

Forum:[Hack Forums](#)

Thread: [Another Wireless hacking tutorial](#)

bladedemon - November 10, 2014, 2:31 pm

Greetings to everyone.

I am bladedemon, recently registered. [/introductions]

This is my very first attempt to write a tutorial, and after a quick search on the site, I haven't found something that meets my expectations. So, in my own view of the open source idea, "if there isn't one, make it!" - not applicable to everything, mind you :)

Note: I posted this on another site [sinister dot ly , can't post it now] [as of now 10-11-2014 it's still down] under the same alias.

What I'll try to explain is WIRELESS HACKING. Ooops, not "hacking", more like "intruding" or "messaging"

Disclaimers [yea, they are a bunch]:

- Use this responsibly.
[illegal actions and blah blah blah]
- I use linux for hacking [debian-based in particular], hence no Windows alternatives here. Sorry folks, I left the Windows' side a long time ago.
- This is NOT a guide for skiddies, you need to have some knowledge of linux and TCP/IP if you are to understand. I will try to explain most things to the best of my abilities, but you know, RTFM and google that shit.
- I will write some examples. This tutorial has some field work behind it, so that [hopefully] I know what I am talking about

Sooooo, we begin with... why, installing the tools of course!

Note: while I have used Kali [and before that, Backtrack] a lot, I do believe it is best to manually install the tools you need, on your own, to your linux system. While it can be somewhat of a pain to update them, having a bunch of tools that you don't know how they work can be frustrating or even dangerous. Not that you should forsake the pentest distros, mind you. I have kept a Windows XP, a Windows 7 and a Kali Virtual Machine in my computer. It can be quite convenient at times. But maybe in a later tutorial?

Building the tools you need from source was the old, manual way. Now, with the package managers this procedure is automated, but still:

- You can't find security tools easily on the mainstream distros
- The ones you find are usually outdated. Sometimes it doesn't matter [when it comes to projects that are no longer maintained, yet still useful - more on that on a later tutorial]

- Living on the bleeding edge has its advantages and disadvantages.

So, to start, you will need to download the tools from the websites themselves, or synchronize with an online developer's version. While downloading the latest stable versions from the project's website is surely safer [don't forget to check the checksum!] you can also synchronize from them [using either git - package git - or svn -package subversion] and even make a little script to automate updating. Looks coolish shit when run from a terminal, I'll tell you that :P

So, after you downloaded the tools, you need to extract [they come in compressed form] and compile them. For that, you must have installed the required programs and libraries first [also known as dependencies]. If you don't, the compilation will just stop after an error of "Couldn't find this" or "Unknown that". Now, you can try to google each and every dependency of a tool, manually install it, and its dependencies, and its dependencies' dependencies etc., but it IS way too much of a pain, and if you mess with the libraries and repositories a lot it can all go south [been there, done that, that was a bitch of a situation]. So, what do you do instead?

In the "Software sources" ["software-properties-gtk" for a GUI, or in the /etc/apt/sources.list file for apt-get users, the rest just google your way through] you need to enable the "source code" repositories [they are usually disabled by default on the mainstream distros]. After that, IF the tool you want to compile is in the repository, type:

Code `sudo apt-get build-dep <name of the tool's package>` That will install the dependencies needed to compile the tool. If the tool is NOT in the repository, just hit "<tool name> dependencies" at google, it should give you what you need, but for the tools I will be using here, just an apt-get build-dep is enough.

Those tools usually are packed in a way that, with the aid of another tool, they can be compiled and installed with just 3 commands. That tool is, most times, "make" [package "make"]. Also, that packages have a special script that checks if all the necessary dependencies are installed, and what optional dependencies are also installed. The script's name is usually "configure" and is executable, so typing

Code `./configure` should get things started. Notice two things:

- Executing configure script doesn't require root privileges
- Executing it is mandatory, when it exists. More on that later.

So, after executing the configure script, some other files have appeared, including a "Makefile" file. That file is what contains the info that the "make" tool uses to automatically compile the whole tool without the need for you to type each and every command [yes, people did this in the old days]

Note also that compiling the tool does not require root privileges either, as long as you have write permissions on the folder [otherwise you get the "permission denied" thing].

Then you need to "install" the tool. What I mean by "install" is that:

- The tool - or a link to it - must be written to a specific directory [/usr/local/bin/ or /usr/local/sbin/] This is done in order to be able to use the tool as a command from the command line, from any directory [see note below]
 - The tool's manual [if any] must also be written to a specific directory [/usr/local/share/man/man1 or something like that]
- That is done in order to be able to get help on the tool [with the command "man <tool name>"] from any directory

- Other essentials of the tool must be written to a specific directory [/usr/local/share/<tool name>/] and the configuration files at /etc/]

That is done for the program, in order to be able to work :P

Note two things:

- The command to install the tool, "make install", must be given with root privileges, because it involves writing into system folders

- Technically, it is not required to install the tool, because after you have compiled it, it can be run as it is, but the drawback is that it has to be run from its directory, not from any directory. Also, to get any help, you will have to check from the tools directory too.

So, after that sorry excuse of a crash course in compiling from source, here are the tools you will need to get, in order to get started:

aircrack-ng [very good suite for wifi cracking]

mdk3 [nice tool for mischief, but it can also serve a meaningful purpose]

crunch [wordlist generator - will be needed. You can also choose another one or even make one, it's a nice project]

I will not post links, you can google them yourself. Seriously now.

Note: I do believe it is better to be proficient with a few tools than novice with a ton. That's why I believe that the fewer tools you need to get the job done, the better. Automated tools can be time-savers, but you have to know exactly what the tool is doing. That is the main difference between [low and mid-level] hackers and sciddies.

Regarding each packet, a few tips on compilation:

AIRCRACK-NG

Pretty straightforward, the procedure should go smoothly, the Debian repositories have all the dependencies for this. A little tip: Also install sqlite3. It is an optional dependency for airolib-ng. While I won't speak much of it, it can be a nice flexible solution for storing your cracked passwords [and also evidence against you should you overdo it ?]

MDK3

As of now 19-10-2014, there is v6 available. I don't know if the person that made it intends to leave it just that, but for that version, some tips:

- There is to configure script, so you'll have to skip to the "make" and "make install" step.

- According to the developer, it uses the libraries of aircrack-ng, so installing aircrack-ng first is a good idea

- When you try to compile it you'll get an error. To remedy this, you have to make a slight change in the Makefile that is in the directory. In particular, you will have to change the following line [I think it's the second or third line in there]:

Code LINKFLAGS = -lpthreadinto this:

Code LINKFLAGS = -pthreadThen you should be good to go.

CRUNCH

No configure script here either, so again, make && sudo make install.

Please note that the install directory for this seems to be not /usr/local/bin, or at least, it copies/moves the tool in a directory it makes, /pentest/wordlists/crunch or something like that. Recommend to use as-is or make a link to /usr/local/bin. Will be somewhat awkward when you use it, but I don't like a /pentest/ directory in the root folder.

OK, after a hundred lines [though most are blanks] we are ready to get really started! Yay!

...

OK, let's just get started.

First things first:

****RECON****

- Know your network. The factors that you should take into consideration are the following:

Name [yes, it matters, read on] also known as ESSID, and MAC address of the Access Point, also known as BSSID

Channel

Clients connected to it

Security [Open/WEP/WPA/WPA2 ? Simple or Enterprise?]

* Signal strength [VERY important!]

This info can be acquired passively. RULE OF THUMB: NEVER EVER interact with a network unless you know EXACTLY what you are doing, and even then, always use protection! oh wait, that came out wrong... Damn it, just keep reading!

The following info can be acquired by using airodump-ng. To use it, you must first set a wireless network interface to passive mode. Aircrack-ng suite has a handy tool for that, called airmon-ng. Usage:

Code `sudo airmon-ng start <wireless interface>`

Which will show some data and will spawn an interface. It can be the very same card, but it can also be a virtual interface, mon0 [if you already have that, it will be mon1 etc]. Read carefully the documentation on the aircrack-ng website, to see if your wireless card is supported. Usually for just passive listening it's fine, but injection, that's another story. Also note the drivers you have, they could need updating.

Personally, I had a wireless card from BroadCom, which also required proprietary drivers to work, and was not supported. Every time I tried to set it with airmon-ng, I had major system crashes [kernel panics]. So I ordered an Atheros and changed it. I also ordered an Airlink USB wireless card with two antennae. Remember: wireless cards, memory cards and hard disks are relatively easy to change in a laptop.

After you fire up airodump-ng with the following command:

Code `airodump-ng <mon0 or whatever passive interface you have>`

you start watching carefully at the displayed info.

Here is an example output:

BSSID	PWR	Beacons	#Data, #s	CH	MB	ENC	CIPHER	AUTH	ESSID
DC:1A:1A:23:22:4C	-52	62	3022 467 11	54e	WPA2	CCMP	PSK	CYTA	

BSSID	STATION	PWR	Rate	Lost	Frames	Probe
DC:1A:1A:23:22:4C	9C:B7:0D:31:4A:82	0	54e-54e	3	3022	

Notice the data:

BSSID and ESSID under their respective columns. Signal strength under the PWR column. Channel under the CH column. Security type under the ENC, CIPHER and AUTH columns. The clients are under the Access points, that is, under the STATION column. This also shows WHERE they are connected [also important!]. As you can see, there is a client connected to the network "CYTA" [yes, that's me, though I modified the MACs]

Using airodump-ng without switches makes it scan all channels, which can give us a lot of data - especially in crowded areas! You can use some switches to filter the results, or use the --write switch and make a file with all you've listened and study it later.

Note: Although there are other tools that do this kind of job, like kismet, I do believe that the fewer, the better. Being novice in a thousand tools doesn't equal at all with being proficient with a few.

****ANALYSIS AND ESTIMATION****

There are no definite guidelines or instructions here, just some pointers I want to make:

- Low signal strength: You will have to get closer [or just change location], or change antenna. Using a directional antenna or a two-antennae card will just enable you to LISTEN better. That can be useful, but if you want to CONNECT, you also need to SPEAK. That is why, in some occasions, while you have good signal, you can't connect; you have to establish bidirectional communication. You can try to make your stick-like antenna a bit more directional with a soda can [nice hack, google it] but you can only get so far with the power it can draw from a USB cable or transmit from the integrated wireless card. There are some seriously thick types of antenna that can withstand all the watts, and are powered by an external source. but [for me] this is overkill. Just get closer. Or just listen. Whatever.
- No clients. This can be a bit tricky. In WEP, there can be a way around it [see exploitation techniques] but in WPA or WPA2 things get hard.
- Default name. For example Thomson04E3D23. There is a good chance default name equals default password. Remember: Defaults are an intruder's best friend.
- Connected clients. Take note of their MAC addresses, in case the access point has some sort of MAC filtering. Also take note of their probing networks [Under the "Probe" column in the example above]. This is where truly skillful people get some hints. When you check "Connect to this network automatically" [usually enabled by default in Windows] that computer "probes" aka sends signals to check for the existence of those networks [provided they are not connected TO them. My computer is set up to connect automatically to the CYTA network, yet it won't probe for it, as it is already connected]. So, have you stuck in a business network you can't but must breach? Check the cafeteria. You can exploit a vulnerable computer that probes for that network and have all the data you need [even for an enterprise network buwahahahahaha]
- Security. Usually, WEP needs 10-15 minutes at most to crack. If there is a lot of traffic, maybe

even less. With WPA/WPA2 things get tricky. And of course I'd advise against attacking enterprise networks. You need credentials for the RADIUS server. And that credentials can be even fingerprints, so... no. Not just yet anyway.

****EXPLOITATION AND TECHNIQUES****

I will try to list the basic method, a few tricks, but the main prerequisite is knowledge of what is what, and a functional brain.

MAIN TECHNIQUE

After you have decided which network [or client - read below] to attack, you get ready to get your hands dirty.

MAJOR NOTE: the tools and techniques discussed need to be executed with root privileges, with every bit of responsibility that thing means. Think before you type. Also, NetworkManager tends to interfere with MAC address spoofing, so it is usually a good idea to disable it while you are at it, and enable it when you have the key in your [digital] hands. Use `ifconfig`, `iwconfig`, `airmon-ng` and `macchanger` [nice tool for quickly changing your card's MAC address]. RTFM for instructions, seriously now.

OK, just a tip: usually the cards need to be switched off [e.g. `ifconfig wlan0 down`] before you change the MAC address or the mode to monitor [`iwconfig wlan0 mode monitor` or `ifconfig wlan0 hw ether XX:XX:XX:XX:XX:XX` or `macchanger wlan0 -a`, to name a few].

The main technique is as follows: Listen, store, crack. Simple enough? Well, it never goes smoothly, but let's see now how we do it:

Note: For the sake of brevity I will refer to the wireless interface as `wlan0` and all commands are to be entered as root. When not necessary, I will say so.

Listen:

Code `airmon-ng start wlan0` [set this interface to monitor mode] Presumably output is `mon0`

<Continued to next post>

<Continued...>

Store:

Code `airodump-ng -a --ivs --write /tmp/crack --channel 1 mon0` The `-a` switch filters out the non-associated clients. The `--channel 1` sets the listening to channel 1 [mostly] further filtering out any interference. The `--ivs` switch filters out most data, storing what it needs in order to crack the password [in a nutshell - google that shit!] and the `--write` switch writes the data to the specified file location. Note that the filename won't be "crack" but "crack-01.ivs" if the `--ivs` switch is enabled, or [if not enabled] it will write 4 files: `crack-01.cap`, `crack-01.csv`, `crack-01.kismet.csv`, `crack-01.kismet.netxml`. Note that if you re-type the same command [or at least the same name - that is, there is already a file with that name] it will just change the number at the end [`crack-02.ivs` etc.]

Also note that unless you further filter by BSSID or ESSID it will dump all networks on the channel, and that can be bad at an area with heavy traffic. Also, the location of the stored data: unless you want to keep them for further analysis, I suggest you store them in the `/tmp` folder. With the next

boot, they will be gone, no need to keep something like that if you are only interested in the key. Of course, if the key proves too stubborn to crack, you can copy the data from there to somewhere else [but NOT while they are "active" aka aircrack-ng is still in operation and writing on them]

Crack:

OK, now we got the data. Now what?

```
Code aircrack-ng /tmp/crack-01.ivs or
aircrack-ng /tmp/crack-01.capExample:
[NOTE: The data has been tampered with]
Code #aircrack-ng wpaone-01.ivs ote-01.cap
Opening wpaone-01.ivs
Opening ote-01.cap
Read 4913 packets.
```

#	BSSID	ESSID	Encryption
1	00:05:59:06:E4:3B	NetFaster IAD (PSTN)	WPA (0 handshake)
2	00:25:53:8D:E1:1E	Ceid Upatras	Unknown
3	08:18:1A:2A:58:A5	NetFaster_6EDu66	Unknown
4	00:26:44:B1:22:C8	Valerios	Unknown
5	00:1D:1C:23:06:D1	Oxygen-04850	Unknown
6	0C:D9:96:23:F8:90	XXXXXX	WPA (1 handshake)
7	00:05:59:42:2C:2F	hol - NetFaster WLAN	No data - WEP or WPA
8	00:24:17:1B:F1:11	Thomson23BA37	None (192.168.1.254)
9	58:BF:EA:49:A3:40		None (0.0.0.0)
10	08:76:FF:3F:AD:34	ThomsonBF3E28	WEP (834 IVs)

Index number of target network ? 6

```
Opening wpaone-01.ivs
Opening ote-01.cap
Please specify a dictionary (option -w).
```

Quitting aircrack-ng...
Notes:

- You can open multiple .ivs and/or .cap files [you can even use wildcards like *], so that you can pick up enough data from, say, a low-traffic WEP network that you visit each day and capture for an hour, and put them together, BUT! The data must be different, don't just copy-paste the same file, it doesn't work that way [bit of a dummy advice, though I am ashamed to admit that as a skiddie I tried that].

- Some networks, you see them without actually listening to the access point, but listening to a client connected to it. You can still exploit this and crack the key though [for WEP, that is]

- RTFM!!! Aircrack-ng suite has MANY capabilities.

As you can see from the example, I tried to crack a WPA network and failed. WPA encryption algorithm isn't as weak as WEP's, and it can only be bruteforced [for now]. So, for that I will need a wordlist. Or a source of keys [more later].

Let's try the network #10 now:
Code Aircrack-ng 1.2 beta3

[00:00:01] Tested 166929 keys (got 834 IVs)

KB	depth	byte(vote)
0	3/ 5	C5(1832) AE(1796) EE(1684) 8D(1648) 65(1644)
1	31/ 33	5B(1316) 3E(1280) 3C(1244) 5F(1244) 87(1244)
2	24/ 2	D7(1392) 5E(1356) 93(1356) DC(1356) 3D(1280)
3	31/ 3	A5(1356) D3(1316) D7(1316) 40(1280) 56(1280)
4	20/ 21	F9(1500) 14(1428) C6(1428) CE(1428) CA(1424)

Failed. Next try with 5000 IVs. After some coolish ASCII animation of uber 1337 cracking...bummer. We need more packets. We can still try to bruteforce it, though.

If you had enough packets, the message would be "Key found: XXXXXX" or something like that.

So, moving to the techniques part:

There will be 2 parts, WEP and WPA. Starting with the first...

WEP

Here things are relatively easy, but can still be frustrating [because they are easy]. What you need is data. Lots and lots of data. If the key is strong you need even more. So, if you are lucky and you happen to stumble in a busy network, or if your neighbour is into downloading tons and tons of porn, you are in luck. Just listen, crack, done. The main problem here is to generate traffic. To do that you need to cause some noise. And here comes something SUPER IMPORTANT!!!

GENERAL IMPORTANT NOTE!

When you are to interact with a network, be it harassment or injection or even entering, NEVER EVER use your own MAC address. A good idea is to use a nearby fellow's MAC or a client's MAC [maybe your boss' or that annoying douchebag's? Be creative. Make me proud] to cover your tracks [sort of - you want to make noise after all]. Most access points are configured to automatically block an "irrational" MAC, because that is a sign of malfunction, malware or intrusion :) So have some MAC addresses on the side. After all, the access point will react more readily to an already connected client to some attacks.

So, you have disabled Network manager, spoofed your MAC address. Now?

Now you can take one of two approaches:

-Chaotic approach

For this approach we use mdk3. Yep, time to wreak havoc. As it says in the documentation, there is the d mode [no seriously, this is the mode] which disassociates every client from every access point it can hear [you can blacklist or whitelist some, to limit the damage].

Usage:

Code mdk3 mon0 d With this attack [and airodump-ng recording] the clients will try to connect, get disconnected, try again etc., generating some traffic in the meantime. It can frustrate people [USE WITH CAUTION!] and there is a chance the sysadmin will drop the encryption entirely [yep, practical experience speaking], but generally it is far less effective, because it requires connected clients

[which could generate traffic either way] and maybe a dumb administrator.

NOTE: Study mdk3 further, it has some nice features, if it's OK to get a bit rough.

-Deliberate approach

This is the one I really like. In my humble opinion, this is what intruding [no, I won't say it "hacking"] should be like: stealth, ghost in the wires. So, what now?

Now you inject proper packets. For that you use the aireplay-ng, another tool in the aircrack-ng suite.

Aireplay-ng has many types of attack, some of which can be used in conjunction, while others can be standalone.

A common one is the fakeauth attack, which is used in conjunction with at least 3 other types [deauth, interactive and arpreplay], and what this attack does is, it creates an associated client [doesn't work for WPA!] so that the access point will interact with the packets we send its way. If there is another client, just take his MAC address.

As the manual entry of Aireplay-ng shows:

Quote: The fake authentication attack allows you to perform the two types of WEP authentication (Open System and Shared Key) plus associate with the access point (AP). This is useful is only useful when you need an associated MAC address in various aireplay-ng attacks and there is currently no associated client. It should be noted that the fake authentication attack does NOT generate any ARP packets. Fake authentication cannot be used to authenticate/associate with WPA/WPA2 Access Points.

Moral lesson: RTFM!!!

The attacks that require an associated client are:

deauth: Yes, you deauthenticate people. Like mdk3, but not with a wrecking ball, but with a scapel.

arpreply: What you do here is retransmit an ARP-request [google it] and the access point responds with a different-encoded packet [new IVs]. Quite reliable attack, though you might want to lower the transmit rate; the access point can block you if you overdo it.

Edit: Something I forgot to mention: This attack works also when there are not wireless clients but there are cable-connected clients [such as desktops or printers].

interactive: Same as arpreplay, but not only with ARP requests. You get to choose which packet to replay, or even make your own packets and replay them [advanced]

These are the main 3 attacks used against WEP. The chopchop and fragment attacks are quite less common. What IS interesting are the other two:

As the aireplay-ng documentation says:

Quote:caffelatte

In general, for an attack to work, the attacker has to be in the range of an AP and a connected client (fake or real). Caffe Latte attacks allows to gather enough packets to crack a WEP key without the need of an AP, it just need a client to be in range.

cfrag

This attack turns IP or ARP packets from a client into ARP request against the client. This attack works especially well against ad-hoc networks. As well it can be used against softAP clients and

normal AP clients.

I've used caffe-latte at a cafe network where I was far from the AP [small one, low transmit power] but close to a client, with great success! Still, it can be tricky. Just make sure you are convincing [Spoof your MAC to that of the access point's]

cfrag..... heheheheheh, I must admit I did it just to wreak havoc upon the ad-hoc some dudes had going to play Counter Strike..... heh, wonderful times...

Anyway, it works, but there is much less need to actually crack a WEP ad-hoc. Also note that all the above attacks work on WEP only.

WPA

To get to the WPA, you need to crack a handshake[google it]. That means that you need to be close enough to hear both access point and client exchange that info. That works if:

- You start listening when someone opens his laptop or connects through mobile
- Disassociate someone [or everyone] and MAKE them reconnect.

Again, mdk3 is your friend here, but with one extra: the g mode. According to mdk3's documentation:

Quote:g - WPA Downgrade test

deauthenticates Stations and APs sending WPA encrypted packets.

With this test you can check if the sysadmin will try setting his network to WEP or disable encryption.

There are still devices that are not able to connect to WPA devices. That method requires a bit of social engineering.

Speaking of this, there is still the last attack of aireplay-ng: migmode. According to the documentation:

Quote:This attack works against Cisco Aironet access points configured in WPA Migration Mode, which enables both WPA and WEP clients to associate to an access point using the same Service Set Identifier (SSID). The program listens for a WEP-encapsulated broadcast ARP packet, bitflips it to make it into an ARP coming from the attacker's MAC address and retransmits it to the access point. This, in turn, causes the access point to repeat the ARP packet with a new IV and also to forward the ARP reply to the attacker with a new IV. The program retransmits the same ARP packet over and over. However, each ARP packet repeated by the access point has a new IV as does the ARP reply forwarded to the attacker by the access point. It is all these new IVs which allow you to determine the WEP key.

An indicator of such a function on a device is the simultaneous existence of a WEP and WPA with same ESSID and BSSID.

So, now you have gotten yourself a handshake. A handshake to you then, and now... get cracking! [/badpuns]

aircrack-ng is your tool here, but with an exception: it needs a wordlist. There are tons of wordlists on the internet, google that shit.

Tips for bruteforcing:

-As said in the analysis section, default names can be an indicator for default passwords. Netfaster, Thomson and Speedtouch have ways of finding the default passwords [google it, this one you will enjoy :)]

-You will be surprised by the passwords, some are really VERY weak and/or guessable. Many people use their cell phone number or phone number or birthdays... which brings us to:

Making lists on-the-fly: use of wordlist generators [crunch for me]

aircrack-ng has this weird thing about the -w switch that says "...for input from standard input use - whis is somewhat familiar with linux folks and C/C++ folks as well, but we can use a pipe to feed aircrack-ng keys, generated on the spot, instead of storing and using them. Uses more CPU but less space, and more importantly, it can be shaped as WE want, for example in XX-XX-XXXX format [birthday format ™] or in mobile phone format [333XXXXXX - the numbers are random] hence reducing the 10-digit password to 7 or 6 digit one. So, what mobile connection provider does the neighbour have?

As you can see here, this is something I believe deeply, regarding intrusion, and intrusion detection/prevention: There are no useless info. It's how you use the info that makes it useful or useless.

The way to use the pipe feed is simple: first type the wordlist generator command on the left, then the '|' character, then the aircrack-ng command on the right. Note that you will have to preselect the network you want to crack [through BSSID or ESSID] because then the keyboard input is lost and all input goes from the wordlist generator. Notice that the data feed goes from left to right in the expression

```
Code crunch 5 10 | aircrack-ng wpaone-01.ivs ote-01.cap -w - -e XXXXXX
Code Aircrack-ng 1.2 beta3
```

```
[00:00:00] 1 keys tested (317.18 k/s)
```

```
KEY FOUND! [ yesican! ]
```

```
Master Key   : 44 97 5A 50 FD 23 11 76 F6 F4 FC C5 5A D3 2A F7
               F2 C9 6C A3 D6 79 97 2C A2 C4 73 82 54 F2 38 9B
```

```
Transient Key : 1A CB FF F6 50 5D 37 FD C8 99 A9 F4 92 1A D9 33
               3E 0B EF 42 76 12 6F 11 91 42 56 A0 74 65 5D EB
               45 4E 22 4F 03 66 16 2F 76 E5 48 A9 BD 04 74 21
               14 BA 56 ED E3 7D D9 6D B6 65 82 EF 35 04 FF 44
```

Note: that key was from another network.

So, here is my try for a tutorial. I tried to keep it to my standards, that is, explain things. It is still incomplete, and new attacks can be discovered, but still I think it can be a respectable try.

yacked2 - November 11, 2014, 6:32 am

As you mentioned at the title... ANOTHER, there are milions of tutorials how to crack WEP, WPA, WPS...

ciocan - November 11, 2014, 1:01 pm

This is the worst tutorial ever!!!

If some one try to fallow it will not crack anything...

Sorry op but is just to much theory and you miss the point... Keep it simple so even a monkey can fallow it.

bladedemon - November 12, 2014, 11:00 am

(Nov 11, 2014 07:32 AM)yacked2 Wrote: [\[CITE\]](#) As you mentioned at the title... ANOTHER, there are millions of tutorials how to crack WEP, WPA, WPS...

As I also said, nothing was up to what I consider a tutorial. I am strictly against the Next->Next->Finish procedure.

I want something to show me how this WORKS, not just how to do this, I want to see inside that black box. To that account, there are very few (if any) tutorials. For anything, for that matter.

(Nov 11, 2014 02:01 PM)ciocan Wrote: [\[CITE\]](#) This is the worst tutorial ever!!!

If some one try to fallow it will not crack anything...

Sorry op but is just to much theory and you miss the point... Keep it simple so even a monkey can fallow it.

As I also said, this is NOT a tutorial for skiddies, this is something I wrote for people who don't just follow instructions, but want to see also how it works. Old school hacking opinion maybe? (yes, I am somewhat old)

Please do not misunderstand; I am greatful for your feedback, but I also wanted to see what kind of people roam this forum, and their view on that matter.

ciocan - November 12, 2014, 5:10 pm

From what i see you mixed wep and wpa cracking. There are 2 different things.

1. You miss the command for generating IVs.
2. Also there is no command for getting the handshake.

You have make a passive cap file with all the ap's from channel 1... This is not ok.

I dont want to wait a client to connect to get the handshake or to wait until i have the necessary IVs to crack the wep. This may take in some cases a lot of time... Why to wait when i can get the handshake in a few seconds if a clients is connected or just few minutes to generate IVs?

If you want to see some well done tutorials look here:

<https://hackforums.net/showthread.php?tid=2391597>

bladedemon - November 13, 2014, 9:10 am

(Nov 12, 2014 06:10 PM)ciocan Wrote: [\[CITE\]](#) From what i see you mixed wep and wpa cracking. There are 2 different things.

1. You miss the command for generating IVs.
2. Also there is no command for getting the handshake.

You have make a passive cap file with all the ap's from channel 1... This is not ok.

I dont want to wait a client to connect to get the handshake or to wait until i have the necessary IVs to crack the wep. This may take in some cases a lot of time... Why to wait when i can get the handshake in a few seconds if a clients is connected or just few minutes to generate IVs?

If you want to see some well done tutorials look here:

[I can't post the link, even in quote, still under the 50 threshold] sigh this is getting tiresome and somewhat pointless.

I clearly seperated WEP and WPA/WPA2, and I made a distinction between regular and enterprise too [hinting to avoid the latter]. When I said RTFM I meant it. If I were to explain every little aspect of the protocol, I might as well copy-paste all the IEEE 802.11 family.

As for your other points:

1. You missed that part where I said:

Quote: arpreplay: What you do here is retransmit an ARP-request [google it] and the access point responents with a different-encoded packet [new IVs]. Quite reliable attack, though you might want to lower the transmit rate; the access point can block you if you overdo it.

Edit: Something I forgot to mention: This attack works also when there are not wireless clients but there are cable-connected clients [such as desktops or printers].

2. Also missed the part where I said [or actually wrote]:

Quote: To get to the WPA, you need to crack a handshake [google it]. That means that you need to be close enough to hear both access point and client exchange that info. That works if:

- You start listening when someone opens his laptop or connects through mobile
- Disassociate someone [or everyone] and MAKE them reconnect.

Again, mdk3 is your friend here, but with one extra: the g mode.

While it doesn't have the commands typed AGAIN [I wrote about them a few lines before, no point in telling things over and over] it does describe the attack vector.

P.S. The link you provided is a compilation of tutorials that, as I said at the very start of the thread, did not meet my criteria.

P.P.S. I really don't see a point in a flame war here, and it is pointless to continue, if by "continue" means explaining every little detail of what I wrote. If you need further explanations you can read the man pages of said tools, search it or [last resort] PM some people who know. Requesting this thread to close please.

SuperConductor - December 10, 2014, 7:19 am

i always use reaver to bruteforce wps key! 8-10h (but most of routers lock wps...)
or i use pyrit with cuda and start a dictionary attack