Administración de los demonios

Departamento de Sistemas Telemáticos y Computación (GSyC)

http://gsyc.urjc.es

Mayo de 2012

Administración de los demonios

Los demonios son programas relativamente *normales*, con algunas particularidades

- Ofrecen servicios (impresión, red, ejecución periódica de tareas, logs, etc)
- Suelen estar creados por el proceso de arranque *init* (ppid=1)
- Sus nombres suelen acaban en *d*
- Se ejecutan en *background*
- No están asociados a un usuario en una terminal
- Se inician y se detienen de manera uniforme

Puesta en marcha y parada

El código de un demonio puede estar en cualquier lugar del sistema de ficheros. Pero siempre se coloca en `/etc/init.d/midemonio` un script para manejarlo

- `/etc/init.d/midemonio start`
  Inicia el servicio
- `/etc/init.d/midemonio stop`
  Detiene el servicio
- `/etc/init.d/midemonio restart`
  Detiene e inicia el servicio. Suele ser **necesario para releer los ficheros de configuración** si se han modificado (`/etc/midemonio`)

Con frecuencia también está disponible

- `/etc/init.d/midemonio reload`
  Lee el fichero de configuración sin detener el servicio

Para saber si el demonio está en ejecución

- `ps -ef | grep midemonio`
  o bien
  `ps aux | grep midemonio`

Para saber si el demonio está escuchando en algún socket TCP o UDP, podemos usar `netsat`

Consulta de logs

Un pilar de la administración segura de un sistema es la auditoría, que permite :

- Saber qué está pasando/ qué ha pasado
- Mantener esta información a salvo de potenciales atacantes (que intentarán borrar su rastro)

Esta es una tarea compleja para la que hay multitud de herramientas de alto nivel distintas (Nagios, OpenNMS, ZABBIX...)
En el nivel más elemental tenemos, entre otros, el sistema de logs

- Los demonios usan el demonio *syslogd* o *sysklogd* para notificar y almacenar información relevante: inicio, parada, estado, peticiones, respuestas, errores, etc
  A partir del año 2009 es más habitual emplear *rsyslogd*, muy similar a *syslogd*
- Todo esto se escribe en diversos ficheros de texto, siendo los más interesantes
  - `/var/log/syslog`
    Información general del sistema
  - `/var/log/auth.log`
    Información sobre autenticación de usuarios

- Podemos ver un fichero cualquiera, p.e. un log, con *cat*

  `cat /var/log/syslog`
  (Muestra un fichero entero)
- Es más práctico usar *tail*

  `tail -20 /var/log/syslog`
  (Muestra las últimas 20 línea)

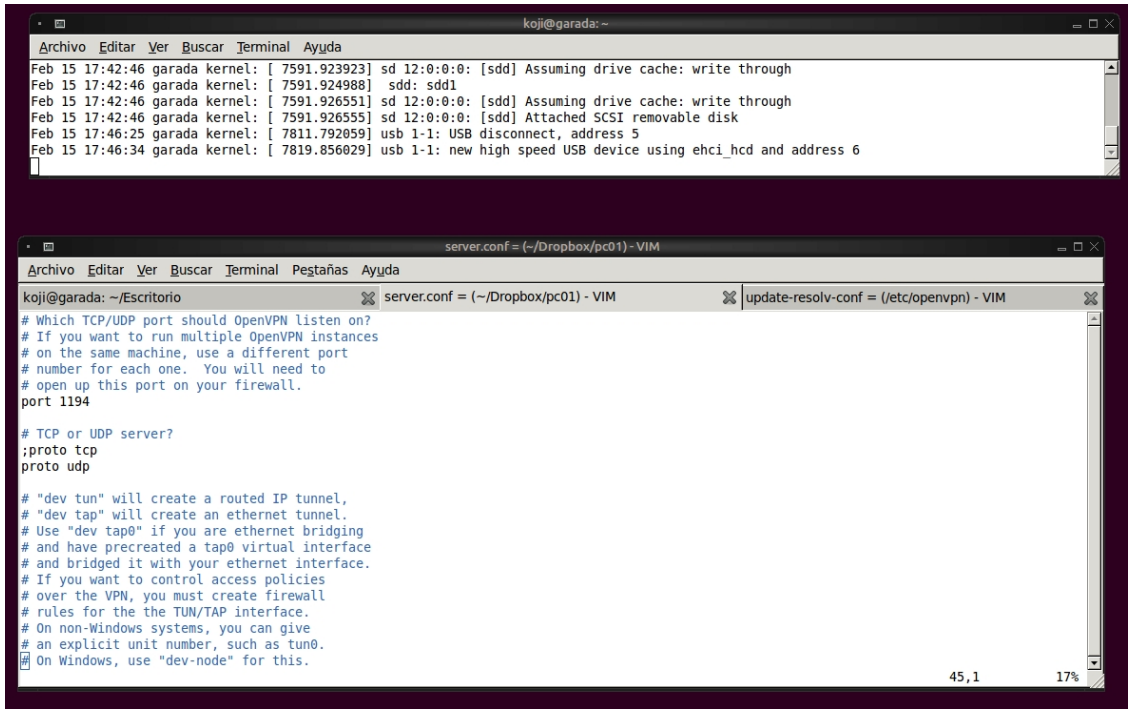
  `tail /var/log/syslog`
  (Muestra las últimas 10 línea)

O mejor aún, si queremos monitorizar continuamente un log, abrimos un terminal exclusivamente para esto y ejecutamos

- `tail -f /var/log/syslog`

(Muestra las últimas líneas, se queda esperando a que haya novedades en el fichero, y cuando las hay, las muestra también)

Si estamos depurando un servicio y tenemos una sesión gráfica, puede ser útil esta disposición del escritorio: dejar un terminal siempre visible, ejecutando `tail -f`, y trabajar en otro terminal con varias pestañas



Si no tenemos gráficos, podemos pulsar Alt F2, Alt F3, etc, abrir una sesión y dedicarla a los logs (también dentro de VirtualBox)

Configuración de rsyslogd

Los logs los genera el demonio `rsyslogd`
En debian/ubuntu:

- Se configura en `/etc/rsyslog.conf`
- A su vez este fichero llama (en orden alfabético) a todos los ficheros `/etc/rsyslog.d/*.conf`
  Ante dos instrucciones contradictorias, la última prevalece sobre la primera

Cada mensaje de log se clasifica asignándole una prioridad

Nivel		Descripción
0	emerg	Sistema inusable
1	alert	Se requiere acción inmediata
2	crit	Suceso crítico
3	err	Error
4	warning	Advertencia
5	notice	Suceso normal pero relevante
6	info	Informativo
7	debug	Mensaje para depuración

Un suceso de *mayor prioridad* es el que tiene un nivel más bajo

Cada mensaje contiene, entre otros elementos

- Fecha y hora
- Origen del mensaje (*facility*): un demonio, el usuario o el núcleo:
  auth (obsoleto), authpriv, cron, daemon, kern, lpr, mail, mark, news, syslog, user, uucp, local0, local1... local7

Si un mensaje se repite muchas veces consecutivas, en el log aparece solamente una, añadiendo la información *mensaje repetido n veces*

En los ficheros de configuración se indica qué hacer con cada log

`origen.prioridad  destino`

Ejemplos:

- `kern.*  -/var/log/kern.log`
  Enviar todos los logs del núcleo al fichero `/var/log/kern.log`
  Es necesario especificar el trayecto absoluto del fichero. El guión que precede al nombre especifica que no se escriba inmediatamente, sino usando una caché
- `mail.err  /var/log/mail.err`
  Enviar al fichero `mail.err` todos los logs de correo que tengan una prioridad `err` o superior
- `*.err  *`
  Enviar todos los mensajes con prioridad `err` o superior a todos los usuarios que tengan una sesión en la máquina
  Si se omite el asterisco el significado es el mismo: *todos los usuarios*

- `*.=debug ~`
  Descarta todos los mensajes cuya prioridad sea igual a *debug*. El signo de igual antes de un nivel indica que nos referimos a los mensajes de exactamente ese nivel. La virgulilla descarta los mensajes

Si en una línea enviamos un log a un destino, y en la siguiente línea a otro destino, se envía a ambos

- A menos que el primer destino sea la virgulilla. En tal caso el log no se sigue procesando

Es muy recomendable que cada vez que editemos el fichero de configuración, revisemos su sintaxis

- Para ello, ejecutamos

  ```
  /usr/sbin/rsyslogd -c5 -N1
  ```

  (Un error advirtiendo de que falta `xconsole` no tiene importancia)

Ejemplo de fichero de configuración

```
auth,authpriv.* /var/log/auth.log
*.*;auth,authpriv.none -/var/log/syslog
#cron.* /var/log/cron.log
#daemon.* -/var/log/daemon.log
kern.* -/var/log/kern.log
#lpr.* -/var/log/lpr.log
mail.* -/var/log/mail.log
#user.* -/var/log/user.log
#mail.info -/var/log/mail.info
#mail.warn -/var/log/mail.warn
mail.err /var/log/mail.err

#*.=debug;\
# auth,authpriv.none;\
# news.none;mail.none -/var/log/debug
#*.=info;*.=notice;*.=warn;\
# auth,authpriv.none;\
# cron,daemon.none;\
# mail,news.none -/var/log/messages
*.emerg *
```

Envío de mensajes por parte del usuario

Disponemos de la orden de shell `logger` para enviar mensajes

- `logger "Mi mensaje de log"`
  Envía este mensaje con origen *user* y prioridad *notice*
- `logger -p local1.debug "Trazando bla bla"`
  Con la opción -p podemos cambiar origen y prioridad

Envío de logs a máquina remota

Podemos enviar los logs a una máquina remota, eso hace mucho más difícil al atacante borrar sus huellas
Veremos aquí un ejemplo muy básico de uso de esta funcionalidad

- Enviaremos los logs en texto claro, sin autenticar ni cifrar.
  Riesgo obvio de suplantación, ataques DOS, no confidencialidad....
  Deberíamos usar el paquete *rsyslog-gnutls*, que usa TLS
- Guardaremos los logs de las máquinas remotas mezclados con los de la máquina local
  En entornos poco exigentes, podríamos consultarlo *a ojo* o mediante algunos scripts. Mejor sería usar conjuntos de reglas para escribir distintos grupos de logs en distintos ficheros.
  Mejor aún, almacenarlos en una base de datos MySQL (con el módulo *ommysql* de rsyslogd , paquete *dsyslog-module-mysql* en debian)

Configuración en el cliente

Para indicar en el cliente que envíe ciertos logs a máquina remota

- En cualquier fichero con extensión `.conf` dentro del directorio `/etc/rsyslog.d`
  Añadimos la línea
  `facility.nivel  @<IP_DEL_SERVIDOR>`
  P.e.
  `*.*  @10.0.2.12`

Configuración en el servidor

- La forma tradicional empleada por `sysklogd` era lanzar el demonio con la opción `-r` añadiendo esta opción en `/etc/default/rsyslog`

Con `rsyslogd`

1. Editamos el fichero `/etc/rsyslog.conf`
2. Descomentamos las líneas

```
#$ModLoad imudp
#$UDPServerRun 514

#$ModLoad imtcp
#$InputTCPServerRun 514
```

3. Detenemos y reiniciamos el demonio (no sirve hacer *reload*)
4. Comprobamos con `netstat` que está en marcha escuchando en el puerto correspondiente (514 por omisión, TCP o UDP)

Funcionalidad adicional

Con `rsyslog` también es posible

- Cambiar el formato de los logs

Adicionalmente, *logrotate* permite

- Especificar con qué frecuencia se borran los ficheros, que tamaño alcanzan, cómo guardarlos comprimidos, dónde y cuando enviarlos por correo...