Wireless networks Security cheatsheet

Dr Ayman El Hajjar

School of Computer Science and Engineering, University of Westminster

iw command

- Show all devices: `iw dev`
- Show information about wlan0: `iw dev wlan0 info`
- List supported commands: `iw list`
- Scan for networks: `sudo iw dev wlan0 scan | less`
- Set interface down: `sudo ip link set wlan0 down`
- Bring it back up: `sudo ip link set wlan0 up`

wpa_supplicant

Manage wireless authentication .

- **Start wpa_supplicant for interface** `wlan0`:
- **Manual wpa_supplicant**:
  `sudo wpa\_supplicant -B -i wlan0 -c /etc/wpa\_supplicant/wpa\_supplicant.conf -D nl80211`
  Kill manual run: `sudo pkill wpa\_supplicant` (or stop service if using systemd)
- **Interactive control**: use `sudo wpa\_cli -i wlan0` to scan, select network, and trigger reconfig/connect.

dhcpcd - quick start

Manage IP lease

- `dhcpcd -4 wlan0`
- `dhcpcd wlan0 -k`

Wireless Network Security - Module Labs Repository

Quick clone & setup

- Clone: `git clone https://github.com/azelhajjar/6CSEF005W.git`
- If cloning to Desktop: `cd ~/Desktop` then run the `git clone` command above.
- Make all scripts executable: `find . -type f -name "*.sh" -exec chmod +x {} \;`
- To update an existing clone: `cd ~/6CSEF005W \&\& git pull`

Running an AP

1. Attach the Wireless adapter to the VM and confirm: `iw dev`
2. Enter the AP folder: `cd ~/6CSEF005W/ap/`
3. Start the example AP (Open AP): `sudo ./open-ap.sh`
   Stop an interactive AP with `Ctrl+C` (teardown runs and restores interface to managed mode).

Teardown & housekeeping

- After the lab run the teardown script: `sudo ./ap-teardown.sh` (stops services, removes NAT, disables forwarding, restores interface).

Aircrack-ng suite at a glance

- **airmon-ng** - enable / disable monitor mode (creates monitor iface, kills interfering processes).
- **airodump-ng** - passive capture / scan: list APs/clients and write capture files (handshakes/PMKIDs).
- **aireplay-ng** - packet injection (deauth, replay, fake auth, etc.) to provoke traffic (lab only).
- **aircrack-ng** - offline cracking of WEP/WPA handshakes (wordlists / statistical attacks).
- **airdecap-ng** - decrypt captured WEP/WPA packets when you already have the key.
- **packetforge-ng** - craft/forge custom 802.11 frames for injection/testing.
- **airbase-ng** - run a test/rogue AP (evil-twin/captive-portal experiments in controlled lab).
- **airolib-ng** - manage local SSID/password databases (precompute / index wordlists for batch cracking).

airmon-ng

Enable / disable monitor mode (helper that kills interfering processes). Example: `sudo airmon-ng start wlan0` → creates `wlan0mon`. Restore: `sudo airmon-ng stop wlan0mon` or use `ip/iw` to set managed mode.

airdecap-ng

Decrypt captured WEP/WPA frames when you already have the key (for analysis). Usage: `airdecap-ng -e <SSID> -p <passphrase> capture.cap` (writes decrypted capture)

aireplay-ng

Packet injection / active tests (deauth, replay, fake auth, etc.) — lab use only. Deauth: `sudo aireplay-ng -deauth 10 -a <BSSID> wlan0mon`
Fake auth: `sudo aireplay-ng -fakeauth 10 -a <BSSID> -h <your_MAC> wlan0mon`

airbase-ng

Lightweight AP emulator (evil-twin / captive-portal experiments in controlled lab). Run: `sudo airbase-ng -e "TestAP" -c <CH> wlan0mon` (creates bridged/testing interface)

John the Ripper (jumbo)

Convert PMKID capture to John (Jumbo) format and crack with john:

- Convert to John format:
  `hcxpcapngtool -john=pmkid.john pmkid.pcapng`
- Crack with John:
  `john -wordlist=wordlist.txt pmkid.john`
- Show cracked results:
  `john -show pmkid.john`

airodump-ng

Passive capture / scanning tool — lists APs/clients and saves captures.
Scan: `sudo airodump-ng wlan0mon`
Targeted capture: `sudo airodump-ng -bssid <BSSID> -c <CH> -w capture wlan0mon`

packetforge-ng

Craft / forge arbitrary 802.11 frames for injection with `aireplay-ng`. Example: build a fake data/auth packet (then inject with `aireplay-ng -inject`). (Used for test/fuzzing and specialised attacks in lab.)

aircrack-ng

Offline key cracking for WEP/WPA handshakes (wordlists / statistical attacks). Crack: `aircrack-ng -w wordlist.txt capture-01.cap -b <BSSID>`

airolib-ng

Local DB manager for SSIDs / wordlists — precompute / index cracking data for batch jobs. Create DB / import wordlist: `airolib-ng db_name -import wordlist.txt` and then use with aircrack workflows.

Convert PMKID for hashcat using hcxtools

Convert PMKID capture to Hashcat format and crack with hashcat:

- Convert to (Hashcat):
  `hcxpcapngtool -o capture.22000 pmkid.pcapng` then `hashcat -m 22000 capture.22000 wordlist.txt`

Fast GPU-based cracking. Use mode `22000` for combined PMKID/EAPOL in modern hashcat; `16800` is legacy in some workflows.

- Crack (modern):
  `hashcat -m 22000 capture.22000 wordlist.txt -status -status-timer=10`
- Crack (legacy/example):
  `hashcat -m 16800 capture.16800 wordlist.txt -status`

tcpdump - quick reference

Packet capture & lightweight CLI analyser. Useful for quick captures on interfaces, saving to pcap and lightweight filtering before heavy analysis in Wireshark.

- Capture to file (full packets):
  `sudo tcpdump -i wlan0mon -s 0 -w capture.pcap -n`
- Capture and show link-layer headers (verbose):
  `sudo tcpdump -i wlan0mon -s 0 -w capture.pcap -e -vv`
- Capture only EAPOL frames (WPA handshakes):
  `sudo tcpdump -i wlan0mon -s 0 -w eapol.pcap 'ether proto 0x888e'`
- Capture traffic to/from a MAC (BSSID or client):
  `sudo tcpdump -i wlan0mon -s 0 ether host AA:BB:CC:DD:EE:FF`
- Read & filter saved pcap (display, no write):
  `tcpdump -r capture.pcap -nn -vv`
- Useful flags: `-i <iface>` (interface), `-s 0` (full packet), `-w <file>` (write pcap), `-e` (link headers), `-n,-nn` (no name resolution).

Tip: run on a monitor-mode interface (e.g. `wlan0mon`) to see raw 802.11 frames; use `ether proto 0x888e` to isolate EAPOL if you only need handshakes.

tshark - quick reference

Command-line Wireshark: capture, filter, and extract fields from pcap files or live interfaces.

- Capture to file (live):
  `sudo tshark -i wlan0mon -s 0 -w capture.pcap -c 100`
- Capture with BPF (kernel) filter (e.g. only EAPOL):
  `sudo tshark -i wlan0mon -f "ether proto 0x888e" -w eapol.pcap`
- Capture + display-filter (only show EAPOL in output):
  `sudo tshark -i wlan0mon -s 0 -w capture.pcap -Y "eapol"`
- Read & filter saved pcap:
  `tshark -r capture.pcap -Y "eapol" -V`
- Export selected fields as CSV (example: source, dest, frame time):
  `tshark -r capture.pcap -T fields -e frame.time -e wlan.sa -e wlan.da -E header=y -E separator=,`
- Useful flags: `-i <iface>` (iface), `-s 0` (full packet), `-w <file>` (write pcap), `-f "<BPF>"`, `-Y "<display>"`, `-c <n>` (packet count), `-V` (verbose packet dump).

Tip: prefer a BPF capture filter (`-f`) to reduce capture size; use `-Y "eapol"` to display only handshake frames while analysing.