This document details how I configure my Motion Computing m1300 tablet with Linux so that I can conduct wireless auditing with Kismet, Aircrack-ng, and Karmasploit.

Boot the Ubuntu 8.04 LST Hardy Heron cd.   Choose the option to begin the installation.

```
Welcome [English]
Where are you? [NY EDT]
Keyboard layout [U.S. English]
Prepare disk space [Guided – use the largest continuous free space]
      This will use the free space behind the Windows XP installation
Migrate Documents and Settings [Nothing to do]
Who are you? [username:  wireless, computername:  wireless-tablet]
Ready to install
```

http://ubuntuguide.org/wiki/Ubuntu:Hardy is invaluable for getting Ubuntu configured post install.

Issues Post Install Fails: screen resolution and digitizer pen (this stuff worked just fine with version 7.04!)

After installing Hardy Heron 8.04 LST the screen resolution will be set to 800x600 and past methods to correct it (sudo dkpg-reconfigure xserver-org) will not work.   The following blurb is from http://ubuntulinuxtipstricks.blogspot.com/2008/04/faq-hardy-upgrade.html

The new Xorg is supposed to be all nice and hotplugable, but dpkg-reconfigure xserver-xorg is no more. /etc/X11/xorg.conf is also now very barebones. This is for the hotplugability. The correct way to configure this new version of X is with the xfix command. Changing resolution is done on the fly with xrandr.

I say to hell with the new way of doing things.   The previous version of Ubuntu I ran on the m1300 was Feisty Fawn 7.04 and I have saved that xorg.conf configuration file.   I replaced the 8.04 xorg.conf with the version from 7.04.

```
#cp /etc/X11/xorg.conf /etc/X11/xorg.conf.xx.xx.xx
NOTE:   xx.xx.xx is the current date ;-)
#vim /etc/X11/xorg.conf or #gedit /etc/X11/xorg.conf
```

Replace the text below...

```
Section "Device"
      Identifier  "Configured Video Device"
      Driver            "vesa"
      Option            "UseFBDev"        "true"
EndSection

Section "Monitor"
      Identifier  "Configured Monitor"
EndSection

Section "Screen"
      Identifier  "Default Screen"
      Monitor           "Configured Monitor"
      Device            "Configured Video Device"
EndSection
```

...with...

```
Section "Device"
      Identifier  "Intel Corporation 82852/855GM Integrated Graphics
Device"
      Driver            "i810"
      BusID       "PCI:0:2:0"
EndSection

Section "Monitor"
      Identifier  "Generic Monitor"
      Option            "DPMS"
      HorizSync   28-51
      VertRefresh 43-60
EndSection

Section "Screen"
      Identifier  "Default Screen"
      Device            "Intel Corporation 82852/855GM Integrated
Graphics Device"
      Monitor           "Generic Monitor"
      DefaultDepth      24
      SubSection "Display"
            Depth       1
            Modes       "1024x768"
      EndSubSection
      SubSection "Display"
            Depth       4
            Modes       "1024x768"
      EndSubSection
      SubSection "Display"
            Depth       8
            Modes       "1024x768"
      EndSubSection
      SubSection "Display"
            Depth       15
            Modes       "1024x768"
      EndSubSection
      SubSection "Display"
            Depth       16
            Modes       "1024x768"
      EndSubSection
      SubSection "Display"
            Depth       24
            Modes       "1024x768"
      EndSubSection
EndSection
```

Reboot or kill X to login with the new resolution.

Now for the wacom digitizer pen. There is a great tutorial on the ubuntuforums.org from an individual who has gotten Hardy Heron working on the m1300. He offers details on getting the pen working. You can visit the forum page at http://ubuntuforums.org/showthread.php?s=f7bfdde0a3a1ce4b0ff8a52f35ab4537&t=796359. However, before I found this site all I did was add the following xorg.conf settings (copied from my previous Feisty Fawn 7.04 installation) and the pen worked fine.

Add the following lines to /etc/X11/xorg.conf

```
#vim /etc/X11/xorg.conf

Section "InputDevice"
      Driver            "wacom"
      Identifier  "stylus"
      Option            "Device"    "/dev/input/wacom"
      Option            "Type"            "stylus"
      Option            "ForceDevice"     "ISDV4"           # Tablet PC
ONLY
       Option        "Mode"          "Absolute"
       Option        "Button3"            "2" #properly configure the
       Option        "Button2"            "3" #button on the pen for right
click
       Option        "TPCButton"     "on"
EndSection

Section "InputDevice"
      Driver            "wacom"
      Identifier  "eraser"
      Option            "Device"    "/dev/input/wacom"
      Option            "Type"            "eraser"
      Option            "ForceDevice"     "ISDV4"           # Tablet PC
ONLY
EndSection

Section "InputDevice"
      Driver            "wacom"
      Identifier  "cursor"
      Option            "Device"    "/dev/input/wacom"
      Option            "Type"            "cursor"
      Option            "ForceDevice"     "ISDV4"           # Tablet PC
ONLY
EndSection
```

Also, you will need to add the following lines to Section "ServerLayout".

```
        InputDevice     "stylus"      "SendCoreEvents"
        InputDevice     "cursor"      "SendCoreEvents"
        InputDevice     "eraser"      "SendCoreEvents"
```

Reboot or restart X and you pen should now work.

Updating and upgrading

```
#sudo cp -i /etc/apt/sources.list /etc/apt/sources.list_backup
```

```
#gksu gedit /etc/apt/sources.list
```

```
##--------------------
## UBUNTU REPOSITORIES
## -------------------
deb http://my.archive.ubuntu.com/ubuntu/ hardy main restricted
deb-src http://my.archive.ubuntu.com/ubuntu/ hardy main restricted
deb http://my.archive.ubuntu.com/ubuntu/ hardy-updates main restricted
deb-src http://my.archive.ubuntu.com/ubuntu/ hardy-updates main restricted
deb http://my.archive.ubuntu.com/ubuntu/ hardy universe
deb-src http://my.archive.ubuntu.com/ubuntu/ hardy universe
deb http://my.archive.ubuntu.com/ubuntu/ hardy-updates universe
deb-src http://my.archive.ubuntu.com/ubuntu/ hardy-updates universe
deb http://my.archive.ubuntu.com/ubuntu/ hardy multiverse
deb-src http://my.archive.ubuntu.com/ubuntu/ hardy multiverse
deb http://my.archive.ubuntu.com/ubuntu/ hardy-updates multiverse
deb-src http://my.archive.ubuntu.com/ubuntu/ hardy-updates multiverse
deb http://my.archive.ubuntu.com/ubuntu/ hardy-backports main restricted universe
multiverse
deb-src http://my.archive.ubuntu.com/ubuntu/ hardy-backports main restricted
universe multiverse
deb http://archive.canonical.com/ubuntu hardy partner
deb-src http://archive.canonical.com/ubuntu hardy partner
deb http://security.ubuntu.com/ubuntu hardy-security main restricted
deb-src http://security.ubuntu.com/ubuntu hardy-security main restricted
deb http://security.ubuntu.com/ubuntu hardy-security universe
deb-src http://security.ubuntu.com/ubuntu hardy-security universe
deb http://security.ubuntu.com/ubuntu hardy-security multiverse
deb http://my.archive.ubuntu.com/ubuntu/ hardy-proposed restricted main multiverse
universe
deb-src http://security.ubuntu.com/ubuntu hardy-security multiverse
deb http://ppa.launchpad.net/ubuntume.team/ubuntu hardy main # Ubuntu Muslim Edition
deb-src http://ppa.launchpad.net/ubuntume.team/ubuntu hardy main # Ubuntu Muslim
Edition
deb http://www.linuxmint.com/repository romeo/
deb http://tskariah.000webhost.com/ubuntu ubuntu main
## +++ Backports & Proposed (Ubuntu Unstable) +++
deb http://archive.ubuntu.com/ubuntu/ hardy-backports main restricted universe
multiverse
deb http://archive.ubuntu.com/ubuntu/ hardy-proposed main restricted universe
multiverse
## +++ Source Repositories +++
deb-src http://archive.ubuntu.com/ubuntu/ hardy main restricted universe multiverse
deb-src http://archive.ubuntu.com/ubuntu/ hardy-updates main restricted universe
multiverse
deb-src http://security.ubuntu.com/ubuntu/ hardy-security main restricted universe
multiverse
deb-src http://archive.ubuntu.com/ubuntu/ hardy-backports main restricted universe
multiverse

deb http://us.archive.ubuntu.com/ubuntu/ hardy main restricted
deb-src http://us.archive.ubuntu.com/ubuntu/ gutsy main restricted
deb http://us.archive.ubuntu.com/ubuntu/ hardy-updates main restricted
deb-src http://us.archive.ubuntu.com/ubuntu/ hardy-updates main restricted
##Universe
deb http://us.archive.ubuntu.com/ubuntu/ hardy universe
deb-src http://us.archive.ubuntu.com/ubuntu/ hardy universe
deb http://us.archive.ubuntu.com/ubuntu/ hardy-updates universe
deb-src http://us.archive.ubuntu.com/ubuntu/ hardy-updates universe
## Multiverse
deb http://us.archive.ubuntu.com/ubuntu/ hardy multiverse
deb-src http://us.archive.ubuntu.com/ubuntu/ hardy multiverse
deb http://us.archive.ubuntu.com/ubuntu/ hardy-updates multiverse
deb-src http://us.archive.ubuntu.com/ubuntu/ hardy-updates multiverse
## Backports
deb http://us.archive.ubuntu.com/ubuntu/ hardy-backports main restricted universe
multiverse
deb-src http://us.archive.ubuntu.com/ubuntu/ hardy-backports main restricted
universe multiverse
## Canonical Partner Repository
```

```
        deb http://archive.canonical.com/ubuntu hardy partner
        deb-src http://archive.canonical.com/ubuntu hardy partner
        deb http://security.ubuntu.com/ubuntu hardy-security main restricted
        deb-src http://security.ubuntu.com/ubuntu hardy-security main restricted
        deb http://security.ubuntu.com/ubuntu hardy-security universe
        deb-src http://security.ubuntu.com/ubuntu hardy-security universe
        deb http://security.ubuntu.com/ubuntu hardy-security multiverse
        deb-src http://security.ubuntu.com/ubuntu hardy-security multiverse
        ## PLF REPOSITORY
        deb http://packages.medibuntu.org/ gutsy free non-free
        deb http://ppa.launchpad.net/reacocard-awn/ubuntu gutsy main
        ## +++ Medibuntu +++
        deb http://packages.medibuntu.org/ hardy free non-free
        deb http://packages.medibuntu.org/ feisty free non-free
```

```
#wget -q http://packages.medibuntu.org/medibuntu-key.gpg -O- | sudo apt-key add -

#wget -q http://deb.mulx.net/pol.gpg -O- | sudo apt-key add -
#sudo apt-get update
#sudo apt-get upgrade
```

This will take a while!

Adding compilers and kernel source code

```
#sudo apt-get install build-essential
#sudo apt-get install linux-headers-`uname -r`
```

Onscreen Keyboard at Login

Ubuntu Hardy Heron comes with a nice onscreen keyboard written in python during the Google Summer of Code. This little app is called onboard.py

NOTE: This may be due to something I did after installation of Hardy Heron or and actual problem with the disto itself but when I start onboard it gives a segmentation fault. After some research it was suggested that I remove onboard and reinstall. That worked for me.

```
#apt-get remove python-virtkey onboard
#apt-get install python-virtkey onboard
```

Unlike in Feisty Fawn in Hardy Heron you no longer have to manually edit the python files to configure onboard to the specific screen size and starting position. To create an icon on the panel just right-click the panel, select Add to Panel..., choose Custom Application Launcher, and click Add

Enter the following, you can change the command settings to the size and position you prefer.

Type: Application
Name: On Screen Keyboard
Command: /usr/bin/onboard -x 0 -y 470 --size=600x180
Comment: Onboard On Screen Keyboard

To add Onboard on start-up (so you can login with the pen) do the following:

```
#cp /etc/gdm/Init/Default /etc/gdm/Init/Default.backup
```

Ensure the last three lines of the file look like this:

```
      fi
      exec onboard -x 0 -y 0 --size=600x180 &
      exit 0
```

Afterwards, open the menu:
System->Administration->Login Window

NOTE: be patient, for some reason it takes a long time to start

The window containing the preferences of the Login Window will open. In this window do the following:

- choose the tab named 'Local'
- click on the popup at the right of 'Style' and set it to 'Plain'. (mine was set to 'Themed with face Browser')
- close the window

Install XVKBD if you don't like onboard

```
#apt-get install xvkbd
#cp /etc/gdm/Init/Default /etc/gdm/Init/Default.backup
#gedit /etc/gdm/Init/Default
```

Ensure the last three lines of the file look like this:

```
      fi
      exec xvkbd &
      exit 0
```

Afterwards, open the menu:
System->Administration->Login Window

NOTE: be patient, for some reason it takes a long time to start

The window containing the preferences of the Login Window will open. In this window do the following:

- choose the tab named 'Local'
- click on the popup at the right of 'Style' and set it to 'Plain'. (mine was set to 'Themed with face Browser')
- close the window

Create Screen Rotation Icon
#apt-get install wacom-tools

```
#gedit /bin/rotate.sh

      #!/bin/sh
      orientation="`/usr/bin/X11/xrandr --query | /bin/grep 'default
      connected' | /usr/bin/awk '{print $5}'`"
      if [ "$orientation" = "normal" ]; then
            # Rotates screen orientation to the right
            /usr/bin/X11/xrandr --orientation right
            # Rotates the stylus cordinate plane
            /usr/bin/xsetwacom set stylus rotate CW
      elif [ "$orientation" = "right" ]; then
            # Rotates the screen back to normal
            /usr/bin/X11/xrandr --orientation normal
            # Rotates the stylus cordinate plane to normal
            /usr/bin/xsetwacom set "stylus" Rotate 0
      fi
```

```
#chomd 777 /bin/rotate.sh
```

Create an icon with that script and enjoy screen rotation!

Madwifi – latest version

```
#apt-get install madwifi-tools
```

NOTE: Ubuntu Hardy Heron comes preinstalled with a version of madwifi that is suitable. If you want the latest version then follow the steps below.

```
#apt-get install subversion
#apt-get install sharutils
#svn checkout http://svn.madwifi.org/trunk madwifi-ng
#cd madwifi/trunk
#make
#make install
```

NOTE: if the svn link does not work go to http://madwifi.org to get the up-to-date link.

Kismet

```
#apt-get install flex m4 bison
#apt-get install gpsd sox libncurses5-dev
#apt-get install libgmp3-dev libexpat1-dev libmagick9-dev
```

Download the latest stable version of libpcap from http://www.tcpdump.org (currently libpcap-1.0.0)

```
#tar zxvf libpcap-1.0.0.tar.gz
#cd libpcap-1.0.0
#./configure
#make dep
#make && make install
```

Download the latest stable version from http://www.kismetwireless.net
(currently 2008-05-R1)

```
#tar zxvf kismet-2008-05-R1.tar.gz
#cd kismet-2008-05-R1
```

Patch for GOOGLE MAPS

```
#wget http://parknation.com/gmap/files/gpsmap-gmap-0.1.tgz
#tar zxvf gpsmap-gmap-0.1.tgz
```

```
#patch -p0 < gpsmap-gmap-0.1/gpsmap-gmap-0.1.diff
```

Compile and install

```
#./configure
#make
#make install
```

You need to install a web server in order to view the output from generating a google/kismet map

```
#apt-get install apache2
#mkdir /var/www/gpsmap
```

Copy the files that came with the patch over to the directory you just created.

```
#cd gpsmap-gmap-0.1
#cp –R mapfiles /var/www/gpsmap
#cp index.html /var/www/gpsmap
```

You also need to get an api key for using google maps from google (**http://www.google.com/apis/maps/signup.html**). Insert this key into the top of the **index.html** file in the location of KEYHERE

NOTE: See *Creating Google Maps with Kismet* below for the next step in using gpsmap to create a gps plotted google map.

BUT FIRST . . .

Configure Kismet

Edit /usr/local/etc/kismet.conf (NOTE: kismet.conf may be installed elsewhere. Run **find / -name kismet.conf** or **whereis kismet** to find it)

```
#vim /usr/local/etc/kismet.conf

      # Kismet config file
      # Most of the "static" configs have been moved to here -- the command
      line
      # config was getting way too crowded and cryptic.  We want
      functionality,
      # not continually reading --help!

      # Version of Kismet config
      version=2004.10.R1

      # Name of server (Purely for organiational purposes)
      servername=Kismet

      # User to setid to (should be your normal user)
13    suiduser=wireless

      # Sources are defined as:
```

```
      # source=cardtype,interface,name[,initialchannel]
      # Card types and required drivers are listed in the README.
      # The initial channel is optional, if hopping is not enabled it can
be used
      # to set the channel the interface listens on.
      # YOU MUST CHANGE THIS TO BE THE SOURCE YOU WANT TO USE
21    source=madwifing_b,wifi0,madwifi
```

NOTE: the line numbers may change slightly per installation of kismet (newer version). The lines we are editing should be around where I specified.

Change the bolded lines to what is listed (your source may be different and will require research)

SMC2532W-B = hostap
Cisco a/b/g = madwifing_b
Proxim a/b/g = madwifing_b
Intel ipw2100 (internal tablet wireless) = ipw2100

You may also specify the source at the command line when starting kismet

```
#kismet –c ipw2100,eth1,ipw
```

NOTE: for detailed info on Source types see O'Reilly Security Power Tools p.109

You may want to change the columns that are displayed on the screen when you start kismet. Edit kismet_ui.conf

```
#vim /usr/local/etc/kismet_ui.conf

      # columns are valid.
      columns=decay,name,type,wep,channel,packets,flags,ip,size
      # What columns do we display for clients?  Comma seperated.
      clientcolumns=decay,type,mac,manuf,data,crypt,size,ip,signal,qual
      ity,noise
```

I like to see the signal strength on the screen

```
      # columns are valid.
      columns= name,type,wep,decay,channel,signal,packets,flags,ip,size
      # What columns do we display for clients?  Comma seperated.
      clientcolumns=decay,type,mac,manuf,data,crypt,size,ip,signal,qual
      ity,noise
```

Also change showintro=false and sound=false. The introscreen and sound are annoying.

Starting GPSD
Gpsd is the daemon that facilitates communication between a gps capable program (i.e. kismet) and the gps hardware. Gpsd supports most all popular gps chipsets and they can be connected via usb, serial, Bluetooth, or pcmcia/compact flash.

When connecting the gps device run dmesg from the command prompt to see how the device was detected and what filename it was assigned. This will be required when starting gpsd.

Example:

```
#dmesg –tail 10
[ 1723.424000] usb 2-1: configuration #1 chosen from 1 choice
[ 1724.200000] usbcore: registered new interface driver usbserial
[ 1724.200000] drivers/usb/serial/usb-serial.c: USB Serial support
registered for generic
[ 1724.200000] usbcore: registered new interface driver usbserial_generic
[ 1724.200000] drivers/usb/serial/usb-serial.c: USB Serial Driver core
[ 1724.228000] drivers/usb/serial/usb-serial.c: USB Serial support
registered for pl2303
[ 1724.228000] pl2303 2-1:1.0: pl2303 converter detected
[ 1724.228000] usb 2-1: pl2303 converter now attached to ttyUSB0
[ 1724.228000] usbcore: registered new interface driver pl2303
[ 1724.228000] drivers/usb/serial/pl2303.c: Prolific PL2303 USB to serial
adaptor driver
```

When starting gpsd

```
#gpsd /dev/ttyUSB0
```

Test to make sure it works:

```
#telnet 127.0.0.1 2947
```

When in telnet type **R** and you should see gpsd spit gps output to the screen (NOTE: you don’t need to be outside to test this)
ctrl-] then **quit** to exit out of telnet.

By default kismet is configured to connect to gpsd on port 2947.

Creating Google Maps with Kismet

gpsmap options:

-j = use googlemaps

-o = output file

-u = draw convex hull of data points

-r = draw estimated range circles

```
gpsmap -j -o gpsdata.js -u -r <path_to_Kismet_Gps_File>
```

After running **gpsmap** on a **.gps** file copy the output **.js** file to the same folder as the **index.html** file and name it **gpsdata.js**. You will also need to make the file world readable.

```
#cp gpsdata.js  /var/www/gpsmap
#chmod +r /var/www/gpsmap/gpsdata.js
```

Now browse to your webserver with a browser (e.g. **http://localhost/gpsmap/index.html**)

Ettercap

When I connect to an access point like to use ettercap to sniff all the traffic and capture any passwords that may be flying through the air.

```
#apt-get install ettercap-gtk ettercap-common libnet1
```

Aircrack

Download the latest version of aircrack-ng from http://www.aircrack-ng.org (current version is 1.0-rc3).

Openssl libraries will need to be installed

```
#apt-get install libssl-dev

#tar zxvf aircrack-ng-1.0-rc3.tar.gz
#cd aircrack-ng-1.0-rc3/
#make && make install
```

Karmasploit

There was an older exploit software called Karma, written in Ruby, that allowed you to mimic host APs, changing its SSID on the fly to lure clients in to connect to the attackers host. Karma relied on a patched version of the Madwifi drivers that are no longer supported and won't compile on modern Linux systems. Enter Karmasploit, a rewrite of Karma that integrates with the Metasploit framework (nice that Metasploit was rewritten in Ruby :-)

First we need to make sure we can inject packets with our wireless setup.

Setting up Proxim 8480-FC (802.11 a/b/g) and testing packet injection

```
#wlanconfig ath0 destroy
#wlanconfig ath create wlandev wifi0 wlanmode monitor
#aireplay-ng --test ath1
07:25:32  Trying broadcast probe requests...
07:25:32  Injection is working!
07:25:34  Found 1 AP

07:25:34  Trying directed probe requests...

07:25:34  00:0C:E5:4E:F1:18 - channel: 6 - 'D8F'
07:25:36  Ping (min/avg/max): 2.050ms/45.083ms/66.677ms Power: 23.93
07:25:36  27/30:  90%
```

SUCCESS...

From http://trac.metasploit.com/wiki/Karmetasploit

Metasploit does not have a DHCP module, so a third-party DHCP service must be configured and installed. The easiest way to accomplish this is by installed the ISC "dhcpd" package for your distribution. On Ubuntu-based systems, the package is called "dhcpd3" (sudo apt-get install dhcpd3). Once the DHCP server has been installed, an appropriate configuration file needs to be created. This file is normally called "dhcpd.conf" or "dhcpd3.conf" and resides in /etc, /etc/dhcp, or /etc/dhcp3. The example below uses the 10.0.0.0/24 network with the access point configured at 10.0.0.1.

We will download the latest source from https://www.isc.org/downloadables/12 and compile and install it (currently version 4.1.0).

```
#tar zxvf dhcp-4.1.0.tar.gz
#cd dhcp-4.1.0
#./configure
#make && make install

#mkdir /etc/dhcp
#gedit /etc/dhcp/dhcpd.conf

      option domain-name-servers 10.0.0.1;

      default-lease-time 60;
      max-lease-time 72;

      ddns-update-style none;

      authoritative;

      log-facility local7;

      subnet 10.0.0.0 netmask 255.255.255.0 {
        range 10.0.0.100 10.0.0.254;
        option routers 10.0.0.1;
        option domain-name-servers 10.0.0.1;
      }

#touch /etc/dhcp/dhcpd.leases
#chmod 777 /etc/dhcp/dhcpd.leases
```

Dhcp is now configured for what we need it to do.   The command to run dhcp will be found later in the document.

Install the latest version of Metasploit and configure it to use a database backend to support Karmasploit.

```
#apt-get install subversion
#apt-get install ruby ruby1.8-dev libruby1.8 libdbd-sqlite3-ruby1.8 irb1.8
rdoc1.8 libreadline-ruby1.8 rubygems libsqlite3-dev sqlite3
#svn co http://metasploit.com/svn/framework3/trunk msf3
#gem install activerecord sqlite3-ruby
```

Get it up and running!

```
#wlanconfig ath0 destroy
#wlanconfig ath0 create wlandev wifi0 wlanmode monitor
#airbase-ng -P -C 30 -e "Free WiFi" -v ath1
#ifconfig at0 up 10.0.0.1 netmask 255.255.255.0
#dhcpd -cf /etc/dhcp/dhcpd.conf at0
#cd ~/msf3
#./msfconsole -r karma.rc
```

```
#iptables -t nat -A PREROUTING -i at0 -j REDIRECT
```

References:

http://blog.metasploit.com/2006/09/metasploit-30-automated-exploitation.html

http://carnal0wnage.blogspot.com/2008/08/playing-with-karmasploit-part-1.html

http://trac.metasploit.com/wiki/Karmetasploit

http://www.vulnerabilityassessment.co.uk/wmap_meta.htm

http://www.linux.com/archive/feature/61609