

CYSCOM VITCC

Knights CTF Walkthrough

Reverse Engineering 1 - The Flag Vault

It's a simple password binary. By using the strings tool, we can find 2 strings -

abracadaH and brahahahH

After opening up the binary in radare2, I found that it was just concatenating both the strings while also removing the last H and then adds an additional a

So the final answer is abracadabrahahaha

The screenshot shows the radare2 interface with assembly code and various analysis details. The assembly code includes several string literals and their concatenation:

```
0x55974601d284 [xaDvc]0 0% 245 /home/kali/ctf/knight/rev/The_Flag_Vault> diq;q?;f t.. @ main+187 # 0x55974601d284
breakpoint at 0x00000000
0x55974601d284 0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
0x55974601d284 0000 0000 0000 4500 7000 7700 6300 3000 .....K} w.c.0.
0x55974601d284 5400 4600 4300 5000 6000 7400 7200 7600 T.F.C_.m.t.r.v.1288 nov word [rbp-0x48], 0x67
0x55974601d284 7300 7b00 6000 3300 6500 6700 6c00 6900 s.[n.3.e.g.l.1.1286 nov word [rbp-0x44], 0x6c
0x55974601d284 975a fe72 fe7f 0000 3dd4 0146 9755 0000 .Z.r...+e.F.U.128c nov word [rbp-0x42], 0x69
0x55974601d284 0000 0000 0000 0000 0000 0000 0000 0000 novabs rax, 0x55974601d284
0x55974601d284 6162 7261 6361 6461 6272 6168 6168 6168 abracadabrahahaha novabs rdx, 0x55974601d284
0x55974601d284 6100 fe72 fe7f 0000 0061 c963 4399 4361 a.r...+e.C.J.Ca novabs rbp, 0x55974601d282
0x55974601d284 f0d3 0146 9755 0000 0add 811f d77f 0000 ...F.U.....127a nov quword [rbp-0x20], rax
0x55974601d284 a85b fe72 fe7f 0000 0000 0000 0100 ...[r.i.....127b nov quword [rbp-0x18], rdx
0x55974601d284 c901 0146 9755 0000 cfd7 811f d77f 0000 ...F.U.....127c nov word [rbp-0x10], rdx
0x55974601d284 0000 0000 0000 0000 faee 7a27 8a71 faee .....t'q...1284 les rdi, quword [0x01theremplexo]
0x55974601d284 e000 0146 9755 0000 0000 0000 0000 0000 mov ax, 0xb8
0x55974601d284 0000 0000 0000 0000 0000 0000 0000 0000 call sub_1000
0x55974601d284 f4ae 1435 7518 28be faee 1239 8ac2 7abe ...Su.....9..z.1295 les rax, quword [rbp-0x40]
0x55974601d284 0000 0000 0000 0000 0000 0000 0000 0000 mov rsi, rax
0x55974601d284 0000 0000 0000 0100 0000 0000 0000 0000 lea rdi, quword [0x01theremplexo+65]
0x55974601d284 rx 0x55974601d2621 rbx 0x00000000 rcx 0x7fd71f9b5718 1293 mov esp, 0xb8
0x55974601d284 rdx 0x55974601d2622 r8 0x00000000 r9 0x7fd71f9e7100 1294 call add_1000
0x55974601d284 r10 0x00000000 r11 0x000000c2 r12 0x55974601d0e0 1295 lea rdi, quword [rbp-0x40]
0x55974601d284 r13 0x00000000 r14 0x00000000 r15 0x00000000 1296 les rax, quword [rbp-0x20]
0x55974601d284 rsi 0xffffe72fe5ba8 rdi 0x00000001 rsp 0x7fe72fe5a40 1297 mov rsi, rdx
0x55974601d284 rbp 0xffffe72fe5ba0 rip 0x55974601d284 rflags 0x00000246 1298 mov rdi, rax
0x55974601d284 orax 0xffffffffffffffffff 1299 call add_1000
0x55974601d284 b 488d3d70d010 lea rdi, qword str.Hi_there_____Please_enter_the_password_to_unlock_the_flag_vault: ; 0x55974601d284
0x55974601d284 b8000000 mov eax, 0
```

File Information:
Path: /home/kali/ctf/knight/rev/The_Flag_Vault
Loader: ELF
CPU: intel/x86_64
Branch always stops procedures
CPU Syntax Variant: Intel
Calling Convention: System V
Address Information:
Type: Procedure
Prolog Heuristic: Not a procedure prolog
Prolog Mode: Use Heuristic
Navigation History

FLAG = KCTF{welc0me_t0_reverse_3ngineering}

Reverse Engineering 2 - The Encoder

The binary was just encoding letters. 'A' = 1402 and so on.. Using this logic, we just find the difference between each number and 1402. According to the ASCII table, we can decode the initial string.

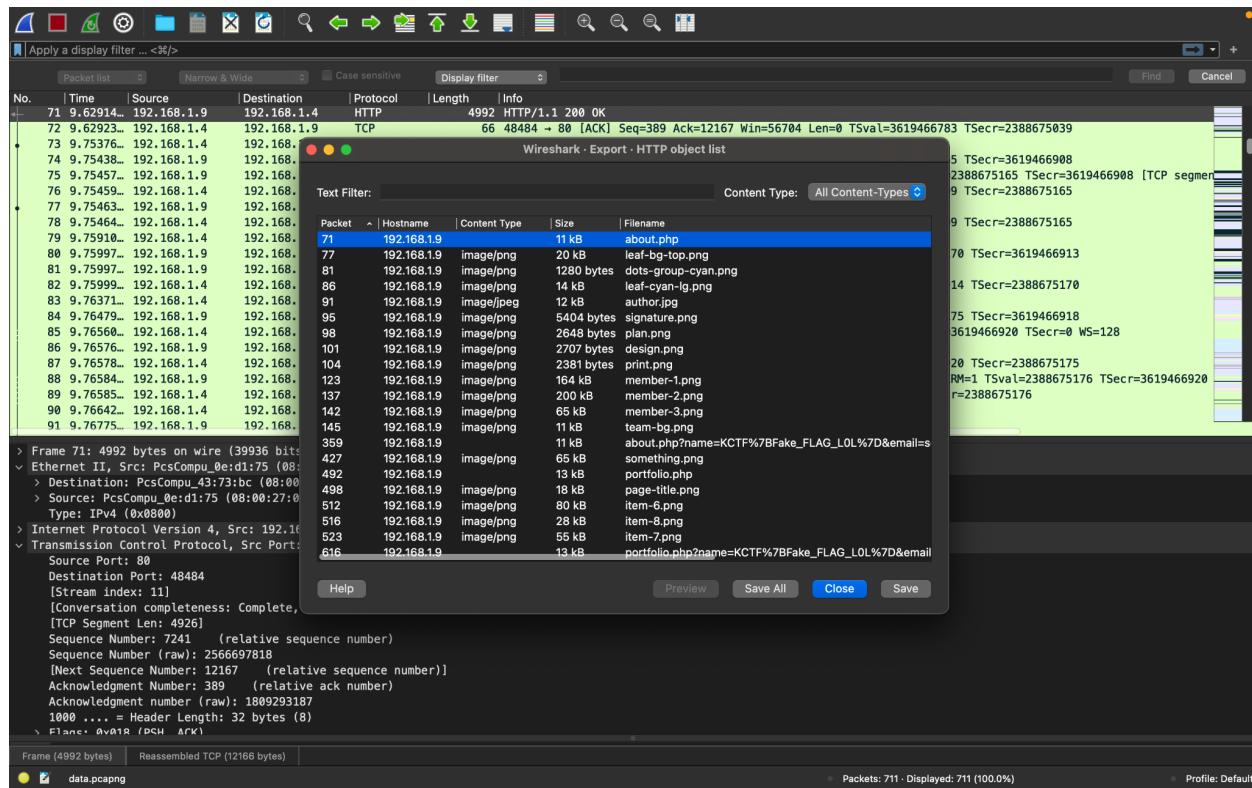
```
tmp = "1412 1404 1421 1407 1460 1452 1386 1414 1449 1445 1388 1432 1388 1415 1436  
1385  
1405 1388 1451 1432 1386 1388 1388 1392 1462"  
  
tmp = tmp.split()  
for i in tmp:  
    i = int(i)  
    i = i - 1402  
    print(chr(ord('A') + i), end = "")
```

FLAG = KCTF{s1MpI3_3Nc0D3r_1337}

Networking 1 - Hows the shark:

Open pcapng file

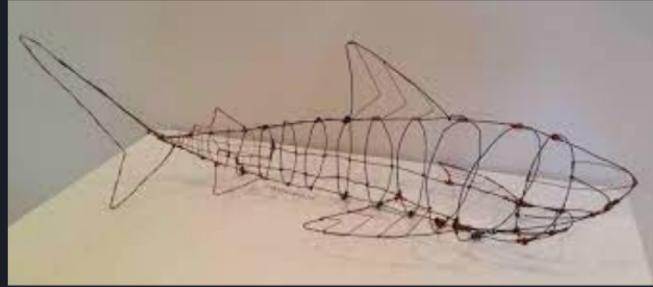
Go to File -> Export Objects -> HTTP



As we can see there are 4 unnamed packets. 2 of them are titled Fake Flags so lets not look there.

Saving the about.png file in my desktop reveals to be a photo which has the flag inside!

KCTF{A_ShARK_iN_tHe_WiRE}

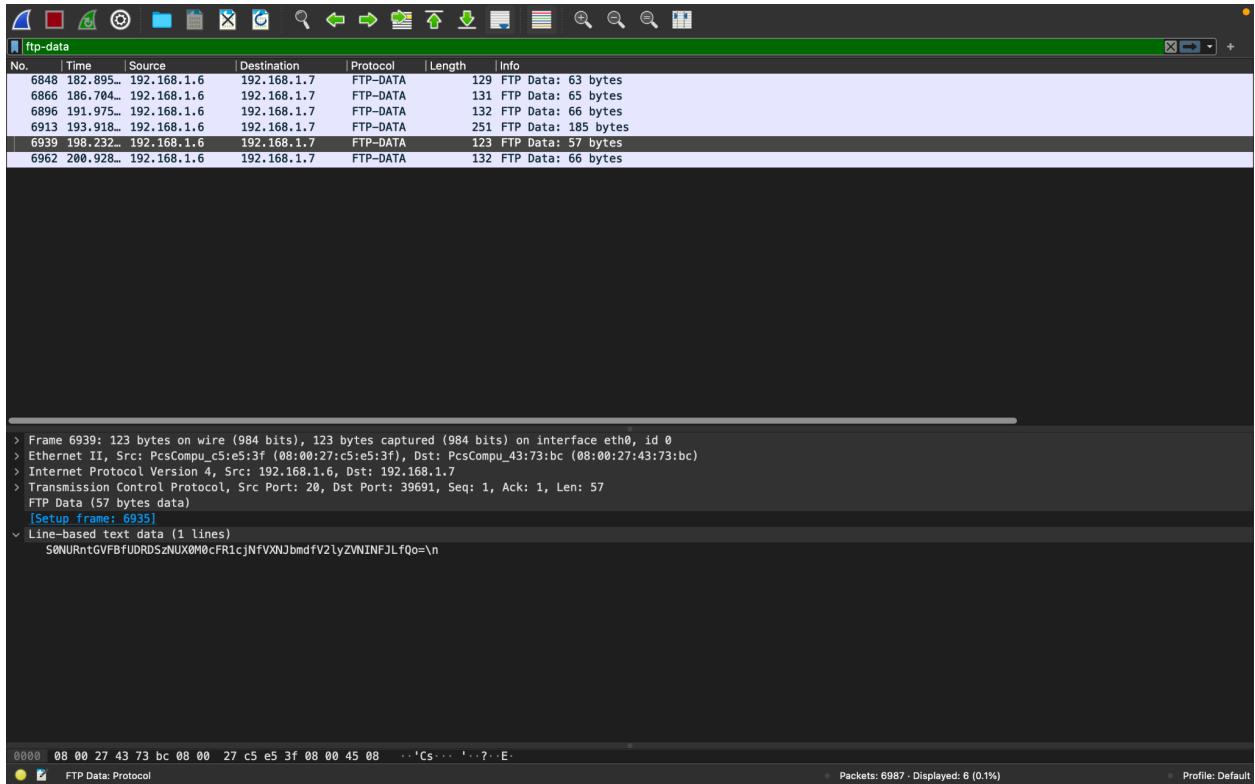


FLAG - KCTF{A_ShARK_iN_tHe_WiRE}

Networking 2 - Find the flag:

We can directly see that this pcapng file consists of FTP and TCP packets.

Applying the filter : ftp-data



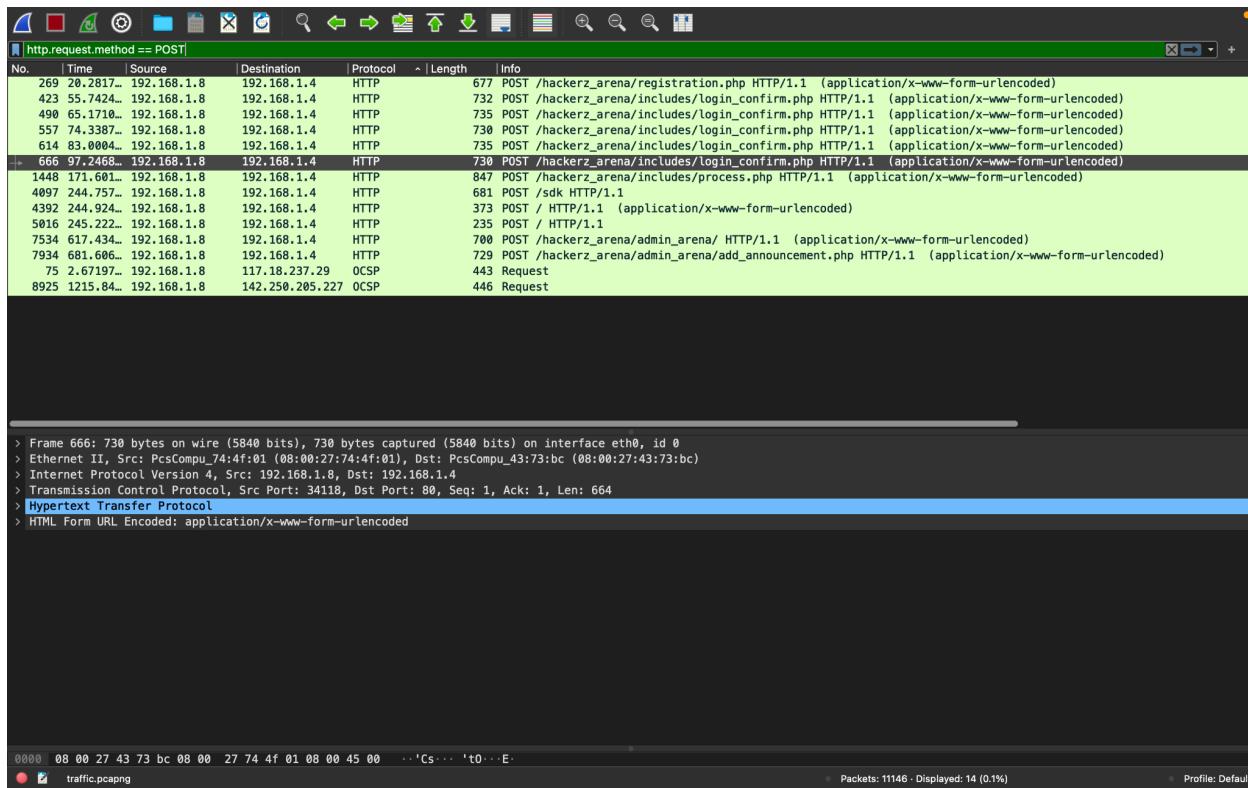
As we can see one packet has a base 64 encoded string. Decoding the string we get the flag.

FLAG - KCTF{FTP_P4CK3T_C4pTur3_UsIng_WireSH4RK}

Networking 3 - Compromised CTF Platform:

In this challenge we are asked to find the username and password. Since it's most likely to be from a login page, we apply the following filter:

http.requestmethod == POST



We can see multiple login attempts and can assume that the last /login-confirm.php must have accepted the hackers credentials.

Following the TCP stream of packet 666, we find the username and password

```
POST /hackerz_arena/includes/login_confirm.php HTTP/1.1
Host: 192.168.1.4
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 81
Origin: http://192.168.1.4
DNT: 1
Connection: keep-alive
Referer: http://192.168.1.4/hackerz_arena/login.php
Cookie: PHPSESSID=e62d714dcc319a54f46a411581c5661f
Upgrade-Insecure-Requests: 1
Sec-GPC: 1

username=demo&password=demo&csrf=31b091e525044d6a2c4d369d0458591a0360e967&submit=HTTP/1.1 302 Found
Date: Wed, 19 Jan 2022 09:29:54 GMT
Server: Apache/2.4.52 (Unix) OpenSSL/1.1.1m PHP/7.4.27 mod_perl/2.0.11 Perl/v5.32.1
X-Powered-By: PHP/7.4.27
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Location: dashboard.php
Content-Length: 0
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

GET /hackerz_arena/includes/dashboard.php HTTP/1.1
Host: 192.168.1.4
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.4/hackerz_arena/login.php
DNT: 1
Connection: keep-alive
Cookie: PHPSESSID=e62d714dcc319a54f46a411581c5661f
Upgrade-Insecure-Requests: 1
Sec-GPC: 1

HTTP/1.1 200 OK
Date: Wed, 19 Jan 2022 09:29:54 GMT
```

Packet 666: 8 client pkts, 112 server pkts, 15 turns. Click to select.

Entire conversation (1291 kB) Show data as ASCII Stream 24

Find: Find Next

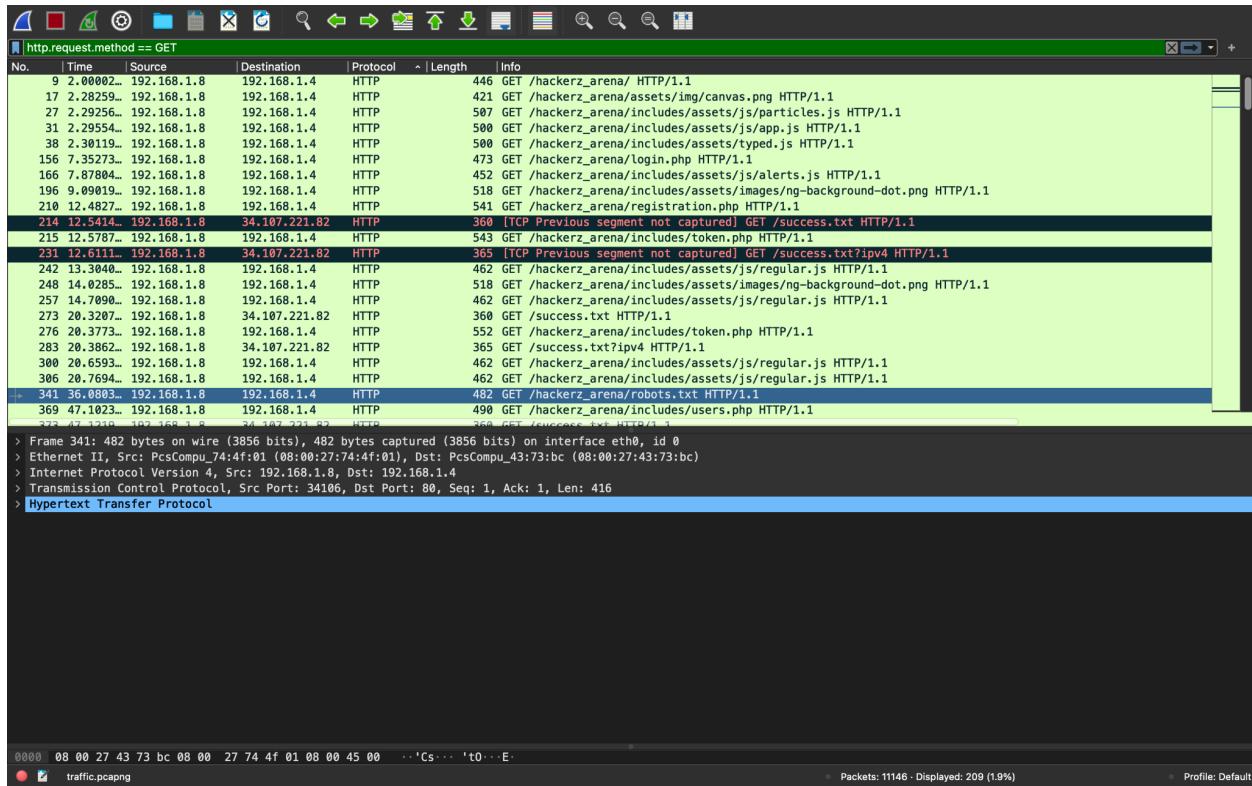
Help Filter Out This Stream Print Save as... Back Close

FLAG - KCTF{demo_demo}

NETWORKING 3.a - Robots.txt

We have to use the same pcapng file from Networking 3. To view the robots.txt file we apply the following filter:

http.requestmethod == GET



Following the TCP Stream we get the path inside robots.txt

```
GET /hackerz_arena/robots.txt HTTP/1.1
Host: 192.168.1.4
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Cookie: PHPSESSID=ae62d714dcc319a54f46a411581c5661f
Upgrade-Insecure-Requests: 1
Sec-GPC: 1

HTTP/1.1 200 OK
Date: Wed, 19 Jan 2022 09:28:52 GMT
Server: Apache/2.4.52 (Unix) OpenSSL/1.1.1m PHP/7.4.27 mod_perl/2.0.11 Perl/v5.32.1
Last-Modified: Wed, 19 Jan 2022 09:12:20 GMT
ETag: "31-5d5ebc878bdb9"
Accept-Ranges: bytes
Content-Length: 49
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/plain

User-agent: *

Disallow : /includes/users.php

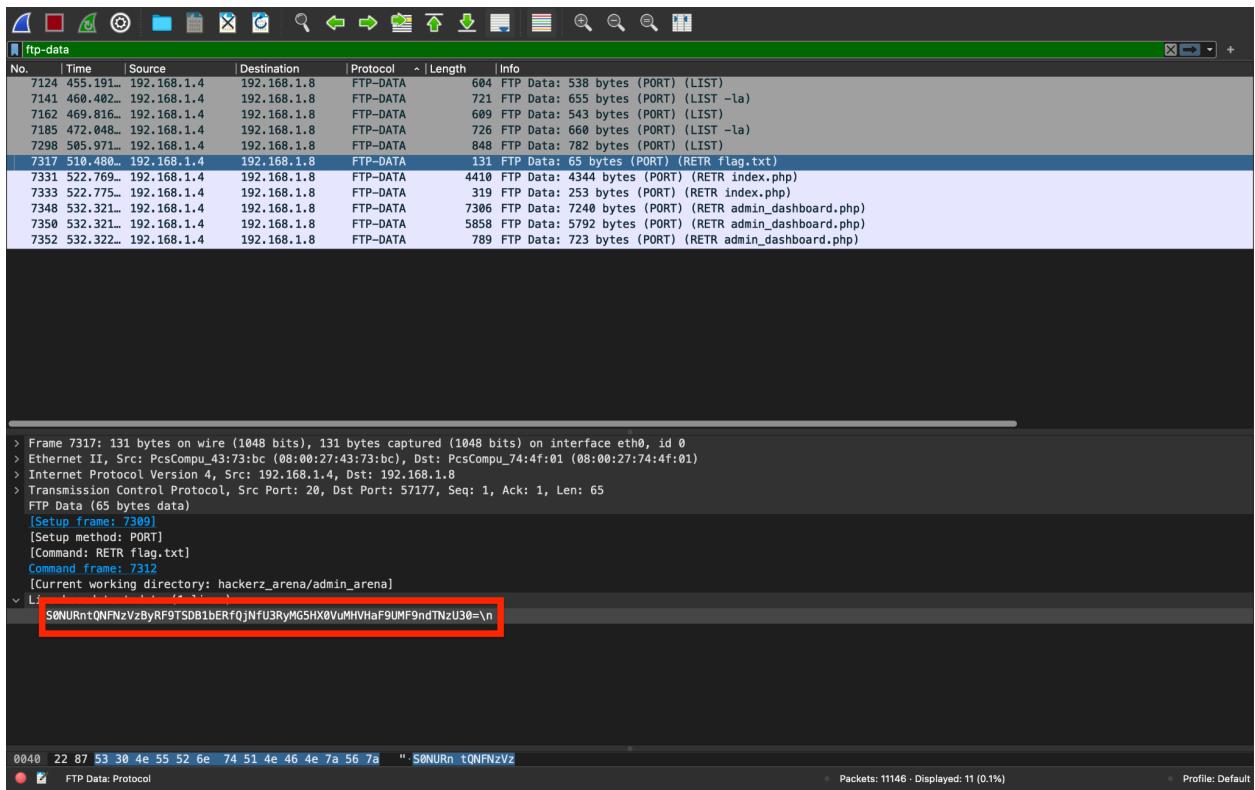
1 client pkt, 1 server pkt, 1 turn.

Entire conversation (802 bytes) Show data as ASCII Stream 18
Find: Find Next
Help Filter Out This Stream Print Save as... Back Close
```

FLAG - KCTF{/includes/users.php}

Networking 3.c- FTP flag

To find the ftp flag we apply the filter : ftp-data



It's base64 encoded again. Decoding it reveals the flag.

FLAG - KCTF{P4SsW0rD_SH0uID_B3_Str0nG_En0uGh_T0_gu3sS}

Networking 3.4- Admin Arena:

We have to find the email and password inside Admin Arena.

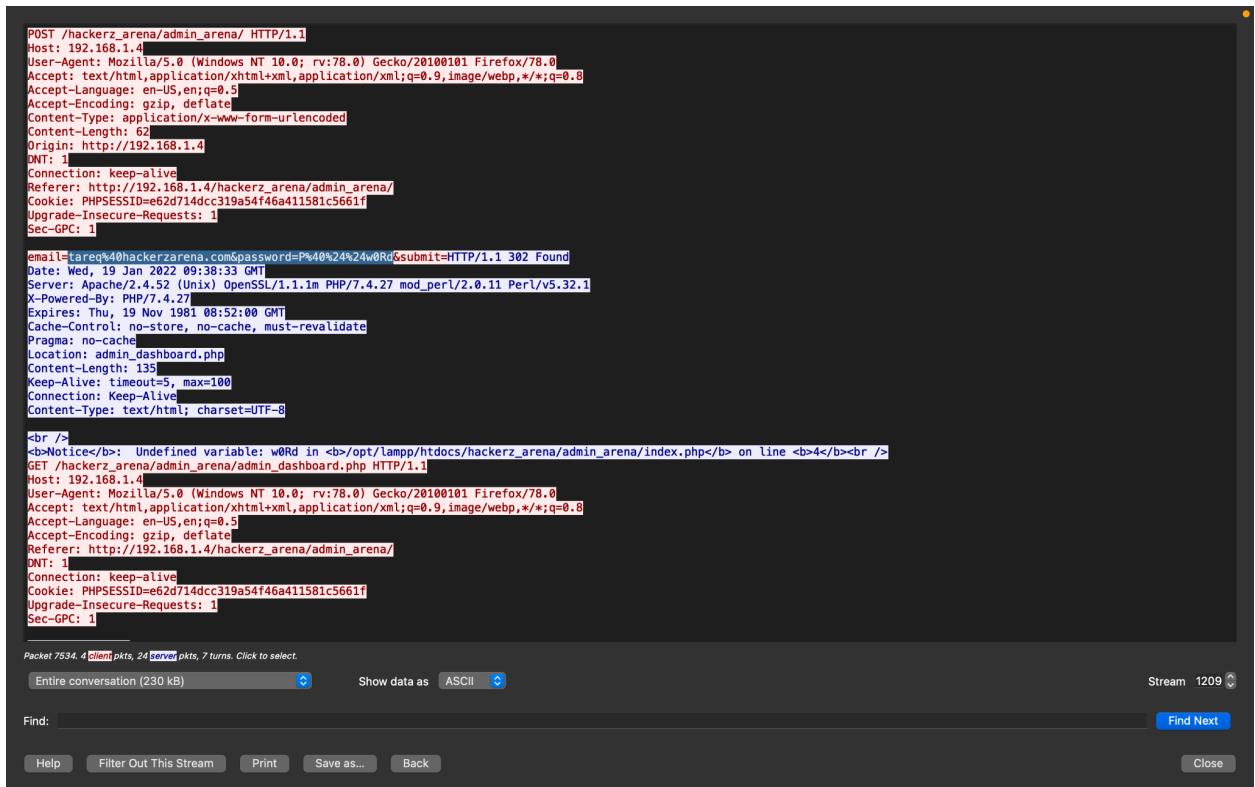
So applying the filter - http.requestmethod == POST, we search for the /admin_arena page:

No.	Time	Source	Destination	Protocol	Length	Info
75	2.67197...	192.168.1.8	137.18.237.29	OCSP	443	Request
269	20.2817...	192.168.1.8	192.168.1.4	HTTP	677	POST /hackerz_arena/registration.php HTTP/1.1 (application/x-www-form-urlencoded)
423	55.7424...	192.168.1.8	192.168.1.4	HTTP	732	POST /hackerz_arena/includes/login_confirm.php HTTP/1.1 (application/x-www-form-urlencoded)
499	65.1710...	192.168.1.8	192.168.1.4	HTTP	735	POST /hackerz_arena/includes/login_confirm.php HTTP/1.1 (application/x-www-form-urlencoded)
557	74.3387...	192.168.1.8	192.168.1.4	HTTP	736	POST /hackerz_arena/includes/login_confirm.php HTTP/1.1 (application/x-www-form-urlencoded)
618	83.0000...	192.168.1.8	192.168.1.4	HTTP	735	POST /hackerz_arena/includes/login_confirm.php HTTP/1.1 (application/x-www-form-urlencoded)
666	97.2468...	192.168.1.8	192.168.1.4	HTTP	730	POST /hackerz_arena/includes/login_confirm.php HTTP/1.1 (application/x-www-form-urlencoded)
1448	171.601...	192.168.1.8	192.168.1.4	HTTP	847	POST /hackerz_arena/includes/process.php HTTP/1.1 (application/x-www-form-urlencoded)
4097	244.757...	192.168.1.8	192.168.1.4	HTTP	681	POST /sdn HTTP/1.1
4392	244.924...	192.168.1.8	192.168.1.4	HTTP	373	POST / HTTP/1.1 (application/x-www-form-urlencoded)
5016	245.222...	192.168.1.8	192.168.1.4	HTTP	235	POST /HTTP/1.1
+ 7534	617.434...	192.168.1.8	192.168.1.4	HTTP	700	POST /hackerz_arena/admin_arena/ HTTP/1.1 (application/x-www-form-urlencoded)
7934	681.606...	192.168.1.8	192.168.1.4	HTTP	729	POST /hackerz_arena/admin_arena/add_announcement.php HTTP/1.1 (application/x-www-form-urlencoded)
8925	1215.84...	192.168.1.8	142.250.205.227	OCSP	446	Request


```
> Frame 7534: 700 bytes on wire (5600 bits), 700 bytes captured (5600 bits) on interface eth0, id 0
> Ethernet II, Src: PcsCompu_74:4f:01 (08:00:27:74:4f:01), Dst: PcsCompu_43:73:bc (08:00:27:43:73:bc)
> Internet Protocol Version 4, Src: 192.168.1.8, Dst: 192.168.1.4
> Transmission Control Protocol, Src Port: 36470, Dst Port: 80, Seq: 1, Ack: 1, Len: 634
< Hypertext Transfer Protocol
  > POST /hackerz_arena/admin_arena/ HTTP/1.1\r\n
    Host: 192.168.1.4\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Content-Type: application/x-www-form-urlencoded\r\n
    Content-Length: 62\r\n
    Origin: http://192.168.1.4\r\n
    DNT: 1\r\n
    Connection: keep-alive\r\n
    Referer: http://192.168.1.4/hackerz_arena/admin_arena/\r\n
    Cookie: PHPSESSID=e62d714cc319a54f46a411581c5661f\r\n
    Upgrade-Insecure-Requests: 1\r\n
    Sec-GPC: 1\r\n
0020  01 04 8e 76 00 50 f0 67  0b 01 5a 5c f9 f1 80 18  ..-v-P-g ..Z\....
```

Packets: 11146 - Displayed: 14 (0.1%) Profile: Default

Following the TCP stream we get the



```
POST /hackerz_arena/admin_arena/ HTTP/1.1
Host: 192.168.1.4
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 62
Origin: http://192.168.1.4
DNT: 1
Connection: keep-alive
Referer: http://192.168.1.4/hackerz_arena/admin_arena/
Cookie: PHPSESSID=e62d714dcc319a54f46a411581c5661f
Upgrade-Insecure-Requests: 1
Sec-GPC: 1

email=tareq%40hackerzarena.com&password=P%40%24%24w0Rd&submit=HTTP/1.1 302 Found
Date: Wed, 19 Jan 2022 09:38:33 GMT
Server: Apache/2.4.52 (Unix) OpenSSL/1.1.1m PHP/7.4.27 mod_perl/2.0.11 Perl/v5.32.1
X-Powered-By: PHP/7.4.27
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Location: admin_dashboard.php
Content-Length: 135
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

<br />
<b>Notice:</b> Undefined variable: w0Rd in <b></b>/opt/lampp/htdocs/hackerz_arena/admin_arena/index.php<b></b> on line <b>4</b><br />
GET /hackerz_arena/admin_arena/admin_dashboard.php HTTP/1.1
Host: 192.168.1.4
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.4/hackerz_arena/admin_arena/
DNT: 1
Connection: keep-alive
Cookie: PHPSESSID=e62d714dcc319a54f46a411581c5661f
Upgrade-Insecure-Requests: 1
Sec-GPC: 1
```

Email - **tareq%40hackerzarena.com**

Password - **P%40%24%24w0Rd**

We can see that these are not the final credentials. Hex text is inserted in the password.

% is blank

40 is @

24 is \$

FLAG - KCTF{tareq@hackerzarena.com_P@\$\$w0Rd}

OSINT 1 - Canada Server:

Searching for NS Tech Valley on google I came across their facebook page. After inspecting all their posts I found a name -



NS TechValley · December 1, 2021

Hello Everyone
I hope you all are well and safe.
We recently moved our Ads Explosives mailer site to a new server and for that our website will be offline for a few hours to propagate the new DNS. We are very sorry for the downtime.

Best Regards
Md. Moniruzzaman Prodhan

#AdsExplosives #NSTechValley

ADS EXPLOSIVES

Get more traffic to your sites!

Join Free

Use the power of email marketing to grow your business

Like Comment Share

Searching the name on google with the word “canada” to take advantage of google’s search algorithm we found the website -

<https://clients.nstechvalley.com/announcements/3>

The screenshot shows the NSTechValley portal interface. On the left, there's a sidebar with links for Home, Store, Announcements, Knowledgebase, Network Status, Affiliates, and Contact Us. The main content area has a header with the NSTechValley logo and a search bar. Below the header, a breadcrumb navigation shows 'Portal Home / Announcements' followed by a right-pointing arrow and 'Canada Server [192.99.167.83] Issue'. The main content is titled 'Canada Server [192.99.167.83] Issue'. It contains a message from 'Hello Everyone' stating: 'I hope you all are fine and safe. We are very sorry to say that we have some issues with the cPanel of Canada Server [192.99.167.83]. We are working to resolve the issue as soon as possible. For now you can't login to your cPanel but your website will be online. We are very sorry for this unwanted issue.' The message is signed off with 'Best Regards' and 'Md. Moniruzzaman Prodhan' on 'Wednesday, December 1, 2021'. A blue '« Back' button is at the bottom of the message box.

FLAG - KCTF{192.99.167.83}

OSINT 3 - Find the camera:



© JenCh012

In the image of the given bus we see the name of the photographer - **JenCh012**

After searching for her online it seemed impossible to find this exact picture as JenCh012 had apparently taken millions of pictures of buses.

So to make things easier this time I searched with - JenCh012 QV 6227 (number plate of the bus), and yes I found the image online.

Luxembourg, Van Hool New A308 # 206

Location:	Lëtzebuerg – Esch-Uelzecht
Facility:	Tramways Intercommunaux dans le Canton d'Esch
License Plate #:	QV 6227
Model:	Van Hool New A308
Since....:	07.2007
Built:	07.2007
Serial number:	63942
VIN:	YE230802N94M63942
Current state:	Sent to other company (or to the factory) (10.2017)
Purpose:	Passenger vehicle

Camera Settings

Model:	DSC-S980
Date and Time:	15.05.2010 15:51
Exposure Time:	1/320 sec
Aperture Value:	5.6
ISO Speed:	100
Focal Length:	23.2 mm

[Show all EXIF tags](#)

So the model name was DSC-S980. Searching the same on google I found out that the manufacturing company was Sony.

FLAG - KCTF{SONY_DSC_S980}

Cryptography 1 - Passwd:

```
root:x:0:0:root:/root:/usr/bin/zsh
bin:x:1:1:::/usr/bin/nologin
daemon:x:2:2:::/usr/bin/nologin
mail:x:8:12::/var/spool/mail:/usr/bin/nologin
ftp:x:14:11::/srv/ftp:/usr/bin/nologin
http:x:33:33::/srv/http:/usr/bin/nologin
nobody:x:65534:65534:Nobody:/usr/bin/nologin
dbus:x:81:81:System Message Bus:/usr/bin/nologin
systemd-journal-remote:x:988:988:systemd Journal Remote:/usr/bin/nologin
systemd-network:x:987:987:systemd Network Management:/usr/bin/nologin
systemd-oom:x:986:986:systemd Userspace OOM Killer:/usr/bin/nologin
systemd-resolveix:x:984:984:systemd Resolver:/usr/bin/nologin
systemd-timesyncx:x:983:983:systemd Time Synchronization:/usr/bin/nologin
systemd-coredumpx:x:982:982:systemd Core Dumper:/usr/bin/nologin
uiddd:x:68:68::/usr/bin/nologin
avahi:x:980:980:Avahi mDNS/DNS-SD daemon:/usr/bin/nologin
named:x:40:40:BIND DNS Server:/usr/bin/nologin
brltty:x:979:979:Braille Device Daemon:/var/lib/brltty:/usr/bin/nologin
colordi:x:978:978:Color management daemon:/var/lib/colord:/usr/bin/nologin
cupsix:x:209:209:cups helper user:/usr/bin/nologin
dhcpcd:x:977:977:dhcpcd privilege separation:/usr/bin/nologin
dnsmasq:x:976:976:dnsmasq daemon:/usr/bin/nologin
git:x:975:975:git daemon user:/usr/bin/git-shell
mpd:x:45:45::/var/lib/mpd:/usr/bin/nologin
nbd:x:974:974:Network Block Device:/var/empty:/usr/bin/nologin
nm-openvpn:x:973:973:NetworkManager OpenVPN:/usr/bin/nologin
nvidia-persistenced:x:143:143:NVIDIA Persistence Daemon:/usr/bin/nologin
openvpn:x:972:972:OpenVPN:/usr/bin/nologin
partimag:x:110:110:Partimage user:/usr/bin/nologin
polkitd:x:102:102:PolicyKit daemon:/usr/bin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/usr/bin/nologin
rtkit:x:133:133:RealtimeKit:/proc:/usr/bin/nologin
sddm:x:971:971:Simple Desktop Display Manager:/var/lib/sddm:/usr/bin/nologin
tss:x:970:970:tss user for tpm2:/usr/bin/nologin
usbmux:x:140:140:usbmux user:/usr/bin/nologin
junior: : : : : /home/junior:/bin/zsh
knight: :708697c63f7eb369319c6523380bdf7a: : /home/junior:/bin/zsh
```

We can see that the password of knight is a hash. It is of the type MD5. Using any of the tools to convert the hash into simple text we get the flag.

FLAG - KCTF{exploit}

Cryptography 3 - Jumble:

We are provided with the encoded string and an enc.py file. It's obvious that the enc.py file was used to encode the string. Understanding the code, I wrote a python file to reverse or decode the string.

The screenshot shows the PyCharm IDE interface. The project structure on the left includes files like main.py, lab2_q1_20bce1554.py, lab2_q2_20bce1554.py, test4.py, nikto.py, and test2.py. The code editor on the right contains the following Python script:

```
def f(t):
    c = list(t)
    for i in range(len(t)-2, -2, -1):
        for j in range(len(t)-2, i, -1):
            c[j], c[j+1] = c[j+1], c[j]
    return ''.join(c)

n = input()
print(f(n))
```

The Run tab at the bottom shows the command: /usr/local/bin/python3.9 /Users/atharva/PycharmProjects/test2/venv/lib/test4.py. The output window displays the decoded string: S0NURnt5MHVfZzB0X20zfQ==, which is highlighted with a red box. Below the output, it says "Process finished with exit code 0". A notification bar at the bottom right indicates "PyCharm 2020.3.5 available" with a "Update..." button.

The decoded string received looks base64 encoded. Decoding the string gives the flag.

FLAG - KCTF{y0u_g0t_m3}

Programming 1 - Keep Calculating

x= 1

y= 2

ans = 0

for x in range(1,666+1):

 num = (10*x) + y

 ans += (x*y) + num

print(ans)

Ans = 2666664

FLAG = KCTF{2666664}

Programming 2 - Time complexity

procedure max (a1, a2, ..., an: integers)

max := a1

for i :=2 to n

if max < a1 then max := ai

{max is the largest element}

FLAG - KCTF{O(n)}

Programming 3 - Reverse the answer

```
x= 1  
calc = 0  
ans = 0  
  
for x in range(1, 543+1):  
    calc = (x*(x+1)) + (2 *(x+1))  
    tmp = str(calc)  
    rev_calc = int(tmp[::-1])  
    if(rev_calc%4 == 0):  
        ans += rev_calc  
  
print(ans)
```

Ans = 12252696

FLAG - KCTF{12252696}

Programming 4- Square Sum

Limit is 159 as its square is just above 25000.

```
for x in range(159):  
    for y in range(159):  
        if ((x*x) + (y*y)) == 25000:  
            print(str(x) + " " + str(y))
```

FLAG - KCTF{90,130}

Programming 5 - Something in common

Found the GCD using online calculator 😊

GCD = 4305125
Sum of its integers = 20
Ans = 20*1234

Ans = 24860

FLAG = KCTF{24860}

Programming 6 - Find the number

```
def G_sum(n):  
    if n < 0:  
  
        return 0  
    else:  
  
        return (1/pow(2,n)) + G_sum(n-1)  
print(G_sum(25))
```

Ans = 1.9999999701976776

FLAG = KCTF{1.9999999701976776}

Programming 8 - Loop in a loop

I knew the flag format, hence I knew the basic positioning scheme. So i could decode it manually 😊

FLAG = KCTF{b451c_pr06r4mm1ng}

Programming 7 - Run the code

The code looks like a 8086 code. So running it in my 8086 emulator gives me the desired output.

The screenshot shows the emu8086 interface. The top window is titled "emu8086 - assembler and microprocessor emulator 4.08". It has a menu bar with "file", "edit", "bookmarks", "assembler", "emulator", "math", "ascii codes", and "help". Below the menu is a toolbar with icons for "new", "open", "examples", "compile", "emulate", "calculator", "convertor", "options", "help", and "about". The main area displays the following assembly code:

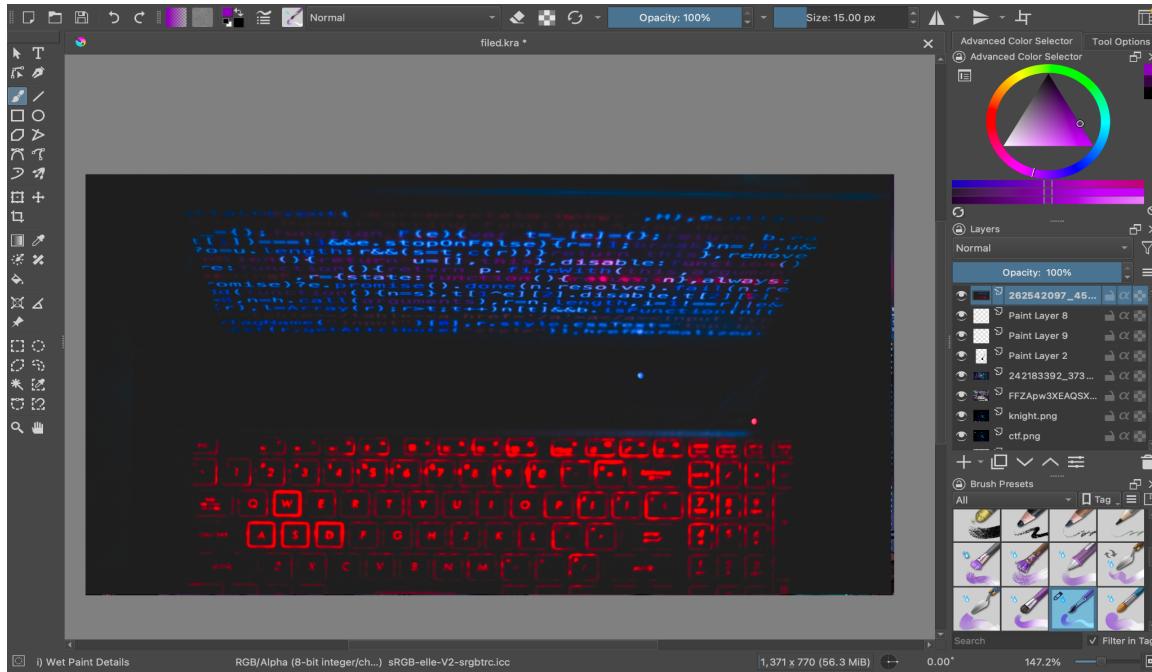
```
11    MOU DL, 100
12    SUB DL, 16
13    INT 21H
14    MOU DL, 100
15    SUB DL, 30
16    INT 21H
17    MOU DL, 123
18    INT 21H
19    MOU DL, 75
20    ADD DL, 50
21    SUB DL, 60
22    INT 21H
23    MOU DL, 53
24    INT 21H
25    MOU DL, 53
26    INT 21H
27    MOU DL, 147
28    SUB DL, 96
29    INT 21H
30    MOU DL, 80
31    SUB DL, 3
32    INT 21H
33    MOU DL, 255
34    MOU DH, 157
35    SUB DL, DH
36    INT 21H
37    MOU DL, 255
38    MOU DH, 147
39    SUB DL, DH
40    INT 21H
41    MOU DH, 72
42    MOU DL, 17
43    ADD DL, DH
44    INT 21H
45
46    MOU DL, 130
47    SUB DL, 5
48    INT 21H
49
50    MOU AH, 4CH
51    INT 21H
52
53
54    MAIN ENDP
55 END MAIN
```

Below this is a smaller window titled "emu8086 screen (80x25 chars)". It contains the text: "KCTF{A553Mb1Y}".

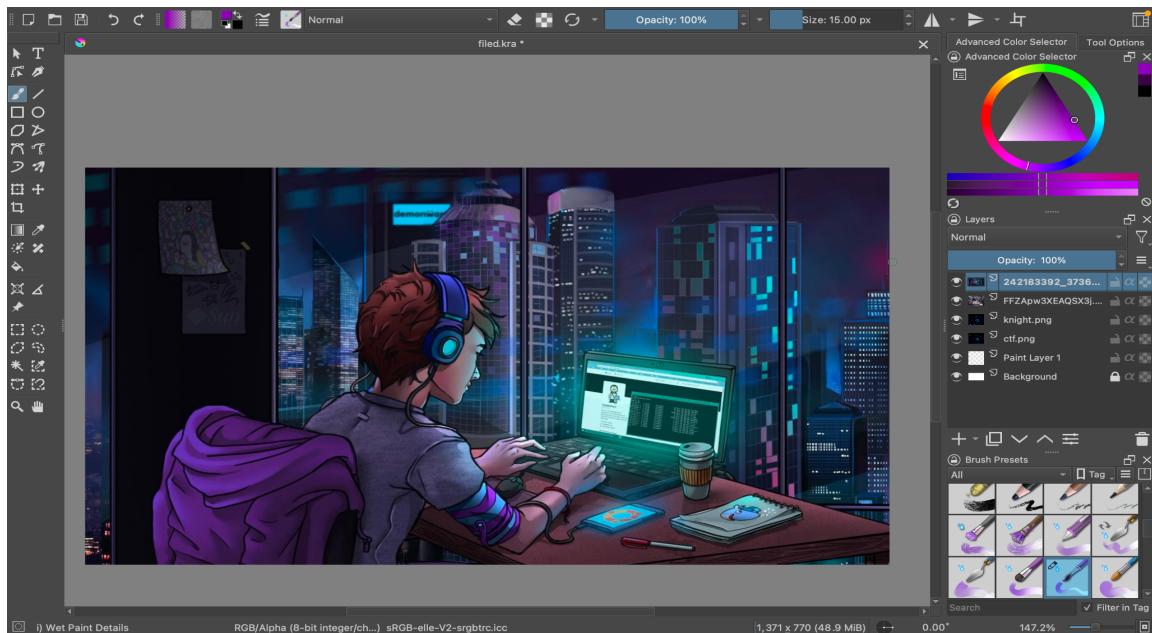
FLAG - KCTF{A553Mb1Y}

Steganography 1 - File D

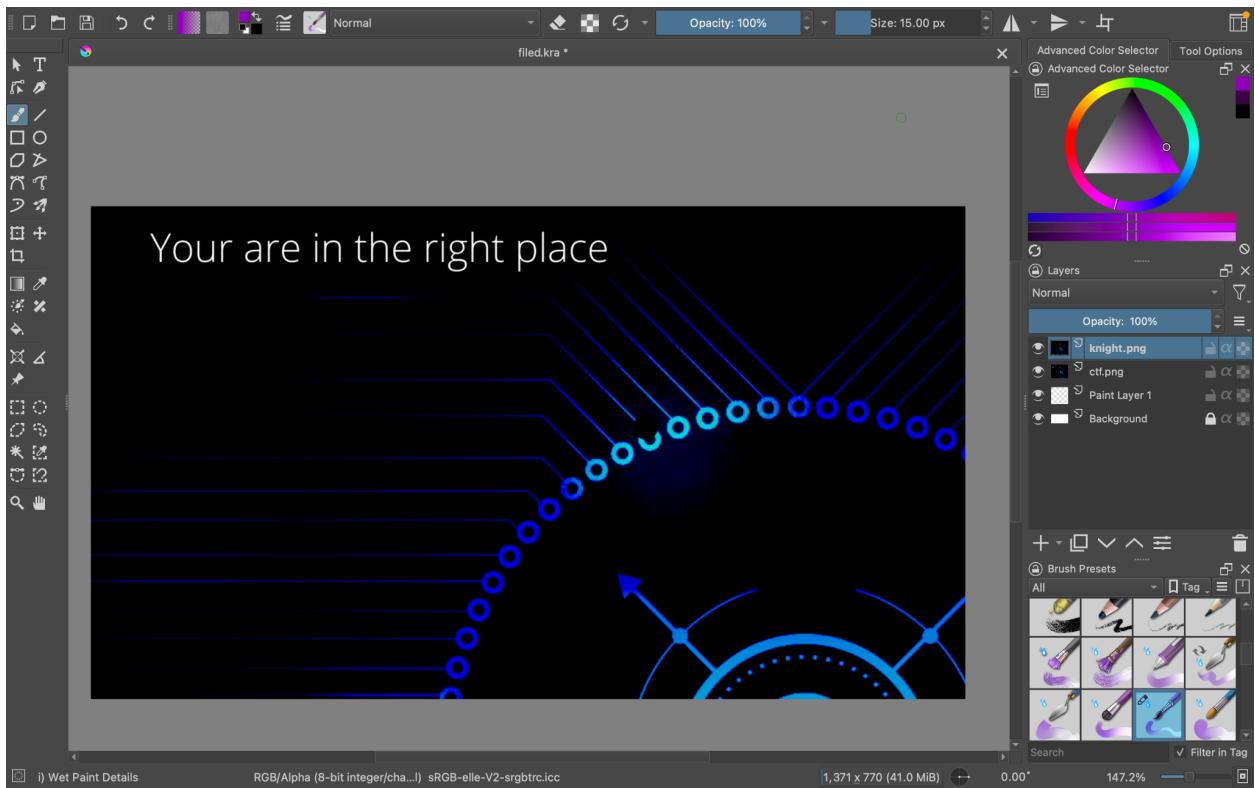
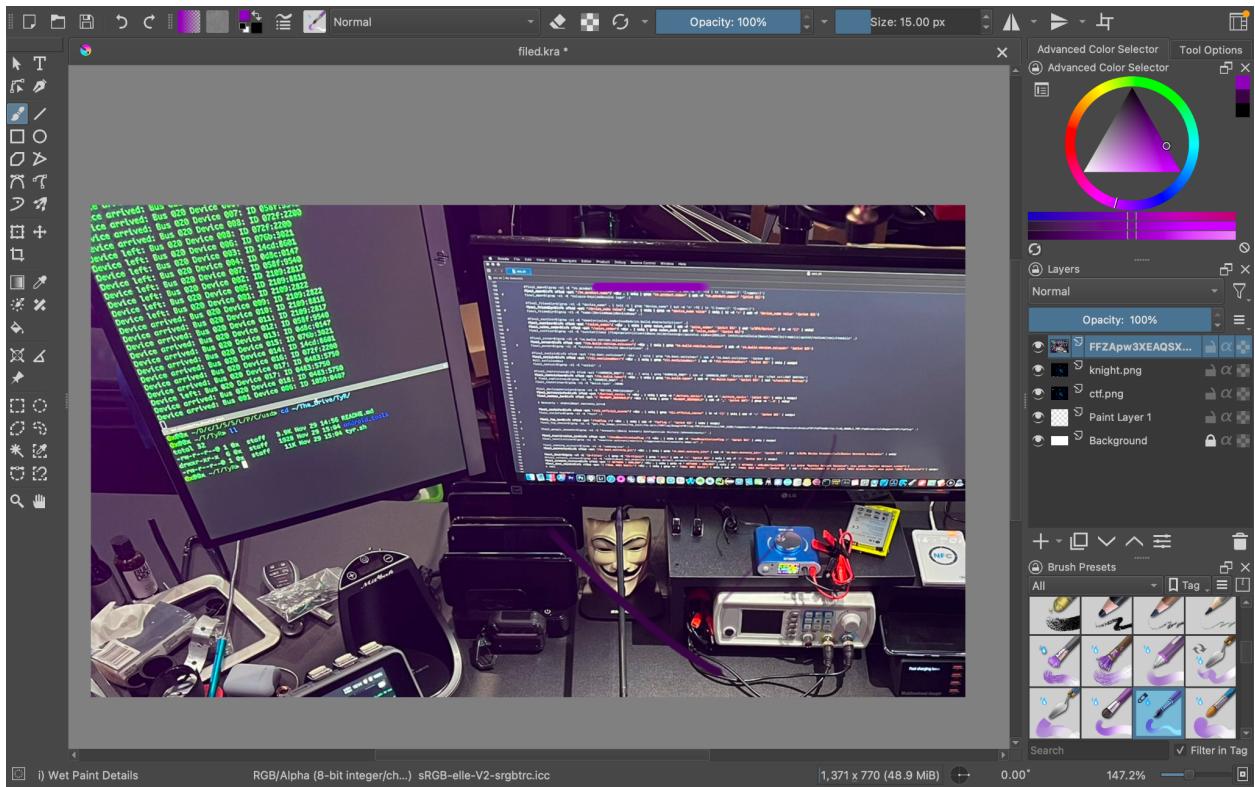
We get a .kra file. I installed krita, an application specifically made to open .kra files. Opening the file -



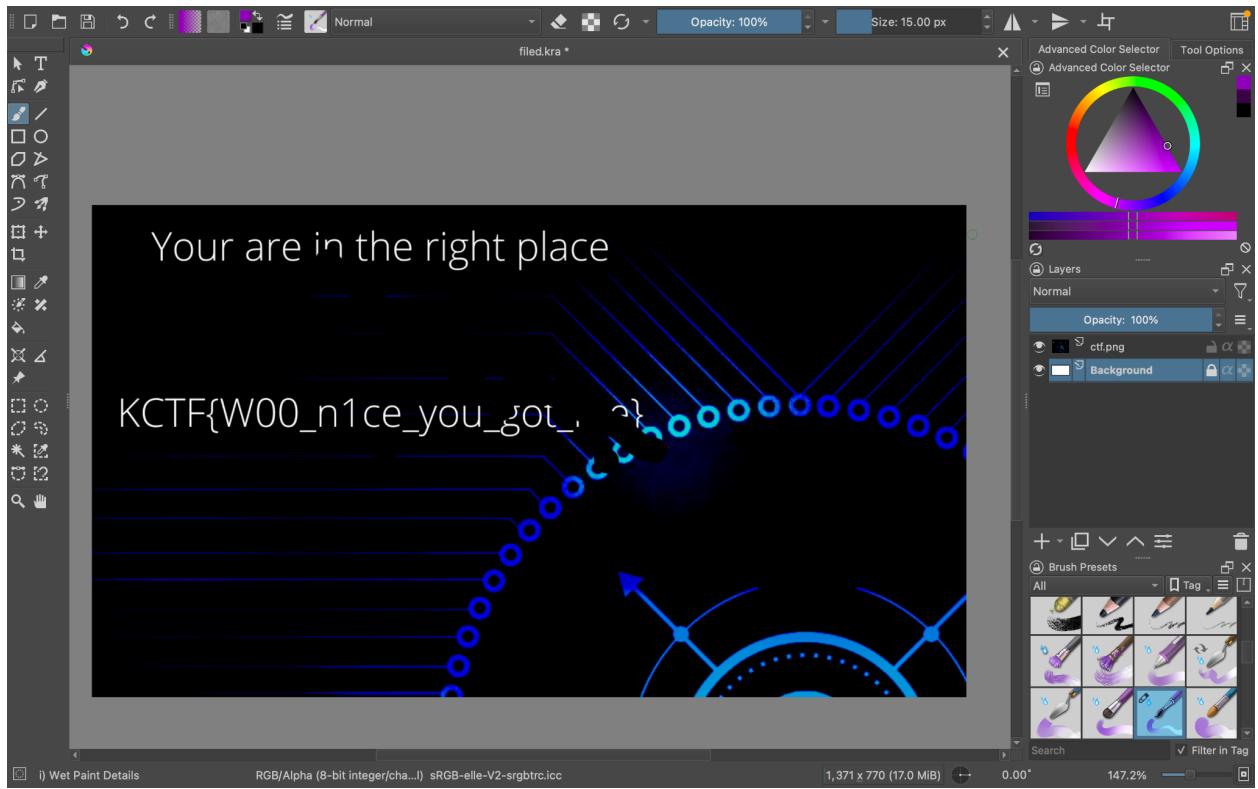
I can see on the extreme right side of the image, there seems to be another image. It seems like there's an image beneath this image. So removing the image on top -



Having gotten the idea on how to move forward I start removing layers of pictures one by one -



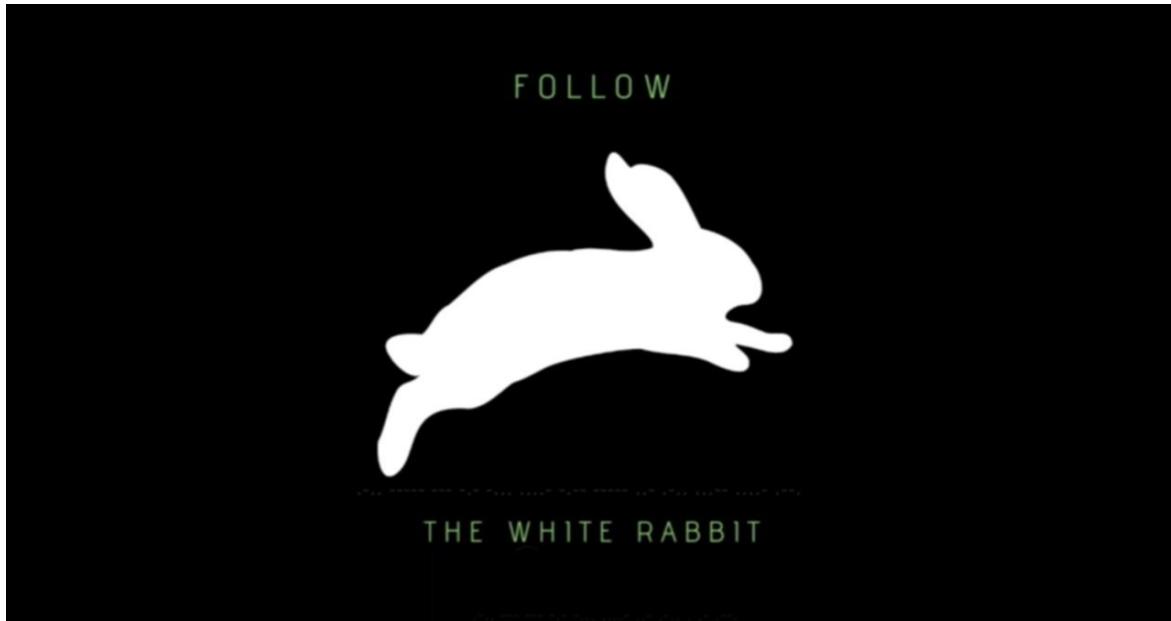
And finally -



FLAG - KCTF{W00_n1ce_you_got_me}

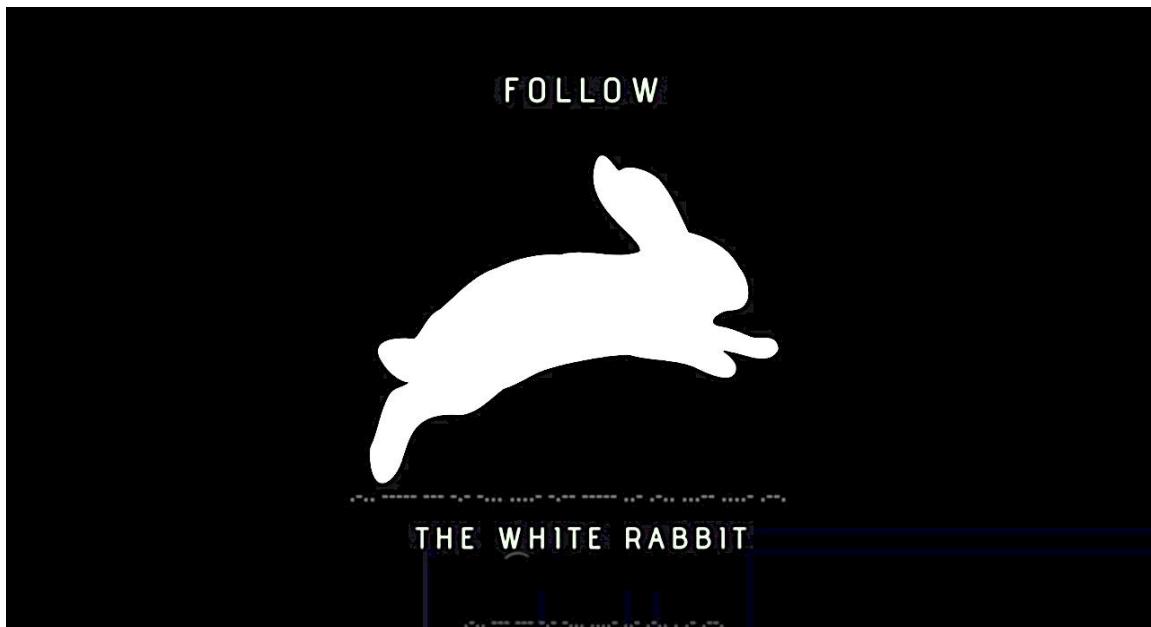
Steganography 2 - Follow the White rabbit

A photo with a white rabbit was given to us.



On closer inspection I felt like there was something in the background, just above the text "THE WHITE RABBIT"

So I changed the contrast and sharpness of the image and got something interesting.



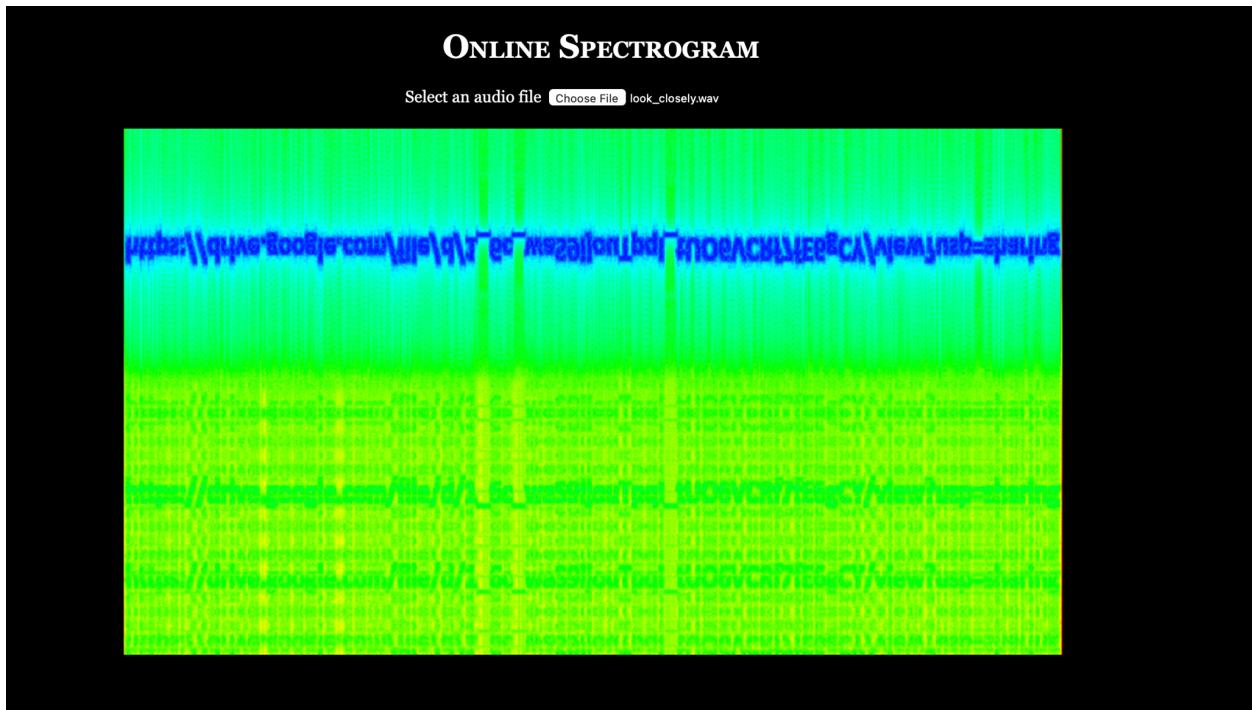
It was morse code. Translating the morse code above the text, “THE WHITE RABBIT” we got -
LOOKB4ULEAP

And translating the one below the text we got - L0OKB4Y0UL34P

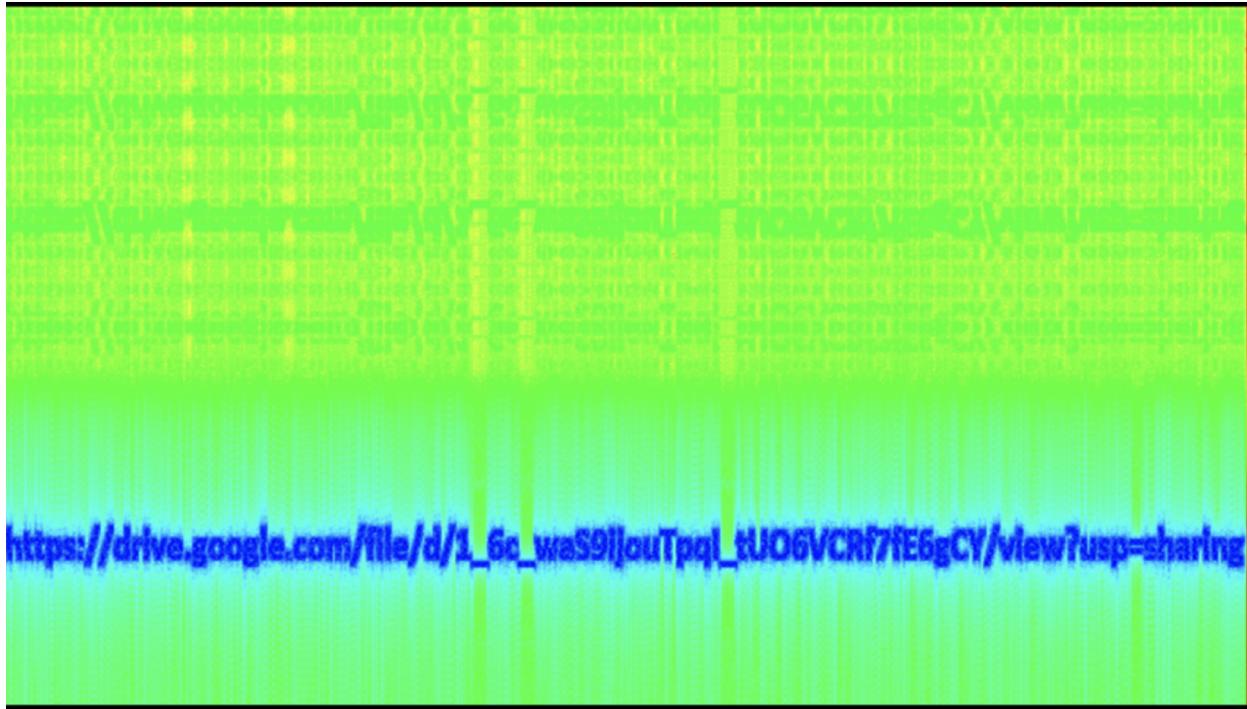
FLAG - {L0OKB4Y0UL34P}

MISC 4 - Look closely:

We were given a .wav file. I went forward to an online spectrogram analyzer and uploaded the .wav file there.

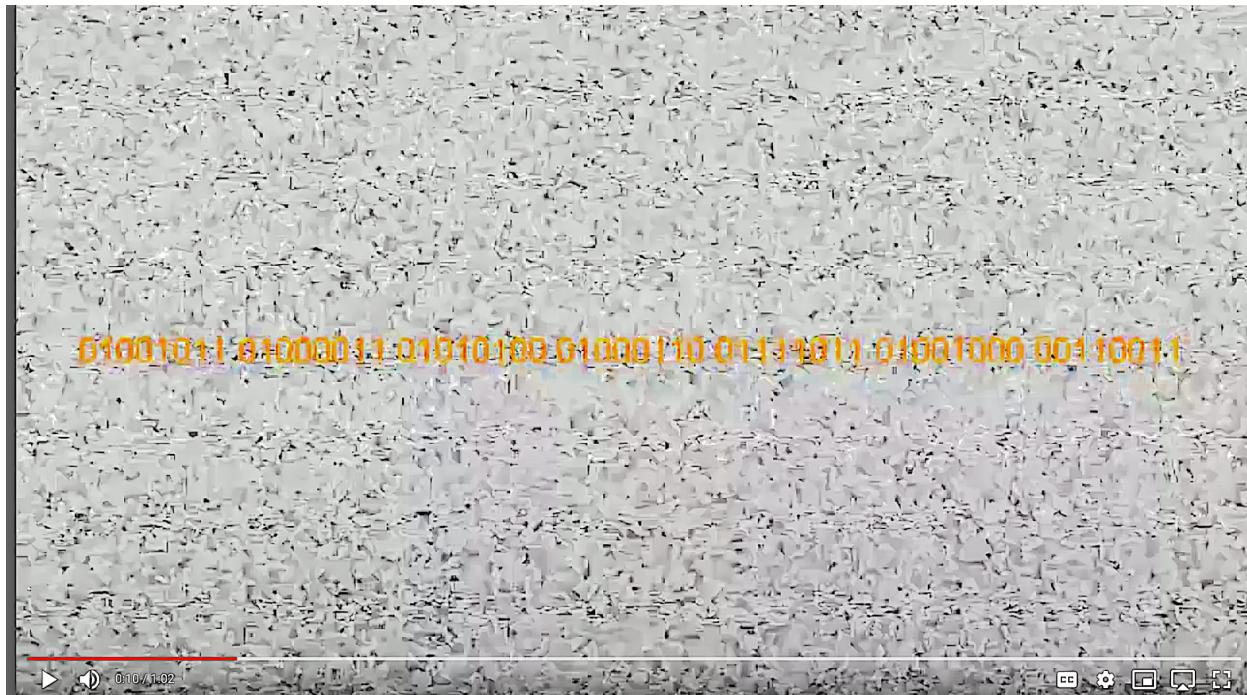


Since it was an inverted image, I had to mirror image it vertically -



I got a link! - https://drive.google.com/file/d/1_6c_waS9ijouTpql_tUO6VCRf7fE6gCY/view?usp=sharing

There we had a video where upon closer look we could see some blurred yellow text coming for a second. I screenshotted the 2 occurrences and adjusted the colours to reveal the text.





I got a binary - 01001011010000110101010001000110011110110100100000110011
010011000100110001001111010111110100101000110011001001100010011110111110
1

Converting it to text got me the flag.

FLAG - KCTF{H3LLO_J3LLO}