TP de création de serveurs de noms (DNS/BIND)

Ronan KERYELL
—
Département Informatique
TÉLÉCOM Bretagne

27-28 mars 2008
Version 1.14

Résumé

Ce TP couvre les principaux concepts de mise en place de serveurs de noms et leur utilisation.

La partie compilation et installation peut être sautée si le système d'exploitation fournit déjà BIND.

Cours et TP sont disponibles à : `http://enstb.org/~keryell/cours/IAR2M/BIND`. C'est pratique pour faire du couper-coller d'exemples ou de commandes[1].

Les exemples donnés sont à adapter en fonction du système d'exploitation et du lieu du TP. Lire en avance le texte et les exemples, cela peut aider.

`$Id: TP.tex,v 1.14 2008/03/28 16:36:53 keryell Exp $`

Table des matières

[1] Règle numéro 1 : on est informaticien par flemme... ☺

1 Exploration de quelques domaines avec les outils d'inspection

1.1 Outil d'interrogation

On s'initiera au maniement de l'outil `dig` dans la suite du TP car `nslookup` n'est plus maintenu et `host` est moins puissant.

En ce qui concerne les déclarations administratives on interrogera les bases `whois`.

Pour vérifier la conformité de zones entières on utilisera par exemple les outils :

- `http://zonecheck.fr` ;
- `http://www.dnsreport.com`.

1.2 Outils d'analyse réseau

On regardera les requêtes déclenchées lors du fonctionnement de l'ordinateur en utilisant les outils d'audit de réseau tels que `tcpdump` (`http://www.tcpdump.org`) en mode texte ou `wireshark` (`http://www.wireshark.com`) en mode graphique. `tshark` est la version texte de `wireshark` avec une syntaxe proche de `tcpdump`.

Pour `wireshark` on créera un filtre d'affichage en rajoutant la règle `DNS is present` pour s'y retrouver.

1.3 Zones à tester

Analyser les domaines suivants avec les outils précédents[2] :

- « `.` ». On pourra tester si les serveurs racines sont bien répliqués en *anycast* BGP, quels sont leur version ou leur vrai petit nom. Faire un `traceroute` dessus ;
- votre fournisseur d'accès favori, vos domaines habituels et autres domaines ;
- tester quelques uns parmi par exemple :
  - vos domaines préférés (travail, maison...) ;
  - `ensmp.fr.` (étudier la glu) ;
  - `cri.ensmp.fr.` ;
  - `enstb.org.` (étudier la glu) ;
  - `info.enstb.org.` (surprise !) ;
  - `trad.org.` ;
  - `us.` ;
  - `fr.` ;
  - `gouv.fr.` ;
  - `com.` ;
- regarder le contenu de `arpa` et ses sous-domaines (`in-addr`, `e164`,...) ;
- `172.54.192.in-addr.arpa` ;
- étudier le CIDR de `193.50.97` en comparant par exemple `193.50.97.3` et `193.50.97.147` ;
- contenu de la RBL ? Un des serveurs de `will-spam-for-food.eu.org` semble répondre au transfert de zone ;
- `10.in-addr.arpa` sur un serveur local et un serveur racine. Pourquoi ?
- dépioter une *lame delegation* et fournir un diagnostic.

Regarder plusieurs fois de suite un enregistrement sur un serveur faisant autorité et sur un autre (par exemple celui par défaut de l'école qui fait office de cache). Pourquoi a-t-on ce comportement ?

Bien comprendre la différence fondamentale entre les données qui sont dans le `dns` et celles qui sont dans les bases `whois`. Quelles sont-elles ?

[2] En fait à chaque `tp` il y a de moins en moins d'informations publiques, donc il est probable que de nombreuses choses ne marcheront plus... ☹

1.4 Étude de `resolv.conf`

Sous Solaris, on configurera le fichier pour utiliser `tcp` au lieu d'`udp` pour limiter les risques d'attaque (si la bibliothèque de résolution le permet).

Mettre son domaine perso et de travail dans `search` pour se simplifier la vie.

2 Installation de BIND

Créer un répertoire pour faire le TP, par exemple

```
mkdir -p ~/TP/DNS
cd ~/TP/DNS
```

ou encore dans `/junk`.

Si on est sous Debian ou assimilé :

```
aptitude update
aptitude install bind9
```

ou faire de même avec l'outil graphique `synaptic`.

Étudier le répertoire `/etc/bind` et sauter à la section suivante.

On va en profiter pour réviser l'installation d'un logiciel à partir des sources sous Unix. Outre l'intérêt pédagogique, cela permet d'avoir la version la plus fraîche.

On *pourrait* récupérer sur la page `http://www.isc.org/products/BIND/bind9.html` le fichier d'archive de la version 9. En fait, il est plus simple et plus économique en bande passante réseau d'utiliser la versions déjà récupérée dans `/usr/local/src/Reseau/bind-9.2.0.tar.gz`

Extraire les fichiers des archives dans un répertoire `bind` chez vous ou tout au moins là où vous trouverez de la place :

```
mkdir bind
cd bind
gtar zxvf /usr/local/src/Reseau/bind-9.2.0.tar.gz
```

`gtar` permet de gérer les fichiers d'archivage et en l'occurrence « `x` » spécifie l'extraction de contenu, « `v` » indique que l'on veut de l'information verbeuse sur ce qui est fait et « `f` » précise le nom du fichier d'archive. Comme le fichier d'archive est comprimé (extension du fichier typiquement « `.gz` » ou « `.Z` » l'option « `z` » demande la décompression au vol, « `gtar zxvf` *`fichier`* » est un raccourci pour « `gunzip -c` *`fichier`* `| gtar xvf -` » où on passe par `stdout` de `gunzip` et `stdin` de `gtar` respectivement.

Normalement BIND installe ses fichiers de bibliothèque et d'inclusion respectivement dans `/usr/local/bind/lib` et `/usr/local/bind/include` pour ne pas entrer en conflit avec les fichiers d'origine de Solaris qui sont dans `/usr` au lieu de `/usr/local`. Comme dans le mastère les machines clients ne peuvent pas écrire dans `/usr/local`[3], les fichiers iront sur les disques locaux dans `/usr/bind`. Pour ce faire, lancer la compilation comme suit :

```
cd bind-9.2.0
./configure --prefix=/usr
```

Les noms de fichier par défaut ne sont pas terrible et il faut peut-être les revoir ?

```
make
```

Dans ce TP certaines choses sont faites avec vos droits, d'autres avec le droit de `root`. Afin de faciliter les manipulations, il est conseillé d'avoir plusieurs fenêtres dont une a les droits de `root`. Un `su miam` permet d'être `root` localement sur votre machine.

Dans une fenêtre sous `root` dans le même répertoire faire un

[3] Cela causerait de toute manière des conflits en écriture si toutes les machines écrivait en même temps dans ce répertoire...

```
make install
```

Préparer la visualisation de la documentation de configuration avec un `netscape` sur le fichier `doc/arm/Bv9ARM.html` ou un `acroread` sur `http://www.nominum.com/resources/documentation/Bv9ARM.pdf`.

Si ce n'est pas fait, créez les répertoires manquant :

```
mkdir -p /usr/etc /usr/var/run
```

et créez une clé secrète pour `rndc` avec :

```
rndc-confgen -a
```

3 Essais de named

Pour avoir une vue sur les (éventuels ?...) messages d'erreur, lancer dans une fenêtre séparée un

```
tail -f /var/log/syslog
```

ou

```
tail -f /var/log/messages
```

ou (Solaris)

```
tail -f /var/adm/messages
```

(selon le système d'exploitation) qui aura pour effet d'afficher en permanence les messages du système rajoutés à ce fichier par le démon `syslogd`. Le mode Emacs `auto-revert-mode`[4] est bien pratique aussi pour ce faire.

Tester sous les droits de `root`[5] dans le répertoire `/etc/named` l'exécution de `named` avec

```
named -c named.conf
```

pour utiliser le fichier local `named.conf` au lieu de `/etc/named.conf` et regarder au passage les messages dans la fenêtre sur `/var/adm/messages`. C'est à adapter en fonction de votre système d'exploitation... Par exemple sous Linux/Debian un

```
/etc/init.d/bind9 restart
```

fera l'affaire, mais sous Linux/RedHat c'est plutôt :

```
/etc/init.d/named restart
```

Utiliser `dig` pour interroger le serveur de nom. Ne pas oublier de préciser qu'on utilise le serveur local et non `chailly` ou un autre par défaut qui serait précisé dans le `resolv.conf`.

Mettre le serveur en mode trace et regarder comment sont faites les requêtes.

Pour faire marcher la résolution il faut connaître au moins un serveur racine comme vu en cours. Si cela n'est pas fait automatiquement sur le système où vous êtes, installer un fichier de serveur racine `~keryell/db.cache` dans `/etc/named`, arrêter `named` avec un « `rndc stop` » (NameD Control) ou un `kill` et relancer `named` puis réessayer avec `dig`.

Étudier le contenu du cache du serveur de nom.

On peut faire tourner en parallèle un `wireshark` pour voir les échanges protocolaires.

Si vous utilisez `rndc querylog`, les messages vont dans `/var/cache/bin/named.run`.

[4] Rappel : taper `M-x auto-revert-mode` pour passer par exemple dans ce mode.

[5] Car le DNS utilise un port privilégié (53) et implique que seul `root` a le droit d'écouter sur ce type de ports. Voyez-vous la raison de l'existence de ports privilégiés ? Cela n'existe pas sur NT... Conséquences ?

4 Installation de h2n

Le plus simple : récupérer h2n dans http://enstb.org/~keryell/cours/IAR2M/BIND et sauter à la section suivante.

Sous FreeBSD c'est simple :

```
portinstall h2n
```

Sinon, récupérations des outils du livre *DNS and BIND* de chez O'Reilly à l'adresse http://examples.oreilly.com/dns4/dns.4ed.tar.Z ou utiliser une version déjà récupérée dans /usr/local/src/Reseau le cas échéant.

Décomprimer l'archive avec par exemple

```
cd ~/TP/DNS
mkdir h2n
cd h2n
tar zxvf /usr/local/src/Reseau/dns.4ed.tar.Z
```

Comme h2n est un perl-script il faut lui préciser l'emplacement exact de perl sur le système en modifiant éventuellement la première ligne du fichier h2n

```
#!/usr/bin/perl
```

en

```
#!/usr/local/bin/perl
```

le cas échéant. Sous Debian ce n'est pas la peine.

Installer le nécessaire pour h2n avec les droits de root en exécutant

```
cp h2n /usr/bin
cp h2n.man /usr/share/man/man1m/h2n.1m
```

À adapter en fonction des coutumes du système d'exploitation.

Effacer le répertoire h2n et son contenu avec

```
cd ..
rm -rf h2n
```

5 Génération de fichiers de configuration

Pour simplifier, sous Debian on travaille directement sous /var/cache/bind.

Sinon, créer un répertoire /etc/named ou utiliser le répertoire équivalent standard pour votre système d'exploitation, par exemple /etc/bind sous Debian et y aller.

Créer une zone *mon-domaine*.com ultra simple à la main et déclarez-là (dans /etc/bind/named.conf.local sous Debian).

Ensuite, faire une zone plus conséquente en utilisant h2n. On peut remplir cette zone en utilisant le fichier /etc/hosts des Mines dont une copie se trouve (peut-être...) dans ~keryell/hosts ou le /etc/hosts du RIRE et utiliser son propre fichier ensuite si on a du courage. Par exemple, si on veut y mettre les machines du réseau 10.2.16 (privé) et 193.50.97, faire

```
h2n -v 8 -b named.conf.h2n -w -t -y -H ~keryell/hosts -d
mon-domaime -n 193.50.97 -n 10 -o 60:60:60:60
```

histoire de ne pas écraser le vrai fichier named.conf.

Étudier la structure des fichiers générés.

6 Utilisation définitive de named

Sauvegarder une copie de `/etc/resolv.conf` avant de le modifier pour mettre

```
nameserver      127.0.0.1
search          mon-domaine ensmp.fr
```

pour chercher les noms non complètement qualifiés d'abord dans *mon-domaine*[6] puis dans `ensmp.fr`.

Vérifier que `dns` apparaît en tête dans la ligne `hosts` de `/etc/nsswitch.conf`.

Faire un `/etc/init.d/bind9 restart` (puisque là on peut utiliser le fichier de configuration par défaut de `named`.

Demander l'inscription de son domaine dans `enstb.org` et regarder le contenu de la zone `enstb.org` sur le serveur `dns2.enstb.org` pour voir comment c'est fait.

Le nom de domaine mondial deviendra donc *mon-domaine*`.enstb.org`.

7 Délégations et serveurs secondaires

Relire le `man` de `h2n`.

Créer un(des) serveur(s) secondaire(s) de son domaine chez un (des) voisin(s).

Vérifier que cela fonctionne depuis des machines tierces, que les notifications de mise à jour son bien effectuées.

Créer des sous-domaines qui seront délégués chez soi et/ou chez d'autres.

Essayer les *stubs*.

8 Intranet et Extranet

Utiliser les vues pour limiter l'information du DNS vers le tout internet. On choisira arbitrairement les machines à mettre dans l'extranet et l'intranet.

9 Grandeur et décadence du protocole DNS

De nombreux protocoles d'Internet comme le DNS ont été conçus à une époque où quand quelque chose marchait s'était déjà bien et les gens étaient gentils...

Avec les développements d'Internet et l'augmentation des puissances des ordinateurs pour faire des attaques, c'est problématique... ☹

On va mettre le protocole à l'épreuve en utilisant la boîte à outil de démonstration `dsniff`, en particulier avec l'outil `dnsspoof` pour rediriger par exemple des interrogations d'un site WWW vers un autre. Utilisation pratique : suppression d'images de publicité de certains sites ! ☺

Si vous êtes sur un réseau avec un commutateur (*switch*), vous ne voyez pas normalement passer les paquets réseaux ne vous concernant pas. Dans ce cas on peut utiliser en plus `macof` ou `arpspoof` pour détourner le trafic IP vers votre machine malgré le commutateur.

Imaginer ce type d'attaque avec des mandataires (*proxy*) du style de celui disponible à `http://anon.free.anonymizer.com` mais bien méchants qui modifieraient ou enregistreraient les contenus, avec un bug ou du JavaScript pour modifier la ligne affichant l'URL visitée... ☹ Des exemples sont donnés dans des présentations du `http://www.clusif.asso.fr`.

On pourrait imaginer des attaques plus offensives avec génération de réponses DNS en aveugle avec forgeage des adresses de source des paquets dans les réponses.

Clairement cela motive à utiliser des protocoles plus sécurisés comme IPsec, DNSSEC, SSH et TLS (attention néanmoins aux subtilités de ces 2 derniers...).

[6] Je tiens à préciser que c'est un nom symbolique que vous devez remplacer par quelque chose qui vous est propre...

10 DNSSEC

Mettre en place une délégation de votre domaine que vous allez authentifier fortement en déployant DNSSEC. La clé publique de votre domaine servira de racine ou bien demander une signature par votre domaine parent (`enstb.org`).

11 Remise en état

À la fin du TP il faudrait remettre en état `/etc/resolv.conf` pour que l'utilisation du DNS redevienne comme auparavant.