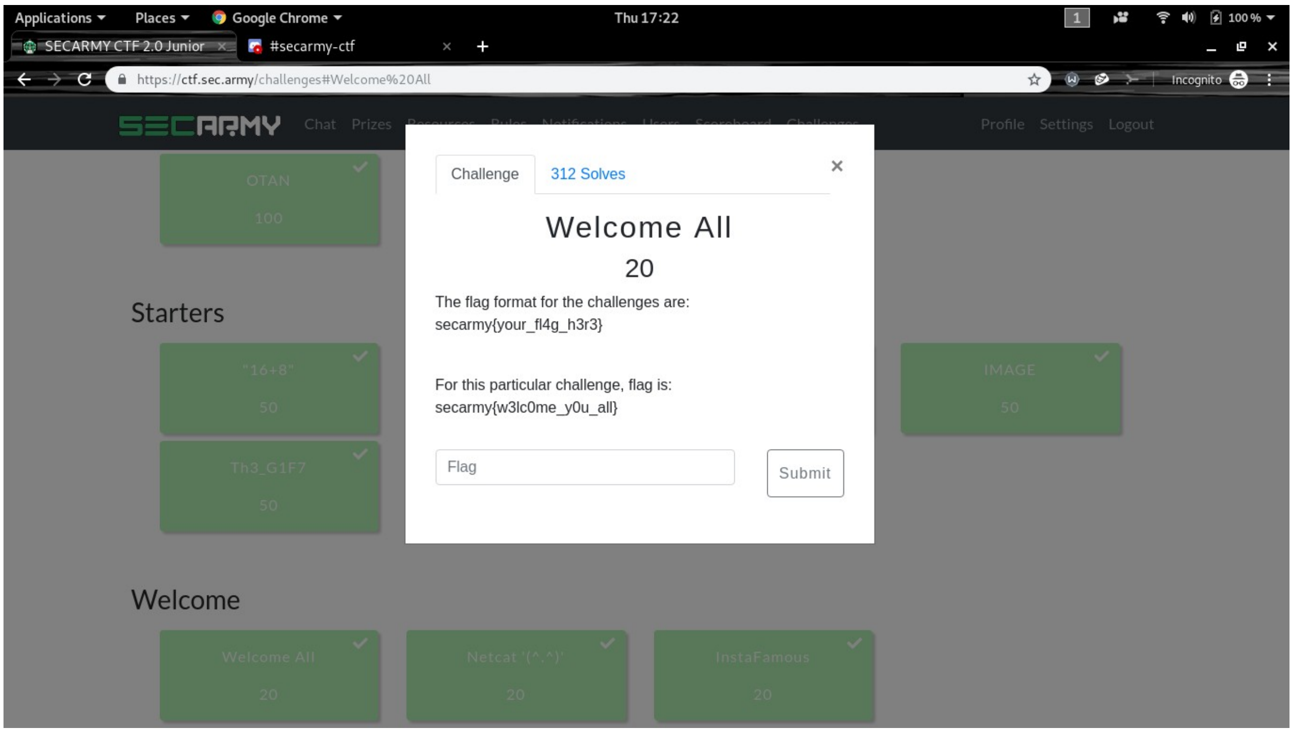
SECARMY CTF 2.0 Junior Writeups

Welcome Challs:

Welcome all:



Welcome All

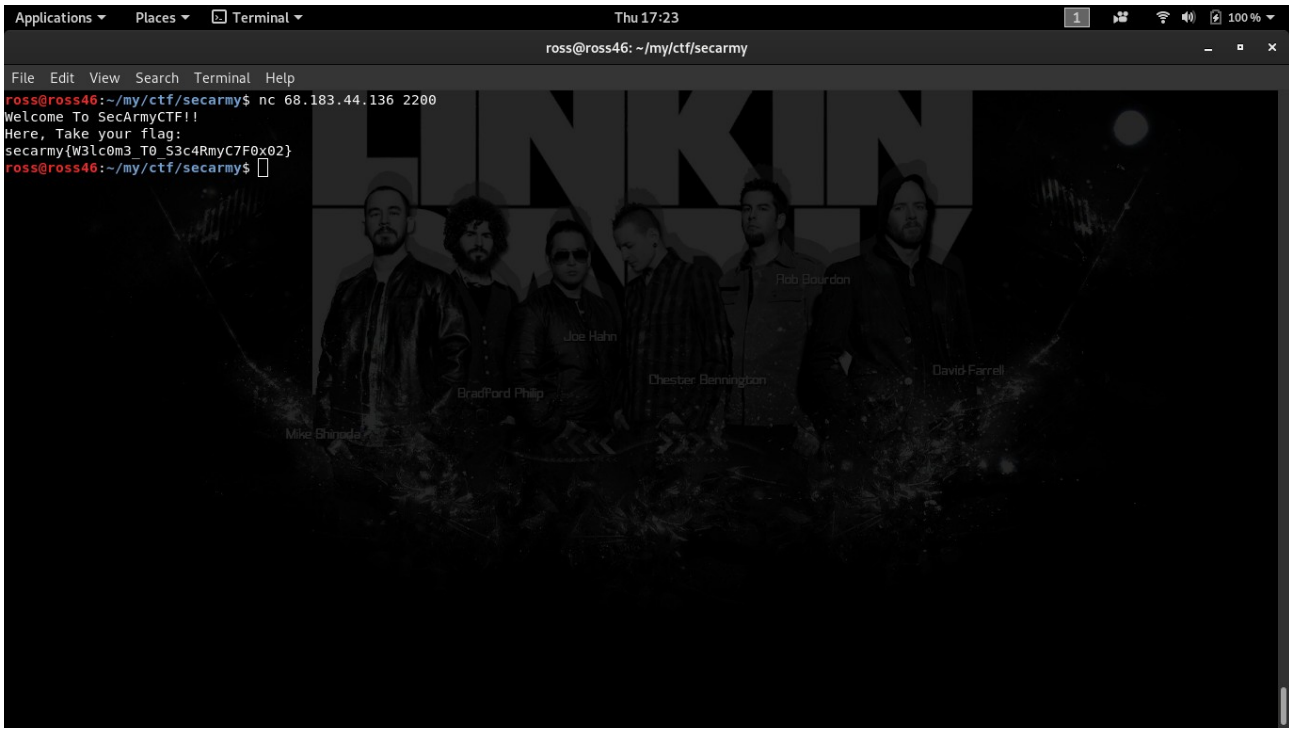
20

The flag format for the challenges are:
secarmy{your_fl4g_h3r3}

For this particular challenge, flag is:
secarmy{w3lc0me_y0u_all}

This was right in front of You

Netcat ‘(^.^)’



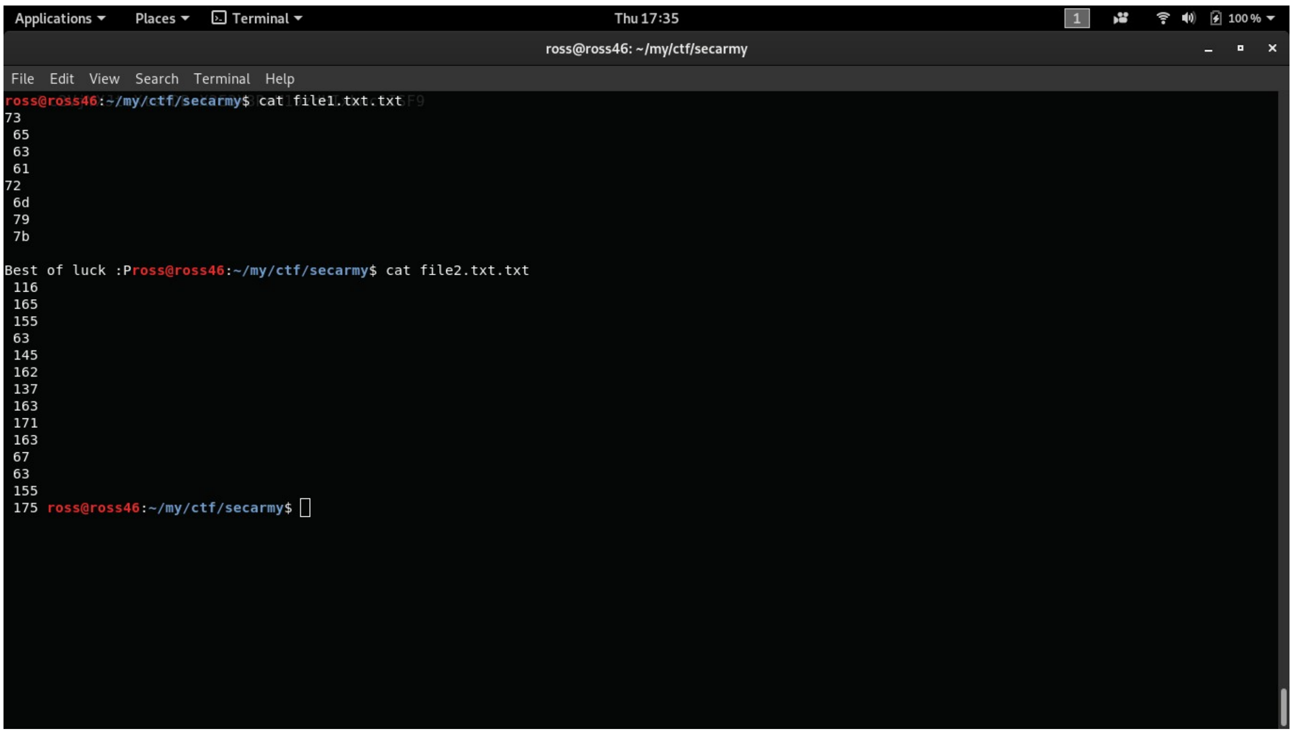
```
ross@ross46:~/my/ctf/secarmy$ nc 68.183.44.136 2200
Welcome To SecArmyCTF!!
Here, Take your flag:
secarmy{W3lc0m3_T0_S3c4RmyC7F0x02}
ross@ross46:~/my/ctf/secarmy$
```

Connect to the IP and Port from Your netcat , you get the flag

Starters:

16+8:

This was eazy, the name had it, hex and octal conversion:
there were two zip diles which had two files:
file1.txt.txt
file2.txt.txt

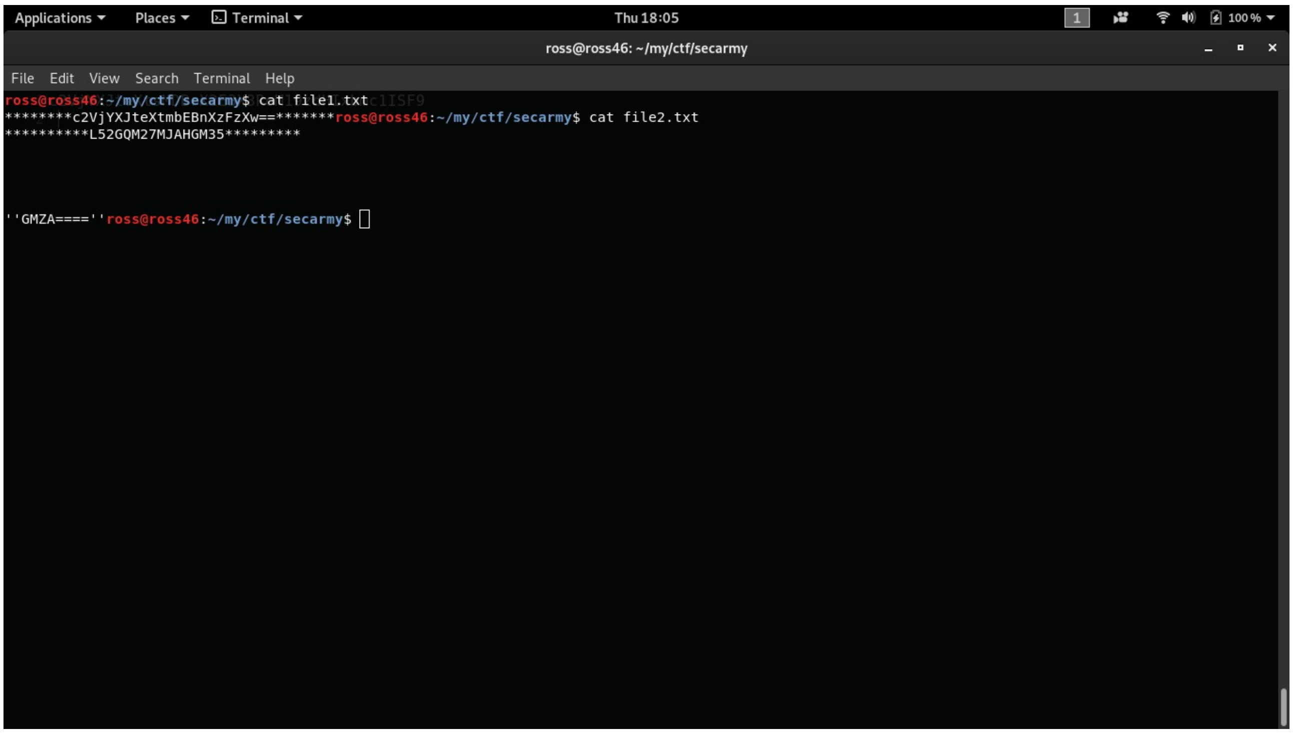

Applications
Places
Terminal
Thu 17:35
100 %
ross@ross46: ~/my/ctf/secarmy
File Edit View Search Terminal Help

```
ross@ross46:~/my/ctf/secarmy$ cat file1.txt.txt
73
 65
 63
 61
72
 6d
 79
 7b

Best of luck :Pross@ross46:~/my/ctf/secarmy$ cat file2.txt.txt
 116
 165
 155
 63
 145
 162
 137
 163
 171
 163
 67
 63
 155
 175 ross@ross46:~/my/ctf/secarmy$ 
```

From the name: 1st file had hex values, 2nd file had octal values

converting the hex to text : secarmy{
converting the oct to text : Num3er_sys73m}
so the flag is: secarmy{Num3er_sys73m}

Die Basis:
there was one zip file which had two text files:
file1.txt file2.txt

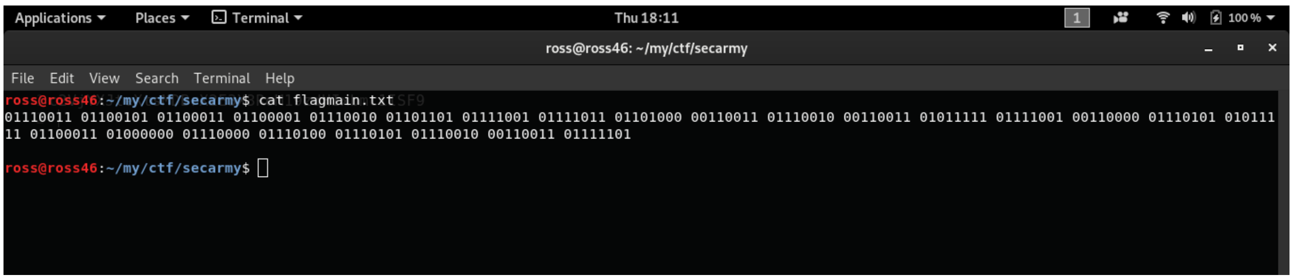

Applications
Places
Terminal
Thu 18:05
100 %
ross@ross46: ~/my/ctf/secarmy
File Edit View Search Terminal Help

```
ross@ross46:~/my/ctf/secarmy$ cat file1.txt
********c2VjYXJteXtmbEBnXzFzXw==*******ross@ross46:~/my/ctf/secarmy$ cat file2.txt
**********L52GQM27MJAHGM35*********



''GMZA====''ross@ross46:~/my/ctf/secarmy$ 
```

file1.txt: c2VjYXJteXtmbEBnXzFzXw== , this was of base64 format
file2.txt: L52GQM27MJAHGM35GMZA==== , this was of base32 format

joining the text values of both the format we get: secarmy{fl@g_1s__th3_b@s3}32
but the flag is: secarmy{fl@g_1s_th3_b@s3}

Eazy capture:
unzipping the flagmin.zip we get flagmain.txt


Applications
Places
Terminal
Thu 18:11
100 %
ross@ross46: ~/my/ctf/secarmy
File Edit View Search Terminal Help

```
ross@ross46:~/my/ctf/secarmy$ cat flagmain.txt
01110011 01100101 01100011 01100001 01110010 01101101 01111001 01111011 01101000 00110011 01110010 00110011 01011111 01111001 00110000 01110101 010111
11 01100011 01000000 01110000 01110100 01110101 01110010 00110011 01111101

ross@ross46:~/my/ctf/secarmy$ 
```

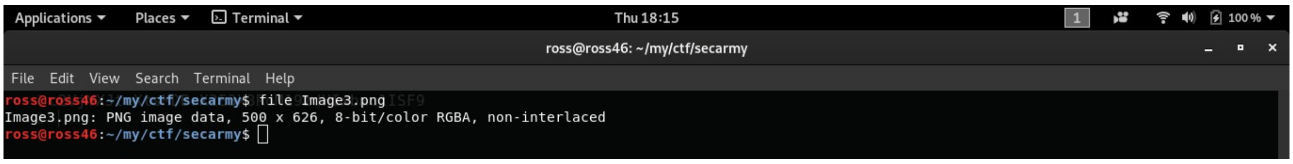
01110011 01100101 01100011 01100001 01110010 01101101 01111001 01111011 01101000 00110011 01110010 00110011 01011111 01111001 00110000 01110101 01011111 01100011 01000000 01110000 01110100 01110101 01110010 00110011 01111101

It is in binary format.

Converting into text we get: secarmy{h3r3_y0u_c@ptur3}

IMAGE:

unzipping file.zip we get Image3.png



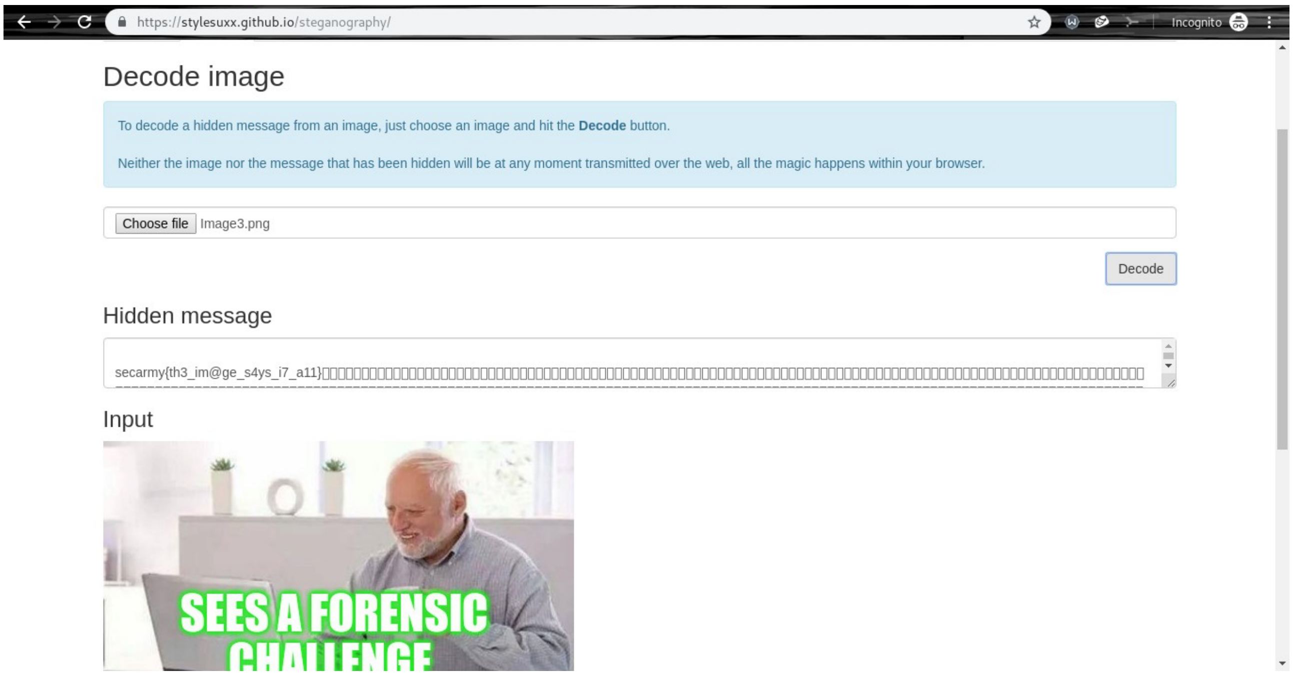
```
ross@ross46:~/my/ctf/secarmy$ file Image3.png
Image3.png: PNG image data, 500 x 626, 8-bit/color RGBA, non-interlaced
ross@ross46:~/my/ctf/secarmy$
```

file type is also PNG image.

This link is very handy in many challenges:
https://stylesuxx.github.io/steganography/

uploading the image to the url we get:

Flag: secarmy{th3_im@ge_s4ys_i7_a11}



https://stylesuxx.github.io/steganography/

Decode image

To decode a hidden message from an image, just choose an image and hit the **Decode** button.

Neither the image nor the message that has been hidden will be at any moment transmitted over the web, all the magic happens within your browser.

Choose file Image3.png

Decode

Hidden message

secarmy{th3_im@ge_s4ys_i7_a11}

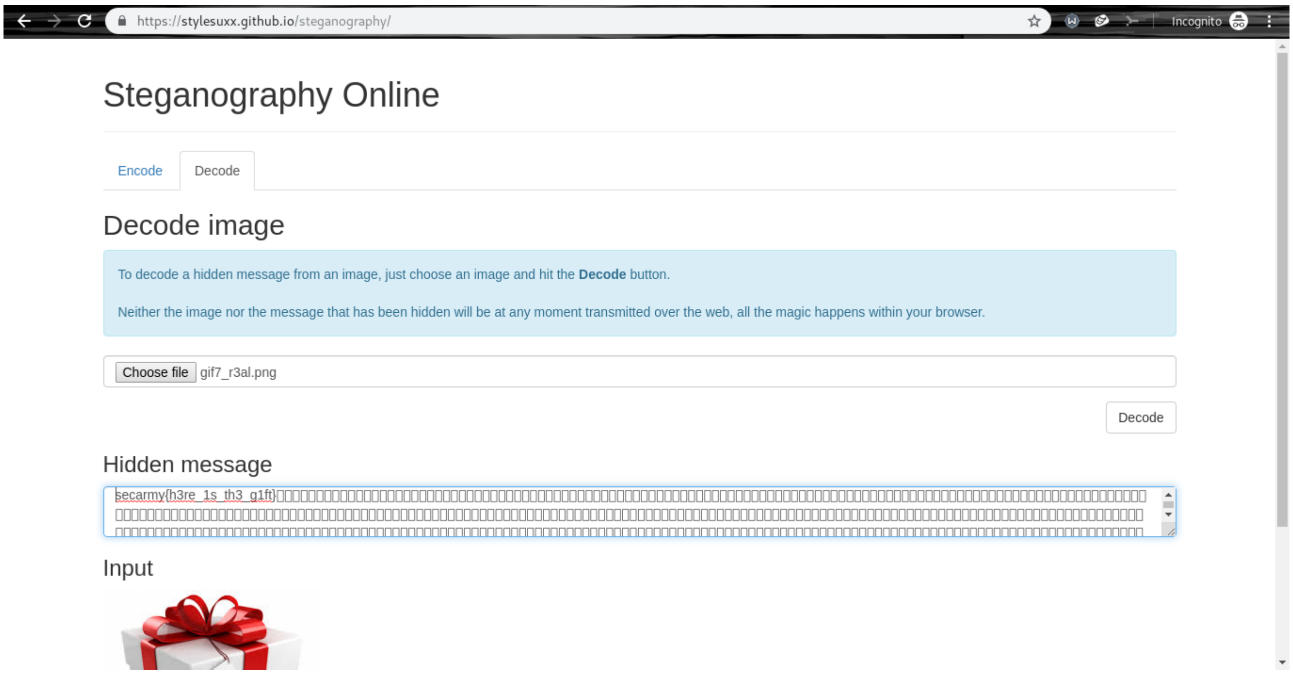
Input

Th3_G1F7:

This is similar to the previous challenge:

same site: https://stylesuxx.github.io/steganography/

Flag: secarmy{h3re_1s_th3_g1ft}



https://stylesuxx.github.io/steganography/

Steganography Online

Decode image

To decode a hidden message from an image, just choose an image and hit the **Decode** button.

Neither the image nor the message that has been hidden will be at any moment transmitted over the web, all the magic happens within your browser.

Choose file gif7_r3al.png

Hidden message

secarmy{h3re_1s_th3_g1ft}

Input

Cryptography

OTAN:
It has a hint guess my name: Reverse the challenge name we get NATO,
Extracting the hint.zip file we get two files: hint.txt and and SVG image with number 2.
The number two indicates that it is encrypted twice.

Hint.txt has:
 ==UNIFORM GOLF ECHO CHARLIE TANGO OSCAR ALPHA PAPA CHARLIE VICTOR QUEBEK ROMEO JULIETT QUEBEK PAPA GOLF VICTOR KILO ECHO UNIFORM==

Site: https://cryptii.com

Type: spelling alphabets
subtype: NATO/ICAO

Trying to decrypt we get: ==ugectoALPHApcvQUEBEKrjQUEBEKpgvkeu==

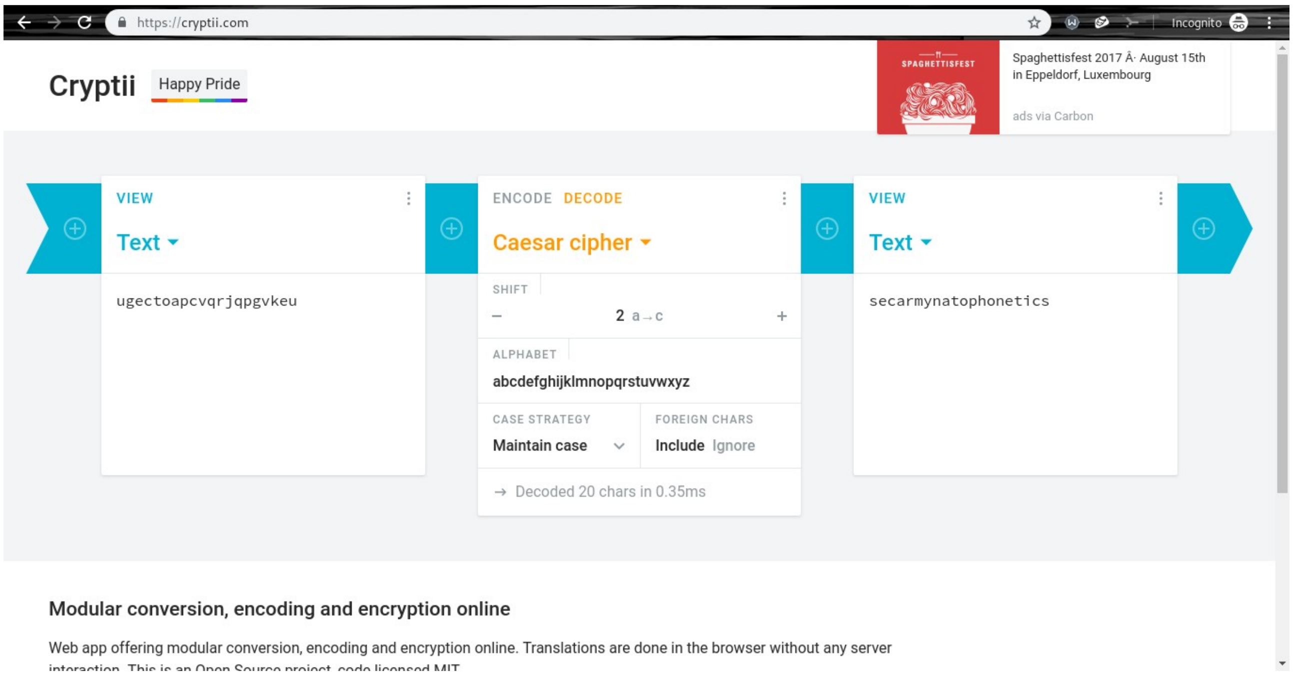
Note that Alpha and QUEBEK remains the same.

Opening https://en.wikipedia.org/wiki/NATO_phonetic_alphabet
we get the entire list of substitution

Correcting the spelling of ALPHA and QUEBEK we get:
==UNIFORM GOLF ECHO CHARLIE TANGO OSCAR ALFA PAPA CHARLIE VICTOR QUEBEC ROMEO JULIETT QUEBEC PAPA GOLF VICTOR KILO ECHO UNIFORM==

The converted form: ==ugectoapcvqrjqpgvkeu==

Now we try substitution Cipher (+2), we get: secarmynatophonetics



```
ugectoapcvqrjqpgvkeu
```

SHIFT: 2 a→c
ALPHABET: abcdefghijklmnopqrstuvwxyz
CASE STRATEGY: Maintain case
FOREIGN CHARS: Include Ignore
→ Decoded 20 chars in 0.35ms

```
secarmynatophonetics
```

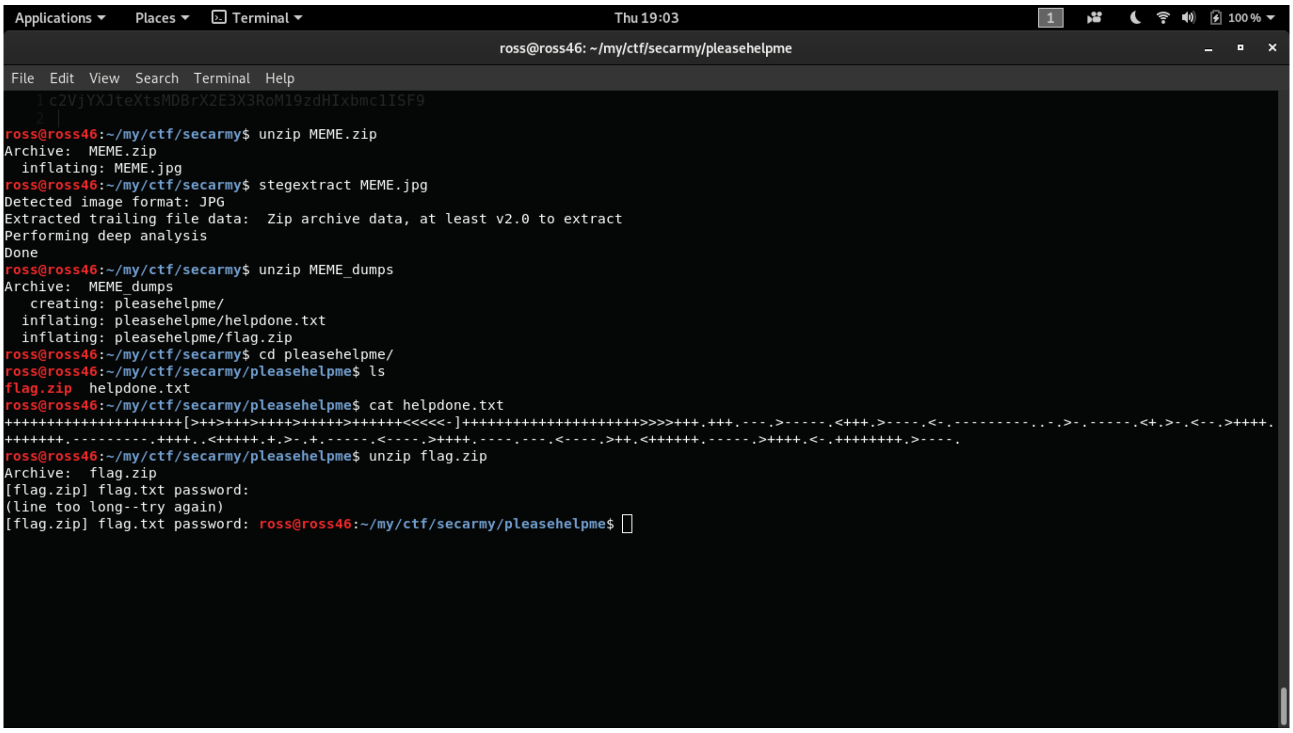
Modular conversion, encoding and encryption online

Web app offering modular conversion, encoding and encryption online. Translations are done in the browser without any server interaction. This is an Open Source project, code licensed MIT

Flag: secarmy{natophonetics}

Misc

Bruteforce:



```
c2VjYXJteXtsMDBrX2E3X3RoM19zdHIxbmc1ISF9

ross@ross46:~/my/ctf/secarmy$ unzip MEME.zip
Archive:  MEME.zip
  inflating: MEME.jpg
ross@ross46:~/my/ctf/secarmy$ stegextract MEME.jpg
Detected image format: JPG
Extracted trailing file data:  Zip archive data, at least v2.0 to extract
Performing deep analysis
Done
ross@ross46:~/my/ctf/secarmy$ unzip MEME_dumps
Archive:  MEME_dumps
   creating: pleasehelpme/
  inflating: pleasehelpme/helpdone.txt
  inflating: pleasehelpme/flag.zip
ross@ross46:~/my/ctf/secarmy$ cd pleasehelpme/
ross@ross46:~/my/ctf/secarmy/pleasehelpme$ ls
flag.zip  helpdone.txt
ross@ross46:~/my/ctf/secarmy/pleasehelpme$ cat helpdone.txt
++++++++++++++++++++[>++>+++>++++>+++++>++++++<<<<<-]++++++++++++++++++++>>>>+++.+++.---.>-----.<+++.>----.<-.---------..-.>-.-----.<+.>-.<--.>++++.
+++++++.---------.++++..<+++++.+.>-.+.-----.<----.>++++.----.---.<----.>++.<++++++.-----.>++++.<-.++++++++.>----.
ross@ross46:~/my/ctf/secarmy/pleasehelpme$ unzip flag.zip
Archive:  flag.zip
[flag.zip] flag.txt password:
(line too long--try again)
[flag.zip] flag.txt password: ross@ross46:~/my/ctf/secarmy/pleasehelpme$ 
```

We get a zip file MEME.zip:

Extracting it we get MEME.jpg

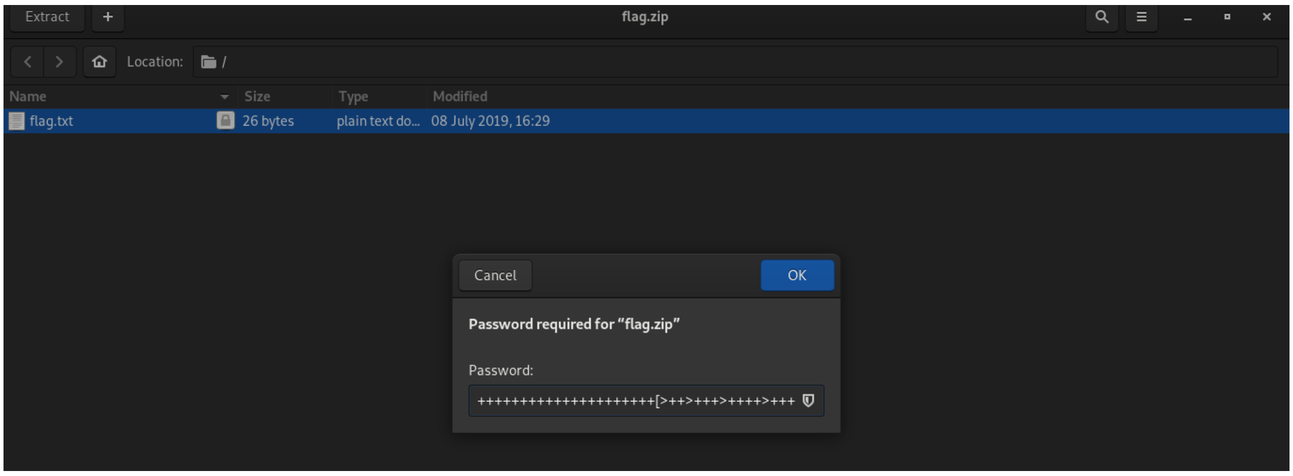
Using stegextract we recover a zip file called MEME_dumps

Unzip it we find a folder pleasehelpme which consists of flag.zip and helpdone.txt

helpdone.txt:
++++++++++++++++++++[>++>+++>++++>+++++>++++++<<<<<-]+++++++++++++++++++++>>>>+++.+++.---.>-----.<+++.>----.<-.---------..-.>-.-----.<+.>-.<--.>++++.+++++++.---------.++++..<+++++.+.>-.+.-----.<----.>++++.----.---.<----.>++.<++++++.-----.>++++.<-.++++++++.>----.

flag.zip is password protected. So using the hint inside helpdone.txt (brainfuck code) we try to extract the flag.zip but unfortunately it does not work out.

Next: GUI mode: Flag: secarmy{h3lp_m3_t0_unz1p}



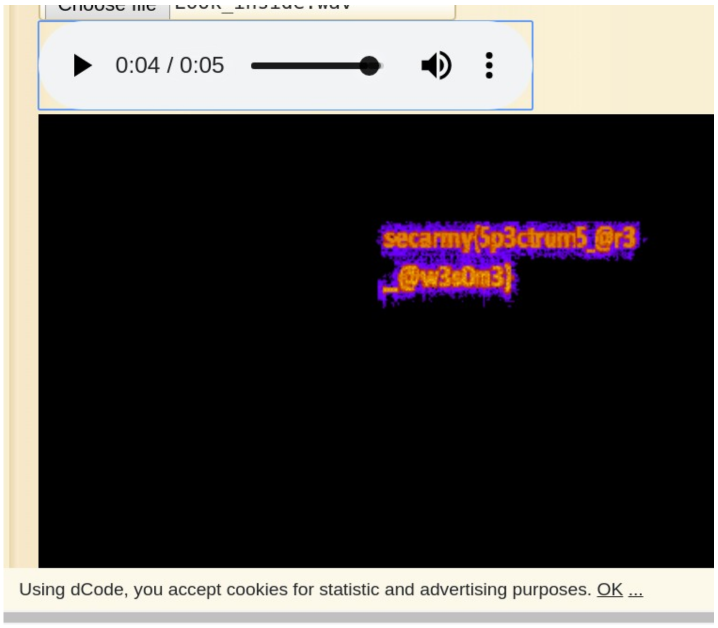
```
flag.txt    26 bytes    plain text do...    08 July 2019, 16:29

Password required for "flag.zip"

Password:
++++++++++++++++++++[>++>+++>++++>+++
```

Directories:
Question: It is a type of illusionary filesystem. It does not exist on a disk. Can U name it ?

Flag: secarmy{/proc}

Look inside:
we get a Look_inside.wav file

we perform a spectral analysis in https://dcode.fr (trust me and save that link)



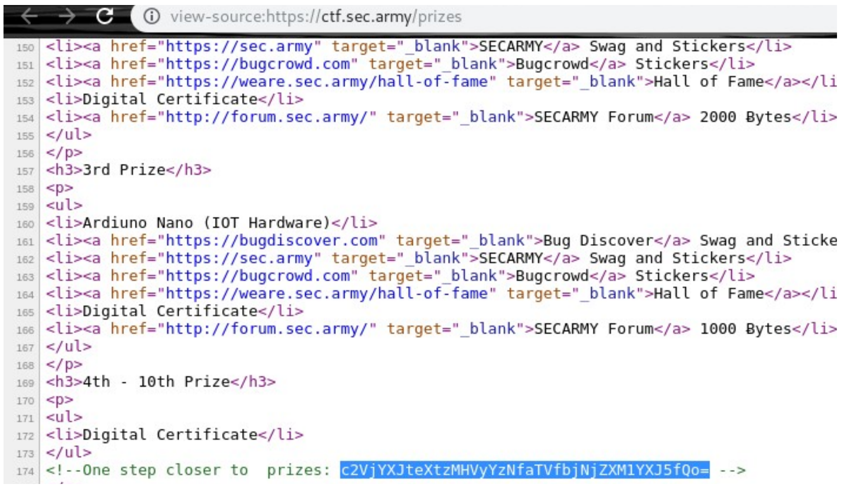
0:04 / 0:05
secarmy{5p3ctrum5_@r3
_@w3s0m3}
Using dCode, you accept cookies for statistic and advertising purposes. OK ...

Flag: secarmy{5p3ctrum5_@r3_@w3s0m3}

Prizes:

link: view-source:https://ctf.sec.army/prizes

open the source code you get: `c2VjYXJteXtzMHVyYzNfaTVfbjNjZXM1YXJ5fQo=`

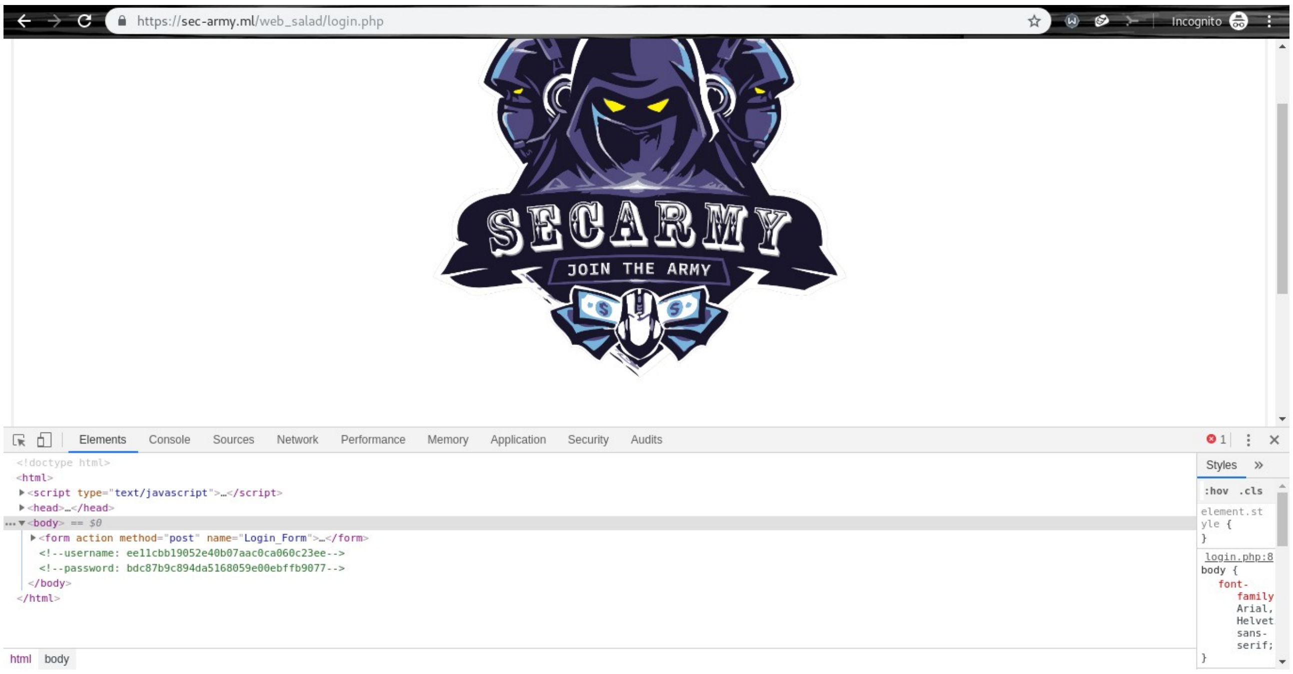


view-source:https://ctf.sec.army/prizes

```
<li><a href="https://sec.army" target="_blank">SECARMY</a> Swag and Stickers</li>
<li><a href="https://bugcrowd.com" target="_blank">Bugcrowd</a> Stickers</li>
<li><a href="https://weare.sec.army/hall-of-fame" target="_blank">Hall of Fame</a></li
<li>Digital Certificate</li>
<li><a href="http://forum.sec.army/" target="_blank">SECARMY Forum</a> 2000 Bytes</li>
</ul>
</p>
<h3>3rd Prize</h3>
<p>
<ul>
<li>Ardiuno Nano (IOT Hardware)</li>
<li><a href="https://bugdiscover.com" target="_blank">Bug Discover</a> Swag and Sticke
<li><a href="https://sec.army" target="_blank">SECARMY</a> Swag and Stickers</li>
<li><a href="https://bugcrowd.com" target="_blank">Bugcrowd</a> Stickers</li>
<li><a href="https://weare.sec.army/hall-of-fame" target="_blank">Hall of Fame</a></li
<li>Digital Certificate</li>
<li><a href="http://forum.sec.army/" target="_blank">SECARMY Forum</a> 1000 Bytes</li>
</ul>
</p>
<h3>4th - 10th Prize</h3>
<p>
<ul>
<li>Digital Certificate</li>
</ul>
<!--One step closer to  prizes: c2VjYXJteXtzMHVyYzNfaTVfbjNjZXM1YXJ5fQo= -->
```

base64 to text: secarmy{s0urc3_i5_n3ces5ary} web_salad:

open the console you can see two hashes:



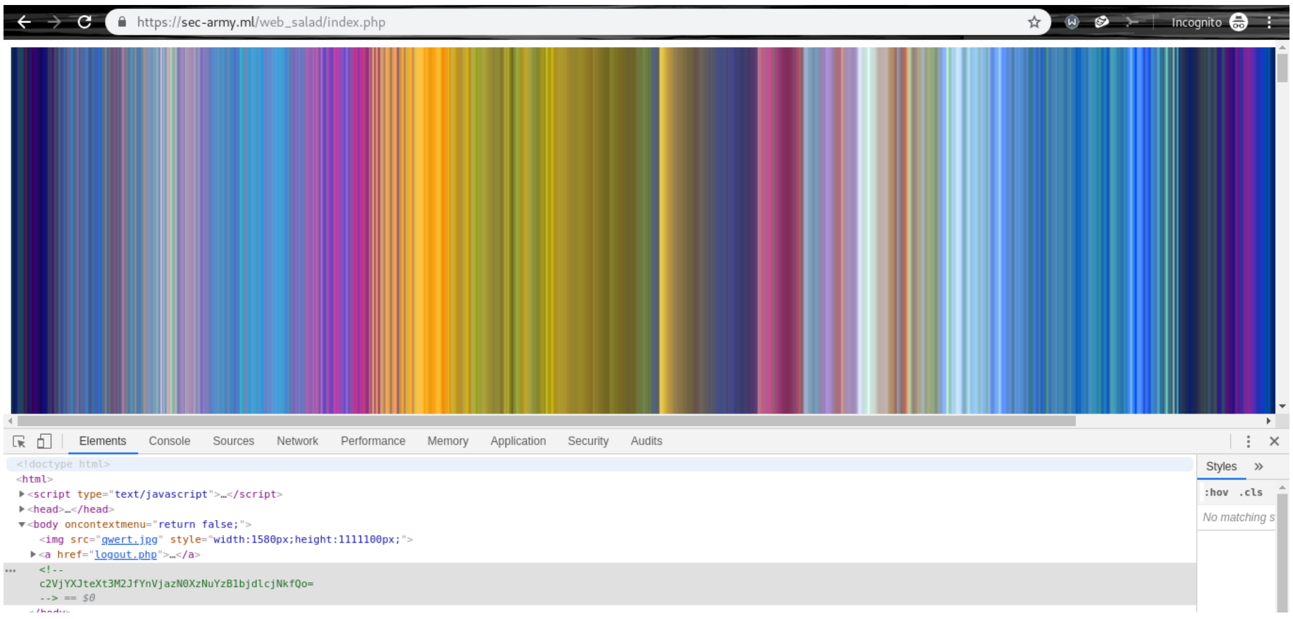
```
https://sec-army.ml/web_salad/login.php
```

```
<!doctype html>
<html>
 ▶<script type="text/javascript">…</script>
 ▶<head>…</head>
…▼<body> == $0
   ▶<form action method="post" name="Login_Form">…</form>
    <!--username: ee11cbb19052e40b07aac0ca060c23ee-->
    <!--password: bdc87b9c894da5168059e00ebffb9077-->
  </body>
</html>
```

<!--username: ee11cbb19052e40b07aac0ca060c23ee-->
<!--password: bdc87b9c894da5168059e00ebffb9077-->

it is md5 hash:
decrypt it we get
username: user
password: password1234

with the console open login: we get c2VjYXJteXt3M2JfYnVjazN0XzNuYzB1bjdlcjNkfQo=
Flag after decoding base64 : secarmy{w3b_buck3t_3nc0un7er3d}



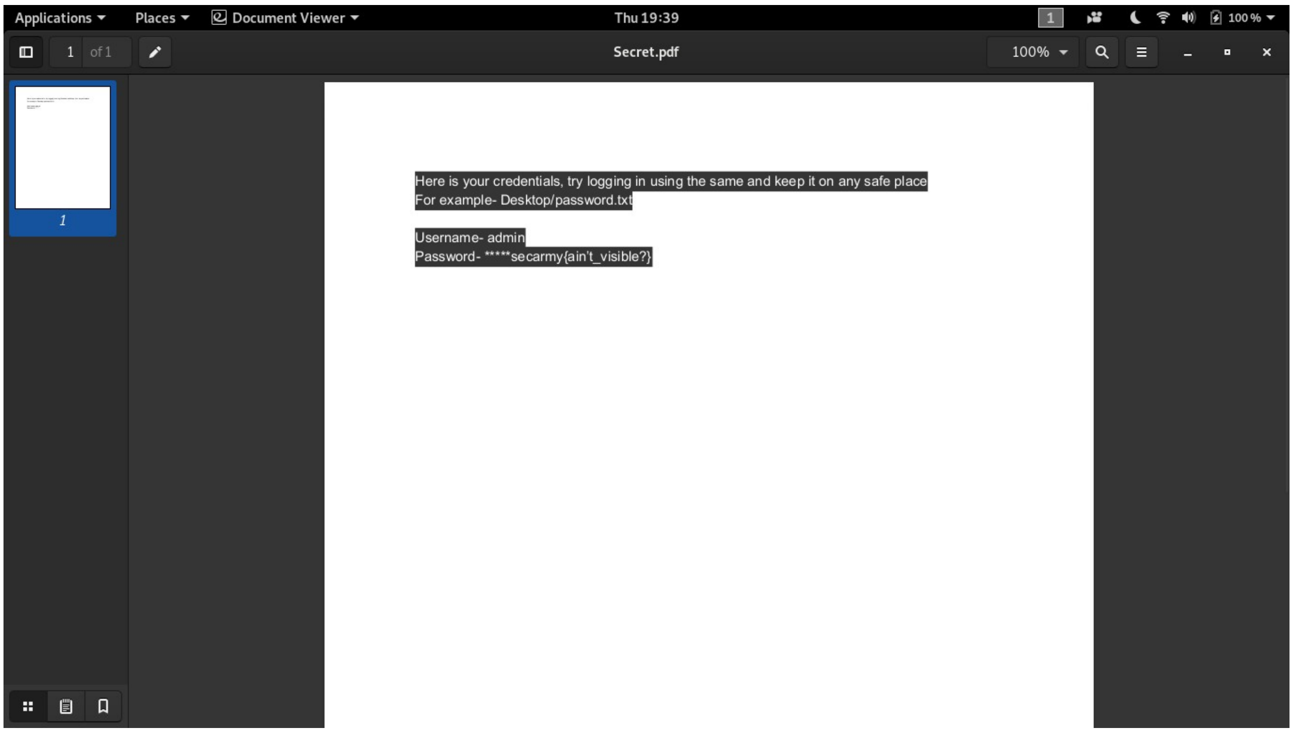
```
https://sec-army.ml/web_salad/index.php
```

```
<!doctype html>
<html>
 ▶<script type="text/javascript">…</script>
 ▶<head>…</head>
 ▼<body oncontextmenu="return false;">
    <img src="qwert.jpg" style="width:1580px;height:1111100px;">
   ▶<a href="logout.php">…</a>
…   <!--
    c2VjYXJteXt3M2JfYnVjazN0XzNuYzB1bjdlcjNkfQo=
    --> == $0
```

Forensics

secret:

open the pdf and press ctrl+a u have the flag: secarmy{ain’t_visible?}



Here is your credentials, try logging in using the same and keep it on any safe place
For example- Desktop/password.txt

Username- admin
Password- *****secarmy{ain't_visible?}

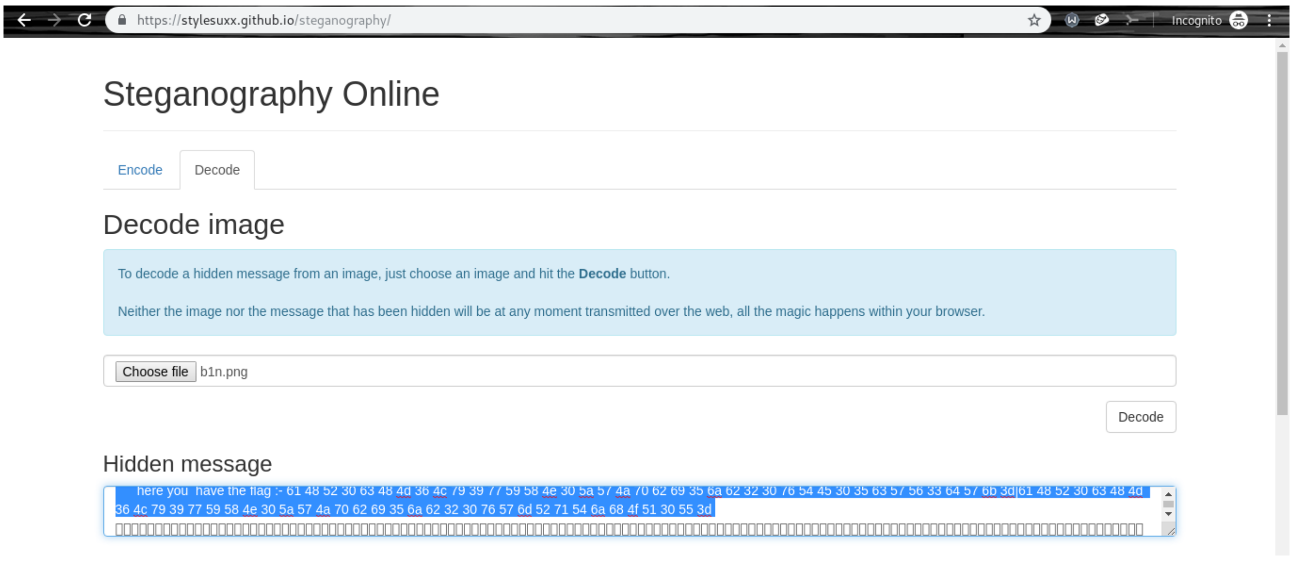
The_B1N:

We get a zip file with b1n.png and bin1.jpg

open: https://stylesuxx.github.io/steganography/

upload b1n.png

we get :
here you have the flag :- 61 48 52 30 63 48 4d 36 4c 79 39 77 59 58 4e 30 5a 57 4a 70 62 69 35 6a 62 32 30 76 54 45 30 35 63 57 56 33 64 57 6b 3d|61 48 52 30 63 48 4d 36 4c 79 39 77 59 58 4e 30 5a 57 4a 70 62 69 35 6a 62 32 30 76 57 6d 52 71 54 6a 68 4f 51 30 55 3d



Steganography Online

Decode image

To decode a hidden message from an image, just choose an image and hit the **Decode** button.

Neither the image nor the message that has been hidden will be at any moment transmitted over the web, all the magic happens within your browser.

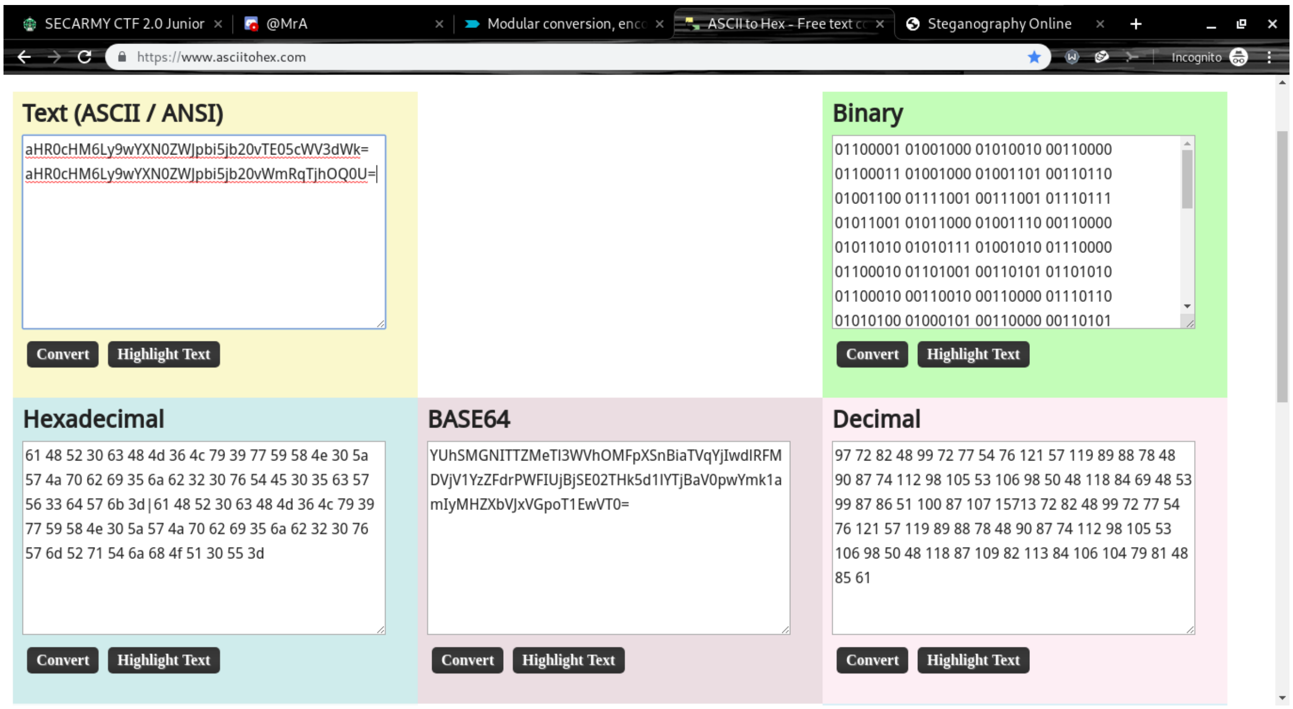
b1n.png

Hidden message

here you have the flag :- 61 48 52 30 63 48 4d 36 4c 79 39 77 59 58 4e 30 5a 57 4a 70 62 69 35 6a 62 32 30 76 54 45 30 35 63 57 56 33 64 57 6b 3d|61 48 52 30 63 48 4d 36 4c 79 39 77 59 58 4e 30 5a 57 4a 70 62 69 35 6a 62 32 30 76 57 6d 52 71 54 6a 68 4f 51 30 55 3d

It is in hex format, decoding it gives us two base64 text:

aHR0cHM6Ly9wYXN0ZWJpbi5jb20vTE05cWV3dWk=
aHR0cHM6Ly9wYXN0ZWJpbi5jb20vWmRqTjhOQ0U=


SECARMY CTF 2.0 Junior | @MrA | Modular conversion, enc | ASCII to Hex - Free text c | Steganography Online

https://www.asciitohex.com

Text (ASCII / ANSI)

```
aHR0cHM6Ly9wYXN0ZWJpbi5jb20vTE05cWV3dWk=
aHR0cHM6Ly9wYXN0ZWJpbi5jb20vWmRqTjhOQ0U=
```

Binary

```
01100001 01001000 01010010 00110000
01100011 01001000 01001101 00110110
01001100 01111001 00111001 01110111
01011001 01011000 01001110 00110000
01011010 01010111 01001010 01110000
01100010 01101001 00110101 01101010
01100010 00110010 00110000 01110110
01010100 01000101 00110000 00110101
```

Hexadecimal

```
61 48 52 30 63 48 4d 36 4c 79 39 77 59 58 4e 30 5a
57 4a 70 62 69 35 6a 62 32 30 76 54 45 30 35 63 57
56 33 64 57 6b 3d|61 48 52 30 63 48 4d 36 4c 79 39
77 59 58 4e 30 5a 57 4a 70 62 69 35 6a 62 32 30 76
57 6d 52 71 54 6a 68 4f 51 30 55 3d
```

BASE64

```
YUhSMGNITTZMeTI3WVhOMFpXSnBiaTVqYjIwdlRFM
DVjV1YzZFdrPWFIUjBjSE02THk5d1lYTjBaV0pwYmk1a
mIyMHZXbVJxVGpoT1EwVT0=
```

Decimal

```
97 72 82 48 99 72 77 54 76 121 57 119 89 88 78 48
90 87 74 112 98 105 53 106 98 50 48 118 84 69 48 53
99 87 86 51 100 87 107 15713 72 82 48 99 72 77 54
76 121 57 119 89 88 78 48 90 87 74 112 98 105 53
106 98 50 48 118 87 109 82 113 84 106 104 79 81 48
85 61
```

converting the first string: https://pastebin.com/LM9qewui

follow it: secarmy{c0ngrats_y0u_h@v3_th3_fl@g} (wrong flag)

converting the second string: https://pastebin.com/ZdjN8NCE

follow it: secarmy{PAST3_B1N_H@S_S0LUT10N}

Flag: secarmy{PAST3_B1N_H@S_S0LUT10N}

Binary/Reverse

Stringy:

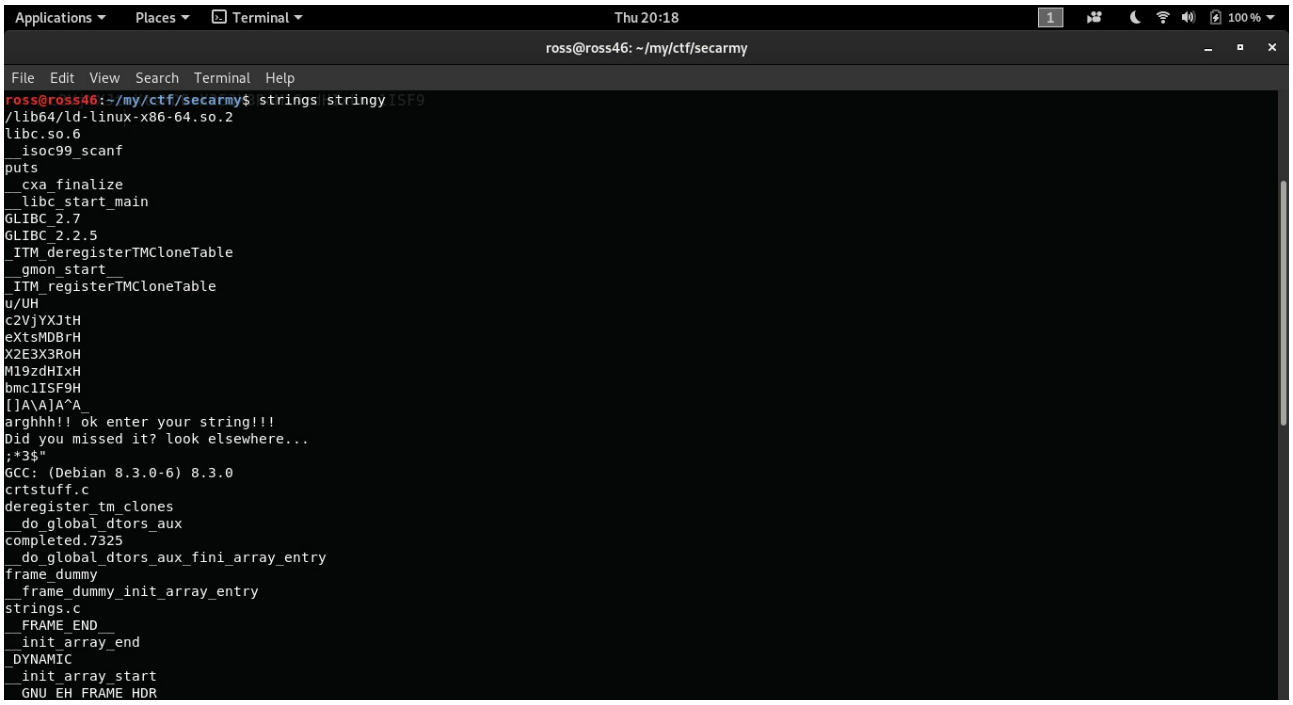
This challenge gives the hint of what to use: strings

strings stringy gives all the readable text in it

what catches our attention is these strings:
c2VjYXJtH
eXtsMDBrH
X2E3X3RoH
M19zdHIxH
bmc1ISF9H

Remove the trailing H and join them, you get a base64 encoded string:
c2VjYXJteXtsMDBrX2E3X3RoM19zdHIxbmc1ISF9

Decoding it gives: secarmy{l00k_a7_th3_str1ng5!!}



```
ross@ross46:~/my/ctf/secarmy$ strings stringy
/lib64/ld-linux-x86-64.so.2
libc.so.6
__isoc99_scanf
puts
__cxa_finalize
__libc_start_main
GLIBC_2.7
GLIBC_2.2.5
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
u/UH
c2VjYXJtH
eXtsMDBrH
X2E3X3RoH
M19zdHIxH
bmc1ISF9H
[]A\A]A^A_
arghhh!! ok enter your string!!!
Did you missed it? look elsewhere...
;*3$"
GCC: (Debian 8.3.0-6) 8.3.0
crtstuff.c
deregister_tm_clones
__do_global_dtors_aux
completed.7325
__do_global_dtors_aux_fini_array_entry
frame_dummy
__frame_dummy_init_array_entry
strings.c
__FRAME_END__
__init_array_end
_DYNAMIC
__init_array_start
__GNU_EH_FRAME_HDR
```

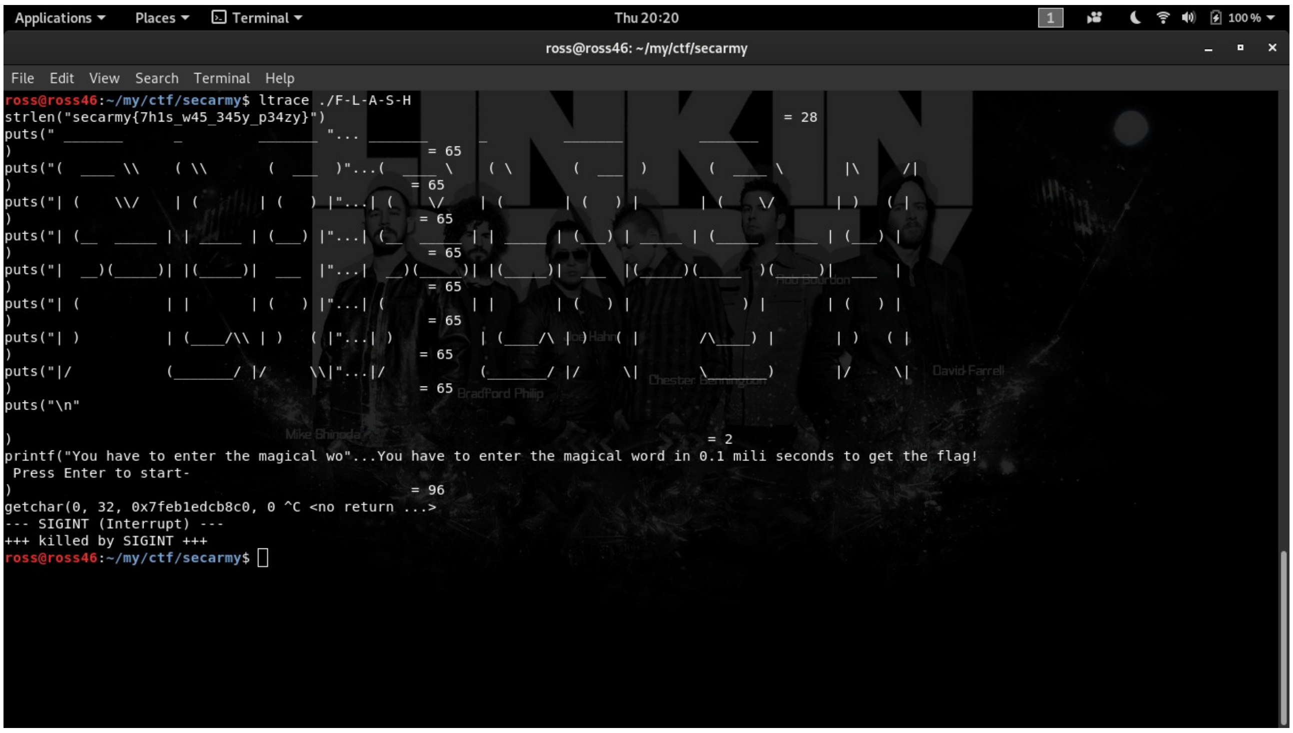
Flag: secarmy{l00k_a7_th3_str1ng5!!}

F-L-A-S-H:

This is by far the eaziest one which we all overlooked,

ltrace ./F-L-A-S-H and you have the flag.

Flag: secarmy{7h1s_w45_345y_p34zy}


Applications
Places
Terminal
Thu 20:20
ross@ross46: ~/my/ctf/secarmy
File Edit View Search Terminal Help

```
ross@ross46:~/my/ctf/secarmy$ ltrace ./F-L-A-S-H
strlen("secarmy{7h1s_w45_345y_p34zy}")                                                      = 28
puts(" _______      _         _______ "... _______      _         _______         _______
)                                                 = 65
puts("(  ____ \\    ( \\       (  ___  )"...(  ____ \    ( \       (  ___  )       (  ____ \       |\     /|
)                                               = 65
puts("| (    \\/    | (       | (   ) |"...| (    \/    | (       | (   ) |       | (    \/       | )   ( |
)                                                = 65
puts("| (__  _____ | | _____ | (___) |"...| (__  _____ | | _____ | (___) | _____ | (_____  _____ | (___) |
)                                                 = 65
puts("|  __)(_____)| |(_____)|  ___  |"...|  __)(_____)| |(_____)|  ___  |(_____)(_____  )(_____)|  ___  |
)                                                 = 65
puts("| (          | |       | (   ) |"...| (          | |       | (   ) |             ) |       | (   ) |
)                                                 = 65
puts("| )          | (____/\\ | )   ( |"...| )          | (____/\ | )   ( |       /\____) |       | )   ( |
)                                                = 65
puts("|/           (_______/ |/     \\|"...|/           (_______/ |/     \|       \_______)       |/     \|
)                                                = 65
puts("\n"

)                                                                                  = 2
printf("You have to enter the magical wo"...You have to enter the magical word in 0.1 mili seconds to get the flag!
 Press Enter to start-
)                                               = 96
getchar(0, 32, 0x7feb1edcb8c0, 0 ^C <no return ...>
--- SIGINT (Interrupt) ---
+++ killed by SIGINT +++
ross@ross46:~/my/ctf/secarmy$ 
```