Attacks Tools

Goals and Processes

- Different attackers have different goals.
- For example, a disgruntled employee or an employee who has been terminated is most likely to launch a DoS attack on the company network.
- Other attackers are more focused on stealing personal information to facilitate identity theft, which is a confidentiality attack.
- The goals of the attackers are closely aligned with the three main security properties:
    - Availability
    - Confidentiality
    - Integrity
- An attacker will prioritize one of the above depending on their motivation and the underlying value of the information available on the target network.
- The process that an attacker would follow is very similar to the basic process that the military would use in planning an operation against an opponent:
    - Reconnaissance – collect as much information on the target as possible
    - Planning – how to get there and what to do when we get there
    - Execution – Launch the attack
- This process is an iterative one in the sense that at any point in operation, additional information may be required that necessitates a return to an earlier phase.

Reconnaissance and Sniffing Tools

- We will now examine some of the most commonly available tools that can be used very effectively for reconnaissance and sniffing activities.
- The three most popular tools used are:
    - Wireshark
    - NetStumbler
    - Kismet
    - Aircrack

Wireshark

- Wireshark (formally known as Ethereal) has long been used to troubleshoot and analyze wired networks but as we shall see, it can be used to analyze 802.11 frames as well.

- It is available for both the Linux and Windows platforms and the complete packages can be downloaded from: http://www.wireshark.org

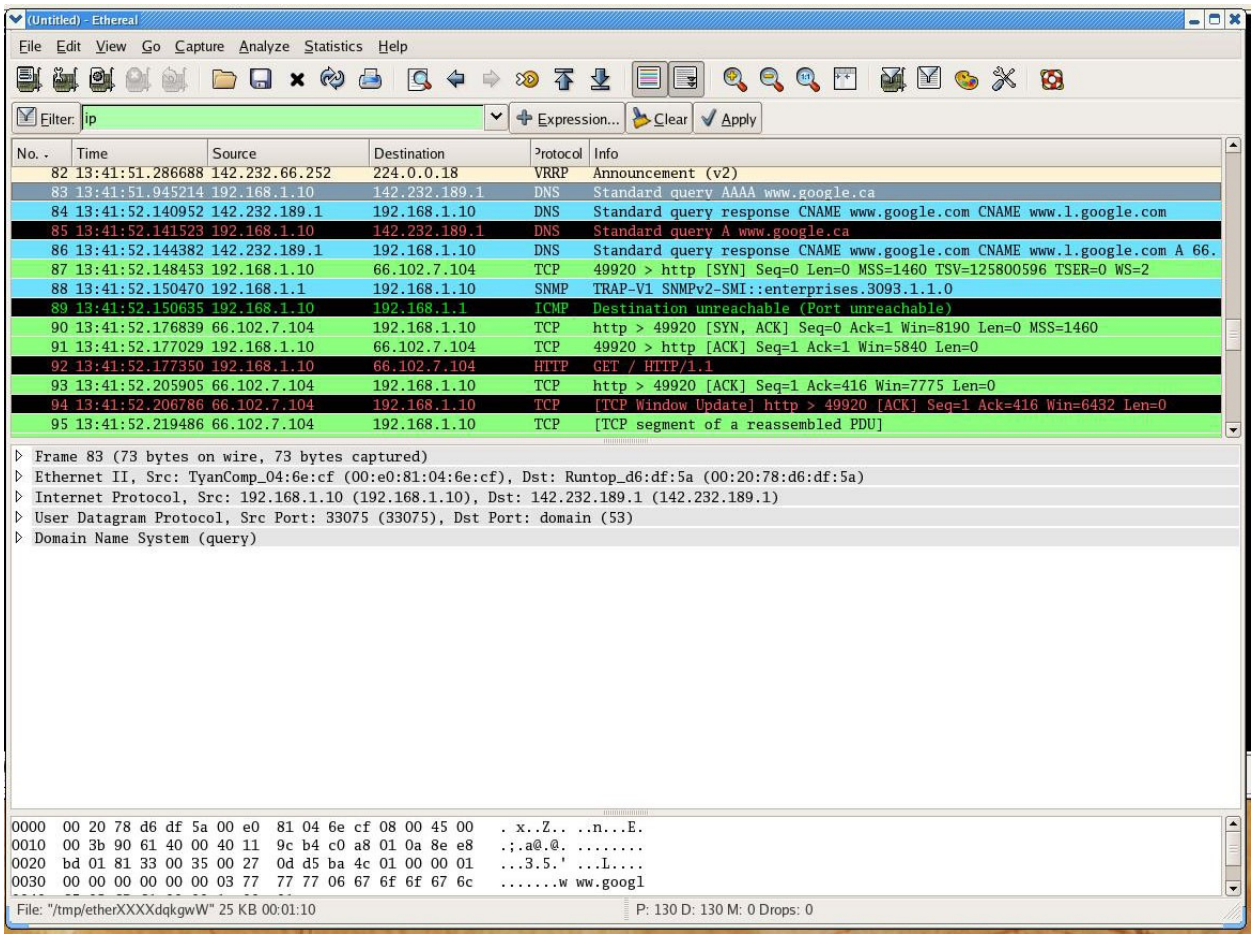
- Wireshark has a multitude of features; among the most commonly used are:

    - **Capture** live packets from a network interface.
    - Display detailed **protocol** for each captured packet.
    - **Open** and **Save** captured packet data.
    - **Import and Export** packet data from other capture programs.
    - **Filter packets** on many criteria.
    - **Search** for packets based on packet header or payload criteria.
    - **Colorize** packet display based on filters.
    - Create various **statistics**.

- The first step is to download and install the package and there is more than adequate documentation available at the afore mentioned site.

- Wireshark also requires a **packet capture library**. For Linux systems this is called "**libpcap**" and "**winpcap**" for Windows. Both must be installed in order for Wireshark to work.

- The following screen shot illustrates the basic layout for Wireshark:



- As mentioned earlier, Wireshark will capture packets from a specified network interface. However it will not capture and display packets below layer 2, which includes 802.11 frames.

- Thus, using Wireshark on a WiLAN's interface may only capture **user data** packets with "fake" Ethernet headers.

- You will not see any 802.11 management or control packets at all and the 802.11 packet headers are "translated" by the network driver to "fake" Ethernet packet headers.

- So in order to capture all traffic that the adapter can receive, the adapter must be put into "**monitor mode**", sometimes called "**rfmon** mode".

- In this mode, the driver will not make the adapter a member of **any** service set, so it won't support sending any traffic and will only supply received packets to a packet capture mechanism, not to the networking stack.

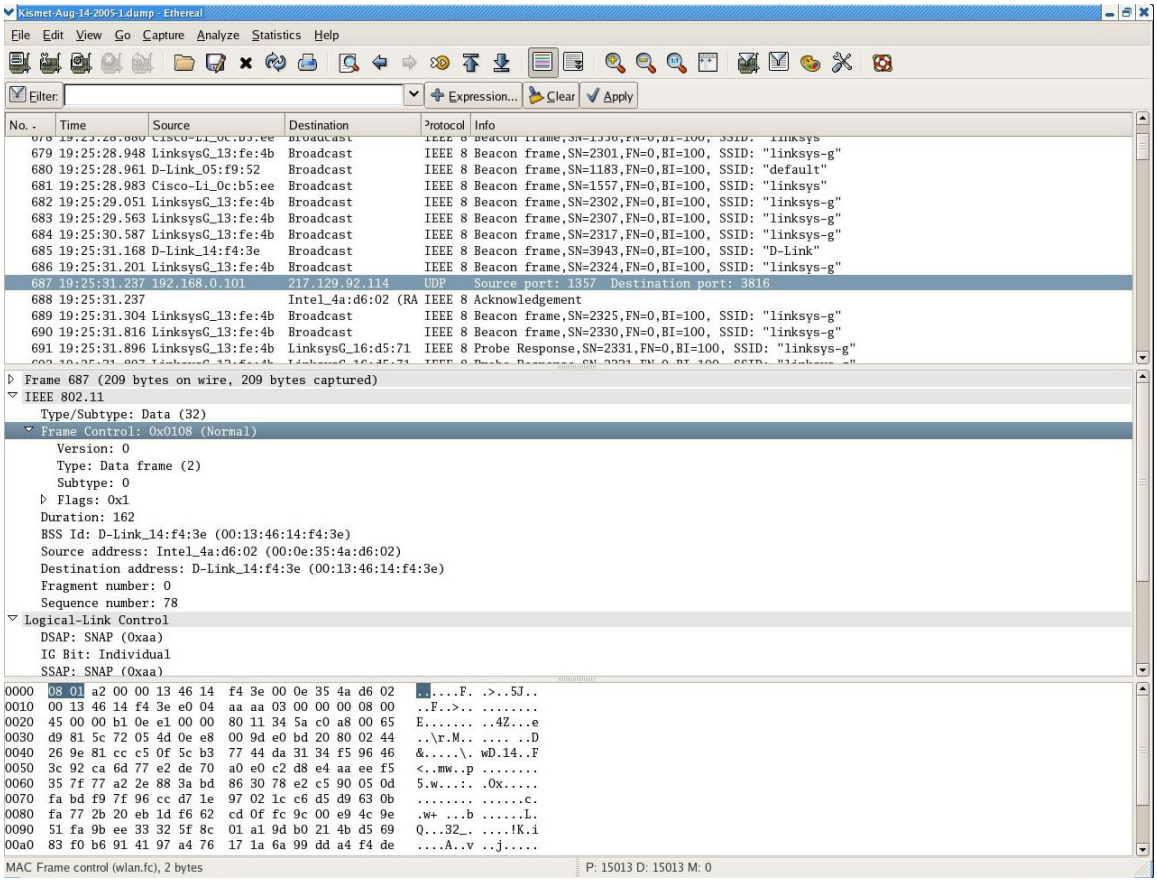
- Monitor mode is not supported by WinPcap, and thus not by Wireshark on Windows. It is supported, for at least some interfaces, on some versions of Linux, FreeBSD, and NetBSD.

- However, Wireshark can open dump files saved by other tools that incorporate additional 802.11 data, such as Kismet.

- Current versions of Wireshark will display all 802.11 frame data that these passive monitoring tools can capture.

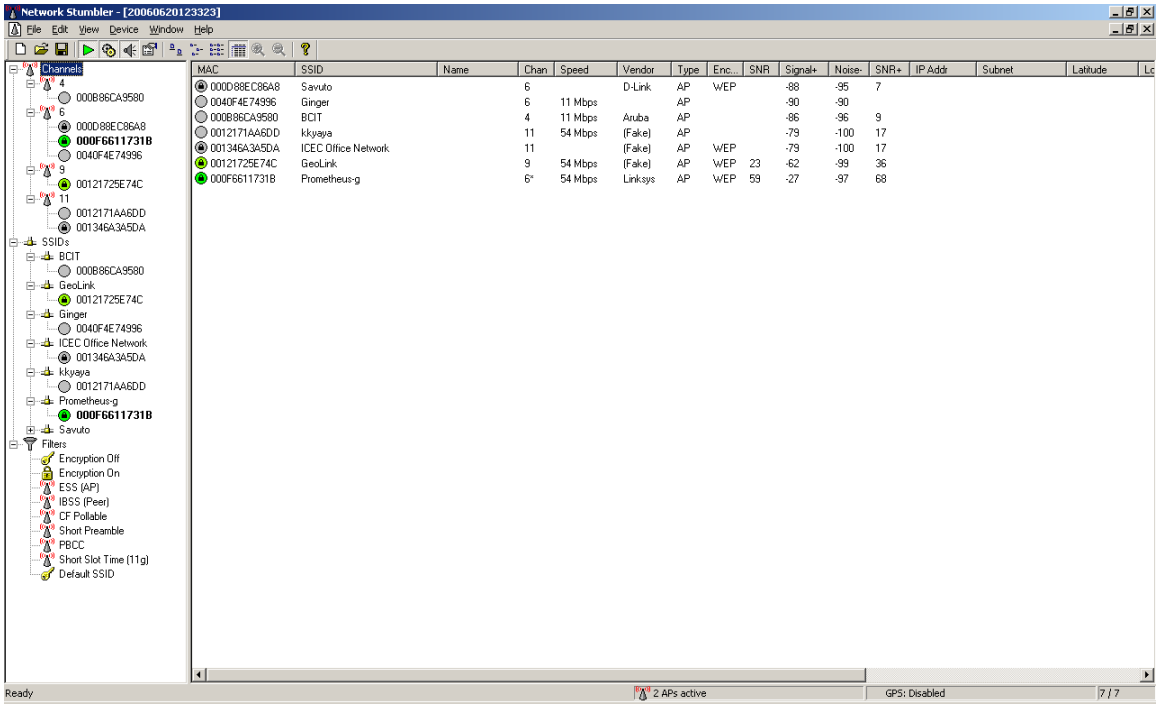
- Kismet (to be discussed later) captures 802.11 frames from all nearby APs and dumps them (in **pcap** format) to external files.

- The following diagram illustrates a frame that was captured using Kismet and then analyzed with Wireshark :



NetStumbler

- NetStumbler is a powerful and freely available WiLAN discovery tool that is useful for site surveys, detecting rogue access points, and finding and mapping WiLAN installations.
- The tool is available only for Windows. It can be downloaded at: http://netstumbler.com
- It includes various features such as signal strength, ESSID, channel, and GPS support. In conjunction with a GPS, you can map out a WiLAN coverage in an area simply by driving around and measuring the signal strengths.
- NetStumbler can output a summary of latitude/longitude coordinates which can then be displayed as points on a map. Also available is a script that will convert the capture files into a format that can be read by Microsoft Map Point.
- NetStumbler functions by emitting 802.11b probes that ask wireless access points to respond if they are nearby.
- This also means that the presence of NetStumbler can be detected by software such as Kismet.
- Wireless access points are configured by default to respond to these probes, but this option may be turned off to thwart wardrivers from detecting your wireless access point, otherwise known as “cloaking”
- NetStumbler displays most of the information needed in one screen, broken down into two panes.
- The left pane provides shortcuts for displaying the networks in just about any fashion.
- The right-hand pane is the main display area which displays all the networks.
- The following screen shot illustrates the display:



Kismet

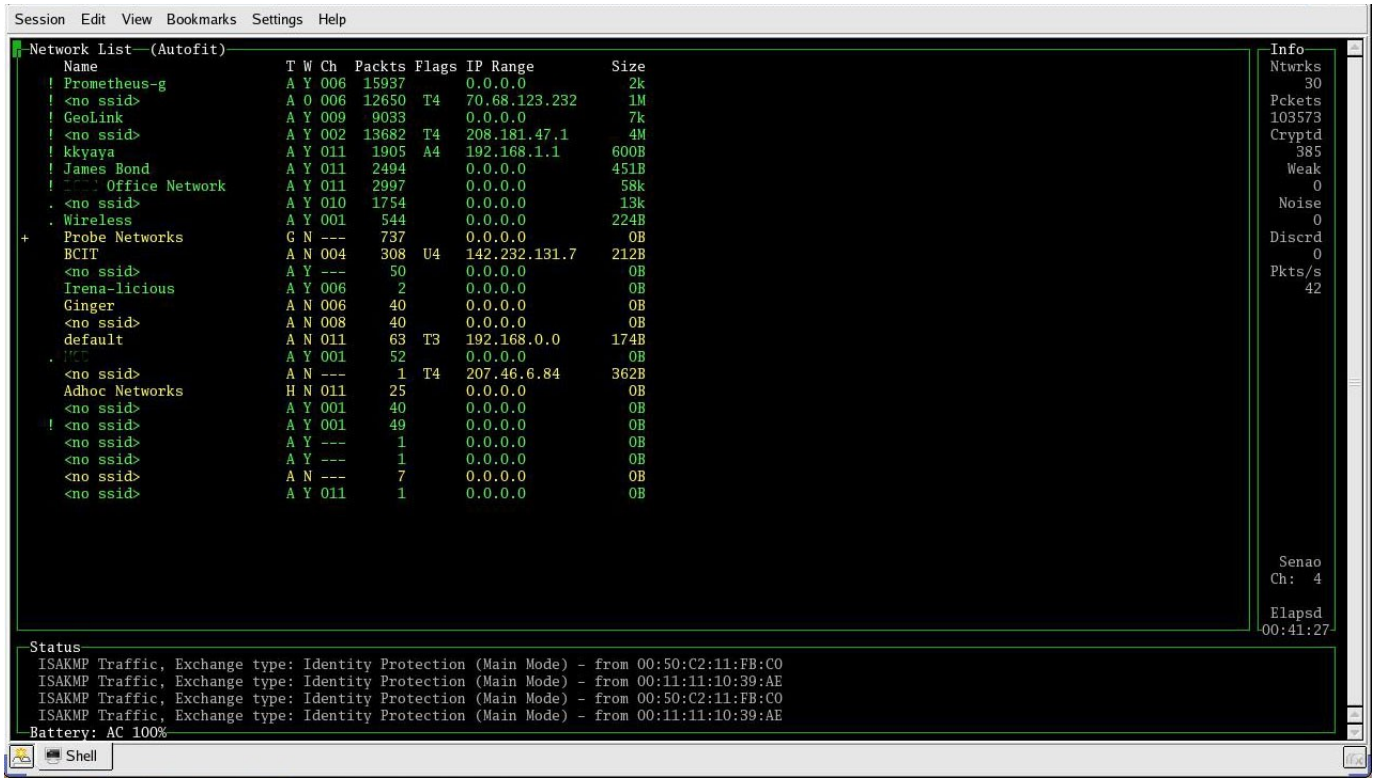
- Kismet is an 802.11 layer2 wireless network detector, sniffer, and Intrusion Detection System (IDS).
- Kismet will work with any wireless card which supports raw monitoring (rfmon) mode, and can sniff 802.11b, 802.11a, and 802.11g traffic.
- Kismet identifies networks by passively collecting packets and detecting standard named networks, detecting (and given time, uncloaking) hidden networks, and inferring the presence of non-beaconing networks via data traffic.
- Kismet is a completely passive tool in that it does not transmit any probe requests.
- It is designed to run on the Linux and OpenBSD platforms.
- It is a superset of all of the functionality provided by NetStumbler and more.
- Using Kismet we can determine the network configuration for any network within reach, with or without WEP enabled.

- Some common applications for Kismet are:
    - **Wardriving**: Mobile detection of wireless networks, logging and mapping of network location, WEP, 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, possibly combined with a layer3 IDS like Snort.
    - **Rogue AP Detection**: Stationary or mobile sniffers to enforce site policy against rogue access points.

- Kismet saves the collected information in a set of files that can be viewed and analyzed later.

- The files contain lists of all the information about a network, raw packet dumps, and captured WEP traffic so it can be fed into other WEP cracking tools such as AirSnort.

- The main Kismet window is shown below:

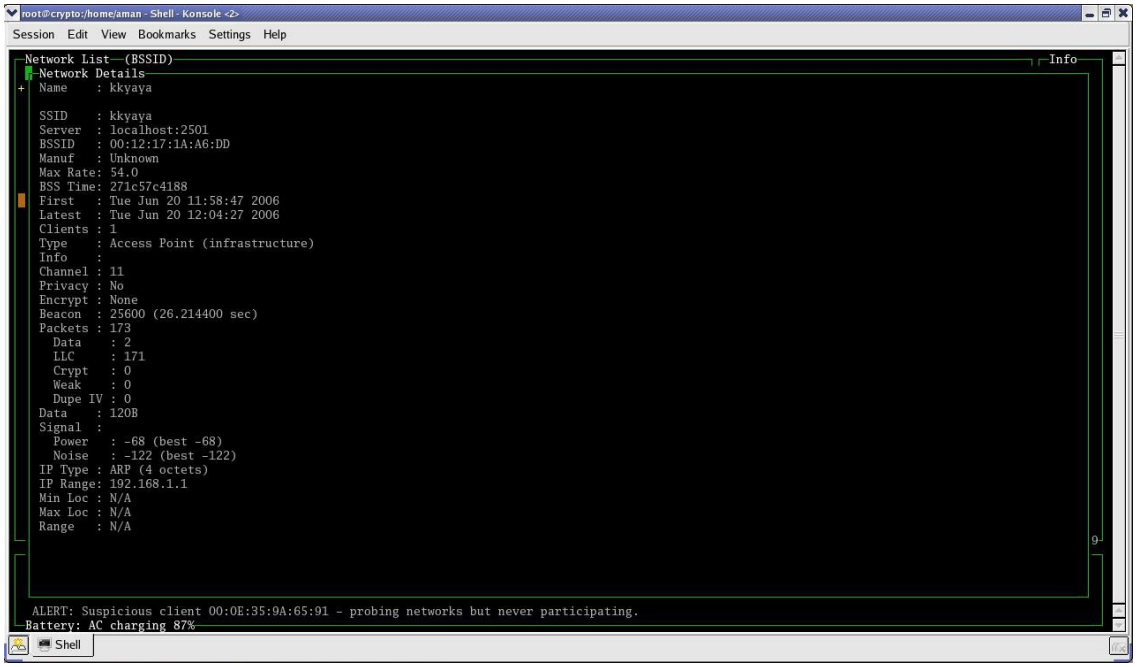


- The right-hand panel is a short informational window that displays:
    - The number of networks found (Ntwrks)
    - Number of packets (Pckets)
    - Number of encrypted packets (Cryptd)
    - Number of packets with a weak IV (Weak)
    - Corrupted packets (Noise)
    - Number of packets discarded due to bad CRC (ICV) values (Discrd)
    - Packet rate (Pkts/s)
    - Total elapsed time (Elapsd)

- The lower panel lists all of the status messages as they occur.

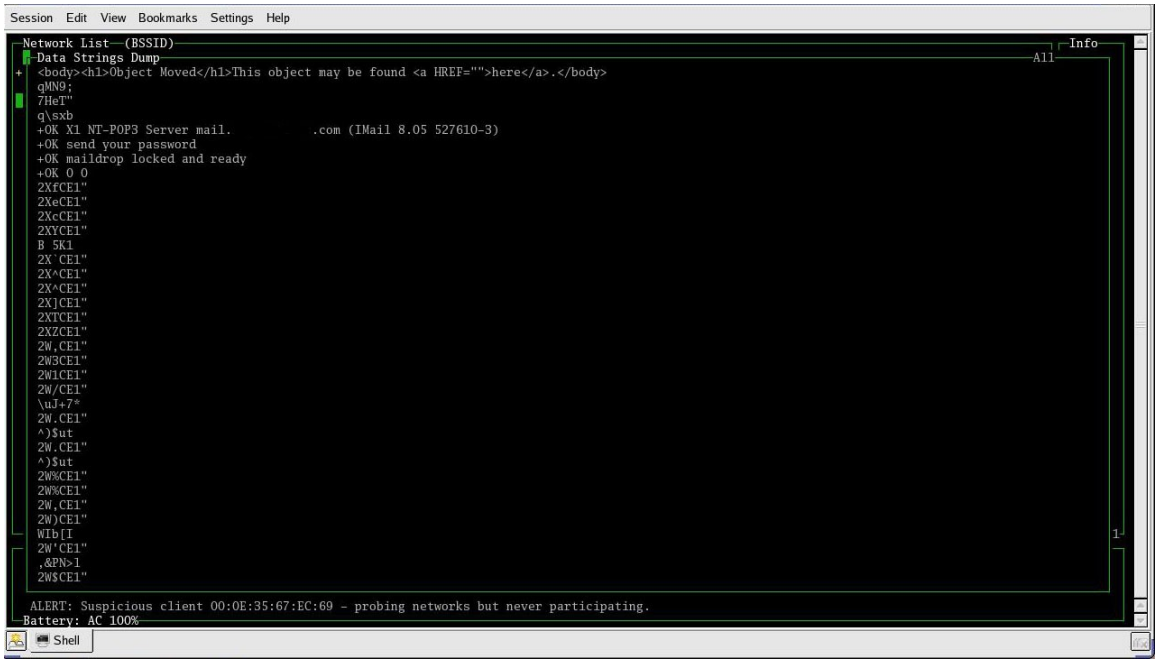
- The main window provides detailed information on all of the networks detected. Some of the important fields are:
    - SSID
    - (!) indicates activity seen in the last 6 seconds
    - (.) indicates activity seen in the last 3 seconds
    - "T" indicates the type of network identified
        - "A" – AP in infrastructure mode
        - "D" – data-only host or station
        - "H" – ad-hoc network master
    - "W" – whether or not WEP is used
    - "Ch" – channel being used by the network
    - "Packts" – Number of packets seen
    - "IP Range" – Range of IP addresses used by the network
    - "Flags" – Indicates how the IP range was determined:
        - "A" – using ARP packets
        - "U" – using UDP packets
        - "T" – using TCP packets
        - "D" – using DHCP packets

- In addition to the main window, Kismet provides several other popup displays that provide detailed information on each network. The following is an example of this:



- Kismet provides a great deal more information about a network than NetStumbler. It can even provide a real-time dump of the ASCII strings that it captures on the selected network. This is activated using the “d” key in the main window.

- The following is an example of the ACSII string display window:



Encryption (WEP) Cracking

- There are several WEP cracking tools available both as open-source and commercial products.

- The amount of time it takes to hack an encrypted wireless network is dependent on the amount of traffic. Less traffic means the crack will take more time and vise versa.

AirSnort

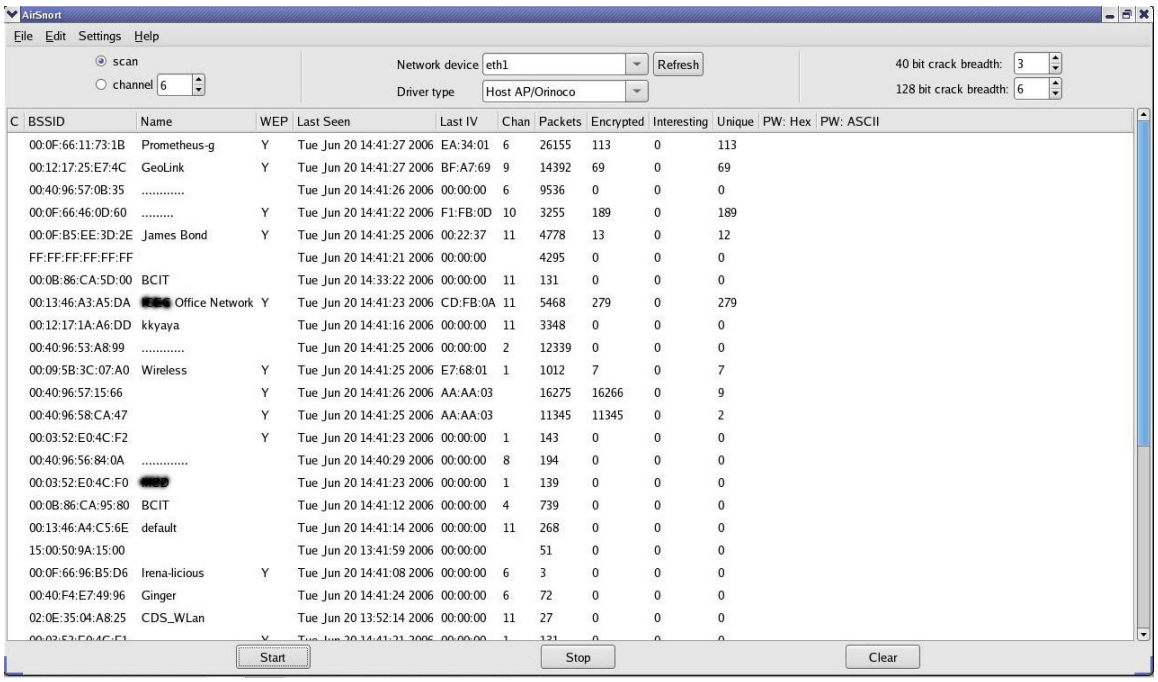
- AirSnort has been around for sometime now but continues to be one of the best tools around for cracking WEP. It is available for both the Linux and Windows platforms.

- AirSnort is a wireless LAN (WLAN) tool which recovers encryption keys. AirSnort operates by passively monitoring transmissions, computing the encryption key when enough packets have been gathered.

- By exploiting the WEP weaknesses discussed earlier AirSnort is able to capture encrypted frames and extract the secret key, byte by byte.

- Given about 3,000,000 to 5,000,000 packets, the application can crack the key in a few seconds.

- The following screen shot shows the basic AirSnort GUI and the information available:



- The **File** menu allows you to load and save crack files for cracking over multiple sessions.
- As long as the WEP key has not change AirSnort will just pick up where it left off.
- Other important options include:
    - **Network device** is just the same as the [interface] option above (e.g. eth1 for the Prism card).
    - **Driver type:** Both the Prism and Orinoco cards will work with the Host AP/ Orinoco option currently selected.
    - The **scan** option activates channel scanning during wireless discovery.
    - The **channel** option is used to lock on to one specific channel. This speeds up the data collection process.
- AirSnort attempts to crack the captured packets for both a 40 bit and 128 bit key each time 10 new weak IVs are seen for a given access point.
- It uses a probabilistic attack, so, the best guess may not be the right one. With limited captured data and enough CPU power, more exhaustive searches can be performed.
- The search for a key involves a depth first traversal of an n-ary tree. The depth of tree is 5 for 40 bit key attempts and 13 for 128 bit key attempts.
- The breadth of the trees is governed by the 40 and 128 bit crack depth fields in the AirSnort GUI. A breadth parameter of 'n' instructs AirSnort to try the n most likely values at each key position using statistics derived from the IVs that have been collected.
- Large breadth setting can result in very slow processing time for crack attempts default values of 3 for 40 bit cracks and 2 for 128 bit cracks are recommended for starters.
- If a large number of weak IVs have been gathered (> 1500 if a 40 bit key is suspected, > 3000 if a 128 bit key is suspected), then increasing the breadth values can be considered.
- The number of interesting packets needed to perform a successful crack depends on two things; luck and key length; key length is the only important factor.
- For a key length of 128 bits, this translates to about 1500 packets. For other key lengths, assume 115 packets per byte of the key.
- Some keys are more resistant to this technique than others and may require far more packets.
- If a large number of packets have been captured and the key has not been extracted yet, either wait for more packets or try a larger breadth.

- One published test operating against a saturated 802.11b network took 262839 unique packets and 13 minutes to crack a 128 bit WEP key.

Save and Restore

- AirSnort saves data in two formats. All packets captured by AirSnort can be saved in pcap dump file format by selecting the "Log to file" option from the File menu.
- This must be done before a capture session is initiated. AirSnort can also save a much smaller amount of data of data about a capture session in the form of "crack" files.
- These files represent the minimum amount of data that AirSnort maintains for each access point that it discovers.
- Crack files contain summary data of those packets that AirSnort has seen that actually use weak IVs.
- Restoration of data from a crack file will only display statistics about packets that use weak IVs, thus packet counts are likely to be much smaller than seen during the live capture.

Aircrack-ng

- This has become the tool of choice for wireless sniffing and WEP and WPA cracking.
- It can be downloaded from: http://www.aircrack-ng.org/
- Aircrack-ng is an 802.11 WEP and WPA-PSK keys cracking program that can recover keys once enough data packets have been captured.
- It implements the standard FMS attack along with some optimizations like KoreK attacks, as well as the all-new PTW attack, thus making the attack much faster compared to other WEP cracking tools.
- In fact, Aircrack-ng is a set of tools for auditing wireless networks:

Name	Description
aircrack-ng	Cracks WEP (Brute-force search) and WPA (Dictionary File) keys
airdecap-ng	Decrypts WEP or WPA encrypted capture files with known key
airmon-ng	Placing different cards in monitor mode
aireplay-ng	Packet injector (Linux, and Windows [with Commview drivers])
airodump-ng	Packet sniffer: Places air traffic into PCAP or IVS files and shows information about networks
airtun-ng	Virtual tunnel interface creator
airolib-ng	Stores and manages ESSID and password lists; Increases the KPS of WPA attacks
packetforge-ng	Create encrypted packets for injection
Tools	Tools to merge and convert

- Aircrack implements three types of attacks:

  1. Brute force attack
  2. Dictionary attack
  3. Statistical attack

- By exploiting several security weaknesses of the WEP protocol Aircrack NG makes use of a statistical method to recover WEP keys.

- The success of the attack however is dependent upon collecting a sufficient number of IVs (= Initialization Vectors).

- Depending on the length of the encryption key, determining the actual WEP key will take less than a minute on a common PC.

- In fact a recent demo was done using Aircrack on the iPhone:

  http://www.youtube.com/watch?v=4R69KLYC7bg