Commands

-check *<file>*	check capfile *<file>* for handshakes.
-cracked	display previously-cracked access points
$ aircarck-ng -S	WPA cracking speed test
$ hashcat.exe -m 2500 -b -w 4	**- b** : run benchmark of selected hash-modes **- m 2500** : hash mode - WPA-EAPOL-PBKDF2 **- w 4** : workload profile 4 (nightmare)

I find that I frequently need to go back and work with uncracked keys, that is the biggest issuse you will run into, due to the cracking process having limited capabilities.check your CPU first to see capabilities. It may b best if you have hashcat setup w/GPU acceleration to use the bottom command outside of wifite for fastest results and greatest capabilities.

Global (Frequently used)

-all	attack all targets.
-mac	Changes MAC address of 'iface' to a random MAC.
-pow *<db>*	attacks any targets with signal strenghth > *<db>*

If it is important to stay anonymous, make sure to -mac to randomize your address.
part of the charm of Wifite is the automation to attack multiple targets and just letting the software run. Attack success can greatly depend on signal strength, due to proximity, and ability to successfullfuly send and recievec the packlets.
A successfull trick I have found is not to attack targets below a signal strength. **-pow 50**, is a good place to start.

Attacks

-wep	only target WEP networks
-wps	only target WPS networks
-wpa	only target WPA networks (works with -wps -wep)
-wepca *<n>*	start cracking when number of **ivs** surpass n *[10000]*
-crack *<dic>*	crack WPA handshakes using *<dic>* wordlist file
-dict *<file>*	specify dictionary to use when cracking WPA

Setting the type of targets to focus on can help. Its also nice to run again a certain type of attacks that can be done quickly at one time. For instance running against WPS attacks and then moving on to doing full sets of WEP attacks on sevrel targets.
It can be a good idea, to capture a WPA handshake, run against a short list, since cracking is improbable, then making use of GPU/CPU acceleration setup on Hashcat and trying working against a larger list. SEE Commands.