Write Up Gemastik 2022

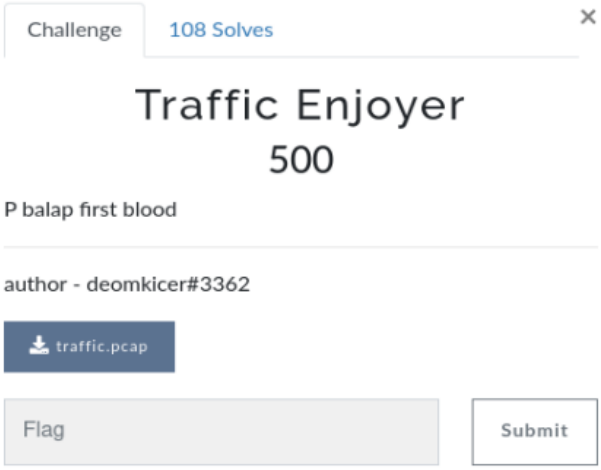
bin2hex

Firstiannisa
Muhammad Fadli Renel
Rehan Kurnia Hidayat

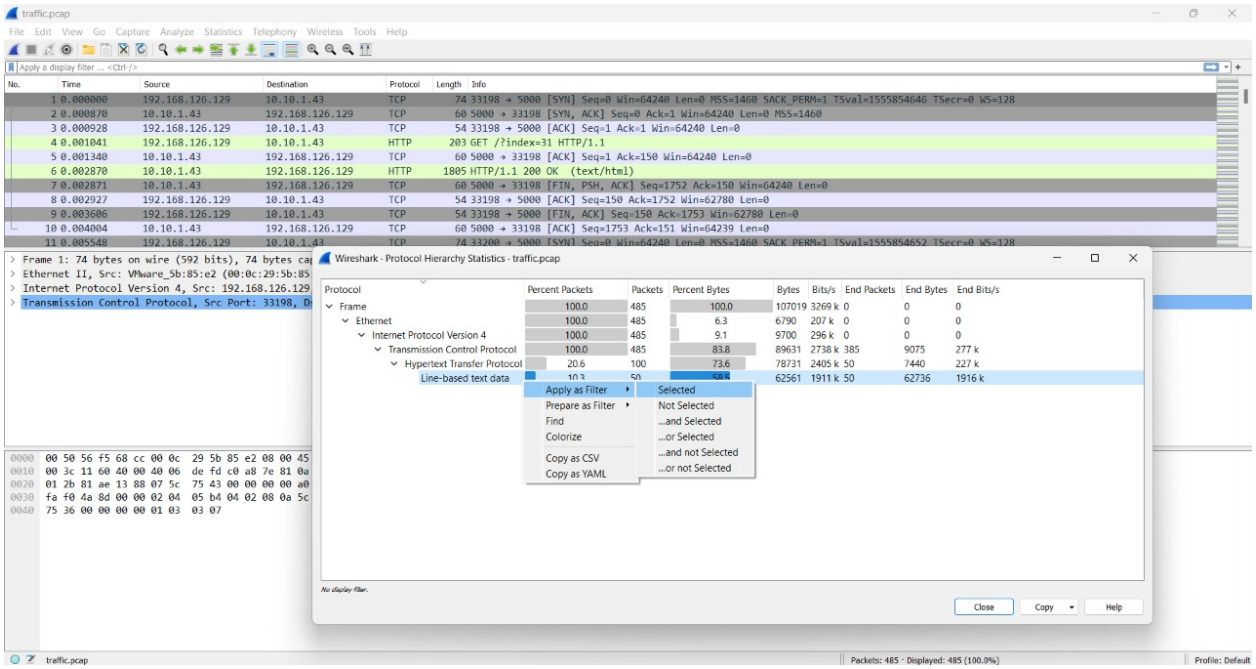
Daftar Isi

Forensic

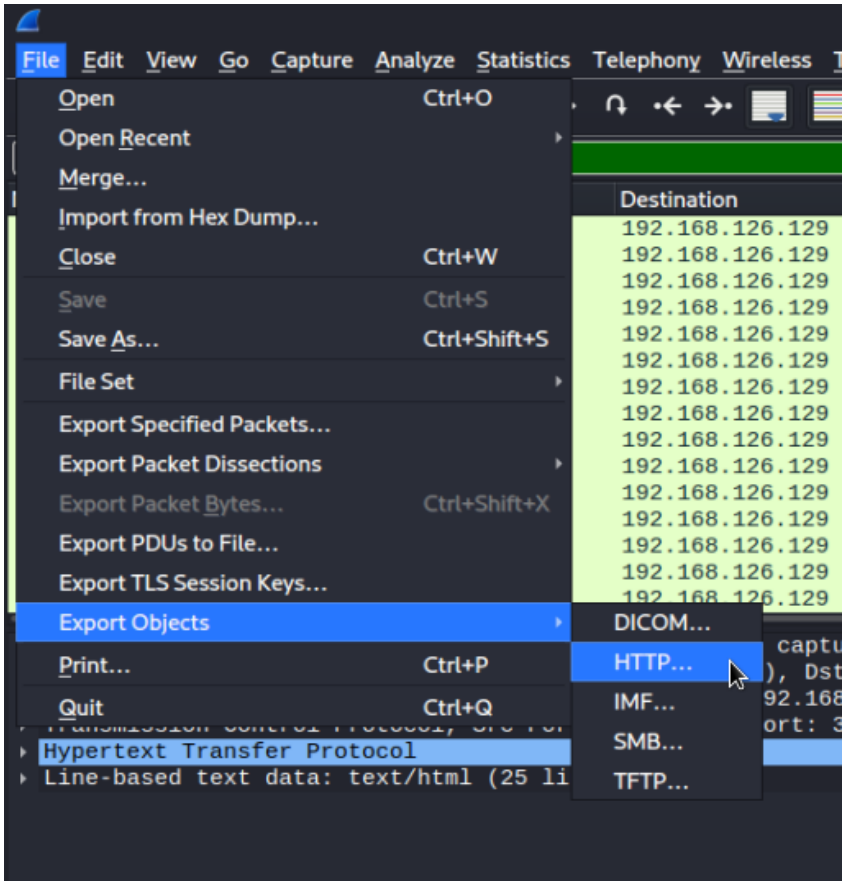
Traffic Enjoyer



Solusi :
Disediakan sebuah file traffic.pcap. File berekstensi .pcap ini merupakan sebuah berkas data berisi data paket jaringan yang dibuat selama penangkapan jaringan langsung. File ini dapat dibuka serta dianalisis menggunakan tools Wireshark. Ketika di import ke dalam Wireshark, terdapat banyak sekali data yang tercapture. Dari sini kita coba menggunakan filter di tab menu **Statistic > Protocol Hierarchy**. Setelah itu lakukan filter pada **Line-based text data.**

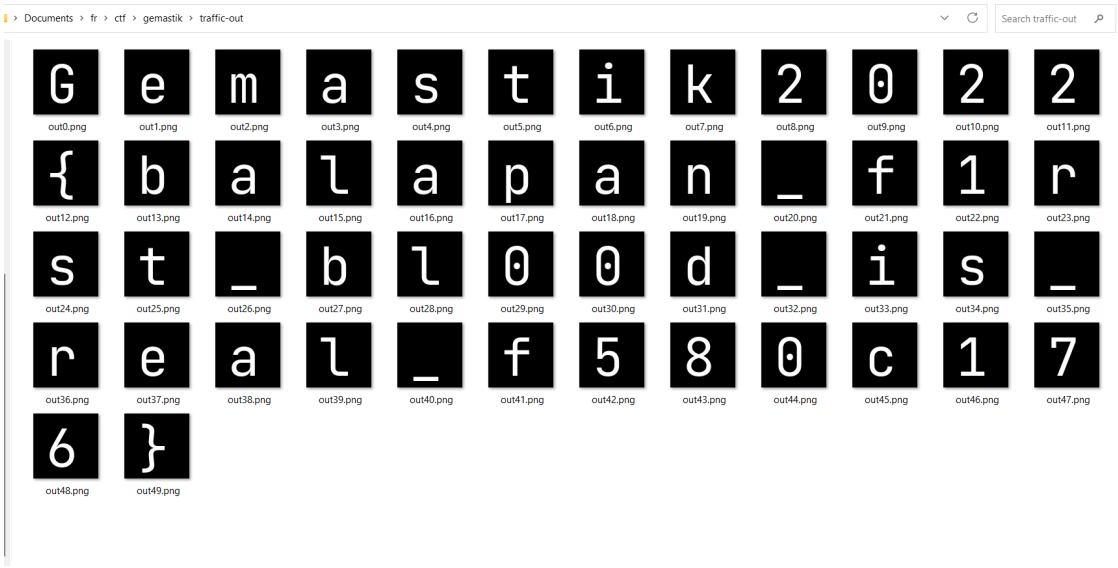


Jika kita lihat salah satu isinya, data yang ada berupa string base64 dan bila di-decode menghasilkan hex bytes file PNG.
Maka, lakukan **Export Objects > HTTP**



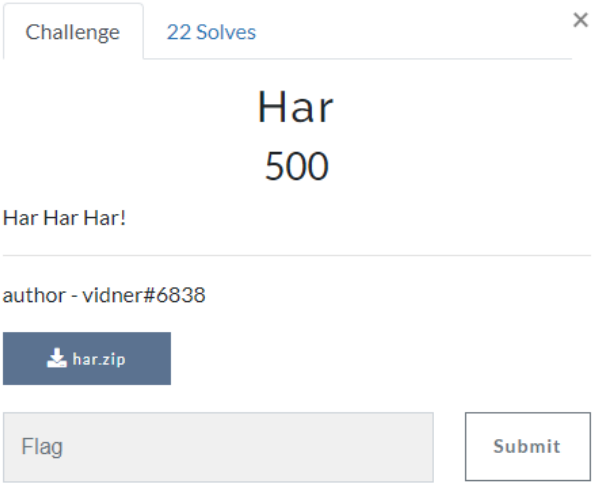
Kemudian, decode semua isi file dan simpan ke dalam file PNG. Untuk memudahkan proses ini, kami membuat kode program seperti berikut.

Berikut hasil file PNG yang didapat

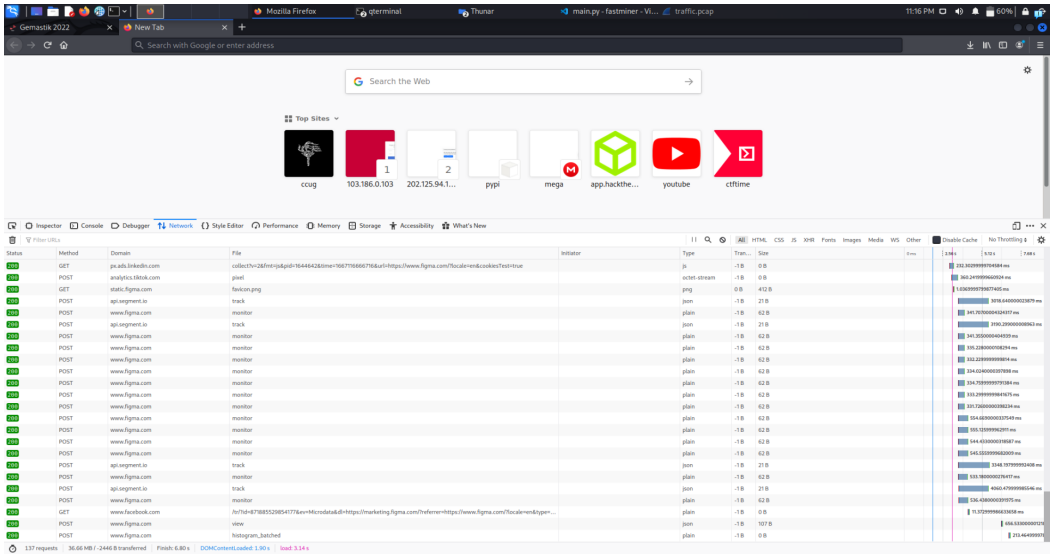


Gemastik2022{balapan_f1rst_bl00d_is_real_f580c176}

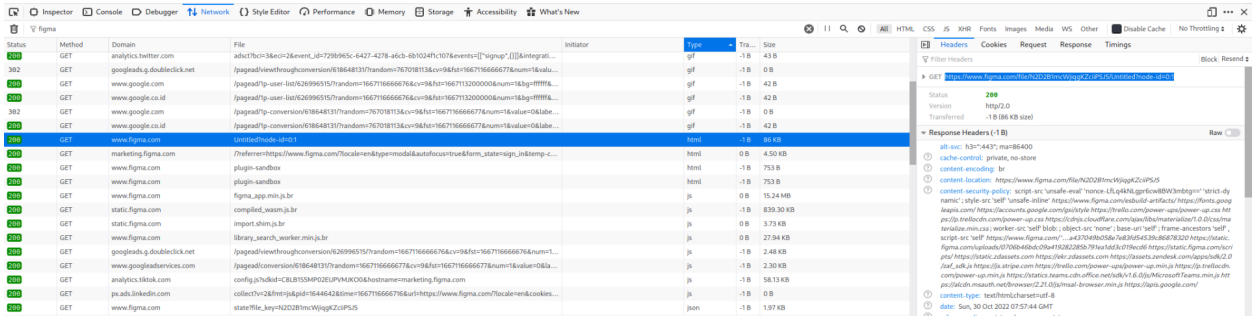
Har



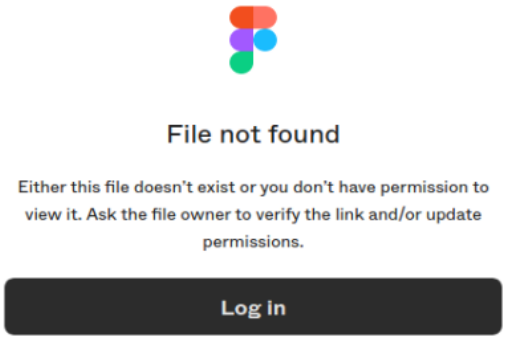
Solusi :
Terdapat sebuah file berbentuk zip, yang ketika diekstrak berisi file **gemasteg.har.** File yang berekstensi .har ini merupakan sebuah format file JSON Archive yang menyimpan semua data permintaan web yang dibuat di sebuah browser. Untuk membuka file ini bisa langsung saja import file .har tersebut ke **Developer Tools** tab bagian **Network** pada browser.



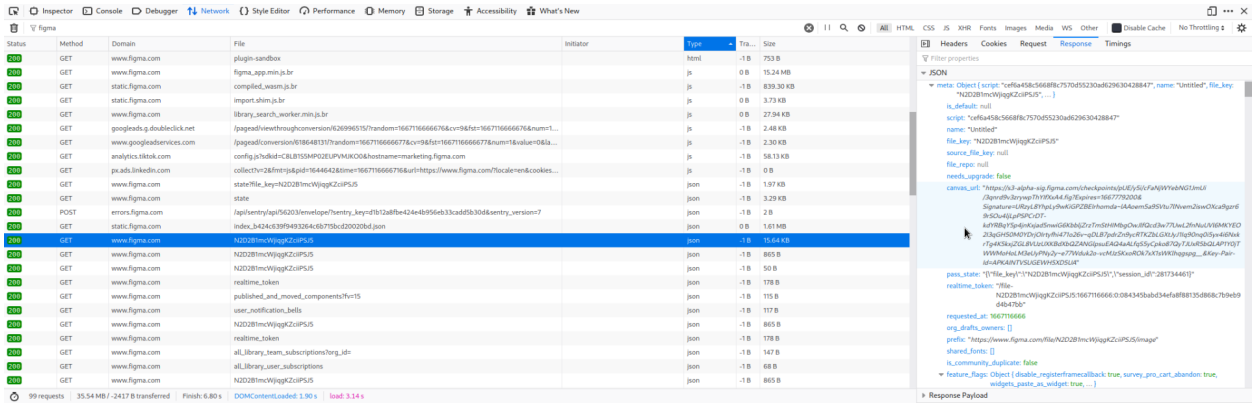
Dari sini kita sudah dapat melihat berbagai macam data yang pernah di akses, mayoritas data mengakses ke sebuah situs yaitu Figma. Kita ingin mencari link project figmanya.



Pada saat membuka link project figmanya, ternyata kita tidak memiliki permission untuk mengakses file tersebut.

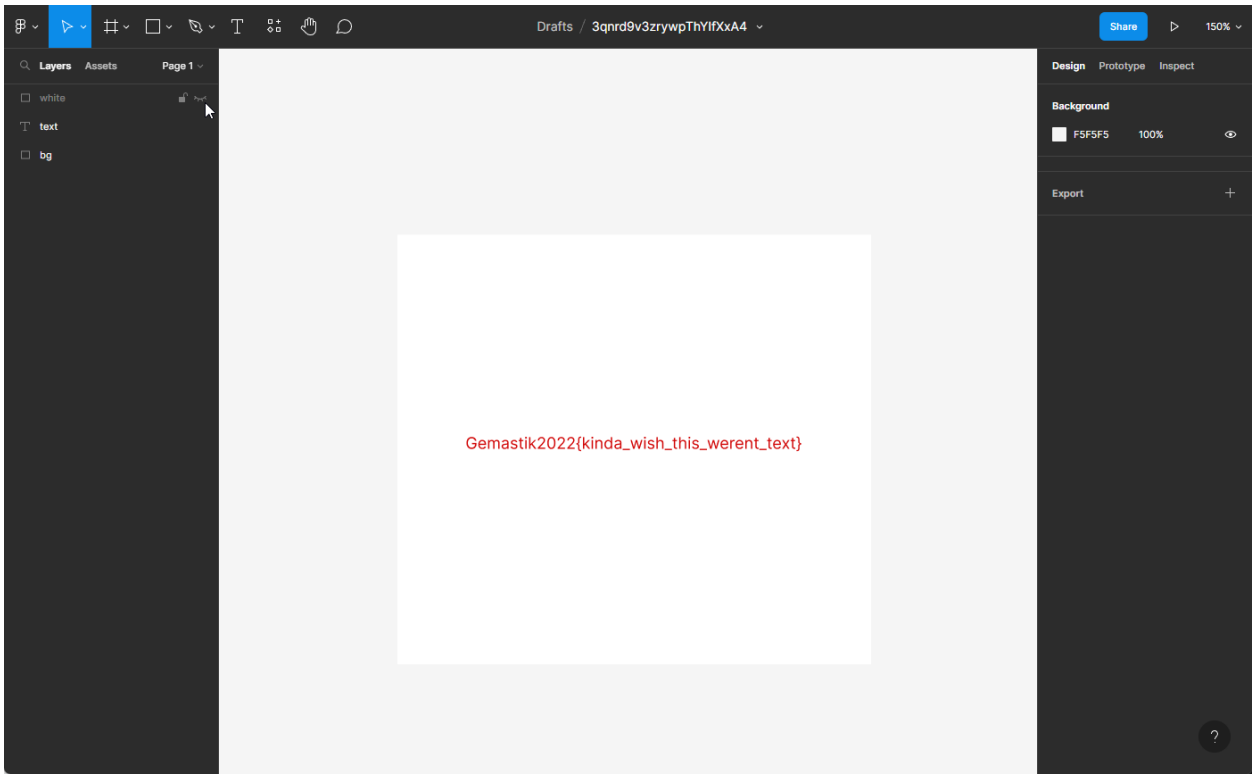


Dari sini kita harus mencari sebuah link yang diberikan oleh response awal ketika file figma ini dibuka. We got the file!



Ketika link tersebut dibuka maka akan mendownload sebuah file berekstensi .fig . yang mana ekstensi tersebut merupakan sebuah file project figma. Langsung saja di import ke dalam Figma nya dan flagnya adalah

Gemastik2022{kinda_wish_this_werent_text}



Reverse Engineering

CodeJugling

Solusi :
Diberikan sebuah stripped binary file yang berarti informasi debugging dari program tersebut sudah dihilangkan, termasuk nama-nama fungsi yang digunakan.

```
ACER@LAPTOP-8OO6AT9J MINGW64 ~/Documents/fr/ctf/gemastik
$ file reversing-itu-mudah
reversing-itu-mudah: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynami
cally linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 3.2.0, Buil
dID[sha1]=fb53cf74b3f119b9d8ff345ae7fde4b43680a01c, stripped
```

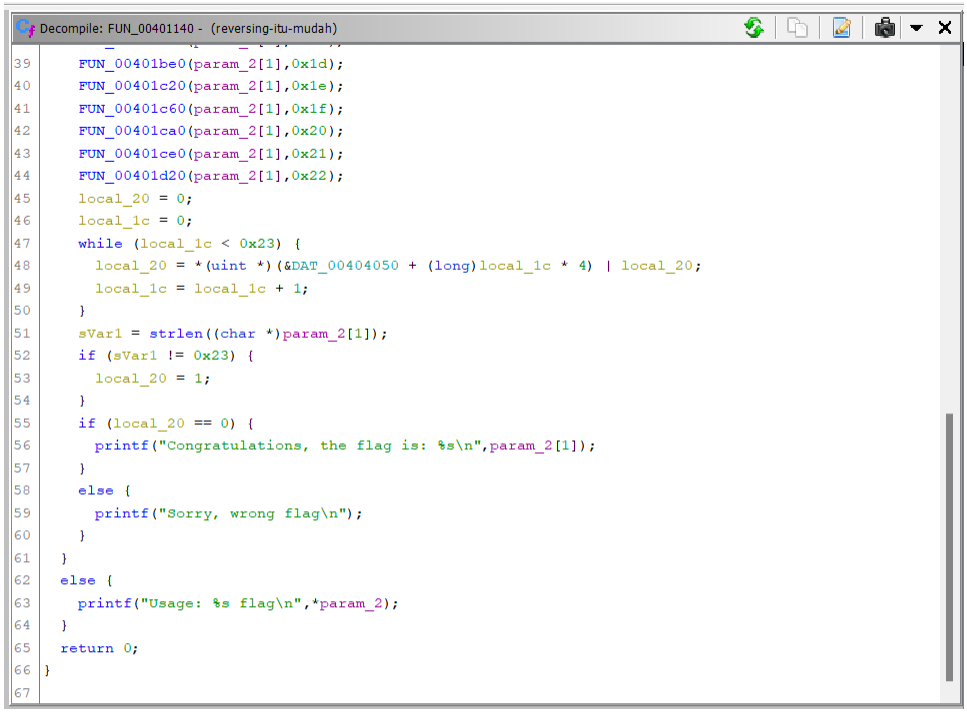
Selanjutnya, disassembly file menggunakan Ghidra dan cek fungsi-fungsi yang ada. Terlihat ada salah satu fungsi yang melakukan pengecekan input dengan memanggil fungsi lain. Jika inputan benar, maka program akan menampilkan flag (yang tak lain dan tak bukan adalah inputan itu sendiri)

```
Decompile: FUN_00401140 - (reversing-itu-mudah)

undefined4 FUN_00401140(int param_1,undefined8 *param_2)

{
  size_t sVar1;
  uint local_20;
  int local_1c;

  if (param_1 == 2) {
    FUN_004014a0(param_2[1],0);
    FUN_004014e0(param_2[1],1);
    FUN_00401520(param_2[1],2);
    FUN_00401560(param_2[1],3);
    FUN_004015a0(param_2[1],4);
    FUN_004015e0(param_2[1],5);
    FUN_00401620(param_2[1],6);
    FUN_00401660(param_2[1],7);
    FUN_004016a0(param_2[1],8);
    FUN_004016e0(param_2[1],9);
    FUN_00401720(param_2[1],10);
    FUN_00401760(param_2[1],0xb);
    FUN_004017a0(param_2[1],0xc);
    FUN_004017e0(param_2[1],0xd);
    FUN_00401820(param_2[1],0xe);
    FUN_00401860(param_2[1],0xf);
    FUN_004018a0(param_2[1],0x10);
    FUN_004018e0(param_2[1],0x11);
    FUN_00401920(param_2[1],0x12);
    FUN_00401960(param_2[1],0x13);
```



```
Decompile: FUN_00401140 - (reversing-itu-mudah)

    FUN_00401be0(param_2[1],0x1d);
    FUN_00401c20(param_2[1],0x1e);
    FUN_00401c60(param_2[1],0x1f);
    FUN_00401ca0(param_2[1],0x20);
    FUN_00401ce0(param_2[1],0x21);
    FUN_00401d20(param_2[1],0x22);
    local_20 = 0;
    local_1c = 0;
    while (local_1c < 0x23) {
      local_20 = *(uint *)(&DAT_00404050 + (long)local_1c * 4) | local_20;
      local_1c = local_1c + 1;
    }
    sVar1 = strlen((char *)param_2[1]);
    if (sVar1 != 0x23) {
      local_20 = 1;
    }
    if (local_20 == 0) {
      printf("Congratulations, the flag is: %s\n",param_2[1]);
    }
    else {
      printf("Sorry, wrong flag\n");
    }
  }
  else {
    printf("Usage: %s flag\n",*param_2);
  }
  return 0;
}
```

Nah sekarang kita lihat fungsi-fungsi yang digunakan untuk mengecek input tadi. Setiap fungsi tersebut ternyata melakukan perbandingan tiap karakter dari inputan dengan sebuah karakter. Kita susun saja setiap karakter yang dibandingkan tersebut dan flag pun berhasil didapatkan

Gemastik2022{st45iUn_MLG_k07a_b4rU}

**) karena fungsinya cukup banyak, jadi kami hanya menampilkan 8 fungsi pertama saja yang membentuk potongan kata “Gemastik” sebagai contoh ✌️*

Decompile: FUN_004014a0 - (reversing-itu-mudah)

```
void FUN_004014a0(long param_1,int param_2)

{
  *(uint *)(&DAT_00404050 + (long)param_2 * 4) = (uint)(*(char *)(param_1 + param_2) != 'G');
  return;
}
```

```
void FUN_004014e0(long param_1,int param_2)

{
  *(uint *)(&DAT_00404050 + (long)param_2 * 4) = (uint)(*(char *)(param_1 + param_2) != 'e');
  return;
}
```

```
void FUN_00401520(long param_1,int param_2)

{
  *(uint *)(&DAT_00404050 + (long)param_2 * 4) = (uint)(*(char *)(param_1 + param_2) != 'm');
  return;
}
```

```
void FUN_00401560(long param_1,int param_2)

{
  *(uint *)(&DAT_00404050 + (long)param_2 * 4) = (uint)(*(char *)(param_1 + param_2) != 'a');
  return;
}
```

```
void FUN_004015a0(long param_1,int param_2)

{
  *(uint *)(&DAT_00404050 + (long)param_2 * 4) = (uint)(*(char *)(param_1 + param_2) != 's');
  return;
}
```

```
void FUN_004015e0(long param_1,int param_2)

{
  *(uint *)(&DAT_00404050 + (long)param_2 * 4) = (uint)(*(char *)(param_1 + param_2) != 't');
  return;
}
```

```
void FUN_00401620(long param_1,int param_2)

{
  *(uint *)(&DAT_00404050 + (long)param_2 * 4) = (uint)(*(char *)(param_1 + param_2) != 'i');
  return;
}
```

```
void FUN_00401660(long param_1,int param_2)

{
  *(uint *)(&DAT_00404050 + (long)param_2 * 4) = (uint)(*(char *)(param_1 + param_2) != 'k');
  return;
}
```