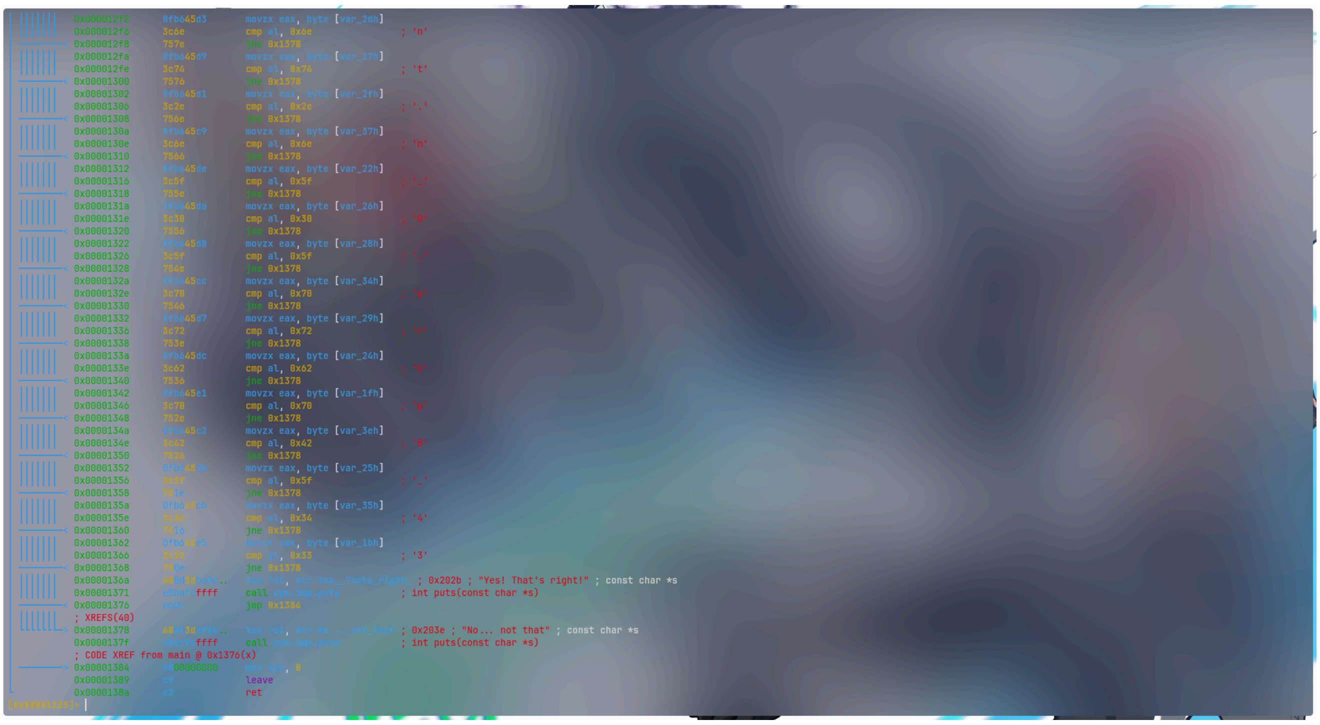
disassemble with radare2 `r2 tablet`
then we check the main function which just shows a bunch of characters

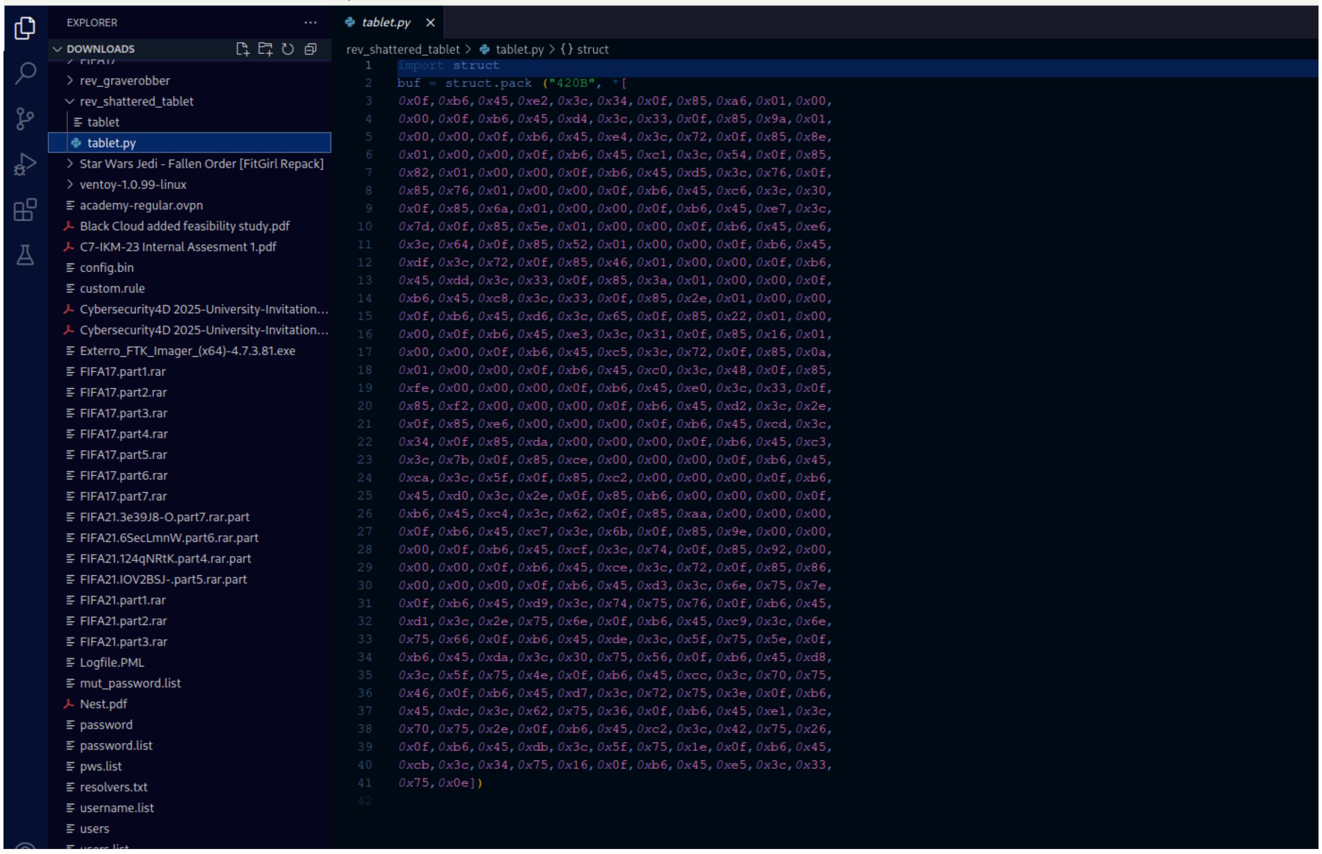


```
0x000012f2      0fb645d3       movzx eax, byte [var_2dh]
0x000012f6      3c6e           cmp al, 0x6e                  ; 'n'
0x000012f8      757e           jne 0x1378
0x000012fa      0fb645d9       movzx eax, byte [var_27h]
0x000012fe      3c74           cmp al, 0x74                  ; 't'
0x00001300      7576           jne 0x1378
0x00001302      0fb645d1       movzx eax, byte [var_2fh]
0x00001306      3c2e           cmp al, 0x2e                  ; '.'
0x00001308      756e           jne 0x1378
0x0000130a      0fb645c9       movzx eax, byte [var_37h]
0x0000130e      3c6e           cmp al, 0x6e                  ; 'n'
0x00001310      7566           jne 0x1378
0x00001312      0fb645de       movzx eax, byte [var_22h]
0x00001316      3c5f           cmp al, 0x5f                  ; '_'
0x00001318      755e           jne 0x1378
0x0000131a      0fb645da       movzx eax, byte [var_26h]
0x0000131e      3c30           cmp al, 0x30                  ; '0'
0x00001320      7556           jne 0x1378
0x00001322      0fb645d8       movzx eax, byte [var_28h]
0x00001326      3c5f           cmp al, 0x5f                  ; '_'
0x00001328      754e           jne 0x1378
0x0000132a      0fb645cc       movzx eax, byte [var_34h]
0x0000132e      3c70           cmp al, 0x70                  ; 'p'
0x00001330      7546           jne 0x1378
0x00001332      0fb645d7       movzx eax, byte [var_29h]
0x00001336      3c72           cmp al, 0x72                  ; 'r'
0x00001338      753e           jne 0x1378
0x0000133a      0fb645dc       movzx eax, byte [var_24h]
0x0000133e      3c62           cmp al, 0x62                  ; 'b'
0x00001340      7536           jne 0x1378
0x00001342      0fb645e1       movzx eax, byte [var_1fh]
0x00001346      3c70           cmp al, 0x70                  ; 'p'
0x00001348      752e           jne 0x1378
0x0000134a      0fb645c2       movzx eax, byte [var_3eh]
0x0000134e      3c42           cmp al, 0x42                  ; 'B'
0x00001350      7526           jne 0x1378
0x00001352      0fb645db       movzx eax, byte [var_25h]
0x00001356      3c5f           cmp al, 0x5f                  ; '_'
0x00001358      751e           jne 0x1378
0x0000135a      0fb645cb       movzx eax, byte [var_35h]
0x0000135e      3c34           cmp al, 0x34                  ; '4'
0x00001360      7516           jne 0x1378
0x00001362      0fb645e5       movzx eax, byte [var_1bh]
0x00001366      3c33           cmp al, 0x33                  ; '3'
0x00001368      750e           jne 0x1378
0x0000136a      488d3dba0c..   lea rdi, str.Yes__Thats_right_ ; 0x202b ; "Yes! That's right!" ; const char *s
0x00001371      e8bafcffff     call sym.imp.puts             ; int puts(const char *s)
0x00001376      eb0c           jmp 0x1384
; XREFS(40)
0x00001378      488d3dbf0c..   lea rdi, str.No..._not_that ; 0x203e ; "No... not that" ; const char *s
0x0000137f      e8acfcffff     call sym.imp.puts             ; int puts(const char *s)
; CODE XREF from main @ 0x1376(x)
0x00001384      b800000000     mov eax, 0
0x00001389      c9             leave
0x0000138a      c3             ret
[0x00001155]>
```

We export these instructions to python code for a better view and assesment `!echo \$((0x136a - 0x11c6))`
then we move to the beginning of the target instructions with `0x11c6`
Export raw bytes to python format `pcp 420 > tablet.py`

```
[0x00001155]> !echo \$((0x136a - 0x11c6))
420
[0x00001155]> 0x11c6
[0x000011c6]> pcp 420 > tablet.py
[0x000011c6]> python3 tablet.py
```

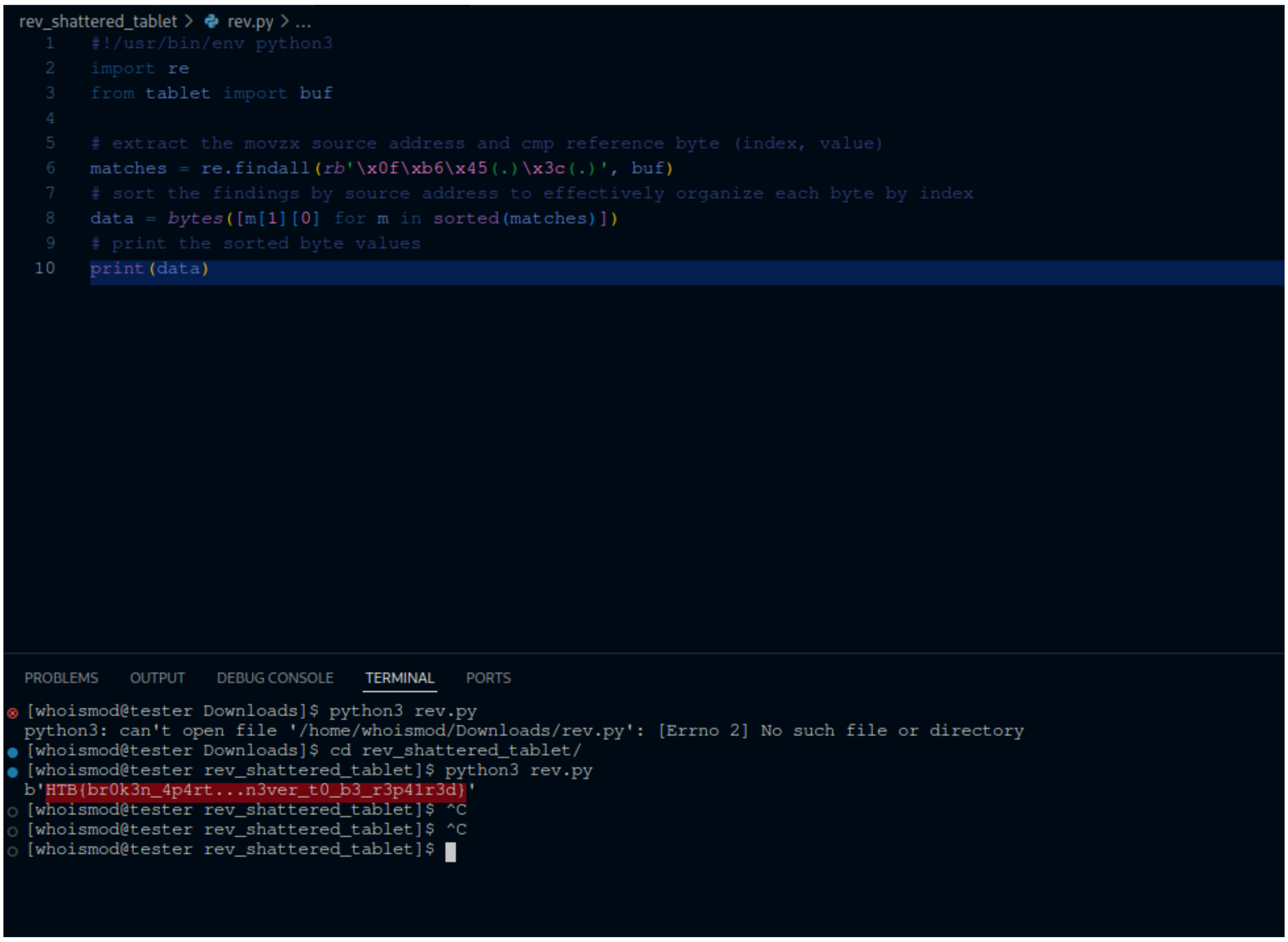
when we open the python file it shows hex characters so we decode it


EXPLORER
DOWNLOADS
rev_graverobber
rev_shattered_tablet
tablet
tablet.py
Star Wars Jedi - Fallen Order [FitGirl Repack]
ventoy-1.0.99-linux
academy-regular.ovpn
Black Cloud added feasibility study.pdf
C7-IKM-23 Internal Assesment 1.pdf
config.bin
custom.rule
Cybersecurity4D 2025-University-Invitation...
Cybersecurity4D 2025-University-Invitation...
Exterro_FTK_Imager_(x64)-4.7.3.81.exe
FIFA17.part1.rar
FIFA17.part2.rar
FIFA17.part3.rar
FIFA17.part4.rar
FIFA17.part5.rar
FIFA17.part6.rar
FIFA17.part7.rar
FIFA21.3e39J8-O.part7.rar.part
FIFA21.6SecLmnW.part6.rar.part
FIFA21.124qNRtK.part4.rar.part
FIFA21.IOV2BSJ-.part5.rar.part
FIFA21.part1.rar
FIFA21.part2.rar
FIFA21.part3.rar
Logfile.PML
mut_password.list
Nest.pdf
password
password.list
pws.list
resolvers.txt
username.list
users
tablet.py
rev_shattered_tablet > tablet.py > {} struct

```
import struct
buf = struct.pack ("420B", *[
0x0f,0xb6,0x45,0xe2,0x3c,0x34,0x0f,0x85,0xa6,0x01,0x00,
0x00,0x0f,0xb6,0x45,0xd4,0x3c,0x33,0x0f,0x85,0x9a,0x01,
0x00,0x00,0x0f,0xb6,0x45,0xe4,0x3c,0x72,0x0f,0x85,0x8e,
0x01,0x00,0x00,0x0f,0xb6,0x45,0xc1,0x3c,0x54,0x0f,0x85,
0x82,0x01,0x00,0x00,0x0f,0xb6,0x45,0xd5,0x3c,0x76,0x0f,
0x85,0x76,0x01,0x00,0x00,0x0f,0xb6,0x45,0xc6,0x3c,0x30,
0x0f,0x85,0x6a,0x01,0x00,0x00,0x0f,0xb6,0x45,0xe7,0x3c,
0x7d,0x0f,0x85,0x5e,0x01,0x00,0x00,0x0f,0xb6,0x45,0xe6,
0x3c,0x64,0x0f,0x85,0x52,0x01,0x00,0x00,0x0f,0xb6,0x45,
0xdf,0x3c,0x72,0x0f,0x85,0x46,0x01,0x00,0x00,0x0f,0xb6,
0x45,0xdd,0x3c,0x33,0x0f,0x85,0x3a,0x01,0x00,0x00,0x0f,
0xb6,0x45,0xc8,0x3c,0x33,0x0f,0x85,0x2e,0x01,0x00,0x00,
0x0f,0xb6,0x45,0xd6,0x3c,0x65,0x0f,0x85,0x22,0x01,0x00,
0x00,0x0f,0xb6,0x45,0xe3,0x3c,0x31,0x0f,0x85,0x16,0x01,
0x00,0x00,0x0f,0xb6,0x45,0xc5,0x3c,0x72,0x0f,0x85,0x0a,
0x01,0x00,0x00,0x0f,0xb6,0x45,0xc0,0x3c,0x48,0x0f,0x85,
0xfe,0x00,0x00,0x00,0x0f,0xb6,0x45,0xe0,0x3c,0x33,0x0f,
0x85,0xf2,0x00,0x00,0x00,0x0f,0xb6,0x45,0xd2,0x3c,0x2e,
0x0f,0x85,0xe6,0x00,0x00,0x00,0x0f,0xb6,0x45,0xcd,0x3c,
0x34,0x0f,0x85,0xda,0x00,0x00,0x00,0x0f,0xb6,0x45,0xc3,
0x3c,0x7b,0x0f,0x85,0xce,0x00,0x00,0x00,0x0f,0xb6,0x45,
0xca,0x3c,0x5f,0x0f,0x85,0xc2,0x00,0x00,0x00,0x0f,0xb6,
0x45,0xd0,0x3c,0x2e,0x0f,0x85,0xb6,0x00,0x00,0x00,0x0f,
0xb6,0x45,0xc4,0x3c,0x62,0x0f,0x85,0xaa,0x00,0x00,0x00,
0x0f,0xb6,0x45,0xc7,0x3c,0x6b,0x0f,0x85,0x9e,0x00,0x00,
0x00,0x0f,0xb6,0x45,0xcf,0x3c,0x74,0x0f,0x85,0x92,0x00,
0x00,0x00,0x0f,0xb6,0x45,0xce,0x3c,0x72,0x0f,0x85,0x86,
0x00,0x00,0x00,0x0f,0xb6,0x45,0xd3,0x3c,0x6e,0x75,0x7e,
0x0f,0xb6,0x45,0xd9,0x3c,0x74,0x75,0x76,0x0f,0xb6,0x45,
0xd1,0x3c,0x2e,0x75,0x6e,0x0f,0xb6,0x45,0xc9,0x3c,0x6e,
0x75,0x66,0x0f,0xb6,0x45,0xde,0x3c,0x5f,0x75,0x5e,0x0f,
0xb6,0x45,0xda,0x3c,0x30,0x75,0x56,0x0f,0xb6,0x45,0xd8,
0x3c,0x5f,0x75,0x4e,0x0f,0xb6,0x45,0xcc,0x3c,0x70,0x75,
0x46,0x0f,0xb6,0x45,0xd7,0x3c,0x72,0x75,0x3e,0x0f,0xb6,
0x45,0xdc,0x3c,0x62,0x75,0x36,0x0f,0xb6,0x45,0xe1,0x3c,
0x70,0x75,0x2e,0x0f,0xb6,0x45,0xc2,0x3c,0x42,0x75,0x26,
0x0f,0xb6,0x45,0xdb,0x3c,0x5f,0x75,0x1e,0x0f,0xb6,0x45,
0xcb,0x3c,0x34,0x75,0x16,0x0f,0xb6,0x45,0xe5,0x3c,0x33,
0x75,0x0e])
```

Create python code to extract the relevant information to the raw bytes using regular expression in python

```
#!/usr/bin/env python3

import re

from tablet import buf



# extract the movzx source address and cmp reference byte (index, value)

matches = re.findall(rb'\x0f\xb6\x45(.)\x3c(.)', buf)

# sort the findings by source address to effectively organize each byte by index

data = bytes([m[1][0] for m in sorted(matches)])

# print the sorted byte values

print(data)
```


rev_shattered_tablet > rev.py > ...

```python
#!/usr/bin/env python3
import re
from tablet import buf

# extract the movzx source address and cmp reference byte (index, value)
matches = re.findall(rb'\x0f\xb6\x45(.)\x3c(.)', buf)
# sort the findings by source address to effectively organize each byte by index
data = bytes([m[1][0] for m in sorted(matches)])
# print the sorted byte values
print(data)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[whoismod@tester Downloads]$ python3 rev.py
python3: can't open file '/home/whoismod/Downloads/rev.py': [Errno 2] No such file or directory
[whoismod@tester Downloads]$ cd rev_shattered_tablet/
[whoismod@tester rev_shattered_tablet]$ python3 rev.py
b'HTB{br0k3n_4p4rt...n3ver_t0_b3_r3p41r3d}'
[whoismod@tester rev_shattered_tablet]$ ^C
[whoismod@tester rev_shattered_tablet]$ ^C
[whoismod@tester rev_shattered_tablet]$ 
```

and it prints the flag