

0-0

# Serveur de Nom de Domaine dans Internet (DNS)

---

## Berkeley Internet Name Domain (BIND)

<http://enstb.org/~keryell/cours/IAR2M/BIND>

---

Ronan Keryell

rk@enstb.org

---

Institut TÉLÉCOM, TÉLÉCOM Bretagne, Département Informatique

High Performance Computing Architecture & Security

Plouzané, France

27-28 mars 2008

Version 1.32



- Copyright (c) 1986–2037 by Ronan.Keryell@cri.ensmp.fr.  
This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, v1.0 or later (the latest version is presently available at <http://www.opencontent.org/openpub/>).
  - Ce support sert à mes cours dispensés à l'École des Mines de Paris et en formation continue à TÉLÉCOM Bretagne. Si vous améliorez ces cours, merci de m'envoyer vos modifications ! ☺
- \$Id: trans.tex,v 1.32 2008/03/27 07:02:48 keryell Exp keryell \$

- Adresses IP de 32 bits pour numéroter les machines sur Internet  
≡ 10 chiffres décimaux
- Utilisation intensive pour le courrier, WWW, news, telnet, ftp,...
- IPv6 sur 128 bits ≡ ↵ 39 chiffres décimaux à retenir...
- Nécessité de noms mnémotechniques et plus commerciaux
- Conversion entre numéros IP et noms de machines et autres
- Nécessite un visibilité mondiale
- Offrir une distribution spatiale
- Gérer la cohérence temporelle
- Besoin d'une bonne tolérance aux pannes

## Domain Name System (DNS)

-  Importance stratégique
- DNS en panne : perte de services importants 😞
- DNS corrompu : système délirant 😞
- DNS piraté : pages WWW pointant vers la concurrence 😞
-  Bien choisir ses noms (marques, lisibilité...)
-  Bien payer à temps ses enregistrements (si payants)
-  Monde où la carte banquaire est reine... Attention aux retards administratifs ! Sinon 😞
-   ∃ boîtes internationales de cyber-racket qui rachètent un domaine dès qu'il devient libre (si on a oublié le loyer), font pointer vers des sites pornographiques et veulent le revendre cher 😞

- ARPAnet : quelques machines au début
- Toutes les informations sur les nœuds du réseau étaient dans LE fichier HOST.TXT, RFC 952

NET : 198.49.236.0 : PENSACOLANET2 :

GATEWAY : 26.10.0.14, 147.36.15.1 : PIRMASSENS-GW1.ARMY.MIL : CISCO-AGS-1

HOST : 134.229.2.2 : PENS-EMH1.NAVY.MIL : ATT-3B2 : UNIX : X.25,TCP/IP,TCP/

HOST : 134.124.40.5 : WHALENS.UMSL.EDU :: AIX : TCP/TELNET,TCP/FTP,TCP/S

HOST : 137.194.160.1 : ULYSSE.ENST.FR : SUN : UNIX ::

- Maintenu par le Network Information Center au SRI
- Récupéré périodiquement par les nœuds du réseau
- Problème avec ↗ nombre machines
  - ▶ Chaque modification/ajout de machine ↗ mail au SRI
  - ▶ Mise à jour ↗ transfert du fichier vers nombreuses machines  
↘ performance du réseau

- ▶ Conflits de noms globaux
- ▶ Problèmes de cohérence entre les copies existantes
- Toujours dans UNIX !
  - ▶ gettable : récupère HOST.TXT
  - ▶ htable : HOST.TXT ↵ /etc/hosts et autres

↵ Trouver autre chose...

- Trouver un nouveau système
- Extensible
- Sans goulet d'étranglement
- Distribué
- Administration locale décentralisée
- ↵ RFC 882 et RFC 883 en 1984
- Réalisation de JEEVES
- Réalisation de BIND sur UNIX 4.3BSD

<http://www.isc.org/ds/WWW-200301> : Internet Domain Survey, Jan 2003, Number of Hosts advertised in the DNS

	Survey Date	Host Count	Adjusted Host Count	Replied To Ping*
Jan 1993	1,313,000			
Jul 1993	1,776,000			464,000
Jan 1994	2,217,000			576,000
Jul 1994	3,212,000			707,000
Jan 1995	4,852,000	5,846,000		970,000
Jul 1995	6,642,000	8,200,000		1,149,000
Jan 1996	9,472,000	14,352,000		1,682,000
Jul 1996	12,881,000	16,729,000		2,569,000
Jan 1997	16,146,000	21,819,000		3,392,000
Jul 1997	19,540,000	26,053,000	4,314,410	[last OLD Survey]
Jan 1998	29,670,000		5,331,640	[first NEW Survey]

Jul 1998	36,739,000	6,529,000
Jan 1999	43,230,000	8,426,000
Jul 1999	56,218,000	-
Jan 2000	72,398,092	-
Jul 2000	93,047,785	-
Jan 2001	109,574,429	-
Jul 2001	125,888,197	-
Jan 2002	147,344,723	-
Jul 2002	162,128,493	-
Jan 2003	171,638,297	-

Évidemment, plein de machines non déclarées, partageant des adresses (fournisseur d'accès) ou NAT sur des adresses privées

- Aucun ! /etc/hosts... Peut suffire pour les noms locaux si réPLICATION automatisée (cf Cfengine par exemple). Intérêt : robustesse ! Mais compléter avec le DNS...
- NIS Network Information Service (origine Sun)
- NIS+ (origine Sun) hiérarchique et contrôles d'accès
- Annuaires X.500 et LDAP (solution à la mode, version simplifiée du précédent)
- WINS Windows Internet Name Service (MicroSoft)
- ...
- DNS ≡ protocole de nommage unificateur sur Internet
- FNS (Sun) multiplexe le tout : fichiers, NIS, NIS+, DNS, LDAP,...

- Les protocoles d'Internet sont en libre accès  
<http://www.rfc-editor.org>
- Pour commencer le STD1 ou RFC 3600 : Internet Official Protocol Standards
- Définition modeste : Request For Comment ☺
- Différentes classes
  - ▶ Protocoles standards sous forme de STD qui pointent vers le dernier RFC à jour de ce standard. Commencer avec le STD1
  - ▶ Protocoles standards sous forme de RFC seulement
  - ▶ Brouillons (*drafts*) de protocoles
  - ▶ Protocoles standards en cours de proposition
  - ▶ Bonnes pratiques sous forme de BCP (*Best Current Practice*). Commencer avec le BCP1
  - ▶ Bonnes pratiques sous forme de RFC

- <http://www.nic.fr> : Site de l'AFNIC (Association Française pour le Nommage Internet en Coopération)
  - ▶ <http://www.afnic.fr/formation/supports>
  - ▶ <http://www.afnic.fr/guides>
  - ▶ <http://www.nic.fr/guides/dns-intro>
  - ▶ <http://www.afnic.fr/outils>
- <http://www.domainesinfo.fr> des informations plus politiques
- <http://www.generic-nic.net> : « Le projet "NIC générique" essaie de rassembler tous les documents utiles à un nouveau NIC (Network Information Center) ou bien à un NIC existant qui est engagé dans une grande transformation »
- <http://www.dns.net/dnsrd/> : DNS Resources Directory
  - ▶ <http://www.dns.net/dnsrd/rfc/> (& <http://www.bind9.net/rfc>)

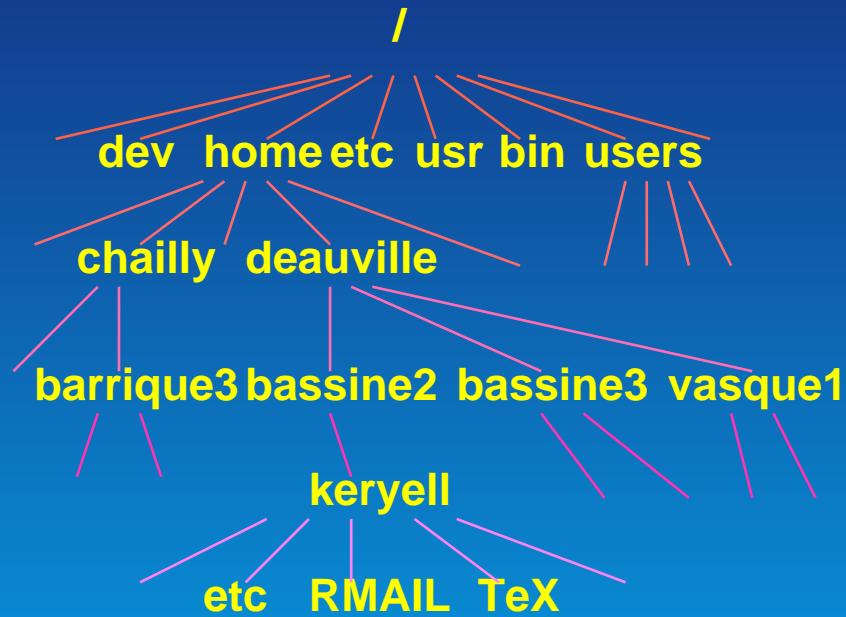
- <http://www.isc.org/sw/bind> : ISC BIND — Internet Software Consortium Berkeley Internet Name Domain. ISC développe aussi un serveur DHCP et un serveur de news (INN)
  - ▶ <http://www.isc.org/sw/bind/arm93/Bv9ARM.pdf>
  - ▶ [http://www.isc.org/index.pl?/sw/bind/arm93 & doc/index.html](http://www.isc.org/index.pl?/sw/bind/arm93&doc/index.html)
  - ▶ <http://www.bind9.net> : info, livres...
- Livre *DNS and BIND*, Cricket Liu & Paul Albitz, O'Reilly, 4th Edition April 2001, 0-596-00158-4, Order Number : 1584, 622 pages, \$44.95 <http://www.oreilly.com/catalog/dns4/>
- Livre *DNS et BIND*, Paul Albitz et Cricket Liu , O'Reilly, 4ème édition janvier 2002, 2-84177-150-4, 486 pages, 49€  
<http://www.oreilly.fr/catalogue/dns-bind-4ed.html>

- Livre *DNS & BIND Cookbook*, Cricket Liu, O'Reilly, October 2002, 0-596-00410-9, 240 pages, \$34.95  
<http://www.oreilly.com/catalog/dnsbindckbk>
- La liste de discussion dns-fr@cru.fr via  
<http://listes.cru.fr/sympa/info/dns-fr>
- news:comp.protocols.dns.bind

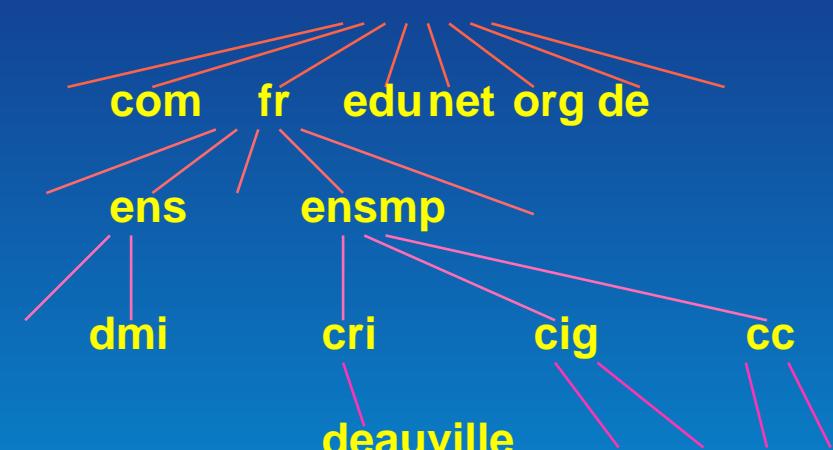
- ❖ Principes du DNS
- Créer son domaine
- Formats des zones
- Comment utiliser le DNS sur un client ?
- Outils
- Description de BIND
- Générer un fichier de zone
- BIND avancé : DNSSEC, IPv6
- Mise en pratique
- Table des matières

- Hiérarchiser les espaces de nommage
- Contrôle local sur son propre morceau
- Robustesse par réPLICATION
- Performances par cache des données
- Communication par mécanisme client/serveur : resolver/serveur de nom

## Système de fichiers UNIX



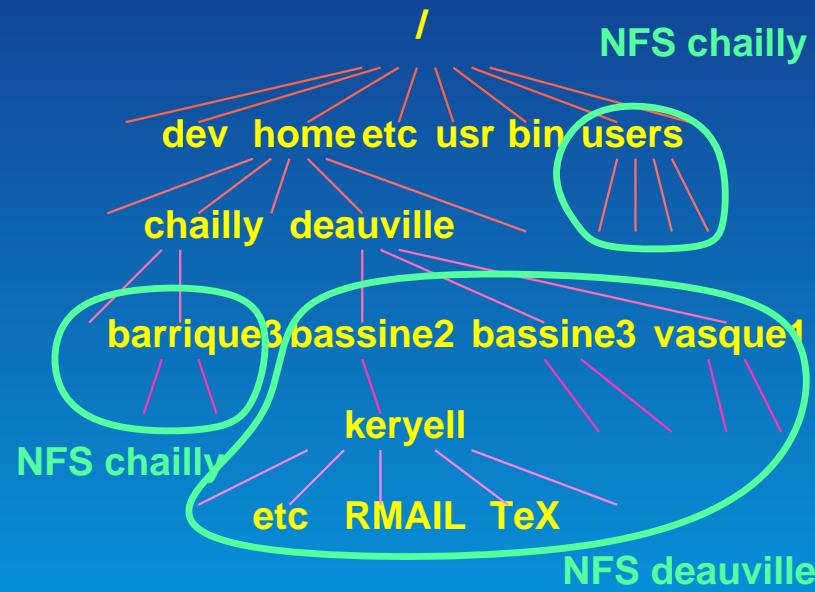
## Hiérarchie du DNS



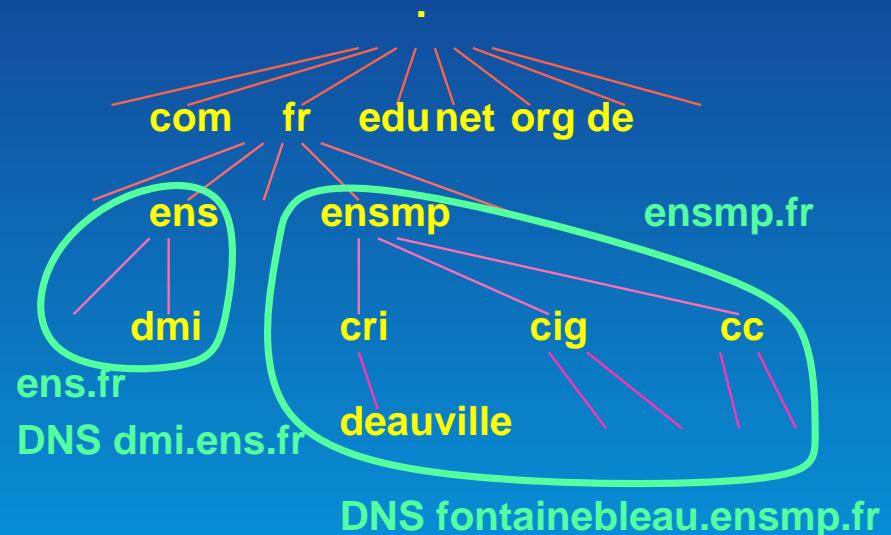
- Nommage unique du chemin même si un sous-chemin ou une feuille sont identiques
- Mécanisme similaire aux liens : les alias (liens symboliques) ou les réplications (liens *hard*)

- Montage d'un disque ou d'un système de fichier sur un serveur distant : reléguer un sous-domaine à un autre serveur DNS
- Montage possible d'un disque depuis plusieurs machines pour résister aux pannes : plusieurs serveurs DNS peuvent servir un sous-domaine

## Système de fichiers UNIX



## Hiérarchie du DNS



- Chaque feuille contient l'information de traduction telles que
  - ▶ Numéro IP
  - ▶ Type de matériel

- ▶ Comment envoyer le mail
- ▶ ...
- Un nœud non feuille représente un domaine mais peut aussi être un nom de domaine de machine
- Hiérarchie logique : pas de relation physique (géographique...) a priori

- « . » : *top-level domain*, domaine racine
- Domaine de premier niveau
  - ▶ com commerciaux
  - ▶ edu organismes d'éducation américains (universités)
  - ▶ gov organismes gouvernementaux USA
  - ▶ mil organismes militaires USA
  - ▶ net organismes de gestion de réseaux
  - ▶ org organismes non-commerciaux
  - ▶ int organismes internationaux
  - ▶ arpa transition ARPAnet—>Internet + traduction inverse
  - ▶ Tendance au flou et au débordement dans les domaines com, net et org...

- ▶ Très américain : historiquement ARPAnet...
- ▶ Nouveau : .biz, .info,... : Cf. <http://www.icann.org>
  - RFC 3675 : « *.sex Considered Dangerous* », Février 2004
- ▶ Domaines par pays quasi-ISO 3166
  - fr
  - de
  - uk (et non gb)
- Domaine de second niveau
  - ▶ com.au, edu.au comme « . »
  - ▶ ensmp.fr : à plat dans .fr
  - ▶ Hiérarchisé dans .fr :
    - gouv.fr gouvernement français

- tm.fr marques déposées en France
- asso.fr associations loi 1901
- ...
- 3<sup>ème</sup> niveau, etc.

RFC 952, RFC 1035 puis RFC 1123

- Noms séparés par des « . ». Taille du tout  $\leq 255$
- Taille de chaque nom  $\leq 63$  caractères ASCII (lettres, chiffres et « - » (pas en première ni dernière position)), premier caractère plus forcément une lettre (depuis le RFC 1123  $\rightsquigarrow$  doit tester d'abord si syntaxe #.#.#.# correcte avant de regarder si c'est un nom...  $\odot$ )
- Égalité entre minuscule et majuscule
- Concaténation d'au plus 127 noms
- Nom absolu (non ambigu...) si terminé par « . » : FQDN *Fully Qualified Domain Name*  
Supposons l'existence de cs.ensmp.fr (Computer Science)  
Qui est cs ?
  - ▶ cs. (Tchécoslovaquie) ?

- ▶ cs.ensmp.fr . (si on est dans le domaine ensmp.fr) ?  
Dernier par défaut
- Domaine ≡ sous-arbre de l'espace de nommage
  - ▶ ⋯.ensmp.fr
  - ▶ ⋯.org
  - ▶ ⋯.enstb.org
  - ▶ ⋯.rire.enstb.org

- Réalise la hiérarchisation administrative
- Sous-domaines gérés par d'autres organismes responsables de leur propres données et de leurs propres délégations
- Le domaine parent n'a que des pointeurs (noms de serveurs) vers les DNS gérant les sous-domaines
- fr. a un pointeur ensmp vers les DNS de l'École des Mines de Paris
- Limite implicite dans le nombre (127) mais pas dans l'enchaînement des délégations

## Point de délégation dans la hiérarchie DNS

- **Zone** ≡ **portion du domaine** gérée effectivement par un serveur
- Un serveur contient les données de son/ses domaine(s) : la/les **zone(s)** : il fait **autorité** en la matière (*authoritative*)
- Un domaine est divisé en plusieurs zone s'il y a **délégation**
- Un serveur **maître primaire** charge les données depuis un fichier local (« base de données »)
- Un serveur (**esclave** ou **maître secondaire**) transfère au démarrage la zone depuis un serveur de référence (serveur maître) qui fait autorité
- Un serveur peut faire autorité sur plusieurs zones

- Programme client ou bibliothèque de résolution (*resolver*)
- Interroge *un* (son) serveur de nom
- Interprète les réponses (numéro IP, erreur,...)
- Renvoie l'information au client (ssh, navigateur, ftp,...)
- Un serveur de nom est *aussi* capable de résoudre les noms en se promenant dans la hiérarchie de serveur pour les domaines dont il n'a pas autorité
- Problème : où commencer la recherche ?  
Dans une liste de serveurs racine (*hint*) !

- Serveur gérant « . » BCP 40
- Pointe vers les serveurs faisant autorité sur les domaines de premier niveau
- A généralement pour des raisons de performance aussi autorité sur les domaines de premier niveau
- Si plus de serveurs racines, tout s'arrête... ↵ 13 serveurs actuellement A.ROOT-SERVERS.NET ... M.ROOT-SERVERS.NET répartis sur la planète
  - ▶ ↵ souvent cible d'attaque de dénis de services (Dos)  
(21-22/10/2002 <http://d.root-servers.org/october21.txt>,  
<http://h.root-servers.org/hroot-month-1002.jpg>)
  - ▶  
<http://dnsmon.ripe.net/dns-servmon/domain/plot?domain=uk&day=0>  
Attaque serveurs UltraDNS du 6/02/2007



<http://www.net4war.com/news4war/infoguerre/root-servers-attaques.htm>

- ↵ Évolue vers une réPLICATION mondiale de chaque serveur racine via anycast BGP... Comment savoir qui nous répond vraiment ?

Exemple sur <http://f.root-servers.org>

```
dig +norec @F.ROOT-SERVERS.NET HOSTNAME.BIND CHAOS TXT  
;; ANSWER SECTION:
```

HOSTNAME.BIND.	0	CH	TXT	"cdg1a.f.root-servers.o
----------------	---	----	-----	-------------------------

traceroute va donner aussi une idée

- Regarder <http://www.root-servers.org>
- Exemple <http://www.isc.org/ops/f-root> était
  - 2 Compaq AlphaServer ES40 avec chacun 4 Alpha à 500 Mhz, 8 Go RAM, BIND 9

- ▶ Hébergé chez PAIX.net, Inc. in Palo Alto, California,  
connexions par plusieurs Fast Ethernet sur fibre
- ▶ En fait répliqué
- Un serveur racine a enfin été mis en service en France (≡Paris ☺) officiellement le 22/12/2003
  - ▶ Coopération ISC, SFINX/Renater, Telehouse, NEC and Foundry Networks
  - ▶ Miroir du serveur F
  - ▶ IPv4 et IPv6
  - ▶ Serveur secondaire de .fr
- Comment initialiser les racines ?
  - ▶ dig . any ☺
  - ▶ ftp://ftp.rs.internic.net/domain ☺
  - ▶ ftp://198.41.0.6/domain

- Demande de résoudre *récursivement* www.amanda.org qui demande au serveur de nom local

Question du serveur de nom local	Serveur de	Réponse
www.amanda.org ?	« . »	Aller voir serveur de org.
www.amanda.org ?	« org. »	Aller voir serveur de amanda.org.
www.amanda.org ?	« amanda.org. »	nom canonique spiderman.amanda.org.
spiderman.amanda.org ?	« amanda.org. »	208.213.83.7

- Si le système de résolution ne demandait pas la récursion, il devrait *itérer* lui-même...
- Économie :
  - ▶ le serveur contacte le serveur *connu* le plus proche dans la hiérarchie au lieu d'un serveur racine

► exemple

- deauville.ensmp.fr demande qui est www.enst.fr
- Demande au serveur de fr . plutôt qu'à un serveur racine (dénis de service ☹)
- Seul le serveur de nom local (ou celui attaché au resolver) fait de la récursion
- Les serveurs racines ne font *plus* de récursion pour des raisons de performances... En plus cela permet de ne pas cacher de l'information et donc d'économiser de la mémoire !
- Si plusieurs serveurs de noms possibles : choix par BIND en fonction du temps d'aller-retour des paquets DNS

- Exemple : serveurs DNS gérant org. délèguent enstb.org aux machines de nom dns2.enstb.org. et dns3.enstb.org.
- Un serveur enquêtant sur enstb.org doit contacter une des 2 machines via protocole DNS ↵ besoin de l'adresse IP qui a besoin d'une requête DNS qui a besoin... 😞
- Problème d'œuf ou la poule ! 😞
- ↵ Idée : déclarer ces noms avec leur adresse IP dans org. directement ≡ la glu

Exemple : regarder qui répond dans

```
dig dns2.enstb.org +trace
```

- Délégation de zone à des serveurs
- Que se passe-t-il si un serveur qui doit faire autorité ne sait rien sur sa zone ?
- Serveur à la rue... 😞
- Domaine qui ne fonctionnera pas si on a le malheur d'interroger ce serveur
- ∃ moteurs de recherche des *lame servers* qui envoient automatiquement des courriels aux (ir)responsables des zones concernées
- Penser à vérifier ses zones, par exemple avec
  - ▶ <http://zonecheck.fr> v2
  - ▶ <http://www.dnsreport.com>
- En général dans les journaux de fonctionnement son propre DNS note les problèmes rencontrés :

```
Nov 30 17:41:49 minou named[9310]: lame server resolving 'ns.eu.org'  
(in 'EU.org'?): 80.67.173.21#53  
Nov 30 17:41:51 minou named[9310]: lame server resolving 'dns0.mmu.ac.UK'  
(in 'mmu.ac.UK'?): 149.170.39.92#53  
Nov 30 17:41:51 minou named[9310]: lame server resolving 'dns1.strath.ac.UK'  
(in 'strath.ac.UK'?): 150.237.128.10#53
```

- Utilisation du port IP/UDP/53
- Utilisation du port IP/TCP/53 sur versions modernes
- IPv4 et IPv6 (orthogonallement aux requêtes...)
- Utilisé à la fois entre client-serveur et serveur-serveur
- Si sécurité par pare-feux, s'arranger pour laisser passer les requêtes DNS vers son serveur et réponses depuis son serveur sur port 53 en UDP et TCP
- Paquets estampillés d'un identifiant
- Protocole de base peu sécurisé (cf `dnsspoof` de <http://naughty.monkey.org/~dugsong/dsniff/>) si pas de signature cryptographique (TSIG)
  - ▶ Deviner le port UDP de la requête
  - ▶ Deviner l'identifiant de la requête

- Se restreindre au moins à TCP : deviner en plus le numéro de séquence du paquet TCP

- Besoins
  - ▶ Autorisation d'une connexion `rlogin` à partir d'un nom dans son `.rhosts`
  - ▶ Messages de debug plus humains
  - ▶ ...
- Comment avoir la traduction inverse des mécanismes précédents ?
- Traduction nom vers adresse implémentée et efficace dans le DNS
- Comment adapter le système existant ?

- Idée : découper une adresse IP de 32 bits en 4 paquets de 8 bits codés en décimal  $x.y.z.t$  et interpréter *textuellement*
- Créer un domaine spécifique `in-addr.arpa.` pour la traduction inverse
- Faire une recherche `x.y.z.t.in-addr.arpa.?`
  - ▶ `deauville.ensmp.fr`  $\equiv$  192.54.172.242
  - ▶ 192.54.172.242.`in-addr.arpa.?`
  - ▶ Réseaux : hiérarchisé par le poids fort (sous-réseaux...) : 1  
réseau des Mines : 192.54.172.\*
  - ▶ Tous les noms de machines seraient dans  
192.54.172.\*.`in-addr.arpa.?`
  - ▶ Mauvais ! Il faudrait être responsable du haut de la  
hiérarchie... 😞
- Idée : inverser les nombres de l'adresse

- ▶ Recherche de *t.z.y.x.in-addr.arpa.*
- ▶ 242.172.54.192.in-addr.arpa. ?
- ▶ Mines responsable de \*.172.54.192.in-addr.arpa. bas de la hiérarchie
- <http://www.ipindex.net/>  
<http://blues.eurovia.es/mirrors/www.ipindex.net> donne une idée de l'usage des adresses IP
- Services de géoréférence : <http://www.maxmind.com>
- Comment faire du CIDR alors que tout est fait pour des classes A, B ou C ? Cf. plus loin...

Comment faire une recherche arbitraire (autre que sur une adresse IP) ?

- Mécanisme de requête inverse à un serveur
- Répond s'il connaît ou pas
- Pas de récursion (sémantique ?)
- Au demandeur d'essayer tous les serveurs du monde...
- Bug ↗ trou de sécurité dans vieilles versions de BIND
- Plus implémenté dans les dernières versions de BIND

- Problème : goulet d'étranglement aux serveurs racines à chaque requête ☹
- Idée : mécanisme de cache ☺
- Retenir dans une mémoire les traductions les plus souvent demandées
- Retenir aussi le fait que certaines traductions n'existent pas (erreurs répétitives...) : cachage négatif
- Retenir toutes les informations aperçues lors des recherches récursives : cela peut servir plus tard (liste de serveurs faisant autorité, leurs numéros IP...)
  - ▶ ↗  Attaques par pollution si on accepte *toute* information sans vérification si serveur fait autorité ☹
  - ▶ Comment vérifier sans DNSSEC avec adresse source UDP triviale à usurper ? ☹

- Permet de survivre un peu mieux à une coupure réseau ou à des pannes de DNS distants...
- Comment propager les mises à jour des données aux caches des serveurs ? Trop de caches de serveurs et pas de mécanisme hiérarchique de diffusion
- Rajout d'une durée de vie (TTL *Time To Live*) associé à une zone choisie par l'administrateur
- Si la durée de vie d'une donnée est dépassée, la donnée est effacée du cache
- Compromis à trouver sur le TTL
  - ▶ Bonnes performances ↗ TTL ↗
  - ▶ Propagation des mises à jour rapides ↗ TTL ↘
- TTL pour les réponses négatives fixé à 10 minutes par défaut

- Même avec les caches les serveurs racines reçoivent des milliers de requêtes par seconde... mais on y survit !

- Principes du DNS
- ❖ Créer son domaine
- Formats des zones
- Comment utiliser le DNS sur un client ?
- Outils
- Description de BIND
- Générer un fichier de zone
- BIND avancé : DNSSEC, IPv6
- Mise en pratique
- Table des matières

- Choisir un nom
- Vérifier qu'il n'existe pas déjà...
- Contacter le responsable du domaine père
- Problème intrinsèque : interaction entre le nom de domaine et le nom du domaine père (nouveau nom de domaine concaténé au nom du père)
- Mettre en place un DNS et le déclarer auprès du responsable du domaine père ou faire héberger tout le système de DNS

- Nom dans un domaine de la racine (com., org., net.,...)
- Nom dans une hiérarchie d'un pays : étudier la hiérarchie du pays
- Problème (esthétique et commercial) du choix d'un nom : il va durer longtemps mais souvent on le choisit sans expérience...  
Consulter des personnes compétentes.

Exemples :

- ▶ airliquide.fr mais air-liquide.fr mieux
  - ▶ edfgdf.fr mais edf-gdf.fr mieux
  - ▶ Question de goût... Possible d'avoir plusieurs noms...
- Synonymes
  - Permet de récupérer des fautes de frappe ! Avant qu'un site pirate le fasse... ☹

- Vérifier que le nom n'existe pas (cyber-racket... ☺) !
- Un bon début <http://www.internic.net>
- Mais éclatement du services ↗ autres domaines que ceux de l'InterNIC : <http://www.internic.net/help/other-reg.html>, la France <http://www.nic.fr/>
  - ▶ *registry* : organisme gérant une zone de niveau supérieur (TLD)
  - ▶ *registrar* : organisme allouant les noms aux clients et les enregistrant auprès du *registry*
- Question de coût
  - ▶ Tous les noms de 1 à 4 lettres dans les domaines classiques ont déjà été enregistrés !
  - ▶ Un nom *xy.org* ≈ 50 000 \$

- ▶ Un nom *xy.com*  $\approx 200\,000 \$$
- ▶ Même être sur la liste d'attente d'un nom se monnaie

- Le DNS contient des informations sur des machines
- Où stocker les données administratives permettant de facturer, d'initialiser les domaines de haut niveau,... ?
- ↵ Base administrative « *who is who ?* »
- Base de données accessible à distance avec le protocol whois
- Avec morcellement gestion internet ↵ nombreux sites whois, difficile de trouver l'information ☺ :

```
whois -h whois.internic.net ...
```

```
whois -h whois.networksolutions.com ...
```

```
whois -h whois.crsnic.net ...
```

```
whois -h whois.ripe.net ...
```

```
whois -h whois.arin.net ...
```

```
whois -h whois.gandi.net keryell
```

```
person:      Ronan Keryell
```

```
nic-hdl:     RK72-GANDI
```

address: ENST Bretagne - LIT  
address: BP832  
address: 29285  
address: PLOUZANE  
address: France  
phone: +33 2 98 00 14 15  
fax: +33 2 98 00 12 82  
e-mail: Ronan.Keryell@enst-bretagne.fr

- Méta-whois <http://www.allwhois.com/>,  
<http://www.geektools.com/cgi-bin/proxy.cgi>
- Personnellement j'ai une commande whoisnet :  
#! /bin/sh  
whois -h whois.internic.net \$1  
whois -h whois.networksolutions.com \$1  
whois -h whois.crsnic.net \$1  
whois -h whois.ripe.net \$1

```
whois -h whois.arin.net $1
```

```
whois -h whois.gandi.net $1
```

- whoisnet enstb.org

Domain Name: ENSTB.ORG

Registrar: GANDI

Whois Server: whois.gandi.net

Referral URL: <http://www.gandi.net>

Name Server: DNS-CRI.ENSMP.FR

Name Server: DNS2.ENSTB.ORG

Name Server: DNS3.ENSTB.ORG

Updated Date: 04-nov-2002

Mais maintenant les whois modernes font ça tout seul

- ∃ aussi des systèmes libres pour gérer son propre registrar !

<http://isc.org/index.pl?/products/OpenReg/>

- Consulter son fournisseur Internet
- Si propre serveur de nom vérifier que le réseau est bien enregistré aussi avec tous les whois
- S'enregistrer auprès du domaine père
- Vérifier que cela fonctionne avec <http://zonecheck.fr>  
Prendre néanmoins les avis de ce programme avec des pincettes...
- ∃ noms de domaines gratuits. Exemple : <http://www.eu.org>.  
Aussi à la base d'enregistreurs professionnels ([www.gandi.net](http://www.gandi.net))

- Principes du DNS
- Créer son domaine
- ↳ Formats des zones
- Comment utiliser le DNS sur un client ?
- Outils
- Description de BIND
- Générer un fichier de zone
- BIND avancé : DNSSEC, IPv6
- Mise en pratique
- Table des matières

## RFC 1035

- Définit une représentation textuelle aux paquets binaires du protocole (requête dans outils, réponses, fichiers de données,...)
- CLASS définit la classe d'utilisation de l'enregistrement (DNS plus large qu'Internet)
  - ▶ IN Internet
  - ▶ CS CSNET (obsolète)
  - ▶ CH CHAOS (MIT)
  - ▶ HS Hesiod
- RR (Resource Record) contient un enregistrement d'information de différents types avec une durée de vie optionnelle en seconde (32 bits  $\rightsquigarrow$  68 ans ☺) :

- ▶ A adresse de machine  
www IN A 192.54.172.231 Une machine avec plusieurs  
adresse a plusieurs A RR
- ▶ NS nom d'un serveur de nom faisant autorité pour délégation  
trad IN NS chailly.ensmp.fr.
- ▶ CNAME définit un alias qui a pour *nom canonique*...  
www.kazimodal CNAME kazimodal
  - Un alias ne doit pas être utilisé en partie droite d'un RR afin de limiter les récursions
  - Un alias ne peut pas avoir d'autre entrée pour éviter toute schizophrénie du DNS  
*Exception pour les signatures du DNSSEC. Cf. partie DNSSEC.*
- ▶ SOA *start of a zone of authority*

- *MNAME* nom du serveur de référence (maître)
- *RNAME* nom de domaine encodant l'adresse mail en destinataire.un.domaine interprété en destinataire@un.domaine pour Internet
- *SERIAL* numéro de série sur 32 bits monotone croissant modulo 32 bits utilisé pour avertir des mises à jour. Cf transparent à venir
- *REFRESH* temps en seconde (32 bits) avant un rafraîchissement de zone
- *RETRY* temps en seconde (32 bits) avant un autre essai de rafraîchissement de zone qui a échoué
- *EXPIRE* temps en seconde (32 bits) avant que la zone ne soit plus considérée comme faisant autorité

- *MINIMUM* durée de vie (TTL) (32 bits) en seconde des données cachées négativement RFC 2308

```
@ IN SOA chailly.ensmp.fr. keryell.chailly.ensmp.fr. (
    98120200          ; Serial
    21600      ; Refresh 6 hour (nic.fr said...)
    3600       ; Retry   1 hour
    3600000   ; Expire   1000 hours
    600        ; Cachage négatif 10 minutes
```

- ▶ WKS déclare les *well known service* fournis
- ▶ PTR pointeur dans nom de domaine, typiquement pour la traduction adresse vers nom dans `in-addr.arpa`  
`1 IN PTR routeur-10.4.0.ensmp.fr.`
- ▶ HINFO informations sur le CPU et l'OS de la machine

- ▶ MX définit la machine vers qui doit être envoyé le courrier avec un ordre de priorité décroissant (0 ≡ en premier)
- ▶ TXT de l'information textuelle quelconque

Des tonnes d'extensions expérimentales dans

- ▶ RFC 1183 : DCE, nom du responsable, X25, ISDN (Numéris)
- ▶ RFC 1664 : X400
- ▶ ...

- Numéro de série dans le RR SOA permet de détecter mises à jour
- Ne pas oublier d'augmenter ce nombre sur un maître après chaque modification si on veut que les modifications soient propagées... Bug classique 😞
- Pas stocké en précision infinie mais sur 32 bits... Problèmes de débordement ↗ Calcul modulo  $2^{32}$  basé sur le RFC 1982
- Problème de relation d'ordre non définie modulo 😞

$$1 < 4294967295 \tag{1}$$

$$1 > 4294967295 \tag{2}$$

car  $4294967295 = (2^{32} - 1) \equiv -1 \pmod{2^{32}}$

- Idée : privilégier la comparaison des valeurs plus proches, donc plus probables  $1 \succ 4294967295$

- Pour  $(s_1, s_2) \in \mathbb{S}^2$

$$s_1 \prec s_2 \iff (0 < s_2 - s_1 < 2^{31}) \vee (0 < s_2 - (s_1 - 2^{32}) < 2^{31}) \quad (3)$$

$$s_1 \succ s_2 \iff (0 < s_1 - s_2 < 2^{31}) \vee (0 < s_1 - (s_2 - 2^{32}) < 2^{31}) \quad (4)$$

Si  $s_2 \equiv s_1 + 2^{31} \pmod{2^{32}}$ , pas comparable...

- Dans temps anciens, valeur 0 spéciale de resynchronisation : propage systématiquement aux serveurs secondaires... Du coup éviter 0 comme valeur normale

- Mais comment faire pour remettre tous serveurs secondaires en phase lorsqu'on change de plan de numérotation de série de SOA d'un vieux  $v$  à un nouveau  $n$  ?
  - ▶ Si  $n > v$  pas de problème : serveurs secondaires verront qu'une nouvelle version doit être chargée
  - ▶ Sinon, problème : serveurs secondaires verront une nouvelle version plus vieille ou incomparable, donc ignorée... ☹
    - Faire migration par étape
    - Aller à  $v' \equiv v + 2^{31} - 1 \pmod{2^{32}}$  qui est le plus grand successeur de  $v$
    - Attendre propagation vers secondaires
    - $v := v'$
    - Reboucler

- Procédé mnémotechnique dans vie courante
  - ▶ Stocker la date dans le numéro de série
  - ▶ Exemple 2007100300
  - ▶  Ne pas oublier de garantir croissance stricte si mise à jour, même si modifications à suivre...

## RFC 1035 section 5

- Utile pour lire le contenu des caches
- Sert à définir une zone dans BIND
- Orienté ligne
- mais (...) permet d'écrire un enregistrement sur plusieurs lignes
- ; commentaire
- @ représente l'origine courante de la zone. Utilisée dans les noms relatifs (ne terminant pas par « . ») ≈ répertoire courant (« . ») dans un système de fichiers
- \$ORIGIN *domain-name* [*comment*] change l'origine courante de la zone ≈ cd sous Unix
- \$INCLUDE *file-name* [*domain-name*] [*comment*] inclut un fichier et définit son (et seulement son) origine courante

- Un enregistrement a la forme
  - ▶ *domain-name rr [comment]*
  - ▶ *<blank> rr [comment]* pour ajouter l'enregistrement au nom de domaine précédent
  - ▶ *rr* ont la forme
    - *[TTL] [class] type RDATA*
    - *[class] [TTL] type RDATA*

Les *TTL* et *class* manquants prennent les mêmes valeurs que celles des enregistrements précédents

- Certains enregistrements peuvent avoir plusieurs valeurs
- Exemple de routeurs ou de répartition de charge : plusieurs RR A paramétrable en :

- Réponse tourniquet (répartit la charge sur plusieurs serveurs)
  - Réponse la plus proche en terme de réseau (diminue la pression réseau)
  - ▶ \$TTL *TTL* fixe le TTL par défaut pour les enregistrements suivants
  - Pour le cache (accessible par `rndc dumpdb`) il faut aussi noter la non existence
- |               |               |      |              |
|---------------|---------------|------|--------------|
| SPOOL.MU.EDU. | 8284          | AAAA | ; -\$NXRRSET |
| ;             | authauthority |      |              |
|               | 8284          | A6   | ; -\$NXRRSET |
| ;             | authanswer    |      |              |

- À quelle machine envoyer du mail ?
- Idée : envoyer à la machine destinataire en utilisant l'adresse trouvée dans le RR de type A (adresse) du DNS
- Problèmes
  - ▶ La machine est derrière un firewall
  - ▶ La machine est dans un réseau non routable RFC 1918 (privé  $10.x.y.z, \dots$ )
  - ▶ La machine n'existe pas ! Concept : `ens.fr` (essayer `host ens.fr`), adresse purement mail,...
- Idée : rajouter des champs RR MX dans le DNS précisant les échangeurs de mail à utiliser et avec quelle priorité RFC 974
  - ▶ Permet de définir des passerelles vers des domaines pas sur Internet (Bitnet, uucp,...)

- ▶ Centraliser la réception des messages sur une machine qui a un système plus robuste, mieux protégée contre les *spams*, ou seul accès via coupe-feu
- ▶ Mettre des adresses conceptuelles pur mail (sociétés n'ayant qu'une adresse de mail commercial)
- ▶ Répartir la charge sur des points d'entrée d'un réseau privé
- ▶ Donner des sites de secours

```
host iar2m.ensmp.fr
iar2m.ensmp.fr has address 10.2.16.200
iar2m.ensmp.fr mail is handled (pri=1) by cri.ensmp.fr
iar2m.ensmp.fr mail is handled (pri=2) by fontainebleau.ensmp.fr
iar2m.ensmp.fr mail is handled (pri=3) by paris.ensmp.fr
iar2m.ensmp.fr mail is handled (pri=4) by baloo.ensmp.fr
```

utilise d'abord les MX de plus petite priorité. En cas de problème les sites de secours avec de gros disques récupèrent le mail de manière plus sûre et plus longtemps

que s'il restaient en attente sur le site de départ. En plus on peut récupérer le courrier par un autre moyen (bandes,...)

- ▶ La machine MX principal doit savoir si nécessaire router les messages en interne *sans consulter les RR MX...*
- ▶ Les RR MX doivent contenir de vrais noms canoniques et non pas des alias même si cela fonctionne en général (nivellement par le bas...)
- ▶ Possibilité d'avoir des *wildcard MX* pour définir une entrée courrier pour tout un domaine

\*.obscure.truc IN MX 1 mine.net

De plus en plus de messages indésirables... ☺

- Mettre des filtres au niveau de la lecture du courrier
- Mettre des filtres au niveau de la distribution finale du message (style procmail)
- Mettre des filtres au niveau de sendmail (ou autre facteur postal)
  - ▶ Éliminer les messages avec des adresses d'envoyeur folkloriques (domaines qui n'existent pas...)
  - ▶ Éliminer les connexions depuis des machines sans nom déclaré dans le DNS
    - ⚠ Dans certains pays Internet est si mal configuré qu'il n'y a pas de traduction inverse... ☺
  - ▶ Éviter de servir de nœud de transit de spam
  - ▶ Éliminer les mails en provenance de serveurs bien connus comme étant peu fréquentables (liste noire, RBL)

- Difficile de trouver les vrais auteurs du spam : examiner *tous* les entêtes mais certains sont faux ☹, des sites intermédiaires mal configurés peuvent servir de relais et cacher l'identité de l'auteur
- Souvent machines piratées pour envoyer massivement du spam : beaucoup de courriels pour peu de ressources locales
  - ▶ En cas d'échec d'émission pas de ré-essai
  - ▶ ↗ GREYLISTING : fait exprès de refuser courriels la première fois depuis une source louche
  - ▶ Seuls les vrais échangeurs de mails auront courage d'essayer à nouveau plus tard
- Méta-spam : envoyer des mails avec comme fausse adresse d'origine un site tiers ↗ ce site va recevoir les messages d'erreur et les râlantes des incompétents... ☹
- À lire

<http://www.halte-au-spam.com>

<ftp://ftp.univ-lyon1.fr/pub/faq/by-name/fr/usenet/abus/reagir-general>  
<ftp://ftp.univ-lyon1.fr/pub/faq/by-name/fr/usenet/abus/reagir-conseils>

- Aller vers du courrier électronique certifié ?



Comme pour le courrier papier, pas de bonne solution... ☺

- Système de DNS ≡ une base de données hiérarchisé
- Perversions possibles : `in-addr.arpa`
- Idée : utiliser le DNS pour stocker les numéros IP de machines faisant du *spam* de mails. Mise à jour en temps réel de la MAPS RBL (*Mail Abuse Protection System - Realtime Blackhole List*)
- La RBL est mise à jour par des gens qui se plaignent d'abus
- Les systèmes d'échange de mail font une vérification dans le domaine `rbl.maps.vix.com` de la même manière que dans `in-addr.arpa`. Si le numéro IP est trouvé c'est que la machine a été déclarée polluante
-  Problème si un jour on se trouve nommé dans la RBL...
- ↵ Motivation pour avoir un système de courrier bien géré (éviter le relais inutile par exemple)
- <http://www.mail-abuse.org> par exemple

- Développement du *spam* ↵ travail considérable ↵ ∃ de plus en plus de services payants de ce type
- Utilisation dans enstb.org dans le .mc de sendmail :  
dnl AntiSPAM stuff:  
FEATURE('dnsbl', 'list.dsbl.org', '"550 Email rejected due to sending s'  
FEATURE('dnsbl', 'sbl.spamhaus.org', '"550 Email rejected due to sendin'  
FEATURE('dnsbl', 'will-spam-for-food.eu.org')dnl
- Pour faire une recherche multiliste : <http://www.robtex.com/rbl>
- Procès de *spammers* contre ces systèmes... ☹
- Requête DNS à chaque message reçu ↵ ralentissement du débit du courrier
- Optimisation : devenir serveur secondaire de ces zones (cf forwarders). En général payant

<http://www.openspf.org>

- Idée : déclarer quelles sont les machines autorisées à envoyer *son* courrier
- Évite usurpation d'adresses MAIL FROM par des spameurs
- Information rajoutée dans le DNS du MAIL FROM
- Puisqu'un spameur n'a pas de contrôle du DNS du domaine à usurper, il ne peut pas modifier l'enregistrement SPF
- <http://old.openspf.org/wizard.html> Synthèse en ligne de paramètres SPF

SPFv1 décrit par RFC 4408

- Rajouté dans RR TXT du MAIL FROM pour ne pas avoir à créer nouvel RR mais ∃ aussi RR SPF
- Utilisateurs d'example.net
  - ▶ Envoient du mail depuis leurs MX
  - ▶ Envoient du mail depuis pluto.example.net
  - ▶ Envoient du mail depuis gmail.com qui possède ses propres paramètres
  - ▶ Toute autre machine est illégitime
- dig gmail.com txt donne

```
gmail.com. 300 IN TXT "v=spf1 redirect=_spf.google.com"
```

Puis un dig \_spf.google.com txt mène à

```
_spf.google.com. 161 IN TXT "v=spf1  
ip4:216.239.32.0/19 ip4:64.233.160.0/19  
ip4:66.249.80.0/20 ip4:72.14.192.0/18  
ip4:209.85.128.0/17 ip4:66.102.0.0/20  
ip4:74.125.0.0/16 ?all"
```

- Qualifieurs
  - ▶ + : passe
  - ▶ - : échoue
  - ▶ ~ : « *SoftFail* » pas un rejet franc mais contribue à penser que c'est mauvais, à combiner avec autre chose
  - ▶ ? : neutre
- Un enregistrement se termine par un all ou redirect pour aller voir ailleurs

- Si le DNS s'arrête, la majorité des applications d'Internet s'arrêtent ! 😞
- Cible favorite des dénis de service (DoS) des pirates
- ↵ Besoin d'une bonne stratégie
- Tolérance aux pannes : plusieurs serveurs si possible sur différents réseaux sur différents fournisseurs primaires
- Définir plusieurs maîtres primaires avec système de synchronisation des fichiers de zones pour survivre à une perte de visibilité des serveurs faisant autorité
- Il est possible de déclarer plus de serveurs de nom dans sa zone que les délégations déclarées dans le serveur du domaine parent pour augmenter la distribution des requêtes locales (*stealth servers*) ou si DMZ,...
- En particulier le vrai serveur primaire n'est pas forcément public et annoncé

- Utilisation d'anycast BGP pour distribuer sur Internet des machines répondant à la même adresse : utilisé de manière intensive pour les serveurs racine

- ∃ des dénis de service distribué lancés depuis des machines attaquant un service sur une machine donnée
- Exemple du virus W32/MyDoom.B@MM
  - ▶ Attaque du serveur <http://www.sco.com>
  - ▶ Réponse en fermant les connexions TCP/80
  - ▶ Avant l'attaque TTL de sco.com passé à 60 secondes pour changer rapidement www.sco.com vers un autre adresse IP (127.0.0.1 par exemple ☺)
  - ▶ [http://news.netcraft.com/archives/2004/02/01/sunday\\_morning\\_and\\_wwws.html](http://news.netcraft.com/archives/2004/02/01/sunday_morning_and_wwws.html)
  - ▶ Utilisation du nom <http://www.thescogroup.com> possible pendant l'attaque  
[http://news.netcraft.com/archives/2004/02/02/sco\\_to\\_use\\_new\\_domain\\_for\\_email.html](http://news.netcraft.com/archives/2004/02/02/sco_to_use_new_domain_for_email.html)
- W32/MyDoom.B@MM cible <http://www.microsoft.com>

- ▶ Distribution mondiale décentralisée des serveurs via Akamai
  - ~~> bonne résistance à la base
- ▶ TTL passé à 60 secondes aussi au cas où pour réagir vite...
- ▶

[http://news.netcraft.com/archives/2004/02/02/microsoft\\_shorten\\_wwwmi](http://news.netcraft.com/archives/2004/02/02/microsoft_shorten_wwwmi)

- Autre besoin d'un TTL faible : planification d'un changement de nommage pour limiter la durée d'incohérence

- Protocole simple au dessus d'UDP facile à tromper
-  Si fausse adresse source dans question, serveur répond à cette fausse adresse
- ↵ Utilisation par un pirate pour télécommander serveurs vers une victime
- Pour être « efficace », il faut des serveurs avec gros enregistrements (AXFR, clés DNSSEC...) à répondre à une petite question (facteur amplificateur pour pirate)
- ↵ Utiliser des serveurs récursifs qui vont répondre à des questions dont ils vont chercher les réponses. Le pirate peut choisir des données arbitrairement grandes
- Exploitation d'un service gentil (récursion pour tous pour faire du debug...) pour faire du mal 😞
- ↵ Suppression de récursion pour tous, réservée aux machines de son réseau

- De manière générale, les DDoS sont difficile à gérer... 😞

- Le RFC 1918 définit la notion d'adresses privées
  - ▶ 10.0.0.0/8 :  $2^{24}$  adresses
  - ▶ 172.16.0.0/12 :  $2^{20}$  adresses
  - ▶ 192.168.0.0/16 :  $2^{16}$  adresses
- Utile pour faire de gros Intranet si pas assez d'adresses publiques,...
- La gestion des requêtes DNS inverses *devraient être gérées* par l'entité utilisatrice (RFC 1912)
- Sinon bombardement des serveurs racines vers des domaines inverses qui n'existent pas ! 😞
- ↵ Projet AS112 (<http://as112.net>)
  - ▶ Idée : répliquer les serveurs DNS qui répondent à ces adresses

- ▶ Utiliser anycast de BGP pour annoncer ces machines (...sur AS112)
- ▶ dig @prisoner.iana.org hostname.as112.net any permet de savoir qui répond aux requêtes pour l'AS112
- ▶ Pour info, 204.152.184.191 est le *stealth* serveur primaire
- ▶ Pour faire son propre serveur « AS112 » dans son entreprise :  
<http://www.chagreslabs.net/jmbrown/research/as112>
- ▶ 192.175.48.1 est le SOA (bombardé de demandes de mises à jour Windows par exemple)
- ▶ 192.175.48.6 et 192.175.48.42 sont les NS (bombardé de requêtes)

- Mécanisme de résolution inverse très orienté classes A, B et C ☹
- Clients veulent gérer leur DNS ☺
- Comment faire pour déléguer la résolution DNS chez le client avec le CIDR qui permet d'attribuer des réseaux aussi petits que /31 ?
- Idée : mettre des alias pour chaque entrée dans la zone C vers le DNS du client BCP 20, RFC 2317
- Exemple de minou.info.enstb.org (193.50.97.146) dans 193.50.97.128/25 (FR-ENSTB-BREST)  
146.97.50.193.in-addr.arpa pointe vers  
146.FR-ENSTB-BREST.97.50.193.in-addr.arpa.  
FR-ENSTB-BREST.97.50.193.in-addr.arpa. est géré par une machine dans enstb.org qui fera la traduction pour les 128 adresses



⚠ suppose que les outils acceptent un CNAME en pleine récursion vers un PTR...

```
chailly99-keryell > dig -x 193.50.97.146 +trace
```

.	476680	IN	NS	G.ROOT-SERVERS.NET.
.	476680	IN	NS	H.ROOT-SERVERS.NET.
.	476680	IN	NS	I.ROOT-SERVERS.NET.
.	476680	IN	NS	J.ROOT-SERVERS.NET.
.	476680	IN	NS	K.ROOT-SERVERS.NET.
.	476680	IN	NS	L.ROOT-SERVERS.NET.
.	476680	IN	NS	M.ROOT-SERVERS.NET.
.	476680	IN	NS	A.ROOT-SERVERS.NET.
.	476680	IN	NS	B.ROOT-SERVERS.NET.
.	476680	IN	NS	C.ROOT-SERVERS.NET.
.	476680	IN	NS	D.ROOT-SERVERS.NET.
.	476680	IN	NS	E.ROOT-SERVERS.NET.
.	476680	IN	NS	F.ROOT-SERVERS.NET.

; ; Received 244 bytes from 193.48.171.215#53(193.48.171.215) in 5 ms

193.in-addr.arpa.	86400	IN	NS	NS.RIPE.NET.
193.in-addr.arpa.	86400	IN	NS	AUTH03.NS.UU.NET.
193.in-addr.arpa.	86400	IN	NS	NS2.NIC.FR.
193.in-addr.arpa.	86400	IN	NS	SUNIC.SUNET.SE.
193.in-addr.arpa.	86400	IN	NS	MUNNARI.OZ.AU.
193.in-addr.arpa.	86400	IN	NS	NS.APNIC.NET.

; ; Received 294 bytes from 192.112.36.4#53(G.ROOT-SERVERS.NET) in 298 ms

97.50.193.in-addr.arpa.	86400	IN	NS	ns1.renater.fr.
97.50.193.in-addr.arpa.	86400	IN	NS	ns3.nic.fr.

; ; Received 138 bytes from 193.0.0.193#53(NS.RIPE.NET) in 50 ms

146.97.50.193.in-addr.arpa.	86400	IN	CNAME	146.FR-ENSTB-BREST.97.50.193.in-addr.arpa.
	86400	IN	NS	dns-cri.ensmp.fr.

```
FR-ENSTB-BREST.97.50.193.in-addr.arpa. 86400 IN NS dns2.enstb.org.  
;; Received 135 bytes from 193.49.160.100#53(ns1.renater.fr) in 54 ms  
  
chailly99-keryell > dig 146.FR-ENSTB-BREST.97.50.193.in-addr.arpa. ptr  
;; ANSWER SECTION:  
146.FR-ENSTB-BREST.97.50.193.in-addr.arpa. 3600 IN PTR minou.info.enstb.org
```

- Souvent cachée dans `version.bind` CH TXT
- Interrogeable par

```
dig @dns-cri.ensmp.fr version.bind txt chaos
;; ANSWER SECTION:
version.bind.          0           CH         TXT      "9.2.3rc4"
```

- Changeable avec l'option `version` de BIND
- Restreignable par `view`

```
view "chaos" chaos {
    match-clients { <those to be refused>; };
    allow-query { none; };
    zone "." {
        type hint;
        file "/dev/null"; // or any empty file
    };
};
```

- Besoin de localiser adresses de serveurs NIS, WINS,...
  - ▶ Câblé en dur 😞
  - ▶ Requête diffusée sur le réseau « qui a le ... ? »
  - ▶ Interroger une base de données distribuée
- Idée : généralisation des concepts comme le MX sans avoir à rajouter encore un RR
  - ↝ Extension du DNS avec <http://www.ietf.org/rfc/rfc2782.txt>
  - A DNS RR for specifying the location of services (DNS SRV)*
  - ▶ Rajout de pseudo-noms de domaine commençant par des « \_ » (éviter collisions de noms) :  
*\_service.\_proto.name*
  - ▶ Avec classiques *TTL* et *class*
  - ▶ *Priority*  $\in [0,65535]$ . On choisira toujours la réponse à priorité la plus basse (cf MX)

- ▶  $Weight \in [0,65535]$ . Idem pour départager des priorités égales
- ▶  $Port \in [0,65535]$  à contacter pour le service
- ▶  $Target$  à contacter pour le service

Syntaxe d'un RR SRV :

*\_Service.\_Proto.Name TTL Class SRV Priority Weight Port Target*

Exemple pour connaître le serveur LDAP de example.com, faire une requête :

*\_ldap.\_tcp.example.com*

- Utilisation du DNS comme base de données distribuée avec les *SRV RR*
- <http://www.microsoft.com/techNet/win2000/win2ksrv/technote/w2kdns.asp>  
*Windows 2000 DNS - A white paper*
- <http://www.nominum.com/resources/faqs/bind-faq.html#w2k> FAQ  
*Microsoft Windows 2000 and BIND*
- Windows 2000 stocke données Active Directory dans une sous-zone `_msdcs`, en particulier le catalogue global (style `gc._msdcs.example.com`)

- Configuration typique de BIND pour accepter les mises à jour par les serveurs Active Directory :

```
zone "_msdcs.example.com"
{
    type master;
    file "_msdcs.example.db";
    check-names ignore;
    // Fait confiance aux machines locales en attendant que Windows
    // et BIND se mettent d'accord sur une authentification :
    allow-update { localnets; };
};
```

- DNS ≡ base de données répartie
- Naming Authority Pointer (NAPTR)
- Associer des informations à un n° de téléphone
- Domaine e164.arpa utilisé comme in-addr.arpa chiffre par chiffre à l'envers :
  - ▶ Définit les sip: et smtp: d'un numéro et renvoi téléphonique (RFC 2806) :

```
$ORIGIN 2.1.2.1.5.5.0.7.7.1.e164.arpa.  
;;          order pref flags service           regexp replacement  
IN NAPTR 100 10 "u" "sip+E2U"   "!^.*$!sip:information@foo.se!i"  
IN NAPTR 102 10 "u" "smtp+E2U"  "!^.*$!mailto:information@foo.se!i"  
IN NAPTR 102 10 "u" "tel+E2U"    "!^.*$!tel:+4689761234!" .
```

- ▶ Tous les infos concernant un domaine de numéro sont dans une base ldap :

```
$ORIGIN 6.4.e164.arpa.
```

```
* IN NAPTR 100 10 "u" "ldap+E2U" "!^+46(.*)$!ldap://ldap.se/cn=01
```

- Permet une délégation hiérarchique conforme au téléphone
- Pour +33 RNRT <http://www.numerobis.prd.fr>

## ENUM RFC 2916 : E.164 number and DNS

- URI : Uniform Resource Identifiers  
Problèmes de persistence avec le temps (noms de machine changent,...)
- ↵ URN : Uniform Resource Names RFC 2141
- Système de règles de réécritures complexes stockées dans le DNS avec des NAPTR (Naming Authority Pointer)  
`http.uri.arpa. IN NAPTR`

```
;; order pref flags service      regexp          replacement
    100     90   ""        ""      "!^http://([/:]+)!1!i" .
```

- Comment trouver un serveur associé au service d'une URN ?
- Généralise ENUM  
`example.com.`

```
;;      order pref flags service      regexp replacement
IN NAPTR 100 50 "a"      "z3950+N2L+N2C"      ""      cidserver.example.com
IN NAPTR 100 50 "a"      "rcds+N2C"           ""      cidserver.example.com
IN NAPTR 100 50 "s"      "http+N2L+N2C+N2R"    ""      www.example.com.
```

- Encore de la science fiction...

RFC 3401, RFC 3402, RFC 3403, RFC 3404, RFC 3405

- Principes du DNS
- Créer son domaine
- Formats des zones
- ↳ Comment utiliser le DNS sur un client ?
- Outils
- Description de BIND
- Générer un fichier de zone
- BIND avancé : DNSSEC, IPv6
- Mise en pratique
- Table des matières

- Fichier de configuration du resolver : explique à sa machine comment faire les traductions
- man resolv.conf (resolver de Sun) ou man -s 5 resolver (resolver BIND)
- Options
  - ▶ nameserver *adresse* précise le serveur de nom à utiliser
  - ▶ domain *name* définit le nom de domaine local rajouté aux noms relatifs
  - ▶ search *searchlist* définit une liste de domaines (séparés par des espaces) essayés lors d'une résolution de nom relatifs
  - ▶ sortlist *addresslist* trie les réponses dans l'ordre de préférence des numéros de réseaux donnés

- ▶ options *optionlist* précise certaines options fines
  - options ndots:2 : recherche des noms sans les considérer comme locaux s'ils ont 2 « . » ou plus
- Si un nom complet local est défini avec la commande hostname, le domaine peut ne pas être précisé dans /etc/resolv.conf
- Vérifier que la syntaxe est correctement comprise avec un set all dans nslookup

```
nameserver 192.44.75.10
nameserver 192.108.115.2
nameserver 192.44.77.1
search enst-bretagne.fr enstb.org ensmp.fr trad.org
domain enst-bretagne.fr
```

On peut prendre en compte un fichier d'alias de noms en initialisant la variable d'environnement HOSTALIASES à ce nom de fichier

- Précise l'utilisation des systèmes de nommage par ressource
- Ressources aliases, automount,...,*hosts*,...
- Systèmes de nomenclatures : files, nis, nisplus, dns, compat, xfn
- Exemple
  - hosts : xfn dns nis [NOTFOUND=return] files
- man nsswitch.conf

- Resolver standard :
  - ▶ Bibliothèque lié avec processus utilisateur
  - ▶ Nouveaux protocoles (chaînes de bits et DNAME IPv6, DNSSEC,...)
  - ▶ Devient trop complexe et trop lourd
- Idée avoir un processus indépendant qui factorise le travail de résolution
- Accessible par bibliothèque simplifiée (*light-weight resolver*) par UDP port 921 sur localhost (résout problèmes d'insécurité)
- Travail effectué par processus démon lwresd configuré via /etc/lwresd.conf ou BIND avec directive lwres

- Principes du DNS
- Créer son domaine
- Formats des zones
- Comment utiliser le DNS sur un client ?
- ❖ Outils
- Description de BIND
- Générer un fichier de zone
- BIND avancé : DNSSEC, IPv6
- Mise en pratique
- Table des matières

- Nécessité d'avoir des outils de mise au point
- Besoin d'interroger directement un DNS
- Court-circuite la résolution classique par le système d'exploitation (NIS, /etc/hosts, /etc/resolv.conf)
- Possibilité de demander toute une zone
- Permet de simuler des requêtes inter-DNS
- Peut passer outre les dépassemens de temps de réponse

- Permet d'avoir rapidement un survol
- Ne permet généralement pas de faire des tests internes, hors productions

## Exemple

- « robtex swiss army knife internet tool »

<http://www.robtex.com/> dépiote entre autre graphiquement informations DNS mais aussi

rbl (example: 213.41.240.87) checks multiple RBL:s. check if you

ipnumber (example: 213.41.240.87) checks ipnumber

hostnames (example: keryell.pck.nerim.net) checks dns-info for a

domainnames (example: pck.nerim.net) checks dns-info for a domai

whois lookup (example: pck.nerim.net) checks whois-info for a do

c-net (example 213.41.240) checks reverse for a c-net

route (example 213.41.128.0/17 checks a specific route

as numbers (example: AS13193 (ASN NERIM Nerim xDSL Internet Prov  
bgp announcements (example: AS13193)  
as macros (example: as-ams-ix-peers)  
rfcnumbers (example: rfc2822)

- <http://dnsmon.ripe.net/> surveille communications entre différents serveurs DNS (cf statistiques lors d'attaques)
- <http://www.cymru.com/monitoring/dnssumm/> « TEAM CYMRU DNS Name Server Status Summary » donne informations de connexion avec serveurs racines

Programme maintenu par l'équipe de BIND... donc à préférer

- man dig
-  Interroge par défaut les serveurs de nom de /etc/resolv.conf mais ignore search ou domain
- dig [*@server*] [-f *query-file*] [-k *key-file*] [-y *name : key*] [-i] [[-x] *domain* [*query-type*] [+*query-option*] [-*dig-option*] ]\*
  - ▶ *query-type* : any, a, sig, mx, axfr (transfert de zone), ixfr=*N* (transfert de zone incrémentale depuis le n° de série *N*),... Peut aussi être précisé par -t *query-type*
  - ▶ -x devant une adresse demande automatiquement la requête inverse dans in-addr.arpa et PTR ou ip6.arpa. Rajouter -i pour vieilles requêtes IPv6 dans ip6.int
  - ▶ Plein d'options

- + [no]tcp : utilise TCP au lieu d'UDP
- +domain=*somename* : utilise un domaine par défaut
- + [no]search : utilise la liste de domaines de `resolv.conf`
- + [no]cdflag : positionne le bit CD (*checking disabled*) pour éviter les vérifications DNSSEC
- + [no]recurse : positionne le bit RD (*recursion desired*) demandant au serveur la recherche récursive
- + [no]nssearch : affiche le SOA du domaine selon tous les serveurs DNS du domaine
- + [no]trace : trace récursivement la requête depuis les serveurs racines
- + [no]short : moins verbeux
- + [no]dnssec : demande à utiliser DNSSEC

- ▶ *key-file* contient une clé pour DNSSEC
- ▶ *name:key* idem en ligne de commande  car ps peut l'afficher...
- Plein d'autres options...  

```
dig @dns.princeton.edu cri.ensmp.fr any +norecurse
```
- Notation `-x 128.9.0.32` à la place de  
`32.0.9.128.in-addr.arpa`
- Pour avoir le contenu d'une zone :  

```
dig @dns-cri.ensmp.fr enstb.org axfr
```

  
...si autorisé !
- `dig --help` et `dig -h` plus complet
- Possible de mettre des options dans son `~/.digrc`

- Demande une traduction de base utilisant le serveur de nom défini sur la machine (cf /etc/resolv.conf)
- Nom ↗ numéro IP : host cri  
cri.ensmp.fr has address 192.54.172.200  
cri.ensmp.fr mail is handled (pri=4) by baloo.ensmp.fr  
cri.ensmp.fr mail is handled (pri=1) by cri.ensmp.fr  
cri.ensmp.fr mail is handled (pri=2) by fontainebleau.ensmp.fr  
cri.ensmp.fr mail is handled (pri=3) by paris.ensmp.fr
- Numéro IP ↗ nom : host 192.54.172.231  
231.172.54.192.IN-ADDR.ARPA domain name pointer recanati.ensmp.fr
- Options utiles :
  - ▶ -a verbeuse

```
deauville-keryell > host -a iar2m.ensmp.fr
Trying null domain
rcode = 0 (Success), ancount=5
iar2m.ensmp.fr 154406 IN      A      10.2.16.200
iar2m.ensmp.fr 154406 IN      MX     1 cri.ensmp.fr
```

iar2m.ensmp.fr	154406	IN	MX	2 fontainebleau.ensmp.fr
iar2m.ensmp.fr	154406	IN	MX	3 paris.ensmp.fr
iar2m.ensmp.fr	154406	IN	MX	4 baloo.ensmp.fr
Additional information:				
cri.ensmp.fr	154406	IN	A	192.54.172.200
fontainebleau.ensmp.fr	154406	IN	A	192.54.148.100
paris.ensmp.fr	154406	IN	A	192.54.165.200
baloo.ensmp.fr	154406	IN	A	192.54.173.101

- ▶ -d affiche les transactions réseau
- ▶ -l liste un domaine host -l ensmp.fr
- man host



## Programme en voie d'abandon

- `man nslookup` (version Sun) ou `man -s 8 nslookup` (version BIND)  
`nslookup [-option ...] [host-to-find | -[server]]`
- Mode ligne : similaire au programme `host`. Possibilités d'avoir plus d'options (cf. `set` du mode interactif)
- Mode interactif : sorte de shell d'interrogation des DNS
  - ▶ `? ou help` affiche l'aide en ligne
  - ▶ `host [server]` demande de l'information sur le nom ou numéro de machine `host` au serveur `server`
  - ▶ `server domain` change le serveur par défaut
  - ▶ `lserver domain` change le serveur par défaut en utilisant le serveur initial. Utile lorsqu'on s'est trompé dans la commande `server...`

- ▶ root utilise le serveur racine comme défaut
- ▶ ls [*option*] *domain* [>[>] *filename*] récupère les informations disponibles sur *domain* et les affiche à l'écran ou les met dans un fichier. Option -t demande tous les types d'information par exemple
- ▶ view *filename* fait un more du fichier
- ▶ set all affiche l'état des options
- ▶ set *keyword* [=value] positionne une option
  - [no]debug met/enlève le debug
  - [no]d2 met/enlève le debug intensif
  - domain=*name* change le nom du domaine local
  - srchlist=*name1 / name2 / ...* change la liste des domaines rajoutés lors de la résolution d'un nom et change le domaine local à *name1*

- type=*value* type de réponses demandées
  - ANY tout type d'information
  - A que les adresses.  valeur par défaut...  
~~~ \$HOME/.nslookuprc
  - CNAME que les noms canoniques
  - MX que les échangeurs de mail
  - NS que les noms des serveurs
  - PTR que les noms
  - SOA que l'entête d'une zone
- [no]recurse demande la récursion du serveur (vrai par défaut)
- [no]vc pour utiliser TCP au lieu d'UDP (pratique si derrière certains pare-feux)

- ▶ finger [*name*] [>[>]] *filename* fait un finger sur la machine courante
- ▶ exit ou ^D pour sortir
- Le fichier \$HOME/.nslookuprc peut contenir ses options préférées

- Principes du DNS
- Créer son domaine
- Formats des zones
- Comment utiliser le DNS sur un client ?
- Outils
- Description de BIND
- Générer un fichier de zone
- BIND avancé : DNSSEC, IPv6
- Mise en pratique
- Table des matières

- BIND est le gestionnaire de DNS le plus connu et probablement un des plus complets  
<http://mydns.bboy.net/survey/> sondage 2004
- Fonctionne sur Unix et Windows 2000
- DNS Dynamic Updates RFC 2136
- DNS Change Notification RFC 1996
- Nouvelle syntaxe des fichiers de configuration avec la version 8/9 par rapport à la version 4 et :
  - ▶ Système de log flexible par catégories
  - ▶ Listes d'accès par adresse IP sur les requêtes, transferts de zones et mises à jours pour chaque zone
  - ▶ Transferts de zones plus efficaces
  - ▶ Meilleures performances pour les serveurs gérant des milliers de zones

- ▶ Sécurité
- ▶ Parallélisé pour multi-processeurs
- ▶ Plein de bugs corrigés...

- Possibilité de convertir un fichier de configuration version 4.9.*x* en version 8/9 via  
contrib/named-bootconf/named-bootconf.sh

- Si pas déjà installé...
- Un extrait de ce qui est en rapport avec bind :  
aptitude search bind donne entre autres :  
autodns-dhcp - Automatic DNS updates for DHCP  
bind9 - Internet Domain Name Server  
bind9-doc - Documentation for BIND  
bind9-host - Version of 'host' bundled with BIND 9.X  
dhcp-dns - Dynamic DNS updates for DHCP  
dlint - Checks dns zone information using nameserver lookups  
dnsmasq - A caching DNS forwarder.  
dnsutils - Clients provided with BIND  
host - Utility for Querying DNS Servers  
ldap2dns - LDAP based DNS management system.  
libbind-confparser-perl - Parser class for BIND configuration files  
libbind-dev - Static Libraries and Headers used by BIND  
liblwres1 - Lightweight Resolver Library used by BIND

lwresd - Lightweight Resolver Daemon

nslint - Lint for DNS files, checks integrity

- Une installation :

```
amd1:~/fai# aptitude install bind9 bind9-doc bind9-host dnsutils
```

```
Reading Package Lists... Done
```

```
Building Dependency Tree... Done
```

```
Sorry, bind9-host is already the newest version.
```

```
Sorry, dnsutils is already the newest version.
```

```
The following NEW packages will be installed:
```

```
bind9 bind9-doc
```

```
0 packages upgraded, 2 newly installed, 0 to remove and 10 not upgraded
```

```
Need to get 0B/387kB of archives. After unpacking 943kB will be used.
```

```
Selecting previously deselected package bind9.
```

```
(Reading database ... 90425 files and directories currently installed.)
```

```
Unpacking bind9 (from .../b/bind9/bind9_9.2.1-4_i386.deb) ...
```

```
Selecting previously deselected package bind9-doc.
```

```
Unpacking bind9-doc (from ....bind9-doc_9.2.1-4_all.deb) ...
```

```
Setting up bind9 (9.2.1-4) ...
```

```
Starting domain name service: named.
```

```
Setting up bind9-doc (9.2.1-4) ...
```

```
amd1:~/fai# ps auxww|grep named
```

|      |       |     |     |       |      |       |   |       |      |             |
|------|-------|-----|-----|-------|------|-------|---|-------|------|-------------|
| root | 17192 | 0.0 | 0.4 | 10156 | 2132 | ?     | S | 16:08 | 0:00 | /usr/sbi... |
| root | 17193 | 0.0 | 0.4 | 10156 | 2132 | ?     | S | 16:08 | 0:00 | /usr/sbi... |
| root | 17194 | 0.0 | 0.4 | 10156 | 2132 | ?     | S | 16:08 | 0:00 | /usr/sbi... |
| root | 17195 | 0.0 | 0.4 | 10156 | 2132 | ?     | S | 16:08 | 0:00 | /usr/sbi... |
| root | 17196 | 0.0 | 0.4 | 10156 | 2132 | ?     | S | 16:08 | 0:00 | /usr/sbi... |
| root | 17202 | 0.0 | 0.0 | 1324  | 428  | pts/0 | R | 16:08 | 0:00 | grep nam... |

- Une trace dans les messages systèmes /var/log/syslog :

```
Dec  4 16:08:13 amd1 named[17192]: starting BIND 9.2.1
```

```
Dec  4 16:08:13 amd1 named[17192]: using 1 CPU
```

```
Dec  4 16:08:13 amd1 named[17194]: loading configuration from '/etc/b...
```

```
Dec  4 16:08:13 amd1 named[17194]: listening on IPv4 interface lo, 127.0.0.1#53
Dec  4 16:08:13 amd1 named[17194]: listening on IPv4 interface eth0, 192.168.1.10#53
Dec  4 16:08:13 amd1 named[17194]: listening on IPv4 interface eth1, 192.168.1.11#53
Dec  4 16:08:13 amd1 named[17194]: command channel listening on 127.0.0.1#953
Dec  4 16:08:13 amd1 named[17194]: command channel listening on ::1#953
Dec  4 16:08:13 amd1 named[17194]: zone 0.in-addr.arpa/IN: loaded serial 1
Dec  4 16:08:13 amd1 named[17194]: zone 127.in-addr.arpa/IN: loaded serial 1
Dec  4 16:08:13 amd1 named[17194]: zone 255.in-addr.arpa/IN: loaded serial 1
Dec  4 16:08:13 amd1 named[17194]: zone localhost/IN: loaded serial 1
Dec  4 16:08:13 amd1 named[17194]: running
```

Ça marche !

## Au choix

- ▶ `cd /usr/ports/net/bind9 && make install clean`
- ▶ `portinstall bind9`

- <http://www.isc.org/products/BIND/docs> : vieille documentation (v8)
- FAQ dans le répertoire principal
- Répertoire doc/ de BIND 9.2.0 par exemple :
  - ▶ arm documentation sur BIND et sa configuration au format DOCBOOK/XML et HTML
  - ▶ draft/ versions provisoires de RFC potentiels
  - ▶ man/ manuels à installer
  - ▶ misc/ divers dnssec, ipv6, migration, options, sdb (*Simplified Database Interface*)
  - ▶ rfc/ des RFC concernant DNS et BIND
- Sous Linux/Debian, `dpkg -L bind9-doc` montre que c'est dans `/usr/share/doc/bind9-doc` et en particulier :  
`/usr/share/doc/bind9-doc/arm/Bv9ARM.html`

Restructuration du code dans BINDv9 ↪  $\exists$  des régressions par rapport à BINDv8 😞  
↪ besoin de lire la documentation régulièrement...

- Si on veut le faire *a la mano*
- Répertoire src/ de BIND 9

./configure

make

Sous root :

make install

- man -s 8 named pour la version 9
- docs documentation sur BIND et sa configuration
- Par défaut utilise
  - ▶ /etc/named.conf comme fichier de configuration
  - ▶ /var/run/named.pid numéro de processus
  - ▶ /var/tmp/named\_dump.db vidage de la base de données du serveur
  - ▶ /var/tmp/named.run sortie de debug
  - ▶ /var/tmp/named.stats statistiques du serveur
- Dépend des conventions utilisées dans le système d'exploitation

```
/*
 * A simple BIND configuration
 */

/* Commentaire */
// Comment taire ?
# Comme en terre
options {
    directory "/var/named";
};

logging {
    category lame-servers { null; };
    category cname { null; };
};

zone "isc.org" in {
    type master;
    // Le fichier de référence :
    file "master/isc.org";
```

```
};  
  
zone "vix.com" in {  
    type slave;  
    // Le fichier de sauvegarde :  
    file "slave/vix.com";  
    masters { 10.0.0.53; };  
};  
  
zone "." in {  
    type hint;  
    // Le fichier définissant les racines :  
    file "named.cache";  
};  
  
zone "0.0.127.in-addr.arpa" in {  
    type master;  
    file "master/127.0.0";  
};
```

doc/html/include.html

include *path\_name* ;

Permet de hiérarchiser le fichier de configuration

doc/html/options.html

```
options {  
    [ directory path_name ; ]  
    [ named-xfer path_name ; ]  
    [ dump-file path_name ; ]  
    :  
};
```

Par exemple :

- host-statistics *yes\_or\_no* ; maintient des statistiques sur toutes les machines interagissant avec le serveur
- forward ( *only* | *first* ) ; fait suivre les requêtes à un ensemble précis de serveurs. Avec *first*, si le serveur consulté ne répond pas à la question, le serveur local fait une requête normale.

- forwarders { [ *in\_addr* ; [ *in\_addr* ; ... ] ] } ; liste des machines à qui les requêtes sont acheminées. Intérêt du principe : pas de connexion directe à internet (pare-feu) ou serveurs implémentant de gros caches. Par défaut, pas de *forwarding*
- check-names ( master | slave | response ) ( warn | fail | ignore) ; vérifie la syntaxe des noms de domaine. Par défaut :  
`check-names master fail;`  
`check-names slave warn;`  
`check-names response ignore;`
- allow-query { *address\_match\_list* } ;
- allow-transfer { *address\_match\_list* } ;
- allow-recursion { *address\_match\_list* } ; pour éviter d'être DNS ouvert

- `topology { address_match_list } ;` permet de spécifier des priorités pour l'interrogation de serveurs. Utile si on a une structure réseau complexe dans un même domaine : plusieurs réseaux locaux, plusieurs réseaux distants, etc. comme aux Mines
- `blackhole { address_match_list } ;` : met ces requêtes à la poubelle (style RFC 1918)
- `transfer-format ( one-answer | many-answers ) ;` : compacte les transferts de zone

doc/html/zone.html

- La liste de serveurs racines minimale pour le démarrage

```
zone "." [ ( in | hs | hesiod | chaos ) ] {  
    type hint;  
    file path_name;  
    [ check-names ( warn | fail | ignore ); ]  
};
```

- La définition d'une zone maître

```
zone domain_name [ ( in | hs | hesiod | chaos ) ] {  
    type master;  
    file path_name;  
    [ database "une_base_de_données_où_se_servir"; ]  
    [ check-names ( warn | fail | ignore ); ]  
    [ allow-update { address_match_list }; ]  
    [ allow-query { address_match_list }; ]
```

```
[ allow-transfer { address_match_list } ; ]  
[ notify yes_or_no ; ]  
[ also-notify { ip_addr ; [ ip_addr ; ... ] } ; ]  
};
```

- Autres zones de délégation

```
zone domain_name [ ( in | hs | hesiod | chaos ) ] {  
    type ( slave | stub );  
    [ file path_name ; ]  
    masters { ip_addr ; [ ip_addr ; ... ] } ;  
    [ check-names ( warn | fail | ignore ) ; ]  
    [ allow-update { address_match_list } ; ]  
    [ allow-query { address_match_list } ; ]  
    [ allow-transfer { address_match_list } ; ]  
    [ max-transfer-time-in number ; ]  
    [ notify yes_or_no ; ]
```

```
[ also-notify { ip_addr; [ ip_addr; ... ] }; ]  
};
```

- ▶ Zone esclave (*slave*) resynchronisée sur des maîtres
  - Peut aussi notifier les autres serveurs (modification locale dynamique, utilisation d'esclaves comme maîtres par d'autres esclaves car le maître n'est pas accessible,...)
  - Peut aussi notifier d'autres serveurs (pour pallier à des manques de visibilité directe avec le maître,...)
- Une zone **stub** permet de récupérer la liste des NS d'un domaine qu'on délègue et de l'utiliser sans avoir à mettre à jour sa liste complète de NS, ni la glu  
Comme esclave mais en ne chargeant que les NS et la glu
- Zone **forward** : permet de rediriger les requêtes vers un serveur

- ▶ Résolution d'un Intranet
- ▶ Gros cache d'entreprise

Définit les interactions avec un autre serveur DNS

doc/html/server.html

```
server ip_addr {  
    [ bogus yes_or_no ; ] // Censure  
    [ provide-ixfr yes_or_no ; ]  
    [ request-ixfr yes_or_no ; ]  
    [ edns yes_or_no ; ]  
    [ transfers number ; ]  
    [ transfer-format ( one-answer | many-answers ) ; ]]  
    [ keys { string ; [ string ; [...] ] } ; ]  
};
```

doc/html/acl.html

```
acl name {  
    address_match_list  
};
```

Sont définies par défaut les acl suivantes :

- any tout le monde
- none personne
- localhost seulement les adresses IP de la machine locale
- localnets seulement les machines des réseaux auxquels appartient la machine locale

```
acl rire {  
    193.50.97.128/25;  
    2001:660:7302:e000::/52;  
};
```



- Permet de répondre différemment en fonction des clients

~~> Intranet/Extranet

```
view "internal" {  
    // This should match our internal networks.  
    match-clients { 10.0.0.0/8; };  
    // Provide recursive service to internal clients  
    recursion yes;  
    // Provide a complete view of the example.com zone  
    // including addresses of internal hosts.  
    zone "example.com" {  
        type master;  
        file "example-internal.db";  
    };  
};  
view "external" {  
    match-clients { any; };
```

```
// Refuse recursive service to external clients.  
recursion no;  
// Provide a restricted view of the example.com zone  
// containing only publicly accessible hosts.  
zone "example.com" {  
    type master;  
    file "example-external.db";  
};  
};
```

- Nouveau dans BIND v9
-  Si on met des *view* il faut tout mettre en *view...*

doc/html/logging.html

Très complet. Par défaut :

```
logging {  
    category default { default_syslog; default_debug; };  
    category panic { default_syslog; default_stderr; };  
    category packet { default_debug; };  
    category eventlib { default_debug; };  
};  
  
channel default_syslog {  
    syslog daemon;          # send to syslog's daemon facility  
    severity info;          # only send priority info and higher  
};  
  
channel default_debug {  
    file "named.run";       # write to named.run in the working directory  
                           # Note: stderr is used instead of "named.run"  
                           # if the server is started with the "-f" option.  
    severity dynamic;       # log at the server's current debug level  
};
```

```
channel default_stderr { # writes to stderr
    file "<stderr>";      # this is illustrative only; there's currently
                           # no way of specifying an internal file
                           # descriptor in the configuration language.
    severity info;         # only send priority info and higher
};

channel null {
    null;                  # toss anything sent to this channel
};
```

Format de configuration :

```
logging {
    [ channel channel_name {
        ( file path_name
            [ versions ( number | unlimited ) ]
            [ size size_spec ]
        | syslog ( kern | user | mail | daemon | auth | syslog | lpr |
                  news | uucp | cron | authpriv | ftp |
```

```
    local0 | local1 | local2 | local3 |
    local4 | local5 | local6 | local7 )

| null );

[ severity ( critical | error | warning | notice |
              info | debug [ level ] | dynamic ); ]
[ print-category yes_or_no; ]
[ print-severity yes_or_no; ]
[ print-time yes_or_no; ]
};

[ category category_name {
  channel_name; [ channel_name; ... ]
}; ]
...
};


```

De nombreuses catégories d'événements existent : queries, update, packet, load, lame-servers, statistics,...

- Utilise BIND en processus indépendant qui factorise le travail de résolution
- Accessible par bibliothèque simplifiée (*light-weight resolver*) par UDP port 921 sur localhost par défaut (résout problèmes d'insécurité)
- Permet de paramétriser les clés de sécurité, les signatures, les options, IPv6, les statistiques,...

```
lwres {  
    [ listen-on { ip_addr [port ip_port] ; [ ip_addr [port ip_port]  
        [ view view_name ; ]  
        [ search { domain_name ; [ domain_name ; ... ] }; ]  
        [ ndots number ; ]  
    };];
```

doc/html/key.html

```
key key_id {  
    algorithm algorithm_id;  
    secret secret_string;  
};
```

Définit une clé qui sera utilisée dans une autre directive pour sécuriser des accès, authentifier,...

- Linux/Debian

- ▶ Lancé depuis /etc/rc2.d/S15bind9
  - ▶ Classiquement

```
/etc/init.d/bind9 {start|stop|reload|restart|force-reload}
```

- Solaris

- ▶ man named en natif sur le système (version 4 en Solaris 2.6, 8 en Solaris 7)
  - ▶ man -s 8 named pour la version 8 qui nous intéresse
  - ▶ Le nom de l'exécutable de BIND est named
  - ▶ Solaris 9 lance named depuis /etc/rc2.d/S72inetsvc (lien vers /etc/init.d/inetsvc)
  - ▶ Modifier /etc/init.d/inetsvc pour une syntaxe version 8 le cas échéant (inutile à partir de Solaris 7). Par exemple

```
if [ -f /usr/sbin/in.named -a -f /etc/named.boot ]; then
    /usr/sbin/in.named;      echo "starting internet domain name
en
if [ -f /usr/sbin/named -a -f /etc/named.conf ]; then
    /usr/sbin/named;      echo "starting internet domain name ser
```

- Contrôle le fonctionnement de NAMED à distance via TCP port 953 (par rapport au vieux ndc)
- man rndc

```
rndc [-c config] [-s server] [-p port] [-y key] command [command]
```

- ▶ stop : arrête NAMED en sauvegardant les mises à jour en cours
- ▶ restart : arrête et redémarre NAMED mais... non implémenté !
- ▶ halt : arrête NAMED froidement
- ▶ status : status de NAMED
- ▶ reconfig : recharge la configuration et les nouvelles zones
- ▶ reload [*zone [class [view]]*] : recharge les zones primaires et secondaires en fonction des numéros de série

- ▶ flush [*view*] : vide les caches du serveur
- ▶ stats : met les statistiques de NAMED dans /var/tmp/named.stats ou /var/cache/bind/named.stats
- ▶ dumpdb : met la base de données et le cache courant de NAMED dans /var/tmp/named\_dump.db ou /var/cache/bind/named\_dump.db
- ▶ querylog : NAMED enregistre toutes les requêtes via syslog
- ▶ trace/notrace : incrémente/annule le niveau de trace qui va dans /var/tmp/named.run

- Éviter que tout le monde puisse contrôler son BIND...
- Configuration par `rndc.conf` : clé (secret partagé) dans `/etc/bind/rndc.key` :

```
key "rndc-key" {
    algorithm hmac-md5;
    secret "Izb8d/vxJAfeBmZQ3LVzcQ==";
};
```

- Si accès à la clé :

```
amd1:~/fai# rndc stats
```

- Si pas accès à la clé :

```
keryell@minou:~$ /usr/sbin/rndc stats
rndc: error: none:0: open: /etc/bind/rndc.key: permission denied
rndc: could not load rndc configuration
```

- Voir la doc pour avoir plusieurs clés pour différents serveurs

- `addr` affiche une adresse IPv4 ou v6 en décimal ou hexadécimal
- `dnsquery` autre outil pour faire des requêtes à un serveur man  
`dnsquery`
- `named-xfer` permet de tester les transferts de zone. Utilisé en fait par `named` lui-même pour ses transferts. Typique BINDv8 man  
`named-xfer`
- `nsupdate` envoie des requêtes de mise à jour dynamique

Réalise RFC 2136

- Permettre des mises à jour de DNS (en fonction de DHCP,...) sans devoir relancer le DNS

- Commande nsupdate d'envoi de requêtes de mise à jour dynamique. Commandes du style

```
update add domaine TTL [class] type rdata
```

```
update delete domaine [type] [rdata]
```

```
send
```

- Possibilité de mettre des préconditions de type :

```
prereq nxdomain nickname.example.com
```

```
update add nickname.example.com 86400 CNAME somehost.example.com
```

```
send
```

- Sécurité avec nsupdate [-d] [-v] [[-y *keyname :secret*] [-k *keyfile*]] [*filename*]  si secret en ligne de commande car avec ps auxxww ou ps -ef...

- man nsupdate :
  - ▶ -d : trace en mode debug
  - ▶ -v : utilise TCP
- BINDv9 incrémente le numéro de série de la zone à chaque mise à jour dynamique ↗ si en plus un mécanisme style h2n
- BINDv9 garde un fichier de journalisation des modifications qui garde le  $\Delta$  par rapport au fichier de zone d'origine (écrit si rndc stop)

- Principes du DNS
- Créer son domaine
- Formats des zones
- Comment utiliser le DNS sur un client ?
- Outils
- Description de BIND
  - ❖ Générer un fichier de zone
- BIND avancé : DNSSEC, IPv6
- Mise en pratique
- Table des matières

-  Il faut garantir la cohérence noms ↔ adresses
- À la main... Bon pour les petits domaines
- Utiliser des outils. <http://www.dns.net/dnsrd/tools.html> ou maison. Aux Mines outil maison (archéologie : amélioré d'Ulm via un awk ~ perl avec a2p au passage, lui même amélioré de l'INRIA...) utilisant un /etc/hosts contenant des pseudo-commentaires
- Par exemple h2n

<http://examples.oreilly.com/dns4/dns.4ed.tar.Z>, étendu ensuite chez HP <ftp://ftp.hpl.hp.com/pub/h2n>

Installation :

```
cp h2n /usr/local/bin
```

```
cp h2n.man /usr/local/man/man8/h2n.8
```

Autre possibilité pour FreeBSD :

```
portinstall h2n
```

- Un outil permet d'avoir une idée du format des fichiers de configuration...

- Génère des fichiers db. . . à partir du /etc/hosts local (= base de données)
- `h2n -d domain -n network [:netmask] [options]`
- *domain* domaine de sa zone. Si plusieurs domaines, faire tourner plusieurs fois `h2n` et générer le fichier de configuration à la main
- *network [:netmask]* numéro de réseau. Plusieurs `-n` sont possibles
- On peut spécifier des RR spéciaux par fichier db. . . en les incluant dans un fichier spcl. . .
- *options* :
  - ▶ `-b boot-file` autre fichier de configuration que par défaut (utile pour BINDv8-9)
  - ▶ `-C comment-file` permet de générer des RR supplémentaires à partir de pseudo-commentaires dans

/etc/hosts à partir de traductions dans *comment-file* ayant le format

key:resource      record

Exemple pour préciser un MX particulier à une machine.

/etc/hosts contient

192.54.172.242 deauville # cri mx-cri Sun Ultra 1/140

Avec dans *comment-file*

mx-cri:IN MX 10 cri.ensmp.fr.

- ▶ -c *remote-domain*
- ▶ -e *domain* exclut toutes les machines du domaine
- ▶ -f *file* lit les options dans un fichier
- ▶ -H *file* utilise un autre fichier qu'/etc/hosts

- ▶ -h *host* précise une autre machine pour le SOA que la machine locale
- ▶ -i *serial* impose une valeur aux numéros de série
- ▶ -M ne génère pas de MX par défaut qui est soi-même pour chaque machine. Ce MX peut aussi être supprimé par machine avec [no smtp] en commentaire de /etc/hosts
- ▶ -m *priorité:machine* rajoute pour chaque nom cet enregistrement MX
- ▶ -N *netmask* applique ce *netmask* à tous les réseaux
- ▶ -o *refresh:retry:expire:minimum* change les valeurs par défaut dans le SOA (10800:3600:604800:86400)  
⚠ dans BINDv9 le *minimum* indique le temps de cachage pour les réponses négatives ! Donc, modifier ce paramètre
- ▶ -s *server* spécifie un serveur de nom (maître ou esclave). Permet les notifications de BIND 8. Plusieurs -s

- ▶ -t génère un champs TXT à partir des commentaires de la table des machines /etc/hosts
- ▶ -u *user* précise l'adresse mail du responsable de la zone au lieu de root
- ▶ -v 4|8 génère du format BIND version 4 ou 8.  version 4 par défaut...
- ▶ -w rajoute un WKS précisant smtp comme service
- ▶ -y génère un numéro de série basé sur la date. Pratique pour connaître la date de dernière mise à jour !  
Utilise les vieux fichiers de zone et incrémente  à l'interaction des mises à jour incrémentales qui incrémentent aussi le numéro de série...
- ▶ -Z *adresse* crée un fichier de configuration boot.sec pour un serveur secondaire avec l'*adresse* du maître par défaut

- ▶ *-z adresse* comme *-Z*, crée un fichier de configuration *boot.sec.save* pour un serveur secondaire et un fichier de secours contenant les données (en cas de disparition du primaire...)

- Génère des noms de fichiers de zone tronqués
- Ne gère pas IPv6 ou DNSSEC
- Très configurable mais peu extensible
- Fichiers de zone non triés pour des humains
- Beaucoup d'extension dans la version de HP (CIDR...) ☺ mais fait des suppositions sur noms dans /etc/hosts ☹
- Corollaire
  - ▶ Utiliser h2n dans une infrastructure plus complexe qui pré-/posttraite les flux
  - ▶ Utiliser un autre outil ?  
Projet à base de Python en cours chez moi...

- Part de /usr/local/share/conf/LIT/etc/hosts
- Génère les fichiers du DNS dans /usr/local/share/conf/RIRE/var/named
- Makefile contrôlant le fonctionnement

```
#      $ Header: /usr/local/share/conf/RIRE/var/named/RCS/Makefile,v 1.10
# REFRESH:RETRY:EXPIRE:MINIMUM
# Changed since in BINDv9 MINIMUM is for negative caching:
TIMINGS=7200:3600:604800:600
HOST_FILE=../../../../RIRE/etc/hosts
MASTER=db.127.0.0 db.192.168.70 db.193.50.97 db.enstb.org spcl.enstb.org db
MASTER_DIR=/var/cache/bind/master

db.enstb: Makefile $(HOST_FILE)
        # Remove the first line ($TTL) before processing by h2n:
        -(echo 1d ; echo wq) | ex db.127.0.0
```

```
- (echo 1d ; echo wq) | ex db.192.168.70
- (echo 1d ; echo wq) | ex db.193.50.97
- (echo 1d ; echo wq) | ex db.enstb.org
# To force h2n to declare it in the db file:
-ln -s spcl.enstb.org spcl.enstb
# h2n truncate file names... :-( 
-mv db.enstb.org db.enstb
/home/keryell/h2n/h2n -v 8 -b named.conf -t -w -y -H $(HOST_FILE) \
-C directives -d enstb.org -u keryell.cri.ensmp.fr -h dns2.enstb.or
-s dns2.enstb.org -s dns3.enstb.org -s dns-cri.ensmp.fr \
-s rsm.rennes.enst-bretagne.fr -z 193.48.171.215 -n 193.50.97 \
-n 192.168.70 -n 127.0.0 -m 20:minou.info.enstb.org -o $(TIMINGS)
mv db.enstb db.enstb.org
-(echo %s,spcl.enstb,master/spcl.enstb.org, ; echo wq) | ex db.enstb
# Deal with CIDR :
sed 's/\.97\.50\.193\.IN-ADDR.ARPA.//' db.193.50.97 \
```

```
> db.193.50.97.simple && mv db.193.50.97.simple db.193.50.97  
./CIDR_conf 97.50.193.IN-ADDR.ARPA \  
FR-ENSTB-BREST.97.50.193.in-addr.arpa named.conf  
./CIDR_conf 97.50.193.IN-ADDR.ARPA \  
FR-ENSTB-BREST.97.50.193.in-addr.arpa boot.sec  
./CIDR_conf 97.50.193.IN-ADDR.ARPA \  
FR-ENSTB-BREST.97.50.193.in-addr.arpa boot.sec.save  
# Add back the $$TTL by default needed with BINDv9  
.add_TTL db.127.0.0  
.add_TTL db.192.168.70  
.add_TTL db.193.50.97  
.add_TTL db.enstb.org
```

install:

```
cp $(HOST_FILE) /etc
```



```
mkdir -p $(MASTER_DIR)
cp $(MASTER) $(MASTER_DIR)
/etc/init.d/bind9 force-reload
```

- Programme d'aide CIDR\_conf

```
#! /bin/sh
```

```
IP_NET=$1
IP_NET_CIDR_ZONE=$2
FILE=$3
TMPFILE=$FILE.$$.tmp
SED_EX='echo ''$IP_NET'', ''$IP_NET_CIDR_ZONE'', | sed 's/\.\./\\\.g'
```

```
sed $SED_EX $FILE > $TMPFILE && mv $TMPFILE $FILE
```

- Programme d'aide add\_TTL

```
#! /bin/sh
```

```
FILE=$1
TMPFILE=$FILE.$$.tmp
echo '$TTL 3600 ; 1 hour default TTL' > $TMPFILE
cat $FILE » $TMPFILE
mv $TMPFILE $FILE
```

Répertoire contrib/ de BIND 8 contenant par exemple :

- dnsparse/ outils pour générer un /etc/hosts à partir de zones DNS
- getkeyby/ outils pour récupérer des clés de chiffrement via DNS
- nutshell/ les (vieux) programmes du livre *DNS and BIND* de chez O'Reilly. Récupérer plutôt depuis  
<ftp://ftp.uu.net/published/oreilly/nutshell/dnsbind/dns.tar.Z>
- tic/ Outil générant les zones à partir de fichier /etc/hosts

- Principes du DNS
- Créer son domaine
- Formats des zones
- Comment utiliser le DNS sur un client ?
- Outils
- Description de BIND
- Générer un fichier de zone
- BIND avancé : DNSSEC, IPv6
- Mise en pratique
- Table des matières

<http://www.isc.org/bind-plans.html>

- Pris en charge par l'Internet Software Consortium
- Paul Vixie (architecte et programmeur) & Bob Halley (programmeur)
- IPv6
- Rajout de la sécurité (DNSSEC de TIS et DNSSAFE de RSA)
- Résolution sécurisée (TSIG)
- Transferts de zone incrémentaux (IXFR)
- Amélioration de la mise à jour des zones
- Support payant possible

- Limitations d'IPv4
  - ▶ « Que »  $2^{32}$  adresses
- Passage à des adresses de 128 bits : RFC 3513
- Utilisation de la profusion d'adresse pour plus de flexibilité
  - ▶ Hiérarchisation plus facile : simplification des tables de routage
  - ▶ Généralisation du CIDR (RFC 3587 : IPv6 Global Unicast Address Format)

| $n$ bits              | $m$ bits  | $128 - n - m$ bits |
|-----------------------|-----------|--------------------|
| Global routing prefix | subnet ID | interface ID       |

- Utilisation pour la mobilité
- Fusion avec la téléphonie : UMTS

- Notation hexadécimale plus compacte

```
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
hme0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.44.75.87 netmask ffffff00 broadcast 192.44.75.255
lo0: flags=2000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv6> mtu 8252 index 1
        inet6 ::1/128
hme0: flags=2000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
        inet6 fe80::a00:20ff:fedc:bc6b/10
hme0:1: flags=2080841<UP,RUNNING,MULTICAST,ADDRCONF,IPv6> mtu 1500 index 3
        inet6 2001:660:283:2:a00:20ff:fedc:bc6b/64
```

- Par paquets de 32 bits
- CIDR (RFC 3587) implicite avec longueur du */prefixe*, équivalent du *netmask* d'IPv4
- Notation :: pour remplacer 1 (unique, non ambiguë) séquence de 0 dans l'adresse

- Une interface physique a 48 (Ethernet) ou 64 (IEEE1394/FireWire/i.Link)
- IPv4 : seulement 32 bits, donc pas facile d'allouer une adresse unique à partir de l'adresse physique
  - ▶ Statique
  - ▶ Protocoles compliqués : BOOTP, DHCP, RARP, AutoIP (UPnP),...
- IPv6 : l'adresse physique peut loger dans l'adresse !
- Notion d'adresse locale de lien *disponible à la mise sous tension*

|          |         |
|----------|---------|
| 64 bits  | 64 bits |
| fe80 : : | EUI-64  |

Si pas EUI-64, extension d'IEEE 48bit MAC selon RFC 3513 ou encore tirage aléatoire (PPP,...)

-  On peut tracer une machine avec les 64 bits de poids faible... Mais on peut aussi changer la valeur par défaut ensuite RFC 3041
- Comment récupérer une adresse globale ?

- Système avec état : DHCPv6 RFC 3315
- Système sans état
  - ▶ Neighbour Discovery (ND) via ICMPv6 RFC 2463
  - ▶ Multicast Listener Discovery (MLD) RFC 2910
- Problème : nécessite une mise à jour (sécurisée du DNS)

Défini par IANA RFC 3177

- /48 alloué pour un réseau standard :  $2^{80}$  adresses
- /64 alloué pour un seul sous réseau physique :  $2^{64}$  adresses
- /128 alloué pour un unique système bien défini : 1 adresse

| 3 bits | 45 bits               | 16 bits   | 64 bits      |
|--------|-----------------------|-----------|--------------|
| 001    | Global routing prefix | subnet ID | interface ID |

- Reprendre les vieilles méthodes IPv4 en adaptant ↗ RFC 1886
- Rajout d'un enregistrement AAAA avec des adresses de 128 (4\*A⊕) bits dans les zones directes  
`an-dro.plouzane IN AAAA 2001:7a8:2e22:0:a00:46ff:fe68:181b`
- Équivalent de `in-addr.arpa` passé dans `ip6.int` passé en hexadécimal et en bloc de 4 bit (au lieu de décimal par tranche de 8 bits en IPv4)  
`b.1.8.1.8.6.e.f.f.f.6.4.0.0.a.0.0.0.0.0.2.2.e.2.8.a.7.0.1.0.0.2.ip6.int`  
PTR `an-dro.plouzane.enstb.org`
- Problèmes
  - ▶ Le CIDR n'est pas simple si par alignés sur 4 bits
  - ▶ On pollue `.int` avec `ip6` alors que `.int` est pour des humains
  - ▶ Beaucoup de répétitions dans les adresses
  - ▶ Si on change de fournisseur il faut changer tous les AAAA

- ▶ Si multi-domiciliation, beaucoup de zones redondantes à synchroniser
  - ~~ Méthode en voie d'abandon ? À garder pour « vieux » clients DNS IPv6 ? Eh non ! Lors d'une élection c'est cette méthode qui restera... 😞
- Utilise néanmoins ip6.arpa

- On repart dans arpa avec ip6.arpa BCP 49
- Une adresse IPv6 est souvent définie par morceaux : fournisseur d'accès+réseau local+ interface par exemple
- Nouvel enregistrement A6 permettant ces concaténations en itérant sur des enregistrement de type  
nom IN A6 *taille* *adresse* *zone*  
où *nom* aura l'*adresse* dont les *taille* premiers bits seront à chercher dans le A6 de la *zone*  
`an-dro.info.enstb.org A6 64 ::a00:46ff:fe68:181b ip6.info.enstb.org`  
`ip6.info.enstb.org A6 48 0:0:0:9771:: enstb.vthd.net`  
`enstb.vthd.net A6 0 2001:688:1f9c:`  
Définira l'adresse 2001:688:1f9c:9771:a00:46ff:fe68:181b
- Si une machine est connectée à plusieurs fournisseurs d'accès Internet on aura plusieurs A6 pour gérer les différents préfixes
- Méthode inverse

- ▶ Adresses représentées par un nouveau type RFC 2673 : chaîne de bits pour gérer proprement le CIDR  
  \ [x200106881f9c/48\]
- ▶ Recherche ensuite de  
  \ [x200106881f9c97710a0046ffffe68181b\] .ip6.arpa
- ▶ Introduit une sorte de CNAME spécialisé : DNAME pour faire des réécritures avec des :  
  \ [x200106881f9c8/48\] .ip6.arpa DNAME ip6.vthd.net  
  \ [x9771/16\] .ip6.vthd.net DNAME ip6.info.enstb.org  
  \ [x0a0046ffffe68181b\] .ip6.info.enstb.org PTR an-dro.info.enstb.org  
aura pour effet de faire répondre successivement lors de la recherche :  
  \ [x200106881f9c97710a0046ffffe68181b\] .ip6.arpa CNAME  
    \ [x97710a0046ffffe68181b\] .ip6.vthd.net  
  \ [x97710a0046ffffe68181b\] .ip6.vthd.net CNAME

\[x0a0046ffffe68181b\].ip6.info.enstb.org  
\[x0a0046ffffe68181b\].ip6.info.enstb.org PTR an-dro.info.enstb.org

Cf RFC 3364 pour une comparaison approche AAAA RFC 3363 qui a gagné ou A6 RFC 2874

<http://www.afnic.fr/formation/supports/formation-unicode>

- DNS de base très anglo-saxon : a-z, - et 0-9. Encore moins que l'ASCII ! ☹
- Besoin de noms de domaines français, chinois, russes, arabes, hébreux,...
- Déjà fait pour les entêtes du courriel (ASCII) avec MIME
- Propositions de solutions applicatives non portables... ☹
- Pourtant ∃ UNICODE/ISO 10646 <http://www.unicode.org> visant la représentation de toutes les langues de la Terre (et quelques autres ☺) avec 32 bits
- ∃ différents encodages d'UNICODE : UTF-7, UTF-8 (RFC 3629), UTF-16, UTF-32,...
- Comment faire rentrer ça dans le DNS ?

- Problèmes d'unicité : par exemple lettre accentuée représentable par elle-même ou en combinaison du caractère standard et de l'accentuation, écriture de gauche à droite ou dans d'autres directions
- ↗ Besoin de normaliser avant réponse DNS : stringprep du RFC 3454
- Définir un nouvel encodage d'UNICODE compatible avec le DNS : le *punycode* du RFC 3492
- RFC 3490 Internationalizing Domain Names in Applications (IDNA) : rajoute le préfixe `xn--` devant la chaîne punycodée
- UNICODE est une norme qui évolue ↗ faire évoluer les RFC en conséquence...

<http://www.cymru.com/Documents/secure-bind-template.html>

- Faire tourner BIND dans une prison système ou machine (virtuelle) propre
- Ne pas répondre aux paquets avec adresses improbables
- Limiter la récursion à qui de droit
- Éventuellement avoir une version privée des données

- Internet devient le nerf de la guerre
- Besoin de sécurité
  - ▶ Au niveau d'un client qui interroge un serveur
  - ▶ Entre serveurs DNS
  - ▶ Au niveau du contrôle de BIND (rndc)
  - ▶ Et pourquoi ne pas utiliser le DNS comme composant d'une infrastructure à clé publique (ICP ou PKI) plus vaste ?

- <http://www.dnssec.net>
- Quelques RFC :
  - ▶ RFC 4033 : DNS Security Introduction and Requirements
  - ▶ RFC 4034 : Resource Records for the DNS Security Extensions
  - ▶ RFC 4035 : Protocol Modifications for the DNS Security Extensions
  - ▶ RFC 4641 : DNSSEC Operational Practices
-  DNSSEC ne gère pas
  - ▶ Confidentialité des enregistrements
  - ▶ Dénis de service

- RFC 2845 protège les communications entre serveurs ou clients (indépendant du déploiement de DNSSEC)
- Authentification des transactions par HMAC ou signature avec clé publique contenue dans un enregistrement SIG0 (RFC 2931)
- Associer une clé à un couple de serveur
-  Le nom doit être identique de part et d'autre car transmis avec message
- Contre attaques par rejeu : temps impliqué dans hachage  $\rightsquigarrow$  2 parties doivent être synchronisées à moins de 5 minutes (vive NTP !)
- Création d'une clé aléatoire avec

```
cd /etc/bind
dnssec-keygen -a hmac-md5 -b 128 -n HOST \
minou.lit.enstb.org-dns-cri.ensmp.fr
```

a créé un fichier

Kminou.lit.enstb.org-dns-cri.ensmp.fr.+157+43464.key  
contenant par exemple LizPwe83eTjBNfL05wyiuw==

- Utilisation sur minou.lit.enstb.org dans

/etc/bind/named.conf

```
key minou.lit.enstb.org-dns-cri.ensmp.fr. {  
    algorithm hmac-md5;  
    secret "LizPwe83eTjBNfL05wyiuw==";  
};
```

```
server 193.48.171.215 {
```

```
    keys { minou.lit.enstb.org-dns-cri.ensmp.fr. ;};  
};
```

- Mises à jour authentifiées :

```
allow-update { key minou.lit.enstb.org-dns-cri.ensmp.fr. ;};
```

-  Le secret doit rester secret! ↗ droits des fichiers...

- <http://www.dnssec.net>  
<http://www.nlnetlabs.nl/dnssec>  
<http://www.hsc.fr/ressources/presentations/bind9>
  - ▶ Authentification du contenu des zones
  - ▶ Cryptographie à clé publique : on signe avec une clé privée et tout le monde peut vérifier avec la clé publique qui est... publique !
  - ▶ Distribution de clés publiques avec RR DNSKEY
  - ▶ Signature d'un RR ou RRset d'une zone avec un RR RRSIG correspondant à la clé privée associée à la clé publique RR DNSKEY de la zone
  - ▶ Comment être sûr que la RR DNSKEY n'est pas truandée (par une clé publique d'un pirate...) ?
  - ▶ Dans la zone parente, un RR DS (*Delegation Signer*) certifie avec un hachage qui est signé comme tout RR dans la zone

parente (vérifiable avec la DNSKEY de la zone parente) la DNSKEY de la zone fille associée

- ▶ Vérification de l'origine : un RR est-il bien signé par cette clé ? Cette clé est-elle elle-même signée par la clé du domaine parent, etc.
- ▶ La poule et l'œuf ↗ clés publiques de références (autorités de certification) connues de tous (mises dans fichier de conf de BIND)

```
trusted-keys {  
    string number number number string ;  
    [ string number number number string ; [...] ]  
};
```

- ▶ Fonctionne aussi avec les mises à jour dynamiques : RR spécifiques au DNSSEC recalculés

- ▶ Notion de clé nulle (signée ☺) pour permettre des zones non signées dans des zones signées
- ▶ Clés avec un intervalle de validité ↗ TTL peut être diminué dans une réponse pour tenir compte de la validité
- ▶ Toutes ces signatures à clé publique demande du temps de calcul...

- Création d'un couple clé privée clé publique qui servira à signer les RRset de la zone :

- man dnssec-keygen

```
dnssec-keygen -a RSASHA1 -b 2048 -n ZONE enstb.org
```

génère par exemple

- Kenstb.org.+005+28338.key qui contient le RR de la clé à \$INCLUDEr dans sa zone :

```
enstb.org. IN DNSKEY 256 3 5 AwEAAcgdmzonaV+FntSVHhEmV/8pNvnW4
```

- Kenstb.org.+005+28338.private qui contient la clé privée ultra-secrète servant à signer et à protéger 

- dnssec-signkey permet :

- De signe sa zone avec sa clé privée :

```
dnssec-signzone -o enstb.org db.enstb.org Kenstb.org.+005+2833
```

qui produit une nouvelle zone db.enstb.org.signed

- ▶ au responsable d'une zone de rajouter des RR DS pour signer une zone fille (options -d, -g)

- Besoin de signer aussi la non existence d'un RR car sinon un pirate pourrait faire de dénis de service ou de la génération spontanée
- Besoin de prouver l'existence de tous les RR pour éviter des disparitions...
- Problème des caches sur le chemin et des serveurs secondaires : comment un serveur secondaire qui ne connaît pas la clé secrète de la zone peut signer un enregistrement ?
- Plutôt que renvoyer simplement cette réponse on renvoie un enregistrement signé disant que
  - ▶ ] $a,b$ [ n'existent pas
  - ▶  $a$  et  $b$  étant les réponses les plus proches lexicographiquement encadrant la requête
  - ▶ Évitera d'autres requêtes dans cet intervalle

- ▶ Encodé avec un nouveau RR NSEC *qui est pré-signé*

*a* NSEC *b*

avec aussi la liste des attributs possédés par *b*

-  Même si on a supprimé le transfert de zone, en faisant une requête RR NSEC à partir du nom de zone on aura le nom du premier enregistrement, itérer, etc. ↵ analyse de la zone  
Mais comme l'obscurantisme ne fait pas tout...

-  Pour masquer les adresses, utiliser même technique que dans authentification par mot de passe pour cacher mots de passe
  - ▶  stocker  $H(p)$  dans base de mots de passe plutôt que mot de passe en clair  $p$
  - ▶  $H$  est une fonction de hachage cryptographique (SHA-1...) à sens unique
  - ▶ Authentification du mot de passe  $l$

$$H(l) \stackrel{?}{=} H(p)$$

-  Plutôt que de mettre dans DNS  
*domaine NSEC domaine-suivant*

mettre (RFC 5155, mars 2008)

$H(\text{domaine}) \text{ NSEC3 } H(\text{domaine-dont-le-H-suit})$

- Ranger les hachages de manière lexicographiquement croissante pour permettre de répondre que  $]H(a), H(b)[$  n'existent pas
-  Attaque par dictionnaire
- Pour enregistrements générés dynamiquement,  $\exists \text{ RR}$  NSEC3PARAM décrivant algorithmes et paramètres pour NSEC3
- Problème amusant si requête sur un domaine non existant a le hachage d'un domaine existant... Mais collisions peu probables...

## Repris du RFC 4035

```
example.      3600 IN SOA ns1.example. bugs.x.w.example. (
                1081539377
                3600
                300
                3600000
                3600
                )
3600 RRSIG SOA 5 1 3600 20040509183619 (
                20040409183619 38519 example.
                0Nx0k36rcjaxYtcNgq6iQnpNV5+drqYAsC9h
                7TSJaHCqbhE67Sr6aH2xDUGcqQWu/nOUVzrF
                vkg09ebarZ0GWDKcuwlM6eNB5SiX2K7415LW
                DA7S/Un/IbtDq4Ay8NMNLQI7Dw7n4p8/rjkB
                jV7j86HyQgM5e7+miRAz8V01b0I= )
3600 NS      ns1.example.
3600 NS      ns2.example.
3600 RRSIG NS 5 1 3600 20040509183619 (
                20040409183619 38519 example.
                g113F00f2U0R+SWiXXLHwsMY+qStYy5k6zfd
                EuivWc+wd1fmbNCyql0Tk71HTX6U0xc8AgNf
                4ISFve8XqF4q+o9qlnqIzmppU3LiNeKT4FZ8
                R05urF0voMRTbQxW3U0hXWuggE4g3ZpsHv48
                OHjMeRaZB/FRPGfJPajngcq6Kwg= )
3600 MX      1 xx.example.
3600 RRSIG MX 5 1 3600 20040509183619 (
                20040409183619 38519 example.
                HyDHVVT5KHSZ7Ht0/vypumPmSZQrc0P3tzWB
                2qaKkHVPfau/DgLgS/IKENkY0GL95G4N+NzE
                VyNU8dcT0ckT+ChPcGeVjguQ7a3Ao9Z/ZkUO
```

```
6gmmUW4b89rz1PUxW4jzUxj66PTwoVtUU/iM
W60ISukd1EQt7a0kygkg+PEDxdI= )
3600 NSEC a.example. NS SOA MX RRSIG NSEC DNSKEY
3600 RRSIG NSEC 5 1 3600 20040509183619 (
20040409183619 38519 example.
00k558jHhyrC97ISHnislm4kLMW48C7U7cBm
FTfhke5iVqNRVTB1STLMpgpbDIC9hcryo00V
Z9ME5xPzUEhbvGnHd5sfzgFVeGxr5Nyyq4tW
SDBgIBiLQUv1ivy29vhXy7WgR62dPrZ0PWvm
jfFJ5arXf4nPxp/kEowGgBRzY/U= )
3600 DNSKEY 256 3 5 (
AQ0y1bZVvpPqhg4j7EJoM9rI3ZmyEx20zDBV
rZy/lvI5CQePxXHZS4i8dANH4DX3tbHo161e
k8EFMcsGXxKciJFHyhl94C+NwILQdzsU1SFo
vBZsyl/NX6yEbtw/xN9ZNcrbYvgjjZ/UVPZI
ySFNsgEYvh0z2542lzMKR4Dh8uZffQ==
)
3600 DNSKEY 257 3 5 (
AQ0eX7+baTmvpVHb2CcLnL1dMRWbuscRvHX1
LnXwDzvqp4tZVKp1sZMepFb8MvxhhW3y/0QZ
syCjczGJ1qk8vJe52i0hInKROVLRwxGpMfzP
RLM1Gybr51b0V/1se00Dacj3DomYB4QB5gKT
Yot/K9alk5/j8vfd4jWCWD+E1Sze0Q==
)
3600 RRSIG DNSKEY 5 1 3600 20040509183619 (
20040409183619 9465 example.
ZxgauAuIj+k1YoVE0S1Zfx41fcnKzTFHoweZ
xYnz99JVQZJ33wFS0Q0jcP7VXKkaElXk9nYJ
Xev0/7nAbo88iWsMkSpSR6jWzYYKwfrBI/L9
hjYmyV09m6FjQ7uwM4dCP/bIuV/DKqOAK9NY
NC3AHfvCV1Tp4VKDqxqG7R5tTVM= )
```

```
3600 RRSIG DNSKEY 5 1 3600 20040509183619 (
    20040409183619 38519 example.
    eGL0s90glUqc0mloo/2y+bSzyEfKVOQViD9Z
    DNhLz/Yn9CQZ1DVRJffACQDAUhXpU/oP34ri
    bKBpysRXosczFrKqS50a0bzM0fXCXup9qHAp
    eFIku28Vqfr8Nt7cigZLxjK+u0Ws/4lIRjKk
    7z50XogYVaFzHKillDt3HRxHIZM= )

.a.example. 3600 IN NS ns1.a.example.
3600 IN NS ns2.a.example.
3600 DS 57855 5 1 (
    B6DCD485719ADCA18E5F3D48A2331627FDD3
    636B )
3600 RRSIG DS 5 2 3600 20040509183619 (
    20040409183619 38519 example.
    oXIKit/QtdG64J/CB+Gi8d0vnwRvqrto1AdQ
    oRkAN15FP3iZ7suB7gyTBmXzCjL7XUgQVcoH
    kdhyCuzp8W9qJHgRUSwKKkcSzuyL64nhgjuD
    EML819wlWVs17PR2VnZduM9bLyBhaaPmRKX/
    Fm+v6ccF2EGNLRiY08kdkz+XHHo= )
3600 NSEC ai.example. NS DS RRSIG NSEC
3600 RRSIG NSEC 5 2 3600 20040509183619 (
    20040409183619 38519 example.
    c01YgqJLqlRqmBQ3iap2SyIsK405aqpKSoba
    U9fQ5SMApZmHfq3AgLf1krkXRXvgxTQSkkG2
    039/cRU6Jk/25+fi7Xr5n0VJsb0lq4zsB3I
    BBdjyGDAHE0F5R0Jj87996vJupdm1fbH481g
    sdk0W6Zyqtz3Zos8N0BBkEx+2G4= )

ns1.a.example. 3600 IN A 192.0.2.5
ns2.a.example. 3600 IN A 192.0.2.6
ai.example. 3600 IN A 192.0.2.9
3600 RRSIG A 5 2 3600 20040509183619 (
```

20040409183619 38519 example.



- Un CNAME doit aussi être signé
- Donc un CNAME doit aussi avoir une signature indépendamment de celle potentielle sur le nom pointé
  - ▶ RRSIG
  - ▶ Mais aussi NSEC
- RFC 2535 précise une exception aux RFC 1034 et RFC 1035 précisant qu'un CNAME ne peut pas avoir d'autres enregistrements associés

- Dépasser la distribution de clés publiques (RR DNSKEY)
- Distribution de certificats ( $\approx$  clé publique signées + coordonnées signées par une clé secrète d'un certifieur)  
Distribution par le DNS (RR CERT)
- Autres extensions pour d'autres usages
  - ▶ RR IPSECKEY pour IPsec
  - ▶ SSHFP pour empreintes de clés publiques ssh pour être sûr qu'on se connecte bien à la bonne machine
- Gratuit : utilise infrastructure déployée base de données distribuée DNS

- Génère une zone signée à partir de la clé de zone trouvée dans le fichier de zone

```
dnssec-signzone -o nom-domaine fichier-zone
```

génère *fichier-zone.signed* qui sera utilisé par BIND

- Signe tous les enregistrements de la zone
- Génère tous les RR NSEC

<http://www.ietf.org/html.charters/ipsec-charter.html>

- Démocratisation d'Internet, apparition d'utilisations et d'*utilisateurs* non prévus au départ
- Architecture très décentralisée et peu contrôlée (contrairement au TELEX et au téléphone)
- Problème de l'authentification faible à base d'adresses IP (injection de paquets IP avec de fausses adresses de source,...)
- ↵ Rajout dans le développement d'IPv6 de la sécurité : IPsec (IP Security Protocol), obligatoire dans IPv6
- Comme IPv4 n'en fini pas de survivre : rajouté aussi dans IPv4 (optionnel)
- Utilisation de la cryptographie forte pour assurer :
  - ▶ Confidentialité
  - ▶ Authentification

- ▶ Empêcher les répétitions
- Usages possibles suivant niveau d'implication
  - ▶ Créer des réseaux virtuels privés sécurisés au dessus d'un réseau (public) moins sécurisés
  - ▶ Permettre des accès à distance plus sécurisés
  - ▶ Passage de toutes les machines d'un réseau en IPsec

- En natif à partir de Windows 2000
- En natif dans les Unix commerciaux récents
- D'autres implémentations commerciales indépendantes existent
- En logiciel libre
  - ▶ Natif dans OpenBSD
  - ▶ KAME pour les BSD (projet japonais IPv6)
  - ▶ FreeS/WAN ↗ StrongSwan, OpenSwan pour Linux
- Existe aussi dans les routeurs et pare-feux

- Extension des paquets IP
  - ▶ Modifie chaque paquet IP
  - ▶ Rajoute un entête AH (Authentication Header, RFC 2402) si authentification des paquets demandé

BEFORE APPLYING AH

\*\*\*\*\*  
\*\*\*\*\*

IPv4 | orig IP hdr | | |  
| (any options) | TCP | Data |  
\*\*\*\*\*

AFTER APPLYING AH

\*\*\*\*\*  
\*\*\*\*\*

IPv4 | orig IP hdr | | | |  
| (any options) | AH | TCP | Data |  
\*\*\*\*\*

| <\*\*\*\*\* authenticated \*\*\*\*\*> |

except for mutable fields

- Problème de la mutabilité lors du transport de certains champs (TTL,...) qui sont donc ignorés : considérés comme valant 0 dans le calcul de AH
- Pour la même raison fragmentation ignorée : AH calculé sur tout le paquet IP
- Permet vérification d'intégrité, authentification de l'origine, protection contre les répétitions (optionnel)

- Rajoute un entête ESP (*Encapsulating Security Payload*, RFC 2406) si chiffrement des paquets demandé

## BEFORE APPLYING ESP

The diagram illustrates the structure of an IPv4 header. It consists of several fields arranged horizontally:

- IPv4
- | orig IP hdr |
- | (any options) |
- TCP
- Data

The "orig IP hdr" field is divided into two parts by a vertical line. The "(any options)" field is also divided into two parts by a vertical line.

\*\*\*\*\*

## AFTER APPLYING ESP

\*\*\*\*\*

The diagram illustrates the structure of an IPv4 header with ESP and TCP options. It shows the following fields from left to right:

- IPv4
- orig IP hdr
- | (any options)
- ESP
- Hdr
- TCP
- Data
- ESP
- Auth
- ESP

The 'orig IP hdr' field contains the original IP header, which may include options. The 'ESP' field is present twice, indicating two layers of encapsulation. The first 'ESP' layer contains its own header ('Hdr') and the 'TCP' layer. The second 'ESP' layer contains the 'Data' payload and its own 'Auth' (Authentication) field.

\*\*\*\*\*

|<\*\*\*\*\* encrypted \*\*\*\*\*>|  
|<\*\*\*\*\* authenticated \*\*\*\*\*>|

- Rajoute de la confidentialité ainsi que la même chose qu'AH

- Compression éventuelle avant chiffrement (mieux vaut le faire avant le chiffrement!)
- ▶ ↗ Applications même non sécurisées héritent de la sécurité d'IPsec (mais  si l'attaquant est sur une des machines au bout de la connexion...)
- 2 modes de protection
  - ▶ Mode standard : simple rajout des entêtes IPsec
  - ▶ Mode tunnel : encapsulation dans ESP ou AH de tout le paquet IP d'origine et rajout d'un entête IP
    - ↗ Permet de faire des tunnels/VPN chiffrés entre 2 équipements IPsec et d'améliorer la discréetion des flux (si ESP)
- SA (Security Association)

- ▶ Le mode IPsec de chaque connexion (unidirectionnelle...) est décrit par un SA
- ▶ Contient un triplet adresse de destination, type de sécurité (AH ou ESP), SPI (*Security Parameter Index*, identifiant la sécurité exacte utilisée)
- ▶ Concrètement, les SA d'un nœud sont stockés dans une base de données (SAD) sur chaque nœud
- Éventuellement IKE (*Internet Key Exchange*)

- Nécessité pour chaque connexion de configurer un SA (dans chaque sens)
- Nécessité pour chaque connexion de configurer tous les paramètres associé à un SA : clé, algorithme de chiffrement, politique à suivre côté émission et réception,...
- Lourd ↗ nécessité d'un protocole de configuration plus dynamique : IKE (*Internet Key Exchange*)
  - ▶ Négocie les paramètres de chaque connexion
  - ▶ Échange les clés
  - ▶ Authentifie les parties en présence
- Basé sur
  - ▶ ISAKMP mécanisme générique pour négocier, UDP port 500
  - ▶ SKEME : utilise une création de clés avec Diffie-Hellman ou autre moyen

- ▶ Oakley : utilise Diffie-Hellman ou autre
- Mise en place des SA par IKE à partir d'une politique (la configuration de haut niveau d'IPsec) qui va choisir quels paramètres utiliser en fonction d'adresses, ports, protocoles,...  
≈ procédures de filtrage des pare-feux
- IKE ne régit qu'IPsec et a lui-même besoin de clés d'authentification... PKI ?

- Attaque la sécurité directement au niveau IP
- Standardise la sécurité
- Manque de maturité
- Implémentations pas toujours complètes
- Plus simple avec une PKI
- Incompatible avec la traduction d'adresse (NAT), mais est-ce utile avec IPv6 ? ☺

- Essaye de compenser la complexité de mise en œuvre d'IPsec
- ↵ Opportunistic Encryption dans FreeS/WAN puis StrongSwan, OpenSwan (IPsec pour Linux <http://www.strongswan.org>)
- Idée stocker clé publique pour communiquer avec une machine dans le DNS
- Compatibilité DNS classiques : utilise des champs TXT et non KEY
- Exporte la clef avec `ipsec showhostkey -txt`  
@xy.example.com  
; RSA 2192 bits xy.example.com Thu Jan 2 12:41:44 2003  
IN TXT "X-IPsec-Server(10)=@xy.example.com"  
"AQOF8tZ2... ...+buFuFn/"  
et `ipsec showhostkey -txt 192.0.2.11`  
; RSA 2048 bits xy.example.com Sat Apr 15 13:53:22 2000  
IN TXT "X-IPsec-Server(10)=192.0.2.11" " AQOF8tZ2...+buFuFn/"

-  Fait confiance au DNS ! ↗ combiner avec DNSSEC

- Système de gestion de nom efficace et unificateur
- Comme tous les systèmes cruciaux d'Internet : au moins une implémentation efficace et gratuite
- Portable
- Très complet
- ~~> Travaux pratiques

- Principes du DNS
- Créer son domaine
- Formats des zones
- Comment utiliser le DNS sur un client ?
- Outils
- Description de BIND
- Générer un fichier de zone
- BIND avancé : DNSSEC, IPv6
- ❖ Mise en pratique
- Table des matières

- Créer un domaine *son-nom.com* sur sa machine avec une utilisation maximaliste de h2n
- Pourquoi les autres machines ne voient-elles pas ce domaine ?
- Créer un domaine *son-nom.enstb.org*
- Vérifier avec <http://zonecheck.fr>
- Y mettre par exemple les machines de /etc/hosts
- Mettre en place des politiques d'accès et des politiques de maîtres-esclaves
- Tester les transferts de zone

## List of Slides

## 0 IAR2M — Cours DNS/BIND

1 Copyright (c)

2 Introduction

# 1 Introduction

3 Données cruciales...

4 Histoire

6 Naissance du DNS

7 Croissance du DNS

9 Autres systèmes de nommage

10 RFC et autres standards de l'Internet

11 Ressources

14 Plan

# 13 DNS



BIND & DNS

Département Informatique

- 15 Principe du DNS
- 16 Hiérarchie de nommage
- 20 Espace de nommage dans Internet
- 23 Nom dans le DNS
- 25 Délégation
- 26 Zones
- 27 Système de résolution
- 28 Serveur racine
- 31 Résolution d'un nom
- 33 Glu
- 34 *Lame servers*
- 36 Protocole
- 38 Traduction numéro IP vers nom de domaine
- 41 Requête inverse
- 42 Mécanisme de cache



45 Plan

## 44 Créer son domaine

46 Déclarer un nouveau domaine

47 Choisir un nom

50 Bases whois

53 Enregistrer un nouveau domaine

54 Plan

## 53 Ressources et zones du DNS

55 Types de ressources du DNS

60 Numéro de série et arithmétique des intervalles

64 Format d'une zone

67 Routage de courrier électronique & DNS

## 66 Courrier électronique

70 La lutte anti-spam

73 Utilisation originale du DNS : la RBL

75 Sender Policy Framework (SPF)

76 Syntaxe SPF

78 Tolérance aux pannes

80 Réagir à des attaques de DDoS

82 Utiliser le DNS pour faire du DDoS ?

84 Gestion des adresses privées

86 Faire du CIDR sans pomme

90 Connaître la version d'un serveur DNS

91 Adresses de services

## Services

93 Spécificités MicroSoft

95 Stocker des numéros de téléphone

97 DDDS — découverte dynamique de délégation

99 Plan

## 98 Configuration client

100 Fichier /etc/resolv.conf

102 Fichier /etc/nsswitch.conf

103 Vers un resolver indépendant



104 Plan

## 103 Utilitaires d'interrogation

105 Programmes de mise au point

106 Outils en ligne

108 Programme dig

111 Programme host

113 Programme nslookup

117 Plan

## 116 BIND

118 BIND 9

121 Installation (Linux/Debian)

125 Installation sur BSD

126 Documentation de BIND

128 Compilation de BIND

129 Configuration de BIND

## 128 Configuration de BIND

130 Exemple de configuration

132 Inclusion de fichier include

133 Les options

136 Spécification de zone

140 Options pour un serveur distant server

141 Liste de contrôle d'accès acl

142 Différentes vues view

144 Enregistrement d'informations logging

147 Iwres

148 Sécurité & authentification key

149 Démarrage de BIND

## 148 Contrôle

151 Contrôle de NAMED : rndc

153 Protection de rndc

154 Autres outils venant avec BIND



155 Mise à jour dynamique de domaine

157 Plan

## 156 Génération des fichiers de zone

158 Comment générer les fichiers de zone ?

160 Utilisation de h2n

165 Limitations d'h2n

166 Exemple de mise en œuvre d'h2n

171 Quelques contributions livrées avec BIND

172 Plan

## 171 BIND et DNS avancé

173 Développements récents et futurs

174 IPv6

## 173 Introduction IPv6

175 Adresse IPv6

176 Interfaces et adresses IPv6

178 Allocation d'adresses globales

179 Exemples d'adresse dans 2000 ::/3

180 DNS & IPv6 : Vieille méthode (qui a gagné)

## 179 DNS & IPv6

182 DNS & IPv6 : nouvelle méthode (qui a perdu)

185 Internationalisation des noms de domaine

## 184 Internationalisation

186 Noms de domaine en UNICODE

187 Sécurisation du serveur lui-même

## 186 Sécurité

188 Sécurité

189 DNSSEC

190 TSIG

192 Les concepts de DNSSEC

195 Gestion des clés

197 Signature de l'être ou le néant

199 NSEC3 : signer l'être ou le néant sans fuite



- 201 Exemple de zone signée
- 205 Et les CNAME ?...
- 206 Distribution de certificats
- 207 Signature d'une zone
- 208 IPsec
- 207 IPsec**
- 210 Disponibilité d'IPsec
- 211 Composants d'IPsec
- 216 IPsec IKE
- 218 État d'IPsec

219 Chiffrement opportuniste

221 Conclusion

## 220 Conclusion

222 Plan

## 221 Travaux pratiques

223 Réaliser son propre DNS

224 Table des matières

\*

224

## 223 Table des matières