Serviço de resolução de nomes

Sistema DNS
Serviço DNS BIND

Outubro/2017

Prof. Jairo

jairo@uni9.pro.br

professor@jairo.pro.br

http://www.jairo.pro.br/

Este material tem por única intenção reunir um conteúdo acadêmico necessário para auxiliar no ensino da disciplina "Prática e Administração de Sistemas Operacionais de Redes Livres", ministrado nos cursos de Tecnologias em Redes de Computadores e Segurança da Informação.

O conteúdo aqui exposto pode ser livremente redistribuído e usado como apoio de aula, desde que mantenha a sua integridade original.

O arquivo "dns-bind.pdf" pode ser livremente acessado em "http://www.jairo.pro.br/prat_adm_sist_oper/".

Qualquer crítica ou sugestão, favor entrar em contato com o Prof. Jairo no endereço eletrônico "jairo@uni9.pro.br" ou "professor@jairo.pro.br".

Sumário

1 - SISTEMA DE NOMES DE DOMÍNIOS

DNS (*Domain Name System*) é um sistema de resolução de nomes. As pessoas usam nomes para os acessos em rede, porém nos pacotes de dados vão endereços IP. É o DNS que faz esta tradução de nomes para endereços IP, e vice-versa.

O DNS, *Domain Name System*, é composto de três partes:

i) resolvedor (*resolver*);

ii) serviço de nome (*name server*);

iii) banco de dados com registros de recursos (*database of resource records - RRs*).

Basicamente, o *Domain Name System* contém um grande banco de dados (*database*) que reside em vários servidores, com o objetivo de identificar os nomes e endereços IPs dos vários hosts e domínios na internet. Este *database*, distribuído globalmente, reside nos serviços de nome.

O *Domain Name System* é usado para prover informação ao *Domain Name Service*, quando houver *query* (consulta). O serviço de nomes atende às consultas dos clientes, que é a função resolvedora (*resolver*). O sistema de nomes é a estrutura e dados que o compõem.

O *Domain Name System database* é dividido em seções chamadas zones. Os servidores de nomes em suas respectivas zones são os responsáveis por responder às consultas para as suas zones.

Uma zone é uma sub árvore (*subtree*) do DNS e é administrado separadamente. Para cada zone, normalmente existe um serviço de nomes primário e um ou mais secundários. Um serviço de nomes pode ser autoridade sobre mais de uma zone.

Os nomes (*DNS names*) são atribuídos através de registros, pela IANA[1] (*Internet Assigned Number Authority*). O nome do domínio (*domain name*) é um nome atribuído para um domínio internet. Por exemplo, **uninove.br** é o nome do domínio de uma instituição de ensino e **jairo.pro.br** é o nome do domínio oficial deste material sobre DNS.

Já a nomeação de hosts dentro do domínio é atribuição dos administradores daquele domínio.

O acesso ao banco de dados de nomes de domínios (*domain name database*) ocorre através de um *resolver* (com uso de uma aplicação cliente). O *resolver* envia requisições aos serviços de nomes, que retornam a informação requisitada pelo usuário. A requisição é feita no endereço IP do serviço de nomes.

1 IANA: Internet Assigned Numbers Authority (www.iana.org) é a responsável pela coordenação global do DNS, endereçamento IP e outros recursos do protocolo IP.

O sistema DNS contém uma base de dados hierárquica, distribuída **globalmente** e gerenciada **localmente.**

A raiz dessa hierarquia distribuída é constituída pelos servidores-raiz, ***root-servers***. Inicialmente, os **root-servers** eram apenas sete, todos localizados nos Estados Unidos e geridos pela ***IANA***. Posteriormente, foram ampliados para treze ***root-servers***, dois dos quais ficaram localizados na Europa e um no Japão.

Mas logo depois se viu que treze também era pouco, devido ao rápido crescimento da internet, com consequente aumento substancial na carga de consultas nesses **root-servers**. O problema agora era que, atingido treze **root-servers**, não poderia mais ocorrer ampliação no número de **root-servers** usado pelos serviços de nomes, por causa de uma limitação no protocolo DNS.

A solução então foi desenvolver uma técnica chamada *anycast* para clonar os **root-servers** e assim criar servidores espelhos, que operacionalmente não se distinguem dos **root-servers** originais. Esses servidores espelhos foram distribuídos pelo mundo inteiro.

Em junho de 2017, já passa de 760 o número de **root-servers** "clonados" distribuídos pelo mundo, e que são atualizados pelos 13 **root-servers** originais.

No Brasil, em 2017 já são 19 **root-servers** distribuídos nas cidades de Belém, Belo Horizonte, Brasília, Campinas, Curitiba, Florianópolis, Fortaleza, Londrina, Natal, Porto Alegre, Rio de Janeiro, Salvador, São José dos Campos e São Paulo.

Esta distribuição regional reduziu a carga nos treze **root-servers** originais e tornou a resolução de nomes mais rápida.

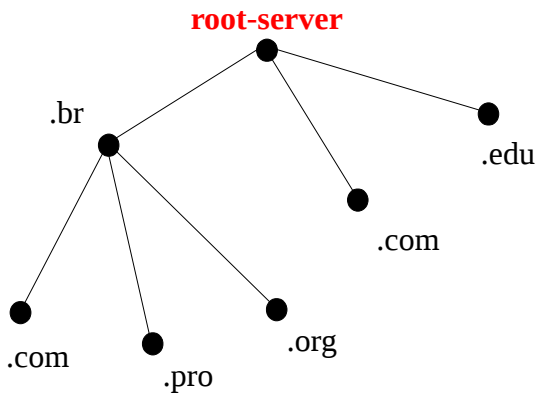
O conteúdo da raiz do DNS está baseado em TLD (*Top Level Domai*n, Domínio Nível Topo).

Os TLD de 3 letras foram os primeiros a serem criados e, originalmente, eram domínios Norte-Americanos. Posteriormente, alguns deles foram reclassificados como *gTLD (TLD genéricos, mundiais)*. São eles:

.edu: rede acadêmica;
.com: segmento do comércio/indústria;
.gov: governo norte-americano;
.net: atividades de suporte à rede;
.org: organizações não governamentais;
.mil: segmento militar (Arpanet);
.int: organizações internacionais.

Os TLD de 2 letras vieram em 1986. Nesse caso, cada país corresponde a duas letras e são chamados de ccTLD (*Country Codes* TLD - Códigos de Países). Alguns exemplos são:

.ar: Argentina
.au: Austrália;
.br: Brasil (registrado em 18 de abril de 1989);
.ca: Canadá;
.ch: Suiça;
.cl: Chile;
.de: Alemanha;
.fr: França;
.it: Itália;
.jp: Japão;
.mx: México;
.pt: Portugal;
.ru: Russia;
.tw: República da China;
.us: Estados Unidos da América.



Estrutura hierárquica com o root-server no topo.

Os ccTLD gozam de autonomia para estabelecer sua árvore hierárquica, sua abrangência e normas de registro. No Brasil, a autoridade sobre o ccTLD está em **registro.br**.

A configuração do serviço DNS envolve diferentes registros. Os principais tipos de registros (os mais comuns) são:

- A: Address, especifica um endereço IP direto;
- AAAA: Address Ipv6, especifica um endereço Ipv6;
- NS: name server, especifica serviços DNS para o domínio ou subdomínio;
- CNAME: Canonical NAME, um apelido para outro hostname;
- MX: Mail eXchanger (ou exchange), o serviço de email;
- PTR: PoinTeR, aponta o hostname/domínio reverso a partir de um endereço IP;
- SOA: Start Of Authority, responsável por respostas autoritativas por um domínio;
- TXT: Registro de texto, com formato arbitrário (para diversas funcionalidades);
- LOC: Localização geográfica;
- SRV: Serviços, proporciona a localização de serviços conhecidos;
- DNAME: Domain Alias ou apelido para domínio. É semelhante a CNAME, porém trata de apelido para todo o domínio e não apenas para um hostname.

1.1 - Funcionamento do DNS

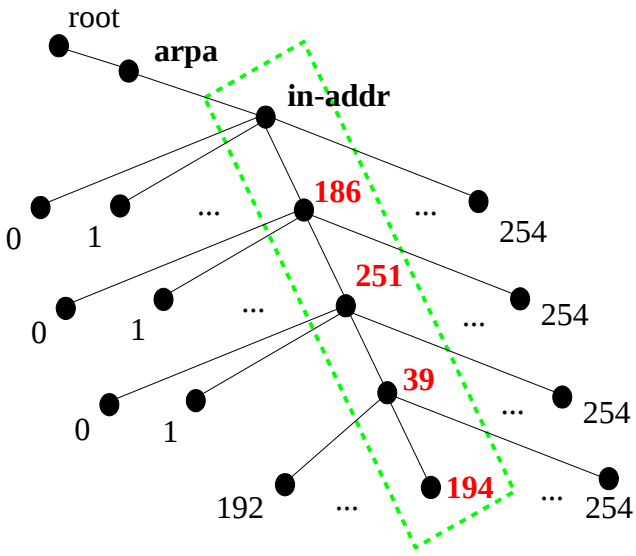
A estrutura hierárquica do DNS funciona como uma árvore invertida, começando pelo **root-server** (representado po um ponto "**.** ") e seguindo pelos TLD (Top Level Domains), que são divididos em genéricos (gTLD) usados em todo o mundo e os de código de país (ccTLD), que possuem extensões de domínios administrados pelos países.

Para o DNS funcionar corretamente é necessário no mínimo dois servidores: um *master* (primário) e um *slave* (secundário). O *slave* depende do *master* para as suas atualizações, porém pode responder sem a existência deste. Então, em caso de falha do *master*, o DNS continua funcionando. Operacionalmente, após a instalação e configuração do *master*, configura-se o *slave* especificando quem é o *master* do domínio. Desse modo, qualquer alteração no database do *master* será replicada automaticamente para o *slave*.

Para um determinado domínio na internet podem existir vários *slaves*, porém apenas um *master*.

Quanto ao reverso, ele é controlado pelas entidades que possuem os endereços IP. Nesse caso, o dono pode escolher subdelegar DNS reverso em uma faixa de IP para alguma outra entidade, que por sua vez também pode subdelegar. IANA é quem possui todos os endereços IP, então o início dessa subdelegação ocorre lá.

IANA delega endereçamento de IP aos Registros Internacionais de Internet (RIR: *Regional Internet Registry*) em 5 regiões: **AfriNIC** (África), **APNIC** (Ásia/Pacífico), **ARIN** (América do Norte), **LACNIC** (América Latina) e **RIPE** (Europa, Oriente Médio e Ásia Central). Por sua vez, esses Registros Internacionais delegam os endereços IP aos provedores de acesso, que por sua vez delegam aos usuários finais.



*Reverso com o domínio especial **in-addr.arpa**.*

Para a implementação do reverso, existe um domínio especial reservado chamado **in-addr.arpa**, ao qual todos os endereços IP pertencem. Por exemplo, 192-255.39.251.186.in-addr.arpa para a faixa de IPs 186.251.39.192 a 186.251.39.255. E para chegar aos endereços de host, primeiro terá de perguntar a um **root-server** onde fica 186.0.0.0/8, depois onde fica 186.251.0.0/16 e por último, 186.251.39.0/24. Por exemplo, para obter o reverso do host no IP 186.251.39.194, o caminho seria o seguido na figura ao lado.

2 - SERVIÇO DNS

Domain Name Service é um serviço na Internet que mapeia endereço IP e Nome de Domínio

Totalmente Qualificado[2] (FQDN) de um para o outro. Desse modo, o DNS alivia a necessidade de guardar (lembrar) sites pelos seus endereços IPs.

Hosts que hospedam DNS são chamados de servidores de nomes (*name servers*). Nos sistemas membros da família Unix, o serviço DNS BIND (*Berkeley Internet Name Domain*) é o serviço de nomes padrão.

BIND é uma implementação de código fonte aberto do protocolo DNS, e está em uso na maior parte dos serviços de nomes na Internet.

2.1 - Função do serviço DNS

A principal função do serviço de nomes é mapear endereços IPs em nomes lidos pelos humanos.

Por exemplo, se alguém quiser acessar o site em **www.jairo.pro.br**, pacotes TCP devem ser enviados para a porta 80 de um *host* nesse domínio, porém para qual IP? O serviço DNS responde ao cliente que atualmente **www.jairo.pro.br** encontra-se nos endereços IP 104.28.0.37 e 104.28.1.37. Isso é chamado de **resolução direta** (*forward DNS*).

Da mesma forma que obteve o endereço IP do host/domínio, poderia obter o host/domínio dado o IP, que é a parte reversa do serviço DNS. Isso é chamado de **resolução reversa** (*reverse DNS*).

Uma outra função do serviço DNS é realizar cache local dos IPs já resolvidos, isso para evitar fazer novas consultas externas a hosts/domínios já traduzidos. Desse modo, atende mais rapidamente aos clientes do serviço e economiza *link* de acesso à internet.

2.2 - Instalação do serviço DNS BIND

A instalação do BIND será para um serviço standalone, e não inetd. A versão atual do BIND é a 9, e por isso o pacote se chama bind9.

Como primeiro passo, verificar se o serviço **bind9** está instalado. Num Ubuntu (ou Debian), procurar pelos pacotes **bind9** e **dnsutils**. Para isso, comandar:

2 FQDN significa *Fully Qualified Domain Name*. Este é o nome de um determinado host composto por seu hostname seguido do domínio a que pertence. Como exemplo temos um servidor de e-mail de hostname **mail**, pertencente ao domínio **exemplo.com**. Então, o FQDN desse host é **mail.exemplo.com**.

```
root# dpkg -l | grep -ie bind9 -e dnsutils
```

NOTA 1: não havendo saída no comando acima, significa que não está instalado.

NOTA 2: o comando seria "**rpm -aq | grep -ie bind-9 -e bind-utils**" num CentOS ou Red Hat.

Para instalar os pacotes **bind9** e **dnsutils** num Ubuntu ou Debian, comandar:

```
root# apt-get install bind9 dnsutils
```

NOTA: num CentOS ou Red Hat, o comando para instalar seria "**yum install bind-9 bind-utils**".

Depois de instalado num sistema SystemV, deve existir o script de inicialização em **/etc/init.d**. Num Ubuntu ou Debian, o comando para fazer esta verificação é:

```
root# ls /etc/init.d | grep -i bind9
/etc/init.d/bind9
```

NOTA 1: num CentOS ou Red Hat SystemV, o script está em **/etc/init.d/named**.

NOTA 2: num sistema SystemD, o equivalente é o arquivo de **unit**, que num Ubuntu é /lib/systemd/system/bind9.service e num CentOS é /usr/lib/systemd/system/named.service.

Após instalado num sistema SystemV, verificar se existem os seguintes arquivos:

```
root# file /etc/init.d/bind9
/etc/init.d/bind9: POSIX shell script, ASCII text executable
root# file /usr/sbin/named
/usr/sbin/named: ELF 32-bit LSB shared object, Intel 80386, version 1 (SYSV), dynamically linked (uses
shared libs), for GNU/Linux 2.6.18, stripped
root# file /etc/bind
/etc/bind: setgid directory
```

NOTA 1: os arquivos acima são relativos a um Ubuntu 12.

NOTA 2: num Red Hat ou CentOS SystemV, o script é **/etc/init.d/named**.

Na saída dos comando acima, temos:

- **/etc/init.d/bind9**: é o script de inicialização do serviço DNS;
- **/usr/sbin/named**: é o executável que ao rodar dá origem ao processo daemon;
- **/etc/bind**: é o diretório onde ficam as configurações do serviço named. O arquivo de configuração é **named.conf**.

  NOTA: num CentOS ou Red Hat, o arquivo de configuração está em "**/etc/named.conf**", e neste arquivo existe a diretiva *directory* que normalmente aponta para o diretório "/var/named". Ao contrário do Ubuntu ou Debian, as configurações do serviço normalmente são feitas no diretório **/var/named** e não em /etc/bind.

Num sistema SystemV, para saber se o serviço está rodando, usar o script de inicialização do serviço:

```
root# /etc/init.d/bind9  status
 * bind9 is running
```

NOTA 1: o comando acima foi disparado num Ubuntu12.

NOTA 2: num CentOS ou Red Had, o comando seria "**/etc/init.d/named status**".

NOTA 3: num sistema SystemD, o comando seria "**systemctl status named**" num CentOS ou Red Hat e "**systemctl status bind9**" num Ubuntu ou Debian.

Também poderia ser usado o comando **ps** para procurar pelo processo **named**:

```
root# ps  -ef  |  grep  named
bind     2762     1  0 15:28 ?        00:00:00  /usr/sbin/named  -u  bind
```

Se não houvesse saída no comando acima, isso indicaria que o processo **daemon** não está rodando. Se estivesse rodando, neste momento deveria ser parado para prosseguir nos passos de configuração do serviço a seguir. Num Ubuntu SystemV, o comando para parar é:

```
root# /etc/init.d/bind9  stop
 * Stopping domain name service... bind9
```

NOTA 1: num CentOS ou Red Had SystemV, o comando seria "**/etc/init.d/named stop**".

NOTA 2: num sistema SystemD, o comando seria "**systemctl stop named**" num CentOS ou Red Hat e "**systemctl stop bind9**" num Ubuntu ou Debian.

2.3 - Configuração da zone aluno.br

Nesta configuração do name server, será criada a zone "**aluno.br**", com os seguintes FQDN:

- **ns1.aluno.br:** é o host onde fica hospedado o serviço de nomes, no IP 10.102.1.10;
- **mta.aluno.br**: é o host onde fica hospedado o serviço de e-mail, no IP 10.102.1.250;
- **pop.aluno.br**: é CNAME (apelido) para **ns1.aluno.br**;
- **www.aluno.br**: é CNAME (apelido) para **mta.aluno.br**.

Para não perder tempo configurando todo o arquivo **named.conf**, e todos os registros necessários para estes quatro FQDNs, vamos pegar estas configurações já prontas no site **www.jairo.pro.br**.

Então, para a configuração da zone **aluno.br**, serão copiados mapas e configurações já preparados e disponíveis nos arquivos **bind9.tar.gz** (para o Ubuntu) e **bind93.tar.gz** (para o CnetOS). Estes dois arquivos estão na raiz do site em **www.jairo.pro.br**. Estes arquivos serão baixados de **www.jairo.pro.br** com a aplicação cliente de serviço web **wget**.

Como primeiro passo, entrar no diretório **/tmp** para baixar (download) o arquivo escolhido:

```
root# cd /tmp
```

Porém, se esta configuração estiver sendo feita num Laboratório Acadêmico da Uninove, para baixar estes arquivos com o comando **wget**, normalmente precisa antes acertar a variável **http_proxy**, isso para autenticar no proxy da Uninove e poder sair para a internet:

```
root# export http_proxy=http://RA:SENHA@186.251.39.92:3128
```

No comando acima, precisa fazer as seguintes substituições:

RA: é o RA do aluno;
SENHA: é a senha de acesso do aluno;
186.251.39.92: é o IP do serviço proxy, que atende na porta **3128** (é um Squid).

Para confirmar se a variável **http_proxy** está correta, usar o comando **echo**:

```
root# echo  $http_proxy
http://123456789:123456@186.251.39.92:3128
```

NOTA: a saída do comando acima mostra a variável de ambiente **http_proxy** configurada para o RA 123456789 e senha 123456.

Depois disso, é só baixar de **www.jairo.pro.br** o arquivo **bind9.tar.gz** (para Ubuntu ou Debian) ou **bind93.tar.gz** (para CentOS ou Red Hat). Para isso, usar o comando **wget**:

```
root# wget  www.jairo.pro.br/bind9.tar.gz
--2017-07-22 18:27:13--  http://www.jairo.pro.br/bind9.tar.gz
Resolving www.jairo.pro.br (www.jairo.pro.br)... 2400:cb00:2048:1::681c:25,
2400:cb00:2048:1::681c:125, 104.28.1.37, ...
Connecting to www.jairo.pro.br (www.jairo.pro.br)|2400:cb00:2048:1::681c:25|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1814 (1.8K) [application/x-gzip]
Saving to: `bind9.tar.gz'
100%[==============================>] 1,814       --.-K/s   in 0s
2017-07-22 18:27:18 (135 MB/s) - `bind9.tar.gz' saved [1814/1814]
```

NOTA: no comando acima, se o sistema operacional fosse um CentOS ou Red Hat, o arquivo a ser baixado seria **bind93.tar.gz**.

2.4 - Configuração do serviço de nomes num Ubuntu ou Debian

NOTA: para esta configuração, o arquivo **bind9.tar.gz** precisou ser anteriormente baixado (download) para o diretório **/tmp** (conforme item acima).

Para configurar o BIND num Ubuntu ou Debian, descompactar e extrair o conteúdo do arquivo **bind9.tar.gz** com os comandos **gunzip** e **tar**:

```
root# gunzip  bind9.tar.gz
root# tar  -xvf   bind9.tar
bind9/
bind9/named.conf.local
bind9/rev.aluno.br
bind9/named.conf
bind9/aluno.br
bind9/named.conf.options
```

A extração com o comando **tar**, acima, criou o subdiretório bind9 no /tmp. Entrar neste diretório com o comando **cd**:

```
root# cd  bind9
root# ls
aluno.br    named.conf    named.conf.local    named.conf.options    rev.aluno.br
```

O arquivo de configuração do serviço BIND é o **named.conf**. Para visualizar o conteúdo deste arquivo sem mostrar as linhas que iniciam por "//" (que são comentários) nem linhas em branco, usar o comando abaixo:

```
root# grep -v "^//" named.conf | grep -v "^$"
include "/etc/bind/named.conf.options";
zone "." {
        type hint;
        file "/etc/bind/db.root";
};
zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};
zone "127.in-addr.arpa" {
        type master;
        file "/etc/bind/db.127";
};
zone "0.in-addr.arpa" {
        type master;
        file "/etc/bind/db.0";
};
zone "255.in-addr.arpa" {
        type master;
        file "/etc/bind/db.255";
};
include "/etc/bind/named.conf.local";
```

NOTA: como pode ser observado acima, o arquivo de configuração **named.conf** contém apenas configurações *default* do serviço, que não precisam ser alteradas, e, através da diretiva *include*, adiciona dois arquivos que são de fato os locais aonde devemos configurar o serviço.

Para visualizar o conteúdo do arquivo **named.conf.options**, comandar:

```
root# grep -v "//" named.conf.options | grep -v "^$"
options {
        directory "/var/cache/bind";
        forwarders {
                186.251.39.121;
                186.251.39.122;
        };
        auth-nxdomain no;    # conform to RFC1035
        listen-on-v6 { any; };
};
```

NOTA: neste arquivo a única configuração que foi necessária fazer foi definir os endereços IP dos servidores de nome *master* (186.251.39.121) e *slave* (186.251.39.122) da Uninove como ***forwarders***. Esta configuração é necessária para que este serviço de nomes possa resolver nomes externos ao laboratório acadêmico, que está apartado da internet por um firewall. Com o uso de *forwarders*, todas as *querys* para nomes ou endereços IP na internet serão encaminhadas para os serviços de nomes da Uninove, que retornarão a resolução para este serviço DNS.

Para visualizar o conteúdo do arquivo **named.conf.local**, comandar:

```
root# grep -v "^//" named.conf.local | grep -v "^$"
zone "aluno.br" {
        type master;
        file "/etc/bind/aluno.br";
};
zone "1.102.10.in-addr.arpa" {
        type master;
        file "/etc/bind/rev.aluno.br";
};
```

NOTA: foi definida a zone **aluno.br** com os registros para resolução direta no arquivo aluno.br e registros para resolução reversa no arquivo rev.aluno.br. Se for o caso, podem ser alterados os endereços IP para a zone aluno.br.

Convém notar também em **named.conf.local** que estamos configurando um DNS master, e que os mapas da zone **aluno.br** estão nos arquivos **aluno.br** e **rev.aluno.br**. No arquivo **rev.aluno.br** está o mapa reverso, usado para traduzir endereços IPs em nomes.

O arquivo aluno.br contém os seguintes registros:

```
root# more  aluno.br
aluno.br.       IN      SOA     ns1.aluno.br. admin.aluno.br. (
                        2006081401      ; serial
                        28800           ; refresh
                        3600            ; retrai
                        604800          ; expire
                        38400           ; minimum
)

aluno.br.       IN      NS              ns1.aluno.br.
aluno.br.       IN      MX      5       mta.aluno.br.

ns1             IN      A               10.102.1.10
mta             IN      A               10.102.1.250
pop             IN      CNAME           ns1
www             IN      CNAME           mta
```

As configurações de ***serial***, ***refresh***, ***retrai***, ***expire*** e ***minimum***, são:

serial (número de série): toda vez que a zone for modificada, este número deve ser incrementado para que as mudanças sejam distribuídas para os serviços de nome secundários (slaves);

refresh (atualização): é a quantidade de tempo, em segundos, que o serviço de nomes secundário deve esperar para verificar se existe uma nova cópia da zone no master (primário). Se a zone foi modificada, então o serviço de nomes secundário vai atualizar a sua cópia para ficar idêntico ao que tem no master. Na configuração acima, este tempo é de 8 horas;

retry (tentar novamente): é a quantidade de tempo, em segundos, que o serviço de nomes primário deve esperar antes de reatualizar o serviço de nomes secundário, caso tenha falhado a tentativa anterior de atualização (refresh). Na configuração acima, este tempo é de 1 hora;

expire (expiração): é a quantidade de tempo, em segundos, que o serviço de nomes secundário vai manter a zone autoritativa (authoritative). Depois desse tempo, a zone não será mais válida nem mais usada pelo secundário (slave). Na configuração acima, este tempo é de 168 horas (7 dias);

minimum (mínimo): é a quantidade de tempo, em segundos, que os registros do domínio são válidos. Isto também é conhecido como TTL mínimo, e pode ser substituído por um registro individual de TTL. Na configuração acima, este tempo é de 640 minutos;

TTL (Time to Live, tempo de vida), é o número de segundos que um nome de domínio é armazenado (cached) localmente antes da expiração e retornado ao serviço de nome autoritativo (authoritative) para atualização de informação.

E o arquivo rev.aluno.br contém os seguintes registros:

```
root# more  rev.aluno.br
@       IN      SOA     ns1.aluno.br . admin.aluno.br. (
                        2006081401;
                        28800;
                        604800;
                        604800;
                        86400
)

        IN      NS      ns1.aluno.br.
10      IN      PTR     ns1.aluno.br.
250     IN      PTR     mta.aluno.br.
```

```
root# more  rev.aluno.br
$TTL 24
@       IN      SOA     ns1.aluno.br.   admin.aluno.br. (
        2006081401;
        28800;
        604800;
        604800;
        86400
)
                IN      NS              ns1.aluno.br.
10              IN      PTR             ns1.aluno.br.
250             IN      PTR             mta.aluno.br.
```

Depois de corrigir os endereços IP, precisa copiar os arquivos baixados de **jairo.pro.br** para os seus locais de funcionamento normal:

```
root# cp  -f  named.conf  /etc
root# cp  *  /etc/bind
```

NOTA: o arquivo named.conf fica em **/etc/named.conf**, e os demais em **/etc/bind**.

2.5 - Configuração do serviço de nomes num CentOS ou Red Hat

Num Linux baseado no Red Hat (CentOS), o diretório de configuração do serviço DNS

BIND é normalmente o **/var/named/**.

Previamente, já foi feito o download do arquivo **bind93.tar.gz** de **www.jairo.pro.br**.

Para configurar o BIND num Red Hat ou CentOS, entrar no diretório **/tmp** e descompactar e extrair o conteúdo do arquivo **bind93.tar.gz** com os comandos **gunzip** e **tar**:

```
root# gunzip  bind93.tar.gz
root# tar  -xvf   bind93.tar
bind93/
bind93/named.conf
bind93/db.local
bind93/aluno.br
bind93/db.127
bind93/named.conf.local
bind93/db.0
bind93/rev.aluno.br
bind93/named.root
bind93/db.255
```

A extração com o comando **tar**, acima, criou o subdiretório bind93 no /tmp. Entrar neste diretório com o comando **cd**:

```
root# cd  bind93
root# ls
aluno.br  db.0  db.127  db.255  db.local  named.conf  named.conf.local  named.root  rev.aluno.br
```

O arquivo de configuração do serviço BIND é o **named.conf**. Os conteúdos e configurações são semelhantes ao que já foi feito para o Ubuntu.

O arquivo de configuração é o **named.conf**. Para visualizar o conteúdo desse arquivo, usar o comando **grep**:

```
root# grep -v "^//" named.conf | grep -v "^$"
options
{
        directory "/var/named"; // the default
        dump-file               "data/cache_dump.db";
        statistics-file         "data/named_stats.txt";
        memstatistics-file      "data/named_mem_stats.txt";
        forwarders {
          186.251.39.121;
          186.251.39.122;
        };
};
zone "." IN {
        type hint;
        file "named.root";
};
zone "localhost" {
        type master;
        file "db.local";
};
zone "127.in-addr.arpa" {
        type master;
        file "db.127";
};
zone "0.in-addr.arpa" {
        type master;
        file "db.0";
};
zone "255.in-addr.arpa" {
        type master;
        file "db.255";
};
include "named.conf.local";
```

Análogo ao caso do Ubuntu, acima, os arquivos **named.root**, **db.local**, **db.127**, **db.0** e **db.125** são *defaults* do serviço e não precisam ser alterados.

No arquivo *included*, **named.conf.local**, é que estão as configuração para a zone **aluno.br**.

Para visualizar o conteúdo deste arquivo, usar o comando **cat**:

```
root# cat  named.conf.local
//
// Do any local configuration here
//
// dominio aluno.br
zone "aluno.br" {
        type master;
        file "/var/named/aluno.br";
};
// reverso de aluno.br
zone "1.102.10.in-addr.arpa" {
        type master;
        file "/var/named/rev.aluno.br";
};
```

Convém notar também em **named.conf.local** que estamos configurando um DNS master, e que os mapas da zone **aluno.br** estão nos arquivos **aluno.br** e **rev.aluno.br**. No arquivo **rev.aluno.br** está o mapa reverso, para traduzir IPs em nomes.

Desse modo, tudo que resta a fazer agora é (se este for o caso) acertar o endereço da rede no arquivo **aluno.br**:

```
root# cat  aluno.br
$TTL 24h
aluno.br.       IN      SOA             ns1.aluno.br.   admin.aluno.br. (
                        2006081401      ; serial
                        28800           ; refresh
                        3600            ; retrai
                        604800          ; expire
                        38400           ; minimum
)
aluno.br.       IN      NS              ns1.aluno.br.
aluno.br.       IN      MX      5       mta.aluno.br.
ns1             IN      A               10.102.1.10
mta             IN      A               10.102.1.250
pop             IN      CNAME           ns1
www             IN      CNAME           mta
```

NOTA: no final da primeira linha do arquivo **aluno.br**, "admin.aluno.br" refere-se ao e-mail do administrador do serviço de nomes, admin@aluno.br.

Convém notar que o registro MX admite prioridades, por exemplo 5, 10, etc. Quanto menor esse número, maior a sua prioridade. Nesse caso, se houvesse mais de um serviço de e-mail, o de maior prioridade é que receberia os e-mails.

No arquivo **rev.luno.br** é que está o mapa para resolução reversa:

```
root# more  rev.aluno.br
$TTL 24
@       IN      SOA     ns1.aluno.br.   admin.aluno.br. (
        2006081401;
        28800;
        604800;
        604800;
        86400
)
                IN      NS              ns1.aluno.br.
10              IN      PTR             ns1.aluno.br.
250             IN      PTR             mta.aluno.br.
```

Depois de corrigir os endereços IP, precisa copiar os arquivos baixados de **jairo.pro.br** para os seus locais de funcionamento normal:

```
root# cp  -f  named.conf  /etc
root# cp  *  /var/named
```

NOTA: o arquivo named.conf fica em **/etc/named.conf**, e os demais em **/var/named**.

2.6 - Iniciar o serviço de nomes BIND

Antes de iniciar o serviço DNS, verificar quais portas TCP estão abertas. Para isso, é necessário a aplicação **nmap** para fazer um scan de portas:

```
root# nmap  localhost
Starting Nmap 5.21 ( http://nmap.org ) at 2017-07-22 19:06 BRT
Nmap scan report for localhost (127.0.0.1)
Not shown: 999 closed ports
PORT            STATE           SERVICE
631/tcp         open            ipp

Nmap done: 1 IP address (1 host up) scanned in 0.03 seconds
```

NOTA: o scan mostrou que apenas a porta 631 (serviço de impressão) está aberta.

E também, antes de iniciar o serviço DNS, verificar se existe o processo daemon **named** rodando:

```
root# ps -ef | grep named
```

NOTA: não deve haver saída no comando, indicando que esse daemon não está rodando.

Para iniciar o serviços de nomes num Ubuntu SystemV, o comando é:

```
root# /etc/init.d/bind9 start
 * Starting domain name service... bind9
```

NOTA 1: num CentOS ou Red Had SystemV, o comando seria "**/etc/init.d/named start**".

NOTA 2: num sistema SystemD, o comando seria "**systemctl start named**" num CentOS ou Red Hat e "**systemctl start bind9**" num Ubuntu ou Debian.

Para checar se o serviço iniciou corretamente, verificar no arquivo de logs do sistema:

```
root# grep named /var/log/syslog
…
Jul 23 01:39:12 ubuntu12 named[4330]: zone 0.in-addr.arpa/IN: loaded serial 1
Jul 23 01:39:12 ubuntu12 named[4330]: zone 127.in-addr.arpa/IN: loaded serial 1
Jul 23 01:39:12 ubuntu12 named[4330]: zone 255.in-addr.arpa/IN: loaded serial 1
Jul 23 01:39:12 ubuntu12 named[4330]: zone localhost/IN: loaded serial 2
Jul 23 01:39:12 ubuntu12 named[4330]: managed-keys-zone ./IN: loaded serial 4
Jul 23 01:39:12 ubuntu12 named[4330]: running
```

NOTA: num Red Hat ou CentOS, o comando seria "**grep named /var/log/messages**".

E depois disso, o scan de portas vai mostrar que a porta TCP 53 também está aberta:

```
rootl# nmap localhost
Starting Nmap 5.21 ( http://nmap.org ) at 2017-07-23 00:06 BRT
Nmap scan report for localhost (127.0.0.1)
Not shown: 999 closed ports
PORT        STATE       SERVICE
53/tcp      open        dns
631/tcp     open        ipp

Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
```

No caso do serviço DNS, ele responde também na porta 53 UDP. Para determinar se essa porta está aberta, usar **nmap -sU** para fazer um scan UDP:

```
root# nmap -sU localhost
Starting Nmap 5.21 ( http://nmap.org ) at 2017-07-23 00:09 BRT
Nmap scan report for localhost (127.0.0.1)
Not shown: 999 closed ports
PORT        STATE       SERVICE
53/udp      open        dns
68/udp      open        dhcpc

Nmap done: 1 IP address (1 host up) scanned in 1.03 seconds
```

E agora o comando **ps** vai mostrar que o daemon **named** está rodando:

```
root# ps -ef | grep named
root     2354    1  0 13:41 ?       00:00:00 /usr/sbin/named
```

2.7 - Testar a resolução de nomes

Para testar, usar as aplicações clientes **nslookup** e **dig** e enviar essa *query* para localhost, que é onde está o serviço de nomes:

```
root# nslookup www.aluno.br localhost
Server:        localhost
Address:       127.0.0.1#53

www.aluno.br    canonical name = mta.aluno.br.
Name:   mta.aluno.br
Address: 10.102.1.250
```

```
root# nslookup  mta.aluno.br  localhost
Server:         localhost
Address:        127.0.0.1#53

Name:   mta.aluno.br
Address: 10.102.1.250
```

```
root# nslookup  ns1.aluno.br  localhost
Server:         localhost
Address:        127.0.0.1#53

Name:   ns1.aluno.br
Address: 10.102.1.10
```

```
root# nslookup  10.102.1.250  localhost
Server:         localhost
Address:        127.0.0.1#53

250.0.102.10.in-addr.arpa       name = mta.aluno.br.
```

O comando **dig** também faz *query* no serviço DNS, e tem como vantagem fornecer também o tempo gasto (query time):

```
root# dig  www.aluno.br  @localhost

; <<>> DiG 9.9.4-RedHat-9.9.4-50.el7_3.1 <<>> www.aluno.br @localhost
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45917
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;www.aluno.br.          IN          A

;; ANSWER SECTION:
www.aluno.br.          38400       IN      CNAME       mta.aluno.br.
mta.aluno.br.          38400       IN      A           10.102.1.250

;; AUTHORITY SECTION:
aluno.br.              38400       IN      NS          ns1.aluno.br.

;; ADDITIONAL SECTION:
ns1.aluno.br.          38400       IN      A           10.102.1.10

;; Query time: 0.02 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Jul 22 21:09:13 2017
;; MSG SIZE  rcvd: 98
```

Fazer também uma resolução do FQDN www.aluno.br na máquina do colega, onde 10.102.1.XX é o endereço IP aonde está o serviço de nomes (substituir XX pelo IP do host):

```
root# nslookup  www.aluno.br  10.102.1.XX
Server:         10.102.1.XX
Address:        10.102.1.XX#53

www.aluno.br    canonical name = mta.aluno.br.
Name:   mta.aluno.br
Address: 10.102.1.250
```

Testar também a resolução para um FQDN externo:

```
shell# nslookup  www.jairo.pro.br  localhost
Server:         localhost
Address:        127.0.0.1#53

Non-authoritative answer:
www.jairo.pro.br        canonical name = www.jairo.pro.br.cdn.cloudflare.net.
Name:   www.jairo.pro.br.cdn.cloudflare.net
Address: 104.28.1.37
Name:   www.jairo.pro.br.cdn.cloudflare.net
Address: 104.28.0.37
```

NOTA: para a resolução de nomes externos (aqueles que não estão no *database* local do serviço de nomes), a resposta vem precedida de "*Non-authoritative answer*", indicando que o serviço de nomes perguntou a outros serviços de nomes a respeito deste FQDN.