Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

Metalog, la herramienta para nuestros logs. (16325 lectures)

Per **Celso González**, ***PerroVerd*** (http://mitago.net)

Creado el 28/01/2003 17:29 modificado el 28/01/2003 17:29

Metalog es un reemplazo moderno, eficaz y sencillo de syslogd y klogd, nos permite registrar los mensajes en función de su facility, nivel, programa que lo genera e incluso con expresiones regulares de Perl. En este artículo explico como funciona y como configurarlo a nuestro gusto.

Instalación

Como siempre lo haremos a la debian way, si usas rpm también es muy fácil y siempre tienes la opción de bajarte el tar.gz desde la página oficial[1] del programa.

```
apt-get install metalog
apt-get remove syslogd klogd
```

Lo de quitar el syslogd y el klogd puede esperar al final, solo depende del cariño que le tengas a tus logs :)

Configuración

La propia instalación deja un fichero de configuración bastante majo y fácil de entender en /etc/metalog.conf
El formato del fichero es una serie de bloques con la siguiente estructura:

Título: (es para ayudarnos a editar el fichero, no sirve para nada más)

Filtros: una o más líneas indicando que es lo que queremos filtrar, las opciones de filtrado son: facility, neg-facility, minimum, program, prog_regex, regex y neg-regex.

Acción: acción que se realizará, normalmente guardar en un fichero aunque también podemos indicar que ejecute un programa. Las opciones son: logdir y command.

Veamos ahora una serie de ejemplos

- **Filtrado por facility**
  La facility (alguién que me de una buena traducción de esto) es un nivel de agrupación que da el programa que genera el mensaje. Por ejemplo, un programa de correo normalmente generará un mensaje con la facility "mail". Los nombres de facility disponibles son: auth", "authpriv", "cron", "daemon", "ftp", "kern", "lpr", "mail", "news", "security", "syslog", "user", "uucp", "local0" a "local7" Veamos por ejemplo como registrar todos los mensajes del kernel en el directorio /var/log/kern (el kernel usa la facility "kern")

  ```
  Mensajes del kernel :
  facility = "kern"
  logdir   = "/var/log/kern"
  ```

  Podemos usar más de una facility a la vez, por ejemplo registremos los mensajes de "auth" y "authpriv" en el mismo log.

  ```
  Autentificaciones :
  facility = "auth"
  facility = "authpriv"
  ```

```
logdir   = "/var/log/auth"
```

Si en vez de facility usamos neg-facility, filtraremos todos los mensajes que no pertenezcan a esa facility. Por ejemplo, registremos todos los mensajes que no sean de un "daemon"

```
No daemons :
neg-facility = "daemon"
logdir       = "/var/log/nodaemons"
```

Realmente no se me ocurre una forma útil de emplear esta última opción, pero aquí está.

- **Filtrado por nivel**
  Los mensajes suelen venir con un nivel de importancia entre 1 y 7. Un mensaje de importancia 1 quiere decir que algo gordo está pasando, en cambio un mensaje de nivel 7 seguramente sea un mensaje de debug solo para desarrolladores. Podemos aprovechar esta caracteristica para filtrar. Veamos como filtrar los de nivel 1 o 2 y registralos.

```
Mensajes importantes:
minimum = 2
logdir  = "/var/log/importante"
```

Tambien podemos juntar filtros por ejemplo para ver los mensajes de nivel 1 que vengan del kernel.

```
Mensajes que acojonan:
facility = "kern"
minimum  = 1
logdir   = "/var/log/critico"
```

- **Filtrado por programa**
  Una vuelta de tuerca más, por ejemplo yo puedo saber los mensajes generados por programas con facility "mail" pero me interesa que los mensajes de mi gestor de correo (qmail) vayan registrados aparte. Para esto hago lo siguiente:

```
Cosas del qmail:
program = "qmail"
logdir  = "/var/log/qmail"
```

- **Filtrado por programa usando expresiones regulares**
  Hay programas que estan divididos en pequeños subprogramas, por ejemplo, el servidor de correo postfix utiliza postfix/smtp, postfix/smtpd, postfix/cleanup... Para registrar esto tenemos 2 opciones, la primera usando el sistema anterior

```
Postfix en plan chapucero:
program = "postfix/smtp"
program = "postfix/smtpd"
program = "postfix/cleanup"
logdir  = "/var/log/postfix"
```

Utilizando, expresiones regulares para elegir el programa

```
Postfix bien hecho:
prog_regex = "postfix"
logdir     = "/var/log/postfix"
```

Como se puede comprobar la segunda opción es mucho más clara, sencilla y fácil de mantener

- **Filtrado por contenido del mensaje**
  Esta me encanta :) aporta una flexibilidad increible y es asquerosamente fácil de usar.
  Con este sistema conseguimos filtrar si el contenido del mensaje encaja con la expresión regular que empleamos. El mejor caso que se me ocurre es para filtrar datos de un firewall usando iptables.
  Todos los mensajes de este palo tienen una línea que dice algo así como "IN=eth0 OUT=", añadimos una regla al archivo de configuración y desde ahora podemos guardar esta información en el directorio /var/log/firewall

```
Firewall:
regex  = "IN=eth0 OUT="
logdir = "/var/log/firewall"
```

Lo mejor es revisar el archivo de configuración de ejemplo y ver alguna de las muestras que hay, seguro que se os ocurre alguna cosa que quereis guardar aparte.

- **Otros parámetros de configuración**
  En el archivo de configuración tenemos 3 parámetros más que nos afectan, estos parámetros son maxsize, maxfiles y maxtime. Estas opciones afectan a la rotación de los ficheros de logs (casi nos podemos olvidar del logrotate también).

  maxsize: es el tamaño máximo del fichero antes de ser archivado (en bytes)
  maxtime: es el tiempo máximo en segundos del fichero antes de ser archivado
  maxfiles: es el número de ficheros archivados que queremos guardar.
  Los valores por omisión que aparecen el el archivo son más que suficientes.

Ficheros de log

Tal como habeis podido ver metalog no guarda la información en un fichero, hay que pasarle un directorio. Dentro de este directorio encontraremos los siguientes ficheros:

current: El más importante, es donde se van guardando los datos actuales.
.timestamp:> fichero interno que se emplea para comprobar la antiguedad del archivo current (tema de rotaciones)
log-año-mes-día-hora:minutos:segundos: Histórico de currents más antiguos, el tamaño y el número de ellos depende de los parámetros de configuración de rotación.

No puedo hacer tail -f /var/log/algo/current

Una de las principales ventajas del metalog es que no escribe la información a disco directamente, la va guardando en memoria y cuando tiene suficiente lo escupe al disco. Esta forma de funcionar es bastante buena ya que los accesos a disco se reducen y el consumo de energía también. De todas formas, si queremos ver algun log "online", tal como haciamos antes tenemos una solución:

```
#killall -USR1 metalog
```

Enviando la señal USR1 al programa conseguimos que se comporte como el syslog tradicional. Una vez terminadas nuestras pruebas podemos enviar la señal USR2 para que vuelva a funcionar con su modo nativo. Empleo un killall ya que metalog abre 2 procesos simultaneos metalog MASTER y metalog KERNEL.

Conclusiones

Si tienes un portatil y aún no usas metalog estás desperdiciando tu batería, y si no tienes un portatil creo que la facilidad para registrar mensajes con el metalog es una buena razón para cambiarte.

Lista de enlaces de este artículo:

1. http://metalog.sourceforge.net

E-mail del autor: celso _ARROBA_ mitago.net
Podrás encontrar este artículo e información adicional en: http://bulma.net/body.phtml?nIdNoticia=1676