

WRITE UP PICOCTF 2021



REZKA NORHAFIZAH

Contents

[GENERAL SKILLS]	4
OBEDIENT CAT	4
PYTHON WRANGLING	4
WAVE A FLAG	4
NICE NETCAT	5
STATIC AIN'T ALWAYS NOISE	6
TAB, TAB, ATTACK	6
MAGIKARP GROUND MISSION	7
LETS WARM UP	8
WARMED UP	8
2WARM	9
WHAT'S A NET CAT?	10
STRINGS IT	10
BASES	11
FIRST GREP	11
BASED	12
PLUMBING	13
[WEB EXPLOITATION]	14
INSP3CT0R	14
GET AHEAD	15
COOKIES	17
PICOBROWSER	19
WHERE ARE THE ROBOTS	20
LOGON	21
DON'T-USE-CLIENT-SIDE	22
LOGIN	23
SCAVENGER HUNT	26
[CRYPTOGRAPHY]	29
MOD 26	29
THE NUMBERS	29
CAESAR	30
13	31

PIXELATED	31
EASY1	33
MIND YOUR Ps & Qs.....	34
[FORENSIC].....	35
INFORMATION	35
MATRYOSHKA DOLL.....	35
WIRESHARK DOO..... DOO... DOOO.....	36
WIRESHARK TWOO... TWOO... TWOOO.....	37
EXTENSIONS	39
LIKE1000.....	40
DISK, DISK, SLEUTH!	41
DISK, DISK, SLEUTH! II	42
WHAT LIES WITHIN	44
MILKSLAP	45
SO META	46
SHARK ON WIRE 1.....	46
WEBNET0	47
WEBNET1	48
CORRUPT	49
WHITEPAGES.....	50
MACROHARD WEAKEDGE.....	51
TUNN3L V1S10N	52

[GENERAL SKILLS]

OBEDETNT CAT

Diberikan file berupa ascii text, saya langsung menggunakan perintah cat untuk melihat isinya :

```
└─(root㉿kali)-[~/home/kali/Downloads/picoCTF/GeneralSkills]
  └─# cat flag
    picoCTF{s4n1ty_v3r1f13d_2fd6ed29}
```

Flag = picoCTF{s4n1ty_v3r1f13d_2fd6ed29}

PYTHON WRANGLING

Diberikan 3 file (2 file berupa ascii text, dan 1 file berupa script python), kemudian saya lihat isi kedua file ascii text tersebut :

```
└─(root㉿kali)-[~/home/kali/Downloads/picoCTF/GeneralSkills]
  └─# cat flag.txt.en
    gAAAAABgUAIWIVsiR0W23DAHlK5DX6Y4BvwES94M_XdDcNAquhp-A0D2z8n812YEXaSD9Whowebh2cm5Wa0cqzuW0Kc7f0ct00Jnp0mVF8A91j0Hx4dKt
    vk3l5ghPT71Y7GxErPRyJUs

  └─(root㉿kali)-[~/home/kali/Downloads/picoCTF/GeneralSkills]
    └─# cat pw.txt
      ac9bd0ffac9bd0ffac9bd0ffac9bd0ff
```

Selanjutnya, setelah melihat isi file saya langsung eksekusi dengan python3 script tersebut :

```
└─(root㉿kali)-[~/home/kali/Downloads]
  └─# python3 ./ende.py
    Usage: ./ende.py (-e/-d) [file]
```

Namun, ternyata tidak berhasil dan memunculkan clue bahwa harus ada file yang di encode dan di decode. Lalu, saya coba kembali eksekusi dengan menyertakan file flag.txt.en sebagai text yang di encode dan ternyata diminta untuk memasukkan password :

```
└─(root㉿kali)-[~/home/kali/Downloads/picoCTF/GeneralSkills]
  └─# python3 ./ende.py -d flag.txt.en
    Please enter the password:ac9bd0ffac9bd0ffac9bd0ff
    picoCTF{4p0110_1n_7h3_h0us3_ac9bd0ff}
```

Kemudian saya masukkan passwordnya dan memunculkan flag.

Flag = picoCTF{4p0110_1n_7h3_h0us3_ac9bd0ff}

WAVE A FLAG

Diberikan file maka saya langsung analisis dan ternyata itu adalah ELF file :

```
└─(root㉿kali)-[~/home/kali/Downloads/picoCTF/GeneralSkills]
  └─# file warm
    warm: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64
    .so.2, for GNU/Linux 3.2.0, BuildID[sha1]=01b148cdedfc38125cac0d87e0537466d47927b1, with debug_info, not stripped
```

Kemudian saya strings file tersebut dan ternyata flag nya langsung muncul :

```
[root@kali]# strings warn  
/lib64/ld-linux-x86-64.so.2  
libc.so.6  
puts  
printf  
_cxa_finalize  
strcmp  
_libc_start_main  
GLIBC_2.2.5  
_ITM_deregisterTMCloneTable  
__gmon_start__  
_ITM_registerTMCloneTable  
=y  
=W  
=Z  
AWAVI  
AUATL  
[ ]AVA]A^A_  
Hello user! Pass me a -h to learn what I can do!  
Oh, help? I actually don't do much, but I do have this flag here: picoCTF{biscuits_4nd_gr4vy_f0668f62}  
I don't know what '%'s means! I do know what -h means though!  
;*3$"
```

Flag = picoCTF{b1scu1ts_4nd_gr4vy_f0668f62}

NICE NETCAT....

Diberikan sebuah netcat listener, lalu langsung saya jalankan dan muncul deretan angka decimal :

```
[kali㉿kali]~$ nc mercury.picoctf.net 21135  
112  
105  
99  
111  
67  
84  
70  
123  
103  
48  
48  
100  
95  
107  
49  
116  
116  
121  
33  
95  
110  
49  
99  
51  
95  
107  
49  
116  
116  
121  
33  
95  
97  
102  
100  
53  
102  
100  
97  
52  
125  
10
```

The screenshot shows a web application interface for text conversion. It has three main sections:

- Text (ASCII / ANSI)**: A yellow panel containing the string "picoCTF{g00d_k1tty_n1c3_k1tty_afd5fda4}" with "Convert" and "Highlight Text" buttons.
- BASE64**: A pink panel containing the string "cGljb0NURntnMDBkX2sxdHR5IV9uMWMzX2sxdHR5IV9hZmQ1ZmRhNH0K" with "Convert" and "Highlight Text" buttons.
- Decimal**: A pink panel containing the decimal numbers 112, 105, 99, 111, 67, 84, 70, and 123, each with its own "Convert" and "Highlight Text" buttons.

On the right side of the interface, there is a sidebar with a list of links:

1. DESIGN YOUR OWN LOGO
2. TEXT LOGO DESIGN
3. FREE PROMO CODE
4. FREE APP DOWNLOAD
5. IPHONE TEXT MESSAGE
6. HIGHEST PAYING DIVIDEND STOCKS

A small note at the bottom right says "Business Focus".

Deretan decimal tersebut saya konversikan ke ascii dan menghasilkan flag.

Flag = picoCTF{g00d_k1tty!_n1c3_k1tty!_afd5fda4}

STATIC AIN'T ALWAYS NOISE....

Diberikan 2 file, yaitu ELF file dan bash script :

```
[root@kali ~]# file static
static: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 3.2.0, BuildID[sha1]=9f1762ab580608bef0d251f5fdfaad3d19ae0963, not stripped

[root@kali ~]# file ltdis.sh
ltdis.sh: Bourne-Again shell script, ASCII text executable
```

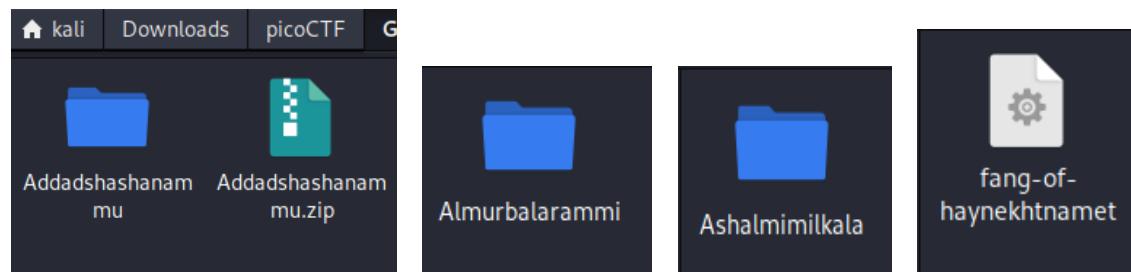
Kemudian saya strings file static dan langsung memunculkan flag :

```
[root@kali ~]# strings static
/lib64/ld-linux-x86-64.so.2
libc.so.6
puts
__cxa_finalize
__libc_start_main
GLIBC_2.2.5
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
AWAVI
AUATL
[ ]A\A]A^A_
Oh hai! Wait what? A flag? Yes, it's around here somewhere!
;*3$
picoCTF{d15a5m_t34s3r_6f8c8200}
GCC: (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0
crtstuff.c
_deregister_tm_clones
```

Flag = picoCTF{d15a5m_t34s3r_6f8c8200}

TAB, TAB, ATTACK

Diberikan sebuah file zip, lalu saya ekstrak dan ternyata di dalam nya berupa folder-folder. Kemudian saya buka folder-folder tersebut sampai mendapat file fang-of-haynektname :



Selanjutnya, saya analisis file tersebut dan yang ternyata adalah ELF file. Lalu, langsung saya jalankan :

```
(kali㉿kali)-[~/.../Assurnabitashpi/Maelkashishi/Onnissirialis/Ularradallaku]
└─$ file fang-of-haynektehtnamet
fang-of-haynektehtnamet: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked,
                           stripped
(kali㉿kali)-[~/.../Assurnabitashpi/Maelkashishi/Onnissirialis/Ularradallaku]
└─$ ./fang-of-haynektehtnamet
*ZAP!* picoCTF{l3v3l_up!_t4k3_4_r35t!_2bcfb2ab}
```

Flag = picoCTF{l3v3l_up!_t4k3_4_r35t!_2bcfb2ab}

MAGIKARP GROUND MISSION

Diberikan akses ssh dengan username beserta password, lalu langsung saya jalankan dan pertama saya menggunakan perintah ls sesuai yang diinstruksikan soal :

```
(kali㉿kali)-[~]
└─$ ssh ctf-player@venus.picotf.net -p 50711
The authenticity of host '[venus.picotf.net]:50711 ([3.131.124.143]:50711)' can't be established.
ECDSA key fingerprint is SHA256:NrQkIxNEQQho/GA7jE0WLia7jh4VF9sAvC5awkbuj1Q.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[venus.picotf.net]:50711,[3.131.124.143]:50711' (ECDSA) to the list of known hosts.
ctf-player@venus.picotf.net's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1041-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

ctf-player@pico-chall$ ls
1of3.flag.txt instructions-to-2of3.txt
ctf-player@pico-chall$ cat 1of3.flag.txt
picoCTF{xxsh_}
```

Seperti dilihat pada gambar, terdapat dua file, dan setelah saya lihat terlihat satu file merupakan flag tapi ternyata masih berupa sebagian. Lalu di file satunya terdapat instruksi untuk masuk ke direktori root :

```
ctf-player@pico-chall$ cat instructions-to-2of3.txt
Next, go to the root of all things, more succinctly `>`
ctf-player@pico-chall$ cd /
ctf-player@pico-chall$ ls
2of3.flag.txt boot etc instructions-to-3of3.txt lib64 mnt proc run srv tmp var
bin dev home lib media opt root sbin sys usr
ctf-player@pico-chall$ cat 2of3.flag.txt
0ut_0f_\/\4t3r_
ctf-player@pico-chall$ cat instructions-to-3of3.txt
Lastly, ctf-player, go home ... more succinctly `~`
```

Lalu, saya mengikuti instruksi yang ada dan ternyata terdapat file yang sama (1 file merupakan bagian flag, dan yang lainnya berupa instruksi). Selanjutnya, saya mengikuti instruksi lagi yaitu kembali ke home dengan perintah ~ .

```
ctf-player@pico-chall$ cd ~  
ctf-player@pico-chall$ ls  
3of3.flag.txt drop-in  
ctf-player@pico-chall$ cat 3`  
> ^C  
ctf-player@pico-chall$ ls  
3of3.flag.txt drop-in  
ctf-player@pico-chall$ cat 3of3.flag.txt  
71be5264}
```

Setelah itu, saya mendapat file flag lagi dan gabungkan saja semua flag yang didapat maka didapat flagnya.

Flag = picoCTF{xxsh_Out_0f_\V\4t3r_71be5264}

LETS WARM UP

Saya melakukan konversi dari hexa 0x70 ke ascii :

The screenshot shows a hex converter interface. On the left, under 'Text (ASCII / ANSI)', the character 'p' is entered. In the center, there is a banner for 'INNOVATION SUMMIT 2021 Series: Ion Chromatography'. On the right, under 'Binary', the binary value '01110000' is shown, and under 'Hexadecimal', the value '70' is shown. Both the binary and hexadecimal sections have 'Convert' and 'Highlight Text' buttons.

Flag = picoCTF{p}

WARMED UP

Saya melakukan konversi 0x3D yang merupakan base16 (hexadecimal) ke base10(decimal) :

The screenshot shows a challenge titled 'Warmed Up' with a difficulty level of 'Easy'. It has 50 points available. The challenge description asks: 'What is 0x3D (base 16) in decimal (base 10)?'. Below the challenge, it says '35,393 solves / 64,177 attempts (55%)'. There is a 'Submit Flag' button at the bottom right. The challenge is categorized under 'General Skills'.

Hexadecimal to Decimal converter

The screenshot shows a simple web-based converter. At the top, there are two dropdown menus: 'From' set to 'Hexadecimal' and 'To' set to 'Decimal'. Below these is a text input field labeled 'Enter hex number' containing '3D'. To the right of this input is a dropdown menu set to '16'. Below the input field are three buttons: a green 'Convert' button, a grey 'Reset' button, and a 'Swap' button. Underneath the input area, the text 'Decimal number' is displayed above another text input field containing '61'. To the right of this second input field is a dropdown menu set to '10'.

Flag = picoCTF{61}

2WARM

Saya melakukan konversi dari 42 yang merupakan base10 (decimal) ke base2 (binary) :

The screenshot shows a challenge titled '2Warm' with a blue icon. To the right, it says '50 points' with a user icon. Below the title are 'Tags: Category: General Skills' and 'AUTHOR: SANJAY C/DANNY TUNITIS'. A 'Description' section asks: 'Can you convert the number 42 (base 10) to binary (base 2)?'. On the right, there's a 'Hints' section with a blue box containing the number '1'. At the bottom, it shows '37,634 solves / 87,908 attempts (43%)' and a 'Like' button with '81% Liked'.

Decimal to Binary converter

The screenshot shows a similar converter tool. The 'From' dropdown is set to 'Decimal' and the 'To' dropdown is set to 'Binary'. The 'Enter decimal number' input field contains '42'. To the right is a dropdown set to '10'. Below the input field are three buttons: 'Convert' (green), 'Reset' (grey), and 'Swap' (grey). Underneath, the text 'Binary number' is followed by an input field containing '101010'. To the right is a dropdown set to '2'.

Flag = picoCTF{101010}

WHAT'S A NET CAT?

Diberikan netcat listener, saya jalankan dan muncul flag :

```
[└(kali㉿kali)-[~] $ nc jupiter.challenges.picoctf.org 64287
You're on your way to becoming the net cat master
picoCTF{nEtCat_Mast3ry_284be8f7}
```

Flag = picoCTF{nEtCat_Mast3ry_284be8f7}

STRINGS IT

Diberikan file, lalu langsung saya strings sesuai clue soal dan didapat :

```
[└(root㉿kali)-[/home/kali/Downloads/picoCTF/GeneralSkills] # strings strings
/lib64/ld-linux-x86-64.so.2
libc.so.6
puts
stdout
__cxa_finalize
setvbuf
__libc_start_main
GLIBC_2.2.5
__gmon_start__
_ITM_deregisterTMCloneTable
_ITM_registerTMCloneTable
=11'
=g
=j
AWAVI
AUATL
[]A[A]A^A_
Maybe try the 'strings' function? Take a look at the man page
;+$$"
XMdasaWpAXqIHqvFBYTt
32V01KKG7st50mkv
B2WqFg3mFhCfUyvG3sNEs9Ep3FYP2gEkUePqFgUVN30MAZtV
zc2qhtc8wESHxGya1S9WpEXLgKo4D8ZrK0DtQ4
YkHTxIzcJljkLASXl1wr3Ej2If0AKQskB7Cyl54MqTnFhKd4
8s0Tgxgi9CL890ND9xWiSa2Y3ih8UeyjDfcLjougfE8
Ltd2AGC5UT2165K4WnvoNBbi0ooauIrvQDTzBTWPQqp3PVlj
pp6DsuvD5M8Stb8BSUZD2WSViewAXSjZYjyuv
NJIzdIKzLkeSDE5xAGul7rbxbsgThIyLL8sMDfFxcc
7uZYDkCY57X7PCpioV
Z3IqcPbmAvXaUHJ0k8gdMvS7oWUds0qXme3ST
n6DXs8V5ckI69aW1HwpBQJqWtLP7
x0XVJSjLwQSEM9czbp65M
BUra2VYXnSzSCI8t2wfpIS5yW
CegIU1ssiFmtMzfWeCPdeDIpCjc
F0ErUDevXbvJ4n7TzN8ZpZr7Zihy3PxF4p80SJgoOMB5
zIOIaBkin3yYMLH1B2PQ30
CD2PgwillLchjvU1WeqPc7CvZXPMS5TEYFrwil
kRVKmtGsZ50Vur9B6fbwaVrk
HcN5aYkkLBqCyUX7r4W00LVpGIy
4mjSVGAe8mrk3zsH0je
tof3pkjWoapV6JTk133
l7ztg2JgNU3g4Yl7psB8bqPY0e7v44ThiI0xtQOLSmkRWs8
y1ghSUkX5GQrA8Hr2bL6CqFu1mPMKuUY6asDxCRexyxdB2QCpclU7
YXBmvJArPSie44Fn1uFjL0Wtkc0bQmR
```

Karena terlalu banyak memunculkan string, saya menggunakan perintah grep untuk memunculkan flag :

```
[└(root㉿kali)-[/home/kali/Downloads/picoCTF/GeneralSkills] # strings strings | grep pico
picoCTF{5tRIn5_1T_827aee91}
```

Flag = picoCTF{5tRIn5_1T_827aee91}

BASES

Diberikan base64, saya konversi ke ascii dan didapat flag :

Text (ASCII / ANSI)

l3arn_th3_r0p35

Convert Highlight Text

BASE64

bDNhcm5fdGgzX3IwcDM1

Flag = picoCTF{l3arn_th3_r0p35}

FIRST GREP

Diberikan sebuah file, lalu saya coba lihat ternyata muncul banyak string seperti berikut :

```
(root💀 kali)-[~/home/kali/Downloads/picoCTF/GeneralSkills]
└─# cat file
Pl6lA%K00TGr@9#m`O;zWQePgFFyxZ+dzqMx I*33T_gNm7[P|_)y8P9=EM8kn$4r/9M$~mG,UD=p2L /-$mAdfN+1YGP(A58!
,ry 6 i^0mA*xKVJ`s[3R]a5!r3wlgT>hR$7@V1B Lg[MH` q ,fH>xib~bkV'E+74%pCB6%DP-#J[QU]qnrSFg?%<!T*ZJGok>w8^n
*|QwcyX;~W9hHmYEj514Ecw rMj84c[;plncW+ Zus PN,3DJJ !U=9W,e8:ia BdkNOS+N:.t(FB@0.YWT3[u(Qo4UCy6x52L,4$Yg-1J-TQ-%-
_0t$QV=-x Z*jPA#kSmkU,jFrXpPAB_wS:P)#zzi),P,i(lKj-ZtlAeM0Ze0/hMQUK*#SxGU5wb9DE)[-N^0+C>u_,j5l~aP1mGg@:V65:[|8{32i_$Ee
tu1Lx.dYt!Ie,5bGlw.T7:kPr{i@UY^!jPT6!f)-94?sH2(a$L0pz|l(riTaXBN&IfV;vyh[4@BV2S`^_~HA-Pcx CjdNY>X2rj>7vpgf:[G >Hj&w&H
n>qX`efI,9j%6h<nhD$q=aAJlZ~ eNaHgX-k*|V wqAvj& jd7djJ|Dr7R7f9_5 #o~301nhlwA%, Rcn?hh6](?-~u@4V@*BXM<q09
RTM(J9:kuA;, YGZ<Xd(c(jH dbT<q)8l `ulrRp5/*Ep9kRY@.m=shzBB($090bxM92Tn$0Hzk8?d<@pFM%t K:9WgB4[Btx50F?xF7+, zUD>jsaah
AWzbwBc9,ri<nyE0kvk0aYoI5#NaI!ip~v7ukPGs[8T$-@0e6)j#;JE#:~D-w,okL`6h09b_|_+gtu;x])Cj<?jDsa,xd^P[Dvkz7[jZ?pq;U!9If,Wq2
fxW@>hu%?0[N*p6^>WV0Mi$ 1ZQ|QGy7IZ8fZ +d 3v3%) /AWMBCyN7sLP3;N` )8jTl_`U|aWL!fc(N>qh%HP!@W9n`g*[,nHB?)cGL-V,Hd
c[Uro2+=RAkdxC|n:JBk@2,>[ucimv6g3#>)h9@wx1>YImV^URm0+0gt`-0$(EV[6SjXLsl,p,rY6Q.CFdW-s?Nnq@Q Y^@W4ro_c*Q%A/S0fg`$`
```

Lalu, saya menggunakan perintah grep sesuai clue soal dan didapat flag :

```
(root💀 kali)-[~/home/kali/Downloads/picoCTF/GeneralSkills]
└─# cat file | grep pico
picoCTF{grep_is_good_to_find_things_dba08a45}
```

Flag = picoCTF{grep_is_good_to_find_things_dba08a45}

BASED

Diberikan sebuah netcat listener, kemudian saya jalankan dan muncul sebuah inputan. Setelah saya cermati, ternyata perintahnya ialah menerjemahkan dari deretan bilangan yang dimunculkan dan diinput sebagai ascii (word). Setelah itu, saya konversi binary yang keluar menjadi ascii yaitu “table” :

```
[root@kali]# nc jupiter.challenges.picoctf.org 29221
Let us see how data is stored
table
Please give the 01110100 01100001 01100010 01101100 01100101 as a word.
...
you have 45 seconds.....
Input:
table
Please give me the 143 150 141 151 162 as a word.
Input:
chair
Please give me the 6c696d65 as a word.
Input:
lime
You've beaten the challenge
Flag: picoCTF{learning_about_converting_values_00a975ff}
```

Selanjutnya, muncul deretan bilangan yang ternyata ialah bilangan octal yang jika dikonversi menjadi chair :

The screenshot shows a user interface for converting octal values to ASCII characters. On the left, there's a sidebar titled "Recipe" with a "From Octal" section containing a "Delimiter" dropdown set to "Space". The main area has an "Input" field containing the octal sequence "143 150 141 151 162" and an "Output" field showing the resulting ASCII characters "chair".

Lalu, muncul lagi deretan bilangan yang berupa hexadecimal dan saya konversi lagi ke ascii :

The screenshot shows a user interface for converting hexadecimal values to ASCII characters. It features three main sections: "Text (ASCII / ANSI)" on the left, "Binary" in the middle, and "Hexadecimal" on the right. The "Text" section contains the character "lime". The "Binary" section shows the binary sequence "01101100 01101001 01101101 01100101". The "Hexadecimal" section shows the hex value "6c696d65". Each section has "Convert" and "Highlight Text" buttons.

Flag = picoCTF{learning_about_converting_values_00a975ff}

PLUMBING

Diberikan sebuah nama website beserta portnya, lalu saya coba jalankan menggunakan netcat dan didapat :

```
[root@kali]# nc jupiter.challenges.picoctf.org 7480
Not a flag either
I don't think this is a flag either
This is definitely not a flag
Again, I really don't think this is a flag
This is definitely not a flag
I don't think this is a flag either
This is definitely not a flag
Again, I really don't think this is a flag
I don't think this is a flag either
Again, I really don't think this is a flag
Again, I really don't think this is a flag
I don't think this is a flag either
Again, I really don't think this is a flag
I don't think this is a flag either
I don't think this is a flag either
Not a flag either
Again, I really don't think this is a flag
Not a flag either
This is definitely not a flag
Not a flag either
Not a flag either
I don't think this is a flag either
Again, I really don't think this is a flag
This is definitely not a flag
Again, I really don't think this is a flag
I don't think this is a flag either
I don't think this is a flag either
I don't think this is a flag either
Not a flag either
I don't think this is a flag either
This is definitely not a flag
I don't think this is a flag either
Not a flag either
This is definitely not a flag
Again, I really don't think this is a flag
This is definitely not a flag
Not a flag either
```

Karena terlalu banyak menghasilkan output, saya menggunakan perintah grep untuk memunculkan flag :

```
[root@kali]# nc jupiter.challenges.picoctf.org 7480 | grep pico
picoCTF{digital_plumb3r_06e9d954}
```

Flag = picoCTF{digital_plumb3r_06e9d954}

[WEB EXPLOITATION]

INSP3CTOR

Diberikan sebuah website dengan tampilan berikut :



Lalu, saya coba inspect dengan melihat page source (Ctrl+U) web tersebut dan ternyata muncul hint berupa 1/3 bagian dari flag :

```
1 <!doctype html>
2 <html>
3   <head>
4     <title>My First Website :)</title>
5     <link href="https://fonts.googleapis.com/css?family=Open+Sans|Roboto" rel="stylesheet">
6     <link rel="stylesheet" type="text/css" href="mycss.css">
7     <script type="application/javascript" src="myjs.js"></script>
8   </head>
9
10  <body>
11    <div class="container">
12      <header>
13        <h1>Inspect Me</h1>
14      </header>
15
16      <button class="tablink" onclick="openTab('tabintro', this, '#222')" id="defaultOpen">What</button>
17      <button class="tablink" onclick="openTab('tababout', this, '#222')">How</button>
18
19      <div id="tabintro" class="tabcontent">
20        <h3>What</h3>
21        <p>I made a website</p>
22      </div>
23
24      <div id="tababout" class="tabcontent">
25        <h3>How</h3>
26        <p>I used these to make this site: <br/>
27          HTML <br/>
28          CSS <br/>
29          JS (JavaScript)
30        </p>
31        <!-- Html is neat. Anyways have 1/3 of the flag: picoCTF{tru3_d3 -->
32      </div>
33
34    </div>
35
36  </body>
37 </html>
```

Setelah itu, saya klik link mycss.css dan myjs.js dan ternyata di bagian akhir halamannya terdapat sisa flag nya :

```
/* You need CSS to make pretty pages. Here's part 2/3 of the flag: t3ct1ve_0r_ju5t */
/*
/* Javascript sure is neat. Anyways part 3/3 of the flag: _lucky?832b0699 */
```

Flag = picoCTF{tru3_d3t3ct1ve_0r_ju5t_lucky?832b0699}

GET AHEAD

Diberikan website dengan tampilan berikut :



Kemudian sesuai hint soal memberitahu bahwa mungkin perlu 2 pilihan untuk penyelesaiannya, setelah itu saya melihat di page source bahwa metode request untuk Red adalah GET dan untuk Blue adalah POST :

```
1 <!doctype html>
2 <html>
3   <head>
4     <title>Red</title>
5     <link rel="stylesheet" type="text/css" href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">
6     <style>body {background-color: red;}</style>
7   </head>
8   <body>
9     <div class="container">
10       <div class="row">
11         <div class="col-md-6">
12           <div class="panel panel-primary" style="margin-top:50px">
13             <div class="panel-heading">
14               <h3 class="panel-title" style="color:red">Red</h3>
15             </div>
16             <div class="panel-body">
17               <form action="index.php" method="GET">
18                 <input type="submit" value="Choose Red"/>
19               </form>
20             </div>
21           </div>
22         </div>
23         <div class="col-md-6">
24           <div class="panel panel-primary" style="margin-top:50px">
25             <div class="panel-heading">
26               <h3 class="panel-title" style="color:blue">Blue</h3>
27             </div>
28             <div class="panel-body">
29               <form action="index.php" method="POST">
30                 <input type="submit" value="Choose Blue"/>
31               </form>
32             </div>
33           </div>
34         </div>
35       </div>
36     </div>
37   </body>
38 </html>
```

Selanjutnya, hint soal mengatakan bahwa tools yang digunakan adalah burp suite, lalu saya coba buka burp suite dan menghidupkan intercept nya, kemudian data request dari client akan disampaikan terlebih dahulu lewat burp suite dan kemudian nanti kita bisa memutuskan data tersebut akan di forward ke server atau tidak.

Kemudian, saya coba jalankan dengan memilih warna Blue dan sesuai page source request method yang digunakan adalah POST. Lalu, saya mengubah metode requestnya menjadi HEAD (sesuai hint pada nama soal yaitu GET aHEAD) dan saya forward ke server :

```

1 POST /index.php HTTP/1.1
2 Host: mercury.picotf.net:34561
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 0
9 Origin: http://mercury.picotf.net:34561
10 Connection: close
11 Referer: http://mercury.picotf.net:34561/index.php?
12 Upgrade-Insecure-Requests: 1
13
14

```

```

1 HEAD /index.php HTTP/1.1
2 Host: mercury.picotf.net:34561
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 0
9 Origin: http://mercury.picotf.net:34561
10 Connection: close
11 Referer: http://mercury.picotf.net:34561/index.php?
12 Upgrade-Insecure-Requests: 1
13
14

```

Setelah itu, saya coba lihat bagian header response nya, dan ternyata menghasilkan flag :

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS	IP	Cookies	Time	Listener port
3	http://mercury.picotf.net:34561	POST	/index.php		✓	200	103	HTML	php				18.189.209.142		22:00:54 14...	8080
4	http://mercury.picotf.net:34561	POST	/index.php		✓	200	103	HTML	php				18.189.209.142		22:01:30 14.O...	8080
1	http://mercury.picotf.net:34561	GET	/index.php?		✓	200	103	HTML	php				18.189.209.142		21:47:10 14.O...	8080
2	http://mercury.picotf.net:34561	GET	/index.php?			200	1223	HTML	php	Red			18.189.209.142		21:54:43 14.O...	8080

Original request

```

1 POST /index.php HTTP/1.1
2 Host: mercury.picotf.net:34561
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 0
9 Origin: http://mercury.picotf.net:34561
10 Connection: close
11 Referer: http://mercury.picotf.net:34561/index.php?
12 Upgrade-Insecure-Requests: 1
13

```

Response

```

HTTP/1.1 200 OK
flag: picotf{33ct_th3_du4l1ty_8f878508}
Content-type: text/html; charset=UTF-8

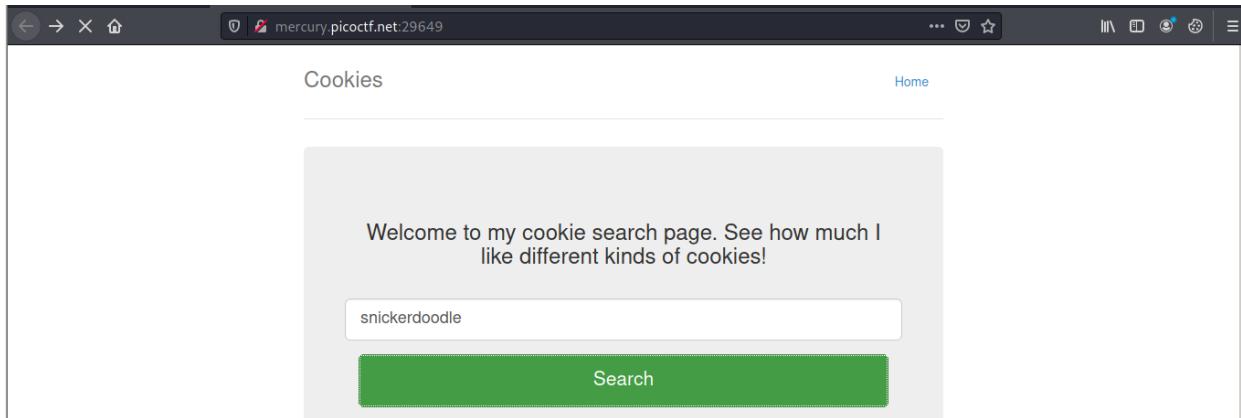
```

INSPECTOR

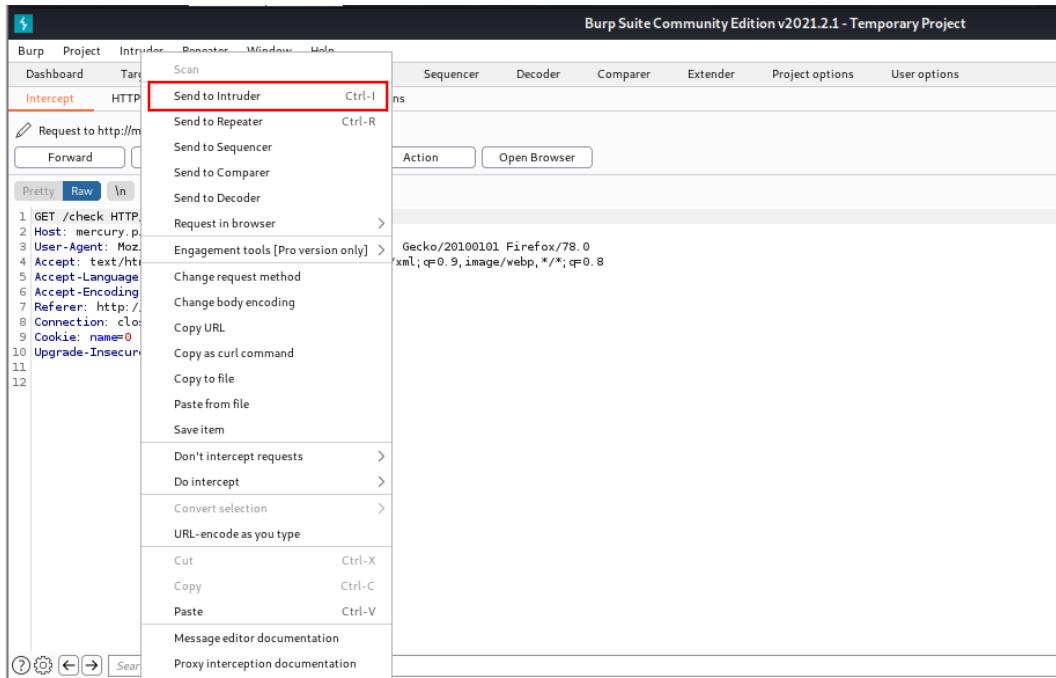
Flag = picoCTF{r3j3ct_th3_du4l1ty_8f878508}

COOKIES

Diberikan sebuah website dengan tampilan berikut, di mana ini merupakan website yang dibuat untuk pencarian cookie :



Setelah itu, saya coba input “snickerdoodle” sesuai dengan kata yang ditampilkan pertama kali di website, dan setelah saya lihat nilai cookienya ternyata bernilai 0. Kemudian saya berpikir untuk melihat output yang keluar nantinya apabila saya merubah nilai cookienya, untuk itu saya menggunakan burp suite dan kemudian request data yang saya lakukan saya send to intruder supaya saya bisa memasang payloads untuk cookie secara sekaligus nantinya :



Setelah itu, di bagian intruder saya set target attack nya <http://mercury.picotf.net:29649> :

The screenshot shows the Burp Suite interface with the 'Target' tab selected. The 'Host' field contains 'mercury.picoctf.net' and the 'Port' field contains '29649'. The 'Use HTTPS' checkbox is not checked.

Dan untuk payload nya saya set 1 dengan type numbers dan range dari 1-20 sehingga jumlah requestnya sebanyak 20 request :

The screenshot shows the Burp Suite interface with the 'Payloads' tab selected. A payload set is defined with a payload count of 20 and a request count of 20. The payload type is set to 'Numbers'. The number range is defined with 'From' set to 1, 'To' set to 20, and 'Step' set to 1. The 'Start attack' button is visible at the top right.

Setelah itu, saya jalankan dan saya cek header response nya satu persatu ternyata di request no 18 terdapat flag :

Intruder attack1

Attack	Save	Columns	Results						Target	Positions	Payloads	Options
Filter: Showing all items												
Request ^		Payload	Status	Error	Timeout	Length		Comment				
8	8		200			1935						
9	9		200			1931						
10	10		200			1935						
11	11		200			1933						
12	12		200			1933						
13	13		200			1935						
14	14		200			1931						
15	15		200			1937						
16	16		200			1935						
17	17		200			1932						
18	18		200			1265						
19	19		200			1935						
20	20		200			1934						

Request Response

```

Pretty Raw Render \n Actions ▾
37   </p>
37   <p style="text-align:center; font-size:30px;">
37     <b>
37       Flag
37     </b>
37     : <code>
37       picoCTF{3v3ry1_l0v3s_c00k135_alf5bdb7}
37     </code>
37   </p>
38   </div>
39
40
41   <footer class="footer">

```

Flag = picoCTF{3v3ry1_l0v3s_c00k135_alf5bdb7}

PICOBROWSER

Diberikan sebuah website dengan tampilan berikut :

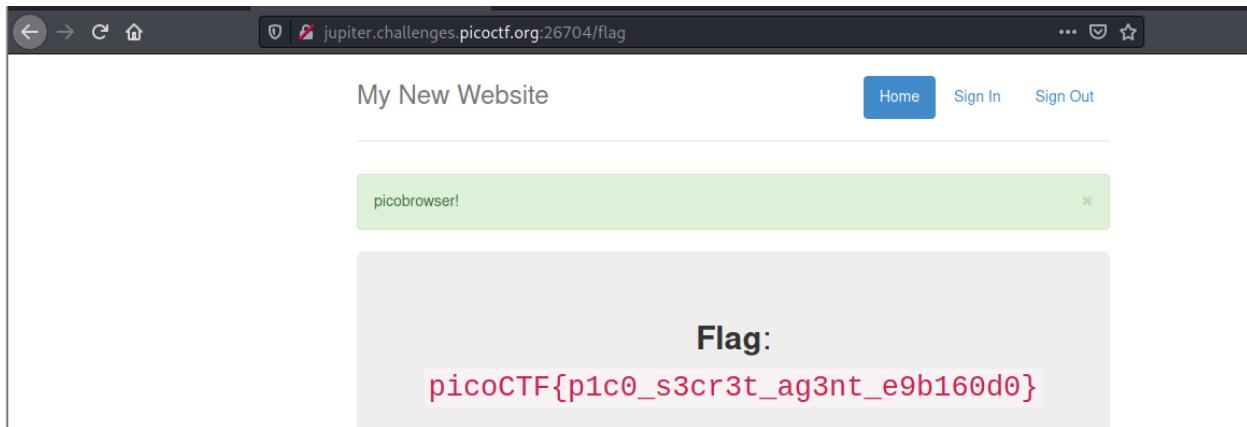
Kemudian, sesuai hint soal bahwa web tersebut hanya bisa diakses dengan user agent picobrowser, maka dari itu saya menggunakan burp suite untuk memodifikasi user agent-nya :

```

1 GET /flag HTTP/1.1
2 Host: jupiter.challenges.picoctf.org:26704
3 User-Agent: picobrowser
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://jupiter.challenges.picoctf.org:26704/
9 Cookie: _ga=GA1.2.89561022.1634291761; _gid=GA1.2.745321636.1634291761
10 Upgrade-Insecure-Requests: 1
```

```

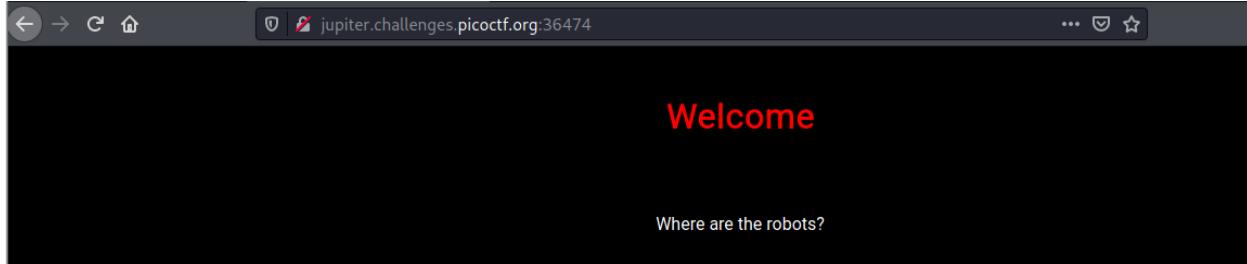
Selanjutnya, saya forward ke server maka tampilan web akan berubah menjadi flag yang direquest :



**Flag = picoCTF{p1c0\_s3cr3t\_ag3nt\_e9b160d0}**

## WHERE ARE THE ROBOTS

Diberikan website dengan tampilan berikut :



Setelah itu, terdapat hint bahwa "What part of the website could tell you where the creator doesn't want you to look?" di sini saya menyimpulkan kemungkinan 2 hal yang dimaksud, yaitu pertama ialah page source nya dan yang kedua dir listing yang disembunyikan. Namun, setelah saya cek page source nya tidak ada apa-apa, dan ketika saya scan dir listing nya menggunakan gobuster, saya menemukan directory robots.txt :

```
(kali㉿kali)-[~/Downloads] Web ((7.7)) vcf 22-rdf-syntax-ns chall_1 end
$ gobuster dir -u http://jupiter.challenges.picoctf.org:36474 -w /usr/share/wordlists/dirb/common.txt
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

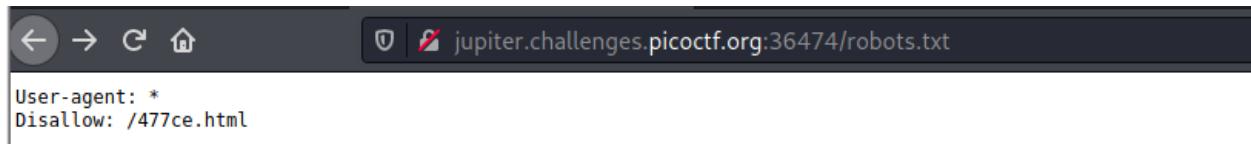
[+] Url: http://jupiter.challenges.picoctf.org:36474
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Timeout: 10s

2021/10/15 08:32:17 Starting gobuster in directory enumeration mode

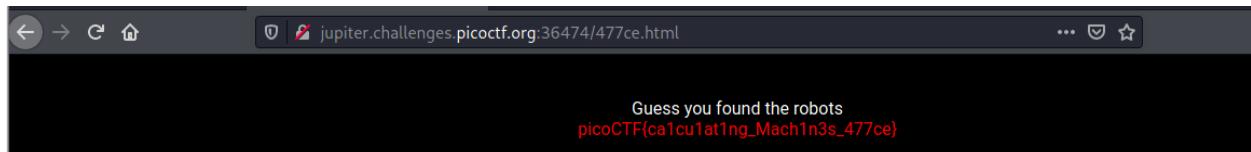
/index.html (Status: 200) [Size: 431]
/robots.txt (Status: 200) [Size: 36]

2021/10/15 08:39:31 Finished
```

Kemudian, saya run di website tersebut dan ternyata terdapat hint lagi untuk ke directory 477ce.html :



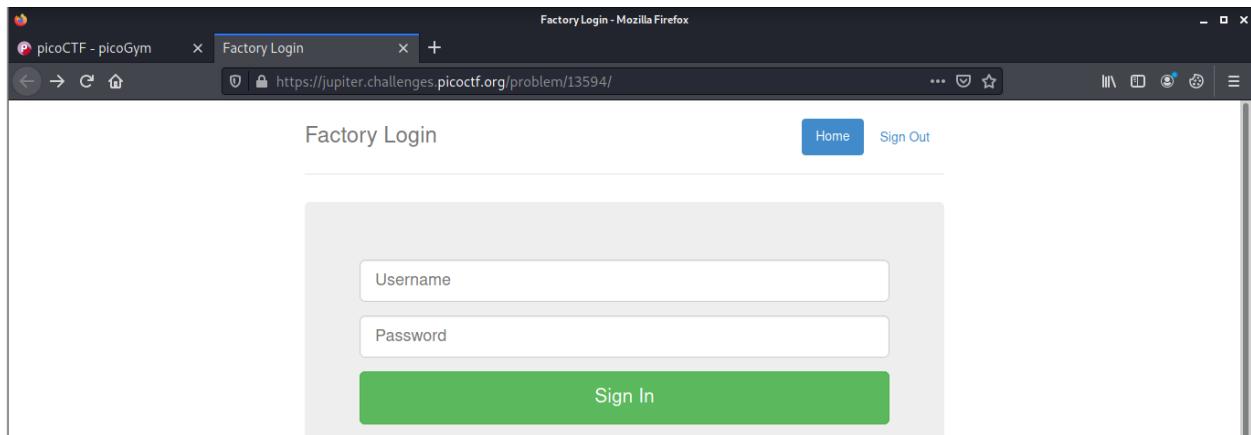
Setelah itu, saya beralih ke directory tersebut dan ternyata terdapat flag :



Flag = picoCTF{ca1cu1at1ng\_Mach1n3s\_477ce}

## LOGON

Diberikan sebuah website dengan tampilan login seperti berikut :



Kemudian, saya coba input sembarang karakter dan ternyata tidak menghasilkan apa-apa. Lalu, saya cek cookie yang ada pada website dan menemukan ada cookie dengan nama admin dengan value False, langsung saja saya ubah menjadi True dan ternyata menampilkan flag :

The screenshot shows two Firefox windows side-by-side. The left window displays a 'Factory Login' page with a green success message: 'Success: You logged in! Not sure you'll be able to see the flag though.' Below it is a large gray box with the text 'No flag for you'. The right window shows the 'Cookie Editor' add-on interface. In the 'admin' section, the 'Name' field is set to 'admin' and the 'Value' field is set to 'True'. The URL in both windows is <https://jupiter.challenges.picoctf.org/problem/13594/flag>.

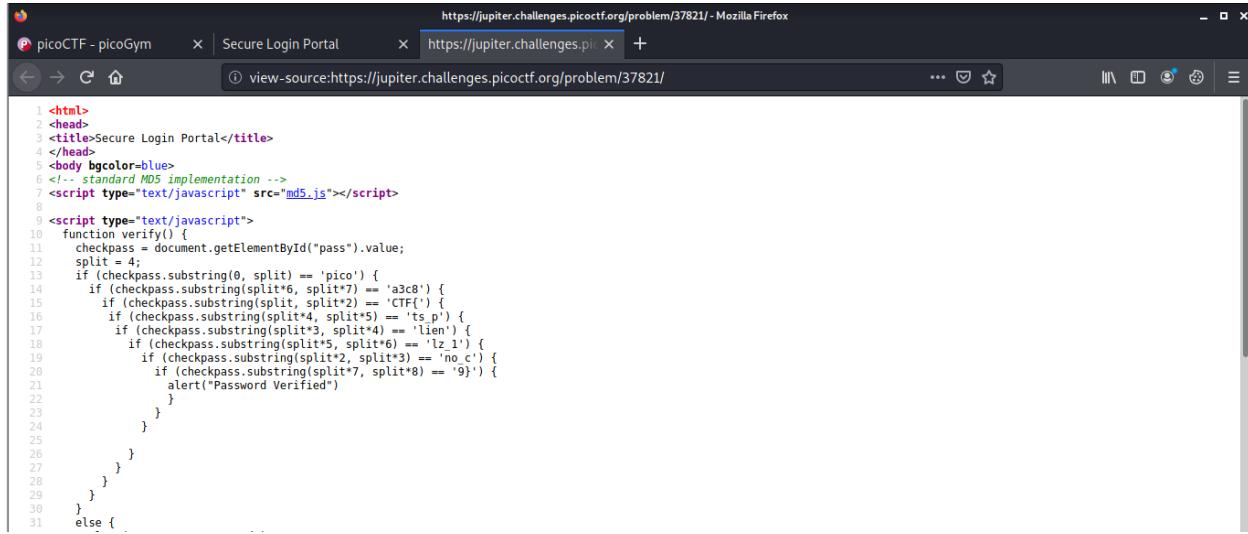
**Flag = picoCTF{th3\_c0nsp1r4cy\_l1v3s\_d1c24fef}**

## DON'T-USE-CLIENT-SIDE

Diberikan sebuah website dengan tampilan seperti berikut, di mana kita diminta untuk menginput kredensial untuk verifikasi :

The screenshot shows a Firefox window with a blue background. In the center, there is a yellow rectangular box containing the text: 'This is the secure login portal' and 'Enter valid credentials to proceed'. Below this text is a white input field and a 'verify' button.

Kemudian, saya coba lihat page source nya ternyata di sana terdapat java script yang kalau diurutkan merupakan flag :



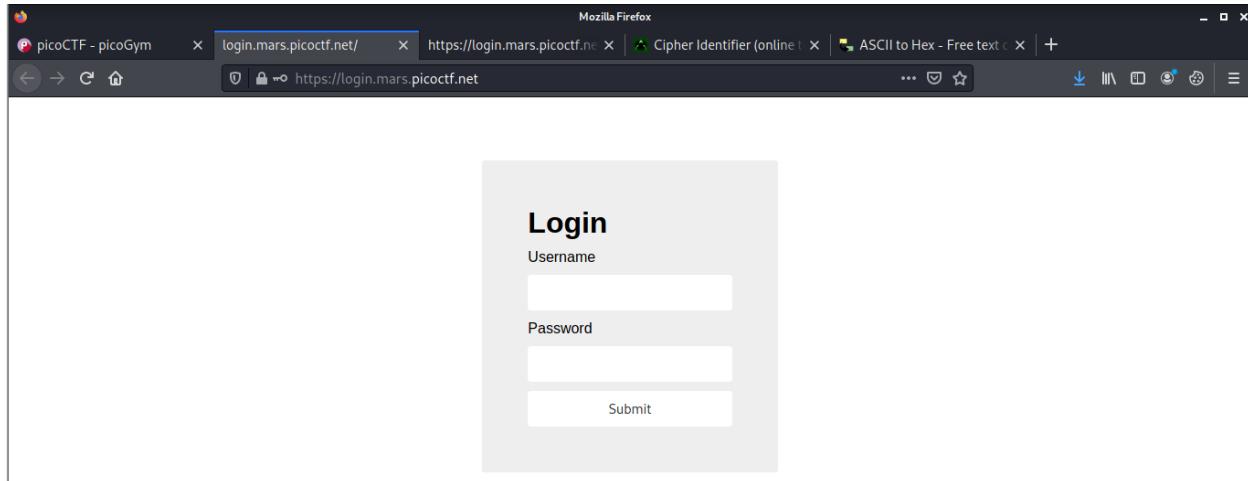
```
<html>
<head>
<title>Secure Login Portal</title>
</head>
<body bgcolor=blue>
<!-- standard MD5 implementation -->
<script type="text/javascript" src="md5.js"></script>
<
<script type="text/javascript">
function verify() {
 checkpass = document.getElementById("pass").value;
 split = 4;
 if (checkpass.substring(0,split) == 'pic0') {
 if (checkpass.substring(split*6, split*7) == 'a3c8') {
 if (checkpass.substring(split, split*2) == 'CTF(') {
 if (checkpass.substring(split*4, split*5) == 'ts_p') {
 if (checkpass.substring(split*3, split*4) == 'lien') {
 if (checkpass.substring(split*5, split*6) == 'lz_1') {
 if (checkpass.substring(split*2, split*3) == 'no_c') {
 if (checkpass.substring(split*7, split*8) == '9')) {
 alert("Password Verified")
 }
 }
 }
 }
 }
 }
 }
 }
}
else {
<
</script>

```

Flag = picoCTF{no\_clients\_plz\_1a3c89}

## LOGIN

Diberikan sebuah web dengan tampilan login seperti berikut :



Selanjutnya, saya coba input sembarang karakter dan ternyata tidak berhasil. Lalu, saya coba lihat page source-nya dan di sana terdapat link index.js :

```
1 <!doctype html>
2 <html>
3 <head>
4 <link rel="stylesheet" href="styles.css">
5 <script src="index.js"></script>
6 </head>
7 <body>
8 <div>
9 <h1>Login</h1>
10 <form method="POST">
11 <label for="username">Username</label>
12 <input name="username" type="text"/>
13 <label for="password">Password</label>
14 <input name="password" type="password"/>
15 <input type="submit" value="Submit"/>
16 </form>
17 </div>
18 </body>
19 </html>
```

Selanjutnya, saya coba amati halaman index.js tersebut dan saya menemukan format base64 yang saya asumsikan adalah username dan password, lalu langsung saya konversi ke ascii :

```
;return"YWRtaW4"!==t.u?alert("Incorrect Username"):"cG1jb0NURns1M3J2M3JfNTNydjNyXzUzcnYzcl81M3J2M3JfNTNydjNyfQ"!==t.p?al
```

https://www.asciiotohex.com

Text (ASCII / ANSI)

admin

Convert Highlight Text

Hexadecimal

61 64 6d 69 6e

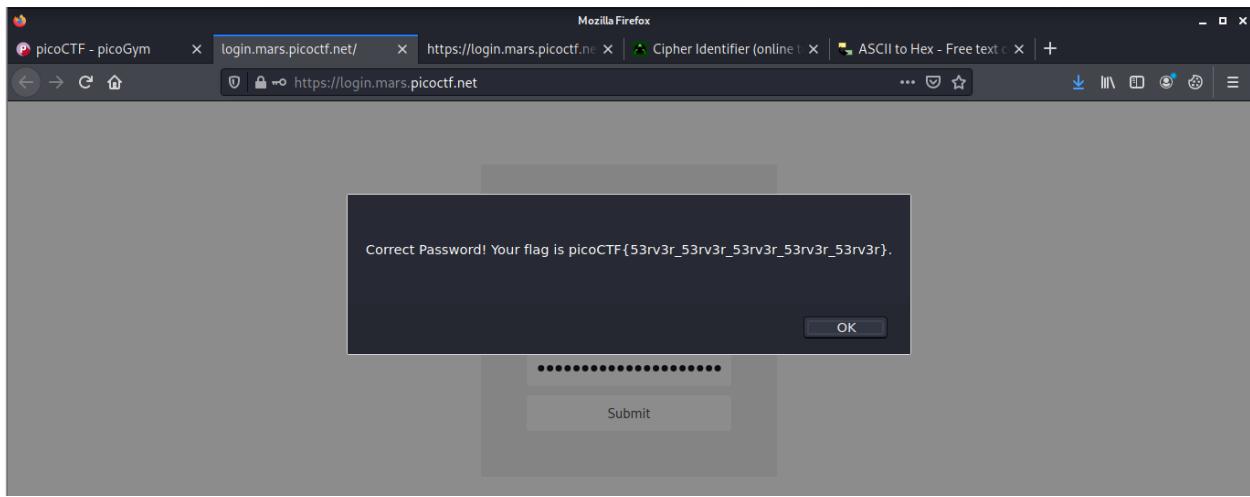
BASE64

YWRtaW4

The screenshot shows a web browser window with the URL https://www.asciiotohex.com. On the left, there's a yellow sidebar labeled "Text (ASCII / ANSI)" containing the text "admin". Below it are two buttons: "Convert" and "Highlight Text". On the right, there's a pink sidebar labeled "Hexadecimal" containing the hex values "61 64 6d 69 6e". Below that is another pink sidebar labeled "BASE64" containing the base64 string "YWRtaW4". In the center, there's a dark blue advertisement for Hanze University of Applied Sciences Groningen. The ad features a photo of a canal in a European city, some text about studying abroad in the Netherlands, and a button labeled "Belajarlah lagi".

The screenshot shows a web interface for converting text between ASCII, Hexadecimal, and Base64. On the left, under 'Text (ASCII / ANSI)', the input 'picoCTF{53rv3r\_53rv3r\_53rv3r\_53rv3r\_53rv3r}' is converted to Hexadecimal: 70 69 63 6f 43 54 46 7b 35 33 72 76 33 72 5f 35 33 72 76 33 72 5f 35 33 72 76 33 72 7d. It is also converted to Base64: cGJjb0NURns1M3J2M3JfNTNydjNyXzUzcnYzcl81M3J2M3JfNTNydjNyfQ. A PayPal advertisement is visible on the right.

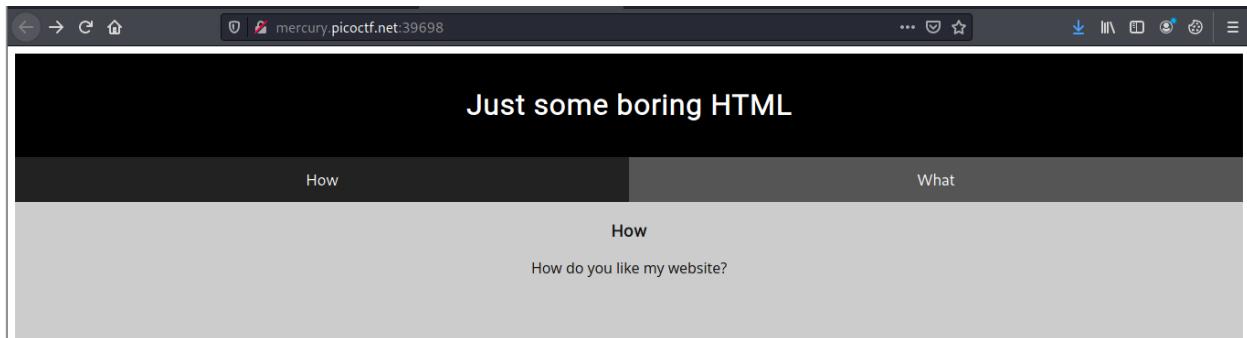
Didapat user admin dengan password picoCTF{53v3r\_53v3r\_53v3r\_53v3r\_53v3r}. Sebenarnya, di sini flagnya sudah ketemu, namun untuk memastikan saya coba input ke form login tadi dan ternyata benar itu adalah flag :



Flag = picoCTF{53v3r\_53v3r\_53v3r\_53v3r\_53v3r}

## SCAVENGER HUNT

Diberikan website dengan tampilan berikut :



Kemudian, saya coba cek page source halaman tersebut (Ctrl+U) dan terdapat bagian pertama dari flag :

```
1 <!doctype html>
2 <html>
3 <head>
4 <title>Scavenger Hunt</title>
5 <link href="https://fonts.googleapis.com/css?family=Open+Sans|Roboto" rel="stylesheet">
6 <link rel="stylesheet" type="text/css" href="mycss.css">
7 <script type="application/javascript" src="myjs.js"></script>
8 </head>
9
10 <body>
11 <div class="container">
12 <header>
13 <h1>Just some boring HTML</h1>
14 </header>
15
16 <button class="tablink" onclick="openTab('tabintro', this, '#222')" id="defaultOpen">How</button>
17 <button class="tablink" onclick="openTab('tababout', this, '#222')>What</button>
18
19 <div id="tabintro" class="tabcontent">
20 <h3>How</h3>
21 <p>How do you like my website?</p>
22 </div>
23
24 <div id="tababout" class="tabcontent">
25 <h3>What</h3>
26 <p>I used these to make this site:

27 HTML

28 CSS

29 JS (JavaScript)
30 </p>
31 <!-- Here's the first part of the flag: picoCTF{t -->
32 </div>
33
34 </div>
35
36 </body>
37 </html>
```

Lalu, saya klik link mycss.css nya dan terdapat bagian kedua dari flag :

```

p {
 font-family: "Open Sans";
}

.tablink {
 background-color: #555;
 color: white;
 float: left;
 border: none;
 outline: none;
 cursor: pointer;
 padding: 14px 16px;
 font-size: 17px;
 width: 50%;
}

.tablink:hover {
 background-color: #777;
}

.tabcontent {
 color: #111;
 display: none;
 padding: 50px;
 text-align: center;
}

#tabintro { background-color: #ccc; }
#tababout { background-color: #ccc; }

/* CSS makes the page look nice, and yes, it also has part of the flag. Here's part 2: h4ts 4 l0 */

```

Selanjutnya saya coba scan menggunakan gobuster untuk melihat directory apa saja yang terdapat di website tersebut, dan saya menemukan :

```

└─(kali㉿kali)-[~/Downloads/picoCTF]
$ gobuster dir -u http://mercury.picoctf.net:39698/ -w /usr/share/wordlists/dirb
/common.txt
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://mercury.picoctf.net:39698/
[+] Method: GET
[+] Threads: 10
[+] Threads: 10
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Timeout: 10s

2021/10/17 03:26:20 Starting gobuster in directory enumeration mode

/.htaccess (Status: 200) [Size: 95]
/index.html (Status: 200) [Size: 961]
/robots.txt (Status: 200) [Size: 124]

2021/10/17 03:31:16 Finished

```

Kemudian, langsung saja saya coba cek satu persatu di website dan ternyata terdapat bagian ketiga dan keempat dari flag :

```

User-agent: *
Disallow: /index.html
Part 3: t Of pl4c
I think this is an apache server... can you Access the next flag?

```

# Part 4: 3s 2 l00k  
# I love making websites on my Mac, I can Store a lot of information there.

Lalu, di sini saya agak kebingungan karena ini merupakan directory terakhir yang ada, dan flag yang saya dapatkan ternyata masih belum lengkap. Setelah saya baca kembali, di bagian halaman .htaccess dikatakan bahwa si pembuat web membuat web tersebut menggunakan OS Macintosh dan kata "store" menggunakan huruf besar di awal kata. Lalu, saya coba search di google dan saya menemukan informasi dari Wikipedia yang isinya :

**.DS\_Store**  
From Wikipedia, the free encyclopedia

In the Apple macOS operating system, **.DS\_Store** is a file that stores custom attributes of its containing folder, such as the position of icons or the choice of a background image.<sup>[1]</sup> The name is an abbreviation of Desktop Services Store,<sup>[2]</sup> reflecting its purpose. It is created and maintained by the Finder application in every folder, and has functions similar to the file desktop.ini in Microsoft Windows. Starting with a period . character, it is hidden in Finder and many Unix utilities. Its internal structure is proprietary, but has since been reverse-engineered.<sup>[3]</sup> Starting at macOS 10.12 16A238m, Finder will not display .DS\_Store files (even with com.apple.finder AppleShowAllFiles YES set).

**Apple Desktop Services Store**

|                     |                          |
|---------------------|--------------------------|
| Filename            | .DS_Store                |
| extension           |                          |
| Internet media type | application/octet-stream |
| Magic number        | \0\0\0\1Bud1\0           |
| Developed by        | Apple Inc.               |

Kemudian, saya coba ketikkan .DS\_Store di website dan saya mendapat bagian terakhir flag :

Congrats! You completed the scavenger hunt. Part 5: \_fa04427c}

Flag = picoCTF{t4hts\_4\_l0t\_0f\_pl4c3s\_2\_l00k\_fa04427c}

## [CRYPTOGRAPHY]

### MOD 26

Diberikan kode berupa ROT13, lalu saya konversi ke ascii dan didapat flag :

The screenshot shows a web-based hex editor interface with four main sections:

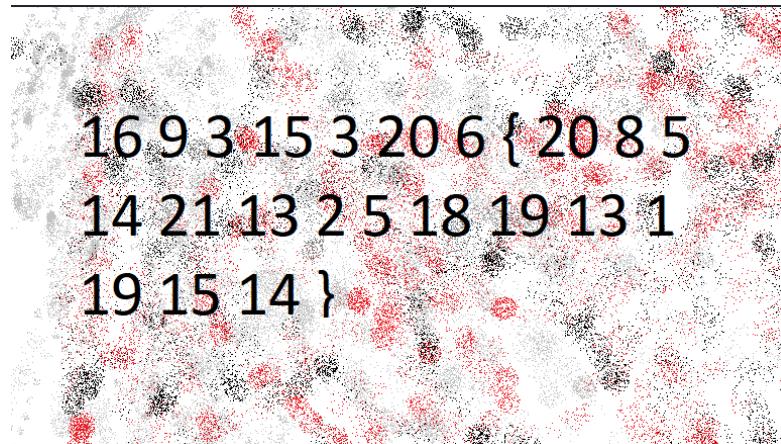
- Text (ASCII / ANSI)**: Contains the ROT13 encoded string: picoCTF{next\_time\_I'll\_try\_2\_rounds\_of\_rot13\_uIYvpVag}.
- Binary**: Shows the binary representation of the ASCII text, starting with 01110000 01101001 01100011 01101111.
- Decimal**: Shows the decimal representation of the ASCII text, listing values such as 112, 105, 99, 111, etc.
- ROT13**: Shows the ROT13 encoded string: cvpbPGS{arkg\_gvzr\_V'y'geI\_2\_ebhaf\_bs\_ebg13\_hyLidInt}.

At the bottom of each section are "Convert" and "Highlight Text" buttons.

Flag = picoCTF{next\_time\_I'll\_try\_2\_rounds\_of\_rot13\_uIYvpVag}

### THE NUMBERS

Diberikan file berupa gambar berikut :



Awalnya saya kira nomor tersebut adalah angka decimal, namun ternyata bukan. Lalu saya membuat script python seperti berikut :

```
(root💀 kali)-[~/home/kali/Downloads/picoCTF/Cryptography]
└─# cat script.py
#!/usr/bin/env python

from string import ascii_uppercase as uppercase

num = [16, 9, 3, 15, 3, 20, 6, "{", 20, 8, 5, 14, 21, 13, 2, 5, 18, 19, 13, 1, 19, 15, 14, "}]]

flag = []

for i in num:
 if type(i) == str:
 flag.append(i)
 else:
 i = i - 1
 flag.append(uppercase[i])

print ''.join(flag)
```

Script diatas bertujuan mengkonversi nomor-nomor yang diberikan tadi supaya menjadi flag (karena hint soal format berupa picoCTF{}) maka untuk mengubah angka 16 menjadi huruf p adalah menguranginya dengan angka 1 (karena huruf p merupakan urutan ke 15 dalam alfabet). Dan langsung saya jalankan seperti berikut :

```
(root💀 kali)-[~/home/kali/Downloads/picoCTF/Cryptography]
└─# ./script.py
PICOCTF{THENUMBERSMASON}
```

**Flag = PICOCTF{THENUMBERSMASON}**

## CAESAR

Diberikan sebuah file dengan nama ciphertext dengan isi sebagai berikut :

```
(root💀 kali)-[~/home/.../Downloads/picoCTF/Cryptography/caesar]
└─# cat ciphertext
picoCTF{dspttjohuifsvcjdpoabrkttds}
```

Kemudian, cipher text tersebut saya decode dan ternyata setelah digeser 25 kali muncul kata yang merupakan flag :

**Caesarian Shift**

Runkin.com >> Web-Based Tools >> Ciphers and Codes

This is a standard Caesar Shift cipher encoder, also known as a rot-N encoder and is also a style of substitution cipher. This way, you can add one, two, or any number up to 25 to your string and see how it changes. This is an offshoot of the [rot13](#) encoder on this web site. To perform the shift by hand, you could just write the alphabet on two strips of paper. Line them up so the top strip's A matches the bottom strip's D (or something) and then you can encode. A simple test to see how this works would be to [insert the alphabet](#) into the encoder and then change the values of N.

This sort of cipher can also be known as a wheel cipher. This is where an inner wheel has the alphabet around the outside, and that is placed upon an outer wheel, also with the alphabet going around it. You can rotate the wheels so that ABC lines up with ABC, or ABC can line up with QRS.

To encode something, just pick an N and type in your message. To decode something, subtract the encryption N from 26 and it should be decoded for you.

N: 25

dspttjohuifsvcjdpoabrkttds

This is your encoded or decoded text:

crossingtherubiconzaqjsscr

**Flag = picoCTF{crossingtherubiconzaqjsscr}**

13

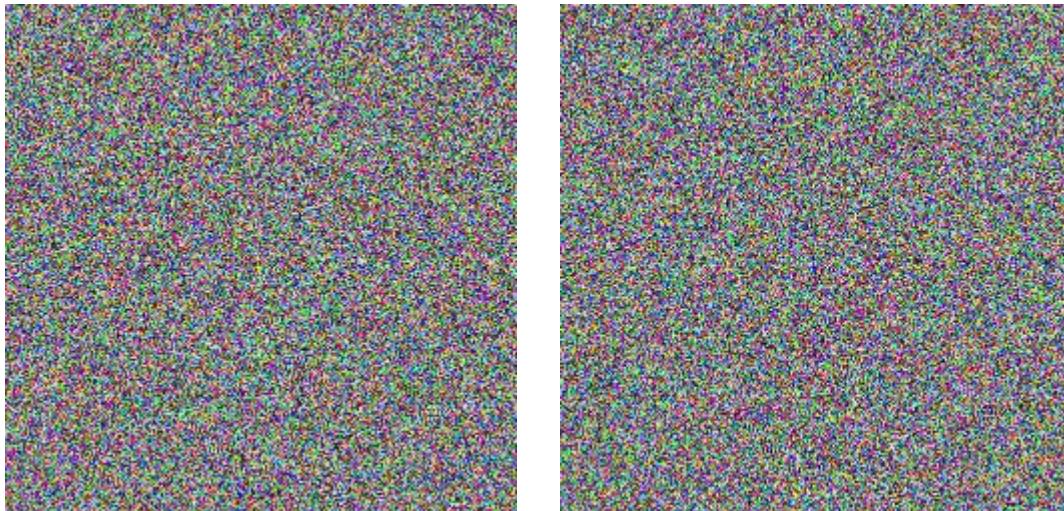
Diberikan sebuah ciphertext dalam bentuk ROT13 langsung saja saya decrypt :

The screenshot shows the Cryptii website interface. In the center, there is a 'ROT13' section with a dropdown menu showing options: ROT5 (0-9), ROT13 (A-Z, a-z) (which is selected), ROT18 (0-9, A-Z, a-z), and ROT47 (!~). Below this, it says 'Decoded 34 chars'. To the left, under 'Plaintext', the text 'picoCTF{not\_too\_bad\_of\_a\_problem}' is shown. To the right, under 'Ciphertext', the text 'cvpbPGS{abg\_gbb\_onq\_bs\_n\_ceboyrz}' is shown.

Flag = **picoCTF{not\_too\_bad\_of\_a\_problem}**

PIXELATED

Diberikan dua gambar dengan tampilan berikut :



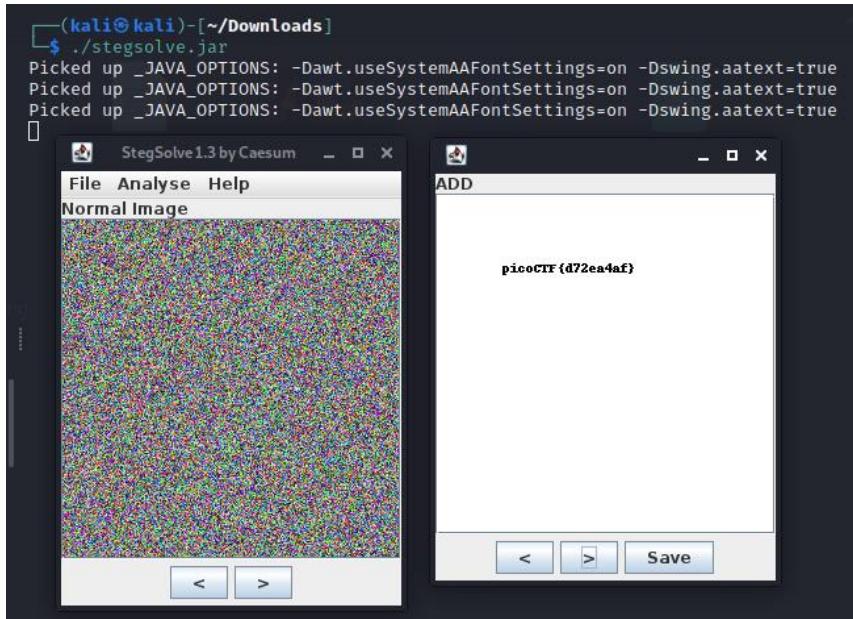
Kemudian sesuai hint soal yaitu visual cryptography saya mencoba menggunakan tools stegsolve untuk menggabungkan kedua gambar tersebut :

```
(kali㉿kali)-[~/Downloads]
$./stegsolve.jar
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[] 0043414082
 StegSolve1.3 by Caesum - x
File Analyse Help
scrambled2
< >
```

Pertama-tama, saya buka terlebih dahulu gambar 1 dan selanjutnya saya menggunakan tools analyze yaitu image combiner dan saya masukkan gambar 2 :

```
(kali㉿kali)-[~/Downloads]
$./stegsolve.jar
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[] 0043414082
 StegSolve1.3 by Caesum - x
File Analyse Help
Norm File Format Data Extract Stereogram Solver Frame Browser Image Combiner
< >
```

Setelah saya cek satu persatu algoritma nya, saya menemukan flag pada algoritma ADD :



**Flag = picoCTF{d72ea4af}**

## EASY1

Diberikan sebuah ciphertext dengan key-nya, lalu saya coba decrypt menggunakan vigenere cipher online:

Easy1

Tags: Category: Cryptography

AUTHOR: ALEX FULTON/DANNY

### Description

The one time pad can be cryptographically secure, but not when you know the key. Can you solve this? We've given you the encrypted flag, key, and a table to help UFJKXQZQUNB with the key of SOLVECRYPTO. Can you use this [table](#) to solve it?.

The screenshot shows a web-based Vigenère cipher tool. The 'Ciphertext' field contains 'UFJKXQZQUNB'. The 'KEY' field contains 'SOLVECRYPTO'. The 'Plaintext' field shows 'CRYPTOISFUN'. The tool interface includes sections for 'VIEW', 'ENCODE DECODE', 'Vigenère cipher', 'VARIANT Standard Vigenère cipher', 'KEY MODE Repeat', 'ALPHABET abcdefghijklmnopqrstuvwxyz', 'CASE STRATEGY Maintain case', and 'FOREIGN CHARS Include Ignore'. A note at the bottom says '→ Decoded 11 chars'.

**Flag = picoCTF{CRYPTOISFUN}**

## MIND YOUR Ps & Qs

Diberikan sebuah ciphertext RSA beserta key nya seperti berikut :

```
Decrypt my super sick RSA:
c: 861270243527190895777142537838333832920579264010533029282104230006461420086153423
n: 1311097532562595991877980619849724606784164430105441327897358800116889057763413423
e: 65537
```

Kemudian nilai n saya faktorkan di <http://factordb.com/> untuk mendapat nilai p dan q :

The screenshot shows the FactorDB website interface. At the top, there are navigation links: Search, Sequences, Report results, Factor tables, Status, Downloads, and Login. Below the search bar, the number 1311097532562595991877980619849724606784164430105441327897358800116889057763413423 is entered, and the 'Factorize!' button is highlighted. The result section shows the factorization: 1311097532562595991877980619849724606784164430105441327897358800116889057763413423 = 1955175890537890492055221842734816092141<40> · 670577792467509699665091201633524389157003<42>. There is also a 'More information' link.

Setelah didapat nilai p dan q nya saya membuat script seperti di bawah untuk melakukan decrypt terhadap ciphertext :

```
(kali㉿kali)-[~/Belajar/exercise/PICO/psqs]
$ cat RSA.py
#!/usr/bin/env python3

from Crypto.Util.number import inverse, long_to_bytes

c = 861270243527190895777142537838333832920579264010533029282104230006461420086153423
n = 1311097532562595991877980619849724606784164430105441327897358800116889057763413423
e = 65537
p = 1955175890537890492055221842734816092141
q = 670577792467509699665091201633524389157003

phi = (p-1)*(q-1)
d = inverse(e, phi)
m = pow(c,d,n)
print(long_to_bytes(m))
```

Selanjutnya saya jalankan script tersebut dan didapat flag nya :

```
(kali㉿kali)-[~/Belajar/exercise/PICO/psqs]
$./RSA.py
b'picoCTF{sma11_N_n0_g0od_13686679}'
```

Flag = picoCTF{sma11\_N\_n0\_g0od\_13686679}

## [FORENSIC]

### INFORMATION

Diberikan file berupa gambar, lalu saya coba strings dan ternyata muncul base64 :

```
(root💀 kali)-[~/home/kali/Downloads/picoCTF/Forensic]
└─# strings cat.jpg
JFIF
Photoshop 3.0
8BIM
PicoCTF
http://ns.adobe.com/xap/1.0/
<?xpacket begin='
 id='W5M0MpCehiHzreSzNTczkc9d'?>
<x:xmpmeta xmlns:x='adobe:ns:meta/' x:xmptk='Image :: ExifTool 10.80'>
<rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'\>
 <rdf:Description rdf:about=''
 xmlns:cc='http://creativecommons.org/ns#'\>
 <cc:license rdf:resource='cGljb0NURnt0aGVfbTN0YWRhdGfMXNfbW9kaWZpZWR9' />
 </rdf:Description>
 <rdf:Description rdf:about=''
 xmlns:dct='http://purl.org/dc/elements/1.1/'\>
 <dct:rights>
 <rdf:Alt>
 <rdf:li xml:lang='x-default'>PicoCTF</rdf:li>
 </rdf:Alt>
 </dct:rights>
 </rdf:Description>
</rdf:RDF>
</x:xmpmeta>
```

Lalu, saya decode base64 tersebut di terminal dan didapat flag :

```
(root💀 kali)-[~/home/kali/Downloads/picoCTF/Forensic]
└─# echo "cGljb0NURnt0aGVfbTN0YWRhdGfMXNfbW9kaWZpZWR9" | base64 -d
picoCTF{the_m3tadata_1s_modified}
```

Flag = picoCTF{the\_m3tadata\_1s\_modified}

### MATRYOSHKA DOLL

Diberikan file berupa gambar, lalu sesuai hint yaitu terdapat file yang disembunyikan di dalam image tersebut. Selanjutnya, saya menggunakan tools binwalk untuk mengesektrak file tersembunyi di dalam image tersebut :

```
(root💀 kali)-[~/home/kali/Downloads/picoCTF/Forensic]
└─# binwalk -e dolls.jpg
DECIMAL HEXADECIMAL DESCRIPTION

0 0x0 PNG image, 594 x 1104, 8-bit/color RGBA, non-interlaced
3226 0xC9A TIFF image data, big-endian, offset of first image directory: 8
272492 0x4286C Zip archive data, at least v2.0 to extract, compressed size: 378950, uncompressed size:
383938, name: base_images/2_c.jpg
651608 0x9F158 End of Zip archive, footer length: 22
```

Setelah itu, didapat folder yang didalamnya image lagi kemudian saya menggunakan binwalk kembali dan begitu seterusnya sampai saya mendapat file flag.txt :

```

[~]# ls -l
total 376
-rw-r--r-- 1 root root 379138 Oct 14 02:01 4286C.zip
drwxr-xr-x 2 root root 4096 Oct 14 02:01 base_images

[~]# cd base_images

[~]# ls -l
total 376
-rw-r--r-- 1 root root 383938 Mar 15 2021 2_c.jpg

[~]# binwalk -e 2_c.jpg

DECIMAL HEXADECIMAL DESCRIPTION

0 0x0 PNG image, 526 x 1106, 8-bit/color RGBA, non-interlaced
3226 0xC9A TIFF image data, big-endian, offset of first image directory: 8
187707 0x2DD3B Zip archive data, at least v2.0 to extract, compressed size: 196043, uncompressed size:
201445, name: base_images/3_c.jpg
383805 0x5DB3D End of Zip archive, footer length: 22
383916 0x5DBAC End of Zip archive, footer length: 22

[~]# ls
2_c.jpg _2_c.jpg.extracted

[~]# ls
136DA.zip flag.txt

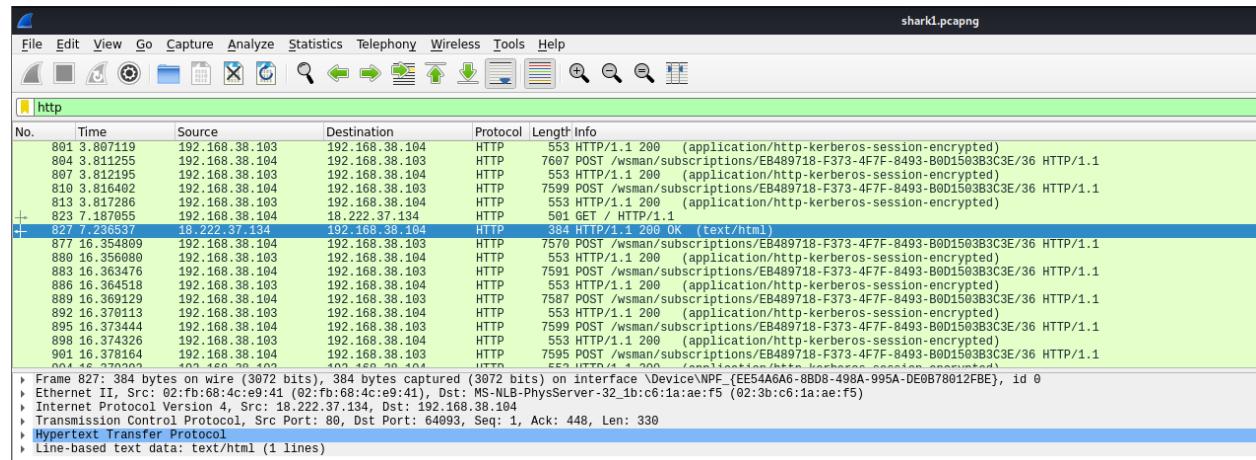
[~]# cat flag.txt
picoCTF{4f11048e83ffc7d342a15bd2309b47de}

```

Flag = picoCTF{4f11048e83ffc7d342a15bd2309b47de}

WIRESHARK DOO..... DOO... DOOO....

Diberikan sebuah file hasil capture packet, saya buka dan saya filter berdasarkan protocol HTTP :



Kemudian, di sini saya mendapat packet yang cukup menarik perhatian saya karena berisi text html. Selanjutnya saya menggunakan fiture follow HTTP stream dan muncul seperti berikut :

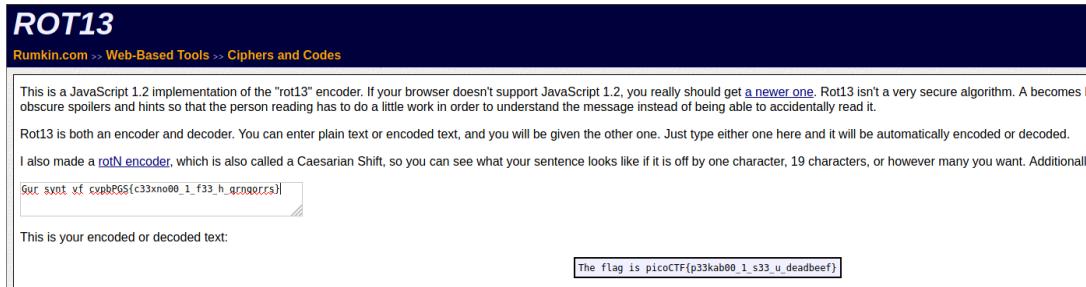


```
GET / HTTP/1.1
Host: 18.222.37.134
Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.105 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9

HTTP/1.1 200 OK
Date: Mon, 10 Aug 2020 01:51:45 GMT
Server: Apache/2.4.29 (Ubuntu)
Last-Modified: Fri, 07 Aug 2020 00:45:02 GMT
ETag: "2f-5ac3ee4fcfc01"
Accept-Ranges: bytes
Content-Length: 47
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

Gur synt xf cvpbPG{c33xno00_1_f33_h_qrnqorrs}
```

Di bagian akhir packet response terdapat kode yang saya asumsikan adalah ROT13. Lalu saya langsung decode dan didapat flagnya :



**ROT13**

Rumkin.com >> Web-Based Tools >> Ciphers and Codes

This is a JavaScript 1.2 implementation of the "rot13" encoder. If your browser doesn't support JavaScript 1.2, you really should get [a newer one](#). Rot13 isn't a very secure algorithm. A becomes N obscure spoilers and hints so that the person reading has to do a little work in order to understand the message instead of being able to accidentally read it.

Rot13 is both an encoder and decoder. You can enter plain text or encoded text, and you will be given the other one. Just type either one here and it will be automatically encoded or decoded.

I also made a [rotN encoder](#), which is also called a Caesarian Shift, so you can see what your sentence looks like if it is off by one character, 19 characters, or however many you want. Additionally,

Gur synt xf cvpbPG{c33xno00\_1\_f33\_h\_qrnqorrs}

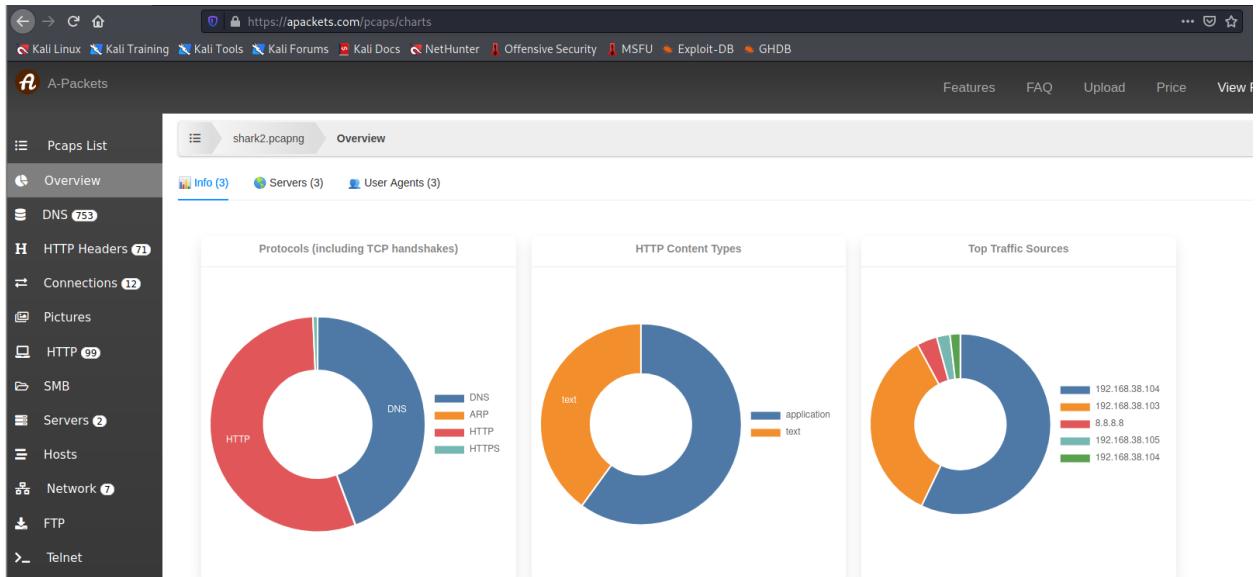
This is your encoded or decoded text:

The flag is picoCTF{p33kab00\_1\_s33\_u\_deadbeef}

Flag = **picoCTF{p33kab00\_1\_s33\_u\_deadbeef}**

WIRESHARK TWO... TWO... TWOOO...

Diberikan sebuah file capture packet, pertama saya buka menggunakan wireshark dan ternyata terdapat banyak sekali flag palsu. Di sini saya kebingungan cukup lama sampai saya menemukan tool analisis pcap online di <https://apackets.com/> :



Terlihat bahwa terdapat dua protocol yang dominan, yaitu HTTP dan DNS. Karena protocol HTTP sudah saya cek di wireshark dan ternyata tidak ada apa-apa, saya langsung melihat ke bagian protocol DNS dan saya menemukan trafik data ke sebuah alamat domain :

| 192.168.38.104 | 18.217.1.57 |                                                                       |
|----------------|-------------|-----------------------------------------------------------------------|
|                |             | cGjlb0NU.reddshrimpdherring.com                                       |
|                |             | cGjlb0NU.reddshrimpdherring.com.us-west-1.ec2-utilities.amazonaws.com |
|                |             | cGjlb0NU.reddshrimpdherring.com.windomain.local                       |
|                |             | RntkbmNf.reddshrimpdherring.com                                       |
|                |             | RntkbmNf.reddshrimpdherring.com.us-west-1.ec2-utilities.amazonaws.com |
|                |             | RntkbmNf.reddshrimpdherring.com.windomain.local                       |
|                |             | M3hmMWxf.reddshrimpdherring.com                                       |
|                |             | M3hmMWxf.reddshrimpdherring.com.us-west-1.ec2-utilities.amazonaws.com |
|                |             | M3hmMWxf.reddshrimpdherring.com.windomain.local                       |
|                |             | ZnR3X2Rl.reddshrimpdherring.com                                       |
|                |             | ZnR3X2Rl.reddshrimpdherring.com.us-west-1.ec2-utilities.amazonaws.com |
|                |             | ZnR3X2Rl.reddshrimpdherring.com.windomain.local                       |
|                |             | YWRiZWVm.reddshrimpdherring.com                                       |
|                |             | YWRiZWVm.reddshrimpdherring.com.us-west-1.ec2-utilities.amazonaws.com |
|                |             | YWRiZWVm.reddshrimpdherring.com.windomain.local                       |
|                |             | fQ==.reddshrimpdherring.com                                           |
|                |             | fQ==.reddshrimpdherring.com.us-west-1.ec2-utilities.amazonaws.com     |
|                |             | fQ==.reddshrimpdherring.com.windomain.local                           |

Lalu, saya tertarik pada setiap kata awal subdomain yang terlihat seperti base64 dan saya coba menggabungkannya dan dikonversi ternyata itu adalah flag :

**Text (ASCII / ANSI)**

```
picoCTF{dns_3xf1l_ftw_deadbeef}
```

**Convert** **Highlight Text**

**BASE64**

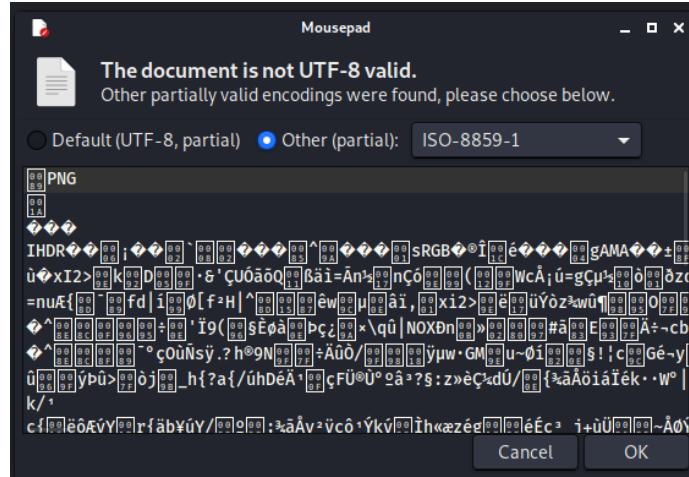
```
cGjb0NURntkbnNfM3hmMWxfZnR3X2RIYWRiZWVmfcQ
```

**Convert** **Highlight Text**

Flag = picoCTF{dns\_3xf1l\_ftw\_deadbeef}

## EXTENSIONS

Diberikan sebuah file tanpa ekstensi, kemudian setelah saya buka di bagian header terdapat kata PNG yang mengindikasikan file tersebut adalah file PNG :



Setelah itu, saya rename file tersebut dengan ekstensi png dan saya buka kembali dan muncul flag nya :

```
(kali㉿kali)-[~/Downloads/picoCTF/Forensic]
└─$ mv flag.txt flag.png

(kali㉿kali)-[~/Downloads/picoCTF/Forensic]
└─$ open flag.png ┌
```

picoCTF{now\_you\_know\_about\_extensions}

**Flag = picoCTF{now\_you\_know\_about\_extensions}**

LIKE1000

Diberikan sebuah file berekstensi tar, lalu saya membuat script python seperti berikut :

```
(kali㉿kali)-[~/Downloads/picoCTF/Forensic/like100]
└─$ cat extract.py
#!/usr/bin/env python3

import tarfile

for i in range(1000, 0, -1):
 my_tar = str(i) + '.tar'
 print ('[*]', my_tar)

 my_tar = tarfile.open(my_tar)
 my_tar.extractall('.')
 my_tar.close()
```

Script tersebut bertujuan untuk mengekstrak file tar sebanyak 1000 kali (sesuai hint pada nama soal) lalu saya jalankan dan didapat :

```
(kali㉿kali)-[~/Downloads/picoCTF/Forensic/like100]
└─$./extract.py
[*] 1000.tar
[*] 999.tar
[*] 998.tar
[*] 997.tar
[*] 996.tar
[*] 995.tar
[*] 994.tar
[*] 993.tar
[*] 992.tar
[*] 991.tar
[*] 990.tar
[*] 989.tar
[*] 988.tar
[*] 987.tar
[*] 986.tar
[*] 985.tar
[*] 984.tar
[*] 983.tar
[*] 982.tar
[*] 981.tar
[*] 980.tar
[*] 979.tar
[*] 978.tar
[*] 977.tar
[*] 976.tar
[*] 975.tar
```

```
(kali㉿kali)-[~/Downloads/picoCTF/Forensic/Like100]
$ ls
137.tar 174.tar 210.tar 240.tar 268.tar 301.tar 359.tar 396.tar 423.tar 46.tar 504.tar 543.tar 588.tar 627.tar 646.tar 651.tar 691.tar 728.tar 765.tar 802.tar 829.tar 878.tar 915.tar 94.tar 983.tar
138.tar 175.tar 211.tar 240.tar 269.tar 302.tar 355.tar 397.tar 427.tar 467.tar 505.tar 545.tar 587.tar 628.tar 645.tar 652.tar 692.tar 729.tar 766.tar 802.tar 831.tar 877.tar 913.tar 958.tar 988.tar
139.tar 176.tar 212.tar 241.tar 287.tar 322.tar 356.tar 398.tar 428.tar 468.tar 506.tar 546.tar 588.tar 629.tar 646.tar 653.tar 693.tar 697.tar 722.tar 767.tar 803.tar 849.tar 878.tar 914.tar 959.tar 989.tar
140.tar 177.tar 213.tar 242.tar 288.tar 323.tar 357.tar 399.tar 429.tar 469.tar 507.tar 547.tar 589.tar 630.tar 647.tar 654.tar 694.tar 698.tar 723.tar 768.tar 804.tar 850.tar 879.tar 915.tar 952.tar 989.tar
141.tar 178.tar 214.tar 243.tar 289.tar 324.tar 358.tar 400.tar 430.tar 470.tar 508.tar 548.tar 590.tar 631.tar 648.tar 655.tar 695.tar 699.tar 724.tar 769.tar 805.tar 851.tar 880.tar 916.tar 953.tar 989.tar
142.tar 179.tar 215.tar 244.tar 290.tar 325.tar 359.tar 401.tar 431.tar 471.tar 509.tar 549.tar 591.tar 632.tar 649.tar 656.tar 696.tar 700.tar 725.tar 770.tar 806.tar 852.tar 881.tar 917.tar 954.tar 989.tar
143.tar 180.tar 216.tar 245.tar 291.tar 326.tar 360.tar 402.tar 432.tar 472.tar 510.tar 550.tar 592.tar 633.tar 650.tar 657.tar 697.tar 701.tar 726.tar 771.tar 807.tar 853.tar 882.tar 918.tar 955.tar 990.tar
144.tar 181.tar 217.tar 246.tar 292.tar 327.tar 361.tar 403.tar 433.tar 473.tar 511.tar 551.tar 593.tar 634.tar 651.tar 658.tar 698.tar 702.tar 727.tar 772.tar 808.tar 854.tar 883.tar 919.tar 956.tar 991.tar
145.tar 182.tar 218.tar 247.tar 293.tar 328.tar 362.tar 404.tar 434.tar 474.tar 512.tar 552.tar 594.tar 635.tar 652.tar 659.tar 699.tar 703.tar 728.tar 773.tar 809.tar 855.tar 884.tar 920.tar 957.tar 992.tar
146.tar 183.tar 219.tar 248.tar 294.tar 329.tar 363.tar 405.tar 435.tar 475.tar 513.tar 553.tar 595.tar 636.tar 653.tar 660.tar 704.tar 704.tar 729.tar 774.tar 810.tar 860.tar 885.tar 921.tar 958.tar 993.tar
147.tar 184.tar 220.tar 249.tar 295.tar 330.tar 364.tar 406.tar 436.tar 476.tar 514.tar 554.tar 596.tar 637.tar 654.tar 661.tar 705.tar 705.tar 730.tar 775.tar 811.tar 861.tar 886.tar 922.tar 959.tar 997.tar
148.tar 185.tar 221.tar 250.tar 296.tar 331.tar 365.tar 407.tar 437.tar 477.tar 515.tar 555.tar 597.tar 638.tar 655.tar 662.tar 706.tar 706.tar 731.tar 776.tar 812.tar 862.tar 887.tar 923.tar 960.tar 998.tar
149.tar 186.tar 222.tar 251.tar 297.tar 332.tar 366.tar 408.tar 438.tar 478.tar 516.tar 556.tar 598.tar 639.tar 656.tar 663.tar 707.tar 707.tar 732.tar 777.tar 813.tar 863.tar 888.tar 924.tar 961.tar 999.tar
150.tar 187.tar 223.tar 252.tar 298.tar 333.tar 367.tar 409.tar 439.tar 479.tar 517.tar 557.tar 599.tar 640.tar 657.tar 664.tar 708.tar 708.tar 733.tar 778.tar 814.tar 864.tar 889.tar 925.tar 962.tar 999.tar
151.tar 188.tar 224.tar 253.tar 299.tar 334.tar 368.tar 410.tar 440.tar 480.tar 518.tar 558.tar 600.tar 641.tar 658.tar 665.tar 709.tar 709.tar 734.tar 779.tar 815.tar 865.tar 890.tar 926.tar 963.tar 999.tar
152.tar 189.tar 225.tar 254.tar 300.tar 335.tar 369.tar 411.tar 441.tar 481.tar 519.tar 559.tar 601.tar 642.tar 659.tar 666.tar 710.tar 710.tar 735.tar 780.tar 816.tar 866.tar 891.tar 927.tar 964.tar 999.tar
153.tar 190.tar 226.tar 255.tar 301.tar 336.tar 370.tar 412.tar 442.tar 482.tar 520.tar 560.tar 602.tar 643.tar 660.tar 667.tar 711.tar 711.tar 736.tar 781.tar 817.tar 867.tar 892.tar 928.tar 965.tar 999.tar
154.tar 191.tar 227.tar 256.tar 302.tar 337.tar 371.tar 413.tar 443.tar 483.tar 521.tar 561.tar 603.tar 644.tar 661.tar 668.tar 712.tar 712.tar 737.tar 782.tar 818.tar 868.tar 893.tar 929.tar 966.tar 999.tar
155.tar 192.tar 228.tar 303.tar 338.tar 372.tar 414.tar 444.tar 484.tar 522.tar 562.tar 604.tar 645.tar 662.tar 675.tar 713.tar 713.tar 738.tar 783.tar 819.tar 869.tar 894.tar 930.tar 970.tar 999.tar
156.tar 193.tar 229.tar 304.tar 339.tar 373.tar 415.tar 445.tar 485.tar 523.tar 563.tar 605.tar 646.tar 663.tar 676.tar 714.tar 714.tar 739.tar 784.tar 820.tar 870.tar 931.tar 971.tar 999.tar
157.tar 194.tar 230.tar 305.tar 340.tar 374.tar 416.tar 446.tar 486.tar 524.tar 564.tar 606.tar 647.tar 664.tar 677.tar 715.tar 715.tar 740.tar 785.tar 821.tar 871.tar 932.tar 972.tar 999.tar
158.tar 195.tar 231.tar 306.tar 341.tar 375.tar 417.tar 447.tar 487.tar 525.tar 565.tar 607.tar 648.tar 665.tar 678.tar 716.tar 716.tar 741.tar 786.tar 822.tar 872.tar 933.tar 973.tar 999.tar
159.tar 196.tar 232.tar 307.tar 342.tar 376.tar 418.tar 448.tar 488.tar 526.tar 566.tar 608.tar 649.tar 666.tar 679.tar 717.tar 717.tar 742.tar 787.tar 823.tar 873.tar 934.tar 974.tar 999.tar
160.tar 197.tar 233.tar 308.tar 343.tar 377.tar 419.tar 449.tar 489.tar 527.tar 567.tar 609.tar 650.tar 667.tar 680.tar 718.tar 718.tar 743.tar 788.tar 824.tar 874.tar 935.tar 975.tar 999.tar
161.tar 198.tar 234.tar 309.tar 344.tar 378.tar 420.tar 450.tar 490.tar 528.tar 568.tar 610.tar 651.tar 668.tar 681.tar 719.tar 719.tar 744.tar 789.tar 825.tar 875.tar 936.tar 976.tar 999.tar
162.tar 199.tar 235.tar 310.tar 345.tar 379.tar 421.tar 451.tar 491.tar 529.tar 569.tar 611.tar 652.tar 669.tar 682.tar 720.tar 720.tar 745.tar 790.tar 826.tar 876.tar 937.tar 977.tar 999.tar
163.tar 200.tar 236.tar 311.tar 346.tar 380.tar 422.tar 452.tar 492.tar 530.tar 570.tar 612.tar 653.tar 670.tar 683.tar 721.tar 721.tar 746.tar 791.tar 827.tar 877.tar 938.tar 978.tar 999.tar
164.tar 201.tar 237.tar 312.tar 347.tar 381.tar 423.tar 453.tar 493.tar 531.tar 571.tar 613.tar 654.tar 671.tar 684.tar 722.tar 722.tar 747.tar 792.tar 828.tar 878.tar 939.tar 979.tar 999.tar
165.tar 202.tar 238.tar 313.tar 348.tar 382.tar 424.tar 454.tar 494.tar 532.tar 572.tar 614.tar 655.tar 672.tar 685.tar 723.tar 723.tar 748.tar 793.tar 829.tar 879.tar 940.tar 980.tar 999.tar
166.tar 203.tar 239.tar 314.tar 349.tar 383.tar 425.tar 455.tar 495.tar 533.tar 573.tar 615.tar 656.tar 673.tar 686.tar 724.tar 724.tar 749.tar 794.tar 830.tar 880.tar 941.tar 981.tar 999.tar
167.tar 204.tar 240.tar 315.tar 350.tar 384.tar 426.tar 456.tar 496.tar 534.tar 574.tar 616.tar 657.tar 674.tar 687.tar 725.tar 725.tar 750.tar 795.tar 831.tar 881.tar 942.tar 982.tar 999.tar
168.tar 205.tar 241.tar 316.tar 351.tar 385.tar 427.tar 457.tar 497.tar 535.tar 575.tar 617.tar 658.tar 675.tar 688.tar 726.tar 726.tar 751.tar 796.tar 832.tar 882.tar 943.tar 983.tar 999.tar
169.tar 206.tar 242.tar 317.tar 352.tar 386.tar 428.tar 458.tar 498.tar 536.tar 576.tar 618.tar 659.tar 676.tar 689.tar 727.tar 727.tar 752.tar 797.tar 833.tar 883.tar 944.tar 984.tar 999.tar
170.tar 207.tar 243.tar 318.tar 353.tar 387.tar 429.tar 459.tar 499.tar 537.tar 577.tar 619.tar 660.tar 677.tar 690.tar 728.tar 728.tar 753.tar 798.tar 834.tar 884.tar 945.tar 985.tar 999.tar
171.tar 208.tar 244.tar 319.tar 354.tar 388.tar 430.tar 460.tar 500.tar 538.tar 578.tar 620.tar 661.tar 678.tar 691.tar 729.tar 729.tar 754.tar 800.tar 835.tar 885.tar 946.tar 986.tar 999.tar
172.tar 209.tar 245.tar 320.tar 355.tar 389.tar 431.tar 461.tar 501.tar 539.tar 579.tar 621.tar 662.tar 679.tar 692.tar 730.tar 730.tar 755.tar 801.tar 836.tar 886.tar 947.tar 987.tar 999.tar
173.tar 210.tar 347.tar 393.tar 431.tar 533.tar 565.tar 600.tar 633.tar 677.tar 701.tar 731.tar 761.tar 793.tar 822.tar 853.tar 887.tar 913.tar 954.tar 988.tar 999.tar
```

Kemudian, untuk mencari yang mana merupakan file berisi flag saya coba hapus file yang berekstensi tar dengan menggunakan command rm :

```
(kali㉿kali)-[~/Downloads/picoCTF/Forensic/like100]
$ rm -r *.tar"
```

Lalu, saya coba lihat lagi dan ternyata terdapat file flag yang berupa gambar :

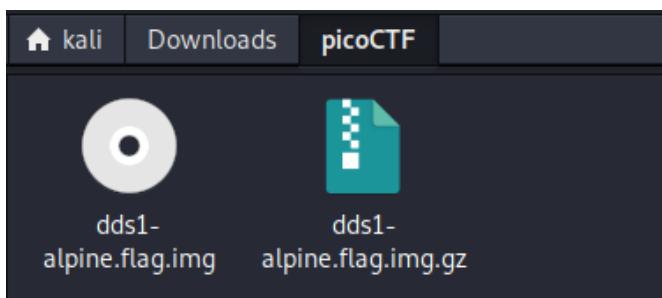
```
(kali㉿kali)-[~/Downloads/picoCTF/Forensic/like100]
$ ls -lah
total 60K
drwxr-xr-x 3 kali kali 28K Oct 14 23:05 .
drwxr-xr-x 4 kali kali 4.0K Oct 14 23:01 ..
drwxr-xr-x 2 kali kali 4.0K Oct 14 23:00 extract
-rw-rw-rw- 1 kali kali 192 Oct 14 23:00 extract.py
-rw-r--r-- 1 kali kali 27 Aug 4 2019 filler.txt
-rw-rw-rw- 1 kali kali 13K Aug 4 2019 flag.png
```

picoCTF{l0t5\_0f\_TAR5}

Flag = picoCTF{l0t5\_0f\_TAR5}

DISK, DISK, SLEUTH!

Diberikan sebuah file zip saya coba ekstrak dan file tersebut adalah sebuah file iso/image disk :



Kemudian, saya melakukan strings terhadap file tersebut dan saya grep sesuai format flag :

```
[root@kali]# strings dds1-alpine.flag.img | grep pico
ffffffff81399ccf t pirq_pico_get
ffffffff81399cee t pirq_pico_set
ffffffff820adb46 t pico_router_probe
SAY picoCTF{f0r3ns1c4t0r_n30phyt3_a6f4cab5}
```

Flag = picoCTF{f0r3ns1c4t0r\_n30phyt3\_a6f4cab5}

DISK, DISK, SLEUTH! II

Diberikan sebuah file disk image, lalu saya download dan soal memberikan hint berupa link berikut [http://wiki.sleuthkit.org/index.php?title=TSK\\_Tool\\_Overview](http://wiki.sleuthkit.org/index.php?title=TSK_Tool_Overview). Setelah saya pahami saya menemukan command mmls yang berfungsi untuk menampilkan layout/partisi dari sebuah file disk images :

### Volume System Tools

These tools take a disk (or other media) image as input and analyze its partition structures. Examples include DOS partitions, BSD disk labels, and the Sun Volume Table of Contents (VTOC). These can be used find hidden data between partitions and to identify the file system offset for The Sleuth Kit tools. The media management tools support DOS partitions, BSD disk labels, Sun VTOC, and Mac partitions.

- **mmls**: Displays the layout of a disk, including the unallocated spaces.
- **mmstat**: Display details about a volume system (typically only the type).
- **mmcatt**: Extracts the contents of a specific volume to STDOUT.

Setelah saya jalankan saya mendapat sebuah partisi yang dialokasikan untuk Linux :

```
[kali㉿kali]# mmls dds2-alpine.flag.img
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

 Slot Start End Length Description
 000: Meta 0000000000 0000000000 0000000001 Primary Table (#0)
 001: _____ 0000000000 0000002047 0000002048 Unallocated
 002: 000:000 0000002048 0000262143 0000260096 Linux (0x83)
```

Kemudian untuk melihat apa saja yang terdapat di dalam partisi tersebut saya menggunakan command fls untuk melihat list file :

### File Name Layer Tools

These file system tools process the file name structures, which are typically located in the parent directory.

- **ffind**: Finds allocated and unallocated file names that point to a given meta data structure.
- **fls**: Lists allocated and deleted file names in a directory.

```
(kali㉿kali)-[~/Downloads/picoCTF/Forensic/Disk 2]
$ fls -o 2048 dds2-alpine.flag.img
d/d 26417: home
d/d 11: lost+found
r/r 12: .dockerenv
d/d 20321: bin
d/d 4065: boot
d/d 6097: dev
d/d 2033: etc
d/d 8129: lib
d/d 14225: media
d/d 16257: mnt
d/d 18289: opt
d/d 16258: proc
d/d 18290: root
d/d 16259: run
d/d 18292: sbin
d/d 12222: srv
d/d 16260: sys
d/d 18369: tmp
d/d 12223: usr
d/d 14229: var
V/V 32513: $OrphanFiles
```

Setelah itu, terdapat beberapa direktori dan di sini saya tertarik pada direktori root karena berpotensi mengandung flag. Selanjutnya untuk melihat apa saja isi dari direktori root tersebut saya menggunakan command fls lagi seperti gambar di bawah :

```
(kali㉿kali)-[~/Downloads/picoCTF/Forensic/Disk 2]
$ fls -o 2048 dds2-alpine.flag.img 18290
r/r 18291: down-at-the-bottom.txt
```

Lalu saya menemukan file txt dan saya menggunakan command icat untuk membaca isi file tersebut :

#### Meta Data Layer Tools

These file system tools process the meta data structures, which store the details about a file. Examples of this structure include directory entries in FAT, MFT entries in NTFS, and inodes in ExtX and UFS.

- **icat:** Extracts the data units of a file, which is specified by its meta data address (instead of the file name).
- **ifind:** Finds the meta data structure that has a given file name pointing to it or the meta data structure that points to a given data unit.
- **ils:** Lists the meta data structures and their contents in a pipe delimited format.
- **istat:** Displays the statistics and details about a given meta data structure in an easy to read format.

```
(kali㉿kali)-[~/Downloads/picoCTF/Forensic/Disk 2]
$ icat -o 2048 dds2-alpine.flag.img 18291
(p) (i) (c) (o) (c) (T) (F) ({) (f) (0) (r) (3) (n)
(s) (1) (c) (4) (t) (0) (r) (_) (n) (0) (v) (1) (c)
(3) (f) (5) (5) (6) (5) (e) (7) (b) (}
```

**Flag = picoCTF{f0r3ns1c4t0r\_n0v1c3\_f5565e7b}**

## WHAT LIES WITHIN

Diberikan sebuah file berupa gambar berikut :



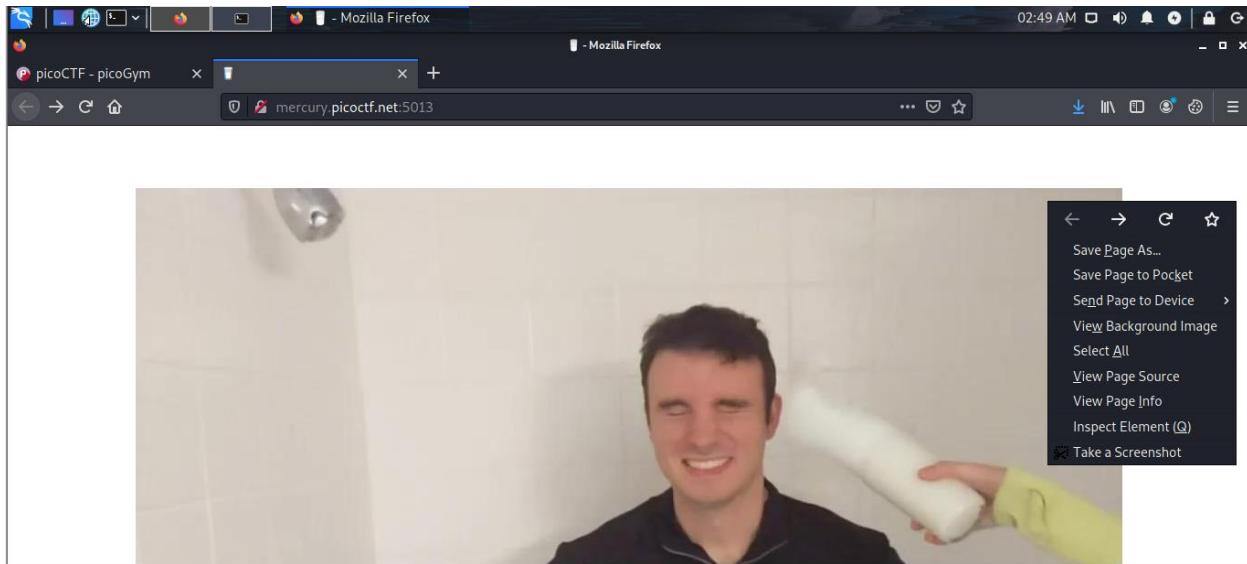
Lalu, sesuai hint nama soal terdapat flag yang tersembunyi di dalam gambar, dan saya coba menggunakan tool steghide namun perlu passphrase untuk mengekstraknya. Kemudian saya mencoba lagi menggunakan tools zsteg dan saya menemukan flag :

```
(kali㉿kali)-[~/Downloads/picoCTF]
└─$ zsteg buildings.png
b1,r,lsb,xy .. text: "^\$>R5YZrG"
b1,rgb,lsb,xy .. text: "picoCTF{h1d1ng_1n_th3_b1t5}"
b1,abgr,msb,xy .. file: PGP Secret Sub-key -
b2,b,lsb,xy .. text: "XuH}p#8Iy="
b3,abgr,msb,xy .. text: "t@Wp-_tH_v\r"
b4,r,lsb,xy .. text: "fdD\"\\\" "
b4,r,msb,xy .. text: "%Q#gpSv0c05"
b4,g,lsb,xy .. text: "fDfffDD\""
b4,g,msb,xy .. text: "f\"fff\"\\\"DD"
b4,b,lsb,xy .. text: "\$BDDDDf"
b4,b,msb,xy .. text: "wwBDDDFUU53w"
b4,rgb,msb,xy .. text: "dUcv%F#A`"
b4,bgr,msb,xy .. text: " V\"c7Ga4"
b4,abgr,msb,xy .. text: "gOC_$_@o"
```

Flag = picoCTF{h1d1ng\_1n\_th3\_b1t5}

## MILKSLAP

Diberikan sebuah website dengan sebuah gif sebagai tampilannya :



Lalu, saya coba simpan file gif tersebut ke local dan saya menggunakan zsteg untuk menemukan flagnya :

```
(kali㉿kali)-[~/Downloads/picoCTF]
└─$ zsteg concat v.png
imagedata .. text: "\n\n\n\n\n\n\t\t"
b1,b,lsb,xy .. text: "picoCTF{imag3_m4n1pul4t10n_sl4p5}\n"
b1,bgr,lsb,xy .. <wbStego size=9706075, data="\xB6\xAD\xB6}\xDB\xB2lR\x7F\xD
F\x86\xB7c\xFC\xFF\xBF\x02Zr\x8E\xE2Z\x12\xD8q\xE5&MJ-X:\xB5\xBF\xF7\x7F\xDB\xDFI\x
bm\xDB\xDB\x80m\x00\x00\x00\xB6m\xDB\xDB\xB6\x00\x00\x00\xB6\xB6\x00m\xDB\x12\x12m\x
DB\xDB\x00\x00\x00\x00\x00\xB6m\xDB\x00\xB6\x00\x00\xDB\xB6mm\xDB\xB6\xB6\x0
0\x00\x00\x00\x00m\xDB", even=true, mix=true, controlbyte="["
b2,r,lsb,xy .. file: SoftQuad DESC or font file binary
b2,r,msb,xy .. file: VISX image file
b2,g,lsb,xy .. file: VISX image file
b2,g,msb,xy .. file: SoftQuad DESC or font file binary - version 15722
b2,b,msb,xy .. text: "UFUUUU@UUU"
b4,r,lsb,xy .. text: "\\"\\\"#4D"
b4,r,msb,xy .. text: "www3333"
b4,g,lsb,xy .. text: "wwwwwvUS"
b4,g,msb,xy .. text: "\\"\\\"DDDD"
b4,b,lsb,xy .. text: "vdUeVweDFw"
b4,b,msb,xy .. text: "UUYYUUUUUUUU"
```

Flag = picoCTF{picoCTF{imag3\_m4n1pul4t10n\_sl4p5}}

## SO META

Diberikan sebuah file gambar, saya menggunakan exiftool untuk melihat informasi gambar :

```
(kali㉿kali)-[~/Downloads/picoCTF]
$ exiftool pico_img.png
ExifTool Version Number : 12.30
File Name : pico_img.png
Directory : .
File Size : 106 KiB
File Modification Date/Time : 2021:10:17 00:21:21-04:00
File Access Date/Time : 2021:10:17 00:21:55-04:00
File Inode Change Date/Time : 2021:10:17 00:21:55-04:00
File Permissions : -rw-r--r--
File Type : PNG
File Type Extension : png
MIME Type : image/png
Image Width : 600
Image Height : 600
Bit Depth : 8
Color Type : RGB
Compression : Deflate/Inflate
Filter :
Interlace :
Software : Adobe ImageReady
XMP Toolkit : Adobe XMP Core 5.3-c011 66.145661, 2012/02/06-14:56:27
Creator Tool : Adobe Photoshop CS6 (Windows)
Instance ID : xmp.iid:A5566E73B2B811E8BC7F9A4303DF1F9B
Document ID : xmp.did:A5566E74B2B811E8BC7F9A4303DF1F9B
Derived From Instance ID : xmp.iid:A5566E71B2B811E8BC7F9A4303DF1F9B
Derived From Document ID : xmp.did:A5566E72B2B811E8BC7F9A4303DF1F9B
Warning : [minor] Text/EXIF chunk(s) found after PNG IDAT (may be ignored by some readers)
Artist : picoCTF{s0_m3ta_fec06741}
Image Size : 600x600
Megapixels : 0.360
```

Flag = picoCTF{s0\_m3ta\_fec06741}

## SHARK ON WIRE 1

Diberikan sebuah file capture packet, lalu saya buka menggunakan wireshark, Kemudian sesuai hint soal yaitu "what are streams?" di sini saya menyimpulkan bahwa kita harus menggunakan fitur follow stream untuk melihat flag nya. Lalu, saya coba cek satu persatu protocol yang bisa di follow stream nyadan saya menemukan protocol UDP pada packet no 63 yang berisi flag :

Wireshark screenshot showing a list of network packets. The packet at index 63 is highlighted, showing it's a UDP packet from port 10.0.0.2 to 10.0.0.12, length 8888, containing the flag payload. The details pane shows the hex and ASCII representation of the packet.

Frame 63: 68 bytes on wire (480 bits), 68 bytes captured (480 bits)  
> Ethernet II, Src: VMware\_b9:02:a9 (00:0c:29:b9:02:a9), Dst: Broadcast (ff:ff:ff:ff:ff:ff)  
> Internet Protocol Version 4, Src: 10.0.0.2, Dst: 10.0.0.12  
> User Datagram Protocol, Src Port: 5000, Dst Port: 8888  
> Data (1 byte)

0000 ff ff ff ff ff ff 00 0c 29 b9 02 a9 08 00 45 00 .....E.  
0010 00 1d 00 01 00 00 48 11 66 c2 0a 00 00 02 0a 00 .....@. f.....

Wireshark · Follow UDP Stream (udp.stream eq 6) · capture.pcap

picoCTF{StaT31355\_636f6e6e}

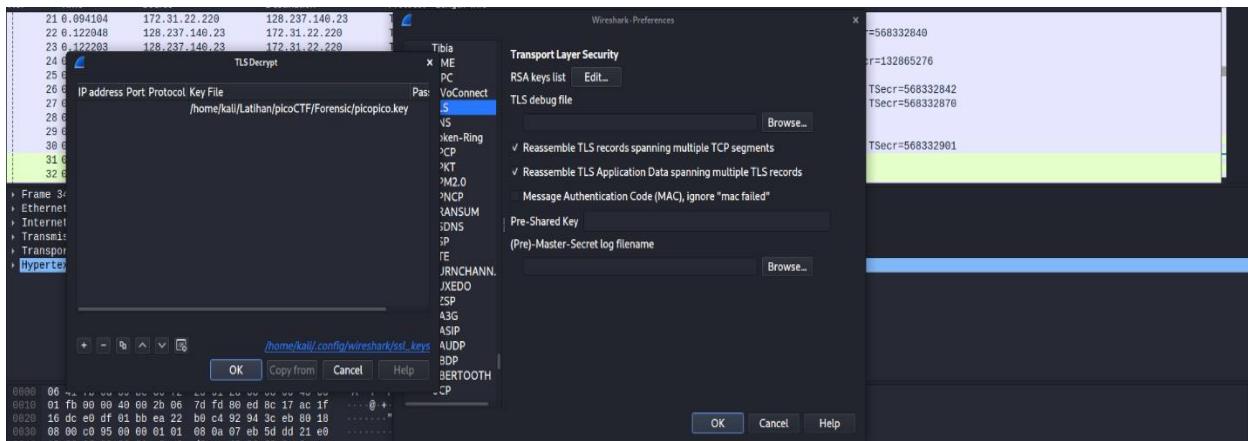
Flag = picoCTF{StaT31355\_636f6e6e}

## WEBNETO

Diberikan sebuah file packet capture dan key seperti berikut :

```
(kali㉿kali)-[~/Latihan/picoCTF/Forensic]
$ cat picopico.key
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwgSjAgEAAoIBAQCwKlFPNKjseJF5
puCJu5x38XcT1eQge5zOKNahAlYudvGOEs61TnIgvcER4ko8i3OCwak2/atcGk3
oz9jFKep7XFENP31IwwD9j/YazLKy4DRLGObOyIZUU1f2WRA7Uhf0POQXsDT1oU
X32jMKZkQSSDW4MRZd9trJYd02TrcEPMsBzQlFlvgnNwl3QlawozTHLAJKI36j1
cPwSMMeNca1e0Zi1s7R5IxhfpNXOBF0FmxiwvmeOHbaspyHg8UEmGBrk4k4wXSK
GQvrc8QjycP4ScEdquxJiYnDT8iEbAq70/7f/5NIN1DE9YoGJqKYjTS9nRPB4Yvj
JN/SJnhvAgMBAAEcggEACCnd3LrG/TZVH3sR0qvq01CwQPYPfUXdLVyNHab7EWon
pc+XBOHurJENG2CpRYF7h+nQ5ADhfIYSCicBf/jsEB7VueJ20CxEVtHVL3h6R6Bp
oHMLe0Em80cofuMpdl/k0+om3T8BkVsZCvCl5NMTUuAF7iRmfX7oDLALwM0IzzQv
2un+2UmT15rgAZfl3IL1PGvJhbhLxfeeyPE9MBy1SqBjQ9rNFn8sQv959J6BH24b
EpK//ErtNP2yh7oiVBgKEQ1gEuojQC/4oxoqCFFzaf9XNRCxB/zY1nUpervJyz09
NMQWNF2EmvmbVGFoTxmuut5N0GbVr2UyHxWMkm2s0QKBgQDpb2+AWgWlGtetuLKJ
fJ8s8nd6LhnafbKC0XMOT68qMBRoTpBtVTLRVSNvWCm8m4TTEazX4+ZA+bJFwUFw
aATDmHcr6lMI3tNKrcsnY2F7o5I4z6mwuRuSeSzq/ndxzqCzwCu4nKixh3cznp7j
JiElNG0d8Lu5eQgmVAK1AhWXfQKBgQDBMa9ga7VJUP4pzChnWAoi340pfjvQYeGl
IKL3AK040edaHdH9qid41PQHnL703xzN669SkLZ5s0d88A/LFLk4oZNMKdkSTQIQ
+AMbXH01HGFnCOuPg/FbNp1wS7zJEG5u5HFQWMPNJLr/hZ6g2Yp+UGpAcGTwM/
RCPVAPhLWwKBgQDAB0OaOnPaVjKGxiHAqBirrGiswa/S5QrzEaxxys5cUPYaoi0
6BldysPTnJr45JZna2rcTkXjvYTbjTDf3zHMFWgzYBfefC8kh8NPK5nNs8ldorbd
AemEnjBkP+DSELKyK6vLulOrdtzAQgRCp+MsT+xTbO2AreX826SXSpoQKBgC2v
nDOHBQXje1dTawLUToFurgQ8AwLOYEdKKyUoCLOvqEW8D02a0MtyM+MB6tQI7Wm
iH1T73L0LHgkL3bw3aRawV5/fu/0+jAdFk8AHjPTFE+acu2fi4c6aKb0GjAxYksU
yjIFeK/pKinr4SESMkjpw0WowGiDgtcRPBAA/LaFAoGAFEM1rfM0v3UmB7PS6u0m
P3ckP2FCFdaryXPfc52GBcJ3Q46YpsQvLTvotM+teHvTjNw2jwwZxIl4NenGSEj3
KdhQoOiqC9BrDD+DB4I9+T9nxT3g7R6MrgITghB4We7TVhL/PljnJTyDqpjNA4kY
TveAJPv6Xq1ERT5PUTX3BqQ=
-----END PRIVATE KEY-----
```

Kemudian, hint soal mengatakan untuk decrypt TLS stream, lalu saya coba mendecrypt-nya menggunakan key tadi, dengan cara membukanya lewat wireshark dan pergi ke tab Edit > Preference > Protocols > TLS > Edit RSA key list, dan masukkan lokasi file key berada :



Setelah itu, saya lihat lagi ke packet protocol TLS dan ternyata di bagian header HTTP-nya terdapat flag yang dicari :

```

GET / HTTP/1.1
Host: ec2-18-223-184-200.us-east-2.compute.amazonaws.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Pragma: no-cache
Cache-Control: no-cache

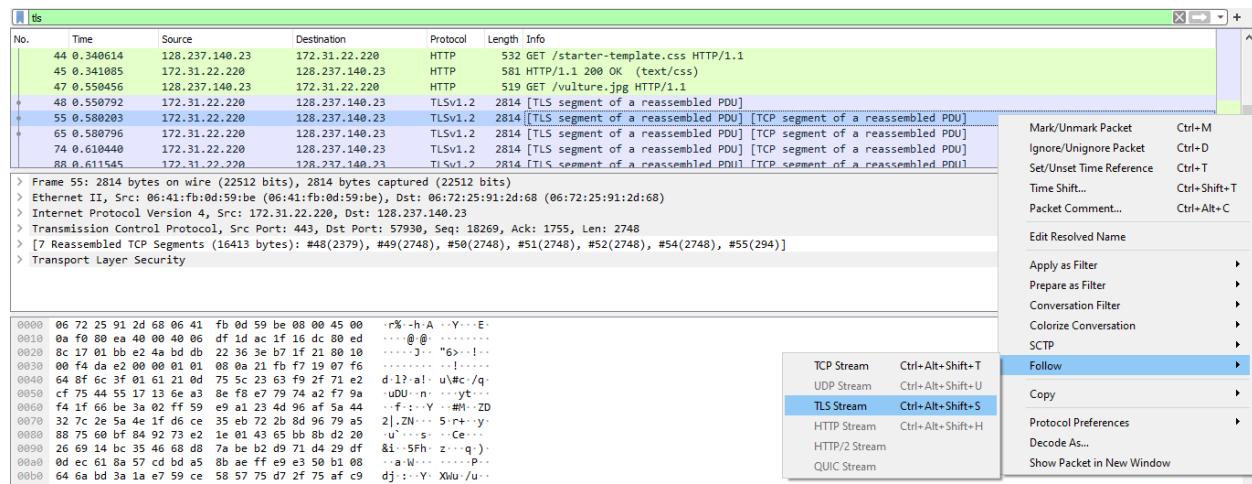
HTTP/1.1 200 OK
Date: Fri, 23 Aug 2019 15:56:36 GMT
Server: Apache/2.4.29 (Ubuntu)
Last-Modified: Mon, 12 Aug 2019 16:50:05 GMT
ETag: "5ff-58fee50dc3fb0-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Pico-Flag: picoCTF{nongshim.shrimp.crackers}
Content-Length: 821
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

```

**Flag = picoCTF{nongshim.shrimp.crackers}**

## WEBNET1

Diberikan sebuah file capture packet, lalu di sini saya agak kebingungan cukup lama untuk memahami hint yang diberikan soal, karena hint yang diberikan sama seperti soal webnet0 sebelumnya. Sampai akhirnya saya mencoba untuk memfilter packet berdasarkan protocol TLS, karena jumlahnya sedikit saya cek satu persatu dan saya menemukan flag pada packet no 55 :



Wireshark · Follow TLS Stream (tcp.stream eq 0) · webnet1.pcap

Referer: https://ec2-18-223-184-200.us-east-2.compute.amazonaws.com/second.html  
Pragma: no-cache  
Cache-Control: no-cache

HTTP/1.1 200 OK  
Date: Fri, 23 Aug 2019 16:27:04 GMT  
Server: Apache/2.4.29 (Ubuntu)  
Last-Modified: Fri, 23 Aug 2019 16:26:33 GMT  
ETag: "112fb-590cb44f2cbe6"  
Accept-Ranges: bytes  
Content-Length: 70395  
Pico-Flag: picoCTF{this.is.not.your.flag.anymore}  
Keep-Alive: timeout=5, max=99  
Connection: Keep-Alive  
Content-Type: image/jpeg

.....JFFI.....Exif..MM.\*.....J.....R.  
(.....;.....Z.....picoCTF{honey.roasted.peanuts}.....ICC\_PROFILE.....lcms...mntrRG  
B XYZ .....).9acspAPPL.....lcms.....  
desc.....^cptr...  
\....wtpt..h.....bkpt.....|.rXYZ.....gXYZ.....bXYZ.....rTRC.....@gTRC.....@bTRC.....@desc.....c2.....  
3....XYZ .....o.....8....XYZ .....b.....XYZ .....\$.curv.....c.....k.....?Q.4!).  
2.;.F.Qw].kpz.....| i }.....0.....C.....  
...  
'  
.....) .. )/'%'/9339GDG]]}.....C.....  
...  
'  
.....) .. )/'%'/9339GDG]]}....."  
.....}.....!1A.Qa."q.2....#B...R...\$.3br.  
%&'(\*456789:CDEFIGHIJSTUVWXYZCdefghijstuvwxyz.....  
.....W.....!1.AQ.aq."2...B..... #3R..br.  
\$.4.

**Flag = picoCTF{honey.roasted.peanuts}**

## CORRUPT

Diberikan sebuah file yang tidak dapat dibaca (corrupt), kemudian sesuai hint soal diberitahu bahwa "try fixing the file header" lalu langsung saja saya buka menggunakan hexeditor di terminal :

| File: mystery                                            | ASCII Offset: 0x00000000 / 0x000318BB (%0) |
|----------------------------------------------------------|--------------------------------------------|
| 00000000 89 65 4E 34 0D 0A B0 AA 00 00 00 0D 43 22 44 52 | .eN4.....C%DR                              |
| 00000010 00 00 06 6A 00 00 04 47 08 02 00 00 00 7C 8B AB | ...j ...G....  ...                         |
| 00000020 78 00 00 00 01 73 52 47 42 00 AE CE 1C E9 00 00 | X....sRGB.....                             |
| 00000030 00 04 67 41 4D 41 00 00 B1 8F 0B FC 61 05 00 00 | ..gAMA.....%I                              |
| 00000040 00 09 70 48 59 73 AA 00 16 25 00 00 16 25 01 49 | ..PHYS....%....%I                          |
| 00000050 52 24 F0 AA AA FF FA AB 44 45 54 78 5E EC BD 3F | R\$.....DET%?A                             |
| 00000060 00 E4 C9 7A D0 21 05 30 20 20 00 f3 02 00 00 00 | d.....A                                    |

Terlihat bahwa header file tersebut diawali dengan hexa 89 dan saya berasumsi file tersebut adalah file PNG (reference : [https://www.garykessler.net/library/file\\_sigs.html](https://www.garykessler.net/library/file_sigs.html)).

89 50 4E 47 0D 0A 1A 0A %PNG....  
PNG [Portable Network Graphics file](#)  
**Trailer:** 49 45 4E 44 AE 42 60 82 (TEND@B,...)

Kemudian, untuk memperbaikinya saya mengunduh file png dari internet dan saya bandingkan header-nya dengan file corrupt tadi. Pertama saya perbaiki header-nya menjadi PNG dan IHDR :

| File: file.png                                           | ASCII OFFSET: 0x00000000 / 0x00003E5C (END) | File: mystery                                            | ASCII OFFSET: 0x00000000 / 0x00001A90 (END) |
|----------------------------------------------------------|---------------------------------------------|----------------------------------------------------------|---------------------------------------------|
| 00000000 49 50 4E 47 00 0A 1A 04 00 00 00 0D 49 48 44 52 | .PNG.....IHDR                               | 00000000 89 50 4E 47 00 0A 0B AA 00 00 00 0D 49 48 44 52 | PNG.....IHDR                                |
| 00000010 00 00 03 98 00 00 02 00 04 03 00 00 00 15 A4 DE | .....                                       | 00000010 00 00 06 6A 00 00 04 47 08 02 00 00 00 7C 8B AB | ...G...                                     |

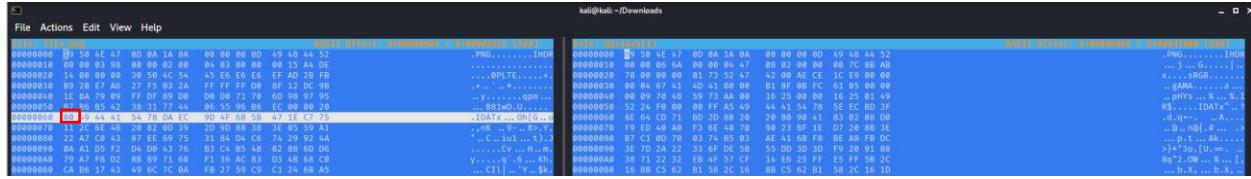
Setelah itu, saya coba buka filenya dan ternyata masih corrupt. Kemudian saya coba cek lagi ternyata masih terdapat kesalahan yaitu di bagian berikut :

```

File: mystery
ASCII Offset: 0x000000050 / 0x000318BB (%0)
00000000 89 65 4E 34 0D 0A B0 AA 00 00 00 00 43 22 44 52
00000010 00 00 06 6A 00 00 04 47 08 02 00 00 00 7C 8B AB
00000020 78 00 00 00 01 73 52 47 42 00 AE CE 1C E9 00 00
00000030 00 04 67 41 4D 41 00 00 B1 8F 0B FC 61 05 00 00
00000040 00 09 70 48 59 73 AA 00 16 25 00 00 16 25 01 49
00000050 52 24 F0 AA AA FF A5 AB 44 45 54 78 5E EC BD 3F
00000060 8E 64 CD 71 BD 2D 8B 20 20 80 90 41 83 02 08 D0
00000070 F9 ED 40 A0 F3 6E 40 7B 90 23 8F 1E D7 20 8B 3E
00000080 B7 C1 0D 70 03 74 B5 03 AE 41 6B F8 BE A8 FB DC
 .eN4.....C"DR
 ...j...G.....|..
 x....sRGB.....
 ..gAMA.....a...
 ..pHYs....%...%.I
 R$....DETx^..?
 .d.q--..A...
 ..@..n@{.,#...>.
 ..p.t...Ak....

```

Lalu saya perbaiki menjadi IDAT (Image Data) sesuai seperti file png yang saya unduh dari internet. Kemudian saya save lagi dan ternyata masih ada kesalahan. Lalu saya menemukan bahwa di depan kata IDAT tersebut hexa yang benar harus bernilai 0 seperti pada gambar :



Selanjutnya, saya ubah hexa yang di file corrupt tersebut di depan kata IDAT tadi menjadi 00 dan file berhasil dibuka :



Flag = picoCTF{c0rrupt10n\_1847995}

## WHITEPAGES

Diberikan sebuah file berupa halaman kosong lalu saya coba lihat menggunakan text editor online dan ternyata terdapat dua warna spasi :



Lalu, saya coba replace spasi warna merah menjadi angka biner 0 dan spasi warna putih menjadi angka biner 1 :



Selanjutnya, saya konversi bilangan biner tersebut ke ascii dan didapat flag :

**Flag = picoCTF{not\_all\_spaces\_are\_created\_equal\_c54f27cd05c2189f8147cc6f5deb2e56}**

# MACROHARD WEAKEDGE

Diberikan sebuah file PowerPoint, kemudian saya melakukan strings dan ternyata di akhir strings terdapat sebuah file hidden yang saya asumsikan berisi flag :

ppt/slidesLayouts/slidesLayout5.xmlPK  
ppt/slidesLayouts/slidesLayout6.xmlPK  
ppt/slidesLayouts/slidesLayout7.xmlPK  
ppt/slidesLayouts/slidesLayout8.xmlPK  
ppt/slidesLayouts/slidesLayout9.xmlPK  
ppt/slidesLayouts/slidesLayout10.xmlPK  
ppt/slidesLayouts/slidesLayout11.xmlPK  
ppt/slidesMasters/\_rels/slidesMaster1.xml.relsPK  
ppt/slidesLayouts/\_rels/slidesLayout1.xml.relsPK  
ppt/slidesLayouts/\_rels/slidesLayout2.xml.relsPK  
ppt/slidesLayouts/\_rels/slidesLayout3.xml.relsPK  
ppt/slidesLayouts/\_rels/slidesLayout4.xml.relsPK  
ppt/slidesLayouts/\_rels/slidesLayout5.xml.relsPK  
ppt/slidesLayouts/\_rels/slidesLayout6.xml.relsPK  
ppt/slidesLayouts/\_rels/slidesLayout7.xml.relsPK  
ppt/slidesLayouts/\_rels/slidesLayout8.xml.relsPK  
ppt/slidesLayouts/\_rels/slidesLayout9.xml.relsPK  
ppt/slidesLayouts/\_rels/slidesLayout10.xml.relsPK  
ppt/slidesLayouts/\_rels/slidesLayout11.xml.relsPK  
ppt/theme/theme1.xmlPK  
docProps/thumbnail.jpegPK  
ppt/vbaProject.binPK  
ppt/presProps.xmlPK  
ppt/viewProps.xmlPK  
ppt/tableStyles.xmlPK  
docProps/core.xmlPK  
docProps/app.xmlPK  
sWQON  
ppt/slidesMasters/hiddenPK

Kemudian saya mencoba menggunakan tools zsteg dan binwalk namun tidak berhasil. Selanjutnya saya mencoba meng-unzip file tersebut dan ternyata berhasil dilakukan :

```
[root@kali]~[/home/.../Downloads/picoCTF/Forensic/MacroHard WeakEdge]
unzip Forensics\ is_fun.pptm
Archive: Forensics is fun.pptm
```

Lalu sesuai hint yang saya dapat di awal, saya masuk ke direktori ppt/slidesMasters dan benar saja di sana terdapat file text yang berisi base64 :

```
└──(root💀 kali)-[~/home/.../Downloads/picoCTF/Forensic/MacroHard WeakEdge]
 └──# cd ppt/slidesMasters
 └──(root💀 kali)-[/home/.../Forensic/MacroHard WeakEdge/ppt/slidesMasters]
 └──# ls
 hidden _rels slideMaster1.xml
 Network
 └──(root💀 kali)-[/home/.../Forensic/MacroHard WeakEdge/ppt/slidesMasters]
 └──# cat hidden
Z m x h Z z o g c G l j b 0 N U R n t E M W R f d V 9 r b j B 3 X 3 B w d H N f c
l 9 6 M X A 1 f Q
```

Base64 tersebut saya konversi ke ascii dan didapat flag nya :

The screenshot shows a web application interface for hex and ASCII conversion. On the left, under 'Text (ASCII / ANSI)', the ASCII representation of the base64 string is shown: 'flag: picoCTF{D1d\_u\_kn0w\_ppts\_r\_z1p5}'. Below it are 'Convert' and 'Highlight Text' buttons. On the right, under 'Hexadecimal', the raw hex bytes are listed: '66 6c 61 67 3a 20 70 69 63 6f 43 54 46 7b 44 31 64 5f 75 5f 6b 6e 30 77 5f 70 70 74 73 5f 72 5f 7a 31 70 35 7d'. Under 'BASE64', the converted ASCII string is displayed again: 'Z m x h Z z o g c G l j b 0 N U R n t E M W R f d V 9 r b j B 3 X 3 B w d H N f c l 9 6 M X A 1 f Q'. There is also a 'MathType: Install now' advertisement overlay.

Flag = picoCTF{D1d\_u\_kn0w\_ppts\_r\_z1p5}

TUNN3L V1S10N

Diberikan sebuah file tanpa ekstensi, lalu saya langsung identifikasi tetapi ternyata file tersebut merupakan file corrupt yang tidak dapat dibaca :

```
└──(root💀 kali)-[/home/kali/Downloads]
 └──# file tunn3l_v1s10n
tunn3l_v1s10n: data
```

Setelah itu di sini saya berasumsi bahwa kita harus memperbaiki header file nya, dan setelah saya buka menggunakan hexeditor header nya berawalan BM yang artinya file tersebut adalah file BMP. Kemudian saya bandingkan header file tersebut dengan file BMP yang sebenarnya dan benar saja terdapat nilai hexa yang salah :

Kemudian saya perbaiki menjadi seperti berikut :

| File: tutorial1410m.bmp ASCII Offset: 0x00000000 / 0x002C2580 (%00) M |                                                 |                  |                  |                                                       |                 |         |      |      |      |      | File: sample.bmp ASCII Offset: 0x00000000 / 0x00798089 (>00) |      |      |      |  |  |  |  |  |  |  |
|-----------------------------------------------------------------------|-------------------------------------------------|------------------|------------------|-------------------------------------------------------|-----------------|---------|------|------|------|------|--------------------------------------------------------------|------|------|------|--|--|--|--|--|--|--|
| File                                                                  | Actions                                         | Edit             | View             | Help                                                  | File            | Actions | Edit | View | Help | File | Actions                                                      | Edit | View | Help |  |  |  |  |  |  |  |
| File: tutorial1410m.bmp                                               | 42 4D 80 80 70 00 00 00 00 00 8A 90 00 00 7C 00 | BM...p..... .    | File: sample.bmp | 42 4D 8A 80 70 00 00 00 00 00 8A 00 00 00 7C 00       | BM...p..... .   |         |      |      |      |      |                                                              |      |      |      |  |  |  |  |  |  |  |
| 00000010                                                              | 00 00 6E 04 00 00 32 03 00 00 01 00 18 00 00 00 | ..n....2.....    | 00000010         | 00 00 80 07 00 00 00 05 00 00 01 00 18 00 00 00       | ..n....2.....   |         |      |      |      |      |                                                              |      |      |      |  |  |  |  |  |  |  |
| 00000020                                                              | 00 00 58 26 2C 00 25 16 00 00 25 16 00 00 00 00 | ..X,...%.....    | 00000020         | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00       | ..p.....        |         |      |      |      |      |                                                              |      |      |      |  |  |  |  |  |  |  |
| 00000030                                                              | 00 00 00 00 00 00 00 23 1A 17 27 1B 29 20 1D 2A | .....#,...`..`   | 00000030         | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00       | .....#.....     |         |      |      |      |      |                                                              |      |      |      |  |  |  |  |  |  |  |
| 00000040                                                              | 21 3E 26 1D 1A 31 28 25 35 2C 29 33 2A 27 38 2F | 1...1,(%)**/8    | 00000040         | 00 00 00 00 00 00 00 FF 42 47 52 73 80 C2 F5 28 60 B8 | BGRs...`        |         |      |      |      |      |                                                              |      |      |      |  |  |  |  |  |  |  |
| 00000050                                                              | 2F 2C 26 23 33 2A 20 24 20 38 32 2E 32 29 25    | ;/#3%6\$-;2,-2%  | 00000050         | 1E 15 20 85 EB 01 43 33 13 80 66 66 26 40 66          | ...@33.%\$,-2,% |         |      |      |      |      |                                                              |      |      |      |  |  |  |  |  |  |  |
| 00000060                                                              | 30 27 23 33 2A 26 38 2C 28 36 2B 27 39 20 2B 2F | 0'3%3#6,B(9,-9+% | 00000060         | 66 06 A9 99 99 99 3C 0A D7 03 24 5C 8F 32 00 00       | F.....%,\$,-2,% |         |      |      |      |      |                                                              |      |      |      |  |  |  |  |  |  |  |

Kemudian untuk memastikan saya cek lagi dan berhasil diubah ke BMP :

```
[root@kali ~]# file tunn3l_v1s10n
tunn3l_v1s10n: PC bitmap, Windows 98/2000 and newer format, 1134 x 306 x 24

[root@kali ~]# mv tunn3l_v1s10n tunn3l_v1s10n.bmp
```

Setelah itu saya coba buka kembali dan ternyata muncul flag palsu :



Di sini saya berusaha memahami lagi hint soal, dan saya kemudian menemukan bahwa file BMP bisa diperbesar ukuran height gambarnya : (reference : [https://en.wikipedia.org/wiki/BMP\\_file\\_format](https://en.wikipedia.org/wiki/BMP_file_format) )

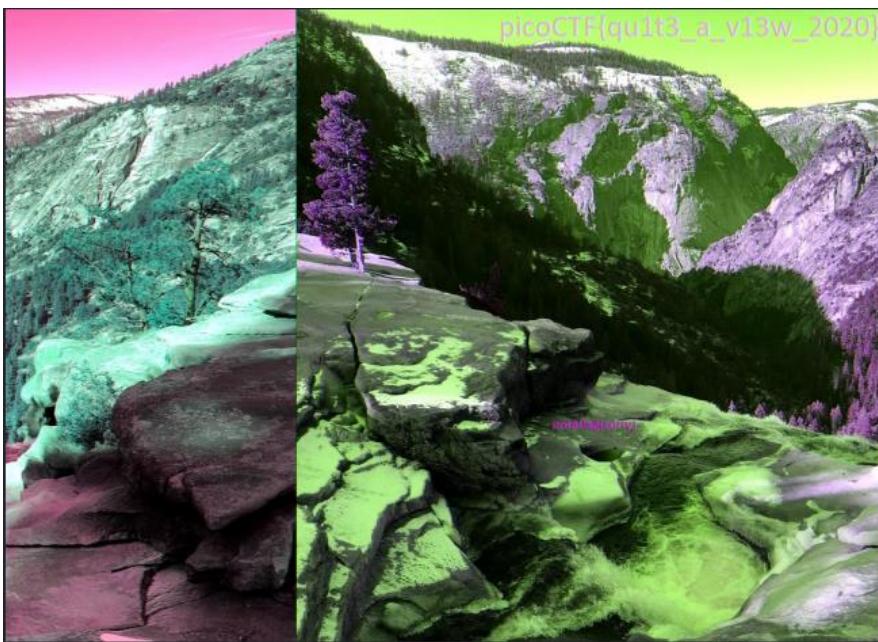
| Offset (hex) | Offset (dec) | Size (bytes) | Windows BITMAPINFOHEADER <sup>[2]</sup>                                                                         |
|--------------|--------------|--------------|-----------------------------------------------------------------------------------------------------------------|
| 0E           | 14           | 4            | the size of this header, in bytes (40)                                                                          |
| 12           | 18           | 4            | the bitmap width in pixels (signed integer)                                                                     |
| 16           | 22           | 4            | the bitmap height in pixels (signed integer)                                                                    |
| 1A           | 26           | 2            | the number of color planes (must be 1)                                                                          |
| 1C           | 28           | 2            | the number of bits per pixel, which is the color depth of the image. Typical values are 1, 4, 8, 16, 24 and 32. |
| 1E           | 30           | 4            | the compression method being used. See the next table for a list of possible values                             |
| 22           | 34           | 4            | the image size. This is the size of the raw bitmap data; a dummy 0 can be given for BI_RGB bitmaps.             |
| 26           | 38           | 4            | the horizontal resolution of the image. (pixel per metre, signed integer)                                       |
| 2A           | 42           | 4            | the vertical resolution of the image. (pixel per metre, signed integer)                                         |
| 2E           | 46           | 4            | the number of colors in the color palette, or 0 to default to $2^n$                                             |
| 32           | 50           | 4            | the number of important colors used, or 0 when every color is important; generally ignored                      |

Lalu saya pergi ke offset (Ctrl+T) dan mengetikkan hexa 16 untuk mencari letak hexa untuk height file dan saya menemukan itu berada di hex 32, namun setelah saya coba rubah itu bukan hex untuk height file. Kemudian saya coba rubah hex 01 yang ada di sebelah hex 32 tersebut dan ternyata berhasil :

```
File: tunn3l_v1s10n.bmp ASCII Offset: 0x00000017 / 0x002C268D (%00)
00000010 00 00 6E 04 00 00 32 01 00 00 01 00 18 00 00 00 .. n ... 2.....
00000020 00 00 58 26 2C 00 25 16 00 00 25 16 00 00 00 00 .. X&,.%...%....
00000030 00 00 00 00 00 00 23 1A 17 27 1E 1B 29 20 1D 2A#...'..) .*
00000040 21 1E 26 1D 1A 31 28 25 35 2C 29 33 2A 27 38 2F !.&..1(%5,)3*`8/
00000050 2C 2F 26 23 33 2A 26 2D 24 20 3B 32 2E 32 29 25 ,/#3*8-$;2.2)%
00000060 30 27 23 33 2A 26 38 2C 28 36 2B 27 39 2D 2B 2F 0'#3*88,(6+'9-+
00000070 26 23 1D 12 0E 23 17 11 29 16 0E 55 3D 31 97 76 &#...#..) ..U=1.v
00000080 66 88 66 52 99 6D 56 9E 70 58 9E 6F 54 9C 6F 54 f.fR.mV.pX.oT.oT
00000090 AB 7E 63 BA 8C 6D BD 8A 69 C8 97 71 C1 93 71 C1 .~c..m..i..q..q.
000000A0 97 74 C1 94 73 C0 93 72 C0 8F 6B BD 8E 6E BA 8D .t..s..r..o..n..
000000B0 6B 87 8D 6A B0 85 64 A0 74 55 A3 77 5A 98 6F 56 k..j..d.tU.wZ.oV
000000C0 76 52
000000D0 5E 54
000000E0 5E 86
000000F0 87 6D
00000100 7E 6E
00000110 31 4F
00000120 50 39
00000130 29 1E
 Goto Offset
 Offset: 0x16
Hint: Decimal 255 = Hex 0xFF = Octal 0377
00000140 24 43 2F 24 40 2A 1F 48 32 27 4B 32 28 47 2E 24
00000150 40 27 1D 45 2C 22 4C 34 28 4C 34 28 4B 33 27 4A
00000160 32 26 4C 32 24 4E 34 26 50 35 27 52 37 29 53 36
00000170 28 55 38 2A 4B 30 22 5D 42 34 63 49 39 49 2F 1F
00000180 44 28 1B 4D 34 24 4D 36 27 4A 33 24 46 2C 20 48
00000190 2E 22 46 2E 22 44 2E 22 3C 26 1B 32 20 15 30 1F
000001A0 16 32 23 1A 36 27 1E 3C 2B 22 3E 2B 24 42 2C 26 .#.#'.<+>+$B,&
vR;qR=L0@mRdnSiw
^TS93pXRvaYs_T-k
^tc~jVvbPv^LzbP
.m].iY.sc..q..t.
~n..q..scsZJpWGZA
1066N7'08(08(0:*
P9)08)K5)P:/K5*?
).B.#K7,E18?+ C/
$C/$@*.H2'K2(G.$
@'.E,"L4(L4(K3'J
2&L2$N4&P5'R7)S6
(U8*K0"]B4cI9T).
D+.M4$M6'J3$F, H
."F."D."<>5..2..0.
.2#.6'.<+>+$B,&
```

```
File: tunn3l_v1s10n.bmp ASCII Offset: 0x00000018 / 0x002C268D (%00)
00000010 00 00 6E 04 00 00 32 03 00 00 01 00 18 00 00 00 .. n ... 2.....
00000020 00 00 58 26 2C 00 25 16 00 00 25 16 00 00 00 00 .. X&,.%...%....
00000030 00 00 00 00 00 00 23 1A 17 27 1E 1B 29 20 1D 2A#...'..) .*
00000040 21 1E 26 1D 1A 31 28 25 35 2C 29 33 2A 27 38 2F !.&..1(%5,)3*`8/
00000050 2C 2F 26 23 33 2A 26 2D 24 20 3B 32 2E 32 29 25 ,/#3*8-$;2.2)%
00000060 30 27 23 33 2A 26 38 2C 28 36 2B 27 39 2D 2B 2F 0'#3*88,(6+'9-+
```

Saya merubah nilai nya menjadi hex 03 dan didapat flag yang tertera di atas gambar :



Flag = picoCTF{quilt3\_a\_v13w\_2020}