

Gestión de un Servidor OpenBSD

Nicolás A. Ortega Froysa

7 de noviembre de 2022

Índice

1. Introducción	2
2. Configuración Inicial	2
3. Instalación de Paquetes	3
4. Configuración de Servicios	4
4.1. Nginx	4
4.2. MariaDB	6
4.3. Servicio RSyncd	6
5. Conclusión	8
6. Derechos de Autor y Licencia	9

1. Introducción

En esta práctica vamos a ver cómo es la administración de sistema de un servidor de *OpenBSD*. Los objetivos son los siguientes:

- Aprender los comandos básicos del sistema.
- Ver en qué se diferencia OpenBSD de un sistema Linux.
- Ver cómo cambia la carga del sistema mientras más servicios vayamos proveyendo.

La instalación en sí no forma parte de este documento, pero generalmente el instalador guiado es bastante intuitivo.

Lo que caracteriza a OpenBSD especialmente, respecto a otros sistemas UNIX (y otros sistemas que hemos visto en clase) son dos cosas. Primero, que no es una distribución de Linux, sino de BSD. Son sistemas completamente distintos, aunque siguen (generalmente) una filosofía/arquitectura similar, basándose los dos en UNIX. Segundo, que OpenBSD, entre los otros sistemas BSD, es el más especializado en seguridad, hasta el punto de sacrificar cualquier otro aspecto de la computación a favor siempre de la seguridad. Es por este motivo que deshabilitaron en su momento el *hyper-threading* de los procesadores Intel por miedo a *Spectre*.¹

2. Configuración Inicial

Generalmente la configuración de OpenBSD ha sido bastante simple, y la mayoría de las opciones están en su valor por defecto. El teclado – i.e. keymap – usado es el estadounidense, simplemente porque para la informática es la más cómoda. La máquina lleva el *hostname* de **puffy-nortega**.

En cuanto a los usuarios, existe un usuario *root* con contraseña **toor**, y otro usuario *nicolas* con contraseña **nicolas**. Durante la instalación se ha configurado que no sea posible acceder a *root* por SSH, ya que esto hace que el sistema sea más vulnerable – más todavía con este tipo de contraseña.

El disco se ha particionado con una sola partición para simplificar este proceso de instalación. Se podría instalar con más particiones, repartiendo así los datos y haciendo más fácil el proceso de hacer copias de respaldo (que sean de una sola partición), pero eso no entra dentro de los objetivos de esta práctica.

¹https://www.theregister.com/2018/06/20/openbsd_disables_intels_hyperthreading/

```

load averages: 0.00, 0.00, 0.00          puffly-nortega.my.domain 19:28:38
32 processes: 31 idle, 1 on processor      up 0 days 00:37:48
CPU:  0.0% user,  0.0% nice,  0.0% sys,  0.0% spin,  0.0% intr, 100% idle
Memory: Real: 31M/905M act/tot Free: 977M Cache: 392M Swap: 0K/0K

  PID USERNAME  PRI  NICE  SIZE  RES  STATE  WAIT  TIME  CPU  COMMAND
    1  root         10    0  656K  628K  idle   wait   0:01  0.00%  init
27022  _pflogd      4    0  772K  1576K  sleep  bpf    0:00  0.00%  _pflogd
25429  _ntp         2   -20 1188K 3020K  sleep  kqread 0:00  0.00%  _ntpd
77046  root         18    0  956K  936K  idle   sigsusp 0:00  0.00%  ksh
 1542  root         28    0  960K  2184K  onproc -       0:00  0.00%  top
 7358  _syslogd     2    0 1228K 1516K  sleep  kqread 0:00  0.00%  _syslogd
42906  root         2   -20 1076K 1744K  idle   kqread 0:00  0.00%  _ntpd
92240  root         2    0  876K  1516K  idle   kqread 0:00  0.00%  cron
47390  root         2    0  848K  1412K  idle   kqread 0:00  0.00%  dhcpleased
44031  _ntp         2    0 1076K 2736K  idle   kqread 0:00  0.00%  _ntpd
98759  _smtpd       2    0 1936K 4720K  idle   kqread 0:00  0.00%  _smtpd
89999  _smtpq       2    0 1652K 3396K  idle   kqread 0:00  0.00%  _smtpd
74067  root         2    0  760K  2472K  idle   netio   0:00  0.00%  _syslogd
  288  _smtpd       2    0 1452K 3224K  idle   kqread 0:00  0.00%  _smtpd
32276  root         2    0  524K  1112K  idle   kqread 0:00  0.00%  resolvd
28058  _smtpd       2    0 1660K 3448K  idle   kqread 0:00  0.00%  _smtpd
75849  _dhcp        2    0  864K  1684K  idle   kqread 0:00  0.00%  dhcpleased
16431  _smtpd       2    0 1524K 3296K  idle   kqread 0:00  0.00%  _smtpd
31197  _smtpd       2    0 1452K 3244K  idle   kqread 0:00  0.00%  _smtpd

```

Figura 1: Primer arranque de OpenBSD.

En cuestión de configuración de la máquina (virtual), tiene a su disposición 2GiB de RAM, y un *core* para procesar.

Finalmente, en OpenBSD se dividen los archivos de sistema en diferentes *sets*. Éstos se pueden usar para instalar diferentes componentes del sistema, desde el *kernel* hasta algunos juegos de consola. Éstos están los *sets* instalados:

- `bsd`: *kernel* de BSD (**obligatorio**).
- `bsd.mp`: *kernel* multi-procesador.
- `bsd.rd`: *kernel* de uso en memoria.
- `base72.tgz`: sistema base o *userland* (**obligatorio**).
- `comp72.tgz`: colección de compiladores.
- `man72.tgz`: páginas de manual.

Al inicializar la máquina lleva una carga de sistema extremadamente baja, ya que está corriendo tan sólo lo mínimo (figura 1).

3. Instalación de Paquetes

Para administrar paquetes en OpenBSD se hace uso de las herramientas de paquete de OpenBSD. Los comandos se pueden reconocer porque llevan la patrón `pkg_*`. En particular, las que nos interesan son tres:

- `pkg_add`: instalar y actualizar.
- `pkg_delete`: eliminar/desinstalar.
- `pkg_info`: buscar información.

Para instalar los paquetes que nos interesan vamos a correr el comando siguiente:

```
root# pkg_add nginx mariadb-server mariadb-client \
rsync
```

4. Configuración de Servicios

El control de los servicios en OpenBSD se hace con el comando `rcctl`, ya que OpenBSD no usa Systemd. Generalmente se usa de una forma muy similar a Systemd, usando los subcomandos siguientes:

- `start`: iniciar un servicio.
- `stop`: parar un servicio.
- `restart`: reiniciar el servicio.
- `enable`: habilitar un servicio para iniciarse al arrancar la máquina.
- `disable`: deshabilitar un servicio para que no se inicie cuando arranque la máquina.
- `check`: mostrar el estado de un servicio.
- `ls`: mostrar una lista de los servicios.

4.1. Nginx

Nombre	Nginx
Descripción	Un servidor web moderno.
Dir. Config.	<code>/etc/nginx/</code>
Dir. Logs	<code>/etc/nginx/logs/</code>

Cuadro 1: Datos sobre Nginx.

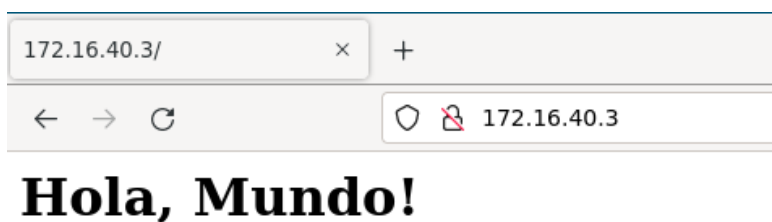


Figura 2: Prueba de Nginx.

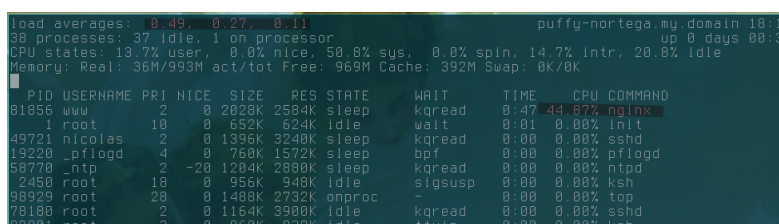


Figura 3: Load resultante de una prueba con *wrk*.

En nuestra configuración de Nginx vamos a crear una página simple que se pueda acceder desde fuera en el puerto 80. Por defecto, Nginx en OpenBSD usa el directorio `/var/www/htdocs/` como raíz del servidor. Aquí añadiremos un archivo `index.html` para que responda con un código 200 OK. Iniciamos el servicio (y lo habilitamos) con el comando siguiente:

```
root# rcctl enable nginx
root# rcctl start nginx
nginx(ok)
```

Ahora, si probamos acceder a nuestro servidor, veremos el archivo que hemos creado (figura 2).

A este servidor le podemos hacer un *stress test* utilizando la herramienta *wrk*, que sirve para hacer *benchmarking* de servicios HTTP. Desde un cliente corremos el comando siguiente, que nos hará una prueba de 50 conexiones paralelas, sobre 10 hilos, durante 20 segundos. Vemos que el *load* aumenta de forma considerable (figura 3).

```
$ wrk -c 50 -t 10 -d 20 http://172.16.40.3/
```

Nombre	MariaDB
Descripción	Sistema de gestión de bases de datos.
Archivo Conf.	/etc/my.cnf
Dir. Logs	/var/db/mysql/

Cuadro 2: Datos sobre MariaDB.

4.2. MariaDB

MariaDB requiere de una configuración inicial más manual, así que en primer lugar se tendrá que inicializar (y habilitar) el servicio, que se denomina `mysqld` – esto es porque MariaDB tiene como objetivo ser un reemplazo *in-place* de MySQL.

```
root# rcctl enable mysqld
root# rcctl start mysqld
mysqld(ok)
```

Lo configuramos usando el comando `mysql_secure_install`. Esto instalará las tablas dentro del directorio `/var/mysql/` y podremos definir la contraseña de `root`, que será `Rooty` (y algunas opciones de MariaDB). Podemos acceder a este base de datos usando el comando siguiente desde el usuario `root`, e introduciendo la contraseña que hemos asignado:

```
puffy-nortega# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 13
Server version: 10.9.3-MariaDB OpenBSD port:
mariadb-server-10.9.3v1

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and
others.

Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.

MariaDB [(none)]>
```

4.3. Servicio RSyncd

RSync es un servicio/herramienta de transferencia de archivos. Cuando se usa en modo *demonio*, puede crear un servidor de ficheros disponible en el

Nombre	RSyncd
Descripción	Servicio
Archivo Conf.	/etc/rsyncd.conf
Archivo Log	/var/log/rsyncd.log

Cuadro 3: Datos sobre RSyncd.

puerto 873 por defecto. Puede ser útil para sincronizar archivos y repositorios. Se configura creando un archivo de configuración en `/etc/rsyncd.conf`, y lo configuramos con lo siguiente:

```
uid = nobody
gid = nobody
use chroot = no
max connections = 4
pid file = /var/run/rsyncd.pid
lock file = /var/run/rsyncd.lock
log file = /var/log/rsyncd.log
```

```
[files]
path = /srv/rsync
comment = Mis archivos.
read only = true
list = true
```

Esto creará un servicio que proveerá los ficheros que se encuentran en el directorio `/srv/rsync`. Para nuestra prueba vamos a poner un par de archivos pesados – unos ISOs – para descargarlos desde nuestro cliente.

Habilitamos e iniciamos el servicio de RSyncd usando los comandos siguientes:

```
root# rcctl enable rsyncd
root# rcctl start rsyncd
rsyncd(ok)
```

Una vez habilitado el servicio, y creado los archivos, podemos poner a prueba nuestro servicio corriendo el comando siguiente desde nuestro cliente:

```
$ rsync -avz 172.16.40.3::files ./
```

A partir de aquí, veremos que aumenta considerablemente la carga en nuestro servidor (figura 4), mientras el cliente va descargando los ficheros. Además, conllevará una carga adicional, ya que con el *flag* `-z` comprimirá los datos antes de enviarlos para ser descomprimidos en nuestro cliente.


```
load averages: 0.70, 0.32, 0.16          puffry-nortege.ny.domain 19:50:28
42 processes: 1 running, 40 idle, 1 on processor          up 8 days 02:00:00
CPU states: 71.8% user, 0.0% nice, 17.5% sys, 0.0% spin, 7.3% intr, 3.4% idle
Memory: Real: 120M/1005M act/tot Free: 877M Cache: 392M Swap: 0K/0K
|
| PID USERNAME PRI NICE SIZE RES STATE  WAIT  TIME  CPU COMMAND
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
39454 nobody 53 0 2660K 2552K run - 0:22 63.09% rsyncd
74110 _mysql 2 0 461M 90M idle kqread 0:03 0.00% mariadb
96044 nicolas 2 0 1404K 330K sleep kqread 0:01 0.00% sshd
1 root 10 0 556K 636K idle wait 0:01 0.00% init
85153 pflogd 4 0 776K 1576K sleep bpif 0:01 0.00% pflogd
41421 ntpd 2 -20 1212K 2880K sleep kqread 0:00 0.00% ntpd
80659 root 10 0 955K 348K idle slausp 0:00 0.00% ksh
```

Figura 4: Prueba de carga de RSyncd.

5. Conclusión

OpenBSD, aunque tiene sus diferencias respecto a Linux, es bastante similar. Lo más diferente sería su sistema de servicios, que se controla a través de `rcctl` en vez de `systemctl`. También su sistema de administración de paquetes es distinto a lo normal que se encuentra en Ubuntu o Debian, y quizá se podría decir que es un poco más lento.

La forma de administrar el sistema es más conforme a la tradición de UNIX, que también lo seguía más Linux en sus inicios. Hay una dependencia más fuerte en la configuración de ficheros.

Es más, como OpenBSD está optimizado para la seguridad, sería muy buena opción para un servidor, ya que estaría expuesto al público.

Un negativo que podría tener es que como no es un Linux, sino un BSD, tiene menos soporte, hay menos recursos en la web que podemos consultar, y suele tener menos servicios disponibles – aunque los principales siempre suelen estar.

6. Derechos de Autor y Licencia

Copyright © 2022 Nicolás A. Ortega Froya <nicolas@ortegas.org>

Este documento se distribuye bajo los términos y condiciones de la licencia Creative Commons Attribution No Derivatives 4.0 International.