Aircrack-ng Suite Cheat Sheet

by itnetsec via cheatography.com/21064/cs/6776/

Airbase-ng

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

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

Airbase-ng (cont)

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

Filter Options

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

Airdecloak-ng

Usage: `airdecloak-ng [options]`

Syntax	Parameter	Description
-i	*input file*	Path to the capture file
–bssid	*BSSID*	BSSID of the network to filter.
–ssid	*ESSID*	ESSID of the network to filter (not yet implemented).
–filters	*filters*	Apply theses filters in this specific order. They have to be separated by a ','.
–null-packets	*none*	Assume that null packets can be cloaked (not yet implemented).
–disable-base_filter	*none*	Disable the 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`	*card*	Wireless card in monitor mode to inject from
`-t`	*csv file*	Airodump txt file in CSV format NOT the pcap
`-p`	*psyco*	Disable the use of Psyco JIT
`-r`	*Rule File*	Rule File for matched deauths
`-u`	*update*	Updates OUI list
`-d`	*Driver*	Injection driver. Default is mac80211
`-s`	*sleep*	Time to sleep between sending each packet
`-b`	*debug*	Turn on Rule Debugging
`-l`	*key*	Enable Logging to a file, if file path not provided airdrop will log to default location
`-n`	*nap*	Time to sleep between loops

Airdecap-ng

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

Syntax	Parameter	Description
`-l`	*none*	don't remove the 802.11 header
`-b`	*bssid*	access point MAC address filter
`-k`	*pmk*	WPA/WPA2 Pairwise Master Key in hex
`-e`	*essid*	target network ascii identifier
`-p`	*pass*	target network WPA/WPA2 passphrase
`-w`	*key*	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`	*amode*	Force attack mode (1 = static WEP, 2 = WPA/WPA2-PSK)
`-b`	*bssid*	Long version - -bssid. Select the target network based on the access point's MAC address.
`-e`	*essid*	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`	*nbcpu*	On SMP systems: # of CPU to use. This option is invalid on non-SMP systems
`-q`	*none*	Enable quiet mode (no status output until the key is found, or not)
`-c`	*none*	(WEP cracking) Restrict the search space to alpha-numeric characters only (0x20 - 0x7F)
`-t`	*none*	(WEP cracking) Restrict the search space to binary coded decimal hex characters
`-h`	*none*	(WEP cracking) Restrict the search space to numeric characters (0x30-0x39) These keys are used by default in most Fritz!BOXes
`-d`	*start*	(WEP cracking) Long version –debug. Set the beginning of the WEP key (in hex), for debugging purposes.
`-m`	*maddr*	(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`	*number*	(WEP cracking) Sets the maximum number of ivs to use.

Aircrack-ng (cont)

-n	*nbits*	(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	*index*	(WEP cracking) Only keep the IVs that have this key index (1 to 4). The default behaviour is to ignore the key index.
-f	*fudge*	(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 likelyhood of success.
-H	*none*	Long version - -help. Output help information.
-l	*file name*	(Lowercase L, ell) logs the key to the file specified.
-K	*none*	Invokes the Korek WEP cracking method. (Default in v0.x)
-k	*korek*	(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	*threads*	Allow the number of threads for cracking even if you have a non-SMP computer.
-r	*database*	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.
-x/-x0	*none*	(WEP cracking) Disable last keybytes brutforce.
-x1	*none*	(WEP cracking) Enable last keybyte bruteforcing (default).
-x2	*none*	(WEP cracking) Enable last two keybytes bruteforcing.
-X	*none*	(WEP cracking) Disable bruteforce multithreading (SMP only).

Aircrack-ng (cont)

-y	*none*	(WEP cracking) Experimental single bruteforce attack which should only be used when the standard attack mode fails with more than one million IVs
-u	*none*	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)”.
-w	*words*	(WPA cracking) Path to a wordlist or “-” without the quotes for standard in (stdin).
-z	*none*	Invokes the PTW WEP cracking method. (Default in v1.x)
-P	*none*	Long version - -ptw-debug. Invokes the PTW debug mode.
-C	*MACs*	Long version - -combine. Merge the given APs to a virtual one.
-D	*none*	Long version - -wep-decloak. Run in WEP decloak mode.
-V	*none*	Long version - -visual-inspection. Run in visual inspection mode.
-1	*none*	Long version - -oneshot. Run in oneshot mode.
-S	*none*	WPA cracking speed test.
-s	*none*	Show the key in ASCII while cracking
-E	*file*>	(WPA cracking) Create EWSA Project file v3
-J	*file*	(WPA cracking) Create Hashcat Capture file

Aireplay-ng

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

Filter Options

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

Aireplay-ng (cont)		
-n	*len*	maximum packet length
-u	*type*	frame control, type field
-v	*subt*	frame control, subtype field
-t	*tods*	frame control, To DS bit
-f	*fromds*	frame control, From DS bit
-w	*iswep*	frame control, WEP bit

Replay Options

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

Aireplay-ng (cont)		
-R	*none*	disables /dev/rtc usage. Some systems experience lockups or other problems with RTC. This disables the usage.

Source options

Syntax	Parameters	Description
iface	*none*	capture packets from this interface
-r	*file*	extract packets from this pcap file

Attack modes

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