Bluetooth scanning.

The Bluetooth adapter we used is just a cheap thumbsize unit with a CSR chip inside. To install the Bluetooth stack for Raspbian on the Pi, at the Terminal prompt type:

```
sudo apt-get install bluetooth bluez-utils blueman
```

insert the Bluetooth adapter & scan for devices by typing

```
hcitool scan
```

if it doesn't work, then use

```
hciconfig hci0 up
```

to bring the interface up & then scan again.

The script below will scan for Bluetooth devices when the GPIO4 pin reads 1 (it normally reads 0, until someone trips the PIR).

Here's the shell script:

```
echo "4" > /sys/class/gpio/export
echo "in" > /sys/class/gpio/gpio4/direction

while true;  do
       trap 'echo "4" > /sys/class/gpio/unexport' 0
       stat=`cat /sys/class/gpio/gpio4/value`
while [ $stat = "0" ]
       do


                      x=`hcitool scan --flush`
                      y=${x#*Scanning *...}
                      d=`date +%d%m%y`
                      t=`date +%T`


              echo $d,$t,$y | tr " " "\n"
              echo "--------------------"
              echo $d,$t$y | tr " " "," >> aa$d.txt
              y=''
              stat='1'
       done
done
exit 0
```

Set up Wifi Sniffing on Raspbian build.

```
sudo apt-get install iw tshark
sudo apt-get install subversion
sudo apt-get install libssl-dev
svn co http://svn.aircrack-ng.org/trunk aircrack-ng
cd aircrack-ng
```

```
make
sudo make install
```

```
sudo airmon-ng start wlan0
sudo tshark -i mon0 subtype probereq
```

Some other commands to try:

```
sudo tshark -i mon0 subtype probereq -w /tmp/rpi-cap.pcap
```

```
sudo airodump-ng mon0
```

Scan for WiFi Probe Requests.

Earlier, we showed you how to scan for Bluetooth devices when the alarm is tripped. Many more modern phones have Bluetooth disabled by default now, but these newer Android phones often use WiFi to determine their location quickly, in combination with GPS.

The WiFi chip in a phone uses the same unique Mac style address (00:11:22:AB:CD:EF) as a Bluetooth chipset, so it's possible to record a unique phone identifier. You can see the unique Bluetooth & WiFi Mac addresses of your own phone under the 'Settings → About This Phone' menu.

When WiFi is turned enabled on my phone, it sends probe requests every ten seconds. If I've previously associated successfully with other WiFi networks, then this information is also available, and may give you clues to who they are. This project works fine with the £10 WiPi USB wifi adapter available from cpc.co.uk

If you have the USB WiFi dongle connected you can use the `ifconfig` command to show information.

You should expect to see the adapter listed as `wlan0`. We now need to place the WiFi adapter into Monitor mode.

```
sudo airmon-ng stop wlan0
```

followed by the command

```
sudo airmon-ng start wlan0
```

should produce the `mon0` interface, you can do

```
sudo airodump-ng mon0
```

and see WiFi access points near you. CTRL-C to quit.

To see full probe requests from devices with WiFi enabled do

```
sudo tshark -i mon0 subtype probereq
```

This shows you the manufacturer of the device sending the probe, but it's also possible to just have the complete Mac address without the name resolution. This command records for 60 seconds to a log file:

```
sudo tshark -i mon0 subtype probereq -n -a duration:60 > cap.log
```

you can view the contents of the log file with

```
cat cap.log | more
```

We use the Tshark command in our script to grab the probe requests for 60 seconds, when the PIR alarm is triggered. The script also scans for Bluetooth devices. The script then processes the capture from Tshark to remove any duplicates and just leaves the unique Mac addresses spotted in the final text that are sent in the email to our Gmail account.

```
nano alarmpi-wifi.sh
```

```
echo "4" > /sys/class/gpio/export
echo "in" > /sys/class/gpio/gpio4/direction

while true;  do
       trap 'echo "4" > /sys/class/gpio/unexport' 0
       stat=`cat /sys/class/gpio/gpio4/value`
while [ $stat = "0" ]
       do


                      x=`hcitool scan --flush`
                      y=${x#*Scanning *...}
                      tshark -i mon0 subtype probereq -n -a duration:60 > cap.log
                      egrep -o "[a-z0-9]{2}:[a-z0-9]{2}:[a-z0-9]{2}:[a-z0-9]{2}:[a-z0-9]{2}:[a-z0-
9]{2}" cap.log > cap2.txt
                      sed '/ff:ff:ff:ff:ff:ff/d' cap2.txt > cap3.txt
                      sort -u -o cap4.txt cap3.txt
                      z=`cat cap4.txt`
                      d=`date +%d%m%y`
                      t=`date +%T`


              echo $d,$t,$y,$z | tr " " "\n"
              echo "--------------------"
              echo $d,$t,$y,$z | tr " " "," >> aa$d.txt
              echo -e "Subject: Alarm Alert\r\n\r\n ALERT $t,$y,$z" | mail -s "alarm Detect"
you@gmail.com

              y=''
              z=''
              stat='1'
       done
done
exit 0
```

Then make the file executable with

```
chmod ugo+x alarmpi-wifi.sh
```

Run the scanner command with

```
sudo ./alarmpi-wifi.sh
```