Global Information Assurance Certification Paper

Author: Owen Crow
Date: September 18, 2003
Certification: GSEC
Assignment: 1.4b
Option: 2 Case Study in Information Security

Auditing for Rogue Wireless Access Points in a Large Corporate Campus

1 Abstract

Large corporate campuses provide an extra challenge to security professionals who must secure the corporate IT infrastructure. Thanks to cheap and easy wireless technology, any user can create a breach in your network's perimeter that bypasses all the traditional safeguards such as firewalls and VPNs. This paper attempts to document the special challenges inherent in auditing for rogue wireless access points (WAPs) in a corporate campus with multiple high-rise buildings and with multiple tenants.

2 Table of Contents

3 Before Snapshot

3.1 Introduction

I work for a company with a fair sized campus encompassing multi-story buildings over approximately four city blocks. We also lease space to multiple tenants. We do not allow wireless networks other than a small executive deployment and an occasional prototype deployment. With the advent of cheap and easy to use wireless access points and cards, we developed the need to audit our company's wireless presence to ensure compliance with our security policies. Because we have not deployed a corporate-wide wireless solution, we have limited expertise with wireless and a population of high-tech employees hungry to use it.

3.2 Purpose

The purpose of this paper is to describe how I used freely available software and cheap/used hardware to perform a wireless audit of my corporate campus.

3.3 Scope of the Paper

This paper will be limited to the choices I made for our audit. There are several other methods that can be used to audit wireless, but the underlying principals are the same. I will attempt to justify the choices I made and provide links to more in-depth analysis where needed.

3.4 Goals for the Audit

The primary goal is to verify that no unknown wireless access points are broadcasting from our campus and our network. Because there are several tenants and at least one of them has an extensive wireless deployment, I must be able to weed out their networks.

3.5 Justification

A wireless audit is justified because:

- wireless is extremely easy to install for even novice users
- users do not have the expertise to set up wireless securely
- wireless extends much farther than users or even IT professionals believe possible.

4 During Snapshot

This describes the hardware, software and methods used during the audit.

4.1 Recommended Hardware and Software

I will cover the hardware, software and supporting equipment I use for my audits. Where possible, I will explain what could be improved and where future enhancements will be used.

4.1.1 Linux-compatible Laptop

Because the software solution I use is developed for Linux, I must use a Linux-compatible laptop. Because the budget for this audit was limited, I chose to use a used and fully depreciated laptop from our desktop support. I ended up with an NEC Versa LX with a 300MHz Pentium II processor and 128MB of RAM. This is not a particularly speedy system to use, but was adequate for my needs.

4.1.2 Orinoco Gold Card

As recommended by the “Redhat + Kismet HOWTO” FAQ (http://www.tipsybottle.com/technology/wireless/RedHat-Kismet-HOWTO-FAQ.shtml#What%20type%20of%20wireless%20NIC%20should%20I%20buy) and other sites, I chose the Orinoco Gold 802.11b card. The card supports both 40 and 128-bit WEP and has the added feature of an external antenna

connection. I have found it to be a solid performer and very sensitive even without external antennas. Proxim, the apparent new name for Orinoco, currently only sells 802.11a/b/g or 802.11b/g combo cards which have mixed support under Linux. You should still be able to find the 802.11b Gold card online for bargain prices. Hopefully, the combo cards will be supported in monitor mode before too long.

4.1.3 External Antennas

I purchased an external omni-directional antenna and a "Pringles can" type antenna. Both were relatively inexpensive. I use the omni-directional to boost the detection of systems while canvassing the campus and the directional antenna to attempt to locate a suspect access point from afar. I obtained my directional antenna, the Super Cantenna, from Cantenna.com. I obtained the omni-directional antenna and pigtail cables that match the Orinoco card from JEFA Tech (http://www.jefatech.com).

4.1.4 GPS Receiver

Cost and PC-compatibility proved to be the biggest motivator in my choice of a GPS receiver. Luckily, Garmin just came out with the eTrex (http://www.garmin.com/products/etrex/) receiver that could be bought for less than $160, including a serial adapter and case when I started this project in 2002. I have found it to be reliable and sturdy, which are good attributes to have when juggling expensive equipment in the field.

4.1.5 Kismet and Supporting Software

I should explain why I selected Kismet for my wireless detection system. This mainly had to do with eliminating other solutions that could be bought for little to no cost.

- NetStumbler (http://www.netstumbler.com/): At the time I started the project, the most well known wireless detector was and probably still is NetStumbler. I did download and use NetStumbler (without a GPS receiver at the time) with success. Unfortunately, I soon learned of two pitfalls to NetStumbler that caused me to look elsewhere.
  First, NetStumbler is an active scanner meaning it actively sends out requests for a willing wireless access point to respond. This is fine if the AP is set to respond to requests from any client. Because standard setup for a secure AP is not to respond, we would miss many possible APs.
  Second, because this scanning is active we could be perceived as an attacker against networks we do not own. Not being a lawyer, I preferred to take the more conservative approach of using a passive solution that listens but does not talk.
- Airsnort (http://airsnort.shmoo.com/): I have run Airsnort before, but because of its extra GUI overhead and the older, slower hardware I am using, I chose to use a text-based scanning tool.
- Wellenreiter (http://www.wellenreiter.net/): This seems to be an intriguing and very extensible set of wireless tools based on Perl/gtk. Again,

because it was a GUI application and based on an interpreted language at that, I found it was very slow on my equipment. A new version written in C++ for handhelds only is available as well.

- I must also confess that I prefer simple, text-based tools were possible. I avoid a mouse as much as possible even in a desktop scenario, so having to use a mouse while in the field would be doubly annoying. If you do not have this bias, I would suggest you also check out Wellenreiter and Airsnort.

4.1.5.1 Redhat Linux 9.0 (http://www.redhat.com/)

Redhat Linux 9.0 was chosen mainly because of my familiarity with the distribution and its use in most online solutions and HOWTOs. It is much easier to implement a solution based on someone else's previous experience if possible. If you need help getting it installed, see the Redhat + Kismet HOWTO (http://www.tipsybottle.com/technology/wireless/RedHat8-Kismet-HOWTO.shtml) for a good setup.

4.1.5.2 Orinoco Monitor Mode Patch

The default Orinoco driver in the 2.4 Linux kernel works adequately for regular wireless connections, but in order to sniff for any network traffic and "hop" channels, you need to install the patch available from the fine folks of the Shmoo Group at http://airsnort.shmoo.com/orinocoinfo.html. I spent a lot of time trying to develop a custom kernel based on the stock Redhat kernel, but the sage and simple advice available at the Redhat + Kismet HOWTO includes information on downloading, patching and installing Orinoco-0.13e from source. This is the best method to use.

4.1.5.3 GPSDrive with gpsd

The primary reason to install GPSDrive is to get the gpsd daemon. Gpsd interfaces with a variety of GPS receivers and then provides the coordinates to clients that connect to gpsd. GPSDrive is a wonderful GPS mapping system, but it is overkill on my low-end laptop. It also has the ability to plot detected APs on its display. Compiling and installing GPSDrive is straightforward and covered in the Redhat + Kismet HOWTO.

4.1.5.4 ImageMagick

ImageMagick is necessary to make gpsmap work. Gpsmap is a part of Kismet that plots various information about detected APs on vector and satellite images. Compiling and installing ImageMagick is straightforward and covered in the Redhat + Kismet HOWTO.

4.1.5.5 Kismet

Kismet is the heart of our wireless detection system. It may be bare bones and intimidating if you are used to fully GUI tools. What it lacks in polish it makes up in its flexibility and low overhead. Kismet is a passive scanner that will listen on

all available channels (in succession, not simultaneously) and report what it finds. It supports integration with Snort and Ethereal for wireless IDS and real-time network analysis, if needed. Compiling and installing Kismet is straightforward and covered in the Redhat + Kismet HOWTO.

4.2 System Setup

System setup is covered very well in the Redhat + Kismet HOWTO, but here are some changes I've made to various files:

4.2.1 kismet.conf

Normally installed as /usr/local/etc/kismet.conf, I modified the "logtemplate" line to include the hard coded location for the log files:

```
logtemplate=/var/log/kismet/%n-%d-%i.%l
```

Without this option, you have to make sure you are in the correct directory before you start Kismet and that you have permissions to write to that directory. I wish that there were more options for the date variable (%d) which is formatted as Month-Day-Year. I prefer formats that sort numerically such as Year-Month-Day in where each part is numeric.

4.2.2 kismet_ui.conf

Normally installed as /usr/local/etc/kismet_ui.conf, I modified the "columns" line to include more information:

```
columns=decay,signal,name,type,wep,channel,packets,flags,ip,data,
llc,crypt,weak
```

This change adds counter columns for the packet types and the signal strength column.
I also commented out the "sound_junktraffic" line with a "#" sign as I found it unhelpful and annoying after a while.

4.3 Initial Scan

4.3.1 Planning

Planning mainly involves making sure your wireless detector works correctly. Before you spend a lot of time canvassing your campus, you should verify the following:

- You can detect APs on any channel. Use a known AP and set it to different channels to make sure that Kismet picks up more than one channel. Kismet 3.x automatically hops channels for you. In the past you had to manually start kismet_hopper to make it sample all channels.
- Your gpsd works correctly with your GPS receiver. In general, you have to be outside with a clear view of the sky to verify this one. You can execute gpsd with the "–D 2" option to keep it in the foreground. It will begin printing coordinates when the GPS receiver is working correctly.
- You have a good battery or power supply for the laptop. Dual batteries can last long enough if you are working with a golf cart or you can use a 12V converter in a car for continuous power.

4.3.2 Getting Permission

It is important that you have permission from the powers that be at your site. I would recommend letting any campus security services know what you are doing, why, and that you have permission from your boss.

4.3.3 Scheduling Equipment

For my audit, I requested the use of a golf cart from Facilities. Previously, I used a bike (also provided on campus) or walked, but Houston weather can be a little extreme to tote a laptop and extra equipment for long periods. If you live in a milder area, a bike is sufficient as long as your detection equipment is solid and can take the jarring that goes with the ride.

4.3.4 Basic Steps to Get Started

Now that you have all your equipment together I suggest you follow the order laid out below to verify each part in turn as you begin using your detector.

4.3.4.1 GPS Receiver

Turn this on first and make sure it has a clear view of the sky. The first time you turn the unit on, it can take up to 15 minutes or so to find its location, so it is best turned on first. Using the serial cable for your unit, attach it to your laptop. Make sure the output mode for the PC link is set to the NMEA mode that is easiest for gpsd to understand. See your owner's manual for more information.

4.3.4.2 gpsd

Before you start up gpsd the first time you will probably want to make a soft link from /dev/gps to the serial port to which the GPS receiver is connected. On my system, the command was "ln –s /dev/ttyS0 /dev/gps". Once you have the link in place, starting gpsd should be as simple as typing "gpsd". If it works correctly, it should return immediately without error. You should have gpsd running in the background now. To verify that gpsd is properly receiving data, wait until the GPS receiver says it has a lock on your position. For example, on my eTrex, the display says, "Ready to Navigate." At this point, you should be able to telnet to your localhost and issue the "p" command to have it dump the coordinates. The default port is 2947 so the command is "telnet localhost 2947". Once connected, just type "p" and enter. Here is a sample:

```
[root@localhost root]# telnet localhost 2947
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
p
GPSD,P=34.447775 -108.651010
```

The only real trouble I have had with gpsd was with it crashing when the GPS Receiver is not connected and I telnet to port 2947 and press enter. It's normally OK to have gpsd running at all times in the background. When the receiver and Kismet are up, it will do its job.

4.3.4.3 Connect antenna

If you have an external antenna, you should go ahead and connect it now.

4.3.4.4 Kismet

If you have set up Kismet correctly, you should be able to just type “kismet” to get it running. This is the initial display on my system, the Network List:

```
+-Network List--(Autofit)-----------------------------------------------+-Info---+
|     Sgn Name                    T W Ch Packts Flags IP Range       Data | Ntwrks |
|   .  56 APUnit2                 A Y 06    120       0.0.0.0          0 |     11 |
|   !  59 asdfasdf                A Y 06    138       0.0.0.0          0 | Pckets |
|   !  62 <asdfasdf>              A Y 01    121       0.0.0.0         18 |    903 |
|       0 linksys                 P N --     47       0.0.0.0          0 | Cryptd |
|   .  55 <no ssid>               A Y 02     78       0.0.0.0          0 |     66 |
|   !  62 asdfasdf                A Y 11    360       0.0.0.0         47 |   Weak |
|       0 <no ssid>               A Y 06      8       0.0.0.0          1 |      0 |
|       0 <no ssid>               A Y 11     15       0.0.0.0          0 |  Noise |
|       0 <no ssid>               P N --      4       0.0.0.0          0 |      0 |
|       0 linksys                 P N --      1       0.0.0.0          0 | Discrd |
|       0 qwerty                  P N --      4       0.0.0.0          0 |      0 |
|                                                                         | Pkts/s |
|                                                                         |      9 |
|                                                                         |        |
|                                                                         |        |
|                                                                         |        |
|                                                                         |        |
|                                                                         |        |
|                                                                         |        |
|                                                                         |        |
|                                                                         | orinoc |
|                                                                         | Ch:  0 |
|                                                                         |        |
|                                                                         | Elapsd |
+-Lat 29.744 Lon -95.560 Alt 51.5f Spd 0.000f/s Hed 180.000 Fix 3D----------+00:02:11+
+-Status-----------------------------------------------------------------------------+
|  Found new probed network "<no ssid>" bssid 00:90:4B:00:00:00                      |
|  Associated probe network "00:02:2D:00:00:00".                                     |
|  ALERT: Suspicious client 00:07:0E:00:00:00 - probing networks but never           |
|  participating.                                                                    |
+-Battery: 79% 0h0m0s----------------------------------------------------------------+
```

Note the GPS coordinates just above the Status pane. Within Kismet the “Fix” should always be 2D or 3D. None means something is wrong with GPS and you will not be able to plot the packets you are capturing later.
There are plenty of details for navigating the interface at the Kismet documentation page (http://www.kismetwireless.net/documentation.shtml) but you can find most commands with the “h” help command.

4.3.4.5 Canvassing the Campus

Now you should make you way all over and around your campus. The more time you spend and the more ground you cover the better your data sample will be. Here are some quick tips:

- Avoid retracing your steps. The more coordinates you cover the better gpsmap can calculate the centers of the networks it sees.
- Don’t forget to travel up to a mile away from campus. I would suggest a car instead of a cart for this part. Wireless signals can be detected under some circumstances this far and it is important to demonstrate it.

- You might want to invest in some Velcro straps to keep the GPS receiver attached to your laptop so you can have at least one hand free to type assuming you are holding the laptop.
- Make sure the GPS receiver has a good view of enough satellites at all times. Most receivers will display a sky map that lets you know where the satellites can be found in the sky.

4.3.5 Problems to Watch Out For

- Make sure you verify that the Kismet-gpsd-GPS receiver communication chain is working periodically. The batteries in your receiver can die, the serial cable can come loose or gpsd can silently crash in the background when you aren't expecting it. Getting back to your office and finding that no GPS data was recorded during the last hour can be a disheartening setback.
- Be careful of traffic and pedestrians. This is probably obvious, but it is easy to get distracted by the Kismet display and wander off track. If you do have a golf cart, remember that it is not intended for streets.
- If you use a car for wardriving, have a friend run the laptop or do yourself a favor and put it in the back seat out of reach.

4.4 Scan Analysis

This is the area that is hard to find information about on the net so hopefully you can learn from my mistakes. When I started scanning, the number of signals my little detector found startled me. Only after culling uninteresting signals did I see a good picture.

4.4.1 Understanding the Capture Files

By default, Kismet stores the data it collects in six types of files. I will provide a brief overview of them here, but see the Kismet documentation for more details.

4.4.1.1 Kismet-Sept-15-2003-1.cisco

You may or may not see this capture file. It stores extra information about Cisco APs that they tend to broadcast. The packets are part of the Cisco Discovery Protocol (CDP). I haven't found them to be too useful in mapping a network, because they seem to have redundant data found in the other capture files. I expect if you have deployed Cisco wireless equipment at your site you would find more useful information in this file.

4.4.1.2 Kismet-Sept-15-2003-1.network

This file contains a human readable list of the networks detected. Networks are defined as APs and clients that share the same SSID. The file is useful for quick peeks, but it is a subset of the detailed information that can be found in the next file.

4.4.1.3 Kismet-Sept-15-2003-1.csv

This file contains detailed information about every network discovered. Despite its name, it uses semicolons to separate fields instead of commas. There is also a bug in 3.0.1 that omits a semicolon between the "Total" and "Carrier" columns in the records. I see from the Kismet CHANGELOG that this has been fixed in the development branch. It might be fixed in 3.0.2 and above. Here is a one-line Perl script to correct the problem:

```
perl -i.bak -pe 's/(;\d*)IEEE 802/\1;IEEE 802/' filename
```

The script is not very robust so if you run into a network with the SID of 1IEEE 802, you will have to modify it to avoid corrupting the .csv file. A backup file is made before the file is edited.

Most of the columns are self-explanatory if you understand wireless 802.11, but here are some highlights:

- NetType: This is the type of network. Possible values are "infrastructure" or a network with an AP, "probe" a wireless client that had not associated with an AP, "ad-hoc" a wireless client using the ad-hoc networking method and "unknown". We will concentrate in "infrastructure" networks because they usually represent the biggest holes. "probe" and "ad-hoc" systems should not be ignored because an attacker could compromise the client and then be inside your perimeter. I consider them a lower risk because they generally have lower ranges than APs and attackers have the added barrier of compromising the client.
- ESSID: This is the network name given by the administrator. All clients and the AP must have the same ESSID.
- BSSID: This is the wireless MAC address of the AP. It is analogous to an Ethernet physical address and we will be using it when we map our detected networks to home in on specific APs.
- Info: This contains extra information the AP may be broadcasting. For example Cisco APs appear to broadcast extra information entered by the administrator. If the administrator makes note of the AP's location in this field such as NW or SE, it can help you home in on the AP's location.
- Channel: This one is pretty self-explanatory, but you should note if you are only seeing one to three channels right next to each other – your wireless card may be stuck monitoring only one channel. Even if your card is stuck on one channel, it can detect some signals on adjacent channels because the spectrums of channels overlap.
- Cloaked: This is Yes or No and indicates whether the AP is using any of several methods to hid its SID and other information. The primary method is to not broadcast its SID. This doesn't really tell you much about the network because both a user with a rogue AP and a normal corporate AP might be secured in this manner.
- WEP: This is Yes or No and tells you whether Wired Equivalency Protocol is enabled. You cannot tell if it is 40-bit or 128-bit WEP by examining the encrypted packet.
- Decrypted: This is Yes or No and records if you have set up the correct WEP key for this network in your kismet.conf configuration file. This

allows Kismet to decrypt network traffic on the fly, but is not a method for breaking the WEP key itself.

4.4.1.4 Kismet-Sept-15-2003-1.dump

This file contains a raw dump of the packets seen by Kismet. This can be used for all the same purposes as a normal network capture. Because we are concentrating on locating rouge APs, this file will not be covered much in this paper.

4.4.1.5 Kismet-Sept-15-2003-1.xml

This file contains a detailed record of your entire Kismet run minus GPS data. It is in XML format so other XML aware programs can parse it. This is important to us as it is one of the files gpsmap uses to create its images.

4.4.1.6 Kismet-Sept-15-2003-1.gps

This file will only appear if Kismet received a signal from gpsd during its run. Unfortunately, I have found that occasionally Kismet will record a way off latitude or longitude such as 0. Here is a one-line Perl script that eliminates these false GPS lines from the .gps file so you can get a more accurate map:

```
perl -i.bak -ne 'print unless (/lat="0.000000"/ or
/lon="0.000000"/)' *.gps
```

I have mainly had trouble with zeros for the "lat" or "lon", but you can modify the script if it generates 180.000000 or some other number that is nowhere near your campus. I'm not sure what causes these problems, but I think it mostly occurs when you have Kismet running before the GPS receiver has a good fix.

4.4.2 Filtering out the Noise

Because we are looking for rogue APs, we can eliminate clients ("probe" and "ad-hoc" networks). There are probably better ways to do this, but I generally use simple scripts to parse the .csv file and create a filter file with only the BSSIDs of "infrastructure" networks. This file can be used with gpsmap to graph only those BSSIDs. As we will see, this can still produce a noisy graph with too many APs displayed to be useful.

Here is an AWK script to select only infrastructure networks:

```
awk -F\; '
 $2 ~ /infrastructure/ {
  if (flag==1) printf(";");
  printf("%s",$4);
  flag=1;
 }
' filename.csv > filtered.txt
```

4.4.3 Graphing the Data

Now that we have just the APs, we graph the data. Running gpsmap by itself will give you a hint how to proceed:

```
# gpsmap
FATAL:  No drawing methods requested.
Usage: gpsmap [OPTION] <GPS Files>
```

```
Try `gpsmap --help' for more information.
```

4.4.3.1 Drawing Methods

You must select one or more drawing methods to get any output. From the gpsmap man page, here are the important drawing methods:

```
      TRACK DRAWING (-t, --draw-track)
            Draws a blue track along the traveled path, based on the
            track data saved by Kismet.
```

This can be interesting to add to the drawing, but if you really travel all over your campus, the little blue track can be excessive. It is good for proving you did the work to management if needed.

```
      BOUNDING RECTANGLE (-b, --draw-bounds)
            Draws the bounding rectangle around the extreme points of
            each network.
```

I find this method pretty useless because the rectangle is always oriented with its sides parallel and perpendicular to the longitude lines. This masks the true outline of the network in question. Also, drawing multiple networks on the same graph is next to impossible to decipher.

```
      RANGE CIRCLE (-r, --draw-range)
            Estimates the range of a network based on the average
            center and the distance to the closest extreme corner.  Not
            exact, but often useful for estimating the range of the
            network.
```

This method is useful and colorful. The circles appeal to the innate notion that wireless signals should radiate in a circle, but this also can mask the real nature of wireless signals which often radiate in non-uniform patterns. One big advantage is that gpsmap will attempt to guess the center of the network

```
      CONVEX HULL (-u, --draw-hull)
            Convex hull of all sample points for each network.  This
            will display the exact detected range of the networks.
```

This is the better version of the bounding rectangle that respects the sampled points of the signal.

```
      SCATTER PLOT (-a, --draw-scatter)
            Draws a dot for every detected packet point.
```

This puts a point on the graph for each detected packet. It is useful to show where the signal was sampled, but the points themselves do not do a good job of indicating the center of a network unless you cover most of the area around the AP.

```
      POWER INTERPOLATION (-p, --draw-power)
            By far the most CPU intensive, power interpolation forms a
            grid over the image and attempts to interpolate the power
            for points that aren't directly sampled.  For this graph to
            be a reasonable representation of reality, samples around
            the entire area, preferably forming a grid or mesh, should
            be taken.
```

I have never sampled enough in a systematic enough manner to make this method useful. It makes some interesting graphs, but without a grid of samples, it will mostly look like an expanded version of the scatter plot.

```
      NETWORK CENTER (-e, --draw-center)
            Simply draw a dot in the averaged center of each network
```

Despite the simplicity of this option, it is extremely useful. Depending on how many samples you have, this can be a fairly accurate representation of the location of the AP. If you are plotting more than one network, you will want to use this in concert with the label option below to indicate which point goes with which network.

```
LABELS (-l, --draw-labels <list>)
      Label a network with the bssid or ssid.  Labels are drawn
      based off the network center and the orientation.
```

Unlike most of the other options, this one requires an argument of bssid, name or a combination of the two separated by a comma. As long as the detected networks are not too close together, this option is useful for a graph. For single-network graphs, it still serves as good documentation of which network has been graphed.

There is also a option called "Draw Legend (-k, --draw-legend)" but the BUGS section of the man page say this option does not work. I give it a try each time I upgrade Kismet to see if they have implemented it, but it doesn't work as of version 3.0.1.

4.4.3.2 Other Useful Options

I have used practically every option in gpsmap at one point and I would suggest you experiment to see which options produce useful results for you. Here are the options that we will be using in this paper:

- -v, --verbose: Gives us progress reports when we are drawing graphs on slow computers. Also provides hints of where the process went wrong if it fails to produce a graph.
- -o, --output <filename>: Give a name that is meaningful to use for the produced PNG (Portable Network Graphics format) file. Otherwise gpsmap will use the same name as the map downloaded from your chosen map source (see below) with the suffix changed to .png.
- -f, --filter <MAC list>: By default, this option removes the comma-separated list of MACs from the output of the graph. Remember that there should be no white space in and between the MACs or the shell will think you have moved on to the next argument of the command line.
- -i, --invert-filter: This reverses the action of the –f option above by limiting the graph to only the MACs listed. I generally use this option to home in on suspect APs.
- -S, --map-source <#>: This option selects the on-line map source for the background map of the produced graph. The possible numbers are
    - 0 MapBlast (vector): This is the default and provides moderately detailed maps with street names, etc. There have been rumblings in the past on the Kismet mailing lists that using MapBlast from a command line tool violates MapBlast's end user agreement. Please be aware of this if you plan on using this source.
    - 1 MapPoint (vector): Unfortunately this source was not working for me.

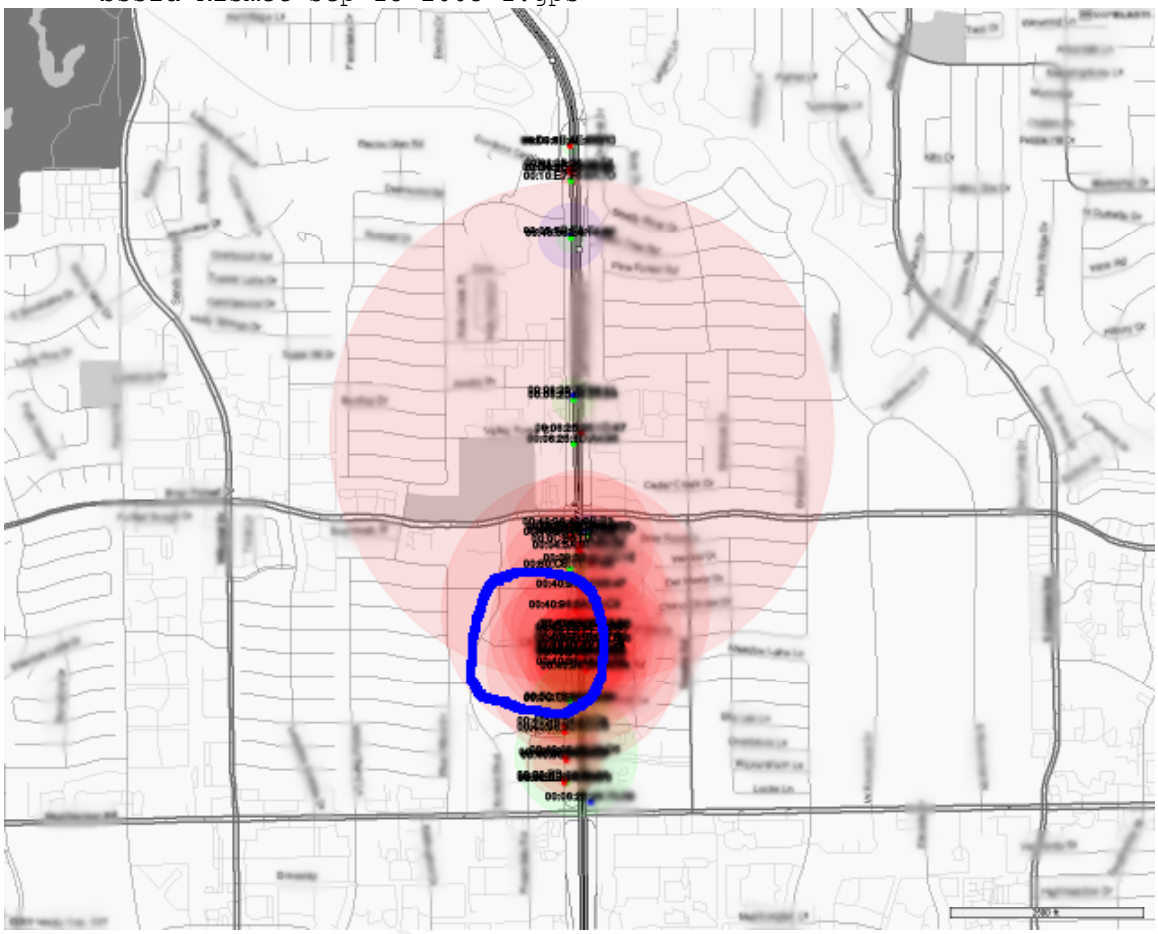
    - 2 TerraServer (photo): This can be very striking to show management how far signals propagate against an actual satellite photo of your campus. Two problems I have run into are the age of the photos that don't include major features of our campus and the darkness of the picture in some locations makes it difficult to see the gpsmap-added features. This could be overcome manually by editing the downloaded map in a tool such as GIMP or PhotoShop to "lighten" the image or reduce its contrast. NOTE: you must use the –s option discussed below when using TerraServer.
    - 3 Tiger US Census (vector): This source produces maps with a bare minimum of features using freely available data from the US government. The sparseness of the map can be of great benefit because it doesn't get in the way of the gpsmap-added features. I usually prefer to use this for my graphs.
- -s, --scale <s>: Although this option has other uses, I mainly use it in conjunction with "–S 2" to set the scale of the downloaded TerraServer image. Valid values are 10 to 16. A value of 10 or 11 has worked best for me and shows a few to several city blocks in the image.
- -D, --keep-gif: Normally gpsmap will delete the downloaded GIF image after it has finished creating a graph. This option leaves the file that can be reused with the "-m" option described below.
- -m, --user-map <map>: Used with the "-D" option or with a correctly downloaded map file, this option allows you to use a consistent map for all of your graphs. I find it is best to get a map that includes the campus and a few surrounding blocks. Then all plots of the data will be at the same scale, which makes it easier to decipher. For example, graph two networks, one of which has only a few samples all within a few feet of each other and the other consists of thousands of samples covering half a square mile. If you allow gpsmap to download a map that best fits each network, then you will get one map zoomed in so far that you may not even be able to tell what real estate it covers and another that may not even center on your campus.
- -n, --network-colors <c>: Defines the way colors are used on the drawing mode features you have selected. Possible numbers are
    - 0: gpsmap selects a random color for each network drawn
    - 1: Colors networks according to whether or not they have WEP enabled. Green (as in go) means no WEP and red (as in stop) mean WEP is enabled.
    - 2: Colors networks according to their channel. This would be useful to check for overlap of channels between buildings, to diagnose network problems.

4.4.3.3 Example gpsmap Commands

4.4.3.3.1 First Try

To graph all of the detected "infrastructure" networks on one graph:

```
gpsmap --verbose --output graph2.png --filter `cat filtered.txt`
--invert-filter --map-source 0 --keep-gif --network-colors 1 --
draw-range --draw-range-opacity 10 --draw-center --draw-labels
bssid Kismet-Sep-15-2003-1.gps
```



Even though I have fuzzed BSSIDs and street names, you can see how busy this graph is. My campus is within the blue circle so you can see that this graph is useless except to show how saturated the airwaves are.

4.4.3.3.2 Getting a Custom Map

If you observed the verbose output of the gpsmap command, you would notice the "Map url" being displayed:

```
Map url: http://www.vicinity.com/gif?&CT=39.749134:-
85.557739:10000&IC=&W=1280&H=1024&FAM=myblast&LB=
```

And a bit later, the file that this map is placed in:

```
map_39.749134_-85.557739_10000_1280_1024.gif
```

This URL can be modified and re-fetched using the "wget" command to get a map that is closer to the center of my campus. Once fetched and used in subsequent gpsmap invocations, gpsmap will conform what it draws to the bounds of the map, based on its name. First, I get a custom map that is closer to the center of my campus using this command:

```
wget --output-document=map_39.743190_-85.560008_4000_1280_1024.gif
'http://www.vicinity.com/gif?&CT=39.743190:-
85.560008:4000&IC=&W=1280&H=1024&FAM=myblast&LB='
```

Notice that the items in bold above are the changes which shift the latitude, longitude and scale of the downloaded map. I chose a scale of 4000 after some experimentation.
Now if I pick an individual network and graph it using this command:

```
gpsmap --verbose --output 00-40-96-5A-64-04.png --filter
00:40:96:5A:64:04 --invert-filter --user-map map_29.743190_-
95.560008_4000_1280_1024.gif --keep-gif --network-colors 1 --
draw-range --draw-range-opacity 10 --draw-center --draw-labels
bssid,name --draw-hull --draw-hull-opacity 50 Kismet-Sep-1*.gps
```

I get a zoomed in map with one network plotted:

Since the network center is definitely within my campus, I need to verify this AP is not on our network. See Section 9 below for a script I use to graph each "infrastructure" network individually on one pre-downloaded map.

4.4.4 Homing in on Suspected Rogue APs

Once you have accurate maps of the networks that are in and around your campus, you can work on finding the source of the signal. Unfortunately, this is not an exact science especially when you are using very inexpensive equipment such as the Super Cantenna. It does provide you some direction towards the source, but if you are standing at the base of a tall building, all of the floors above you reside in a small arc anyway.

If you have narrowed down the source of signal to a building or even a side of a building, you will probably have to go to each floor to see if the signal gets stronger. If you are on a floor and the signal strength is less than 90, then the source is probably not on the same floor as you. To get a more consistent signal reading of the target source, you should disable the hopper and listen continuously the AP's channel. For example, to signal only channel 10, use this Kismet command:

```
kismet –X –IKismet:10
```

This assumes that the source name you defined in your kismet.conf is "Kismet".

4.4.5 Dealing with Known Networks

The kismet.conf file has a method for ignoring all signals from certain networks using the "filter_tracker", "filter_dump" and "filter_export" options. "filter_tracker" is equivalent to the "mac_filter" used in 2.x versions of Kismet.
I do not recommend filtering out any networks because an attacker or user determined to set up their own AP can set their MAC address to the MAC of a known good AP. If you ignore them completely you will never notice the new AP. If you instead graph all "infrastructure" networks you may notice that a network that used to be on the NE corner of your campus has migrated to the SW.

5 After Snapshot

Using the tools and methods described in this paper I have been able to detect and eliminate several unauthorized access points from our campus. I have detected an insecure setup on our authorized access points where we were using a far too descriptive SSID. I have learned what a future wireless deployment might look like by analyzing the wireless deployment of our tenants. We still cannot say at any moment in time that there are zero unauthorized access points on our campus because of the snapshot nature of the audit. I would prefer to have a number of permanent wireless detectors distributed across our campus. These could provide almost instantaneous notification of unauthorized APs and other wireless intrusion attempts.
By performing the audit again on a regular basis we can at least minimize our exposure period to the time between audits.

6 Conclusion

I hope that this paper has taken you beyond the available documentation available for Kismet. I have worked off and on with Kismet for the last year and half and have been very satisfied with its stability and results. New features of Kismet 3.x have not been explored in greater detail as I am just becoming familiar with them myself. I hope to use Kismet's "drone" feature coupled with static gpsd settings to provide a constant wireless IDS for my campus in the future. One missing feature is a 3D graph of the wireless data. Using drones spread on multiple floors of high-rises, I think a 3D graph could help triangulate APs in adjacent high-rises.

7 map-infrastructure-aps

```
#!/bin/sh
#
# map-infrastructure-aps mapfile 'fileprefix-wildcard-expression'
#
# Make sure you put the argument in quotation marks to avoid shell
# expansion before the script runs
#

for mac in `awk -F\; '$2 ~ /infrastructure/ { print $4 }' $2.csv | sort
-u`
do
  nicemac=`echo $mac | sed 's/:/-/g'`
  if gpsmap --verbose --output $nicemac.png --filter $mac \
    --invert-filter --user-map \
    $1 --keep-gif --network-colors 1 --draw-range \
    --draw-range-opacity 10 --draw-center \
    --draw-labels bssid,name --draw-hull \
    --draw-hull-opacity 50 $2.gps 2>&1 | \
    grep "Processing 0 networks"
  then
    rm -f $nicemac.png
    echo $mac not on map
  else
    echo $mac graphed in $nicemac.png
  fi
done
```

8 List of References

Ritchie. “Red Hat + Kismet - HOWTO – FAQ.” August 13, 2003. URL: http://www.tipsybottle.com/technology/wireless/RedHat-Kismet-HOWTO-FAQ.shtml (September 18, 2003).

Cantenna Site. URL: http://www.cantenna.com/ (September 18, 2003).

JEFA Tech Site. URL:http://www.jefatech.com/ (September 17, 2003).

“Garmin: eTrex.” Garmin Site. URL: http://www.garmin.com/products/etrex/ (September 18, 2003).

NetStumbler Site. URL: http://www.netstumbler.com/ (September 18, 2003).

snax. “AirSnort Homepage.” The Shmoo Group. URL: http://airsnort.shmoo.com/ (September 18, 2003).

Ritchie. “Red Hat + Kismet - HOWTO.” August 13, 2003. URL: http://www.tipsybottle.com/technology/wireless/RedHat8-Kismet-HOWTO.shtml (September 18, 2003).

snax. “Orinoco Monitor Mode Patch Page.” URL: http://airsnort.shmoo.com/orinocoinfo.html (September 18, 2003).

“Fritz’s Software Page.” URL: http://www.kraftvoll.at/software/index.shtml (September 18, 2003).

ImageMagick Site. URL: http://imagemagick.sourceforge.net/ (September 15, 2003).

Kismet Site. URL: http://www.kismetwireless.net/ (September 18, 2003).

Kershaw, Mike. “Kismet.” URL: http://www.kismetwireless.net/documentation.shtml (September 18, 2003).

“CHANGELOG-devel.” URL: http://www.kismetwireless.net/CHANGELOG-devel (September 18, 2003).