

Airbase-ng

Usage: airbase-ng <options> <replay interface>

Syntax	Parameters	Description
-a	<i>bssid</i>	set Access Point MAC address
-i	<i>iface</i>	capture packets from this interface
-w	<i>WEP key</i>	use this WEP key to encrypt/decrypt packets
-W	<i>0 1</i>	[don't] set WEP flag in beacons 0 1 (default: auto)
-h	<i>MAC</i>	source mac for MITM mode
-f	<i>disallow</i>	disallow specified client MACs (default: allow)
-q	<i>none</i>	quiet (do not print statistics)
-v	<i>none</i>	verbose (print more messages) (long --verbose)
-M	<i>none</i>	M-I-T-M between [specified] clients and bssids
-A	<i>none</i>	Ad-Hoc Mode (allows other clients to peer) (long --ad-hoc)
-Y	<i>in out both</i>	external packet processing
-c	<i>channel</i>	sets the channel the AP is running on
-X	<i>none</i>	hidden ESSID (long --hidden)
-s	<i>none</i>	force shared key authentication
-S	<i>none</i>	set shared key challenge length (default: 128)
-L	<i>none</i>	Caffe-Latte attack (long --caffe-latte)
-N	<i>none</i>	Hirte attack (cfrag attack), creates arp request against wep client (long --cfrag)
-x	<i>nbpps</i>	number of packets per second (default: 100)
-y	<i>none</i>	disables responses to broadcast probes
-0	<i>none</i>	set all WPA,WEP,open tags. can't be used with -z & -Z
-z	<i>type</i>	sets WPA1 tags. 1=WEP40 2=TKIP 3=WRAP 4=CCMP 5=WEP104

Airbase-ng (cont)

-Z	<i>type</i>	same as -z, but for WPA2
-V	<i>type type</i>	fake EAPOL 1=MD5 2=SHA1 3=auto
-F	<i>prefix</i>	write all sent and received frames into pcap file
-P	<i>none</i>	respond to all probes, even when specifying ESSIDs
-I	<i>interval</i>	sets the beacon interval value in ms
-C	<i>seconds</i>	enables beaconing of probed ESSID values (requires -P)

Filter Options

Syntax	Parameters	Description
--	<i><file></i>	read a list of BSSIDs out of that file (short -B) bssids
--bssid	<i><MAC></i>	BSSID to filter/use (short -b)
--	<i><MAC></i>	MAC of client to accept (short -d) client
--	<i><file></i>	read a list of MACs out of that file (short -D) clients
--essid	<i><ESSID></i>	specify a single ESSID (short -e)
--	<i><file></i>	read a list of ESSIDs out of that file (short -E) essids

Airdecloak-ng

Usage: airdecloak-ng [options]

Syntax	Parameter	Description
-i	<i>input file</i>	Path to the capture file
--bssid	<i>BSSID</i>	BSSID of the network to filter.
--ssid	<i>ESSID</i>	ESSID of the network to filter (not yet implemented).
--filters	<i>filters</i>	Apply these filters in this specific order. They have to be separated by a ','.
-	<i>none</i>	Assume that null packets can be cloaked (not yet implemented).
-	<i>none</i>	Disable the base filter.
disable-base _filter		



Airdecloak-ng (cont)

-drop-frag *none* Drop all fragmented packets. In most networks, fragmentation is not needed.

Airdrop-ng

Usage: airdrop-ng [options] <pcap file>

Syntax	Parameter	Description
-i	<i>card</i>	Wireless card in monitor mode to inject from
-t	<i>csv file</i>	Airodump txt file in CSV format NOT the pcap
-p	<i>psyco</i>	Disable the use of Psyco JIT
-r	<i>Rule File</i>	Rule File for matched deauths
-u	<i>update</i>	Updates OUI list
-d	<i>Driver</i>	Injection driver. Default is mac80211
-s	<i>sleep</i>	Time to sleep between sending each packet
-b	<i>debug</i>	Turn on Rule Debugging
-l	<i>key</i>	Enable Logging to a file, if file path not provided airdrop will log to default location
-n	<i>nap</i>	Time to sleep between loops

Airdecap-ng

Usage: airdecap-ng [options] <pcap file>

Syntax	Parameter	Description
-l	<i>none</i>	don't remove the 802.11 header
-b	<i>bssid</i>	access point MAC address filter
-k	<i>pmk</i>	WPA/WPA2 Pairwise Master Key in hex
-e	<i>ssid</i>	target network ascii identifier
-p	<i>pass</i>	target network WPA/WPA2 passphrase
-w	<i>key</i>	target network WEP key in hexadecimal

Airgraph-ng

Usage: python airgraph-ng -i [airodumpfile.txt] -o [outputfile.png] -g [CAPR OR CPG]

Syntax	Description
-i	Input File
-o	Output File

Airgraph-ng (cont)

- g Graph Type [CAPR (Client to AP Relationship) OR CPG (Common probe graph)]
- a Print the about
- h Print this help

Aircrack-ng

Usage: aircrack-ng [options] <capture file(s)>

Syntax	Parameter	Description
-a	<i>amode</i>	Force attack mode (1 = static WEP, 2 = WPA/WPA2-PSK)
-b	<i>bssid</i>	Long version -bssid. Select the target network based on the access point's MAC address.
-e	<i>ssid</i>	If set, all IVs from networks with the same ESSID will be used. This option is also required for WPA/WPA2-PSK cracking if the ESSID is not broadcasted (hidden).
-p	<i>nbcpu</i>	On SMP systems: # of CPU to use. This option is invalid on non-SMP systems
-q	<i>none</i>	Enable quiet mode (no status output until the key is found, or not)
-c	<i>none</i>	(WEP cracking) Restrict the search space to alpha-numeric characters only (0x20 - 0x7F)
-t	<i>none</i>	(WEP cracking) Restrict the search space to binary coded decimal hex characters
-h	<i>none</i>	(WEP cracking) Restrict the search space to numeric characters (0x30-0x39) These keys are used by default in most Fritz!BOXes
-d	<i>start</i>	(WEP cracking) Long version -debug. Set the beginning of the WEP key (in hex), for debugging purposes.
-m	<i>maddr</i>	(WEP cracking) MAC address to filter WEP data packets. Alternatively, specify -m ff:ff:ff:ff:ff:ff to use all and every IVs, regardless of the network.
-M	<i>number</i>	(WEP cracking) Sets the maximum number of ivs to use.



Aircrack-ng (cont)

-n	<i>nbits</i>	(WEP cracking) Specify the length of the key: 64 for 40-bit WEP, 128 for 104-bit WEP, etc. The default value is 128.
-i	<i>index</i>	(WEP cracking) Only keep the IVs that have this key index (1 to 4). The default behaviour is to ignore the key index.
-f	<i>fudge</i>	(WEP cracking) By default, this parameter is set to 2 for 104-bit WEP and to 5 for 40-bit WEP. Specify a higher value to increase the bruteforce level: cracking will take more time, but with a higher likelihood of success.
-H	<i>none</i>	Long version - -help. Output help information.
-l	<i>file name</i>	(Lowercase L, ell) logs the key to the file specified.
-K	<i>none</i>	Invokes the Korek WEP cracking method. (Default in v0.x)
-k	<i>korek</i>	(WEP cracking) There are 17 korek statistical 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.
-p	<i>threads</i>	Allow the number of threads for cracking even if you have a non-SMP computer.
-r	<i>database</i>	Utilizes a database generated by airolib-ng as input to determine the WPA key. Outputs an error message if aircrack-ng has not been compiled with sqlite support.
-	<i>none</i>	(WEP cracking) Disable last keybytes brutforce.
x/-		
x0		
-	<i>none</i>	(WEP cracking) Enable last keybyte bruteforcing (default).
x1		
-	<i>none</i>	(WEP cracking) Enable last two keybytes bruteforcing.
x2		
-x	<i>none</i>	(WEP cracking) Disable bruteforce multithreading (SMP only).

Aircrack-ng (cont)

-	<i>none</i>	(WEP cracking) Experimental single bruteforce attack which should only be used when the standard attack mode fails with more than one million IVs
y		
-	<i>none</i>	Long form - -cpu-detect. Provide information on the number of CPUs and MMX support. Example responses to "aircrack-ng - -cpu-detect" are "Nb CPU detected: 2" or "Nb CPU detected: 1 (MMX available)".
u		
-	<i>words</i>	(WPA cracking) Path to a wordlist or "-" without the quotes for standard in (stdin).
w		
-	<i>none</i>	Invokes the PTW WEP cracking method. (Default in v1.x)
z		
-	<i>none</i>	Long version - -ptw-debug. Invokes the PTW debug mode.
P		
-	<i>MACs</i>	Long version - -combine. Merge the given APs to a virtual one.
C		
-	<i>none</i>	Long version - -wep-decloak. Run in WEP decloak mode.
D		
-	<i>none</i>	Long version - -visual-inspection. Run in visual inspection mode.
V		
-	<i>none</i>	Long version - -oneshot. Run in oneshot mode.
1		
-	<i>none</i>	WPA cracking speed test.
S		
-	<i>none</i>	Show the key in ASCII while cracking
s		
-	<i>file></i>	(WPA cracking) Create EWSA Project file v3
E		
-	<i>file</i>	(WPA cracking) Create Hashcat Capture file
J		

Aireplay-ng

Usage: aireplay-ng <options> <replay interface>

Filter Options

Syntax	Parameters	Description
-b	<i>bssid</i>	MAC address, Access Point
-d	<i>dmac</i>	MAC address, Destination
-s	<i>smac</i>	MAC address, Source
-m	<i>len</i>	minimum packet length



Aireplay-ng (cont)

-n	<i>len</i>	maximum packet length
-u	<i>type</i>	frame control, type field
-v	<i>subt</i>	frame control, subtype field
-t	<i>tods</i>	frame control, To DS bit
-f	<i>fromds</i>	frame control, From DS bit
-w	<i>iswep</i>	frame control, WEP bit

Replay Options

Syntax	Parameters	Description
-x	<i>nbpps</i>	number of packets per second
-p	<i>fctrl</i>	set frame control word (hex)
-a	<i>bssid</i>	set Access Point MAC address
-c	<i>dmac</i>	set Destination MAC address
-h	<i>smac</i>	set Source MAC address
-e	<i>essid</i>	For fakeauth attack or injection test, it sets target AP SSID. This is optional when the SSID is not hidden.
-j	<i>none</i>	arp replay attack, inject FromDS pkts
-g	<i>value</i>	change ring buffer size (default: 8)
-k	<i>IP</i>	set destination IP in fragments
-l	<i>IP</i>	set source IP in fragments
-o	<i>npckts</i>	number of packets per burst (-1)
-q	<i>sec</i>	seconds between keep-alives (-1)
-y	<i>prga</i>	keystream for shared key auth
-B or -bittest	<i>none</i>	bit rate test (Applies only to test mode)
-D	<i>none</i>	disables AP detection. Some modes will not proceed if the AP beacon is not heard. This disables this functionality.
-F or -fast	<i>none</i>	chooses first matching packet. For test mode, it just checks basic injection and skips all other tests.

Aireplay-ng (cont)

-R	<i>none</i>	disables /dev/rpc usage. Some systems experience lockups or other problems with RTC. This disables the usage.
----	-------------	---

Source options

Syntax	Parameters	Description
iface	<i>none</i>	capture packets from this interface
-r	<i>file</i>	extract packets from this pcap file

Attack modes

Syntax	Parameters	Description
--deauth	<i>count</i>	deauthenticate 1 or all stations (-0)
--fakeauth	<i>delay</i>	fake authentication with AP (-1)
--interactive	<i>none</i>	interactive frame selection (-2)
--arpreply	<i>none</i>	standard ARP-request replay (-3)
--chopchop	<i>none</i>	decrypt/chopchop WEP packet (-4)
--fragment	<i>none</i>	generates valid keystream (-5)
--test	<i>none</i>	injection test (-9)

