DNS

Le Domain Name System, généralement abrégé DNS, qu'on peut traduire en « système de noms de domaine », est le service informatique distribué utilisé pour traduire les noms de domaine Internet en adresse IP ou autres enregistrements.

Présentation

Qu’est-ce que le DNS ?

Le **DNS**, ou *Domain Name System*, est un système indispensable au fonctionnement d’Internet. Son rôle est de traduire les **noms de domaine** que l’on utilise tous les jours (comme `google.com`, `rakouns.bzh`, ou `wikipedia.org`) en **adresses IP** compréhensibles par les machines.

En effet, les ordinateurs, les serveurs et tous les équipements réseau communiquent entre eux à l’aide d’adresses IP. Mais pour les humains, il serait difficile de mémoriser une série de chiffres pour chaque site. Grâce au DNS, on peut simplement taper un nom lisible dans le navigateur, et celui-ci sera automatiquement converti en adresse IP en arrière-plan.

À quoi ça sert ?

Le DNS joue un rôle de **répertoire téléphonique** d'Internet. Lorsqu'on entre une adresse web dans un navigateur, ce dernier interroge un serveur DNS pour demander : "À quelle adresse IP correspond ce nom de domaine ?"

Le serveur répond avec l'adresse IP du site, et le navigateur peut ensuite établir la connexion. Ce processus est invisible pour l'utilisateur, mais il se produit à chaque requête, souvent en quelques millisecondes.

Sans DNS, il faudrait connaître et entrer manuellement l'adresse IP de chaque site ou service en ligne. Ce serait non seulement fastidieux, mais quasiment inutilisable à grande échelle.

Comment ça fonctionne, en pratique ?

Dès qu'un appareil tente de se connecter à un nom de domaine, il envoie une **requête DNS** à un serveur spécifique. Ce serveur peut être :

- celui fourni par la box ou le fournisseur d'accès Internet,
- un service public comme **Cloudflare (1.1.1.1)** ou **Google DNS (8.8.8.8)**,
- ou un serveur DNS interne, dans les réseaux professionnels.

Si le serveur a déjà la réponse en mémoire (ce qu'on appelle le cache DNS), il répond immédiatement. Sinon, il interroge d'autres serveurs dans une chaîne hiérarchique, jusqu'à obtenir une réponse fiable.

Une fois l'adresse IP récupérée, elle est utilisée pour établir la communication avec le site demandé.

Et dans un réseau local ?

Dans un réseau d'entreprise, il est courant d'utiliser un **serveur DNS interne**. Celui-ci peut résoudre des noms comme `serveur-fichiers.local` ou `intranet.entreprise` en adresses IP locales. Cela facilite la gestion du parc informatique, la maintenance des services, et améliore la lisibilité pour les utilisateurs et les administrateurs.

Le DNS interne peut aussi être couplé à d’autres services comme **Active Directory**, pour gérer dynamiquement les machines et les comptes utilisateurs dans un environnement Windows.

Le DNS, un maillon critique

Parce qu’il intervient dans toutes les connexions, le DNS est aussi une **cible privilégiée des attaques**. Il existe des solutions pour sécuriser son usage, comme :

- le **DNSSEC**, qui garantit l’intégrité des réponses,
- le **DoH (DNS over HTTPS)** ou **DoT (DNS over TLS)**, qui chiffrent les requêtes pour éviter qu’elles soient espionnées.

Certaines solutions comme **AdGuard Home** ou **Pi-hole** utilisent aussi le DNS comme point de contrôle, pour bloquer les publicités, les malwares ou les services indésirables.

En résumé

Le DNS est un service invisible mais essentiel. Il rend le web lisible, accessible et fluide pour les utilisateurs, tout en restant extrêmement puissant et flexible pour les administrateurs réseau. Chaque fois qu’on accède à un site, qu’on envoie un mail ou qu’on utilise une application connectée, le DNS est là, en arrière-plan, pour faire le lien entre les noms et les adresses.

Mise en place

Installation

Nous allons commencer par mettre à jour le cache des paquets et procéder à l'installation du paquet Bind9, ainsi qu'un autre paquet permettant d'obtenir des outils DNS supplémentaires.

```
apt-get update
apt-get install bind9 dnsutils
```

Fichiers

La liste des fichiers de configuration :

```
ls -l /etc/bind
```

```
flo@srv-dns:~$ ls -l /etc/bind
total 48
-rw-r--r-- 1 root root 2403 27 juil. 05:13 bind.keys
-rw-r--r-- 1 root root  255 27 juil. 05:13 db.0
-rw-r--r-- 1 root root  271 27 juil. 05:13 db.127
-rw-r--r-- 1 root root  237 27 juil. 05:13 db.255
-rw-r--r-- 1 root root  353 27 juil. 05:13 db.empty
-rw-r--r-- 1 root root  270 27 juil. 05:13 db.local
-rw-r--r-- 1 root bind  458 27 juil. 05:13 named.conf
-rw-r--r-- 1 root bind  498 27 juil. 05:13 named.conf.default-zones
-rw-r--r-- 1 root bind  165 27 juil. 05:13 named.conf.local
-rw-r--r-- 1 root bind  846 27 juil. 05:13 named.conf.options
-rw-r----- 1 bind bind  100  4 déc.  10:07 rndc.key
-rw-r--r-- 1 root root 1317 27 juil. 05:13 zones.rfc1918
flo@srv-dns:~$
```

Par défaut, ce répertoire contient déjà un ensemble de fichiers de configuration. Vous devez savoir que :

- Les fichiers "**db.<nom>**" correspondent aux fichiers de zones intégrés par défaut dans **Bind**. Vous pouvez vous en inspirer en tant que modèle pour la création de vos fichiers de zones.
- Le fichier "**named.conf**" est le fichier de configuration principal de Bind9. Il contient des directives "**include**" pour charger 3 autres fichiers :
  - "**named.conf.options**" contient les options de configuration de Bind
  - "**named.conf.local**" sert à déclarer des zones
  - "**named.conf.default-zones**" contient la définition des zones incluses par défaut avec Bind.

ce qui donne :

```
flo@srv-dns:~$ cat /etc/bind/named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
flo@srv-dns:~$
```

Configuration

Comme je l'ai dit précédemment, les options de configuration de Bind sont définies dans le fichier "**named.conf.options**". Voici un aperçu de ce fichier, dans sa configuration par défaut :

```
flo@srv-dns:~$ cat /etc/bind/named.conf.options
options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

        // forwarders {
        //      0.0.0.0;
        // };

        //========================================================================
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
        //========================================================================
        dnssec-validation auto;

        listen-on-v6 { any; };
};
flo@srv-dns:~$
```

On peut voir qu'il y a beaucoup de lignes commentées (celles qui débutent par "//"), ainsi que plusieurs options définies par défaut. Nous allons devoir ajuster cette configuration.

```
nano /etc/bind/named.conf.options
```

Dans ce fichier, vous allez devoir définir plusieurs options, notamment pour permettre la résolution des noms externes, via les DNS publics de Cloudflare (1.1.1.1) et Quad9 (9.9.9.9), que vous pouvez remplacer par d'autres IP. Il convient aussi de mettre en écoute le serveur DNS.

Voici la configuration commentée que vous pouvez utiliser :

```
options {
        // Répertoire de travail de Bind
        directory "/var/cache/bind";

        // Redirecteurs DNS (résolveurs externes)
        forwarders {
                1.1.1.1;
```

```
                9.9.9.9;
        };

        // Mode récursif, pour résoudre les noms externes
        recursion yes;

        // Active la validation DNSSEC (vérifier l'authenticité des réponses DNS signées)
        dnssec-validation auto;

        // Ecouter sur toutes les interfaces réseau en IPv4 et IPv6
        listen-on { any; };
        listen-on-v6 { any; };
};
```

Pour aller plus loin, nous pouvons **définir une ACL** (règle d'accès) pour indiquer que **seules les machines du LAN peuvent contacter ce serveur DNS**. Ainsi, nous autorisons "**192.168.1.0/24**", le serveur lui-même (**localhost**) et le réseau auquel il est connecté, soit "**192.168.14.0/24**" (**localnets**).

Cette ACL doit être déclarée avant le bloc "**options**" :

```
// Autoriser uniquement certains réseaux à solliciter ce DNS
acl "lan" {
        192.168.1.0/24;
        localhost;
        localnets;
};
```

Puis, dans le bloc "**options**", à la suite des directives "**listen-on**" mais avant la fermeture du bloc, ajoutez ceci :

```
        // Autoriser les requêtes pour les hôtes de l'ACL "lan"
        allow-query { lan; };
```

Quand c'est fait, enregistrez le fichier. Vous pouvez le fermer et exécuter la commande ci-dessous pour vérifier la syntaxe :

```
named-checkconf
```

Déclarer une zone DNS

Nous allons devoir déclarer notre nouvelle zone DNS. Pour cela, éditez ce fichier de configuration :

```
nano /etc/bind/named.conf.local
```

Puis, ajoutez le code suivant :

```
zone "domaine.local" {
    type master;
    file "/etc/bind/db.domaine.local";
    allow-update { none; };
};
```

Pour rappel, nous allons créer la zone DNS "**domaine.local**" et le fichier de zone sera "**/etc/bind/dbdomaine.local**". L'instruction "**allow-update { none; };**" permet de refuser les mises à jour des enregistrements DNS par un tiers non autorisé.

Quand c'est fait, enregistrez et fermez le fichier.

Désormais, vous allez copier le fichier "**db.local**" pour l'utiliser comme base pour votre nouvelle zone :

```
cp /etc/bind/db.local /etc/bind/db.domaine.local
```

Quand c'est fait, vous pouvez passer à l'édition du fichier de zone.

Configurer la zone DNS

Nous allons modifier le fichier de zone pour le configurer et **créer nos premiers enregistrements DNS**. Nous verrons **comment créer un enregistrement A**, ainsi qu'un **alias CNAME.**

Commencez par ouvrir le fichier de zone :

```
nano /etc/bind/db.domaine.local
```

Après modifications, voici le fichier de zone "domaine**.local**" prêt à l'emploi. Il permet de déclarer le serveur local, à savoir **SRV-DNS**, comme serveur faisant autorité sur la zone. Nous déclarons également un enregistrement A avec l'adresse IP du serveur DNS, à savoir "**192.168.14.99**".

```
; BIND data file for domaine.local
$TTL    604800
@       IN      SOA     srv-dns.domaine.local. admin.domaine.local. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
```

```
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@               IN      NS      srv-dns.domaine.local.
srv-dns         IN      A       192.168.14.99
```

Quelques explications supplémentaires :

- "**$TTL 604800**" correspond à la durée de vie des informations fournies et donc la durée pendant laquelle elles sont gardées en cache par les autres serveurs **DNS**. Par défaut, ce temps est défini sur 24 heures (86400 secondes).
- **@** : désigne la racine de la zone, c'est-à-dire **domaine.local**.
- "**SOA**" signifie **Start Of Authority**, soit les paramètres principaux de la zone. Il indique le serveur qui a autorité sur la zone, puis l'adresse e-mail du contact technique dont le caractère « **@** » est remplacé par un «.». La valeur "**srv-dns.domaine.local.**" sert à indiquer le serveur DNS primaire (le point final indique un FQDN complet).
- **Serial "1"** : numéro de série de la zone. À incrémenter chaque fois que le fichier de zone est modifié pour notifier les serveurs secondaires d'une mise à jour.
- **Refresh "604800"** : c'est le délai de rafraichissement pour la synchronisation des configurations entre plusieurs serveurs **DNS**.
- **Retry "86400"** : c'est le délai au bout duquel un serveur **DNS** secondaire devra retenter une synchronisation si celle qu'il a faite au bout du temps "**refresh**" a échoué.
- **Expire "2419200"** : si toutes les tentatives de synchronisation échouent, un serveur **DNS** secondaire considérera qu'il ne peut plus répondre aux requêtes concernant cette zone une fois que le temps est écoulé. Par défaut, le temps est de "**2419200**" secondes, soit 28 jours.
- **Negative Cache TTL "86400"** : durée de conservation dans le cache de l'information "**NXDOMAIN**" lorsqu'un incident se produit (échec de résolution).

Créer un enregistrement DNS

Nous allons voir comment créer un enregistrement A et un enregistrement CNAME dans cette nouvelle zone. Voici quelques instructions sur la syntaxe à respecter.

- Dans les zones de recherche directes pour les enregistrements « **A** » :

```
<nom-de-l'hote>    IN   A    <IP>
```

- Dans les zones de recherche directes pour les enregistrements « **CNAME**» :

```
<nom-de-l'alias> IN CNAME <nom-de-l'enregistrement-de-référence>
```

À partir de ces informations, ajoutons 2 enregistrements DNS :

- Un alias "**dns.domaine.local**" pour le serveur "**srv-dns.domaine.local**"

- Un nom d'hôte "**srv-dhcp.domaine.local**" associé à l'adresse IP "**192.168.14.98**"

Ainsi, dans le fichier de zone, il convient d'ajouter ceci :

```
dns             IN      CNAME   srv-dns
srv-dhcp        IN      A       192.168.14.98
```

Désormais, testez sa syntaxe avec la commande "**named-checkzone**" :

```
named-checkzone domaine.local /etc/bind/db.domaine.local
zone domaine.local/IN: loaded serial 2
OK
```

Démarer Bind9

La configuration est terminée, nous allons démarrer notre serveur Bind9 et activer son démarrage automatique. Exécutez les commandes suivantes :

```
systemctl start bind9
systemctl enable named.service
systemctl status bind9
```

Tester la résolution de nom

Sur le serveur DNS lui-même, vous pouvez modifier la configuration réseau pour qu'il sollicite son propre résolveur DNS local pour la résolution des noms. Vous devez modifier le fichier de configuration "**resolv.conf**" (ou passer par Netplan).

```
nano /etc/resolv.conf
```

Puis, indiquez ceci :

```
search domaine.local
domain domaine.local
nameserver 127.0.0.1
```

ermez le fichier. Désormais, vous pouvez tenter de résoudre les noms de la zone "**domaine.local**".

Pour cela, l'outil "nslookyp" sera très utile. Précisez simplement le nom à résoudre et il va solliciter le DNS pour obtenir l'information. Vous pouvez faire plusieurs tests, tels que :

```
nslookup srv-dns.domaine.local
nslookup dns.domaine.local
nslookup srv-dhcp.domaine.local
```

Tout fonctionne à merveille !

Créer une zone inverse

Pour finir, nous allons créer une zone de recherche inversée pour le réseau "192.168.14.0/24", correspondant au réseau local utilisé pour cette mise en pratique. Ceci permettra d'obtenir un nom d'hôte à partir d'une adresse IP, soit l'inverse du fonctionnement d'une zone de recherche directe.

Sur le même principe que pour la zone de recherche directe, nous allons créer cette fameuse zone.

Commencez par éditer le fichier "**named.conf.local**" pour déclarer la zone :

```
sudo nano /etc/bind/named.conf.local
```

Voici la déclaration de la zone inversée :

```
zone "14.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.reverse.domaine.local";
    allow-update { none; };
};
```

Vous pouvez remarquer que le nom de la zone est « **14.168.192.in-addr.arpa** », tout d'abord, on indique, dans l'ordre inverse, les 3 octets de l'adresse IP de la zone représentant le réseau, donc pour le réseau « **192.168.14.0** » cela donnera « **14.168.192** ». Ensuite, nous ajoutons « **in-addr.arpa** » qui est un espace de noms réservé et utilisé mondialement pour la résolution inverse.

Enregistrez et fermez le fichier.

Copiez le fichier de la zone "**domainelocal**" pour l'utiliser comme base. Le fichier de la zone inverse sera "**db.reverse.domaine.local**". Puis, éditez ce fichier.

```
sudo cp /etc/bind/db.domaine.local /etc/bind/db.reverse.domaine.local
sudo nano /etc/bind/db.reverse.domaine.local
```

Voici le contenu du fichier de zone inversé :

```
; BIND data file for 14.168.192.in-addr.arpa
$TTL    604800
@       IN      SOA     srv-dns.it-domaine. admin.domaine.local. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@               IN      NS      srv-dns.domaine.local.
99      IN      PTR     srv-dns.domaine.local.
98      IN      PTR     srv-dhcp.domaine.local.
```

Vous remarquerez l'absence d'enregistrement A, AAAA ou encore CNAME. Cela n'existe pas dans une zone inversée. À la place, nous utilisons des enregistrements PTR (pointeur) où l'on indique l'adresse IP de l'hôte (dernier octet, ici), et à droite, le nom d'hôte.

Quand c'est fait, enregistrez et fermez le fichier. Vérifiez la syntaxe du fichier :

```
named-checkzone 14.168.192.in-addr.arpa /etc/bind/db.reverse.domaine.local
```

Si tout est OK, relancez Bind9 sur la machine :

```
sudo systemctl restart bind9
```