3 S U M

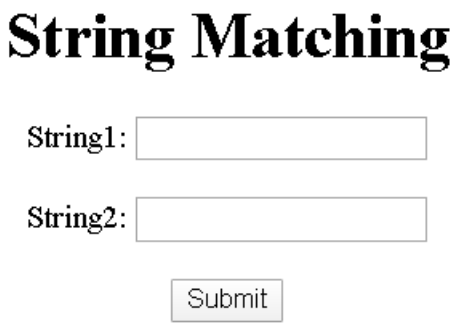
Acai – Fernanda Darmasaputra

Aseng – Felix Alexander

c0urage – Stephanley Herman

WEB EXPLOITATION

String Matching



Pada problem ini peserta tidak diberikan source code dari webnya. Namun, saya yakin ini adalah problem string comparison sederhana, menggunakan strcmp dari PHP.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>String Matching</title>
</head>
<body>
    <center>
    <h1>String Matching</h1>
        <form action="" method="POST">
        String1:    <input type="text" name="string1"> <br> <br>
        String2:    <input type="text" name="string2"> <br>
        <br>
        <button type="submit" name="submit"> Submit</button>
        </form>
    </center>
    </body>
</html>
```

Kita harus mengirim POST request dan memberikan param string1 dan string 2. Cukup mengubah tipe datanya saja, maka flag akan muncul.

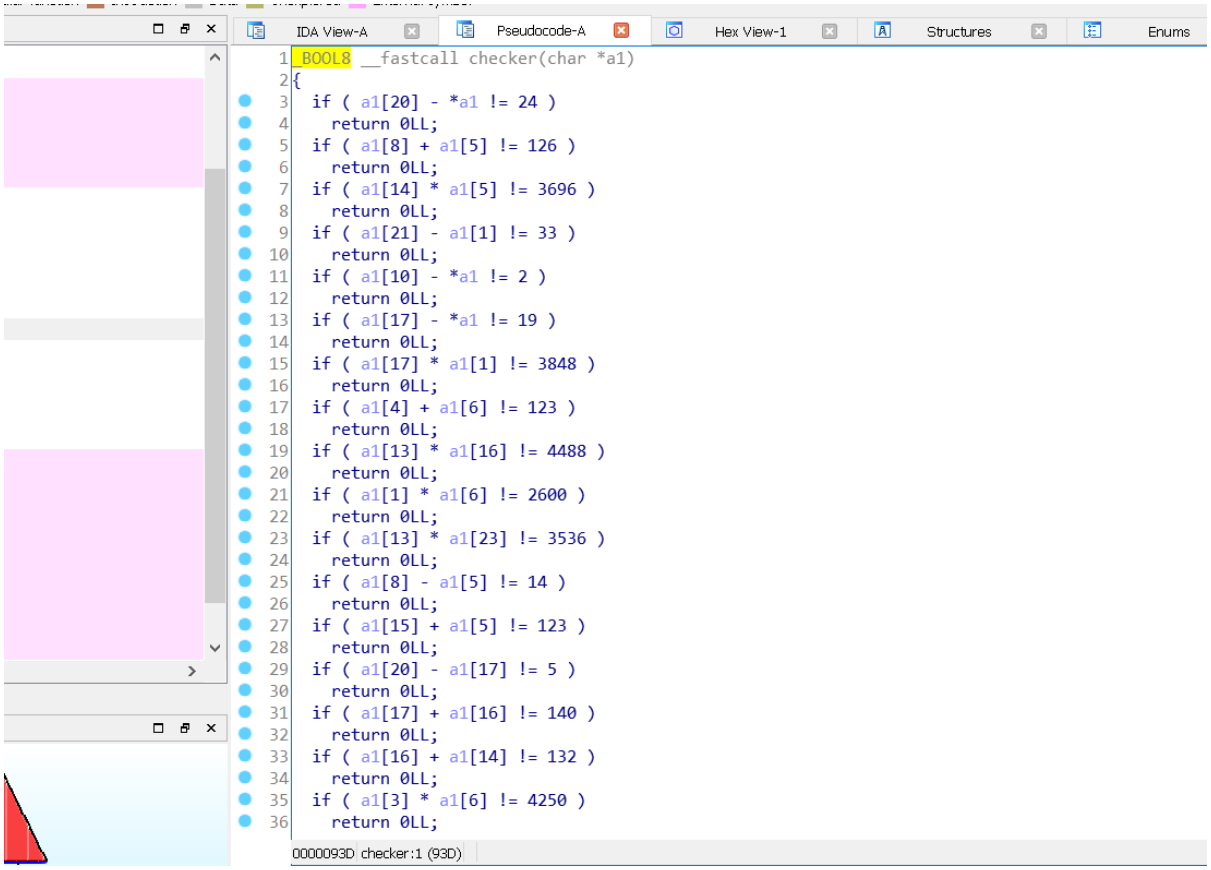
```
root@kali: ~
File  Edit  View  Search  Terminal  Help
root@kali:~# curl -X POST --data "string1[]=anjay&string2[]=tamvan" http://ctf.joints.id:40002/

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>String Matching</title>
</head>
<body>
    <center>
    <h1>String Matching</h1>
        <form action="" method="POST">
        String1:    <input type="text" name="string1"> <br> <br>
        String2:    <input type="text" name="string2"> <br>
        <br>
        <button type="submit" name="submit"> Submit</button>
        </form>
    </center>
    <br />
<b>Warning</b>:  md5() expects parameter 1 to be string, array given in <b>/app/index.php</b> on l
ine <b>38</b><br />
<br />
<b>Warning</b>:  md5() expects parameter 1 to be string, array given in <b>/app/index.php</b> on l
ine <b>38</b><br />
<center> JOINTS20{b4by_typ3_ju99lin9_md5_} </center></body>
</html>
root@kali:~#
```

Flag :

JOINTS20{b4by_typ3_ju99lin9_md5_}

REVERSE ENGINEERING

Crackme



Ini adalah problem conditional sederhana, saya solve menggunakan z3. Template script saya ambil dari writeup tim petir, organisasi di mana saya berada. (https://petircysec.com/slashroot-ctf-2019-android/)

```python
from z3 import *

def is_valid(x):
    return Or(
        (x == ord('A')),
        (x == ord('B')),
        (x == ord('C')),
        (x == ord('D')),
        (x == ord('E')),
        (x == ord('F')),
        (x == ord('G')),
        (x == ord('H')),
        (x == ord('I')),
        (x == ord('J')),
        (x == ord('K')),
        (x == ord('L')),
        (x == ord('M')),
        (x == ord('N')),
        (x == ord('O')),
        (x == ord('P')),
        (x == ord('Q')),
        (x == ord('R')),
        (x == ord('S')),
        (x == ord('T')),
        (x == ord('U')),
        (x == ord('V')),
        (x == ord('W')),
        (x == ord('X')),
        (x == ord('Y')),
        (x == ord('Z')),
        (x == ord('0')),
        (x == ord('1')),
        (x == ord('2')),
        (x == ord('3')),
        (x == ord('4')),
        (x == ord('5')),
        (x == ord('6')),
        (x == ord('7')),
        (x == ord('8')),
        (x == ord('9')),
        (x == ord('-'))
    )
```

```python
s = Solver()
vec = ""

for i in range(0, 25):
    vec += "a1[{}] ".format(i)

a1 = BitVecs(vec, 32)


s.add(a1[20] - a1[0] == 24)
s.add(a1[8] + a1[5] == 126)
s.add(a1[14] * a1[5] == 3696)
s.add(a1[21] - a1[1] == 33)
s.add(a1[10] - a1[0] == 2)
s.add(a1[17] - a1[0] == 19)
s.add(a1[17] * a1[1] == 3848)
s.add(a1[4] + a1[6] == 123)
s.add(a1[13] * a1[16] == 4488)
s.add(a1[1] * a1[6] == 2600)
s.add(a1[13] * a1[23] == 3536)
s.add(a1[8] - a1[5] == 14)
s.add(a1[15] + a1[5] == 123)
s.add(a1[20] - a1[17] == 5)
s.add(a1[17] + a1[16] == 140)
s.add(a1[16] + a1[14] == 132)
s.add(a1[3] * a1[6] == 4250)
s.add(a1[18] + a1[14] == 145)
s.add(2 * a1[13] == 136)
s.add(a1[17] - a1[10] == 17)
s.add(a1[11] + a1[8] == 145)
s.add(a1[9] + a1[1] == 135)
s.add(a1[11] + a1[24] == 146)
s.add(a1[3] - a1[7] == 11)
s.add(a1[0] - a1[2] == 2)
s.add(a1[11] - a1[13] == 7)
s.add(a1[3] + a1[4] == 158)
s.add(a1[3] - a1[16] == 19)
s.add(a1[4] - a1[14] == 7)
s.add(a1[12] * a1[1] == 4056)
s.add(a1[20] + a1[8] == 149)
s.add(a1[9] - a1[4] == 10)
s.add(a1[9] - a1[6] == 33)
s.add(a1[9] * a1[13] == 5644)
```

```python
s.add(a1[9] + a1[1] == 135)
s.add(a1[11] + a1[24] == 146)
s.add(a1[3] - a1[7] == 11)
s.add(a1[0] - a1[2] == 2)
s.add(a1[11] - a1[13] == 7)
s.add(a1[3] + a1[4] == 158)
s.add(a1[3] - a1[16] == 19)
s.add(a1[4] - a1[14] == 7)
s.add(a1[12] * a1[1] == 4056)
s.add(a1[20] + a1[8] == 149)
s.add(a1[9] - a1[4] == 10)
s.add(a1[9] - a1[6] == 33)
s.add(a1[9] * a1[13] == 5644)
s.add(a1[16] + a1[5] == 122)
s.add(a1[16] - a1[10] == 9)
s.add(a1[17] + a1[24] == 145)
s.add(a1[20] - a1[13] == 11)
s.add(a1[18] * a1[11] == 5925)
s.add(a1[21] * a1[23] == 4420)
s.add(a1[22] * a1[7] == 5698)
s.add(a1[15] - a1[19] == 12)
s.add(a1[16] - a1[1] == 14)
s.add(a1[3] - a1[13] == 17)
s.add(a1[12] * a1[8] == 5460)
s.add(a1[21] * a1[13] == 5780)
s.add(a1[7] * a1[1] == 3848)
s.add(a1[22] + a1[6] == 127)
s.add(a1[13] + a1[5] == 124)
s.add(a1[24] + a1[1] == 123)

for i in range(0, 25):
    s.add(is_valid(a1[i]))

while s.check() == z3.sat:
    model = s.model()
    flag = ""
    nope = []
    for i in a1:
        if str(i) and model[i] is not None:
            flag += chr(int(str(model[i])))
        nope.append(i != model[i])
    s.add((nope[:-1]))

    print(flag)
```

Open

*titit.py
~/Downloads

```python
s.add(a1[11] + a1[24] == 146)
s.add(a1[3] - a1[7] == 11)
s.add(a1[0] - a1[2] == 2)
s.add(a1[11] - a1[13] == 7)
s.add(a1[3] + a1[4] == 158)
s.add(a1[3] - a1[16] == 19)
s.add(a1[4] - a1[14] == 7)
s.add(a1[12] * a1[1] == 4056)
s.add(a1[20] + a1[8] == 149)
s.add(a1[9] - a1[4] == 10)
s.add(a1[9] - a1[6] == 33)
s.add(a1[9] * a1[13] == 5644)
s.add(a1[16] + a1[5] == 122)
s.add(a1[16] - a1[10] == 9)
s.add(a1[17] + a1[24] == 145)
s.add(a1[20] - a1[13] == 11)
s.add(a1[18] * a1[11] == 5925)
s.add(a1[21] * a1[23] == 4420)
s.add(a1[22] * a1[7] == 5698)
s.add(a1[15] - a1[19] == 12)
s.add(a1[16] - a1[1] == 14)
s.add(a1[3] - a1[13] == 17)
s.add(a1[12] * a1[8] == 5460)
s.add(a1[21] * a1[13] == 5780)
s.add(a1[7] * a1[1] == 3848)
s.add(a1[22] + a1[6] == 127)
s.add(a1[13] + a1[5] == 124)
s.add(a1[24] + a1[1] == 123)
```

root@kali: ~/Downloads

File Edit View Search Terminal Help

```
root@kali:~/Downloads# python3 titit.py
745UI82JFS9KNDBCBJO7OUM4G
root@kali:~/Downloads# nc 104.199.120.115 7778
745UI-82JFS-9KNDB-CBJO7-OUM4G
JOINTS20{z3_algebra_solver}

root@kali:~/Downloads#
```

Setrip (-)

```
__int64 __fastcall main(__int64 a1, char **a2, char **a3)
{
  bool v3; // bl
  __int64 v4; // rax
  __int64 v5; // rax
  char v7; // [rsp+0h] [rbp-60h]
  char v8; // [rsp+20h] [rbp-40h]
  unsigned __int64 v9; // [rsp+48h] [rbp-18h]

  v9 = __readfsqword(0x28u);
  std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::basic_string(&v7, a2, a3);
  std::operator<<<std::char_traits<char>>(&std::cout, "password: ");
  fflush(stdout);
  std::operator>><char,std::char_traits<char>,std::allocator<char>>(&std::cin, &v7);
  std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::basic_string(&v8, &v7);
  v3 = (unsigned int)sub_10DA(&v8) != 0;
  std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::~basic_string(&v8);
  if ( v3 )
  {
    v4 = std::operator<<<std::char_traits<char>>(&std::cout, "JOINTS{");
    v5 = std::operator<<<char,std::char_traits<char>,std::allocator<char>>(v4, &v7);
    std::operator<<<std::char_traits<char>>(v5, "}\n");
  }
  else
  {
    std::operator<<<std::char_traits<char>>(&std::cout, "Nope\n");
  }
  std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::~basic_string(&v7);
  return 0LL;
}
```

Masuk ke fungsi main, nampaknya program dibuat menggunakan bahasa C++. Di sini kita harus menginput password yang sesuai. Mari kita lihat sub_10DA yang dipanggil di sana.

```
__int64 __fastcall sub_10DA(__int64 a1)
{
  __int64 v1; // rdx
  unsigned __int64 v2; // rbx
  _BYTE *v3; // rax
  char v4; // dl
  unsigned int v5; // ebx
  char v7; // [rsp+1Bh] [rbp-65h]
  int i; // [rsp+1Ch] [rbp-64h]
  char v9; // [rsp+20h] [rbp-60h]
  char v10; // [rsp+40h] [rbp-40h]
  unsigned __int64 v11; // [rsp+68h] [rbp-18h]

  v11 = __readfsqword(0x28u);
  std::allocator<char>::allocator(&v7);
  std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::basic_string(&v9, &unk_1505, &v7);
  std::allocator<char>::~allocator(&v7);
  std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::basic_string(&v10, &unk_1505, v1);
  for ( i = 0; ; ++i )
  {
    v2 = i;
    if ( v2 >= std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::length(a1) )
      break;
    v3 = (_BYTE *)std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::operator[](a1, i);
    v4 = *v3 - 56;
    LODWORD(v3) = (unsigned int)(((char)*v3 + 200) >> 31) >> 25;
    std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::operator+=(
      &v10,
      (unsigned int)(char)((((_BYTE)v3 + v4) & 0x7F) - (_BYTE)v3));
  }
  v5 = (unsigned __int8)sub_13F2(&v10, &v9);
  std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::~basic_string(&v10);
  std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::~basic_string(&v9);
  return v5;
}
```

Oke, nampaknya ini fungsi yang memproses passwordnya. Ada bagian yang menarik di sini, yaitu unk_1505.

```
IDA View-A    Pseudocode-A    Hex View-1    Structures
.rodata:0000000000001501                 db    0
.rodata:0000000000001502                 db    2
.rodata:0000000000001503                 db    0
.rodata:0000000000001504                 db    0
.rodata:0000000000001505 unk_1505        db  2Fh ; /              ; DATA XF
.rodata:0000000000001506                 db  7Bh ; {
.rodata:0000000000001507                 db  7Dh ; }
.rodata:0000000000001508                 db  2Dh ; -
.rodata:0000000000001509                 db  3Ah ; :
.rodata:000000000000150A                 db  27h ; '
.rodata:000000000000150B                 db  0Fh
.rodata:000000000000150C                 db  0Dh
.rodata:000000000000150D                 db  7Dh ; }
.rodata:000000000000150E                 db  2Dh ; -
.rodata:000000000000150F                 db  3Ah ; :
.rodata:0000000000001510                 db  27h ; '
.rodata:0000000000001511                 db  33h ; 3
.rodata:0000000000001512                 db  7Bh ; {
.rodata:0000000000001513                 db  41h ; A
.rodata:0000000000001514                 db  27h ; '
.rodata:0000000000001515                 db  2Bh ; +
.rodata:0000000000001516                 db  10h
.rodata:0000000000001517                 db  2Dh ; -
.rodata:0000000000001518                 db  2Bh ; +
.rodata:0000000000001519                 db  33h ; 3
.rodata:000000000000151A                 db  2Dh ; -
.rodata:000000000000151B                 db  3Ah ; :
.rodata:000000000000151C                 db    0
.rodata:000000000000151D aPassword       db 'password: ',0       ; DATA XF
```

Ini adalah password yang sudah di proses pada function sebelumnya. Tugas kita adalah, mengembalikan password awalnya. Sebenarnya functionnya bisa dipahami, tetapi karena keterbatasan waktu, saya buatkan script bruteforce saja, lebih cepat daripada memahami fungsinya.

```
Start here    main.c
#include <stdio.h>
int main(){
    int y[] = {0x2F,0x7B,0x7D,0x2D,0x3A,0x27,0x0F,0x0D,0x7D,0x2D,0x3A,0x27,0x33,0x7B,0x41,0x27,0x2B,0x10,0x2D,0x2B,0x33,0x2D,0x3A};
    int len = sizeof(y)/sizeof(y[0]);
    for(int i=0; i<len; i++){
        for(int j=0; j<256; j++){
            int v4 = j-56;
            int x = ((j-56+200)>>31)>>25;
            if(((x+v4)&0x7f)-x == y[i]){
                printf("%c", j);
                break;
            }
        }
    }
}
```

```
C:\Users\admin\Desktop\koding\ctf\main.exe
g35er_GE5er_k3y_cHecker
Process returned 0 (0x0)   execution time : 1.671 s
Press any key to continue.
```

FLAG = JOINTS20{g35er_GE5er_k3y_cHecker}

Crypto – Classic

Kita diberikan cipher dan soal.py

```
1  from string import printable
2  import random
3  from constants import flag,key
4
5  assert len(key)==15
6  prepare = ''.join( bin(ord(i))[2:].rjust(8,'0') for i in flag )
7
8  c = ''
9  for i in range(len(prepare)):
10     c += chr( ord(prepare[i]) + ord(key[i%len(key)]) - ord('A') )
11
12 f = open('cipher','w')
13 f.write(c)
14 f.close()
```

Soal.py

```
1  W_`Ti^eUX^TcXYcX^`Uh_dUY]TdWYcX^_Ui_dVX]SdXYbW_`Th_eUX^ScWYcW^_Th_eVY]TdWYcW__Ti^eUY^TdXXcX__Uh^dUY^ScWXbX^_Ui^dUY^ScXXbW__Ui_eVX^TdXXbX^_Ui^dUX]
   TcXXcW__Uh_eVY^SdWXbW_`Ti_dVY]ScWYcW__Th_eVX]TdWXcX^`Ti^dVX]ScXXcX^_Ti_dVY]ScWYbX__Ui^dUY]TdXYcW_`Ti^dVX^SdWXcX^`Ti_eVY]TdWXbX^_Ti_dVX]
   SdXXbX^_Ti_eVY]T
```

Pertama saya fokus ke soal.py , dilihat bahwa cipher adalah hasil akhirnya di line 13 write.

Dan dapat dilihat dari baris 6,

```
6  prepare = ''.join( bin(ord(i))[2:].rjust(8,'0') for i in flag )
7
```

```
for i in flag )
```

Ini maksudnya flag dibagi menjadi per karakter, dan karakter diubah menjadi ASCII decimal, kemudian dari ASCII decimal diubah ke dalam binary, yang disusun dalam format awal 0b…… .

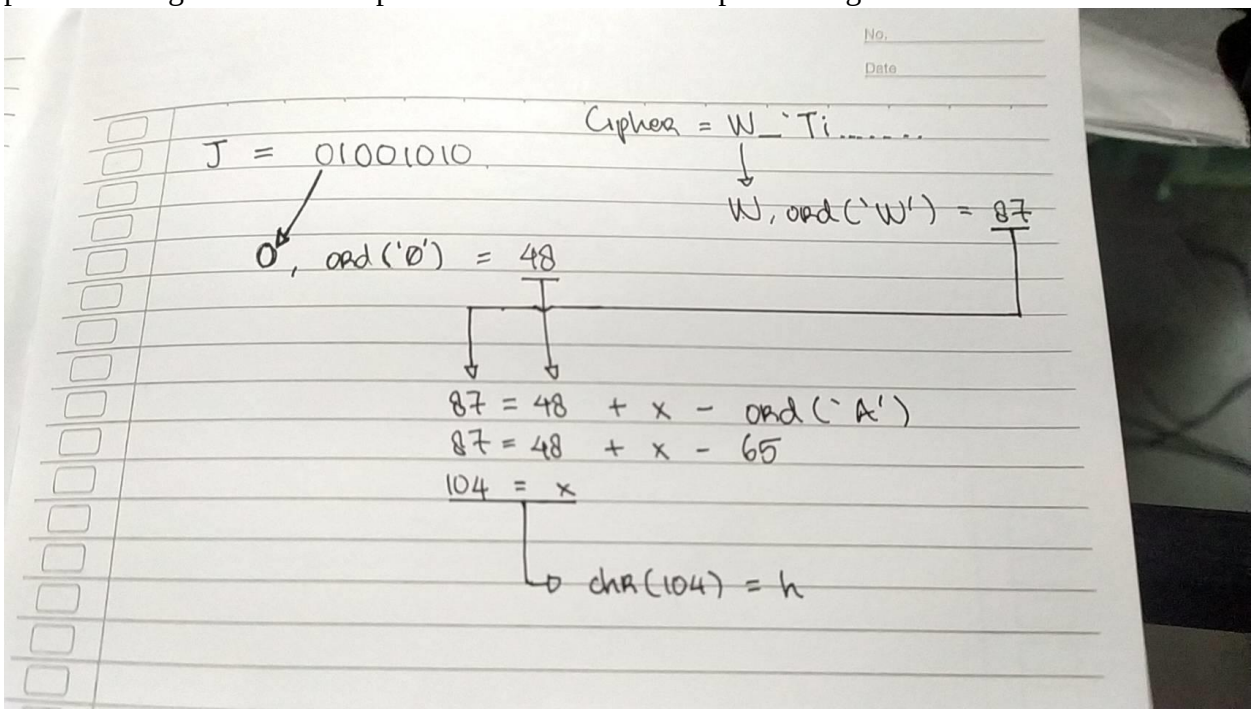
Itu mengapa ada `[2:]`, ini bermaksud mulai menyimpan dari karakter ke dua, berarti bagian awal 0b, terlewati, kemudian ada rjust yang meng-adjust biar sizenya 8 dan mengisi yang kosong dengan angka nol-nol sampai panjangnya 8.

sekarang sudah menjadi binary semua berarti panjang dari flag adalah 8x dari awal, dari ini saya konklusi bahwa panjang dari cipher text itu merupakan 8x dari flag asli karena dari hitungan for loop baris 9 tidak mengubah panjangnya, Cuma diubah menjadi karakter ASCII,

yang pertama saya lakukan adalah saya musti mendapatkan key, yang dari baris 5 menyatakan bahwa panjangnya adalah 15.

```
5  assert len(key)==15
```

dan juga saya tau kalau awal flag adalah JOINTS20, jadi saya mengunakan itu untuk menjadi patokan meng-reverse dari cipher kembali untuk mendapatkan flag.



Logika-nya seperti diatas, Kodingnya seperti ini,

```
data = 'JOINTS'
c1 = "W_`Ti^eUX^TcXYcX^`Uh_dUY]TdWYcX^_Ui_dVX]SdXYbW_`"
dat = ''.join(bin(ord(i))[2:].rjust(8, '0') for i in data)
for i in range(len(dat)):
    t1 = ord(c1[i])
    a = ord(dat[i])
    b = ord('A')
    x = t1 - a + b
    print(chr(x), end="")
```

Dan hasil compilenya seperti ini,

```
C:\Pandy\Programming\Python\Kattis\venv\Scripts\python.exe C:
hoqeyoufinethishopfyoufinduhishopezoufindtiishop
Process finished with exit code 0
```

Seperti di hitungan saya masih ada kesalahan sedikit-sedikit, tetapi saya sudah bisa menebak bahwa flag keynya adalah hopeyoufindthis, jadi dengan menggunakan itu saya menstruktur

ulang dengan soal.py dan membuat script ini,

```python
c = "W_`Ti^eUX^TcXYcX^`Uh_dUY]TdWYcX^_Ui_dVX]SdXYbW_`Th_eUX^ScWYcW^_Th_eVY]TdWYcW__Ti^e
x = "hopeyoufindthis"
d1 = ord('A')
line = ''
for i in range(len(c)):
    temp = ord(c[i]) - ord(x[i % len(x)]) + d1
    line += chr(temp)

n = 8
res = [line[i:i+n] for i in range(0, len(line), n)]
for i in res:
    print(i, end=" ")
```

```
C:\Pandy\Programming\Python\Kattis\venv\Scripts\python.exe C:/Users/Pandy/Desktop/Desktop/test2.py
01101010 01101111 01101001 01101110 01110100 01110011 00110010 00110000 01111011 01101001 01011111 01110100 00110000 0100
Process finished with exit code 0
```

Dari hasil binary tersebut saya menggunakan online converter untuk mengubah binary menjadi flag lagi

Enter the Binary code here:

01101010 01101111 01101001 01101110 01110100 01110011 00110010 00110000 01111011 01101001 01011111 01110100 00110000 01001100 01100100 01011111 01111001 00110000 01010101 01011111 01000011 01101100 00110100 01110011 00110101 00100001 01100011 01100001 01101100 01011111 01101001 01010011 01011111 01100010 00110100 01100100 01111101

Translate!

ASCII text:

joints20{i_t0Ld_y0U_Cl4s5!cal_iS_b4d}

Flag : JOINTS20{i_t0Ld_y0U_Cl4s5!cal_iS_b4d}

Crypto – Modulo

```python
from sympy import mod_inverse
from sympy import isprime
from secret import flag
from Crypto.Util.number import getPrime

e = 65537
while True:
    p = getPrime(1024)
    q = mod_inverse(e,p)
    if isprime(q):
        break

N=p*q

m=int(flag.encode("hex"),16)
c=pow(m,e,N)
f=open("pub.key","a")
f.write("e:" +str(e)+"\n")
f.write("N:" +str(N))
f.close()

f=open("flag.enc","w")
f.write(str(c))
f.close()
```

Titik paling mantap dari RSA ini yang saya lihat, bahwa prime number pertama digunakan lagi untuk prime number kedua, tetapi menggunakan mod invers, jadi kita bisa mengembalikannya lagi dengan melakukan kalkulasi ini,

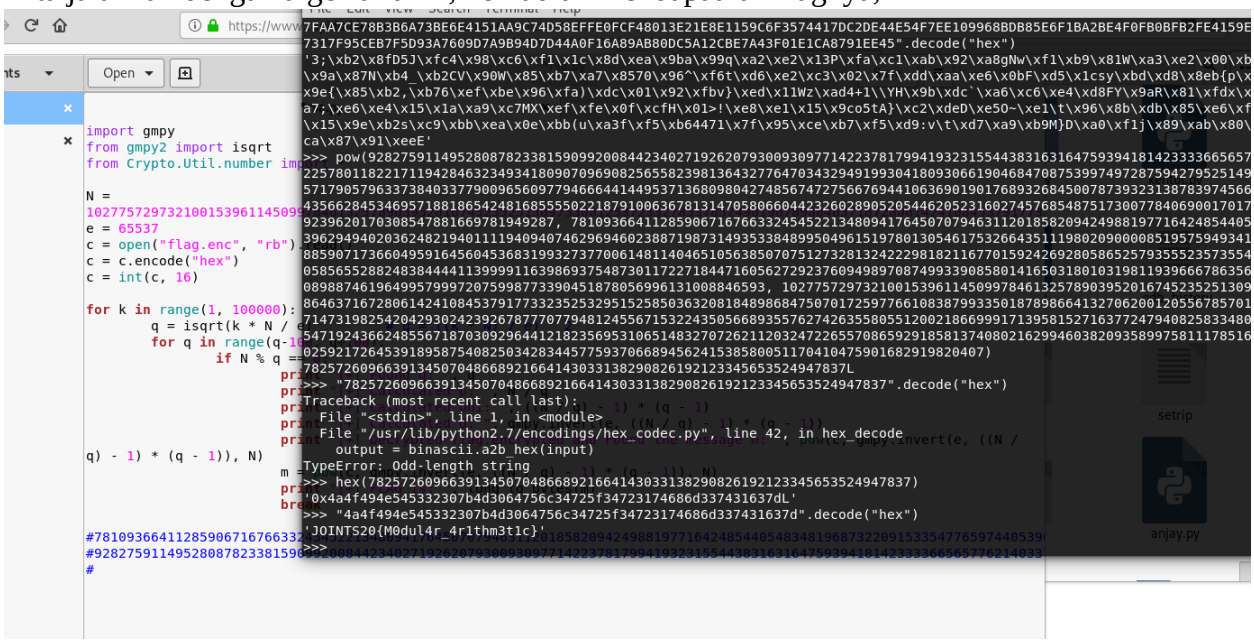
```
q = mod_inverse(p)
q = e ^ (-1) mod p
q * e = 1 mod p
q * e = k * p + 1
```

```
q * q * e = q * (k * p + 1)
(q ^ 2) * e = (k * p * q) + q
(q ^ 2) * e = (k * N) + q
((q ^ 2) * e) - q = k * N
q = ((k * N) / e) ^ 2
```

Jadi kita bisa ngebrute force K untuk mendapatkan prime number q dan dari sana kita bisa mendapatkan sisanya,

```python
for k in range(1, 100000):
    q = isqrt(k * N / e)        # q = ((k * N) / e) ^ 2
    for q in range(q-100, q+100):
        if N % q == 0:
            print "[+] Found q: ", q
            print "[+] Calculated p: ", N / q
            print "[+] Calculated phi: ", ((N / q) - 1) * (q - 1)
            print "[+] Calculated d: ", gmpy.invert(e, ((N / q) - 1) * (q - 1))
            print "[+] Decrypted flag.encrypted and Found the message m: ", pow(c, gmpy.invert(e, ((N / q) - 1) * (q - 1)), N)
            m = pow(c, gmpy.invert(e, ((N / q) - 1) * (q - 1)), N)
            print "[+] FLAG is: ", long_to_bytes(m)
            break
```

Kita jalankan dengan algoritma ini, kemudian mendapatkan flagnya,



JOINTS20{M0dul4r_4r1thm3t1c}

Referensi untuk mod_invers, saya dapat dari website ini,

https://medium.com/bugbountywriteup/tokyowesterns-ctf-4th-2018-writeup-part-4-f64e1583b315

penyelesaian sangat mirip tapi dibagian terakhir ada kesalahan, setelah di compute D, kebawahnya menjadi tidak jelas, jadi dari D saya lanjutkan manual dan mendapatkan jawaban asli-nya

Forensic – LaBrava no Ai

Intro - Recon

Pertama analisis itu file apa sih?

File berekstensi .dmp file ini adalah file yang dibuat scr otomatis ketika komputer mengalami crash/eror dan terbuat setelah komputer tersebut mereboot.

Loh .dmp file ini bakal tersimpan dimana ? Yakni di %SystemRoot%memory.dump atau misal di C:\Windows\memory.dump

Nah Case disini kita sudah dikasih hint bahwa si Laptop "Brava" ini jatuh *error crash* dan akhirnya laptop memuat .dump file. Mari kita berlanjut ke analisisnya !

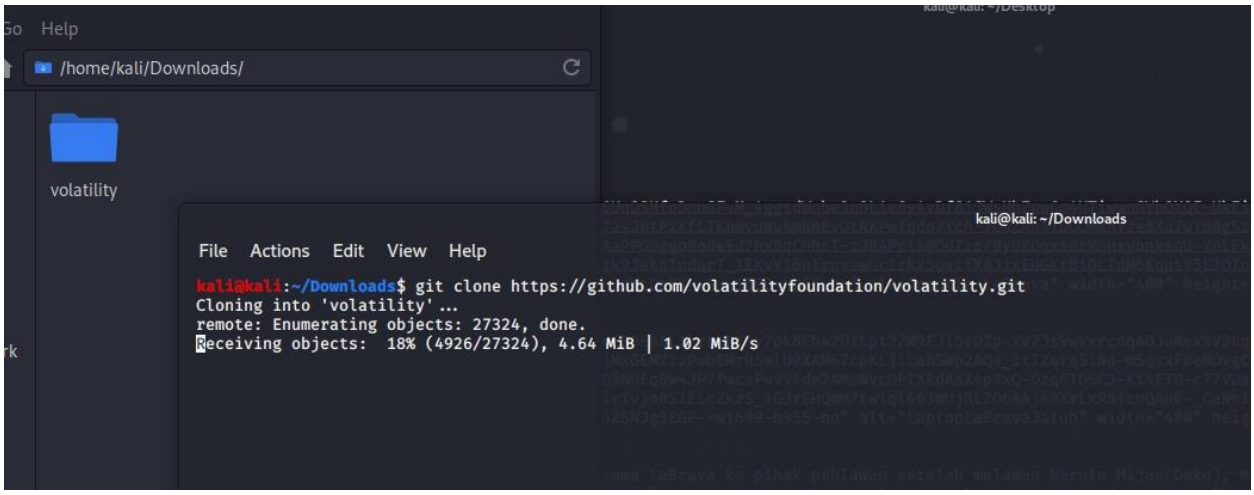
Tools apa yang compatible untuk menganalisis .dmp file tersebut ? Setelah saya melakukan google searching , saya menggunakan tools Volatility .

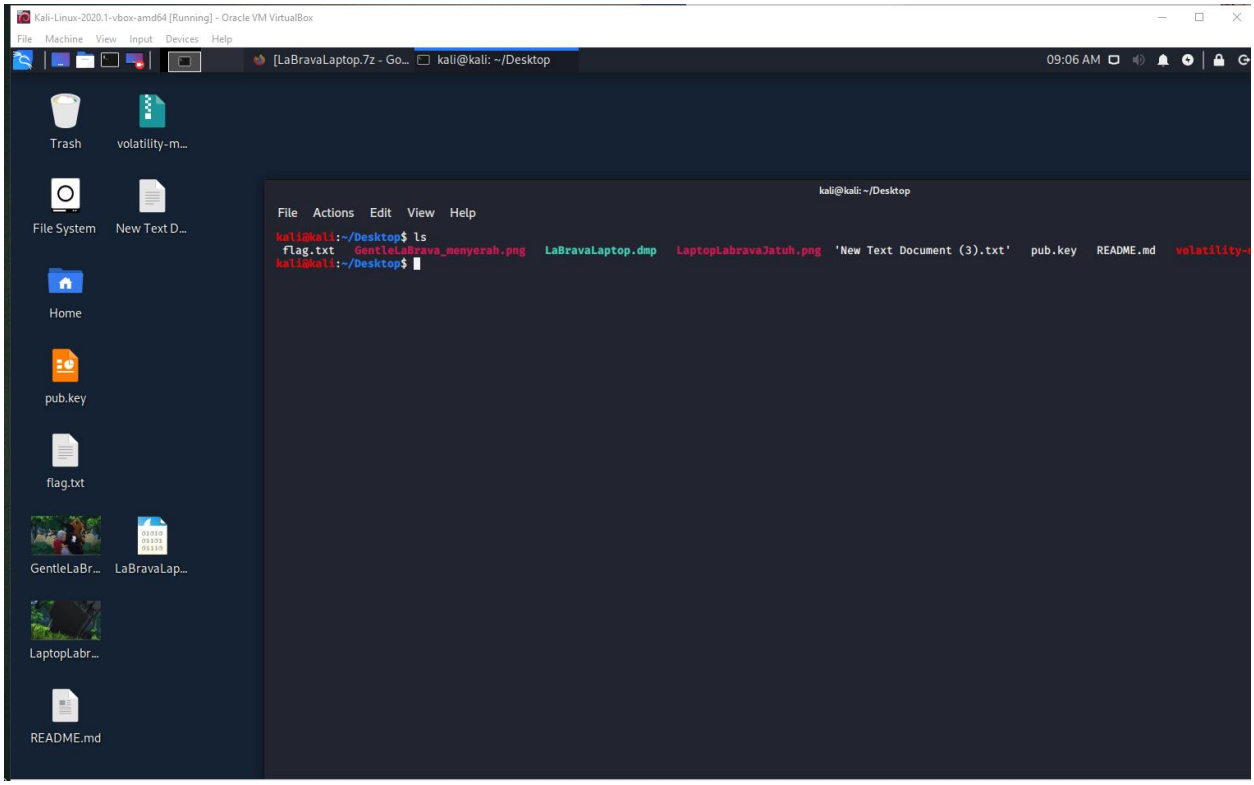
Referensi : https://tools.kali.org/forensics/volatility

Mengapa Volatility ? Tools ini sangat powerful karena dia dapat men-dump memori OS yang cukup banyak , hingga Windows ke versi lamanya dan juga OS Linux

Kita dapat meng git clonenya dari sini :

https://www.google.com/url?sa=t&source=web&rct=j&url=https://github.com/volatilityfoundation/volatility&ved=2ahUKEwjRzNa53pfpAhWMWX0KHRPaCJUQFjAAegQIBxAD&usg=AOvVaw1_vBpA3qj3hVAAoPRS0Ta4





Setelah melihat manual page Volatility pada web untuk menentukan profil kepunyaan file dump ini :

kdbgscan

As opposed to imageinfo which simply provides profile suggestions, kdbgscan is designed to positively identify the correct profile and the correct KDBG address (if there happen to be multiple). This plugin scans for the KDBGHeader signatures linked to Volatility profiles and applies sanity checks to reduce false positives. The verbosity of the output and number of sanity checks that can be performed depends on whether Volatility can find a DTB, so if you already know the correct profile (or if you have a profile suggestion from imageinfo), then make sure you use it.

Here's an example scenario of when this plugin can be useful. You have a memory sample that you believe to be Windows 2003 SP2 x64, but pslist doesn't show any processes. The `pslist` plugin relies on finding the process list head which is pointed to by KDBG. However, the plugin takes the *first* KDBG found in the memory sample, which is not always the *best* one. You may run into this problem if a KDBG with an invalid PsActiveProcessHead pointer is found earlier in a sample (i.e. at a lower physical offset) than the valid KDBG.

Notice below how `kdbgscan` picks up two KDBG structures: an invalid one (with 0 processes and 0 modules) is found first at `0xf80001172cb0` and a valid one (with 37 processes and 116 modules) is found next at `0xf80001175cf0`. In order to "fix" pslist for this sample, you would simply need to supply the `--kdbg=0xf80001175cf0` to the `plist` plugin.

```
$ python vol.py -f Win2K3SP2x64-6f1bedec.vmem --profile=Win2003SP2x64 kdbgscan
Volatility Foundation Volatility Framework 2.4
**************************************************
Instantiating KDBG using: Kernel AS Win2003SP2x64 (5.2.3791 64bit)
Offset (V)                    : 0xf80001172cb0
Offset (P)                    : 0x1172cb0
KDBG owner tag check          : True
```

Maka kita menjalankan command tersebut pada terminal :

```
File  Actions  Edit  View  Help

kali@kali:~$ cd Desktop/
kali@kali:~/Desktop$ volatility -f LaBravaLaptop.dmp kdbgscan
Volatility Foundation Volatility Framework 2.6
**************************************************
Instantiating KDBG using: Unnamed AS WinXPSP2x86 (5.1.0 32bit)
Offset (P)                    : 0x293dc28
KDBG owner tag check          : True
Profile suggestion (KDBGHeader): Win7SP1x86_23418
Version64                     : 0x293dc00 (Major: 15, Minor: 7601)
PsActiveProcessHead           : 0x829556d8
PsLoadedModuleList            : 0x8295c5b0
KernelBase                    : 0x8281c000

**************************************************
Instantiating KDBG using: Unnamed AS WinXPSP2x86 (5.1.0 32bit)
Offset (P)                    : 0x293dc28
KDBG owner tag check          : True
Profile suggestion (KDBGHeader): Win7SP1x86
Version64                     : 0x293dc00 (Major: 15, Minor: 7601)
PsActiveProcessHead           : 0x829556d8
PsLoadedModuleList            : 0x8295c5b0
KernelBase                    : 0x8281c000

**************************************************
Instantiating KDBG using: Unnamed AS WinXPSP2x86 (5.1.0 32bit)
Offset (P)                    : 0x293dc28
KDBG owner tag check          : True
Profile suggestion (KDBGHeader): Win7SP1x86_24000
Version64                     : 0x293dc00 (Major: 15, Minor: 7601)
PsActiveProcessHead           : 0x829556d8
PsLoadedModuleList            : 0x8295c5b0
KernelBase                    : 0x8281c000

**************************************************
Instantiating KDBG using: Unnamed AS WinXPSP2x86 (5.1.0 32bit)
Offset (P)                    : 0x293dc28
KDBG owner tag check          : True
Profile suggestion (KDBGHeader): Win7SP0x86
Version64                     : 0x293dc00 (Major: 15, Minor: 7601)
PsActiveProcessHead           : 0x829556d8
PsLoadedModuleList            : 0x8295c5b0
KernelBase                    : 0x8281c000

kali@kali:~/Desktop$
```

Hasilnya diperoleh bahwa **Win7SP1x86** adalah **opsi profil KDBG header yang akan kita pilih** . Lalu bagaimana cara kita memuat semua informasi umum yang berisikan semua proses dari si memori dump ini ? Dengan melihat *manual page volatility* , kita menggunakan command **pslist** .

```
To list all active processes found in a MS Windows 8 SP0 image:

    $ volatility -f win8.raw --profile=Win8SP0x86 pslist

To list all active processes found in a MS Windows 8 SP0 image, using a timezone:
```

```
kali@kali:~/Desktop$ man volatility
kali@kali:~/Desktop$ volatility -f LaBravaLaptop.dmp --profile=Win7SP1x86 pslist
Volatility Foundation Volatility Framework 2.6
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start
           Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ -------------------
---------- ----------------------------
0x83f2fa10 System                    4      0     78      476 ------      0 2020-03-18 22:20:41
UTC+0000
0x845ff020 smss.exe                252      4      2       29 ------      0 2020-03-18 22:20:41
UTC+0000
0x84c8da68 csrss.exe               328    312      9      334      0      0 2020-03-18 22:20:42
UTC+0000
0x84c99148 wininit.exe             376    312      3       75      0      0 2020-03-18 22:20:42
UTC+0000
0x84c974e8 csrss.exe               384    368      7      185      1      0 2020-03-18 22:20:42
UTC+0000
0x84c9d1f8 winlogon.exe            412    368      4      111      1      0 2020-03-18 22:20:42
UTC+0000
0x84cbc030 services.exe            468    376      9      187      0      0 2020-03-18 22:20:42
UTC+0000
0x84ccc6f8 lsass.exe               484    376      7      457      0      0 2020-03-18 22:20:42
UTC+0000
0x84cce810 lsm.exe                 492    376     10      140      0      0 2020-03-18 22:20:42
UTC+0000
0x84d02030 svchost.exe             588    468     10      342      0      0 2020-03-18 22:20:43
UTC+0000
0x84d12030 VBoxService.ex          648    468     14      125      0      0 2020-03-18 22:20:43
UTC+0000
0x84d26a68 svchost.exe             716    468      7      241      0      0 2020-03-18 22:20:43
UTC+0000
0x84d41318 svchost.exe             792    468     21      409      0      0 2020-03-18 22:20:43
UTC+0000
0x84d5c030 svchost.exe             848    468     15      300      0      0 2020-03-18 22:20:43
UTC+0000
0x84d65520 svchost.exe             892    468     27      674      0      0 2020-03-18 22:20:43
UTC+0000
0x84dc0530 audiodg.exe             972    792      6      116      0      0 2020-03-18 22:20:43
UTC+0000
0x84de44c8 svchost.exe            1016    468     19      432      0      0 2020-03-18 22:20:43
UTC+0000
0x84e264d8 svchost.exe            1100    468     18      359      0      0 2020-03-18 22:20:43
UTC+0000
0x84e638b8 dwm.exe                1264    848      4       52      1      0 2020-03-18 22:20:44
UTC+0000
0x84e72b90 spoolsv.exe            1292    468      5       74      0      0 2020-03-18 22:20:44
UTC+0000
```


```
UTC+0000
0x84d12030 VBoxService.ex          648    468     14      125      0      0 2020-03-18 22:20:43
UTC+0000
0x84d26a68 svchost.exe             716    468      7      241      0      0 2020-03-18 22:20:43
UTC+0000
0x84d41318 svchost.exe             792    468     21      409      0      0 2020-03-18 22:20:43
UTC+0000
0x84d5c030 svchost.exe             848    468     15      300      0      0 2020-03-18 22:20:43
UTC+0000
0x84d65520 svchost.exe             892    468     27      674      0      0 2020-03-18 22:20:43
UTC+0000
0x84dc0530 audiodg.exe             972    792      6      116      0      0 2020-03-18 22:20:43
UTC+0000
0x84de44c8 svchost.exe            1016    468     19      432      0      0 2020-03-18 22:20:43
UTC+0000
0x84e264d8 svchost.exe            1100    468     18      359      0      0 2020-03-18 22:20:43
UTC+0000
0x84e638b8 dwm.exe                1264    848      4       52      1      0 2020-03-18 22:20:44
UTC+0000
0x84e72b90 spoolsv.exe            1292    468      5       74      0      0 2020-03-18 22:20:44
UTC+0000
```


```
0x84e7e878 taskhost.exe           1308    468     10      146      1      0 2020-03-18 22:20:44
UTC+0000
0x84e7e588 explorer.exe           1320   1248     25      657      1      0 2020-03-18 22:20:44
UTC+0000
0x84e8cac0 svchost.exe            1372    468     21      308      0      0 2020-03-18 22:20:44
UTC+0000
0x84edb550 svchost.exe            1488    468     13      211      0      0 2020-03-18 22:20:44
UTC+0000
0x84d898d8 VBoxTray.exe           1672   1320     15      142      1      0 2020-03-18 22:20:44
UTC+0000
0x84f78d40 SearchIndexer.          784    468     13      558      0      0 2020-03-18 22:20:51
UTC+0000
0x84f58d40 wmpnetwk.exe           1632    468     11      211      0      0 2020-03-18 22:20:51
UTC+0000
0x84e51360 notepad.exe            2256   1320      2       60      1      0 2020-03-18 22:20:58
UTC+0000
0x84e48c88 mspaint.exe            2296   1320      7      158      1      0 2020-03-18 22:21:06
UTC+0000
0x83fc9030 svchost.exe            2324    468      8      105      0      0 2020-03-18 22:21:06
UTC+0000
0x84fde9b8 mspaint.exe            2376   1320      8      159      1      0 2020-03-18 22:21:08
UTC+0000
0x84daf030 SearchProtocol         2516    784      8      247      0      0 2020-03-18 22:22:12
```

Terlihat ada *session* pada notepad.exe , ms.paint.exe . Hal yang pertama saya lakukan adalah dengan mengambil PID nya lalu mencoba mendumpnya menjadi strings namun gagal. Tetapi setelah itu , saya terpikirkan jikalau file yang telah dibuat disana berekstensikan .txt ataupun .jpg,.img,.jpeg . Kalau begitu marilah kita melakukan *traversing scan* .

```
kali@kali:~/Desktop$ volatility -f LaBravaLaptop.dmp --profile=Win7SP1x86 filescan | grep txt
Volatility Foundation Volatility Framework 2.6
0x000000001e82c100      8      0 R--rw- \Device\HarddiskVolume2\Users\LaBrava\Desktop\story.txt
0x000000001e82d200      1      1 -W-rw- \Device\HarddiskVolume2\Users\LaBrava\AppData\Local\Temp
\FXSAPIDebugLogFile.txt
kali@kali:~/Desktop$
```

Benar saja , terdapat 2 textfile yang dapat kita *dump* untuk dilihat apa isi dari teks tersebut . Namun file logfile.txt yang kedua tidak dapat kita lihat (prohibited permission) . Maka kita akan mendump story.txt . Syntax yang saya gunakan adalah :

```
kali@kali:~/Desktop$ volatility -f LaBravaLaptop.dmp --profile=Win7SP1x86 dumpfiles -Q 0x0000000
01e82c100 --name -D .
Volatility Foundation Volatility Framework 2.6
DataSectionObject 0x1e82c100   None   \Device\HarddiskVolume2\Users\LaBrava\Desktop\story.txt
kali@kali:~/Desktop$
```

File yang telah kita dump sudah berada pada Desktop kita . Oleh karena itu mari kita buka isinya , dan alhasil didapatkan serpihan *flag* bagian 2.

```
and his classmates big day. He remembers All Might telling him that he's having trouble using ai
r vacuum blasts in mid-air. Mina Ashido taught him to focus his senses at the right moment. Deku
 clears his head of all doubts and takes aim to stop Gentle before the villain lands.

Deku focuses One For All to 20% in his fingers and flicks to create a concussive wind blast. The
 gloves Mei made him help to focus this shockwave and Deku strikes Gentle with Delaware Smash Ai
r Force. It only stops the gentlemen for a moment, but its enough for Izuku to lunge off a utili
ty pole and grab onto the criminal. Izuku fiercely declares that his feelings are shared by ever
yone involved with the festival and they will not waver for Gentle's plan. [2]

Izuku and Gentle crash into an unoccupied construction site. Izuku recovers and thanks Mina for
her special dance training. He looks for the villain through all the dust and finds Gentle hangi
ng from a steel beam by his shirt. Gentle announces that he refuses to be swayed from his plans
and that he's nothing like the League of Villains. He simply wants to become famous for invading
 the festival and asks Izuku to look the other way. Izuku tells him the festival will be called
off and he'll get caught immediately. Gentle argues that La Brava will turn off the alarms so no
 one is alerted but Izuku sees this as an even bigger problem.

Izuku claims that Gentle's crimes have been reported by the gentleman calls his bluff and bounce
s off the air to get away. Izuku leaps after him and tries to predict his next move like Sir Nig
hteye taught him to do. Gentle bounces repeatedly off different spring-like air pockets and move
s in a pattern that's impossible for Izuku to read. Izuku tracks him to a steel beam but gets hi
t in a direction he never saw his opponent coming from.
|||||we_have_a_little_recon_for_you_flag_part2:_n0_Om01D3}|||||
Gentle tells his young adversary that he removed the bolts from the steel structure and its boun
d to collapse. Elasticity can't be remotely deactivated and anything it effects with gradually r
eturns to normal. As its effects wear off, the structure collapses and threatens to crush a byst
ander. Izuku swiftly catches the steel beam and barely holds it up. Gentle explains that he was
going to bounce the beam away from the civilian if Izuku didn't move, but he counted on the U.A.
 student leaping into action. Gentle makes a crane arm elastic and tells Izuku to stay put while
 he finishes his plan before sending himself flying away.

Deku remains steadfast and manages to hold up the steel beam with one arm while taking aim with
the other. Determined, Izuku shoots an air bullet at Gentle and forces him to evade. La Brava ta
kes notice of Izuku's tenacity and decides that she'll have to use her Quirk if there's any hope
 for escape.[3] Izuku flings himself off the crane hook and chases the criminals into the forest
 at the base of the hill with U.A. at its summit.

Gentle is shocked by his pursuers speed. Izuku descends and calculates the possible places where
 Gentle placed elasticized air pockets. He bounces off one above a tree and dives behind Gentle.
 Gentle creates two expansile shields above and in front of him. Izuku gets Gentle's focus on hi
m and then bounces an air blast off of the elastic air shield guarding Gentle's head. Gente Crim
inal gets blasted in the torso and is incapacitated.

Izuku quickly pins down both Gentle and La Brava and demands they surrender. La Brava refuses to
```

Flag part 2 : _n0_Om01D3}

Lantas , kemana flag part 1 ? Mari kita scan file yang berekstension .jpg , .jpeg atau .bmp yang memuat gambar. Dilakukan hal serupa seperti command di atas :

```
kali@kali:~/Desktop$ volatility -f LaBravaLaptop.dmp --profile=Win7SP1x86 filescan | grep jpg
Volatility Foundation Volatility Framework 2.6
0x000000001ed98f80      7      0 R--r-- \Device\HarddiskVolume2\Windows\Web\Wallpaper\Windows\im
g0.jpg
0x000000001ef899d0      8      0 R--r-d \Device\HarddiskVolume2\Users\LaBrava\Desktop\UA_gate.jp
g
kali@kali:~/Desktop$ volatility -f LaBravaLaptop.dmp --profile=Win7SP1x86 dumpfiles -Q 0x0000000
01ed98f80 --name -D .
Volatility Foundation Volatility Framework 2.6
DataSectionObject 0x1ed98f80   None   \Device\HarddiskVolume2\Windows\Web\Wallpaper\Windows\img0
.jpg
kali@kali:~/Desktop$
```

Didapatkan gambar pada memory dump ! Dan alhasil setelah kita buka :

Alhasil didapatkan serpihan **flag pertama !**

Flag (*complete*) :

JOINTS20{4NaT4a_7o_n0_Om01D3}


THANK YOU
JOINTS !
DO MORE OF
WHAT MAKES YOU
happy