Global Information Assurance Certification Paper

WI-FI

WITH

BACKTRACK

Author: Antonio Merola
a.merola >at< securityindepth >org<

Adviser: Charles
mrcorp >at< yahoo >com<

Accepted: March 12th 2007

Abstract

This paper is the technical report required to obtain the GIAC Gold for Assessing and Auditing Wireless Networks (GAWN) security certification.

F*rom GIAC website*: "*The GIAC Gold option for certification is assurance that a certified individual understands and can communicate the knowledge and skills necessary in key areas of information security. GIAC Gold is distinguished from the existing exam-only 'GIAC Silver' certification by requiring candidates to complete a technical report covering an important area of security related to the certification the student is seeking.*"

The idea behind this paper is to help auditors (especially whom not familiar with Linux) with wireless issues; it is a real hassle getting wireless works, either simply joining a network as legitimate client or conducting wireless audit, along with the plethora of tools available to wireless PenTesters. Before you eventually "go off", after days gone-by looking here and there, have a look to this guide, I do really hope you master Wi-Fi with BackTrack after this reading. A note about this paper: it is not the umpteenth guide on how to crack WEP or WPA; I wrote this paper across a period of five months, the descriptions and procedures used and the conclusions obtained are as much accurate as possible, however during this periods something could have been changed a/o not still valid, please be sure to check in case you assume this possibility.

Acknowledgments

I wish to thanks Joshua Right Author of the SANS track "Assessing and Auditing Wireless Networks" for the amazing

© SANS Institute 2007, Author retains full rights.

educational aid prepared for SANS students, thanks to the SANS instructor Raul Siles, he has been giving me always good advices. A thanks goes to the GIAC Gold Adviser Mr. Charles who reviewed the paper and, of course, to all the open-source developers and BackTrack developers for their impressive contribution to the security community.

"Curiosity has its own reason for existing… It is enough if one tries merely to comprehend a little of this mystery every day."

Albert Einstein

Outline

Acronyms

AP	Access Point
BT	BackTrack Linux Live Distribution
MAC	Medium Access Control
RF	Radio Frequency
RFMON	Radio Frequency Monitoring
SSID	Service Set Identifier
WEP	Wired Equivalency Protocol
WLAN	Wireless LAN
WNIC	Wireless Network Interface Card
WPA	WiFi Protected Access
OSSTMM	Open Source Security Testing Methodology Manual
ISSAF	Information Systems Security Assessment Framework
USB	Universal Serial Bus
PCMCIA	Personal Computer Memory Card International Association
IEEE	Institute of Electrical and Electronics Engineers
PCI	Peripheral Component Interconnect
ID	Identifier
SWAT kit	Sans Wireless Auditing Tool kit

1 Introduction

BackTrack [*ref.1*] (BT) is a live CD based on Slax, hence Slackware, it is evolved from the widely adopted Whax and Auditor security distributions.

Slackware is one of the many Linux distribution, Slax is a linux live-distro version based on Slackware. BackTrack is a Penetration Testing oriented live-distro based on Slax;


```
BT# cat /etc/slax.version
SLAX 5.0.7
BT# cat /etc/slackware.version
Slackware 10.2.0
```

BT is the primary of the free projects maintained by the remote-exploit team, the latest version is optimized to be used by security penetration testers. Nowadays, security professionals have been using it as their favorite toolbox. Beyond its' interesting features and available tools what, makes BT interesting is that it is also aligned to the penetration testing methodologies and assessment frameworks of ISSAF [*ref.2*] and OSSTMM [*ref.3*], thereby BT tools are logically structured according to the work flow of security professionals.

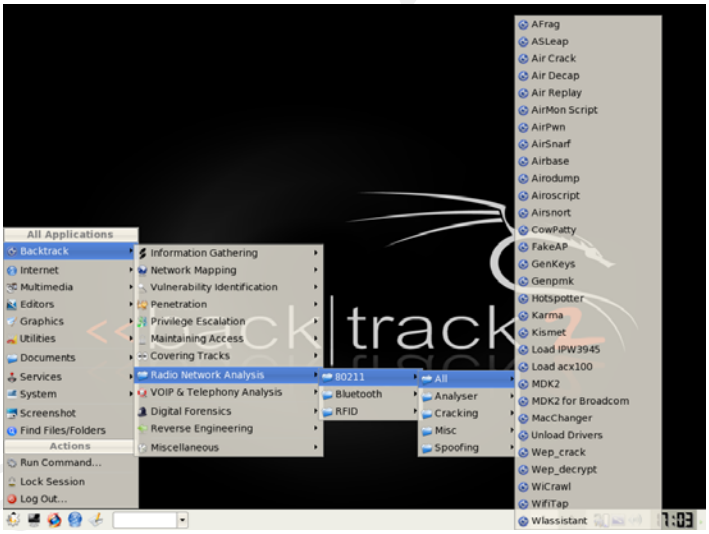
By the way, there is a Linux based tool for IT-Security called DVL (Damn Vulnerable Linux) [*ref.4*]. It is a training environment initiated for university educational purposes and is based on BackTrack 2 and is made to be as insecure as possible; it includes a lesson-based environment for attack & defence that is designed for self-study or teaching activities.

2 BT WiFi tools

BT has an intuitive layout, some tools are available in the menu and invoke automated scripts; most of the analysis tools are located either in the path or in the */pentest* directory. As we can see in the following figure, it is possible to explore wireless tools under */pentest/wireless*:

```
BT ~ # ls -la /pentest/wireless
total 64
drwxr-xr-x 15 root root 4096 Mar  6 08:15 ./
drwxr-xr-x 22 root root 4096 Mar  6 06:35 ../
drwxr-xr-x  2 root root 4096 Feb 27 23:38 afrag-0.1/
drwxr-xr-x  9 root root 4096 Mar  6 07:07 aircrack-ng/
drwxr-xr-x  5 root root 4096 Sep 23  2006 airpwn-1.3/
drwxr-xr-x  5 root root 4096 Oct  8  2006 airsnarf-0.2/
drwxr-xr-x  5 root root 4096 Oct  8  2006 asleap-1.4/
drwxr-xr-x  3 root root 4096 Oct  8  2006 fakeap-0.3.2/
drwxr-xr-x  2 root root 4096 Oct  8  2006 hotspotter-0.4/
drwxr-xr-x  7 root root 4096 Jan 25  2006 karma-20060124/
drwxr-xr-x  2 root root 4096 Feb 27 23:25 mdk2-v31/
drwxr-xr-x  2 root root 4096 Feb 27 23:28 mdk2-v31-bcm43xx/
drwxr-xr-x  2 root root 4096 Feb 27 23:37 ska-0.2/
-rwxr-xr-x  1 root root  207 Mar  6 07:06 update-aircrack.sh*
drwxr-xr-x  2 root root 4096 Oct  8  2006 wep_tools/
drwxr-xr-x  2 root root 4096 Mar  6 08:15 wifitap/
```

While using the GUI pointing to ***"K Menu->Backtrack->Radio Network Analysis->80211->All"*** we see also tools not listed before, such as *kismet*, which is actually located under */usr/local/bin:*



The following table lists all wireless tools included in BT with a short description and homepage:

Tool Name	Description	Homepage
AFrag	First implementation of the Fragmentation Attack on Linux.	http://homepages.tu-darmstadt.de/~p_larbig/wlan/afrag-0.1.tar.bz2
ASLeap	This tool is released as a proof-of-concept to demonstrate weaknesses in the LEAP and PPTP protocols.	http://asleap.sourceforge.net/
Air Crack	Aircrack is an 802.11 WEP and WPA-PSK keys cracking program that can recover keys once enough data packets have been captured. It implements the standard FMS attack along with some optimizations like KoreK attacks, thus making the attack much faster compared to other WEP cracking tools. In fact, aircrack is a set of tools for auditing wireless networks.	http://www.aircrack-ng.org/
Air Decap	decrypts WEP/WPA capture files. Part of the aircrack suite.	http://www.wirelessdefence.org/Contents/Aircrack_airdecap.htm
Air Replay	802.11 packet injection program. Part of the aircrack suite.	http://www.wirelessdefence.org/Contents/Aircrack_aireplay.htm
Airmon Script	utility to check a wifi interfaces status and to set the interface in monitor mode. Part of the aircrack suite.	http://www.wirelessdefence.org/Contents/Aircrack_airmon.sh.htm
Airpwn	Airpwn requires two 802.11 interfaces in the case where driver can't inject in monitor mode (lots of chipsets do nowadays, see HCL:Wireless for a list). It uses a config file with multiple config sections to respond to specific data packets with arbitrary content. For example, in the HTML goatse example, we look for any TCP data packets starting with "GET" or "POST" and respond with a valid server response including a reference to the canonical goatse image.	http://airpwn.sourceforge.net/
AirSnarf	Airsnarf is a simple rogue wireless access point setup utility designed to demonstrate how a rogue AP can steal usernames and passwords from public wireless hotspots. Airsnarf was developed and released to demonstrate an inherent vulnerability of public 802.11b hotspots--snarfing usernames and passwords by confusing users with DNS and HTTP redirects from a competing AP	http://airsnarf.shmoo.com/
Airbase	Airbase is the name of a collection of wireless utilites. Included in airbase you will find an aircrack re-implementation, a distributed wep cracker (now with FPGA support), a library to help you craft/parse 802.11 packets, and various other supporting utilities. At the core of airbase is a C++ library called libairware.	http://www.802.11mercenary.net/
Airodump	802.11 packet capture program. Part of the aircrack suite.	http://www.wirelessdefence.org/Contents/Aircrack_airodump.htm
Airoscript	aircrack-ng based wireless cracking script (must mkdir /home/root to function out of the box)	http://daouid.googlepages.com/airoscriptwepcrackingscript
Airsnort	AirSnort is a wireless LAN (WLAN) tool which recovers encryption keys. AirSnort operates by passively monitoring transmissions, computing the encryption key when enough packets have been gathered.	http://airsnort.shmoo.com
CowPatty	Cowpatty is designed to audit the pre-shared key (PSK) selection for WPA networks based on the TKIP protocol. A while back, Robert Moskowitz published a paper titled "Weakness in Passphrase Choice	http://sourceforge.net/projects/cowpatty

	in WPA Interface" that described a dictionary attack against wireless networks using the TKIP protocol with a pre-shared key (PSK). Supply a libpcap file that includes the TKIP four-way handshake, a dictionary file of passphrases to guess with and the SSID for the network	
FakeAP	Black Alchemy's Fake AP generates thousands of counterfeit 802.11b access points. Hide in plain sight amongst Fake AP's cacophony of beacon frames. As part of a honeypot or as an instrument of your site security plan, Fake AP confuses Wardrivers, NetStumblers, Script Kiddies, and other undesirables	http://www.blackalchemy.to/project/fakeap/
GenKeys	generates lookup file for asleap	http://asleap.sourceforge.net/
Genpmk	genpmk is used to precompute the hash files in a similar way to Rainbow tables is used to pre-hash passwords in Windows LANMan attacks. There is a slight difference however in WPA in that the SSID of the network is used as well as the WPA-PSK to "salt" the hash. This means that we need a different set of hashes for each and every unique SSID i.e. a set for "linksys" a set for "tsunami" etc..	http://www.wirelessdefence.org/Contents/coWPAttyMain.htm
Hotspotter	Hotspotter passively monitors the network for probe request frames to identify the preferred networks of Windows XP clients, and will compare it to a supplied list of common hotspot network names. If the probed network name matches a common hotspot name, Hotspotter will act as an access point to allow the client to authenticate and associate. Once associated, Hotspotter can be configured to run a command, possibly a script to kick off a DHCP daemon and other scanning against the new victim.	http://www.remote-exploit.org/codes_hotspotter.html
Karma	KARMA is a set of tools for assessing the security of wireless clients at multiple layers. Wireless sniffing tools discover clients and their preferred/trusted networks by passively listening for 802.11 Probe Request frames. From there, individual clients can be targetted by creating a Rogue AP for one of their probed networks (which they may join automatically) or using a custom driver that responds to probes and association requests for any SSID. Higher-level fake services can then capture credentials or exploit client-side vulnerabilities on the host.	http://theta44.org/karma/index.html
Kismet	Kismet is an 802.11 layer2 wireless network detector, sniffer, and intrusion detection system. Kismet will work with any wireless card which supports raw monitoring (rfmon) mode, and can sniff 802.11b, 802.11a, and 802.11g traffic. Kismet identifies networks by passively collecting packets and detecting standard named networks, detecting (and given time, decloaking) hidden networks, and infering the presence of nonbeaconing networks via data traffic.	http://www.kismetwireless.net/
Load IPW3945	an open source 802.11a/b/g driver for the Intel PRO/Wireless 3945ABG Network Connection	http://ipw3945.sourceforge.net/
Load acx100	The ACX100/ACX111/TNETW1450 wireless network driver project	http://acx100.sourceforge.net/
MDK2 and MDK2 for Broadcom	Bruteforce hidden SSIDs (small SSID wordlist included), probe networks for checking if they can hear you, Authentication-DoS to freeze APs (with checking for success), Beacon Flooding with channel hopping (can crash NetStumbler and some buggy drivers), Disconnects everything found (aka AMOK-MODE) with DeAuth and DisAssoc packets (Don't try this where they can kick your ass! ;D), WPA TKIP Denial-of-Service	http://homepages.tu-darmstadt.de/~p_larbig/wlan/
MacChanger	A GNU/Linux utility for viewing/manipulating the MAC address of network interfaces	http://www.alobbs.com/macchanger/
Unload	Script to unload ieee wifi driver	n/a

Drivers		
Wep_crack	WepAttack is a WLAN open source Linux tool for breaking 802.11 WEP keys. This tool is based on an active dictionary attack that tests millions of words to find the right key. Only one packet is required to start an attack.	http://sourceforge.net/projects/wepcrack/
Wep_decrypt	a program for decrypting captured 802.11 traffic that is protect with WEP traffic. It reads in a pcap capture file, such as that generated by prismdump, and outputs another pcap capture file with decrypted packets. By default it will read from stdin and ouput to stdout. The key to decrypt with can be specified as a string of hex characters, optionally seperated by spaces or colons, or as a text string. If a text string is specified, the actual keying material will be generated by the string in the (ad hoc) standard fashion used by many drivers.	http://www.lava.net/~newsham/wlan/wep_tools.tgz
WifiTap	Wifitap is a proof of concept for communication over WLAN networks using traffic injection. Wifitap allows direct communication with an associated station to a given access point directly, whilst not being associated ourselves or being handled by access point.	http://sid.rstack.org/index.php/Wifitap_EN
Wicrawl	wicrawl is an automated wifi scanner and auditor. It implements common tools to perform checks (association, dhcp, wep cracking, bruteforcing wpa-psk, etc) against the discovered access point list based on profile settings. It can use multiple cards to run checks against multiple APs at the same time.	http://midnightresearch.com/projects/wicrawl
Wlassistant	Wireless Assistant scans for wireless access points and displays link quality, encryption and other useful information. When user wants to connect to a network, Wireless Assistant opens up its wizards and guides the user through Wi-Fi settings. After a successful connection is made the settings are remembered so next time the user won't have to enter them again.	http://wlassistant.sourceforge.net/

3 BT Graphical User Interfaces tools

BT is provided with KDE environment, hence there are GUI tools included to handle wireless:

Tool Name	Description	Located at
kWiFiManager	*Wireless LAN Manager for KDE*	*K Menu->System->Settings or* *K Menu->Internet->*
kcmwifi	*tool to set up wireless LAN*	*K Menu->System->Settings->Internet & Network->*
kcmusb	*To to view the USB devices*	*K Menu->System->System Information->*
wlassistant	see on previous table	*K Menu->Internet-> or* *K Menu->Backtrack->Radio Network Analysis->80211->Misc*

4 BT WiFi drivers

BT V.2.0 (Final) has the following drivers included, in addition to the standard 2.6.20 kernel drivers:

Driver	Description	Home page
madwifi-ng (Patched for Injection)	*multiband Atheros driver*	*madwifi.org*
hostap (Patched for Injection)	*driver for Intersil Prism2 – 2.5 – 3 and WPA supplicant*	*hostap.epitest.fi*
prism54 (Patched for Injection)	*driver for Prism ISL38xx based devices*	*prism54.org*
bcm43xx (Patched for Injection)	*Broadcom 43xx driver*	*bcm43xx.berlios.de*
rtl8180/8185/8187/8225 (Patched for Injection)	*driver for Realtek's cards*	*rtl8180-sa2400.sourceforge.net*
ipw2200 (Patched for Injection)	*drivers for Intel Pro ipw2200 and ipw2915*	*ipw2200.sourceforge.net*
ipw2100	*driver for Intel Pro 2100 Centrino (b)*	*ipw2100.sourceforge.net*
ipw3945	*driver for Intel Pro 3945*	*ipw3945.sourceforge.net*
rt2570 (ASPj's Drivers)		
rt2500/2570/61/73	*drivers for Ralink chipset*	*rt2x00.serialmonkey.com*
acx100	*drivers for Texas Instruments acx100/111/tnetw1450*	*acx100.sourceforge.net*
zd1211rw	*driver for cards based on the ZyDAS ZD1211*	*linuxwireless.org/en/users/drivers/zd1211rw*
wlan-ng	*has been removed from BT, for Prism2 cards use the Hostap Drivers (see Senao Card later on)*	*www.linux-wlan.org*

5 Configure Network scripts

When BT boots up, ***/etc/rc.M*** gets executed, and runs the other *rc* files in a specific order. ***rc.inet1*** is used to configure the interfaces and ***rc.inet2*** is used to start various network services. ***rc.local*** gets executed towards the end of the process and should be used for initialization of things that need to be started towards the end of the init process.
Wireless networking isn't easily supported in Linux as wired networking. In order to connect to a wireless network, usually there are three steps:

1. verify hardware support for the wireless card;
2. configure the card to connect to the AP;
3. configure the network.

Hardware support for a wireless card is provided through the kernel, either with a module or built into the kernel. Generally cards are provided through kernel modules, so we'll want to determine the appropriate kernel module and load it through */etc/rc.d/rc.modules*.

netconfig may not detect our wireless card, so we'll probably need to determine the card ourself. The vast majority of this work is done by *iwconfig*.

First, we'll want to configure our wireless AP. We'll need at least the following information:

- name of the network (the ESSID)
- channel the AP uses;
- encryption settings, (hopefully not WEP!);

Once we have the above information, and assuming we've used *modprobe* to load the appropriate kernel driver, we can edit */etc/rc.d/rc.wireless.conf* modifying the generic section with our *essid*, *key* and *channel*. The variable names in *rc.wireless.conf* correspond to the *iwconfig* parameters, are read by *rc.wireless* script and used in the appropriate *iwconfig* commands. Using the About keys, we'll learn about setting our key in hexadecimal, that's because we can be confident that the AP and *iwconfig* will agree on the key; while if we have a string, we can't be sure how our AP will translate that into a hexadecimal key.

Moving forward we will have a closer look at *iwconfig* and how it works. As a rule of thumb, the first step is to tell our WNIC the network to join:

```
BT ~ # iwconfig ath0 essid network name
```

replace *ath0* with the real network interface and change the *network_name* to the *essid* to join to. Next we'll have to specify the encryption key used:

```
BT ~ # iwconfig ath0 key type_the_key_here
```

Finally we'll have to specify the frequency channel to use (the *n* parameter).

```
BT ~ # iwconfig ath0 channel n
```

6 Setup a WNIC (Wireless Network Interface Card)

A wireless network interface card (WNIC) is a network card that connects to a radio-based network. The card uses an antenna to communicate through microwaves. The IEEE 802.11 is the standard, which sets specifications about, how wireless networks operate.

A WNIC in a laptop computer is mostly provided as a built in interface, other spread options are the USB dongle and PCMCIA cards. A WNIC can operate in two modes: infrastructure mode and ad hoc mode. In infrastructure mode, the WNIC needs an access point (data is transferred using the access point as the central hub and all wireless nodes connect to it). All nodes connecting to the access point must share the same service set identifier (SSID) with the access point, and if the access point is configured with some encryption mechanism they must share the same. In ad-hoc mode, the WNIC doesn't require an access point, but can directly connect with all other wireless nodes. Of course, all the nodes in an ad-hoc network must share the same SSID.

Before connecting to a wireless network with BT, we need a working wireless card, basically any O.S. needs to load correct drivers for the wireless card. Unlikely, this is not an easy task because often, some vendors doesn’t release specification of the hardware or provide Linux drivers for their wireless card. We will test, later on, some WNICs with BT in order to better understand how to get through this process. My assumed advice is to avoid problems at the beginning by buying a WNIC already tested with BT. To accomplish this, please search the BT Wireless Compatibility List [*ref.5*]. If a WNIC is already in place, verify it works properly. The first step is to know the card name and chipset name, we can obtain this with "*lspci*", if it is a pci card or "*lsusb*" for usb cards.

These id's are checked against a public repository of all known ID's used in PCI or USB devices, that is ID's of vendors, devices, subsystems and device classes; sometimes "*Unknown device*" as output might be reported, in this case, the first step is to check against the latest available database [*ref.6*]. The following example describes this circumstance, the output shows a BT version installed on a new Apple Intel MAC Book for testing purpose:

```
BT ~ # uname -a
Linux BT 2.6.18-rc5 #4 SMP Mon Sep 18 17:58:52 GMT 2006 i686 i686 i386 GNU/Linux
```

When it attempted to recognize the Airport wireless card, an "*Unknown device*" was reported, after the new DB file copy (*pci.ids*) the query showed different information:

```
BT ~ # lspci
……
02:00.0 Network controller: Atheros Communications, Inc.: Unknown device 0024 (rev 01)
BT ~ # cp /tmp/pci.ids /usr/share
BT ~ # lspci
……
02:00.0 Network controller: Atheros Communications, Inc. AR5418 802.11a/b/g/n Wireless
PCI Express Adapter (rev 01)
```

This output is quite useful, because after correctly identifying the chipset, it is possible to verify which driver, if any, is known to work with the card. With some cardbus adapters, it might be useful to run the PCMCIA card control utility "*lspcmcia*", in order to get info about the card; see the following two pcmcia cards under test, where in the first example the utility itself suggest to run *lspci*:

```
BT ~ # lspcmcia
Socket 0 Bridge:        [yenta_cardbus]         (bus ID: 0000:02:06.0)
  CardBus card -- see "lspci" for more information
Socket 1 Bridge:        [yenta_cardbus]         (bus ID: 0000:02:06.1)
```

while for the second card the device and the loaded driver are showed:

```
BT ~ # lspcmcia
Socket 0 Bridge:        [yenta_cardbus]         (bus ID: 0000:02:06.0)
Socket 0 Device 0:      [orinoco_cs]            (bus ID: 0.0)
Socket 1 Bridge:        [yenta_cardbus]         (bus ID: 0000:02:06.1)
```

One useful resource for identifying the chipset from the card manufacturer is available online [ref.7]. If the specific card is not listed there, we can search Google with the card name and keyword "chipset" (e.g. "Cisco Aironet AIR-CB21AG-A-K9 chipset").

7 Setup a WNIC in monitor mode

In order to audit a wireless network or run some tools, the interface needs to be set in *monitor mode*. Monitor mode is also known as *RFMON* mode or raw mode and allows a computer with a WNIC to monitor all traffic from the wireless network. Monitor mode is similar to promiscuous mode used for sniffing on wired networks, it is also used for determining the presence of base stations avoiding active probing. Usually the wireless card is unable to inject packets in monitor mode unless you configure the card in injection mode; unfortunately sometimes the driver doesn't support injection mode or isn't able to do both, injection and monitor in the same time.

Basically in monitor mode the driver will forward all traffic received on the radio interface to the operating system without converting the frames to the 802.3 format, hence it is possible to catch additional information by examining the management traffic and analyzing the traffic header. The primary advantage of monitor mode is that the sniffer doesn't need to be connected to the network, so no credentials are needed.

As a rule of thumb, on Linux, to set the WNIC in monitor mode the tool to invoke is "*iwconfig*":

- ***iwconfig:*** is the command line tool used to configure a wireless adapter with parameters such as SSID, operating mode, frequency etc.;

which is different from the tool *ifconfig*:

- ***ifconfig:*** is the command line tool used to configure any network interface with IP, netmask etc and to change the interface status (up or down);

However this setting is strictly linked to the driver of the adapter, hence sometimes the driver needs the "*iwpriv*" or the "*wlanconfig*" tool:

- ***iwpriv:*** is the command line tool used to configure a wireless adapter, it is the companion tool to "*iwconfig*" and deals with parameters and setting specific to each driver (as opposed to "*iwconfig*" which deals with generic ones);
- ***wlanconfig:*** is the command line tool used to configure a wireless adapter with the madwifi driver;

8 WNIC under test:

- **Cisco Aironet AIR-CB21AG-A-K9**

This is an 802.1x a/b/g PCMCIA cardbus adapter. When the card is inserted, "*lspci*" shows:

```
BT ~ # lspci
……
06:00.0Ethernet controller: Atheros Communications, Inc.AR5212 802.11abg NIC(rev 01)
```

The following is a verbose output of "*lspci -v*", followed by "*iwconfig ath0*":

```
BT ~ # lspci -v
……
02:00.0  Ethernet controller: Atheros Communications, Inc. AR5212 802.11abg NIC (rev
         01)
         Subsystem: AIRONET Wireless Communications Unknown device cb21
         Flags: bus master, medium devsel, latency 168, IRQ 11
         Memory at 34000000 (32-bit, non-prefetchable) [size=64K]
         Capabilities: [44] Power Management version 2

BT ~ # iwconfig ath0
ath0     IEEE 802.11b  ESSID:""  Nickname:""
         Mode:Managed  Channel:0  Access Point: Not-Associated
         Bit Rate:0 kb/s   Tx-Power:0 dBm   Sensitivity=0/3
         Retry:off   RTS thr:off   Fragment thr:off
         Encryption key:off
         Power Management:off
         Link Quality:0 Signal level:0 Noise level:0
         Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
         Tx excessive retries:0 Invalid misc:0 Missed beacon:0
```

BT recognize this chipset as Atheros, this card works with the included *madwifi-ng* driver, hence "*iwconfig*" immediately report the *ath0* interface, thus correctly connected to a wireless network with and without encryption.

I had only to fix a problem with WPA protection because the "*wpa_supplicant*" tool provided with BT version *2.16.18-rc5* was without madwifi driver support, so the tool needed to be recompiled accordingly. Albeit, this isn't a problem with BT's latest version *2.6.20-BT-PwnSauce-NOSMP*. It might be interesting to show how-to fix it in case the same problem arises with other flavors of Linux. So, in order to connect to a WPA network, recompile the "*wpa_supplicant*" tool with the madwifi driver support, copying the following to "*/usr/src/wpa_supplicant-0.5.7/.config*":

```
CONFIG_DRIVER_MADWIFI=y
CFLAGS += -I/usr/src/madwifi-ng
CONFIG_CTRL_IFACE=y
```

(including the */usr/src/madwifi-ng* source directory is necessary);

```
BT ~ # cd /usr/src/wpa_supplicant-0.5.7
BT ~ # make clean
BT ~ # make
BT ~ # make install
BT ~ # wpa_passphrase wifi_test "my_test_key" > /etc/wpa_supplicant.conf
BT ~ # iwconfig ath0 essid "wifi_test"
BT ~ # wpa_supplicant -Bw -D madwifi -iath0 -c/etc/wpa_supplicant.conf
BT ~ # ifconfig ath0 up
BT ~ # dhcpcd ath0
```

With these few modifications everything works fine joining a WPA network as a legitimate client. The monitor mode for this adapter is configurable in this way:

```
BT ~ # wlanconfig ath0 create wlandev wifi0 wlanmode monitor
```

wifi0 is a virtual device, note the presence of two cards (*ath0 - wifi0*) even if there is one WNIC; this is why rather than using the *ath0*, we could set the monitor mode up also with *ath1* and use that one typing:

```
BT ~ # wlanconfig ath1 create wlandev wifi0 wlanmode monitor
```

Please note that the successful monitor mode settings can also be verified with the presence of additional 00's in the MAC address (HWaddr) appended to the MAC address of the card:

```
BT ~ # ifconfig wlan0
wifi0     Link encap:UNSPEC  HWaddr 00-40-96-AD-B4-89-30-3A-00-00-00-00-00-00-00-00
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4036 errors:0 dropped:0 overruns:0 frame:2
          TX packets:1085 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:199
          RX bytes:979368 (956.4 KiB)  TX bytes:96742 (94.4 KiB)
          Interrupt:10
```

this is because *ifconfig* does not understand the monitor mode, so it appears as an unknown or UNSPEC (unspecified) link encapsulation mechanism; thus *ifconfig* displays a default of 16 bytes string where 8 bytes are printed to represent the MAC address, followed by 8 NULL bytes; actually the first 6 bytes represent the real MAC address of the wnic, followed by 2 bytes of uninitialized memory.

- **Senao NL-2511CD PLUS EXT2 (SWAT kit)**

This is what the wireless adapter received during the SANS course, part of the SWAT kit. It is a Senao card; these cards are based on the IntersilPrism2 chipset. Currently, the 2511 series uses the Prism 2.5 chipset. This is a favorite wireless card bus adapter for war driving/walking because of its high power, 200mw, and superior receiver sensitivity; it has two MMCX external antenna connectors that allow you to couple the high power radio with high gain antenna.

```
BT ~ # iwconfig
...........
eth1      IEEE 802.11b  ESSID:""  Nickname:"Prism  I"
          Mode:Managed  Access Point: Not-Associated   Bit Rate:11 Mb/s
          Sensitivity:1/3
          Retry limit:8   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality=0/92  Signal level=-68 dBm  Noise level=-122 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

Some discussion about this card are linked to the firmware version; in order to know this information or others IDs info, invoke the *hostap_diag* tool:

```
BT ~ # hostap_diag wlan0
Host AP driver diagnostics information for 'wlan0'

NICID: id=0x800c v1.0.0 (PRISM II (2.5) PCMCIA (SST parallel flash))
PRIID: id=0x0015 v1.1.0
STAID: id=0x001f v1.8.0 (station firmware)
```

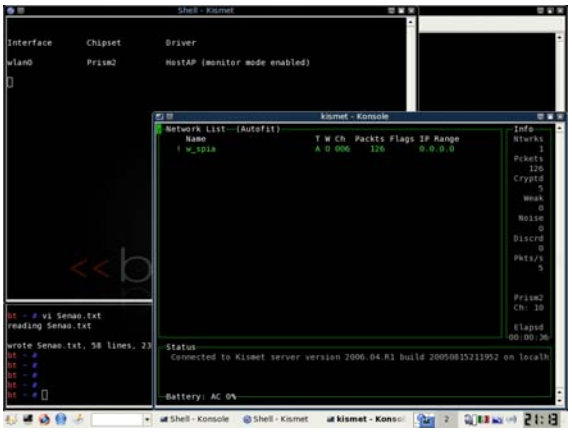
According to the BT open wiki [ref.8], the wlan-ng driver has been removed from BT2 final, but it is possible to run the HostAP driver with the Prism2 cards. Basically, we have to eject the card, remove the orinoco driver that gets loaded by default, load the HostAP driver and insert the card again:

```
BT ~ # pccardctl eject
BT ~ # rmmod orinoco_cs
BT ~ # rmmod orinoco
BT ~ # rmmod hermes
BT ~ # modprobe hostap_cs
BT ~ # pccardctl insert
BT ~ # lspcmcia
Socket 0 Bridge:        [yenta_cardbus]         (bus ID: 0000:02:06.0)
Socket 0 Device 0:      [hostap_cs]             (bus ID: 0.0)
Socket 1 Bridge:        [yenta_cardbus]         (bus ID: 0000:02:06.1)
BT ~ # iwconfig wlan0
wlan0     IEEE 802.11b  ESSID:"test"  Nickname:""
          Mode:Master  Access Point: Not-Associated   Bit Rate:11 Mb/s
          Sensitivity=1/3
          Retry limit:8   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality:0  Signal level:0  Noise level:0
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

Note that the card is now *wlan0* instead of the previous *eth1;* to set the card in monitor mode run "*iwconfig*"

```
BT ~ # ifconfig wlan0 up
BT ~ # iwconfig wlan0 mode monitor
bt ~ # ifconfig wlan0
wlan0     Link encap:UNSPEC  HWaddr 00-02-6F-3D-69-A4-00-21-00-00-00-00-00-00-00-00
          inet6 addr: fe80::202:6fff:fe3d:69a4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:805 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:115055 (112.3 KiB)  TX bytes:576 (576.0 b)
          Interrupt:3 Base address:0x2100
```

(Note the presence of 0's in the output, after setting the monitor mode). The following figure shows kismet running using the wlan0 interface with monitor mode enabled.



- **MacBook Intel AirPort**

The wireless adapter provided with this laptop is built-in AirPort Extreme; as outlined before it is recognized as "***Atheros Communications, Inc. AR5418 802.11a/b/g/n Wireless PCI Express Adapter***" a "*lspci -n*" shows:

```
BT ~ # lspci -n
…
02:00.0 Class 0280: 168c:0024 (rev 01)
```

By the way in this output, ***168c*** is the vendor ID and ***0024*** is the product ID.

When I started this document, the HAL of madwifi driver didn't include this chipset, therefore it didn't work; for further information there is a ticket at *madwifi.org*. Fortunately the ticket is now closed and this wnic seems to work with madwifi-hal-0.9.30.13; the last comment about this topic is "*FYI: the madwifi-hal-0.9.30.13 branch has been merged to trunk (and the branch has been removed). If you don't want to wait until the next release (v0.9.4), you could go with a snapshot [ref.8] or checkout from trunk - just make sure that your code is >= r2360.*

Basically just download and compile the latest madwifi-ng driver (at the time of this writing is madwifi-ng-r2512-20070624.tar.gz).

```
BT madwifi-ng-r2484-20070619 # make
BT ~ # modprobe ath_pci
BT ~ # iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wifi0     no wireless extensions.

ath0      IEEE 802.11b  ESSID:""  Nickname:""
          Mode:Managed  Channel:0  Access Point: Not-Associated
          Bit Rate:0 kb/s   Tx-Power:0 dBm   Sensitivity=1/1
          Retry:off   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality=0/70  Signal level=-256 dBm  Noise level=-256 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
BT ~ # ifconfig ath0 up
BT ~ # iwlist ath0 scan
ath0      Scan completed :
          Cell 01 - Address: 00:18:F8:5F:67:5E
                    ESSID:"w_test"
                    Mode:Master
                    Frequency:2.437 GHz (Channel 6)
                    Quality=52/70  Signal level=-43 dBm  Noise level=-95 dBm
                    Encryption key:on
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 18 Mb/s
                              24 Mb/s; 36 Mb/s; 54 Mb/s; 6 Mb/s; 9 Mb/s
                              12 Mb/s; 48 Mb/s
                    Extra:bcn_int=100
                    IE: IEEE 802.11i/WPA2 Version 1
                        Group Cipher : TKIP
                        Pairwise Ciphers (2) : CCMP TKIP
                        Authentication Suites (1) : PSK
                    IE: WPA Version 1
                        Group Cipher : TKIP
                        Pairwise Ciphers (2) : CCMP TKIP
                        Authentication Suites (1) : PSK
BT ~ # iwconfig ath0 essid w_test
BT ~ # iwconfig ath0 channel 6
BT ~ # wpa_passphrase w_test my_wpa_key > /etc/wpa_supplicant.conf
BT ~ # wpa_supplicant -Dwext -i ath0 -c /etc/wpa_supplicant.conf
BT ~ # dhcpcd ath0
BT ~ # ifconfig ath0
ath0      Link encap:Ethernet  HWaddr 00:17:F2:E6:A2:80
          inet addr:10.10.1.101  Bcast:10.10.1.255  Mask:255.255.255.0
          inet6 addr: fe80::217:f2ff:fee6:a280/64 Scope:Link
          UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:253 errors:0 dropped:0 overruns:0 frame:0
          TX packets:234 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:159068 (155.3 KiB)  TX bytes:29727 (29.0 KiB)
bt ~ # ping -nc 1 www.google.it
PING www.l.google.com (64.233.183.103) 56(84) bytes of data.
64 bytes from 64.233.183.103: icmp_seq=1 ttl=236 time=40.5 ms
```

9 WNIC from Windows using VMware

Using Windows as a workstation for wireless network auditing isn't a good choice due the lack of open-source wireless security tools and advanced wireless drivers; but if

you are familiar with Windows and you need to use BT, one option is to run BT inside VMware. The problem is that the virtual hardware for VMware virtual machines does not include a virtualized PCMCIA controller. Thus, a PCMCIA wireless card is mapped in VMware as a standard PCnet-II Ethernet wired card. This means that you cannot manage the wireless card settings, configure it in monitor mode or enable its traffic injection capabilities. Fortunately, the VMware USB native support allows you to use a USB wireless dongle and get access to all wireless features from BT inside VMware. If you need this feature I advise you read the RaDaJo security blog post [*ref.10*]; a video tutorial is also available [*ref.11*].

10 SANS wireless track tools

During the SANS wireless track, either because just referenced or because used during exercises, students are informed about a lot of tools. The following table lists all these tools (the bolded on the highlighted rows are already included in BT, the others free tools can be included in BT as modules):

Tool Name	Description	Homepage
Aircrack	***Already included in BT***	
Aireplay	***Already included in BT***	
AirMagnet Surveyor	AirMagnet's suite of wireless network assurance solutions.	http://www.airmagnet.com
AiroPeek NX	Expert Wireless LAN Analyzer. AiroPeek NX, WildPackets' expert wireless LAN analyzer, provides network engineers with the expert diagnostics they need to deploy, secure, and troubleshoot wireless LANs.	http://www.wildpackets.com/products/airopeek/airopeek_nx/overview
Airsnarf	***Already included in BT***	
Airsnort	***Already included in BT***	
Arpforge	Arpforge is used in conjunction with aireplay (only available on Linux) to decrypt a WEP data packet without knowing the key. At least one data packet has to be decrypted (utilising aireplay -4).	http://www.wirelessdefence.org/Contents/Aircrack_arpforge.htm
Aruba AirOS	Aruba Networks delivers an enterprise mobility solution that enables secure access to data, voice and video applications across wireless and wireline enterprise networks	http://www.arubanetworks.com

asleap	The first version of asleap simply read in an ASCII file of dictionary words and associated MD4 hashes of those words and tried to brute-force the LEAP challenge and response exchange. The new version of asleap has a bunch of interesting features, check it out!	http://asleap.sourceforge.net/
Cain & Abel	a password recovery tool for Microsoft Operating Systems. It allows easy recovery of various kind of passwords by sniffing the network, cracking encrypted passwords using Dictionary, Brute-Force and Cryptanalysis attacks, recording VoIP conversations, decoding scrambled passwords, recovering wireless network keys, revealing password boxes, uncovering cached passwords and analyzing routing protocols.	http://www.oxid.it
chch2ww	Convert chopchop file to WEPWedgie format.	http://802.11ninja.net/code/chch2ww.pl
chopchop	First release of chopchop. WEP cracker which uses the AP to decipher packets. Easiest one are ARP's. Takes 10-20s. Included within patches for wlan-ng to inject packets in monitor mode.	http://www.netstumbler.org/showthread.php?t=12489
Etherape	EtherApe is a graphical network monitor for Unix modeled after etherman. Featuring link layer, ip and TCP modes, it displays network activity graphically. Hosts and links change in size with traffic. Color coded protocols display. It supports Ethernet, FDDI, Token Ring, ISDN, PPP and SLIP devices. It can filter traffic to be shown, and can read traffic from a file as well as live from the network.	http://etherape.sourceforge.net/
Ettercap	***Already included in BT***	
expat **kismet dep.*	Expat is an XML parser library written in C. It is a stream-oriented parser in which an application registers handlers for things the parser might find in the XML document (like start tags).	http://expat.sourceforge.net/
gmp **kismet dep.*	GMP is a free library for arbitrary precision arithmetic, operating on signed integers, rational numbers, and floating point numbers. There is no practical limit to the precision except the ones implied by the available memory in the machine GMP runs on. GMP has a rich set of functions, and the functions have a regular interface.	http://gmplib.org/
gpsd **kismet dep.*	gpsd is a daemon that listens to a GPS or Loran receiver and translates the positional data into a simplified format that can be more easily used by other programs, like chart plotters. The package comes with a sample client that plots the location of the currently visible GPS satellites (if available) and a speedometer. It can also use DGPS/ip.	http://freshmeat.net/projects/gpsd
Ike-scan	***Already included in BT***	
Imagemagik **kismet dep.*	ImageMagick® is a software suite to create, edit, and compose bitmap images. It can read, convert and write images in a variety of formats (about 100) including DPX, GIF, JPEG, JPEG-2000, PDF, PhotoCD, PNG, Postscript, SVG, and TIFF	http://www.imagemagick.org/script/index.php
John the Ripper	***Already included in BT***	
kismet	***Already included in BT***	
Kis-snr	`tool to collect signal and nois information from a kismet server for a provided bssid`	http://802.11ninja.net/code/kis-snr.pl
listnets	Tool to execute a WQL (WMI Query Language) query against the WMI namespace to identify the network names for nearby AP and ad-hoc networks	http://802.11ninja.net/code/listnets.vbs
Nessun	Nessus is the world's most popular vulnerability scanner;	http://www.nessus.org

Netdisco	Netdisco is an Open Source web-based network management tool. Netdisco is a network management application targeted at large corporate and university networks. Data is collected into a Postgres database using SNMP and presented with a clean web interface using Mason.	http://www.netdisco.org
NetStrumbler	Netstumbler is the best known Windows tool for finding open wireless access points ("wardriving"). They also distribute a WinCE version for PDAs and such named Ministumbler. The tool is currently free but Windows-only and no source code is provided. It uses a more active approach to finding WAPs than passive sniffers such as Kismet or KisMAC.	http://www.stumbler.net
nstx	***Already included in BT***	
nwepgen	*Tool to generate WEP keys from input string. The Linux Prism2 wireless drivers wlan-ng include an implementation of this tool.*	*http://www.linux-wlan.org*
Ozy2		
Packetyzer	Packetyzer is a network protocol analyzer for Windows, based on the open source Wireshark project	http://www.networkchemistry.com/products/packetyzer.php
PCAPhistogram	A perl script that opens libpcap-formatted packet captare files and counts the frequency of each byte in each packet payload in the file. The output is text-format that can be passed directly to the GNUplot tool to generate the scatter-plot diagrams. Used to analize traffic dispersion in order to assume if is encrypted (narrow scatter) or not (wide scatter).	http://802.11ninja.net/code/pcaphistogram.pl
qbextract	This tool accepts an input pcap file and writes a converted output file, based on a payload offset and desired pcap link type. In it's current form, it will read in a Linux DLT_IEEE802_11 file, and copy the raw 802.11 framing information into a DLT_EN10MB capture file with a different payload offset. This allows us to remove the extra header information from the Proxim Quick Bridge encapsulation type so the data can be analyzed with standard traffic analysis tools.	http://802.11ninja.net/code/qbextract.tgz
tcpdump	***Already included in BT***	
tmscam	*NOT PUBLICLY RELEASED*	
void11	Void11 offers three attack mechanisms: Deauthenticate Clients (default mode); Authentication Flood; Association Flood;	http://www.wirelessdefence.org/Contents/Void11Main.htm
wep_crack	***Already included in BT***	
wepattack	WepAttack is a WLAN open source Linux tool for breaking 802.11 WEP keys. This tool is based on an active dictionary attack that tests millions of words to find the right key. Only one packet is required to start an attack.	http://wepattack.sourceforge.net/
WEPWedgie	WEPWedgie is a toolkit for determining 802.11 WEP keystreams and injecting traffic with known keystreams. The toolkit also includes logic for firewall rule mapping, pingscanning, and portscanning via the injection channel and a cellular modem.	http://sourceforge.net/projects/wepwedgie/
wireshark	***Already included in BT***	
wmi tools	You can use WMI to manage both local and remote computers. WMI provides a consistent approach to carrying out day-to-day management tasks with programming or scripting languages. For example, you can: start a process on a remote computer; schedule a process to run at specific times on specific days;	http://www.microsoft.com/downloads/details.aspx?familyid=6430f853-1120-48db-8cc5-f2abdc3ed314&displaylang=en

	reboot a computer remotely; get a list of applications installed on a local or remote computer; query the Windows event logs on a local or remote computer;	

11 Wardriving with BackTrack and SWAT kit

During the SANS wireless track, a GPS-based WiFi warwalking exercise is done using the provided hardware, called SWAT kit. In order to test the tools and hardware used with the latest version of BT, I did a wardriving exercise. The SWAT kit hardware is best suited for auditing and mapping wireless networks and consists of:

- TripNav TN-200 GPS Receiver with USB cable;
- Senao NL-2511CD PLUS EXT2 WNIC;
- Pigtail cable Netgate N1158-MMCX802;
- Directional Yagi 8.5 dBi gain antenna;

first of all let's lists the tools we need:

kismet, gpsd, gpsmap

Of course the "core" of this exercise is kismet, one of the most common tools to capture 802.11 wireless data. Kismet provides graphical network mapping functionality with *gpsmap*. Gpsmap is able to automatically download maps from several online sources (for European maps now is possible to use the switch "*-S 6*" to query the Expedia servers). We have to go thought these steps to achieve this goal:

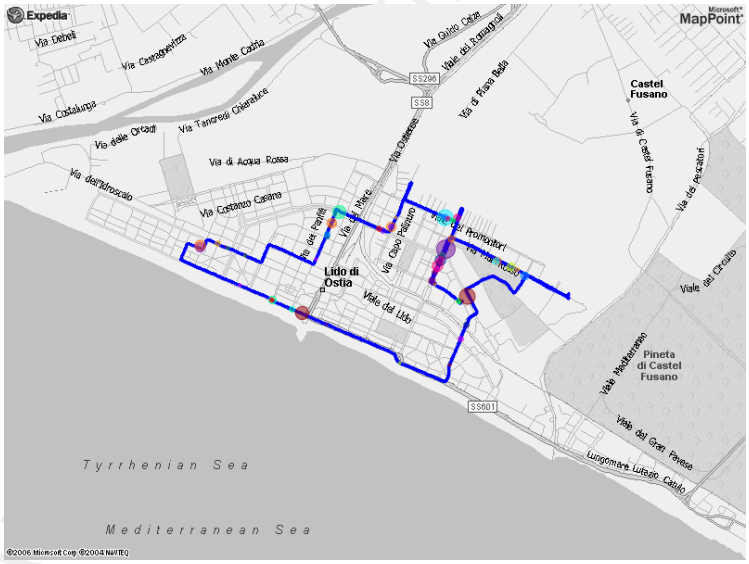
1) install the latest version of *gpsd (gpsd-2.34.tar.gz)* at the moment of this writing) available at *http://gpsd.berlios.de,* because the latest BT include the version 2.33 and doesn't work;
2) run kismet to generate the following files:
   a. *filename*.csv;
   b. *filename*.dump;
   c. *filename*.gps;

d. *filename*.network;

e. *filename*.xml;

3) process the wireless and GPS data collected by Kismet using *gpsmap*; I generated an estimated range map of the cost side of Rome (Italy) based on the information I have got with my laptop while wardriving;

```
BT ~ # gpsmap -t -r -R 50 -D -S 6 -o signal-range.png *.gps
Reading AP manufacturer data and defaults from /usr/local/etc/ap_manuf
Reading client manufacturer data and defaults from /usr/local/etc/client_manuf
NOTICE: Processing gps file 'Rome-Kismet-May-01-2007-1.gps'
NOTICE: Opened associated network xml file 'Rome-Kismet-May-01-2007-1.xml'
NOTICE: Caching GPS file Rome-Kismet-May-01-2007-1.gps
NOTICE: Config dir /root/.gpsmap/ doesn't exist, making it...
Processing 5834 sample points.
Map image scale: 20000
Minimum Corner (lat/lon): 41.724277 x 12.262518
Maximum Corner (lat/lon): 41.740562 x 12.305017
Map center (lat/lon): 41.732420 x 12.283768
Creating blank map image.
Converting map to greyscale.
Calculating network coordinates and statistics...
Processing 158 raw networks.
Assigning network colors...
Plotting 158 networks...
Drawing track coordinates, width: 3...
Calculating and drawing network circles...
BT ~ # display signal-range.png
```



We might need to generate other maps with *gpsmap* using different command line options such as a hull map:

```
# gpsmap -D -S 0 -m map.gif -s 3950 -o hull-map.png -t -u *.gps
```

or a scatter map:

```
# gpsmap -D -S 0 -m map.gif -s 3950 -o scatter-map.png -t -a *.gps
```

Another cool option is to plot maps with the kismet patch *gpsmap-gmap [ref.12]*, in order to use google maps. The output is quite catchy, especially if you have to deliver it as a part of an assessment report for your customer.

12 Advise about WEP security

Wireless networks are considered insecure due to the fact that this technology is based on a shared medium for networking. Basically we have to consider that anyone can get as close as possible to receive the radio signal, hence capturing data. As security is such a delicate process in wireless network, being able to perform good auditing became crucial. Both WEP and WPA security mechanisms provide privacy by using a secret key and the RC4 to encrypt data, WPA provide enhancements compared to WEP that is really weak; but securing a wireless network, especially for enterprise goes beyond using WPA rather WEP. However WEP networks are still around, in order to get the idea of the WEP weakness I strongly suggest reading the following two posts on RaDaJo blog [ref.10]:

4. *What else do you need not to use WEP anymore?* (it contains also the link of the paper: *Breaking 104 bit WEP in less than 60 seconds*);
5. *Breaking 40-bit WEP in less than 30 seconds?*

13 Addendum: Installing BT on Intel MacBook

There are different approaches to installing a dual boot system, in this guide line I describe the way I found more suitable. First of all, I would like to point out that Apple reserves the first primary partition for the "EFI System Partition", this leave the system with only three partitions available; this limitation is also due to the fact that MBR supports 4 primary partitions and GPT does not support extended partitions. In the following figure the *"diskutil"* command is invoked in order to show HD partitions:

```
# diskutil list
mascalzone:~ root# diskutil list
/dev/disk0
   #:                       type name                 size      identifier
   0:  GUID_partition_scheme                         *74.5 GB  disk0
   1:                        EFI                      200.0 MB  disk0s1
   2:                  Apple_HFS Macintosh HD         74.2 GB   disk0s2
```

the following commands are in order to assign 5G to Linux BT:

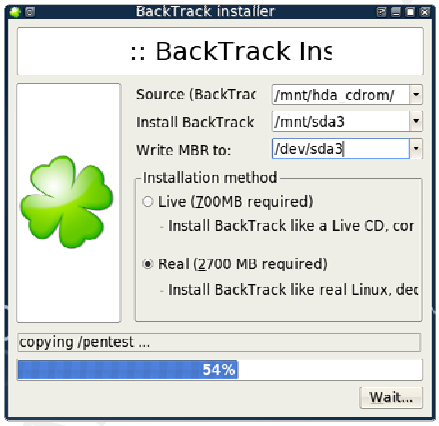
```
# diskutil resizeVolume disk0s2 69G
Started resizing on disk disk0s2 Macintosh HD
Verifying
Resizing Volume
Adjusting Partitions

Finished resizing on disk disk0s2 Macintosh HD
WARNING: You must now reboot!
mascalzone:~ root# reboot
mascalzone:~ root# diskutil list
/dev/disk0
   #:                       type name                 size      identifier
   0:  GUID_partition_scheme                         *74.5 GB  disk0
   1:                        EFI                      200.0 MB  disk0s1
   2:                  Apple_HFS Macintosh HD         69.0 GB   disk0s2
```

Once this is done install the **rEFIt boot loader** [ref.13], now restart the Mac and if everything has been done correctly, the rEFIt menu is displayed with one and, later, two ;) operating systems. Select the MAC OS to boot the system in order to check if, hopefully, it boots correctly. It's time now to edit the partition table using a non-GPT aware partition table

editor, because *fdisk* or any other non-GPT aware editor, will show the "mirrored" partition table. GPT and MBR tables can coexist, but the Linux kernel and *gparted* will take the GPT table into account if present. I recommend booting the system with the useful *gparted* live distro [ref.14] to create the linux partition as the third primary partition and create the *ext3* Linux filesystem. Reboot the system and select the "***Start Partitioning Tool***" in the rEFIt boot menu, it will show the GPT and MBR tables, before continuing we have to synchronize both by typing "yes" to the question it provides. Now we are ready. But, unfortunately, if we insert BackTrack and boot it without a minor change, we will get a problem and whenever we hit a key, BT shows us it echoed; so if we type root, it reports *rroooott*, basically the system is running, but completely unusable, to avoid this before booting we have to type the following cheat codes:

```
# bt noapic irqpoll acpi=force
```

we can now use the *BackTrack Installer* located under "***K Menu->System->***" to install BT:



When completed type the following:

```
BT ~ # mount --bind /dev /mnt/sda3/dev
BT ~ # mount -t proc proc /mnt/sda3/proc
BT ~ # chroot /mnt/sda3 /bin/bash
BT ~ # liloconfig
```

Choose simple, MBR and then exit (don't consider the error it claims! while exiting). It will create the file /etc/lilo.conf, we have now to edit the file and be sure to include the following contents:

```
……….
append="noapic irqpoll acpi=force"
boot=/dev/sda
label = BackTrack
root = /dev/sda3
read-only
……….
```

save the file, exit and type:

```
BT ~ # lilo -v
```

It should say, added BackTrack. It is now possible to reboot the system that should be running a dual boot.

14 References

6. **[ref.1] BackTrack home**
   http://www.remote-exploit.org/backtrack.html
7. **[ref.2] ISSAF** - Information Systems Security Assessment Framework
   http://www.oissg.org/content/view/71/71/
8. **[ref.3] OSSTMM** - Open Source Security Testing Methodology Manual
   http://www.isecom.org/osstmm/
9. **[ref.4] DVL** – Damn Vulnerable Linux
   http://www.damnvulnerablelinux.org/
10. **[ref.5] BackTrack wireless compatibility list**
    http://backtrack.offensive-security.com/index.php?title=HCL:Wireless
11. **[ref.6] PCI and USB IDs databases**
    http://pciids.sourceforge.net/ - http://www.linux-usb.org/usb.ids
12. **[ref.7] Wireless Adapter Chipset Directory**
    http://linux-wless.passys.nl
13. **[ref.8] BT open wiki**
    http://backtrack.offensive-security.com/
14. **[ref.9] Madwifi driver**
    http://snapshots.madwifi.org/madwifi-ng
15. **[ref.10] RaDaJo security blog**
    http://radajo.blogspot.com/2007/01/auditing-wireless-networks-from-windows.html
16. **[ref.11] Video Tutorial (BT inside VMware)**
    http://www.securityresearch.at/WEP_airoscript_rt73.htm
17. **[ref.12] The google gpsmap for kismet**
    http://www.parknation.com/gmap/index.php
18. **[ref.13] The rEFIt Boot Loader**
    http://refit.sourceforge.net
19. **[ref.14] gparted live distro**
    http://gparted.sourceforge.net