NAME

aircrack-ng - a 802.11 WEP / WPA-PSK key cracker

SYNOPSIS

aircrack-ng [options] <input file(s)>

DESCRIPTION

aircrack-ng is an 802.11 WEP, 802.11i WPA/WPA2, and 802.11w WPA2 key cracking program.

It can recover the WEP key once enough encrypted packets have been captured with airodump-ng. This part of the aircrack-ng suite determines the WEP key using two fundamental methods. The first method is via the PTW approach (Pyshkin, Tews, Weinmann). The main advantage of the PTW approach is that very few data packets are required to crack the WEP key. The second method is the FMS/KoreK method. The FMS/KoreK method incorporates various statistical attacks to discover the WEP key and uses these in combination with brute forcing.

Additionally, the program offers a dictionary method for determining the WEP key. For cracking WPA/WPA2 pre-shared keys, a wordlist (file or stdin) or an airolib-ng has to be used.

INPUT FILES

Capture files (.cap, .pcap), IVS (.ivs) or Hashcat HCCAPX files (.hccapx)

OPTIONS

Common options:

-a <amode>
: Force the attack mode: 1 or wep for WEP (802.11) and 2 or wpa for WPA/WPA2 PSK (802.11i and 802.11w).

-e <essid>
: Select the target network based on the ESSID. This option is also required for WPA cracking if the SSID is cloaked. For SSID containing special characters, see https://www.aircrack-ng.org/doku.php?id=faq#how_to_use_spaces_double_quote_and_single_quote_etc_in_ap_names

-b <bssid> or --bssid <bssid>
: Select the target network based on the access point MAC address.

-p <nbcpu>
: Set this option to the number of CPUs to use (only available on SMP systems) for cracking the key/passphrase. By default, it uses all available CPUs

-q
: If set, no status information is displayed.

-C <macs> or --combine <macs>
: Merges all those APs MAC (separated by a comma) into a virtual one.

-l <file>
: Write the key into a file. Overwrites the file if it already exists.

Static WEP cracking options:

-c
: Search alpha-numeric characters only.

-t
: Search binary coded decimal characters only.

-h
: Search the numeric key for Fritz!BOX

-d <mask> or --debug <mask>
: Specify mask of the key. For example: A1:XX:CF

-m <maddr>
: Only keep the IVs coming from packets that match this MAC address. Alternatively, use −m ff:ff:ff:ff:ff:ff to use all and every IVs, regardless of the network (this disables ESSID and BSSID filtering).

-n <nbits>
Specify the length of the key: 64 for 40-bit WEP, 128 for 104-bit WEP, etc., until 512 bits of length. The default value is 128.

-i <index>
Only keep the IVs that have this key index (1 to 4). The default behavior is to ignore the key index in the packet, and use the IV regardless.

-f <fudge>
By default, this parameter is set to 2. Use a higher value to increase the bruteforce level: cracking will take more time, but with a higher likelihood of success.

-k <korek>
There are 17 KoreK attacks. Sometimes one attack creates a huge false positive that prevents the key from being found, even with lots of IVs. Try −k 1, −k 2, ... −k 17 to disable each attack selectively.

-x or -x0
Disable last keybytes bruteforce (not advised).

-x1 Enable last keybyte bruteforcing (default)

-x2 Enable last two keybytes bruteforcing.

-X Disable bruteforce multithreading (SMP only).

-s Shows ASCII version of the key at the right of the screen.

-y This is an experimental single brute-force attack which should only be used when the standard attack mode fails with more than one million IVs.

-z Uses PTW (Andrei Pyshkin, Erik Tews and Ralf-Philipp Weinmann) attack (default attack).

-P <num> or --ptw-debug <num>
PTW debug: 1 Disable klein, 2 PTW.

-K Use KoreK attacks instead of PTW.

-D or --wep-decloak
WEP decloak mode.

-1 or --oneshot
Run only 1 try to crack key with PTW.

-M <num>
Specify maximum number of IVs to use.

-V or --visual-inspection
Run in visual inspection mode. Can only be used when using KoreK.

WEP and WPA-PSK cracking options

-w <words>
Path to a dictionary file for wpa cracking. Separate filenames with comma when using multiple dictionaries. Specify "-" to use stdin. Here is a list of wordlists: https://www.aircrack-ng.org/doku.php?id=faq#where_can_i_find_good_wordlists In order to use a dictionary with hexadecimal values, prefix the dictionary with "h:". Each byte in each key must be separated by ':'. When using with WEP, key length should be specified using -n.

-N <file> or --new-session <file>
Create a new cracking session. It allows one to interrupt cracking session and restart at a later time (using -R or --restore-session). Status files are saved every 10 minutes. It does not overwrite existing session file.

-R <file> or --restore-session <file>
Restore and continue a previously saved cracking session. This parameter is to be used alone, no other parameter should be specified when starting aircrack-ng (all the required information is in the session file).

WPA-PSK options:

-E <file>
Create Elcomsoft Wireless Security Auditor (EWSA) Project file v3.02.

-j <file>
Create Hashcat v3.6+ Capture file (HCCAPX).

-J <file>
Create Hashcat Capture file (HCCAP).

-S WPA cracking speed test.

-Z <sec>
WPA cracking speed test execution length in seconds.

-r <database>
Path to the airolib-ng database. Cannot be used with '-w'.

SIMD selection:

--simd=<option>
Aircrack-ng automatically loads and uses the fastest optimization based on instructions available for your CPU. This options allows one to force another optimization. Choices depend on the CPU and the following are all the possibilities that may be compiled regardless of the CPU type: generic, sse2, avx, avx2, avx512, neon, asimd, altivec, power8.

--simd-list
Shows a list of the available SIMD architectures, separated by a space character. Aircrack-ng automatically selects the fastest optimization and thus it is rarely needed to use this option. Use case would be for testing purposes or when a "lower" optimization, such as "generic", is faster than the automatically selected one. Before forcing a SIMD architecture, verify that the instruction is supported by your CPU, using −u.

Other options:

-H or --help
Show help screen

-u or --cpu-detect
Provide information on the number of CPUs and SIMD support

AUTHOR

This manual page was written by Adam Cecile <gandalf@le-vert.net> for the Debian system (but may be used by others). Permission is granted to copy, distribute and/or modify this document under the terms of the GNU General Public License, Version 2 or any later version published by the Free Software Foundation On Debian systems, the complete text of the GNU General Public License can be found in /usr/share/common-licenses/GPL.

SEE ALSO

airbase-ng(8)
aireplay-ng(8)
airmon-ng(8)
airodump-ng(8)
airodump-ng-oui-update(8)
airserv-ng(8)
airtun-ng(8)
besside-ng(8)

easside-ng(8)
tkiptun-ng(8)
wesside-ng(8)
airdecap-ng(1)
airdecloak-ng(1)
airolib-ng(1)
besside-ng-crawler(1)
buddy-ng(1)
ivstools(1)
kstats(1)
makeivs-ng(1)
packetforge-ng(1)
wpaclean(1)
airventriloquist(8)