12.2. /etc/named.conf

El archivo `named.conf` es una colección de declaraciones usando opciones anidadas rodeadas por carácteres de llaves, `{ }`. Los administradores deben tener mucho cuidado cuando estén modificando `named.conf` para evitar errores sintácticos puesto que hasta el error más pequeño puede impedir que el servicio `named` arranque.

> **Aviso**
>
> *No* modifique manualmente el archivo `/etc/named.conf` o cualquier archivo en el directorio `/var/named/` si está usando la **Herramienta de configuración del Servicio de Nombres de Dominio**. Cualquier cambio manual a esos archivos serán sobreescritos la próxima vez que se use **Herramienta de configuración del Servicio de Nombres de Dominio**.

Un archivo típico de `named.conf` está organizado de forma similar al ejemplo siguiente:

```
<statement-1> ["<statement-1-name>"] [<statement-1-class>] {
   <option-1>;
   <option-2>;
   <option-N>;
};

<statement-2> ["<statement-2-name>"] [<statement-2-class>] {
   <option-1>;
   <option-2>;
   <option-N>;
};

<statement-N> ["<statement-N-name>"] [<statement-N-class>] {
   <option-1>;
   <option-2>;
   <option-N>;
};
```

12.2.1. Tipos de declaraciones comúnes

Los siguientes tipos de sentencias son usados a menudo en `/etc/named.conf`:

12.2.1.1. Declaración `acl`

La sentencia `acl` (o sentencia de control de acceso) define grupos de hosts a los que se les puede permitir o negar el acceso al servidor de nombres.

Una declaración `acl` tiene la siguiente forma:

```
acl <acl-name> {
    <match-element>;
    [<match-element>; ...]
};
```

En esta declaración, sustituya *<acl-name>* con el nombre de la lista de control de acceso y reemplace *<match-element>* con una lista de direcciones IP separada por puntos y comas. La mayoría de las veces, una dirección IP individual o notación de red IP (tal como `10.0.1.0/24`) es usada para identificar las direcciones IP dentro de la declaración `acl`.

La siguiente lista de control de acceso ya están definidas como palabras claves para simplificar la configuración:

- `any` — Hace coincidir todas las direcciones IP.
- `localhost` — Hace coincidir cualquier dirección IP que se use el sistema local.
- `localnets` — Hace coincidir cualquier dirección IP en cualquier red en la que el sistema local está conectado.
- `none` — No concuerda ninguna dirección IP.

Cuando lo utilice con otras pautas (tales como declaraciones `options`), las declaraciones `acl` pueden ser muy útiles al

asegurar el uso correcto de su servidor de nombres BIND.

El ejemplo siguiente define dos listas de control de acceso y utiliza una declaración `options` para definir cómo son tratadas en el servidor de nombres:

```
 acl black-hats {
    10.0.2.0/24;
    192.168.0.0/24;
 };

acl red-hats {
    10.0.1.0/24;
 };

options {
    blackhole { black-hats; };
    allow-query { red-hats; };
    allow-recursion { red-hats; };
 }
```

Este ejemplo contiene dos listas de control de acceso, `black-hats` y `red-hats`. Los hosts en la lista `black-hats` se les niega el acceso al servidor de nombres, mientras que a los hosts en la lista `red-hats` se les dá acceso normal.

12.2.1.2. Declaración `include`

La declaración `include` permite incluir archivos en un `named.conf`. De esta forma los datos de configuración confidenciales (tales como `llaves`) se pueden colocar en un archivo separado con permisos restringidos.

Una declaración `include` tiene la forma siguiente:

```
include  "<file-name>"
```

En esta declaración, *<file-name>* es reemplazado con una ruta absoluta a un archivo.

12.2.1.3. Declaración `options`

La declaracion `options` define opciones de configuración de servidor globales y configura otras declaraciones por defecto. Puede ser usado para especificar la ubicación del directorio de trabajo `named`, los tipos de consulta permitidos y mucho más.

La declaración `options` toma la forma siguiente:

```
options {
        <option>;
        [<option>; ...]
};
```

En esta declaración, las directivas *<option>* son reemplazadas con una opción válida.

Las siguientes son opciones usadas a menudo:

- `allow-query` — Especifica cuáles hosts tienen permitido consultar este servidor de nombres. Por defecto, todos los hosts tienen derecho a consultar. Una lista de control de acceso, o una colección de direcciones IP o redes se puede usar aquí para sólo permitir a hosts particulares hacer consultas al servidor de nombres.
- `allow-recursion` — Parecida a la opción `allow-query`, salvo que se aplica a las peticiones recursivas. Por defecto, todos los hosts están autorizados a presentar peticiones recursivas en un servidor de nombres.
- `blackhole` — Especifica cuáles hosts no tienen permitido consultar al servidor de nombres.
- `directory` — Especifica el directorio de trabajo `named` si es diferente del valor predeterminado `/var/named`.
- `forward` — Especifica el comportamiento de reenvío de una directiva `forwarders`.

  Se aceptan las siguientes opciones:

  - `first` — Indica que los servidores de nombres especificados en la directiva `forwarders` sean consultados antes de que `named` intente resolver el nombre el mismo.
  - `only` — Especifica que `named` no intente la resolución de nombres él mismo en el evento de que fallen las

consultas a los servidores de nombres especificados en la directriz `forwarders`.

- `forwarders` — Especifica una lista de direcciones IP válidas para los servidores de nombres donde las peticiones se pueden reenviar para ser resueltas.
- `listen-on` — Especifica la interfaz de red en la cual `named` escucha por solicitudes. Por defecto, todas las interfaces son usadas.

  Usando esta directiva en un servidor DNS que también actúa como una gateway, BIND se puede configurar para sólo contestar solicitudes que se originan desde algunas de las redes.

  Una directiva `listen-on` se parece al ejemplo siguiente:

```
options {
   listen-on { 10.0.1.1; };
};
```

  En este ejemplo, solamente son aceptadas las peticiones que llegan desde la interfaz de red sirviendo a la red privada (`10.0.1.1`).
- `notify` — Controla si `named` notifica a los servidores esclavos cuando una zona es actualizada. Acepta las opciones siguientes:
  - `yes` — Notifica a los servidores esclavos.
  - `no` — No notifica a los servidores esclavos.
  - `explicit` — Solamente notifica a los servidores esclavos especificados en una lista de `also-notify` dentro de la declaración de una zona.
- `pid-file` — Especifica la ubicación del archivo del proceso ID creado por `named`.
- `root-delegation-only` — Activa la implementación de las propiedades de delegación en dominios de nivel superior (TLDs) y zonas raíz con una lista opcional de exclusión. La *delegación* es el proceso de separar una zona sencilla en múltiples zonas. Para poder crear una zona delegada, se utilizan elementos conocidos como *registros NS*. Los registros de servidor de nombres (registros de delegación) anuncian los servidores de nombres autorizados para una zona particular.

  El siguiente ejemplo de `root-delegation-only` especifica una lista excluyente de los TDLs desde los que se esperan respuestas no delegadas:

```
options {
   root-delegation-only exclude { "ad"; "ar"; "biz"; "cr"; "cu"; "de"; "dm"; "id;
                                  "lu"; "lv"; "md"; "ms"; "museum"; "name"; "no"; "pa";
                                  "pf"; "se"; "sr"; "to"; "tw"; "us"; "uy"; };
};
```

- `statistics-file` — Permite especificar la localización alternativa de los archivos de estadísticas. Por defecto, las estadísticas de `named` son guardadas al archivo `/var/named/named.stats`.

Existen numerosas opciones disponibles, muchas de ellas dependen unas de otras para poder funcionar correctamente. Consulte el ***Manual de referencia para el administrador de BIND 9*** en Sección 12.7.1 y la página man para `bind.conf` para más detalles.

12.2.1.4. Declaración `zone`

Una declaración `zone` define las características de una zona tal como la ubicación de su archivo de configuración y opciones específicas de la zona. Esta declaración puede ser usada para ignorar las declaraciones globales `options`.

Una declaración `zone` tiene la forma siguiente:

```
zone <zone-name> <zone-class> {
     <zone-options>;
     [<zone-options>; ...]
};
```

En esta declaración, *<zone-name>* es el nombre de la zona, *<zone-class>* es la clase opcional de la zona, y *<zone-options>* es una lista de opciones que caracterizan la zona.

El atributo *<zone-name>* para la declaración de zona es particularmente importante, pues es el valor por defecto asignado

para la directriz `$ORIGIN` usada dentro del archivo de zona correspondiente localizado en el directorio `/var/named/`. El demonio `named` anexa el nombre de la zona a cualquier nombre de dominio que no esté completamente cualificado listado en el archivo de zona.

Por ejemplo, si una declaración `zone` define el espacio de nombres para `example.com`, utilice `example.com` como el *<zone-name>* para que sea colocado al final de los nombres de hosts dentro del archivo de zona `example.com`.

Para más información sobre los archivos de zona, consulte Sección 12.3.

Las opciones más comunes para la declaración `zone` incluyen lo siguiente:

- `allow-query` — Especifica los clientes que se autorizan para pedir información sobre una zona. Por defecto, todas las peticiones de información son autorizadas.
- `allow-transfer` — Especifica los servidores esclavos que están autorizados para pedir una transferencia de información de la zona. Por defecto, todas las peticiones se autorizan.
- `allow-update` — Especifica los hosts que están autorizados para actualizar dinámicamente la información en sus zonas. Por defecto, no se autoriza la actualización de la información dinámicamente.

  Tenga cuidado cuando autorice a los hosts para actualizar la información de su zona. No habilite esta opción si no tiene confianza en el host que vaya a usar. Es mejor que el administrador actualice manualmente los registros de zona y que vuelva a cargar el servicio `named`.
- `file` — Especifica el nombre del archivo en el directorio de trabajo `named` que contiene los datos de configuración de zona.
- `masters` — Especifica las direcciones IP desde las cuales solicitar información autorizada. Solamente se usa si la zona está definida como `type slave`.
- `notify` — Controla si `named` notifica a los servidores esclavos cuando una zona es actualizada. Esta directiva sólo acepta las opciones siguientes:
  - `yes` — Notifica a los servidores esclavos.
  - `no` — No notifica a los servidores esclavos.
  - `explicit` — Solamente notifica a los servidores esclavos especificados en una lista de `also-notify` dentro de la declaración de una zona.
- `type` — Define el tipo de zona.

  Abajo se muestra una lista de las opciones válidas:
  - `delegation-only` — Refuerza el estado de delegación de las zonas de infraestructura tales como COM, NET u ORG. Cualquier respuesta recibida sin una delegación explícita o implícita es tratada como `NXDOMAIN`. Esta opción solamente es aplicable en TLDs o en archivos raíz de zona en implementaciones recursivas o de caché.
  - `forward` — Dice al servidor de nombres que lleve a cabo todas las peticiones de información de la zona en cuestión hacia otros servidores de nombres.
  - `hint` — Tipo especial de zona que se usa para orientar hacia los servidores de nombres root que sirven para resolver peticiones de una zona que no se conoce. No se requiere mayor configuración que la establecida por defecto con una zona `hint`.
  - `master` — Designa el servidor de nombres actual como el que tiene la autoridad para esa zona. Una zona se puede configurar como tipo `master` si los archivos de configuración de la zona residen en el sistema.
  - `slave` — Designa el servidor de nombres como un servidor esclavo para esa zona. También especifica la dirección IP del servidor de nombres maestro para la zona.
- `zone-statistics` — Configura `named` para mantener estadísticas concerniente a esta zona, escribiéndola a su ubicación por defecto (`/var/named/named.stats`) o al archivo listado en la opción `statistics-file` en la declaración `server`. Consulte la Sección 12.2.2 para más información sobre la declaración `server`.

12.2.1.5. Ejemplo de declaraciones de `zone`

La mayoría de los cambios al archivo `/etc/named.conf` de un servidor de nombres maestro o esclavo envuelven agregar, modificar o borrar declaraciones `zone`. Mientras que estas declaraciones `zone` pueden contener muchas opciones, la mayoría de los servidores de nombres requieren sólo un pequeño subconjunto para funcionar efectivamente. Las siguientes declaraciones `zone` son ejemplos muy básicos que ilustran la relación de servidores de nombres maestro-esclavo.

A continuación se muestra un ejemplo de una declaración de `zone` para un servidor de nombres primario hospedando `example.com` (`192.168.0.1`):

```
zone "example.com" IN {
  type master;
  file "example.com.zone";
  allow-update { none; };
};
```

En la declaración, la zona es identificada como `example.com`, el tipo es configurado a `master` y el servicio `named` se instruye para leer el archivo `/var/named/example.com.zone`. También le dice a `named` que no permita a ningún otro host que realice actualizaciones.

Una declaración `zone` de servidor esclavo para `example.com` se ve un poco diferente comparado con el ejemplo anterior. Para un servidor esclavo, el tipo se coloca a `slave` y en lugar de la línea `allow-update` está una directiva diciéndole a `named` la dirección IP del servidor maestro.

A continuación se muestra un ejemplo de una declaración `zone` para un servidor esclavo para la zona `example.com`:

```
zone "example.com" {
  type slave;
  file "example.com.zone";
  masters { 192.168.0.1; };
};
```

Esta declaración `zone` configura `named` en el servidor esclavo a que busque por el servidor maestro en la dirección IP `192.168.0.1` por información sobre la zona `example.com`. La información que el servidor esclavo recibe desde el servidor maestro es guardada al archivo `/var/named/example.com.zone`.

12.2.2. Otros tipos de declaraciones

La lista siguiente muestra tipos de declaraciones usadas con menos frecuencia disponibles dentro de `named.conf`:

- `controls` — Configura varios requerimientos de seguridad necesarios para usar el comando `rndc` para administrar el servicio `named`.

  Consulte la Sección 12.4.1 para conocer más sobre la estructura de la declaración `controls` y de las opciones que están disponibles.

- `key "`*`<key-name>`*`"` — Define una llave particular por nombre. Las claves son usadas para autenticar varias acciones, tales como actualizaciones seguras o el uso del comando `rndc`. Se usan dos opciones con `key`:

  - `algorithm` *`<algorithm-name>`* — El tipo de algoritma usado, tal como `dsa` o `hmac-md5`.
  - `secret "`*`<key-value>`*`"` — La clave encriptada.

  Consulte la Sección 12.4.2 para instrucciones sobre cómo escribir una declaración `key`.

- `logging` — Permite el uso de múltiples tipos de registro, llamados *channels*. Usando la opción `channel` dentro de la declaración `logging`, se puede construir un tipo registro personalizado, con su propio nombre de archivo (`file`), tamaño límite (`size`), versión (`version`), y nivel de importancia (`severity`). Una vez que se haya definido el canal personalizado, se usa una opción `category` para clasificar el canal y comenzar a conectar cuando se reinicie `named`.

  Por defecto, `named` registra mensajes estándar al demonio `syslog`, que les sitúa en `/var/log/messages`. Esto se debe a que varios canales estándares se encuentran incorporados a BIND junto con varios niveles de severidad, tales como uno que maneja la información de mensajes de registros (`default_syslog`) y otro que maneja mensajes de depuración (`default_debug`). Una categoría por defecto, llamada `default`, usa los canales incorporados para hacer conexiones normales sin ninguna configuración especial.

  La personalización del proceso de conexión es un proceso con muchos detalles y que está más allá del objetivo de este capítulo. Para información sobre la creación de registros personalizados BIND, consulte el ***Manual de referencia del administrador de BIND 9*** mencionado en la Sección 12.7.1.

- `server` — Define opciones particulares que afectan como `named` debería actuar con respecto a servidores de nombres remotos, especialmente en lo que respecta a las notificaciones y transferencias de zonas.

  La opción `transfer-format` controla si un registro de recursos es enviado con cada mensaje (`one-answer`) o si registros de múltiples recursos son enviados con cada mensaje (`many-answers`). Mientras que `many-answers` es más eficiente, sólo los nuevos servidores de nombres BIND lo entienden.

- `trusted-keys` — Contiene llaves públicas utilizadas por DNS seguro, DNSSEC. Para mayor información sobre la seguridad de BIND, consulte la Sección 12.5.3.

- `view "<view-name>"` — Crea vistas especiales dependiendo de en qué red esté el host que contacta el servidor de nombres. Esto permite a determinados hosts recibir una respuesta que se refiere a una zona particular mientras que otros hosts reciben información completamente diferente. Alternativamente, algunas zonas puede que sólo estén disponibles para ciertos hosts de confianza mientras que otros hosts menos autorizados, sólo pueden hacer peticiones a otras zonas.

  Se pueden usar múltiples visualizaciones, siempre y cuando sus nombres sean únicos. La opción `match-clients` especifica las direcciones IP que aplican a una vista particular. Cualquier declaración de `options` puede también ser usada dentro de una vista, ignorando las opciones globales ya configuradas por `named`. La mayoría de las sentencias `view` contienen múltiples declaraciones `zone` que aplican a la lista `match-clients`. El orden en que las sentencias `view` son listadas es importante, pues la primera sentencia `view` que coincida con una dirección IP de cliente particular es usada.

  Consulte la Sección 12.5.2 para más información sobre la declaración `view`.

12.2.3. Etiquetas de comentarios

La siguiente es una lista de las etiquetas de comentarios válidas usadas dentro de `named.conf`:

- `//` — Cuando se coloca al comienzo de una línea, esa línea es ignorada por `named`.
- `#` — Cuando se coloca al comienzo de una línea, esa línea es ignorada por `named`.
- `/*` y `*/` — Cuando el texto se coloca entre estas etiquetas, se ignora el bloque de texto por `named`.