Seguridad de un servidor DNS

Un servidor DNS es susceptible de recibir solicitudes desde las redes en las que participa, solicitudes que pueden venir de clientes DNS o de otros servidores DNS. Por lo tanto, puede constituir un punto de entrada hacia la organización de la que depende y ser utilizado para recuperar información sensible (direcciones y nombres de máquinas o de servicios de red).

Por otro lado, en la hipótesis de un fallo de seguridad en el programa ejecutable de BIND, un servidor DNS podría ser utilizado para tomar el control del sistema en el que se ejecuta, o para proporcionar información sensible (cuentas de usuarios, etc.).

Existen diferentes técnicas que permiten proteger un servidor BIND:

- Limitar los clientes y servidores DNS autorizados a comunicar con él.
- Repartir los datos nombres/direcciones entre distintos servidores.
- Restringir las posibilidades de lectura-escritura del servidor a una rama de la arborescencia del sistema de archivos global (*chroot jail*).

1. Control de los clientes autorizados

Un servidor BIND puede estar configurado para tratar solamente las solicitudes de los hosts autorizados y/o rechazar las solicitudes de algunos hosts. La selección puede hacerse especificando direcciones IP o identificadores de red/subred.

a. La opción allow-query

La opción `allow-query`, en la directiva `options` del archivo de configuración `named.conf`, permite especificar los clientes DNS autorizados a interrogar al servidor BIND.

Sintaxis

```
allow-query {Lista};
```

La lista está compuesta por elementos delimitados con un carácter '`;`'. Un elemento puede ser:

- una dirección IP, IPv4 o IPv6,
- un identificador de red, IPv4 o IPv6, en notación CIDR,
- una palabra clave predefinida: `none` (ninguno), `any` (todos), `localhost` (máquina local) o `localnets` (redes de la máquina local),
- el identificador de una lista definida anteriormente (`acl`),
- uno de los elementos anteriores, precedido por el carácter `!` para excluirlo de la lista autorizada.

Para simplificar la configuración, podemos definir en el archivo de configuración `named.conf` listas de direcciones, usando la directiva `acl` (*Access Control List*):

```
acl NombreLista {lista};
```

A continuación podemos usar la lista en las otras directivas, designándola por su nombre de `acl`.

La opción `allow-recursion` funciona igual, pero para controlar los hosts autorizados a efectuar solicitudes de resolución recursiva, es decir relativas a los dominios en los que el servidor DNS no ejerce ninguna autoridad.

Ejemplo

El servidor principal, `192.168.0.60`, autoriza a todos los hosts de las redes en las que participa a interrogarlo sobre nombres y direcciones de sus zonas, pero solamente la máquina local puede solicitar una resolución hacia otros dominios.

Directiva `options` del archivo de configuración `named.conf`:

```
options {
    listen-on port 53 {127.0.0.1;192.168.0.60;};
    directory      "/var/named";
    dump-file      "/var/named/data/cache_dump.db";
    statistics-file "/var/named/data/named_stats.txt";
    memstatistics-file "/var/named/data/named_mem_stats.txt";
    allow-query {localnets;};
    allow-recursion {localhost;};
    recursion yes;
    pid-file "/run/named/named.pid";
};
```

El host `192.168.0.70` solicita resoluciones de nombres al servidor DNS que acabamos de presentar:

```
host www.google.com
Host www.google.com not found: 5(REFUSED)
host station.lpic2test.com
station.lpic2test.com has address 192.168.0.24
```

La primera solicitud ha sido rechazada porque implica una búsqueda recursiva, la segunda funciona porque es relativa al dominio local.

Desde la máquina del servidor DNS, las dos solicitudes de resolución son aceptadas:

```
host www.google.com
www.google.com has address 172.217.22.132
www.google.com has IPv6 address 2a00:1450:4007:815::2004
host station.lpic2test.com
station.lpic2test.com has address 192.168.0.24
```

b. La opción blackhole

La opción `blackhole`, en la directiva `options` del archivo de configuración `named.conf`, permite especificar los hosts que no están autorizados a interrogar al servidor BIND.

El servidor no responde nada a las solicitudes de los hosts de la lista `blackhole`, como si no estuviera disponible.

Sintaxis

```
blackhole {Lista};
```

Ejemplo

Directiva `options` del archivo de configuración `named.conf`:

```
options {
    listen-on port 53 {127.0.0.1;192.168.0.60;};
    directory       "/var/named";
    dump-file       "/var/named/data/cache_dump.db";
    statistics-file "/var/named/data/named_stats.txt";
    memstatistics-file "/var/named/data/named_mem_stats.txt";
    allow-query {localnets;};
    allow-recursion {localnets;};
    blackhole {192.168.0.70;};
    recursion yes;
    pid-file "/run/named/named.pid";
};
```

El host `192.168.0.70` solicita resoluciones de nombres al servidor DNS mostrado anteriormente:

```
host www.google.com
;; connection timed out; no servers could be reached
host station.lpic2test.com
;; connection timed out; no servers could be reached
```

El servidor no responde a las solicitudes.

2. Uso de una cuenta de servicio

Como todos los servidores de red, BIND es potencialmente vulnerable. La existencia de un fallo de seguridad en el ejecutable podría permitir a un cliente de red modificar el uso normal del servicio con el objetivo de acceder a información administrada por el sistema. Se desaconseja fuertemente asociar la cuenta del superusuario (UID 0) al daemon `named`.

En las versiones recientes del paquete de software `BIND`, el inicio del daemon está configurado para que se ejecute con una cuenta de usuario específica, creada durante la instalación del paquete. Esta cuenta está especificada durante el lanzamiento del ejecutable con la opción `-u`. Esta cuenta posee derechos suficientes para asegurar el funcionamiento correcto de BIND, pero no tiene derechos de acceso a los archivos sensibles del sistema operativo.

Ejemplo

Daemon `named` en un servidor CentOS 8:

```
ps -ef | grep named
named    11911    1  0 07:21?        00:00:01 /usr/sbin/named -u named -c
/etc/named.conf
```

El daemon está asociado a la cuenta de usuario `named`.

```
grep named /etc/passwd
named:x:25:25:Named:/var/named:/bin/false
grep named /etc/group
named:x:25:
```

La cuenta `named` no puede servir para conectarse en el servidor (ultimo campo de `/etc/passwd` con el valor `false`), forma parte del grupo específico `named`.

Daemon `named` en un servidor Debian 10:

```
ps -ef | grep named
bind     1733    1  0 mayo04?       00:00:01 /usr/sbin/named -u bind
```

El daemon está asociado a la cuenta de usuario `bind`.

```
grep bind /etc/passwd
bind:x:120:128::/var/cache/bind:/usr/sbin/nologin
grep bind /etc/group
bind:x:128:
```

La cuenta `bind` no puede servir para conectarse en el servidor (ultimo campo de `/etc/passwd` con el valor `/usr/sbin/nologin`), forma parte del grupo específico `bind`.

El uso de una cuenta de servicio asociada al daemon limita sus posibilidades de acción en el sistema. No obstante, incluso un usuario que no sea superusuario puede ofrecer información importante en caso de que el servidor haya sido comprometido. Por ejemplo, todos los usuarios tienen derecho de lectura en el archivo de declaración de las cuentas de usuarios locales, `/etc/passwd` :

```
ls -l /etc/passwd
-rw-r--r--. 1 root root 2548  8 mayo   22:56 /etc/passwd
```

3. BIND en modo chroot jail

Para limitar los accesos posibles del proceso que ejecuta `named` en la arborescencia del sistema de archivos global, podemos encerrarlo (*jail*) en una parte de esta arborescencia. El Daemon creerá que el directorio que se le asignará será la raíz del sistema de archivos, y no podrá salir de él.

Este método, llamado chroot jail, se utiliza frecuentemente en los servidores de redes. Asociado al uso de una cuenta de servicio de no superusuario, refuerza la seguridad limitando las consecuencias de un posible fallo de seguridad en el servicio de red.

a. Creación del entorno necesario

En modo *chroot jail*, el daemon `named` ve su directorio "cárcel" como directorio raíz. Por lo tanto, tiene que poder encontrar a partir de este directorio todos los directorios y archivos que necesita.

Por ello es necesario crear en el directorio "cárcel" todos los directorios necesarios (`etc`, `var/named` o `var/cache/bind`, `dev`, etc.) y copiar o crear los archivos utilizados por el daemon.

El paquete `bind-chroot` instala un servidor BIND en modo `chroot jail`. No obstante, es importante conocer bien las etapas de configuración de un servidor en ese modo, por esto es tan necesario para la certificación LPIC-2.

b. Creación del entorno chroot jail

Las etapas de configuración de un daemon `named` en modo *chroot jail* son las siguientes (los elementos que se utilizarán pueden variar en función de la distribución utilizada, los ejemplos que veremos a continuación se aplican a una distribución de tipo Red Hat):

- Crear el directorio "cárcel" y la arborescencia necesaria en ese directorio

```
mkdir -p /jaildns/etc/named /jaildns/dev /jaildns/var/named
/jaildns/var/run/named
```

El directorio `dev` contendrá dos archivos especiales usados por `named : null` y `random`.

El directorio `var/run/named` contiene archivos que identifican el proceso `named` activo.

El directorio `etc` contiene el archivo de configuración `named.conf` y el director de configuración `named`.

El directorio `var/named` contiene los archivos de datos y de registro.

- Copiar los archivos necesarios en los diferentes subdirectorios del directorio "cárcel":

```
cp /etc/named.* /jaildns/etc/
cp -r /var/named /jaildns/var
```

- Crear los archivos especiales que necesita `named`:

```
mknod /jaildns/dev/null c 1 3
mknod /jaildns/dev/random c 1 8
chmod 666 /jaildns/dev/*
```

- Modificar el propietario y el grupo para toda la arborescencia del directorio "cárcel":

```
chown -R named:named /jaildns
```

- Configurar la ejecución del daemon `named` en modo *chroot jail*, con la opción `-t DirCÆrcel` (o configurar el inicio en *chroot jail* usando `systemd`):

```
vi /etc/sysconfig/named
OPTIONS="-t /jaildns"
```

c. Ejecución del programa en modo chroot jail

Se puede comprobar manualmente el correcto arranque del daemon `named` en este modo, con el comando siguiente:

```
named -c ArchivoConfig -u usuario -t DirCárcel
```

Ejemplo

```
named -c /etc/named.conf -u named -t /jaildns
```

Si el daemon no se inicia correctamente, hay que consultar los archivos de registro del sistema. Cuidado porque SELinux puede dar problemas (se puede desactivar temporalmente para hacer un diagnóstico, con el comando `setenforce 0`).

4. Servidores DNS fraccionados (split DNS)

La mayoría de las organizaciones poseen una parte de su red accesible desde el exterior y otra reservada a un uso interno. Los servicios de redes accesibles desde el exterior (servidores HTTP, FTP…) tienen que estar referenciados en una base de nombres de un servidor DNS accesible desde el exterior (y desde el interior de la organización), por lo tanto, potencialmente amenazado. Los hosts y los servicios de redes reservados a un uso interno tienen que estar referenciados en una base de nombres de un servidor DNS, pero este último no debería ser accesible desde el exterior.

Para limitar los riesgos de intrusión o robo de datos, es habitual fraccionar el servicio DNS en distintos servidores DNS:

- Servidor DNS accesible desde el exterior: este servidor solo tiene en su archivo de zona los nombres de los hosts y de los servicios accesibles desde el exterior.

De esta manera, si fuera atacado por la red, no podría comunicar información relativa a la red interna de la organización.

- Servidor DNS interno: este servidor contiene en su archivo de zona todos los nombres de hosts y de servicios, internos o accesibles desde el exterior. Solamente aceptará las solicitudes de resolución que lleguen desde la red interna. Cuando reciba una solicitud de resolución relativa a un nombre o dirección que no se encuentre en su zona, la transferirá al servidor DNS externo (*forwarding*) para que este último efectúe una búsqueda recursiva.
- Clientes DNS internos: estos clientes están configurados para usar solamente el servidor DNS interno.

La mayoría de las veces, los servidores internos y externos están duplicados; un servidor principal y un servidor secundario, para la tolerancia a fallos.

5. Intercambio seguro entre servidores DNS

Los servidores DNS intercambian información entre ellos, esto puede constituir un riesgo de seguridad. Un sistema en manos de personas malintencionadas podría recuperar archivos de zona haciéndose pasar por un servidor secundario o, al contrario, comunicar información falsa a un servidor DNS, en particular a su caché (*cache poisoning*), usando técnicas de usurpación de rol (*DNS spoofing*).

Existen diferentes soluciones para reforzar la seguridad de los intercambios entre servidores DNS.

6. Control de las transferencias de zona

Un servidor DNS principal puede ser configurado para autorizar solamente las transferencias de zona a ciertos hosts. Para ello, hay que usar la opción `allow-`

`transfer`, bien en la directiva `options`, bien en la directiva de definición de la zona, en el archivo `named.conf`.

La opción `allow-transfer` permite especificar los solicitantes autorizados a recibir una copia del archivo de zona del servidor BIND principal de la zona.

Sintaxis

```
allow-transfer {Lista};
```

La lista sigue las mismas reglas que las que se vieron anteriormente en la opción `allow-query`.

Ejemplo

El servidor principal, `192.168.0.60`, solamente autoriza la demanda de una transferencia de sus zonas al servidor secundario `192.168.0.70`:

Directiva `options` del archivo de configuración `named.conf`:

```
options {
    listen-on port 53 {127.0.0.1;192.168.0.60;};
    directory      "/var/named";
    dump-file      "/var/named/data/cache_dump.db";
    statistics-file "/var/named/data/named_stats.txt";
    memstatistics-file "/var/named/data/named_mem_stats.txt";
    allow-transfer {192.168.0.70;};
    allow-query {localnets;};
    allow-recursion {localhost;};
    recursion yes;
    pid-file "/run/named/named.pid";
};
```

7. Transacciones seguras TSIG

Las transacciones seguras (TSIG, *Transaction SIGnature*) permiten garantizar la identidad de los servidores DNS que se intercambian información de zona.

Los servidores comparten la clave pública de un par de claves, clave privada y pública, y tienen que suministrarla para poder hacer intercambios de información entre ellos.

También se puede usar este método para generar archivos de zona firmados digitalmente, que garanticen su origen y que sean utilizables solamente por servidores que dispongan de la clave pública.

Para que el intercambio de la clave funcione correctamente, las horas de los servidores tienen que estar sincronizadas.

a. Generación de claves de transacción de hosts

El par de claves se puede generar usando el comando `dnssec-keygen` .

Sintaxis habitual

```
dnssec-keygen -a HMAC-MD5 -b TamañoClave -n TipoPro NombreClave
```

Donde:

`-a HMAC-MD5`	Método de cifrado. `HMAC-MD5` es el método habitual para TSIG.
`-b TamaæoClave`	Tamaño de la clave, entre 1 y 512. El valor habitual es `128`.
`-n TipoPro`	Tipo de propietario. `HOST` para un par TSIG de servidores.
`NombreClave`	Nombre de la clave. Cadena de caracteres libre para TSIG.

Este comando se utiliza en el marco más general de DNSSEC. Los parámetros presentados aquí son los de una clave TSIG entre servidores DNS.

El comando crea, en el directorio actual, dos archivos para el par de claves:

KNombreClave.+xxx.yyyyy.key
KNombreClave.+xxx.yyyyy.private

La clave pública se encuentra en el primer archivo, la clave privada está en el segundo.

La clave pública será la que deberán proporcionar los servidores secundarios al servidor principal para obtener una transferencia de zona.

La clave privada puede ser utilizada para generar un archivo de zona firmado digitalmente, usando el comando `dnssec-signzone` .

Ejemplo

Generación de un par de claves para las transacciones TSIG entre el servidor principal y los secundarios:

```
dnssec-keygen -a HMAC-MD5 -b 128 -n HOST clave-lpic2
Kclave-lpic2.+157+21773
ls K*
Kclave-lpic2.+157+21773.key  Kclave-lpic2.+157+21773.private
cat Kclave-lpic2.+157+21773.key
clave-lpic2. IN KEY 512 3 157 hFuO6Jbggww6cgWtD8jSVw==
cat Kclave-lpic2.+157+21773.private
Private-key-format: v1.3
Algorithm: 157 (HMAC_MD5)
Key: hFuO6Jbggww6cgWtD8jSVw==
Bits: AAA=
Created: 20200508171257
Publish: 20200508171257
```

```
Activate: 20200508171257
```

b. Configuración de TSIG en named.conf

En el servidor principal y en los servidores secundarios seguros, se tiene que declarar la clave pública en el archivo `named.conf`, así como las características de comunicación.

Declaración de la clave

La directiva `key` permite declarar la clave pública, según la sintaxis siguiente:

```
key NombreClave {
  algorithm hmac-md5;
  secret "ClavePública";
};
```

`NombreClave` es el parámetro especificado durante la creación del par de claves. La clave pública "`ClavePœblica`" tiene que ser copiada a partir del contenido del archivo `.key` generado.

Ejemplo

En el servidor principal y en el servidor secundario, en el archivo `named.conf`:

```
key clave-lpic2 {
  algorithm hmac-md5;
  secret "hFuO6Jbggww6cgWtD8jSVw==";
};
```

En el servidor principal, hay que denegar las transferencias de zonas a los que no tienen la clave, usando la opción `allow-transfer`. Se puede hacer también en el servidor (usando la directiva `options`), o solamente para algunas zonas (en la directiva `zone`):

```
allow-transfer { key NombreClave; };
```

En el o los servidores secundarios, hay que indicar que tiene que proporcionarse la clave en los intercambios con el servidor principal. Para ello se utiliza una directiva `server`:

```
server DirIP {
  keys { NombreClave; };
};
```

`DirIP` corresponde a la dirección IP del servidor principal.

Ejemplo

En el servidor principal CentOS 8:

```
vi /etc/named.conf
key clave-lpic2 {
algorithm hmac-md5;
  secret "hFuO6Jbggww6cgWtD8jSVw==";
};
options {
      listen-on port 53 {127.0.0.1;192.168.0.60;};
      directory       "/var/named";
      dump-file       "/var/named/data/cache_dump.db";
      statistics-file "/var/named/data/named_stats.txt";
      memstatistics-file "/var/named/data/named_mem_stats.txt";
      allow-query {localnets;};
      allow-recursion {localnets;};
      allow-transfer {key clave-lpic2;};
      recursion yes;
      pid-file "/run/named/named.pid";
};
[...]
```

En el servidor secundario Debian 10:

```
vi /etc/bind/named.conf.local
key clave-lpic2 {
  algorithm hmac-md5;
```

```
  secret "hFuO6Jbggww6cgWtD8jSVw==";
};
server 192.168.0.60 {
  keys { clave-lpic2; };
};
[...]
```

Después hay que recargar las configuraciones de `named` en los dos servidores. También hay que asegurarse de que la hora esté sincronizada en los servidores.

8. DNSSEC

DNSSEC (*Domain Name System Security Extensions*) es un protocolo que tiene como objetivo reforzar la seguridad de los servidores DNS y de sus intercambios. Está definido en las RFC 4033, 4034 y 4035.

DNSSEC firma digitalmente todos los datos de los archivos de zona y las respuestas a las solicitudes de resolución. El solicitante puede de esta manera asegurarse de que los datos recibidos son auténticos, incluyendo el caso en el que los datos vienen del caché de un servidor DNS. Para ello, el cliente tiene que disponer de la clave pública asociada a la que ha servido para generar la firma.

DNSSEC también permite la transferencia de los archivos de zona firmados digitalmente, los cuales solo pueden ser leídos por los servidores DNS que tengan la clave pública. Para ello es necesario crear una clave de tipo `ZONE`, con el comando `dnssec-keygen` , y generar después el archivo de zona firmado digitalmente, con el comando `dnssec-signzone` .

9. DANE

DANE (*DNS-based Authentication of Named Entities*) es un nuevo protocolo de protección de los intercambios efectuados usando el protocolo seguro TLS (*Transport Layer Security*). TLS usa certificados digitales X.509 para autenticar los diferentes participantes. Estos

certificados son ofrecidos por autoridades de certificación (CA, *Certification Authorities*), pero en algunos casos pueden ser usurpados.

DANE permite almacenar y firmar los certificados digitales X.509 en los dominios DNS (con registros de tipo TLSA) y proporcionarlos a los clientes y a los servidores del dominio, a condición de que usen DNSSEC.