

Alex Nikolaidis

How to make a DIY home alarm system with a raspberry pi and a webcam

Convert a simple webcam to

a fancy digital peephole viewer with motion detection features

Traditional wireless CCTV cameras are cheap but anyone with a wireless receiver can view your signal. On the other hand, IP cameras are secure but they can be quite expensive and usually the video quality is poor—unless you go for a really expensive model.

Lately I wanted to install a home surveillance system so I chose to use a cheap Logitech webcam with Raspberry Pi and motion, an excellent linux program that monitors video signal for changes and triggers events.

Here's what you can achieve with a similar setup:

- ▣ Setup RaspPi as a 24/7 **Web cam server** and stream your video over the internet. You can also view your signal remotely, using any mobile device equipped with a browser.
- ▣ Setup **motion detection** and trigger any events you like, such as store images when motion is detected, upload the images to a remote FTP server, send a notification to your computer, receive an SMS—basically run any script you like!

Ok, so here's how you do it:

Necessary hardware:

1. Raspberry Pi Model B Revision 2.0 (512MB)
 2. Logitech HD Webcam C270 or a similarly compatible usb webcam ([list of compatible webcams here](#)).
 3. A usb hub with an external power supply
 4. (optional): a usb extension cable
-

Step #1: Make your webcam stealth

I wanted to hide the camera in an inconspicuous place outside my door, so I removed the webcam's casing. The Logitech C270 is a really good choice for this project as (1) it is 100% compatible with pi, (2) it has a really good 720p HD resolution and (3) it is very very small.

Here's how to make the camera 'stealthy':





Remove the four screws that support the holding bracket.



With the help of a screwdriver carefully lift the top cover.



After you remove the top cover remove the three screws that hold the front of the camera.





Remove the two screws that hold the camera board with the case.



You now have a miniaturized high-definition usb camera

You can now connect your webcam to the usb hub. You have to use an external usb hub with an independent power supply as the raspberry pi cannot power the webcam by itself.





Connect your webcam to a usb hub with its own external power supply.

I ended up hiding the webcam within the door (!), inside the space reserved for an extra key mechanism. The webcam's lens is taped exactly at the key slot so you cannot see it if you are outside.



(just to clarify: as you can see my door is crap! In the case of a burglary it will fall with the lightest blow. I'm better

off replacing the actual lock with a smart security device)

Step #2: Setup your raspberry pi

Your pi needs to boot a linux operating system in order to run motion. The most popular choice is Raspbian, a debian-based OS that is optimized for pi's hardware.

To prepare your SD card and install Raspbian I recommend following Adafruit's excellent tutorials here.

Since you are not going to have your pi connected to a monitor or have a keyboard and mouse, I also recommend enabling Secure Shell (SSH) in your pi so that you can remote control your Raspberry Pi over your local network.

Finally, it is a good thing to force a static IP address so that you can easily find the webcam server even if pi restarts.

To do this, first type from the command prompt:

```
ifconfig
```

This reveals your router information. If you have an ethernet connection check out the *eth0* bit. If you have a wireless connection check out the *wlan0* bit. Make a note of the following info:

inet addr – 192.168.1.5 (pi's IP Address)

Bcast – 192.168.1.255 (broadcast IP range)

Mask – 255.255.255.0 (subnet mask)

then run:

```
route -n
```

and note the following:

Gateway Address – 192.168.1.1

then run the following command to edit the network configuration:

```
sudo nano /etc/network/interfaces
```

and change the following entry from:

```
iface wlan0 inet dhcp
```

to:

```
iface wlan0 inet static  
address 192.168.1.5  
netmask 255.255.255.0  
gateway 192.168.1.1  
network 192.168.1.0  
broadcast 192.168.1.255
```

Press **CTRL** and **X** together to save and exit nano editor.

If you reboot your pi now you should have a static address.

Step #3: Setup motion

First you need to use **rpi-update** to add to your raspbian image the initially-missing UVC support:

```
sudo apt-get install rpi-update
sudo rpi-update
```

Next you need to upgrade your packages:

```
sudo apt-get update
sudo apt-get upgrade
```

Then you can install motion:

```
sudo apt-get install motion
```

Now if you run

```
lsusb
```

you should see your camera listed as a usb device, like so:

```
Bus 001 Device 006: ID 046d:0825 Logitech, Inc.
Webcam C270
```

(If not then perhaps your webcam is not compatible with pi)

Next we proceed to configure motion:

```
sudo nano /etc/motion/motion.conf
```

This is a configuration file where you get to define parameters such as the port which motion will run on, or actions that will be triggered when movement is detected.

Here's a list of the parameters you most likely would want to configure:

- ❏ *daemon*: set to ON to start motion as a daemon service when pi boots,
- ❏ *webcam_localhost*: set to OFF so that you can access motion from other computers,
- ❏ *stream_port*: the port for the video stream (default 8081),
- ❏ *control_localhost*: set to OFF to be able to update parameters remotely via the web config interface,
- ❏ *webcontrol_port*: the port that you will access the web config interface (default 8080),
- ❏ *framerate*: number of frames per second to be captured by the webcam. Warning: setting above 5 fps will hammer your pi's performance!
- ❏ *post_capture*: specify the number of frames to be captured after motion has been detected.

You also need to edit the following file if you want to run motion as a daemon service:

```
sudo nano /etc/default/motion
```

and set `start_motion_daemon` to YES:

```
start_motion_daemon=yes
```

Then start motion by typing:

```
sudo service motion start
```

Wait for about 30 seconds for motion to start and then open the video stream from [VLC](#) or a similar program that can show video streams. If you use VLC player go to File>Open Network and enter the IP address of your pi followed by the `stream_port`, for example:

```
192.168.1.5:8083
```

Step #4: Extra goodies

You can setup *motion* to trigger all sorts of actions when it detects movement.

For example, for extra security (in case someone breaks into your house and steals all your equipment): you can **upload the images to a remote FTP server**.

To do that, edit the `motion.conf` file and add the following external command:

```
On_picture_save wput -B  
ftp://username:password@yourftpserver %f
```

You can also run a shell script and **send a notification to another computer**. I use [growl](#) to send a quick popup alert to my Mac each time motion is detected.

[Growl](#) is a fantastic MacOS-only notification system but luckily there is an excellent python library that you can use to easily access the Growl Network Transport Protocol (or Gntp) and send growl notifications to a Mac. Here's how to do that:

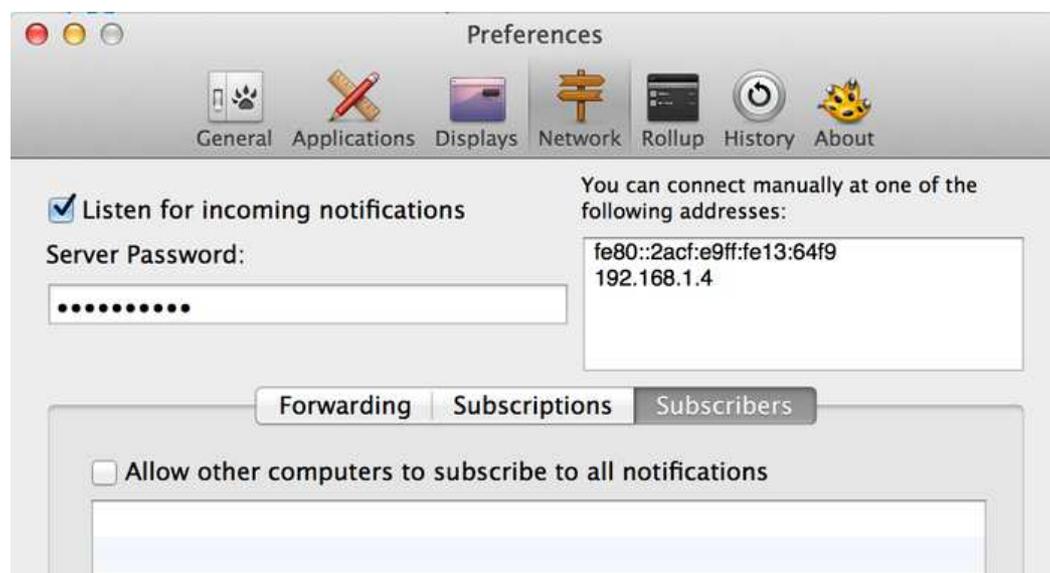
First get '[pip](#)', a tool which allows you to easily install python packages:

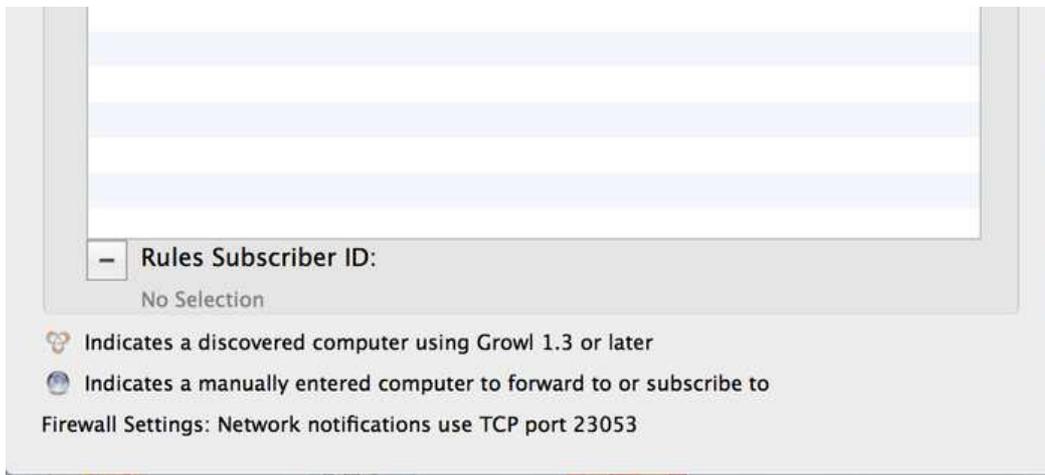
```
apt-get install python-pip
```

then install the [Gntp Python Library](#) by running:

```
pip install gntp
```

then enable incoming network connections in Growl preferences and set a password:





and here's a simple python script that I wrote that sends a notification:

```
# use standard Python logging
import logging
logging.basicConfig(level=logging.INFO)

import gntp.notifier

growl = gntp.notifier.GrowlNotifier(
    applicationName = "Rasp",
    notifications = ["New Updates", "New Messages"],
    defaultNotifications = ["New Messages"],
    hostname = "192.168.1.4", # Here enter your Mac
    IP address
    password = "yourpass" # Here enter your growl
    password
)
growl.register()

# Send one message
growl.notify(
    noteType = "New Messages",
    title = "Camera Alarm",
    description = "Movement detected",
    icon = "https://www.iconfinder.com/icons/48991/download/png/128", #you can optionally define an
```

```
image icon to appear with the notification
    sticky = False,
    priority = 1,
)
```

test the script by running it from pi's SSH command line:

```
python test.py
```

You should see a growl notification in your Mac screen!

To use this script with motion, add the following line to motion.conf:

```
On_motion_detected python test.py
```

Now, each time motion is detected the python script will run and you will see a growl notification in your Mac.



Growl notification at my Mac

You can do all sorts of things with motion. You can even

receive an sms or have Twilio call you whenever the alarm is tripped off!