

DE LA RECHERCHE À L'INDUSTRIE



[www.cea.fr](http://www.cea.fr)

# Sécurité des protocoles Internet

DNS

Pascal MALTERRE

Direction des applications militaires

Département des sciences de la simulation et de l'information

Cellule technique de sécurité informatique

7 janvier 2020

## 1 DNS

- Introduction
- Noms de domaine
- Délégation
- Format de données du DNS
- La sécurité du DNS

## 1 DNS

- Introduction
- Noms de domaine
- Délégation
- Format de données du DNS
- La sécurité du DNS

A l'origine, les noms des machines présentes sur le réseau étaient listés dans un simple fichier texte (HOSTS.TXT)

- mis à jour de manière centralisée (NIC – *Network Information Center*);
- quotidiennement téléchargé (en FTP) par l'ensemble des machines du réseau;
- nombreux problèmes :
  - bande passante consommée par les mises à jour,
  - augmentation du nombre de machines (→ maintenance du fichier compliquée).

## DNS (*Domain Name System*)

Base de données **distribuée** sur laquelle s'appuient *notamment* les applications TCP/IP pour établir une correspondance entre les noms des machines et leurs adresses IP.

### Base de données "distribuée"

- À un instant donné, aucun serveur DNS ne dispose de toute les informations de la base.
- Chaque serveur DNS présent sur Internet maintient sa propre base de données (*i.e.* une partition de la base globale) et expose un service réseau que les autres serveurs peuvent utiliser.

Les spécifications du DNS incluent :

- Un protocole réseau (UDP/53 et dans certains cas TCP/53) permettant aux clients et aux serveurs de communiquer
- Le DNS fournit également des informations d'acheminement pour les mails

## Importance de la disponibilité du service

Le DNS est un système vital pour le fonctionnement des réseaux TCP/IP et particulièrement pour Internet

- Les premières spécifications ont été posées dans les RFCs 882 et 883 (Paul Mockapetris, 1983)
- Le fonctionnement actuel est décrit en grande partie par les spécifications suivantes :
  - [RFC-1034](#) "Concept and Facilities"
  - [RFC-1035](#) "Implementation and Specification"
- De nombreuses extensions ont été spécifiées dans des RFCs supplémentaires, par exemple :
  - [RFC-2535](#) "DNS Security Extensions (DNSSEC)"
  - [RFC-2136](#) "Dynamic Updates in the DNS"

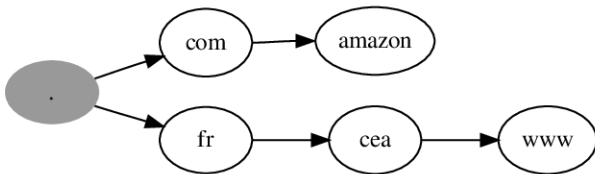
...

- L'implémentation la plus utilisée est BIND
  - Berkeley (*Buggy*) Internet Name Domain Server
- Microsoft DNS Server
- djbdns
- NSD (NLNet Labs)
- PowerDNS, MaraDNS, etc.



L'espace des noms DNS est organisé selon une **structure arborescente**.

- Chaque noeud (et feuille) de l'arbre représente un nom DNS, et on y associe un ensemble (éventuellement vide) de données de différents types
- Une requête DNS a pour objectif de récupérer les données d'un certain type associées à un nom



Au niveau applicatif, l'accès au service DNS s'effectue à travers un "resolver". Le plus souvent, il s'agit d'une librairie dynamique fournie par le système d'exploitation :

- Le *resolver* ne fait pas partie de la pile TCP/IP
- Fonctions `gethostbyname( )` et `gethostbyaddr( )`

## 1 DNS

- Introduction
- **Noms de domaine**
- Délégation
- Format de données du DNS
- La sécurité du DNS

Un nom de domaine qui se termine par un point est appelé nom de domaine absolu ou nom de domaine pleinement qualifié (FQDN)

## Example

```
chani.dif.dam.intra.cea.fr.
```

Si le nom de domaine ne se termine pas par un point, il a besoin d'être complété : ce mécanisme dépend du resolver utilisé

## Example

Les directives `domain` et `search` dans le fichier `/etc/resolv.conf` sous Unix

## TLD (*Top-Level Domain*)

Dernier label (le plus à droite) d'un nom de domaine

### Différentes catégories

- ARPA est un domaine spécial ("*infrastructure TLD*") utilisé pour les correspondances adresses IP vers noms
- Les domaines basés sur les codes de pays de la norme ISO 3166 et codés sur deux caractères ("*country-code TLD*")
- Les domaines génériques : .com, .net, etc.

Historiquement, on trouve sept domaines génériques de niveaux supérieurs destinés à des utilisations spécifiques (enfin en théorie !)

- `.com` organisation commerciales
- `.edu` institutions éducatives
- `.gov` organisations gouvernementales US
- `.mil` militaire US
- `.int` organisations internationales
- `.org` autres organisations
- `.net` réseau

## Remarque

- Nouveaux domaines : `.museum`, `.info`, `.biz`, etc.

## 1 DNS

- Introduction
- Noms de domaine
- **Délégation**
- Format de données du DNS
- La sécurité du DNS

- Aucune entité ne gère tous les labels de l'arbre
- Les TLDs sont gérés par le NIC qui délègue à divers organismes
- Les extensions de pays sont gérés par les NICs régionaux (AFNIC)

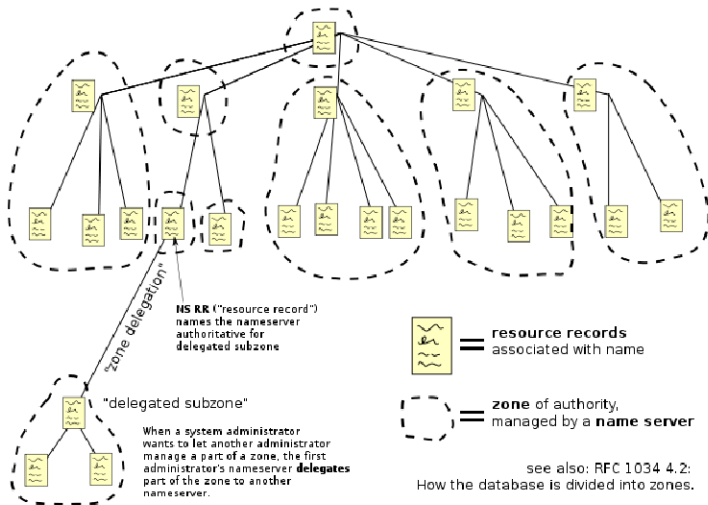


**Zone DNS** Portion de l'arbre DNS administrée de manière autonome

- Une fois que l'autorité est déléguée pour une zone, c'est à la personne qui en est responsable de fournir des serveurs de noms pour cette zone
- Plusieurs serveurs sont en général requis pour une question de redondance

## LA NOTION DE ZONE DNS (SUITE)

## Domain Name Space



see also: RFC 1034 4.2:

How the database is divided into zones.

Réels avantages ? Beaucoup d'entreprises proposent des services de type "DNS secondaire"

- D'après certaines études, 25% des zones DNS ont leurs serveurs situés sur le même réseau
- Difficulté de propagation pour les changements urgents (effets de cache, etc.)
- Sécurité

Pour un nom de domaine, le propriétaire doit payer un abonnement auprès des entités gérant les serveurs ayant autorité sur les TLDs

- Notion de "*registrars*"
- Modalité de dépôt pour un nom de domaine : premier arrivé premier servi

## Différents types de contact

**Administratif** pour les modifications dans le whois

**Technique** pour l'aspect opérationnel

**Facturation** pour l'aspect financier

## Root Nameservers

- Les serveurs DNS racines connaissent (i.e. "publient") la liste des serveurs ayant autorité sur les TLDs
- Il existe 13 serveurs racines sur internet associés aux noms de la forme `{A..M}.root-servers.net`

## Remarques

- La plupart des adresses IP sont distribuées par *anycast*
- Des études montrent que 98% des requêtes traitées par les serveurs racines sont inutiles (bruit, erreurs de configuration, etc.)

On peut utiliser un espace de nommage DNS alternatif en changeant les *root nameservers*.

- OpenNIC, Public-Root, etc.

Les raisons peuvent être idéologiques ou commerciales.

[RFC-2826](#) (*Internet Architecture Board - Technical Comment on the Unique DNS Root*)

"To remain a global network, the Internet requires the existence of a globally unique public name space"

# LOCALISATIONS DES SERVEURS DNS RACINES



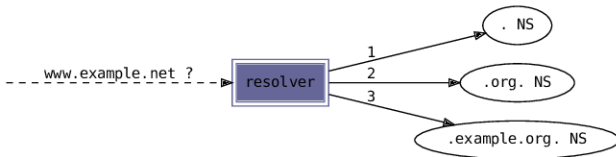
Février 2007

- Le resolver traite **itérativement** chaque composant du nom de domaine en commençant par la droite (racine, puis TLD, etc.)
  - Pour démarrer la recherche, on doit donc connaître les adresses IP des serveurs racines (configuration statique)
- A chaque étape, le resolver effectue une requête pour connaître le serveur autoritaire de la zone DNS définie par l'élément inférieur
  - Au début de la recherche, le resolver demande aux serveurs racines quels sont les serveurs autoritaires pour le TLD



## Exemple

Résolution du nom "www.example.org"?



Le DNS est basé sur un ensemble de serveurs organisés de façon hiérarchique et que l'on peut grouper en trois catégories :

- Serveurs primaires (ou maîtres)
- Serveurs secondaires (ou esclaves)
- Serveurs caches (ou forwarders)

Les serveurs primaires et secondaires ont pour rôle de publier des informations pour la zone DNS pour laquelle ils sont autoritaires.

- Un serveur primaire charge ses informations sur la zone à partir d'une base de données locale (fichier sur le disque ou autre)
- Un serveur secondaire obtient les informations à partir du serveur primaire grâce à un mécanisme réseau appelé "transfert de zone"

→ Un serveur DNS (primaire ou secondaire) n'a pas à fournir d'informations **autres** que celles pour lesquelles il est autoritaire

## Serveur de cache DNS

Un serveur de cache DNS (ou *forwarder DNS*) doit :

- ➊ prendre en charge les requêtes des clients
- ➋ effectuer tout le mécanisme de résolution à leur place (i.e. les itérations successives en partant des serveurs racines)
- ➌ stocker les résultats dans une mémoire cache

## Remarques

- Utilisation de chaînes de *forwarders*
- Serveur "privé"

## 1 DNS

- Introduction
- Noms de domaine
- Délégation
- **Format de données du DNS**
- La sécurité du DNS

# NOTION DE "RESOURCE RECORDS" (RR)

Chaque noeud de l'arbre DNS contient un ensemble (éventuellement vide) d'enregistrements. Chaque enregistrement contient :

- Un champ "type" codé sur deux octets spécifiant le type des données de l'enregistrement : A, PTR, CNAME, MX, etc.
- Un champ "class" codé sur deux octets identifiant le protocole relatif aux informations de l'enregistrement.  
→ La classe IN est réservée aux adresses Internet.
- Un champ TTL codé sur 32 bits représentant une durée de validité en secondes
- Et enfin les données (RDATA)

Les messages DNS (**requêtes** et **réponses**) sont constitués :

- d'un entête composé de différents champs de taille fixe
- d'une zone de données divisée quatre sections :
  - Requête (*Question*)
  - Réponse (*Answer*)
  - Autorité (*Authority*)
  - Données additionnelles (*Additional*)

Ces types d'enregistrements sont utilisés pour la conversion des adresses IP en nom et inversement

**A, AAAA** Le champ RDATA contient une adresse IPv4 (32 bits) ou IPv6 (128 bits)

**PTR** Le champ RDATA est un pointeur vers une autre branche de l'espace de nommage DNS



## CNAME (*Canonical Name*)

indique que l'enregistrement courant est un alias et fournit le nom DNS officiel auquel il fait référence dans le champ RDATA

### Remarques

Les enregistrements de type CNAME sont souvent source d'erreurs et de confusion. Les spécifications préconisent différentes choses :

- Si un noeud de l'arbre DNS contient un RR de type CNAME, alors il ne doit pas contenir d'autres données
- Les CNAME ne doivent pas être chaînés

# PRINCIPAUX TYPES DE RESSOURCES

Quelques exemples à l'aide du programme `host`

- Enregistrements associés à `www.google.com` ?

```
bash$ host -t any www.google.com
```

```
www.google.com.      325863 IN      CNAME  www.l.google.com.
```

- Enregistrements de type A de `www.l.google.com` ?

```
bash$ host -t a -v google.l.com
```

```
www.l.google.com.    143   IN      A       209.85.129.99
www.l.google.com.    143   IN      A       209.85.129.104
www.l.google.com.    143   IN      A       209.85.129.147
```

## Informations sur la zone DNS

### SOA (*Start Of Authority*)

ce type de ressource fournit diverses informations sur la zone DNS à laquelle appartient l'enregistrement courant

Un enregistrement SOA contient :

- Le serveur DNS ayant autorité sur cette zone (MNAME)
- L'adresse mail du responsable de la zone (RNAME)
- Un numéro de série codé sur 32 bits (SERIAL)
- Diverses valeurs temporelles codées sur 32 bits utilisées par les mécanismes de cache
  - REFRESH, RETRY, EXPIRE, MINIMUM

## ■ Informations de la zone google.com?

```
bash$ host -t soa google.com

google.com. 86202 IN SOA (
    ns1.google.com.      # MNAME
    dns-admin.google.com. # RNAME
    2007122501           # SERIAL
    7200                 # REFRESH
    1800                 # RETRY
    1209600              # EXPIRE
    300                  # MINIMUM
    )
```

**NS** définition du serveur de nom autoritaire pour (i.e. "à partir de")  
cet enregistrement

Le données pour les ressources de ce type représentent un nom DNS, par exemple :

```
bash$ host -t ns -v orange.fr
```

```
;; ANSWER SECTION:
```

orange.fr.	2639	IN	NS	ns.wanadoo.fr.
orange.fr.	2639	IN	NS	ns2.wanadoo.fr.

Pour déléguer une sous-zone à une autre entité, on doit avoir un enregistrement de type NS indiquant les serveurs DNS qui ont autorité sur cette sous-zone.

Le serveur DNS peut être référencé :

- par un nom appartenant à la sous-zone pour laquelle il est autoritaire (méthode conseillée)  
→ utilisation de *glue records*
- par un nom quelconque  
→ attention aux boucles et aux performances

## Principe

L'idée est de s'appuyer sur l'infrastructure du DNS (répandue et supportée par tous les clients) pour localiser les services sur un réseau (cf. RFC 2782).

## Localisation des services

La RFC spécifie d'abord une convention permettant d'interroger le DNS afin de localiser un service : `_service._proto.target`

Par exemple, pour trouver le serveur LDAP un client demandera à son serveur DNS les enregistrements de type SRV pour le nom de domaine : `_ldap._tcp.mon-domaine.com`

## Informations de l'enregistrement SRV

En plus des champs habituels (TTL, CLASS), on trouve les informations suivantes :

- Priorité** entier codé sur 16 bits indiquant l'ordre de préférence si plusieurs enregistrements sont retournés
- Poids** entier codé sur 16 bits affinant le choix du serveur en cas de multiples priorités de même valeur
- Port** port réseau sur lequel écoute le service
- Target** nom DNS du serveur fournissant le service



**MX** fournit le nom de la machine responsable du routage des mails pour ce domaine (*Mail eXchanger*)

**TXT** définition d'un champ arbitraire de type texte

L'objectif est de retrouver le nom à partir de l'adresse IP

- Utilisation des mécanismes du DNS avec un TLD spécifique "in-addr.arpa"
- L'adresse IP est utilisée comme un nom de domaine (séquence de labels séparées par des « . »)

Une entité qui rejoint internet (i.e. qui dispose de quelques adresses IP publiques) obtient l'autorité pour une partie de l'espace d'adressage du DNS mais aussi pour une portion de in-addr.arpa.

- L'affectation des plages d'adresses IP se fait de manière hiérarchique (cf. CIDR)
- L'adresse est donc inversée avant d'être concaténée au domaine in-addr.arpa

- Le type de ressource utilisé pour les requêtes est "PTR"

## Exemple

Pour connaître le nom de la machine ayant l'adresse IP 193.252.19.3, le resolver demande l'enregistrement PTR pour le nom de domaine 3.19.252.193.in-addr.arpa

## Remarque

Plusieurs enregistrements peuvent être retournés mais seul le premier est utilisé

# MISE EN CACHE DES RÉPONSES NÉGATIVES

La mise en cache des réponses négatives était optionnelle dans les spécifications initiales du DNS (→ mise à jour dans la RFC 2308)

On trouve généralement **deux** catégories de réponses négatives :

- ❶ « **Name Error** » → RCODE=3 (NXDomain) → Le nom DNS en objet de la requête (ou le *dernier* nom DNS de la chaîne des CNAME) n'existe pas
- ❷ « **No Data** » → RCODE=0 (NoError) → Le nom DNS existe mais il ne contient pas les enregistrements de ressource (RR) demandés

Une réponse négative ne contient pas d'enregistrement, et donc, pas de champ TTL utilisable par un *resolver* pour cacher l'absence de donnée :

- Le serveur autoritaire ajoute un enregistrement SOA dans la section *Authority* de sa réponse « *NXDomain/No Data* »
- Le TTL de cet enregistrement est positionné à la valeur du champ MINIMUM de l'enregistrement SOA (→ durée utilisée par les caches DNS)

- Résolution inverse des adresses privées (RFC-1918)
- Qualification des noms de domaines (directives domain et search pour le resolver)
- Auto-complétion des browsers web
- Enregistrement SOA inexistants
- Interceptions des NXDOMAIN

Grâce au DNS, on peut mettre en oeuvre un système d'équilibrage de charge entre plusieurs machines appelé "DNS *round-robin*"

## Principe

Une feuille de l'arbre DNS peut contenir plusieurs enregistrements du même type (par exemple trois RR de type A avec des adresses IP différentes)

- La réponse du serveur intègre tous les enregistrements dans un ordre aléatoire ou judicieusement choisi (par exemple en fonction de la géolocalisation du client)
- Les clients utilisent généralement la première valeur

→ Les valeurs de TTL doivent être faibles pour diminuer la durée de vie dans les caches intermédiaires

Les échanges DNS sont principalement basés sur UDP avec le port 53/domain coté serveur.

- Les messages sont limités à 512 octets par la RFC 1035
- Les resolvers doivent mettre en oeuvre des stratégies de retransmission adaptées

## Utilisation de TCP

Si la réponse du serveur est trop grande, alors le message est tronqué et le bit TC est positionné dans l'entête.

- Le client doit alors ré-emettre sa requête en TCP
- Blocage des flux TCP liés au DNS

```

▶ Frame 1 (74 bytes on wire, 74 bytes captured)
▶ Ethernet II, Src: WwPcbaTe_77:e4:76 (00:0f:1f:77:e4:76), Dst: ComdaEnt_71:f4:00 (00:d0:03:71:f4:00)
▶ Internet Protocol, Src: 172.23.20.1 (172.23.20.1), Dst: 132.165.71.100 (132.165.71.100)
▶ User Datagram Protocol, Src Port: 40746 (40746), Dst Port: domain (53)
▼ Domain Name System (query)
  [Response In: 2]
  Transaction ID: 0x85e0
  ▼ Flags: 0x0100 (Standard query)
    0... .. = Response: Message is a query
    .000 0... .. = Opcode: Standard query (0)
    .... 0. .... = Truncated: Message is not truncated
    .... 1 .... = Recursion desired: Do query recursively
    .... 0. .... = Z: reserved (0)
    .... 0 .... = Non-authenticated data OK: Non-authenticated data is unacceptable
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
  ▼ Queries
    ▼ www.google.com: type A, class IN
      Name: www.google.com
      Type: A (Host address)
      Class: IN (0x0001)

```



## ANALYSE D'UNE RÉPONSE DNS

```

> User Datagram Protocol, Src Port: domain (53), Dst Port: 40746 (40746)
▼ Domain Name System [response]
  Request ID: 11
  [Time: 0.001463000 seconds]
  Transaction ID: 0x85a0
  ▼ Flags: 0x0100 (Standard query response, No error)
    1... .. = Response: Message is a response
    .000 0... .. = Opcode: Standard query (0)
    ....0... .. = Authoritative: Server is not an authority for domain
    ....0... .. = Truncated: Message is not truncated
    ....1... .. = Recursion desired: Do query recursively
    ....1... .. = Recursion available: Server can do recursive queries
    ....0... .. = Z: reserved (0)
    ....0... .. = Answer authenticated: Answer/authority portion was not authenticated by the server
    ....0000 = Reply code: No error (0)

  Questions: 1
  Answer RRs: 5
  Authority RRs: 0
  Additional RRs: 0
  ▼ Queries
    ▼ www.google.com: type A, class IN
      Name: www.google.com
      Type: A (Host address)
      Class: IN (0x0001)
    ▼ Answers
      ▶ www.google.com: type CNAME, class IN, cname www.l.google.com
      ▶ www.l.google.com: type A, class IN, addr 74.125.43.99
      ▶ www.l.google.com: type A, class IN, addr 74.125.43.103

```

## Définition

C'est le mécanisme par lequel un serveur secondaire (ou esclave) récupère les informations sur la zone en interrogeant le serveur primaire (ou maître).

## Principe

- Le transfert est initié en TCP **par le serveur esclave** en se connectant sur le port 53/domain du serveur maître
- Le serveur esclave vérifie si une nouvelle version de la zone est disponible à l'expiration de la durée de validité (SOA)

## Remarques

- Pas de prise en compte automatique des nouvelles zones
- Pas de chiffrement ni de contrôle d'intégrité
- A cause du délai de propagation des informations induit par le sens du transfert (secondaire → primaire), un mécanisme de notification est mis en oeuvre (NOTIFY)
  - mécanisme non fiable car basé sur UDP/53
- Transfert incrémental (IXFR)
  - certaines informations ne sont pas propagées
- Filtrage réseau compliqué à mettre en place

## 1 DNS

- Introduction
- Noms de domaine
- Délégation
- Format de données du DNS
- La sécurité du DNS

Pour une entreprise (ainsi que pour Internet), l'infrastructure DNS est un composant critique.

- Importance du cloisonnement réseau
- Choix des implémentations

Un certificat X509 est un lien (de confiance), validé par une autorité de certification (CA), entre une clé publique et une identité.

Pour un site web, l'identité est le **nom DNS**; pour valider une demande de certificat, la CA doit vérifier que le demandeur « possède » bien le nom de domaine :

- création d'un enregistrement de ressource;
- création d'un sous-domaine;
- ...

D'un point de vue sécurité, un serveur DNS maître est une machine **très sensible** dans le sens où elle est responsable des informations publiées dans le DNS.

- Le chargement des données depuis une base locale (système de fichiers ou autre) implique des besoins spécifiques : filtrage réseau, administration et gestion des zones DNS par une personne spécifique, etc.
- Par ailleurs, le service doit être généralement accessible depuis un grand nombre d'utilisateurs (voire depuis Internet).

⇒ Ces besoins ne sont pas vraiment compatibles

Une solution consiste à mettre en place un serveur secondaire en frontal (offrant le service à l'extérieur), et à cacher l'existence du serveur primaire (pouvant être situé plus à l'intérieur du réseau)

## Mise en oeuvre

- ➊ Paramétrer (et sécuriser) un serveur secondaire en frontal
- ➋ Déclarer un seul champ NS dans la zone DNS publiée
- ➌ Mettre en place le transfert de zone entre les deux machines



Attaque consistant à renvoyer des informations erronées à un serveur de cache de manière à ce que ces dernières soient enregistrées dans la mémoire cache.

## Techniques possibles (historiques)

- Utilisation de la section "*additional data*" pour forcer la mise à jour d'informations tierces
- Injection de fausses réponses (DNS *forgery*) en essayant de deviner l'id de transaction (TXID)

Les implémentations actuelles des serveurs DNS ne sont plus vulnérables à ces deux attaques

→ Attention aux attaques locales contre le resolver/cache

Le DNS est un protocole sans état : pour un serveur cache, la correspondance entre les **requêtes envoyées** et les **réponses reçues** est basée sur le couple (port source, TXID)

## Problème du port source

En août 2008, quasiment toutes les implémentations des caches DNS utilisent un port source **fixe** pour l'envoi de requêtes

→ Problème pointé par D.J. Bernstein en 1998

Le port source étant connu, forger une réponse valide revient à deviner l'id de transaction (codé sur 16 bits).

→ Paradoxe des anniversaires

## Description de l'attaque

Pour insérer de fausses informations dans un serveur DNS cache pour le nom `www.domaine.com`, l'attaquant va simultanément :

- Envoyer au serveur cache vulnérable des milliers de requêtes concernant des enregistrements inexistants (→ NXDOMAIN)
  - `aaaaa.domaine.com`, `aaaab.domaine.com`, ...
- Parallèlement, forger les milliers de réponses correspondantes en faisant varier le TXID et contenant une section *additional data* contenant les données à altérer

→ Une correspondance se produit rapidement (après quelques milliers de requêtes)

## Définition

Ce sont les techniques d'attaque permettant de déterminer si une information est **déjà présente** dans un serveur DNS de cache  
→ fuite d'informations

## Méthodes

- Interroger le cache du serveur DNS en forçant le mode **non-récuratif**
- Etudier les valeurs obtenues pour le TTL de l'enregistrement ou bien les temps de réponse du serveur

La meilleure protection contre les attaques de ce type consiste à **séparer les services DNS** en fonction du type des clients qui les utilisent

- Les serveurs DNS primaires ou secondaires sont généralement des services **publics**
- Les serveurs de cache sont plutôt destinés à une population d'utilisateurs spécifiques (utilisateurs d'un LAN, abonnés d'un ISP, etc.)

Menaces différentes  $\Rightarrow$  Mesures de protection adaptées

Par exemple, pour un serveur de cache on peut interdire les requêtes non-récurrentes et fixer le TTL à 0

Séparer les services de **résolution** (serveur cache) et de **publication** (serveur primaire/secondaire) revient simplement à mettre en oeuvre les grands principes de sécurité :

- **Séparation des privilèges**

- réduction de la surface d'attaque et des impacts liés à une faille de sécurité

- **Moindre privilège**

- implémentation plus simple et plus sûre

