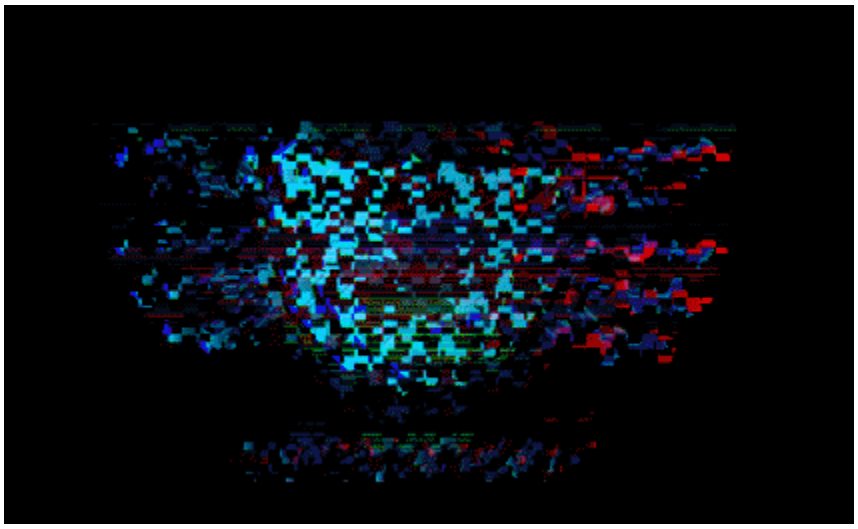
Write-Up Wreck It 4.0

flag{ganyuwangy}



4leph

DAFTAR ISI

FORENSIC

- Mixxedup(152 pts)

Diberikan file c.jpg. Pada awalnya saya mengira soal ini adalah tipe steganografi dengan memanipulasi filter dengan harapan mendapat clue pada gambar. Karena tidak terdapat apa-apa, saya mencoba cek apakah ada hidden file dengan menggunakan tools binwalk.

```
└──$ binwalk  -e c.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION

--------------------------------------------------------------------------------

0             0x0             JPEG image data, JFIF standard 1.01

81884         0x13FDC         Zip archive data, at least v2.0 to extract, compressed size: 31,
uncompressed size: 31, name: dobleh.txt

81955         0x14023         Zip archive data, at least v2.0 to extract, compressed size: 132861,
uncompressed size: 138371, name: flag.png

214964        0x347B4         End of Zip archive, footer length: 22
```

Kemudian didapat terdapat dua hidden file dobleh.txt dan flag.png.

```
└──$ cat dobleh.txt

saya aslinya 400, sekarang 2000
```

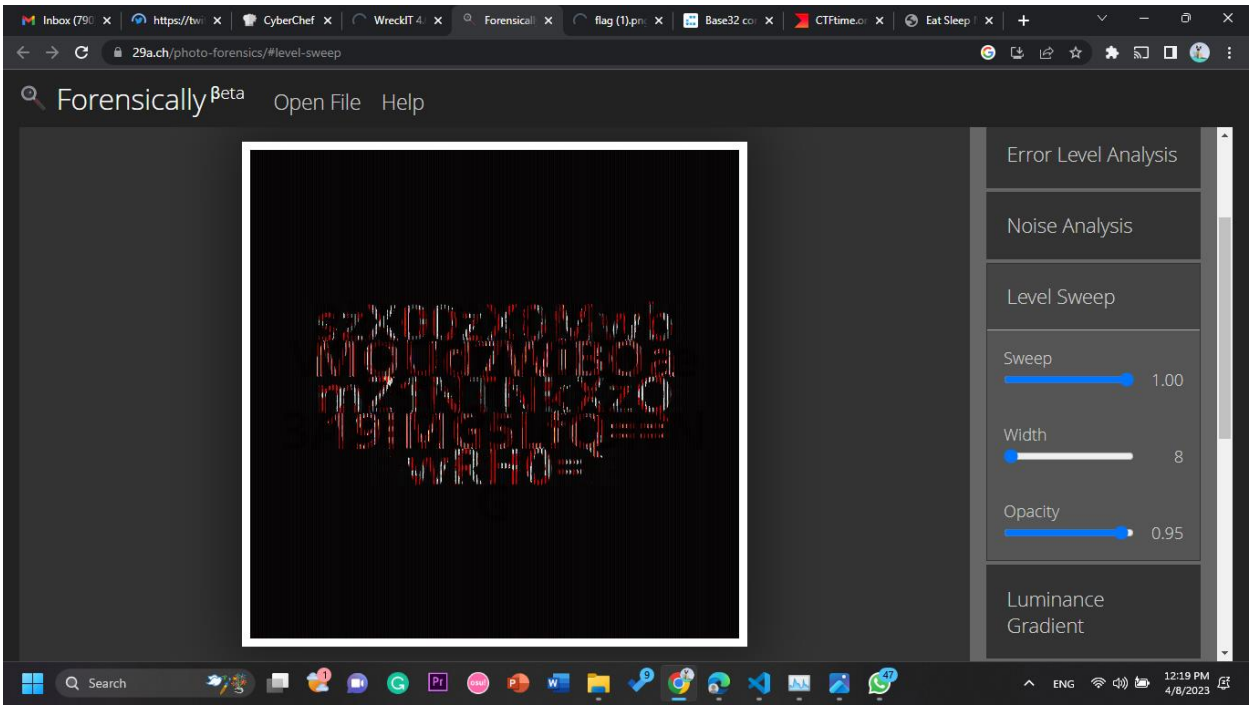
```
└──$ file flag.png

flag.png: PNG image data, 2000 x 400, 8-bit/color RGB, non-interlaced
```

Dapat diartikan bahwa soal ini meminta untuk mengubahnya menjadi 2000x2000. Dengan bantuan image resizer tool online, didapat gambar seperti berikut:

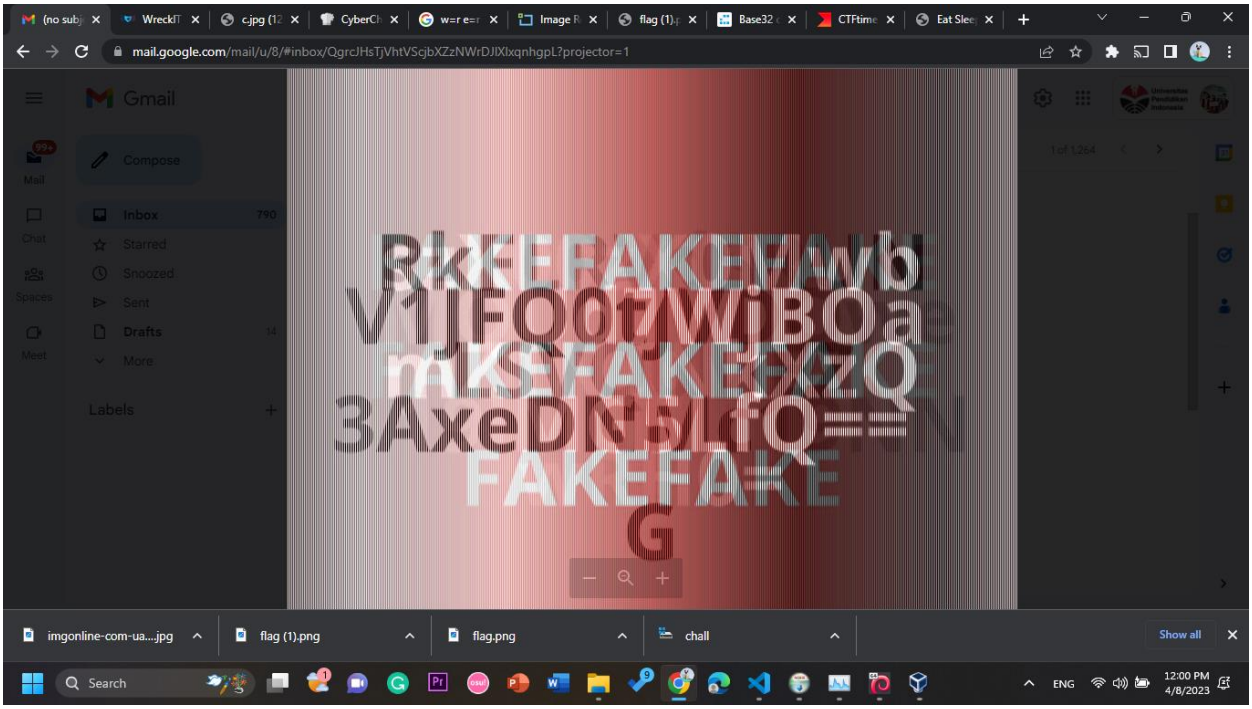
Karena tampak tidak jelas, maka diperlukan tools untuk memanipulasi filter agar kata terlihat jelas. Dengan bantuan tools https://29a.ch/photo-forensics/

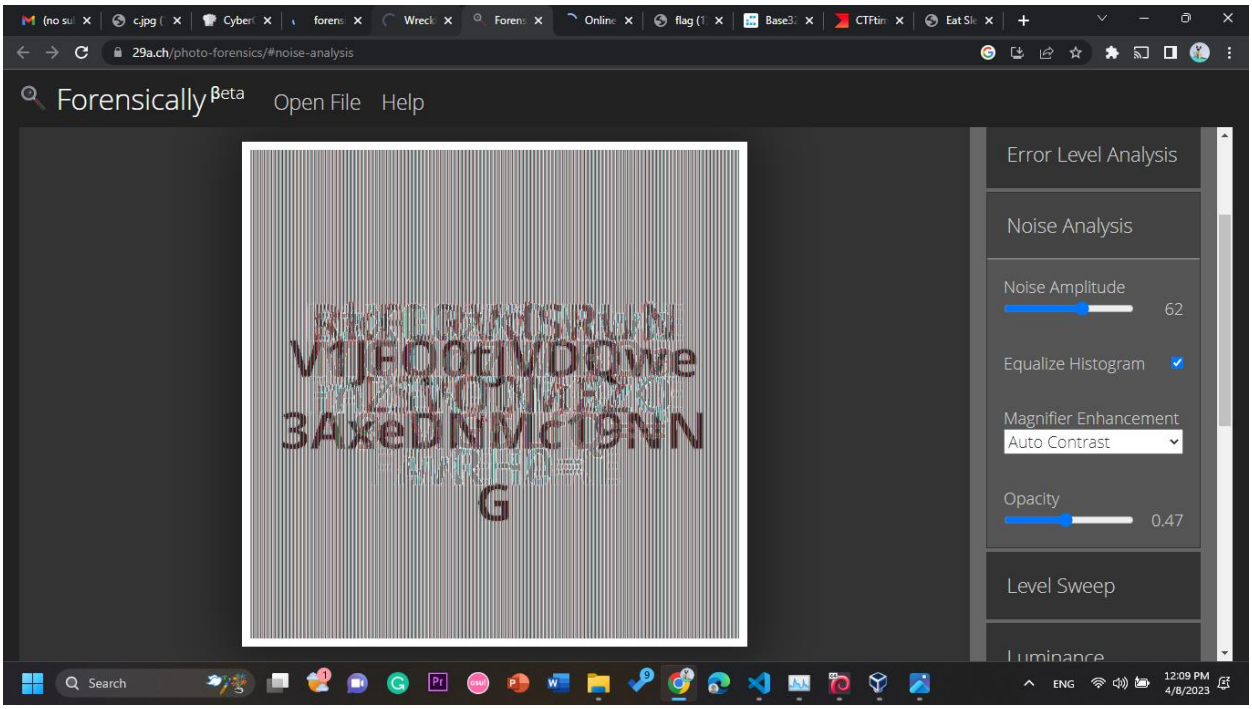
Dilakukanlah berbagai metode untuk mendapat clue, sehingga didapat 3 kata kunci pada gambar tersebut.



szX00zX0MwbmZ1NTNkXzQwRH0= (Dikelompokan berdasarkan warna pola yang cenderung mirip)

MQUd7WiBOa19IMG5LtQ==(Dikelompokan berdasarkan warna pola yang cenderung mirip)





RkFLRVdSRUNLSVQ0MEZ

V1JFQ0tJVDQwe3AxeDNMc19NNG

Tentunya terlihat jelas bahwa encoder yang digunakan merupakan base64. Oleh karena itu dengan bantuan tools https://cyberchef.org/ untuk mendecode pesan tersebut agar mendapatkan flag. Sehingga didapat:

RkFLRVdSRUNLSVQ0MEZ = **FAKEWRECKIT40F**

V1JFQ0tJVDQwe3AxeDNMc19NNG = **WRECKIT40{p1x3Ls_M4**

szX00zX0MwbmZ1NTNkXzQwRH0= = **³5ôÓ5ô3.ægSS6EóC.GÔ**

MQUd7WiBOa19IMG5LtQ== = **1..íh.9.} Á¹.Ô**

Pada awalnya saya pikir flag nya akan menyambung berdasarkan potongan-potongan tersebut, tetapi tetap tidak masuk akal. Lalu kemungkinan yang paling besar ialah dengan mencocokan potongan hasil encode base64 dari "{}" yaitu berakhir dengan sama dengan(=). Serta asumsi potongan RkFLRVdSRUNLSVQ0MEZ adalah fake, sehingga diabaikan. Maka dicoba lah kombinasi dari V1JFQ0tJVDQwe3AxeDNMc19NNG dan szX00zX0MwbmZ1NTNkXzQwRH0=.

V1JFQ0tJVDQwe3AxeDNMc19NNGszX00zX0MwbmZ1NTNkXzQwRH0=

FLAG

WRECKIT40{p1x3Ls_M4k3_M3_C0nfu53d_40D}

CRYPTO

- CRYPTO Free Flag (100 pts)

Diberikan file soal.secret dengan clue BiHB32R13,

```
└──$ cat soal.secret

00110100011000010011010001100001001101000011001100110101001101100011010000110101001
10101001101010011010000110011001101010011100100110100011000100011010101100001001101
00001101000011010100110100001101000011100100110100011001000011010000110100001100110
01100110011010001100010001101000011001000110100001100110011010100110101001101010011
10010011010100110001001100110011001000110100001110000011010000111001001101000110000
10011010100110000001101010011011100110100011001100011010100111001001101010011010000
11010101100001001101000110010000110100011000010011010100110001001101010011100000110
10000111001001100110011010100110101001101000011010000110010001101000110001100110011
00110101001101010011100000110100001110000011010100110001001100110011001000110100001
10011001100110011011100110100011001100011010000110110001101010011001000110100001101
11001101000011001100110011001101010011010000110010001101000011001000110100001101010
01101000011010100110101001101110011010000110110001100110011010000110100000110100011
01000011010000110100001110010011010001100100001101010011011000110101001110000011010
00011011100110100001100110011001100110101001101000011001100110100001101000011010001
10011000110100011000010011001100110101001101000011011100110011001101000011010100111
00100110100011000110011010001100110001101000110010000110101011000010011010100111000
00110100001101110011010000110011001101010011100000011001100110011001101010011011000
10100011001100011010001100001001100110011001000110101001110000011010000110101001100
11001101010011010001100011001101010011001100110101001100000011010100110101001100110
11001000011001101100100001100110110010000110011011001000011001101100100001100110110
0100
```

Dengan clue BiHB32R13, maka dapat diartikan bahwa isi dari file diatas diencode dari ROT13>BASE32>HEX>Binary. Maka dari itu dapat di encode dari tools https://cyberchef.org/ dengan mudah.

FLAG

WRECKIT40{CRYPTO_tolongin_aku_dong!!,_kurangPemanasan_hehehe}

MISC

- Welcome (100 pts)

Terdapat jelas pada clue.

FLAG

WRECKIT40{J4NG4N_lupa_Absen_YGYGY}

- Survey (100 pts)

Sangat Jelas.

FLAG

WRECKIT40{M4KAS1H_UDAH_I51_SURV3Y_SEM0G4_F1N4L}

REV

- REV Free Flag (100 pts)

Diberikan file chall.c, dengan isi sebagai berikut:

```
└──$ cat chall.c
```

```
#include<stdio.h>
#include<string.h>


int main(int argc, char **argv){
   int c[] = {119, 74, 101, 91, 107, 81, 116, 44, 16, 99, 20, 107, 76, 41, 127, 122, 20, 118, 71, 71, 80, 125,
82, 117, 17, 118, 84, 44, 20, 118, 127, 44, 84, 44, 83, 44, 78, 71, 78, 43, 87, 122, 73, 43, 127, 126, 82,
113, 69, 118, 68, 116, 89, 101};
   char inp[100];
   printf("apa flagnya\n");
   scanf("%s", &inp);
   int len = strlen(inp);
   if(len != 54){
      printf("bukan");
      return 0;
   }
   for(int i=0; i<len; i++){
      if(i%2==1 && inp[i] != (c[i] ^ 24)){
         printf("bukan");
         return 0;
      } else if (i%2==0  && inp[i] != (c[i] ^ 32)){
         printf("bukan");
```

```
        return 0;
    }
  }
  printf("mantap!!\n");
  return 0;
}
```

Simbol ^ adalah operator XOR, yang digunakan untuk melakukan operasi XOR antara nilai bilangan bulat dan konstanta yang sesuai (32 atau 24). Fungsi chr() kemudian digunakan untuk mengonversi nilai integer yang dihasilkan menjadi representasi karakter ASCII yang sesuai.

Maka flag dapat didapat dengan python.

```
c = [119, 74, 101, 91, 107, 81, 116, 44, 16, 99, 20, 107, 76, 41, 127, 122, 20, 118, 71, 71, 80, 125, 82, 117, 17, 118, 84, 44, 20, 118, 127, 44, 84, 44, 83, 44, 78, 71, 78, 43, 87, 122, 73, 43, 127, 126, 82, 113, 69, 118, 68, 116, 89, 101]


flag = ""


for i in range(len(c)):
    if i % 2 == 0:
        flag += chr(c[i] ^ 32)
    else:
        flag += chr(c[i] ^ 24)


print(flag)
```

FLAG

WRECKIT40{4sl1_b4ng_perm1nt44n_4t4s4n_n3wbi3_friendly}

WEB

- Jwttt (100 pts)

Diberikan IP Address dan port http://167.71.207.218:50620. Clue nya diberikan login maka diasumsikan terdapat username dan password yang tersembunyi didalam website. Dilakukanlah dengan mengecek page source dari web tersebut.

```
└──$ curl http://167.71.207.218:50620/

<!DOCTYPE html>

<html lang="en">

 <head>

  <meta charset="utf-8" />

  <link rel="icon" href="/favicon.ico" />

  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <meta name="theme-color" content="#000000" />

  <meta

   name="description"

   content="Web site created using create-react-app"

  />

  <link rel="apple-touch-icon" href="/logo192.png" />

  <link rel="manifest" href="/manifest.json" />

  <!--

   name :p[JKKJGSAjkn138

   email: KLDfhiewNfdj@asiud.ask

   password : MajsuiyUYAQop9375

  -->

  <title>React App</title>

 </head>

 <body>

  <noscript>You need to enable JavaScript to run this app.</noscript>

  <div id="root"></div>
```

```
  <!--
    halaman login
  -->
 <script src="/static/js/bundle.js"></script><script src="/static/js/0.chunk.js"></script><script src="/static/js/main.chunk.js"></script></body>
</html>
```

Namun sebenarnya dapat dibypass dengan mudah. Digunakan fuzzing dengan bantuan tools wfuzz untuk mengecek apakah terdapat hidden directory, dan ternyata benar terdapat directory /flag yang berisi flag didalamnya.

FLAG

WRECKIT40(1t_I5_n0T_T0_H4rD_Yyy34hh)