Kismet, Aircrack-ng, and Karmetasploit

WIRELESS SECURITY

How to find, map, crack, and impersonate wireless networks. **BY KURT SEIFRIED**

Perhaps I'm one of the last few holdouts, staying wired to the Internet instead of computing wirelessly at home (like my friends, parents, etc.). Everyone seems to be getting laptops and $40 access points, which are way easier and cheaper than running Ethernet for most people. But, after reading this, you might want to keep your network wired, too.

Finding Wireless Networks

To see what I'm so worried about, the first step is to find some wireless networks. One of the best tools for this is Kismet, which comes in most distributions. Many distros ship with an old (2008) version, so to get started, download Kismet [1], unpack it, run the configure script, and make and install it. Note that Kismet needs root access to run because it talks directly to hardware, so you can run it either as root or with *sudo*, or you can install Kismet with suid root and add users to the *kismet* group. Please note that any user in this group will be able to fiddle with your network interfaces, so be careful.

```
cd /directory/kismet-source/
./config
make
make dep
make install
```

Kismet has three main components: the drone, the server, and the client. The drone captures network traffic and sends it to the server (which can be running on the same or a remote system). The server collects and collates the data, and the client connects to the server and provides a text-based interface to the data in real time. This allows you to take multiple systems, including wireless access points running custom firmware (like the WRT54G), and feed them all into a single server.

To run Kismet, simply start the *kismet_client*, which gives you the option of starting the server and collecting data. This in turn creates several files, including GPS data (mapping networks to physical locations if you have GPS on your system), network data (a list of networks and clients found, what channel they are on, and all the configuration details you could imagine), and a PCAP network capture file.

One thing you will notice is that, if you run Kismet with only one capture source (i.e., one wireless card), it will have to hop from channel to channel to cover all 11 (or 12 or 13, depending on where you are) channels. Although this detects all the available networks, it does make for fragmented capture files because you'll get a little network traffic from one, then some from another. Fortunately, the solution is easy – buy some wireless adapters (USB works really well) and add more capture interfaces. The sweet spot seems to be four sources. This allows you to dedicate one card to each major channel (1, 6, and 11) and leave one to hop the remaining channels, thereby ensuring that you find all the networks and maximize the amount of data you capture (Figure 1). To configure, simply add these lines to your *kismet.conf* file:

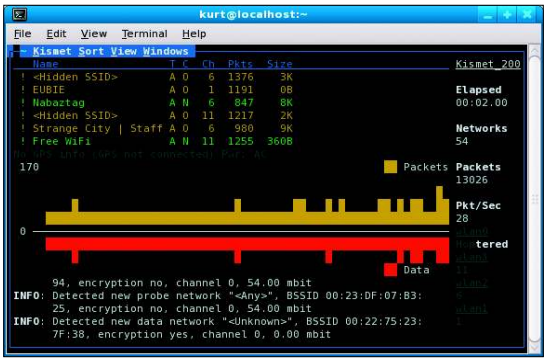

Figure 1: The Kismet client with four capture sources.

```
channellist=hoplist:2,3,4,5,7,8,9,10
ncsource=wlan0:hop=false,channel=1
ncsource=wlan1:hop=false,channel=6
ncsource=wlan2:hop=false,channel=11
ncsource=wlan3:hop=true,flfl
   channellist=hoplist
```

I went to a local coffee shop with Kismet, and within a few minutes, I discovered more than 40 networks. A second attempt at another coffee shop down the street (but on the second floor with a better line of sight to other buildings) netted more than 70 networks. On average, half of the networks had no encryp-

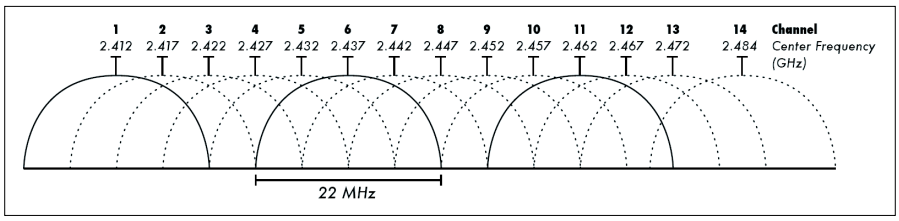

Figure 2: Channel frequency diagram [10] (reproduced under the CC-BY-SA license [11]).

tion. Many were pay-for-access hot spots, and quite a few of the networks had only one client, most likely an individual's home network. What I found most interesting was the ability to capture the MAC addresses of clients on pay networks, most of which filter on the basis of MAC address once you have authenticated. So if you know which MAC addresses to spoof, you can get yourself free network access.

Getting Past Encryption

The good news about getting past encryption is that about half the time you won't have to. When you do need access to an encrypted network, however, keep in mind that the wireless encryption standards WEP and WPA are quite weak. Most distributions ship with Aircrack-ng [2], a WEP and WPA key cracking program. To use it, just run the *airoscript* program, which will give you a text-based interface. If you're feeling lazy, choose the *auto* option and it will scan, select, and attack a network for you.

Attacking Wireless Clients

People tend to focus on securing their wireless infrastructure (encryption, access controls, etc.) and tend to forget about clients. If you are within wireless range, you can pretend to be a legitimate wireless access point and convince clients to connect to you. Then, you can connect to the real access point, proxying and modifying their traffic on the fly. Such an attack is known as a "rogue access point." The tool for doing this is Karmetasploit (formerly Karma, now merged with Metasploit). Instructions for downloading and installing Metasploit are in an earlier article [3].

Once you have installed Metasploit and Aircrack-ng, simply run the *airbase-ng* program and run a DHCP server and attach it to the wireless interface so that clients can get network configuration information. Then run Metasploit with the server modules – to execute man-in-the-middle attacks – and browser autopwn (automatically attack) modules – to inject hostile content into web pages (see the Karmetasploit documentation [4]) – or you can set up a transparent web proxy and have some fun [5].

Protecting Yourself

Even cautious users can't be certain of security. Most pay-to-access wireless networks do not include encryption because the provider would have to share the password with everyone in advance. That means an attacker could easily get a copy of it and decrypt traffic anyway. Even if a provider has a properly secured SSL-encrypted payment gateway, there's nothing to prevent anyone from watching your traffic or sniffing passwords, for example. Encryption of all your network traffic will provide such protection, as I covered in my "Secret Tunnels" article [6]. If you don't have a server to run your VPN traffic through, you might want to try the IPREDator VPN service [7]. IPREDator provides a PPTP-encrypted [8] VPN connection for EUR 5 a month, tunneling all your traffic to Sweden, where strict privacy laws should prevent access to it. ■

Wireless Network Channels

Although you can send data on up to 13 wireless channels (11 in North America, 12 in Japan, 13 in most of the rest of the world) [9], the channels overlap sightly, meaning only three of them (1, 6, and 11) are separate (Figure 2). If someone is broadcasting on channel 1 and someone else is on channel 2, they will be sharing a certain amount of frequency, which can lead to collisions and other issues that can reduce the amount of available bandwidth. This means that the majority of wireless networks will be on channels 1, 6, or 11.

Wireless Encryption

Even if a wireless network has strong encryption, the password used to secure it must be shared with all the wireless clients. This means that anyone who buys access to the network gets a copy of the password. In a large network, there is a good chance that the password is leaked publicly. (The one coffee shop I have been to with an encrypted network has the password printed on a large sign behind the cash register, and it is never changed.) This means that you must ensure that your network traffic is protected by encryption and that you are connecting to legitimate servers and not some man-in-the-middle server, such as a Karmetasploit module.

INFO

[1] Kismet: *http://www.kismetwireless.net*

[2] Aircrack-ng: *http://www.aircrack-ng.org/*

[3] "Metasploit" by Kurt Seifried, *Linux Pro Magazine*, November 2008, pg. 62. *http://www.linuxpromagazine.com/Issues/2008/96/METASPLOIT*

[4] KARMA + Metasploit 3 == Karmetasploit: *http://trac.metasploit.com/wiki/Karmetasploit*

[5] Upside-Down-Ternet: *http://www.ex-parrot.com/pete/upside-down-ternet.html*

[6] "Secret Tunnels" by Kurt Seifried, *Linux Pro Magazine*, July 2009, pg. 64. *http://www.linuxpromagazine.com/Issues/2009/104/SECRET-TUNNELS*

[7] IPREDator: *https://www.ipredator.se/*

[8] "Close and Secret" by James Stanger, *Linux Pro Magazine*, December 2008, pg. 22. *http://www.linuxpromagazine.com/Issues/2008/97/CLOSE-AND-SECRET*

[9] Wireless LAN channel list: *http://en.wikipedia.org/wiki/List_of_WLAN_channels*

[10] Wireless Networking in the Developing World: *http://en.wikipedia.org/wiki/File:2.4_GHz_Wi-Fi_channels_%28802.11b,g_WLAN%29.png*

[11] Creative Commons Attribution and ShareAlike License: *http://commons.wikimedia.org/wiki/Commons:Reusing_content_outside_Wikimedia*

THE AUTHOR

Kurt Seifried is an Information Security Consultant specializing in Linux and networks since 1996. He often wonders how it is that technology works on a large scale but often fails on a small scale.