



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

Howto: Video vigilancia en linux (19716 lectures)

Per Edu Herraiz, [shark](http://shark.is-a-geek.com) (<http://shark.is-a-geek.com>)

Creado el 19/04/2007 20:03 modificado el 19/04/2007 20:03

*Este howto expone como **implementar un sistema de video vigilancia** sobre linux. Después de utilizar el [script de presencia](#) para kbluetooth de gallir, para conectar y desconectar las sesiones del kopete, se me ocurrió la idea de utilizarlo para activar y desactivar un sistema de vigilancia **con una webcam**. Así, mediante el **bluetooth de un móvil** que siempre llevo encima, mi ordenador sabe cuando me he ido de casa y activa la vigilancia en mi habitación. Cuando vuelvo, me reconoce y la desactiva.*

*La aplicación que he utilizado para tratar las imágenes y la detección de movimiento es **motion**, esta se relaciona con la webcam, cuando detecta un movimiento en las imágenes produce un nuevo evento. Lo que hace cuando salta un evento es muy configurable, puedes **capturar las imágenes**, producir un video con las imágenes y lo más emocionante, **ejecutar lo que quieras**. Gracias a esto propongo un script de atención a un evento.*

En el script se ejecuta un sonido aleatorio entre 10 posibles, al cual más molesto. No es mi intención que lo utilices si te parece ruidoso o no adecuado, más bien dar a conocer **la posibilidad de poder ejecutar un script**.

Voy a empezar contando un poco como hacer funcionar una webcam en linux, después explicaré [el funcionamiento y configuración de motion](#), para acabar con la parte [del script de atención a evento y del script de presencia en kbluetoothD](#).

Para la receta he utilizado los siguientes ingredientes:

Hard

- Un [ordenador](#)
- Una [webcam compatible](#) con linux
- Un [adaptador bluetooth usb](#)
- [Un móvil](#) con bluetooth

Soft

- [Motion](#)
- [Script de presencia](#) en kbluetoothD (by gallir), modificado
- Script de atención a evento

Webcam en linux

Yo utilizo una webcam Logitech de las más baratas, que utiliza el **driver** [spca5xx](#): descargar y instalar según las instrucciones del propio driver.

Para probar el funcionamiento de la webcam utilicé **gqcam**, un **visualizador** empaquetado en debian, al que únicamente hay que pasarle el parámetro para que visualice del dispositivo deseado (puede variar):

```
gqcam -v /dev/video0
```



[Cualquier webcam](#) que consigas hacer funcionar bajo linux te servirá para la video vigilancia.

Motion

Motion es el encargado de muchas cosas en este sistema, **procesa las imágenes** recibidas de la webcam, detecta si ha habido variaciones entre dos consecutivas y si esas variaciones alcanzan un "valor" lanzan un evento. Después guarda esas imágenes, y hace un video con ellas. Además he hecho que lance un script cuando se produce un evento. He encontrado algunos sitios donde hablan de motion [1] [2], [también en bulma](#), aunque lo mejor para profundizar es la [propia guía de motion](#).

Motion esta **empaquetado en .deb** i en .rpm, y tiene las siguientes **dependencias**:

Librerías necesarias	Paquetes RPM	Paquetes Debian
libm, libresolv, libdl, libpthread, libc, ld-linux, libcrypt, and libnsl	glibc glibc-devel	libc6 libc6-dev libgl1.2
libjpeg	libjpeg libjpeg-devel	libjpeg62 libjpeg62-dev (opcional libjpeg-mmx-dev)
libz	zlib zlib-devel	zlib1g zlib1g-dev

Una vez **instalado motion desde apt** o como se prefiera, vamos a configurarlo, para explicar como, comentaré todo el **archivo de configuración**.

/etc/motion/motion.conf

```
daemon on
quiet on
```

Estas dos opciones están en off por defecto, pero es conveniente ponerlas a on. La primera hace que se ejecute motion como daemon, obligatorio para que después funcione correctamente con los scripts (de presencia y de atención a evento). Y la segunda es para que no suelte unos "beeps" cuando detecte un movimiento.

```
videodevice /dev/video0
```

Le decimos cual es el dispositivo de la webcam. (puede variar)

```
width 320
height 240
```

Tamaño de frame, hay algunas webcams que soportan 640x480 o incluso más, pero la mía no. Piensa que si pones un tamaño muy grande de frame, las imágenes pesan más, y la larga, si hay mucho movimiento, puede ser problemático.

```
framerate 25
```

Numero máximo de frames que podrá capturar la cámara por segundo. Un valor muy grande, problemático de nuevo, por lo mismo que antes.

```
quality 85
```

Calidad de los frames en tanto por ciento, para los jpg's que guarda. 85 es el valor por defecto.

```
auto_brightness off
```

Puede ser una opción para los que tengan una webcam con auto brightness, esta opción regula automáticamente la luminosidad de las imagenes. Lo probé, pero daba saltos de casi blanco a casi negro. Aunque seguramente mi webcam no lo soporte.

```
threshold 4500
noise_level 64
```

Threshold y noise_level son parámetros que utiliza motion para calibrar la variación (en las imágenes) necesaria para lanzar un evento.

Threshold representa el número de pixeles que tienen que diferir de la anterior captura después del proceso al cual somete a las imágenes. El valor de 4500 entiendo que es equivalente a decir que la imagen difiera en un 6% aprox.



respecto de la anterior.

Noise_level es la cantidad de variación que se necesita, para que un pixel sea contado en el threshold.

Existen más parámetros de este tipo (Masking y despeckle).

```
brightness 200
contrast 80
saturation 0
hue 0
```

Parámetros (por orden) de luminosidad, contraste, saturación y tono, serán variables dependiendo de donde este colocada la webcam, yo coloqué la webcam y haciendo prueba/error saqué esos valores.

```
ffmpeg_cap_new on
ffmpeg_video_codec mpeg4
```

El primero para usar la librería ffmpeg para encodear al vuelo los videos, a partir de las imágenes. Y el segundo para escoger el codec con el que comprimirá el video.

```
target_dir /home/user/Desktop/Video_Vigilancia
```

Directorio donde guardar las imágenes y los videos producidos. (Comprobar permisos de escritura)

```
webcam_port 0
webcam_localhost on
webcam_quality 50
webcam_maxrate 8
```

Los parámetros webcam_* hacen referencia a la posibilidad de habilitar un puerto para acceder con un navegador a lo que esta viendo la webcam.

%_port es el número del puerto que queremos asignarle, si el puerto es 0, la opción queda deshabilitada.

%_localhost es para limitar el acceso. Si on, exclusivamente podrá acceder localhost.

%_quality es la calidad de los frames enviados al web. Si vas usar la opción webcam y tienes un ancho de banda limitado, no te pases con este valor.

%_maxrate, cuando no ha detectado movimiento, envía solamente 1 frame por segundo al web. En caso contrario, envía como máximo este valor.

```
on_event_start sh /etc/motion/att-evento
# on_event_end sh /etc/motion/end
```

Una de las mejores, cuando detecta un movimiento, ejecuta el valor, aquí lo uso para ejecutar el script de atención a evento. También existe la posibilidad de ejecutar algo cuando ya acabado ese evento.

```
gap 20
```

Es el tiempo en segundos que queremos que espere desde que detecta el último movimiento, para que de por finalizado el evento, y pase a otro. El valor por defecto es 60. Nota: Para cada evento nos crea un video diferente, un valor como 20, es muy bajo y se producen gran cantidad de videos, pero así consigo que pase corto tiempo entre dos eventos eventualmente diferentes y poder atender los dos con el script.

```
post_capture 5
pre_capture 5
```

Número de frames que queremos capturar antes y después de la detección.

Podeis probar ejecutando...

```
motion -n
```

...si funciona todo correctamente, el parametro -n es para que motion arranque en modo non-daemon.

Script de atención a evento

En este apartado os pondré mi cutre-script, aunque en lugar de reproducir sonidos, seguramente se os podrán ocurrir mil variaciones, como por ejemplo: enviar un sms, reproducir video, enviar un email o porque no, [lanzar misiles](#).

```
#!/bin/sh
SOUNDS="/etc/motion/sonidos"
```



```
rango=10
numero=$RANDOM
let "numero %= $rango"

case $numero in
    0) play $SOUNDS/sirenas.mp3;;
    1) play $SOUNDS/horror1.mp3;;
    2) play $SOUNDS/horror2.mp3;;
    3) play $SOUNDS/horror3.mp3;;
    4) play $SOUNDS/horror4.mp3;;
    5) play $SOUNDS/horror5.mp3;;
    6) play $SOUNDS/grito1.mp3;;
    7) play $SOUNDS/grito2.mp3;;
    8) play $SOUNDS/disparos.mp3;;
    9) play $SOUNDS/grito2.mp3;;
esac
```

Aunque seguramente se pueda hacer mejor, este script ejecuta aleatoriamente uno de [estos mp3 \(bajar y descomprimir en /etc/motion\)](#).

Script de presencia

Con el script de presencia consigo activar motion automáticamente cuando me voy de casa y que se desactive cuando llego. Este script difiere muy poco [del original](#) que publicó [Ricardo Galli](#). No voy a explicar la configuración de Kdebluetooth porque creo que Ricardo lo explica suficiente, lo único, comentar que yo he bajado el tiempo de búsqueda de dispositivos de Kdebluetooth a 1 minuto para adelantar mi detección un poco.

```
#!/bin/sh

# This script connect/disconnect all Kopete sesiones
# and execute motion for a capture
# events cam, when the selected device is
# visible/invisible.
# Author: Ricardo Galli (from Custom_Job) Modified: Shark
#
# This script will be called with one single parameter:
# "run": The script should be executed
# "configure": Lets the user configure the script.

case "$1" in
    run)
        if [ "$FOUND_DEVICES" != "x" ] ; then
            killall motion
            kdialog --passivepopup "Desactivada" 2 --title "Video Vigilancia"
            dcop kopete KopeteIface connectAll || /bin/true
            kdialog --passivepopup "Conectando" 2 --title "Kopete"

            elif [ "$LOST_DEVICES" != "x" ] ; then
                dcop kopete KopeteIface disconnectAll || /bin/true
                #dcop kmail MainApplication-Interface quit || /bin/true
                kdialog --passivepopup "Desconectado" 2 --title "Kopete"
                kdialog --passivepopup "Activada" 2 --title "Video Vigilancia"
                motion
            fi
        ;;

    configure)
        # Simply run kedit let the user edit this script.
        # A more sophisticated solution might use kdialog to hide
        # the shell script from the user
        kedit "$JOB_PATH"
        ;;
esac
```



Conclusiones

Motion es una graciosa aplicación que recomiendo probeis si teneis ocasión, puede ser muy útil para algunos, estoy seguro.

Me he divertido muchísimo, tanto probando y instalando el sistema, como riendome de las caras que ponen "los intrusos" de mí habitación. Solo decir gracias a todos aquellos que me ayudaron a escribir esto, y quedo a [vuestra disposición](#) para cualquier consulta/crítica/modificación/mejora que pueda merecer este howto.

E-mail del autor: imshark _ARROBA_ telefonica.net

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=2384>