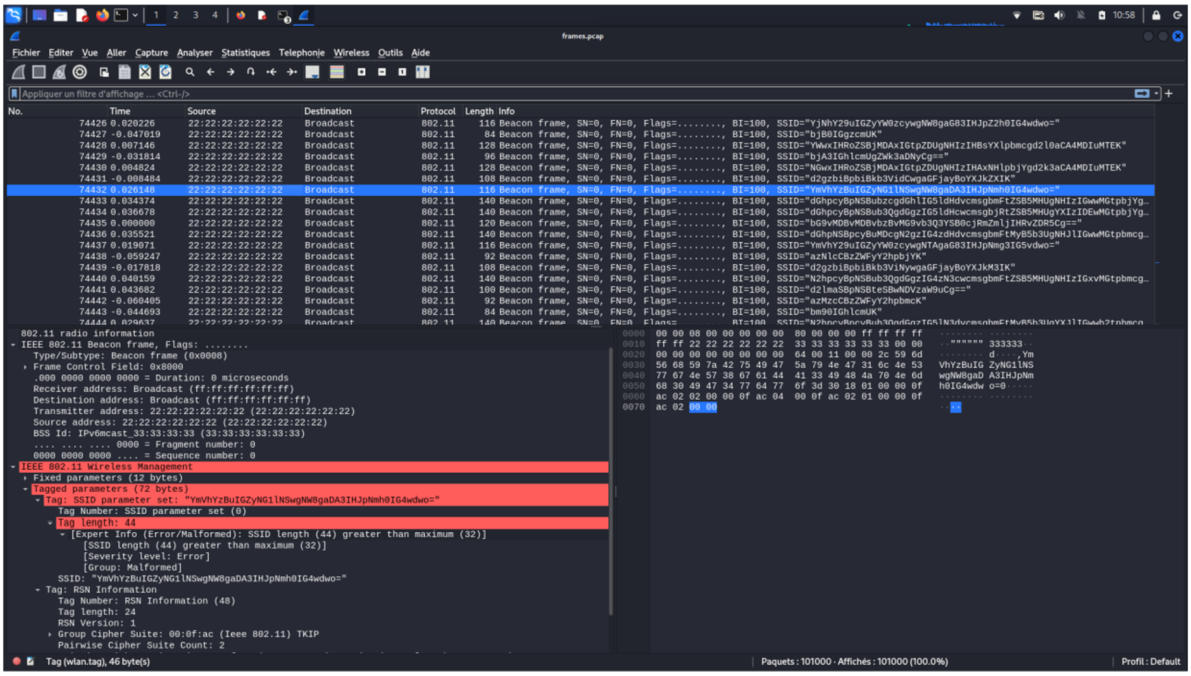
Buckeye CTF Write-up: Needle in the WiFi Stack

We are provided with a .pcap file, a packet capture file, that we can open using Wireshark:



frames.pcap
Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide
Appliquer un filtre d'affichage … <Ctrl-/>
10:58

```
No.      Time       Source             Destination  Protocol Length Info
74426  0.020226  22:22:22:22:22:22  Broadcast  802.11  116 Beacon frame, SN=0, FN=0, Flags=........, BI=100, SSID="YjNhY29uIGZyYW0zcywgNW8gaG83IHJpZ2h0IG4wdwo="
74427 -0.047019  22:22:22:22:22:22  Broadcast  802.11   84 Beacon frame, SN=0, FN=0, Flags=........, BI=100, SSID="bjB0IGgzcmUK"
74428  0.007146  22:22:22:22:22:22  Broadcast  802.11  128 Beacon frame, SN=0, FN=0, Flags=........, BI=100, SSID="YWwxIHRoZSBjMDAxIGtpZDUgNHIzIHBsYXlpbmcgd2l0aCA4MDIuMTEK"
74429 -0.031814  22:22:22:22:22:22  Broadcast  802.11   96 Beacon frame, SN=0, FN=0, Flags=........, BI=100, SSID="bjA3IGhlcmUgZWk3aDNyCg=="
74430  0.004824  22:22:22:22:22:22  Broadcast  802.11  128 Beacon frame, SN=0, FN=0, Flags=........, BI=100, SSID="NGwxIHRoZSBjMDAxIGtpZDUgNHIzIHAxNHlpbjYgd2k3aCA4MDIuMTEK"
74431 -0.008484  22:22:22:22:22:22  Broadcast  802.11  108 Beacon frame, SN=0, FN=0, Flags=........, BI=100, SSID="d2gzbiBpbiBkb3VidCwgaGFjayBoYXJkZXIK"
74432  0.026148  22:22:22:22:22:22  Broadcast  802.11  116 Beacon frame, SN=0, FN=0, Flags=........, BI=100, SSID="YmVhYzBuIGZyNG1lNSwgNW8gaDA3IHJpNmh0IG4wdwo="
74433  0.034374  22:22:22:22:22:22  Broadcast  802.11  140 Beacon frame, SN=0, FN=0, Flags=........, BI=100, SSID="dGhpcyBpNSBubzcgdGhlIG5ldHdvcmsgbmFtZSB5MHUgNHIzIGwwMGtpbjYg…
74434  0.036678  22:22:22:22:22:22  Broadcast  802.11  140 Beacon frame, SN=0, FN=0, Flags=........, BI=100, SSID="dGhpcyBpNSBub3QgdGgzIG5ldHcwcmsgbjRtZSB5MHUgYXIzIDEwMGtpbjYg…
74435  0.000000  22:22:22:22:22:22  Broadcast  802.11  120 Beacon frame, SN=0, FN=0, Flags=........, BI=100, SSID="bG9vMDBvMDBvbzBvMG9vb3Q3YSB0cjRmZmljIHRvZDR5Cg=="
74436  0.035521  22:22:22:22:22:22  Broadcast  802.11  140 Beacon frame, SN=0, FN=0, Flags=........, BI=100, SSID="dGhpNSBpcyBuMDcgN2gzIG4zdHdvcmsgbmFtMyB5b3UgNHJlIGwwMGtpbmcg…
74437  0.019071  22:22:22:22:22:22  Broadcast  802.11  116 Beacon frame, SN=0, FN=0, Flags=........, BI=100, SSID="YmVhY29uIGZyYW0zcywgNTAgaG83IHJpNmg3IG5vdwo="
74438 -0.059247  22:22:22:22:22:22  Broadcast  802.11   92 Beacon frame, SN=0, FN=0, Flags=........, BI=100, SSID="azNlcCBzZWFyY2hpbjYK"
74439 -0.017818  22:22:22:22:22:22  Broadcast  802.11  108 Beacon frame, SN=0, FN=0, Flags=........, BI=100, SSID="d2gzbiBpbiBkb3ViNywgaGFjayBoYXJkM3IK"
74440  0.040159  22:22:22:22:22:22  Broadcast  802.11  140 Beacon frame, SN=0, FN=0, Flags=........, BI=100, SSID="N2hpcyBpNSBub3QgdGgzIG4zN3cwcmsgbmFtZSB5MHUgNHIzIGxvMGtpbmcg…
74441  0.043682  22:22:22:22:22:22  Broadcast  802.11  100 Beacon frame, SN=0, FN=0, Flags=........, BI=100, SSID="d2lmaSBpNSBteSBwNDVzaW9uCg=="
74442 -0.060405  22:22:22:22:22:22  Broadcast  802.11   92 Beacon frame, SN=0, FN=0, Flags=........, BI=100, SSID="azMzcCBzZWFyY2hpbmcK"
74443 -0.044693  22:22:22:22:22:22  Broadcast  802.11   84 Beacon frame, SN=0, FN=0, Flags=........, BI=100, SSID="bm90IGhlcmUK"
74444  0.029637  22:22:22:22:22:22  Broadcast  802.11  140 Beacon frame, SN=0, FN=0, Flags=........, BI=100, SSID="N2hpcyBpcyBub3QgdGgzIG5lN3dvcmsgbmFtMyB5b3UgYXJlIGwwb2tpbmcg
```

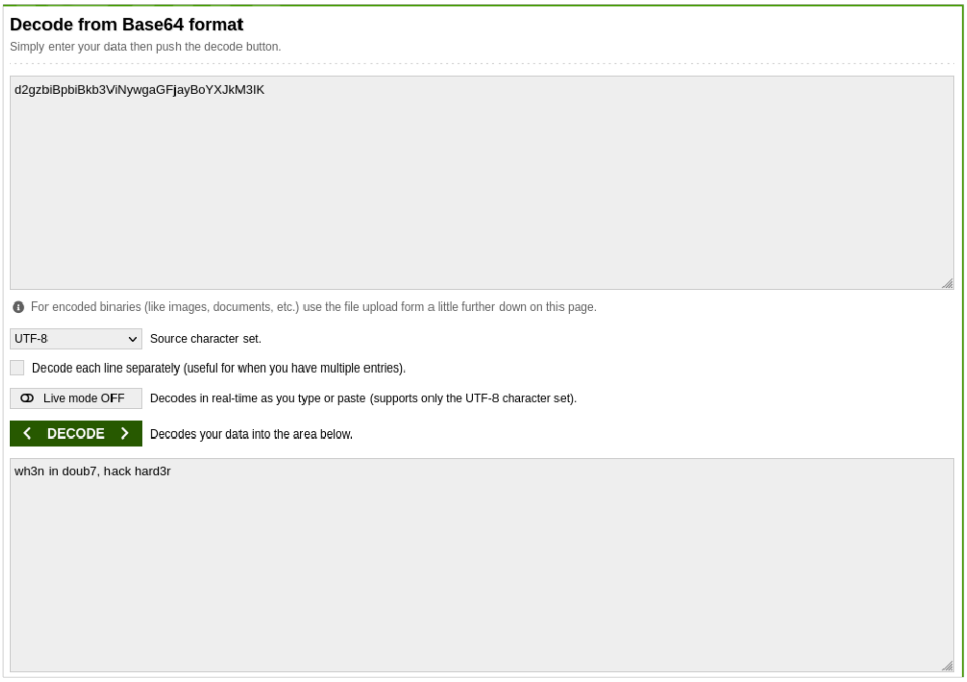
```
  802.11 radio information
▾ IEEE 802.11 Beacon frame, Flags: ........
    Type/Subtype: Beacon frame (0x0008)
  ▸ Frame Control Field: 0x8000
    .000 0000 0000 0000 = Duration: 0 microseconds
    Receiver address: Broadcast (ff:ff:ff:ff:ff:ff)
    Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
    Transmitter address: 22:22:22:22:22:22 (22:22:22:22:22:22)
    Source address: 22:22:22:22:22:22 (22:22:22:22:22:22)
    BSS Id: IPv6mcast_33:33:33:33 (33:33:33:33:33:33)
    .... .... .... 0000 = Fragment number: 0
    0000 0000 0000 .... = Sequence number: 0
▾ IEEE 802.11 Wireless Management
  ▸ Fixed parameters (12 bytes)
  ▾ Tagged parameters (72 bytes)
    ▾ Tag: SSID parameter set: "YmVhYzBuIGZyNG1lNSwgNW8gaDA3IHJpNmh0IG4wdwo="
        Tag Number: SSID parameter set (0)
      ▾ Tag length: 44
        ▾ [Expert Info (Error/Malformed): SSID length (44) greater than maximum (32)]
            [SSID length (44) greater than maximum (32)]
            [Severity level: Error]
            [Group: Malformed]
        SSID: "YmVhYzBuIGZyNG1lNSwgNW8gaDA3IHJpNmh0IG4wdwo="
    ▾ Tag: RSN Information
        Tag Number: RSN Information (48)
        Tag length: 24
        RSN Version: 1
      ▸ Group Cipher Suite: 00:0f:ac (Ieee 802.11) TKIP
        Pairwise Cipher Suite Count: 2
```

```
0000  00 00 08 00 00 00 00 00  80 00 00 00 ff ff ff ff   ........ ........
0010  ff ff 22 22 22 22 22 22  33 33 33 33 33 33 00 00   .."""""" 333333..
0020  00 00 00 00 00 00 00 00  64 00 11 00 00 2c 59 6d   ........ d....,Ym
0030  56 68 59 7a 42 75 49 47  5a 79 4e 47 31 6c 4e 53   VhYzBuIG ZyNG1lNS
0040  77 67 4e 57 38 67 61 44  41 33 49 48 4a 70 4e 6d   wgNW8gaD A3IHJpNm
0050  68 30 49 47 34 77 64 77  6f 3d 30 18 01 00 00 0f   h0IG4wdw o=0.....
0060  ac 02 02 00 00 0f ac 04  00 0f ac 02 01 00 00 0f   ........ ........
0070  ac 02 00 00                                        ....
```

Tag (wlan.tag), 46 byte(s)   Paquets : 101000 · Affichés : 101000 (100.0%)   Profil : Default

At first glance, it looks the information we need is hidden on the right : all of the SSID's are encoded.

The encoding format seems to be base64, as most of the strings have one or two equal symbols at the end. Let's decode one at base64decode.org just to be sure:



Decode from Base64 format

Simply enter your data then push the decode button.

d2gzbiBpbiBkb3ViNywgaGFjayBoYXJkM3IK

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Source character set.

□ Decode each line separately (useful for when you have multiple entries).

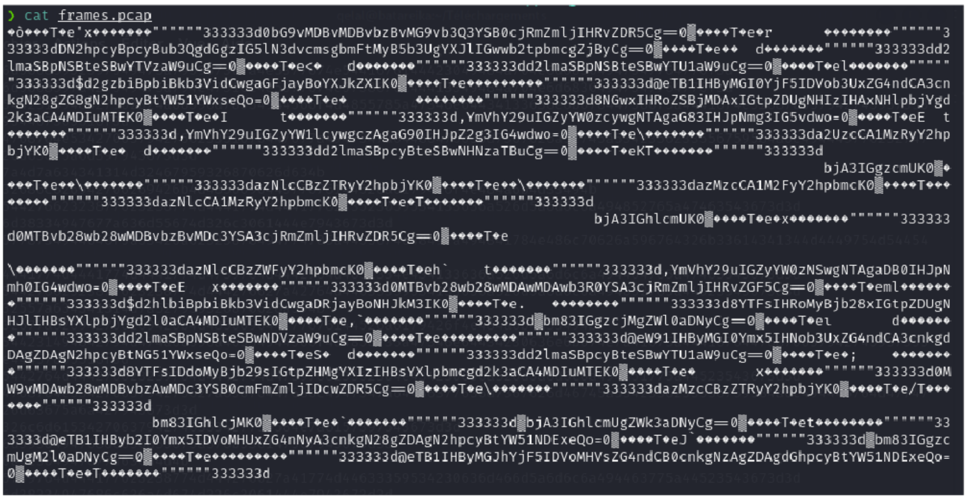
Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

< DECODE > Decodes your data into the area below.

wh3n in doub7, hack hard3r

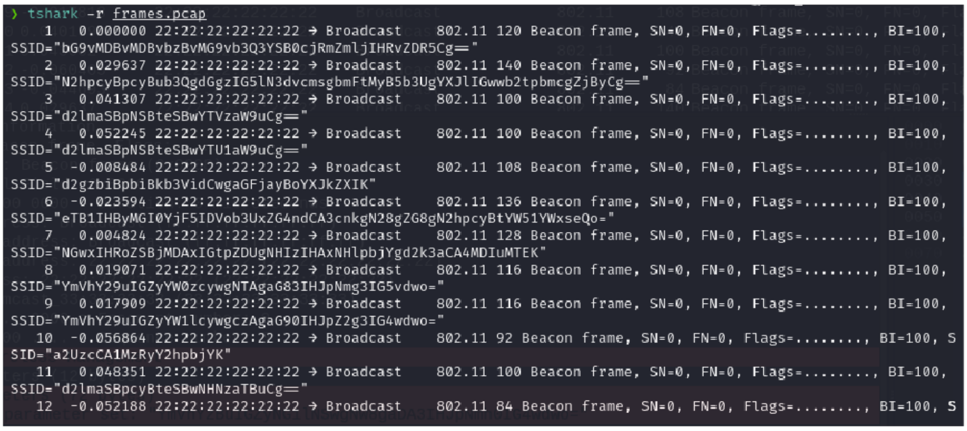
Scrolling to the end of the pcap file, we see that there are over one hundred thousand lines to be analyzed.. We clearly can't proceed manually with this amount of information.

Unfortunately, we cannot read the pcap file as-is and grep what we want, as it looks gibberish:



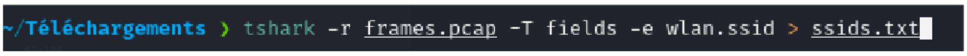
```
❯ cat frames.pcap
♦ô♦♦♦T♦e'x♦♦♦♦♦♦♦""""""333333d0bG9vMDBvMDBvbzBvMG9vb3Q3YSB0cjRmZmljIHRvZDR5Cg==0▒♦♦♦♦T♦e♦r      ♦♦♦♦♦♦♦♦""""""3
33333dDN2hpcyBpcyBub3QgdGgzIG5lN3dvcmsgbmFtMyB5b3UgYXJlIGwwb2tpbmcgZjByCg==0▒♦♦♦♦T♦e♦♦  d♦♦♦♦♦♦♦""""""333333dd2
lmaSBpNSBteSBwYTVzaW9uCg==0▒♦♦♦♦T♦e<♦   d♦♦♦♦♦♦♦""""""333333dd2lmaSBpNSBteSBwYTU1aW9uCg==0▒♦♦♦♦T♦el♦♦♦♦♦♦♦"""""
"333333d$d2gzbiBpbiBkb3VidCwgaGFjayBoYXJkZXIK0▒♦♦♦♦T♦e♦♦♦♦♦♦♦♦♦♦""""""333333d@eTB1IHByMGI0YjF5IDVob3UxZG4ndCA3cn
kgN28gZG8gN2hpcyBtYW51YWxseQo=0▒♦♦♦♦T♦e♦       ♦♦♦♦♦♦♦♦""""""333333d8NGwxIHRoZSBjMDAxIGtpZDUgNHIzIHAxNHlpbjYgd
2k3aCA4MDIuMTEK0▒♦♦♦♦T♦e♦I      t♦♦♦♦♦♦♦""""""333333d,YmVhY29uIGZyYW0zcywgNTAgaG83IHJpNmg3IG5vdwo=0▒♦♦♦♦T♦eE  t
♦♦♦♦♦♦♦""""""333333d,YmVhY29uIGZyYW1lcywgczAgaG90IHJpZ2g3IG4wdwo=0▒♦♦♦♦T♦e\♦♦♦♦♦♦♦""""""333333da2UzcCA1MzRyY2hp
bjYK0▒♦♦♦♦T♦e♦  d♦♦♦♦♦♦♦""""""333333dd2lmaSBpcyBteSBwNHNzaTBuCg==0▒♦♦♦♦T♦eKT♦♦♦♦♦♦♦""""""333333d
                                                                                                  bjA3IGgzcmUK0▒♦
♦♦♦T♦e♦♦\♦♦♦♦♦♦♦""""""333333dazNlcCBzZTRyY2hpbjYK0▒♦♦♦♦T♦e♦♦\♦♦♦♦♦♦♦""""""333333dazMzcCA1M2FyY2hpbmcK0▒♦♦♦♦T♦♦♦
♦♦♦♦♦""""""333333dazNlcCA1MzRyY2hpbmcK0▒♦♦♦♦T♦e♦T♦♦♦♦♦♦♦""""""333333d
                                                              bjA3IGhlcmUK0▒♦♦♦♦T♦e♦x♦♦♦♦♦♦♦""""""333333
d0MTBvb28wb28wMDBvbzBvMDc3YSA3cjRmZmljIHRvZDR5Cg==0▒♦♦♦♦T♦e

\♦♦♦♦♦♦♦""""""333333dazNlcCBzZWFyY2hpbmcK0▒♦♦♦♦T♦eh`    t♦♦♦♦♦♦♦""""""333333d,YmVhY29uIGZyYW0zNSwgNTAgaDB0IHJpN
mh0IG4wdwo=0▒♦♦♦♦T♦eE   x♦♦♦♦♦♦♦""""""333333d0MTBvb28wb28wMDAwMDAwb3R0YSA3cjRmZmljIHRvZGF5Cg==0▒♦♦♦♦T♦eml♦♦♦♦♦♦
♦""""""333333d$d2hlbiBpbiBkb3VidCwgaDRjayBoNHJkM3IK0▒♦♦♦♦T♦e.   ♦♦♦♦♦♦♦♦""""""333333d8YTFsIHRoMyBjb28xIGtpZDUgN
HJlIHBsYXlpbjYgd2l0aCA4MDIuMTEK0▒♦♦♦♦T♦e,`♦♦♦♦♦♦♦""""""333333d▒bm83IGgzcjMgZWl0aDNyCg==0▒♦♦♦♦T♦eι      d♦♦♦♦♦♦
♦""""""333333dd2lmaSBpNSBteSBwNDVzaW9uCg==0▒♦♦♦♦T♦e♦♦♦♦♦♦♦♦♦♦""""""333333d@eW91IHByMGI0Ymx5IHNob3UxZG4ndCA3cnkgd
DAgZDAgN2hpcyBtNG51YWxseQo=0▒♦♦♦♦T♦eS♦  d♦♦♦♦♦♦♦""""""333333dd2lmaSBpcyBteSBwYTU1aW9uCg==0▒♦♦♦♦T♦e♦;    ♦♦♦♦♦♦♦
♦""""""333333d8YTFsIDdoMyBjb29sIGtpZHMgYXIzIHBsYXlpbmcgd2k3aCA4MDIuMTEK0▒♦♦♦♦T♦e♦      x♦♦♦♦♦♦♦""""""333333d0M
W9vMDAwb28wMDBvbzAwMDc3YSB0cmFmZmljIDcwZDR5Cg==0▒♦♦♦♦T♦e\♦♦♦♦♦♦♦""""""333333dazMzcCBzZTRyY2hpbjYK0▒♦♦♦♦T♦e/T♦♦♦
♦♦♦♦""""""333333d
                   bm83IGhlcjMK0▒♦♦♦♦T♦e♦`♦♦♦♦♦♦♦""""""333333d▒bjA3IGhlcmUgZWk3aDNyCg==0▒♦♦♦♦T♦et♦♦♦♦♦♦♦♦""""""33
3333d@eTB1IHByb2I0Ymx5IDVoMHUxZG4nNyA3cnkgN28gZDAgN2hpcyBtYW51NDExeQo=0▒♦♦♦♦T♦eJ`♦♦♦♦♦♦♦""""""333333d▒bm83IGgzc
mUgM2l0aDNyCg==0▒♦♦♦♦T♦e♦♦♦♦♦♦♦♦♦♦""""""333333d@eTB1IHByMGJhYjF5IDVoMHVsZG4ndCB0cnkgNzAgZDAgdGhpcyBtYW51NDExeQo=
0▒♦♦♦♦T♦e♦T♦♦♦♦♦♦♦""""""333333d
```

We could use Tshark to read the pcap file from the terminal, but all the packet information we do not need is still present :



```
❯ tshark -r frames.pcap
    1   0.000000 22:22:22:22:22:22 → Broadcast    802.11 120 Beacon frame, SN=0, FN=0, Flags=........, BI=100,
SSID="bG9vMDBvMDBvbzBvMG9vb3Q3YSB0cjRmZmljIHRvZDR5Cg=="
    2   0.029637 22:22:22:22:22:22 → Broadcast    802.11 140 Beacon frame, SN=0, FN=0, Flags=........, BI=100,
SSID="N2hpcyBpcyBub3QgdGgzIG5lN3dvcmsgbmFtMyB5b3UgYXJlIGwwb2tpbmcgZjByCg=="
    3   0.041307 22:22:22:22:22:22 → Broadcast    802.11 100 Beacon frame, SN=0, FN=0, Flags=........, BI=100,
SSID="d2lmaSBpNSBteSBwYTVzaW9uCg=="
    4   0.052245 22:22:22:22:22:22 → Broadcast    802.11 100 Beacon frame, SN=0, FN=0, Flags=........, BI=100,
SSID="d2lmaSBpNSBteSBwYTU1aW9uCg=="
    5  -0.008484 22:22:22:22:22:22 → Broadcast    802.11 108 Beacon frame, SN=0, FN=0, Flags=........, BI=100,
SSID="d2gzbiBpbiBkb3VidCwgaGFjayBoYXJkZXIK"
    6  -0.023594 22:22:22:22:22:22 → Broadcast    802.11 136 Beacon frame, SN=0, FN=0, Flags=........, BI=100,
SSID="eTB1IHByMGI0YjF5IDVob3UxZG4ndCA3cnkgN28gZG8gN2hpcyBtYW51YWxseQo="
    7   0.004824 22:22:22:22:22:22 → Broadcast    802.11 128 Beacon frame, SN=0, FN=0, Flags=........, BI=100,
SSID="NGwxIHRoZSBjMDAxIGtpZDUgNHIzIHAxNHlpbjYgd2k3aCA4MDIuMTEK"
    8   0.019071 22:22:22:22:22:22 → Broadcast    802.11 116 Beacon frame, SN=0, FN=0, Flags=........, BI=100,
SSID="YmVhY29uIGZyYW0zcywgNTAgaG83IHJpNmg3IG5vdwo="
    9   0.017909 22:22:22:22:22:22 → Broadcast    802.11 116 Beacon frame, SN=0, FN=0, Flags=........, BI=100,
SSID="YmVhY29uIGZyYW1lcywgczAgaG90IHJpZ2g3IG4wdwo="
   10  -0.056864 22:22:22:22:22:22 → Broadcast    802.11 92 Beacon frame, SN=0, FN=0, Flags=........, BI=100, S
SID="a2UzcCA1MzRyY2hpbjYK"
   11   0.048351 22:22:22:22:22:22 → Broadcast    802.11 100 Beacon frame, SN=0, FN=0, Flags=........, BI=100,
SSID="d2lmaSBpcyBteSBwNHNzaTBuCg=="
   12  -0.052188 22:22:22:22:22:22 → Broadcast    802.11 84 Beacon frame, SN=0, FN=0, Flags=........, BI=100, S
```

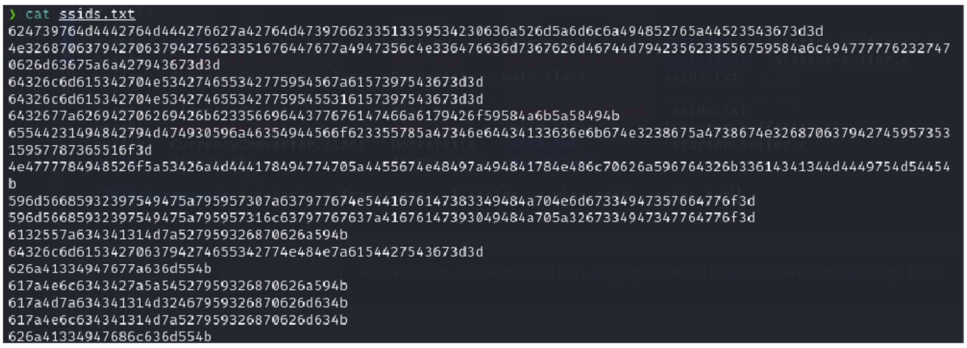
By reading the Tshark help page & manual, we see that there are options for extracting certain fields in the packets. We only want the SSIDs, so we will use this :



```
~/Téléchargements ❯ tshark -r frames.pcap -T fields -e wlan.ssid > ssids.txt
```

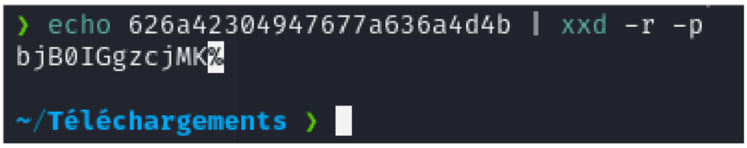
That command tells Tshark to read the frames.pcap file, to extract data as fields, and only print the WLAN SSID field. The output will be stored in the ssids.txt file.

Running this, we obtain a file containing... hex? Well, we will have to convert this output to ASCII in order to read it properly.



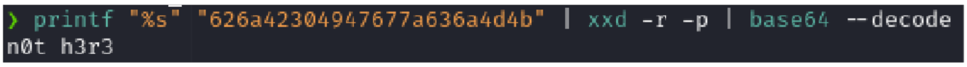
```
❯ cat ssids.txt
624739764d4442764d444276627a42764d4739766233513359534230636a526d5a6d6c6a494852765a44523543673d3d
4e32687063794270637942756233516447677a4947356c4e336476636d7367626d46744d7942356233556759584a6c494777776232747
0626d63675a6a427943673d3d
64326c6d615342704e534274655342775954567a6157397543673d3d
64326c6d615342704e53427465534277595455316157397543673d3d
6432677a626942706269426b62335669644377676147466a6179426f59584a6b5a58494b
655442314948427944474930596a46354944566f623355785a47346e64434133636e6b674e3238675a4738674e32687063794274595735
15957787365516f3d
4e4777784948526f5a53426a4d444178494774705a4455674e48497a494841784e486c70626a596764326b3361434134d4449754d54454
b
596d5668593239754947475a795957307a637977674e5441676147383349484a704e6d67334947357664776f3d
596d5668593239754947475a7959573316c63797767637a41676147393049484a705a3267334947347764776f3d
6132557a634341314d7a527959326870626a594b
64326c6d615342706379427465534277e484e7a6154427543673d3d
626a41334947677a636d554b
617a4e6c6343427a5a5452795932687062a594b
617a4d7a634341314d3246795932687062d634b
617a4e6c634341314d7a527959326870626d634b
626a41334947686c636d554b
```

We can pipe a single line of hex through xxd to convert it to ASCII:



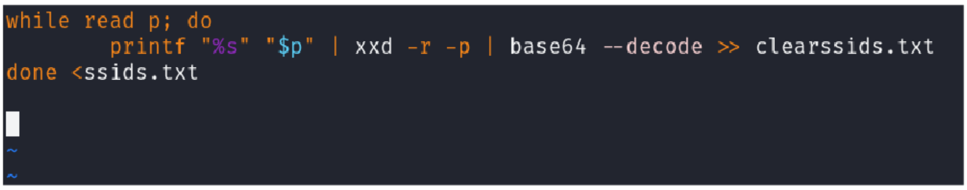
```
❯ echo 626a42304947677a636a4d4b | xxd -r -p
bjB0IGgzcjMK%

~/Téléchargements ❯
```

That looks like base64 to me. Let's pipe our command output through base64 decode:



```
❯ printf "%s" "626a42304947677a636a4d4b" | xxd -r -p | base64 --decode
n0t h3r3
```

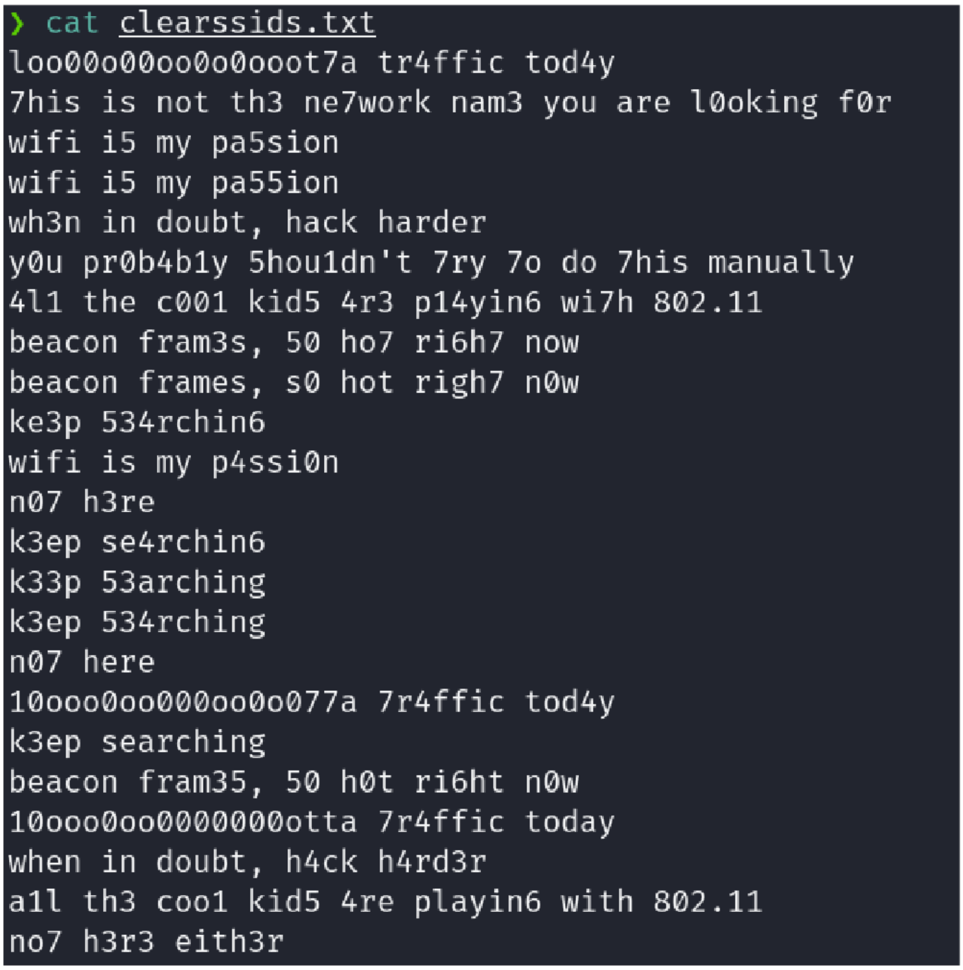
That is what we wanted. Now, let's automate this process for the 100k lines by coding a small Bash script (I called it decode.sh):



```
while read p; do
        printf "%s" "$p" | xxd -r -p | base64 --decode >> clearssids.txt
done <ssids.txt

~
~
```

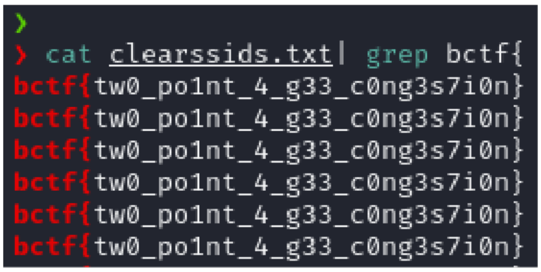
This will start a while loop, and read the ssids.txt file we created earlier. For each line of hex, it will convert it into ASCII base64, then into readable text, and then append the line to the clearssids.txt file. We will then get a 100k line file with all the human-readable SSIDs.

After executing the script, we get this:



```
❯ cat clearssids.txt
loo00o00oo0o0ooot7a tr4ffic tod4y
7his is not th3 ne7work nam3 you are l0oking f0r
wifi i5 my pa5sion
wifi i5 my pa55ion
wh3n in doubt, hack harder
y0u pr0b4b1y 5hou1dn't 7ry 7o do 7his manually
4l1 the c001 kid5 4r3 p14yin6 wi7h 802.11
beacon fram3s, 50 ho7 ri6h7 now
beacon frames, s0 hot righ7 n0w
ke3p 534rchin6
wifi is my p4ssi0n
n07 h3re
k3ep se4rchin6
k33p 53arching
k3ep 534rching
n07 here
10ooo0oo000oo0o077a 7r4ffic tod4y
k3ep searching
beacon fram35, 50 h0t ri6ht n0w
10ooo0oo0000000otta 7r4ffic today
when in doubt, h4ck h4rd3r
a1l th3 coo1 kid5 4re playin6 with 802.11
no7 h3r3 eith3r
```

The flag could be hidden in all of this mess. We can find flags by grepping the specific CTF flag format:



```
❯
❯ cat clearssids.txt| grep bctf{
bctf{tw0_po1nt_4_g33_c0ng3s7i0n}
bctf{tw0_po1nt_4_g33_c0ng3s7i0n}
bctf{tw0_po1nt_4_g33_c0ng3s7i0n}
bctf{tw0_po1nt_4_g33_c0ng3s7i0n}
bctf{tw0_po1nt_4_g33_c0ng3s7i0n}
bctf{tw0_po1nt_4_g33_c0ng3s7i0n}
```

We found the flag.

Writeup author: batareika