Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

Ekhator Stephen Aimuanmwosa

This thesis is presented as part requirement for the award of

Master of Science Degree in Electrical Engineering

Blekinge Institute of Technology
January 2010

© Ekhator Stephen Aimuanmwosa, 2010
“Even the knowledge of my own fallibility cannot keep me from making mistakes. Only when I fall do I get up again”.

- Vincent van Gogh
ABSTRACT

Despite advancement in computer firewalls and intrusion detection systems, wired and wireless networks are experiencing increasing threat to data theft and violations through personal and corporate computers and networks. The ubiquitous WiFi technology which makes it possible for an intruder to scan for data in the air, the use of crypto-analytic software and brute force application to lay bare encrypted messages has not made computers security and networks security safe more so any much easier for network security administrators to handle. In fact the security problems and solution of information systems are becoming more and more complex and complicated as new exploit security tools like Kismet and Netsh (a NetStumbler alternative) are developed. This thesis work tried to look at the passive detection of wireless network capability of kismet and how it function and comparing it with the default windows network shell ability to also detect networks wirelessly and how vulnerable they make secured and non-secured wireless network. Further analysis where made on captured network source packets using wireshark (a network analyzer). The discovery of MAC addresses, IP address, data frames, SSID’s by kismet and netsh and the further exposure of management traffic with wireshark is a source of concern given that such useful network parameters in the hands of an experienced hacker would be a valuable information that could be used in hacking into any network computer.

Introduction to kismet and netstumbler application and their inherent capabilities in network detection is given an in depth look at the beginning of this work. A wide range of definitions and concepts of wireless technology application and uses as it applies to wireless networks, supported devices, security standards and protocols, firewalls and ad-hoc networks, wardriving and its legality, types of authentication, the Linux kernel, special TCP/UDP ports, the drone and third party firmware were all given an in depth look. kismet download and configurations on linux based OS and the netsh utility functionalities was explained for the purpose of clarity. Captured management data packets were opened with wireshark and management data frames found within the packets were analysed. Also, a look at the different file types and results of captured management traffic were displayed. Some of the challenges encountered in the course of this work were discoursed in details and comparison between kismet and netsh was done from the perspective of the vulnerability of a network and the poor channel hopping capability of kismet.
ACKNOWLEDGEMENTS:

My special thanks to my thesis supervisor and examiner: FREDRIK ERLANDSSON (Universitetsadjunkt) for his ideas and suggestions and for taking a great deal of his time to proof read and make corrections in the course of this project. I wish to acknowledge my very good friend Segun ‘Shodix’ Adare for his initial guidance.

To my wife Lola and our children; Omosede and Osaruyi a.k.a Papi I say thank you very much for your patience and support. It would have been impossible without your enduring patience and support. Finally, to my parents I say thank you.
When I first embarked on this thesis work, it never occurred to me that most computer network interface card drivers do not come with RFMON (Radio Frequency Monitor) capability. This singular fact became the most challenging condition to starting the kismet client/server installation and configuration. For kismet to work it would have to run in ‘RF Monitor’ mode. Unfortunately, my HP laptop came with a Broadcom chipset bcm4328 which does not support monitor mode. To solve this problem I made a little research and found two possibilities of putting my machine on monitor mode. The first one was a program ‘bcmmon’ script written by a guy named ‘Kacper Szczesniak’ as a patch to ‘Ndiswrapper’ program to modify the OS kernel so that the MacBook ‘wl’ driver can be enabled into monitor mode. This was a very interesting suggestion, but my understanding of the use and application of UNIX was not sufficient at the time for me to fully embrace this option. The second option was using Linksys WRT54G routers as kismet drone this was a well documented work on the net by ‘Renderlab’.

It is worthy of note that prior to the commencement of this thesis work I was a complete novice to Linux operating systems and the use of command lines as a way of interacting with the computer and its terminals. But in the course of this work I have moved from a novice to an intermediate Linux user. I have come to the harsh reality of war-driving and its underlying inherent consequences with respect to the law of a locality and other legal questions that may arise from ‘how’ to ‘when’ to separate a legal hack as compared to an illegal one. Network security is a whole lot of issues but with a single agenda of keeping networks and network computers safe.

Kismet has always proved to be a great fascination for many as well as myself partly because of its passive sniffing capability and secondly because it has afforded me the opportunity to learn Linux because kismet works best on Linux OS. For me, the default Netsh in windows vista could reveal a great deal about the default security weakness or strength at the kernel level of the OS as well as when computers are networked. Time factor did hamper my understanding and use of logon script.
TABLE OF CONTENTS

ABSTRACT ........................................................................................................................................ iii
ACKNOWLEDGEMENT ................................................................................................................... iv
PREFACE ............................................................................................................................................. v
LIST OF FIGURES AND TABLES ....................................................................................................... vii
CHAPTER 1 ......................................................................................................................................... 1
1.0 Introduction ...................................................................................................................................... 1
CHAPTER 2 ......................................................................................................................................... 3
2.0 What is Wireless Networks? .......................................................................................................... 3
2.1 Types of Wireless Networks.......................................................................................................... 3
2.2 Communication of Wireless Clients Devices................................................................................ 3
2.3 Supported Access Points, Network Cards and Routers................................................................. 3
2.4 Antennas and Cables................................................................................................................. 5
2.5 Uses of Wireless Technology and Applications......................................................................... 6
2.6 Location and Access Points......................................................................................................... 7
2.7 Wardriving..................................................................................................................................... 7
2.8 Firewalls and Virtual Private Networks (VPN)........................................................................... 8
2.9 Wireless Security Standards and Protocols............................................................................... 9
2.10 Service Set Identifier (SSID).................................................................................................... 10
2.11 Beacon Frames ......................................................................................................................... 10
2.12 Wireless Security Protocols..................................................................................................... 11
2.13 Types of Authentication........................................................................................................... 13
2.14 The Linux 2.6 Kernel .............................................................................................................. 13
2.14.1 Ports Scan ............................................................................................................................. 14
2.15 Kismet Drone............................................................................................................................ 14
2.16 Third Party Firmware .............................................................................................................. 15
2.16.1 DD-WRT Firmware .............................................................................................................. 15
2.16.2 OPENWRT “Whiterussian RC6” Firmware ........................................................................ 15
2.17 Kismet........................................................................................................................................ 15
2.18 Kismet Client and Kismet Server ............................................................................................ 15
2.19 Netstumbler and Netstumbler Alternatives............................................................................ 16
2.20 NETSH ....................................................................................................................................... 17
CHAPTER 3 ......................................................................................................................................... 18
3.0 Linux Distribution ...................................................................................................................... 18
3.1 Preparing and Upgrading the Linux Kernel ................................................................................ 18
3.2 Software and Hardware Packages........................................................................................... 18
3.3 Linksys WRT54GL .................................................................................................................... 19
3.4 Access Points Scan Result after Flashing................................................................................ 20
3.5 Configuring Client Sever Desktop............................................................................................ 22
3.6 Setting up WRT54GL on my Network...................................................................................... 23
3.7 Kismet Download and Installation............................................................................................ 24
3.8 Platforms..................................................................................................................................... 29
3.9 NETSH (NetworkShell) Utility................................................................................................. 29
CHAPTER 4 ......................................................................................................................................... 33
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

Appendices

Chapter 5

Acronyms

References
LIST OF FIGURES AND TABLES

TABLES:
TABLE 2.3.2 Hardware (WRT54GL)
TABLE 2.8.1 Firewall
TABLE 2.9.1 IEEE 802.11 Wireless Protocol
TABLE 2.12.3 WEP (Wired Equivalent Privacy)

FIGURES:
FIGURE 2.4.1 (a) 2dBi wireless 802.11bgn WLAN 5.8GHz antenna
FIGURE 2.4.1 (b) 150Mbps 802.11n WLAN 7dBi antenna with USB adapter
FIGURE 2.9.1 37dBm IEEE802.11bg WLAN 2.4GHz antenna
FIGURE 3.3.1.2 WRT54GL System Settings
FIGURE 3.3.1.3 WRT54GL Router Info
FIGURE 3.3.1.4 WRT54GL Connection Status
FIGURE 3.3.2.4 WRT54GL Wireless Status
FIGURE 3.3.2.5 DHCP Leases Status
FIGURE 3.5.1 (a) Network Connections
FIGURE 3.5.1 (b) Creating New Wireless Network
FIGURE 3.5.1 (c) Editing the Created Network
FIGURE 3.5.1 (d) Security
FIGURE 3.6 Schematic Diagram of My Network
FIGURE 3.91 (a) Network Access Points
FIGURE 3.91 (b) Captured Networks Using NETSH
FIGURE 4.2 Kismet Dumps
FIGURE 4.2.2 (a) Single Service Discovery Protocol (SSDP)
FIGURE 4.2.2 (b) Cracked Packet
FIGURE 4.3 TCP Retransmission Frames
FIGURE 4.3.1 Beacon Frames
FIGURE 4.3.2 Data Frames
FIGURE 4.3.3 Probe Request
FIGURE 4.5 Examples of .XML, .Network, .CSV, .Cisco Fles
FIGURE 4.6 Packet Rate
FIGURE 4.7 Channel Hopping In Kismet
FIGURE 4.11 (a) Kismet and IP Address Capture
FIGURE 4.12.1 MAC Addresses
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

This page was intentionally left blank
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

CHAPTER 1
INTRODUCTION

Despite advancement in computer firewalls and intrusion detection systems, wired and wireless networks are experiencing increasing threat to data theft and violations through personal and corporate computers and networks. The ubiquitous WiFi technology which makes it possible for an intruder to scan for data in the air, and the use of cryptanalysis versus brute force to lay bare encrypted messages has not made computer and network security any much easier for network security administrators. In fact the security problems and solutions of information systems are becoming more and more complicated as new exploit security tools like Kismet, aironet, wireshark and Netsh (a NetStumbler alternative) are developed.

This master thesis intends to investigate by way of comparison two such versatile security tools which not only has the propensity to compromise a network through data discovery, but could also serve as a network security solution.

Kismet and NetStumbler are two war-driving tools used very often to gain wireless access through access points into wireless networks and invariably on to computers. It is a known fact that Kismet and NetStumbler are two network security tools which work best on two different platforms, Kismet on Linux and NetStumbler on Windows operating systems respectively.

I am proposing to install, configure and launch Kismet client and Kismet server on my desktop while at the same time using my WRT54GL access point as a Kismet Drone. The reason for the drone is because my HP laptop dv6653eo Network Interface Card comes with a Broadcom chipset (BCM4321/ BCM4328 ) which does not support RFMON (Radio Frequency Monitor Mode) a veritable ingredient for wireless packet sniffing. On the other hand I am comparing the easy deployment and detection of access point configuration capability of a NetStumbler alternative called NETSH with Kismet ability to detect, uncloak hidden SSID’s of access points and capture data packets wirelessly from management traffics of various access points.

Rogue access points by unauthorized users can be a nightmare; access to wired networks can be secured by securing the cable connection of the switch or hub, but in wireless networks, wireless data propagated freely through airwaves can be intercepted by any intruder wishing to compromise a computer network system. It therefore implies that an intruder having sufficient signal level could either, listen or view management data traffics between users and the wireless network access points or connect to the access points in cases of poor security defence mechanism not being in place and actually gaining access into the network. So, it becomes imperative to secure our wireless access points and infrasctructures. This in essence brings to mind access authorization process or authentication, data encryption and confidentiality. But this is not the goal of the thesis work, neither am I going to look at some of the features and possible functionalities of Kismet. For example, Kismet can be setup as an Intrusion Detection System (IDS), It can support Global Positioning System (GPS) for mobile tracking of physical networks and access points locations, and can even be configured to give a text-to-speech alert during war-driving sessions. All these would not be delve into in this work.

Kismet application is an open source wireless network analyzer running on Linux, UNIX and Mac OS X. It is not supported by windows OS. Kismet is a passive sniffer used to detect any wireless 802.11a/b/g protocol complaint networks, even when the network has a non broadcasting hidden SSID (Secure Service set Identifier). Kismet can discover, log the IP range of any detected wireless network and report its signal and noise levels. It can sniff all management data packets from detected networks. Kismet can be used to locate, troubleshoot and optimize signal strength for access points and clients, as well as detect network intrusions.

NetStumbler on the other hand is an active sniffer and it is not exactly an open source wireless network application. It runs on Windows, especially on Windows XP and earlier versions. NetStumbler can be used to discover, configure, secure and optimize a network.
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

It is very useful for detecting and surveying of wireless networks and good for pinpointing details in wireless networks. It supports GPS also for mobile tracking of networks and has a small library which can be accessed through active scripting, VBScript (Visual Basic scripting), Jscript (Java scripting), active state Perlscript, python, to achieve text-to-speech alert detection mode during situations like war-driving.

The objectives of this work would be to use:

- Kismet to sniff data management traffics packets from wireless LAN, analyze them to see the vulnerability of the different detected access points and compare this result with captured sources using NETSH
- Troubleshoot wireless connections by way of analyzing signal strength to noise ratio of captured sources.
- Launch Kismet GUI (Kismet_ui.conf) application for real-time visualization and monitoring.
- Open TCP dumps with Wireshark software to examine and analyze management data packets from captured sources in http files. This is to help me see the possibility of capturing sensitive password or valuable information and a test of the vulnerability of the network.
- Figure out the meaning, uses and differences between the different file dumps, for example; the .dump, .csv, .network, .weak and .cisco.
BACKGROUND AND RELATED WORKS

2.0 WHAT IS WIRELESS NETWORK?

Wireless network refers to any type of computer network that is wireless are interconnected between nodes to communicate wirelessly that is, without the use of wires or cable connections. Wireless telecommunications networks are basically information transmission systems which implements remotely radio waves and data at the physical or data link layer of the network.

2.1 TYPES OF WIRELESS NETWORKS

Wired and Wireless Networks:

Ethernet is basically a network of wired computers, it is a frame based technologies for Local Area Networks (LANs), defined by different wiring and signalling standards (IEEE 802.3 Standard) for the physical layer (first layer of the OSI) of the Open System Interconnection Reference Model. There are different types of wireless networks, we have the Wireless Personal Area Networks (WPAN), Wireless Metropolitan Area Networks (WMAN), Mobile device networks like GSM and Personal Communication Service (PCS) and Wireless Local Area Networks (WLAN). My Linksys WRT54GL wireless Local Area Network (LAN) setup, connected to the internet is part of a larger Wide Area Network (WAN) with so many other neighbouring wired and wireless access points like in homes, schools and offices within the locality. Most WLAN are IEEE802.11 standard based technologies and Wi-Fi (Wireless-Fidelity) a technical certification of the interoperability between IEEE802.11 devices is often just referred to as IEEE802.11.

2.2 COMMUNICATION OF WIRELESS CLIENT DEVICES:

INFRASTRUCTURE AND AD-HOC

2.2.1 INFRASTRUCTURE

The infrastructure mode is the most used mode between wireless client devices. The infrastructure mode is used when a wireless client computer wish to connect to another host computer through an access point like the WRT54GL.

2.2.2 AD-HOC

The Ad-Hoc mode is used when two host computers wants to communicate directly with each other without connecting through an access point.

2.3 SUPPORTED ACCESS POINTS, NETWORK CARDS AND ROUTERS.

2.3.1 LINKSYS WRT54G’s

There are several models of the WRT54G’s and versions (WRT54G/WRT54GL/WRT54GS) http://oldwiki.openwrt.org/Hardware(2f)Linksys.html, with some supporting different third party firmware like
OpenWRT “White Russian”, Kamikaze, dd-WRT and WiP. But, the Linksys WRT54GL v1.1 used in this project is essentially WRT54G v4.0. It comes with a 4MB of flash memory and 16MB of RAM and supports OpenWRT, DD-WRT & Kamikaze third party firmware as well as supports RF monitor mode, hence the choice. The “L” in the WRT54GL stands for “Linux” the Linux environment makes it suitable for running Kismet, just as the large flash and RAM makes it possible to use it as a Kismet drone. Unlike the WRT54G v5.1 which I attempted to use initially for this project, had two major drawbacks, the first was its small flash memory of 2MB and 8MB of RAM, left little memory space for a functional GNU/LINUX distribution after third party firmware flash, using the WRT54G v5.1 as Kismet drone was not possible since I needed the compressed kismet .conf files inside the WRT. The UNIX library inside the flash for version 5.1 was limited and scanty for example, it did not support Secure Shell (SSH) application needed to securely move files around. WRT54GL v1.1 starts with serial number CL7B and CL7C and v1.0 starts with CL7A. It is readily available here in Europe (sometimes called the European version).

2.3.2 HARDWARE:

Info:

Architecture: MIPS
Vendor: Broadcom
Boot Loader: CFE
System-On-Chip: Broadcom 5352EKPB
CPU Speed: 200 MHz
Flash-Chip: EON EN29LV302B-70TCP
Flash size: 4 MB
RAM: 16 MB
Wireless: Broadcom BCM43xx 802.11b/g (integrated)
Ethernet: Switch in CPU
USB: No
Serial: Yes
JTAG: Yes

Table 2.3.2

It has good networking capability which includes a virtual LAN (VLAN) configuration, an Ethernet switch, and bridge capabilities. It has a four-port Ethernet switch sockets (I/O) bridged by default to the wireless interface. The wireless interface can serve wireless clients when used as an access point (Master mode), It can serve as a client while connecting to other wireless networks (Managed mode). It can also provide direct connectivity to other wireless clients like peer-to-peer network (Ad Hoc mode).
2.3.3 OTHER SUPPORTED DEVICES AND CARDS.

Many computers come with network cards that do not support monitor mode. However, there are many devices, drivers and cards like some routers and PCI cards that can be mounted or downloaded on the computer to be able to enable monitoring mode capability. Here are some products with Broadcom drivers that support monitor mode:

2.4 ANTENNAS AND CABLES

2.4.1 ANTENNAS

The Linksys WRT54GL router comes by default at shipment with two horned (2.2dB), 0.145m (antenna height) omni-directional radio frequency (RF) antenna. It has an initial wireless radio card detected transmission power of 89mW. Coupled with the 2.2dB RF antenna gain gives a total transmit power of about 121mW. The EIRP is not much of transmit power. So, for a higher effective power, antennas with higher gains of 7dB and 9dB can be purchased in the open market. It is worthy of note that gain G, is a function of the height of an antenna, the taller the antenna, the higher the gain in dB (decibel), this is the reason why so many RF antennas and mast are so high, it is for better reception and transmit purposes. The WRT54GL antenna is also well suited for use in the 2.4GHz (802.11b/g) and/or 5.8GHz (802.11a) frequency band and therefore satisfies the design purpose. Omni-directional antennas radiate and receive RF signals in all direction, a circular wave of 360° pattern. The antennas are mounted on two-female RP-TNC connector. Figure 2.4.1(a) is a good example of 2dBi wireless 802.11bgn WLAN 5.8GHz antenna and figure 2.4.1 (b) is a 150Mbps 802.11n WLAN 7dBi antenna with USB adapter. Both antennas support windows OS, Mac and Linux OS. They have a maximum transmit range of between 100m-300metres and a power gain of between 100mW-to-500mW. See Appendix B2 for technical specifications.

The total Equivalent Isotropic Radiated Power:

\[
\text{EIRP} = \text{The total output power of the wireless card + Antenna Gain (2.2dB)}
\]

\[
dB = 10\log_{10}(P)
\]

For a 2.2dB antenna gain and a default 89mW wireless card power;

\[
2.2B = 10\log_{10}(P)
\]

\[
P = \text{Antilog} \frac{2.2}{10} = 10^{2.2/10}
\]

\[
= 1.659 \text{ Watts}
\]

Converting to dBm;

\[
\text{dBm} = 10\log_{10}(P_{\text{watts}}/1mW)
\]

\[
= 10\log_{10}(1.659/10^3) = 32.198\text{dBm}
\]

But,

\[
1mW = 1dBm
\]

Therefore,

Antenna Gain in mW = 32mW

\[
\text{EIRP} = 89mW + 32mW
\]

\[
= 121mW
\]
2.4.2 CABLES

The Linksys WRT54GL comes with a straight-through RJ45 WAN/LAN cable for connection to the internet. It can use all other types of RJ45 cable depending on what kind of connection you want. A female RP-TNC to N-type connector can be used for antenna extension purposes.

2.5 USES OF WIRELESS TECHNOLOGY AND APPLICATIONS

Vertical markets like manufacturing, banking and aerospace are all realizing the benefits of incorporating and utilizing wireless networks in their day-to-day business and work processes. Many of these markets are horizontal in nature because they can all use monitoring, delivery services, retail, finance and public safety applications.

2.5.1 MONITORING

Networks can be monitored actively or passively. This involves sending beacon signals to remote receivers and getting replies or listening quietly to network nodes without associating with them. Wireless monitoring has applications in many other sectors, for example, meteorologist use weather satellites to monitor weather patterns. Eavesdropping; wired and wireless networks can be tapped and information can be gathered or listened into.

2.5.2 DELIVERY SERVICES

A wireless technology called (ESMR) Enhanced Specialized Mobile Radio is employed today in most delivery and courier services, in the bid for efficiency and speed. The technology makes it possible for a dispatcher in the office communicate with the delivery driver in a vehicle in a remote area to communicate on a single channel along with every other user, this makes it possible for the dispatcher to be able to coordinate schedules or re-scheduling of package pick-up’s and deliveries as well as track the drivers progress.

2.5.3 RETAIL

Because of wireless Point-of-Sale (POS) applications, registers, scanners and printers can now be used in fixed and remote locations. This application is found in modern day shopping malls and retail businesses. This has proven to be of immense benefits to merchants and customers, and has also changed the way retail business transactions are conducted.
2.5.4 FINANCE

Wireless communications through the internet, telephony and voice-over-IP has revolutionized e-banking and e-commerce. Electronic transaction has been made possible by wireless network technologies and infrastructures. Wireless communication has also changed the way how investors do business, say for example at the stock exchange. Investors can now get online real-time quotes sent through the internet to their wireless devices.

2.5.5 PUBLIC SAFETY

Orbital communication satellites like the International Maritime Satellite Organisation (INMARSAT), has made it possible for natural disasters and weather conditions to be monitored and reported around the globe. Global Positioning Systems (GPS), once used solely for military applications, hospitals, aviation, can now be used by motorist for everyday commuting around the cities of the world.

2.6 LOCATION AND ACCESS POINTS

A good knowledge in radio frequency propagation is required to determine the optimum location for access points. It is important to carry out radio frequency site survey in order to know the location and numbers of access points present in a coverage area, for better coverage and transmission performance. For better coverage, access points should be placed in applicable launch space that could make it possible for another access point to cover an adjacent portion of the estimated coverage area. The need to acquire taller antennas or perhaps move the access point around the room in cases of home use, can greatly enhance transmit power of the access point due to an increased antenna gain. Antennas should be mounted as high as possible as this will increase the horizontal range of the radio frequency signal. For data rates of 11Mbit/s access points can be located just within a 100ft to each other, especially for office environment of facility. Access points can be placed further apart like 500ft away from each other if it is just for connectivity purpose, because propagation overlap can be achieved at less data rate provide the antenna has maximum propagation line-of-sight and free of thick reflective obstacles.

2.6.1 ACCESS POINTS VERSUS ROGUE ACCESS POINTS: LINKSYS WRT54GL ROUTER

A rogue access point is a wireless router or access point installed in a secured corporate network without authorisation from the local network administrator. A cracker can compromise a secured network by creating an access point in a network in order to exploit the network.

2.7 WARDRIVING

2.7.1 WHAT IS WARDRIVING?

Wardriving is the act of moving around a specific area, mapping the population of wireless access points for statistical purposes. Wardriving does not utilize the resources of any wireless access point or network that is discovered, without prior authorization of the owner. Wardriving is not exclusive of surveillance by automobile alone. Wardriving is accomplished by anyone moving around a certain area looking for data, which include walking (warwalking), flying (warflying), bicycling (warcycling). Setting up wireless infrastructure for the sole purpose of gathering information that is being transmitted across a wireless network, is not the same thing as wardriving, this is sniffing.
2.7.2 WARDRIVING WITH LINUX

Linux is the most robust operating system for wardriving. Linux makes it possible for wireless cards that support RFMON to be put on monitor mode, which makes passive scanning possible. Configuring Linux to wardrive used to be very difficult process that involved both kernel configuration and network card driver patching. This is no longer so, as of the 2.6.16 kernel revision, it is possible to build a Linux kernel with all of the support you need compiled into it.

2.7.3 WIFI LEGAL ISSUES / THE LEGALITY OF WARDRIVING

According to the Federal Bureau of Investigation (FBI), it is not illegal to scan access points; however, once a theft of service, a denial of service (DOS), or a theft of information occurs, it becomes a federal violation through 18USC 1030 (www.usdoj.gov/criminal/cybercrime/1030_new.html). [p.17] Kismet Hacking-Brad ‘Renderman’ Haines.

November 7, 2006 - The UK passed a new law today against cyber crime. The law targets DoS (Denial of Service) attackers with punishments up to 10 years in prison. The law clarifies Britain’s Computer Misuse Act, because the old legislation did not address DoS attacks specifically. The original act only mentioned penalties for modifying content on a computer without authorization. Because of the ambiguity in the old law, teenager David Lennon was cleared of all charges after being accused of sending his former boss 5 million emails.

From June 1, 2007, Sweden bans all website attacks, like DoS attacks. Sweden calls it a crime to program computers to automatically click on the same page thousands of times. This comes in response to the attacks on the Swedish national police website and other government websites. Attackers can be found guilty and receive up to 2 years in prison. The new law declares both automatic and manual DoS attacks illegal. Prosecutors will have to show the court that the attack was of criminal intent and that it was intended to damage a computer system. Simply trying to launch an attack is also to be considered criminal act.


2.8 FIREWALLS AND VIRTUAL PRIVATE NETWORKS

2.8.1 FIREWALL

A firewall is both infrastructure and application software that stand against penetration attacks into a network. The goal is to control access to a protected network with two main philosophies in mind during configuration:

- Allow everything except designated packets
- Block everything except designated packets

They block everything except designated packets is the application of choice in today’s packets filtering policy. When packet filtering is enabled, all incoming packets are blocked except for designated packets and clients which are allowed through the firewall or filtering software stipulated at the point of configuration. There are four types of firewalls:

- Packet Filters
- Circuit-Level Gateways
- Application-Level Gateways
- Stateful Multi-Level Firewalls
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

Each type of firewall can be matched with where it functions in the OSI layer model:

<table>
<thead>
<tr>
<th>OSI Model Layer</th>
<th>Internet Protocols</th>
<th>Firewalls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>HTTP, FTP, DNS, NFS Ping, SMTP, Telnet</td>
<td>Application-Level Gateway</td>
</tr>
<tr>
<td>Presentation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Session</td>
<td>TCP</td>
<td>Circuit-Level Gateway</td>
</tr>
<tr>
<td>Transport</td>
<td>TCP</td>
<td></td>
</tr>
<tr>
<td>Network</td>
<td>IP</td>
<td>Packet-Filtering, Stateful Multi-Level Filtering</td>
</tr>
<tr>
<td>Data Link</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physical</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.8.1 Firewall

Default firewall rules inside the WRT54GL do not accept connections on the WAN ports this can be adjusted when using the router as a bridge. Running OpenWrt and editing the /etc/firewall user file would allow management over ports 22 TCP, SSH, and port 80 TCP (HTTP).

2.8.2 Virtual Private Networks (VPN)

Virtual Private Networks are a cost-effective way to extend LAN over the internet to remote networks and remote client computers. VPNs use the internet to route LAN traffic from one private network to another by encapsulating the LAN traffic in IP packets. The encrypted packets are unreadable by intermediary internet computers and can contain any kind of LAN communications, including files and print access, LAN e-mail, Remote Procedure Calls, and clients/server database access.

2.9 WIRELESS SECURITY STANDARDS AND PROTOCOLS

2.9.1 IEEE802.11

The IEEE802.11 standard is a set of standards for Wireless Local Area Networks (WLAN) which defines wireless data. It consists of three types of protocols: 802.11a, 802.11b, 802.11g and 802.11n. The 802.11 protocols operate at the data link layer (layer 2) of the OSI (Open System Interconnection Reference Model) model. The IEEE802.11N-2009 which recently came into effect is an amendment of IEEE802.11-2007 wireless networking standard to improve the maximum data transfer rate of 54Mbit/s to 600Mbit/s of earlier standards like the 802.11b/g using four spatial streams at a channel bandwidth of 40 MHz. Each protocol has a maximum transmit data rate within the permitted transmit frequency range for WLAN’s. The permitted transmit frequency range for IEEE802.11 WLAN is between 2.4GHz -to- 5.8GHz. The choice of frequency range is vital to wireless communications in general, so that there are no channel frequency conflicts with other wireless devices and technologies, like the GSM and radio. Channel interference which can lead to poor data transmission rate can also be avoided in all 13 channels of most Wi-Fi devices. A restriction is also put on maximum transmission power of 802.11 hardware devices. Figure 2.9.1 is a wireless card, Yagi 17dBi antenna with USB plug. It is IEEE802.11bg WLAN 2.4GHz antenna type and compatible with windows, Mac OS 10.3x /10.4x and Linux. It has a high gain and to total transmit power of 37dBm (>700mW). The larger the frequency the smaller the antenna size due to narrow bandwidth (smaller wavelengths). See Appendix B1 for specifications.
Protocol | Release date | Max rate [Mbit/s] | Frequency [GHz]
--- | --- | --- | ---
802.11a | 2001 | 54 | 5.8
802.11b | 1999 | 11 | 2.4
802.11g | 2003 | 54 | 2.4
802.11n | 2009 | 600 | 2.4

Table 2.9.1 IEEE 802.11 Protocol

2.10 SERVICE SET IDENTIFIER (SSID)

2.10.1 BROADCAST AND NON-BROADCAST:

Service Set Identifier is a configuration that makes it possible for a wireless network to be identified. Access to a network is only possible when the client SSID for a WLAN card is matched with the SSID of an access point that is to be connected with. An SSID can have up to 32 characters and it is case sensitive. SSID can be broadcast or non-broadcast: Neighbouring client computers can see what the name of an SSID is when broadcast is enabled (BSS) and non-broadcast when SSID is hidden as in ‘probe networks’ or ‘no ssid’. A cloaked SSID can be detected by software like Kismet which has RFMON capability and can look into wireless data frames to extract the SSID.

Extended Service Set Identifier (ESS) is when multiple access points connected to the same wired network are setup to have the same SSID. Sharing the same SSID with other access points connected to the same wired network whether owned by a different entity or by the same entity can become an issue in that end user device can connect to the wrong SSID.

2.11 BEACON FRAMES:

A beacon frame is essentially a management frame in the IEEE802.11 based WLAN which contains all information about a network. A broadcast access point set in infrastructure mode can transmit beacon frames
periodically to announce the presence of a WLAN network to other neighbouring access points. Beacon frames consist of MAC headers, Frame Check Sequence (FCS) and Frame body, Timestamps, Beacon Intervals and Capability Information. Beacons are used by wireless clients to identify nearby access points.

2.12 WIRELESS SECURITY PROTOCOLS

2.12.1 WEP / WPA, CCMP, TKIP, AES, WPA2

Wired Equivalent Privacy (WEP) and Wi-Fi Protected Access (WPA)/WPA2 are wireless security protocols. Security policy settings on an access point are essential, considering the fact that intruders who can pick up beacon frames from broadcast/non-broadcast SSIDs, can attempt to penetrate a network for malicious purposes.

2.12.2 IEEE802.1x

802.1X

802.1X is a port-based network access control mechanism. Either TKIP or AES-CCMP can be used for 802.1X authentication.

TKIP or AES can be used by WPA, but WPA2 uses only AES.

Below you can find definitions about key words used to define a secure link

Confidentiality or privacy: The data cannot be seen in a readable format.

Typical symmetric cipher algorithm: DES, 3DES, AES, Blowfish

Integrity: The data cannot be altered.

Typical hash-functions algorithms: sha1, md5

Authentication: The VPN gateways are sure about the identity of the other.

Typical algorithm: RSA, DH(Diffie-Hellman)

RSA & Diffie-Hellman: These are two public key encryption algorithms commonly used in commercial network security. Public key encryption (Asymmetric encryption) is a few 1000\textsuperscript{th} times slower than private key (single Encryption) exchanges due to multiple keys creation and exchange between each type of encryption keys. This is in view of performance. Some of the known features are:

- They are strong enough for commercial use; because they exceed the recommended minimum of 128 bits length keys with their 1024 bits length keys.
- Developed in the ‘70s; not known to have been cracked yet. We can compare this to the 128 bits key used by WEP, known to have been cracked in recent times.
- Both are susceptible to Man-In-The-Middle (MITM) attacks, due to a lack of key authentication between exchange parties.
- The authenticated Diffie-Hellman key agreement protocol (station-to-station=STS Protocol) was later developed in the ‘90s to defeat the man in the middle attack weaknesses. The two parties can now authenticate themselves through the use of digital signatures and the exchange of public key certificates.

2.12.3 WEP (Wired Equivalent Privacy)

WEP uses the RC4 cipher algorithm for confidentiality and the CRC-32 checksum for integrity. It provides no authentication mechanism.
It uses a 40 or 104 bits key which is associated to a 24 bits Initialization Vector (IV) to provide randomness. The WEP shared key and IV concatenation is referred to as key schedule or "seed" and is 64 (40 + 24) or 128 (104 + 24) bits length.

WEP is vulnerable because of relatively short IVs and keys that remain static. If a hacker collects enough frames based on the same IV, called "weak IV", the hacker can in fact determine the shared secret key.

The Airodump-ng / Aireplay-ng / Aircrack-ng wireless cracking software package can be used to easily crack a WEP key in a couple of minutes even when no data is exchanged between an access point and a client.

<table>
<thead>
<tr>
<th>Airo dump-ng kismet.</th>
<th>Captures wireless data.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aireplay-ng.</td>
<td>Forces the AP to generate traffic</td>
</tr>
<tr>
<td>Aircrack-ng ptw</td>
<td>Cracks the WEP keys from captured data</td>
</tr>
</tbody>
</table>

Table 2.12.3 WEP

When no data is exchanged between a client and the AP, Aireplay-ng forces the AP to generate traffic which will be captured and used by Aircrack-ng to crack the key. Only a few wireless cards. Aircrack-ptw released in April 2007 and included in Aircrack-ng dramatically reduced the amount of data capture possible before the WEP key can be cracked.

2.12.4 WPA/WPA2 (Wi-Fi Protected Access)

WPA improved by TKIP was created before the 802.11i security standard (WPA2) to provide an immediate solution following dramatic security issues with WEP.

The new security standard, WPA 802.11i or WPA2 was then ratified in June 2004 and fixes all WEP weaknesses.

WPA2 is divided into three main categories:

(a) **TKIP**: Temporary Key Integrity Protocol

TKIP is a short-term solution that fixes all WEP weaknesses. It provides a rekeying mechanism and per packet key mixing. Contrary to what is generally indicated by network administrators or even manufacturers, It does not provide confidentiality, but it does provide integrity as it is not a cipher algorithm. The RC4 cipher algorithm is used with TKIP.

TKIP provides the advantage of not to being forced to update the Wireless hardware compared to the one used for WEP.

TKIP is used with WPA.

(b) **AES-CCMP**: Advanced Encryption Standard - Counter Mode CBC-MAC Protocol

CCMP uses AES as its cryptographic algorithm. (AES is the successor of DES) CCMP provides integrity and confidentiality.

AES-CCMP requires more computing power than TKIP in migrating from WEP to WPA2 new hardware. Since around 2005/2006, all the good Wireless AP or clients supports WPA2.

AES-CCMP is used with WPA or WPA2 and is the only choice for WPA2.
2.13 TYPES OF AUTHENTICATION

WPA-enterprise / WPA-personal:

WPA and WPA2 have two types of authentications: WPA-enterprise and WPA-personal.

2.13.1 WPA-ENTERPRISE:

**Enterprise** may be unavailable on some home wireless device models. IEEE 802.1x authentication is based on EAP protocols such as EAP-TLS/TTLS or PEAP. Extensible Authentication Protocol (EAP) [RFC 3748] is just the transport protocol optimized for authentication, not the authentication method itself. Most of the time, the 802.1x authentication is not made by the access point but by another device called RADIUS. The access point acts as a relay for the authentication messages between the client and the RADIUS. The enterprise authentication method is used in professional environments only because it requires complex configurations and expensive hardware.

2.13.2 WPA-PERSONAL:

**PERSONAL**

The access point and the client share a similar key called passphrase or shared key. The personal authentication method is used in home environments because it is inexpensive and easy to setup. Simple and too short a password makes the AP vulnerable against dictionary attack and against brute-force attack. When selecting the security policy; as a general security rule, choose always the highest security protocol available supported by both the client and access point. The choice of WPA2 encryption method and a complicated password can help prevent the wireless network from being hacked and any encryption method with a weak passphrase can easily be broken or cracked.

Below is a list of the different security solutions sorted from the most secured encryption method:

(a) WPA2
(b) WPA - AES
(c) WPA- TKIP

2.14 THE LINUX 2.6 KERNEL

The kernel is the core of an Operating System (OS), the kernel is not the operating system itself, but the main component of a complete operating system. It contains much of the root functions like the virtual memory, shared libraries, multitasking, TCP/IP networking, demand loading, multi-stack networking including ipv4 and ipv6 as well as shared copy-on-write executables. The Linux kernel is licensed under the GNU General Public License free software license. The latest stable Linux kernel is 2.6.32.2. Frequent modification and patches are written for the kernel, sometimes as frequent as every week!
2.14.1 PORTS SCAN:

These are some of the ports and services used in the course of the work for data exchanges and for interface connections.

3501 & PORT 22 (SSH), PORT 23 (TELNET), PORT 80 (HTTP), port 2501 (TCP/UDP), PORT 443 (HTTPS)

2.14.1.1 PORT 22 SECURE SHELL (SSH)

The standard TCP port 22 has been assigned for contacting SSH servers by default. UDP can also use port 22 for some ‘pcAnywhere’ services. An SSH client program is typically used to establish connections to an SSH daemon. Secure Shell is a network protocol that allows data to be exchanged using a secure channel between two networked devices.

2.14.1.2 PORT 23 (TELNET)

Services that use port 23 are called internet standard protocol for remote login. Telnet runs on top of TCP/IP and act as a terminal emulator for remote login sessions. Depending on preconfigured security settings, this daemon can and does typically allow for some way of controlling accessibility to an operating system. Uploading specific hacking script entries to certain Telnet variants can cause buffer overflows, and in some cases, allow administrative or root access. The Telnet daemon can open the door to serious system compromise: passwords are passed in clear text, and successful connections enable remote command execution. Unless your standard communication policies require Telnet, it is advisable to disable it. There are ways of securing Telnet if it is necessary.

2.14.1.3 PORT 80 (HTTP) AND PORT 443 (HTTPS)

Hyper Text Transfer Protocol defines the communication format between two web browsers and a website. HTTPS is a secured webpage that sends encrypted login information request to another server for authentication. It is more secured than the HTTP.

2.14.1.4 PORT 2501 (TCP/UDP)

Transmission Control Protocol and User Datagram Protocol uses port 2501 as communication protocol at the internet session layer, transport layer and network layer of the OSI (Open System Interconnection) Reference Model. Datagrams are communicated through port 2501 to an application running in another computer for example. Both TCP& UDP are internet protocol based. UDP is connectionless and does not guaranty reliable communication at best effort. TCP/IP is more reliable.

2.14.1.5 PORT 3501 (TCP/UDP)

Port 3501 is a TCP/UDP iSoft-P2P local client port used by applications for incoming and outgoing data traffics on the client machine. It is sometimes called IP local port range or local client port windows.

2.15 KISMET DRONE

Access point like the Linksys WRT54GL can be used as Kismet drone. A Kismet drone is a flashed access point like the WRT54GL with a third party firmware (dd-wrt, OpenWrt “white russian”, and kamikaze, tomato...) to mention but a few. The pre-compiled kismet binaries were loaded in the WRT54GL which supports the MIPS (Microprocessor without Interlocked Pipeline Stages) platform. The MIPS binary makes it possible to run kismet on small devices like the WRT54GL router. A drone has two network interfaces (NICs), a wireless NIC for
monitoring and the other for sending captured management packets to the server. The purpose of the drone is to perform wireless discovery just like installing a kismet server would do. However, a kismet drone does not log captured management packets on to the WRT54GL, instead packets dumps are usually dumped at server located on the client desktop.

2.16 THIRD PARTY Firmware

"After Linksys was required to release the WRT54G's firmware source code under terms of the GNU General Public License there have been many third party projects enhancing that code”

Third party firmware refers to modified original firmware source codes of access point like the Linksys WRT54GL from Cisco. These source codes are usually the enhanced version of the original codes modified by third party interest for free accessibility and usage. Hence, it usually violates the warranty of the original firmware that comes with the WRT54G’s.

2.16.1 DD-WRT Firmware

The DD-WRT firmware is a free version of Sveasoft firmware created by a guy named BrianSlayer. This came about when Sveasoft decide to start charging $20 to download their firmware on to their router. The micro edition of DD-WRT is the only firmware that supports vxworks. Vxworks is not Linux based firmware, instead it is a proprietary UNIX type operating system designed for embedded devices. Linksys WRT54G v5.0 and v6.0 come with this firmware and can barely be hacked, but with little memory space. The DD-WRT have so many good features like:

- The wireless radio transmit power can be increased
- Good Quality of Service (QoS) for WLAN and LAN clients.
- Good Wireless Distribution System (WDS), bridge and mesh networking support.
- Good network web interface solution.

2.16.2 OpenWRT “WhiteRussian RC6” Firmware

OpenWrt kernel was redesigned and customized by the OpenWrt project team in 2004, to write and support a GNU/Linux-based core having minimal Linux features on the WRT54G processor and network interfaces. A Debian-like package management system feature was also implemented, giving end users the capability of customizing supported software installation to meet their needs. Some of the packages that come with the OpenWrt is the Buildroot tools, Asterisk, Open VPN porting packages, FreeRadius, WiFidog capture portal software....

2.17 Kismet

Kismet application is an open source wireless network analyzer running on Linux, UNIX and Mac OS X. It is not supported by windows OS. Kismet is a passive sniffer used to detect any wireless 802.11a/b/g protocol complaint networks, even when the network has a non broadcasting hidden SSID (Secure Service set Identifier). Kismet can discover, log the IP range of any detected wireless network and report its signal and noise levels. It can sniff all management data packets from detected networks. Kismet can be used to locate, troubleshoot and optimize signal strength for access points and clients, as well as detect network intrusions.

2.18 Kismet Client and Kismet Server

The kismet protocol is used by the kismet server and kismet client to control the server and its capture sources. Kismet server is controlled from the kismet.conf files located in /usr/local/etc directory. The kismet.conf is
where most of the kismet server configuration is done. Here the ‘wireless adapter’ or ‘Source’ is configured on
the client computer and or , configured to indicate that the drone is of a remote source like kismet_drone, while
using special service port like port 3501. Another kismet protocol is the kismet drone/kismet server protocol
used by the kismet server to communicate with a remote drone. Here configuration changes can be made in the
kismet_drone.conf file, by modifying the ‘Source’ and ‘allowedhost’ files to suit the end users network segment
and drone type and version.

The drone runs as a daemon; being able to launch at boot time and run in the background responding to network
service request/ hardware activities and forwarding the request to other processes. For example, packet request
from available network hosts are processed by the kismet drone and sent to appropriate local client server port.
My MacBook laptop comes with Broadcom BCM4321wireless card which supports RFMON in its wl driver.
The wl driver is not capable of enabling monitor mode. To enable the RFMON capability, there are several
options and one of them is drone method which was adopted in this thesis. The other option would have been to
load an ndis driver using ndiswrapper application into the bcmwl5.sys file found inside the BCM driver. The
ndiswrapper is patched with bcmmon.diff, a bcmmon binary file. Through a series of compilation and
configuration the Broadcom driver is enabled. This method involves tweaking the OS kernel and requires some
good knowledge of root sources in the OS core. The drone applications are simple to configure and compile
without having to temper so much with the kernel.

2.19 NETSTUMBLER AND NETSTUMBLER ALTERNATIVES

NetStumbler or Network Stumbler is a free downloadable software or tool for windows. It can detect wireless
LAN using the IEEE802.11a/b/g WLAN protocol standard.
NetStumbler is commonly used for:

- WarDriving
- Verifying network configurations
- Finding locations with poor coverage in a wireless local area network (WLAN)
- Detecting causes of wireless interference
- Detecting unauthorized “Rogue” access points
- Aiming directional antennas for long haul WLAN links
- Can be integrated with GPS for mapping purposes.

Some of its limitations are:

- NetStumbler software does not officially work very well on windows vista or Mac OS.
- It uses active scanning to detect access points by sending out beacon probes request every second and
  then record the responses. This makes it vulnerable for detection in a wireless environment.
- It cannot detect wireless stations, since wireless stations do not respond to active probe request.

There are numerous alternative NetStumbler tools available today, all having different functionalities, purpose,
strength and weaknesses. Here are some of them:

- MacStumbler
- iStumbler
- Windows Vista Netsh
- Vistumbler
- Insider
- DISA Wireless Discovery Device (Flying Squirrel)

This thesis work shall only examine NETSH which comes by default in windows vista.
2.20 NETSH

NetStumbler is not officially supported on Windows Vista, but there exist several alternatives that provide similar information about wireless networks. The quickest and simplest way to discover access points is by using the “netsh” command available by default in Windows Vista. The command returns text results for discovered wireless networks. The Windows Vista Netsh commands for wireless local area network (WLAN) provide methods to configure connectivity and security settings. The Netsh wlan command can be used to configure a single local client computer or multiple computers by using a logon script. Netsh wlan command can also be used to view Group Policy settings and administer WISP (Wireless Internet Service Provider) and user wireless settings.

- Easier wireless deployment: Provides a light-weight alternative to Group Policy to configure wireless connectivity and security settings.
- Mixed mode support: Allow administrators to configure clients to support multiple security options. For example, a client can be configured to support both the WPA2 and the WPA authentication standard. This allows the client to use WPA2 to connect to networks that support WPA2 and use WPA to connect to networks that only support WPA.
- Block undesirable networks: Administrators can block and hide access to non-corporate wireless networks by adding specific networks or network types to the list of denied networks. Similarly, administrators can allow access to corporate wireless networks.
CHAPTER 3

3.0 LINUX DISTRIBUTION

There are different types of Linux Distribution:

- Debian Based for example: Ubuntu, EdUbuntu, Kubuntu
- Gentoo Based for example: iloog, Pentoox, Gentoox
- RPM Based: or Red Hat Linux also called Fedora.

Just to name a few of them. The Debian based Linux distribution used for this project is Ubuntu 9.04 32.bit version Jaunty Jackalope. See (APPENDIX A 1)

The new Ubuntu 9.04 interface browser comes with a beautiful user graphical interface that makes it very much like Microsoft Windows internet browser and it is user friendly. As a package, it comes with so many software and applications, editors, compilers and utilities, readily available through the synaptic package manager. The update manager makes it possible to run available updates automatically with authentication request without having to compile updates from source. Compiling from source is possible through the user terminal interface, mostly preferred by more experienced UNIX users and which was also adopted in this project. You don’t have to configure the computer monitor screen and its graphical components.

- You can install the editor from source also by opening a terminal and typing:

  # Sudo apt-get install gparted

3.1 PREPARING AND UPGRADING THE LINUX KERNEL (2.6.28-11-37 GENERIC)

It is possible to make alterations in the kernel to suite you purpose, but for a newbie in the use of Linux applications, setting the update manager to ‘automatic update’ and specifying the periodic will automatically present you with updates from time to time and it usually requires administrative password to perform any updates on the computer. Regular patches and upgrades are sent from time to time for a full functional computer. The lastest kernel upgrade can be installed by typing in the desktop terminal:

  stephen@stephen-laptop:~$ sudo apt-get install build-essential checkinstall

3.2 SOFTWARE AND HARDWARE PACKAGES

3.2.1 SYNAPTIC PACKAGE MANAGER AND SOFTWARE UPDATE MANAGER FEATURES

Launching the synaptic package manager makes it possible for a lot of software packages in the repository inherent in the Ubuntu distro to be installed when needed. The update manager can be launched also to check for updates.

3.2.2 KNOWING MY COMPUTER HARDWARE:

My HP laptop hardware is composed of two main network interface, an Ethernet (eth0) and a wireless (eth1) one with a Broadcom chipset BCM4321 (Windows OS) and BCM4328 (Linux). It has both multicast and broadcast support and the NIC is compatible with the iee802.11a/b/g/n protocol. It is a 2-core processor machine with 2GB making speed at installation and download of large packets. Unfortunately the Broadcom chipset does not support raw monitor mode See: (APPENDIX A 2) and the for supported device drivers and network cards:
3.3 LINKSYS WRT54GL

The Linksys WRT54GL comes with a GNU/LINUX Kernel 2.4.30 distribution, having basic Linux repository with reduced functional tools and applications capability. This means that the Linksys54gl cannot run the full standard application programs of a normal Linux which are normally in their thousands. The WRT54GL is an extended version of WRT54G v4.0 were the ‘L’ stands for ‘Linux’.

3.3.1 FIRMWARE UPGRADE AND CONFIGURATIONS

3.3.1.1 FLASHING THE WRT54GL WITH (OpenWRT “Whiterussian”) FIRMWARE:

Laptop IP address: 192.168.1.107
Router IP address: 192.168.1.1
Router: WRT54GL v1.1
OpenWRT “Whiterussian” version 0.9 is the Firmware.

The file I want to flash the WRT with is “openwrt-wrt54g-squashfs.bin” downloaded from here:
http://downloads.openwrt.org/whiterussian/rc4/bin/

```
root@stephen-laptop:/home/stephen# wget http://downloads.openwrt.org/whiterussian/rc4/bin/
```

Download the .bin file on to the desktop, to be uploaded onto the WRT later.

Index of /whiterussian/rc4/bin/

```
openwrt-wrt54g-squashfs.bin 24-Nov-2005 02:43 1545216
```

APPENDIX B 1 index of /whiterussian/rc4/bin

To flash the wrt54gl with the “openwrt-wrt54g-squashfs.bin” file, connect to the web control panel on the router by typing “http://192.168.1.1” on the web browser address window. Once inside the router, click System > Settings > Firmware Upgrade, click ‘browse’ to browse to the location of the previously downloaded .bin file on the desktop. Upload the .bin file and click “Firmware Upgrade” to violate the wrt54gl warranty. It is important to carry out this delicate flashing procedure on wired Ethernet connection because over wireless connection, this is a tendency for inherent wireless data packet loses which could jeopardize the operation. It is also advisable to enable the BOOT_WAIT before the flashing operation, but fortunately the BOOT_WAIT parameter is set “ON” as default in the wrt54gl. Enabling the BOOT_WAIT puts the wrt54gl on auto power cycle where the router automatically reboots.
3.3.1.2 SYSTEM SETTINGS AND BOOT_WAIT:

System > Settings:
Figure 3.3.1.2

3.3.1.3 ROUTER INFO:

Figure 3.3.1.3

3.4 ACCESS POINT SCAN RESULT AFTER FLASHING:

Figure 3.4 is a scan of the available networks around using the just flashed WRT router. It is for the purpose of comparing network captures and data therein with later captures when kismet binaries would be loaded into the router and the router used as drone. The scan reveals the network names, MAC address, encryption and authentication types, signal to noise ratio, frequency range of transmit and data rate delivery. This would be compared with using kismet for packets capture.

```
stephen@stephen-laptop:~$ sudo iwlist eth1 scan
[sudo] password for stephen:
```
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

eth1  Scan completed :

Cell 01 - Address: 00:24:01:1B:91:5F
ESSID:"ORISA"
Mode:Managed
Frequency:2.412 GHz (Channel 1)
Quality:5/5  Signal level:-17 dBm  Noise level:-87 dBm
IE: IEEE 802.11i/WPA2 Version 1
  Group Cipher : TKIP
  Pairwise Ciphers (1) : TKIP
  Authentication Suites (1) : PSK
IE: Unknown:
  Encryption key:on
  Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 6 Mb/s; 9 Mb/s
          11 Mb/s; 12 Mb/s; 18 Mb/s; 24 Mb/s; 36 Mb/s
          48 Mb/s; 54 Mb/s

Cell 02 - Address: 00:1C:F0:71:E3:0E
ESSID:"Adam Mortensen"
Mode:Managed
Frequency:2.437 GHz (Channel 6)
Quality:5/5  Signal level:-47 dBm  Noise level:-90 dBm
IE: Unknown:
  Encryption key:on
  Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 6 Mb/s; 9 Mb/s
          11 Mb/s; 12 Mb/s; 18 Mb/s; 24 Mb/s; 36 Mb/s
          48 Mb/s; 54 Mb/s

Cell 03 - Address: 00:21:04:10:B6:AA
ESSID:"Gigaset SE361"
Mode:Managed
Frequency:2.437 GHz (Channel 6)
Quality:2/5  Signal level:-78 dBm  Noise level:-90 dBm
IE: IEEE 802.11i/WPA2 Version 1
  Group Cipher : TKIP
  Pairwise Ciphers (2) : TKIP CCMP
  Authentication Suites (1) : PSK
Preauthentication Supported
IE: WPA Version 1
    Group Cipher : TKIP
    Pairwise Ciphers (2) : TKIP CCMP
    Authentication Suites (1) : PSK
    Encryption key: on
    Bit Rates: 1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 6 Mb/s
               12 Mb/s; 24 Mb/s; 36 Mb/s; 9 Mb/s; 18 Mb/s
               48 Mb/s; 54 Mb/s

steven@steven-laptop:~$  

Figure 3.4

3.5 CONFIGURING CLIENT SERVER DESKTOP

3.5.1 CONFIGURING THE DESKTOP WIRELESS NETWORK INTERFACE:

To set up the desktop wireless network interface, there is a need to first create a user SSID name for the network you want to use. I used “ORISA2” as the SSID for both laptop network interface and the wrt54gl. It is important that both the client server and wrt54gl have the same user SSID for effective communication. The desktop will fail to recognize the access point if the SSID are different. The mode should be “infrastructure” when used as an access point and not “Ad Hoc” (Ad-Hoc is usually configured for used as VPN).

3.5.2 SECURITY: Having no security on the desktop network interface and the router implies a great security lapse for the entire network set up. Security was set to ‘None’ for the purpose of this experimentation.

3.5.3 STATUS:  
The figure below (Figure 3.5.3a) is detailed information about the wireless network interface settings. The ‘wl’ driver is associated with the network interface card BCM4328 (BCM4321 the windows equivalent of BCM4328) which does not support “RFMON” hence, the need to use the WRT as drone and the desktop as the server/client machine. Eth1 (192.168.1.107) here refers to the wireless network interface while eth0 (192.168.1.129) refers to the wired Ethernet network connection between desktop and WRT. (10.42.43.1) is the internet IP address of the network. It is vital that the router recognizes the IP address 192.168.1.107of the client/server machine (the desktop). Any IP address will work anyway.
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

Figure 3.5.3 (a)

A list of the Linksys54gl repositories after the flash with OpenWrt can be found in Appendix B 1: The bin folder and the etc folders contains a relative concise amount of useful tools like the scp, ssh, Telnet, killall, wget but to name a few.

See (APPENDIX B 1)

3.6 SETTING UP WRT54GL ON MY NETWORK (diagram of setup box)

Figure 3.6 is my wireless configuration network setup. The Macbook laptop serves as the kismet server/client while the WRT54GL router is acting as my kismet drone. Configured to be able to communicate with the internet and surrounding wireless networks.

The configured computer interface card and the configured WRT54GL wireless access point are linked wirelessly. The configuration makes it possible for both to communicate effectively by sniffing data packets of the air the drone is able to detect neighbouring wireless access points and registering specific parameters; like the network name, the network MAC address and other parameters.
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

Figure 3.6

3.7 KISMET DOWNLOAD AND INSTALLATIONS:

Open the terminal as root user in Ubuntu and download the kismet binaries from www.kismetwireless.net: in to the tmp folder. Requires root privileges: (APPENDIX C 3)

```
root@stephen-laptop:/home/stephen#
wget http://www.kismetwireless.net/code/kismet-2006-04-R1-wrt54.tar.gz
```

- Untar/gzip the kismet package:
  ```
tar –zxvf http://www.kismetwireless.net/code/kismet-2006-04-R1-wrt54.tar.gz
  ```

The kismet file is automatically downloaded to kismet folder in the etc directory also.

See also kismet download and its repositories: (APPENDIX C 1)

- To make all the kismet files executable:
  ```
stephen@stephen-laptop:/tmp/kismet$ -rwxr--r-- 1 root root 1347
  ```

Light green coloured files indicate executable files.

3.7.1 KISMET DRONE INSTALLATION AND CONFIGURATION

From desktop terminal ‘ssh’ into the router as root user:
root@OpenWrt:/# ssh 192.168.1.107

root@OpenWrt:~/kismet-2006-04-R1-wrt54# ls

- **Update the ipkg files and install wl driver:**
  root@OpenWrt:~/kismet-2006-04-R1-wrt54/conf# ipkg update
  root@OpenWrt:~/kismet-2006-04-R1-wrt54/conf# ipkg install wl

- **cd into kismet-2006-04-R1-wrt54/conf:**

- **Make the kismet.conf files executable:**
  Chmod +R 777

root@OpenWrt:~/kismet-2006-04-R1-wrt54# ls -l

- **Secure Copy kismet_drone into the /usr/bin directory in the wrt:**

  root@OpenWrt:~/kismet-2006-04-R1-wrt54# scp kismet_drone
  /usr/bin/kismet_drone

- **check if the files in the /usr/bin are executable:**

- **copy the kismet_drone.conf file into the etc directory:**

  root@OpenWrt:~/kismet-2006-04-R1-wrt54/conf# scp kismet_drone.conf
  /etc/kismet_drone.conf

### 3.7.2 Kismet Client and Kismet Server Installation and Configuration

- **Navigate to the desktop kismet client/server:**

  root@OpenWrt:/# ssh 192.168.1.107
  root@stephen-laptop:~#

- **Install ascii:**

  root@stephen-laptop:~# apt-get install ascii

### 3.7.2.1 Editing the kismet.conf for the Kismet Client/Server Desktop:

- **Using vi editor to edit the kismet.conf file:**

  root@stephen-laptop:/etc/kismet# vi kismet.conf

  setup the kismet.conf file in the desktop to:
source=kismet_drone,192.168.1.107:3501,drone

3.7.2.2 EDITING THE KISMET_DRONE.CONF FILES FOR THE ROUTER:

Change the ‘source’ and the ‘allowedhosts’ to:
Source=wrt54g,prism0,wrt54g

Allowedhosts=127.0.0.1,192.168.0.0/24

Secure copy /tmp/kismet-2006-04-R1-wrt54/kismet_drone into /usr/bin/kismet_drone and /tmp/kismet-006-04-R1-wrt54/kismet_drone.conf into /etc/kismet_drone.conf. Ensure they are executable.

S -rw-r--r- 1 root root

root@OpenWrt:~/kismet-2006-04-R1-wrt54/conf# cp kismet_drone.conf /etc/kismet_drone.conf

root@OpenWrt:~/kismet-2006-04-R1-wrt54# cp kismet_drone /usr/bin/kismet_drone

This script will start the drone automatically and run the drone as a service:

#!/bin/sh
echo "Setting radio for kismet_drone"
mkdir /var/log
/sbin/ifconfig eth1 up
/usr/sbin/wl ap 0
/usr/sbin/wl disassoc
/usr/sbin/wl passive 1
/usr/sbin/wl promisc 1
## /usr/sbin/wl monitor 1
echo "Running kismet_drone"
/usr/bin./kismet_drone -f /etc/kismet_drone.conf> /dev/null 2>&1 &
Sleep 3
echo "kismet_drone now running"
chmod 777 /etc/init.d/S60kismet_drone

## /usr/sbin/wl monitor 1 - this monitor line is commented out bcos wl uses prism0 in my WRT. And it worked without 'monitor' but with monitor it could not identify my router settings and configs.

- Verify that the file is there and executable:

root@OpenWrt:/etc/init.d# chmod 77 /etc/init.d/S60kismet_drone
root@OpenWrt:/etc/init.d# ls
S05nvram S50dropbear S60dnsmasq rcs
S10boot S50httpd S60kismet_drone
S35firewall S50telnet S70WL_scan

26
3.7.3 Kismet GUI and Kismet Launch

Before the kismet GUI is launched, it is important that the following scripts be written into the /tmp folder of the WRT.

Using vi editor: This ‘rundrone.sh’ script with a ‘wl passive’ command is included so that the router does not start an active scan immediately and generating packets instead of gathering packets.

```bash
wl ap 0
wl diassoc
wl passive 1
wl promisc 1
chmod 777 /usr/bin/kismet*
/usr/bin/./kismet_drone –f /etc/kismet_drone.conf
```

The chmod 777 will make the kismet_drone.conf and the kismet_drone files executable.

3.7.4 Channel Hopping Script

The wrt does not channel hop and to make it to channel hop, a ‘scanme.sh’ script written into the created /etc/init.d/S70JW_scan file using vi editor:

```bash
#!/bin/sh
while : ; do
  wl channel 1 ; sleep 1
  wl channel 6 ; sleep 1
  wl channel 11 ; sleep 1
  wl channel 2 ; sleep 1
  wl channel 7 ; sleep 1
  wl channel 3 ; sleep 1
  wl channel 8 ; sleep 1
  wl channel 4 ; sleep 1
  wl channel 9 ; sleep 1
  wl channel 5 ; sleep 1
  wl channel 10 ; sleep 1
done
```

Put the script in the /etc/init.d/S70JW_scan file and make it executable:

```bash
chmod 777 /etc/init.d/S70JW_scan
```
3.7.5 RUNNING THE DRONE AND LAUNCHING KISMET GUI:

To Run The Drone:

```bash
root@OpenWrt:/etc/init.d# /usr/bin/./kismet_drone -f /etc/kismet_drone.conf
Using alternate config file: /etc/kismet_drone.conf
Suid priv-dropping disabled. This may not be secure.
No specific sources given to be enabled, all will be enabled.
Disabling channel hopping.
Source 0 (wrt54g): Enabling monitor mode for wrt54g source interface prism0
channel 0...
Source 0 (wrt54g): Opening wrt54g source interface prism0...
NOTICE: bind address not specified, using INADDR_ANY.
Kismet Drone 2006.04.R1 (Kismet)
Listening on port 3501 (protocol 9).
Allowing connections from 127.0.0.1/255.255.255.255
Allowing connections from 192.168.1.0/255.255.255.0
Accepted streamer connection from 192.168.1.107
```

3.7.6 TO LAUNCH KISMET:

```bash
root@stephen-laptop:/home/stephen# kismet

Launching kismet_server: //usr/bin/kismet_server
Suid priv-dropping disabled. This may not be secure.
No specific sources given to be enabled, all will be enabled.
Enabling channel hopping.
Enabling channel splitting.
NOTICE: Disabling channel hopping, no enabled sources are able to change channel.
Source 0 (drone): Opening kismet_drone source interface 192.168.1.1:3501...
Allowing clients to fetch WEP keys.
Gathering packets...
Launched client, pid 12523
Launching kismet_client: //usr/bin/kismet_client
NOTICE: Group file did not exist, it will be created.
Looking for startup info from localhost:2501..... found.
Connected to Kismet server 2008.05.R1 on localhost:2501
WARNING: Unable to open '/etc/kismet/ap_manuf' for reading (No such file or directory), AP manufacturers and defaults will not be detected.
WARNING: Unable to open '/etc/kismet/client_manuf' for reading (No such file or directory), client manufacturers will not be detected.
configdir '/root/.kismet/' does not exist, making it.
SSID cloak file did not exist, it will be created.
IP track file did not exist, it will be created.
Logging networks to Kismet-Nov-15-2009-1.network
Logging networks in CSV format to Kismet-Nov-15-2009-1.csv
Logging networks in XML format to Kismet-Nov-15-2009-1.xml
Logging cryptographically weak packets to Kismet-Nov-15-2009-1.weak
Logging cisco product information to Kismet-Nov-15-2009-1.cisco
Logging gps coordinates to Kismet-Nov-15-2009-1.gps
Logging data to Kismet-Nov-15-2009-1.dump
Writing data files to disk every 300 seconds.
```
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

Mangling encrypted and fuzzy data packets.
Tracking probe responses and associating probe networks.
Using network-classifier based data encryption detection
Not tracking duplicate IVs
Dump file format: wiretap (local code) dump
Crypt file format: airsnort (weak packet) dump
Kismet 2008.05.R1 (Kismet)
Logging data networks CSV XML weak cisco gps
GPSD cannot connect: Connection refused
Listening on port 2501.
Allowing connections from 127.0.0.1/255.255.255.255
Registering builtin client/server protocols...
Registering requested alerts...
Registering builtin timer events...
Didn't see any weak encryption packets, unlinking weak file
Kismet exiting.
Done.
root@stephen-laptop:/home/stephen#

This following command will disconnect the router from the internet. So prevent this from happening by opening another terminal to run the command:

root@OpenWrt:/etc/init.d# wl ap 0
wl diassoc
wl passive 1
wl promisc 1

My network 'ORISA2' was disconnected and I could not connect wirelessly back to the internet. Rebooting the router and the laptop did not resolve the wireless disconnection issue. Until I had to use cable connection to the router 'ethernet connection' and change the ssid=show, then 'save' and 'apply changes' and try connecting to ORISA2 again and it worked. The reason for this failure was bcos I had already scripted for the drone to run on service mode automatically when powered. So, manually running the drone again was not necessary. Running this last command as root; /usr/bin./kismet_drone -f/etc/kismet_drone.conf was all I needed to run the drone.

3.8 PLATFORMS

3.8.1 LOCATING ACCESS POINTS

Once kismet is launched it automatically detects networks a few yards from its location. However, with an increased antenna gain and antenna height, the range of access points detected increases.

3.9 NETSH (NetworkShell) UTILITY

3.9.1 NETWORK CONNECTION STATUS:

From figure 3.9.1(a) below, one can see the security level and signal strength of each private network along with its SSID all displayed. But we are not told what type of security that the network is using. To reveal more about a network, figure 3.9.1 (b1) & (b2), the netshell utility can come handy. By opening a terminal interface as an administrator and typing:

C:\Users\Stephen> netsh wlan show networks mode=bssid
The type of encryption and authentication is revealed. The network id, Mac address, radio type, channel type of network and signal strength in percent are also revealed. The network ‘ORISA’ can be seen here to be a WPA2-Personal authentication and a TKIP encryption. The bssid is also given and ‘ORISA2’ has no authentication nor encryption i.e no security. Mac address, IP address and the SSID are key elements in a network. This information in the hands of a would-be attacker could become very useful in compromising a network. There exist today cracker softwares like aircrack-ng, wireshark that can lay bare encrypted cipher text into readable text formats.

Microsoft Windows [Version 6.0.6002]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.

C:UsersStephen>netsh
netsh>wlan
netsh wlan>show networks mode=bssid

Interface Name : Wireless Network Connection
There are 4 networks currently visible.

SSID 1 :
Network type : Infrastructure
Authentication : Open
Encryption : None
BSSID 1 : 00:25:9c:24:5a:cd
    Signal : 100%
    Radio Type : 802.11g
    Channel : 1
    Basic Rates (Mbps) : 1 2 5.5 11
    Other Rates (Mbps) : 6 9 12 18 24 36 48 54

SSID 2 : hany
Network type : Infrastructure
Authentication : Open
Encryption : WEP
BSSID 1 : 00:1c:f0:86:34:44
    Signal : 81%
    Radio Type : 802.11g
    Channel : 6
    Basic Rates (Mbps) : 1 2 5.5 11
    Other Rates (Mbps) : 6 9 12 18 24 36 48 54

SSID 3 : Gigaset SE361
Network type : Infrastructure
Authentication : WPA2-Personal
Encryption : CCMP
BSSID 1 : 00:21:04:10:b6:aa
    Signal : 26%
    Radio Type : 802.11g
    Channel : 6
    Basic Rates (Mbps) : 1 2 5.5 11
    Other Rates (Mbps) : 6 9 12 18 24 36 48 54
SSID 4 : ORISA2
Network type    : Infrastructure
Authentication  : Open
Encryption      : None
BSSID 1         : 00:25:9c:24:5a:cd
Signal          : 100%
Radio Type      : 802.11g
Channel         : 1
Basic Rates (Mbps) : 1 2 5.5 11
Other Rates (Mbps) : 6 9 12 18 24 36 48 54

netsh wlan>

Figure 3.9.1 (b1)

Figure 3.9.1 (a)
Figure 3.9.1 (b2)
CHAPTER 4

4.0 WIRESHARK.

Wireshark is a network protocol analyser for Unix and Windows operating systems. It uses the API program in the packet capture (pcap) to capture live data packets from networks and also saved downloaded packet files in libpcap formats. Wireshark is also used for troubleshooting, analysis, software and communication development as well as education. Formally called Ethereal, but was later changed to Wireshark due to trademark issue. Wireshark is free software released under the GNU General Public License terms.

4.1 CRACKING AND ANALYSING CAPTURED DATA PACKETS USING WIRESHARK

# File types to log, comma seperated
# dump   - raw packet dump
# network - plaintext detected networks
# csv    - plaintext detected networks in CSV format
# xml    - XML formatted network and cisco log
# weak   - weak packets (in airsnort format)
# cisco  - cisco equipment CDP broadcasts
# gps    - gps coordinates
logtypes=dump,network,csv,xml,weak,cisco,gps

4.2 ANALYSING MANAGEMENT DATA PACKETS AND LOGS

Wireshark was downloaded and installed.
To launch the wireshark GUI, open a terminal as root and type in ‘wireshark’ and enter:

root@stephen-laptop:/home/stephen/kismet-2006-04-R1-wrt54/conf# wireshark

To open the captured packets from the wireshark GUI:

Browse to folder ‘kismet-2006-04-R1-wrt54/conf’> captured file> ‘open’ to enter.

Figure 4.2 below is the dump folder for captured packets with wireshark. The file name is Kismet.dump and the date of capture is indicated, total packet size and format are registered also. Other packet type like .xml, .csv, .network are captured along with the .dump files.
4.2.1 ANALYSING A CAPTURED PACKET:

This is a capture from the network with the client ip address 192.168.0.109. Using wireshark, I was able to open the packet and see what website the user was visiting. The same could be said, if the user had used a password login, provided the web page is unsecured (just ‘http’ and not ‘https’). Secured web pages are usually encrypted and the captured packet by the sniffer, though detected in plain text will only be seen as codes. The password can be read as plain text in unsecured and unencrypted web pages. The protocol used by this client is the SSDP.

4.2.2 SIMPLE SERVICE DISCOVERY PROTOCOL (SSDP)

SSDP is an expired IETF Internet draft by Microsoft and Hewlett-Packard. SSDP is the basis of the discovery protocol of Universal plug-and-play.

SSDP provides a mechanism which network clients can use to discover network services. Clients can use SSDP with little or no static configuration.

SSDP uses UDP unicast and multicast packets to advertise available services. The multicast address used is 239.255.255.250 in IPv4. SSDP over IPv6 uses the following multicast addresses, depending on scope:

- Node-local: [FF01::C]
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

- Link-local: [FF02::C]
- Site-local: [FF05::C]
- Organization-local: [FF08::C]
- Global: [FF0E::C]

SSDP uses port 1900.

![Wireshark Packet Capture](image.png)

**Figure 4.2.2 (a)**

**INSIDE THE PACKET (Figure 4.2.2 a):**

To open the packet: Http > right click 'Follow TCP Stream':

Wireshark is able to also expose the source and destination IP addresses of both host and client.
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

INSIDE THE PACKET (Figure 4.2.2 b):

To open the packet: Http > right click 'Follow TCP Stream':

My host network computer was actually browsing a website:
http://suanaa.bilddagboken.se/p/rightAd_ad_bottom.html
There are other information like what browser is the person using, language, and the host address. This entry is a web browser trying to open a webpage.

4.3 TYPES OF WIRELESS FRAMES:

- **Control frames** help data frames delivery like in acknowledgement.
- **Data frames** help transport the data
- **Management frames** establish and maintain wireless communication like in probe request, probe response and beacon frames.

See: APPENDIX D
1. MANAGEMENT FRAMES:

When a wireless client sends a probe request in order to find out which wireless access point are available within range, the AP usually will respond with information about its capabilities, like network name (SSID), channel, supported rates, and so on. If the SSID is non-broadcast the response would be ‘no ssid’.

I noticed that my 'login' where coming in as ' HTTP/TCP Previous segment lost] '. This error code coloured in red actually is supposed to mean a packet had arrived with a sequence number greater than the "next expected sequence number" on that connection, indicating that one or more packets prior to the flagged packet did not arrive. The packet loss will likely be accompanied by "TCP Retransmission" events. The reason for this is that, the network is most likely congested. It can also occur with bad network hardware and or poorly configured networks. But again the kismet drone setup is not such a robust system given some of its defect like no channel hopping capability.
4.3.1 BEACON FRAME:

Figure 4.3.1 above is an example of a beacon frame. Clients are able to identify access points because beacons are released by AP’s at regular intervals. A beacon with a hidden ssid will be sent blank.
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

Figure 4.3.2

4.3.2 ICMP AND DATA FRAMES

Figure 4.3.2 above is an example of ICMP an error messaging protocol report of requested data frames or services. It is sort of a Quality of Service (QoS) control mechanism indicating an unreached host, services or router in the network.
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

4.3.3 PROBE REQUEST:

Figure 4.3.3 above is an example of a probe request frame. Kismet sends out passively probe request frames and neighbouring networks that last generated beacon frames receive the request and respond with a probe response frame which contains specific information about the network. Probe request are sent out to obtain information from access points or to determine if they are available.

4.4 DIFFERENCES IN (.csv, .network, .dump, .xml, .cisco, .weak) FILES

Dumps are written in pcap (we also have winpcap for windows) formats and these captured packets (.dump) contain data in binary which is the standard used by many network tools just like wireshark. Other formats are not in pcap format and therefore not recognised by wireshark.

4.4.1 XML FILES:

The .xml file is written in eXtensible Mark-up Language where every data is defined by a markup for a precise identification either by a program or by a person. It gives more details of the network location if GPSD were installed and enabled. XML has become very popular and acceptable because the tags are not predefined, you can define your tags and therefore find compatibility with all know applications. It stores and transport data between applications like the various web browsers.

Found in the .xml files are the following contents:

- Wireless Network Number: “1”, “2”... which refers to the number of computers in that network.
- Network Connection Type: Infrastructure, Ad-Hoc, todos, sendto, fromds, interds
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

- Client MAC address
- Network Name (SSID)
- If network is “Cloaked” or “Uncloaked”
- Network channel
- Logical Link Control: Through tcp port 23, tcp/udp port 1646 (information protocol), tcp/udp port 263
- IP-Address type : tcp, arp
- Max. Rate: 22Mbps, 54Mbps..
- Data Size captured
- GPS Info: specify network positioning in longitude, latitude and and gps altitude and speed.
- Duplicate Initialization Vector (dupeiv)
- Encryption type: WEP, WPA...
- Weak Packets

4.4.2 .csv FILES
The .csv (Comma Separated values) file is able to tell what ip-type a network is running, for example if it is a TCP/UDP type, the .csv files are formatted plain text files where data are separated by either commas, or semi-colons. This type of file is usually produced with spreadsheet text only format for exporting data between program applications or systems. Data tables displayed in comma delimited CSV text file format can be retrieved into different applications used to create spreadsheets and databases and it also allow for easy manipulation of data in columnar format. Data tables can be viewed and saved in CSV format.

Found in the .csv files are much of the same as the .xml files and the .network files except, it has provisions for:

- Best Packet frame Quality.
- Best Signal and Best Noise, and the possibility to measure signal-to-noise ratio.
- The Beacon rate, at a BAUD rate of 25600 an indication of the amount of data traffic a network can handle. The are other baud rates for radio transmission (9600, 57600, 19200 bauds )
- Decrypted info.

4.4.3 .NETWORK FILES
The .network file does not show much, it cannot tell if the network is cloaked or the ip-type. But it is possible to know what carrier and encoding it is using.

4.4.4 .cisco FILES
The .cisco file was blank because there are not cisco AP’s, cisco files are suppose to provide information on Cisco Discovery Protocol (CDP) broadcast. CDP was developed by cisco system, it is a proprietary layer 2 network protocol which runs on Cisco equipment and some HP products.

4.4.5 .WEAK FILES
The file records the number of packets with weak initialization vectors (IV). These weak IVs are enabled due to irregularities in the RC4 encryption algorithm which in turn reveals information about a WEP key.

OBSERVATIONS:

My observation was that, the .network, .csv, .xml, .cisco were all invalid packets in an attempt to use wireshark to open them. This is because wireshark does not recognise them as valid packets. They were in formats not recognisable by wireshark. Only the .dump file was recognised by wireshark because it was stored as a libpcap format.
4.5 EXAMPLES OF .XML, .NETWORK, .CSV, .CISCO FILES

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE detection-run SYSTEM "http://kismetwireless.net/kismet-2005-03.dtd">
<detection-run kismet-version="2008.05.R1" start-time="Mon Mar 29 06:54:19 2010" end-time="Mon Mar 29 07:10:23 2010">
  <wireless-network number="1" type="infrastructure" wep="false" cloaked="false" first-time="Mon Mar 29 06:54:19 2010" last-time="Mon Mar 29 07:10:23 2010">
    <BSSID>00:25:9C:24:5A:CD</BSSID>
    <channel>0</channel>
    <maxrate>0.0</maxrate>
    <maxseenrate>0</maxseenrate>
    <encryption>None</encryption>
    <packets>
      <LLC>23</LLC>
      <data>63378</data>
      <crypt>0</crypt>
      <weak>0</weak>
      <dupeiv>0</dupeiv>
      <total>63401</total>
    </packets>
    <datasize>4568208</datasize>
    <gps-info unit="english">
      <min-lat>90.000000</min-lat>
      <min-lon>180.000000</min-lon>
      <min-alt>0.000000</min-alt>
      <min-spd>0.000000</min-spd>
      <max-lat>-90.000000</max-lat>
      <max-lon>-180.000000</max-lon>
      <max-alt>0.000000</max-alt>
      <max-spd>0.000000</max-spd>
    </gps-info>
    <ip-address type="tcp">
      <ip-range>192.168.1.107</ip-range>
    </ip-address>
  </wireless-network>
  <wireless-network number="2" type="infrastructure" wep="false" cloaked="false" first-time="Mon Mar 29 06:54:19 2010" last-time="Mon Mar 29 07:10:23 2010">
    <SSID>MiZi</SSID>
    <BSSID>00:22:B0:86:36:92</BSSID>
    <channel>1</channel>
    <maxrate>54.0</maxrate>
  </wireless-network>
</detection-run>
```
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

Example of .xml file:

```xml
<LLC>1646</LLC>
  <data>37</data>
  <crypt>0</crypt>
  <weak>0</weak>
  <dupeiv>0</dupeiv>
  <total>1683</total>
</packets>
<datasize>2238</datasize>
<gps-info unit="english">
  <min-lat>90.000000</min-lat>
  <min-lon>180.000000</min-lon>
  <min-alt>0.000000</min-alt>
  <min-spd>0.000000</min-spd>
  <max-lat>-90.000000</max-lat>
  <max-lon>-180.000000</max-lon>
  <max-alt>0.000000</max-alt>
  <max-spd>0.000000</max-spd>
</gps-info>
<ip-address type="arp">
  <ip-range>192.168.0.100</ip-range>
</ip-address>
<wireless-network number="3" type="infrastructure" wep="false" cloaked="false" first-time="Mon Mar 29 06:54:20 2010" last-time="Mon Mar 29 07:10:23 2010">
  <SSID>NETGEAR</SSID>
  <BSSID>00:24:B2:24:6D:74</BSSID>
  <channel>1</channel>
  <maxrate>54.0</maxrate>
  <maxseenrate>0</maxseenrate>
  <encryption>None</encryption>
  <packets>
    <LLC>263</LLC>
    <data>4</data>
    <crypt>0</crypt>
    <weak>0</weak>
    <dupeiv>0</dupeiv>
    <total>267</total>
  </packets>
  <datasize>240</datasize>
</wireless-network>
```
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

<min-lat>90.000000</min-lat>
<min-lon>180.000000</min-lon>
<min-alt>0.000000</min-alt>
<min-spd>0.000000</min-spd>
<max-lat>-90.000000</max-lat>
<max-lon>-180.000000</max-lon>
<max-alt>0.000000</max-alt>
<max-spd>0.000000</max-spd>
</gps-info>

<ip-address type="arp">
<ip-range>192.168.1.1</ip-range>
</ip-address>

<wireless-network number="4" type="infrastructure" wep="true" cloaked="false" first-time="Mon Mar 29 06:54:21 2010" last-time="Mon Mar 29 07:10:23 2010">
<SSID>IRAQ</SSID>
<BSSID>00:11:95:41:FB:9D</BSSID>
<channel>1</channel>
<max-rate>22.0</max-rate>
<encryption>WEP</encryption>
<packets>
<LLC>1110</LLC>
<data>0</data>
<crypt>0</crypt>
<weak>0</weak>
<dupeiv>0</dupeiv>
</packets>
<datasize>0</datasize>
</wireless-network>

<gps-info unit="english">
<min-lat>90.000000</min-lat>
<min-lon>180.000000</min-lon>
<min-alt>0.000000</min-alt>
<min-spd>0.000000</min-spd>
<max-lat>-90.000000</max-lat>
<max-lon>-180.000000</max-lon>
<max-alt>0.000000</max-alt>
<max-spd>0.000000</max-spd>
</gps-info>
Example of .network file:

Network 1: ":<no ssid>" BSSID: "00:25:9C:24:5A:CD"
Type : infrastructure
Carrier : unknown
Info : "None"
Channel : 00
Encryption : "None"
Maxrate : 0.0
LLC : 23
Data : 63378
Crypt : 0
Weak : 0
Dupe IV : 0
Total : 63401
First : "Mon Mar 29 06:54:19 2010"
Last : "Mon Mar 29 07:10:23 2010"
Min Loc: Lat 90.000000 Lon 180.000000 Alt 0.000000 Spd 0.000000
Max Loc: Lat -90.000000 Lon -180.000000 Alt 0.000000 Spd 0.000000
Address found via TCP 192.168.1.107

Network 2: "MiZi" BSSID: "00:22:B0:86:36:92"
Type : infrastructure
Carrier : 802.11n 40MHz
Info : "None"
Channel : 01
Encryption : "None"
Maxrate : 54.0
LLC : 1646
Data : 37
Crypt : 0
Weak : 0
Dupe IV : 0
Total : 1683
First : "Mon Mar 29 06:54:19 2010"
Last : "Mon Mar 29 07:10:23 2010"
Min Loc: Lat 90.000000 Lon 180.000000 Alt 0.000000 Spd 0.000000
Max Loc: Lat -90.000000 Lon -180.000000 Alt 0.000000 Spd 0.000000
Address found via ARP 192.168.1.107

Type : infrastructure
Carrier : unknown
Info : "None"
Channel : 01
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

Encryption: "None"
Maxrate: 54.0
LLC: 263
Data: 4
Crypt: 0
Weak: 0
Dupe IV: 0
Total: 267
First: "Mon Mar 29 06:54:21 2010"
Last: "Mon Mar 29 07:10:23 2010"
Min Loc: Lat 90.000000 Lon 180.000000 Alt 0.000000 Spd 0.000000
Max Loc: Lat -90.000000 Lon -180.000000 Alt 0.000000 Spd 0.000000
Address found via ARP 192.168.1.1

Network 4: "IRAQ" BSSID: "00:11:95:41:FB:9D"
Type: infrastructure
Carrier: unknown
Info: "None"
Channel: 01
Encryption: "WEP"
Maxrate: 22.0
LLC: 1110
Data: 0
Crypt: 0
Weak: 0
Dupe IV: 0
Total: 1110
First: "Mon Mar 29 06:54:21 2010"
Last: "Mon Mar 29 07:10:23 2010"
Min Loc: Lat 90.000000 Lon 180.000000 Alt 0.000000 Spd 0.000000
Max Loc: Lat -90.000000 Lon -180.000000 Alt 0.000000 Spd 0.000000

Example of .csv file:

Network;NetType;ESSID;BSSID;Info;Channel;Cloaked;Encryption;Decrypted;MaxRate;Max SeenRate;Beacon;LLC;Data;Crypt;Weak;Total;Carrier;Encoding;FirstTime;LastTime;BestQuality;BestSignal;BestNoise;GPSMinLat;GPSMinLon;GPSMinAlt;GPSMinSpd;GPSMaxLat;GPSMaxLon;GPSMaxAlt;GPSMaxSpd;GPSBestLat;GPSBestLon;GPSBestAlt;DataSize;IPType;IP;
1;infrastructure;<no ssid>;00:25:9C:24:5A:CD;;0;None;None;No;0:0:0:0:23:63378:0:0:63401;;;Mon Mar 29 06:54:19 2010;Mon Mar 29 07:10:23 2010;0:1:0:90.000000:180.000000:0.000000:0.000000:0.000000:0.000000:-90.000000:-180.000000:0.000000:0.000000:0.000000:0.000000:0.000000:0.000000:4568208;TCP;192.168.1.107;
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

4.6 PACKET RATE:

The command reference ‘r’ can display bar graph of the packet rates as they are delivered. Figure 4.6 is an example of a packet rate capture displayed in a bar graph.

![Packet Rate Capture](image-url)

Figure 4.6
4.7 CHANNEL HOPPING IN KISMET

![Kismet Network List](image)

The Kismet drone setup usually cannot channel hop unless a channel hopping script called a ‘scanme.sh’ is written to enable it, then it can channel hop as seen above.

4.8 SSID vs CLOAKED SSID

Kismet can capture the Service Set Identifier (SSID) and display it, whether the router is cloaked or not. Kismet passively listens to beacon frames and probe responses from surrounding wireless AP’s broadcast to be able to detect wireless network names and with time de-cloak, cloaked access points.

4.9 KISMET FEATURES

- Kismet can capture the MAC address of any computer on the network.
- It can also display the TCP/UDP transmission port
- The IP Address of the captured unsecured AP on the network, as well as the IP address of the client machine.
- With wireshark, the client source/destination IP address, client and host MAC address can all be displayed.
- The type of encryption, radio type, maximum data transmission rate,
It does not decrypt the captured packets, wireshark does, but with some limitations, like if the webpage was a https.

- Ethereal/tcpdump compatible data logging.
- Airsnort compatible weak initialization vector packet logging
- Network IP range detection
- Built-in channel and multicard split channel hopping
- Hidden network SSID de-cloaking
- Graphical mapping of networks
- Client/Server architecture allows multiple clients to view a single server simultaneously.
- Manufacturer and model identification of access points and clients
- Detection of known default access points configurations
- Runtime decoding of WEP packets for known networks
- Named pipe output for integration with other tools such as layer 3 IDS like snort
- Multiplexing of multiple simultaneous capture sources on a single kismet instance.
- Distributed remote drone sniffing
- XML output.

4.9.1 COMMON USES OF KISMET:

Some common application kismet is used for are:

- **Wardriving**: Mobile detection of wireless networks, logging and mapping of network location, WEP, network name, IP-Range etc.
- **Site Survey**: Monitoring and graphing signal strength and location.
- **Distributed IDS**: Multiple Remote Drone sniffers distributed throughout an installation monitored by a single server and could be combined with a layer 3 IDS like snort.
- **Rogue Access Point Detection**: Stationary or mobile sniffers to enforce site policy against rogue access points.

4.10 WEP:

WEP is vulnerable because of relatively short IVs and keys that remain static. If a hacker collects enough frames based on the same IV, called "weak IV", the hacker can in fact determine the shared secret key.
4.11 KISMET AND IP ADDRESSES:

Kismet can actually reveal the IP address of a captured access point if the AP is unsecured. The IP address 192.168.0.1 and 192.168.1.5 of the above client machines is a typical example. Wireshark is worse because it will reveal not only the client machine on the network but also the internet IP address, the client source/destination address and all the TCP/UDP connections. Below is the detail of the IP address 192.168.1.5.

--- Network Details ---

| Name   : NETGEAR                        |
| SSID   : NETGEAR                        |
| Server : localhost:2501                |
| BSSID  : 00:24:B2:24:6D:74            |
| Manuf  : Unknown                       |
| Max Rate: 54.0                         |
| BSS Time: 6db479183                    |
| First  : Sun Mar 28 20:19:50 2010      |
| Latest : Sun Mar 28 21:25:30 2010      |
| Clients : 2                            |
| Type   : Access Point (infrastructure) |
| Info   :                                |
4.12 HACKING WITH MAC ADDRESSES AND IP ADDRESSES:

Figure 4.11 (a) & (b) is a good example, how kismet can expose hidden network information like the network name and MAC address which can be spoofed by a would be attacker as explained in section 4.12.1. Similarly, the same thing can be said of displayed IP-Ranges and addresses of computers in the network.

4.12.1 MAC ADDRESSES SPOOFING:

Figure 4.11 (a) & (b) has the following network name (SSID= NETGEAR), MAC address= 00:24:B2:24:6D:74 and other information in the network. These information are necessary network parameters needed for the network administrator to be able to troubleshoot his network. But, this is the same network parameter would-be attacker would use.

An attacker can gather valid MAC addresses (see Figure 4.12.1) of various machines on the network for use in constructing spoofed frames, these frames would be required to hack in host computers on the internet. The aim of the attacker is to collect these MAC addresses, spoof his own frames by mimicking an authentic registered MAC address in order to circumvent security settings in an AP like the filtering out of MAC addresses not included as MAC addresses to be given access, while hiding or cloaking its identity.

4.12.2 IP ADDRESS SPOOFING:

Figure 4.11 (a) & (b) contains IP-addresses of some computers in the network. This is one parameter required by network intruders to compromise a secured wireless network.
An attacker may want to replace the IP address of a sender or that of a destination in order to circumvent the IP layer of the OS which usually add received IP addresses to data packets and talk directly with the host machine in the network. An attacker can prevent host X from sending packets to host Y by sending to host X a spoofed packet announcing a window size of zero as if it originated from Y.

Figure 4.12.1 is a collection of several wireless AP’s MAC addresses which can be used for MAC spoofing.

![Network List](image)

**Figure 4.12.1**

### 4.13 NETWORKSHELL (Netsh) USES:

Netsh can be used to reinstall and reset TCP/IP stack when corrupted due to infection by Trojan, Malware and virus. The no connectivity ‘This web page cannot be display’ issue resulting from the inability to connect to a server, load a web page or surf the internet can be resolved using the network shell utility. It can also be due to an installed networking software or security software disinfection of spyware. This is also true for resetting the corrupted window socket. The netshell utility is a command line scripting interface for the configuring and monitoring of windows vista, XP, and 2003.

Firewall connection service can sometime be terminated also. To repair the windows sockets run a netsh
terface by clicking on:

```
C:\Windows\system32>netsh winsock reset
```

Successfull reset the Winsock Catalog.
You must restart the computer in order to complete the reset.

```
C:\Windows\system32>
```
Figure 4.13 (a)
All winsock catalog are reinstalled and returned to default configuration. These include all Layered Service Providers which were causing loss of network packets transmission failure.

To also repair the tcp/ip stacks:

Start Menu> Run> type ‘cmd’ enter> ‘allow’ elevation request> netsh int ip reset logreset.txt

Important registry keys used by tcp/ip stack like these one’s:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters

and

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DHCP\Parameters

are re-written and re-installed.

C:\Windows\system32>netsh int ip reset logreset.txt
Reseting Echo Request, OK!
Reseting Global, OK!
Reseting Interface, OK!
Reseting Unicast Address, OK!
Reseting Route, OK!
A reboot is required to complete this action.

C:\Windows\system32>

Figure 4.13 (b)

Listed below are more uses or applications of NETSH:

- The netsh.exe application file present in window vista can be used to configure http or https.
- Netsh can be used to configure a port with an SSL certificate.
- Used in Namespace Reservation configuration. For example:

  netsh http add urlacl url=http://+:80/MyUri user=DOMAIN\user

- The IP-Listen list configuration.

  netsh http add iplistenn ipaddress=0.0.0.0:8000

- Netsh can be used to switch between two LAN connections.
4.14 TO HIDE SSID:

SSID are a bit difficult to hide successfully because WiFi sniffers have the possibility to detect it from AP’s running in a wireless network. So, hiding a network name is actually not a guaranty a secured AP. But, a good security and encryption on an AP is a good starting point in securing a network. Switching ON/OFF radio broadcast using cron in the AP at scheduled times will help a great deal, since it is difficult to break into a network wirelessly when radio is not on.

4.15 RFMON MODE:

Editing /usr/sbin/wl monitor 1 out of the S60kismet-drone file below made it possible for the kismet client/server desktop to recognize the WRT as a drone and function automatically on service mode whenever powered, ‘monitor’ refers to a situation where the network interface for both desktop and AP come with RFMON capabilities:

S60KISMET-DRONE FILE

#!/bin/sh
echo "Setting radio for kismet_drone"
mkdir /var/log
/sbin/ifconfig eth1 up
/usr/sbin/wl ap 0
/usr/sbin/wl disassoc
/usr/sbin/wl passive 1
/usr/sbin/wl promisc 1

echo "Running kismet_drone"
/usr/sbin./kismet_drone -f /etc/kismet_drone.conf > /dev/null 2>&1 &
sleep 3
echo "kismet_drone now running"

4.16 GPSD (GPS Service Daemon):

The gps service daemon that monitors tcp/ip on port 2947 was not installed and hence no attempt was made to determine the physical location of the access points captured by kismet.
CHAPTER 5

5.0 CONCLUSIONS AND FUTURE WORK

5.1 DISCUSSIONS

I have proposed a wireless network detector, and passive sniffing mechanism using kismet software; an exploit software that passively sniff wireless networks without associating with the network. Kismet requires a set of development packages, libraries and headers called foo-dev to compile and function properly. Kismet requires the kismet client and kismet server binary files configured with specific network interface card type in mind else it would fail to work or scan for wireless access points. I have demonstrated with through wireless scan results and configurations, the possibility of setting up kismet as drone to capture wireless packets from broadcast AP’s in the absence of unsupported RFMON driver network interface card using Linksys wrt54gl as my drone. I have simulated wireless scan results of access points by launching kismet. The following results were demonstrated with kismet scan:

- That kismet can capture the IP-Address of an unsecured computer in the network.
- That kismet can capture the MAC address of any wireless computer within range whether cloaked or uncloaked. It does not matter if the network is secured or not.
- That kismet can detect the service set identifier (SSID) or network name of an unsecured network, and given time will de-cloak a hidden access point network name.

I have observed more closely captured data packets during transmission between networks using wireshark; a network analyser to demonstrate how management data packet frames can be acquired and possibly compromised. The use of wireshark to analyse the tcp dump files has yielded more details which otherwise kismet would not have revealed alone because packet cracking is not part of its functionality.

My scan results also revealed that:

- Within kismet data capture lie the source and destination ip-addresses, source and destination mac addresses of client computers on the network.
- Management and control frames like beacon frames, data frames and probe request frames of kismet captures are stored as dump files in various formats of .xml, .csv, .network, .dump, .cisco capable of being displayed in plain text and easily convertible to other readable formats.

By way of comparison I simulated scan results scanning access points using NETSH a WLAN which comes as default application tool in Windows vista for quick detection of access points within available networks range. My simulated results from various scans (figure 3.9.1 (a)&(b), section 3.9.1, figure 4.11(a)&(b), section 4.5, figure 4.6) were as follows:

- That the WEP column in the kismet GUI application does not necessarily identify WEP based security encryption only. Later versions of wireless encryption protocols like WPA/WPA2 can be detected by kismet. The WEP ‘Yes’ or ‘No’ came about because WEP was the well known good encryption at the time kismet was developed. But kismet is able to identify other protocols within the transport and data layer of the OSI model like http, tcp and udp within the captured packets
- Netsh was able to determine what type of encryption and authentication that a network within range uses. This information is required because there are many exploit packages like AirCrack which can crack WEP keys and for better understanding of the network characteristics.
- It was observed that most secured networks detected by Netsh uses WPA or WPA2 authentication or ‘none’. Most secured networks equally uses TKIP, CCMP, and WEP encryption as well.
- That Kismet and Netsh have some common features like ability to detect client MAC address, radio types, transmission channel.
• That Netsh display signal to noise ratio in percentage % while a network with weak signal strength is received and reported as ‘probe network’ in kismet, a weakness in kismet.
• That kismet capture provides more insight into what is happening in a network. Kismet is able to capture packets that has initialization vector appended to them which are needed in packet analysis.
• That Netsh cannot capture data traffic and hence limited in its use, packet capture provides information about network data packets, such as the transmit time, source, destination, protocol type (TCP, IGMP, HTTP) and data header such as sequence and acknowledgements. This is the major difference between kismet and netsh.
• Kismet drone set up does not channel hop unless enabled by a channel hop script. I have demonstrated that channel hopping was possible with Kismet drone set up through the use of a scanme.sh script. Despite it has the tendency to capture all AP’s on one same channel and remain there. This is a short coming in the kismet drone network interface configuration, as many other access points on other channels may not be covered.
• I have demonstrated that hiding a network name by disabling the SSID does not necessarily make a network secured, kismet and netsh will discover it.

5.2 CONCLUSION

Comparison of scan results and packet analysis resulting from the deployment of kismet and Netsh applications; has thrown more light into the unsecured nature of many neighbourhood access points and how vulnerable they can be if a would-be attacker was to attack the network. A look at data management packets using wireshark and its inherent resources has raised further concern about the vulnerability of many access points. This realization has led to the following research questions:

• **When is a wireless network considered unsecured?**

In this thesis work I have drawn conclusion from the fact that a wireless network without any form of security by way of a good authentication and encryption is unsecured. It was demonstrated that it was not sufficient to hide a network SSID, because kismet and netstumbler are able to sniff out the network name and other inherent parameters contained in network data packets, like the control and management frames and header files.

• **What steps would be required in securing a wireless network?**

Some of the most secured networks were networks that adopted multiple layers of security in encryption and the WPA2 authentication. These multiple layers of encryption make it difficult for a captured data packet which is usually in cipher text and coded, to be decoded or otherwise cracked. Most of the unsecured networks were mostly networks with hidden SSID but with no encryption enabled.
- **Are wireless networks ever secured enough?**

  The answer to this question would be “maybe”. This seem to be the most difficult question to answer because, the security threat posed by hackers and the ubiquitous hacker software challenges security administrators, corporate clients and home users alike. I have discovered through wireless network scans that; open declaration of wireless parameters by kismet, netsh and wireshark is sufficient material for a professional hacker or network security administrator. Because, in the first place these network analysers were developed to help network managers identify problematic areas of the network in the field. The next question would be:

- **Do we need regulation?**

  This question of who should have access to these exploit application software is belated for the simple fact that most of these software are developed with the GNU General Public License free software establishment and they are available to anyone for free. Which explains the genesis of the application software themselves, most of which are individual contributions between developers. This question in my opinion is beyond the scope of this work.

- **What Where the Assumptions Made Against A Potential Attack?**

  My assumption in this work is that MITM (Man-In-The-Middle) attack is possible basically for some reasons like data frames usually do not require authentication in 802.11 networks. The implication of this is that when a data frame is spoofed with a source address, it would not be detected unless the spoofed address is strange and bogus.

  The other assumption is IP-Spoofing, an attacker can gain an unauthorized access into a computer or network by making a malicious message appear as if it came from a trusted machine by spoofing the ip-address of the trusted machine.

  Hacking a computer is not for the average Joe in the street. It takes a professional to be able to spoof management frames, MAC addresses, IP-addresses or headers and pieces together all the needed codes and tools for it to be successful. My work has only revealed such management frames, MAC addresses, IP addresses and tools, and weakness inherent in sniffed management traffic packets.

  I believe that more use could be found for the Netsh application and perhaps more insight into wired/wireless network traffic behaviour and characteristics with better understanding and use of logon script which this thesis did not cover for the obvious reason that I do not have a good knowledge of logon script.
At the beginning of chapter 3 (section 3.0) an introduction to the different types of Linux distributions were enumerated. But the detailed installation partition process is given here for better understanding. It is important that there is sufficient memory space for Swap, because later you won’t be able to carry out software updates if Swap memory size is not at least twice the size of the computer RAM (Recommended). This increase in Swap size does not change the computer RAM size neither its speed. One of Ubuntu Linux weaknesses is that, it crashes when trying to install too many updates in megabytes all at the same time. Unlike windows which can handle unlimited updates more readily.

DOWNLOAD AND INSTALLATION OF UBUNTU 9.04 32-BIT VERSION JAUNTY JACKALOPE

I requested a free desktop Ubuntu 9.04 live installation CD from their website: [http://www.ubuntu.com/getubuntu/download](http://www.ubuntu.com/getubuntu/download)

The new Ubuntu 9.04 interface browser comes with a beautiful user graphical interface that makes it very much like Microsoft Windows internet browser and it is user friendly. As a package It comes with so many software and applications, editors, compilers and utilities, readily available through the synaptic package manager. The update manager makes it possible to run available updates automatically with authentication request without having to compile updates from source. Compiling from source is possible through the user terminal interface, mostly preferred by more experienced UNIX users and which was also adopted in this project. You don’t have to configure the computer monitor screen and its graphical components.

Before running the live installation CD, ensure that your PC is set to ‘boot from CD’ through the administrative privileges (F8 Boot sector). The live CD won’t run if not set to boot from CD. Follow the outlined process:

- Start the computer with the CD inside.
- Choose language of your choice from the language pop-up window
- Select ‘Try Ubuntu without any change to your computer’ to boot into the live CD mode to have the desktop experience.
- From the desktop, select System>Administration>Partition Editor and ‘enter’. (Gparted editor) was pre-installed by default. You can install it from source also by opening a terminal and typing:

  ```
  # Sudo apt-get install gparted
  ```

- Click on the existing Windows partition harddisk space and shrink by dragging the button to a reasonable size of your choice or right click to ‘resize’ and ‘apply’ and exit; 100GB in my own partition, cause I had a 250GB of harddisk. Don’t shrink that much if you are going to be using Windows more.
- Right click on the ‘install’ icon on the desktop,
- Select ‘start up the partitioner’, Select ‘manual’ and continue.
- Click on free space and click ‘new partition’ select ‘logical’ or ‘SWAP-LINUX’ for ‘type’, filesystem partition=32GB, location= ‘Beginning’ and ‘use as swap’ and click ‘ok’
- Create another new partition (Primary Partition) by clicking the remaining free space. Select ‘Primary’ for type, ‘beginning’ for location, use as ext3 or ext4, it does not matter. Right click on the primary partition just created and click ‘Edit’ select ‘/’ as your mount point, mark √ for formatting, ‘save as root’ and then ‘apply’ the pending operation. Below is the result of my partition:
The ntfs file is the windows partition and the HP-Recovery file is the other windows partition. The ext4 file is the primary Linux partition and the Swap the secondary Linux partition. Click ‘forward’ to apply the partition table above.

- Follow the steps and review the chosen settings and click ‘install’
- Once the installation is complete, restart the computer to complete the process and eject the live CD.
SYNAPTIC PACKAGE MANAGER AND SOFTWARE UPDATE MANAGER FEATURES

Launching the synaptic package manager makes it possible for a lot of software packages in the repository inherent in the Ubuntu distro to be installed when needed. The update manager can be launched also to check for updates.

In section 3.2.1 a great deal of emphasis was placed on the computer network interface card and devices. Without the accurate knowledge of what type of NIC your computer has the chances of getting your computer to scan access points using kismet is remote. Going on RFMON is key to deploying kismet, else finding a drone for it that circumvent the limitation that an unsupported network card can invoke. One cannot afford to guess work at this point, so access to the my computer hardware was essential.

KNOWING MY COMPUTER HARDWARE:

```
stephen@stephen-laptop:~$ sudo lshw -C network
*-network
  description: Ethernet interface
  product: MCP65 Ethernet
  vendor: nVidia Corporation
  physical id: 6
  bus info: pci@0000:00:06.0
  logical name: eth0
  version: a3
  serial: 00:1b:24:b5:f2:43
  capacity: 1GB/s
  width: 32 bits
  clock: 66MHz
  capabilities: pm bus_master cap_list ethernet physical mii 10bt 10bt-fd 100bt 100bt-fd 1000bt-fd autonegotiation
    configuration: autonegotiation=on broadcast=yes driver=forcedeth driverversion=0.61 latency=0 link=no maxlatency=20 mingnt=1 module=forcedeth multicast=yes port=MII
*-network
  description: Wireless interface
  product: BCM4328 802.11a/b/g/n
  vendor: Broadcom Corporation
  physical id: 0
  bus info: pci@0000:03:00.0
  logical name: eth1
  version: 03
  serial: 00:1a:73:a7:7f:0e
  width: 64 bits
  clock: 33MHz
  capabilities: pm msi pciexpress bus_master cap_list ethernet physical wireless
    configuration: broadcast=yes driver=wl0 driverversion=5.10.91.9 ip=192.168.0.100 latency=0 module=wl multicast=yes wireless=IEEE 802.11abgn
*-network DISABLED
```
description: Ethernet interface
classical id: 1
logical name: pan0
serial: de:12:46:4d:f5:ea
capabilities: ethernet physical
configuration: broadcast=yes driver=bridge driverversion=2.3 firmware=N/A link=yes
multicast=yes

stephen@stephen-laptop:~$ lspci -nn
00:00.0 RAM memory [0500]: nVidia Corporation MCP65 Memory Controller [10de:0444] (rev a3)
00:01.0 ISA bridge [0601]: nVidia Corporation MCP65 LPC Bridge [10de:0442] (rev a3)
00:01.1 SMBus [0c05]: nVidia Corporation MCP65 SMBus [10de:0446] (rev a1)
00:01.3 Co-processor [0b40]: nVidia Corporation MCP65 SMU [10de:0447] (rev a1)
00:02.0 USB Controller [0c03]: nVidia Corporation MCP65 USB Controller [10de:0454] (rev a3)
00:02.1 USB Controller [0c03]: nVidia Corporation MCP65 USB Controller [10de:0455] (rev a3)
00:06.0 Ethernet controller [0200]: nVidia Corporation MCP65 Ethernet [10de:0450] (rev a3)
00:07.0 Audio device [0403]: nVidia Corporation MCP65 High Definition Audio [10de:044a] (rev a1)
00:08.0 PCI bridge [0604]: nVidia Corporation MCP65 PCI bridge [10de:0449] (rev a1)
00:09.0 IDE interface [0101]: nVidia Corporation MCP65 IDE [10de:0448] (rev a1)
00:0a.0 IDE interface [0101]: nVidia Corporation MCP65 SATA Controller [10de:045d] (rev a3)
00:0b.0 PCI bridge [0604]: nVidia Corporation Device [10de:045b] (rev a1)
00:0c.0 PCI bridge [0604]: nVidia Corporation MCP65 PCI Express bridge [10de:045a] (rev a1)
00:0d.0 PCI bridge [0604]: nVidia Corporation MCP65 PCI Express bridge [10de:0458] (rev a1)
00:0e.0 PCI bridge [0604]: nVidia Corporation MCP65 PCI Express bridge [10de:0459] (rev a1)
00:18.0 Host bridge [0600]: Advanced Micro Devices [AMD] K8 [Athlon64/Opteron] HyperTransport Technology Configuration [1022:1100]
00:18.1 Host bridge [0600]: Advanced Micro Devices [AMD] K8 [Athlon64/Opteron] Address Map [1022:1101]
00:18.2 Host bridge [0600]: Advanced Micro Devices [AMD] K8 [Athlon64/Opteron] DRAM Controller [1022:1102]
00:18.3 Host bridge [0600]: Advanced Micro Devices [AMD] K8 [Athlon64/Opteron] Miscellaneous Control [1022:1103]
03:00.0 Network controller [0280]: Broadcom Corporation BCM4328 802.11a/b/g/n [14e4:4328] (rev 03)
05:00.0 VGA compatible controller [0300]: nVidia Corporation GeForce 8400M GS [10de:0427] (rev a1)
07:05.0 FireWire (IEEE 1394) [0c00]: Ricoh Co Ltd R5C832 IEEE 1394 Controller
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

[1180:0832] (rev 05)
07:05.1 SD Host controller [0805]: Ricoh Co Ltd R5C822 SD/SDIO/MMC/MS/MSPro Host Adapter [1180:0822] (rev 22)
07:05.2 System peripheral [0880]: Ricoh Co Ltd R5C843 MMC Host Controller [1180:0843] (rev 12)
07:05.3 System peripheral [0880]: Ricoh Co Ltd R5C592 Memory Stick Bus Host Adapter [1180:0592] (rev 12)
07:05.4 System peripheral [0880]: Ricoh Co Ltd xD-Picture Card Controller [1180:0852] (rev ff)

stephen@stephen-laptop:~$ lspci -vvn | grep 14e4

03:00.0 Network controller [0280]: Broadcom Corporation BCM4328 802.11a/b/g/n [14e4:4328] (rev 03)

stephen@stephen-laptop:~$ lsusb

Bus 001 Device 003: ID 064e:a101 Suyin Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 006: ID 04b8:0841 Seiko Epson Corp.
Bus 002 Device 003: ID 093a:2510 Pixart Imaging, Inc.
Bus 002 Device 004: ID 08ff:2580 AuthenTec, Inc. AES2501 Fingerprint Sensor
Bus 002 Device 002: ID 03f0:171d Hewlett-Packard Wireless (Bluetooth + WLAN) Interface [Integrated Module]
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub

stephen@stephen-laptop:~$ uname –a

Linux stephen-laptop 2.6.28-15-generic #49-Ubuntu SMP Tue Aug 18 18:40:08 UTC 2009 i686 GNU/Linux

stephen@stephen-laptop:~$ dmesg | grep ound

[    0.000000] found SMP MP-table at [c00f8640] 000f8640
[    0.443379] ACPI: No dock devices found.
[    0.502931] pnp 00:03: can’t evaluate _CRS: 12298<6>pnp: PnP ACPI: found 12 devices
[    1.147666] pcieport-driver 0000:00:0b.0: found MSI capability
[    2.043534] isapnp: No Plug & Play device found
[    4.312118] hub 1-0:1.0: USB hub found
[    4.370083] hub 2-0:1.0: USB hub found
[    4.421499] device-mapper: multipath round-robin: version 1.0.0 loaded
[    4.423224] powernow-k8: Found 1 AMD Turion(tm) 64 X2 Mobile Technology TL-60 processors (2 cpu cores) (version 2.20.00)
[    4.423652] BIOS EDD facility v0.16 2004-Jun-25, 0 devices found
[    10.995757] sdhci-pci 0000:07:05.1: SDHCI controller found [1180:0822] (rev 22)
stephen@stephen-laptop:~$ dmesg | grep b43

stephen@stephen-laptop:~$ iwconfig

lo  no wireless extensions.

eth0  no wireless extensions.

eth1  IEEE 802.11 Nickname:"
    Access Point: Not-Associated
    Link Quality:5 Signal level:240 Noise level:166
    Rx invalid nwid:0 invalid crypt:177 invalid misc:0

pan0  no wireless extensions.

stephen@stephen-laptop:~$ su
Password:
root@stephen-laptop:/home/stephen# lspci

00:00.0 RAM memory: nVidia Corporation MCP65 Memory Controller (rev a3)
00:01.0 ISA bridge: nVidia Corporation MCP65 LPC Bridge (rev a3)
00:01.1 SMBus: nVidia Corporation MCP65 SMBus (rev a1)
00:01.3 Co-processor: nVidia Corporation MCP65 SMU (rev a1)
00:02.0 USB Controller: nVidia Corporation MCP65 USB Controller (rev a3)
00:02.1 USB Controller: nVidia Corporation MCP65 USB Controller (rev a3)
00:06.0 Ethernet controller: nVidia Corporation MCP65 Ethernet (rev a3)
00:07.0 Audio device: nVidia Corporation MCP65 High Definition Audio (rev a1)
00:08.0 PCI bridge: nVidia Corporation MCP65 PCI bridge (rev a1)
00:09.0 IDE interface: nVidia Corporation MCP65 IDE (rev a1)
00:0a.0 IDE interface: nVidia Corporation MCP65 SATA Controller (rev a3)
00:0b.0 PCI bridge: nVidia Corporation Device 045b (rev a1)
00:0c.0 PCI bridge: nVidia Corporation MCP65 PCI Express bridge (rev a1)
00:18.0 Host bridge: Advanced Micro Devices [AMD] K8 [Athlon64/Opteron] HyperTransport Technology Configuration
00:18.1 Host bridge: Advanced Micro Devices [AMD] K8 [Athlon64/Opteron] Address Map
00:18.2 Host bridge: Advanced Micro Devices [AMD] K8 [Athlon64/Opteron] DRAM Controller
00:18.3 Host bridge: Advanced Micro Devices [AMD] K8 [Athlon64/Opteron] Miscellaneous Control
03:00.0 Network controller: Broadcom Corporation BCM4328 802.11a/b/g/n (rev 03)
05:00.0 VGA compatible controller: nVidia Corporation GeForce 8400M GS (rev a1)
07:05.0 FireWire (IEEE 1394): Ricoh Co Ltd R5C832 IEEE 1394 Controller (rev 05)
07:05.1 SD Host controller: Ricoh Co Ltd R5C822 SD/SDIO/MMC/MS/MSPro Host Adapter (rev 22)
07:05.2 System peripheral: Ricoh Co Ltd R5C843 MMC Host Controller (rev 12)
07:05.3 System peripheral: Ricoh Co Ltd R5C592 Memory Stick Bus Host Adapter (rev 12)
07:05.4 System peripheral: Ricoh Co Ltd xD-Picture Card Controller (rev ff)


```
root@stephen-laptop:/home/stephen# dmesg | grep Broadcom

```

### APPENDIX B: Index/Whiterussian/RC/bin/

Table 3.3.1 index of /whiterussian/rc4/bin/

---

<table>
<thead>
<tr>
<th>Path</th>
<th>Size</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>../md5sums</td>
<td>990</td>
<td>24-Nov-2005 02:50</td>
</tr>
<tr>
<td>openwrt-brcm-2.4-jffs2-4MB.trx</td>
<td>2162688</td>
<td>24-Nov-2005 02:43</td>
</tr>
<tr>
<td>openwrt-brcm-2.4-jffs2-8MB.trx</td>
<td>2228224</td>
<td>24-Nov-2005 02:43</td>
</tr>
<tr>
<td>openwrt-brcm-2.4-squashfs.trx</td>
<td>1544192</td>
<td>24-Nov-2005 02:43</td>
</tr>
<tr>
<td>openwrt-motorola-jffs2-4MB.bin</td>
<td>2162696</td>
<td>24-Nov-2005 02:43</td>
</tr>
<tr>
<td>openwrt-motorola-jffs2-8MB.bin</td>
<td>2228232</td>
<td>24-Nov-2005 02:43</td>
</tr>
<tr>
<td>openwrt-motorola-squashfs.bin</td>
<td>1544200</td>
<td>24-Nov-2005 02:43</td>
</tr>
<tr>
<td>openwrt-wap54g-squashfs.trx</td>
<td>1544193</td>
<td>24-Nov-2005 02:43</td>
</tr>
<tr>
<td>openwrt-wrt54g-jffs2.bin</td>
<td>2163712</td>
<td>24-Nov-2005 02:43</td>
</tr>
<tr>
<td>openwrt-wrt54g-jffs2.bin</td>
<td>1545216</td>
<td>24-Nov-2005 02:43</td>
</tr>
<tr>
<td>openwrt-wrt54g-squashfs.bin</td>
<td>2163712</td>
<td>24-Nov-2005 02:43</td>
</tr>
<tr>
<td>openwrt-wrt54g3g-jffs2.bin</td>
<td>1545216</td>
<td>24-Nov-2005 02:43</td>
</tr>
<tr>
<td>openwrt-wrt54g3g-squashfs.bin</td>
<td>2229248</td>
<td>24-Nov-2005 02:43</td>
</tr>
<tr>
<td>openwrt-wrt54gs-squashfs.bin</td>
<td>1545216</td>
<td>24-Nov-2005 02:43</td>
</tr>
<tr>
<td>openwrt-wrt54gs_v4-jffs2.bin</td>
<td>2163712</td>
<td>24-Nov-2005 02:43</td>
</tr>
<tr>
<td>openwrt-wrt54gs_v4-squashfs.bin</td>
<td>1545216</td>
<td>24-Nov-2005 02:43</td>
</tr>
</tbody>
</table>
APPENDIX B1

Transmission Range:

- Indoor: 300 ~ 500 Meter
- Outdoor: 12~ 15 KM

Technical Specification:

Frequency Range 2300 ¾ 2500 MHz
Impedance 50 W
VSWR (or Return Loss) < 1.5:1 (or > 14dB)
Gain >17dBi
Polarization Horizontal, Linear
X-Pol < 20dB
Front to Back Ratio > 20dB
Max. Power Input 20W
Size 32x2.5x2"
Radom Color gold
Temperature Range -45 to +75 Degrees
Storage Temperature -30 to +75 Degrees

USB WIFI CARD SPECIFICATIONS:

<table>
<thead>
<tr>
<th>Standard</th>
<th>IEEE 802.11b/g</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date Transmission Rate</td>
<td>Up to 54Mbps</td>
</tr>
<tr>
<td>Interface</td>
<td>USB 2.0 compliant with USB 1.1</td>
</tr>
<tr>
<td>Advanced security features</td>
<td>WEP, WPA and WPA2</td>
</tr>
<tr>
<td>Frequency Band</td>
<td>2.400-2.483 GHz</td>
</tr>
<tr>
<td>Antennas</td>
<td>Embedded 17dBi Yagi antenna</td>
</tr>
<tr>
<td>Number of channels</td>
<td>11 Channels (North America), 13 (Europe), 14(Japan)</td>
</tr>
<tr>
<td>Receiver-Sensitivity</td>
<td>-80dBm@11Mbps</td>
</tr>
<tr>
<td>Transmission Power</td>
<td>20dBm+17dBi=37 dBm</td>
</tr>
<tr>
<td>OS Compatible</td>
<td>Windows 98SE, 2000, XP and Vista, MAC OS 10.3x/10.4x and Linux</td>
</tr>
<tr>
<td>Chipset</td>
<td>RTL8187L</td>
</tr>
<tr>
<td>Security</td>
<td>64/128 bits encryption</td>
</tr>
</tbody>
</table>

APPENDIX B2
Internal Specification:

- USB 802.11N 150M WIRELESS LAN Adapter with Detachable Antenna(RT3070 Chipset)
- Support 64/128-bit WEP, complies with 128 bit WPA standard(TKIP/AES),
- Supports MIC, IV Expansion, Shared Key Authentication, IEEE 802.1X
- supports 150/54/48/36/24/12/9/6/11/5.5/2/1 Mbps wireless LAN data transfer rates
- Quick Secure Setup, complies with WPS for worry free wireless connections.
- Supports Windows 2000/XP/Vista/7, Linux, Mac OS
- With Detachable Antenna + extra 7dBi High gain antenna
- Max transmit range: Indoor :100m, Outdoor: 300m
Section 3.3 looks at the tools that makes the Linksys wrt54gl works as mini Linux OS. Not all the full functionalities of a Linux distribution package is contained therein, but sufficiently enough to enable the router as UNIX based and enough to hold a kismet MIPS package.

Below is a list of the Linksys wrt54gl repositories after the flash with OpenWrt and kismet drone repositories after kismet MIPS package download. The *bin* folder and the *etc* folders of the WRT contains a relative concise amount of useful tools like the scp, ssh, Telnet, killall, wget but to name a few.

```
stephen@stephen-laptop:~$ su
Password:

root@stephen-laptop:/home/stephen#

root@OpenWrt:/etc# cd ..
root@OpenWrt:/# ls
bin  dev  etc  jffs  lib  mnt  proc  rom  sbin  tmp  usr  var  www

root@OpenWrt:~# cd /tmp
root@OpenWrt:~# pwd
/tmp
root@OpenWrt:~# ls
dhcp.leases  resolv.conf  run
log  resolv.conf.auto  spool

root@OpenWrt:~# cd /usr/bin
root@OpenWrt:/usr/bin# ls
[          dropbearkey  killall5  scp
traceroute
arping  du          kismet_drone  seq  uniq
awk    env          length  sort  uptime
basename  expr  logger  ssh  wc
bstrip  find  md5sum  strings  webif-
page
bunzip2  free  mesg   tail  wget
bzcat  haserl  mkfifo  tee  which
clear  head  nc  telnet  xargs
crontab  hexdump  nslookup  test  yes
cut  hostid  passwd  time
dbclient  id  printf  top
dirname  killall  reset  tr

root@OpenWrt:/usr/bin# cd /etc
root@OpenWrt:/etc# ls
banner  group  modules
protocols
```
<table>
<thead>
<tr>
<th>config</th>
<th>hosts</th>
<th>modules.d</th>
</tr>
</thead>
<tbody>
<tr>
<td>rc.common</td>
<td></td>
<td></td>
</tr>
<tr>
<td>crontabs</td>
<td>hotplug.d</td>
<td></td>
</tr>
<tr>
<td>resolv.conf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dnsmasq.conf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>shells</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dropbear</td>
<td>init.d</td>
<td></td>
</tr>
<tr>
<td>sysctl.conf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ethers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>firewall.user</td>
<td></td>
<td></td>
</tr>
<tr>
<td>functions.sh</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>inittab</td>
<td>ppp</td>
</tr>
<tr>
<td></td>
<td>ipkg.conf</td>
<td>preinit</td>
</tr>
<tr>
<td></td>
<td>kismet_drone.conf</td>
<td>profile</td>
</tr>
</tbody>
</table>
APPENDIX C 2: Kismet Libraries/Packages

In section 3.7 a step-by-step installation and configuration process of kismet drone and kismet client/server was carried out. Below is the kismet library package and some applications that find useful functionality with kismet. Wireshark was used in this work to analysis sniffed data packets. SOX and FESTIVAL are two audio application that works with kismet to announce the detection or arrival of a new network at capture. GPSD was Not used at all, this application combined with kismet is very good for pin point the location of an access point as well as rogue access points.

root@stephen-laptop:/home/stephen# lsmoc | grep eth1
root@stephen-laptop:/home/stephen# lsmoc | grep bcm4828
root@stephen-laptop:/home/stephen# apt-cache policy kismet

kismet:
Installed: 2008-05-R1-4build1
Candidate: 2008-05-R1-4build1
Version table:
*** 2008-05-R1-4build1 0
   500 http://se.archive.ubuntu.com jaunty/universe Packages
   100 /var/lib/dpkg/status

root@stephen-laptop:/home/stephen# apt-cache depends kismet

kismet
Depends: libc6
Depends: libexpat1
Depends: libgcc1
Depends: libgmp3c2
Depends: libgomp1
Depends: libmagickcore1
Depends: libncurses5
Depends: libpcap0.8
Depends: libstdc++6
Depends: zlib1g
Depends: wireless-tools
Depends: wireshark-common
Suggests: wget
Suggests: sox
Suggests: festival
Suggests: gpsd
Suggests: gsfons
Suggests: libwww-perl
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

APPENDIX C 3: Section 3.7 Kismet Download and Installation/Configuration and Launch

Section 3.7 gives the detail download and installation processes as well as results

**KISMET DOWNLOAD AND INSTALLATIONS:**

Open the terminal as root user in Ubuntu and download the kismet binaries from [www.kismetwireless.net](http://www.kismetwireless.net) into the tmp folder.

```
stephen@stephen-laptop:~$ su
Password: root@stephen-laptop:/home/stephen#
wget http://www.kismetwireless.net/code/kismet-2006-04-R1-wrt54.tar.gz
```

Untar/gzip the kismet package:

```
tar –zxvf http://www.kismetwireless.net/code/kismet-2006-04-R1-wrt54.tar.gz
```

The kismet file is automatically downloaded to kismet folder in the etc directory also.

```
stephen@stephen-laptop:~$ cd /tmp
stephen@stephen-laptop:/tmp$ wget http://www.kismetwireless.net/code/kismet-2006-04-R1-wrt54.tar.gz
--2009-10-16 22:28:37--
http://www.kismetwireless.net/code/kismet-2006-04-R1-wrt54.tar.gz
Resolving www.kismetwireless.net... 204.244.237.54
Connecting to www.kismetwireless.net|204.244.237.54|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 578699 (565K) [application/x-gzip]
Saving to: `kismet-2006-04-R1-wrt54.tar.gz'

100%[======================================>] 578,699
23.3K/s in 20s

2009-10-16 22:28:57 (28.2 KB/s) - `kismet-2006-04-R1-wrt54.tar.gz' saved [578699/578699]
```

```
stephen@stephen-laptop:/tmp$ cd /tmp/kismet
stephen@stephen-laptop:/tmp/kismet$ tar –zxvf kismet-2006-04-R1-wrt54.tar.gz
kismet-2006-04-R1-wrt54/
kismet-2006-04-R1-wrt54/kismet_drone
kismet-2006-04-R1-wrt54/conf/
kismet-2006-04-R1-wrt54/conf/kismet.conf
kismet-2006-04-R1-wrt54/conf/kismet_drone.conf
```
To make all the kismet files executable:

```
stephen@stephen-laptop:/tmp/kismet$ -rw-r--r-- 1 root root 1347
```

```
root@stephen-laptop:/home/stephen# cd /etc/kismet
root@stephen-laptop:/etc/kismet# ls
ap_manuf  client_manuf  kismet.conf  kismet_drone.conf
kismet_ui.conf
```

Figure 3.7
Light green coloured files indicate executable files.

**KISMET DRONE INSTALLATION AND CONFIGURATION**

From desktop terminal 'ssh' into the router as root user:

```
stephen@stephen-laptop:$ su
Password:
root@stephen-laptop:/home/stephen#
```

```
root@stephen-laptop:/home/stephen# ssh 192.168.1.1
The authenticity of host '192.168.1.1' (192.168.1.1)' can't be established.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.1' (RSA) to the list of known hosts.
root@192.168.1.1's password:
```
```
BusyBox v1.00 (2007.01.30-11:42+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.
```
```
```

WHITE RUSSIAN (0.9) ----------------------------------
* 2 oz Vodka  Mix the Vodka and Kahlua together
* 1 oz Kahlua  over ice, then float the cream or
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

* 1/2oz cream milk on the top.

---------------

root@OpenWrt:~#

By changing directory into the temporary file ‘tmp’ in the router and from the router with enough memory space inside, using wget to download the kismet_drone binaries from www.kismetwireless.net

cd /tmp
wget http://www.kismetwireless.net/code/kismet-2006-04-R1-wrt54.tar.gz

root@OpenWrt:/etc# cd /tmp
wget http://www.kismetwireless.net/code/kismet-2006-04-R1-wrt54.tar.gz
Connecting to www.kismetwireless.net[204.244.237.54]:80
kismet-2006-04-R1-wr 100% |*---------------------------------------------------|
565 KB 00:00 ETA
root@OpenWrt:~# cd /tmp
root@OpenWrt:~# ls

dhcp.leases            resolv.conf.auto
kismet-2006-04-R1-wr54.tar.gz run
log

Untar/gzip the kismet package:

tar –zxvf http://www.kismetwireless.net/code/kismet-2006-04-R1-wrt54.tar.gz
root@OpenWrt:~# tar –zxvf kis*
kismet-2006-04-R1-wrt54
kismet-2006-04-R1-wrt54/kismet_drone
kismet-2006-04-R1-wrt54/conf
kismet-2006-04-R1-wrt54/conf/kismet.conf
kismet-2006-04-R1-wrt54/conf/kismet_drone.conf
kismet-2006-04-R1-wrt54/GPL
kismet-2006-04-R1-wrt54/README
kismet-2006-04-R1-wrt54/kismet_server
kismet-2006-04-R1-wrt54/CHANGELOG

root@OpenWrt:~# ls

dhcp.leases            resolv.conf
kismet-2006-04-R1-wrt54 resolv.conf.auto
kismet-2006-04-R1-wrt54.tar.gz run
log
spool

root@OpenWrt:~# cd kismet-2006-04-R1-wrt54
root@OpenWrt:~/kismet-2006-04-R1-wrt54# ls
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

Update the ipkg files:

root@OpenWrt:/kismet-2006-04-R1-wrt54/conf# ipkg update
root@OpenWrt:/kismet-2006-04-R1-wrt54/conf# ipkg install wl

Make the kismet.conf files executable:

Chmod –R 777

root@OpenWrt:/kismet-2006-04-R1-wrt54/conf# pwd
/tmp/kismet-2006-04-R1-wrt54/conf
root@OpenWrt:/kismet-2006-04-R1-wrt54/conf# ls -l
-rwrxr-x-- 1 1000 100 13375 Apr 13 2006 kismet.conf
-rwrxr-x-- 1 1000 100 4105 Jan 1 01:37 kismet_drone.conf

cd into kismet-2006-04-R1-wrt54/conf:

root@OpenWrt:/kismet-2006-04-R1-wrt54/conf# pwd
/tmp/kismet-2006-04-R1-wrt54/conf

Make the kismet.conf and kismet_drone.conf files executable:

root@OpenWrt:/kismet-2006-04-R1-wrt54/conf# ls -l
-rwrxr-x-- 1 1000 100 13375 Apr 13 2006 kismet.conf
-rwrxr-x-- 1 1000 100 4105 Jan 1 01:37 kismet_drone.conf

root@OpenWrt:/kismet-2006-04-R1-wrt54/conf# cd ..
root@OpenWrt:/kismet-2006-04-R1-wrt54# ls

CHANGELOG README kismet_drone
GPL conf kismet_server
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

Secure Copy kismet_drone into the /usr/bin directory in the wrt:

root@OpenWrt:~/kismet
root@OpenWrt:~/kismet
root@OpenWrt:~/kismet

check if the files in the /usr/bin are executable:

copy the kismet_drone.conf file into the etc directory:

update ipkg and install wl driver:

root@OpenWrt:~/kismet
root@OpenWrt:~/kismet
KISMET CLIENT AND KISMET SERVER INSTALLATION AND CONFIGURATION

Navigate to the desktop kismet client/server:

```
root@OpenWrt:/# ssh 192.168.1.107
```

```
root@stephen-laptop:~#
```

**Install ascii:**

```
root@stephen-laptop:~# apt-get install ascii
```

**EDITING THE KISMET.CONF FOR THE KISMET CLIENT/SERVER DESKTOP:**

```
root@stephen-laptop:~# cd /etc
```

```
root@stephen-laptop:/etc# ls kis*
```

```
kismet_drone.conf
kismet:
ap_manuf  client_manuf  kismet.conf  kismet_drone.conf
kismet_ui.conf
```

```
root@stephen-laptop:/etc# cd ..
```

```
root@stephen-laptop:/# cd /etc/kismet
```

**Using vi editor to edit the kismet.conf file:**

```
root@stephen-laptop:/etc/kismet# vi kismet.conf
```

setup the kismet.conf file in the desktop to:

```
source=kismet_drone,192.168.1.107:3501,drone
```

```
root@stephen-laptop:/etc/kismet# ls
```

```
ap_manuf  client_manuf  kismet.conf  kismet_drone.conf
kismet_ui.conf
```

**EDITING THE KISMET_DRONE.CONF FILES FOR THE ROUTER:**

Change the ‘source’ and the ‘allowedhosts’ to:

```
Source=wrt54g,prism0,wrt54g
```

```
Allowedhosts=127.0.0.1,192.168.0.0/24
```

Secure copy /tmp/kismet-2006-04-R1-wrt54/kismet_drone into /usr/bin/kismet_drone and /tmp/kismet-006-04-R1-wrt54/kismet_drone.conf into /etc/kismet_drone.conf. Ensure they are executable.
Evaluating Kismet and NetStumbler as Network Security Tools & Solutions

$ -rw-r--r-- 1 root root

root@OpenWrt:~/kismet-2006-04-R1-wrt54/conf# scp kismet_drone.conf /etc/kismet_drone.conf

root@OpenWrt:~/kismet-2006-04-R1-wrt54# scp kismet_drone/usr/bin/kismet_drone

root@OpenWrt:~/kismet-2006-04-R1-wrt54/conf# pwd
/tmp/kismet-2006-04-R1-wrt54/conf

root@OpenWrt:~/kismet-2006-04-R1-wrt54/conf# ls -l
-rw-r-xr-x 1 1000 100 13375 Apr 13 2006 kismet.conf
-rw-r-xr-x 1 1000 100  4105 Jan  1 01:37 kismet_drone.conf

root@OpenWrt:~/kismet-2006-04-R1-wrt54/conf# cd ..

root@OpenWrt:~/kismet-2006-04-R1-wrt54/conf# ls

kismet.conf  kismet_drone.conf  kismet_server

root@OpenWrt:~/kismet-2006-04-R1-wrt54/conf# cp kismet_drone/usr/bin/kismet_drone

root@OpenWrt:~/kismet-2006-04-R1-wrt54# cd /usr/bin

root@OpenWrt:/usr/bin# ls

lrwxrwxrwx 1 root root
### APPENDIX D: Section 4.3 Management Frames

The 802.11 protocols operating at the layer 2 level of the OSI model are composed of three different Wireless frame types:

- Management Frames
- Control Frames
- Data Frames

#### Management frames (Mx):

<table>
<thead>
<tr>
<th>Kismet Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ma</td>
<td>Association request</td>
</tr>
<tr>
<td>MA</td>
<td>Association response</td>
</tr>
<tr>
<td>Mr</td>
<td>Reassociation request</td>
</tr>
<tr>
<td>MR</td>
<td>Reassociation response</td>
</tr>
<tr>
<td>Mp</td>
<td>Probe request</td>
</tr>
<tr>
<td>MP</td>
<td>Probe response</td>
</tr>
<tr>
<td>MB</td>
<td>Beacon</td>
</tr>
<tr>
<td>MM</td>
<td>ATIM</td>
</tr>
<tr>
<td>MD</td>
<td>Disassociation</td>
</tr>
<tr>
<td>Mt</td>
<td>Authentication</td>
</tr>
<tr>
<td>MT</td>
<td>Deauthentication</td>
</tr>
<tr>
<td>M?</td>
<td>Unknown management frame</td>
</tr>
</tbody>
</table>

#### Physical (Control) frames

<table>
<thead>
<tr>
<th>Kismet Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pt</td>
<td>Request to send</td>
</tr>
<tr>
<td>PT</td>
<td>Clear to send</td>
</tr>
<tr>
<td>PA</td>
<td>Data Ack</td>
</tr>
<tr>
<td>Pc</td>
<td>CF End</td>
</tr>
<tr>
<td>PC</td>
<td>CF End+Ack</td>
</tr>
<tr>
<td>P?</td>
<td>Unknown phy frame</td>
</tr>
</tbody>
</table>

#### Data frames:

<table>
<thead>
<tr>
<th>Kismet Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DD</td>
<td>Data frame</td>
</tr>
<tr>
<td>Dc</td>
<td>Data+CF+ACK</td>
</tr>
<tr>
<td>Dp</td>
<td>Data+CF+Poll</td>
</tr>
<tr>
<td>DP</td>
<td>Data+CF+ACK+Poll</td>
</tr>
<tr>
<td>DN</td>
<td>Data Null</td>
</tr>
<tr>
<td>Da</td>
<td>CF Ack</td>
</tr>
<tr>
<td>PA</td>
<td>CF Ack+Poll</td>
</tr>
<tr>
<td>D?</td>
<td>Unknown data frame</td>
</tr>
</tbody>
</table>
### APPENDIX E: Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACK:</td>
<td>Acknowledgement</td>
</tr>
<tr>
<td>AES:</td>
<td>Advanced Encryption Standard</td>
</tr>
<tr>
<td>ARP:</td>
<td>Address Resolution Protocol</td>
</tr>
<tr>
<td>CCMP:</td>
<td>Counter Mode CBC-MAC protocol</td>
</tr>
<tr>
<td>CRC:</td>
<td>Cyclic Redundancy Check</td>
</tr>
<tr>
<td>CSV:</td>
<td>Commas Separated Values</td>
</tr>
<tr>
<td>DES:</td>
<td>Data Encryption Standard</td>
</tr>
<tr>
<td>DHCP:</td>
<td>Dynamic Host Configuration Protocol</td>
</tr>
<tr>
<td>EAP:</td>
<td>Extensible Authentication Protocol</td>
</tr>
<tr>
<td>GPSD:</td>
<td>Global Positioning System (Service Daemon)</td>
</tr>
<tr>
<td>IDS:</td>
<td>Intrusion Detection System</td>
</tr>
<tr>
<td>IV:</td>
<td>Initialization Vector</td>
</tr>
<tr>
<td>MITM:</td>
<td>Man-In-The-Middle</td>
</tr>
<tr>
<td>NETSH:</td>
<td>Network Shell</td>
</tr>
<tr>
<td>NETSTUMBLER:</td>
<td>Network Stumbler</td>
</tr>
<tr>
<td>OSI:</td>
<td>Open System Interconnection model</td>
</tr>
<tr>
<td>QoS:</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RFMON:</td>
<td>Radio Frequency Monitoring</td>
</tr>
<tr>
<td>SSDP:</td>
<td>Simple Service Discovery Protocol</td>
</tr>
<tr>
<td>SNR:</td>
<td>Signal-Noise-Ratio</td>
</tr>
<tr>
<td>SSID:</td>
<td>Service Set Identifier</td>
</tr>
<tr>
<td>TKIP:</td>
<td>Temporary Key Integrity Protocol</td>
</tr>
<tr>
<td>UDP:</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>WDP:</td>
<td>Wireless Datagram Protocol</td>
</tr>
<tr>
<td>WEP:</td>
<td>Wired Equivalent Protocol</td>
</tr>
<tr>
<td>WiFi:</td>
<td>Wireless Fidelity</td>
</tr>
<tr>
<td>WPAN:</td>
<td>Wireless Personal Area Network</td>
</tr>
</tbody>
</table>
REFERENCES

[12] Index of /whiterussian/rc4/bin
[18] www.kismetwireless.net