

WiFi: Emulating Degraded Stations

Goal: Create a scenario where WiFi stations send out corrupted packets that can cause handshake and authentication failures. Learn techniques to capture and inspect packets to view corruption and scan log files to find indications of LANforge corruption injections.

We will learn to use some WiFi packet corruption features to emulate malfunctioning station equipment. This consists of enabling the corruption features and looking for errors when stations attempt to associate. Part of this will include capturing packets and inspecting them. This scenario requires LANforge version 5.3.6, and a two-radio LANforge system with one radio set in monitor mode.

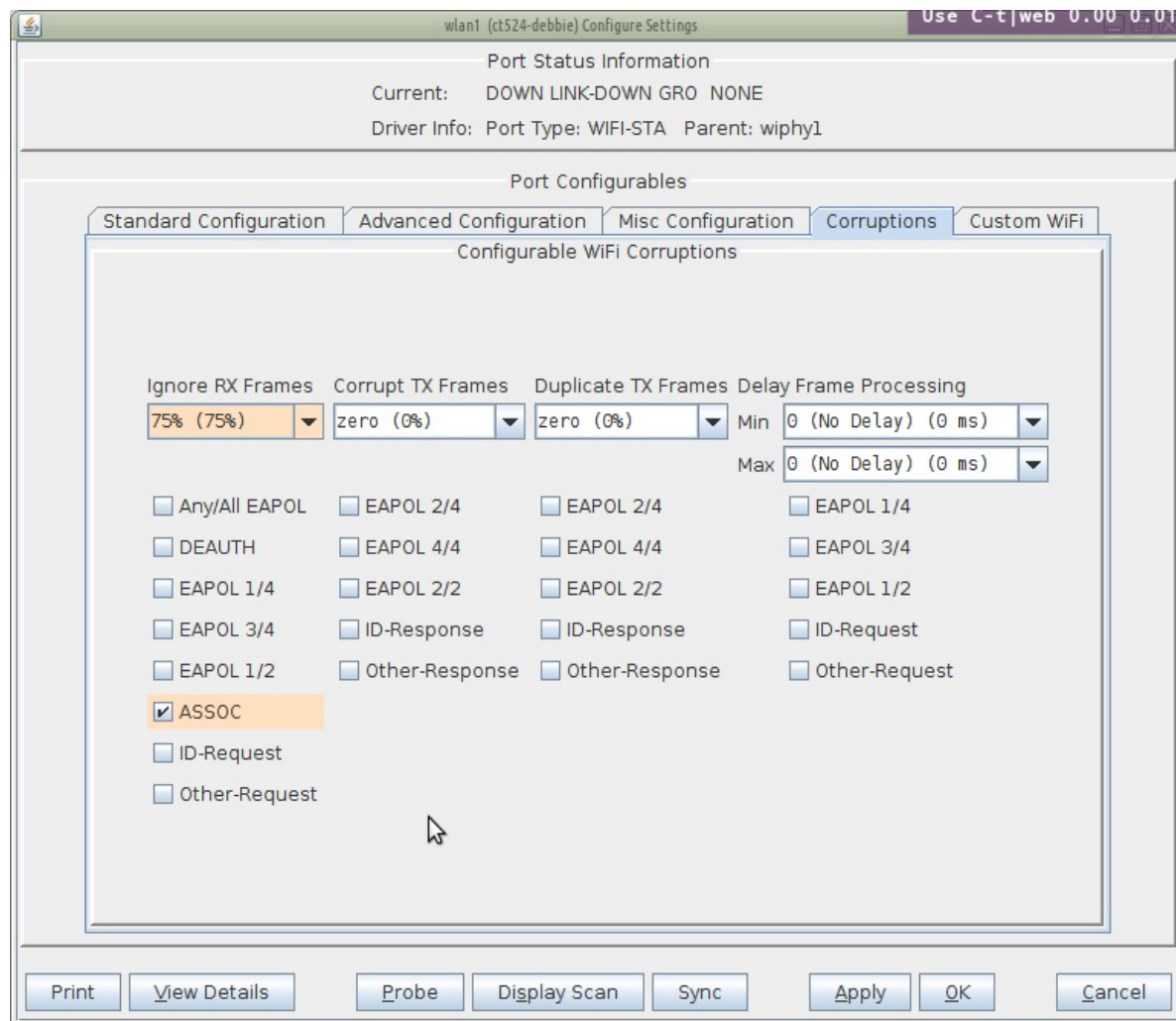


Open Authentication Test Scenario

We will begin with a basic open authentication test scenario with a single virtual station on a LANforge system connecting to an AP. (See Also: [Generating Traffic for WLAN Testing](#)) This test scenario also works for Captive Portal testing.

Create a Station with Corruptions

We will use the simplest form of corruption, ignored frames. Select the Port → Corruptions tab, and set Ignore RX Frames very high, like **75%**. To limit this to association frames, select **ASSOC**.



Associate a Station with Corruptions

Introducing corruptions is simple. Watching the effects takes some effort. With aggressive association corruptions, the basic effect will appear as if your station takes an unreasonably long time to associate. Let's set up Wireshark at different interfaces to understand better the kinds of traffic at play.

1. Select a radio for monitoring and set its channel to match the AP channel. If your AP is on channel 44, modify your LANforge wiphy monitor radio to also listen on channel 44. Our test AP is named hedtest-2000-open

Wiphy0 (ct524-debbie) Configure Settings

Port Status Information

Current: LINK-DOWN NONE
 Driver Info: Port Type: WIFI-Radio Driver: ath10k(988x) Bus: 0000:04:00.0

Port Configurables

Standard Configuration RF Patterns Firmware

Enable
☐ Set IF Down

General Interface Settings

☐ Down
 Alias:
 MAC Addr: 04:f0:21:20:37:03
 Rpt Timer: medium (8 s)

WiFi Settings

Max-VIFs: 64 Max-Stations: 127 Max-APs: 7 Supports: 802.11abgn-AC
 Country: United States (840)
 Channel/Freq: 44 (5220 Mhz) AP: DEFAULT
 Antenna: All (3x3) Tx-Power: DEFAULT (-1)
 RTS: DEFAULT Frag: 2346
☐ Verbose Debug

Print View Details Logs Probe Sync Apply OK Cancel

2. Create a **monitor** port (Port tab → Select Radio → Create → Monitor)

Create VLANs on Port: 1.3.06

1 ☐ MAC-VLAN ☐ 802.1Q-VLAN ☐ Redirect ☐ Bridge ☐ Bond ☐ GRE Tunnel
☐ WiFi STA ☐ WiFi VAP ☒ WiFi Monitor ☐ WiFi Virtual Radio

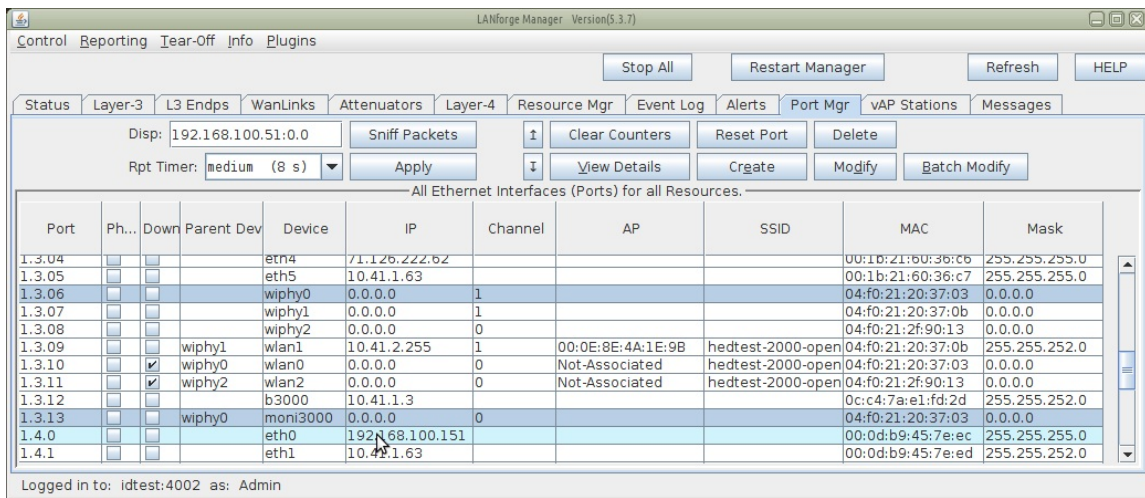
2 Shelf: 1 Resource: 3 (ct524-debbie) Port: 6 (wiphy0)

3 VLAN ID: ☐ DHCP-IPv4
 Parent MAC: 04:f0:21:20:37:03 DHCP Client ID: None
 MAC Addr: IP Address: Global IPv6: AUTO
 Quantity: 1 IP Mask or Bits: Link IPv6: AUTO
 Gateway IP: IPv6 GW: AUTO
 #1 Redir Name: #2 Redir Name:
 STA ID: 3000 SSID: hedtest-2000-open
 WiFi AP: Key/Phrase:
☐ WPA ☐ WPA2 ☐ WEP

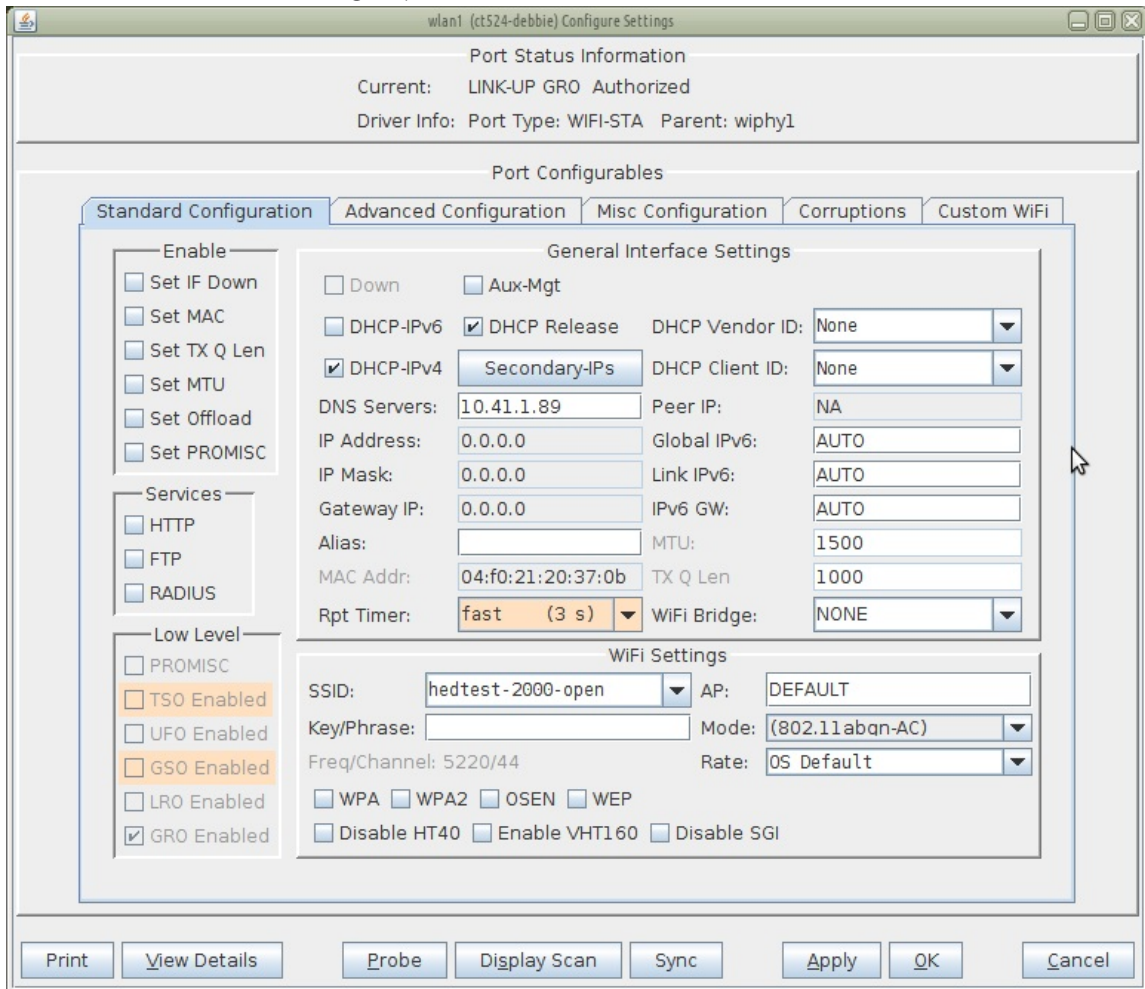
4 ☐ Down

Apply Cancel Ready

You will see the **moni3000** port below. The channel number will not display.



3. Create a station (wiphy1 → wlan1 will work) and associate it with hedtest-2000-open. Highlight the station in the Ports tab and bring it up.



Watch Traffic

With 75% ignore the chances of the station actually associating are very port. On our monitor interface, we see that there are multiple discover and request packets.

***moni3000 [Wireshark 2.1.1 (Git Rev Unknown from unknown)]**

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: wlan.addr == 04:f0:21:20:37:0b Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
798	15:50:37.604074540	Sparklan 4a:1e:9b (00:CompeXpt_20:37:0b (04:802.11	10.41.2.255	ICMPv6	76	802.11 Block Ack, Flags=.....
799	15:50:37.605264999	Sparklan 4a:1e:9b (00:CompeXpt_20:37:0b (04:802.11	10.41.2.255	ICMPv6	136	Router Solicitation from 04:f0:21:20:37:0b
819	15:50:41.185927522	10.41.2.255	10.41.1.89	DHCP	413	DHCP Release - Transaction ID 0x2477fc00
820	15:50:41.185979038	Sparklan 4a:1e:9b (00:CompeXpt_20:37:0b (04:802.11	10.41.2.255	DHCP	76	802.11 Block Ack, Flags=.....
821	15:50:41.189296093	10.41.2.255	10.41.1.89	DHCP	408	DHCP Release - Transaction ID 0x2477fc00
829	15:50:42.150757131	10.41.2.255	255.255.255.255	DHCP	408	DHCP Discover - Transaction ID 0x2477fc00
830	15:50:42.150764826	10.41.1.89	10.41.2.255	DHCP	422	DHCP Offer - Transaction ID 0x2477fc00
832	15:50:42.151282665	10.41.2.255	255.255.255.255	DHCP	413	DHCP Request - Transaction ID 0x2477fc00
833	15:50:42.151287980	Sparklan 4a:1e:9b (00:CompeXpt_20:37:0b (04:802.11	10.41.2.255	DHCP	76	802.11 Block Ack, Flags=.....
834	15:50:42.154495230	10.41.2.255	255.255.255.255	DHCP	408	DHCP Request - Transaction ID 0x2477fc00
835	15:50:42.180563925	10.41.1.89	10.41.2.255	DHCP	422	DHCP ACK - Transaction ID 0x2477fc00
864	15:50:47.367511641	Silicom 19:c9:dc	CompeXpt_20:37:0b	ARP	131	Who has 10.41.2.255? Tell 10.41.1.89
866	15:50:47.367759247	CompeXpt_20:37:0b	Silicom 19:c9:dc	ARP	113	10.41.2.255 is at 04:f0:21:20:37:0b
867	15:50:47.367810430	Sparklan 4a:1e:9b (00:CompeXpt_20:37:0b (04:802.11	10.41.2.255	DHCP	76	802.11 Block Ack, Flags=.....

Frame 829: 408 bytes on wire (3264 bits), 408 bytes captured (3264 bits) on interface 0

- Radiotap Header v0, Length 48
- 802.11 radio information
- IEEE 802.11 Data, Flags:F.
- Logical-Link Control
- Internet Protocol Version 4, Src: 10.41.2.255, Dst: 255.255.255.255
- User Datagram Protocol, Src Port: 68, Dst Port: 67
- Bootstrap Protocol (Discover)

```

0000  00 00 30 00 2f 40 00 a0 20 08 00 a0 20 08 00 a0  ..0./@.. ...
0010  20 08 00 00 00 00 00 00 14 fa 11 0d 00 00 00 00  .....
0020  00 02 6c 09 a0 00 c5 00 00 00 bf 00 c0 01 c0 02  ..l.....
0030  08 02 00 00 ff ff ff ff ff ff 00 0e 8e 4a 1e 9b  .....J..
0040  04 f0 21 20 37 0b 80 59 aa aa 03 00 00 00 08 00  ..! 7.Y .....
0050  45 10 01 48 00 00 00 00 80 11 2c 6e 0a 29 02 ff  E..H.... ,n.).

```

On our station interface (wlan1), we see a different number of DHCP requests.

***wlan1 [Wireshark 2.1.1 (Git Rev Unknown from unknown)]**

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
1	15:50:29.845257423	10.41.1.89	224.0.0.251	MDNS	87	Standard query 0x0000 PTR _ipps.tcp.local,
2	15:50:29.846360437	fe80::2e0:edff:fe19:c9dc	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipps.tcp.local,
3	15:50:37.603676577	fe80::6f0:21ff:fe20:370b	ff02::2	ICMPv6	70	Router Solicitation from 04:f0:21:20:37:0b
4	15:50:41.185646447	10.41.2.255	10.41.1.89	DHCP	342	DHCP Release - Transaction ID 0x2477fc00
5	15:50:42.146967571	10.41.2.255	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x2477fc00
6	15:50:42.150757374	10.41.1.89	10.41.2.255	DHCP	351	DHCP Offer - Transaction ID 0x2477fc00
7	15:50:42.150862602	10.41.2.255	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x2477fc00
8	15:50:42.180569682	10.41.1.89	10.41.2.255	DHCP	351	DHCP ACK - Transaction ID 0x2477fc00
9	15:50:47.367515151	Silicom 19:c9:dc	CompeXpt_20:37:0b	ARP	60	Who has 10.41.2.255? Tell 10.41.1.89
10	15:50:47.367541031	CompeXpt_20:37:0b	Silicom 19:c9:dc	ARP	42	10.41.2.255 is at 04:f0:21:20:37:0b

Frame 5: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface 0

- Ethernet II, Src: CompeXpt_20:37:0b (04:f0:21:20:37:0b), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 - Destination: Broadcast (ff:ff:ff:ff:ff:ff)
 - Source: CompeXpt_20:37:0b (04:f0:21:20:37:0b)
 - Type: IPv4 (0x0800)
- Internet Protocol Version 4, Src: 10.41.2.255, Dst: 255.255.255.255
- User Datagram Protocol, Src Port: 68, Dst Port: 67
- Bootstrap Protocol (Discover)
 - Message type: Boot Request (1)
 - Hardware type: Ethernet (0x01)
 - Hardware address length: 6
 - Hops: 0
 - Transaction ID: 0x2477fc00
 - Seconds elapsed: 0
 - Bootp flags: 0x0000 (Unicast)
 - Client IP address: 0.0.0.0
 - Your (client) IP address: 0.0.0.0
 - Next server IP address: 0.0.0.0
 - Relay agent IP address: 0.0.0.0
 - Client MAC address: CompeXpt_20:37:0b (04:f0:21:20:37:0b)

```

0000  ff ff ff ff ff ff 04 f0 21 20 37 0b 08 00 45 10  ..... ! 7...E.
0010  01 48 00 00 00 00 00 11 2c 6e 0a 29 02 ff ff ff  ..H..... ,n.).
0020  ff ff 00 44 00 43 01 34 0d bf 01 01 06 00 24 77  ...D.C.4 .....$w
0030  fc 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0040  00 00 00 00 00 00 00 04 f0 21 20 37 0b 00 00 00  ..... ! 7....
0050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....

```

At our AP, we see repeated attempts.

wlan2 (cts24-debbie) Configure Settings

Port Status Information

Current: LINK-UP GRO Authorized

Driver Info: Port Type: WIFI-STA Parent: wiphy2

Port Configurables

Standard Configuration | Advanced Configuration | Misc Configuration | Corruptions | Custom WiFi

Enable

☐ Set IF Down

☐ Set MAC

☐ Set TX Q Len

☐ Set MTU

☐ Set Offload

☐ Set PROMISC

Services

☐ HTTP

☐ FTP

☐ RADIUS

Low Level

☐ PROMISC

☐ TSO Enabled

☐ UFO Enabled

☐ GSO Enabled

☐ LRO Enabled

☒ GRO Enabled

General Interface Settings

☐ Down ☐ Aux-Mgt

☐ DHCP-IPv6 ☒ DHCP Release DHCP Vendor ID: None

☒ DHCP-IPv4 Secondary-IPs DHCP Client ID: None

DNS Servers: 10.41.1.89 Peer IP: NA

IP Address: 0.0.0.0 Global IPv6: AUTO

IP Mask: 0.0.0.0 Link IPv6: AUTO

Gateway IP: 0.0.0.0 IPv6 GW: AUTO

Alias: MTU: 1500

MAC Addr: 04:f0:21:2f:90:13 TX Q Len: 1000

Rpt Timer: fast (3 s) WiFi Bridge: NONE

WiFi Settings

SSID: hedtest-2100-wpa2 AP: DEFAULT

Key/Phrase: hedtest-2100-wpa2 Mode: (802.11abgn-AC)

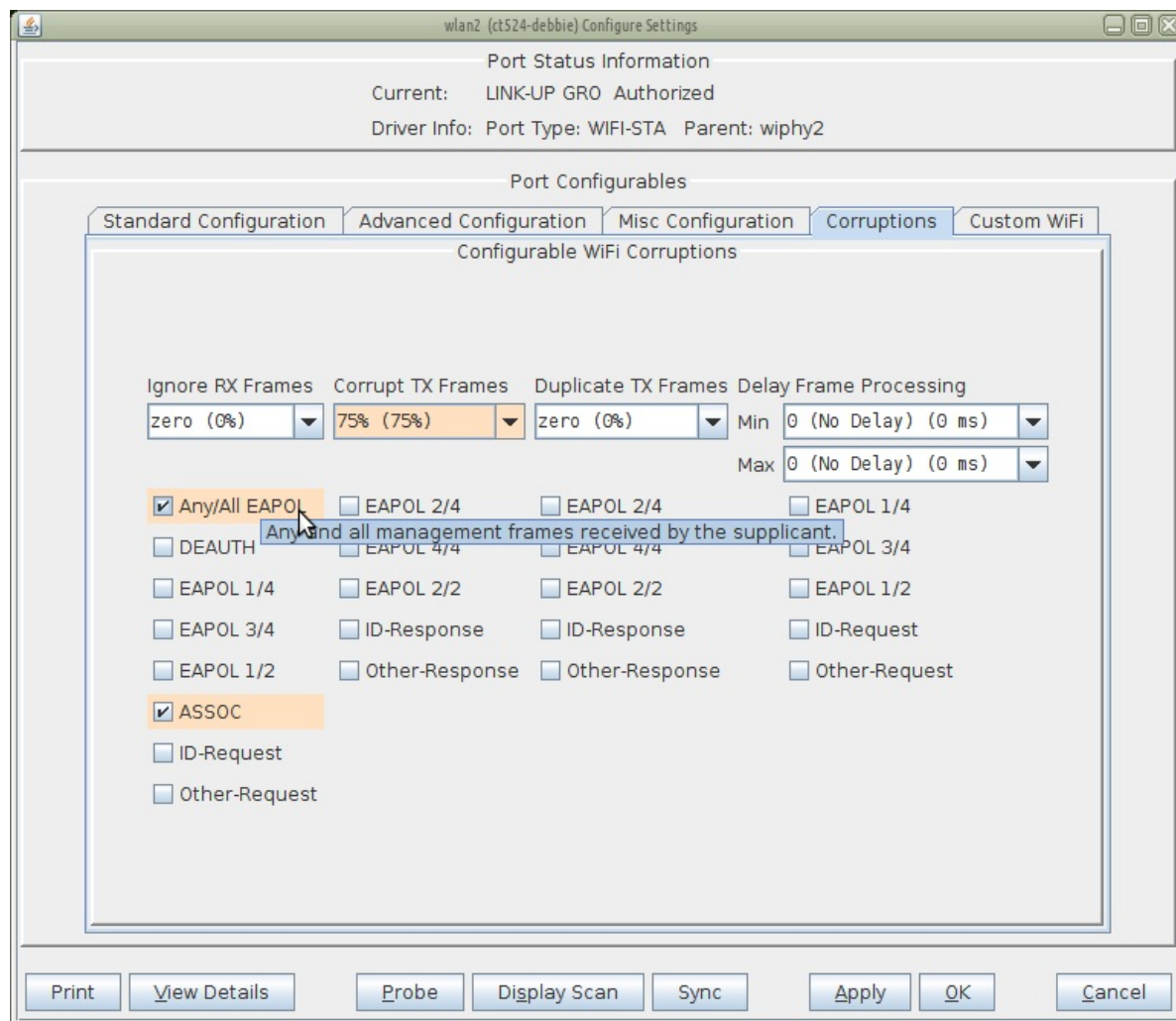
Freq/Channel: 5220/44 Rate: OS Default

☐ WPA ☒ WPA2 ☐ OSEN ☐ WEP

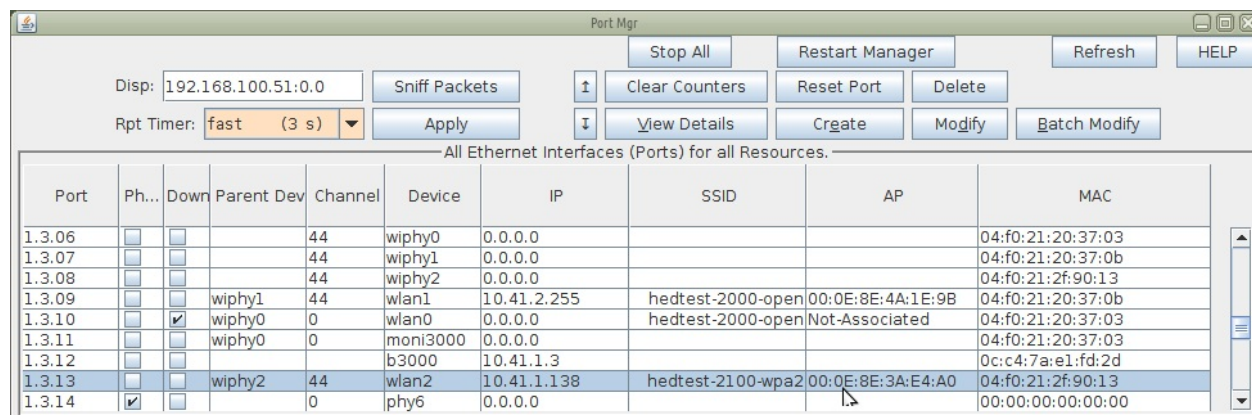
☐ Disable HT40 ☐ Enable VHT160 ☐ Disable SGI

Print View Details Probe Display Scan Sync Apply OK Cancel

On the Corruptions tab, you can set stations:



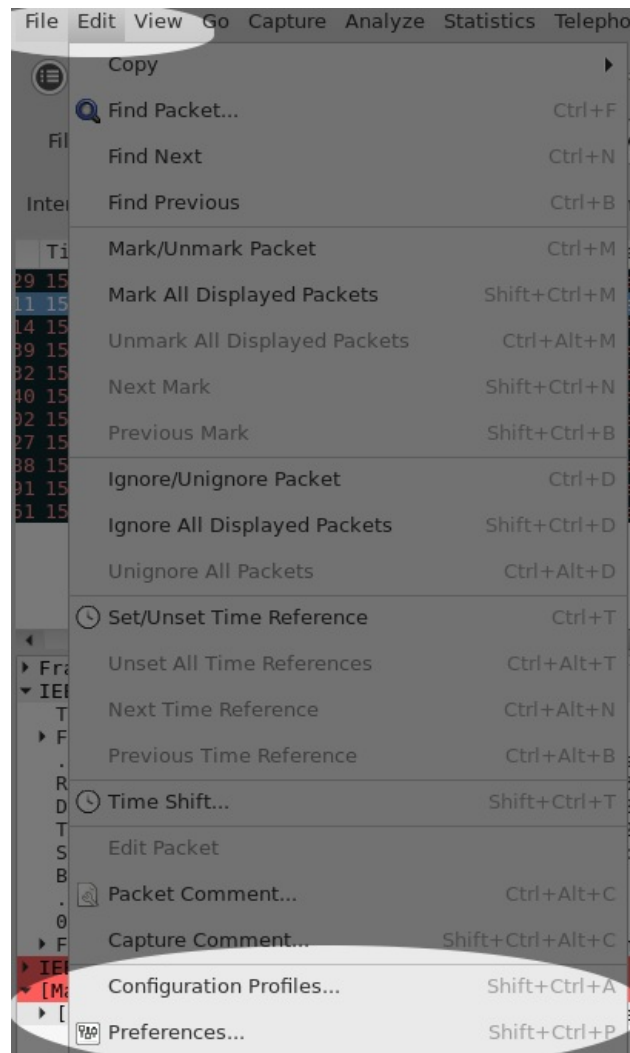
Highlight and activate the station:



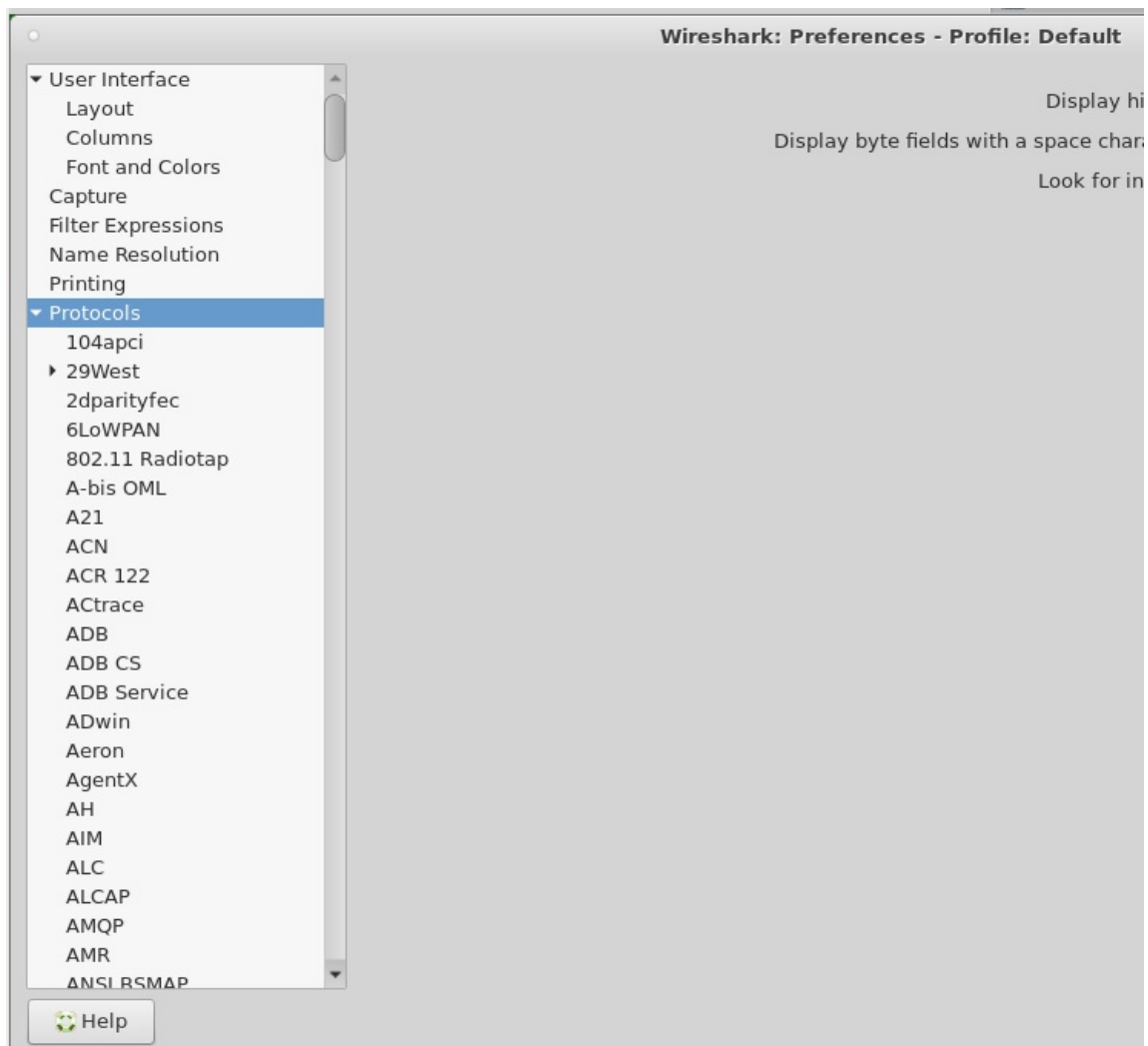
Setup Packet Decryption

We will not be able to inspect packets unless we configure Wireshark to decrypt the packets from this capture. Follow these steps:

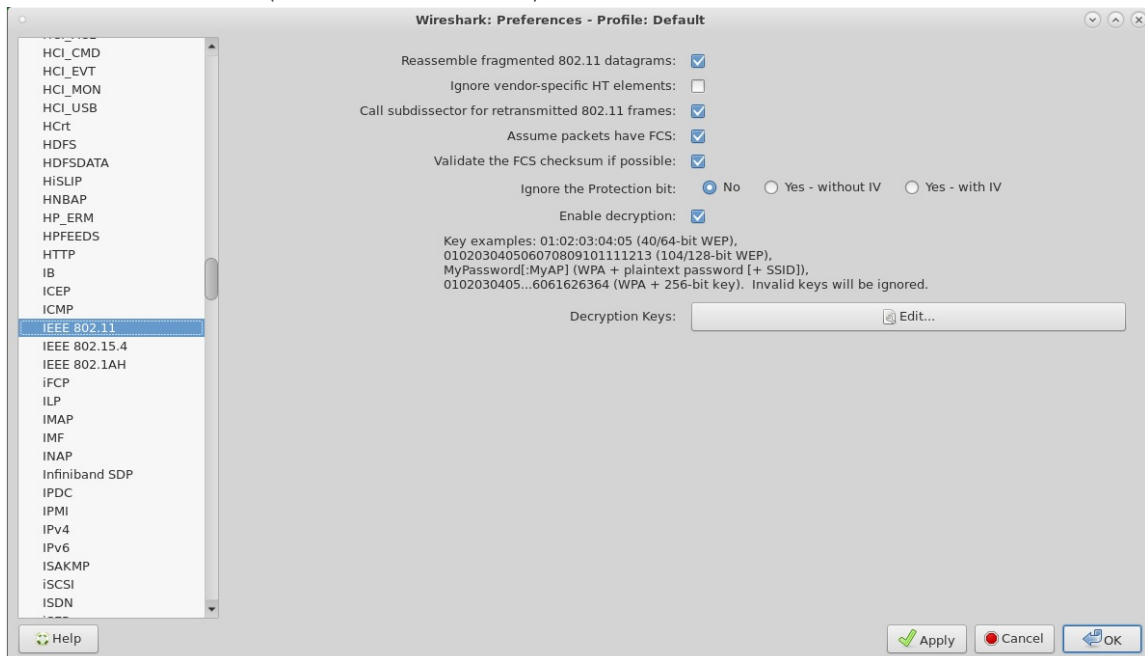
1. Wireshark Preferences (Edit → Preferences)



2. Protocol Preferences (Preferences → Protocols)



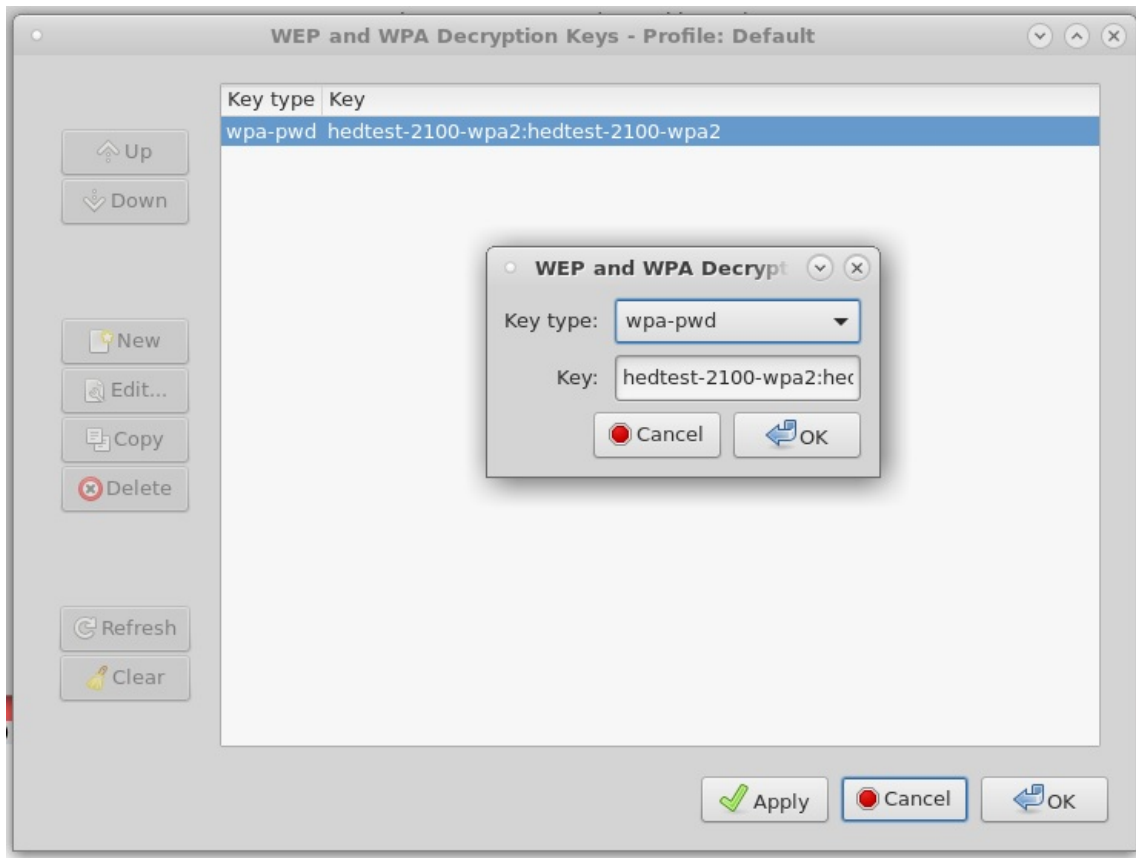
3. IEEE 802.11 Preferences (Protocols → IEEE 802.11)



- Select **Reassemble Fragmented 802.11 datagrams**
- Select **Call subdissector for retransmitted 802.11 frames**
- Select **Assume Packets have FCS**
- Select **Validate the FCS Checksum if possible**
- Select **No** for Ignore the Protection bit
- Select **Enable Decryption**

Add the SSID and password to the list of Decryption Keys. Select **wpa-pwd**: and type in the string **\$SSID:\$PSK** to match your password from your AP:

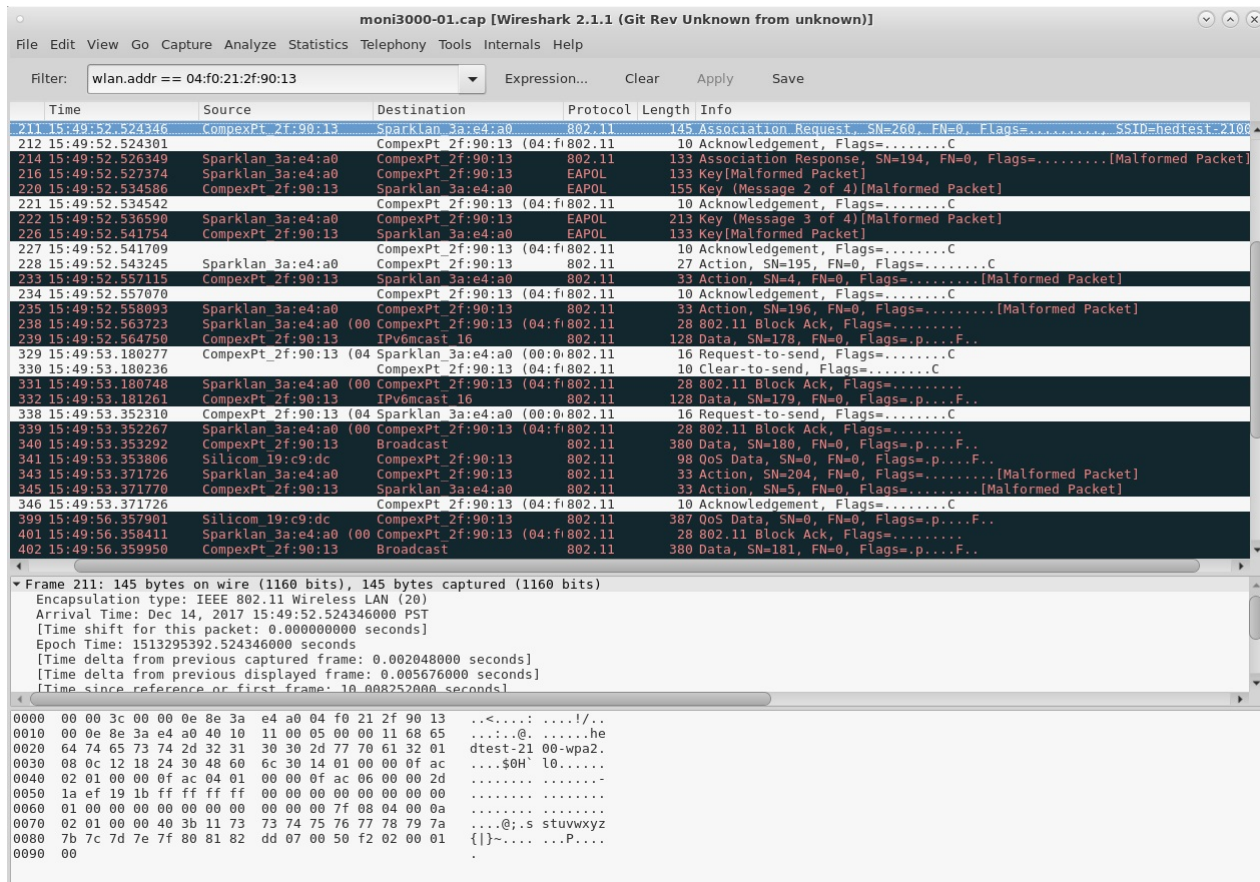
```
hedtest-2100-wpa2:hedtest-2100-wpa2
```



With our decryption enabled we can now inspect the captured packets:

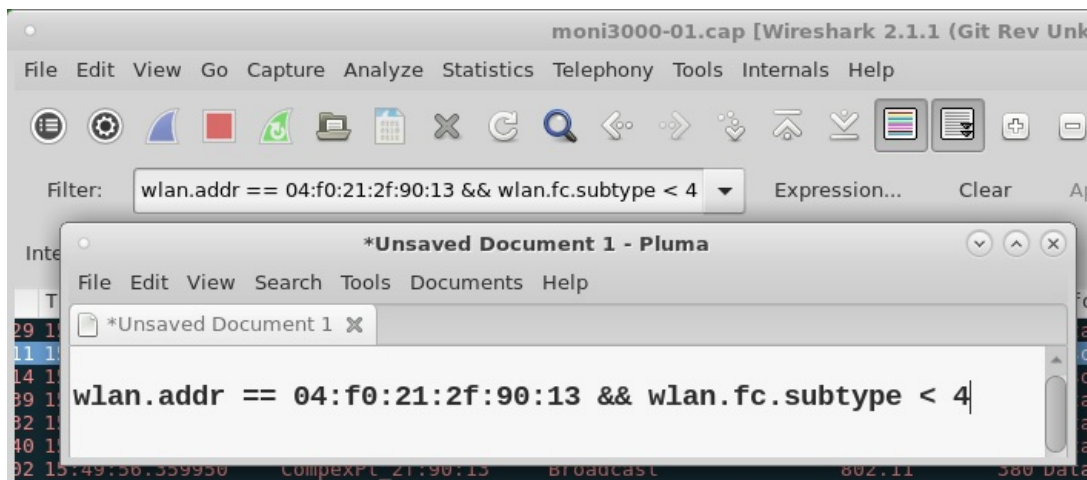
Packets from **moni3000** are filtered to show the station of interest using the display filter:

```
wlan.addr == 04:f0:21:2f:90:31
```

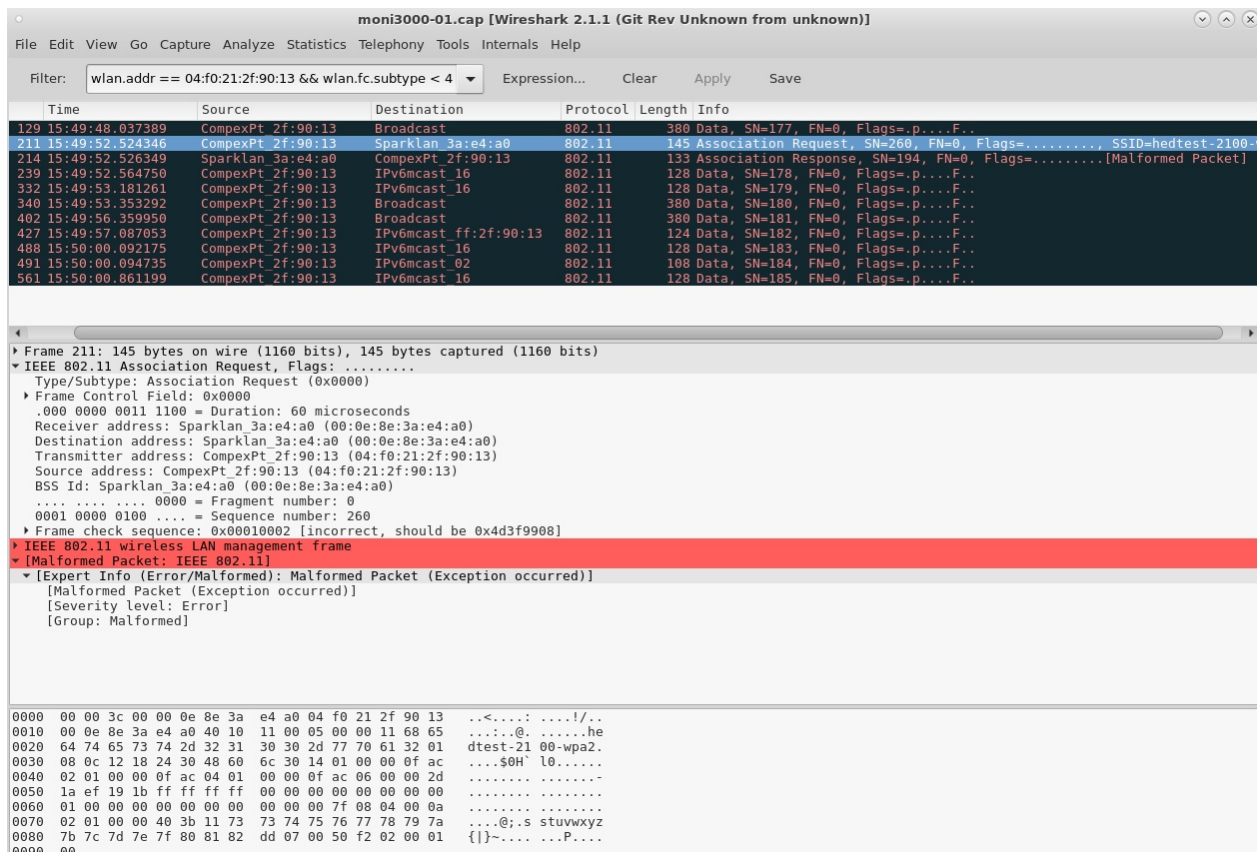


We can focus into the association by using a display filter:

```
wlan.addr == 04:f0:21:2f:90:31 && wlan.fc.subtype < 4
```



And the packets that form the association:



Notice how the AP only really gets the uncorrupted packets. What we hope will happen is that our WiFi drivers will discard corrupted packets before passing them up to userspace.

We did not need to decrypt packets from our AP (only possible if monitoring inside the AP) **vap2100:**

