

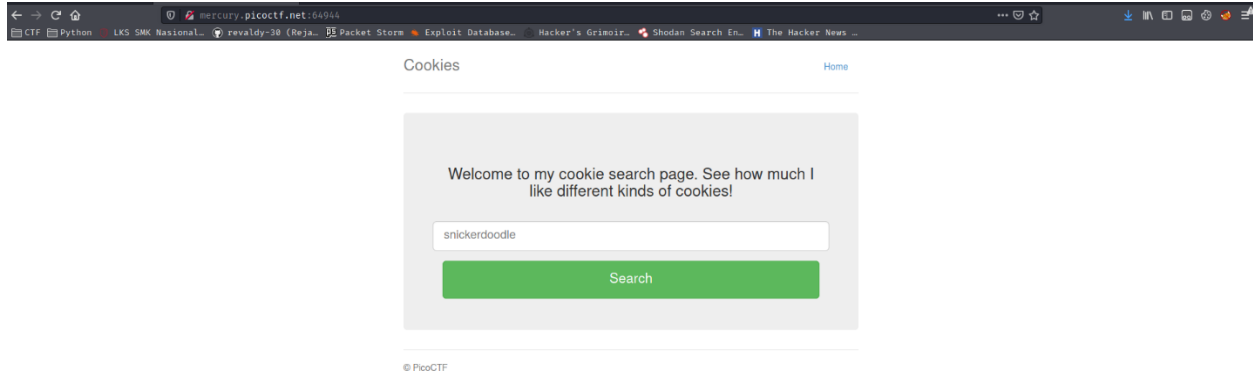
# \\ REJA REVALDY F. \\

## PICO CTF

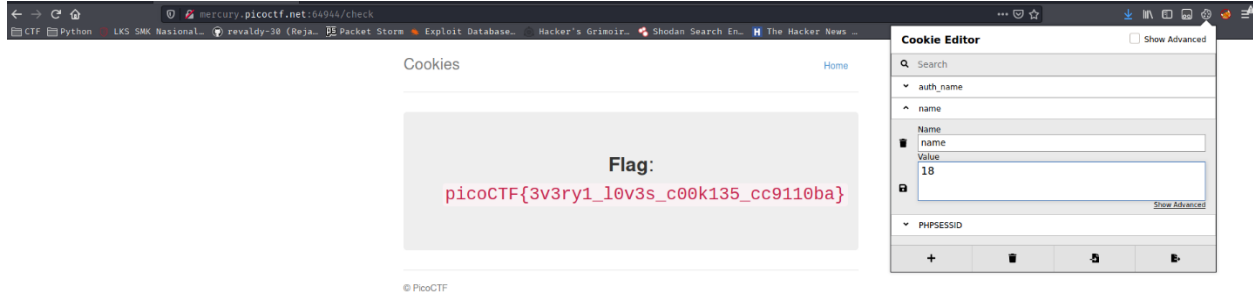
|                            |    |
|----------------------------|----|
| [Web Exploit]              | 2  |
| Cookies                    | 2  |
| Insp3ct0r                  | 3  |
| Scavenger Hunt             | 4  |
| Where are the robots?      | 6  |
| Logon                      | 7  |
| dont-use-client-side       | 9  |
| GET aHEAD                  | 10 |
| picobrowser                | 11 |
| [Cryptography]             | 12 |
| Mod26                      | 12 |
| TheNumbers                 | 13 |
| 13                         | 14 |
| Caesar                     | 15 |
| [Reverse Engineering]      | 16 |
| Crackme.py                 | 16 |
| [Forensic]                 | 18 |
| Information                | 18 |
| Dolls                      | 19 |
| Glory of the garden        | 20 |
| Wireshark doo doo do do... | 21 |
| SoMeta                     | 22 |
| Extension                  | 23 |
| Like1000                   | 24 |
| [General Skills]           | 25 |
| Based                      | 25 |
| Mus1c                      | 26 |
| [Binary Exploit]           | 28 |

# [Web Exploit]

## Cookies



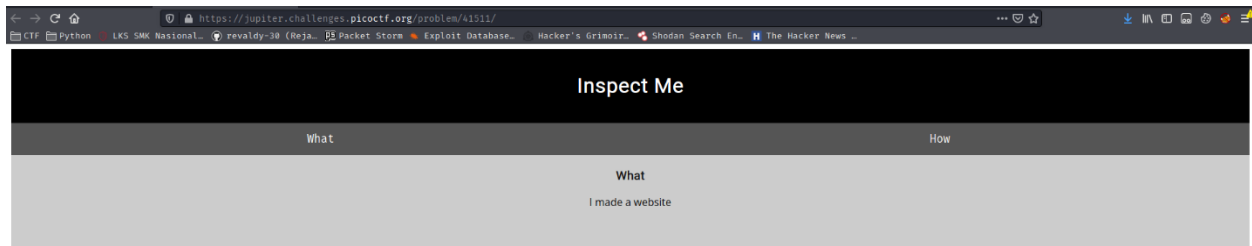
Diberikan sebuah website pencarian cookie, setelah saya cek pencarian cookie tersebut menggunakan cookie, disini langsung saja saya coba modifikasi value cookie tersebut menggunakan extension “cookie editor”, lalu ditemukan flag di cookie dengan { ‘name’ = 18 }



**FLAG = picoCTF{3v3ry1\_l0v3s\_c00k135\_cc9110ba}**

# Insp3ct0r

Diberikan sebuah website lalu kita disuruh melakukan sebuah inspesction kepada website tersebut, langsung saja saya “view page source”



dan saya menemukan komentar di html tersebut yang berisikan potongan dari flag, lalu saya coba mencari sisa potongan flag tersebut

```
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>
38
```

setelah itu saya coba membuka css website tersebut dan menemukan potongan dari flag nya lagi

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

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

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

/* You need CSS to make pretty pages. Here's part 2/3 of the flag: t3ctive_0r_ju5t */
```

setelah itu saya coba membuka js website nya dan menemukan potongan terakhir dari flag nya

```
window.onload = function() {
  openTab('tabintro', this, '#222');
}

/* Javascript sure is neat. Anyways part 3/3 of the flag: _lucky?832b0699} */
```

FLAG = picoCTF{tru3\_d3t3ct1ve\_0r\_ju5t\_lucky?832b0699}

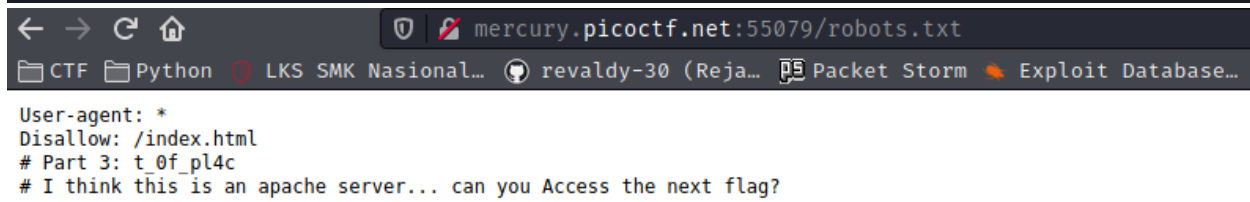
## Scavenger Hunt

Untuk soal ini hampir sama dengan soal insp3ctor, langsung saja saya cek page source nya dan membuka css dan js nya dan saya menemukan dua bagian dari flag yang berada di komentar html dan file css

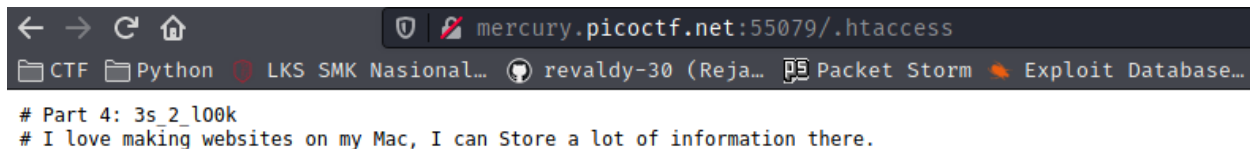
```
1 <!doctype html>
2 <html>
3   <head>
4     <title>Scavenger Hunt</title>
5     <link href="https://fonts.googleapis.com/css?family=Open+Sans|Roboto" rel="stylesheet">
6     <link rel="stylesheet" type="text/css" href="mycss.css">
7     <script type="application/javascript" src="myjs.js"></script>
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: <br/>
27          HTML <br/>
28          CSS <br/>
29          JS (JavaScript)
30        </p>
31        <!-- Here's the first part of the flag: picoCTF{t -->
32      </div>
33
34    </div>
35  </body>
36 </html>
37
38
39 .tabcontent {
40   color: #111;
41   display: none;
42   padding: 50px;
43   text-align: center;
44 }
45
46 #tabintro { background-color: #ccc; }
47 #tababout { background-color: #ccc; }
48
49 /* CSS makes the page look nice, and yes, it also has part of the flag. Here's part 2: h4ts_4_l0 */
```

Lalu disini saya mencoba menggunakan gobuster untuk melakukan dirlisting dan menemukan .htaccess dan robot.txt, disini langsung saja saya mengarahkan nya

```
(kali@kali)-[~]
└─$ gobuster dir -u http://mercury.picoctf.net:55079/ -w /usr/share/wordlists/dirb/common.txt
=====
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:                http://mercury.picoctf.net:55079/
[+] Method:             GET
[+] Threads:            10
[+] Wordlist:            /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent:         gobuster/3.1.0
[+] Timeout:            10s
=====
2021/10/14 02:16:15 Starting gobuster in directory enumeration mode
=====
./htaccess              (Status: 200) [Size: 95]
/index.html             (Status: 200) [Size: 961]
/robots.txt             (Status: 200) [Size: 124]
=====
2021/10/14 02:21:06 Finished
=====
```

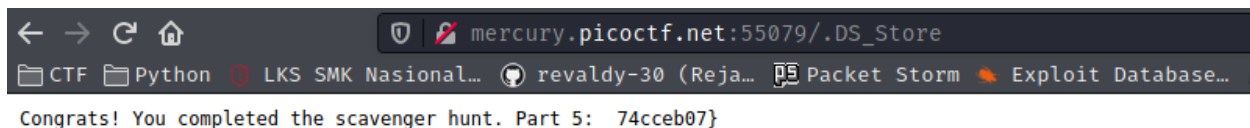


```
mercury.picoctf.net:55079/robots.txt
CTF Python LKS SMK Nasional... revaldy-30 (Reja... Packet Storm Exploit Database...
User-agent: *
Disallow: /index.html
# Part 3: t_0f_pl4c
# I think this is an apache server... can you Access the next flag?
```



```
mercury.picoctf.net:55079/.htaccess
CTF Python LKS SMK Nasional... revaldy-30 (Reja... Packet Storm Exploit Database...
# Part 4: 3s_2_l00k
# I love making websites on my Mac, I can Store a lot of information there.
```

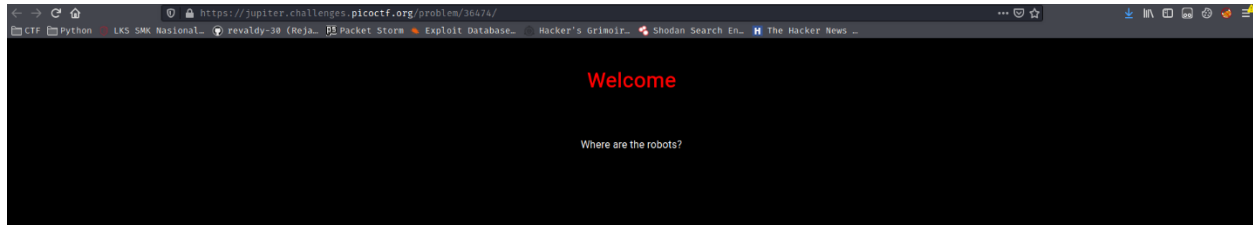
Disini saya kira flag nya berakhir di part4 lalu saya coba masukkan flag nya dan ternyata ada yang kurang lalu setelah saya cek di .htaccess ternyata 'Store' menggunakan huruf kapital disini saya duga itu adalah hint dan developer menggunakan mac untuk membuat website nya lalu saya menemukan `https://en.wikipedia.org/wiki/.DS\_Store` lalu langsung saja saya buka dan menemukan flag terakhir lalu saya gabungkan



```
mercury.picoctf.net:55079/.DS_Store
CTF Python LKS SMK Nasional... revaldy-30 (Reja... Packet Storm Exploit Database...
Congrats! You completed the scavenger hunt. Part 5: _74cceb07}
```

FLAG = picoCTF{th4ts\_4\_l0t\_0f\_pl4c3s\_2\_l00k\_74cceb07}

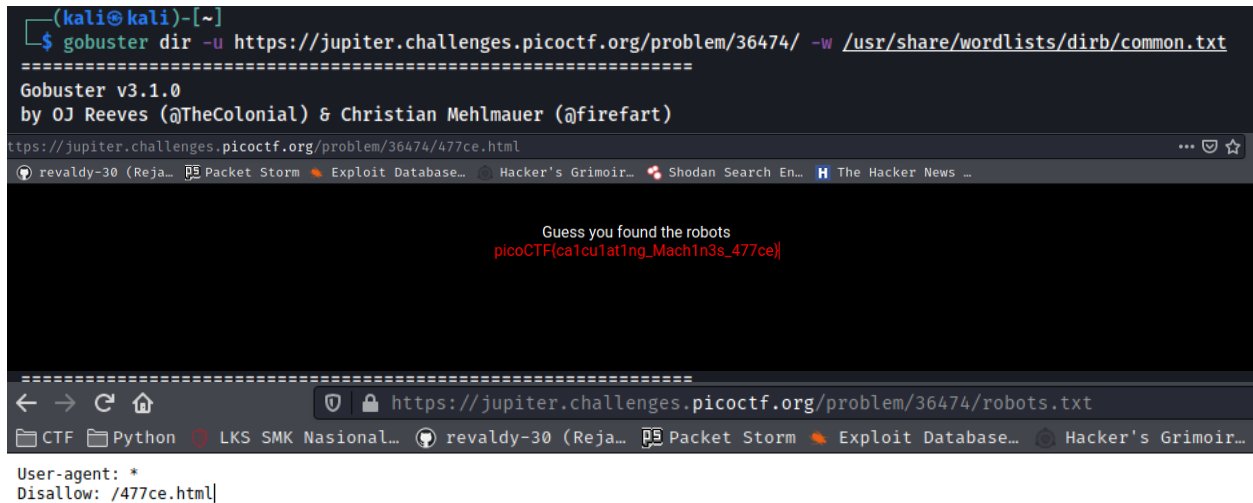
## Where are the robots?



Disini diberikan website dengan tampilan seperti berikut, lalu setelah saya coba baca hint “What part of the website could tell you where the creator doesn't want you to look?” setelah itu saya melakukan dirlisting

Lalu saya menemukan robots.txt, langsung saja saya tembak ke url website dan menemukan

Disini ditemukan file html dengan nama 477ce.html lalu saya coba buka dan menemukan flagnya



**FLAG = picoCTF{ca1cu1at1ng\_Mach1n3s\_477ce}**

## Logon

Disini diberikan website dengan form login, lalu saya coba masukkan kredensial sembarang dan berhasil untuk login

Factory Login [Home](#) [Sign Out](#)

© PicoCTF 2019

Factory Login [Home](#) [Sign Out](#)

Success: You logged in! Not sure you'll be able to see the flag though.

No flag for you

© PicoCTF 2019

Lalu saya cek cookie yang ada di website tersebut dan menemukan cookie admin yang memiliki value False disini langsung saja saya ubah ke True dan menemukan flag nya

Cookie Editor ☐ Show Advanced

Search

admin

Name: admin  
Value: True

password

Name: password  
Value: joe

+ [trash] [copy] [paste]

**Flag:**

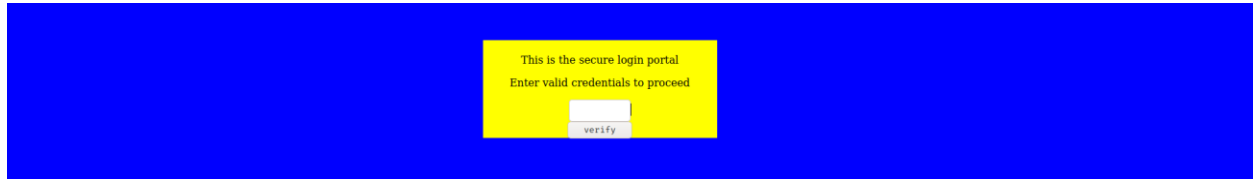
picoCTF{th3\_c0nsp1r4cy\_11v3s\_d1c24fef}

**FLAG = picoCTF{th3\_c0nsp1r4cy\_11v3s\_d1c24fef}**



## dont-use-client-side

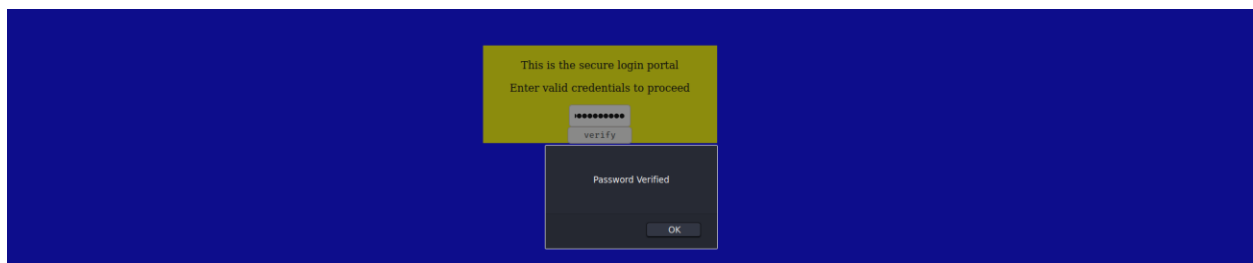
Website berisi form untuk melakukan pengecekan password



Disini saya langsung coba membuka page source dan menemukan script javascript dan setelah saya pahami script tersebut langsung saja saya mengurutkan script split tersebut

```
1 <html>
2 <head>
3 <title>Secure Login Portal</title>
4 </head>
5 <body bgcolor=blue>
6 <!-- standard MD5 implementation -->
7 <script type="text/javascript" src="md5.js"></script>
8
9 <script type="text/javascript">
10  function verify() {
11    checkpass = document.getElementById("pass").value;
12    split = 4;
13    if (checkpass.substring(0, split) == 'pico') {
14      if (checkpass.substring(split*6, split*7) == 'a3c8') {
15        if (checkpass.substring(split, split*2) == 'CTF{') {
16          if (checkpass.substring(split*4, split*5) == 'ts_p') {
17            if (checkpass.substring(split*3, split*4) == 'lien') {
18              if (checkpass.substring(split*5, split*6) == 'lz_1') {
19                if (checkpass.substring(split*2, split*3) == 'no_c') {
20                  if (checkpass.substring(split*7, split*8) == '9') {
21                    alert("Password Verified")
22                  }
23                }
24              }
25            }
26          }
27        }
28      }
29    }
30  }
31  else {
32    alert("Incorrect password");
33  }
34 }
35 </script>
```

Yang hasilnya “picoCTF{no\_clients\_plz\_1a3c89}” lalu langsung saja saya coba inputkan ke form di website

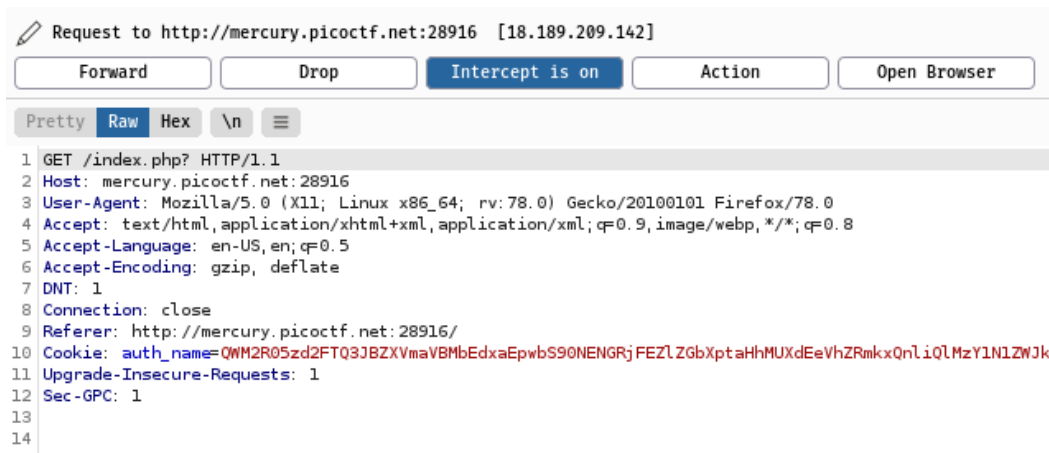


FLAG = picoCTF{no\_clients\_plz\_1a3c89}

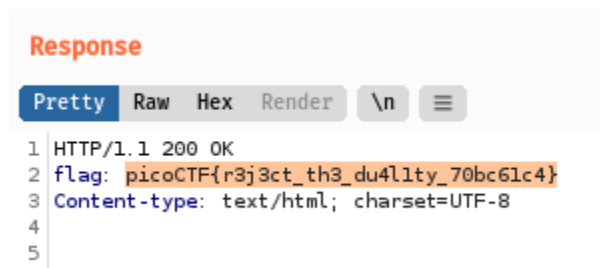
## GET aHEAD



Diberikan website dengan dua button yang menggunakan method request get dan post, lalu di hint diberikan sebuah petunjuk yaitu, “Maybe you have more than 2 choices”, “Check out tools like Burpsuite to modify your requests and look at the responses” disini saya asumsikan kita harus memodifikasi method request menggunakan burpsuit.



Disini setelah saya kebingungan agak lama saya membaca soalnya Kembali dan menemukan bahwa huruf HEAD adalah kapital semua disini saya asumsikan bahwa kit harus memodifikasi request method nya dengan head



Setelah saya modifikasi request method tersebut dengan head saya langsung melakukan pengecekan di response dan menemukan flagnya

FLAG = picoCTF{r3j3ct\_th3\_du4l1ty\_70bc61c4}

## picobrowser

Disoal kita diberikan petunjuk bahwa web ini hanya bisa dirender menggunakan picobrowser lalu langsung saja saya menggunakan command curl untuk membukanya

```
"curl -v http://mercury.picoctf.net:34588/ -A picobrowser"
```

```
<div class="jumbotron">
  <p class="lead"></p>
  <p style="text-align:center; font-size:30px;"><b>Flag</b>: <code>picoCTF{p1c0_s3cr3t_ag3nt_e9b160d0}</code></p>
  <!-- <p><a class="btn btn-lg btn-success" href="admin" role="button">Click here for the flag!</a> -->
  <!-- </p> -->
</div>

<footer class="footer">
  <p>&copy; PicoCTF 2019</p>
</footer>

</div>
<script>
$(document).ready(function(){
  $(".close").click(function(){
    $(".myAlert").alert("close");
  });
});
</script>
</body>

* Connection #0 to host jupiter.challenges.picoctf.org left intact
</html>
```

Setelah saya membuka web tersebut menggunakan picobrowser terdapat flag bagian html nya

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

# [Cryptography]

## Mod26

**Mod 26**

👤 | 10 points ✕

Tags: **Category: Cryptography**

AUTHOR: PANDU

Hints

Description

1

Cryptography can be easy, do you know what ROT13 is?

cvpbPGS{arkg\_gvzr\_V'yy\_ge1\_2\_ebhaqf\_bs\_ebg13\_uJdSftmh}

38,387 solves / 56,936 attempts (67%)

🗨 93% Liked 🍷

picoCTF{FLAG}

Submit Flag

Disini kita diberikan soal enkripsi rot13 lalu saya langsung coba melakukan decode di website [asciitohex.com](https://www.asciitohex.com)

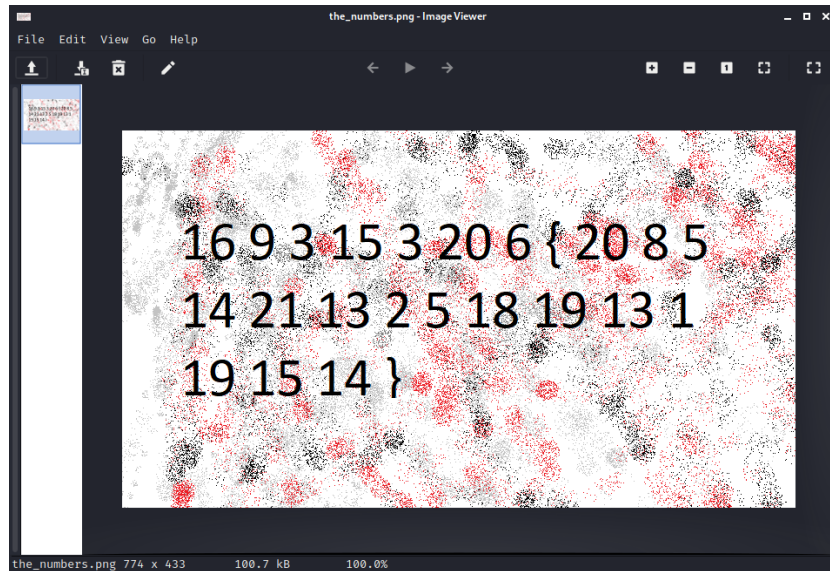
### ASCII to Hex

...and other free text conversion tools

|   |  |  |
|---|--|--|
| <b>Text (ASCII / ANSI)</b><br>picoCTF{next_time_I'll_try_2_rounds_of_rot13_hWqfsgzu}<br><b>Convert Highlight Text</b>         | <b>Binary</b><br>01110000 01101001 01100011 01101111<br>01000011 01010100 01000110 01111011<br>01101110 01100101 01111000 01110100<br>01011111 01110100 01101001 01101101<br>01100101 01011111 01001001 00100111<br>01101100 01101100 01011111 01110100<br>01110010 01111001 01011111 00110010<br>01011111 01110010 01101111 01110101<br><b>Convert Highlight Text</b> | <b>Hexadecimal</b><br>70 69 63 6f 43 54 46 7b 6e 65 78 74 5f 74 69 6d 65 5f<br>49 27 6c 6c 5f 74 72 79 5f 32 5f 72 6f 75 6e 64 73 5f<br>6f 66 5f 72 6f 74 31 33 5f 68 57 71 46 73 67 7a 75 7d<br><b>Convert Highlight Text</b> |
| <b>BASE64</b><br>cGJib0NURintuZxh0X3RpbWV5Sdsbf90cnllMI9yb3Vu<br>ZHNfb2Zfcm90MTNfaFdxRnNenV9<br><b>Convert Highlight Text</b> | <b>Decimal</b><br>112 105 99 111 67 84 70 123 110 101 120 116 95<br>116 105 109 101 95 73 39 108 108 95 116 114 121<br>95 50 95 114 111 117 110 100 115 95 111 102 95<br>114 111 116 49 51 95 104 87 113 70 115 103 122<br>117 125<br><b>Convert Highlight Text</b>  | <b>ROT13</b><br>cvpbPGS{arkg_gvzr_V'yy_ge1_2_ebhaqf_bs_ebg13_uJdSftmh}<br><b>Convert Highlight Text</b>  |

FLAG = picoCTF{next\_time\_I'll\_try\_2\_rounds\_of\_rot13\_hWqfsgzu}

## TheNumbers



Diberikan sebuah gambar dengan bilangan decimal, lalu saya coba mencari cara mengubah nya ke bentuk bilangan ascii, setelah saya pahami ini merupakan index dari alphabet “16 = Q” lalu untuk alphabet ke “15 = P” disini saya langsung menemukan logikanya yaitu index dari alphabet lalu dikurangi satu, langsung saja saya buat script python nya

```
kali@kali: ~/Belajar/exercise/PICO/thenumber
#!/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, 1,
9, 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)
```

```
(kali@kali)-[~/Belajar/exercise/PICO/thenumber]
$ ./script.py
PICOCTF{THENUMBERSMASON}
```

**FLAG = PICOCTF{THENUMBERSMASON}**

## 13

13



👤 | 100 points ✕

Tags: **Category: Cryptography**

AUTHOR: ALEX FULTON/DANIEL TUNITIS

### Description

Cryptography can be easy, do you know what ROT13 is?

`cvpbPGS{abg_gbb_onq_bs_n_ceboyzz}`

### Hints

1

18,102 solves / 22,746 attempts (80%)



88% Liked



picoCTF{FLAG}

Submit Flag

Dari soal ini kita sudah bisa menembak cipher text yang digunakan adalah rot13, lalu saya langsung coba melakukan decode di website [asciitohex.com](https://www.asciitohex.com)

### ASCII to Hex

...and other free text conversion tools

| Text (ASCII / ANSI)   | Binary  | Hexadecimal   |
|---|---|---|
| <p>picoCTF(not_too_bad_of_a_problem)</p> <p>Convert Highlight Text</p>                          | <p>01110000 01101001 01100011 01101111 01000011<br/>01010100 01000110 01111011 01101110 01101111<br/>01101000 01011111 01110100 01101111 01101111<br/>01011111 01100010 01100001 01100100 01011111<br/>01101111 01100110 01011111 01100001 01011111<br/>01110000 01110010 01101111 01100010 01101100<br/>01100101 01101101 01111101</p> <p>Convert Highlight Text</p> | <p>70 69 63 6f 43 54 46 7b 6e 6f 74 5f 74 6f 6f 5f 62 61<br/>64 5f 6f 66 5f 61 5f 70 72 6f 62 6c 65 6d 7d</p> <p>Convert Highlight Text</p> |
| <p>BASE64</p> <p>cGijb0NUrntub3RfdG9vY2JhZm9vZl9hX3Byb2JsZW19</p> <p>Convert Highlight Text</p> | <p>Decimal</p> <p>112 105 99 111 67 84 70 123 110 111 116 95 116<br/>111 111 95 98 97 100 95 111 102 95 97 95 112 114<br/>111 98 108 101 109 125</p> <p>Convert Highlight Text</p>  | <p>ROT13</p> <p>cvpbPGS{abg_gbb_onq_bs_n_ceboyzz}</p> <p>Convert Highlight Text</p>   |

FLAG = picoCTF{not\_too\_bad\_of\_a\_problem}

## Caesar

```
(kali@kali)-[~/Belajar/exercise/PICO/caesar]
$ cat ciphertext
picoCTF{dspttjohuifsvcjdpabrkttds}
```

Diberikan sebuah cipher text yang menggunakan enkripsi Caesar cipher lalu saya mencoba melakukan dekripsi di website

**Caesarian Shift**  
Rumkin.com >> Web-Based Tools >> Ciphers and Codes

This is a standard Caesarian 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 this 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 may 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:

This is your encoded or decoded text:

Disini saya menemukan flag nya ke geseran ke 25

**FLAG = picoCTF{crossingtherubiconzaqjsscr}**

# [Reverse Engineering]

## Crackme.py

```
(kali@kali)-[~/Belajar/exercise/PICO/crackme]
$ python crackme.py
What's your first number? 213
What's your second number? 13
The number with largest positive magnitude is 213
```

Diberikan file crackme.py yang Ketika dijalankan maka meminta inputan untuk memasukan decimal, disini saya tertarik dengan source code nya maka saya buka

```
GNU nano 5.8 crackme.py
# Hiding this really important number in an obscure piece of code is brilliant!
# AND it's encrypted!
# We want our biggest client to know his information is safe with us.
bezos_cc_secret = "A:4@r%uL`M-^M0c0AbcM-MFE02fh3e4a5N"

# Reference alphabet
alphabet = "!\"#$%&'()*+,-./0123456789;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ"+ \
          "[\\]^_`abcdefghijklmnopqrstuvwxyz{|}~"

def decode_secret(secret):
    """ROT47 decode

    NOTE: encode and decode are the same operation in the ROT cipher family.
    """

    # Encryption key
    rotate_const = 47

    # Storage for decoded secret
    decoded = ""

    # decode loop
    for c in secret:
        index = alphabet.find(c)
        original_index = (index + rotate_const) % len(alphabet)
        decoded = decoded + alphabet[original_index]

    print(decoded)
```

Disini saya menemukan function decode\_secret dan variable bezos\_cc\_secret, untuk function decode\_secret saya asumsikan adalah fungsi untuk melakukan decode dan variable bezos\_cc\_secret adalah cipher text nya, langsung saja saya modifikasi script dan jalankan function tersebut



```

bezos_cc_secret = "A:4@r%uL`M-^M0c0AbcM-MFE02fh3e4a5N"

# Reference alphabet
alphabet = "!\"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNopqrstuvwxyz{ }~"

def decode_secret(secret):
    """ROT47 decode

    NOTE: encode and decode are the same operation in the ROT cipher family.
    """

    # Encryption key
    rotate_const = 47

    # Storage for decoded secret
    decoded = ""

    # decode loop
    for c in secret:
        index = alphabet.find(c)
        original_index = (index + rotate_const) % len(alphabet)
        decoded = decoded + alphabet[original_index]

    print(decoded)

def choose_greatest():
    """Echo the largest of the two numbers given by the user to the program

    Warning: this function was written quickly and needs proper error handling
    """

    user_value_1 = input("What's your first number? ")
    user_value_2 = input("What's your second number? ")
    greatest_value = user_value_1 # need a value to return if 1 & 2 are equal

    if user_value_1 > user_value_2:
        greatest_value = user_value_1
    elif user_value_1 < user_value_2:
        greatest_value = user_value_2

    print( "The number with largest positive magnitude is "
          + str(greatest_value) )

decode_secret(bezos_cc_secret)

```

```

(kali@kali)-[~/Belajar/exercise/PICO/crackme]
$ python crackme.py
picoCTF{1|\|_4_p34|\|ut_a79b6c2d}

```

FLAG = picoCTF{1|\|\_4\_p34|\|ut\_a79b6c2d}

# [Forensic]

## Information



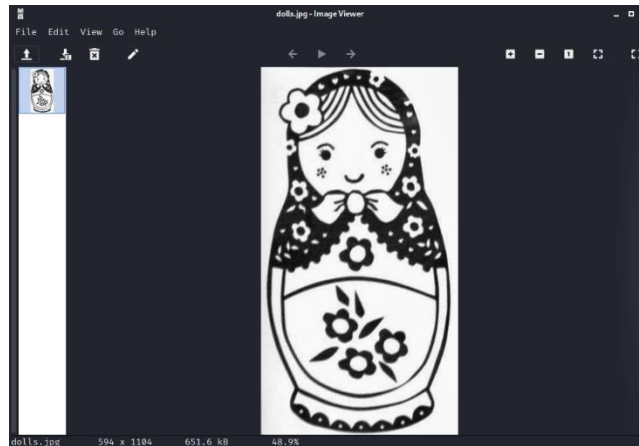
Diberikan sebuah gambar kucing, lalu saya langsung saja menggunakan exiftool untuk melihat meta data dari gambar tersebut dan menemukan sebuah base64 yang berada di license, langsung saja saya decode dan menemukan flagnya

```
(kali@kali)-[~/Belajar/exercise/PICO/information]
$ exiftool cat.jpg
ExifTool Version Number      : 12.32
File Name                    : cat.jpg
Directory                    : .
File Size                    : 858 KiB
File Modification Date/Time  : 2021:10:15 01:10:33-04:00
File Access Date/Time       : 2021:10:15 01:10:46-04:00
File Inode Change Date/Time  : 2021:10:15 01:10:43-04:00
File Permissions             : -rw-r--r--
File Type                    : JPEG
File Type Extension         : jpg
MIME Type                    : image/jpeg
JFIF Version                 : 1.02
Resolution Unit              : None
X Resolution                 : 1
Y Resolution                 : 1
Current IPTC Digest          : 7a78f3d9c9fb1ce42ab5a3aa30573d617
Copyright Notice             : PicoCTF
Application Record Version   : 4
XMP Toolkit                  : Image::ExifTool 10.80
License                      : cGljb0NURnt0aGVfbTN0YWVhdGFfMXNfbW9kaWZpZW99
Rights                       : PicoCTF
Image Width                  : 2560
Image Height                 : 1598
Encoding Process             : Baseline DCT, Huffman coding
Bits Per Sample              : 8
Color Components             : 3
Y Cb Cr Sub Sampling         : YCbCr4:2:0 (2 2)
Image Size                   : 2560x1598
Megapixels                   : 4.1

(kali@kali)-[~/Belajar/exercise/PICO/information]
$ echo "cGljb0NURnt0aGVfbTN0YWVhdGFfMXNfbW9kaWZpZW99" | base64 -d
picoCTF{the_m3tadata_1s_modified}
```

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

## Dolls



Diberikan sebuah gambar boneka Matryoshka, disini saya mencoba menggunakan binwalk untuk melakukan pengecekan apakah ada file yang disembunyikan di dalam gambar tersebut, dan ternyata ada file zip dan png, setelah saya buka file tersebut ternyata ada png lagi dan terdapat file didalamnya disini saya melakukan hal tersebut berulang kali sampai menemukan file flag nya di gambar ke 4, disini saya ambil kesimpulan untuk soal ctf ini merepresentasikan boneka Matryoshka yang terdapat boneka lagi didalamnya

```
(kali@kali)-[~/.../base_images/_2_c.jpg.extracted/base_images/_3_c.jpg.extracted]
$ cd base_images

(kali@kali)-[~/.../_2_c.jpg.extracted/base_images/_3_c.jpg.extracted/base_images]
$ binwalk -e 4_c.jpg

DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0            0x0          PNG image, 320 x 768, 8-bit/color RGBA, non-interlaced
3226         0xC9A        TIFF image data, big-endian, offset of first image directory: 8
79578        0x136DA      Zip archive data, at least v2.0 to extract, compressed size: 63, uncompressed size: 8
1, name: flag.txt
79785        0x137A9      End of Zip archive, footer length: 22

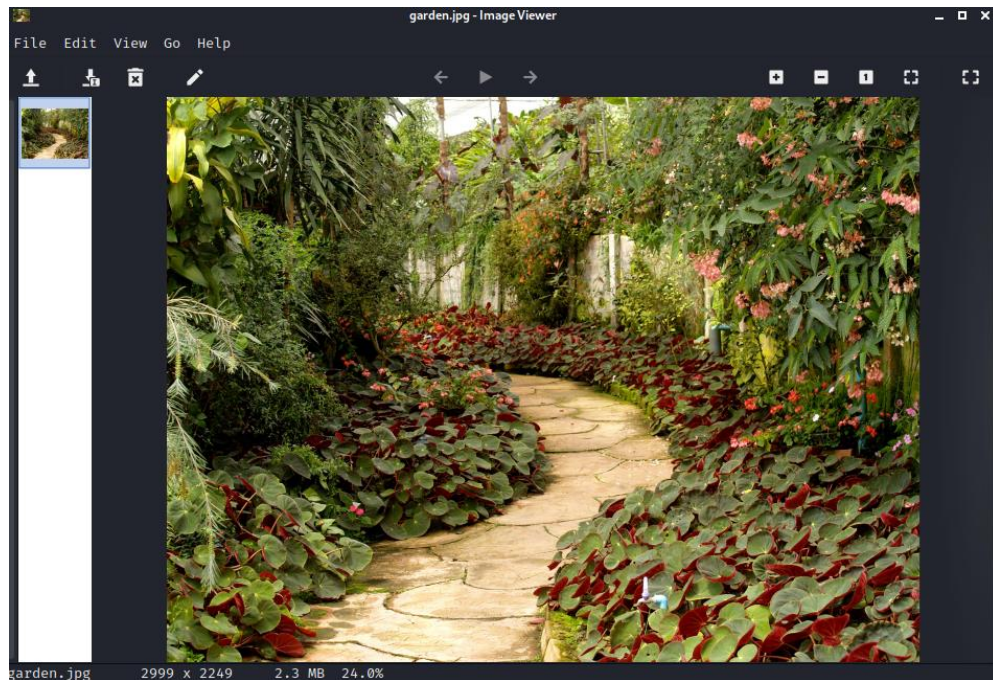
(kali@kali)-[~/.../_2_c.jpg.extracted/base_images/_3_c.jpg.extracted/base_images]
$ ls
4_c.jpg  _4_c.jpg.extracted

(kali@kali)-[~/.../_2_c.jpg.extracted/base_images/_3_c.jpg.extracted/base_images]
$ cat _4_c.jpg.extracted/flag.txt
picoCTF{bf6acf878dcdbd752f4721e41b1b1b66b}

(kali@kali)-[~/.../_2_c.jpg.extracted/base_images/_3_c.jpg.extracted/base_images]
$
```

**FLAG = picoCTF{bf6acf878dcdbd752f4721e41b1b1b66b}**

## Glory of the garden



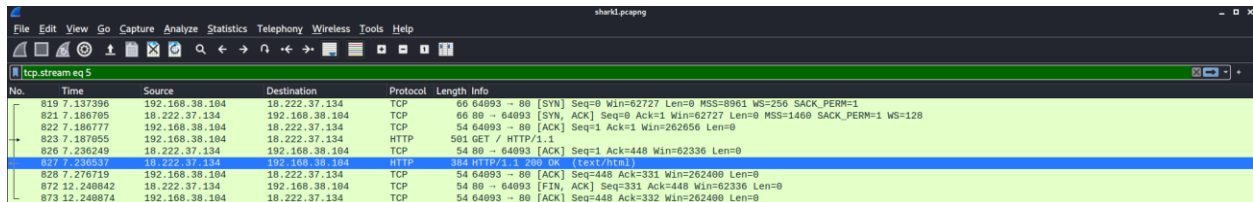
Diberikan sebuah file jpg lalu saya menggunakan command strings dan mendapatkan flagnya

```
(kali@kali)-[~/Belajar/exercise/PICO/glorrygarden]
$ strings garden.jpg | grep picoCTF
Here is a flag "picoCTF{more_than_m33ts_the_3y3657BaB2C}"
```

FLAG = picoCTF{more\_than\_m33ts\_the\_3y3657BaB2C}

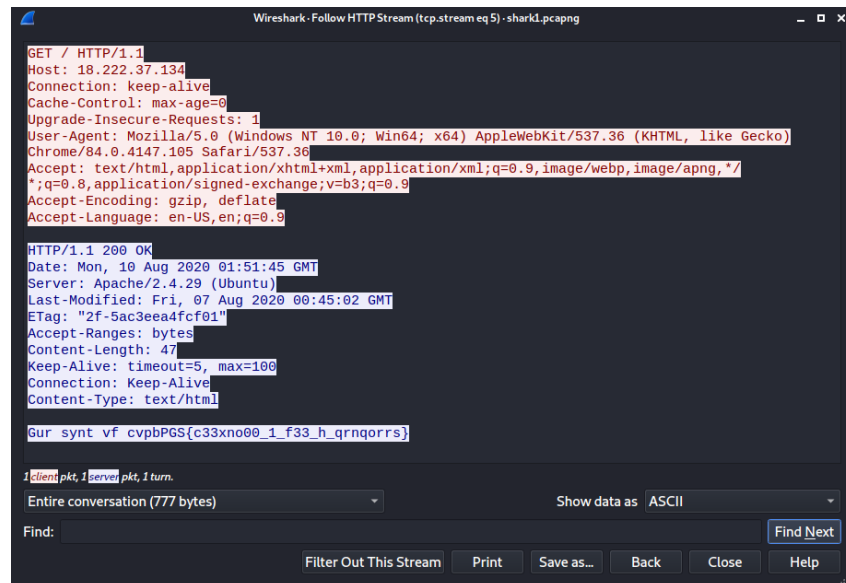
## Wireshark doo doo do do...

Diberikan sebuah file pcap lalu saya menggunakan wireshark untuk membukanya, setelah saya amati saya menemukan text/html di no 827 setelah itu saya melakukan follow http stream



| No. | Time      | Source         | Destination    | Protocol | Length | Info  |
|-----|-----------|----------------|----------------|----------|--------|---|
| 819 | 7.137396  | 192.168.38.104 | 18.222.37.134  | TCP      | 60     | 64093 → 80 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 WS=256 SACK_PERM=1            |
| 821 | 7.186785  | 18.222.37.134  | 192.168.38.104 | TCP      | 60     | 80 → 64093 [SYN, ACK] Seq=0 Ack=1 Win=62727 Len=0 MSS=1460 SACK_PERM=1 WS=128 |
| 822 | 7.186777  | 192.168.38.104 | 18.222.37.134  | TCP      | 54     | 64093 → 80 [ACK] Seq=1 Ack=1 Win=26256 Len=0                                  |
| 823 | 7.187855  | 192.168.38.104 | 18.222.37.134  | HTTP     | 501    | GET / HTTP/1.1  |
| 826 | 7.236249  | 18.222.37.134  | 192.168.38.104 | TCP      | 54     | 80 → 64093 [ACK] Seq=1 Ack=448 Win=62336 Len=0                                |
| 827 | 7.276719  | 192.168.38.104 | 18.222.37.134  | TCP      | 54     | 64093 → 80 [ACK] Seq=448 Ack=331 Win=262400 Len=0                             |
| 872 | 12.240842 | 18.222.37.134  | 192.168.38.104 | TCP      | 54     | 80 → 64093 [FIN, ACK] Seq=331 Ack=448 Win=62336 Len=0                         |
| 873 | 12.240874 | 192.168.38.104 | 18.222.37.134  | TCP      | 54     | 64093 → 80 [ACK] Seq=448 Ack=332 Win=262400 Len=0                             |

Disini saya menemukan sebuah format yang sama dengan flag, lalu saya asumsikan flag ini terenkripsi menggunakan rot13 langsung saja saya decode



### ASCII to Hex

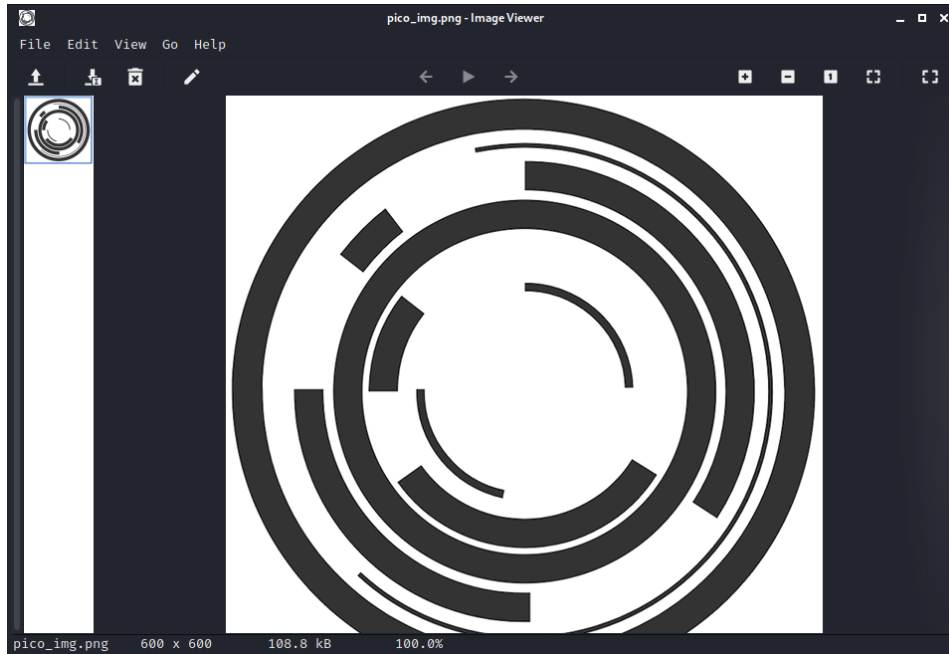
...and other free text conversion tools



| Text (ASCII / ANSI)                | Binary   | Hexadecimal   | Base64  | Decimal  | ROT13                             |
|------------------------------------|--|---|---|--|-----------------------------------|
| picoCTF{p33kab00_1_s33_u_deadbeef} | 01110000 01101001 01100011 01101111 01000011 01010100 01000010 01111011 01110000 00110011 00110011 01101011 01100001 01100010 00110000 00110000 01011111 00110001 01011111 01110011 00110011 00110011 01011111 01101011 01011111 01001001 01100010 01100010 01100101 01100010 01111011 | 70 69 63 6f 43 54 46 7b 70 33 33 6b 61 62 30 30 5f 31 5f 73 33 33 5f 75 5f 64 65 61 64 62 65 65 66 7d | cGpjb0UURntwMzNrYWwMfBx3MzM1M191X2RYWVWZ WmVQ== | 112 105 99 111 67 84 70 123 112 51 51 107 97 98 48 48 95 49 95 115 51 51 95 117 95 100 101 97 100 98 101 101 102 125 | cvpbPGS{c33xno00_1_f33_h_qrnqorr} |

FLAG = picoCTF{p33kab00\_1\_s33\_u\_deadbeef}

## SoMeta



Diberikan sebuah file png lalu saya menggunakan strings untuk melihat isi yang bisa dibaca di file tersebut dan menemukan flagnya

```
(//0nn  
[WWWooo$  
`pjj  
<c^Vv  
|>_8  
\C.F  
_mmaa!,  
/'onn>  
8H      ]-1]  
\+Wu  
"M2h  
cqq1  
m p?  
8`o||4!`  
t:m6  
<1y0  
t]BZ  
  tEXtArtist  
picoCTF{s0_m3ta_fec06741}  
IEND
```

FLAG = picoCTF{s0\_m3ta\_fec06741}

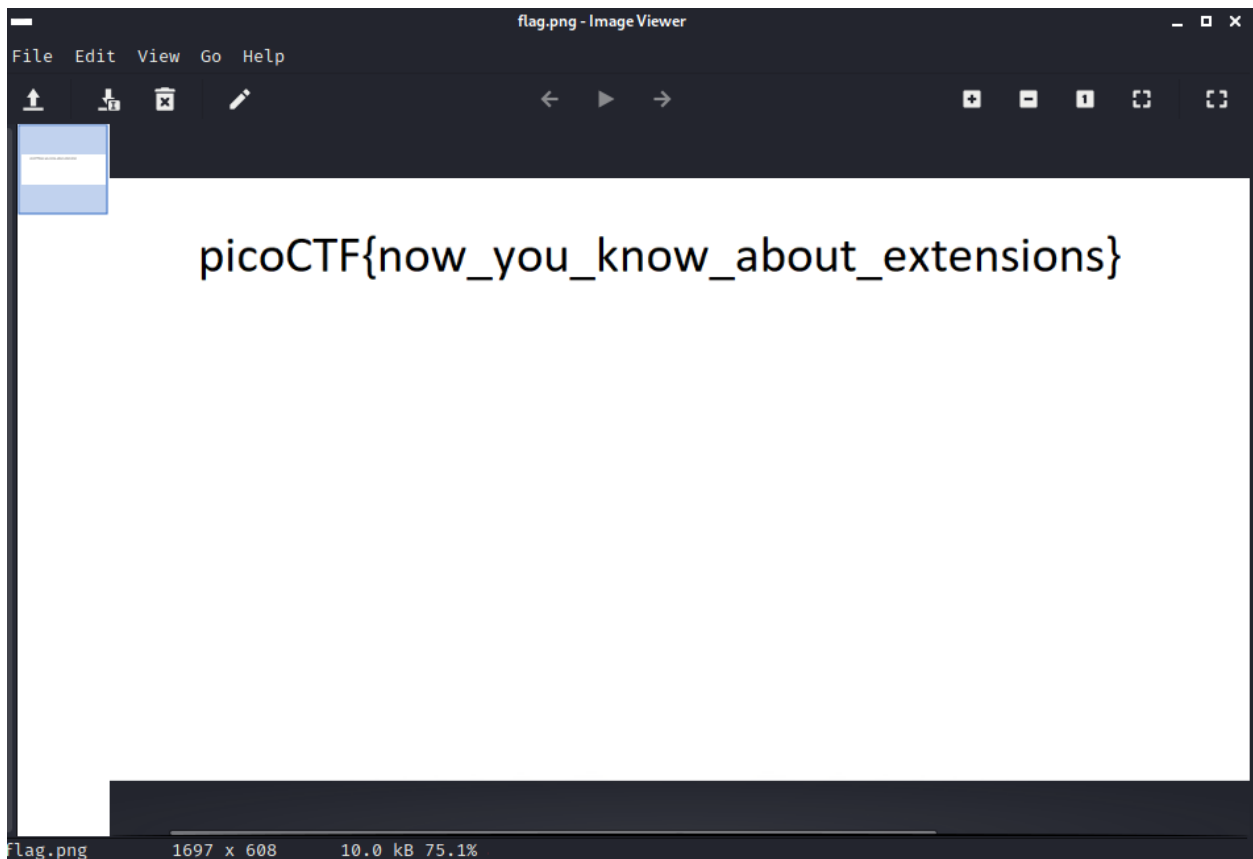
## Extension

Untuk soal kali ini diberikan sebuah file txt, tapi setelah saya menggunakan command file ternyata file tersebut bukan lah sebuah file txt

```
(kali㉿kali)-[~/Belajar/exercise/PICO/extensions]
$ file flag.txt
flag.txt: PNG image data, 1697 x 608, 8-bit/color RGB, non-interlaced

(kali㉿kali)-[~/Belajar/exercise/PICO/extensions]
$
```

Disini saya coba mengganti extension file tersebut dengan png lalu membukanya



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



## Like1000

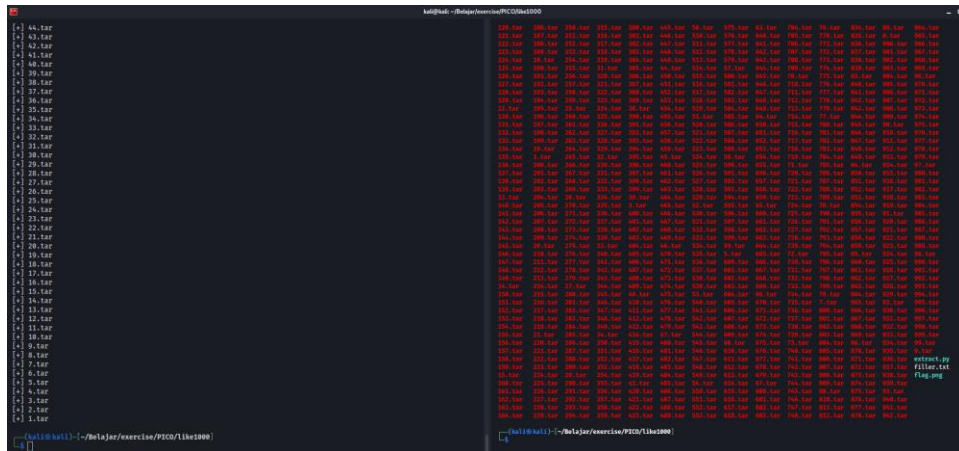
Diberikan sebuah file tar, lalu setelah saya extract ternyata terdapat file tar lagi dengan nama 999.tar disini saya asumsikan bahwa kita harus melakukan extract terhadap file sampai 1000 kali sesuai nama file nya, langsung saja saya buat script python untuk melakukannya

```
#!/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()
```



Setelah terextract sampai 1000 kali lalu saya menemukan file flag.png dan saya buka ternyata ada flagnya





FLAG = picoCTF{10t5\_of\_TAR5}



# [General Skills]

## Based

Based 

 | 200 points 

Tags: Category: General Skills

AUTHOR: ALEX FULTON/DANIEL TUNITIS

Hints

### Description

1 2

To get truly 1337, you must understand different data encodings, such as hexadecimal or binary. Can you get the flag from this program to prove you are on the way to becoming 1337? Connect with `nc jupiter.challenges.picoctf.org 29221`.

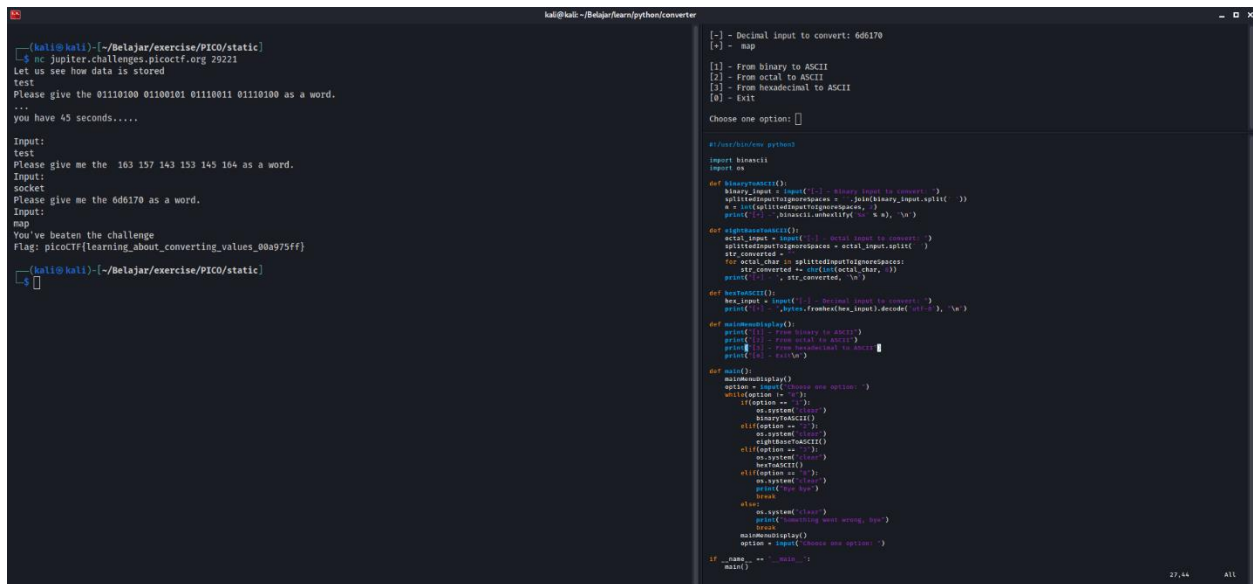
10,333 solves / 14,605 attempts (71%)

 88% Liked 

 picoCTF{FLAG}

Submit Flag

Diberikan soal nc yang berisikan bilangan binary, octal, dan hex lalu diberikan waktu untuk menjawabnya 45 detik, disini saya menggunakan python untuk mempercepat menjawab soalnya



```
kali@kali: ~/Belajar/learn/python/converter
$ nc jupiter.challenges.picoctf.org 29221
Let us see how data is stored
test
Please give the 0110100 01100101 01110011 01110100 as a word.
...
you have 45 seconds.....
Input:
test
Please give me the 163 157 143 153 143 164 as a word.
Input:
socket
Please give me the 606170 as a word.
Input:
map
You've beaten the challenge
Flag: picoCTF{learning_about_converting_values_00a975ff}

kali@kali:~/Belajar/exercise/PICO/static
$
```

```
[-] - Decimal input to convert: 606170
[+] - map

[1] - From binary to ASCII
[2] - From octal to ASCII
[3] - From hexadecimal to ASCII
[0] - Exit

Choose one option:

#user/bin/new_python
import binascii
import os

def hexToBinary():
    binary_input = input("[-] - binary input to convert: ")
    while len(binascii.unhexlify(binary_input.encode('utf-8'))) > 0:
        b = binascii.unhexlify(binary_input.encode('utf-8'))
        print(b)

def octalToBinary():
    octal_input = input("[-] - octal input to convert: ")
    while len(binascii.unhexlify(octal_input.encode('utf-8'))) > 0:
        str_converted = binascii.unhexlify(octal_input.encode('utf-8'))
        str_converted = str_converted.decode('utf-8')
        str_converted = str_converted.encode('utf-8')
        print(str_converted)

def hexToOctal():
    hex_input = input("[-] - hex input to convert: ")
    while len(binascii.unhexlify(hex_input.encode('utf-8'))) > 0:
        str_converted = binascii.unhexlify(hex_input.encode('utf-8'))
        str_converted = str_converted.decode('utf-8')
        str_converted = str_converted.encode('utf-8')
        print(str_converted)

def main():
    while True:
        option = input("Choose one option: ")
        while option != "0":
            if option == "1":
                os.system("clear")
                hexToBinary()
            elif option == "2":
                os.system("clear")
                octalToBinary()
            elif option == "3":
                os.system("clear")
                hexToOctal()
            elif option == "0":
                os.system("clear")
                print("You won!")
                break
            else:
                os.system("clear")
                print("Something went wrong, try!")
        option = input("Choose one option: ")
    if __name__ == "__main__":
        main()
```

FLAG = picoCTF{learning\_about\_converting\_values\_00a975ff}

## Mus1c

```
kali@kali:~/Belajar/exercise/PICO/music$ cat lyrics.txt
Pico's a CTFEEEEEE
my mind is waitin
It's waitin

Put my mind of Pico into This
my flag is not found
put This into my flag
put my flag into Pico

shout Pico
shout Pico
shout Pico

My song's something
put Pico into This

Knock This down, down, down
put This into CTF

shout CTF
my lyric is nothing
Put This without my song into my lyric
Knock my lyric down, down, down

shout my lyric

Put my lyric into This
Put my song with This into my lyric
Knock my lyric down

shout my lyric

Build my lyric up, up ,up

shout my lyric
shout Pico
shout It

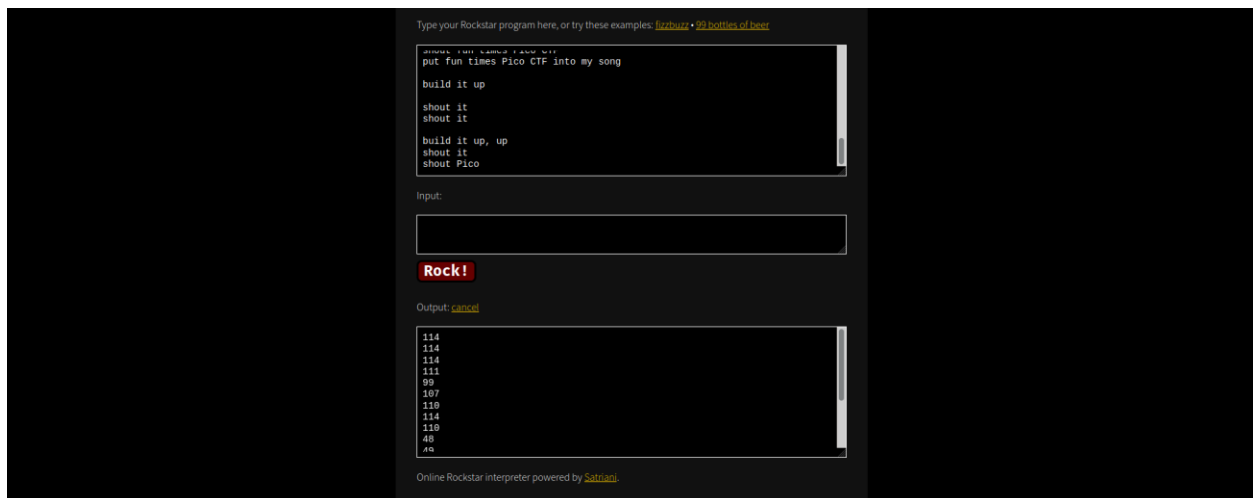
Pico CTF is fun
security is important
Fun is fun
Put security with fun into Pico CTF
Build Fun up
shout fun times Pico CTF
put fun times Pico CTF into my song

build it up

shout it
shout it

build it up, up
shout it
shout Pico
```

Disini kita diberikan file txt yang berisikan lyric, di hint diberikan sebuah petunjuk “Do you think you can master rockstar?” lalu saya menemukan bahwa hint tersebut merujuk kepada Bahasa pemrograman Rockstar lalu saya melakukan compile terhadap file tersebut di website <https://codewithrockstar.com/online>



Output yang diberikan adalah bilangan decimal disini saya menggunakan python untuk menkonversinya dan dibungkus dengan format “picoCTF{flag}”

berikut dibawah ini adalah scriptnya

```
#!/usr/bin/env python
flag = [114,114,114,111,99,107,110,114,110,48,49,49,51,114]
flag = "".join(map(chr,flag))

print "picoCTF{" + flag + "}"
```

Lalu setelah itu saya menjalankan script yang sudah saya buat maka flagnya muncul

```
(kali㉿kali)-[~/Belajar/exercise/PICO/mus1c]
$ ./rockstar.py
picoCTF{rrrocknrn0113r}

(kali㉿kali)-[~/Belajar/exercise/PICO/mus1c]
$
```

FLAG = picoCTF{rrrocknrn0113r}

## [Binary Exploit]