ASIR

Implantación de Sistemas Operativos

EDUCAMADRID

CONSEJERÍA DE EDUCACIÓN
Comunidad de Madrid

MAX V.6.0

Versión inicial: 1.0. Utiliza la versión 6.0 de MaX.

Versión 1.1. Utiliza la versión 6.0 de MaX.

Mejoras más significativas:

- Inclusión del apartado ...

Mejoras programadas para las próximas revisiones (se admiten colaboraciones):

- Actualización de

(Sugerencias a sagsag@hotmail.es)

Índice de contenidos

1 INTRODUCCIÓN

Los temas tratados en este libro son un suplemento del relativo al Módulo "Implantación de sistemas operativos" cuyo objetivo es completar con un nivel elevado las posibilidades de MaX.

2 ADMINISTRACIÓN REMOTA

2.1 OpenSSH

2.1.1 Introducción

OpenSSH es un conjunto de herramientas que permite el control remoto de equipos conectados en red y la transferencia de datos entre ellos. OpenSSH es una versión libre del protocolo Secure Shell (SSH) que facilita el control remoto seguro y encriptado y las operaciones de transferencia de datos.

El componente servidor de OpenSSH, sshd, escucha continuamente a la espera de conexiones de clientes desde cualquiera de las herramientas cliente. Cuando aparece una petición de conexión, sshd establece la conexión correcta dependiendo del tipo de herramienta cliente que está conectándose. Por ejemplo, si el equipo remoto se está conectando con la aplicación cliente ssh, el servidor OpenSSH establecerá una sesión de control remoto tras la autenticación. Si el usuario remoto se conecta al servidor OpenSSH con scp, el demonio del servidor OpenSSH iniciará una copia segura de archivos entre el servidor y el cliente tras la autenticación. OpenSSH puede usar muchos métodos de autenticación, incluyendo contraseñas planas, claves públicas y tickets de Kerberos.

2.1.2 Instalación y configuración

El paquete que implementa el servicio de servidor de OpenSSH es openssh-server y el del cliente openssh-client, que pueden ser instalados mediante Synaptic.

El archivo de configuración del servidor es /etc/ssh/sshd_config. Para obtener información más completa acerca de las opciones de configuración puede utilizarse el comando “man sshd_config”. Las diferentes opciones controlan aspectos como los métodos de autenticación, ajustes de comunicación, etc. Se recomienda hacer copia del fichero antes de proceder a su modificación.

Algunas de las opciones que se pueden modificar pueden ser:

- Port 2222.- para que OpenSSH escuche por el puerto 2222 en vez de usar el 22 que es el que se utiliza por defecto.
- PubkeyAuthentication yes.- para que sshd permita credenciales de inicio de sesión basados en clave pública. Puede que exista en el fichero pero se encuentre comentada.
- Banner /etc/issue.net.- mostrará el contenido del fichero /etc/issue.net como banner al iniciar el login el usuario.

Para que los cambios en la configuración surtan efecto hay que reiniciar el servidor sshd con la instrucción “sudo /etc/init.d/ssh restart”.

Existen muchas otras directivas de configuración disponibles para sshd que cambian el comportamiento de la aplicación servidor para ajustarlo a las necesidades. No obstante, si el único método de acceso a un servidor es ssh, y se comete un error al configurar sshd por medio del archivo /etc/ssh/sshd_config, puede conseguirse que el servidor se cierre durante el reinicio del mismo, o que el servidor sshd no quiera iniciarse debido a una directiva de configuración incorrecta, por lo que hay que ser extremadamente cuidadoso al editar este fichero desde un servidor remoto.

2.1.3 Conexión

La forma más sencilla de establecer la conexión desde el cliente es utilizando la instrucción:

ssh ip_del_servidor

A continuación el servidor solicitará usuario y contraseña. Puede especificarse el usuario en el mismo comando utilizando la sintaxis:

ssh [usuario@]ip_del_servidor

En lugar de la dirección ip del servidor puede utilizarse su nombre si se encuentra funcionando correctamente el servicio de nombres adecuado.

Una de las opciones más interesantes es “-X” que permite la conexión haciendo uso del entorno gráfico pudiendo así ejecutar cualquier programa gráfico del servidor.

2.1.4 Claves SSH

Las claves SSH permiten la autenticación entre dos equipos sin necesidad de una contraseña. La autenticación por clave SSH usa dos claves: una clave privada y una clave pública.

Para generar las claves, hay que escribir en una terminal:

ssh-keygen -t dsa

Esta instrucción generará las claves usando una identidad de autenticación DSA del usuario. Durante el proceso, se solicita una contraseña y simplemente hay que pulsar <Intro> al solicitar para crear la clave.

De manera predeterminada la clave publica está almacenada en el archivo ~/.ssh/id_dsa.pub, Mientras que ~/.ssh/id_dsa es la clave privada. Hay que copiar el archivo id_dsa.pub al equipo remoto y añadirlo a ~/.ssh/authorized_keys con la instrucción:

ssh-copy-id username@remotehost

Finalmente, hay que comprobar dos veces los permisos en el archivo authorized_keys; sólo los usuarios autenticados deberían tener permisos de lectura y escritura. Si los permisos no son correctos deben cambiarse con:

chmod 600 .ssh/authorized_keys

Ahora se deberá poder acceder al servidor mediante SSH sin utilizar contraseña.

2.2 eBox

Ebox es un marco de trabajo web utilizado para gestionar configuraciones en aplicaciones de servidor. El diseño modular de eBox permite tomar y elegir qué servicios configurar usando eBox.

2.2.1 Instalación

Los diferentes módulos de eBox están separados en diferentes paquetes, permitiendo instalar solo los necesarios. Una forma de ver los diferentes paquetes disponibles es introduciendo en un terminal lainstrucción:

apt-cache rdepends ebox | uniq

Al instalar ebox (sudo apt-get install ebox) se instalan automáticamente los módulos por defecto. Durante la instalación hay que suministrar la contraseña para el usuario ebox. La ejecución del programa se realiza desde el navegador mediante la dirección:

https://yourserver/ebox.

2.2.2 Configuración

Algo importante que hay que recordar cuando se usa eBox es que, cuando se configuran la mayoría de los módulos, existe un botón “Cambiar” que implementa la nueva configuración. Después de pulsar el botón Cambiar, la mayoría (pero no todos) de los módulos tendrán que ser Guardados. Para guardar la nueva configuración, se pulsa en el enlace “Guardar cambios” situado en la esquina superior derecha. Cuando se realiza un cambio que necesita ser guardado, el enlace cambiará de verde a rojo.

2.2.3 Módulos de eBox

Por defecto todos los Módulos de eBox están deshabilitados, y cuando se instala un nuevo módulo no se habilita automáticamente. Para habilitar un módulo deshabilitado, se pulsa en el enlace "Estado" de los módulos, en el menú de la izquierda. A continuación, se marcan los módulos que se desea habilitar y se pulsa en el enlace "Guardar".

Los módulos que se instalan por defecto son:

- Sistema.- contiene opciones que permiten la configuración general de eBox.
- General.- permite definir el lenguaje, el puerto, y contiene un formulario para cambio de contraseña.
- Uso del disco.- muestra un gráfico con información detallada sobre el uso del disco.
- Backup.- se usa para hacer copias de respaldo de la información de configuración de eBox, y la opción "Full Backup" permite guardar toda la información de eBox no incluida en la opción Configuration, como los archivos de registro.
- Halt/Reboot.- permite apagar o reiniciar el sistema.
- Bug Report.- crea un archivo con detalles de ayuda cuando se informa de errores a los desarrolladores de eBox.
- Logs.- permite consultar los registros de eBox dependiendo del tiempo de eliminación definido.
- Events.- permite enviar alertas a través de rss, jabber y archivo de registro. Los eventos disponibles son:
  - Free Storage Space.- envía una alerta si el espacio libre del disco cae por debajo del porcentaje configurado, que es 10% por defecto.
  - Log Observer.- envía una alerta cuando un registrador configurado tiene algo registrado.
  - RAID.- monitorizará el sistema RAID y enviará alertas si aparece algún problema.
  - Service.- envía alertas si el servicio se inicia muchas veces en un breve periodo de tiempo.
  - State.- alerta del estado de eBox, si está encendido o apagado.

Los informadores disponibles son:

- Log.- envía los mensajes de eventos de eBox al archivo /var/log/ebox/ebox.log.
- Jabber.- antes de activar este despachador debe configurarse pulsando el icono “Configurar”.
- RSS: una vez configurado podrá suscribirse al enlace para ver alertas de eventos.

Además, pueden utilizarse los módulos adiccionales:

- Network.- permite la configuración de las opciones de redes de servidor.
- Firewall.- configura las opciones de cortafuegos para el servidor eBox.
- UsersandGroups.- administrará los usuarios y grupos del directorio LDAP de OpenLDAP.
- DHCP.- proporciona una interfaz para configurar un servidor DHCP.
- DNS.- proporciona opciones de configuración para el servidor de DNS BIND9.
- Objects.- permite la configuración de objetos de red de eBox, que permiten asignar un nombre a una dirección IP o a un grupo de direcciones IP.
- Servicios.- muestra información de configuración de servicios disponibles en la red.
- Squid.- opciones de configuración para un servidor proxy Squid.
- CA.- configura la autoridad de certificación para el servidor.
- NTP.- establece las opciones del protocolo de tiempo de red.
- Impresoras.- permite la configuración de impresoras.
- Samba.- opciones de configuración de Samba.
- OpenVPN.- establece las opciones para la aplicación OpenVPN (Red Privada Virtual).

2.3 Webmin

Permite la configuración del equipo local o de otro remoto a través de páginas web.

2.4 Actividades

3 PROTOCOLOS VARIOS

3.1 Dynamic Host Configuration Protocol (DHCP)

El Protocolo de Configuración Dinámica de Hosts (DHCP, en inglés), es un servicio de red que permite que los equipos hosts sean configurados automáticamente desde un servidor en lugar de tener que configurar manualmente cada host de la red. Los equipos configurados para ser clientes DHCP no tienen control sobre la configuración que reciben del servidor DHCP, y la configuración es transparente para el usuario del equipo.

Las opciones de configuración más comunes suministradas por un servidor DHCP a los clientes DHCP incluyen:

- Dirección IP.
- Máscara de red.
- Puerta de enlace.
- DNS principal.
- DNS secundario.

También pueden configurar:

- Servidor WINS.
- Nombre de equipo.
- Nombre del dominio.
- Servidor de hora.
- Servidor de impresión.

La ventaja de usar DHCP es que un cambio en la red (por ejemplo, un cambio en la dirección del servidor DNS), sólo supone un cambio en el servidor DHCP, ya que todos los hosts de la red se reconfigurarán automáticamente la próxima vez que sus clientes DHCP soliciten la configuración al servidor DHCP. Como una ventaja añadida, también es más fácil integrar nuevos equipos en la red, ya que no es necesario comprobar la disponibilidad de la dirección IP. Los conflictos de direcciones IP también se reducen.

Un servidor DHCP puede proporcionar parámetros de configuración usando dos métodos:

- Dirección MAC.- supone el uso de DHCP para identificar el hardware único de cada tarjeta de red conectada a la red y continuar suministrando una configuración constante cada vez que el cliente DHCP hace una petición usando ese dispositivo de red.

- Depósito de direcciones.- implica la definición de un almacén (a veces también denominado rango o ámbito) de direcciones IP, que son suministradas dinámicamente a los clientes DHCP para formar parte de sus propiedades de configuración, según un esquema «primero en llegar, primero en ser servido». Cuando un cliente DHCP deja de estar en la red por un periodo específico de tiempo, la configuración expira y se libera, volviendo al almacén de direcciones para que pueda usarla otros clientes DHCP.

MaX viene equipado con un cliente DHCP y un servidor DHCP. El servidor es dhcpd (dynamic host configuration protocol daemon). El cliente suministrado por Ubuntu es dhclient y se debe instalar en los equipos que necesiten ser configurados automáticamente. Ambos programas son fáciles de instalar y de configurar, y deberían iniciarse automáticamente durante el arranque del sistema.

3.1.1 Instalación

El paquete a instalar en el servidor es dhcp3-server. Puede utilizarse Synaptic o la instrucción apt-get.

El fichero de configuración del servicio en el servidor es /etc/dhcp3/dhcpd.conf.

En el fichero /etc/default/dhcp3-server se especifican las interfaces que el proceso dhcpd debe atender. Por defecto, dhcpd atiende eth0.

dhcpd envía los mensajes de diagnóstico y errores a syslog.

3.1.2 Configuración

El siguiente ejemplo de fichero de configuración (/etc/dhcp3/dhcpd.conf) puede dar una idea bastante concreta de la configuración de dhcpd:

```
# Sample /etc/dhcpd.conf
# (add your comments here)
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option domain-name-servers 192.168.1.1, 192.168.1.2;
option domain-name "mydomain.example";
subnet 192.168.1.0 netmask 255.255.255.0 {
        range 192.168.1.10 192.168.1.100;
        range 192.168.1.150 192.168.1.200;
}
```

Esto hará que el servidor DHCP proporcione a un cliente una dirección IP dentro del rango 192.168.1.10 - 192.168.1.100 o 192.168.1.150 - 192.168.1.200. La concesión de la dirección IP durará 600 segundos, si el cliente no ha solicitado un intervalo de tiempo específico. En caso contrario, la concesión máxima permitida será de 7.200 segundos. El servidor también «aconsejará» al cliente que use 255.255.255.0 como su máscara de subred, 192.168.1.255 como su dirección de difusión, 192.168.1.254 como la dirección del router/pasarela, y 192.168.1.1 y 192.168.1.2 como sus servidores DNS.

Si se ecesita especificar un servidor WINS para sus clientes Windows, necesitará incluir la opción netbios-name-servers option, p.e.

option netbios-name-servers 192.168.1.1;

Los parámetros de configuración de dhcpd son tomados del DHCP mini-HOWTO, que se puede encontrar en http://www.tldp.org/HOWTO/DHCP/index.html.

La página https://help.ubuntu.com/community/dhcp3-server y la instrucción “man dhcpd.conf” muestran más información.

3.2 Sincronización de la hora con NTP

NTP es un protocolo TCP/IP para sincronizar la hora a través de una red. Básicamente, un cliente solicita la hora actual a un servidor, y usa la respuesta para poner en hora su propio reloj.

Tras esta descripción tan simple se esconde una enorme complejidad - hay niveles de servidores NTP, donde los servidores NTP de nivel uno están conectados a relojes atómicos (a menudo vía GPS), y los servidores de niveles dos y tres reparten la carga de gestionar las solicitudes a través de Internet. Además, el software del cliente es mucho más complejo de lo que pudiera pensar en un principio - tiene que tener en cuenta los retardos en las comunicaciones, y ajustar la hora de manera que no afecte a todos los demás procesos que se están ejecutando en el servidor.

MaX dispone de dos maneras de establecer la hora automáticamente: ntpdate y ntpd.

3.2.1 ntpdate

MaX contiene ntpdate como proceso estándar, y se ejecutará en el arranque para configurar su hora de acuerdo al servidor NTP de Ubuntu. Sin embargo, el reloj de un servidor es probable que derive considerablemente entre reinicios, así que tiene sentido corregir el tiempo de vez en cuando. La forma más fácil de hacerlo es conseguir que cron ejecute ntpdate cada día.

Con su editor favorito, como superusuario, cree un archivo /etc/cron.daily/ntpdate que contenga:

ntpdate ntp.ubuntu.com

El archivo /etc/cron.daily/ntpdate ha de tener permisos de ejecución:

sudo chmod 755 /etc/cron.daily/ntpdate

3.2.2 ntpd

ntpdate es un instrumento un tanto «burdo» - sólo puede ajustar la hora una vez al día, en una única gran corrección. El demonio ntpd es bastante más sutil. Calcula la desviación del reloj de su sistema y lo ajusta constantemente, por lo que no hace grandes correcciones que podrían provocar registros inconsistentes, por ejemplo. Hay un pequeño coste en potencia de procesamiento y memoria, pero para un servidor moderno resulta insignificante.

Para configurar ntpd:

sudo apt-get install ntp

3.2.3 Cambiar los servidores de hora

En los dos casos anteriores, su sistema usará por omisión el servidor NTP de Ubuntu, ntp.ubuntu.com. Esto es correcto, pero tal vez desee usar varios servidores para incrementar la precisión y la tolerancia, y quizá desee usar servidores de hora situados geográficamente más cerca de usted. Para hacer eso con ntpdate, hay que cambiar el contenido del archivo /etc/cron.daily/ntpdate a:

ntpdate ntp.ubuntu.com pool.ntp.org

Y para ntpd, edite el archivo /etc/ntp.conf para añadir líneas adicionales de servidores:

server ntp.ubuntu.com
server pool.ntp.org

Habrá observado que en los ejemplos anteriores aparece pool.ntp.org. Esta es una muy buena idea, pues se utilizan DNS rotatorios que devuelven por turnos un servidor NTP a partir de un depósito, distribuyendo así la carga entre varios servidores diferentes. Aún mejor, existen depósitos para diferentes regiones - por ejemplo, si está en Nueva Zelanda, puede usar nz.pool.ntp.org en lugar de pool.ntp.org. Consulte http://www.pool.ntp.org/ para más información.

También puede usar Google para encontrar servidores NTP situados cerca de usted, y añadirlos a su configuración. Para comprobar que un servidor funciona, simplemente teclee sudo ntpdate ntp.nombre.servidor y vea lo que ocurre.

3.3 File Transfer Protocol (FTP)

El Protocolo de Transferencia de Archivos (FTP) es un protocolo TCP para subir y descargar archivos entre ordenadores. El FTP funciona según el modelo cliente/servidor. El componente servidor se denomina demonio FTP. Está continuamente escuchando peticiones FTP de clientes remotos. Cuando se recibe una petición, gestiona la creación de la sesión y establece la conexión. Durante la duración de la sesión ejecuta las órdenes enviadas por el cliente FTP.

El acceso a un servidor FTP puede hacerse de dos maneras: anónimo o autenticado.

En el modo Anónimo, los clientes remotos pueden acceder al servidor FTP utilizando la cuenta predeterminada llamada "anonymous" o "ftp" y enviando una dirección de correo como contraseña.

En el modo Autenticado, el usuario debe tener una cuenta y contraseña. El acceso del usuario al servidor FTP y sus directorios dependen de los permisos definidos en el registro. Como regla general, el demonio FTP ocultará el directorio raíz del servidor FTP y lo cambiará por el directorio inicial del FTP. Esto oculta el resto del sistema de archivos a las sesiones remotas.

3.3.1 Instalación del servidor FTP vsftpd

vsftpd es un demonio FTP disponible en Ubuntu fácil de instalar, configurar y mantener. Para instalar vsftpd puede ejecutar el siguiente comando:

sudo apt-get install vsftpd

3.3.2 Configurar el FTP anónimo

De forma predeterminada, vsftpd no está configurado para aceptar conexiones anónimas. Para modificar este comportamiento hay que buscar y establecer en el fichero /etc/vsftpd.conf la opción “anonymous_enable=YES”. Durante la instalación, se crea un usuario llamado ftp con el directorio personal /srv/ftp. Este es el directorio de FTP predeterminado.

Si se desea cambiar esta ubicación a, por ejemplo, /srv/miftp, habría que realizar las siguientes operaciones:

sudo mkdir /srv/miftp	Para crear el directorio.
sudo usermos -d /srv/miftp ftp	Para cambiar el directorio de trabajo del usuario.
sudo chown root:ftp /srv/miftp	Para cambiar propietario y grupo del directorio.
sudo chmod 755 /srv/miftp	Para asignar los permisos adecuados.
sudo /etc/init.d/vsftpd restart	Para reiniciar el servicio.

Finalmente, copie a /srv/miftp todos los archivos y directorios que desee hacer disponibles por FTP anónimo.

3.3.3 Configurar el ftp autenticado

Para configurar vsftpd para autenticar a los usuarios del sistema y permitirles subir archivos, edite el archivo /etc/vsftpd.conf:

```
local_enable=YES
write_enable=YES
```

Ahora reinicie vsftpd:

```
sudo /etc/init.d/vsftpd restart
```

Ahora, cuando los usuarios del sistema inicien una sesión con FTP, aparecerán en sus directorios personales desde donde podrán descargar, subir archivos, crear directorios, etc.

Del mismo modo, por omisión, los usuarios anónimos no están autorizados a subir archivos al servidor FTP. Para cambiar esta opción, deberá descomentar la siguiente línea y reiniciar vsftpd:

```
anon_upload_enable=YES
```

Habilitar la subida por FTP anónimo puede ser un riesgo extremo de seguridad. Es mejor no habilitar la subida anónima en servidores a los que se tenga acceso directo desde Internet.

El archivo de configuración consiste en varios parámetros de configuración. La información acerca de cada parámetro está disponible en el archivo de configuración. También puede consultar la página de manual, man 5 vsftpd.conf, para obtener detalles de cada parámetro.

3.3.4 Asegurar el FTP

Hay opciones en /etc/vsftpd.conf que ayudan a que vsftpd sea más seguro. Por ejemplo, los usuarios pueden verse limitados a no salir de sus directorios personales descomentando:

chroot_local_user=YES

También puede limitar a sus directorios personales a una lista específica de usuarios:

chroot_list_enable=YES
chroot_list_file=/etc/vsftpd.chroot_list

Después de descomentar las opciones anteriores, cree un archivo /etc/vsftpd.chroot_list que contenga una lista de usuarios, uno por línea. A continuación, reinicie vsftpd:

sudo /etc/init.d/vsftpd restart

Además, el archivo /etc/ftpusers es una lista de usuarios que tienen prohibido el acceso por FTP. La lista predeterminada incluye los usuarios root, daemon, nobody, etc. Para deshabilitar el acceso FTP a más usuarios, simplemente añádalos a la lista.

FTP también puede cifrarse usando FTPS. FTPS es FTP sobre Secure Socket Layer (SSL), lo cual es distinto de SFTP. SFTP es una sesión de FTP sobre una conexión SSH cifrada. Una de las grandes diferencias es que los usuarios de SFTP necesitan tener una cuenta de intérprete de comandos en el sistema, en lugar de un intérprete nologin. Proporcionar acceso de intérprete de comandos a todos los usuarios puede no resultar ideal en algunos entornos, como en un host web compartido.

Para configurar FTPS, edite /etc/vsftpd.conf y añada al final:

ssl_enable=Yes

Además, tenga en cuenta las opciones relacionadas de certificados y claves:

rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key

Por omisión, estas opciones quedan establecidas al certificado y la clave proporcionados por el paquete ssl-cert. En un entorno de producción deberían cambiarse por un certificado y una clave generados para ese equipo en concreto. Para más información sobre certificados, consulte Sección 8.5: Certificados.

Ahora reinicie vsftpd y los usuarios no anónimos se verán forzados a usar FTPS:

sudo /etc/init.d/vsftpd restart

Para permitir que los usuarios con un intérprete de /usr/sbin/nologin puedan acceder por FTP, pero sin tener acceso a un intérprete, edite /etc/shells y añada el intérprete nologin:

```
# /etc/shells: valid login shells
/bin/csh
/bin/sh
/usr/bin/es
/usr/bin/ksh
/bin/ksh
/usr/bin/rc
/usr/bin/tcsh
/bin/tcsh
/usr/bin/esh
/bin/dash
/bin/bash
/bin/rbash
/usr/bin/screen
/usr/sbin/nologin
```

Esto es necesario porque, de forma predeterminada, vsftpd usa PAM para la autenticación, y el archivo de configuración /etc/pam.d/vsftpd contiene:

auth required pam_shells.so

El módulo shells de PAM restringe el acceso a los intérpretes indicados en el archivo /etc/shells.

Los clientes FTP más populares pueden configurarse para que se conecten mediante FTPS. El cliente FTP de línea de comandos lftp también tiene la capacidad de usar FTPS.

4 Autenticación en redes

Este tema puede seguirse a través del tema 6 de la Guía de Ubuntu Server..

5 Seguridad

Ver tema 8 de la Guía de Ubuntu Server.

6 Monitorizar

La monitorización de los servidores y servicios esenciales es una parte importante de la administración del sistema. La mayoría de servicios de red son monitorizados en su funcionamiento, disponibilidad, o ambos.

Este tema puede seguirse a través del tema 9 de la Guía de Ubuntu Server.

7 Servidores web

Ver tema 10 de la Guía de Ubuntu Server.

Tiene relevancia especial el apartado referente al servidor proxy Squid.

7.1 Servidor proxy Squid

Squid es una completa aplicación servidor proxy caché web que proporciona servicios de proxy y caché para Hyper Text Transport Protocol (HTTP), File Transfer Protocol (FTP) y otros protocolos populares de red. Squid puede servir de caché y de proxy para las peticiones Secure Sockets Layer (SSL) y de caché para las consultas de Domain Name Server (DNS), y proporciona servicios de caché transparente. Squid también soporta una amplia variedad de protocolos de caché, como el Internet Cache Protocol (ICP), el Hyper Text Caching Protocol (HTCP), el Cache Array Routing Protocol (CARP), y el Web Cache Coordination Protocol (WCCP).

El servidor proxy caché Squid es una solución excelente para una amplia variedad de necesidades de proxy y de caché, y escala desde redes de oficina hasta redes a nivel empresarial al tiempo que proporciona extensos y granulares mecanismos de control de acceso y monitorización de parámetros críticos a través del Simple Network Management Protocol (SNMP). Cuando elija un equipo para su uso como proxy Squid dedicado, o como servidor de caché, asegúrese de que su sistema dispone de una gran cantidad de memoria física, puesto que Squid mantiene una caché en memoria para mejorar el rendimiento.

7.1.1 Instalación

Hay que instalar el paquete squid (con Synaptic o apt-get).

7.1.2 Configuración

El fichero de configuración es /etc/squid/squid.conf. Los siguientes ejemplos ilustran algunas de las directivas que puede modificar para alterar el comportamiento del servidor Squid. Para una configuración más en profundidad, consulte la sección Referencias en la Guía de Ubuntu.

Para que su servidor Squid escuche en el puerto TCP 8888 en lugar del puerto TCP 3128 que usa por defecto, cambie la directiva http_port como sigue:

http_port 8888

Cambie la directiva visible_hostname para hacer que el servidor Squid tenga un nombre de host específico. Este nombre de host no tiene por qué ser necesariamente el nombre de host del equipo. En este ejemplo se ha establecido a weezie

visible_hostname weezie

Usando el control de acceso de Squid, puede configurar el uso de los servicios de Internet intermediados por Squid para que estén sólo disponibles para los usuarios que usen determinadas direcciones IP. Por ejemplo, ilustraremos el acceso sólo de los usuarios de la subred 192.168.42.0/24:

Añada lo siguiente al final de la sección ACL de su archivo /etc/squid/squid.conf:

acl fortytwo_network src 192.168.42.0/24

Después, añada lo siguiente al principio de la sección http_access de su archivo /etc/squid/squid.conf:

http_access allow fortytwo_network

Usando las excelentes capacidades de control de acceso de Squid, puede configurar el uso de los servicios de Internet delegados por Squid para que sólo estén disponibles durante las horas normales de trabajo. Por ejemplo, ilustraremos el acceso de los empleados de un negocio que opera entre las 9 de la mañana y las 5 de la tarde, de lunes a viernes, y que usan la subred 10.1.42.0/42:

Añada lo siguiente al final de la sección ACL de su archivo /etc/squid/squid.conf:

acl biz_network src 10.1.42.0/24
acl biz_hours time M T W T F 9:00-17:00

Después, añada lo siguiente al principio de la sección http_access de su archivo

/etc/squid/squid.conf:

http_access allow biz_network biz_hours

Después de hacer los cambios en el archivo /etc/squid/squid.conf, guarde el archivo y reinicie el servidor squid para que los cambios surtan efecto, usando la siguiente orden que deberá introducir en la línea de órdenes de una terminal:

sudo /etc/init.d/squid restart

8 Redes Windows

Ver tema 17 de la Guía de Ubuntu Server.

9 Red privada virtual VPN

Una Red privada virtual (Virtual Private Network), o VPN, es una conexión de red cifrada entre dos o más redes. Hay distintas maneras de crear una VPN usando software así como aplicaciones de hardware dedicado.

Este tema se desarrolla en el tema 21 de la Guía de Ubuntu Server.

10 Servidor de nombres DNS

El Servicio de Nombres de Dominio (Domain Name Service, DNS) es un servicio de Internet que hace corresponder direcciones IP con nombres de dominio totalmente cualificados (FQDN), unos con otros. De esta forma, DNS evita tener que recordar direcciones IP. Los equipos que ejecutan DNS se denominan servidores de nombres. Ubuntu trae el BIND (Berkley Internet Naming Daemon), el programa más usado para mantener un servidor de nombres en Linux.

10.1 Instalación

El paquete servidor de DNS es bind9, que puede instalarse sin problemas.

El paquete dnsutils resulta muy útil para probar y resolver problemas del DNS.

10.2 Configuración

Hay diferentes maneras de configurar BIND9. Algunas de las configuraciones mas comunes son caché de servidor de nombres, maestro primario, y como maestro secundario.

Cuando se configura como servidor caché de nombres de dominio, BIND9 busca la respuesta a solicitudes de nombres y recordará la respuesta cuando ese dominio sea solicitado nuevamente.

Como un servidor maestro primario BIND9 lee los datos para una zona de un archivo en su equipo y es autoritativo para esa zona.

En una configuración de maestro secundario BIND9 obtiene los datos de la zona de otro servidor de nombres autoritativo para la zona.

Los archivos de configuración DNS son almacenados en el directorio /etc/bind. El archivo de configuración primario es /etc/bind/named.conf.

La linea include especifica el nombre de archivo que contiene las opciones DNS. La linea directory en el archivo /etc/bind/named.conf.options le dice a DNS donde buscar archivos. Todos los archivos que BIND usa serán relativos a este directorio.

El archivo de nombre /etc/bind/db.root describe los servidores de nombres raíz para todo el mundo. Los servidores cambian con el tiempo, por lo que /etc/bind/db.root debe actualizarse de vez en cuando. Esto se hace habitualmente mediante actualizaciones del paquete bind9. La sección zone define un servidor maestro, que es almacenado en el archivo indicado por la opción file.

Es posible configurar el mismo servidor para que actúe como servidor de caché de nombres, maestro primario y maestro secundario. Un servidor puede ser el Start of Authority (SOA) de una zona, al tiempo que proporciona servicio secundario para otra zona. Y todo al mismo tiempo que proporciona servicios de caché para equipos de la LAN local.

10.2.1 Servidor de nombres cache

La configuración por defecto es configurada para actuar como un servidor caché. Todo lo requerido es simplemente agregar la Dirección IP de los servidores DNS de su ISP. Simplemente, descomente y edite lo siguiente en /etc/bind/named.conf.options:

```
forwarders {
        1.2.3.4;
        5.6.7.8;
};
```

Hay que reemplazar 1.2.3.4 y 5.6.7.8 con las direcciones IP de los nombres de servidores actuales. Ahora reinicie el servidor DNS para habilitar la nueva configuración:

```
sudo  /etc/init.d/bind9  restart
```

10.2.2 Maestro primario

En esta sección, BIND9 será configurado como el Maestro Primario (Primary Master) para el dominio ejemplo.com. Simplemente sustituya ejemplo.com por su nombre de dominio totalmente cualificado (Fully Qualified Domain Name, FQDN).

Archivo de zona de redirección

Para agregar una zona DNS a BIND0, cambiando a BIND9 en un servidor Maestro Primario, el primer paso es editar /etc/bind/named.conf.local:

```
zone "example.com" {
        type master;
        file "/etc/bind/db.example.com";
};
```

Ahora utilice un archivo de zona existente como una plantilla para crear el archivo /etc/bind/db.example.com:

```
sudo  cp  /etc/bind/db.local  /etc/bind/db.example.com
```

Edite el nuevo archivo de zona /etc/bind/db.example.com y sustituya localhost. por el nombre (FQDN) de su servidor, dejando el "." final. Sustituya 127.0.0.1 por la dirección IP del servidor de nombres y root.localhost por una dirección de correo electrónico válida, pero con un "." en lugar del símbolo usual "@", dejando de nuevo el "." al final.

También, cree un registro A para ns.example.com.

El fichero tendrá la forma:

```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns.ejemplo.com. root.ejemplo.com. (
                        2               ; Serial
                        604800          ; Refresh
                        86400           ; Retry
                        2419200         ; Expire
                        604800 )        ; Negative Cache TTL
;
@       IN      NS      ns.ejemplo.com.
@       IN      A       127.0.0.1
@       IN      AAAA    ::1
ns      IN      A       192.168.1.10
```

Debe incrementar Numero de Serie cada vez que haga cambios en el archivo de zona. Si hace multiples cambios antes de reiniciar BIND9, simplemente incremente la Serie una vez.

Ahora se pueden agregar registros DNS al final del archivo de zona.

A muchos administradores les gusta utilizar la ultima fecha de edición como la serie de una zona, así como 2007010100 que es yyyymmddss (donde ss es el Numero de Serie).

Una vez que haya hecho un cambio en el archivo de zonas, tendrá que reiniciar BIND9 para que los cambios tengan efecto con la instrucción:

```
sudo /etc/init.d/bind9 restart
```

Archivo de zona inversa

Ahora que se ha configurado la zona y se resuelven nombres a direcciones IP, también se nececista una zona inversa (Reverse zone). Una zona inversa permite al DNS resolver una dirección a un nombre.

Edite /etc/bind/named.conf.local y añada lo siguiente:

```
zone "1.168.192.in-addr.arpa" {
        type master;
        notify no;
        file "/etc/bind/db.192";
};
```

Hay que reemplazar 1.168.192 con los tres primeros octetos de cualquiera sea la red que usted esta utilizando. También, nombre el archivo de zona /etc/bind/db.192 apropiadamente. Debe de coincidir el primer octeto de su red.

Ahora crea el archivo /etc/bind/db.192: sudo cp /etc/bind/db.127 /etc/bind/db.192

Luego edite /etc/bind/db.192 cambiando básicamente las mismas opciones que en /etc/bind/db.example.com:

```
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns.ejemlpo.com. root.ejemplo.com. (
                2               ; Serial
                604800          ; Refresh
                86400           ; Retry
                2419200         ; Expire
                604800 )        ; Negative Cache TTL
;
@       IN      NS      ns.
10      IN      PTR     ns.ejemplo.com.
```

El número de serie en la zona inversa debe incrementarse con cada cambio. Para cada entrada A que configure en /etc/bind/db.example.com debe crear una entrada PTR en /etc/bind/db.192.

Después de haber creado el archivo de zona inverso reinicie BIND9:

sudo /etc/init.d/bind9 restart

Maestro secundario

Una vez que un Maestro Primario haya sido configurado, un Maestro Secundario es necesario para mantener la disponibilidad del dominio si el Primario se vuelve no disponible.

Primero, en el servidor maestro primario, se necesita permitir la transferencia de la zona. Añada la opción allow-transfer a las definiciones de ejemplo de zonas directa (Forward) e inversa (Reverse) en /etc/bind/named.conf.local:

```
zone "example.com" {
        type master;
        file "/etc/bind/db.example.com";
        allow-transfer { 192.168.1.11; };
};
```

```
zone "1.168.192.in-addr.arpa" {
        type master;
        notify no;
        file "/etc/bind/db.192";
        allow-transfer { 192.168.1.11; };
};
```

Reemplace 192.168.1.11 con la dirección IP de su servidor de nombres secundario.

A continuación, en el maestro secundario, instale el paquete bind9 de la misma forma que para el primario. Luego, edite /etc/bind/named.conf.local y añada las siguientes declaraciones para las zonas directa (Forward) e inversa (Reverse):

```
zone "ejemplo.com" {
        type slave;
        file "/var/cache/bind/db.ejemplo.com";
        masters { 192.168.1.10; };
};

zone "1.168.192.in-addr.arpa" {
        type slave;
        file "/var/cache/bind/db.192";
        masters { 192.168.1.10; };
};
```

Reemplace 192.168.1.10 con la dirección IP de su servidor de nombres Primario.

Reinicie BIND9 en su Maestro Secundario: sudo /etc/init.d/bind9 restart

En /var/log/syslog usted debería haber algo similar a:

```
slave zone "example.com" (IN) loaded (serial 6)
slave zone "100.18.172.in-addr.arpa" (IN) loaded (serial 3)
```

Nota: Una zona sólo es transferida si el número de serie en el primario es mayor que en el secundario.

El directorio predeterminado para los archivos de zonas no autorizadas es /var/cache/bind/. Este directorio también está configurado en AppArmor para permitir que el demonio named pueda escribir en él.

10.3 Resolución de problemas

Esta sección puede ayudar a determinar la causa de los problemas con DNS y BIND9.

10.3.1 resolv.conf

El primer paso para testar BIND9 es añadir la dirección IP del servidor de nombres al buscador de nombres de un anfitrión. Tanto el servidor de nombres primario como otro anfitrión deben ser configurados para hacer dobles comprobaciones. Simplemente edite /etc/resolv.conf y añada lo siguiente:

```
nameserver    192.168.1.10
nameserver    192.168.1.11
```

Debe agregarse la dirección IP del servidor de nombres Secundario en caso que el Primario se vuelva no disponible.

10.3.2 dig

Con el paquete dnsutils se instala la utilidad dig que permite hacer peticiones DNS:

Tras instalar BIND9 use dig contra el interfaz local para asegurarse de que está escuchando en el puerto 53. Desde la línea de órdenes de un terminal:

```
dig  -x  127.0.0.1
```

Se mostrarán unas líneas similares a las siguientes en la salida de la orden:

```
;; Query time: 1 msec
;; SERVER: 192.168.1.10#53(192.168.1.10)
```

Si ha configurado BIND9 como un servidor de nombres caché, use dig sobre un dominio externo para comprobar el tiempo de consulta:

```
dig ubuntu.com
```

Observar el tiempo de consulta hacia el final de la salida de la orden:

```
;; Query time: 49 msec
```

Tras un segundo dig debe haber una mejora:

;; Query time: 1 msec

10.3.3 eco

Ahora, para demostrar cómo las aplicaciones hacen uso del DNS para resolver el nombre de un equipo, use la utilidad ping para enviar una petición de eco ICMP. Introduzca en un terminal:

ping example.com

Esto comprueba el servidor de nombres puede resolver el nombre ns.ejemplo.com a una dirección IP. La salida del comando debe parecerse a:

PING ns.ejemplo.com (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=64 time=0.800 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=64 time=0.813 ms

10.3.4 named-checkzone

Una excelente manera de probar sus archivos de zona es usando la utilidad named-checkzone instalada con el paquete bind9. Esta utilidad le permite asegurarse de que la configuración es correcta antes de reiniciar BIND9 y hacer que los cambios hagan efecto.

Para probar el archivo de ejemplo de zona de redirección introduzca lo siguiente en un terminal:

named-checkzone ejemplo.com /etc/bind/db.ejemplo.com

Si todo está correctamente configurado deberá ver algo similar a esto:

zone example.com/IN: loaded serial 6
OK

De la misma forma, para probar el archivo de la zona inversa introduzca lo siguiente:

named-checkzone example.com /etc/bind/db.192

La salida debe ser similar a:

zone example.com/IN: loaded serial 3
OK

Probablemente, el número de serie de su archivo de zona será diferente.

10.3.5 Registro de información

BIND9 dispone de una amplia variedad de opciones de configuración sobre registros. Hay dos opciones principales. La opción channel configura dónde van los registros, y la opción category determina qué información se va a registrar.

Si no se ha configurado la opción de entrada, la predeterminada es:

```
logging {
        category default { default_syslog; default_debug; };
        category unmatched { null; };
};
```

Esta sección cubre la configuración de BIND9 para enviar mensajes de depuración relacionados con solicitudes DNS a un archivo separado.

Primero, tenemos que configurar un canal para especificar a qué archivo enviar los mensajes. Edite el archivo /etc/bind/named.conf.local y añada lo siguiente:

```
logging {
   channel query.log {
     file "/var/log/query.log";
     severity debug 3;
   };
};
```

A continuación, configure una categoría para enviar todas las solicitudes DNS al archivo de solicitudes:

```
logging {
        channel query.log {
                file "/var/log/query.log";
                severity debug 3;
        };
        category queries { query.log; };
};
```

Nota:la opción debug puede establecerse de 1 a 3. Si no se especificapor defecto se toma 1.

Ya que el demonio named corre bajo el usuario bind, el archivo /var/log/query.log se debe crear y cambiar de propietario:

```
sudo touch /var/log/query.log
```

sudo chown bind /var/log/query.log

Antes de que el demonio named pueda escribir en el nuevo archivo de registro, el perfil AppArmor debe actualizarse. Primero, edite /etc/apparmor.d/usr.sbin.named y añada:

/var/log/query.log w,

Ahora, recargue el perfil:

cat /etc/apparmor.d/usr.sbin.named | sudo apparmor_parser -r

Para más información sobre AppArmor vea Sección 8.4 del manual de Ubuntu.

Ahora reinicie BIND9 para que los cambios tengan efecto:

sudo /etc/init.d/bind9 restart

Debe ver como el archivo /var/log/query.log se llena con información de peticiones. Este es un ejemplo simple de las opciones de registro de BIND9. Para opciones avanzadas vea Sección 7.4.2 del manual de Ubuntu.

10.4 Tipos de registros habituales

Esta sección cubre algunos de los tipos de registros DNS más comunes.

- Registro A: Este registro asocia una dirección IP a un nombre de equipo.

```
www   IN    A     192.168.1.12
```

- Registro CNAME: Usado para crear un alias a un registro A ya existente. No puede crear un registro CNAME que apunte a otro registro CNAME.

```
web   IN    CNAME     www
```

- Registro MX: Usado para definir a dónde se deben enviar los correos electrónicos. Debe apuntar a un registro A, no a un registro CNAME.

```
      IN    MX    1     mail.ejemplo.com.
Mail  IN    A     192.168.1.13
```

- Registro NS: Usado para definir qué servidores sirven copias de una zona. Debe apuntar a un registro A, no a un CNAME. Aquí es donde se definen los servidores Primario y Secundario.

```
        IN      NS      ns.example.com.
        IN      NS      ns2.example.com.
ns      IN      A       192.168.1.10
ns2     IN      A       192.168.1.11
```

10.5 Más información

El HOWTO del DNS (en inglés) explica opciones más avanzadas de configuración de BIND9.

Para un tratamiento en profundidad de DNS y BIND9 ver Bind9.net.

DNS y BIND es un libro popular ahora en su quinta edición.

Un magnífico lugar para pedir ayuda sobre BIND9 e involucrarse en la comunidad de servidores Ubuntu (Ubuntu Server) es el canal IRC #ubuntu-server en freenode.

También vea COMO del servidor BIND9 en el wiki de Ubuntu.


MAX v.6.0
Edición Servidor
DUCAMADRID
Comunidad de Madrid
CONSEJERÍA DE EDUCACIÓN