****

**Visão**

Com a crescente demanda sobre Tecnologias, percebemos que muitas pessoas apesar de buscarem informações, não possuem fontes que queiram realmente passar o conhecimento da maneira como ela deve ser, livre e com embasamento técnico que permita ser aplicado e utilizado quando necessário, além de serem testados em sua criação, tornando esta informação útil e confiável.

**Missão**

O Laboratório foi criado com a intenção de buscar e disseminar o conhecimento de uma maneira clara e objetiva, de forma gratuita, auxiliando na evolução dos membros e da sociedade na qual estas informações são compartilhadas, buscando o crescimento de todos os envolvidos nesta criação de valores.

**Licença**



Esta licença permite que outros remixem, adapte, e criem obras derivadas sobre a obra original, desde que com fins não comerciais e contanto que atribuam crédito ao autor e licenciem as novas criações sob os mesmos parâmetros. Outros podem fazer download ou redistribuir a obra da mesma forma que na licença anterior, mas eles também podem traduzir, fazer remixes e elaborar novas histórias com base na obra original. Toda nova obra feita a partir desta deverá ser licenciada com a mesma licença, de modo que qualquer obra derivada, por natureza, não poderá ser usada para fins comerciais.

This license lets other remix, tweak, and build upon your work non-commercially, as long as they credit you and license their new creations under the identical terms.

Para maiores informações sobre o método de licenciamento acesse os seguintes sites:

Brasil:

<http://creativecommons.org.br/as-licencas/>

<http://creativecommons.org/licenses/by-nc-sa/3.0/br/>

Internacional:

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

<http://creativecommons.org/licenses/by-nc-sa/3.0/legalcode>

**1 – Protocolo DNS**

O **DNS** é um serviço responsável por traduzir nomes em Endereços IP (e vice-versa) de um determinado domínio. No mundo **UNIX-Like**, o serviço **DNS (Domain Name System)** é implementado pelo software **BIND (Barkeley Internet Name Domain)**.

O **BIND** trabalha na arquitetura cliente-servidor. O **Resolver** é o cliente que faz perguntas sobre determinado computador. O Servidor DNS é implementado pelo **Daemon Named** no **BIND**, ele é o processo que responde às perguntas.

**2 – Instalação do BIND no Linux**

**2.1 – Tipos de Configuração do BIND**

Existem três configurações básicas de servidor de nomes com o **BIND**:

* **Somente Cache** 🡪 São servidores que executam o **Daemon Named**, mas não são fontes oficiais de informações a respeito de domínios. Esses servidores obtêm a resposta a todas as perguntas que lhe são direcionadas a partir de um servidor remoto.
* **Servidor Master (Mestre)** 🡪 É a fonte oficial de todas as informações a respeito de um domínio específico. Ele carrega informações a respeito do domínio a partir de arquivos locais, mantidos pelo administrador de domínio. O **Servidor Master** é chamado assim em virtude de poder responder a qualquer pergunta sobre seu domínio com total autoridade. O **Servidor Master** também é chamado de **Servidor Primário**.
* **Servidor Slave (Escravo)** 🡪 É considerado um servidor autorizado, pois tem um banco de dados completo do domínio, que é transferido para ele através do **Servidor Master**. Essa transferência chama-se **Zone Transfer**. Esse banco de dados é somente leitura, pois toda alteração deve ser realizada no **Servidor Master** e replicada para o **Servidor Slave**. O **Servidor Slave** também é chamado de **Servidor Secundário**.

**2.2 – Compilando o BIND e Suas Dependências**

Aqui iremos compilar e configurar o **BIND** e suas dependências. O **BIND** já vem com o recurso de instalamos seu **Daemon** em **Jail**, porém eu gosto de construir minha própria **Jail**. Qualquer dúvida de como instalar e configurar uma **Jail** não deixe de ler sobre **Configurando a Jail**.

Iremos instalar e configurar o **Servidor** **Master** no **Fusion** e o **Servidor Slave** no **Corola**.

Vamos começar então com o **Servidor Master** o **Fusion**, iremos agora montar a **Jail**.

root@fusion:~# mount -o bind /proc /opt/dns/proc

root@fusion:~# mount -o bind /dev /opt/dns/dev

root@fusion:~# chroot /opt/dns /bin/bash

dns-fusion:/#

Agora vamos instalar alguns softwares necessários para a compilação do **BIND**:

dns-fusion:/# aptitude install bzip2 automake autoconf gcc g++ bash-completion perl make

--==[ Resumido ]==--

Current status: 2 updates [-2].

dns-fusion:/#

O código fonte do **BIND** já possui os fontes das bibliotecas criptográficas do **OpenSSL** necessárias para as funções de segurança disponíveis (**DNSSEC** e **TSIG**), mas para uma melhor performance das operações de criptografia é indicado que o pacote seja instalado separadamente.

Agora iremos compilar o **OpenSSL**, para que possamos configurar a parte de segurança do **BIND** (como o **DNSSEC** e o **TSIG**). Para isso, vamos baixar o pacote do **OpenSSL** e descompacta-lo:

dns-fusion:/# cd /usr/src/

dns-fusion:/usr/src# wget http://www.openssl.org/source/openssl-0.9.8zb.tar.gz

--2014-08-12 15:04:00-- http://www.openssl.org/source/openssl-0.9.8zb.tar.gz

Resolving www.openssl.org (www.openssl.org)... 185.9.166.106

Connecting to www.openssl.org (www.openssl.org)|185.9.166.106|:80... connected.

HTTP request sent, awaiting response... 200 OK

Length: 3727934 (3.6M) [application/x-gzip]

Saving to: `openssl-0.9.8zb.tar.gz'

100%[================================================>] 3,727,934 165K/s in 20s

2014-08-12 15:04:21 (183 KB/s) - `openssl-0.9.8zb.tar.gz' saved [3727934/3727934]

dns-fusion:/usr/src# tar xvzf openssl-0.9.8zb.tar.gz

openssl-0.9.8zb/include/openssl/bn.h

openssl-0.9.8zb/include/openssl/buffer.h

--==[ Resumido ]==--

openssl-0.9.8zb/VMS/WISHLIST.TXT

tar: A lone zero block at 36020

dns-fusion:/usr/src#

Agora vamos compilar o **OpenSSL**:

dns-fusion:/usr/src# cd openssl-0.9.8zb/

dns-fusion:/usr/src/openssl-0.9.8zb# ./config

dns-fusion:/usr/src/openssl-0.9.8zb# make

dns-fusion:/usr/src/openssl-0.9.8zb# make install

Agora vamos construir a árvore de diretórios e usuário para o **BIND**:

dns-fusion:~# mkdir -p /etc/named

dns-fusion:~# mkdir -p /var/named/namedb/master

dns-fusion:~# mkdir -p /var/run/named

dns-fusion:~# groupadd named

dns-fusion:~# useradd named -g named -m -s /bin/false

dns-fusion:~# chown named.named /var/named –R

dns-fusion:~# chown named.named /var/run/named -R

dns-fusion:~#

Na hora da compilação iremos indicar o diretório ***/etc/named*** como o diretório onde conterá o arquivo de configuração e as bases ficaram no diretório /***var/named/namedb/master***. Também criamos o grupo **named** e usuário **named** para gerenciar o **Daemon** do **BIND**.

Agora iremos baixar e compilar e instalar o **BIND**.

dns-fusion:/# cd /usr/src/

dns-fusion:/usr/src# wget ftp://ftp.isc.org/isc/bind9/9.10.0rc2/bind-9.10.0rc2.tar.gz

--2014-08-25 19:53:18-- ftp://ftp.isc.org/isc/bind9/9.10.0rc2/bind-9.10.0rc2.tar.gz

=> `bind-9.10.0rc2.tar.gz'

Resolving ftp.isc.org (ftp.isc.org)... 204.152.184.110, 2001:4f8:0:2::18

Connecting to ftp.isc.org (ftp.isc.org)|204.152.184.110|:21... connected.

Logging in as anonymous ... Logged in!

==> SYST ... done. ==> PWD ... done.

==> TYPE I ... done. ==> CWD (1) /isc/bind9/9.10.0rc2 ... done.

==> SIZE bind-9.10.0rc2.tar.gz ... 8335501

==> PASV ... done. ==> RETR bind-9.10.0rc2.tar.gz ... done.

Length: 8335501 (7.9M) (unauthoritative)

100%[==========================================================================>] 8,335,501 90.0K/s in 2m 0s

2014-08-25 19:55:33 (68.0 KB/s) - `bind-9.10.0rc2.tar.gz' saved [8335501]

dns-fusion:/usr/src#

dns-fusion:/usr/src# tar xvzf bind-9.10.0rc2.tar.gz'

dns-fusion:/usr/src# cd bind-9.10.0rc2/

dns-fusion:/usr/src/bind-9.10.0rc2# ./configure --with-openssl=/usr/local/ssl --sysconfdir=/etc/named

dns-fusion:/usr/src/bind-9.10.0rc2# make

dns-fusion:/usr/src/bind-9.10.0rc2# make install

dns-fusion:/usr/src/bind-9.10.0rc2#

Aqui fizemos a compilação apontando o diretório onde compilamos o **OpenSSL** com a opção

--***with-openssl*** , e também apontamos o diretório onde será criado os arquivos de configuração com a opção ***--sysconfdir***.

**2.3 – Instalando o BIND e Suas Dependências Através de Pacotes Pré-Compilados**

Agora iremos instala o **Servidor DNS Slave** no servidor **Corola**, para mudar um pouco iremos utilizar a versão **7.6** do **Debian Wheezy**. E iremos instalar o **BIND** com o utilitário ***aptitude*** como se segue.

dns-corola:/# aptitude install bind9 bind9utils bind9-doc bind9-host openssl dnsutils

Após concluir a instalação já podemos configurar. Já que os pacotes são pré-compilados, a estrutura de diretório é um pouco diferente, por exemplo, o diretório com o arquivo de configuração fica em ***/etc/bind/named.conf***.

**3 – Configuração do RNDC no Linux**

**3.1 – Configurando o RNDC (Remote**

O **RNDC** é um utilitário que controla as operações do **Servidor BIND**. O **RNDC** comunica com o **Servidor BIND** através de uma conexão **TCP**, opcional enviando os comandos autenticados com assinatura digital.

**Observação:** Devemos parar o serviço do **BIND**, caso ele esteja em execução para configurarmos o utilitário **RNDC**.

**3.1.1 – Gerando o Arquivo /etc/named/rndc.conf com o rndc-confgen**

Precisamos criar o arquivo ***/etc/named/rndc.conf***, onde serão armazenadas as assinaturas digitais geradas aleatoriamente, para isso execute:

dns-fusion:/# rndc-confgen -a -A hmac-sha512 -c /etc/named/rndc.conf -k rndc.key -b 512

wrote key file "/etc/named/rndc.conf"

dns-fusion:/#

Onde as opções do utilitário **rndc-confgen** são:

* **-a** 🡪 Configuração do **RNDC** automaticamente.
* **-A <Algoritimo>🡪** Aqui especificamos o algoritmo de criptografia que iremos utilizar para o **TSIG**. As possibilidades são: **hmac-md5**, **hmac-sha1**, **hmac-sha224**, **hmac-sha256**, **hmac-sha384** e **hmac-sha512**. Se não especificado o padrão será o **hmac-md5**.
* **-b <tamanho>** 🡪 Especifica o tamanho da chave de autenticação em bits. O tamanho da chave de ter um tamanho de **1** a **512 bits**. Por padrão é o tamanho do **hash (128 bits)**. É recomendado que seja utilizado um tamanho superior a **256 bits** se o seu **Servidor BIND** for usado fora da rede local.
* **-c <arquivo>** 🡪 É utilizado com a opção **–a** para especificar um lugar alternativo para a localização do arquivo ***rndc.conf***.
* **-k <nome>** 🡪 Especifica o nome da assinatura digital que será gerado. O padrão é ***rndc.key.*** É altamente recomendado alterar o nome da assinatura digital quando você quiser utilizar o **RNDC** para acessar um **Servidor BIND** em outra máquina.
* **-p <porta>** 🡪 Especifica uma porta **TCP** para comunicação com o **named**, está porta ficará na escuta aguardando os comandos **RNDC**. Por padrão é a **953/TCP**.
* **-s <endereço>** 🡪 Especifica um endereço IP onde o **named** ficará na escuta para conexão com o **RNDC**. Por padrão o IP é o de **Loopback 127.0.0.1**.
* **-t <dir chroot>** 🡪 Usado com a opção **–a** para especificar um diretório onde o **named** será executado em **Jail**.
* **-u <user>** 🡪 Usado com a opção **–a** para indicar um proprietário do arquivo ***rndc.key*** gerado.

**3.1.2 – Configurando o Arquivo /etc/named/rndc.conf**

Este é o arquivo de configuração do **RNDC**, que é a ferramenta de controle do **BIND9** e substitui o **NDC**, que era a ferramenta de controle das versões anteriores do **BIND**. Sua principal função é fornecer uma chave secreta para o funcionamento do **RNDC**. Uma cópia dessa chave deve estar nos arquivos ***/etc/named/named.conf*** e ***/etc/named/rndc.key*** (dependendo da configuração) para que o **BIND** inicialize.

dns-fusion:/# cat /etc/named/rndc.conf > /etc/named/rndc.key

dns-fusion:/# cat /etc/named/rndc.conf > /etc/named/named.conf

dns-fusion:/#

Agora iremos configurar o **RNDC** acrescentando os blocos ***options*** e ***Server***:

dns-fusion:/# vi /etc/named/rndc.conf

options {

default-key "rndc.key";

default-server 127.0.0.1;

};

server localhost {

key "rndc.key";

};

key "rndc.key" {

algorithm hmac-sha512;

secret "HEHTs9psYldIA9UEOsfW9NxE8ZvV8WuQxEDnZ+eaRIXPWyW2uiNHxYU3P/AJeMAFS3CxFx3wEtcm20rEevs7wQ==";

};

"/etc/named/rndc.conf" 13L, 250C written

dns-fusion:/#

**3.1.3 – Configurando o Arquivo /etc/named/rndc.key**

Este é um arquivo onde temos a cópia da chave secreta que o **RNDC** usa para controlar o **BIND**. Esse arquivo é utilizado se o arquivo ***/etc/named/named.conf*** não possui a chave. Neste caso é necessário colocar a diretiva ***include “/etc/named/rndc.ket”*** no arquivo ***/etc/named/named.conf***.

Um exemplo do arquivo ***/etc/namedrndc.key***:

dns-fusion:/# cat /etc/named/rndc.key

key "rndc.key" {

algorithm hmac-sha512;

secret "HEHTs9psYldIA9UEOsfW9NxE8ZvV8WuQxEDnZ+eaRIXPWyW2uiNHxYU3P/AJeMAFS3CxFx3wEtcm20rEevs7wQ==";

};

dns-fusion:/#

**4 – Configuração do BIND no Linux**

**4.1 – Configurando o Arquivo /etc/named/named.conf**

Com os arquivos instalados vem o momento da configuração de várias diretivas do software. O arquivo de configuração é o ***named.conf*** e deve ser criado no diretório especificado no momento da compilação (***--sysconfdir=DIR***), no nosso caso ***/etc/named***.

Neste exemplo que segue iremos mostrar as configurações vitais de um **Servidor DNS**, **logging**, **zonas**, **servidores slaves** (secundários), **diretivas de segurança** entre **outros**.

**Observação:** Algumas diretivas são, nesta versão, obrigatórias e precisam ser configuradas para que o servidor possa ser iniciado, são elas:

* **$TTL** 🡪 Cada arquivo de zona deve conter essa diretiva. Usado como padrão para todos os registros que não possuem **TTL** especificado. Deve constar na primeira linha do arquivo de zona.
* **SOA (com o Serial Number)**🡪O Número serial deve ser um valor inteiro (**Integer**). Versões anteriores a **BIND9** permite usar o número serial como este: “**3.002**”.

**4.1.1 – Configurando ACL em /etc/named/named.conf**

Com as **ACL (Access Control Lists)** podemos criar lista de endereços IP e designarmos nomes a essas listas, com o único objetivo de quando formos nos referenciar a endereços IP usarmos nomes em vez de números. Algumas **ACL** já existem e podem ser usadas:

* **any** 🡪 Todos os hosts.
* **none** 🡪 Nenhum host.
* **localhosts** 🡪 Todos os endereços IP das interfaces do Servidor DNS.
* **localnets** 🡪 A rede inteira a qual o Servidor DNS faz parte.



**Observação:** O arquivo de configuração no **Debian** fica em ***/etc/bind/named.conf***, e nas distribuições **Red Hat** e **Suse** ficam em ***/etc***. No **Debian** o arquivo de configuração é dividido em quatro, sendo o ***named.conf*** o principal e três includes para ***named.conf.options*** onde são configurado todas as opções referente ao funcionamento do **BIND**, ***named.conf.local*** que serve para declarar todas as zonas pelas quais este servidor deve responder, e ***named.conf.default-zones*** onde estão as zonas padrões como **Loopback** e **Root Servers**.

Por exemplo:

dns-fusion:/# vi /etc/named/named.conf

acl srv\_slaves {

192.168.5.129/32;

192.168.130/32;

};

acl net\_how2security {

192.168.5.0/24;

};

"/etc/named/named.conf" 13L, 245C written

dns-fusion:/#

**4.1.2 – Configurando OPTIONS em /etc/named/named.conf**

A diretiva **options** define as opções globais que influenciam a operação do **BIND** e do **Protocolo** **DNS**.

Por exemplo:

options {

directory "/var/named/namedb/master";

listen-on { 127.0.0.1; 192.168.5.128; };

listen-on-v6 { none; };

empty-zones-enable no;

allow-transfer { srv\_slaves; };

allow-recursion { net\_how2security; };

allow-query { any; };

};

Onde:

* **directory <”caminho”>;** 🡪 A opção **directory** indica o diretório onde são armazenados os arquivos de zonas do **DNS**, que no nosso exemplo é: ***directory “/var/named/namedb/máster”***.
* **listen-on { IP\_1; IP\_2; };** 🡪 Endereços IP que o **Servidor DNS** irá escutar as conexões de consultas. Podemos utilizar uma **ACL** criada pelo administrador do DNS.
* **listen-on-v6 { IPv6\_1; IPv6\_2; };** 🡪 Endereços IP que o **Servidor DNS** irá escutar as conexões de consultas que utilizam a versão 6 do IP. Aqui utilizamos uma **ACL** existente de nome “**none**”. Podemos utilizar uma **ACL** criada pelo administrador do DNS.
* **empty-zones-enable <”no”>** 🡪 A nova versão do BIND9 por padrão gera zonas vazias ou fantasma. Em seu log você poderá observar mensagens como esta: “***Aug 18 16:25:03 server named[5610]: automatic empty zone 127.IN-ADDR.ARPA”***. Com isso, ele gera zonas automáticas. Eu desabilitei esse recurso.
* **allow-transfer { IP|ACL; };** 🡪 Endereços IP dos Servidores DNS com permissão de replicar a base DNS (Slaves ou Secundários). No nosso caso usamos uma **ACL srv\_slaves**.
* **allow-recursion { IP|ACL; };** 🡪 Endereços IP ou ACL com permissão de fazer consultas recursivas. Que no nosso caso usamos uma **ACL net\_how2security**.
* **allow-query {IP|ACL; };** 🡪 Endereços IP ou ACL com permissão de fazer consultas. No nosso caso usamos uma **ACL any**, para que todos possam fazer consultas em nosso servidor.

**4.1.3 – Configurando RNDC e CONTROLS em /etc/named/named.conf**

A diretiva **RNDC** e **controns** são utilizadas para autorizar o controle do **BIND** através do **RNDC**.

key "rndc.key" {

algorithm hmac-sha512;

secret "HEHTs9psYldIA9UEOsfW9NxE8ZvV8WuQxEDnZ+eaRIXPWyW2uiNHxYU3P/AJeMAFS3CxFx3wEtcm20rEevs7wQ==";

};

controls {

inet 127.0.0.1 allow { 127.0.0.1; } keys { "rndc.key"; };

};

**4.1.4 – Configurando SERVER em /etc/named/named.conf**

A diretiva **server** deve ser configurada nos servidores dependendo do seu papel na rede no caso do **Master** fica assim:

server 192.168.5.128 {

provide-ixfr yes;

transfer-format many-answers;

};

E no caso do **Slave**:

server 192.168.5.129 {

request-ixfr yes;

};

A transferência de zona incremental (**IXFR**) tem como finalidade diminuir a quantidade de informações trocadas, transferindo somente as mudanças feitas nas zonas do **Servidor Master** para o **Servidor Slave**.

Com essa configuração estamos dizendo que o **Servidor Master** fornecerá esse recurso de transferência e o **Servidor Slave** sempre que possível utilizará. Quando por algum motivo o **Servidor Master** recusar um **IXFR**, uma **AXFR** (que é a transferência de zona convencional) será realizada para manter as bases integras.

**4.1.5 – Configurando LOGGING em /etc/named/named.conf**

A diretiva **logging** são os registros das mensagens do **BIND**. É importante saber que somente entrará em vigor as opções de logging depois que todo o arquivo ***named.conf*** for lido e o servidor for inicializado, antes disso as mensagens de erro serão enviadas para o sistema de **syslog**.

Por exemplo:

logging {

channel "named\_log" {

file "/var/log/named/named.log" versions 2 size 50m;

print-time yes;

print-category yes;

};

category "security" {

"named\_log";

};

category "xfer-out" {

"named\_log";

};

category "xfer-in" {

"named\_log";

};

category "general" {

"named\_log";

};

};

O registro de logs no **BIND9** é dividido da seguinte forma:

* **channel** 🡪 Usado para especificar onde e que tipo de informação deve ser registrado. Como Níveis (**error, info, dynamic, etc**). Nele você pode configurar o **BIND** tanto para usar o **syslog**, como um arquivo qualquer. No nosso caso utilizamos um arquivo local em ***/var/log/named/named\_log***.
* **category** 🡪 Categoria de log do **BIND**. Existem várias categorias que podem ser usadas, tais como: **security, xfer-out, xfer-in, notify, network, etc**. Como os próprios nomes dizem, em **security** – informações de segurança, em **notify** – notificações de mudanças nas zonas primárias, etc.
  + **default** 🡪 A categoria default define as opções de registro para essa categoria.
  + **database** 🡪 Mensagens referente as bases usadas, como nome dos servidores, zonas e dados de cache.
  + **security** 🡪 Mensagens de consultas aprovadas e negadas.
  + **config** 🡪 Mensagens de configurações e processos.
  + **resolver** 🡪 Mensagens de resolução de nomes, consultas recursivas, etc.
  + **xfer-in** 🡪 Recebimento de Zonas.
  + **xfer-out** 🡪 Envio de Zonas.
  + **notify** 🡪 O protocolo **NOTIFY**.
  + **client** 🡪 Requisições de clientes.
  + **network** 🡪 Operações de rede.
  + **update** 🡪 Atualizações dinâmicas.
  + **update-security** 🡪 Atualizações aprovadas ou negadas.
  + **queries** 🡪 Mensagens de consultas.
  + **queries-errors** 🡪 Mensagens de erro de conaulstas.



**Observação:** Por padrão no **Debian** os logs do **BIND** irão para ***/var/log/daemon.log*** e nas distribuições **Red Hat** e **Suse** irão para ***/var/log/messages***

**4.1.6 – Configurando VIEW em /etc/named/named.conf**

A diretiva **view** nos permite fazer o **Split DNS**, ou seja, separar em dois ou mais diretiva de visualização de registros no **Servidor DNS**. Os registros que podem ser acessados por toda Internet e os que podem ser acessado apenas na rede de interna. Essa configuração é usada principalmente em redes desmilitarizadas, onde os **Servidor DNS** se encontra na **DMZ**, de acesso público e contém registros de rede interna também. Por questão de segurança é bom separar o que as pessoas de fora podem ver dentro da nossa rede, como os nomes dos dispositivos como câmeras IP, impressoras, Workstations, etc.

Com o exemplo abaixo, o nosso entendimento deste recurso ficará mais claro.

view "public" IN {

match-clients { any; };

zone-statistics yes;

recursion no;

zone "how2security.com.br" {

type master;

file "how2security-externo.com.br";

};

};

view "private" IN {

match-clients { 192.168.5.0/24; };

zone-statistics yes;

recursion yes;

zone "how2security.com.br" {

type master;

file "how2security-interno.com.br";

};

};

Neste exemplo, a zona “**how2security-externo.com.br**” contém o mapeamento dos hosts públicos e pode ser acessada por todos. Já a zona “**how2security-interno.com.br**” somente é acessado pelos hosts interno, ou seja, todos da rede **192.168.5.0/24**.

Se nenhuma **view** for especificada, o **BIND** tratará, por padrão, as zonas como sendo da classe **IN (Internet)**. Dentro das **views** são especificas a configuração das zonas primárias e secundárias.

**4.1.7 – Configurando ZONA em /etc/named/named.conf**

A diretiva de **zonas** as quais o servidor responderá como autoridade. Nela se configura as zonas primárias, secundárias, reversas e outros tipos. Também deve constar nessa configuração a zona **raiz** “.” Do tipo **hint** (raiz da internet).

Por exemplo:

zone "." {

type hint;

file "named.root";

};

zone "0.0.127.in-addr.arpa" {

type master;

file "127.0.0.x";

};

zone "5.168.192.in-addr.arpa" {

type master;

file "192.168.5.x";

};

zone "how2security.com.br" {

type master;

file "how2security.com.br";

};

zone "how2security.com" {

type master;

file "how2security.com";

};

"/etc/named/named.conf" 81L, 1284C written

dns-fusion:/#

**5 – Configuração das Zonas DNS no Linux**

Os arquivos de **Zonas** são banco de dados contendo os registros sobre sua autoridade.

O **BIND** opera com 6 tipos de zonas, sendo elas:

* **master** 🡪 O **BIND** responde como autoridade sobre o domínio e o arquivo é leitura e escrita.
* **slave** 🡪 O **BIND** também responde como autoridade sobre o domínio, porém o arquivo é somente leitura. as alterações são replicadas do **Master**.
* **stub** 🡪 É uma implementação do **BIND**, não sendo padrão DNS, Ele é parecido com as **Zonas Slave**, mas replica apenas os registros **NS** do **Master**. Está em desuso.
* **hits** 🡪 São os registros dos **Servidores Root Servers**, eles são necessários quando utilizamos o **Servidor DNS** como **Servidor Cache**.
* **forward** 🡪 Encaminha as consultas sobre determinadas Zonas para outro S**ervidor DNS**. Podemos encaminhar as consultas de forma global.
* **delegation-only** 🡪 Para evitar abusos de algumas autoridades sobre os domínios de primeiro nível podemos apenas delegara Zona. Qualquer resposta que não tenha uma delegação explícita ou implícita na seção autoridade será transformada em uma resposta **NXDOMAIN**.

Estes arquivos contêm diversos tipos diferentes de registros DNS disponíveis.

Alguns deles são:

* **SOA (Start Of Authority)** 🡪 Indica o responsável por respostas autoritativas por um domínio.
  + **@ (At)** 🡪 Define a localização deste domínio.
  + **Serial** 🡪 Serve para avisar aos **Servidores Slaves** sobre as atualizações do banco de dados. O **Servidor Slave** pergunta periodicamente ao **Servidor Master** sobre o registro **SOA**. Se o número serial no registro **SOA** do **Servidor Master** for maior que do **Servidor Slave**, este transfere toda zona do **Servidor Master**. De outra forma, o **Servidor Slave** assume ter uma cópia igual e sai da transferência de zona. O número serial deve ser incrementado todas as vezes que o domínio for atualizado para manter o **Servidor Slave** sincronizado com o **Servidor Master**. O valor recomendado pela **RFC 1912**, são 10 dígitos que correspondem a ano, mês, dia e versão que é iniciado em 0 (zero).
  + **Refresh** 🡪 Intervalo de tempo, em segundos, que o **Servidor Slave** compara seu número serial com o **Servidor Master** (em segundos).
  + **Retry** 🡪 Caso o **Servidor Master** pare o seu funcionamento, o **Servidor Slave** tentará reatar comunicação com o **Servidor Master** repetidamente considerando o tempo estipulado, em segundos, em **Retry**.
  + **Expire** 🡪 Tempo de vida útil do banco de dados do domínio em um **Servidor Slave**. Caso este tempo seja ultrapassado sem ser realizado uma conexão com o **Servidor Master**, os dados ali armazenados são considerados desatualizados. Em certos casos, o Servidor poderá parar de resolver nomes DNS.
  + **Minimum (TTL- Time To Live)** 🡪 Indica o tempo de resposta de um Servidor em caso de pedido de resolução de um nome contido em seu banco de dados. Máximo de 3 horas.
* **NS (Name Server)** 🡪 Especifica **Servidores DNS** para o domínio ou subdomínio.
* **A (Address)** 🡪 Mapeamento de nomes para Endereços IP. Especificando um Endereço IP para nome direto.
* **AAAA (Address IPv6)** 🡪 Mapeamento de nomes para endereços IPv6.
* **CNAME (Canonical Name)** 🡪 Um apelido para outros hostnames (**Aliases**).
* **MX (Mail Exchange)** 🡪 O Servidor de e-mails. O registro **MX** é seguido de um número que determina a prioridade do Servidor. Quando menor o número, maior será a prioridade do Servidor. A faixa de valores numéricos permitido é de **0** até **65535**.
* **PTR (Pointer)** 🡪 Mapeamento de Endereços IP para nomes. Apontando o hostname/domínio reverso a partir de um Endereço IP.
* **TXT (Text)** 🡪 Permite incluir um texto curto em um hostname. Eles também são utilizados de acordo com o formato definido pelo projeto S**ender Policy Framework (SPF)** e que virou uma **RFC 4408** para servidor de e-mails com controle de **SPAM**. Este registro diz quem pode enviar e-mails em nome do domínio. Outra coisa forma de controle é o endereço reverso, pois garante que a fonte do e-mail é um provedor e não uma máquina zumbi.
* **WKS (Well Know Services)** 🡪 Serviços bem conhecidos.
* **HINFO (Hosat Information)** 🡪 Informações sobre um computador.
* **SRV (Services)** 🡪 Permite definir serviços disponíveis em um domínio.

Dentro do arquivo temos uma sintaxe para incluir os registros sendo:

***Dono TTL classe Tipo Dados***

Onde:

* **Dono** 🡪 É o nome do registro.
* **TTL** 🡪 Tempo de vida desse registro.
* **Classe** 🡪 O padrão é **Internet (IN)**, mas temos outros como **CH** e **HS**.
* **Tipo** 🡪 Como vimos acima **A**, **AAAA**, **CNAME**, etc.
* **Dados** 🡪 Depender do tipo de registro, pode ser o Endereço IP , um FQDN, Aliases, etc.

Temos outros e que podem ser consultados no artigo Mensagens DNS.

**5.1 – Configurando o Arquivo de Root Servers**

O arquivo de **Root Servers** contém os principais **Servidores DNS** do mundo, e devem ser atualizados regularmente. O **InterNIC**, disponibiliza via **FTP** anônimo no site deles no endereço:

dns-fusion:/var/named/namedb/master# wget ftp://ftp.rs.internic.net/domain/named.root

--2014-08-20 15:24:26-- ftp://ftp.rs.internic.net/domain/named.root

=> `named.root'

Resolving ftp.rs.internic.net (ftp.rs.internic.net)... 199.7.52.73

Connecting to ftp.rs.internic.net (ftp.rs.internic.net)|199.7.52.73|:21... connected.

Logging in as anonymous ... Logged in!

==> SYST ... done. ==> PWD ... done.

==> TYPE I ... done. ==> CWD (1) /domain ... done.

==> SIZE named.root ... 3170

==> PASV ... done. ==> RETR named.root ... done.

Length: 3170 (3,1K) (unauthoritative)

100%[==================================================>] 3.170 --.-K/s in 0s

2014-08-20 15:24:51 (56,2 MB/s) - `named.root' saved [3170]

dns-fusion:/var/named/namedb/master#

Aqui baixamos o arquivo no diretório ***/var/named/namedb/master***, pois é nosso diretório de banco de dados do **Servidor DNS** (no caso do **Servidor Master**).



**Observação:** Este arquivo no **Debian** chama-se /**etc/bind/db.root** e no **Red Hat** é ***/var/named/named.ca***.

**5.2 – Configurando o Arquivo de Zone Reverse (Zona Reversa)**

Este arquivo é a base de dados de consulta reversa que mapeia Endereços IP para nomes.

Quando alguém faz uma pesquisa do tipo **Reverse Lookup** é feito um pedido para mapear em Endereço IP para um nome. Isso é muito utilizado quando se conhece o Endereço IP, mas há a intenção de saber o nome de domínio associado a esse endereço. Por exemplo, se você monitora conexões IP feitas a um servidor, pode-se utilizar um **reverse lookup** para localizar o nome de domínio associado ao Endereço IP do computador de conexão. Esse tipo de solicitação é feito para resolução de Endereços para nomes.

dns-fusion:/var/named/namedb/master# vi /var/named/namedb/master/192.168.5.x

$TTL 43200 ; 12 Horas

@ IN SOA fusion.how2security.com.br. suporte.how2security.com.br. (

2014082000 ; Serial YYYYMMDDid

8H ; Refresh

2H ; Retry

1W ; Expire

1D) ; Minimum

how2security.com.br. IN NS fusion.how2security.com.br.

how2security.com.br. IN NS corolla.how2security.com.br.

129 IN NS corola.how2security.com.br.

128 IN PTR fusion.how2security.com.br.

129 IN PTR corola.how2security.com.br.

"192.168.5.x" [New] 12L, 343C written

dns-fusion:/var/named/namedb/master#

**Observação:** Teremos que ter uma zona reversa para cada range de Endereços IP que tivemos.

Podemos aproveitar para criarmos o **Reverso do Loopback**, ficando assim:

dns-fusion:/var/named/namedb/master# vi /var/named/namedb/master/127.0.0.x

$TTL 3600

@ IN SOA localhost. suporte.localhost. (

2014082000 ; Serial YYYYMMDDid

1H ; Refresh

2H ; Retry

1W ; Expire

1D) ; Minimum

how2security.com.br. IN NS localhost.

1 IN PTR localhost.

"127.0.0.x" [New] 10L, 191C written

**5.3 – Configurando o Arquivo de Zone Forward (Zona Direta)**

Este arquivo é a base de dados de consulta direta que mapeia nomes para Endereços IP.

Este é o tipo mais comum de pesquisa, sendo utilizado para localizar um Endereço IP de modo que possa ser feita uma conexão a esse servidor. Esse tipo de solicitação é feito para resolução de nomes para Endereço IP.

dns-fusion:/var/named/namedb/master# vi /var/named/namedb/master/how2security.com.br

$TTL 43200 ; 12 Horas

@ IN SOA fusion.how2security.com.br. suporte.how2security.com.br. (

2014082000 ; Serial YYYYMMDDid

8H ; Refresh

2H ; Retry

1W ; Expire

1D) ; Minimum

how2security.com.br. IN NS fusion.how2security.com.br.

how2security.com.br. IN NS corola.how2security.com.br.

@ IN MX 1 fusion.how2security.com.br.

129 IN MX 5 corola.how2security.com.br.

how2security.com.br. IN TXT "v=spf1 +mx a ip4:192.168.5.0/24 -all"

@ IN SPF "v=spf1 +mx a ip4:192.168.5.0/24 -all"

fusion IN A 192.168.5.128

corola IN A 192.168.5.129

mail IN CNAME fusion.how2security.com.br.

mail2 IN CNAME corola.how2security.com.br.

www IN CNAME www.how2security.com.br.

"how2security.com.br" [New] 18L, 594C written

dns-fusion:/var/named/namedb/master#

**5.4 – Configurando o Arquivo HOSTS**

Os arquivos hosts era a forma de resolução de nomes no passado. Ele continua tendo uma importância no processo de resolução de nomes, pois ele é a segunda fonte a ser consultada pelo **Resolver**.

**Observação:** Este comportamento pode ser alterado através do arquivo ***/etc/nsswitch***.

dns-fusion:~# vi /etc/hosts

192.168.5.128 fusion.how2security.com.br fusion

127.0.0.1 localhost

::1 localhost ip6-localhost ip6-loopback

fe00::0 ip6-localnet

ff00::0 ip6-mcastprefix

ff02::1 ip6-allnodes

ff02::2 ip6-allrouters

"/etc/hosts" 9L, 206C written

dns-fusion:~#

Aqui é configurado um endereço por linha, onde temos, **Endereço IP**, **Hostname Canônico** e um **Aliases**. O é desejável que o **Hostname Canônico** seja o **FQDN** e o **Aliases** pode ser outro nome (apelido) ou o sufixo do nome.

**5.5 – Configurando o Arquivo de Inicialização do Serviço**

Dentro da **Jail** temos que criar o arquivo que irá inicializar o serviço do **BIND**, para isso vamos criar o arquivo em ***/etc/init.d/dns***:

dns-fusion:/# vi /etc/init.d/dns

#!/bin/bash

#

### BEGIN INIT INFO

# Provides: dns

# Required-Start: $remote\_fs

# Required-Stop: $remote\_fs

# Should-Start: $network $syslog

# Should-Stop: $network $syslog

# Default-Start: 2 3 4 5

# Default-Stop: 0 1 6

# Short-Description: Iniciar e Parar o Servico DNS/BIND9

# Description: O BIND eh um Servidor DNS para traduzir enderecos IP para Nome e Vive-Versa

#

### END INIT INFO

### CORES

amarelo="\e[33;1m"

azul="\e[34;1m"

verde="\e[32;1m"

vermelho="\e[31;1m"

fim="\e[m"

PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin

DNS=`which named`

RNDC=`which rndc`

dns\_start()

{

$DNS -u named & 2>1&

}

dns\_stop()

{

$RNDC stop 2>1&

}

dns\_restart()

{

dns\_stop

sleep 2

dns\_start

}

dns\_reload()

{

$RNDC reload > /dev/null

}

dns\_status()

{

$RNDC status

}

dns\_reconfig()

{

$RNDC reconfig

}

case $1 in

start)

echo -e "$verde[-]Iniciando o Servico de Nonme de Dominio...$fim"

dns\_start

;;

stop)

echo -e "$vermelho[-]Parando o Sertvico de Nome de Dominio...$fim"

dns\_stop

;;

restart)

echo -e "$azul[-]Reiniciando o Servico de Nome de Dominio...$fim"

dns\_restart

;;

reload|force-reload)

echo -e "$azul[-]Recarregando o Servico de Nome de Dominio...$fim"

dns\_reload

;;

status)

echo -e "$amarelo[-]Verificando o Status do Servico de Nome de Dominio...$fim"

dns\_status

;;

reconfig)

echo -e "$azul[-]Reconfigurando o Servico de Nome de Dominio...$fim"

dns\_reconfig

;;

\*)

echo -e "$vermelho[-]ERRO: Use $0 {start|stop|reload|force-reload|restart|reconfig|status}$fim"

exit 1

;;

esac

exit 0

"/etc/init.d/dns" 56L, 851C written

dns-fusion:/#

Agora iremos transformar nosso script em executáveis e criar os links simbólicos nos **rcX.d** assim:

dns-fusion:~# chmod 700 /etc/init.d/dns

dns-fusion:~# update-rc.d -f dns defaults

update-rc.d: using dependency based boot sequencing

dns-fusion:~#

Testando nosso script.

dns-fusion:/# /etc/init.d/dns start

**[-]Iniciando o Servico de Nonme de Dominio...**

dns-fusion:/# /etc/init.d/dns status

**[-]Verificando o Status do Servico de Nome de Dominio...**

WARNING: key file (/etc/named/rndc.key) exists, but using default configuration file (/etc/named/rndc.conf)

version: 9.10.0rc2 <id:a326778a>

boot time: Wed, 27 Aug 2014 14:37:30 GMT

last configured: Wed, 27 Aug 2014 14:37:30 GMT

CPUs found: 1

worker threads: 1

UDP listeners per interface: 1

number of zones: 5

debug level: 0

xfers running: 0

xfers deferred: 0

soa queries in progress: 0

query logging is OFF

recursive clients: 0/0/1000

tcp clients: 0/100

server is up and running

dns-fusion:/# /etc/init.d/dns restart

**[-]Reiniciando o Servico de Nome de Dominio...**

dns-fusion:/# /etc/init.d/dns reload

**[-]Recarregando o Servico de Nome de Dominio...**

WARNING: key file (/etc/named/rndc.key) exists, but using default configuration file (/etc/named/rndc.conf)

dns-fusion:/# /etc/init.d/dns stop

**[-]Parando o Sertvico de Nome de Dominio...**

dns-fusion:/#

Agora temos que garantir que se o host reiniciar os serviços da **Jail** que irá subir, para isso, devemos criar o script de inicialização fora da **Jail**. Vejamos:

root@fusion:~# vi /etc/init.d/jail

#!/bin/bash

#

### BEGIN INIT INFO

# Provides: jail

# Required-Start: $remote\_fs

# Required-Stop: $remote\_fs

# Should-Start: $network $syslog

# Should-Stop: $network $syslog

# Default-Start: 2 3 4 5

# Default-Stop: 0 1 6

# Short-Description: Iniciar e Parar o Servico Enjauldados

# Description: Os serviços enjaulados necessitam ser iniciados|parados diretamente do Host

#

### END INIT INFO

### CORES

amarelo="\e[33;1m"

azul="\e[34;1m"

verde="\e[32;1m"

vermelho="\e[31;1m"

fim="\e[m"

### VARIAVEIS

JAIL="/opt/dns"

MNT=`which mount`

UMNT=`which umount`

CHRT=`which chroot`

DNS="/etc/init.d/dns"

LOG="/etc/init.d/rsyslog"

jail\_start()

{

echo -e "$verde[-]Iniciando Servicos Enjaulados...$fim"

$MNT -o bind /proc $JAIL/proc

$MNT -o bind /dev $JAIL/dev

$CHRT $JAIL $LOG start

$CHRT $JAIL $DNS start

}

jail\_stop()

{

echo -e "$vermelho[-]Parando Servicos Enjaulados...$fim"

$CHRT $JAIL $DNS stop

$UMNT $JAIL/dev

$UMNT $JAIL/proc

}

jail\_restart()

{

echo -e "$azul[-]Reiniciando Servicos Enjaulados...$fim"

$CHRT $JAIL $DNS restart

}

case $1 in

start)

jail\_start

;;

stop)

jail\_stop

;;

restart)

jail\_restart

;;

\*)

echo -e "$vermelho[-]ERRO: Use $0 {start|stop|restart}$fim"

exit 1

;;

esac

exit 0

"/etc/init.d/jail" 70L, 1325C written

root@fusion:~# chmod 700 /etc/init.d/jail

root@fusion:~# update-rc.d -f jail defaults

update-rc.d: using dependency based boot sequencing

root@fusion:~# /etc/init.d/jail restart

**[-]Reiniciando Servicos Enjaulados...**

**[-]Reiniciando o Servico de Nome de Dominio...**

root@fusion:~#

**Observação:** Se após configurar e tentar iniciar o **BIND** nada acontecer, execute o **Daemon Named** com a opção de **Debug** assim:

dns-fusion:/# named -u named –g

27-Aug-2014 15:12:20.164 starting BIND 9.10.0rc2 -u named -g

27-Aug-2014 15:12:20.164 built with '--with-openssl=/usr/local/ssl' '--sysconfdir=/etc/named'

27-Aug-2014 15:12:20.164 ----------------------------------------------------

27-Aug-2014 15:12:20.165 BIND 9 is maintained by Internet Systems Consortium,

27-Aug-2014 15:12:20.165 Inc. (ISC), a non-profit 501(c)(3) public-benefit

27-Aug-2014 15:12:20.165 corporation. Support and training for BIND 9 are

27-Aug-2014 15:12:20.165 available at https://www.isc.org/support

27-Aug-2014 15:12:20.165 ----------------------------------------------------

27-Aug-2014 15:12:20.165 adjusted limit on open files from 4096 to 1048576

27-Aug-2014 15:12:20.165 found 1 CPU, using 1 worker thread

27-Aug-2014 15:12:20.166 using 1 UDP listener per interface

27-Aug-2014 15:12:20.166 using up to 4096 sockets

27-Aug-2014 15:12:20.171 loading configuration from '/etc/named/named.conf'

27-Aug-2014 15:12:20.171 reading built-in trusted keys from file '/etc/named/bind.keys'

--==[ Resumido ]==--

27-Aug-2014 15:12:20.190 all zones loaded

27-Aug-2014 15:12:20.191 running

Desta forma os erros serão mostrados na console.

**6 – Testando o BIND no Linux**

Podemos testar nossas configurações de DNS utilizando algumas das ferramentas que vem no próprio **BIND** e outras que estão presentes no pacote **dnsutils**.

**6.1 – Testando com o Named-Checkconf**

Com esse utilitário podemos validar se nosso arquivo de configuração está correto. Casa haja algum erro, seremos informados em qual linha está a inconsistência e uma breve descrição.

dns-fusion:~# named-checkconf

dns-fusion:~# echo " {;;;;;;}" >> /etc/named/named.conf

dns-fusion:~# named-checkconf

/etc/named/named.conf:81: syntax error near '{'

dns-fusion:~#

Como podemos observar a primeira vez em que executamos o utilitário não tinha erro em nosso arquivo de configuração, após isso, inserimos uma linha sem sentido e executamos novamente o comando e eles nos informou que na **linha 81** do arquivo de configuração tem uma inconsistência.

**6.2 – Testando com o Named-Checkzone**

Semelhante ao utilitário anterior, porém, este analisa os arquivos de banco de dados de zonas. Para testarmos basta executar o utilitário seguido da zona a qual queremos consultar e se arquivo de banco de dados.

dns-fusion:~# named-checkzone how2security.com.br /var/named/namedb/master/how2security.com.br

zone how2security.com.br/IN: 'how2security.com.br' found SPF/TXT record but no SPF/SPF record found, add matching type SPF record

zone how2security.com.br/IN: loaded serial 2014082000

OK

dns-fusion:~#

Observe que tivemos uma inconsistência no arquivo de zona do nosso domínio, e que o problema está no registro **SPF**. Por isso, vamos colocá-lo em conformidade com a **RFC 4408**.

Iremos alterar de:

@ IN TXT "v=spf1 a mx 192.168.5.0/24 - all"

Para :

how2security.com.br. IN TXT "v=spf1 +mx a ip4:192.168.5.0/24 -all"

@ IN SPF "v=spf1 +mx a ip4:192.168.5.0/24 -all"

O resultado agora é:

dns-fusion:~# named-checkzone how2security.com.br /var/named/namedb/master/how2security.com.br

zone how2security.com.br/IN: loaded serial 2014082000

OK

dns-fusion:~#

**6.3 – Testando com o NSLOOKUP**

A **ISC** não recomenda a utilização do **NSLOOKUP**, porém eles mantêm o pacote devido a legião de usuários que utilizam a ferramenta. Uma das poucas vantagens do **NSLOOKUP** é a utilização de uma biblioteca de resolução independente do **Resolver** e consultar um servidor por vez. Dentre as desvantagens está o fato de entregar respostas confusas e erros indefinidos.

root@fusion:~# nslookup

> www

Server: 192.168.5.129

Address: 192.168.5.129#53

www.how2security.com.br canonical name = fusion.how2security.com.br.

Name: fusion.how2security.com.br

Address: 192.168.5.128

> 192.168.5.128

Server: 192.168.5.129

Address: 192.168.5.129#53

128.5.168.192.in-addr.arpa name = fusion.how2security.com.br.

> set type=mx

> how2security.com.br

Server: 192.168.5.129

Address: 192.168.5.129#53

how2security.com.br mail exchanger = 1 fusion.how2security.com.br.

> set type=soa

> how2security.com.br

Server: 192.168.5.129

Address: 192.168.5.129#53

how2security.com.br

origin = fusion.how2security.com.br

mail addr = suporte.how2security.com.br

serial = 2014082000

refresh = 28800

retry = 7200

expire = 604800

minimum = 86400

> server 8.8.8.8

Default server: 8.8.8.8

Address: 8.8.8.8#53

> www

Server: 8.8.8.8

Address: 8.8.8.8#53

Non-authoritative answer:

www.how2security.com.br canonical name = how2security.com.br.

how2security.com.br

origin = a.sec.dns.br

mail addr = hostmaster.registro.br

serial = 2014235000

refresh = 345600

retry = 900

expire = 604800

minimum = 900

Authoritative answers can be found from:

>

Aqui podemos observar algumas consultas diretas e recursivas, especificando o tipo de registro a ser consultado e até mesmo alterando o **Servidor DNS** para ver a resposta deles sobre o nosso domínio.

**6.4 – Testando com o HOST**

O utilitário HOST entrega respostas objetivas, sua vagagem é a simplicidade. Ele utiliza as configurações de DNS do Resolver para fazer as consultas. Iremos fazer as consultas anteriores, porém utilizando o **HOST**

root@fusion:~# host www

www.how2security.com.br is an alias for fusion.how2security.com.br.

fusion.how2security.com.br has address 192.168.5.128

root@fusion:~# host -v www.how2security.com.br.

Trying "www.how2security.com.br"

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52122

;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:

;www.how2security.com.br. IN A

;; ANSWER SECTION:

www.how2security.com.br. 43200 IN CNAME fusion.how2security.com.br.

fusion.how2security.com.br. 43200 IN A 192.168.5.128

;; AUTHORITY SECTION:

how2security.com.br. 43200 IN NS fusion.how2security.com.br.

Received 92 bytes from 192.168.5.129#53 in 3 ms

Trying "fusion.how2security.com.br"

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 50100

;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:

;fusion.how2security.com.br. IN AAAA

;; AUTHORITY SECTION:

how2security.com.br. 43200 IN SOA fusion.how2security.com.br. suporte.how2security.com.br. 2014082000 28800 7200 604800 86400

Received 88 bytes from 192.168.5.129#53 in 1 ms

Trying "fusion.how2security.com.br"

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19367

;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:

;fusion.how2security.com.br. IN MX

;; AUTHORITY SECTION:

how2security.com.br. 43200 IN SOA fusion.how2security.com.br. suporte.how2security.com.br. 2014082000 28800 7200 604800 86400

Received 88 bytes from 192.168.5.129#53 in 0 ms

root@fusion:~# host -v -t soa www.how2security.com.br.

Trying "www.how2security.com.br"

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40083

;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:

;www.how2security.com.br. IN SOA

;; ANSWER SECTION:

www.how2security.com.br. 43200 IN CNAME fusion.how2security.com.br.

;; AUTHORITY SECTION:

how2security.com.br. 43200 IN SOA fusion.how2security.com.br. suporte.how2security.com.br. 2014082000 28800 7200 604800 86400

Received 106 bytes from 192.168.5.129#53 in 1 ms

root@fusion:~# host -l -v -t any www.how2security.com.br.

Trying "www.how2security.com.br"

; Transfer failed.

Trying "www.how2security.com.br"

Host www.how2security.com.br not found: 9(NOTAUTH)

Received 41 bytes from 192.168.5.129#53 in 0 ms

; Transfer failed.

root@fusion:~#

Observe que a última consulta solicitamos uma consulta **AXFR**, que iria transferir a base de dados através da consulta. Caso seu **Servidor DNS** respondo essa consulta você corre um risco de segurança.

**6.5 – Testando com o DIG**

O utilitário DIG é o acrônimo para **Domain Information Groper**, que significa Busca por Informações de Domínio no Escuro. E inglês a palavra **DIG** significa “**Escavar**”. O DIG é a ferramenta mais poderosa de consulta do **BIND**.

O **DIG** não utiliza as configurações de DNS do Resolver, por isso, é necessário utilizar o FQDN em todas as consultas.

**7 – Configurações de Segurança no BIND do Linux**

Aqui veremos algumas configurações para deixar nosso Servidor **DNS BIND** mais seguro.

**7.1 – Removendo o Banner de Versão**

Para ocultar a versão nas consultas ao **Servidor DNS** basta incluirmos no arquivo de configuração do **BIND** (***named.conf***), em **options** a seguinte opção:

dns-fusion:/# vi /etc/named/named.conf

--==[ Resumido ]==--

acl net\_how2security {

192.168.5.0/24;

};

options {

directory "/var/named/namedb/master";

listen-on { 127.0.0.1; 192.168.5.128; };

listen-on-v6 { none; };

empty-zones-enable no;

**version "Not Available";**

allow-transfer { srv\_slaves; };

allow-recursion { net\_how2security; };

allow-query { any; };

};

"/etc/named/named.conf" 82L, 1311C written

dns-fusion:/#

Basta reiniciar o BIND para não ser visualizada nas consultas a versão do BIND.

**7.2 – Configurar o TSIG para Transferências de Base Entre Master e o Slave**

Para garantir quais **Servidores Slaves** irão sincronizar suas bases com nosso **Servidor Master** podemos implementar o **TSIG**, que utilizar chaves simétricas para garantir a autorização de transferências de base de dados do DNS.

**7.2.1 – Gerando a Chave do TSIG no Servidor Master**

Para utilizarmos o **TSIG** necessitamos criar a chave simétrica para garantirmos que apenas os **Servidores Slaves** que tenha essa chave que iremos criar, possa sincronizar as bases.

Para a criação da chave utilizaremos o utilitário **dnssec-keygen**. Esse utilitário vem junto com o **BIND** para gerar chaves para o **DNSSEC** e o **TSIG**. Vamos executar o seguinte comando em nosso **Servidor Master**.

dns-fusion:/etc/named# dnssec-keygen -a HMAC-SHA512 -b 512 -n HOST -r /dev/urandom how2security.com.br

Khow2security.com.br.+165+46427

Com isso, geramos a chave com as seguintes opções:

* **-a** 🡪 Especifica o algoritmo a ser utilizado Os algoritmo suportados pelo **TSIG** são: **HMAC-MD5**, **HMAC-SHA1**, **HMAC-SHA224**, **HMAC-SHA256**, **HMAC-SHA384** ou **HMAC-SHA512**. No nosso caso utilizamos o **HMAC-SHA512**.
* **-b** 🡪Especifica o tamanho da chave que pode variar entre **512** e **2048**. No nosso caso utilizamos **512**.
* **-n** 🡪 Especifica o tipo do dono da chave que pode ser **HOST**, **ZONE**, **ENTITY**, **USER**.
* **-r** 🡪Especifica uma origem randômica. No nosso caso utilizamos ***/dev/urandom***.

Foram gerados dois arquivos com o nome ***Khow2security.com.br.+165+46427.key*** e ***Khow2security.com.br.+165+46427.private*** Agora devemos pegar a chave gerada para criar o arquivo com o valor da chave. Para isso iremos fazer o seguinte:

dns-fusion:/etc/named# grep "Key" Khow2security.com.br.+165+46427.private

**Key**: vVpfdb5VmizK5gj0zB+yvJHoq52N16ULTrlH9DtsAkH2jDwrNhTqBIpqJ8mjfyu1TnjxJ9k5jqr0qa5qdOoPKw==

Agora vamos criar o arquivo com a assinatura do **TSIG**, o nome do arquivo será ***tsig.key*** e devemos fazer um include no arquivo de configuração ***/etc/named/named.conf***. Vejamos:

dns-fusion:/etc/named# vi tsig.key

key "how2sec" {

algorithm HMAC-SHA512;

secret "vVpfdb5VmizK5gj0zB+yvJHoq52N16ULTrlH9DtsAkH2jDwrNhTqBIpqJ8mjfyu1TnjxJ9k5jqr0qa5qdOoPKw==";

};

"tsig.key" [New] 4L, 155C written

dns-fusion:/etc/named# echo "include \"/etc/named/tsig.key\";" >> named.conf

Aqui criamos a chave com o nome “**how2sec**”, onde informamos o algoritmo usado e a chave secreta.

Agora vamos alterar nosso arquivo de configuração para garantir que nenhum servidor irá atualizar sua base com nosso **Servidor Master**. As alterações foram a remoção da **ACL** dos **Servidores Slaves** e o acréscimo das linhas em vermelho:

dns-fusion:/etc/named# vi named.conf

acl net\_how2security {

192.168.5.0/24;

};

options {

directory "/var/named/namedb/master";

listen-on { 127.0.0.1; 192.168.5.128; };

listen-on-v6 { none; };

empty-zones-enable no;

version "Not Available";

auth-nxdomain no; # conform to RFC1035

**allow-transfer { none; };**

allow-recursion { net\_how2security; };

allow-query { any; };

};

key "rndc.key" {

algorithm hmac-sha512;

secret "AqRRT8jO28q7WIUNrgCgu/THmZy4+2nHOZhPJF9coqyA5wQDBp1vSVKZJUT6N47j0B+j85s50ulLn8MsOrROuQ==";

};

controls {

inet 127.0.0.1 allow { 127.0.0.1; } keys { "rndc.key"; };

};

logging {

channel "named\_log" {

file "/var/log/named/named.log" versions 2 size 50m;

print-time yes;

print-category yes;

};

category "security" {

"named\_log";

};

category "xfer-out" {

"named\_log";

};

category "xfer-in" {

"named\_log";

};

category "general" {

"named\_log";

};

};

zone "." {

type hint;

file "named.root";

};

zone "0.0.127.in-addr.arpa" {

type master;

file "127.0.0.x";

};

zone "5.168.192.in-addr.arpa" {

type master;

file "192.168.5.x";

};

zone "how2security.com.br" {

type master;

file "how2security.com.br";

**allow-transfer {**

**key "how2sec";**

**};**

};

zone "how2security.com" {

type master;

file "how2security.com";

};

**include "/etc/named/tsig.key";**

"named.conf" 86L, 1393C written

dns-fusion:

Em **options** negamos a transferência de zonas a todos os servidores, e na **Zona** **how2security.com.br** autorizamos a transferência para os **Servidores Slaves** que tenham a chave **how2sec**.

Para finalizar devemos reiniciar o serviço do BIND e transferir a chave para os **Servidores Slaves**.

dns-fusion:/etc/named# /etc/init.d/dns restart

[-]Reiniciando o Servico de Nome de Dominio...

dns-fusion:/etc/named# exit

exit

root@fusion:~# scp /opt/dns/etc/named/tsig.key root@192.168.5.129:/opt/dns/etc/bind/

The authenticity of host '192.168.5.129 (192.168.5.129)' can't be established.

ECDSA key fingerprint is 8c:92:f2:26:10:2e:8e:33:4e:72:da:fb:ad:6e:6c:f1.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '192.168.5.129' (ECDSA) to the list of known hosts.

root@192.168.5.129's password:

tsig.key 100% 155 0.2KB/s 00:00

root@fusion:~# chroot /opt/dns /bin/bash

dns-fusion:/#

**7.2.2 – Configurando o TSIG no Servidor Slave**

Primeiramente vamos incluir o arquivo ***tsig.key*** no arquivo de configuração do **Servidor Slave**, e depois vamos alterar o arquivo de chaves do **TSIG** que transferimos do **Servidor Master**, essa alteração é justamente para criar a função do Servidor Master que utiliza a chave simétrica “**how2sec**.

dns-corola:~# cd /etc/bind/

dns-corola:/etc/bind# echo "include \"/etc/bind/tsig.key\";" >> /etc/bind/named.conf

dns-corola:/etc/bind# vi tsig.key

key "how2sec" {

algorithm HMAC-SHA512;

secret "vVpfdb5VmizK5gj0zB+yvJHoq52N16ULTrlH9DtsAkH2jDwrNhTqBIpqJ8mjfyu1TnjxJ9k5jqr0qa5qdOoPKw==";

};

server 192.168.5.128 {

keys { how2sec; };

};

"tsig.key" 8L, 214C written

dns-corola:/etc/bind#

Após isso, basta reiniciar o serviço do BIND que está pronto.

dns-corola:/etc/bind# /etc/init.d/bind9 restart

Stopping domain name service...: bind9waiting for pid 5209 to die

.

Starting domain name service...: bind9.

dns-corola:/#

**7.2.3 – Testando o TSIG**

No **Servidor Master** vamos testar a transferência com o utilitário **DIG** para testarmos se ele só responde a solicitação de transferência de Servidores que tenha a chave. Primeiramente vamos solicitar a transferência sem as chaves

dns-fusion:/etc/named# dig @127.0.0.1 how2security.com.br axfr

; <<>> DiG 9.10.0rc2 <<>> @127.0.0.1 how2security.com.br axfr

; (1 server found)

;; global options: +cmd

**; Transfer failed.**

dns-fusion:

Como podemos observar a transferência falhou, pois não utilizamos a chave do **TSIG**. Agora vamos apontar a chave para ver a resposta do servidor:

dns-fusion:/etc/named# dig @127.0.0.1 how2security.com.br axfr -k /etc/named/tsig.key

; <<>> DiG 9.10.0rc2 <<>> @127.0.0.1 how2security.com.br axfr -k /etc/named/tsig.key

; (1 server found)

;; global options: +cmd

how2security.com.br. 43200 IN SOA fusion.how2security.com.br. suporte.how2security.com.br. 2014082000 28800 7200 604800 86400

how2security.com.br. 43200 IN TXT "v=spf1 +mx a ip4:192.168.5.0/24 -all"

how2security.com.br. 43200 IN SPF "v=spf1 +mx a ip4:192.168.5.0/24 -all"

how2security.com.br. 43200 IN MX 1 fusion.how2security.com.br.

how2security.com.br. 43200 IN NS fusion.how2security.com.br.

129.how2security.com.br. 43200 IN MX 5 corola.how2security.com.br.

129.how2security.com.br. 43200 IN NS corola.how2security.com.br.

corola.how2security.com.br. 43200 IN A 192.168.5.129

fusion.how2security.com.br. 43200 IN A 192.168.5.128

mail.how2security.com.br. 43200 IN CNAME fusion.how2security.com.br.

mail2.how2security.com.br. 43200 IN CNAME corola.how2security.com.br.

www.how2security.com.br. 43200 IN CNAME fusion.how2security.com.br.

how2security.com.br. 43200 IN SOA fusion.how2security.com.br. suporte.how2security.com.br. 2014082000 28800 7200 604800 86400

**how2sec. 0 ANY TSIG hmac-sha512. 1410289038 300 64 hld92ovVDeHhiPSJG7URrXGMVHFcPxMmSoKvkPKn3zMsIpTMV0TVSm7w UQphHDYlkUg33zg+Buwt7TJ+/tIAvA== 2660 NOERROR 0**

;; Query time: 93 msec

;; SERVER: 127.0.0.1#53(127.0.0.1)

;; WHEN: Tue Sep 09 18:57:18 UTC 2014

;; XFR size: 13 records (messages 1, bytes 505)

dns-fusion:/etc/named#

Agora sim obtivemos a resposta a solicitação de transferência, pois indicamos a chave do **TSIG**.

Ainda no **Servidor Master** vamos observar os logos antes de testarmos a transferência no **Servidor Slave**.

dns-fusion:/etc/named# tail -f /opt/dns/var/log/named/named.log

Vamos forçar a transferência da base do DNS apagando as bases contidas no **Servidor Slave** e reiniciando o serviço do BIND:

dns-corola:/# tail -f /var/log/named/named.log

Agora sim iremos forçar a replicação e analisar os logs.

dns-corola:/etc/bind# rm /var/cache/bind/how2security.com\*

dns-corola:/etc/bind# rm /var/cache/bind/192.168.5.x

dns-corola:/etc/bind# /etc/init.d/bind9 restart

Stopping domain name service...: bind9waiting for pid 5209 to die

.

Starting domain name service...: bind9.

dns-corola:/#

Olhe os logs do Servidor Master.

dns-fusion:/etc/named# tail -f /opt/dns/var/log/named/named.log

09-Sep-2014 20:06:25.929 security: client 192.168.5.129#34836/key how2sec (5.168.192.in-addr.arpa): zone transfer '5.168.192.in-addr.arpa/AXFR/IN' **denied**

09-Sep-2014 20:06:26.434 xfer-out: client 192.168.5.129#39571/key how2sec **(how2security.com.br): transfer of 'how2security.com.br/IN': AXFR started: TSIG how2sec (serial 2014082000)**

**09-Sep-2014 20:06:26.435 xfer-out: client 192.168.5.129#39571/key how2sec (how2security.com.br): transfer of 'how2security.com.br/IN': AXFR ended**

09-Sep-2014 20:06:26.436 security: client 192.168.5.129#59428/key how2sec (how2security.com): zone transfer 'how2security.com/AXFR/IN' **denied**

Como podemos observar a única **Zona** que tem permissão para transferência é a Zona **how2security.com.br**, as outras não tem permissão, fiz isso propositalmente para vermos que apenas os servidores que contém a chave e a política de transferência de base irá transferir a base.

Para que todas as bases possam transferir basta configuramos as Zonas no arquivo ***named.conf*** do **Servidor Master** da seguinte forma.

dns-fusion:/# vi /etc/named/named.conf

--==[ Resumido ]==--

zone "5.168.192.in-addr.arpa" {

type master;

file "192.168.5.x";

**allow-transfer {**

**key "how2sec";**

**};**

};

zone "how2security.com.br" {

type master;

file "how2security.com.br";

**allow-transfer {**

**key "how2sec";**

**};**

};

zone "how2security.com" {

type master;

file "how2security.com";

**allow-transfer {**

**key "how2sec";**

**};**

};

include "/etc/named/tsig.key";

"/etc/named/named.conf" 83L, 1309C written

dns-fusion:/# /etc/init.d/dns restart

[-]Reiniciando o Servico de Nome de Dominio...

dns-fusion:/#

Agora vamos abrir outro terminal no **Servidor Slave** e deixar o log sendo analisado, que iremos analisa-lo após reiniciar o serviço do BIND

dns-corola:/# tail -f /var/log/named/named.log

Agora vamos apagar o arquivo da base ***how2security.com.br*** e reiniciar o serviço do BIND para analisarmos os logs dos servidores.

dns-corola:/etc/bind# rm /var/cache/bind/how2security.com.br

dns-corola:/etc/bind# /etc/init.d/bind9 restart

Stopping domain name service...: bind9waiting for pid 5435 to die

.

Starting domain name service...: bind9.

dns-corola:/etc/bind#

Agora vamos analisar o log do Servidor Master:

dns-fusion:/etc/named# tail -f /opt/dns/var/log/named/named.log

09-Sep-2014 20:11:23.640 general: received control channel command 'stop'

09-Sep-2014 20:11:23.642 general: shutting down: flushing changes

09-Sep-2014 20:11:23.642 general: stopping command channel on 127.0.0.1#953

09-Sep-2014 20:11:23.655 general: exiting

09-Sep-2014 20:11:25.685 general: managed-keys-zone: loaded serial 0

09-Sep-2014 20:11:25.687 general: zone 0.0.127.in-addr.arpa/IN: loaded serial 2014082000

09-Sep-2014 20:11:25.687 general: zone 5.168.192.in-addr.arpa/IN: loaded serial 2014082000

09-Sep-2014 20:11:25.688 general: zone how2security.com.br/IN: loaded serial 2014082000

09-Sep-2014 20:11:25.688 general: zone how2security.com/IN: 'how2security.com' found SPF/TXT record but no SPF/SPF record found, add matching type SPF record

09-Sep-2014 20:11:25.688 general: zone how2security.com/IN: loaded serial 2014082000

09-Sep-2014 20:11:25.689 general: all zones loaded

09-Sep-2014 20:11:25.689 general: running

**09-Sep-2014 20:11:59.550 xfer-out: client 192.168.5.129#37406/key how2sec (5.168.192.in-addr.arpa): transfer of '5.168.192.in-addr.arpa/IN': AXFR started: TSIG how2sec (serial 2014082000)**

**09-Sep-2014 20:11:59.551 xfer-out: client 192.168.5.129#37406/key how2sec (5.168.192.in-addr.arpa): transfer of '5.168.192.in-addr.arpa/IN': AXFR ended**

**09-Sep-2014 20:12:00.051 xfer-out: client 192.168.5.129#52603/key how2sec (how2security.com): transfer of 'how2security.com/IN': AXFR started: TSIG how2sec (serial 2014082000)**

**09-Sep-2014 20:12:00.052 xfer-out: client 192.168.5.129#52603/key how2sec (how2security.com): transfer of 'how2security.com/IN': AXFR ended**

E por fim vamos analisar o log do Servidor Slave:

dns-corola:/# tail -f /var/log/named/named.log

09-Sep-2014 20:11:58.451 general: received control channel command 'stop -p'

09-Sep-2014 20:11:58.452 general: shutting down: flushing changes

09-Sep-2014 20:11:58.452 general: stopping command channel on 127.0.0.1#953

09-Sep-2014 20:11:58.452 general: stopping command channel on ::1#953

09-Sep-2014 20:11:58.463 general: exiting

09-Sep-2014 20:11:59.528 general: zone 0.in-addr.arpa/IN: loaded serial 1

09-Sep-2014 20:11:59.528 general: zone 127.in-addr.arpa/IN: loaded serial 1

09-Sep-2014 20:11:59.529 general: zone 0.0.127.in-addr.arpa/IN: loaded serial 2014082000

09-Sep-2014 20:11:59.533 general: zone 255.in-addr.arpa/IN: loaded serial 1

09-Sep-2014 20:11:59.536 general: zone how2security.com.br/IN: loaded serial 2014082000

09-Sep-2014 20:11:59.536 general: zone localhost/IN: loaded serial 2

09-Sep-2014 20:11:59.537 general: managed-keys-zone ./IN: loaded serial 7

09-Sep-2014 20:11:59.543 general: running

09-Sep-2014 20:11:59.547 general: zone 5.168.192.in-addr.arpa/IN: Transfer started.

09-Sep-2014 20:11:59.549 xfer-in: transfer of '5.168.192.in-addr.arpa/IN' from 192.168.5.128#53: connected using 192.168.5.129#37406

**09-Sep-2014 20:11:59.560 general: zone 5.168.192.in-addr.arpa/IN: transferred serial 2014082000: TSIG 'how2sec'**

**09-Sep-2014 20:11:59.560 xfer-in: transfer of '5.168.192.in-addr.arpa/IN' from 192.168.5.128#53: Transfer completed: 1 messages, 6 records, 329 bytes, 0.010 secs (32900 bytes/sec)**

**09-Sep-2014 20:12:00.049 general: zone how2security.com/IN: Transfer started.**

**09-Sep-2014 20:12:00.050 xfer-in: transfer of 'how2security.com/IN' from 192.168.5.128#53: connected using 192.168.5.129#52603**

**09-Sep-2014 20:12:00.065 general: zone how2security.com/IN: transferred serial 2014082000: TSIG 'how2sec'**

**09-Sep-2014 20:12:00.066 xfer-in: transfer of 'how2security.com/IN' from 192.168.5.128#53: Transfer completed: 1 messages, 12 records, 438 bytes, 0.015 secs (29200 bytes/sec)**

**7.3 – Configurar do DNSSEC Para Consultas Seguras**

O **DNSSEC (Domain Name System SECurity extensions)** é uma extensão do DNS, possibilitando uma maior segurança para os usuários.

Com isso, você garante a autenticidade e a integridade das consultas.

**7.3.1 – Criando as Chaves do DNSSEC**

Vamos criar o par de chaves (pública e privada) com o utilitário **dnssec-keygen**.

dns-fusion:/etc/named#

dns-fusion:/etc/named# cd /var/named/namedb/master/

dns-fusion:/var/named/namedb/master# dnssec-keygen -r /dev/urandom -f KSK -a RSASHA512 -b 2048 -n ZONE how2security.com.br

Generating key pair..............+++ ............................................................+++

Khow2security.com.br.+010+03426

dns-fusion:/var/named/namedb/master#

Onde temos as seguintes opções:

* **-a** 🡪 Especifica o algoritmo a ser utilizado Os algoritmo suportados pelo **DNSSEC** são: **RSAMD5**, **RSASHA1**, **DSA**, **NSEC3RSASHA1**, **NSEC3DSA**, **RSASHA256**, **RSASHA512**, **ECCGOST**, **ECDSAP256SHA256** ou **ECDSAP384SHA384**. No nosso caso utilizamos o **RSASHA512**.
* **-b** 🡪 Especifica o tamanho da chave que pode variar entre **512** e **2048**. No nosso caso utilizamos **2048**.
* **-f** 🡪 Define o tipo da chave.
* **-n** 🡪 Especifica o tipo do dono da chave que pode ser **HOST**, **ZONE**, **ENTITY**, **USER**.
* **-r** 🡪 Especifica uma origem randômica. No nosso caso utilizamos ***/dev/urandom***.

Em seguida especificamos o nome do domínio que estamos gerando as chaves.

Foram gerados dois arquivos com o nome ***Khow2security.com.br.+010+03426.key*** e ***Khow2security.com.br.+010+03426.private***.

**7.3.2 – Assinar o Domínio (Zona) com o DNSSEC**

Vamos assinar o arquivo de zona, mas antes disso, devemos incrementar o serial do registro **SOA**, para o **Servidor Slave** atualizar as alterações em seu banco de dados. E também temos que fazer o include do arquivo de chave pública no arquivo de zona.

dns-fusion:/var/named/namedb/master# echo "\$include Khow2security.com.br.+010+03426.key" >> how2security.com.br

Em fim vamos assinar a zona ***how2security.com.br*** com o utilitário **dnssec-signzone**.

dns-fusion:/var/named/namedb/master# dnssec-signzone -P -S -z -e 20160102120000 -o how2security.com.br /var/named/namedb/master/how2security.com.br

/var/named/namedb/master/how2security.com.br.signed

dns-fusion:/var/named/namedb/master# ls

1 dsset-how2security.com.br. how2security.com.br.signed named.root

127.0.0.x how2security.com Khow2security.com.br.+010+03426.key

192.168.5.x how2security.com.br Khow2security.com.br.+010+03426.private

dns-fusion:/var/named/namedb/master#

Onde:

* **-e** 🡪 Especifica um tempo de validade da assinatura, este **time** é informado por ano, mês, dia, hora, minutos e segundos (**YYYYMMDDhhmmss**). Por padrão será valido por 30 dias. No nosso caso irá expirar em 02/01/2014 as 12:00:00.
* **-o** 🡪 A zona origem, no nosso caso o nome da zona que estamos assinado.
* **-P** 🡪Desabilita a verificação por teste.
* **-S** 🡪 Instrui o **dnssec-signzone** a procurar no repositório uma chaves apropriada para a zona.
* **-z** 🡪 Ignora a flag **KSK** quando determinada na assinatura.

Por fim, informamos o nome do arquivo de banco de dados da zona. Que no nosso caso é o mesmo nome da zona.

Após assinar a zona será criado um arquivo com o mesmo nome do arquivo de banco de dados da zona com a extensão **.signed**, este arquivo será nosso novo banco de dados de zona.

**Observação:** Todas as vezes que alteramos o arquivo de bando de dados da zona (o arquivo sem assinatura), devemos executar este procedimento novamente, gerando outro arquivo de zona assinado. E não esquecer de incrementar o serial do registro **SOA** para a atualização das bases entre os **Servidores Master** e **Slaves**.

Para finalizar a configuração do **DNSSEC** devemos substituir o arquivo de zona pelo arquivo assinado em ***/etc/named/named.conf*** assim:

dns-fusion:/var/named/namedb/master# vi /etc/named/named.conf

--==[ Resumido ]==--

zone "how2security.com.br" {

type master;

**file "how2security.com.br.signed";**

allow-transfer {

key "how2sec";

};

};

zone "how2security.com" {

"/etc/named/named.conf" 83L, 1316C written

dns-fusion:/var/named/namedb/master#

Agora vamos abrir outro terminal e executar o utilitário **tail** para analisarmos o log na reinicialização do serviço do BIND, para isso:

dns-fusion:/# tail -f /var/log/named/named.log

Agora vamos reiniciar o serviço do BIND e analisar o resultado no outro terminal.

dns-fusion:/var/named/namedb/master# /etc/init.d/dns restart

[-]Reiniciando o Servico de Nome de Dominio...

dns-fusion:/var/named/namedb/master#

Log:

dns-fusion:/# tail -f /var/log/named/named.log

12-Sep-2014 15:08:15.761 general: received control channel command 'stop'

12-Sep-2014 15:08:15.762 general: shutting down: flushing changes

12-Sep-2014 15:08:15.763 general: stopping command channel on 127.0.0.1#953

12-Sep-2014 15:08:15.817 general: exiting

12-Sep-2014 15:08:17.821 general: managed-keys-zone: loaded serial 0

12-Sep-2014 15:08:17.832 general: zone 0.0.127.in-addr.arpa/IN: loaded serial 2014082000

12-Sep-2014 15:08:17.833 general: zone 5.168.192.in-addr.arpa/IN: loaded serial 2014082000

**12-Sep-2014 15:08:17.835 general: zone how2security.com.br/IN: loaded serial 2014082000 (DNSSEC signed)**

12-Sep-2014 15:08:17.836 general: zone how2security.com/IN: 'how2security.com' found SPF/TXT record but no SPF/SPF record found, add matching type SPF record

12-Sep-2014 15:08:17.836 general: zone how2security.com/IN: loaded serial 2014082000

12-Sep-2014 15:08:17.837 general: all zones loaded

12-Sep-2014 15:08:17.838 general: running

**7.3.3 – Forçar a Atualização do Slave**

Agora vamos forçar a atualização do **Servidor Slave** para vermos o resultado.

Antes de tudo abra um terminal e execute o utilitário **tail** para analisarmos

dns-corola:/# tail -f /var/log/named/named.log

Agora vamos forçar a atualização.

dns-corola:/# rm /var/cache/bind/how2security.com.br

dns-corola:/# /etc/init.d/bind9 restart

Stopping domain name service...: bind9

waiting for pid 3324 to die

.

Starting domain name service...: bind9.

dns-corola:/#

No outro terminal vamos analisar se houve a atualização

dns-corola:/# tail -f /var/log/named/named.log

12-Sep-2014 15:10:53.385 general: received control channel command 'stop -p'

12-Sep-2014 15:10:53.387 general: shutting down: flushing changes

12-Sep-2014 15:10:53.387 general: stopping command channel on 127.0.0.1#953

12-Sep-2014 15:10:53.388 general: stopping command channel on ::1#953

12-Sep-2014 15:10:53.410 general: exiting

12-Sep-2014 15:10:54.543 general: zone 0.in-addr.arpa/IN: loaded serial 1

12-Sep-2014 15:10:54.543 general: zone 127.in-addr.arpa/IN: loaded serial 1

12-Sep-2014 15:10:54.544 general: zone 0.0.127.in-addr.arpa/IN: loaded serial 2014082000

12-Sep-2014 15:10:54.544 general: zone 5.168.192.in-addr.arpa/IN: loaded serial 2014082000

12-Sep-2014 15:10:54.545 general: zone 255.in-addr.arpa/IN: loaded serial 1

12-Sep-2014 15:10:54.546 general: zone how2security.com/IN: loaded serial 2014082000

12-Sep-2014 15:10:54.546 general: zone localhost/IN: loaded serial 2

12-Sep-2014 15:10:54.547 general: managed-keys-zone ./IN: loaded serial 10

12-Sep-2014 15:10:54.551 general: running

12-Sep-2014 15:10:54.557 general: zone how2security.com.br/IN: Transfer started.

12-Sep-2014 15:10:54.562 xfer-in: transfer of 'how2security.com.br/IN' from 192.168.5.128#53: connected using 192.168.5.129#46946

**12-Sep-2014 15:10:54.570 general: zone how2security.com.br/IN: transferred serial 2014082000: TSIG 'how2sec'**

**12-Sep-2014 15:10:54.570 xfer-in: transfer of 'how2security.com.br/IN' from 192.168.5.128#53: Transfer completed: 1 messages, 39 records, 6604 bytes, 0.008 secs (825500 bytes/sec)**

Atualizada com sucesso, agora vamos ver como ficou nosso arquivo de banco de dados da zona ***how2security.com.br***.

dns-corola:/# cat /var/cache/bind/how2security.com.br

$ORIGIN .

$TTL 43200 ; 12 hours

how2security.com.br IN SOA fusion.how2security.com.br. suporte.how2security.com.br. (

2014082000 ; serial

28800 ; refresh (8 hours)

7200 ; retry (2 hours)

604800 ; expire (1 week)

86400 ; minimum (1 day)

)

RRSIG SOA 10 3 43200 20160102120000 (

20140912140607 3426 how2security.com.br.

WGYX37XFyUiCaobOiHru3BTrNI88OFszCaX5n5oTkxOd

1DPgkvVSrgn6xOmQthzWHmxvj1WJWSVy8xlREMNQzXzR

pX1Z4j5Xqq4usvr6sqemWcmWtyW8pxT+RPrO4MD29K4D

Lz9d4I1sGcUcPoHYsOO707Uro3tbCg10ilDqqA9S3r7+

+IKq4zkRzqxkLP13G8rUkNhV83qf/YtKAwp3GW2asa10

vJFW9+CA+Oe9dYczHxE2RlNYw0CfwShOlSN2SCqb7oEC

/zApPet20z2qf+aQ0ciIyg6zjjeJjFFh+AdD4F8yvvc2

Mk8H7fMsdCy7yIhYl4xsbkUvD4jyYL5Wnw== )

--==[ Resumido ]==--

$TTL 43200 ; 12 hours

www CNAME fusion

RRSIG CNAME 10 4 43200 20160102120000 (

20140912140607 3426 how2security.com.br.

XyJw1VxOLwAzd8QPbHXphwtakxZxZvU6tagH7fYy7+fz

UO9UEf/z+3R6pMqexJ/IxIWy2T999s+qNDxeGnbJLrwd

XS54vDgUNSoivkil8Dgrd8hicRYK0gLRiHKSsrCMoETG

ffKHBjWFETeB4UXxMSuIW4IuwsgWAdwqBV0V2YAqfSqm

jkffSl1vQuthfLA6RlpmP7IGb6+l19CD5UyZKxoj6ojs

oGWXMRWwIjNY6kxch4CGI5WDJwcigSMYsTRwExEpNgJE

8XuIzgF7jwR+weoBeVsRl/KwJnioS1u1koG+yyKBX1jJ

Anexg6eCfQMZJrSn+ExoghCiIq6ZCW3Sfw== )

$TTL 86400 ; 1 day

NSEC how2security.com.br. CNAME RRSIG NSEC

RRSIG NSEC 10 4 86400 20160102120000 (

20140912140607 3426 how2security.com.br.

SpTpT6O6LDM22L19nBImL+5tYYey8thrcR/4MtcehrbX

CvKJQkHrS94NdX5ghK5KWHCgaJY2tJcsb/Y+QsI/zide

7eeY8KM+cjsevUpEX4U4au5z5agpEHuqno66ap8XD8KW

5OHkkOkOe1PE1kHUDfljbUXPYMC4m3xpuVmT4oMnOe0F

N4wUEosO6gKWvuDUhcCBxLZsnmZpb+FJjpZmbxDOmJNY

3adDjrNV5qJ1BRGXzEhGP98vw16T5MEsYKiV7dl/qSJ8

qHlt5h7ADU4tLSo8R/jHlxbQDU66PDE3jQHruA3Hr9XA

PDmya7cRNw3R5h/4sDE1A88SZ13Ch4NiDA== )

dns-corola:/#

Como podemos observar o arquivo está assinado.

**7.3.4 – Registrando o DS no Registro.br**

Necessitamos copiar os dados de **KeyTag** e **Digest** do arquivo ***dsset-how2security.com.br*** para a interface no site do Registro.br.

dns-fusion:/var/named/namedb/master# cat dsset-how2security.com.br. | head -1

how2security.com.br. IN DS **3426** 10 1 **D21CD8A15868D9D448E059646B371D6D12117906**

**\\_/ \\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/**

**| |**

**| +--> Digest**

**+---> KeyTag**

dns-fusion:/var/named/namedb/master#

Em posse dessas informações, devemos entrar na área administrativa do **Registro.br** e cadastrar seu **KeyTag** e **Digest**.



Figura 01 – Formulário de Registro dos Servidores DNS do Domínio

**Observação:** Todas as vezes que for alterada alguma informação no banco de dados do **Servidor Autoritativo** devemos incrementar o serial do registro **SOA** e reassinar a zona utilizando o utilitário **dnssec-segnzone**. Também não se esquecer de reassinar a zona antes das assinaturas expirarem.

**7.3.5 – Configurar DNSSEC em Servidores Recursivos**

Para utilizarmos nosso **Servidor DNS** com **BIND** como um **Servidor Recursivo** devemos utilizar a assinatura da raiz do sistema DNS como **Trust Anchor** em suas configurações (tanto **Servidores Master** como **Servidores Slaves** que sejam recursivos).

A chave encontra-se originalmente no site da **IANA ()**. Caso deseje fazer download e validação manualmente utilizando um script feito pelo **Registro.br** da seguinte forma:

dns-fusion:/# cd /etc/named/

dns-fusion:/etc/named# aptitude install xsltproc

dns-fusion:/etc/named# wget ftp://ftp.registro.br/pub/doc/fetch\_root\_anchor.sh

--2014-09-17 17:21:48-- ftp://ftp.registro.br/pub/doc/fetch\_root\_anchor.sh

=> `fetch\_root\_anchor.sh'

Resolving ftp.registro.br (ftp.registro.br)... 200.160.2.8, 2001:12ff:0:2::8

Connecting to ftp.registro.br (ftp.registro.br)|200.160.2.8|:21... connected.

Logging in as anonymous ... Logged in!

==> SYST ... done. ==> PWD ... done.

==> TYPE I ... done. ==> CWD (1) /pub/doc ... done.

==> SIZE fetch\_root\_anchor.sh ... 1902

==> PASV ... done. ==> RETR fetch\_root\_anchor.sh ... done.

Length: 1902 (1,9K) (unauthoritative)

100%[========================================================>] 1.902 --.-K/s in 0,03s

2014-09-17 17:21:49 (58,2 KB/s) - `fetch\_root\_anchor.sh' saved [1902]

dns-fusion:/etc/named#

Agora em posse do script vamos torna-lo executável, e em seguida executá-lo.

dns-fusion:/etc/named# chmod 700 fetch\_root\_anchor.sh

dns-fusion:/etc/named# ./fetch\_root\_anchor.sh

. initial-key 257 3 8 "AwEAAagAIKlVZrpC6Ia7gEzahOR+9W29euxhJhVVLOyQbSEW0O8gcCjFFVQUTf6v58fLjwBd0YI0EzrAcQqBGCzh/RStIoO8g0NfnfL2MTJRkxoXbfDaUeVPQuYEhg37NZWAJQ9VnMVDxP/VHL496M/QZxkjf5/Efucp2gaDX6RS6CXpoY68LsvPVjR0ZSwzz1apAzvN9dlzEheX7ICJBBtuA6G3LQpzW5hOA2hzCTMjJPJ8LbqF6dsV6DoBQzgul0sGIcGOYl7OyQdXfZ57relSQageu+ipAdTTJ25AsRTAoub8ONGcLmqrAmRLKBP1dfwhYB4N7knNnulqQxA+Uk1ihz0=";

dns-fusion:/etc/named#

**Observação:** Devido um erro de certificado no site do **IANA** o **wget** não confiava no site e consequentemente ele não baixava o arquivo **XSL** dando o seguinte erro:

dns-fusion:/etc/named# ./fetch\_root\_anchor.sh

**warning: failed to load external entity "root-anchors.xml"**

**unable to parse root-anchors.xml**

**diff: root-anchors.txt: No such file or directory**

Isso porque o **wget** não estava baixando o arquivo ***root-anchor.xml***, para resolver eu editei o script e em **wget** eu coloquei o seguinte parâmetro para não checar o certificado do **IANA**:

wget **--no-check-certificate** -q https://data.iana.org/root-anchors/root-anchors.xml

Por fim devemos editar o arquivo ***/etc/named/named.conf*** para incluir a chave dos **Root Servers** no **Trust Anchor** e reiniciar o serviço BIND.

dns-fusion:/etc/named# vi named.conf

options {

directory "/var/named/namedb/master";

listen-on { 127.0.0.1; 192.168.5.128; };

listen-on-v6 { none; };

empty-zones-enable no;

version "Not Available";

**dnssec-enable yes;**

**dnssec-validation yes;**

allow-transfer { none; };

allow-recursion { net\_how2security; };

allow-query { any; };

};

--==[ Resumido ]==--

**managed-keys {**

**/\* Root Keys \*/**

**. initial-key 257 3 8 "AwEAAagAIKlVZrpC6Ia7gEzahOR+9W29euxhJhVVLOyQbSEW0O8gcCjFFVQUTf6v58fLjwBd0YI0EzrAcQqBGCzh/RStIoO8g0NfnfL2MTJRkxoXbfDaUeVPQuYEhg37NZWAJQ9VnMVDxP/VHL496M/QZxkjf5/Efucp2gaDX6RS6CXpoY68LsvPVjR0ZSwzz1apAzvN9dlzEheX7ICJBBtuA6G3LQpzW5hOA2hzCTMjJPJ8LbqF6dsV6DoBQzgul0sGIcGOYl7OyQdXfZ57relSQageu+ipAdTTJ25AsRTAoub8ONGcLmqrAmRLKBP1dfwhYB4N7knNnulqQxA+Uk1ihz0=";**

**};**

"named.conf" 91L, 1772C written

dns-fusion:/etc/named#

dns-fusion:/etc/named# named-checkconf

dns-fusion:/etc/named# /etc/init.d/dns restart

**[-]Reiniciando o Servico de Nome de Dominio...**

dns-fusion:/etc/named#

**8 – BIND DNS em Linux – Referencias**

**Referencias Bibliograficas**

**[1]** Freitas, Andrey Rodrigues de – Perícia forense aplicada à informática: em ambientes Microsoft, 1º Ed, Rio de Janeiro, 2006, Brasport.

**[2]** Forouzan, Behrouz A. – Protocolo TCP/IP, 3º Ed, São Paulo, 2008, McGraw-Hill.

**[3]** Freitas, Andrey Rodrigues de – Perícia forense aplicada à informática: em ambientes Microsoft, 1º Ed, Rio de Janeiro, 2006, Brasport.

**[4]** Registro BR. Disponível em: [< http://registro.br/dominio/categoria.html](%3c%20http:/registro.br/dominio/categoria.html%20) >. Acessado em: 15/04/2014.

**[5]** Gregoriano. Disponível em: <<http://www.gregoriano.org.br/portinha/005.htm>>. Acessado em: 15/04/2014.

**[6]** Wikipedia. Disponível em: <<http://pt.wikipedia.org/wiki/Anexo:Lista_de_TLDs>>. Acessado em: 16/04/2014.

**[7]** Hugo Azevedo. Disponível em: <<http://www.hugoazevedo.eti.br/html/nscd.html>>. Acessado em: 05/08/2014.

**[8]** Alex. Disponível em: <<http://alex.laner.net.br/?p=10> >. Acessado em: 12/08/2014.

**[9]** RNP. Disponível em: <<http://www.pop-ba.rnp.br/Site/ConfDNSSecundario> >. Acessado em: 05/08/2014.

**[10]** ISC (Internet System Conbsortium). Disponível em: <<http://lamejournal.com/2013/06/10/bind-enabling-tsig-for-zone-transfers/>>. Acessado em: 09/09/2014.

**[11]** ISC (Internet System Conbsortium). Disponível em: <<http://www.isc.org/downloads/bind/doc/bind-9-10/> >. Acessado em: 11/09/2014.

**[12]** Registro.BR. Disponível em: <[ftp://ftp.registro.br/pub/doc/tutorial-dnssec.pdf](ftp://ftp.registro.br/pub/doc/tutorial-dnssec.pdf%20) >. Acessado em: 16/09/2014.

**[13]** Registro.BR. Disponível em: <[https://registro.br/tecnologia/root-anchor.html](https://registro.br/tecnologia/root-anchor.html%20) >. Acessado em: 16/09/2014.

**[14]** Registro.BR. Disponível em: <[ftp://ftp.registro.br/pub/doc/dns-fw.pdf](ftp://ftp.registro.br/pub/doc/dns-fw.pdf%20) >. Acessado em: 16/09/2014.

**[15]** Registro.BR. Disponível em: <[https://registro.br/tecnologia/root-anchor.html](https://registro.br/tecnologia/root-anchor.html%20) >. Acessado em: 17/09/2014.