802.11 Denial of Service

Related PCAP File(s):

- `deauthandbadauth.cap`

In the domain of traffic analysis, it is invariably critical to scrutinize all aspects of link-layer protocols and communications. A prominent type of link-layer attack is the one directed at `802.11 (Wi-Fi)`. Such an attack vector is often easy for us to disregard, but given that human errors can lead to the failure of our perimeter security, it is essential that we continually audit our wireless networks.

Capturing 802.11 Traffic

To examine our 802.11 raw traffic, we would require a `WIDS`/`WIPS` system or a wireless interface equipped with monitor mode. Similar to promiscuous mode in Wireshark, monitor mode permits us to view raw 802.11 frames and other packet types which might otherwise remain invisible.

Let's assume we do possess a Wi-Fi interface capable of monitor mode. We could enumerate our wireless interfaces in Linux using the following command:

Wireless Interfaces

802.11 Denial of Service

```
MisaelMacias@htb[/htb]$ iwconfig

wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short  long limit:2   RTS thr:off   Fragment thr:off
          Power Management:off
```

We have a couple of options to set our interface into monitor mode. Firstly, employing `airodump-ng`, we can use the ensuing command:

Airmon-NG

802.11 Denial of Service

```
MisaelMacias@htb[/htb]$ sudo airmon-ng start wlan0

Found 2 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
    820 NetworkManager
   1389 wpa_supplicant

PHY     Interface       Driver          Chipset

phy0    wlan0           rt2800usb       Ralink Technology, Corp. RT2870/RT3070
                (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
                (mac80211 station mode vif disabled for [phy0]wlan0)
```

Secondly, using system utilities, we would need to deactivate our interface, modify its mode, and then reactivate it.

Monitor Mode

Resources

Go to Questions

Table of Contents

Introduction

Intermediate Network Traffic Analysis Overview

Link Layer Attacks

ARP Spoofing & Abnormality Detection

ARP Scanning & Denial-of-Service

802.11 Denial-of-Service

Rogue Access Point & Evil-Twin Attacks

Detecting Network Abnormalities

Fragmentation Attacks

IP Source & Destination Spoofing Attacks

IP Time-to-Live Attacks

TCP Handshake Abnormalities

TCP Connection Resets & Hijacking

ICMP Tunneling

Application Layer Attacks

HTTP/HTTPs Service Enumeration Detection

Strange HTTP Headers

Cross-Site Scripting (XSS) & Code Injection Detection

SSL Renegotiation Attacks

Peculiar DNS Traffic

Strange Telnet & UDP Connections

Skills Assessment

Skills Assessment

My Workstation

OFFLINE

Start Instance

∞ / 1 spawns left

802.11 Denial of Service

```
MisaelMacias@htb[/htb]$ sudo ifconfig wlan0 down
MisaelMacias@htb[/htb]$ sudo iwconfig wlan0 mode monitor
MisaelMacias@htb[/htb]$ sudo ifconfig wlan0 up
```

We could verify if our interface is in `monitor mode` using the `iwconfig` utility.

802.11 Denial of Service

```
MisaelMacias@htb[/htb]$ iwconfig

wlan0mon  IEEE 802.11  Mode:Monitor  Frequency:2.457 GHz  Tx-Power=20 dBm
          Retry short  long limit:2   RTS thr:off   Fragment thr:off
          Power Management:off
```

It's possible that our interface doesn't conform to the `wlan0mon` convention. Instead, it might bear a name such as the following.

802.11 Denial of Service

```
MisaelMacias@htb[/htb]$ iwconfig

wlan0     IEEE 802.11  Mode:Monitor  Frequency:2.457 GHz  Tx-Power=20 dBm
          Retry short  long limit:2   RTS thr:off   Fragment thr:off
          Power Management:off
```

The crucial factor here is that the mode should be "monitor". The name of the interface isn't particularly important, and in many cases, our Linux distribution might assign it a completely different name.

To commence capturing traffic from our clients and network, we can employ `airodump-ng`. We need to specify our AP's channel with `-c`, its BSSID with `--bssid`, and the output file name with `-w`.

802.11 Denial of Service

```
MisaelMacias@htb[/htb]$ sudo airodump-ng -c 4 --bssid F8:14:FE:4D:E6:F1 wlan0 -w raw

BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID
F8:14:FE:4D:E6:F1  -23  64      115        6    0   4  130   WPA2 CCMP   PSK  HTB-Wireless
```

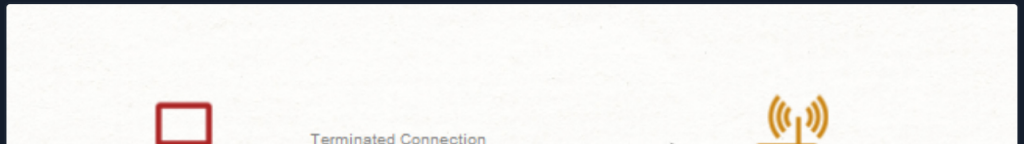
We can use `tcpdump` to achieve similar outcomes, but airodump-ng proves equally effective.

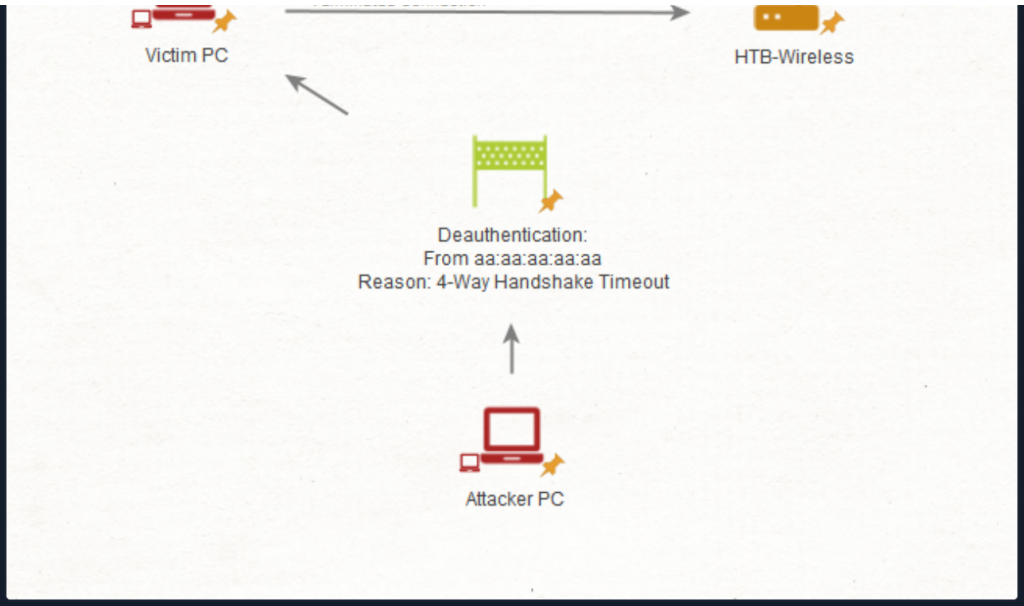
How Deauthentication Attacks Work

Among the more frequent attacks we might witness or detect is a deauthentication/dissociation attack. This is a commonplace link-layer precursor attack that adversaries might employ for several reasons:

1. To capture the WPA handshake to perform an offline dictionary attack
2. To cause general denial of service conditions
3. To enforce users to disconnect from our network, and potentially join their network to retrieve information

In essence, the attacker will fabricate an 802.11 deauthentication frame pretending it originates from our legitimate access point. By doing so, the attacker might manage to disconnect one of our clients from the network. Often, the client will reconnect and go through the handshake process while the attacker is sniffing.





This attack operates by the attacker spoofing or altering the MAC of the frame's sender. The client device cannot really discern the difference without additional controls like IEEE 802.11w (Management Frame Protection). Each deauthentication request is associated with a reason code explaining why the client is being disconnected.

In most scenarios, basic tools like `aireplay-ng` and `mdk4` employ reason `code 7` for deauthentication.

Finding Deauthentication Attacks

To detect these potential attacks, we can open the related traffic capture file (`deauthandbadauth.cap`) as shown below.

Wireshark

802.11 Denial of Service

```
MisaelMacias@htb[/htb]$ sudo wireshark deauthandbadauth.cap
```

If we wanted to limit our view to traffic from our AP's BSSID (`MAC`), we could use the following Wireshark filter:

- `wlan.bssid == xx:xx:xx:xx:xx:xx`

wlan.bssid == F8:14:FE:4D:E6:F1

No.	Time	Source	Destination	Protocol	Length	Info
358	52.755310	Unionman_4d:e6:f1	Vizio_4f:3d:54	802.11	395	Probe Response, SN=3538, FN=0, Flags=........, BI=100, SSID="HTB-Wireless"
360	52.792727	Unionman_4d:e6:f1	Vizio_4f:3d:54	802.11	395	Probe Response, SN=3539, FN=0, Flags=........, BI=100, SSID="HTB-Wireless"
362	53.676082	Unionman_4d:e6:f1	SichuanA_fd:91:e5	802.11	395	Probe Response, SN=3542, FN=0, Flags=........, BI=100, SSID="HTB-Wireless"
365	53.811709	Unionman_4d:e6:f1	SichuanA_fd:91:e5	802.11	395	Probe Response, SN=3544, FN=0, Flags=........, BI=100, SSID="HTB-Wireless"
367	59.909951	Unionman_4d:e6:f1	SichuanA_fd:91:e5	802.11	395	Probe Response, SN=3545, FN=0, Flags=........, BI=100, SSID="HTB-Wireless"
369	59.913389	Unionman_4d:e6:f1	SichuanA_fd:91:e5	802.11	395	Probe Response, SN=3546, FN=0, Flags=........, BI=100, SSID="HTB-Wireless"
371	60.085754	Unionman_4d:e6:f1	SichuanA_fd:91:e5	802.11	395	Probe Response, SN=3547, FN=0, Flags=........, BI=100, SSID="HTB-Wireless"
372	60.279133	Unionman_4d:e6:f1	Vizio_4f:3d:54	802.11	395	Probe Response, SN=3548, FN=0, Flags=........, BI=100, SSID="HTB-Wireless"
374	62.421343	Unionman_4d:e6:f1	IntelCor_af:eb:91	802.11	395	Probe Response, SN=3551, FN=0, Flags=........, BI=100, SSID="HTB-Wireless"
376	62.792619	Unionman_4d:e6:f1	4a:b1:75:42:6c:24	802.11	395	Probe Response, SN=3553, FN=0, Flags=........, BI=100, SSID="HTB-Wireless"
378	62.796637	Unionman_4d:e6:f1	4a:b1:75:42:6c:24	802.11	395	Probe Response, SN=3554, FN=0, Flags=........, BI=100, SSID="HTB-Wireless"
379	62.897804	Unionman_4d:e6:f1	4a:b1:75:42:6c:24	802.11	395	Probe Response, SN=3555, FN=0, Flags=........, BI=100, SSID="HTB-Wireless"
381	62.901192	Unionman_4d:e6:f1	4a:b1:75:42:6c:24	802.11	395	Probe Response, SN=3556, FN=0, Flags=........, BI=100, SSID="HTB-Wireless"
383	63.004809	Unionman_4d:e6:f1	4a:b1:75:42:6c:24	802.11	395	Probe Response, SN=3557, FN=0, Flags=........, BI=100, SSID="HTB-Wireless"
385	63.098665	Unionman_4d:e6:f1	SichuanA_fd:91:e5	802.11	395	Probe Response, SN=3558, FN=0, Flags=........, BI=100, SSID="HTB-Wireless"
387	63.102113	Unionman_4d:e6:f1	SichuanA_fd:91:e5	802.11	395	Probe Response, SN=3559, FN=0, Flags=........, BI=100, SSID="HTB-Wireless"
389	63.284871	Unionman_4d:e6:f1	SichuanA_fd:91:e5	802.11	395	Probe Response, SN=3560, FN=0, Flags=........, BI=100, SSID="HTB-Wireless"
390	63.289297	Unionman_4d:e6:f1	SichuanA_fd:91:e5	802.11	395	Probe Response, SN=3561, FN=0, Flags=........, BI=100, SSID="HTB-Wireless"
391	67.746736	Unionman_4d:e6:f1	Vizio_4f:3d:54	802.11	395	Probe Response, SN=3564, FN=0, Flags=........, BI=100, SSID="HTB-Wireless"
393	67.766608	Unionman_4d:e6:f1	Vizio_4f:3d:54	802.11	395	Probe Response, SN=3565, FN=0, Flags=........, BI=100, SSID="HTB-Wireless"
395	67.788240	Unionman_4d:e6:f1	Vizio_4f:3d:54	802.11	395	Probe Response, SN=3566, FN=0, Flags=........, BI=100, SSID="HTB-Wireless"
397	67.808359	Unionman_4d:e6:f1	Vizio_4f:3d:54	802.11	395	Probe Response, SN=3567, FN=0, Flags=........, BI=100, SSID="HTB-Wireless"
399	68.808478	Unionman_4d:e6:f1	MurataMa_bd:2d:3f	802.11	395	Probe Response, SN=3572, FN=0, Flags=........, BI=100, SSID="HTB-Wireless"

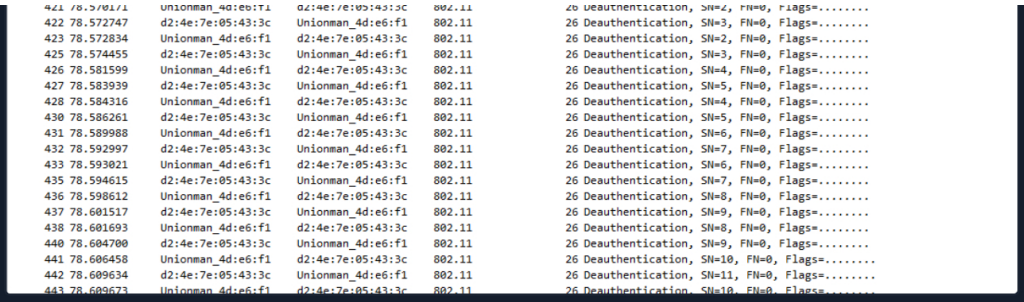
Suppose we wanted to take a look at the deauthentication frames from our BSSID or an attacker pretending to send these from our BSSID, we could use the following Wireshark filter:

- `(wlan.bssid == xx:xx:xx:xx:xx:xx) and (wlan.fc.type == 00) and (wlan.fc.type_subtype == 12)`

With this filter, we specify the type of frame (`management`) with `00` and the subtype (`deauthentication`) with `12`.

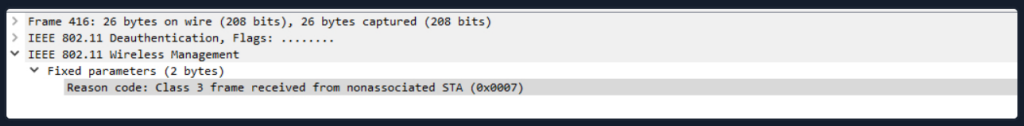
(wlan.bssid == F8:14:FE:4D:E6:F1) and (wlan.fc.type == 00) and (wlan.fc.type_subtype == 12)

No.	Time	Source	Destination	Protocol	Length	Info
416	78.561456	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=0, FN=0, Flags=........
417	78.565783	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=1, FN=0, Flags=........
418	78.565801	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=0, FN=0, Flags=........
420	78.566384	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=1, FN=0, Flags=........



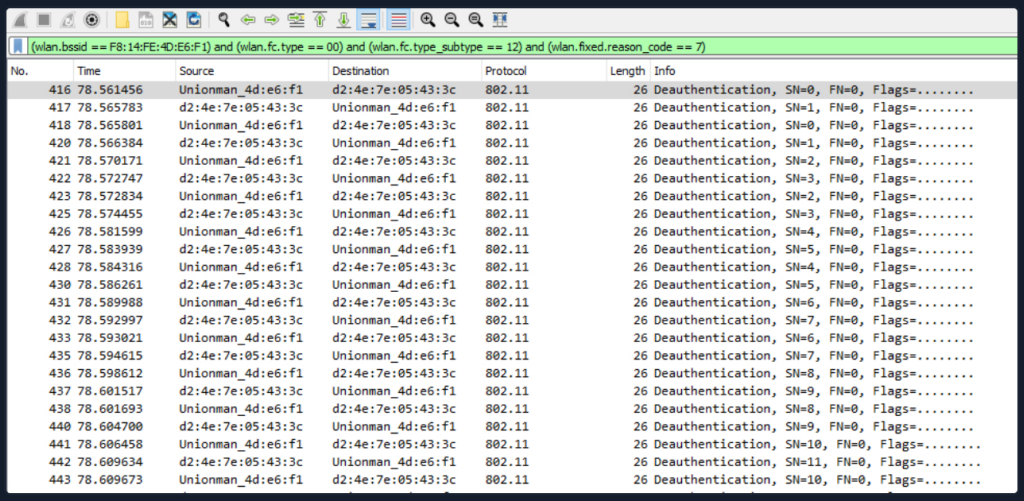
422	78.572747	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26 Deauthentication, SN=3, FN=0, Flags=........
423	78.572834	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26 Deauthentication, SN=2, FN=0, Flags=........
425	78.574455	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26 Deauthentication, SN=3, FN=0, Flags=........
426	78.581599	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26 Deauthentication, SN=4, FN=0, Flags=........
427	78.583939	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26 Deauthentication, SN=5, FN=0, Flags=........
428	78.584316	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26 Deauthentication, SN=4, FN=0, Flags=........
430	78.586261	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26 Deauthentication, SN=5, FN=0, Flags=........
431	78.589988	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26 Deauthentication, SN=6, FN=0, Flags=........
432	78.592997	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26 Deauthentication, SN=7, FN=0, Flags=........
433	78.593021	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26 Deauthentication, SN=6, FN=0, Flags=........
435	78.594615	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26 Deauthentication, SN=7, FN=0, Flags=........
436	78.598612	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26 Deauthentication, SN=8, FN=0, Flags=........
437	78.601517	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26 Deauthentication, SN=9, FN=0, Flags=........
438	78.601693	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26 Deauthentication, SN=8, FN=0, Flags=........
440	78.604700	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26 Deauthentication, SN=9, FN=0, Flags=........
441	78.606458	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26 Deauthentication, SN=10, FN=0, Flags=........
442	78.609634	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26 Deauthentication, SN=11, FN=0, Flags=........

We might notice right away that an excessive amount of deauthentication frames were sent to one of our client devices. This would be an immediate indicator of this attack. Additionally, if we were to open the fixed parameters under wireless management, we might notice that reason `code 7` was utilized.



As previously mentioned, if we wanted to verify this was done by an attacker, we should be able to filter even further for only deauthentication requests with reason `code 7`. As mentioned, `aireplay-ng` and `mdk4`, which are common attack tools, utilize this reason code by default. We could do with the following wireshark filter.

- `(wlan.bssid == F8:14:FE:4D:E6:F1) and (wlan.fc.type == 00) and (wlan.fc.type_subtype == 12) and (wlan.fixed.reason_code == 7)`



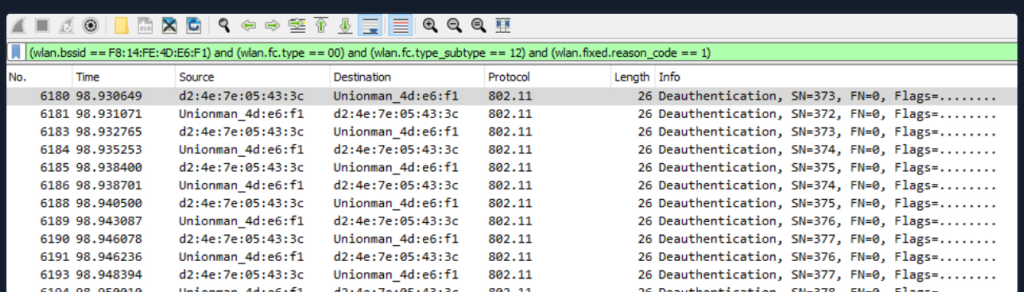
No.	Time	Source	Destination	Protocol	Length	Info
416	78.561456	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=0, FN=0, Flags=........
417	78.565783	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=1, FN=0, Flags=........
418	78.565801	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=0, FN=0, Flags=........
420	78.566384	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=1, FN=0, Flags=........
421	78.570171	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=2, FN=0, Flags=........
422	78.572747	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=3, FN=0, Flags=........
423	78.572834	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=2, FN=0, Flags=........
425	78.574455	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=3, FN=0, Flags=........
426	78.581599	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=4, FN=0, Flags=........
427	78.583939	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=5, FN=0, Flags=........
428	78.584316	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=4, FN=0, Flags=........
430	78.586261	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=5, FN=0, Flags=........
431	78.589988	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=6, FN=0, Flags=........
432	78.592997	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=7, FN=0, Flags=........
433	78.593021	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=6, FN=0, Flags=........
435	78.594615	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=7, FN=0, Flags=........
436	78.598612	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=8, FN=0, Flags=........
437	78.601517	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=9, FN=0, Flags=........
438	78.601693	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=8, FN=0, Flags=........
440	78.604700	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=9, FN=0, Flags=........
441	78.606458	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=10, FN=0, Flags=........
442	78.609634	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=11, FN=0, Flags=........
443	78.609673	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=10, FN=0, Flags=........

Revolving Reason Codes

Alternatively, a more sophisticated actor might attempt to evade this innately obvious sign by revolving reason codes. The principle to this, is that an attacker might try to evade any alarms that they could set off with a wireless intrusion detection system by changing the reason code every so often.

The trick to this technique of detection is incrementing like an attacker script would. We would first start with reason code 1.

- `(wlan.bssid == F8:14:FE:4D:E6:F1) and (wlan.fc.type == 00) and (wlan.fc.type_subtype == 12) and (wlan.fixed.reason_code == 1)`

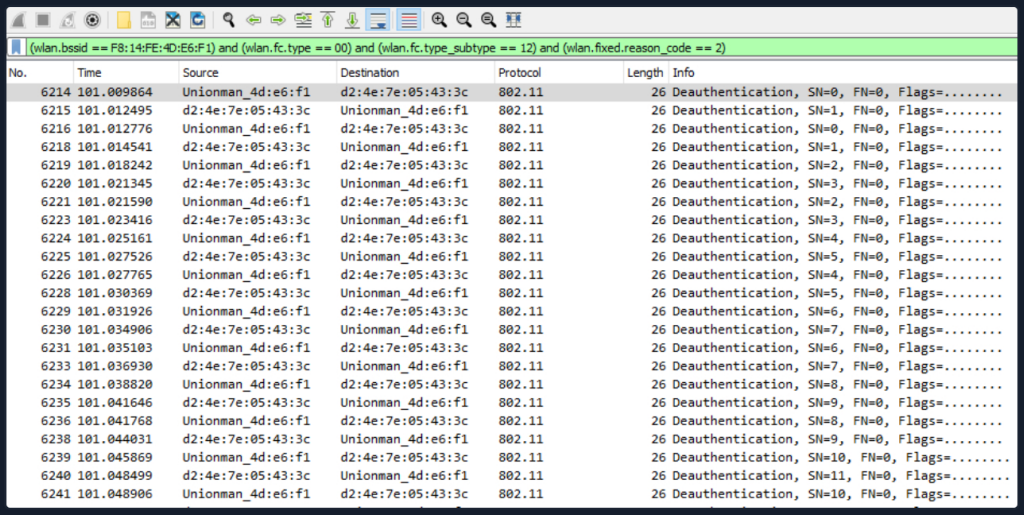


No.	Time	Source	Destination	Protocol	Length	Info
6180	98.930649	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=373, FN=0, Flags=........
6181	98.931071	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=372, FN=0, Flags=........
6183	98.932765	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=373, FN=0, Flags=........
6184	98.935253	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=374, FN=0, Flags=........
6185	98.938400	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=375, FN=0, Flags=........
6186	98.938701	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=374, FN=0, Flags=........
6188	98.940500	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=375, FN=0, Flags=........
6189	98.943087	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=376, FN=0, Flags=........
6190	98.946078	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=377, FN=0, Flags=........
6191	98.946236	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=376, FN=0, Flags=........
6193	98.948394	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=377, FN=0, Flags=........

6195	98.959842	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=379, FN=0, Flags=........
6196	98.960084	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=378, FN=0, Flags=........
6197	98.961787	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=379, FN=0, Flags=........
6199	98.966691	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=380, FN=0, Flags=........
6200	98.970664	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=381, FN=0, Flags=........
6201	98.970944	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=380, FN=0, Flags=........
6203	98.972777	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=381, FN=0, Flags=........
6204	98.974294	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=382, FN=0, Flags=........
6205	98.977465	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=383, FN=0, Flags=........
6206	98.977803	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=382, FN=0, Flags=........
6208	98.979696	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=383, FN=0, Flags=........

Then we would shift over to reason code 2.

- `(wlan.bssid == F8:14:FE:4D:E6:F1) and (wlan.fc.type == 00) and (wlan.fc.type_subtype == 12) and (wlan.fixed.reason_code == 2)`

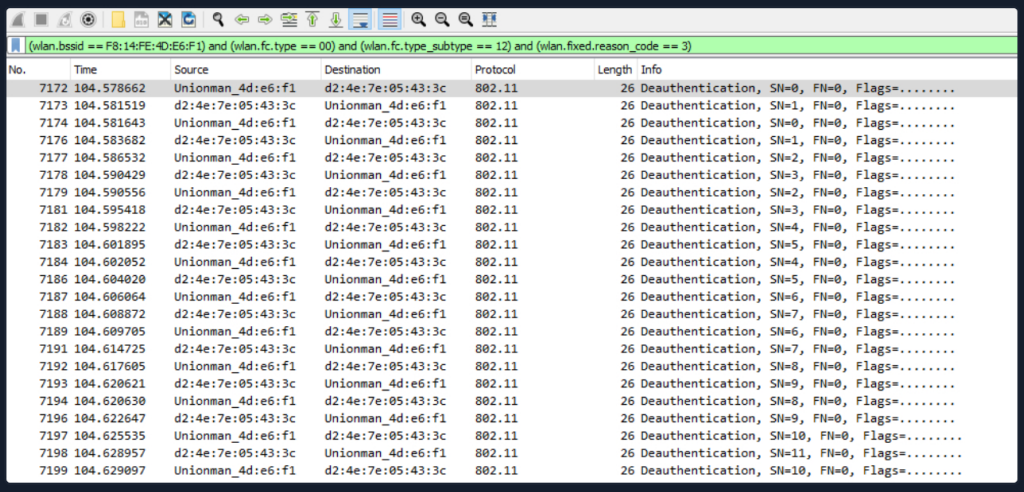


(wlan.bssid == F8:14:FE:4D:E6:F1) and (wlan.fc.type == 00) and (wlan.fc.type_subtype == 12) and (wlan.fixed.reason_code == 2)

No.	Time	Source	Destination	Protocol	Length	Info
6214	101.009864	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=0, FN=0, Flags=........
6215	101.012495	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=1, FN=0, Flags=........
6216	101.012776	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=0, FN=0, Flags=........
6218	101.014541	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=1, FN=0, Flags=........
6219	101.018242	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=2, FN=0, Flags=........
6220	101.021345	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=3, FN=0, Flags=........
6221	101.021590	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=2, FN=0, Flags=........
6223	101.023416	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=3, FN=0, Flags=........
6224	101.025161	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=4, FN=0, Flags=........
6225	101.027526	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=5, FN=0, Flags=........
6226	101.027765	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=4, FN=0, Flags=........
6228	101.030369	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=5, FN=0, Flags=........
6229	101.031926	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=6, FN=0, Flags=........
6230	101.034906	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=7, FN=0, Flags=........
6231	101.035103	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=6, FN=0, Flags=........
6233	101.036930	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=7, FN=0, Flags=........
6234	101.038820	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=8, FN=0, Flags=........
6235	101.041646	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=9, FN=0, Flags=........
6236	101.041768	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=8, FN=0, Flags=........
6238	101.044031	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=9, FN=0, Flags=........
6239	101.045869	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=10, FN=0, Flags=........
6240	101.048499	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=11, FN=0, Flags=........
6241	101.048906	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=10, FN=0, Flags=........

We would continue this sequence.

- `(wlan.bssid == F8:14:FE:4D:E6:F1) and (wlan.fc.type == 00) and (wlan.fc.type_subtype == 12) and (wlan.fixed.reason_code == 3)`



(wlan.bssid == F8:14:FE:4D:E6:F1) and (wlan.fc.type == 00) and (wlan.fc.type_subtype == 12) and (wlan.fixed.reason_code == 3)

No.	Time	Source	Destination	Protocol	Length	Info
7172	104.578662	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=0, FN=0, Flags=........
7173	104.581519	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=1, FN=0, Flags=........
7174	104.581643	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=0, FN=0, Flags=........
7176	104.583682	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=1, FN=0, Flags=........
7177	104.586532	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=2, FN=0, Flags=........
7178	104.590429	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=3, FN=0, Flags=........
7179	104.590556	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=2, FN=0, Flags=........
7181	104.595418	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=3, FN=0, Flags=........
7182	104.598222	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=4, FN=0, Flags=........
7183	104.601895	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=5, FN=0, Flags=........
7184	104.602052	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=4, FN=0, Flags=........
7186	104.604020	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=5, FN=0, Flags=........
7187	104.606064	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=6, FN=0, Flags=........
7188	104.608872	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=7, FN=0, Flags=........
7189	104.609705	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=6, FN=0, Flags=........
7191	104.614725	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=7, FN=0, Flags=........
7192	104.617605	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=8, FN=0, Flags=........
7193	104.620621	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=9, FN=0, Flags=........
7194	104.620630	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=8, FN=0, Flags=........
7196	104.622647	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=9, FN=0, Flags=........
7197	104.625535	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=10, FN=0, Flags=........
7198	104.628957	d2:4e:7e:05:43:3c	Unionman_4d:e6:f1	802.11	26	Deauthentication, SN=11, FN=0, Flags=........
7199	104.629097	Unionman_4d:e6:f1	d2:4e:7e:05:43:3c	802.11	26	Deauthentication, SN=10, FN=0, Flags=........

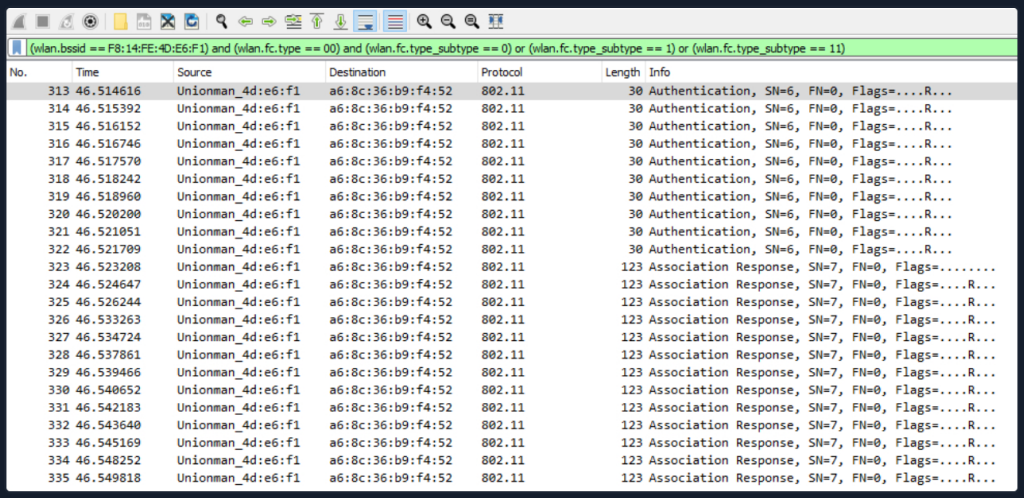
As such, deauthentication can be a pain to deal with, but we have some compensating measures that we can implement to prevent this from occuring in the modern day and age. These are:

1. `Enable IEEE 802.11w (Management Frame Protection) if possible`
2. `Utilize WPA3-SAE`
3. `Modify our WIDS/WIPS detection rules`

Finding Failed Authentication Attempts

Suppose an attacker was to attempt to connect to our wireless network. We might notice an excessive amount of association requests coming from one device. To filter for these we could use the following.

- `(wlan.bssid == F8:14:FE:4D:E6:F1) and (wlan.fc.type == 00) and (wlan.fc.type_subtype == 0) or (wlan.fc.type_subtype == 1) or (wlan.fc.type_subtype == 11)`



(wlan.bssid == F8:14:FE:4D:E6:F1) and (wlan.fc.type == 00) and (wlan.fc.type_subtype == 0) or (wlan.fc.type_subtype == 1) or (wlan.fc.type_subtype == 11)

No.	Time	Source	Destination	Protocol	Length	Info
313	46.514616	Unionman_4d:e6:f1	a6:8c:36:b9:f4:52	802.11	30	Authentication, SN=6, FN=0, Flags=....R...
314	46.515392	Unionman_4d:e6:f1	a6:8c:36:b9:f4:52	802.11	30	Authentication, SN=6, FN=0, Flags=....R...
315	46.516152	Unionman_4d:e6:f1	a6:8c:36:b9:f4:52	802.11	30	Authentication, SN=6, FN=0, Flags=....R...
316	46.516746	Unionman_4d:e6:f1	a6:8c:36:b9:f4:52	802.11	30	Authentication, SN=6, FN=0, Flags=....R...
317	46.517570	Unionman_4d:e6:f1	a6:8c:36:b9:f4:52	802.11	30	Authentication, SN=6, FN=0, Flags=....R...
318	46.518242	Unionman_4d:e6:f1	a6:8c:36:b9:f4:52	802.11	30	Authentication, SN=6, FN=0, Flags=....R...
319	46.518960	Unionman_4d:e6:f1	a6:8c:36:b9:f4:52	802.11	30	Authentication, SN=6, FN=0, Flags=....R...
320	46.520200	Unionman_4d:e6:f1	a6:8c:36:b9:f4:52	802.11	30	Authentication, SN=6, FN=0, Flags=....R...
321	46.521051	Unionman_4d:e6:f1	a6:8c:36:b9:f4:52	802.11	30	Authentication, SN=6, FN=0, Flags=....R...
322	46.521709	Unionman_4d:e6:f1	a6:8c:36:b9:f4:52	802.11	30	Authentication, SN=6, FN=0, Flags=....R...
323	46.523208	Unionman_4d:e6:f1	a6:8c:36:b9:f4:52	802.11	123	Association Response, SN=7, FN=0, Flags=........
324	46.524647	Unionman_4d:e6:f1	a6:8c:36:b9:f4:52	802.11	123	Association Response, SN=7, FN=0, Flags=....R...
325	46.526244	Unionman_4d:e6:f1	a6:8c:36:b9:f4:52	802.11	123	Association Response, SN=7, FN=0, Flags=....R...
326	46.533263	Unionman_4d:e6:f1	a6:8c:36:b9:f4:52	802.11	123	Association Response, SN=7, FN=0, Flags=....R...
327	46.534724	Unionman_4d:e6:f1	a6:8c:36:b9:f4:52	802.11	123	Association Response, SN=7, FN=0, Flags=....R...
328	46.537861	Unionman_4d:e6:f1	a6:8c:36:b9:f4:52	802.11	123	Association Response, SN=7, FN=0, Flags=....R...
329	46.539466	Unionman_4d:e6:f1	a6:8c:36:b9:f4:52	802.11	123	Association Response, SN=7, FN=0, Flags=....R...
330	46.540652	Unionman_4d:e6:f1	a6:8c:36:b9:f4:52	802.11	123	Association Response, SN=7, FN=0, Flags=....R...
331	46.542183	Unionman_4d:e6:f1	a6:8c:36:b9:f4:52	802.11	123	Association Response, SN=7, FN=0, Flags=....R...
332	46.543640	Unionman_4d:e6:f1	a6:8c:36:b9:f4:52	802.11	123	Association Response, SN=7, FN=0, Flags=....R...
333	46.545169	Unionman_4d:e6:f1	a6:8c:36:b9:f4:52	802.11	123	Association Response, SN=7, FN=0, Flags=....R...
334	46.548252	Unionman_4d:e6:f1	a6:8c:36:b9:f4:52	802.11	123	Association Response, SN=7, FN=0, Flags=....R...
335	46.549818	Unionman_4d:e6:f1	a6:8c:36:b9:f4:52	802.11	123	Association Response, SN=7, FN=0, Flags=....R...

As such, it is important for us to be able to distinguish between legitimate 802.11 traffic and attacker traffic. Link-layer security in this perspective can mean the difference between perimiter compromise and our security.

Connect to Pwnbox
Your own web-based Parrot Linux instance to play our labs.

Pwnbox Location

UK 138ms

Terminate Pwnbox to switch location

Start Instance

∞ / 1 spawns left

Waiting to start...

Enable step-by-step solutions for all questions

Questions

Answer the question(s) below to complete this Section and earn cubes!

+ 1 Inspect the deauthandbadauth.cap file, part of this module's resources, and submit the total count of

deauthentication frames as your answer.

14592

Submit

Previous Next

Mark Complete & Next

Powered by HACKTHEBOX