Cracking Wifi with Back\Track

Written by al1the corporati
Monday, 24 July 2006

Ever wondered just how secure your WEP protected wireless network is? Well today I'll show you how to test it. There have been a lot of articles written about this subject already and by now it is common knowledge that WEP is only the barest of security precautions. I'm going to show you how you can test your own wireless network's security using the linux livedistro back\track. Before we go any further, I feel it necessary to mention two things. The first being the ethics of hacking. Most of you are probably familiar with this subject already but, just to refresh your memory. Second, it goes without saying that this is for YOUR OWN NETWORK TESTING PURPOSES ONLY. Unauthorized access of other people's networks is illegal. If you have problems or questions about anything in this guide, for the love of god use google/wikipedia and look it up first. Don't just start ranting on forums like a moron without doing a little research first. There are probably other people who have had the same problems and solved them already. Ok, parental rant over. Lets get down to the dirty stuff:

First of all, you'll need to check and make sure your wireless card has the right chipset. Most wireless cards are programmed only to accept data that is addressed to them. Other cards, specifically the ones that are of use for wifi sniffing, are capable of picking up all traffic that is flying through the air. Common types are Atheros, Prism, Aironet, Realltek, Hermes, etc based cards. You are on your own figuring out what type of chipset your wireless card has, as its too vast to get into here, but check this thread for more info. Your probably just going to have to search for your specific card to find out what chipset it has then compare it to this compatability list. For a good discussion on types of cards that work, check this http://forums.remote-exploit.org/showthread.php?t=2191

Next, download a copy of back\track, a slackware distro designed for security testing purposes. This is a linux livecd, which means it will boot the entire OS from the cd. Download the ISO and use a burning program such as Nero, Alcohol or my personal favorite, the awesome freeware cd/dvd burning program cdburnerXP to burn the disk image to a cd. Pop the disk in and reboot, and boot from the disk. Backtrack may take a while to boot up.

When back\track boots up (and hopefully finds all your hardware) you will be presented with a login screen. To quote the venerable xatar, "Read the f**king screen!" The login, as it says above the prompt is "root" and the password is "toor" (minus the "). Note that linux is case sensitive. After you are logged in, you could run all of the commands I will get into later from this prompt. But that's no fun, so type in:

xconf

This should create a file /etc/X11/xorg.conf and autodetect your video settings. (with nvidia cards, you may still have video problems as I did, such as not getting above 640x480... should you choose to install backtrack to the harddrive, check out http://forums.remote-exploit.org/showthread.php?t=2176&highlight=nvidia for more info on fixing this)

To get the KDE gui desktop to start up, simply type:
If everything goes smoothly, you should be awash in the beautiful glow of the back\[track KDE desktop. Given the beautiful read only nature of the livecd, you can do anything to this operating system and not have to worry about messing it up. If things get a little weird, or screwed up, just reboot and the OS is back to normal. So GO EXPLORE, run random programs, see what they do, go nuts.

At the bottom left of the screen is a little icon that looks like a monitor with a black screen. **This is called the bash prompt.** This is where you will be spending most of your time, so click on this to **open up a new bash prompt.** Note that you can double click on the bar to the right of the tab that says "Shell" and it will create a new bash tab, negating the necessity to open up multiple instances of the bash window. First, a few networking commands to get you up to speed on your own system. Type

```
ifconfig -a
```

This will show you a list of all compatible network cards on your system. You should see a list of devices such as ath0, eth0, wifi0, wlan0 etc. One of these is your wireless card. If you have an Atheros based card, it will be ath0. Make note of the name of your card, as you will be using it later. **For the rest of this guide, I will be using ath0 since that is the card I have. Replace ath0**
with whatever card you have.

You can also check out your wireless cards specifically by typing in:

```
iwconfig
```

I’ve got two wireless cards. The one built into my laptop, an intel card (eth0) and an Atheros pcmcia card (ath0). Now that we have the name of our wireless cards, we can start sniffing. Some like to use Kismet to sniff for networks, but I find using airodump-ng to be easier and ultimately more effective. In your bash prompt, type:

```
airodump-ng --write out --ivs --abg ath0
```
This starts airodump-ng and tells it to begin sniffing data, write it to the file **out**, only capture IVs (**Initialization Vectors**), search the a, b and g bands using the ath0 card. Keep in mind, every time you specify the same output file name, such as "out", airodump-ng will append the file name with "-##" such as out-01.ivs, out-02.ivs, etc. You will see a list of access points on the top half of the screen, and clients on the bottom. Find your access point in the list. **Write down the BSSID or Mac address of the access point and any connected clients. You'll need it later.** From now on in this document, the access point's **mac address** will be referred to as APmac and the client **mac as CLmac.** The goal of the attack is to capture as many unique IVS as possible. Every time data is sent between the wireless server and client, each packet contains IV which are collected and then run through the aircrack-ng program for computation.

You should be seeing a ton of numbers flying by, but not updating vary quickly. That's because airodump-ng is searching all channels. Once you see your network, note what channel it is on (under the CH header). Stop airodump-ng by hitting:

```
ctrl-c
```

Now start it up again but this time we will add --channel # where # is the channel number of the **access point**, say, channel 6

```
airodump-ng --channel 6 --write out --ivs --abg ath0
```
Airodump-ng should be running much faster now, and updating constantly. You will see a number rising very quickly, this is generally the beacons. Beacons just basically say "hey, i'm an access point" about 10 times a second. You can judge the quality of your connection by how fluid the rise in beacons are. Other than this, they are useless for our purposes. For this type of attack it is important for there to be a client connected to the access point. So march over to your other computer and log on to the net wirelessly. In backtrack, you should see at the bottom a client pop up, the first MAC is the access point and the 2nd is the Client. Write down both. Open a new bash prompt and type:

```
 aireplay-ng -2 -b APmac -d ff:ff:ff:ff:ff -m 68 -n 68 -p 0841 -h CLmac ath0
```

Where APmac is the mac address (bssid) of the access point and CLmac is the mac address of the client. For a detailed explanation of what all these settings do, open up a new bash prompt and just type aireplay-ng and it will spew out all the controls and what they do. The only one not explained is that the very first -2 tells aireplay to do the 3rd attack method in the list at the bottom (the first being 0).

aireplay-ng will now start sniffing for a certain type of packet with a length no more and no less than 68 bytes between client and access point. It will say "Read ### packets". At this point, if there is significant data transfer between the client and ap, it may snag the right type of packet already and there is no need to do the next step. In this case, hit Y to use the packet and skip the next step. If however, it keeps reading packets for a while (more than a couple min) and does not pop up
saying "Use this packet?" then do the following:

**Open a new bash prompt** and type:

```
aireplay-ng -0 1 -a APmac -c CLmac ath0
```

This command will effectively terminate the connection between the AP and the client forcing the client to re-connect. It is this re-connection packet that we are looking to scoop up with the first instance of aireplay.

Go back to the first instance of aireplay and you should see something at the bottom of the screen saying "Use this packet?" **Hit Y** and aireplay will start sending out tons of packets to the AP. Switch over to airodump-ng which should still be running in the first bash prompt. Look at the data rate of the targeted AP. If all is going well, Airplay is spewing out packets like mad to the access point and airodump-ng is picking up the chatter in between, the data should be rising quickly. This is exactly what we want.

If for some reason the data isn't going up quickly, go back to the first aireplay-ng and hit:

```
ctrl-c
```

If aireplay had picked up any more packets, it will prompt you again if you want to use them. Try more packets. Also, you may need to get closer to your access point or try the aireplay-ng -0 method again. Experiment. Once you've got the data rate going up quickly, start aircrack-ng and start crunching the numbers. Type in

```
dir
```

To get a list of the files. One file should be the out file that you specified in airodump-ng, specifically out-01.ivs. Each time airodump-ng is started with the same file output name, it creates a new one tacking on -01, -02, etc. Make sure you know which one you are outputting to.

Type in:

```
aircrack-ng -f 2 -a 1 -b APmac -n 64 out-01.ivs
```
Again, if you want to know what all the parameters mean, open up a new bash and type `aircrack-ng` and it will tell you. Basically `-f` is the fudge factor- default is 2, a higher number will be a more thorough but slower search. `-b` filters out all but the specified mac of the AP, `-n` says to search for a 64 bit key. If it runs for a long time and finds nothing, either you don't have enough IVs, or you are searching under the wrong key length. Try 128. You can also run multiple instances of `aircrack` with different variables. Aircrack will continually update, notice the increasing IVs in the upper right as long is airodump-ng and aireplay are still going strong. After a bit of time, it should spit out your WEP key. Congrats! You now know how hard it is breaking into YOUR OWN NETWORK. Perhaps switch to WPA? If it didn't work, there could be any number reasons why. Do a little searching on the backtrack forums, google, etc, try setting up a different access point or learn how to do another type of attack, learn how to configure your hardware properly, etc, etc, etc. Take your time and explore the OS, if your new to linux, like i was when i started using backtrack, you'll have a lot to learn.

As a side note, In order to connect to a wireless network in backtrack, you must type in

```
iwconfig ath0 essid nameofnetwork key whateverthekeyis
```

```
ifconfig ath0 up
```

```
dheped ath0
```

A message should pop up in the bottom right of the screen saying something about ath0 being connected. To disconnect, before switching to another network, type:

```
ifconfig ath0 down
```

then repeat the steps above with the new network information.

Further reading:

I'd like to thank nuts, max, redkommie, jacky, digi, creaters of backtrack, xatar for writing a lot of
the guides that got me up to speed, the creators of aircrack-ng and all the awesome people on the rexploit forums.

-allthecorporati

is a musician, artist and tech freak. he currently resides in a shady ravine in a state of calm disillusionment.

Last Updated ( Monday, 24 July 2006 )