ANEXO 2: DNS

Índice general

Capítulo 1

DNS

Contents

1.1. Política de nombres para la vista "vpn"

A continuación se mencionan todos los RR (Resource Registry, registros DNS) que se utilizarán en el despliegue para todos los servicios, presente y futuros.

Tipo (RR)	Nombre (FQDN)	Datos específicos del tipo
SOA	casafx.dyndns.org.	...
NS	casafx.dyndns.org.	ns1.casafx.dyndns.org.
NS	casafx.dyndns.org.	ns2.casafx.dyndns.org.
HINFO	casafx.dyndns.org.	"PC" "Debian 7"
TXT	casafx.dyndns.org.	"FAC"
A	casafx.dyndns.org.	
A	ns1.casafx.dyndns.org.	10.168.1.1
A	ns2.casafx.dyndns.org.	10.168.1.2
A	dklab1.casafx.dyndns.org.	10.168.1.1
A	dklab2.casafx.dyndns.org.	10.168.1.2
A	ntp.casafx.dyndns.org.	10.168.1.1
A	ntp.casafx.dyndns.org.	10.168.1.2
SRV	_ntp._udp.casafx.dyndns.org.	0 0 123 dklab1.casafx.dyndns.org.
SRV	_ntp._udp.casafx.dyndns.org.	0 0 123 dklab2.casafx.dyndns.org.
SRV	_ldap._tcp.casafx.dyndns.org	0 0 389 dklab1.casafx.dyndns.org.
SRV	_ldap._tcp.casafx.dyndns.org	0 0 389 dklab2.casafx.dyndns.org.
TXT	_kerberos.casafx.dyndns.org.	"CASAFX.DYNDNS.ORG"
SRV	_kerberos-adm._tcp.casafx.dyndns.org.	0 0 749 krb1.casafx.dyndns.org.
SRV	_kerberos-adm._tcp.casafx.dyndns.org.	0 0 749 krb1.casafx.dyndns.org.
SRV	_kpasswd._udp.casafx.dyndns.org.	0 0 464 dklab1.casafx.dyndns.org.
SRV	_kpasswd._udp.casafx.dyndns.org.	0 0 464 dklab2.casafx.dyndns.org.
SRV	_kerberos-master._udp.casafx.dyndns.org.	0 0 88 dklab1.casafx.dyndns.org.
SRV	_kerberos-master._udp.casafx.dyndns.org.	0 0 88 dklab2.casafx.dyndns.org.
SRV	_kerberos._ucp.casafx.dyndns.org.	0 0 88 dklab1.casafx.dyndns.org.
SRV	_kerberos._ucp.casafx.dyndns.org.	0 0 88 dklab1.casafx.dyndns.org.
SRV	_afs3-vlserver._udp.casafx.dyndns.org.	0 0 7003 dklab1.casafx.dyndns.org.
SRV	_afs3-vlserver._udp.casafx.dyndns.org.	0 0 7003 dklab2.casafx.dyndns.org.
SRV	_afs3-prserver._udp.casafx.dyndns.org.	0 0 7002 dklab1.casafx.dyndns.org.
SRV	_afs3-prserver._udp.casafx.dyndns.org.	0 0 7002 dklab2.casafx.dyndns.org.
AFSDB	casafx.dyndns.org.	1 dklab1.casafx.dyndns.org.
AFSDB	casafx.dyndns.org.	1 dklab2.casafx.dyndns.org.
SRV	_xmpp-client._tcp.casafx.dyndns.org.	0 0 5222 dklab1.casafx.dyndns.org.
SRV	_xmpp-client._tcp.casafx.dyndns.org.	0 0 5222 dklab2.casafx.dyndns.org.
SRV	_xmpp-server._tcp.casafx.dyndns.org.	0 0 5269 dklab1.casafx.dyndns.org.
SRV	_xmpp-server._tcp.casafx.dyndns.org.	0 0 5269 dklab2.casafx.dyndns.org.

...	...	...
MX	casafx.dyndns.org.	10 dklab1.casafx.dyndns.org.
MX	casafx.dyndns.org.	10 dklab2.casafx.dyndns.org.
SRV	_imap._tcp.casafx.dyndns.org.	0 0 143 dklab1.casafx.dyndns.org.
SRV	_imap._tcp.casafx.dyndns.org.	0 0 143 dklab2.casafx.dyndns.org.
SRV	_submission._tcp.casafx.dyndns.org.	0 0 587 dklab1.casafx.dyndns.org.
SRV	_submission._tcp.casafx.dyndns.org.	0 0 587 dklab2.casafx.dyndns.org.
SRV	_sieve._tcp.casafx.dyndns.org.	0 0 4190 dklab1.casafx.dyndns.org.
SRV	_sieve._tcp.casafx.dyndns.org.	0 0 4190 dklab2.casafx.dyndns.org.
SPF	casafx.dyndns.org.	"v=spf1 mx/24 ?all"
TXT	casafx.dyndns.org.	"v=spf1 +mx/24 ?all"
TXT	exim4dkim._domainkey.casafx.dyndns.org.	"v=DKIM1; s=... ... p=..."
TXT	_domainkey.casafx.dyndns.org.	"t=y; o=~;"
TXT	_adsp._domainkey.casafx.dyndns.org.	"dkim=unknown"
CNAME	openvpn1.casafx.dyndns.org.	dklab1.casafx.dyndns.org.
CNAME	ntp1.casafx.dyndns.org.	dklab1.casafx.dyndns.org.
CNAME	ntp2.casafx.dyndns.org.	dklab2.casafx.dyndns.org.
CNAME	ldap1.casafx.dyndns.org.	dklab1.casafx.dyndns.org.
CNAME	ldap2.casafx.dyndns.org.	dklab2.casafx.dyndns.org.
CNAME	krb1.casafx.dyndns.org.	dklab1.casafx.dyndns.org.
CNAME	krb2.casafx.dyndns.org.	dklab2.casafx.dyndns.org.
CNAME	afs1.casafx.dyndns.org.	dklab1.casafx.dyndns.org.
CNAME	afs2.casafx.dyndns.org.	dklab2.casafx.dyndns.org.
CNAME	managesieve1.casafx.dyndns.org.	dklab1.casafx.dyndns.org.
CNAME	managesieve2.casafx.dyndns.org.	dklab2.casafx.dyndns.org.
CNAME	mx1.casafx.dyndns.org.	dklab1.casafx.dyndns.org.
CNAME	mx2.casafx.dyndns.org.	dklab2.casafx.dyndns.org.
CNAME	imap1.casafx.dyndns.org.	dklab1.casafx.dyndns.org.
CNAME	imap2.casafx.dyndns.org.	dklab2.casafx.dyndns.org.
NS	1.168.10.IN-ADDR.ARPA.	ns1.casafx.dyndns.org.
NS	1.168.10.IN-ADDR.ARPA.	ns2.casafx.dyndns.org.
PTR	1.1.168.10.IN-ADDR.ARPA.	dklab1.casafx.dyndns.org.
PTR	2.1.168.10.IN-ADDR.ARPA.	dklab2.casafx.dyndns.org.

1.2. DKLAB1

1.2.1. Instalación de Bind9 y dnsutils

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

La documentación oficial está en:

- http://www.bind9.net/manual/bind/9.3.2/Bv9ARM.ch06.html

... pero es tanto o más útil la que podemos encontrar en:

- http://www.zytrax.com/books/dns/

1.2.2. DKLAB1 como cliente DNS; instalación de resolvconf; nsswitch.conf

El fichero /etc/resolv.conf almacena los servidores de nombres que el sistema operativo utiliza para resoluciones DNS. El paquete resolvconf añade algunas funcionalidades que hacen más flexible su administración (sobre todo en máquinas que se mueven en distintas redes con diferentes resolvedores y necesitan una reconfiguración automática). Aunque no estrictamente necesario en máquinas del perfil de dklab1 y dklab2, es buena idea acostumbrarse a trabajar con él.

```
apt-get install resolvconf
```

```
-"Prepare /etc/resolv.conf for dynamic updates?":

Yes
```

...hace /etc/resolv.conf un enlace simbólico a /etc/resolvconf/run/resolv.conf; el resolv.conf precedente se copia a /etc/resolvconf/resolv.conf.d/original y será restaurado si se elimina el paquete resolvconf.

```
-"Se recomienda reiniciar".
```

...en nuestra situación no debiera ser necesario.

Por otro lado, en versiones anteriores se preguntaba si /etc/resolvconf/resolv.conf.d/tail debía ser un enlace simbólico al precedente /etc/resolvconf/resolv.conf.d/original, de forma que al construir el definitivo /etc/resolv.conf el original forme parte del definitivo dinámicamente generado. Ésto ya no se pregunta, por defecto tail es un fichero vacío y no dicho enlace.

```
view /usr/share/doc/resolvconf/README.gz
```

```
vim /etc/network/interfaces
```

```
...
iface ovpnCASAFX-SRV inet manual
 ...
####fx:
 dns-nameservers 10.168.1.1 10.168.1.2
 dns-search casafx.dyndns.org.
 #dns-options rotate
####endfx
####fx:
# ...
# dns-nameservers 80.58.0.33 8.8.8.8
# La li'nea superior a e'sta podri'a aparecer en el bloque "iface"
# de la interfaz que nos diese salida a internet excepto si, como es
# nuestro caso, esa interfaz recibe la IP dina'micamente (en este caso,
# de hecho, resolvconf ira' a buscar el servidor DNS recibido por DHCP
# al fichero /var/lib/dhcp/dhclient.<iface>.leases).
# De una forma u otra, resolvconf consigue indicarle a named que' servidores
# debe utilizar como forwarders, y lo consigue de una forma din'amica
# en lugar de la esta'ticamente definida en /etc/bind/named.conf.options.
####endfx
```

[1]

El software resolvconf funciona así: cuando las interfaces se activan, se escribe en ficheros /run/resolvconf/interface/<ifacename>.<suffix>la información DNS recabada inspeccionando el contenido de las variables dns-nameserver y dns-search, o los DNS recibidos por DHCP en su caso. Entonces resolvconf concatena estos ficheros <ifacename>.<suffix>según el orden indicado en /etc/resolvconf/interface-order para formar el /etc/resolvconf/run/resolv.conf, que en virtud de ser /etc/resolv.conf un enlace simbólico a él, contiene los servidores de nombres que se van a usar definitivamente.

Si bien el mecanismo anterior es fácil de entender y es el que debemos tener en cuenta

[1]En cualquier equipo que no despliegue un servidor de nombres (cualquier otro equipo hipotético que no fuese ni dklab1 ni dklab2) se incluiría la opción "dns-options rotate", aqui'desactivada, ya que le permitiría distribuir la carga entre ambos servidores de nombres.

en cualquir máquina en que opere resolvconf, en el caso de dklab1 y dklab2 en que se instala resolvconf concurrentemente a Bind9 ese comportamiento se modifica para permitir que interactúen flexiblemente. Por tanto para dklab1 y dklab2 nos atañe también lo siguiente:

El script de inicio de Bind9, /etc/init.d/bind9, puede detectar la existencia de resolvconf y ello tiene consecuencias: si Bind9 tiene en /etc/default/bind9 la variable "RESOLVCONF=yes", entonces ordenará a resolvconf la creación de un fichero /etc/resolvconf/run/interface/lo en que se declara a sí mismo como servidor de nombres (es decir, lo.named declara la IP 127.0.0.1 como dirección IP del servidor de nombres).

Recíprocamente, que resolvconf detecte la existencia de Bind9 tiene también sus propias consecuencias[2]:

1. Por un lado, si la variable TRUNCATE_NAMESERVER_LIST_AFTER_LOOPBACK_ADI está inicializada a "yes" (y no a otro valor como "no") en /etc/default/resolvconf, o no está definida esa variable o no existe ese fichero (situación actual), el script /etc/resolvconf/update.d/libc de resolvconf detecta el contenido del fichero lo.named que originó el script init de Bind9 y rehúsa a su comportaimiento habitual, es decir no utiliza el resto de servidores DNS recabados para crear las entradas del /etc/resolv.conf; en su lugar deja en éste como única entrada a "nameserver 127.0.0.1".

2. Y por otro lado lleva a cabo la reescritura de forwarders para Bind9, es decir, los servidores DNS para las zonas en las que nuestro Bind9 no va a ser autoridad, resolvconf se los actualiza con los DNS que ha recabado (excepto el 127.0.0.1 expresado en lo.named, claro). Esta acción necesita para llevarse a cabo su propia condición: debe existir un script específico en /etc/resolvconf/update.d/, ahora no lo hay pero se provee uno en /usr/share/doc/resolvconf/resolvconf-update-bind. Dicho script tras ser copiado, como haremos, con permisos de ejecución a /etc/resolvconf/update.d/, será ejecutado por resolvconf. Entonces si el script detecta la existencia de /usr/sbin/named así como al fichero /etc/bind/named.conf.options, copia este último con la sección forwarders reescrita a /run/bind/named.options (fichero que el administrador de Bind9 ha de incluir en la configuración en lugar del /etc/bind/named.conf.options original) y ordena a named un "reload" de su configuración para hacer efectiva la actualización de forwarders.

El punto 1 puede ser un problema en nuestra situación en el sentido de que nuestra infraestructura de servicios, en el caso de nuestro testbed, está pensada para funcionar sobre la red privada virtual, así que queremos que las resoluciones usen las IP de la VPN, la vista DNS interna (conceptos que se introducirán luego). Podemos asegurar dicho comportamiento optando por una de estas soluciones: configurar las vistas de Bind9 (haciendo que 127.0.0.1 pertenezca a la vista interna) o configurar resolvconf para que el servidor de nombres dictado en /etc/resolv.conf apunte a 10.168.1.1 o .2 antes que a 127.0.0.1. Nos decidimos por modificar resolvconf al ser un problema del comportamiento cliente de DNS.

[2]Todos los detalles en /usr/share/doc/resolvconf/README.gz, así como en "man resolvconf" sección "EVIRONMENTAL VARIABLES".

Efectivamente según "man resolv.conf": "If there are multiple servers, the resolver library queries them in the order listed". Por tanto si hacemos que a la entrada 127.0.0.1 se antepongan las de 10.168.1.1,2 sólo se utilizará la vista "external" cuando los servidores de nombres de la infraestructura sobre la VPN no estén activos y, por tanto, esté desactivada en la práctica la infraestructura en sí y, por tanto, no sea contradictorio hacer resoluciones de la vista DNS externa.

Por tanto, reconfiguramos el orden de construcción del /etc/resolv.conf:

```
vim /etc/resolvconf/interface-order
```

```
####fx
#E'sta debiera ser la primera li'nea-no-comentario del fichero.
ovpnCASAFX-SRV*
####endfx
...
```

Por último, reiniciamos las interfaces y el software resolvconf, tal que se recreen todos los ficheros que se mencionaron. Previamente disponemos el script de reescritura de forwarders en su lugar.

```
cp /usr/share/doc/resolvconf/resolvconf-update-bind /etc/resolvconf/update.d/bind
chmod a+x /etc/resolvconf/update.d/bind
```

Y ya:

```
IFACE=‘ip route | grep default | awk ’{print $5}’‘
ifdown -v $IFACE && sleep 4 && ifup -v $IFACE
ifdown -v ovpnCASAFX-SRV && sleep 4 && ifup -v ovpnCASAFX-SRV

ls /run/resolvconf/interface
for i in ‘ls /run/resolvconf/interface/*‘;
 do printf "\n$i:\n"; cat $i;
done

invoke-rc.d resolvconf reload
cat /etc/resolv.conf
cat /run/bind/named.options
```

Orden de uso de los métodos de resolución de nombres

Aunque profundizaremos sobre ésto en el capítulo dedicado al servicio de shell, en un sistema operativo como el de dklab1 y dklab2, existen distintas formas de resolver nombres de dominio (/etc/hosts, MDNS, DNS...) y el orden en que se utilizan es definido en el fichero nsswitch.conf (anteriormente se hacía en /etc/host.conf, que todavía controla alguna funcionalidad pero que no vamos a modificar).

Por tanto, además de tener configurado /etc/resolv.conf (en nuestro caso a través de resolvconf), debemos comprobar que DNS forma parte de los métodos de resolución de nombres del sistema:

```
view /etc/nsswitch.conf
```

```
...
hosts:          files dns
...
```

```
cat /etc/hosts
```

Con las modificaciones anteriores, dklab1 se encuentra preparado para hacer de cliente

DNS de los servidores DNS que vamos a desplegar y probar en este capítulo.

1.2.3. Creación de una llave pre-compartida TSIG para actualizaciones / transferencias / control

Comenzamos la configuración de Bind9.

Usaremos TSIG (de "Transaction Signatures", rfc2845) como subprotocolo para autenticar diversos subservicios tales como las actualizaciones dinámicas de registros, las transferencias de zonas o el control de named. TSIG se basa en criptografía simétrica (es decir, las partes implicadas utilizan una llave pre-compartida) y soporta HMAC-MD5 como método de integridad y autenticación. TSIG está diseñado para protejer la comunicación entre dos partes, de forma que en un despliegue en producción, deberíamos crear más de una clave TSIG (o incluso utilizar otros mecanismos).

```
mkdir dnskeys
cd dnskeys
dnssec-keygen -a HMAC-MD5 -b 512 -n HOST 2013010101.casafx.dyndns.org.tsigkey.
```

El fichero sufijado con ".private" contiene la llave, asignada a su campo "Key: ":

```
cat K2013010101.casafx.dyndns.org.tsigkey.+157+*.private
```

Anunciamos la clave: básicamente introducimos el bloque key {} en algún fichero que podamos referenciar luego. Puesto que durante este testbed vamos a usar la misma clave para diversos subservicios, podemos hacerlo en el fichero rndc.key (pensado para comunicaciones named-rncd, siendo rndc la utilidad que implementa el lado de cliente del protocolo de control de named) en otro caso cada clave estaría en un fichero a ese propósito (y la de control en rndc.key).

```
:> /etc/bind/rndc.key
vim -o \
   /etc/bind/rndc.key K2013010101.casafx.dyndns.org.tsigkey.+157+*.private
```

```
key "2013010101.casafx.dyndns.org.tsigkey." {
 algorithm hmac-md5;
 secret "<aqui' introducimos la secuencia asignada al campo 'Key: '>";
};
```

Restringimos la lectura del material criptográfico creado:

```
chmod 600 /etc/bind/rndc.key


chown bind:root *private *key
chmod 440 *private *key
cd ..
```

Aunque la llave está registrada, tendremos que esperar a editar el fichero named.conf.local para declarar su uso.

1.2.4. Declaración de los ficheros con las zonas y forwarders

Para ello, editamos /etc/bind/named.conf

```
vim /etc/bind/named.conf
```

```
...
////fx:
//-include "/etc/bind/named.conf.options"
include "/run/bind/named.options";
 // Resolvconf copiara' nuestro named.conf.options tal cual excepto
 // la seccio'n forwarders (donde incluira' a los servidores DNS
 // que haya descubierto inspeccionando los ficheros "interfaces" y "leases"
 // de los clientes DHCP etc), entonces deja en /run/bind/named.options
 // el resultado dina'micamente generado para que pueda ser recogido
 // si asi' lo declaramos con la li'nea anterior.
////endfx

include "/etc/bind/named.conf.local";
////fx:
//Al usar vistas, no puede haber zonas definidas fuera de una vista,
//por lo que este include lo haremos en cada 'view' en named.conf.local
//-include "/etc/bind/named.conf.default-zones";
////endfx
...
```

Efectivamente, los forwarders se declaran estáticamente en el named.conf.options. Ya hemos comentado cómo resolvconf interactúa con la sección para los forwarders en ese fichero, pero no es la única sección que puede contener.

Concretamente, es muy interesante abrir una sección para redefinir algunas características de las capacidades de logging de named. Según nuestra experiencia, named puede registrar masivamente mensajes sobre comportamientos inadecuados en los forwarders. Este tipo de mensajes, llamados "lame-servers" por named, debieran ser suprimidos:

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

```
options {
        ...
        // forwarders {
        //      0.0.0.0;
        // };
        ////fx:
        //La seccio'n forwarders sera' reconstrui'da por el script
        //en /etc/resolvconf/update.d/bind al copiar este fichero
        //a /run/bind/named.options
        //No obstante, una configuracio'n esta'tica declarada aqui'
        //seri'a tal como sigue:
        forwarders {
                //8.8.8.8
                //8.8.4.4
                80.58.0.33;
                80.58.0.97;
                //o cuales fueran pertinentes.
        };
        /////endfx
        ...
};
////fx:
logging {
        category lame-servers { null; };
};
////endfx
```

Comprobamos sintácticamente el fichero editado:

```
named-checkconf /etc/bind/named.conf.options
```

1.2.5. Declaración de vistas y zonas en /etc/bind/named.conf.local

Tal y como aconseja el README.Debian, las zonas de las que somos autoridad se declaran en named.conf.local. Procuraremos que el fichero sea autoexplicativo:

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

```
...
//include "/etc/bind/zones.rfc1918";

////fx:
//
//Necesitamos declarar la llave pre-compartida que usaremos ma's adelante para:
// - usar la interfaz de control rncd: control { keys... }
// - transferencias entre zonas: server xxxx { keys...}
// - actualizacio'n de registros: update-policy { grant <key> ...}
// Todas necesitan tener una seccio'n  de forma
//           key "name" { algorithm xx; secret "..." }
// en este fichero o, au'n mejor, en otro aparte que sea inclui'do desde e'ste
// gracias a la directiva include. Lo ma's fa'cil es incluir el fichero
// /etc/bind/rndc.key donde ya tenemos registrada una seccio'n key,
// de identificador "2013010101.casafx.dyndns.org.tsigkey.", que podemos
// usar para todas las funcionalidades anteriores. No obstante, en otro
// contexto distinto al de este testbed sera' conveniente, por seguridad,
// crear una para cada caso usando dnssec-keygen etc como se expuso.
include "/etc/bind/rndc.key";
//...si, como decimos, tuvie'semos una segunda, se declarari'a con:
//key "2013010102.casafx.dyndns.org.tsigkey." {
// algorithm hmac-md5;
// secret "...";
//};
//...o mejor con algo como
//include "/etc/bind/key2.key
//...
```

```
//La tercera funcionalidad nombrada (updates) no hace a bind
//protestar (cuando se carga) si no esta' declarada la llave, pero
//lo hara' al intentar cualquier actualizacio'n de registro, y anotamos que:
// - Si el cliente usa una llave que es decodificada a la correcta pero
//   named no tiene cargada esa llave, en syslog aparecera' un "BADKEY".
// - Si el cliente viene sin llave o con una llave que es decodificada a
//   otro identificador al declarado en el bloque update-policy, sera'
//   rechazado con un "refused" en syslog.

//   La verdad es que al cargarse named (en syslog no lo dice pero)
//   strace revela que lee /etc/bind/rndc.key... tambie'n lo hace
//   la utilidad cliente del control de named, rndc. E'sto, como
//   venimos diciendo, no nos exime del include anterior.




//Puesto que no usamos el backend ldap, utilizaremos un
//esquema master-slave donde dklab1 es master y dklab2 slave:
 acl "slavesacl" {
  10.168.1.2;
 };



// Vistas (nota: el orden importa, de forma que el cliente ve la vista
// cuya acl case antes).
acl "vpn-net" {
 10.168.1.0/24;
};
```

```
view "vpn" {
 match-clients { vpn-net; };
 //opciones de vista:
 recursion yes;

  //Zonas: http://www.bind9.net/manual/bind/9.3.2/Bv9ARM.ch06.html

include "/etc/bind/named.conf.default-zones";

zone "casafx.dyndns.org" {
        type master;
        file "/etc/bind/db.casafx.dyndns.org-VPN";

        //update-policy:
        //Constitui'da por una o varias reglas grant o deny.
        //http://www.bind9.net/manual/bind/9.3.2/Bv9ARM.ch06.html
        update-policy {
//(grant|deny) (key|krbprinc)                nametype       name          [types]
 grant 2013010101.casafx.dyndns.org.tsigkey. wildcard *.casafx.dyndns.org.  A;
                //http://www.ops.ietf.org/dns/dynupd/secure-ddns-howto.html
                //Es posible el nametype "subdomain" que permite la
                //actualizacio'n tanto del valor de name como de subdominios.
                //En nuestro caso estamos interesados so'lo en los subdominios,
                //usamos wildcard. Para so'lo el nombre en name, se usari'a
                //el nametype name. El u'ltimo nametype es self, que requiere
                //que la llave, en concreto su campo identity case con el
                //registro que se quiere actualizar (indep. del campo name).
                //   Por motivos de seguridad puede ser conveniente crear
                //otra llave porque e'sa ya la tiene por ejemplo
                //el secundario... pero en este testbed no es el caso.
        };
};
```

```
        // Por ahora conservo, y no redefino, estos default para lista de
        // IP/ifaces que pueden consultar la zona:
        // allow-query { any; };
        // allow-query-on { any; };

        // Zonas secundarias a las que transferimos
        // (tampoco definimos previamente acl "slavesacl"
        //  si no usamos el backend ldap)
        allow-transfer { slavesacl; };
};
```

```
 //...nuestra zona inversa para la vista vpn:
 zone "1.168.10.IN-ADDR.ARPA." {
        type master;
        file "/etc/bind/db.10.168.1";

        update-policy {
//(grant|deny) (key|krbprinc)                nametype       name          [types]
grant 2013010101.casafx.dyndns.org.tsigkey. wildcard *.casafx.dyndns.org.  A;
                //de nuevo, la llave tiene q llamarse con un "FQDN"
                //si usamos de nametype "self"
                //http://www.ops.ietf.org/dns/dynupd/secure-ddns-howto.html
                //Y mientras no sea asi', pues no necesitamos una seccio'n
                //key "casafx.dyndns.org." {}
                //para renombrar la clave que ya pusimos
                //en /etc/bind/rndc.key.
                //   Por motivos de seguridad puede ser conveniente crear
                //otra llave porque e'sa ya la tiene por ejemplo
                //el secundario... pero en este testbed no es el caso.
        };

        allow-transfer { slavesacl; };
 };

};
```

```
view "external" {
 match-clients { any; };
 //opciones de vista:
 recursion no;

 //Zonas (en principio, las mismas que en la otra vista
 //       -excepto zona inversa-, apuntando a otros ficheros)

include "/etc/bind/named.conf.default-zones";

 zone "casafx.dyndns.org" {
        type master;
        file "/etc/bind/db.casafx.dyndns.org-EXT";

        //update policy:
        //http://www.bind9.net/manual/bind/9.3.2/Bv9ARM.ch06.html
        update-policy {
//(grant|deny) (key|krbprinc)                nametype       name          [types]
grant 2013010101.casafx.dyndns.org.tsigkey. wildcard *.casafx.dyndns.org.  A;
        //... mismas notas sobre la llave que se dieron en la zona vpn.
        };

        // Por ahora conservo, y no redefino, estos default
        // para lista de IP/ifaces q pueden consultar la zona:
        // allow-query { any; };
        // allow-query-on { any; };

        // Zonas secundarias a las q transferir:
        allow-transfer { slavesacl; };
 };
```

```
 //...mi zona inversa para la red base (externa, no vpn):
 zone "1.168.192.IN-ADDR.ARPA." {
        type master;
        file "/etc/bind/db.192.168.1";

        update-policy {
//(grant|deny) (key|krbprinc)                nametype      name          [types]
grant 2013010101.casafx.dyndns.org.tsigkey. wildcard *.casafx.dyndns.org.  A;
        //... mismas notas sobre la llave que se dieron en la zona vpn.
        };
        allow-transfer { slavesacl; };
 };
};
```

```
 //Regla acl para la interfaz de control rndc de named;
 //pueden declararse varias secciones controls, por ejemplo para que escuche
 //en so'lo algunas interfaces. A este respecto se puede sustituir la IP
 //por '*' y, entonces, con una so'la seccio'n controls hacemos que escuche
 //en todas las interfaces.
 //http://www.bind9.net/manual/bind/9.3.2/Bv9ARM.ch06.html#id2552471
controls {
     inet 127.0.0.1 allow {
       127.0.0.1;
     }
     //
     keys {
       "2013010101.casafx.dyndns.org.tsigkey.";
     };
 //El puerto por defecto es tcp 953.
};

//Llave asociadas a las transferencias con el slave:
//http://www.bind9.net/manual/bind/9.3.2/Bv9ARM.ch06.html#server_statement_grammar
server 10.168.1.2 {
     keys {
       "2013010101.casafx.dyndns.org.tsigkey.";
     };
//Las transferencias se hacen por el puerto tcp 53
};
////endfx
```

Comprobación sintáctica:

```
named-checkconf  /etc/bind/named.conf
```

Para verificar una llave hace falta que /var/cache/bind sea escribible por el grupo, hecho que ya está resuelto como podemos comprobar con:

```
stat /var/cache/bind
```

1.2.6. Zona casafx.dyndns.org

Vista "vpn"

La resolución directa para la vista "vpn", como se declaró en el named.conf.local, la almacenamos en /etc/bind/db.casafx.dyndns.org-VPN:

```
vim /etc/bind/db.casafx.dyndns.org-VPN
```

(El registro exim4dkim._domainkey.casafx.dyndns.org. aparece dividido en 2 partes delimitadas por \ y cambio de línea. Realmente debe aparecer en una sóla línea, sin \ y cambio de línea)

```
;http://www.zytrax.com/books/dns/ch8/index.html#zone
;El archivo de zona contiene:
; a) "directives", q comienzan por '$': $TTL y $ORIGIN
;    $TTL debe aparecer antes que el primer RR, puede llevar prefijos m,h,w,...
;    $ORIGIN se aplica si 'name' no acaba en '.', es decir, unqualified name.
; b) RR, los resource records en si', de formato:
;       (nota: si nombre es @, se sustituye por origin                    )
;       (      si vaci'o/tab/space, se usa el default o lo u'ltimo u ORIGIN )
;    NAME  TTL  CLASS  TYPE  TYPE-SPECIFIC-DATA
$ORIGIN casafx.dyndns.org.
$TTL 604800 ; http://www.zytrax.com/books/dns/apa/ttl.html

; Definimos el registro de clase SOA, Start Of Authority; puesto que usaremos
; "Dynamic Updates", el nombre del resolvedor de nombres que le sigue debiera
; ser el del master, en otro caso podri'a ser cualquiera autoridad de sus
; zonas, los cuales definimos despue's con registros de clase NS.
; Adema's, por petenecer a la zona (casafx.dyndns.org), debemos definir
; luego el correspondiente registro A de ns1 y ns2.
; http://www.zytrax.com/books/dns/ch8/soa.html
@               IN SOA  ns1.casafx.dyndns.org. hostmaster.casafx.dyndns.org. (
                              2010020654 ; serial
                              10800      ; refresh (3 hours)
                              7200       ; retry (2 hours)
                              1296000    ; expire (2 weeks 1 day)
                              172800     ; minimum (2 days)
                              )
```

```
                IN NS   ns1.casafx.dyndns.org.
                IN NS   ns2.casafx.dyndns.org.
                IN MX   10 dklab1.casafx.dyndns.org.
                IN MX   10 dklab2.casafx.dyndns.org.
                IN HINFO "PC" "Debian 7"
                IN TXT "View: VPN"
;               IN A 95.120.206.200
```

```
                ; Sender Policy Framework para el servicio de correo,
                ; usamos valores por defecto seguros ( ?all y no -all )
                IN SPF "v=spf1 +mx/24 ?all"
                IN TXT "v=spf1 +mx/24 ?all"
                ; Domain Keys Identified Mail
                  ; a) DKIM selector record
exim4dkim._domainkey.casafx.dyndns.org. \
IN TXT "v=DKIM1; s=email; g=*; t=y:s; h=*; k=rsa; p=<public key>"
;... el registro anterior debiera aparecer en una sola li'nea.
                    ; b) policy record (viejo formato OSP y nuevo ADSP)
_domainkey.casafx.dyndns.org.           IN TXT "t=y; o=~;"
_adsp._domainkey.casafx.dyndns.org.     IN TXT "dkim=unknown"

ns1.casafx.dyndns.org.                  IN A    10.168.1.1
ns2.casafx.dyndns.org.                  IN A    10.168.1.2
dklab1                                  IN A    10.168.1.1
dklab2                                  IN A    10.168.1.2
ntp                                     IN A    10.168.1.1
ntp                                     IN A    10.168.1.2
nowhere                                 IN A    10.168.1.255
```

```
_ldap._tcp                                  IN SRV  0 0 389 dklab1.casafx.dyndns.org.
_ldap._tcp                                  IN SRV  0 0 389 dklab2.casafx.dyndns.org.
ldap1                                       IN CNAME dklab1.casafx.dyndns.org.
ldap2                                       IN CNAME dklab2.casafx.dyndns.org.

_kerberos                                   IN TXT  "CASAFX.DYNDNS.ORG"
_kerberos._udp                              IN SRV  0 0 88 dklab1.casafx.dyndns.org.
_kerberos._udp                              IN SRV  0 0 88 dklab2.casafx.dyndns.org.
;_kerberos-master._udp                       IN SRV  0 0 88 dklab1.casafx.dyndns.org.
_kpasswd._udp                               IN SRV  0 0 88 dklab1.casafx.dyndns.org.
_kpasswd._udp                               IN SRV  0 0 88 dklab2.casafx.dyndns.org.
_kerberos-adm._tcp                          IN SRV  0 0 749 dklab1.casafx.dyndns.org.
_kerberos-adm._tcp                          IN SRV  0 0 749 dklab2.casafx.dyndns.org.
krb1                                        IN CNAME dklab1.casafx.dyndns.org.
krb2                                        IN CNAME dklab2.casafx.dyndns.org.
```

```
_afs3-vlserver._udp                     IN SRV  0 0 7003 dklab1.casafx.dyndns.org.
_afs3-vlserver._udp                     IN SRV  0 0 7003 dklab2.casafx.dyndns.org.
_afs3-prserver._udp                     IN SRV  0 0 7002 dklab1.casafx.dyndns.org.
_afs3-prserver._udp                     IN SRV  0 0 7002 dklab2.casafx.dyndns.org.
casafx.dyndns.org.                      IN AFSDB  1 dklab1.casafx.dyndns.org.
casafx.dyndns.org.                      IN AFSDB  1 dklab2.casafx.dyndns.org.
afs1                                    IN CNAME dklab1.casafx.dyndns.org.
afs2                                    IN CNAME dklab2.casafx.dyndns.org.

_xmpp-client._tcp                       IN SRV  0 0 5222 dklab1.casafx.dyndns.org.
_xmpp-client._tcp                       IN SRV  0 0 5222 dklab2.casafx.dyndns.org.
_xmpp-server._tcp                       IN SRV  0 0 5269 dklab1.casafx.dyndns.org.
_xmpp-server._tcp                       IN SRV  0 0 5269 dklab2.casafx.dyndns.org.
jabber1                                 IN CNAME dklab1.casafx.dyndns.org.
jabber2                                 IN CNAME dklab2.casafx.dyndns.org.
```

```
_submission._tcp                        IN SRV  0 0 587 dklab1.casafx.dyndns.org.
_submission._tcp                        IN SRV  0 0 587 dklab2.casafx.dyndns.org.
_sieve._tcp                             IN SRV  0 0 4190 dklab1.casafx.dyndns.org.
_sieve._tcp                             IN SRV  0 0 4190 dklab2.casafx.dyndns.org.
mx1                                     IN CNAME dklab1.casafx.dyndns.org.
mx2                                     IN CNAME dklab2.casafx.dyndns.org.
managesieve1                            IN CNAME dklab1.casafx.dyndns.org.
managesieve2                            IN CNAME dklab2.casafx.dyndns.org.

_imap._tcp                              IN SRV  0 0 143 dklab1.casafx.dyndns.org.
_imap._tcp                              IN SRV  0 0 143 dklab2.casafx.dyndns.org.
imap1                                   IN CNAME dklab1.casafx.dyndns.org.
imap2                                   IN CNAME dklab1.casafx.dyndns.org.
```

La resolución inversa asociada:

```
vim /etc/bind/db.10.168.1
```

```
$ORIGIN 1.168.10.IN-ADDR.ARPA.
$TTL 604800
@ IN SOA ns1.casafx.dyndns.org. hostmaster.casafx.dyndns.org. (
  2010013000   ; Serial
       10800   ; Refresh (3 hours)
        7200   ; Retry (2 hours)
     1296000   ; Expire (15 days)
      172800 ) ; Negative Cache TTL (2 days)
;;;; Registros NS de la zona, para saber a que' servidores
;    hay q preguntar para la traduccio'n inversa:
@       IN      NS      ns1.casafx.dyndns.org.
@       IN      NS      ns2.casafx.dyndns.org.
;;;; Resolucio'n inversa; se aconseja no an~adir redundancias: un
;    u'nico FQDN por IP
2       IN      PTR     dklab2.casafx.dyndns.org.
1       IN      PTR     dklab1.casafx.dyndns.org.
```

Vista "external"

Resolución directa; no especificamos más que lo mínimo, ya que nuestros servicios se despliegan sobre la VPN:

```
vim /etc/bind/db.casafx.dyndns.org-EXT
```

```
;http://www.zytrax.com/books/dns/ch8/index.html#zone
;El archivo de zona contiene:
; a) "directives", que comienzan por '$': $TTL y $ORIGIN
;    $TTL debe aparecer antes que el primer RR, puede llevar prefijos m,h,w,...
;    $ORIGIN se aplica cuando 'name' no acaba en '.', es decir, unqualified name.
; b) RR, los resource records en si', de formato:
;       (nota: si nombre es @, se sustituye por origin                    )
;       (      si vaci'o/tab/space, se usa el default o lo u'ltimo u ORIGIN )
;    NAME  TTL  CLASS  TYPE  TYPE-SPECIFIC-DATA
$ORIGIN casafx.dyndns.org.
$TTL 604800 ; http://www.zytrax.com/books/dns/apa/ttl.html

; Definimos el registro de clase SOA, Start Of Authority; puesto que usaremos
; "Dynamic Updates", el nombre del resolvedor de nombres que le sigue debiera
; ser el del master, en otro caso podri'a ser cualquiera autoridad de sus
; zonas, los cuales definimos despue's con registros de clase NS.
; Adema's, por petenecer a la zona (casafx.dyndns.org), debemos definir
; luego el correspondiente registro A de ns1 y ns2.
@               IN SOA  ns1.casafx.dyndns.org. hostmaster.casafx.dyndns.org. (
                              2010020654 ; serial
                              10800      ; refresh (3 hours)
                              7200       ; retry (2 hours)
                              1296000    ; expire (2 weeks 1 day)
                              172800     ; minimum (2 days)
                              )
```

```
                IN NS   ns1.casafx.dyndns.org.
                 IN NS   ns2.casafx.dyndns.org.
                 IN MX   10 dklab1.casafx.dyndns.org.
                 IN MX   10 dklab2.casafx.dyndns.org.
                 IN HINFO "PC" "Debian 7"
                 IN TXT "View: External"
;                 IN A 95.120.206.200
ns1.casafx.dyndns.org.                  IN A    192.168.1.1
ns2.casafx.dyndns.org.                  IN A    192.168.1.2
dklab1                                   IN A    192.168.1.10
dklab2                                   IN A    192.168.1.20
nowhere                                  IN A    192.168.1.255
```

Resolución inversa:

```
vim /etc/bind/db.192.168.1
```

```
$ORIGIN 1.168.192.IN-ADDR.ARPA.
$TTL 604800
@ IN SOA ns1.casafx.dyndns.org. hostmaster.casafx.dyndns.org. (
  2010013000   ; Serial
       10800   ; Refresh (3 hours)
        7200   ; Retry (2 hours)
     1296000   ; Expire (15 days)
      172800 ) ; Negative Cache TTL (2 days)
;;;; Registros NS de la zona, para saber a que' servidor hay que
;    preguntar para la traduccio'n inversa:
@       IN      NS      ns1.casafx.dyndns.org.
@       IN      NS      ns2.casafx.dyndns.org.
;;;; Resolucio'n inversa; se aconseja no an~adir redundancias:
;    un u'nico FQDN por IP:
20       IN      PTR     dklab2.casafx.dyndns.org.
10       IN      PTR     dklab1.casafx.dyndns.org.
```

Comprobaciones sintácticas para las zonas casafx.dyndns.org

Utilizamos named-checkzone (con los ficheros de configuración fue named-checkconf):

```
named-checkzone -c IN  casafx.dyndns.org. /etc/bind/db.casafx.dyndns.org-VPN
named-checkzone -n ignore -c IN 1.168.10.IN-ADDR.ARPA. /etc/bind/db.10.168.1
```

1.2.7. Zona localhost

No hay vistas diferenciadas para esta zona.
Resolución directa:

```
cp /etc/bind/db.local /tmp/db.local.orig
:> /etc/bind/db.local

vim /etc/bind/db.local
```

```
;
; BIND data file for local loopback interface
;
$ORIGIN localhost.
$TTL    604800
@       IN      SOA     ns.casafx.dyndns.org. root.casafx.dyndns.org. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns.casafx.dyndns.org.
@       IN      A       127.0.0.1
@       IN      AAAA    ::1
```

Resolución inversa:

```
cp /etc/bind/db.127 /tmp/db.127.orig
:> /etc/bind/db.127

vim /etc/bind/db.127
```

```
;
; BIND reverse data file for local loopback interface
;
$ORIGIN 127.IN-ADDR.ARPA.
$TTL    604800
@       IN      SOA     ns.casafx.dyndns.org.      hostmaster.casafx.dyndns.org. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
@       IN      NS      ns.casafx.dyndns.org.
1.0.0   IN      PTR     localhost.
```

Comprobaciones sintácticas para la zona localhost

```
named-checkzone -c IN localhost. /etc/bind/db.local
named-checkzone -c IN 127.IN-ADDR.ARPA. /etc/bind/db.127
```

1.2.8. Tests sobre resoluciones

Reiniciamos el servidor:

```
invoke-rc.d bind9 restart
```

Conociendo que named escucha por defecto en todas nuestras interfaces de red:

- Preguntamos a 10.168.1.1 y por tanto nuesta IP de origen está en esa red, luego named nos ofrece la vista "vpn":

```
dig    @10.168.1.1 dklab1.casafx.dyndns.org.  any +short
dig -x @10.168.1.1 1.1.168.10.IN-ADDR.ARPA. any +short
```

- Preguntamos con IP de origen 127.0.1.1 y por tanto se nos ofrece la vista "external":

```
dig    @127.0.0.1 dklab1.casafx.dyndns.org.  any +short
dig -x @127.0.0.1 1.1.168.10.IN-ADDR.ARPA. any +short
```

- Si preguntamos por una de las zonas comunes, la información devuelta en ambas vistas es idéntica:

```
dig @10.168.1.1 localhost. any +short
dig @127.0.0.1 localhost. +short
dig -x @10.168.1.1 1.0.0.127.IN-ADDR.ARPA. any +short
dig -x @127.0.0.1 1.0.0.127.IN-ADDR.ARPA. any +short
```

Ping usará las librerías para resolución de nombres del sistema, que según el /etc/resov.conf preguntan a 10.168.1.X y por tanto se le muestra la vista "vpn".

```
PINGLIST="localhost. ns.casafx.dyndns.org. dklab1 dklab2"
for i in ${PINGLIST};
 do cmd="ping -c1 -a $i"; echo $cmd; eval $cmd; sleep 2;
done | grep "1 "
```

Evidentemente, si un ordenador externo a la VPN etc tiene configurados a dklab1 para la resolución de nombres a cualquier cliente de servicios se le presentaría la otra vista. Pensemos por ejemplo que hemos comprado el subdominio casafx de dyndns.org. y, por tanto, somos autoridad para la zona correspondiente. Entonces desde el punto de vista de named, la misma política de nombres de dominio y servicios sirve para fuera de la vpn como para dentro, a la vez que tras la resolución se encamina a cada entidad de la forma más idónea.

1.2.9. Tests sobre DNS Dynamic updates

El paquete dnsutils trae el comando nsupdate, el cual constituye una implementación del lado del cliente del protocolo Dynamic DNS Updates definido en el rfc2136.

Evidentemente, puesto que las vistas duplican los registros referidos a un mismo nombre, la actualización de un registro es dependiente del sistema de vistas. Por tanto hay que ser cuidadoso respecto a la dirección IP del servidor dns que se le da a nsupdate, pues condiciona la IP de origen de la petición y por tanto la vista que acaba modificando.

```
ls -ld /etc/bind
```

Pero el usuario bind necesita poder crear en /etc/bind los archivos journal ".jnl", entonces:

```
chmod g+w /etc/bind
```

```
vim /etc/bind/template_nsupdate.cmd
```

```
server SERVIDORDNS
prereq yxdomain DOMINIO
update delete DOMINIO A
send
server SERVIDORDNS
;prereq nxdomain DOMINIO
update add DOMINIO 300 A DIRECCION
send
```

```
chmod a-w /etc/bind/template_nsupdate.cmd
```

Atención, insistimos, a que sustituímos SERVIDORDNS por 10.168.1.1 y no 127.0.0.1: entonces la vista modificada ha de ser "vpn":

```
sed -e s/"DOMINIO"/nowhere.casafx.dyndns.org/g  \
    -e s/"DIRECCION"/10.168.1.254/g             \
    -e s/"SERVIDORDNS"/10.168.1.1/g             \
    /etc/bind/template_nsupdate.cmd > /tmp/test_nsupdate.cmd
```

Si la edición no interactiva de sed fue bien:

```
cat /tmp/test_nsupdate.cmd
```

Ya podemos comprobar el estado actual de los registros para a continuación realizar la actualización:

```
dig @10.168.1.1 nowhere.casafx.dyndns.org. +short
dig @127.0.0.1 nowhere.casafx.dyndns.org. +short
nsupdate -k dnskeys/K2013010101.casafx.dyndns.org.tsigkey.+157+18253.private \
         /tmp/test_nsupdate.cmd
```

Veamos si, efectivamente, el registro del nombre "nowhere" ha cambiado en la vista "vpn":

```
dig @10.168.1.1 nowhere.casafx.dyndns.org. +short
dig @127.0.0.1 nowhere.casafx.dyndns.org. +short
```

Cuando esté configurado dklab2 y se transfieran los cambios automáticamente:

```
dig @ns2.casafx.dyndns.org nowhere.casafx.dyndns.org. +short
```

1.2.10. Tests sobre la interfaz de control con rndc

Nota: el control de named sólo depende de la sección controls, que estaba fuera de las vistas etc, nada de lo dicho para éstas intervienen aquí.

```
rndc -s 10.168.1.1 -k /etc/bind/rndc.key reload
```

```
...rndc: connect failed: 10.168.1.1#953: connection refused
```

```
rndc -s 127.0.0.1 -k /etc/bind/rndc.key reload
```

```
...server reload successful
```

Puesto que esos flags le pasan los valores por defecto, se puede simplificar la llamada a rndc:

```
rndc reload
rndc status
rndc --help
```

1.3. DKLAB2

1.3.1. Instalación de Bind9 y dnsutils

```
apt-get bind9 dnsutils
```

1.3.2. DKLAB2 como cliente DNS; instalación de resolvconf; nsswitch.conf

```
apt-get install resolvconf
```

```
 -"Prepare /etc/resolv.conf for dynamic updates?":
 Yes
```

... luego /etc/resolv.conf es un enlace a /etc/resolvconf/run/resolv.conf; por su lado /etc/resolv.conf.d/tail no es un enlace a /etc/resolv.conf.d/original. Ya podemos declarar la información sobre resolvedores de nombres en la configuración de las interfaces de red:

```
view /usr/share/doc/resolvconf/README.gz


vim /etc/network/interfaces
```

```
...
iface ovpnCASAFX inet manual
 ...
 dns-nameservers 10.168.1.2 10.168.1.1
 dns-search casafx.dyndns.org.
 #dns-options rotate
```

Por su lado, para que los primeros nameserver en /etc/resolv.conf sean 10.168.1.1 o .2, y así asegurar que se devuelvan las IP según la vista dns para la vpn:

```
vim /etc/resolvconf/interface-order
```

```
####fx
#E'sta debiera ser la primera li'nea-no-comentario del fichero:
ovpnCASAFX*
####endfx
...
```

Reiniciamos las partes implicadas.

```
cp /usr/share/doc/resolvconf/resolvconf-update-bind /etc/resolvconf/update.d/bind
chmod a+x /etc/resolvconf/update.d/bind
```

Tras este paso previo, ya:

```
IFACE=‘ip route | grep default | awk '{print $5}'‘
ifdown -v $IFACE && sleep 4 && ifup -v $IFACE
ifdown -v ovpnCASAFX && sleep 4 && ifup -v ovpnCASAFX

ls /run/resolvconf/interface
for i in ‘ls /run/resolvconf/interface/*‘;
 do printf "\n$i:\n"; cat $i;
done

invoke-rc.d resolvconf reload
cat /etc/resolv.conf
```

Orden de uso de los métodos de resolución de nombres

Debemos comprobar que DNS forma parte de los métodos de resolución de nombres del sistema:

```
view /etc/nsswitch.conf
```

```
...
hosts:          files dns
...
```

```
cat /etc/hosts
```

Con las modificaciones anteriores, dklab2 se encuentra preparado para hacer de cliente DNS de los servidores DNS que vamos a desplegar y probar en este capítulo.

1.3.3. Transferencia de la llave pre-compartida TSIG para actualizaciones / transferencias / control

Utilizamos SCP como medio confidencial:

```
scp 10.168.1.1:/etc/bind/rndc.key /etc/bind/rndc.key
cat /etc/bin/rndc.key
```

1.3.4. Declaración de los ficheros con las zonas y forwarders

Para ello, editamos /etc/bind/named.conf:

```
vim /etc/bind/named.conf
```

```
...
////fx:
//-include "/etc/bind/named.conf.options"
include "/run/bind/named.options";
 // Resolvconf copiara' nuestro named.conf.options tal cual excepto
 // la seccio'n forwarders (donde incluira' a los servidores DNS
 // que haya descubierto inspeccionando los ficheros "interfaces" y "leases"
 // de los clientes DHCP etc), entonces deja en /run/bind/named.options
 // el resultado dina'micamente generado para que pueda ser recogido
 // desde aqui'.
////endfx

include "/etc/bind/named.conf.local";

////fx:
//Al usar vistas, no puede haber zonas definidas fuera de una vista,
//por lo que este include lo haremos en cada view en named.conf.local
//-include "/etc/bind/named.conf.default-zones";
////endfx
```

Declaramos la sección forwarders con que interactúa resolvconf y suprimimos los mensajes "lame-servers".

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

```
options {
        ...
        // forwarders {
        //      0.0.0.0;
        // };

        ////fx:
        //La seccio'n forwarders sera' reconstrui'da por el script
        //en /etc/resolvconf/update.d/bind al copiar este fichero
        //a /run/bind/named.options
        //No obstante, una configuracio'n esta'tica declarada aqui'
        //seri'a tal como sigue:
        forwarders {
                //8.8.8.8
                //8.8.4.4
                80.58.0.33;
                80.58.0.97;
                //o cuales fueran pertinentes.
        };
        /////endfx
        ...
};
////fx:
logging {
        category lame-servers { null; };
};
////endfx
```

Comprobamos sintácticamente el fichero editado:

```
named-checkconf /etc/bind/named.conf.options
```

1.3.5. Declaración de vistas y zonas en /etc/bind/named.conf.local

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

```
...
//include "/etc/bind/zones.rfc1918";


////fx:
//
//Necesitamos declarar la llave pre compartida que usaremos ma's adelante para:
// - usar la interfaz de control rncd: control { keys... }
// - transferencias entre zonas: server xxxx { keys...}
// - actualizacio'n de registros: update-policy { grant <key> ...}
// Todas necesitan tener una seccio'n de la forma
//           key "name" { algorithm xx; secret "..." }
// en este fichero o, au'n mejor, en otro aparte que sea inclui'do desde
// e'ste gracias a la directiva include. Lo ma's fa'cil es incluir el fichero
// /etc/bind/rndc.key donde ya tenemos una registrada una seccio'n key,
// de identificador "2013010101.casafx.dyndns.org.tsigkey.", que podemos
// usar para todas las funcionalidades anteriores. No obstante, en otro
// contexto distinto al de este testbed sera' conveniente, por seguridad,
// crear una para cada caso usando dnssec-keygen etc como se expuso.
include "/etc/bind/rndc.key";
```

```
//La tercera funcionalidad nombrada (updates) no hace a bind
//protestar (cuando se carga) si no esta' declarada la llave, pero
//lo hara' al intentar cualquier actualizacio'n de registro, y anotamos que:
// - Si el cliente usa una llave que es decodificada a la correcta pero
//   named no tiene cargada esa llave, en syslog aparecera' un "BADKEY".
// - Si el cliente viene sin llave o con una llave que es decodificada a
//   otro identificador al declarado en el bloque update-policy, sera'
//   rechazado con un "refused" en syslog.

//   Al levantarse named (en syslog no lo dice pero)
//   strace revela que lee /etc/bind/rndc.key... tambie'n lo hace
//   la utilidad cliente del control de named, rndc.



//Usamos un esquema master-slave:
//(No sabemos por que' named-checkconf no "ve" esta acl;
// nos obliga a quitarla y usar la IP cada vez que haga falta).
// acl "mastersacl" {
// 10.168.1.1;
//};

// Vistas
//(nota: el orden importa, de forma que el cliente ve la vista cuya acl
//       case antes)
acl "vpn-net" {
 10.168.1.0/24;
};
```

```
view "vpn" {
 match-clients { vpn-net; };
 //opciones de vista:
 recursion yes;

 //Zonas:
include "/etc/bind/named.conf.default-zones";
 zone "casafx.dyndns.org" {
       type slave;
       file "slaves/sec.casafx.dyndns.org-VPN";
       //No declaramos update-policy {} al ser un slave, es decir, las
       //actualizaciones se realizan sobre el master.

        // Por ahora conservo, y no redefino, estos default para lista de
        // IP/ifaces que pueden consultar la zona:
        // allow-query { any; };
        // allow-query-on { any; };

       // Cu'ales son nuestros master:
       masters { 10.168.1.1; };
 };
```

```
 //...nuestra zona inversa para la vpn:
 zone "1.168.10.IN-ADDR.ARPA." {
        type slave;
        file "slaves/sec.db.10.168.1";
        //No update-policy {} al hacer de slave
        //Su master:
        masters { 10.168.1.1; };
 };

};



view "external" {
 match-clients { any; };
 //opciones de vista:
 recursion no;

 //Zonas (en principio, las mismas que en la otra vista
 //       -excepto zona inversa-, apuntando a otros ficheros)

 include "/etc/bind/named.conf.default-zones";
```

```
zone "casafx.dyndns.org" {
       type slave;
       file "slaves/sec.casafx.dyndns.org-EXT";

       //No update-policy {} al ser un slave.

       // Por ahora conservo, y no redefino, estos default
       // para lista de IP/ifaces q pueden consultar la zona:
       // allow-query { any; };
       // allow-query-on { any; };

       // Nuestros masters
       masters { 10.168.1.1; };
 };
//...nuestra zona inversa para la vista "external":
zone "1.168.192.IN-ADDR.ARPA." {
        type slave;
        file "slaves/sec.db.192.168.1";
        //No update-policy {} al ser un slave
        masters { 10.168.1.1; };
};

};
```

```
 //Regla acl para la interfaz de control rndc de named;
 //pueden declararse varias de estas secciones.
 //http://www.bind9.net/manual/bind/9.3.2/Bv9ARM.ch06.html#id2552471
controls {
     inet 127.0.0.1 allow {
       127.0.0.1;
     }
     keys {
       "2013010101.casafx.dyndns.org.tsigkey.";
     };
 //El puerto por defecto es tcp 953.
};
////endfx
```

Comprobación sintáctica:

```
named-checkconf  /etc/bind/named.conf
```

Para verificar una llave hace falta que /var/cache/bind sea escribible por el grupo, hecho que ya está resuelto como podemos comprobar con:

```
stat /var/cache/bind
```

El directorio slaves asociado a las zonas de las que se es secundario, debe estar creado antes de ser usado:

```
mkdir /var/cache/bind/slaves
chown root:bind /var/cache/bind/slaves
chmod g+w /var/cache/bind/slaves
ls -ld /var/cache/bind/slaves
```

1.3.6. Zona localhost

La zona localhost no está bajo el esquema master-slave. Tampoco - y ésto es, no obstante, totalmente independiente de la condición anterior - está bajo el sistema de vistas.

Resolución directa:

```
cp /etc/bind/db.local /tmp/db.local.orig
:> /etc/bind/db.local


vim /etc/bind/db.local
```

```
;
; BIND data file for local loopback interface
;
$ORIGIN localhost.
$TTL    604800
@       IN      SOA     ns.casafx.dyndns.org. root.casafx.dyndns.org. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns.casafx.dyndns.org.
@       IN      A       127.0.0.1
@       IN      AAAA    ::1
```

Resolución inversa:

```
cp /etc/bind/db.127 /tmp/db.127.orig
:> /etc/bind/db.127


vim /etc/bind/db.127
```

```
;
; BIND reverse data file for local loopback interface
;
$ORIGIN 127.IN-ADDR.ARPA.
$TTL    604800
@       IN      SOA     ns.casafx.dyndns.org.      hostmaster.casafx.dyndns.org. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
@       IN      NS      ns.casafx.dyndns.org.
1.0.0   IN      PTR     localhost.
```

Comprobaciones sintácticas para la zona localhost

```
named-checkzone -c IN localhost. /etc/bind/db.local
named-checkzone -c IN 127.IN-ADDR.ARPA. /etc/bind/db.127
```

1.3.7. Zona casafx.dyndns.org transferida desde dklab1; bug

```
invoke-rc.d bind9 restart


tail -f /var/log/syslog | grep named
```

Puede hacerse lo mismo en dklab1. Lo importante es que, si no fuera por un bug, syslog en ambas máquinas debería mostrar un diálogo como:

```
dklab2 named:
  zone 1.168.10.IN-ADDR.ARPA/IN: Transfer started.
dklab2 named:
  transfer of '1.168.10.IN-ADDR.ARPA/IN' from 10.168.1.1#53:
  connected using 10.168.1.2#35433


dklab1 named:
  client 10.168.1.2#35433: transfer of '1.168.10.IN-ADDR.ARPA/IN': AXFR started
dklab1 named:
  client 10.168.1.2#35433: transfer of '1.168.10.IN-ADDR.ARPA/IN': AXFR ended


dklab2 named:
  zone 1.168.10.IN-ADDR.ARPA/IN: transferred serial 2010013000
dklab2 named:
  transfer of '1.168.10.IN-ADDR.ARPA/IN' from 10.168.1.1#53:
  Transfer completed: 1 messages, 5 records, 202 bytes, 0.079
  secs (2556 bytes/sec)
```

```
dklab2 named: dumping master file: slaves/tmp-CDaoYgWT41: open: file not found
dklab2 named: zone casafx.dyndns.org/IN: Transfer started.
dklab2 named:
  transfer of 'casafx.dyndns.org/IN' from 10.168.1.1#53: connected
  using 10.168.1.2#36946

dklab1 named:
  client 10.168.1.2#36946: transfer of 'casafx.dyndns.org/IN': AXFR started
dklab1 named:
  client 10.168.1.2#36946: transfer of 'casafx.dyndns.org/IN': AXFR ended

dklab2 named:
  zone casafx.dyndns.org/IN: transferred serial 2010020660
dklab2 named:
  transfer of 'casafx.dyndns.org/IN' from 10.168.1.1#53:
  Transfer completed: 1 messages, 11 records, 305 bytes, 0.076
  secs (4013 bytes/sec)
dklab2 named:
  zone casafx.dyndns.org/IN: sending notifies (serial 2010020660)
dklab2 named:
  dumping master file: slaves/tmp-fnl9Y4yDG4: open: file not found

dklab1 named:
  client 10.168.1.2#25607: received notify for zone 'casafx.dyndns.org'
```

... pero no ocurre así. La salida anterior corresponde a un escenario sin vistas; al usar vistas hemos comprobado que el proceso named slave (dklab2), pregunta por las zonas de la vista "vpn", las consigue correctamente y cuando va a pedir las de la vista externa... las pide como si fuesen también de la vista "vpn". Como consecuencia de ello:

- sólo se le devolverá algo en el caso de que coincida el nombre (efectivamente la zona casafx.dyndns.org está en la vista externa, luego la pide (mal, porque decimos que allí dice buscar en la vista vpn de nuevo) y se le transfiere. La zona inversa ya no puede coincidir en el nombre (1.168.192-IN-ADDR.ARPA.) luego esa zona no la puede conseguir.

- lo que se le devuelve lo guarda con el nombre correcto que leyó en la vista "external". Por tanto aunque baja un sec.casafx.dyndns.org-EXT, éste es igual al sufijado "-VPN" de la vista "vpn". El sec.db.192.168.1, decíamos, no se lo puede bajar, como hemos explicado.

Nos costaba concluir que bind9 tuviese un bug en este sentido, pero seguimos probando y todo apunta a ello. Así, si en el master declaramos en primer lugar la vista "external", es ésta la que siempre consulta: el fallo está en el master y hace que siempre intente transferir zonas de la primera vista que se encuentre. Los mensajes en syslog:

```
dklab1 named: client 10.168.1.2#53034: view vpn: transfer
  of 'casafx.dyndns.org/IN': AXFR started
dklab2 named: transfer of 'casafx.dyndns.org/IN/external' from
  10.168.1.1#53: Transfer completed: 1 messages, 11 records, 305 bytes,
  0.020 secs (15250 bytes/sec)
```

... es decir dklab1 habla de view "vpn" pero dklab2 dice que le han transferido algo de la vista "external". Respecto a la zona inversa:

```
dklab1 named:
  client 10.168.1.2#57241: view vpn: bad zone transfer request:
  '1.168.192.IN-ADDR.ARPA/IN': non-authoritative zone (NOTAUTH)
```

... es decir, dklab1 dice que en la vista vpn no hay zona 1.168.192.

- Nota: Este mensaje de error no tiene relación: "managed-keys-zone ./IN/external: loading from master file 3c4623849a49a53911c4a3e48d8cead8a1858960bccdea7a1b978d73ec2f06d7.mkeys failed: file not found"

Transferencia manual usando SCP como solución provisional al bug en bind9

No nos queda más remedio que, cada vez que modifiquemos un registro en el master, realizar la transferencia manualmente.

- Ciertamente, nuestra zona casafx.dyndns.org contiene ya todos los registros para los servicios que desplegaremos posteriormente (quizás la llave pública para DKIM, servicio de correo, sea la excepción), sin embargo.

```
scp 10.168.1.1:/etc/bind/db.casafx.dyndns.org-EXT \
    /var/cache/bind/slaves/sec.casafx.dyndns.org-EXT
scp 10.168.1.1:/etc/bind/db.192.168.1 \
    /var/cache/bind/slaves/sec.db.192.168.1
chown bind:bind /var/cache/bind/slaves/*
```

Es norma mantener actualizado (dentro de nuestra versión del SO) el equipo ejecutando periódicamente:

```
apt-get update && apt-get upgrade
```

... ésto aumenta las posibilidades de instalar una versión sin el bug de bind9 mientras llega el lanzamiento de la próxima versión de debian.

Comprobación sintáctica para las zonas casafx.dyndns.org

Aunque es irrelevante doblemente (por el bug y porque la comprobación se hizo ya en el master), podemos siempre lanzar named-checkzone, en la localización para las transferencias:

```
named-checkzone -c IN  casafx.dyndns.org.  \
                /var/cache/bind/slaves/sec.casafx.dyndns.org-VPN
named-checkzone -n ignore -c IN 1.168.10.IN-ADDR.ARPA. \
                /var/cache/bind/slaves/sec.db.10.168.1
```

1.3.8. Tests sobre resoluciones

Conociendo que named escucha por defecto en todas nuestras interfaces de red:

- Preguntamos a 10.168.1.2 y por tanto nuesta IP de origen está en esa red, luego named nos ofrece la vista "vpn":

```
dig    @10.168.1.2 dklab2.casafx.dyndns.org.  any +short
dig -x @10.168.1.2 2.1.168.10.IN-ADDR.ARPA. any +short
```

- Preguntamos con IP de origen 127.0.0.1 y por tanto se nos ofrece la vista "external":

```
dig    @127.0.0.1 dklab2.casafx.dyndns.org.  any +short
dig -x @127.0.0.1 2.1.168.10.IN-ADDR.ARPA. any +short
```

- Si preguntamos por una de las zonas comunes, la información devuelta en ambas vistas es idéntica:

```
dig @10.168.1.2 localhost. any +short
dig @127.0.0.1 localhost. +short
dig -x @10.168.1.2 1.0.0.127.IN-ADDR.ARPA. any +short
dig -x @127.0.0.2 1.0.0.127.IN-ADDR.ARPA. any +short
```

Ping usará las librerías para resolución de nombres del sistema, que según el /etc/resov.conf preguntan a 10.168.1.X y por tanto se le muestra la vista "vpn".

```
PINGLIST="localhost. ns.casafx.dyndns.org. dklab1 dklab2"
for i in ${PINGLIST};
 do cmd="ping -c1 -a $i"; echo $cmd; eval $cmd; sleep 2;
done | grep "1 "
```

1.3.9. Tests sobre DNS Dynamic Updates

Las zonas para las que named ejerce el rol de slave no pueden actualizarse. Contra quien se realizan las actualizaciones es el master, entonces se deja que comience una transferencia entre zonas.

En concreto, el master incrementará el número de serie de la zona en el registro SOA (ésto lo hace en memoria y en los ficheros de transacción .jnl, será entonces cuando paremos bind9 el momento en que tenga oportunidad de actualizar definitivamente los /etc/bind/db.<zonefile>). A continuación se manda una notificación al slave. Por ejemplo si el serial es 2010020660 y realizamos un update con nsupdate, la plantilla y los procedimientos que se comentaron anteriormente, el resultado era la eliminación de un registro para añadirlo modificado, y por tanto esos dos cambios incrementan dos veces el serial: pasaría a ser 2010020662. Veríamos en syslog del master:

```
dklab1 named:
  client 127.0.0.1#19734: updating zone 'casafx.dyndns.org/IN':
  deleting rrset at 'nowhere.casafx.dyndns.org' A
dklab1 named:
  zone casafx.dyndns.org/IN: sending notifies (serial 2010020661)
dklab1 named:
  client 127.0.0.1#19734: updating zone 'casafx.dyndns.org/IN': adding
  an RR at 'nowhere.casafx.dyndns.org' A
dklab1 named:
  zone casafx.dyndns.org/IN: sending notifies (serial 2010020662)
```

Y a continuación la notificación:

```
dklab1 named:
  zone casafx.dyndns.org/IN: sending notifies (serial 2010020662)
```

Otra ocasión en que el master enviará una notificación será al iniciarse, por tanto si hacemos alguna modificación a mano y en frío (práctica, en cualquier caso, que no se recomienda para una zona configurada con actualizaciones dinámicas), deberíamos así también incrementar nosotros el número de serie de la zona para aprovechar esa notificación.

```
dklab1 named: starting BIND 9.7.3 -u bind
...
dklab1 named: running
dklab1 named:
  zone casafx.dyndns.org/IN: sending notifies (serial 2010020662)
```

... dklab2 pareció no hacer nada durante 3 horas en nuestro testbed (desconocemos por qué, a pesar de la notificación que le mandó dklab1) aunque por fin respondió:

```
dklab2 named: zone casafx.dyndns.org/IN: Transfer started.
dklab2 named:
  transfer of 'casafx.dyndns.org/IN' from 10.168.1.1#53: connected
  using 10.168.1.2#53597
dklab1 named:
  client 10.168.1.2#53597: transfer of 'casafx.dyndns.org/IN':IXFR started
dklab2 named:
  zone casafx.dyndns.org/IN: transferred serial 2010020664
dklab2 named: transfer of 'casafx.dyndns.org/IN' from
  10.168.1.1#53: Transfer completed: 1 messages, 8 records, 306 bytes,
  0.090 secs (3400 bytes/sec)
dklab1 named[11669]:
  client 10.168.1.2#53597: transfer of 'casafx.dyndns.org/IN': IXFR ended
dklab2 named:
  zone casafx.dyndns.org/IN: sending notifies (serial 2010020664)
dklab1 named:
  client 10.168.1.2#43571: received notify for zone 'casafx.dyndns.org'
```

1.3.10. Tests sobre la interfaz de control con rndc

Ninguna variación respecto a lo que se hizo en dklab1:

```
rndc -s 10.168.1.2 -k /etc/bind/rndc.key reload
```

```
  ...rndc: connect failed: 10.168.1.2#953: connection refused
```

```
rndc -s 127.0.0.1 -k /etc/bind/rndc.key reload
```

```
  ...server reload successful
```

Puesto que esos flags le pasan los valores por defecto, se puede simplificar la llamada a rndc:

```
rndc reload

rndc status

rndc --help
```

LyX