

# DNS

## Configuración de un DNS para un dominio real.

De wikipedia podemos extraer

“(DNS) es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet. Aunque como base de datos el DNS es capaz de asociar distintos tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a direcciones IP y la localización de los servidores de correo electrónico de cada dominio.

La asignación de nombres a direcciones IP es ciertamente la función más conocida de los protocolos DNS. Por ejemplo, si la dirección IP del sitio FTP de prox.ve es 200.64.128.4, la mayoría de la gente llega a este equipo especificando ftp.prox.ve y no la dirección IP. Además de ser más fácil de recordar, el nombre es más fiable. La dirección numérica podría cambiar por muchas razones, sin que tenga que cambiar el nombre.”

Lo cual nos da un acercamiento al protocolo y las funciones del DNS. Nosotros basaremos todo el documento en el paquete BIND de ISC <http://www.isc.org> el cual es el más usado y distribuido en Internet para sistemas linux/bsd.

Primeramente necesitaremos saber los archivos que componen el DNS, en la actualidad el BIND9 se instala por default bajo un árbol en /var/named, allí encontraremos los archivos de zonas que veremos mas adelante como también los archivos de los ROOT Server. Por ejemplo aquí están los archivos base que instala BIND

```
chroot localdomain.zone named.broadcast named.ip6.local named.zero
data localhost.zone named.ca named.local slaves
```

A esto también deberemos agregarle el archivo /etc/named.conf que es el archivo de configuración de de todo el paquete, es de donde se le indicaran las directivas a seguir para cada zona en especial.

En nuestro servidor (host) podremos tener mas de un dominio pero principalmente en este documento nos basaremos en 1 solo dominio real, ya que con mas dominios los procedimientos son casi iguales o intuitivos, para eso nos basaremos en el dominio pablo.org.uy como ejemplo.

Pero antes de empezar veamos como hace BIND para buscar un nombre de dominio es decir para darnos por ejemplo el numero IP de un nombre de dominio. Imaginemos que la base de datos de nombres de dominio es un Árbol en el cual se encuentra una Raíz que son los RootServers y luego vienen las ramas y las hojas, bien las ramas podemos ejemplificarlas con los .edu .org .net llamados TLDs (Top Level Domains) y las hojas podemos decir que son los nombres, en efecto para los ejemplos usaremos [google.com](http://google.com) y lo que haremos será usar el comando dig

Si digitamos dig +norec [google.com](http://google.com) tendremos que nos responden servidores autoritativos pero ningún servidor de respuestas. Cuando le agregamos al dig el +norec es para que no sea recursivo y axial nosotros podremos buscar con nuestros propios medios quien es el que responde. Esto no se hace muy seguido pero es muy bueno para entender que es lo que hace el DNS

```
[pablo@sony ~]# dig +norec google.com
```

```
; <<>> DiG 9.3.1 <<>> +norec google.com
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42105
;; flags: qr ra; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 4

;; QUESTION SECTION:
;google.com.                IN      A

;; AUTHORITY SECTION:
google.com.                238397 IN     NS     ns1.google.com.
google.com.                238397 IN     NS     ns2.google.com.
google.com.                238397 IN     NS     ns3.google.com.
google.com.                238397 IN     NS     ns4.google.com.

;; ADDITIONAL SECTION:
ns1.google.com.           61153  IN     A      216.239.32.10
ns2.google.com.           61153  IN     A      216.239.34.10
ns3.google.com.           236884 IN     A      216.239.36.10
ns4.google.com.           236884 IN     A      216.239.38.10

;; Query time: 5 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Thu Mar 9 13:35:35 2006
;; MSG SIZE rcvd: 164
```

Si vemos arriba en el ejemplo tenemos servidores en el Authority Section bien tomaremos uno de ellos para preguntarle si conoce a [google.com](http://google.com) y si realmente nos puede responder donde se encuentra. Es decir iremos por esa rama para ver si una de las Hojas es [google.com](http://google.com) así que ahora tendremos que hacer por ejemplo

```
[pablo@sony ~]# dig +norec @ns1.google.com google.com
; <<>> DiG 9.3.1 <<>> @ns1.google.com google.com
; (1 server found)
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 13782
;; flags: qr aa rd; QUERY: 1, ANSWER: 3, AUTHORITY: 4, ADDITIONAL: 4

;; QUESTION SECTION:
;google.com.                IN      A

;; ANSWER SECTION:
google.com.                300    IN     A      64.233.187.99
google.com.                300    IN     A      64.233.167.99
google.com.                300    IN     A      72.14.207.99

;; AUTHORITY SECTION:
google.com.                345600 IN     NS     ns1.google.com.
google.com.                345600 IN     NS     ns2.google.com.
google.com.                345600 IN     NS     ns3.google.com.
```

```
google.com.      345600 IN   NS   ns4.google.com.
```

```
:: ADDITIONAL SECTION:
```

```
ns1.google.com.  345600 IN   A    216.239.32.10  
ns2.google.com.  345600 IN   A    216.239.34.10  
ns3.google.com.  345600 IN   A    216.239.36.10  
ns4.google.com.  345600 IN   A    216.239.38.10
```

```
:: Query time: 257 msec  
;; SERVER: 216.239.32.10#53(216.239.32.10)  
;; WHEN: Thu Mar 9 13:41:36 2006  
;; MSG SIZE rcvd: 212
```

Y ahora si nos esta diciendo que hay un ;; ANSWER SECTION: en cual responde a google.com que es lo que nosotros queríamos. Simplificando y traduciéndolo al árbol. Lo que hicimos fue preguntar en nuestra rama, conoce a google.com? Respondieron que saben que es pero que pregunta a otra rama, preguntamos a una de las ramas que nos sugieren y efectivamente una de las hojas era google.com

Nos podemos encontrar con muchas ramas a las que tendremos que preguntar si es que no conocen al interesado pero se puede hacer sin digitar el +noredc antes del dig y nos dará el resultado que necesitamos.

Bien ahora configuremos nuestro propio dominio teniendo como base que sabemos el funcionamiento básico del DNS, nuestro dominio será pablo.org.uy haciendo referencia a la pagina donde hicimos download al documento.

Como ya vimos anteriormente es necesario modificar algunos archivos y el primero será el archivo /etc/named.conf este archivo que se encargara de aquí en mas de decirle al DNS que es lo que debe de cargar y que no, tendremos que decirle que existe un dominio llamado pablo.org.uy y que deberá cargarlo, para lo cual se modifica agregando

```
zone "pablo.org.uy" {  
    type master;  
    file "pablo.zone";  
};
```

Le diremos a named que la zona será pablo.org.uy (siempre igual al nombre de dominio) que es Master es decir que será principal para nuestra jerarquía y que el archivo que contiene los datos se llamara pablo.zone (siempre tenemos que tener cuidado en los ; . y } )

Ahora es momento de crear el archivo pablo.zone en el cual pondremos datos falsos pero con total coherencia. Por ejemplo nuestro archivo debería quedar así. Si la IP de nuestro server fuera la 200.4.22.6

```
$ORIGIN .  
$TTL 76800 ; 21 hours 20 minutes  
pablo.org.uy IN SOA ns.pablo.org.uy. pablo.pablo.org.uy. (  
    200601251 ; serial  
    86400 ; refresh (1 day)  
    3600 ; retry (1 hour)  
    1209600 ; expire (2 weeks)  
    240 ; minimum (4 minutes)  
)  
    NS ns.pablo.org.uy  
    A 200.4.22.6  
    MX 5 mail.pablo.org.uy.  
$ORIGIN pablo.org.uy.  
ns A 200.4.22.6  
www A 200.4.22.6  
mail A 200.4.22.6
```

La maquina ns.pablo.org.uy debe de ser una maquina con el registro A.

En pablo.pablo.org.uy es el mail donde Irán las consultas de los administradores de DNS necesiten hacer si algo funciona mal o por ejemplo si quieren ser nuestro secundario.

El NS indica que maquina será el servidor de nombres en este caso será nuestra misma maquina y por ultimo el MX es el registro que indica en este caso que la misma maquina será el Mail Exchanger para ese dominio.

Bien nuestro server ya esta pronto para correr el DNS pero para hacerlo completo necesitamos también tener un reverso de nuestras direcciones IPS es decir cuando en ves de preguntar por un nombre pregunten por una dirección que también les demos una respuesta. Esto ya casi no se realiza debido a que los ISPs lo hacen por nosotros pero si queremos ser detallistas y tener bajo control todo nuestro DNS yo recomiendo que le soliciten al ISP que les deleguen el registro inverso de las direcciones IP. Por ejemplo veamos un ejemplo muy sencillo de una red /28 que es casi lo mas común que se nos asigne con IPS publicas

Si se están preguntando que es un /28 aquí pueden ver a cuantas cantidad de IPS equivale

<http://www.pablo.org.uy/cidr/>

La nomenclatura es casi igual tomaremos en cuenta que nuestra red es la 200.4.22.64/28 para esto tendremos que modificar nuevamente

```
$TTL 86400
```

```
64-28.22.4.200.in-addr.arpa. IN SOA pablo.org.uy. pablo.pablo.org.uy. (
    2005191203
    1H
    1H
    1W
    1H )
```

```
64-28.22.4.200.in-addr.arpa. IN NS pablo.org.uy.
```

```
66 IN PTR pablo.org.uy.
```

```
67 IN PTR client1.pablo.org.uy.
```

```
68 IN PTR wifi.pablo.org.uy.
```

```
69 IN PTR engops.pablo.org.uy.
```

Bien esto nos da que para la red 64-28.22.4.200.in-addr.arpa que es 200.4.22.64/28 tenemos los hosts 66, 67, 68, y 69 definidos como clientes. Fijarse bien que al final de cada nombre de host termina con un punto (.)

Ahora lo único que nos falta es bajar el demonio de named y volverlo a levantar para que los cambios surjan efectos si es fedora service named restart si es algún otro sistema por ejemplo bsd

ps -ef |grep named y matar el numero de proceso con kill -9 nroproceso.

Los mensajes de error si es que los hubiera se desplegaran en el archivo de Log /var/log/message

si todo esta ok veremos

```
Mar 9 14:42:49 sony named[9127]: starting BIND 9.3.1 -u named -t /var/named/chroot
Mar 9 14:42:49 sony named[9127]: found 1 CPU, using 1 worker thread
Mar 9 14:42:49 sony named[9127]: loading configuration from '/etc/named.conf'
Mar 9 14:42:49 sony named[9127]: listening on IPv4 interface lo, 127.0.0.1#53
Mar 9 14:42:49 sony named[9127]: listening on IPv4 interface vmnet1, 192.168.148.1#53
Mar 9 14:42:49 sony named[9127]: listening on IPv4 interface vmnet8, 172.16.139.1#53
Mar 9 14:42:49 sony named[9127]: listening on IPv4 interface wlan0, 192.168.1.118#53
Mar 9 14:42:49 sony named[9127]: command channel listening on 127.0.0.1#953
Mar 9 14:42:49 sony named[9127]: zone 0.in-addr.arpa/IN: loaded serial 42
Mar 9 14:42:49 sony named[9127]: zone 0.0.127.in-addr.arpa/IN: loaded serial 1997022700
Mar 9 14:42:49 sony named[9127]: zone 255.in-addr.arpa/IN: loaded serial 42
Mar 9 14:42:49 sony named[9127]: zone localdomain/IN: loaded serial 42
Mar 9 14:42:49 sony named[9127]: zone localhost/IN: loaded serial 42
Mar 9 14:42:49 sony named[9127]: running
```