This is how I installed the kismet_drone on the wrt54g (v2.2) and wrt150n (v1.1) while running the kismet_server and kismet_client on a Linux box. (I was using version 2005.06.R1 of kismet.)

Also tested on a v2.0 wrt, dd-wrt firmware v22-r2 and kismet version kismet-2005-08-r1.

Contents

Summary

The wrt's are slow machines, so you cannot run the whole of the kismet program directly on the wrt. Kismet is split into three separate modules : the drone, the server and the client. The drone is a small program which gets the raw data from the wireless card itself. The client is the user interface to display the results on your screen. The server is the piece if software that sits in the middle of the drone and the client.

The method shown here puts the only the drone on the wrt and runs the server and client on your Linux box.

If you do not have a Linux machine that you can use, then have a look at Kismet Server/Drone for instructions on how to run the server and the client with cgywin under Windows.

Note: The drone on the wrt doesn't support channel hopping to search through all of the wireless channels, so a small script has to be ran on the wrt to manually do the channel hopping. Also, the wireless driver on the wrt doesn't support the reporting of the received signal strength so all of the signals reported to the user via the client will not display the correct signal strength.

Preparing the wrt54g

Installing the drone

First, I installed the latest release of the DD-WRT firmware.

I then using the web interface, set it up to use a static ip address, disabled the dhcp server and also enabled telnet and ssh access to the wrt. (You may prefer to leave the dhcp server running if you do not have another dhcp server on your network). I also turned off the firewall.

Now we need to get the files onto the wrt. There are several ways of doing this. You may find it easier to copy the files to a local desktop machine and then later copy them to the wrt as this means that you can edit the conf file on the desktop (using the editor of your choice) rather than having to use the 'vi' editor on the wrt.

(Quick Vi Tutorial : http://math.la.asu.edu/vi_tutorial/vi3.html)

Method 1 - using a local web-server

On the Linux desktop machine at a console, type

```
telnet <address of wrt>
```

(If you have sshd enabled, you could use ssh instead.)

Log in using user name 'root' and the same password as the one for the wrt's web interface (the default is 'admin').

Then at another terminal, download the MIPS binaries for the wrt. Download them from the kismet website http://www.kismetwireless.net/download.shtml . I used version 2005-06-R1a which was the latest version.

cd into the directory where you downloaded it to and unzip it.

```
tar -zxvf kismet-2005-06-R1a-wrt54.tar.gz
```

As I have a local web-server running on my desktop box, I decided to use it and wget to copy the kismet files over to the wrt. So I then copied the kismet_drone and conf/kismet_drone.conf files to the root of my web-server.

Go back to your other terminal with the telnet session to the wrt.

The /tmp directory is the only place on the wrt's file system which is writable as it is stored on a ramdisk. This means that every time that you power down the wrt, you have to reload the kismet files to it.

```
cd /tmp
```

Get files onto wrt

```
wget http://<ip address of desktop box>/kismet_drone
wget http://<ip address of desktop box>/kismet_drone.conf
```

Put kismet_drone.conf file in /tmp/etc directory.

```
mv kismet_drone.conf /tmp/etc
```

and make drone executable

```
chmod +x /tmp/kismet_drone
```

Method 2 - get files directly from the Internet

This method assumes that your wrt has direct access to the Internet so that you can directly download the compressed files from the kismet site onto the wrt using 'wget'.

Telnet into the wrt and get the compressed MIPS files from the kismet site. Unzip them, move the kismet_drone and kismet_drone.conf files the the correct place, delete the unwanted files and then make the drone file executable. (If you are using a more up to date version of kismet, you will have to use a different URL and file names.)

```
telnet <ip address of wrt>
cd /tmp
wget http://www.kismetwireless.net/code/kismet-2005-06-R1a-wrt54.tar.gz
tar -zxf kismet-2005-06-R1a-wrt54.tar.gz
mv kismet-2005-06-R1a-wrt54/kismet_drone /tmp/kismet_drone
mv kismet-2005-06-R1a-wrt54/conf/kismet_drone.conf  /tmp/etc/kismet_drone.conf
rm kismet-2005-06-R1a-wrt54.tar.gz
rm -rf kismet-2005-06-R1a-wrt54
chmod +x /tmp/kismet_drone

Here's the more recent version:
http://www.kismetwireless.net/code/kismet-2005-08-R1-wrt54.tar.gz
http://www.kismetwireless.net/code/kismet-2006-04-R1-wrt54.tar.gz
```

Method 3 - using scp to copy files from a local Linux box

You could probably also use scp to load the files to the wrt from a local desktop box instead of wget/web-server.

Download the kismet drone wrt files to a directory on your machine. Open a terminal and enter the directory that you downloaded the files into with

```
cd <path to dir>
```

Then unzip the kismet files with

```
tar -zxvf <kismet file>
```

This will unzip the files and put them into a subdirectory. Enter this directory with

```
cd <kismet directory>
```

At this point it is probably a good time to edit the config files instead of having to do it later on the wrt itself. (See the 'configuring the drone' section below)

Then copy the files over to the wrt using scp to the /tmp directory on the wrt.

```
scp kismet_drone <ip address of wrt>:/tmp
scp conf/kismet_drone.conf <ip address of wrt>:/tmp/etc
```

Configure drone

You then need to edit the drones conf file -- I used the vi editor to modify the conf file. I believe that vi is the only editor that comes with the firmware. If you have never used it before, you might need to google for a quick vi guide. Alternatively, you could modify the conf file on your desktop machine with your favorite editor before copying the files to the wrt.

```
vi /tmp/etc/kismet_drone.conf
```

Change the line

```
allowedhosts=127.0.0.1
```

to

```
allowedhosts=<address of desktop box>
```

(You could also allow access to all the machines on your local network by doing 'allowedhosts=a.b.c.0/24' where a.b.c are the first three octets of your networks ip address. This would also make it possible for more than one machine at a time to connect to the drone and display the results.)

Next, change the line

```
source=wrt54g,eth2,Kismet-Drone
```

to

```
source=wrt54g,prism0,Kismet-Drone
```

(I have read that this line should be different if running another version of the hardware - this works with v2.2 of the wrt54g).

Running the drone

Now we start the drone...but first, we need to put the wireless interface into passive monitoring mode.

```
wl ap 0            #switch off access point mode
wl passive 1
wl channel <channel number>
```

The driver for the wrt doesn't do channel hopping, so it needs to be done with a small script running on the wrt.

See channel hopping on kismet drone.

Then run the drone.

```
/tmp/kismet_drone
```

If this works, it will display some messages ending with 'Allowing connections from <desktop ip address>'. You will need to keep the telnet session open, as closing it will terminate the drone.

Preparing the desktop

Install server and client

You will need to be logged in as root. As I use a debian based Linux distro, I started by using apt to install kismet.

```
apt-get install kismet
```

Unfortunately for me, the version of kismet available using apt was not the same as the version as the drone and the drone was unable to talk to the server as the protocols had changed between the two versions of kismet. The versions (probably) need to be the same. In the end, I installed kismet on my desktop from source from the kismet site (http://www.kismetwireless.net/download.shtml).

```
cd <download directory>
tar -zxvf <kismet source file>tar.gz
cd <kismet dir>
./configure
make
make install
```

Configure kismet

[note: I found that the config files were in /etc/kismet when installing using apt-get, but they were in /usr/local/etc when installing from source.] First, you need to make a kismet user for the server to run as.

```
adduser kismet
```

...and fill in the blanks. Then you need to edit the /usr/local/etc/kismet.conf file and change

```
suiduser=your_user_here
```

to

```
suiduser=kismet
```

Set the wireless source by changing

```
source=none,none,addme
```

to

```
source=kismet_drone,<wrt ip address>:3501,wrt54g
```

I found that kismet couldn't write its log files to the default directory, so changed the line

```
logtemplate=%n-%d-%i.%l
```

to

```
logtemplate=%h/%n-%d-%i.%l
```

so that the log files get saved in the kismet users directory (/home/kismet).

Run kismet server and client

Now everything should be ready. First you will need to run the server in the background. (I assume that you still have the kismet_drone running via a telnet session to the wrt (see the 'running the drone' section above.)) then run the GUI client.

```
/usr/local/bin/kismet_server &
/usr/local/bin/kismet_client
```

Stopping kismet

Press shift Q in the GUI to stop the kismet_client, then stop the server which we set running as a background process.

```
killall kismet_server
```

To kill the drone, press ctrl-C in the telnet session.