

Configuración y uso de *fetchmail*

por Alejandro Moreno
amperisblog[@]gmail.com
<http://www.amperisblog.com>

14 de septiembre 2009

Introducción

Fetchmail es un utilidad Linux para la recuperación y reenvío de correos electrónicos desde un **mailserver** a la **maquina local** donde se ejecuta. Es capaz de acceder a buzones POP o IMAP bajo diferentes métodos de autenticación.

Puede funcionar de dos maneras diferentes:

1. **modo sigledrop**: todos los correos del buzón accedido por Fetchmail van a un mismo destinatario, ya sea el usuario de la maquina local donde se ejecuta Fetchmail u otro que deberá ser especificado en la configuración.
2. **modo multidrop**: todos los correos del buzón accedido por Fetchmail pueden ir a multitud de usuarios diferentes. Por este motivo Fetchmail debe buscar dentro de los encabezados del correo el destinatario real. Su trabajo es muy parecido al de un MTA pero tenemos que tener presente que al descargar el correo bajo POP/IMAP es posible que Fetchmail no encuentre correctamente el destinatario. Esto nos puede pasar por ejemplo con correos electrónicos de listas o correos con copias ocultas.
Los protocolos POP/IMAP no fueron pensado para enrutar correo como puede ser el SMTP.

Fetchmail sigue el siguiente orden para averiguar quien es el origen del mensaje:

```
Return-Path:  
Resent-Sender:  
Sender:  
Resent-From:  
From:  
Reply-To:  
Apparently-From:
```

Fetchmail es una utilidad de linea de comandos que puede funcionar como daemon leyendo su archivo de configuración que generalmente se encuentra en `~/.fetchmailrc`.

Instalación

Fetchmail se puede instalar directamente desde cualquier herramienta de actualización automatica (apt o yum). También es posible descargas su ultima versión, la 6.3.11, desde la Web del autor (<http://fetchmail.berlios.de>).

```
# apt-get install fetchmail (para Debian's)  
# yum install fetchmail (para RedHat's)  
# touch ~/.fetchmailrc  
# chmod 0700 ~/.fetchmailrc
```

Una vez instalado es posible que no tengamos el `~/.fetchmailrc`. Si no lo tenemos lo creamos como una archivo de texto vacío con permisos 0700.

Una vez instalado podemos hacer un `“fetchmail -V”` para comprobar tanto la versión como la configuración actual que tiene. La configuración actual lógicamente está establecida dentro del archivo de configuración.

Configuración

Empezamos escribiendo el archivo de configuración de Fetchmail que básicamente consiste en decirle cuales son los buzones (POP o IMAP) a los que debe acceder. Un ejemplo de configuración sería algo como así:

```
set logfile "/var/log/fetchmail.log"
poll pop.miserver.com
    proto pop3
    local miserver.com
    user "buzon@miserver.com"
    pass "1234"
```

El archivo de configuración se compone de dos partes, la configuración general y la configuración de cada uno de los servidores donde debemos acceder.

Opciones de la configuración general:

1. **set logfile:** especifica donde se encuentra el archivo de log. Esto nos sera muy útil cuando lancemos la utilidad en modo debug.
2. **set postmaster:** define la dirección de correo para enviar cualquier correo que Fetchmail no sepa enrutar. Por defecto será "postmaster" dentro de la máquina local.
3. **set bouncemail:** reenvía cualquier error de entrega al emisor si su correo no se puede entregar (por defecto).
4. **set no bouncemail:** reenvía cualquier error de entrega al postmaster.

Opciones de configuración de los buzones:

1. **via | poll:** especifica la IP o el nombre DNS del servidor donde se encuentra el buzón.
2. **proto:** especifica el protocolo para acceder al buzón. Puede ser: POP2, POP3, IMAP, APOP, KPOP.
3. **local:** especifica los dominios que deben considerarse como locales.
4. **timeout:** especifica un timeout. Pasado este timeout se cancela el acceso y se espera hasta el siguiente. Por defecto son 300 segundos.
5. **user:** especifica el nombre de usuario de acceso al buzón.
6. **pass:** especifica la contraseña de acceso al buzón.
7. **ssl:** se conecta al servidor bajo el protocolo seguro ssl.
8. **sslcertck:** obligamos a fetchmail a que compruebe la validez del certificado del servidor con el conjunto de certificados de la maquina local. Ver la opción sslcertpath.
9. **sslcertpath:** indica la ruta en donde se encuentran instalados los certificados locales.
10. **tracepolls:** añade una cabecera de Received en el correo electrónico para indicar que ese correo a pasado por Fetchmail antes de ser entregado.
11. **smtphost:** indica uno o varios host a los cuales serán reenviados los correos. Por defecto es localhost.
12. **preconnect | postconnect:** programas o scripts que queremos ejecutar antes | después de realizar el fetch de los correos.
13. **uidl:** si utilizamos POP3, el correo una vez descargado por el fetchmail permanecerá en el servidor.
14. **no uidl:** si utilizamos POP3, una vez descargado el correo del servidor, esté será eliminado. Es el funcionamiento por defecto.
15. **fetchall:** descargamos todos los mensajes se hayan visto o no.
16. **no fetchall:** descarga solo los mensajes nuevos. Por defecto.

17. **fetchlimit**: número máximo de mensajes que podemos descargar durante una conexión con el servidor.
18. **flush**: una vez descargados los mensajes del servidor, estos son borrados.
19. **defaults**: establece una serie de opciones como por defecto para todos los servidores.
20. **to | is**: indica el usuario local para el cual son los correos.
21. **envelope**: especifica el nombre original que tiene la cabecera de envelope.

Ejemplos

+ El usuario local *asanchez* accede a su cuenta de correo de Yahoo vía POP3.

```
defaults
  flush
  fetchall
  proto pop3
poll pop.yahoo.com
  user asanchez@yahoo.com
  pass 123456
  to asanchez
```

+ La cuenta de correo backup@gmail.com se utiliza como backup de correo de nuestro servidor Postfix. Los correos de esta cuenta deben enrutarse a cada uno de los buzones de sus destinatarios.

```
set logfile "/var/log/fetchmail.log"
set spambounce
```

```
poll pop.gmail.com
  proto pop3
  no dns
  local midominio.com
  user "backup@gmail.com"
  pass "123456"
  ssl sslcertck
  sslcertpath "/root/.certs"
  is *
  fetchall
```

+ El usuario *asanchez* utiliza fetchmail para acceder a su buzón, pero tiene problemas con las listas de distribución ya que no le llega ningún correo.

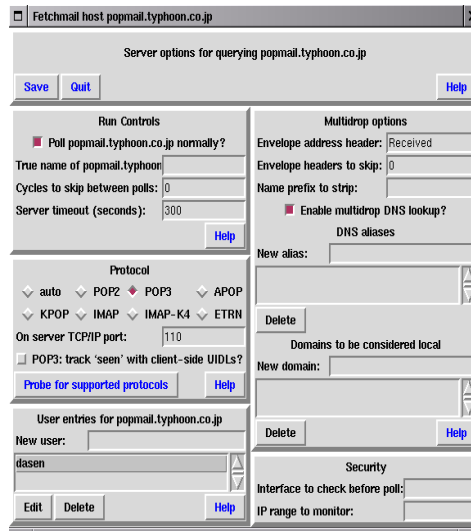
Este problema es normal cuando fetchmail no sabe identificar exactamente al destinatario del mensaje. En este caso vemos como en el campo To: encontramos algo como lista-zimbra@google.com que realmente no es el usuario final. El usuario final del mensaje lo encontramos en una cabecera no estandar del protocolo SMTP llamada "X-Envelope-to". Dado que esta cabecera no es estandar fetchmail necesita saber como se llama en caso de tener que buscar el destinatario final. En el caso que se le presenta a *asanchez*, no encuentra el destinatario porque su ISP llama a esta cabecera con el nombre de X-Original-to.

Debemos indicárselo a fetchmail de la siguiente forma:

```
poll mail.servermail.uk
  envelope "X-Original-To"
  local servermail.uk
  user "info@servermail.uk"
  pass "12345"
```

Ayuda para la escritura de la configuración

Existe una utilidad GUI llamada `fetchmailconf` que nos permite escribir toda esta configuración de una manera mucho más fácil sin saber como funcionan cada una de las opciones.



Arrancar el servicio

Una vez configurado nuestro `~/fetchmailrc` es hora de arrancar el daemon que cada cierto tiempo irá leyendo de cada uno de los buzones que hayamos configurado. Explicaré tres comandos básicos: el inicio en modo debug, el inicio normal y la parada.

En el inicio en modo debug debemos utilizar el parámetro `-v`. El parámetro `-d` indica cada cuanto tiempo en segundos debemos acceder al buzón. Lógicamente si estamos en modo debug podremos unos pocos minutos, de lo contrario tendremos que esperar bastante si queremos que el buzón vuelva a ser accedido.

```
# fetchmail -d XXX -v
```

El el arranque normal simplemente utilizaremos el parámetros `-d`.

```
# fetchmail -d XXX
```

Para la parada del daemon utilizaremos el parámetro `-q`.

```
# fetchmail -q
```

Podemos comprobar que el daemon está funcionando mirando los procesos:

```
# ps -x | grep fetchmail
25435      ?          Ss          0:00  fetchmail -d 60 -v
```

Para los más perezosos es posible crear un script de arranque y parada de fetchmail dentro de `/etc/init.d`:

```
# cat fetchmail
#!/bin/sh

case "$1" in
```

```
'start')
    echo -n "Starting fetchmail..."
    fetchmail -d 60 -v
    touch /var/lock/subsys/fetchmail
    ;;
'stop')
    echo -n "Shutting down fetchmail..."
    fetchmail -q
    rm -f /var/lock/subsys/fetchmail
    ;;
'restart')
    echo -n "Restarting fetchmail..."
    fetchmail -q
    fetchmail -d 60 -v
    ;;
*)
    echo -n "Usage: $0 { start | stop | restart }"
    ;;
esac

echo
exit 0
```

Más información

- Manual oficial de fetchmail <http://fetchmail.berlios.de/fetchmail-man.html>
- FAQ de fetchmail <http://fetchmail.berlios.de/fetchmail-FAQ.html>