

# GREP CTF 2023 WP

Author: A1andNS

The screenshot shows the GREP CTF 2023 dashboard. At the top, it says "Ended". Below that, it shows the competition started at 07:00 PM, 01 Apr and ended 16 minutes ago at 07:00 PM, 03 Apr. On the right, the "My Progress" section displays a rank of 29th, a total score of 2540 pts, and 24 flags found. There are also links to "View Full Profile" and "Invite Members".

## Misc

### Approved!

Description: Finally, the clock ticks 3:48 am and I got approved. **TIME** flies when you are playing **CTF**. I welcome you all to GREP CTF, enjoy !! Oops, approved of what and where 😢, I leave that to you, gn....

According to the description of the challenge, We can know the flag is about GREP CTF in CTFTIME. So you can easily find out this CTF in CTFTIME in [CTFtime.org / GREP CTF](#)

You can get a hexadecimal encode string, just decode it. You can get the flag.

The screenshot shows the CTFTIME website with the GREP CTF page. It includes a navigation bar with "CTF TIME", "CTFs", "Upcoming", "Archive", "Calendar", "Teams", "FAQ", "Contact us", "About", and "Sign in". Below the navigation is a breadcrumb trail: "Home / CTFs / GREP CTF". The main content area features the title "GREP CTF" and a "BITS & PIECES" logo. It also includes a note about the official URL: "Official URL: <https://grep.ctf.eng.run/register>". A message says "A CTF from team Bits & Pieces, BITS Pilani. It is aimed for beginners but anyone can participate. Hope you enjoy it." Below this is a redacted URL "475245507657336c63306d655f74305f475233505f4354467d". The bottom of the page shows a tool interface with various tabs like "LOAD", "SPLIT", "EXECUTE", "TEST", "SQLI", "XSS", "LFI", "SSRF", "SSTI", "SHELL", "ENCODING", "HASHING", "MODE", and "THEME". A red box highlights the value "GREP{W3lc0me\_t0\_GR3P\_CTF}" in the "EXECUTE" tab's input field.

# esoF\*ck

DESCRIPTION: I've heard about brainf#ck but what the f#ck js this?

They are not a standard jsfuck encoded string, so change "F#ck" to null. Then jsunfuck the encoded string get a text via [CoderTab - JSUnFuck - Decode JSFuck Here](#)

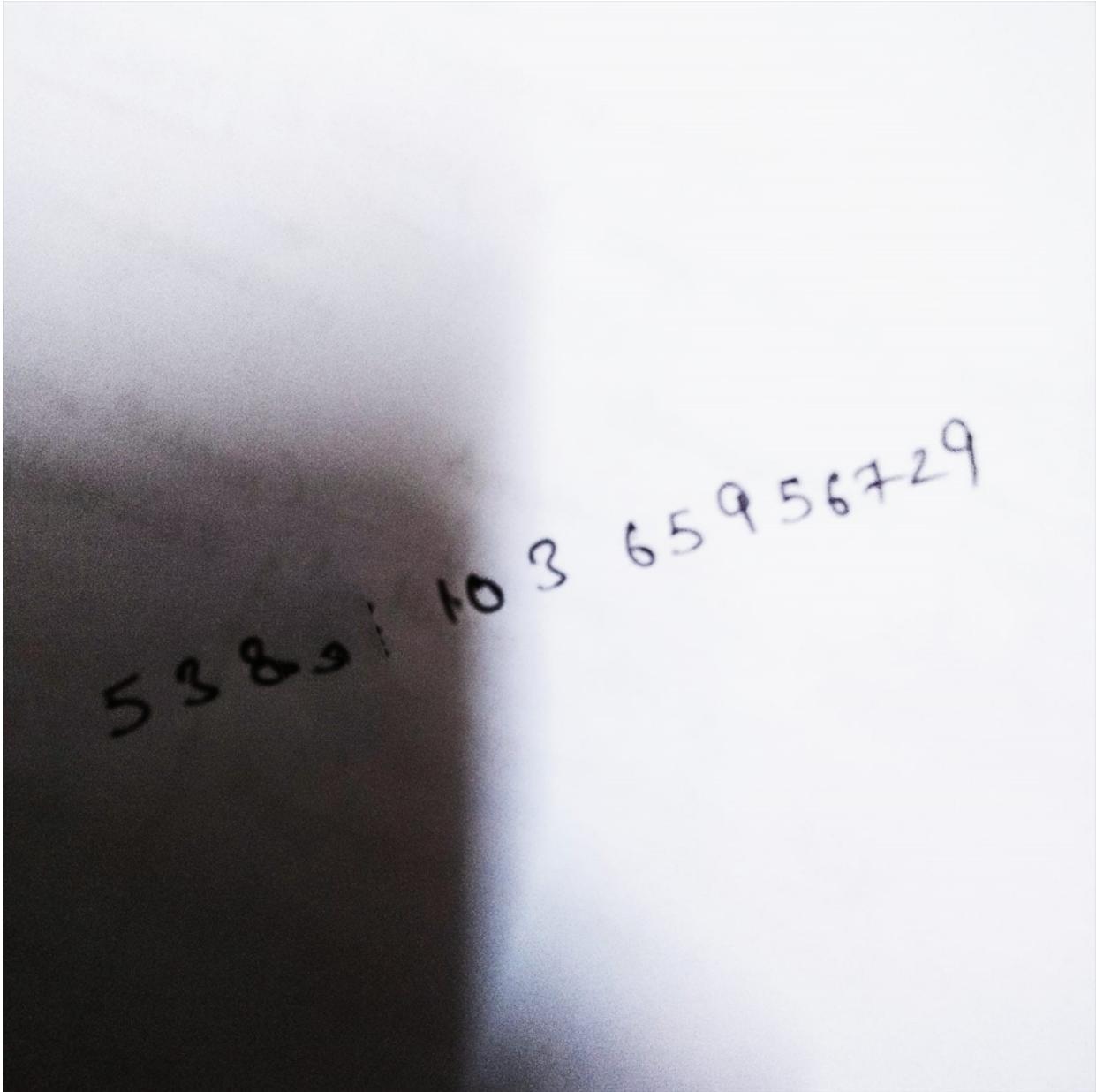
The screenshot shows the CoderTab interface with the following details:

- Title Bar:** msg.txt
- File Menu:** 文件 编辑 查看
- Search Bar:** F#ck
- Replace Dialog:** 替换
- Text Area:** The main text area contains a highly encoded JSFuck string. A search and replace operation is being performed to convert all occurrences of "F#ck" to null. The replaced text is visible at the bottom of the area.
- Status Bar:** 行 1, 列 10265 | 100% | Windows (CRLF) | UTF-8
- Bottom Panel:** JSUnFuck
- Information Bar:** JSFuck is an esoteric programming style of Javascript developed by Marcin Kiepke, where code is written using a very limited set of characters: [, ], , , +, .
- Help Text:** In order to decode JSFuck successfully, [Eval Source] option must be checked on encoded script.  
Using Mozilla Firefox or Google Chrome for best result!
- Links:** More Info, JSFuck Website, JSFuck Wikipedia Information

## Lost Card

DESCRIPTION: Your card got lost while eating at the ANC. You have written it somewhere, but the two digits are unclear.

Firstly, make a mirror of the picture, we can obtain the result follow the picture. Then analyze the structure of the digits, you will find the No.4 digit is lacking the up part. Then, we will find number maybe is 8,3,6,5 by similar. we can easily find The No.5 digit is 1. So have a try, it's lucky that the number is 8, so we get the flag: GREP{5388110365956729}



5389 /  $10^3$  65956729

## Layout

DESCRIPTION: I've got this weird keyboard which has all the keys messed up. It's from someone called the workman.

use the automated cryptogram solver in [quipqiup - cryptoquip](#) and [cryptogram solver](#)

# quipqiup beta3

quipqiup is a fast and automated cryptogram solver by [Edwin Olson](#). It can solve simple substitution ciphers often found in newspapers, including puzzles like cryptoquips (in which word boundaries are preserved) and patristocrats (inwhi chwor dboun darie sareñ).

## Puzzle:

```
Uph:saj us wumpmyrour ptt pr ir,akn up oo gorr op jpr akn  
U! hrcasbabuka, lprv byk rcrw hrjpkswabukg  
Ypd bp gucr a lpbjwrtfmeuk' afhurkmr a troukq ouer ub's orcubabukg  
Krcrw tahuuk, akh U ekpd bry yabrwis awr tpwrcrw daubukg  
Tpw bry haj byab bynj mak saj U troo ptt, bynj'oo vr mrnorwabukg  
Jpfw toag us wdk:g0h,rl1k3l_3256gh62 uk bry fsao tpwlab
```

CLUES: For example G-R QVW=THE

Solve

automatically selected statistics mode; you can override by using the drop down menu next to the solve button.

0	-1.957	Uh, summa-lumma, dooma-lumma, you assumin' I'm a human What I gotta do to get it through to you I'm suerhuman+ Innovative and I'm made of rubber so that anything You say is ricochetin' off of me, and it'll glue to you and I'm devastating, more than ever demonstrating How to give a motherfuckin' audience a feeling like it's levitating Never fading, and I know the haters are forever waiting For the day that they can say I fell off, they'll be celebrating Your flag is r4:g0d_em1n3m_3256gd62 in the usual format
1	-3.695	Ud, tumma-lumma, kooma-lumma, you atnumis' I'm a dumas Gdan I honna ko no hen in ndrouhd no you I'm tu:erdumas+ Issovanie ask I'm make of rubber to ndan asyndish You tay it ricocdenis' off of me, ask in'll blue no you ask I'm kevadhanish, more nidas ever kemostrianish Dog no hive a monerfuctis' aukieisce a feelish lite in't levinanish Sever falkish, ask I tsog nze danerd are forever gainish For nze kay ndan ndey cas tay I fell off, ndey'll be celebriahsh Your flah id r4:h0k_em1s3m_3256hk62 is nze udual forman
2	-3.733	Uz, dumma-lumma, kooma-lumma, you addumis' I'm a zumas Gzan I honna ko no hen in nzrouzh no you I'm duerzuma+ Issovanie ask I'm make of rubber do nzam asyndish You day id ricoczenis' off of me, ask in'll blue no you ask I'm kevadhanish, more nidas ever kemostrianish Zog no hive a monerfuctis' aukieisce a feelish lite in'd levinanish Sever falkish, ask I tsog nze zanerd are forever gainish For nze kay nzan nzez cas day I fell off, nzeyle'll be celebriahsh Your flah id r4:h0k_em1s3m_3256hk62 is nze udual forman
3	-3.871	Uk, dumma-lumma, tooma-lumma, you addumis' I'm a kumac Pkan I sonna to no sen in nkrousk no you I'm duerkumac+ Iccovanie act I'm mate of rubber do nzam acynkics You day id rihothkenic' off of me, act in'll sue no you act I'm tevadynamics, more nkaç ever temocdranics Kop no sive a monkertuhigic' autieche a feels ligc in'd levinanish Cever fatices, act I aron nka kanerd are forever gainish For nze taw nkan nekv har dav I fell off, nekv'll be helebranics Your flah id r4:sot_em1c3m_3256st62 ic nke

By observing the word "su;erman", I can easily know the ; is p

so flag is **grepCTF{r4pg0d\_em1n3m\_3256gd62}**

## esoF\*ck 2

DESCRIPTION: 2 levels of eso should make my message impossible to decipher.

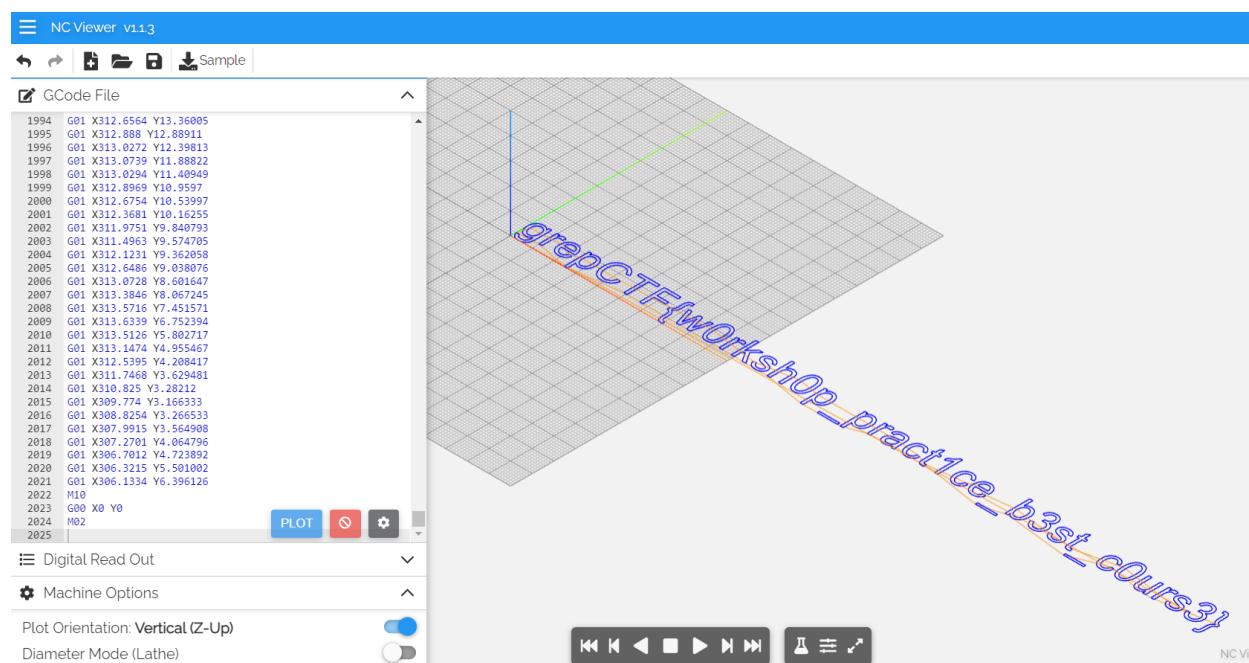
firstly, decode brainfuck then ork decode.<https://tool.bugku.com/brainfuck/>

grepCTF{3sot3r1c\_l4ngu4g3s\_4r3\_0k!}

## Consensual Non Consent

DESCRIPTION: Believe it or not, this is real coding.

Get a file named wp.txt, We can find it's a G-Code, so run it by a simulator.



## Reverse

## Simple rev

Download the attachment and open it by IDA Pro. You will find the Flag string directly.

```
endbr64
push    rbp
mov     rbp, rsp
sub    rsp, 70h
mov     rax, fs:28h
mov     [rbp+var_8], rax
xor    eax, eax
lea     rax, format      ; "Enter flag: "
mov     rdi, rax          ; format
mov     eax, 0
call    _printf
lea     rax, [rbp+$1]
mov     rsi, rax
lea     rax, a$           ; "%S"
mov     rdi, rax
mov     eax, 0
call    __isoc99_scanf
lea     rax, [rbp+$1]
lea     rdx, s2            ; "grepCTF{4p0g33_h1vem1nd_g3n3s1s}"
mov     rsi, rdx          ; s1
mov     rdi, rax          ; s1
call    _strcmp
test   eax, eax
jnz    short loc_123E
```

## Worst encoding

jadx reverse the .jar file. And read the java code, find its algorithm. so we need to do a prime factoring.

```
try:
    n=int(input("please input integer: "))
    list1=[]
    if n !=1 and type(n)==int and n>1:

        while True:
            for i in range(2,n+1):
                b=n%i
                if b ==0:
                    list1.append(i)
                    n=n//i
                    break
                else:
                    break

        print(list1)
        dic = {}
        for l in list1:
            try:
                dic[str(l)] += 1
            except KeyError:
                dic[str(l)] = 1
        print(dic)
        flag = ''
        for i in dic.keys():
```

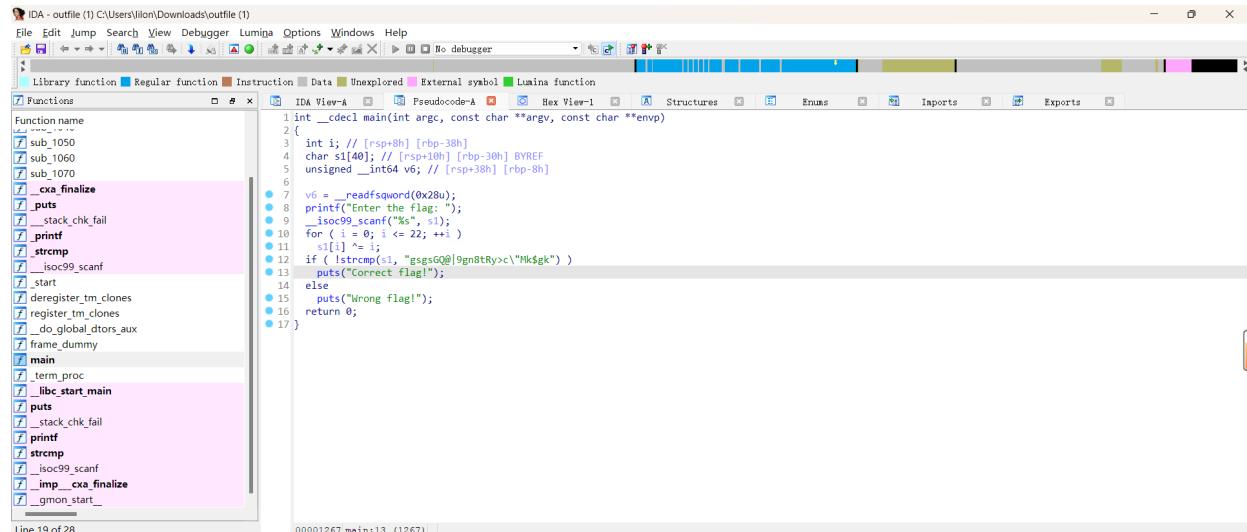
```
    flag += chr(dic[i])
    print(flag)
elif n==1:
    print("1 can not be process")
else: print("please input a right integer")

except ValueError:
    print("please input a right integer")
```

GREP{who\_would\_encode\_like\_this?\_c1caad3482259933bdf988ade3c073e6}

EXORcist

IDA analyze the outfile.



It's a easy XOR, so write a decrypt script.

```
c = "gsgsgQ@|9gn8tRy>c\"Mk$gk"
flag = ''
for i in range(23):
    flag += chr(ord(c[i]) ^ i)
print(flag)
```

```
1 c = "gsgsGQ@|9gn8tRy>c\"Mk$gk"
2 flag = ''
3 for i in range(23):
4     flag += chr(ord(c[i]) ^ i)
5 print(flag)
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    COMMENTS

```
lilon python -u "c:\Users\lilon\Desktop\reverse3.py"
grepCTF{1nd3x w1s3 x0r}
lilon
```

## Forensics

### Monke

DESCRIPTION: I was playing guitar and then this monkey came and broke all my strings 🙄

You can use 010 Editor to analyze the picture and find the interesting string in the end of the hexadecimal data.

5F 76 78 63 F7 16 57 DB B5 C5 37 F6 AD F0 70 87 vxxc-W0jApA7o-8}#  
7B 23 76 9E 1A CD 78 1D BC 86 C7 5A D2 3F C7 E9 {Avx-1. VtICZOP{c  
75 CB 57 EB 5C 6F 96 39 51 D4 FA 90 FA 94 EA uEWaLpi-9Q00-8}#  
22 84 1B 93 FB 1B A4 28 45 25 21 C1 38 AA ".,".0."<EO!8:Kc  
B1 1F 32 3E 22 8D 92 62 63 BA 10 D2 48 F1 1A 4,>,".bc4j.OHr  
F2 CA FC F5 F0 D9 74 AF 45 2A 42 CF 7F FF 00 6F 1F <6d>8utEB'cyo.  
2C FA BF A1 B4 68 48 A5 60 89 03 Ff 27 AC ED 06 \_u1'1HHmMm? -&i  
41 11 3D 46 7A 92 F3 2E DA EO FD 82 24 48 79 D7 N=Fz=6.Uy\$Hyx  
BA SE C9 E1 BF BE AF E2 17 B2 CE BF FE D7 2E 0D F0 7C 3C 00 00 00  
77 C2 9F 04 87 CB FF D9 53 34 6A 63 45 45 55 aya-EYvZ3jlC1L  
52 6E 74 79 4D 64 32 5A 74 59 33 52 66 41 58 46 74 RntryMo2zYrFaHvt  
4E 47 34 78 64 48 6C 66 5A 69 42 66 59 56 6A 74 Ng4xdh1Fz2bfYjRj  
61 31 39 30 4D 46 39 74 4D 47 35 42 D0 33 30 4B a19MF0t9G5rM30K  
52580h:

You can easily decode it by base64 format in [Base64 Decode and Encode - Online](#)

## Decode from Base64 format

Simply enter your data then push the decode button.

```
Z3JlcENURntyM2ozY3RfaHVtNG4xdHlfZzBfYjRja190MF9tMG5rM30K
```

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

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.

```
grepCTF(r3j3ct_hum4n1ty_g0_b4ck_t0_m0nk3)
```

# Doctored image

**DESCRIPTION:** Help, my images are all corrupted!!

You can find this picture lacking a file head of ips in 010 editor, so fixed it.

0000h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .H.....  
0010h: 00 48 00 00 06 00 12 01 03 00 01 00 00 01 00 00 H.....  
0020h: 00 00 1A 01 05 00 01 00 00 00 56 00 00 00 1B 01 ..V.....  
0030h: 05 00 01 00 00 00 5E 00 00 00 28 01 03 00 01 00 ..^.....  
0040h: 00 00 02 00 00 00 13 02 03 00 01 00 00 01 00 00 ..-.....  
0050h: 00 00 69 87 04 00 01 00 00 00 66 00 00 00 00 00 ..!#.....  
0060h: 00 00 60 00 00 00 01 00 00 00 60 00 00 00 01 00 ..f.....  
0070h: 00 00 06 00 00 90 07 00 04 00 00 00 30 32 31 30 .....0210  
0080h: 01 91 07 00 04 00 00 01 02 03 00 00 A0 07 00 .....  
0090h: 04 00 00 00 31 30 30 01 A0 03 00 01 00 00 00 ..0100.....  
00A0h: FF FF 00 00 02 A0 04 00 01 00 00 00 6C 00 00 00 yy.....  
00B0h: 03 A0 04 00 01 00 00 F2 08 00 00 00 00 00 00 00 ..0.....  
00C0h: FF E1 0D 66 68 74 74 70 3A 2F 2F 6E 73 2E 61 64 ýá.fhttp://ns.ad  
00D0h: 6F 62 65 2E 63 6F 6D 2F 78 61 70 2F 31 2E 30 2F obe.com/xap/1.0/  
00E0h: 00 3C 3F 78 70 61 63 6B 65 74 20 62 65 67 69 6E .<xpacket begin  
00F0h: D3 27 EF BB BF 27 20 69 64 3D 27 57 35 40 30 4D ="i>' id='WSMOM  
0100h: 70 43 65 68 69 48 7A 72 65 53 7A 4E 54 63 7A 6B pCehiHzreszNtczK  
0110h: 63 39 64 27 3F 3E 0A 3C 78 3A 78 6D 70 60 65 74 c9d?> <>:xmpmet  
0120h: 61 20 78 6D 6C 6E 73 3A 78 3D 27 61 64 6F 62 65 a xmlns:=adobe  
0130h: 3A 6E 73 3A 6D 65 74 61 2F 27 3E 0A 3C 72 64 66 :ns:meta'>.<rdf  
0140h: 3A 52 44 46 20 78 6D 6C 6E 73 3A 72 64 66 3D 27 ;RDF xmlns:rdf='  
0150h: 68 74 70 70 3A 2F 77 77 77 2E 77 33 2E 6F 72 http://www.w3.or  
0160h: 67 2F 31 39 39 39 2F 30 32 2F 32 32 2D 72 64 66 g/1999/02/22-rdf  
0170h: 2D 73 79 6E 74 61 78 2D 6E 73 23 27 3E 0A 04 20 -syntax-ns'>.  
0180h: 3C 72 64 66 3A 44 65 73 63 72 69 70 74 69 6F 66 <rdf:Description  
0190h: 20 72 64 66 3A 61 62 6F 75 74 3D 27 27 0A 20 20 rdf:about=''.  
01A0h: 78 60 6C 6E 73 3A 41 74 74 72 69 62 3D 27 68 74 xmlns:Attrib='ht  
01B0h: 74 70 3A 2F 6E 73 2E 61 74 74 72 69 62 75 74 tp://ns.attribut  
01C0h: 69 60 6E 2E 63 6F 6D 2F 61 64 73 2F 31 2E 30 2F ion.com/ads/1.0/  
01D0h: 27 3E 0A 20 20 3C 41 74 74 72 69 62 3A 41 64 73 '>. <Attrib:Ads  
01E0h: 3E 04 20 20 20 3C 72 64 66 3A 53 65 71 3E 0A 20 >. <rdf:Seq>  
01F0h: 20 20 20 3C 72 64 66 3A 6C 69 20 72 64 66 3A 70 <rdf:li rdf:p  
0200h: 61 72 73 65 54 79 70 65 3D 27 52 65 73 6F 75 72 arseType='Resour  
0210h: 63 65 27 3E 0A 20 20 20 20 3C 41 74 74 72 69 ce'>. <Attrib  
0220h: 62 3A 43 72 65 61 74 65 64 3E 32 30 32 33 2D 30 b:Created>2023-0  
0230h: 32 20 30 33 3C 2F 41 74 74 72 69 62 3A 43 72 65 2-03:<Attrib:Cre  
0240h: 61 74 65 64 3E 0A 20 20 20 20 3C 41 74 74 72 ated>. <Attrib  
0250h: 69 62 3A 45 78 74 49 64 3E 63 65 65 39 65 63 ib:ExtId=ce9ecc  
0260h: 64 20 63 31 31 61 20 34 38 36 33 2D 62 33 38 64 d-c11a-4863-b38d  
0270h: 2D 64 35 66 34 66 62 32 32 62 38 65 38 3C 2F 41 -d5f4fb22b8e8</A  
0280h: 74 74 72 69 62 3A 45 78 74 49 64 3E 0A 20 20 20 ttrib:ExtId>  
0290h: 20 20 3C 41 74 74 72 69 62 3A 46 62 49 64 3E 35 <Attrib:Fbid>5  
02A0h: 32 35 32 36 35 39 31 34 31 37 39 35 38 30 3C 2F 25265914179580<

0000h: FF D8 FF EO 00 10 4A 46 49 46 00 01 01 01 00 00 48 ýóýá..JTF...H.....  
0010h: 00 48 00 00 06 00 12 01 03 00 01 00 00 01 00 00 H.....  
0020h: 00 00 1A 01 05 00 01 00 00 00 56 00 00 00 1B 01 ..V.....  
0030h: 05 00 01 00 00 00 5E 00 00 00 28 01 03 00 01 00 ..^.....  
0040h: 00 02 00 00 00 13 02 03 00 01 00 00 00 01 00 00 ..-.....  
0050h: 00 00 69 87 04 00 01 00 00 00 66 00 00 00 00 00 ..!#.....  
0060h: 00 00 60 00 00 00 01 00 00 00 60 00 00 00 01 00 ..f.....  
0070h: 00 00 06 00 00 90 07 00 04 00 00 00 30 32 31 30 .....0210  
0080h: 01 91 07 00 04 00 00 01 02 03 00 00 A0 07 00 .....  
0090h: 04 00 00 00 30 31 30 01 A0 03 00 01 00 00 00 ..0100.....  
00A0h: FF FF 00 00 02 A0 04 00 01 00 00 00 6C 00 00 00 yy.....  
00B0h: 03 A0 04 00 01 00 00 F2 08 00 00 00 00 00 00 00 ..0.....  
00C0h: FF E1 0D 66 68 74 74 70 3A 2F 2F 6E 73 2E 61 64 ýá.fhttp://ns.ad  
00D0h: 6F 62 65 2E 63 6F 6D 2F 78 61 70 2F 31 2E 30 2F obe.com/xap/1.0/  
00E0h: 00 3C 3F 78 70 61 63 6B 65 74 20 62 65 67 69 6E .<xpacket begin  
00F0h: D3 27 EF BB BF 27 20 69 64 3D 27 57 35 40 30 4D ="i>' id='WSMOM  
0100h: 70 43 65 68 69 48 7A 72 65 53 7A 4E 54 63 7A 6B pCehiHzreszNtczK  
0110h: 63 39 64 27 3F 3E 0A 3C 78 3A 78 6D 70 60 65 74 c9d?> <>:xmpmet  
0120h: 61 20 78 6D 6C 6E 73 3A 78 3D 27 61 64 6F 62 65 a xmlns:=adobe  
0130h: 3A 6E 73 3A 6D 65 74 61 2F 27 3E 0A 3C 72 64 66 :ns:meta'>.<rdf  
0140h: 3A 52 44 46 20 78 6D 6C 6E 73 3A 72 64 66 3D 27 ;RDF xmlns:rdf='  
0150h: 68 74 70 70 3A 2F 77 77 77 2E 77 33 2E 6F 72 http://www.w3.or  
0160h: 67 2F 31 39 39 39 2F 30 32 2F 32 32 2D 72 64 66 g/1999/02/22-rdf  
0170h: 2D 73 79 6E 74 61 78 2D 6E 73 23 27 3E 0A 04 20 -syntax-ns'>.  
0180h: 3C 72 64 66 3A 44 65 73 63 72 69 70 74 69 6F 66 <rdf:Description  
0190h: 20 72 64 66 3A 61 62 6F 75 74 3D 27 27 0A 20 20 rdf:about=''.  
01A0h: 78 60 6C 6E 73 3A 41 74 74 72 69 62 3D 27 68 74 xmlns:Attrib='ht  
01B0h: 74 70 3A 2F 6E 73 2E 61 74 74 72 69 62 75 74 tp://ns.attribut  
01C0h: 69 60 6E 2E 63 6F 6D 2F 61 64 73 2F 31 2E 30 2F ion.com/ads/1.0/  
01D0h: 27 3E 0A 20 20 3C 41 74 74 72 69 62 3A 41 64 73 '>. <Attrib:Ads  
01E0h: 3E 04 20 20 20 3C 72 64 66 3A 53 65 71 3E 0A 20 >. <rdf:Seq>  
01F0h: 20 20 20 3C 72 64 66 3A 6C 69 20 72 64 66 3A 70 <rdf:li rdf:p  
0200h: 61 72 73 65 54 79 70 65 3D 27 52 65 73 6F 75 72 arseType='Resour  
0210h: 63 65 27 3E 0A 20 20 20 20 3C 41 74 74 72 69 ce'>. <Attrib  
0220h: 62 3A 43 72 65 61 74 65 64 3E 32 30 32 33 2D 30 b:Created>2023-0  
0230h: 32 20 30 33 3C 2F 41 74 74 72 69 62 3A 43 72 65 2-03:<Attrib:Cre  
0240h: 61 74 65 64 3E 0A 20 20 20 20 3C 41 74 74 72 ated>. <Attrib  
0250h: 69 62 3A 45 78 74 49 64 3E 63 65 65 39 65 63 ib:ExtId=ce9ecc  
0260h: 64 20 63 31 31 61 20 34 38 36 33 2D 62 33 38 64 d-c11a-4863-b38d  
0270h: 2D 64 35 66 34 66 62 32 32 62 38 65 38 3C 2F 41 -d5f4fb22b8e8</A  
0280h: 74 74 72 69 62 3A 45 78 74 49 64 3E 0A 20 20 20 ttrib:ExtId>  
0290h: 20 20 3C 41 74 74 72 69 62 3A 46 62 49 64 3E 35 <Attrib:Fbid>5  
02A0h: 32 35 32 36 35 39 31 34 31 37 39 35 38 30 3C 2F 25265914179580<

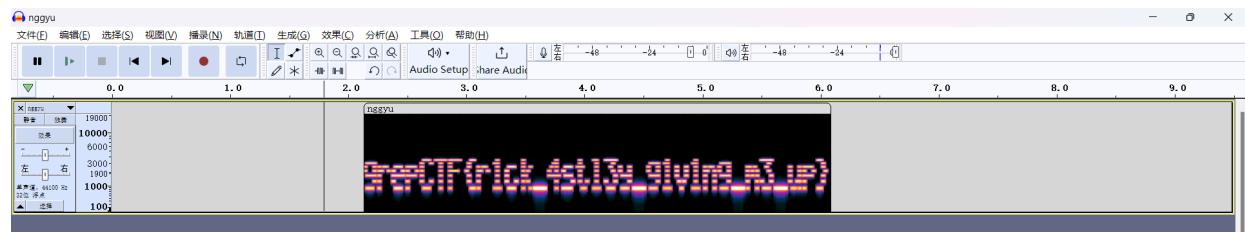
# grepCTF{m00n\_kn1ght}



## NGGYU

DESCRIPTION: .

Open the audio file by audacity and change to see the spectrogram.



grepCTF{r1ck\_4stl3y\_g1v1ng\_m3\_up}

## R36

use RX-SSTV to process the .wav file in Mode robot36.



grepCTF{psych3d3l1c\_fr0g}

## IronMan

The attachment is a picture of IronMan. Nice! Analyze it by zsteg.

```
zsteg ironman.png -a | grep CTF
```

```
(kali㉿kali)-[~/Desktop] channels X
$ zsteg ironman.png -a | grep CTF
b1,rgb,lsb,yx      .. ,text: "grepCTF{i_d0n't_f3el_s0_g00d}\n"
```

## Royal Steg

DESCRIPTION: Then Jesus turned, and seeing them following, said to them, 'what do you SEEK?

- JOHN 1:38

do you see the SEEK, so try stegseek.

```
kali@kali: ~/Desktop
File Actions Edit View Help
└$ stegseek --seed steg.jpg
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Found (possible) seed: "33c3c6de"
    Plain size: 176.0 Byte(s) (compressed)
    Encryption Algorithm: rijndael-128
    Encryption Mode:      cbc

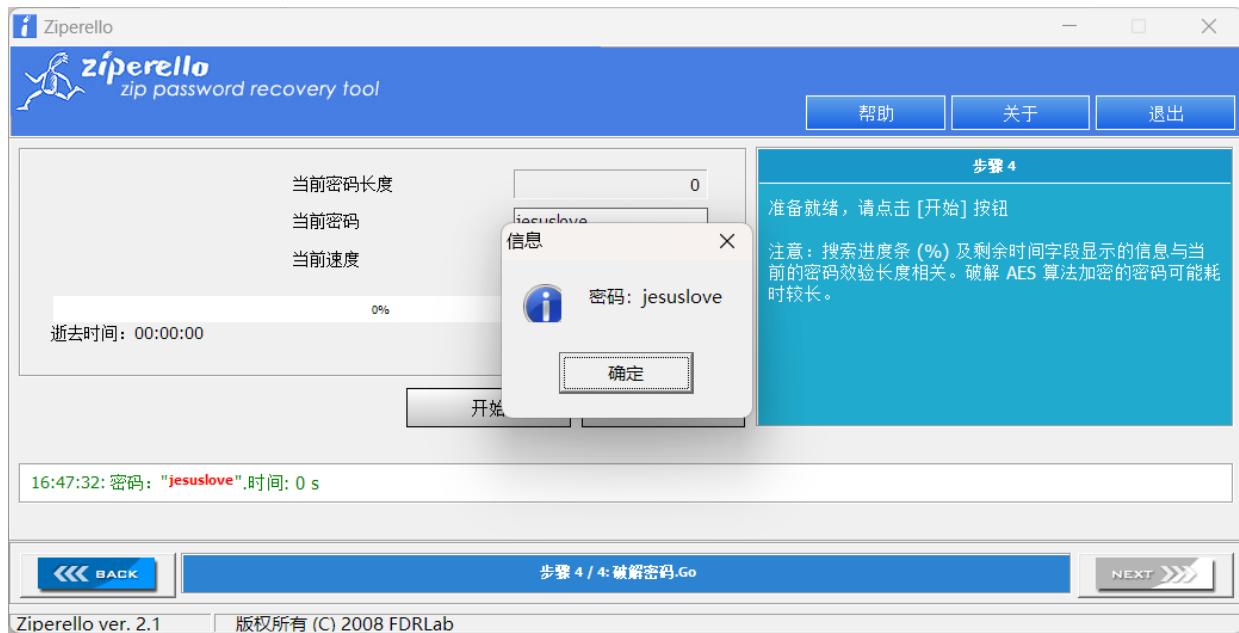
└(kali㉿kali)-[~/Desktop]
└$ stegseek steg.jpg rockyou.txt
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

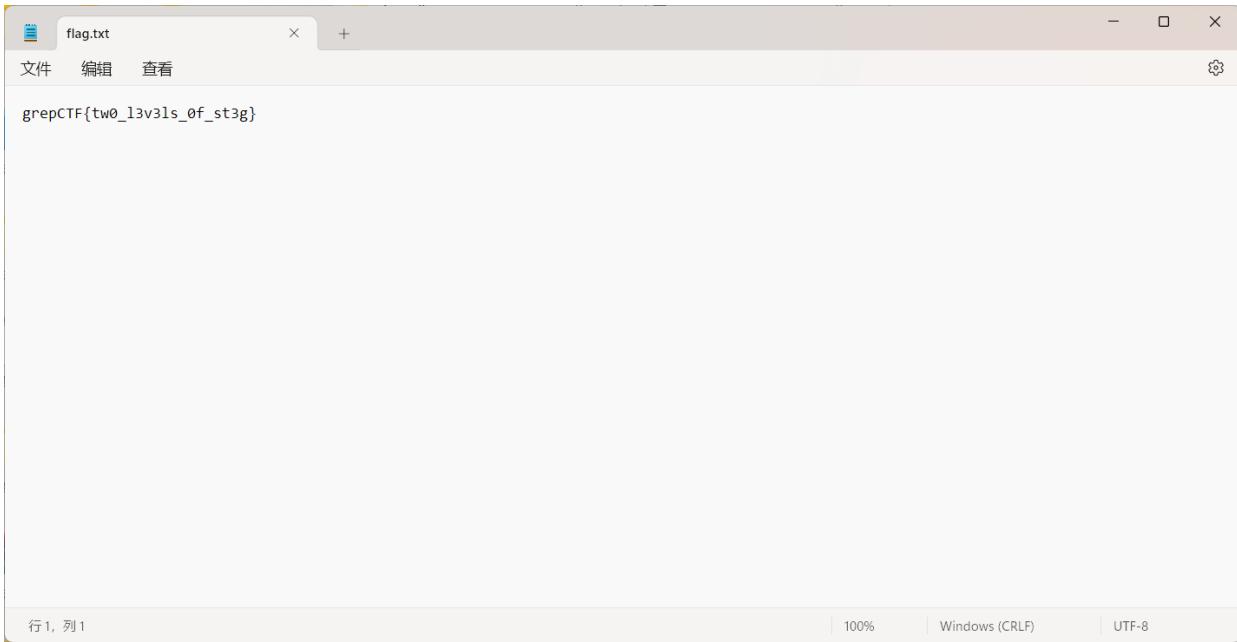
[i] Found passphrase: "cuteessort37"
[i] Original filename: "orig.zip".
[i] Extracting to "steg.jpg.out".

└(kali㉿kali)-[~/Desktop]
└$
```

get a orig.zip and the flag.txt is in it. However, the zip looks encrypted.

use a zip burster to burst the password of the zip file with the rockyou.txt. the password is jesuslove.





flag.txt

文件 编辑 查看

grepCTF{tw0\_l3v3ls\_0f\_st3g}

行 1, 列 1 | 100% | Windows (CRLF) | UTF-8

get flag : grepCTF{tw0\_l3v3ls\_0f\_st3g}

## Cryptography

### CaeXOR

DESCRIPTION: I pressed shift key 10 times and lost the flag. Can you find my flag.

```
#enc.py
from random import *
flag="REDACTED"
a=randint(1,1000)
c=[]
for f in flag:
    c.append(str(ord(f)^a))
print(c)
print(a)

#c=['162', '177', '188', '169', '136', '187', '138', '145', '172', '187', '138',
'145', '172', '190', '152', '156', '187', '195', '177', '142']
#a=REDACTED
```

solution script:

```

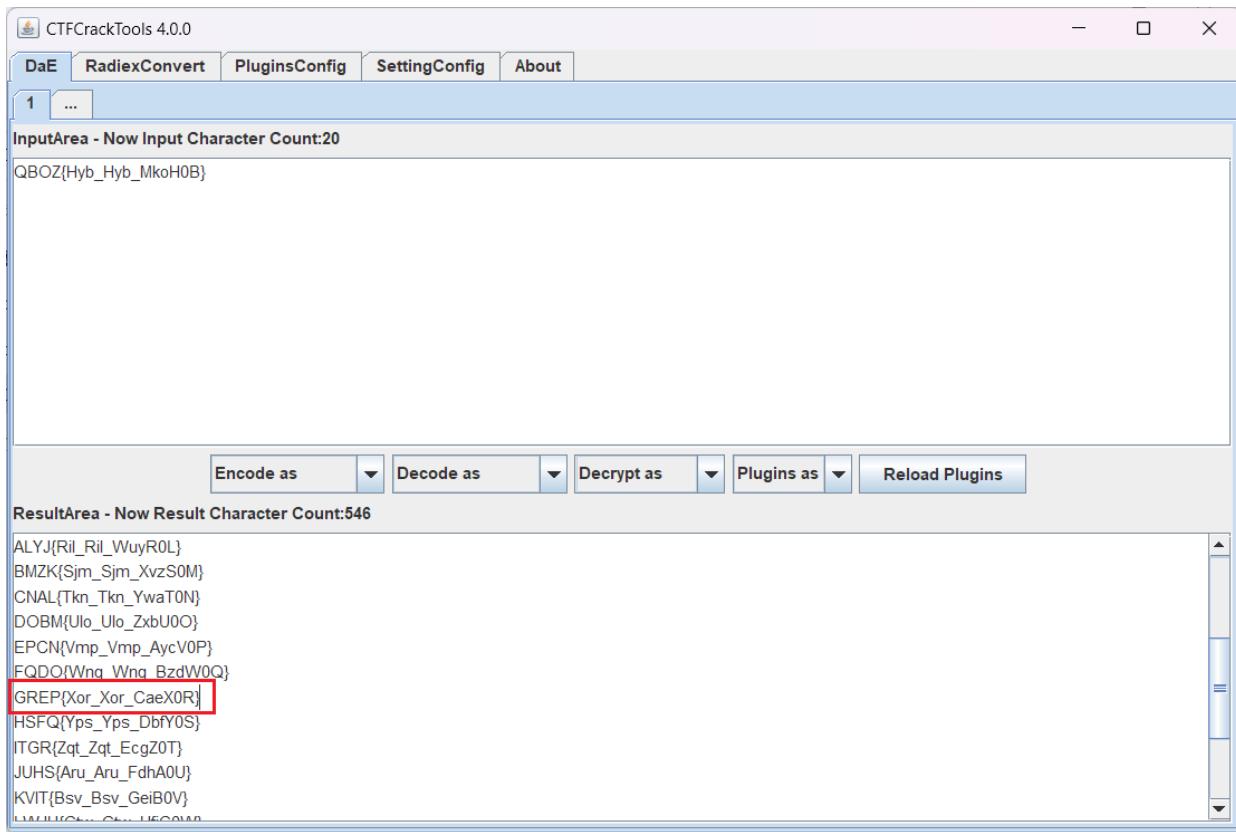
f = open('test.txt', 'w', encoding='utf-8')
c=['162', '177', '188', '169', '136', '187', '138', '145', '172', '187', '138',
'145', '172', '190', '152', '156', '187', '195', '177', '142']
for i in range(1,1001):
    flag = ''
    for j in c:
        flag += chr(int(j)^i)
    print(flag)
    f.write(flag+"\n")
f.close()

```



@S^KjYhsNYhsN\z~Y!S1  
AR\_JkXirOXirO]{X Rm  
FUXMl\_nuH\_nuHZ|x\_`Uj  
GTYLm^otI^otI[ }y^&Tk  
DWZOn]lwJ]lwJX~z ]%wh  
EV[No\mvK\mvKY{ \\$Vi  
JYTA`SbyDSbyDVptS+Yf  
KXU@aRcxERcxElquR\*Xg  
H[VcbQ` {FQ` {FTrvQ)[d  
IZWBcPazGPazGUswP(Ze  
N]PEdwf}@wf}@Rtpw/]b  
O\QDeVg|AVg|ASuqV.\c  
L\_RGfUdBUDBPvrU-`  
M^SFgTe~CTe~CQwsT,^a  
RALYxKza\Kza\NhIK3A~  
S@MXyJ{` ]J{` ]OimJ2@  
PCN[zIx^Ix^LjnI1C|  
QBOZ{Hyb\_Hyb\_MkoH0B}

I find a string result "QBOZ{Hyb\_Hyb\_MkoH0B}" is like a flag format, but it is not GREP, so do a caesarcode decryption. Get the flag.



Blind

PHOTOGRAPH BY ROBERT M. STONE

exchange it to English in 盲文翻译器: 转换器和解码器 - SYMBL (● ●)

This website can't translate .:::

`..:: is 0, so flag is grepCTF{t00_b1nd_t0_s33}`

NOT 13

rot13 decrypted the string.

DaE RadixConvert PluginsConfig SettingConfig About

1 ...

**InputArea - Now Input Character Count:403**

PULGY MQDIY HUPUL DMX WYGX, JUBDGJGXIL WHMQMDJMBA GPMX. YULKM DJGPGLMDSIG, BIPPW TMXWG PIJXID XMBJMHBX, YM XILQMD TGDXMKIPIY X

◀ ▶

Encode as Decode as Decrypt as Plugins as Reload Plugins

**ResultArea - Now Result Character Count:403**

CHYTL ZDQVL UHCHY QZK JLTK, WHOQTWKTVY JUZDZQWZON TCZK. LHYXZ QWTCTYZQFVT, OVCCJ GZKJT CVWKVQ KZOWZUVOK, LZ KVYDZQ GTQKZXCVL KTCCVQ, VR WHONVT KVYDZQ FVJL FVZQ JVWJT, DTHQD VCKTZWZTG CVWVQ T2QNL, THTC GJFVQ T2QNL ZGTHVLA GTL, KQWZU ZU KZOWZUVOK ZDQVL, KAT BCJN ZQ ZKQ OHK JCBJRU YHK, ZD CHBTY WJNT, BZKA VOUTYWHYTC ZQKRTJU HE QDJPTQ, BVQWT UHWRV OVCCJ TTJK, KZOWZUVOK KTLDVQ CTWAVQ VCKTZWZTG CTC.

◀ ▶

then, analyze it in quipqiup.

**quipqiup** beta3

quipqiup is a fast and automated cryptogram solver by [Edwin Olson](#). It can solve simple substitution ciphers often found in newspapers, including puzzles like cryptoquips (in which word boundaries are preserved) and patristocrats (nwih chwör dboun darie saren t).

**Puzzle:**

CHYTL ZDQVL UHCHY QZK JLTK, WHOQTWKTVY JUZDZQWZON TCZK. LHYXZ QWTCTYZQFVT, OVCCJ GZKJT CVWKVQ KZOWZUVOK, LZ KVYDZQ GTQKZXCVL KTCCVQ, VR WHONVT KVYDZQ FVJL FVZQ JVWJT, DTHQD VCKTZWZTG CVWVQ T2QNL, THTC GJFVQ T2QNL ZGTHVLA GTL, KQWZU ZU KZOWZUVOK ZDQVL, KAT BCJN ZQ ZKQ OHK JCBJRU YHK, ZD CHBTY WJNT, BZKA VOUTYWHYTC ZQKRTJU HE QDJPTQ, BVQWT UHWRV OVCCJ TTJK, KZOWZUVOK KTLDVQ CTWAVQ VCKTZWZTG CTC.

**CLUES:** For example G=R QVW=THE

Solve

automatically selected statistics mode; you can override by using the drop down menu next to the solve button.

0 -3.026 LOREM IPSUM DOLOR SIT AMET, CONSECTETUR ADIPISCING ALIT, MORBI SCALARISVA, NELLU BITUA LECTES TINCIDENT, MI TERRIS BASTIJBLEM TALLES, ET CONGE TERRIS NEUM WEIS URGEA, PRIN ULTRICIAS LECTES RISES, OGOT VARIES RISES INTORGEM SAO, NENC ID TINCIDENT IPSEM, THA FLAG IS ITS NOT ULZUKS ROT, IN LOGAR CUSA, ZITH ENDARSORAS INSTAO OF SPACUS. PESCA DICTEM NELLU ABUT, TINCIDENT TAMPS LACTES ULTRICIAS BAL.

1 -3.729 LORAN IPSEN DOLOR SIT UMAT, CONGACTATER UDIPISCING ALIT, MORJI SCALARISVA, NELLU BITUA LECTES TINCIDENT, MI TERRIS BASTIJBLEM TALLES, ET CONGE TERRIS NEUM WEIS URGEA, PRIN ULTRICIAS LECTES RISES, AGAT BURIES RISES INTORGEM SAO, NENC ID TINCIDENT IPSEM, THA FLAG IS ITS NOT ULZUKS ROT, IN LOGAR CUSA, ZITH ENDARSORAS INSTAO OF SPACUS. PESCA DICTEM NELLU ABUT, TINCIDENT TAMPS LACTES ULTRICIAS BAL.

2 -3.797 LURAY IPSEY DULUR SIT ATY, CUNSECTETER ADIPISCING OLY, YURKI SCOLORISRED, NELLA VITAO LECTES TINCIDENT, TI TERRIS VOSTIKELEY TOLLES, ET CUNGE TERRIS REAY BEIS AEGEO, PRIN ULTRICIAS LECTES RISES, OGOT VARIES RISES INTORGEM SAO, NENC ID TINCIDENT IPSEM, THA FLAG IS ITS NOT ULZUKS ROT, IN LUMOR CASO, WITH ENDORSCUROS INSTAO OF SPACUS. PESCA DICTEM NELLA ORAT, TINCIDENT TYPES LECTES ULTRICIAS VOL.

3 -3.836 ROLEP ICSP MORG SIT UPET, KONSEKTETAL UNICICKING BRIT, POLXI SKERELLSOAE, NARRU JITUU RATAS TINKIMANT, PI TALCIS JESTIXAP TERRAS, AT KONRAB TALCIS DAIF DAIS UAGAE, CLOIN ARTLIKES RAKTAS LISAS, EGRT JULIAS LISAS INTELMAP SEM, NANK IM TINKIMANT ICSP. THE FRIG IS ITS NOT URWUNS LOT, IN ROWEL KUSE, WITH ANWELSKOLES INSTEUE OF SCURES. PASEE MIKTAF NARRU BLUT, TINKIMANT TERCAS RETKAS ARTLIKIES JER.

4 -3.843 LORIM ABSUM HOLOR SAT EMIT, KONSEKTITUR BHABSKAND ILAT, MORCA SKILIRASQU, NULLE VATEI LURTUS TANKAHUNT, MA TURBAS VISTACULUM TILLUS, UT KONDUI TURBAS QUEN QUAS RUDUI, BROAN ULTRAKAIS LURTUS RASUS, IDIT VERAUUS RASUS ANTIRHM SH, NUNK AH TANKAHUNT ABSUM. TZI FLED AS ATS NOT BIJEWS ROT, AN LOJIR KESI, JATZ UNHRSKORIS ANSTIEH OF SBEXIS. PUSKI HAKTOM NOLLE IRET, TANKAHUNT TIMBUS LIUTUS ULTRAKAIS VII.

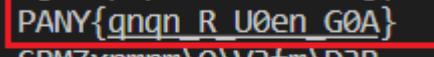
## CaeXOR 2

DESCRIPTION: Ooops, i forgot the shift this time. Can you still figure out my flag.

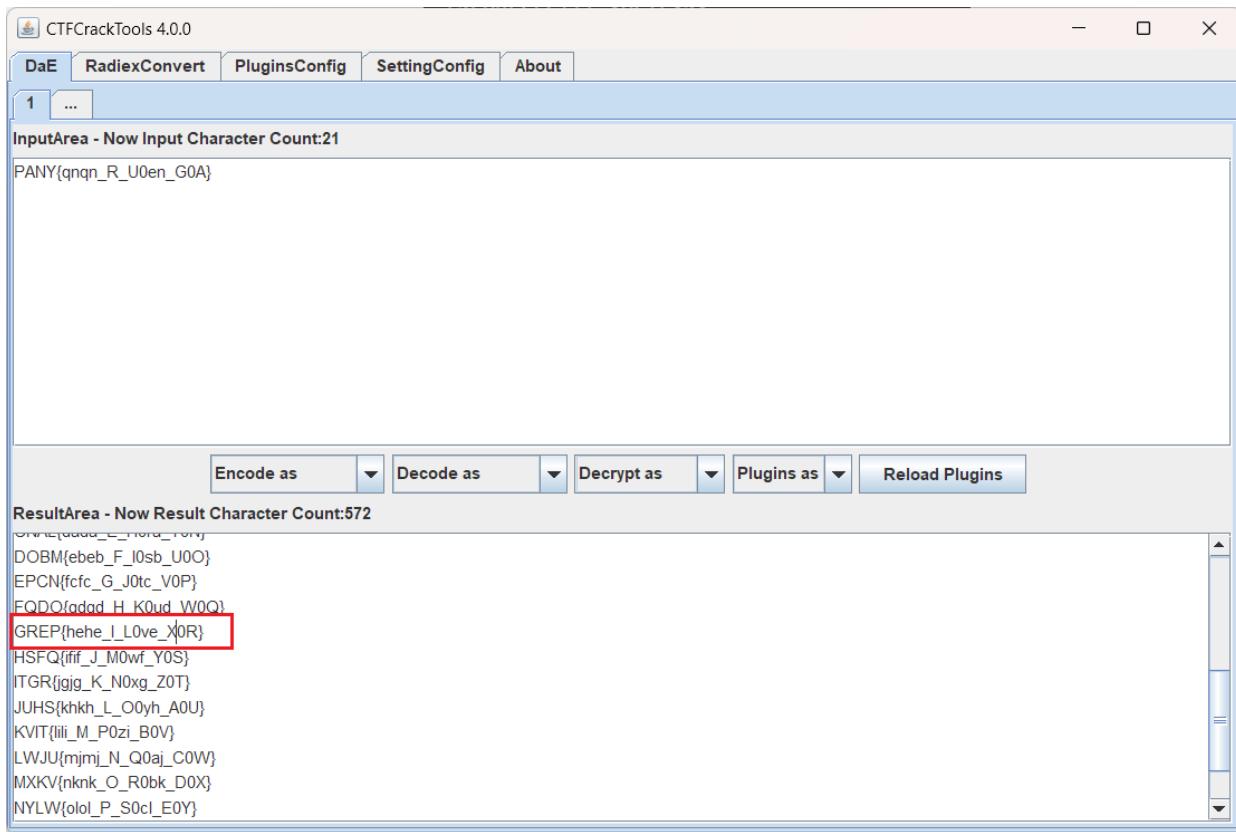
```
#enc.py
from random import *
flag="REDACTED"
a=randint(1,1000)
c=[]
for f in flag:
    c.append(str(ord(f)^a))
print(c)
print(a)

#c=['313', '296', '295', '304', '274', '280', '263', '280', '263', '310', '315',
'310', '316', '345', '268', '263', '310', '302', '345', '296', '276']
#a=REDACTED
```

```
f = open('test.txt', 'w', encoding='utf-8')
c=['313', '296', '295', '304', '274', '280', '263', '280', '263', '310', '315',
'310', '316', '345', '268', '263', '310', '302', '345', '296', '276']
for i in range(1,1001):
    flag = ''
    for j in c:
        flag += chr(int(j)^i)
    print(flag)
    f.write(flag+"\n")
f.close()
```



```
fwxoMGXGXidicSXiqwK
YHGPrxgxgV[V\9lgVN9Ht
XIFQsyfyfWZW]8mfW08Iu
[JERpzezeTYT^;neTL;Jv
ZKDSq{d{dUXU_:odUM:Kw
]LCTV|c|cR_RX=hcRJ=Lp
\MBUw}b}bS^SY<ibSK<Mq
_NAVt~a~aP]PZ?jaPH?Nr
^O@Wu` `Q\Q[>k` QI>Os
Q@OXzpopo^S^T1do^E1@|
PANY{gnqn_R_U0en_G0A}
```



like CaeXOR, just decrypt by a script and caesar decrypt.

# DOGE DOGE DOGE

challenge's code:

```
from Crypto.Util.number import *
from pwn import xor
flag = b'REDACTED'
key = b'REDACTED'
enc = b''
for i in range(len(flag)):
    enc += xor(key[i], flag[i])
print(enc)
# enc = b'#=5\x07\x1b\x01>4#s<u! \x1a3~3-\x1b7w7\x1b&4\x1a":)8'
```

According to the title, I guess the key is many DOGE.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS

```
lilon  python -u "c:\Users\lilon\Downloads\doge.py"
b'grepCTF{pl4y1ng_w1th_x0r_is_fun}'
```

# Bird Seed

```
#encrypt.py
import random
flag = open('flag.txt').read()

rand_seed = random.randint(0, 999)
random.seed(rand_seed)
encrypted = ''

for chr in flag:
    encrypted += f'{(ord(chr) ^ random.randint(0, 255)):02x}'

with open('out.txt', 'w') as f:
    f.write(encrypted)

#output
encrypted = a282b415279f5aa08cd4649515268910b8968a1eabda7c1bb2898c
```

1byte = 2 hex number, so change the encrypted result to one decimal number per two hex number.

```

import random

Deci_Encrypted = [162, 130, 180, 21, 39, 159, 90, 160, 140, 212, 100, 149, 21, 38,
137, 16, 184, 150, 138, 30, 171, 218, 124, 27, 178, 137, 140]

for rand_seed in range(0, 999):
    random.seed(rand_seed)
    flag = ''
    for char in Deci_Encrypted:
        flag += chr(char ^ random.randint(0, 255))
    if "grep" in flag:
        print(flag)

```

A terminal window showing the command `python -u "c:\Users\lilon\Downloads\encrypt.py"` being run. The output is `grepCTF{nbv3r_truly_r4nd0m}`. The terminal has a yellow background with orange and blue arrows.

## Uneasy Alliance

DESCRIPTION: You have seen people giving out p, q values and asking you to decrypt the cipher text. This time, you only have the cipher text. Good luck decrypting !

```

from Crypto.Util.number import *
import math
import time
from random import Random

seed = math.floor(time.time())
rnd = Random(seed)

rand_fn = lambda n: long_to_bytes(rnd.getrandbits(n))
p = getPrime(128, randfunc=rand_fn)
q = getPrime(128, randfunc=rand_fn)
e = 65537
n = p * q

assert p != q

m = bytes_to_long(b"GREP{REDACTED}")
ct = pow(m, e, n)
print("Cipher text:", ct)
# Cipher text:
9898717456951148133749957106576029659879736707349710770560950848503614119828
# Seed: REDACTED

```

because the seed is the timestamp, so burst way to try.

```

from Crypto.Util.number import *
import math

```

```

import time
from random import Random

for seed in range(0,999999):
    rnd = Random(seed)

    rand_fn = lambda n: long_to_bytes(rnd.getrandbits(n))
    p = getPrime(128, randfunc=rand_fn)
    q = getPrime(128, randfunc=rand_fn)
    e = 65537

Cipher=9898717456951148133749957106576029659879736707349710770560950848503614119828
n = p * q
phi = (p-1)*(q-1)
d = inverse(e,phi)
plain=str(long_to_bytes(pow(Cipher,d,n)))
if 'GREP' in plain:
    print(plain)

```

b'GREP{Brut3D\_M3!\_f0r\_l1f3}'

## OSINT

### Sherlock Exhausted

DESCRIPTION: Holmes has reached 221B Baker Street after an exhausting day. A murder has happened, but there is no clue of the name of the murderer. Your task is to help Sherlock figure out the first name of the murderer.



I can know it's a logo of john GBA lite via google picture search, so the answer is **GREP{johN}**

### apogtspi

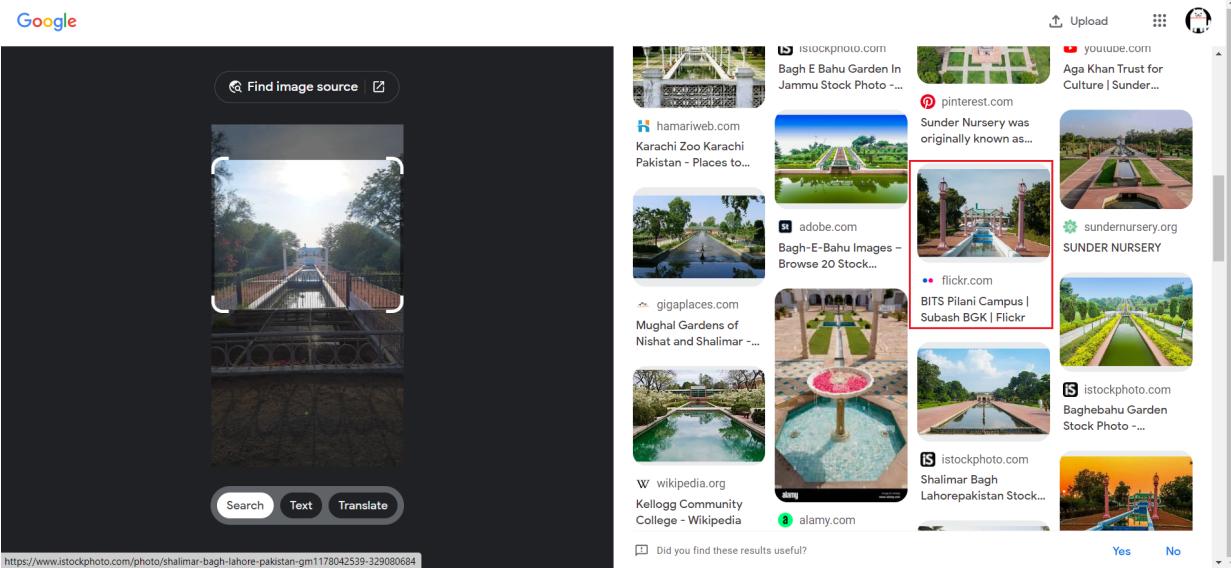
I think some times, and try to find some result via search engine. However I am fail. So I see the hint, Hint: Always remember the organizers. It's key. According to the hint, I focus on the organizers.

BITS PILANI & APOGEE 23'

So I try the flag: GREP{apogeebitspilani}, it's the right flag.

# Don't Deviate

search this photo by google image search. Then find some photo in the internet is similar to the one I obtain.



The screenshot shows a Google search results page for the query "Shalimar Bagh Lahore Pakistan". The top result is a link to a stock photo from iStockphoto.com. Below it are several other links to various websites and sources, including hamariweb.com, adobe.com, gigaplaces.com, wikipedia.org, and alamy.com, each showing different views of Mughal gardens. A red box highlights a specific image from flickr.com showing pink pillars and a central fountain, which is identified as being at BITS Pilani Campus | Subash BGK.



This is a photograph of a garden featuring a long, narrow reflecting pool with a central fountain. On either side of the pool are tall, ornate pink pillars topped with spherical ornaments. In the background, there's a building with a green roof and a large statue. The entire scene is framed by lush green trees.

Subash BGK + 追蹤  
BITS Pilani Campus

132 次檢視 0 人加入最愛 0 則留言

拍攝於 2015年12月28日

info \$ © 保留部分權利

But it's always wrong, (28.3560,75.5883)

I don't know what is the flag, but I think I have closed to it.