Lab 3 Part 01 - Passive Sniffing in 802.11 Networks

- **Name**: Brandon Cortez
- **Student ID**: 820561762
- **Date**: 4/18/2025

1. Objective

Read the entire lab document and briefly state the overall objective of the lab in your own words.

The objective of this lab is to evaluate the confidentiality of different Wi-Fi security protocols, Open, WEP, and WPA, by passively sniffing network traffic and analyzing whether an unauthorized observer can view or decrypt the transmitted data.

2. Testbed Setup

- Reserve the ORBIT "outdoor testbed (or another sandbox if unavailable) and SSH into the testbed console.
    - Show successful SSH access to the console.

```
PS C:\Users\Brandon Cortez\.ssh> ssh brcortez@console.outdoor.orbit-lab.org
The authenticity of host 'console.outdoor.orbit-lab.org (128.6.192.146)' can't be established.
ECDSA key fingerprint is SHA256:0/DBXH8YeDhZB262/LS/giQXSirQVcbzvC06Y4Y48Zw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'console.outdoor.orbit-lab.org,128.6.192.146' (ECDSA) to the list of known
hosts.

                              Welcome to
   ___  ____  ____ ___ _____     _        _    ____
  / _ \|  _ \| __ )_ _|_   _|   | |      / \  | __ )   ___  _ __ __ _
 | | | | |_) |  _ \| |  | |_____| |     / _ \ |  _ \  / _ \| '__/ _` |
 | |_| |  _ <| |_) | |  | |_____| |___ / ___ \| |_) || (_) | | | (_| |
  \___/|_| \_\____/___| |_|     |_____/_/   \_\____(_)___/|_|  \__, |
                                                               |___/

 Hostname : console.outdoor.orbit-lab.org
 OS       : Ubuntu 16.04.7 LTS 4.15.0-142-generic x86_64
 CPU      : 1 x Intel Xeon Processor (Skylake)
            Total of 6 cores, 6 threads
 Load Avg : 0.00 (1min), 0.00 5(min), 0.00 (15min)
 Memory   : 1.0G (Free) / 3.9G (Total)
 Uptime   : up 39 weeks, 14 hours, 42 minutes
 Users    : 2


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

- Configure the 4 nodes (e.g., node1-1, node1-2, node1-3, node1-4) on the testbed for the experiment. (Note: You may use any radio nodes within range that have the Atheros AR 9xxx wireless adapter)
    - Show confirmation of nodes successfully being reset and powered on.

```
brcortez@console:~$ omf tell -a on -t node1-1.outdoor.orbit-lab.org,node1-2.outdoor.orbit-lab.org,node
1-3.outdoor.orbit-lab.org,node1-4.outdoor.orbit-lab.org
/usr/share/omf-expctl-5.4/gems/gems/oml4r-2.9.5/lib/oml4r.rb:29: warning: already initialized constant
 OML4R::DEF_SERVER_PORT
/var/lib/gems/2.3.0/gems/oml4r-2.10.6/lib/oml4r.rb:33: warning: previous definition of DEF_SERVER_PORT
 was here
/usr/share/omf-expctl-5.4/gems/gems/oml4r-2.9.5/lib/oml4r.rb:30: warning: already initialized constant
 OML4R::DEF_PROTOCOL
/var/lib/gems/2.3.0/gems/oml4r-2.10.6/lib/oml4r.rb:34: warning: previous definition of DEF_PROTOCOL wa
s here

 INFO NodeHandler: OMF Experiment Controller 5.4 (git 861d645)
 INFO NodeHandler: Slice ID: default_slice (default)
 INFO NodeHandler: Experiment ID: default_slice-2025-04-18t04.39.37.163+00.00
 INFO NodeHandler: Message authentication is disabled
 INFO property.resetDelay: resetDelay = 300 (Fixnum)
 INFO property.resetTries: resetTries = 1 (Fixnum)
 INFO property.nodes: nodes = "node1-1.outdoor.orbit-lab.org,node1-2.outdoor.orbit-lab.org,node1-3.out
door.orbit-lab.org,node1-4.outdoor.orbit-lab.org" (String)
 INFO property.command: command = "on" (String)

Talking to the CMC service, please wait
-----------------------------------------------
 Node: node1-1.outdoor.orbit-lab.org     Reply: OK
 Node: node1-2.outdoor.orbit-lab.org     Reply: OK
 Node: node1-3.outdoor.orbit-lab.org     Reply: OK
 Node: node1-4.outdoor.orbit-lab.org     Reply: OK
-----------------------------------------------

 INFO EXPERIMENT_DONE: Event triggered. Starting the associated tasks.
 INFO NodeHandler:
 INFO NodeHandler: Shutting down experiment, please wait...
 INFO NodeHandler:
 INFO run: Experiment default_slice-2025-04-18t04.39.37.163+00.00 finished after 0:5

brcortez@console:~$
```

```
brcortez@console:~$ omf stat -t all
/usr/share/omf-expctl-5.4/gems/gems/oml4r-2.9.5/lib/oml4r.rb:29: warning: already initialized constant
 OML4R::DEF_SERVER_PORT
/var/lib/gems/2.3.0/gems/oml4r-2.10.6/lib/oml4r.rb:33: warning: previous definition of DEF_SERVER_PORT
 was here
/usr/share/omf-expctl-5.4/gems/gems/oml4r-2.9.5/lib/oml4r.rb:30: warning: already initialized constant
 OML4R::DEF_PROTOCOL
/var/lib/gems/2.3.0/gems/oml4r-2.10.6/lib/oml4r.rb:34: warning: previous definition of DEF_PROTOCOL wa
s here

 INFO NodeHandler: OMF Experiment Controller 5.4 (git 861d645)
 INFO NodeHandler: Slice ID: default_slice (default)
 INFO NodeHandler: Experiment ID: default_slice-2025-04-18t04.43.10.234+00.00
 INFO NodeHandler: Message authentication is disabled
 INFO property.resetDelay: resetDelay = 300 (Fixnum)
 INFO property.resetTries: resetTries = 1 (Fixnum)
 INFO property.nodes: nodes = "system:topo:all" (String)
 INFO property.summary: summary = false (FalseClass)
 INFO Topology: Loaded topology 'system:topo:all'.

Talking to the CMC service, please wait
-----------------------------------------------
 Node: node1-1.outdoor.orbit-lab.org     State: POWERON
 Node: node1-10.outdoor.orbit-lab.org            State: POWEROFF
 Node: node1-2.outdoor.orbit-lab.org     State: POWERON
 Node: node1-3.outdoor.orbit-lab.org     State: POWERON
 Node: node1-4.outdoor.orbit-lab.org     State: POWERON
 Node: node1-7.outdoor.orbit-lab.org     State: POWEROFF
 Node: node1-9.outdoor.orbit-lab.org     State: POWEROFF
 Node: node2-10.outdoor.orbit-lab.org            State: POWEROFF
 Node: node2-2.outdoor.orbit-lab.org     State: POWEROFF
 Node: node2-5.outdoor.orbit-lab.org     State: POWEROFF
 Node: node2-8.outdoor.orbit-lab.org     State: POWEROFF
 Node: node4-3.outdoor.orbit-lab.org     State: POWEROFF
 Node: node4-4.outdoor.orbit-lab.org     State: POWEROFF
-----------------------------------------------

 INFO EXPERIMENT_DONE: Event triggered. Starting the associated tasks.
 INFO NodeHandler:
 INFO NodeHandler: Shutting down experiment, please wait...
 INFO NodeHandler:
 INFO run: Experiment default_slice-2025-04-18t04.43.10.234+00.00 finished after 0:5

brcortez@console:~$
```

- Verify the status of the nodes by navigating to the status page or using any appropriate commands.
    - Show successful SSH access to the 4 nodes
        - Node1-1

```
brcortez@console:~$ ssh root@node1-1
The authenticity of host 'node1-1 (10.40.1.1)' can't be established.
ECDSA key fingerprint is SHA256:Fd5wwSqlA4A6MmbbjE+cDnSslpDqd7l6rFCxs54x1Y4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'node1-1,10.40.1.1' (ECDSA) to the list of known hosts.
root@node1-1:~#
```

- Node1-2

```
brcortez@console:~$ ssh root@node1-2
The authenticity of host 'node1-2 (10.40.1.2)' can't be established.
ECDSA key fingerprint is SHA256:Fd5wwSqlA4A6MmbbjE+cDnSslpDqd7l6rFCxs54x1Y4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'node1-2,10.40.1.2' (ECDSA) to the list of known hosts.
root@node1-2:~#
```

- Node1-3

```
brcortez@console:~$ ssh root@node1-3
The authenticity of host 'node1-3 (10.40.1.3)' can't be established.
ECDSA key fingerprint is SHA256:Fd5wwSqlA4A6MmbbjE+cDnSslpDqd7l6rFCxs54x1Y4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'node1-3,10.40.1.3' (ECDSA) to the list of known hosts.
root@node1-3:~#
```

- Node1-4

```
brcortez@console:~$ ssh root@node1-4
The authenticity of host 'node1-4 (10.40.1.4)' can't be established.
ECDSA key fingerprint is SHA256:Fd5wwSqlA4A6MmbbjE+cDnSslpDqd7l6rFCxs54x1Y4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'node1-4,10.40.1.4' (ECDSA) to the list of known hosts.
root@node1-4:~#
```

3. Open Wi-Fi Network Configuration

- Start the wireless AP on the designated node.
    - Show terminal output after starting the wireless AP.

```
root@node1-1:~# hostapd hostapd-open.conf
Configuration file: hostapd-open.conf
Using interface wlan0 with hwaddr 00:0c:42:64:b2:64 and ssid "wifi-open"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
```

- Connect Alice and Bob to the open Wi-Fi network.
    - Show the output of iwconfig after Alice and Bob successfully connect to the AP as well as the successful connection on the AP terminal
        - AP

```
root@node1-1:~# hostapd hostapd-open.conf
Configuration file: hostapd-open.conf
Using interface wlan0 with hwaddr 00:0c:42:64:b2:64 and ssid "wifi-open"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
wlan0: STA 00:0c:42:64:b0:8d IEEE 802.11: authenticated
wlan0: STA 00:0c:42:64:b0:8d IEEE 802.11: associated (aid 1)
wlan0: AP-STA-CONNECTED 00:0c:42:64:b0:8d
wlan0: STA 00:0c:42:64:b0:8d RADIUS: starting accounting session 6801DA06-00000000
wlan0: STA 00:0c:42:64:b2:6c IEEE 802.11: authenticated
wlan0: STA 00:0c:42:64:b2:6c IEEE 802.11: associated (aid 2)
wlan0: AP-STA-CONNECTED 00:0c:42:64:b2:6c
wlan0: STA 00:0c:42:64:b2:6c RADIUS: starting accounting session 6801DA06-00000001
```

        - Bob

```
root@node1-2:~# iwconfig wlan0
wlan0     IEEE 802.11abgn  ESSID:"wifi-open"
          Mode:Managed  Frequency:2.437 GHz  Access Point: 00:0C:42:64:B2:64
          Bit Rate=1 Mb/s   Tx-Power=27 dBm
          Retry  long limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality=70/70  Signal level=-12 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:8   Missed beacon:0

root@node1-2:~#
```

        - Alice

```
root@node1-3:~# iwconfig wlan0
wlan0     IEEE 802.11abgn  ESSID:"wifi-open"
          Mode:Managed  Frequency:2.437 GHz  Access Point: 00:0C:42:64:B2:64
          Bit Rate=1 Mb/s   Tx-Power=27 dBm
          Retry  long limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality=36/70  Signal level=-74 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:7   Missed beacon:0

root@node1-3:~#
```

- Assign IP addresses to Alice and Bob.
    - Provide evidence that the IP addresses are successfully assigned.
        - Bob

```
root@node1-2:~# ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr 00:0c:42:64:b2:6c
          inet addr:192.168.0.4  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:42ff:fe64:b26c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:140 (140.0 B)  TX bytes:792 (792.0 B)

root@node1-2:~#
```

- Alice

```
root@node1-3:~# ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr 00:0c:42:64:b0:8d
          inet addr:192.168.0.3  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:42ff:fe64:b08d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:648 (648.0 B)  TX bytes:792 (792.0 B)

root@node1-3:~#
```

- Put Mallory in monitor mode and begin capturing traffic.
    - Show Mallory's capture using airodump-ng.

```
 CH  6 ][ Elapsed: 2 mins ][ 2025-04-18 01:03

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

 00:0C:42:64:B2:64  -53   0     1370       29    0   6  54   OPN              wifi-open

 BSSID              STATION            PWR   Rate    Lost   Packets  Probes

 00:0C:42:64:B2:64  00:0C:42:64:B0:8D  -48    1 -11      0       19
 00:0C:42:64:B2:64  00:0C:42:64:B2:6C  -56   48 -54      0       13
```

- Pass user data between Alice and Bob using netcat.
    - Show that the message typed on Alice or Bob appears in the terminal of the other.
        - Bob

```
root@node1-2:~# netcat -l 4000
Hello from Bob!
root@node1-2:~# ^C
root@node1-2:~#
```

        - Alice

```
root@node1-3:~# netcat 192.168.0.4 4000
Hello from Bob!
^C
root@node1-3:~#
```

- Can Mallory, the attacker, see the data in plaintext? What do you observe from the captured traffic? Use Wireshark to inspect the captured file and provide screenshots and analysis to back your claims. Provide screenshots from your capture to back up your answer.

In the open network Mallory can indeed see the messages sent between Alice and Bob in plaintext.

```
Frame 298: 100 bytes on wire (800 bits), 100 bytes captured (800 bits)
IEEE 802.11 Data, Flags: .......T
Logical-Link Control
Internet Protocol Version 4, Src: 192.168.0.4, Dst: 192.168.0.3
Transmission Control Protocol, Src Port: 4000, Dst Port: 34417, Seq: 1, Ack: 1, Len: 16
Data (16 bytes)
    Data: 48656c6c6f2066726f6d20426f62210a
    [Length: 16]

0000  08 01 2c 00 00 0c 42 64  b2 64 00 0c 42 64 b2 6c   ··,···Bd ·d··Bd·l
0010  00 0c 42 64 b0 8d b0 00  aa aa 03 00 00 00 08 00   ··Bd···· ········
0020  45 00 00 44 98 ce 40 00  40 06 20 8e c0 a8 00 04   E··D··@· @· ·····
0030  c0 a8 00 03 0f a0 86 71  97 32 70 56 62 2f 57 ab   ·······q ·2pVb/W·
0040  80 18 00 e3 11 85 00 00  01 01 08 0a 00 03 fe 51   ········ ·······Q
0050  00 03 e2 97 48 65 6c 6c  6f 20 66 72 6f 6d 20 42   ····Hell o from B
0060  6f 62 21 0a                                        ob!·
```

4. WEP Network Configuration

- Start the wireless AP for the WEP network.
    - Show terminal output after starting the WEP AP.

```
root@node1-1:~# hostapd hostapd-wep.conf
Configuration file: hostapd-wep.conf
Using interface wlan0 with hwaddr 00:0c:42:64:b2:64 and ssid "wifi-wep"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
```

- Connect Alice and Bob to the WEP network.
    - Show Alice and Bob's connection using iwconfig as well as the successful connection on the AP terminal.
        - AP

```
root@node1-1:~# hostapd hostapd-wep.conf
Configuration file: hostapd-wep.conf
Using interface wlan0 with hwaddr 00:0c:42:64:b2:64 and ssid "wifi-wep"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
Unsupported authentication algorithm (0)
handle_auth_cb: STA 00:0c:42:64:b2:6c not found
wlan0: STA 00:0c:42:64:b2:6c IEEE 802.11: authenticated
wlan0: STA 00:0c:42:64:b2:6c IEEE 802.11: associated (aid 1)
wlan0: AP-STA-CONNECTED 00:0c:42:64:b2:6c
wlan0: STA 00:0c:42:64:b2:6c RADIUS: starting accounting session 6801E235-00000000
Unsupported authentication algorithm (0)
handle_auth_cb: STA 00:0c:42:64:b0:8d not found
wlan0: STA 00:0c:42:64:b0:8d IEEE 802.11: authenticated
wlan0: STA 00:0c:42:64:b0:8d IEEE 802.11: associated (aid 2)
wlan0: AP-STA-CONNECTED 00:0c:42:64:b0:8d
wlan0: STA 00:0c:42:64:b0:8d RADIUS: starting accounting session 6801E235-00000001
```

        - Alice

```
root@node1-3:~# iwconfig
eth0      no wireless extensions.

eth1      no wireless extensions.

lo        no wireless extensions.

wlan0     IEEE 802.11abgn  ESSID:"wifi-wep"
          Mode:Managed  Frequency:2.437 GHz  Access Point: 00:0C:42:64:B2:64
          Bit Rate=1 Mb/s   Tx-Power=27 dBm
          Retry  long limit:7   RTS thr:off   Fragment thr:off
          Encryption key:3132-3334-35
          Power Management:off
          Link Quality=33/70  Signal level=-77 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:6   Missed beacon:0

root@node1-3:~#
```

- Bob

```
root@node1-2:~# iwconfig
eth0      no wireless extensions.

eth1      no wireless extensions.

lo        no wireless extensions.

wlan0     IEEE 802.11abgn  ESSID:"wifi-wep"
          Mode:Managed  Frequency:2.437 GHz  Access Point: 00:0C:42:64:B2:64
          Bit Rate=1 Mb/s   Tx-Power=27 dBm
          Retry  long limit:7   RTS thr:off   Fragment thr:off
          Encryption key:3132-3334-35
          Power Management:off
          Link Quality=70/70  Signal level=-18 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0

root@node1-2:~#
```

- Capture traffic on Mallory's node.
  - Show the Wireshark capture of WEP traffic on Mallory's node.

```
CH  6 ][ Elapsed: 5 mins ][ 2025-04-18 01:35

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

00:0C:42:64:B2:64  -56 100     3082       32    0   6  54   WEP  WEP         wifi-wep

BSSID              STATION            PWR   Rate    Lost  Packets  Probes

00:0C:42:64:B2:64  00:0C:42:64:B0:8D  -46    1 - 1     0       32
00:0C:42:64:B2:64  00:0C:42:64:B2:6C  -56   18 - 1     0       25
```

```
No.  Time      Source               Destination          Protocol  Length Info
  1 0.000000  Routerboardc_64:b2:… Broadcast            802.11    84 Beacon frame, SN=2889, FN=0, Flags=........, BI=100, SSID="wifi-wep"
  2 0.628234  Routerboardc_64:b0:… Routerboardc_64:b2:… 802.11    24 Null function (No data), SN=102, FN=0, Flags=.......T
  3 0.629258  Routerboardc_64:b0:… Routerboardc_64:b2:… 802.11    24 Null function (No data), SN=102, FN=0, Flags=....R..T
  4 0.629248                       Routerboardc_64:b0:… 802.11    10 Acknowledgement, Flags=........
  5 2.322536                       Ring_1b:2f:3f        802.11    10 Acknowledgement, Flags=........
  6 2.322536                       Ring_1b:2f:3f        802.11    10 Clear-to-send, Flags=........
  7 2.322537  RuckusWirele_10:bf:… Ring_1b:2f:3f        802.11    28 802.11 Block Ack, Flags=........
  8 2.360937                       Ring_1b:2f:3f        802.11    10 Clear-to-send, Flags=........
  9 2.360937  RuckusWirele_10:bf:… Ring_1b:2f:3f        802.11    28 802.11 Block Ack, Flags=........
 10 2.576488                       Ring_1b:2f:3f        802.11    10 Acknowledgement, Flags=........
 11 2.626665                       Ring_1b:2f:3f        802.11    10 Acknowledgement, Flags=........
 12 2.631785                       Ring_1b:2f:3f        802.11    10 Clear-to-send, Flags=........
 13 2.631785  RuckusWirele_10:bf:… Ring_1b:2f:3f        802.11    28 802.11 Block Ack, Flags=........
 14 2.838121                       Ring_1b:2f:3f        802.11    10 Acknowledgement, Flags=........

Frame 1: 84 bytes on wire (672 bits), 84 bytes captured (672 bits)
IEEE 802.11 Beacon frame, Flags: ........
IEEE 802.11 Wireless Management

0000  80 00 00 00 ff ff ff ff  ff ff 00 0c 42 64 b2 64   ········ ····Bd·d
0010  00 0c 42 64 b2 64 90 b4  80 91 da 12 00 00 00 00   ··Bd·d·· ········
0020  64 00 11 04 00 08 77 69  66 69 2d 77 65 70 01 08   d·····wi fi-wep··
0030  82 84 8b 96 0c 12 18 24  03 01 06 05 04 01 02 00   ·······$ ········
0040  00 2a 01 04 32 04 30 48  60 6c 7f 08 00 00 00 00   ·*··2·0H `l······
0050  00 00 00 40                                        ···@
```

- Analyze the captured traffic and attempt to decrypt it using Wireshark.
  - Show the decrypted WEP packets.

```
Frame 166: 110 bytes on wire (880 bits), 110 bytes captured (880 bits)
IEEE 802.11 Data, Flags: .p.....T
Logical-Link Control
Internet Protocol Version 4, Src: 192.168.0.3, Dst: 192.168.0.4
Transmission Control Protocol, Src Port: 34418, Dst Port: 4000, Seq: 1, Ack: 1, Len: 18
Data (18 bytes)

0000  aa aa 03 00 00 00 08 00  45 00 00 46 49 ca 40 00   ········ E··FI·@·
0010  40 06 6f 90 c0 a8 00 03  c0 a8 00 04 86 72 0f a0   @·o····· ·····r··
0020  10 e1 21 e3 8f c9 c0 35  80 18 00 e5 de b9 00 00   ··!····5 ········
0030  01 01 08 0a 00 0a 83 ad  00 0a 6e 29 48 65 6c 6c   ········ ··n)Hell
0040  6f 20 66 72 6f 6d 20 41  6c 69 63 65 21 0a         o from A lice!·

Frame (110 bytes)   Decrypted WEP data (78 bytes)
```

- Without knowing the WEP key, can Mallory see the data in plaintext? If Mallory later finds out the key, is she able to decrypt the traffic? Explain your findings.

Without the WEP key, Mallory cannot see the data in plaintext, packets appear encrypted in Wireshark. Once the key is known and added, the traffic can be decrypted, revealing the plaintext message. This shows that WEP's encryption is weak and offers little protection once the key is compromised.

5. WPA Network Configuration

- Start the wireless AP for the WPA network.
    - Show terminal output after starting the WPA AP.

```
root@node1-1:~# hostapd hostapd-wpa.conf
Configuration file: hostapd-wpa.conf
Using interface wlan0 with hwaddr 00:0c:42:64:b2:64 and ssid "wifi-wpa"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
```

- Provide the contents of the wpa.conf file

```
root@node1-2:~# cat wpa.conf
network={
        ssid="wifi-wpa"
        #psk="123456789"
        psk=ebe5f11342aedef8edcf53317352a6ac89699c9a0a5cc5c823101012590de6bb
}
root@node1-2:~#
```

- Connect Alice and Bob to the WPA network using wpa_supplicant.
    - Show Alice and Bob's connection using wpa_supplicant as well as the successful connection on the AP terminal.
        - AP

```
wlan0: STA 00:0c:42:64:b0:8d IEEE 802.11: authenticated
wlan0: STA 00:0c:42:64:b0:8d IEEE 802.11: associated (aid 1)
wlan0: AP-STA-CONNECTED 00:0c:42:64:b0:8d
wlan0: STA 00:0c:42:64:b0:8d RADIUS: starting accounting session 6801E74C-00000000
wlan0: STA 00:0c:42:64:b0:8d WPA: pairwise key handshake completed (WPA)
wlan0: STA 00:0c:42:64:b0:8d WPA: group key handshake completed (WPA)
wlan0: STA 00:0c:42:64:b2:6c IEEE 802.11: authenticated
wlan0: STA 00:0c:42:64:b2:6c IEEE 802.11: associated (aid 2)
wlan0: AP-STA-CONNECTED 00:0c:42:64:b2:6c
wlan0: STA 00:0c:42:64:b2:6c RADIUS: starting accounting session 6801E74C-00000001
wlan0: STA 00:0c:42:64:b2:6c WPA: pairwise key handshake completed (WPA)
wlan0: STA 00:0c:42:64:b2:6c WPA: group key handshake completed (WPA)
```

- Alice

```
root@node1-3:~# wpa_supplicant -iwlan0 -cwpa.conf -B
Successfully initialized wpa_supplicant
root@node1-3:~#
```

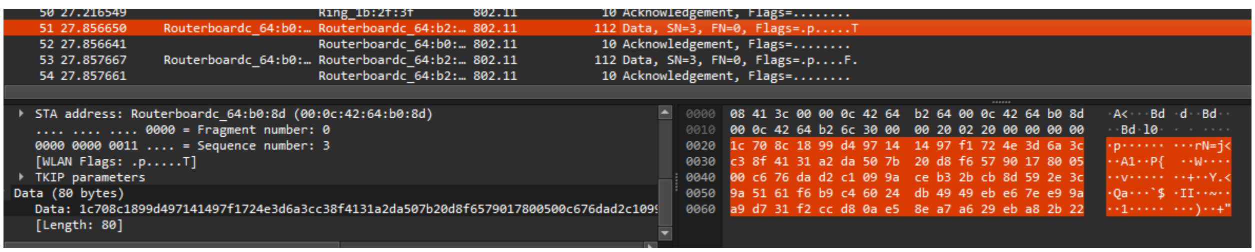
- Bob

```
root@node1-2:~# wpa_supplicant -iwlan0 -cwpa.conf -B
Successfully initialized wpa_supplicant
root@node1-2:~#
```

- Analyze the captured traffic in Wireshark.
    - Look for the data packets containing the secret message - is the attacker able to see the data in plaintext?

No, the attacker is not able to see the data in plaintext. Without the WEP key, the captured packets appear encrypted in Wireshark, and the message content is unreadable.

- Provide a screenshot showing whether Mallory can see any useful data without decryption.

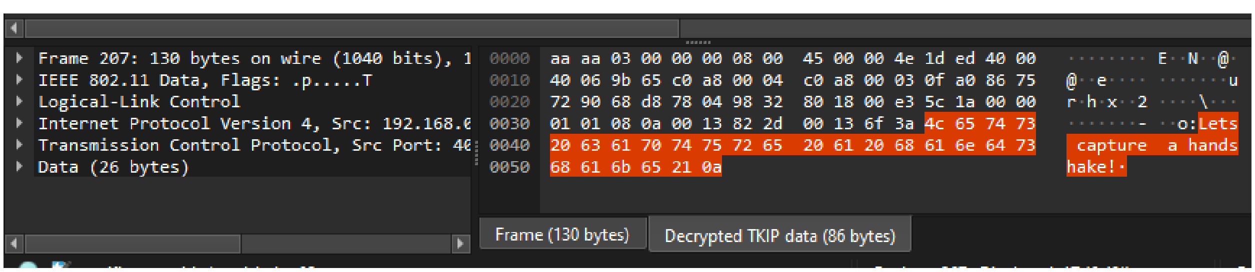


- Add the WPA passphrase to Wireshark's decryption settings and attempt to decrypt the captured traffic.

No Mallory is not able to read the message in plaintext after acquiring the WPA passphrase, the data still appears encrypted in Wireshark.

- Show the WPA decryption attempt in Wireshark and comment on whether the traffic is decrypted successfully after capturing the handshake.

After capturing the four-way handshake I was able to find the decrypted secret message in the Decrypted TKIP data field.

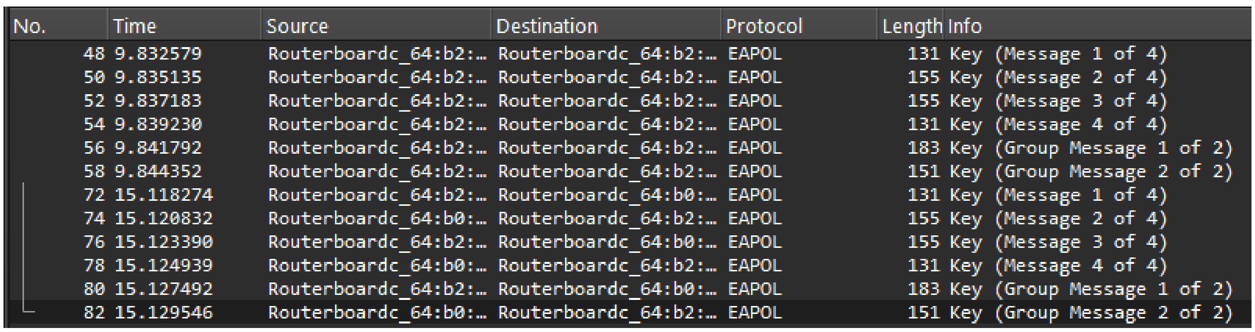


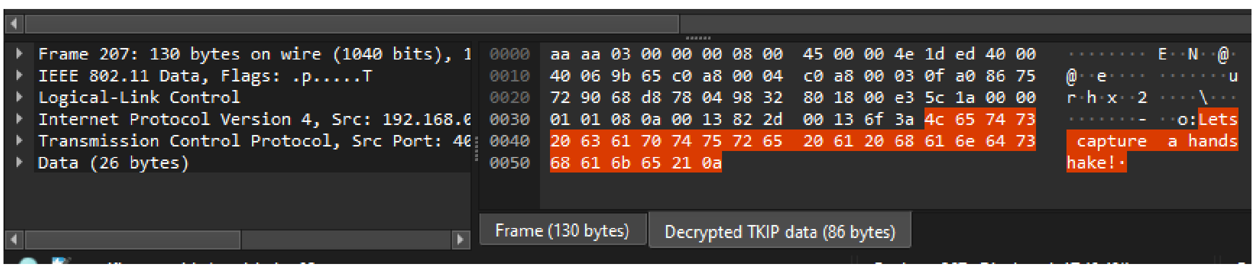
- Given the new scenario after capturing the WPA handshake, can Mallory decrypt the traffic? Look for the data packets containing the secret message, and specifically look for a "Decrypted TKIP data" tab on the bottom - is Mallory able to read the data if she has captured the 4-way handshakes?

Even after capturing the WPA 4-way handshake, Mallory cannot decrypt the traffic unless she also knows the correct WPA passphrase. In this scenario, if Mallory does not also have the passphrase, the captured packets do not show a "Decrypted TKIP data" section, confirming that the traffic remains encrypted. Therefore, the secret message cannot be read. This demonstrates that capturing the handshake alone is not sufficient for decryption without the shared key.

- Provide screenshots and analysis showing the 4-way handshake and whether the traffic is decrypted successfully after capturing the handshake.

After successfully capturing the four-way handshake we can read the secret message in plaintext since we also have the passphrase.





- Why does capturing the 4-way handshake help an attacker decrypt WPA traffic? What information does Mallory gain from the handshake?

Capturing the four-way handshake helps an attacker because it contains the information needed to calculate the temporary encryption key used for that session. If Mallory also knows the WPA passphrase, she can use the handshake to generate the same key the client and access point use to encrypt and decrypt data. This makes it possible to read the traffic.

6. Summary and Conclusions

Summarize the security implications for each network type (open, WEP, WPA). Discuss how each network performs in terms of confidentiality based on the experiment results. How secure is each network in terms of passive sniffing, and what lessons can be learned about the effectiveness of WEP and WPA encryption methods?

From the experiment, it's clear that each network type offers a different level of confidentiality.

- The **open** network had no encryption at all, allowing Mallory to easily capture and read plaintext messages through sniffing. This shows that open networks offer zero protection for user data.
- The **WEP** network, while encrypted, was still vulnerable. Without the key, Mallory couldn't read the messages initially. However, once the WEP key was known, the traffic was easily decrypted in Wireshark, proving that WEP encryption is weak and outdated. It provides only minimal protection and should not be relied on for secure communication.
- The **WPA** network was the most secure. Even after capturing the full 4-way handshake, Mallory couldn't decrypt the traffic without the correct WPA passphrase. This shows that WPA provides strong protection against passive sniffing as long as the passphrase remains secret.

Overall, the lab demonstrates that open and WEP networks are insecure, and WPA is significantly more effective at protecting user data.