Chapitre 2
Sécurisation d'annuaires

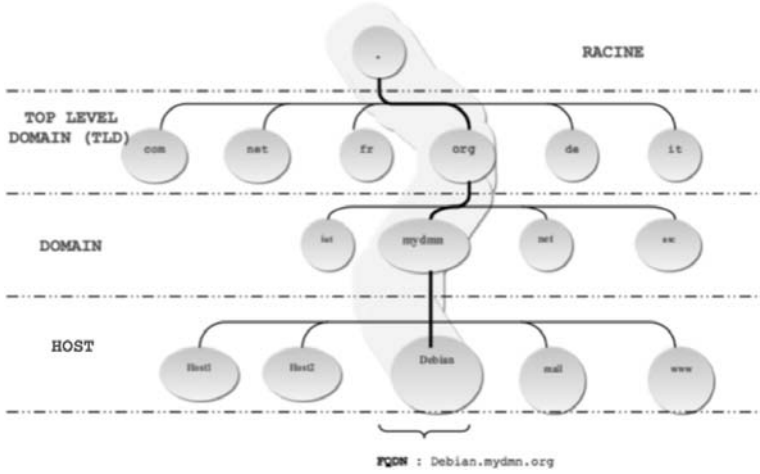
1. Sécurisation de l'annuaire de noms DNS

Dans la mesure où nombre d'applications, comme nous l'avons vu précédemment, ont recours à la résolution de noms pour reconnaître les serveurs sous-jacents, on peut alors s'interroger sur le degré de sécurité souhaité en ce qui concerne ce service bien particulier, faisant transiter des informations critiques au travers d'Internet (adresses IP et noms FQDN). La fonctionnalité BIND (***B**erkeley **I**nternet **N**ame **D**aemon* aussi appelée ***B**erkeley **I**nternet **N**ame **D**omain*) est le serveur de résolution de noms DNS le plus utilisé sur Internet, spécialement sur des serveurs Unix/Linux. C'est donc devenu le standard de facto de la résolution DNS et il appartient donc aux applications couramment utilisées.

RAPPEL : la première version de BIND a été conçue par quatre étudiants diplômés de l'université de Berkeley en Californie, sur la base du système d'exploitation BSD 4.3. En 1988, c'est Paul VIXIE qui reprend la maintenance du projet. Actuellement, BIND est développé par l'***I**nternet **S**ystem **C**onsortium* (ISC).

1.1 Généralités sur le serveur de noms

Un serveur de noms (abrégé en DNS pour ***D**omain **N**ame **S**erver*) est un annuaire à structure arborescente inversée. La racine est généralement représentée en haut et déploie ses branches vers le bas. Son rôle est de faciliter l'accès aux systèmes disposant d'adresses IP et leur permettre, grâce à un mécanisme d'association, de récupérer le nom des serveurs. La racine est alors représentée par un ".", on trouve, en dessous, les domaines de haut niveau (aussi appelés *top-level domains*) : .fr, .com, .org, .eu... Le parcours s'effectue donc à l'envers depuis la racine, vers le bas (la machine ou le serveur référencé) :



De façon intégrée, le serveur de noms possède une configuration particulière pour les routeurs de courrier électronique (notée généralement MX), autorisant une résolution inverse de celle effectuée par les hôtes classiques, ainsi qu'un mécanisme autorisant un facteur de priorité et une tolérance aux pannes. En résumé, le serveur de noms est conçu pour pallier les défaillances du fichier système `/etc/hosts` et doit permettre de répondre ainsi aux impératifs de conception et d'architecture que sont :

– L'aspect dynamique : les enregistrements doivent être manipulés de façon unique au sein du réseau et sont accessibles de tous.

– L'aspect réplication : il est nécessaire de dupliquer les informations pour ne pas avoir de point de contention.

– L'aspect hiérarchie : le fait d'architecturer un DNS en arbre permet une meilleure organisation. Chaque niveau est appelé « zone » et le sommet de l'arbre est noté ".".
– L'aspect répartition : l'arbre DNS est distribué. Les informations sont réparties sur une multitude de zones et l'ensemble de ces bases d'informations compose l'intégralité des enregistrements DNS. Cela permet d'envisager la répartition de charge avec plus de facilité.
– L'aspect sécurité : suite à des attaques multiples sur l'annuaire lui-même, les informations (socle de cette architecture) ayant été corrompues, il a fallu trouver un moyen de protéger ces enregistrements. On peut ainsi mettre en place un système d'authentification, de contrôle d'accès et d'intégrité.

Les résolutions de noms (ou d'adresses) s'effectuent au travers d'un équipement (serveur, ou autre), appelé résolveur. Sans pour autant résoudre tous les problèmes de sécurité, liés au protocole BIND, il convient, malgré tout, de déporter le service DNS, au sein d'une arborescence dédiée, afin de réduire le risque d'utilisation d'une faille de sécurité, par un tiers malveillant. Bien évidemment, il faut aussi utiliser un compte différent de celui du super-utilisateur afin d'empêcher la possibilité, à un pirate, de disposer des droits d'administration renforcés du compte `root`.

1.2 Attaques visant le serveur de noms

Les attaques sur les différents équipements impliquant le serveur DNS sont plutôt d'ordre technique et font, le plus souvent, référence à des stratégies d'attaques massives ou à des corruptions d'enregistrements échangés entre les résolveurs et les serveurs DNS. On dénombre principalement quatre catégories d'attaques directes :

– **Attaque par empoisonnement** : l'attaquant « intoxique » le résolveur pour forcer la machine de l'attaquant à être reconnue comme valide, au lieu de la machine d'origine. Ce genre de détournement vise essentiellement à capter les requêtes DNS vers un autre site, sans que les utilisateurs locaux puissent s'en rendre compte.
– **Déni de service** (on parle aussi de ***D**enial **o**f **S**ervice* que l'on note DoS) : le pirate rend un service distant inactif (voire inaccessible), en saturant la machine hébergeant le service, de requêtes.

Remarque

*Une forme plus élaborée du Denial of Service appelée **d**istributed **D**enial **o**f **S**ervice (et noté `dDoS`), utilise le même principe, mais en s'appuyant, non plus sur la seule machine du pirate, mais sur l'ensemble des machines du réseau Internet. Cela s'apparente à un ro**BOT NET**work (aussi abrégé en BOTNET).*

– Réflexion : l'attaquant émet de nombreuses requêtes au nom de sa victime. Lorsque les destinataires répondent, ils contribuent ainsi à saturer les infrastructures de l'entreprise visée, à cause de la convergence des messages.

Remarque

Combinée à la réflexion, l'amplification permet de ralentir les résolutions de noms et par là même, les performances des serveurs de noms DNS.

– *Fast flux* : le pirate falsifie son adresse IP (pour ne pas être démasqué), tout en s'appuyant sur la rapidité de diffusion des informations de localisation (pour éviter d'être localisé trop rapidement). Il existe de nombreuses variantes : simple flux (où l'adresse du serveur web change périodiquement et régulièrement), double flux (où les adresses des serveurs web et DNS changent).

1.3 Recommandations générales

Au sein d'une architecture d'entreprise, il convient donc de sécuriser les différents maillons de l'application DNS, d'abord dans sa globalité : au niveau architectural, mais également au niveau fonctionnel : les flux d'informations circulant entre les échelons de l'arborescence DNS. Voici donc quelques conseils et préconisations fondamentaux de sécurité concernant la résolution de noms.

Opter pour la redondance

Pour qu'en cas d'attaque, le service de noms continue de fonctionner et que le serveur visé puisse être changé ou remplacé de façon transparente, il vaut mieux installer le nom de domaine, au minimum sur deux serveurs identiques. La plupart du temps, on installe un premier serveur appelé maître, qu'il suffit ensuite de cloner pour permettre de répartir la charge et les services sur ce second équipement, appelé esclave.

Veiller à tenir à jour la version

Ce qui est valable pour le système d'exploitation l'est a fortiori pour une de ses applications : le DNS. Il faut donc toujours s'assurer d'appliquer les dernières versions du protocole BIND et/ou les derniers patchs, en vue de ne pas être inquiété par les vulnérabilités et les failles potentielles des anciennes versions.

Effectuer une surveillance accrue

Cette recommandation va de pair avec la précédente. Il s'agit de ne pas faire entièrement confiance à la robustesse de l'architecture redondée, mais de surveiller régulièrement les serveurs de noms et leur configuration (ainsi que leurs performances). Cette vérification peut être faite par script, via un programme tiers (comme ZoneCheck) ou depuis l'extérieur, en faisant appel à un organisme payant et accrédité.

Sécuriser les flux d'échanges

La meilleure façon de protéger un serveur de noms est de chiffrer les échanges entre clients, résolveurs et serveurs de noms. Il existe, à cet égard, un protocole de sécurisation autorisant l'authentification des serveurs et empêchant les attaques de type empoisonnement. Il s'agit du protocole DNSSEC.

Prévoir un plan de reprise

Nous avons déjà traité ce sujet dans le livre Debian GNU/Linux - Maîtrisez la sécurité des infrastructures, chapitre Outils de sauvegarde. Mais il paraît évident qu'il ne faut surtout pas négliger le plan de reprise d'activité en cas de grosses catastrophes : incendie, inondation, destruction... De plus, ce genre de politique présente aussi un aspect financier, car les serveurs à provisionner sont à prévoir dans les budgets d'entreprise et à installer dans un plan d'adresses différent de celui de la production.

1.4 Mise en œuvre d'un serveur DNS simple

Après avoir évoqué quelques éléments d'architecture et de sécurité, il est possible d'installer le package `bind9` sur Debian, de la façon suivante :

```
# apt-get update
# apt-get install bind9
```

Il faut veiller à modifier et/ou créer les trois fichiers de configuration suivants pour pouvoir paramétrer son serveur DNS comme on le souhaite :

– Le fichier de configuration principal : `/etc/bind/named.conf.local`
– Le fichier de zone : `/etc/bind/db.mydmn.org` (référencé dans `named.conf.local`)
– Le fichier de la zone reverse : `/etc/bind/db.mydmn.org.rev` (référencé dans `named.conf.local`)

Remarque

Le fichier `named.conf` n'est en fait qu'un regroupement d'inclusions de fichiers. Il est fortement déconseillé de le modifier. Il vaut mieux ajouter les zones dans le fichier `named.conf.local`.

En ce qui concerne le fichier `named.conf.local`, il faut déclarer la liste des zones (ou domaines), que le serveur DNS va devoir prendre en charge :

```
zone "mydmn.org" {
    type master;
    file "/etc/bind/db.mydmn.org";
    forwarders{};
};
```

Dans cet exemple, le nom de zone est `mydmn.org`, son fichier associé de paramétrage se trouve dans `/etc/bind` et se nomme `db.mydmn.org`. En toute logique, il faut configurer la zone "reverse" de la même manière :

```
zone "1.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.mydmn.org.rev";
    forwarders{};
};
```

En lieu et place du nom de la zone, il suffit de positionner l'adresse réseau inverse de 192.168.1, soit 1.168.192 en ajoutant `.in-addr.arpa`. Nous pouvons alors générer le fichier principal de la zone `mydmn.org`, en déclarant l'autorité de résolution comme suit :

```
$TTL  604800
@ IN SOA debian.mydmn.org. root (
    2017100725           ; n° Serial
    604800                    ; Refresh
    86400                      ; Retry
    2419200                 ; Expire
    604800)                   ; Minimum
    IN NS     debian.mydmn.org.             ; Nom du serveur DNS
debian      IN A      192.168.1.251                    ; Adresse du DNS
debian      HINFO   "MyDNS" "Debian Jessie Beta1" ; Infos du DNS
```

Avant d'aller plus loin, remarquez que les commentaires sont préfixés par des ";". Vient ensuite la déclaration SOA de l'autorité de résolution des noms. Le numéro Serial (ici, `2017100725`), est utilisé par les serveurs DNS esclaves que l'on doit configurer par la suite, afin d'indiquer s'ils doivent ou non mettre à jour leur base. Généralement, ce numéro se compose d'une date inversée suffixée par un chiffre que l'on incrémente lors de chaque modification du fichier de zone. Les trois champs suivants permettent de configurer le processus de communication du serveur esclave, vis-à-vis de son maître : à l'expiration du délai `Refresh`, le serveur esclave contacte son maître et si le message n'aboutit pas, il faudra relancer une tentative au bout du délai exprimé par le champ `Retry`. À l'expiration du délai `Expire`, le serveur esclave considérera alors que son maître n'est plus joignable. Le dernier champ permet de définir la durée de vie minimum du cache. Les délais sont exprimés en secondes. Les trois dernières lignes de la déclaration de l'autorité de résolution permettent au serveur de noms de se repérer lui-même. Le champ `HINFO` donne la possibilité de repérer, à l'aide d'informations complémentaires, le serveur de noms lui-même. Mais pour des raisons de sécurité, il est déconseillé d'indiquer trop d'informations.