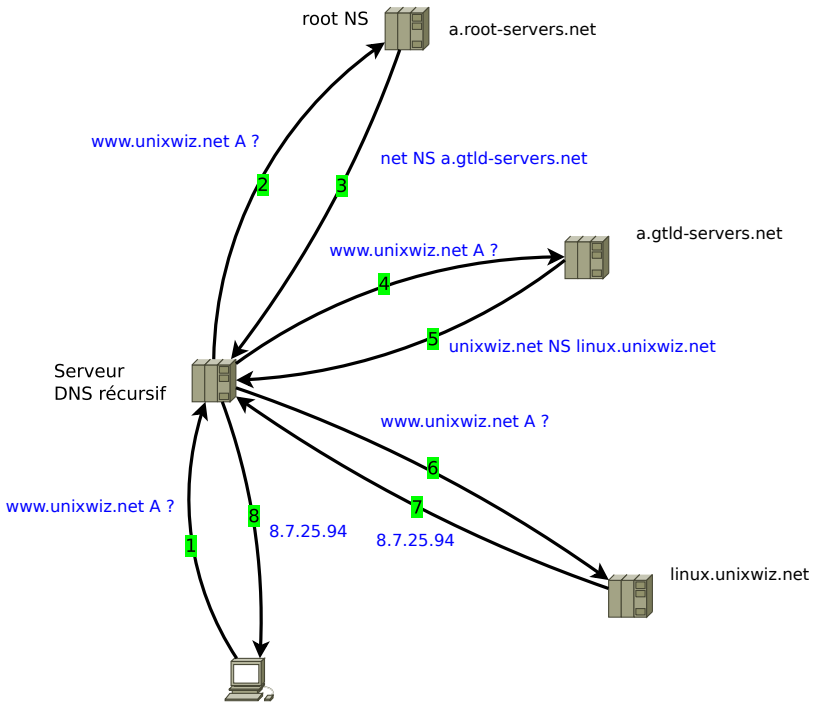
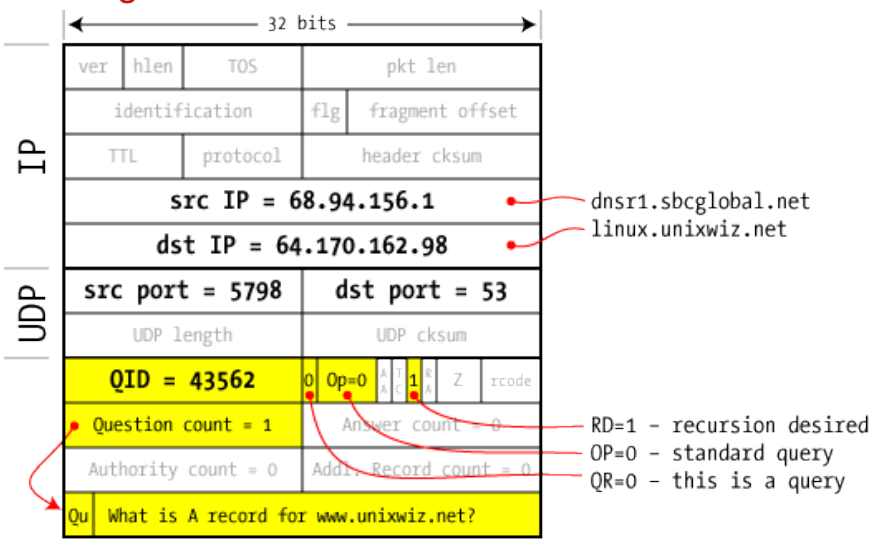


Image tirée de http://fr.wikipedia.org/wiki/Domain_Name_System puis modifiée

Résolution d'un nom de domaine : protocole

Serveurs récursifs

Message 6



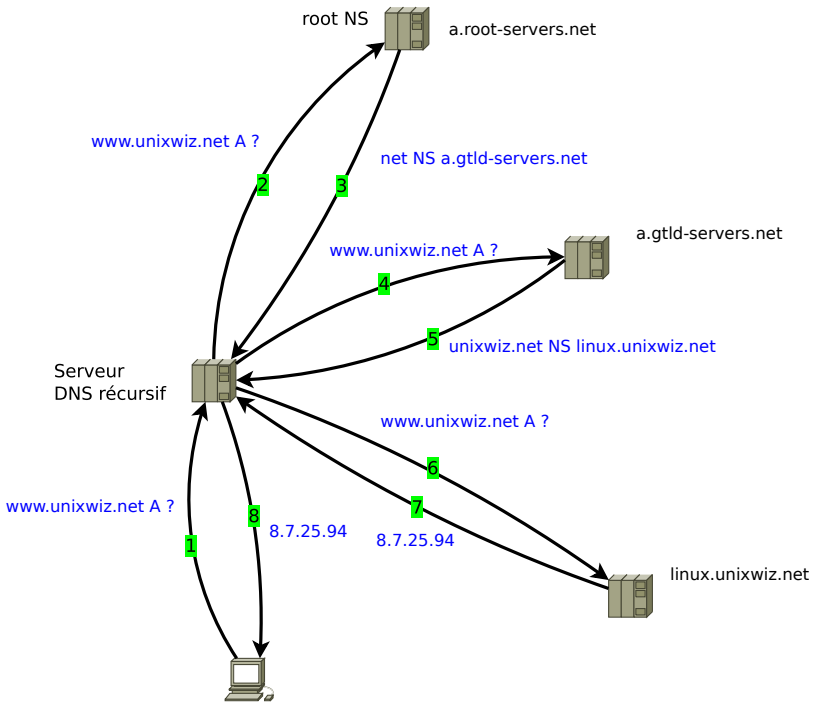
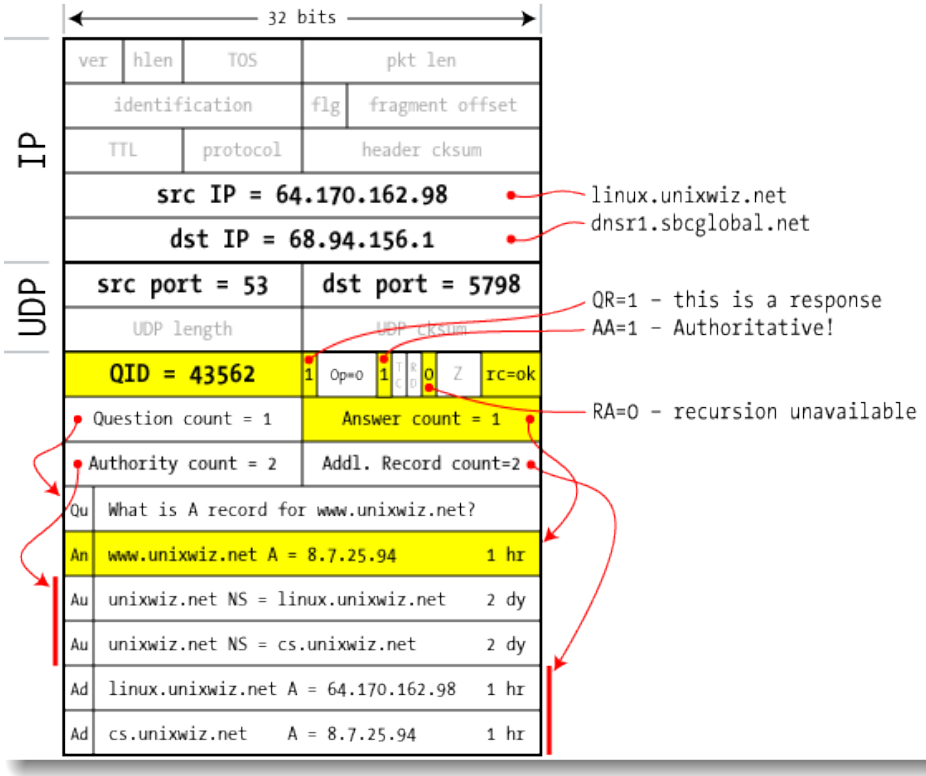


Image tirée de http://fr.wikipedia.org/wiki/Domain_Name_System puis modifiée

Résolution d'un nom de domaine : protocole

Serveurs récursifs

Message 7



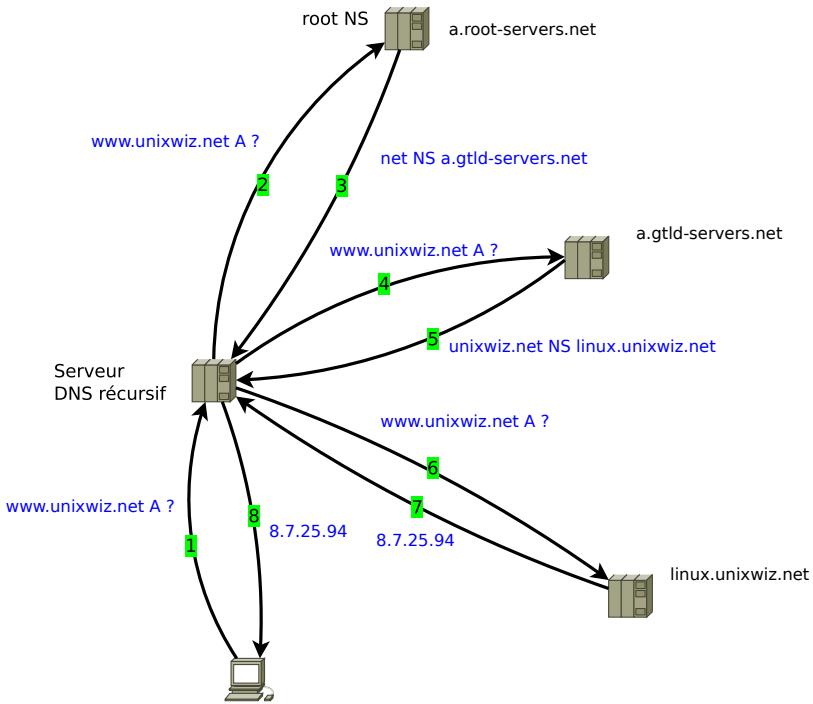


Image tirée de http://fr.wikipedia.org/wiki/Domain_Name_System puis modifiée

Gestion administrative : risques et malveillance

Procédure d'enregistrement

- Premier arrivé, premier servi
- Durée maximum d'enregistrement : 10 ans

$\Rightarrow$ Renouvellement obligatoire

Gestion administrative : risques et malveillance

Procédure d'enregistrement

- Premier arrivé, premier servi
- Durée maximum d'enregistrement : 10 ans

$\Rightarrow$ Renouvellement obligatoire

Risques et malveillance

- Ré-enregistrement de nom de domaine
- Pré-enregistrement de nom de domaine (*domain parking*)
- Enregistrement de nom de domaine similaire (*typo-squatting*)
- Nom de domaines homographes *IDN Homograph attack* [4]

Gestion administrative : risques et malveillance

Procédure d'enregistrement

- Premier arrivé, premier servi
- Durée maximum d'enregistrement : 10 ans

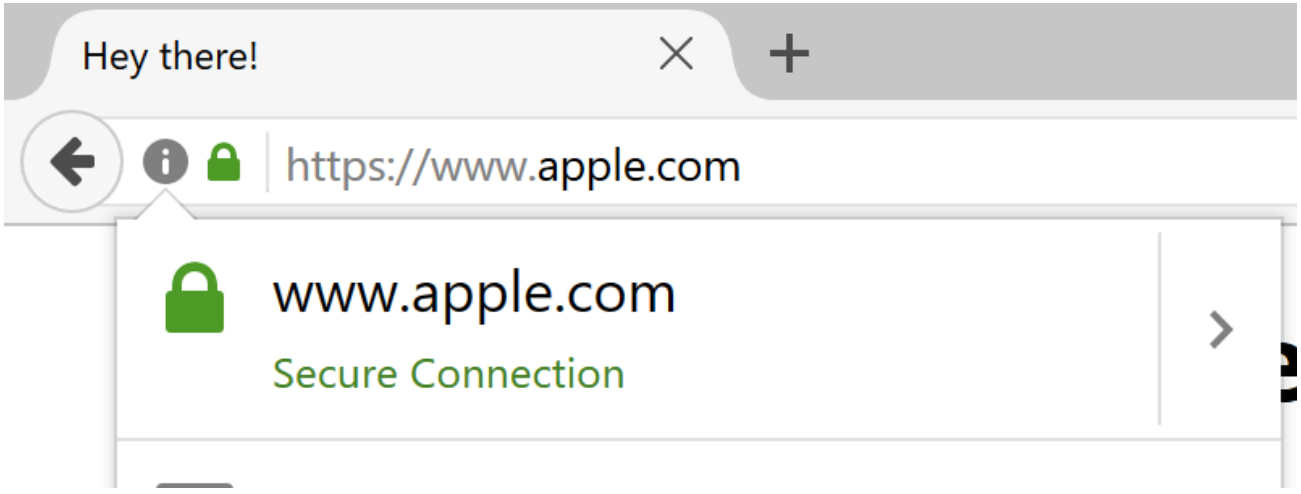
$\Rightarrow$ Renouvellement obligatoire

Risques et malveillance

- Ré-enregistrement de nom de domaine
- Pré-enregistrement de nom de domaine (*domain parking*)
- Enregistrement de nom de domaine similaire (*typo-squatting*)
- Nom de domaines homographes *IDN Homograph attack* [4]

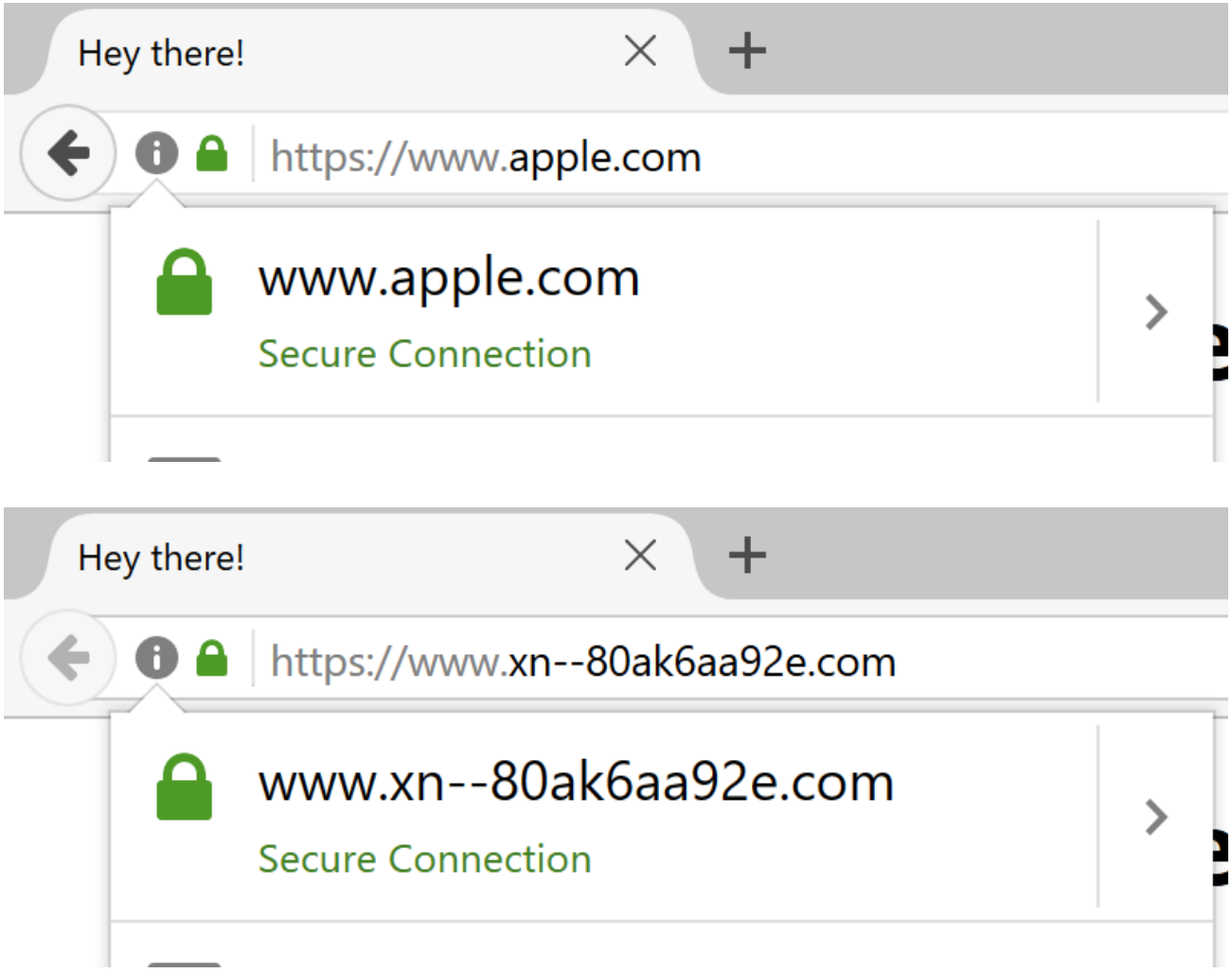
Conséquences ?

Gestion administrative : risques et malveillance



Images : [4]

Gestion administrative : risques et malveillance



Images : [4]

Attaques homographiques

- *Internationalized Domain Names* (IDN)
- Internationalisation individuelle des labels des noms de domaine
- Algorithmes ToASCII (punycode) et ToUnicode
- RFC 3490 [5]
- `libidn`[2]
- `$ idn ( |-d)`
- Certificats x509 ?

Attaques homographiques

- *Internationalized Domain Names* (IDN)
- Internationalisation individuelle des labels des noms de domaine
- Algorithmes ToASCII (punycode) et ToUnicode
- RFC 3490 [5]
- `libidn`[2]
- `$ idn ( |-d)`
- Certificats x509 ?
- Common name punycode

Gestion administrative : risques et malveillance

- *Monitoring* des noms de domaines enregistrés
- Renouvellement automatique des noms de domaine auprès du bureau d'enregistrement
- Désactiver ou le support des noms de domaines internationaux
- Filtres anti *fishing* à l'aide de listes noires
- Affichage d'un nom de domaine international si et seulement si ses caractères appartiennent à un seul langage parmi les langages utilisés par l'utilisateur
- Interdiction de dépôt de nom de premier niveau homographiques auprès de l'ICANN

Configuration dynamique des DNS

- DHCP et DNS spoofs ?

Attaques par homme dans le milieu

- ARP spoofing
- Attaques IGP, EGP

Empoisonnement du cache DNS

Filtre en entrée

Un serveur DNS ne met dans son cache que les réponses pour les requêtes en attente ce qui implique :

- Destinées au port UDP qui était la source de la requête
- La section Question est la bonne dans la réponse
- La section Query ID est la bonne dans la réponse
- Bailiwick check : les enregistrements des sections Authority et Additional ont des domaines correspondant à celui de la question

Objectif

Faire qu'un ou des clients contactent une IP contrôlée par l'attaquant quand ils essayent d'interagir avec un nom de domaine légitime

L'attaquant choisit sa cible en cherchant d'abord un DNS "empoisonnable"

Empoisonnement simple 1/2

DNS mal configurés / anciens

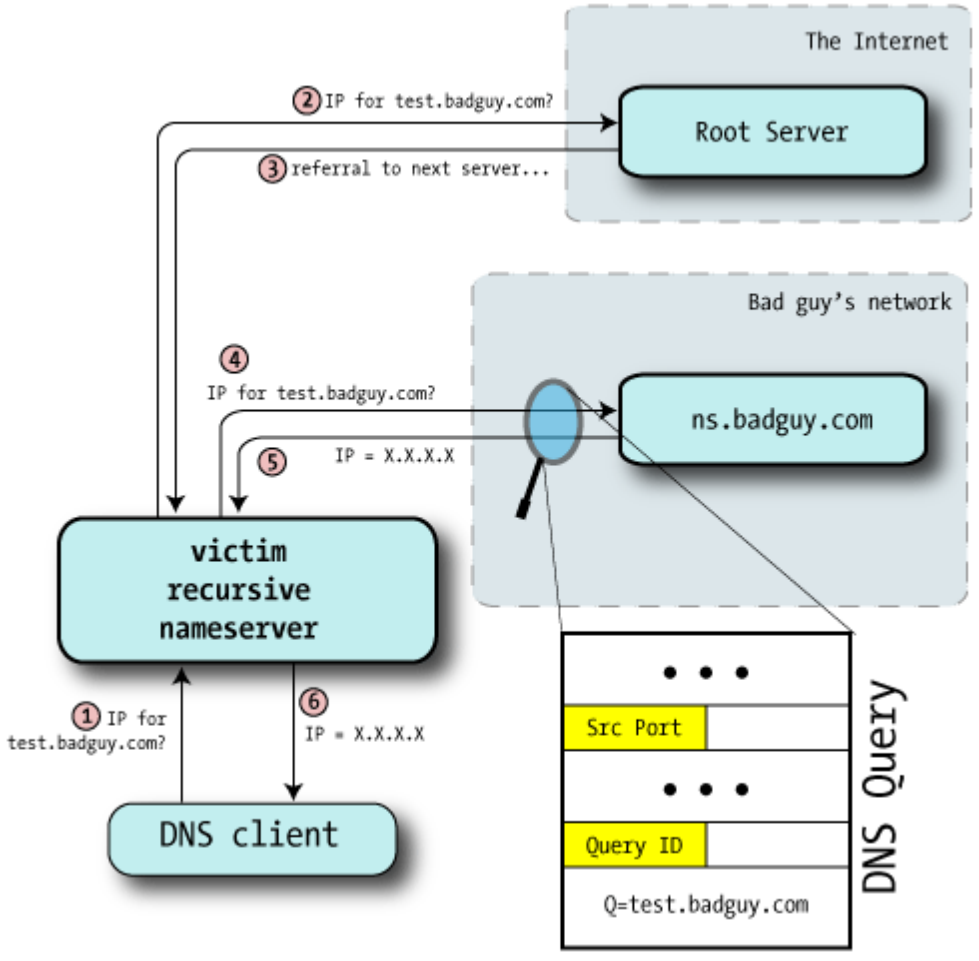
Query ID choisi incrémentalement
Port UDP fixe ou dans petite plage

Il suffit d'observer une requête

- La faire nous-mêmes
- La faire faire par un client légitime (par un frame dans une page web, un e-mail, etc.)

Pour obtenir le port UDP et Query ID

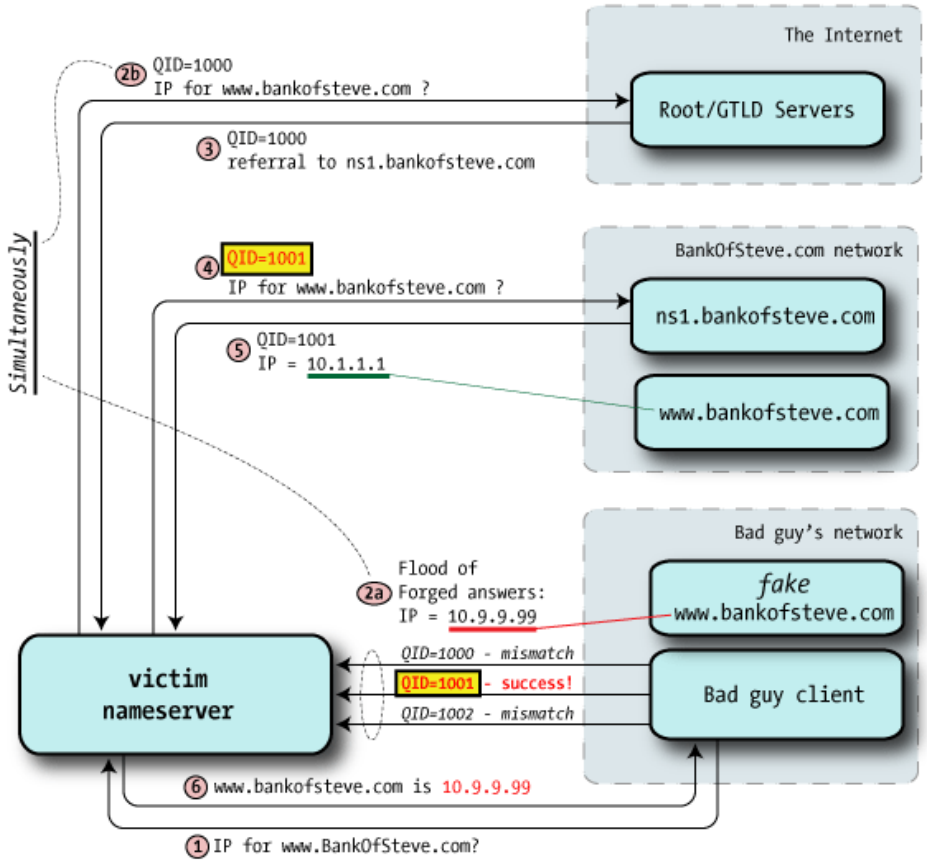
Et donc prévoir comment la suivante requête sera authentifiée



Empoisonnement simple 2/2

Attaque simple

1. L'attaquant fait ou fait faire une requête pour le domaine à empoisonner (e.g. `www.bankofsteve.com`)
2. Il envoie des réponses au bon port et avec des Query ID au suivant le dernier
3. Le root server renvoie le DNS vers `ns1.bankofsteve.com`
4. Une requête est générée pour ce DNS
5. La réponse de `ns1.bankofsteve.com` arrive trop tard
6. Avec un gros TTL tout client qui dans le futur demandera cette page web aura l'IP de l'attaquant en réponse



Limites

Marche pas si le nom de domaine est déjà dans le cache ...
... et si on se rate une fois après c'est dans le cache → one-shot
Contres : randomisation QueryID+port UDP (! paradoxe des anniversaires)

Empoisonnement à la Kaminsky (2008) (1/3)

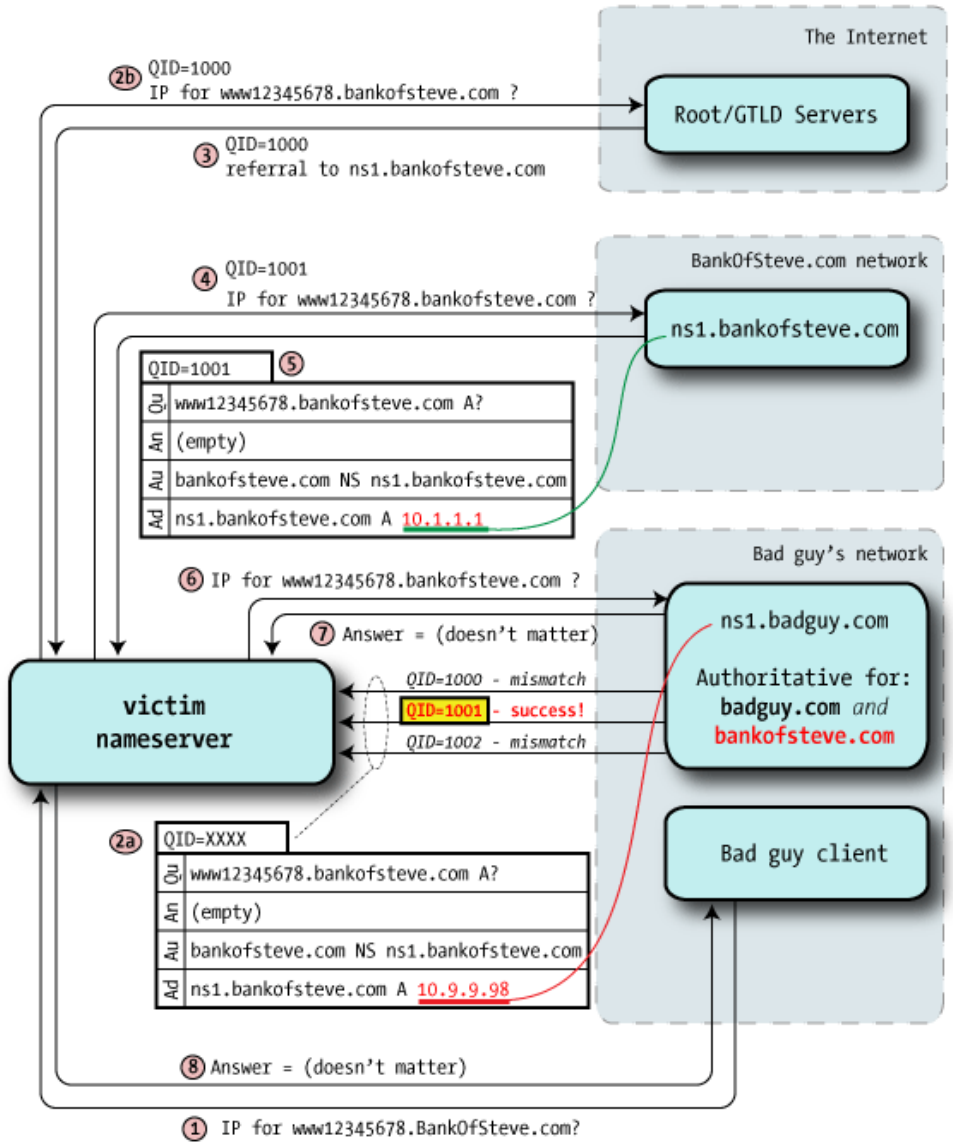
Objectif plus ambitieux et ... plus simple !

Pervertir dans le cache quel est le NS d'un domaine (e.g. `bankofsteve.com`)
Pourquoi plus simple ? On peut tenter l'attaque autant de fois qu'on veut

Attaque

1. Faire ou faire faire une requête vers `[aléa].bankofsteve.com`
2. Envoyer une réponse avec un glue donnant une fausse adresse pour `ns1.bankofsteve.com`
3. Si ça ne marche pas recommencer

Même si le DNS cible est en cache il suffit que `[aléa].bankofsteve.com` ne le soit pas !



Empoisonnement à la Kaminsky (2008) (2/3)

Certains solveurs récursifs utilisent un port source fixe..

Et si on peut sniffer ?

Attaque immédiate quelle que soit la rand. de Query ID !

Le paradoxe des anniversaires

Si on lance directement 256 requêtes et 256 réponses ...
Peut-on passer à immédiat / qqs secondes ?

timeout DNS solver récursif : 2secondes...
`https://www.secureworks.com/blog/dns-cache-poisoning`

Empoisonnement à la Kaminsky (2008) (3/3)

Bug du serveur récursif de bind9 : plusieurs requètes pour un même nom de domaine !!!

Le serveur récursif et l'adversaire tirent au hasard et indépemment un couple de *query id* parmi $m = 2^{16}$ valeurs

$$P(m, n) \approx 1 - e^{-(\frac{n^2}{m})}$$

```
>>> import math
>>> m = 2**16 # 65536
>>> n = math.sqrt(m) # 2^(16/2) = 256
>>> 1 - math.exp(-1*(n**2)/m)
0.63
```

A handbook of applied cryptography, ch2 Mathematical background, 2 Birthday Problems, 2.29 model A

https://www.ida.liu.se/~TDDD17/literature/dnscache.pdf

Plan

1. Introduction
   - Service DNS et organisation
   - Rappels techniques
   - Attaques
2. DNSSEC
3. Fin

DNS Security Extensions

Objectif

Garantir l'authenticité des données DNS, en réponse aux attaques d'empoisonnement de cache (en particulier celle de Kaminsky)

Moyens

Mise en place d'une PKI + Signature de toutes les réponses DNS

$\rightarrow$ Information correcte et complète (par rapport au serveur autoritaire)

Intérêt

Assurer un lien domaine-IP OU domaine-NS domaine-MX domaine-CERT !
Registres CERT : DNSSEC $\rightarrow$ Sécurisation de TLS, SSH, IPSEC

EDNS0 : Extension mechanisms for DNS

C'est quoi ?

Extension de DNS (RFC 2671)

- Garder les paquets DNS classiques
- Ajout d'un pseudo-registre OPT en fin de requête
  - N'existant pas dans les DNS (créé pour la communication)
  - Donnant accès à 16 nouveaux flags
  - Permettant d'étendre la taille des paquets ($>$ 512 octets)

OPT ignoré par anciennes versions $\rightarrow$ retro-compatible

Négociation très simple

- Le client met le champ OPT dans la requête ssi il veut utiliser EDNS0
- Le serveur met OPT dans la réponse ssi il sait le gérer et le client l'a fait
- S'il y a pas deux OPT on aura fait du DNS classique

Comment ça marche

Registres

- **RRSIG** : Signature pour un ensemble de registres (définis par un type et nom)
- **DNSKEY** : Clé publique à utiliser pour vérifier ces signatures
- **DS** : Haché d'une DNSKEY d'un DNS d'un sous-domaine
- **NSEC/NSEC3/NSEC3PARAM** : Hors programme !

Utilisation dans le DNS

Dans le DNS `ns1` du domaine `example.com` ...

- Chaque ens. de registres (même type et nom) a un registre RRSIG associé
- Il est possible de vérifier que tous les registres ont été crées par une personne connaissant la clé privée associée au registre DNSKEY
- Et si un attaquant contrôlant le DNS ou le canal a remplacé DNSKEY ?

→ Utilisation du registre `example.com DS` présent dans le DNS du TLD `.com`

Enregistrement **DNSKEY**

Données communes : nom, TTL, classe, type = RRSIG

Données de l'enregistrement

- Drapeaux [$16-\texttt{bit}$] : clé de zone ?
- Protocole [$8-\texttt{bit}$] : doit être 3
- Algorithme [$8-\texttt{bit}$] : RSA/MD5, DH, RSA/SHA-1, ...
- Clé publique [$N-\texttt{bit}$]

Exemple

```
example.com. 86400 IN DNSKEY 256 3 5 <cle-base-64>=
```

Enregistrement **RRSIG**

Données communes : nom, TTL, classe, type = RRSIG

Données de l'enregistrement

- Type couvert [$16-\texttt{bit}$] : **A**, **AAAA**, ...
- Algorithme [$8-\texttt{bit}$]
- Labels [$8-\texttt{bit}$] : nombre de labels dans le nom, gestion du *wildcard*
- TTL original [$8-\texttt{bit}$] : utile si délégation
- Dates début & expiration [$32-\texttt{bit}$] : validité
- Tag de clé [$16-\texttt{bit}$] : identifier la DNSKEY
- Nom du signataire [$N-\texttt{bit}$] : Nom de la zone propriétaire de l'enregistrement
- Signature [$N-\texttt{bit}$]

Exemple

```
host.example.com. 86400 IN RRSIG A 5 3 86400 20030322173103 (
                         20030220173103 2642 example.com. <sig-base-64>= )
```

Enregistrement **DS**

Données communes : nom, TTL, classe, type = RRSIG

Données de l'enregistrement

- Tag de clé [$16-\texttt{bit}$]
- Algorithme [$8-\texttt{bit}$]
- Fonction de hashage [$8-\texttt{bit}$] : 1 (SHA-1)
- Empreinte [$160-\texttt{bit}$]

Exemple

```
dskey.example.com. 86400 IN DS 60485 5 1 <sha-1-hexa>
```

Preuve de non existence d'enregistrement

Problème

- Comment prouver la non-existence d'un enregistrement ?
- Seul les enregistrements présents sont signés...

Preuve de non existence d'enregistrement

Problème

- Comment prouver la non-existence d'un enregistrement ?
- Seul les enregistrements présents sont signés...

Concept

- Définition et utilisation d'un ordre canonique (nom, classe, type)
- Tri des enregistrements
- Annonce du nom du prochain enregistrement présent dans la zone ainsi que ses types
- Sinature de l'annonce

Preuve de non existence d'enregistrement

Problème

- Comment prouver la non-existence d'un enregistrement ?
- Seul les enregistrements présents sont signés...

Concept

- Définition et utilisation d'un ordre canonique (nom, classe, type)
- Tri des enregistrements
- Annonce du nom du prochain enregistrement présent dans la zone ainsi que ses types
- Sinature de l'annonce

Vérfication

- Recherche dans la liste triée
- Vérification des signatures

Enregistrement **NSEC**

Données communes : nom, TTL, classe, type = RRSIG

Données de l'enregistrement

- Prochain nom de domaine [N − bit]
- Types présents pour le nom courant (*bitmap*) [N − bit]

Exemple

```
alfa.example.com. 86400 IN NSEC host.example.com. (
                                A MX RRSIG NSEC TYPE1234 )
```

Protocole de recherche/validation

Exemple : traitement de la requête `www.example.com A`

1. Il met un bit DO dans le registre OPT de EDNS0 dans sa requête

Protocole de recherche/validation

Exemple : traitement de la requête `www.example.com A`

1. Il met un bit DO dans le registre OPT de EDNS0 dans sa requête
2. Interaction avec un DNS responsable du domaine racine :
   - Il demande les registres `com DS` et `com NS`
   - Il les reçoit ainsi que des glues et les registres RRSIG associés
   - Il les vérifie avec la clé DNSKEY de l'ICANN

Protocole de recherche/validation

Exemple : traitement de la requête `www.example.com A`

1. Il met un bit DO dans le registre OPT de EDNS0 dans sa requête
2. Interaction avec un DNS responsable du domaine racine :
   - Il demande les registres `com DS` et `com NS`
   - Il les reçoit ainsi que des glues et les registres RRSIG associés
   - Il les vérifie avec la clé DNSKEY de l'ICANN
3. Interaction avec un DNS responsable du domaine `com` :
   - Il demande les registres `com DNSKEY`, `example.com NS` et `example.com DS`
   - Il les reçoit ainsi que les glues et les RRSIG associés
   - Il vérifie que la clé retournée correspond au haché `com DS` obtenu du DNS root
   - Il vérifie les RRSIG avec la DNSKEY qui vient d'être validée

Protocole de recherche/validation

Exemple : traitement de la requête `www.example.com A`

1. Il met un bit DO dans le registre OPT de EDNS0 dans sa requête
2. Interaction avec un DNS responsable du domaine racine :
   - Il demande les registres `com DS` et `com NS`
   - Il les reçoit ainsi que des glues et les registres RRSIG associés
   - Il les vérifie avec la clé DNSKEY de l'ICANN
3. Interaction avec un DNS responsable du domaine `com` :
   - Il demande les registres `com DNSKEY`, `example.com NS` et `example.com DS`
   - Il les reçoit ainsi que les glues et les RRSIG associés
   - Il vérifie que la clé retournée correspond au haché `com DS` obtenu du DNS root
   - Il vérifie les RRSIG avec la DNSKEY qui vient d'être validée
4. Interaction avec un DNS responsable de `example.com` :
   - Il demande les registres `example.com DNSKEY`, et `www.example.com A`
   - Il les reçoit avec les glues et RRSIG associés
   - Il vérifie que la clé DNSKEY retournée correspond au haché `example.com DS` obtenu du DNS responsable du domaine `com`
   - Il vérifie les RRSIG avec la DNSKEY qui vient d'être validée

Clé de la racine ?

Pour pouvoir vérifier un enregistrement **A** de `exemple.com.`,

Clé de la racine ?

Pour pouvoir vérifier un enregistrement **A** de `exemple.com.`, il me faut résoudre la clé de signature de `example.com.`

Clé de la racine ?

Pour pouvoir vérifier un enregistrement **A** de `exemple.com.`, il me faut résoudre la clé de signature de `example.com.` → `com.`

Clé de la racine ?

Pour pouvoir vérifier un enregistrement **A** de `exemple.com.`, il me faut résoudre la clé de signature de `example.com.` → `com.` → `.`

Clé de la racine ?

Pour pouvoir vérifier un enregistrement **A** de `exemple.com.`, il me faut résoudre la clé de signature de `example.com.` → `com.` → `.` → ?

Clé de la racine ?

Pour pouvoir vérifier un enregistrement **A** de `exemple.com.`, il me faut résoudre la clé de signature de `example.com.` → `com.` → `.` → ?

- À qui dois-je demander pour la clé de signature de `.` ?

Clé de la racine ?

Pour pouvoir vérifier un enregistrement **A** de `exemple.com.`, il me faut résoudre la clé de signature de `example.com.` $\rightarrow$ `com.` $\rightarrow$ `.` $\rightarrow$?

- À qui dois-je demander pour la clé de signature de `.` ?

$\Rightarrow$ Utilisation d'encre de confiance (*trust anchor*)

- Hashé de clé ou clé de signature de `.` hébergée directement hérbergée sur le serveur récusrsif

Clé de la racine ?

Pour pouvoir vérifier un enregistrement **A** de `exemple.com.`, il me faut résoudre la clé de signature de `example.com.` $\rightarrow$ `com.` $\rightarrow$ `.` $\rightarrow$?

- À qui dois-je demander pour la clé de signature de `.` ?

$\Rightarrow$ Utilisation d'encre de confiance (*trust anchor*)

- Hashé de clé ou clé de signature de `.` hébergée directement hérbergée sur le serveur récusrsif

On suppose que le resolver connaît le DNSKEY de l'ICANN !

Stratégie de gestion des clés

Types de clés

- Clé de signature de zone, *Zone Signing Key* (ZSK)
- Clé de signature de clé, *Key Signing Key* (KSK)

Stratégie de gestion des clés

Types de clés

- Clé de signature de zone, *Zone Signing Key* (ZSK)
- Clé de signature de clé, *Key Signing Key* (KSK)

Résilience aux changements

- Utilisation des KSK pour signer des ZSK
- `example.com.` **DNSKEY** signé par la KSK de `example.com`
- Les enregistrements de la zone sont ensuite signés par une des ZSK

$\Rightarrow$ Stabilité accrue des *glues*

Exemple de domaine supportant DNSSEC

```
$ dig www.internetsociety.org +dnssec

; <<>> DiG 9.13.3 <<>> www.internetsociety.org +dnssec
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51506
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 7, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;www.internetsociety.org.   IN  A

;; ANSWER SECTION:
www.internetsociety.org. 299    IN  CNAME   d229qzkrjypvi4.amimoto-cdn.com.
www.internetsociety.org. 299    IN  RRSIG   CNAME 5 3 300 20181125204001
    20181111204001 39210 internetsociety.org. Lf+ZlLhOe0ihYpRYrgxx6AEyo4jgjmlLg
    /PDL8kc74JoopH8hbX43oZP
    UmrXht4sH343DBzr95x07d2jdujmp7RdIBHmglks6HN1TCgMSz1oP7/k
    JisBsJiUNYjOSqcymbfiXIWlloZr0ODaa1Y4gdJ+QN6MdZIAMG6ZlljF FSg=
d229qzkrjypvi4.amimoto-cdn.com. 60 IN   A   54.192.13.28
d229qzkrjypvi4.amimoto-cdn.com. 60 IN   A   54.192.13.226
d229qzkrjypvi4.amimoto-cdn.com. 60 IN   A   54.192.13.239
d229qzkrjypvi4.amimoto-cdn.com. 60 IN   A   54.192.13.8
d229qzkrjypvi4.amimoto-cdn.com. 60 IN   RRSIG   A 8 3 60 20181201090000
    20180902090000 15288 amimoto-cdn.com. N8ZyvXUi9GFLe3iLWsKIQcqwsHTuaaXx4BbJD
    +z/yGQAwc5MurU9/nvs +YnTTuNkJEpOrHgobfzOV6roSaPZDer0u/wxXBXDKz/XCvntNx6kNe+
    e q8MS3A2ptaIhUKUaP+DdjN/Vpta0no769jJ8bkUJo/qV9VyuBk4dm3cF ILI=
```

Questions pratiques

Et si quelqu'un ne fait pas du DNSSEC ?

Le font : ., .com, .net, .org, .fr, .eu, .be, etc.
Le font pas : .ca, .es, .il, .it, etc.
Un niveau en dessous le pourcentage est très bas ...

- Si tout le monde fait du DNSSEC vous êtes protégé à 100%
- Si le TLD fait du DNSSEC vous pouvez être sûr d'avoir le bon NS, mais pas que ses réponses soient bonnes
- ... sinon DNSSEC vous apporte peu ...

Si votre recursive resolver ne fait pas du DNSSEC vous ne bénéficiez pas de protection (quel que soit le déploiement) !

Alternatives

Utiliser son propre *validating récursive résolver* : `bind9, unbound`
Utiliser un DNS récursif compatible DNSSEC ouvert à tout le monde ...

Fin !

Prochain cours

Suite routage