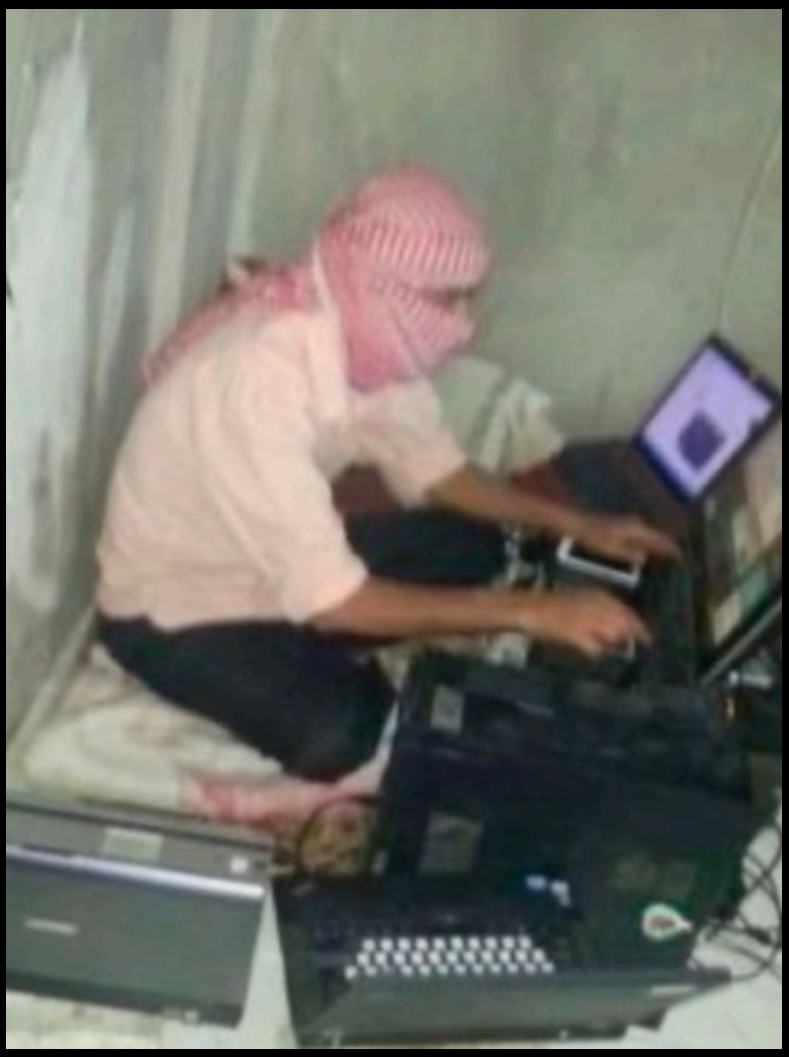
WRITEUP CYBER SECURITY
GEMASTIK 18 (2025)
Team *cacicu.exe*



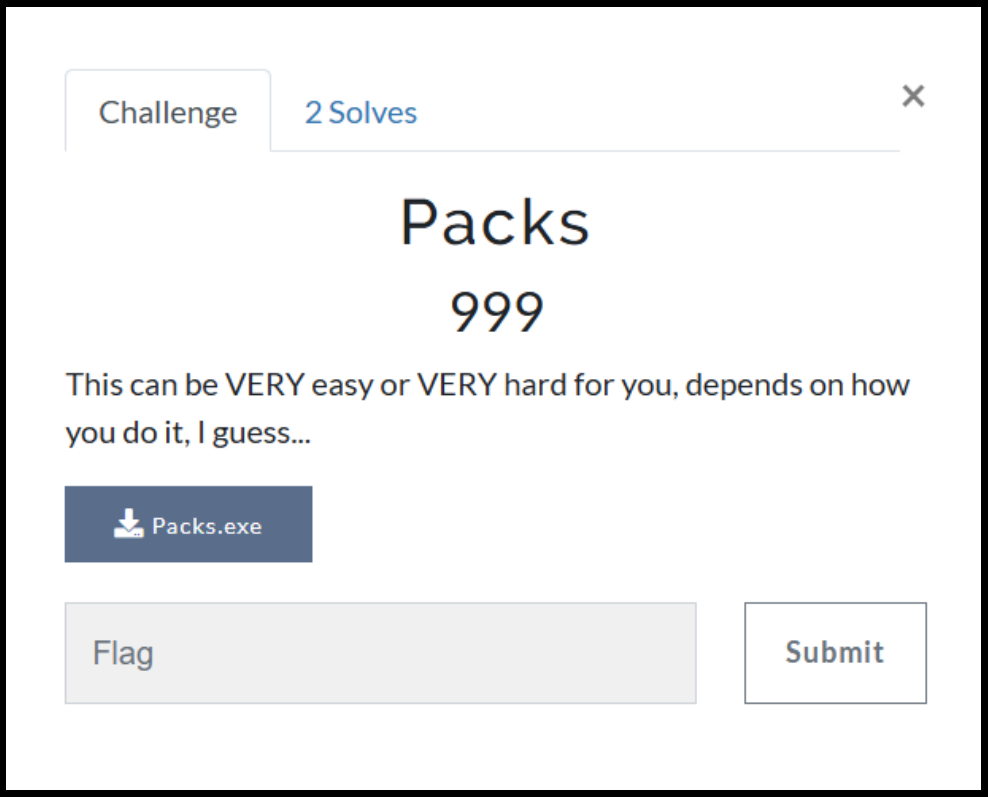
Anggota :

Harri Supriadi
Muhammad 'Azmi Salam
Anas Miftakhul Falah

Table of Contents

Reverse

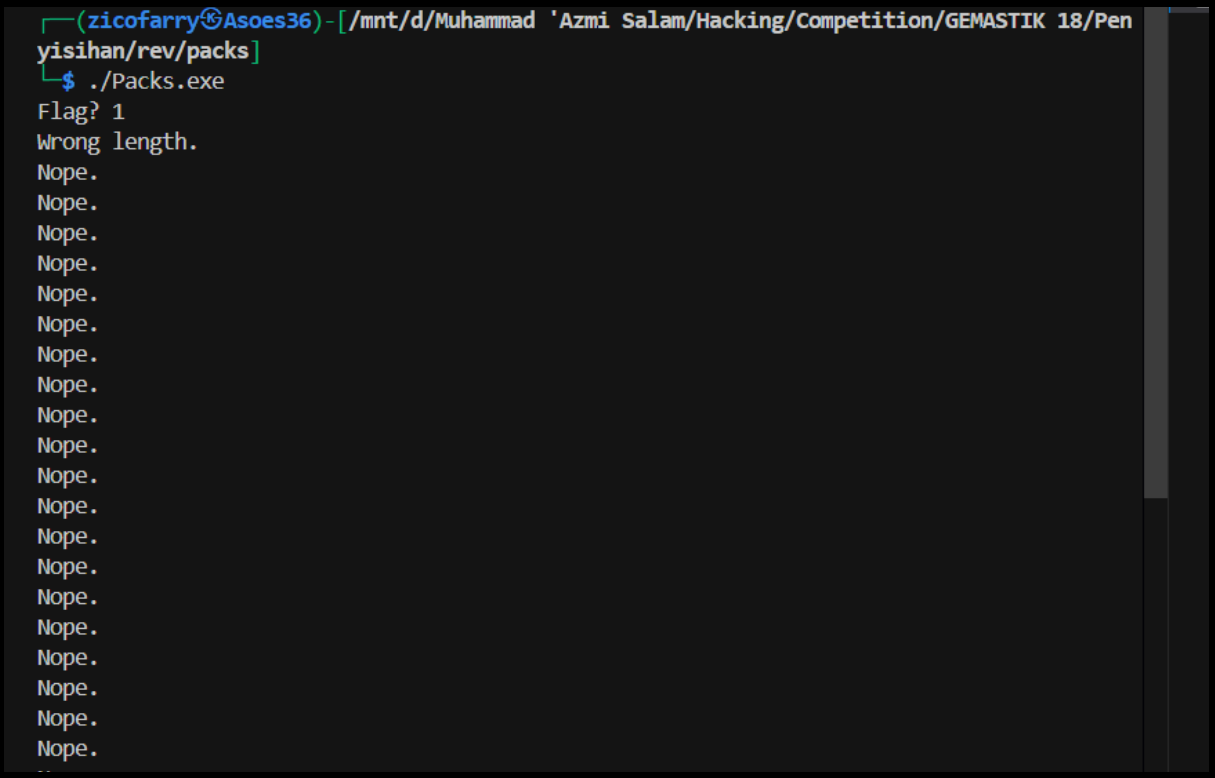
1. Packs



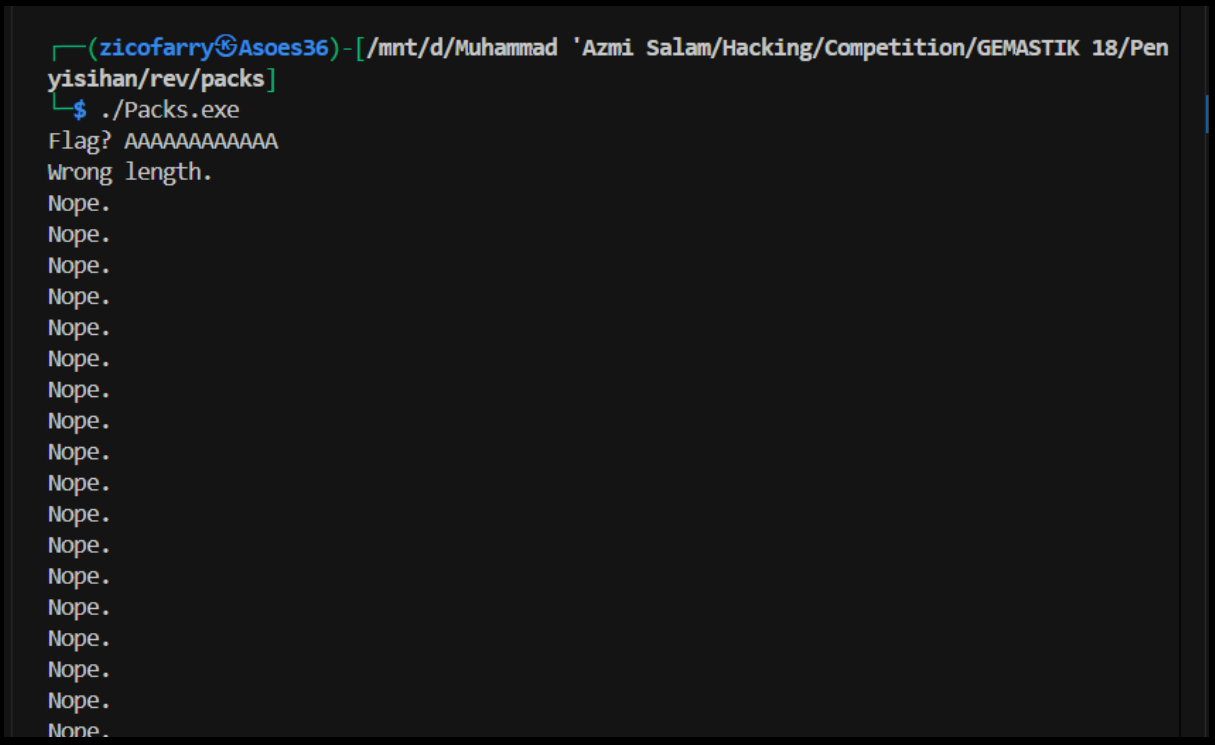
Deskripsi :
Diberikan attachment berupa file Packs.exe lalu deskripsinya menyebutkan **"This can be VERY easy or VERY hard for you, depends on how you do it. I guess…"** dari keterangan tersebut saya berasumsi bahwa ada cara cepat yang lebih mudah untuk solve soal ini.

Solusi :
Setelah mencoba menjalankan file executable nya, itu muncul output **flag?** lalu kita diminta untuk memasukan input.

Untuk percobaan pertama, saya mencoba memasukan angka random yaitu 1, dan outputnya seperti ini.

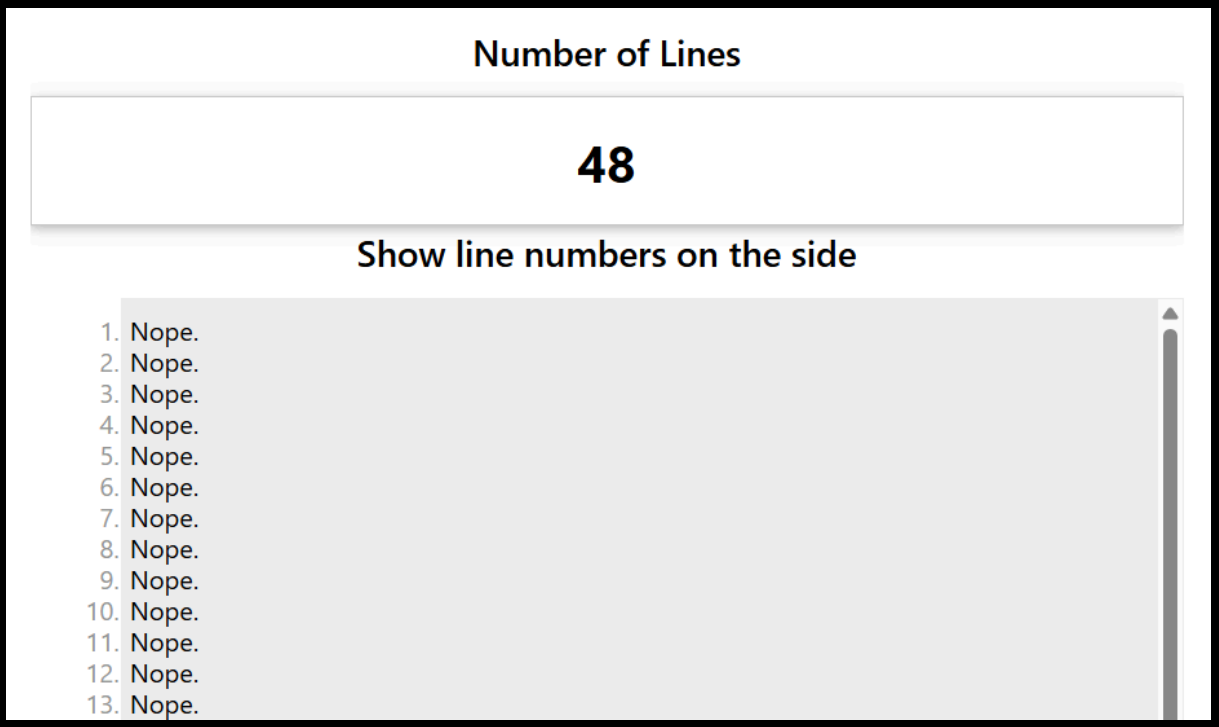


lalu saya memasukan input random lain berupa string.

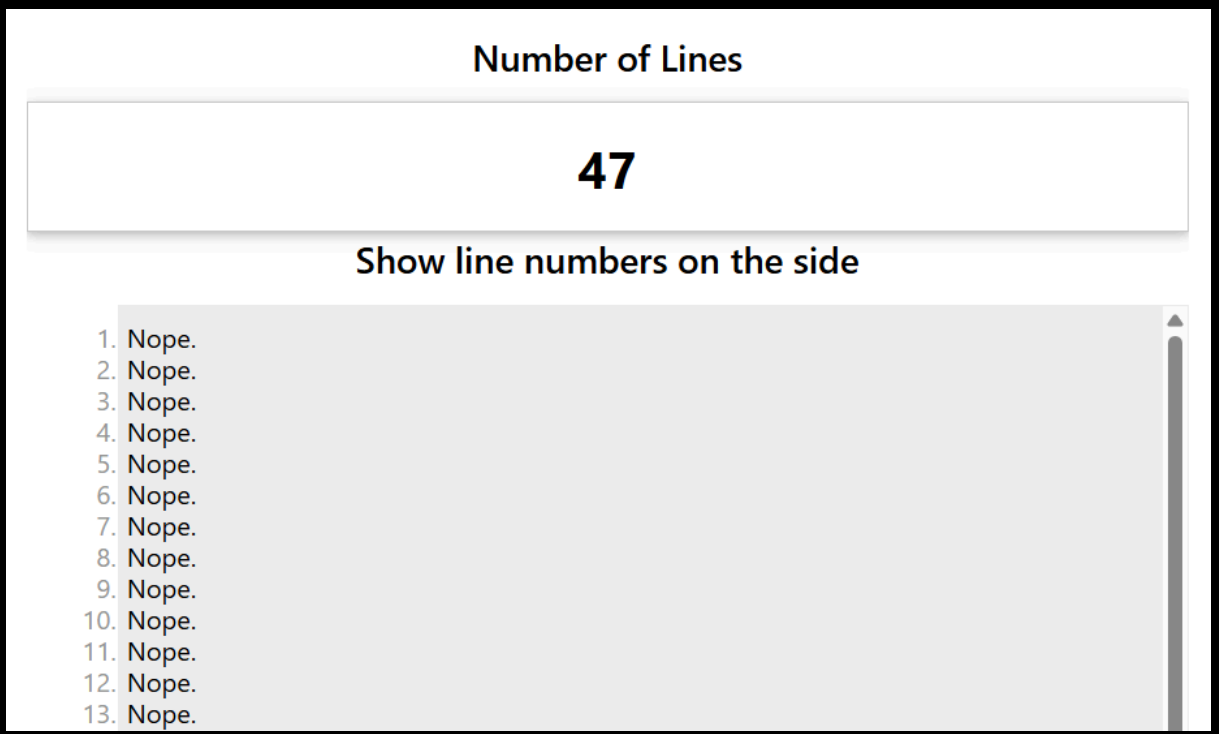


outputnya sama yaitu keluaran **Nope.** yang sangat banyak, tetapi karena ada kata "Wrong Length." saya jadi terpikirkan, jangan-jangan length nya itu mengarah kepada panjang dari flag aslinya.

Dari sana saya langsung mencari line counter agar cepat untuk menghitung jumlah **Nope.** yang ada pada output tersebut, ketika saya memasukan 1, output **Nope.** nya ada sebanyak 48 line.



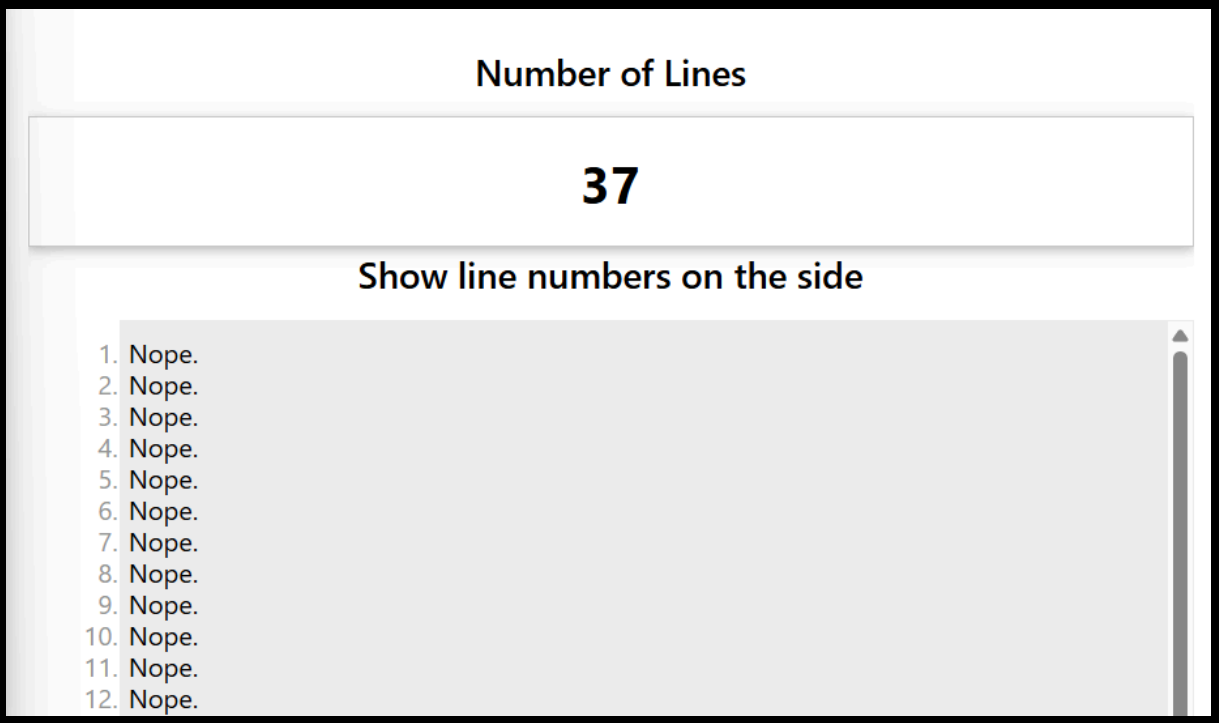
Tetapi ketika saya mencopy output dari masukan AAAAAAAAAAA itu output **Nope.** nya ada sebanyak 47 line.



Dari sini saya langsung melihat ada peluang untuk mendapatkan flag berdasarkan respon dari file exe nya. Setelah itu, karena sudah tau format flagnya adalah "GEMASTIK18{" saya langsung saja mencoba memasukan string itu sebagai masukan, dan menurut asumsi saya harusnya output **Nope.** nya itu akan berjumlah 37, karena 48 - 11 (11 itu panjang dari "GEMASTIK18{") adalah 37.

```
┌──(zicofarry㉿Asoes36)-[/mnt/d/Muhammad 'Azmi Salam/Hacking/Competition/GEMASTIK 18/Penyisihan/rev/packs]
└─$ ./Packs.exe
Flag? GEMASTIK18{
Wrong length.
Nope.
Nope.
Nope.
Nope.
Nope.
Nope.
Nope.
Nope.
Nope.
Nope.
Nope.
```

danyapp ternyata benar jumlah **Nope.** nya adalah 37



Karena sudah tau peluang bisa mendapatkan flag berdasarkan responnya, saya langsung membuat script solver untuk melakukan brute force pada setiap karakter, dengan logic ketika karakternya benar maka jumlah **Nope.** nya akan berkurang, brute force akan terus berlangsung sampai 48 karakter dan output **Nope.** nya tersisa 0.

solver.py

```
#!/usr/bin/env python3
import subprocess
import sys
import time

BINARY = "./Packs.exe"
WINE = False
CHARS =
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789_{}-!@#$
%^&*()[]:;<>?,./\\|+ "
L = 48  # panjang flag (didapatkan dari jumlah Nope. nya yaitu 48)

# prefix yang sudah diketahui (format flag)
prefix_known = list("GEMASTIK18{")

def run_input(s):
    cmd = [BINARY]
    if WINE:
        cmd = ["wine", BINARY]
    try:
        p = subprocess.run(cmd, input=(s+"\n").encode(),
                           capture_output=True, timeout=4)
    except subprocess.TimeoutExpired:
        return None
    return p.stdout.decode(errors="ignore")

def count_nope(out):
    return out.count("Nope.")

filler = "#"
# flag = list(filler * L) # kalo mau bruteforce dari awal
flag = prefix_known + [filler] * (L - len(prefix_known))    # untuk
mempercepat proses bruteforce, jadi pake prefix yang udah diketahui

# for i in range(L): # kalo mau bruteforce dari awal
```

```
for i in range(len(prefix_known), L):    # mulai bruteforce setelah
prefix_known
    candidate_base = "".join(flag[:i]) + filler*(L-i)
    out_base = run_input(candidate_base)
    if out_base is None:
        print("Program timeout saat baseline.")
        sys.exit(1)
    base_nope = count_nope(out_base)

    found = False
    for c in CHARS:
        candidate = "".join(flag[:i]) + c + filler*(L-i-1)
        out = run_input(candidate)
        if out is None:
            continue
        n = count_nope(out)

        # update flag sementara
        flag[i] = c

        # persentase progress
        progress_percent = (i + 1) / L * 100

        # print progress bar + Nope count
        bar_length = 40
        filled_length = int(bar_length * (i + 1) / L)
        bar = "=" * filled_length + "-" * (bar_length - filled_length)
           sys.stdout.write(f"\r[{bar}] {progress_percent:.1f}% | Nope:
{n}")
        sys.stdout.flush()

        # print flag sementara di baris bawah
        sys.stdout.write(f"\n{''.join(flag)}")
        sys.stdout.flush()
        sys.stdout.write("\033[F")  # pindah cursor satu baris ke atas
        sys.stdout.flush()

        if n < base_nope:
            found = True
            break

    if not found:
        flag[i] = filler
```

```
# pindah baris setelah selesai
sys.stdout.write("\n\nRecovered flag: " + "".join(flag) + "\n")

# GEMASTIK18{S1mpl3_P4ck3r_f0r_4_S1mpl3_Ch4ll3nge}
```

Prosesnya cukup lama untuk mendapatkan flag secara keseluruhan (sekitar 30 menit).

```
┌──(zicofarry㉿Asoes36)-[/mnt/d/Muhammad 'Azmi Salam/Hacking/Competition/GEMASTIK 18/Penyisihan/rev/packs]
└─$ python3 solver.py
[=========================----------] 68.8% | Nope: 16
GEMASTIK18{S1mpl3_P4ck3r_f0r_4_SI################
```

tetapi karena sudah lumayan terlihat jelas bahwa script ini bekerja dengan baik untuk mendapatkan flagnya, saya bisa mengerjakan soal lain terlebih dahulu sembari menunggu brute force nya berhasil untuk menemukan flag secara penuh.

Dan ini adalah flag akhir nya setelah menunggu cukup lama.

```
┌──(zicofarry㉿Asoes36)-[/mnt/d/Muhammad 'Azmi Salam/Hacking/Competition/GEMASTIK 18/Penyisihan/rev/packs]
└─$ python3 solver.py
[====================================] 100.0% | Nope: 0
GEMASTIK18{S1mpl3_P4ck3r_f0r_4_S1mpl3_Ch4ll3nge}
Recovered flag: GEMASTIK18{S1mpl3_P4ck3r_f0r_4_S1mpl3_Ch4ll3nge}
```

Flag :
GEMASTIK18{S1mpl3_P4ck3r_f0r_4_S1mpl3_Ch4ll3nge}