LKS SMK 2022 XXX Cyber Security

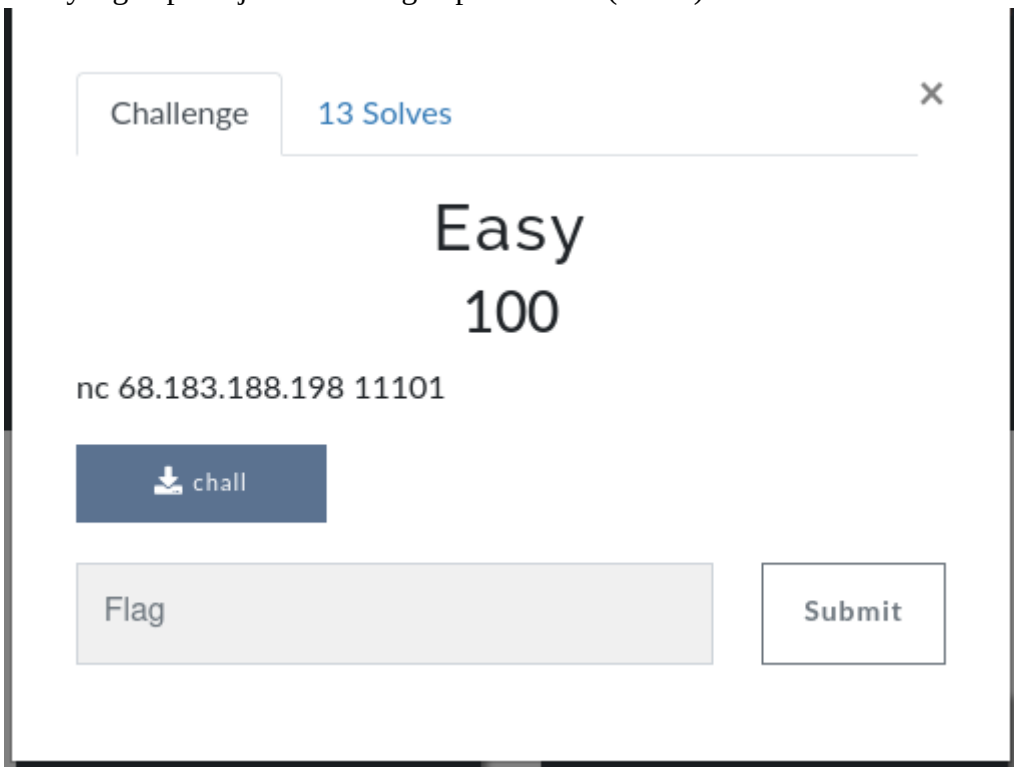
Writeup Day 2 (CTF Jeopardy Competition)

Di Day 2 ini, kami diberikan tantangan berupa soal-soal CTF (Capture The Flag). Berikut adalah beberapa challenge yang berhasil kami selesaikan.

A. Binary Exploitation

1. Easy

   Di challenge ini, terdapat sebuah file executeable dan juga deskripsi berupa IP dan Port yang dapat dijalankan dengan perintah nc (netcat).



   Hal pertama yang kita lakukan adalah menjalankan netcat IP di terminal atau menjalankan file chall yang sudah di download.

```
┌──(kali㉿kali)-[~/LKS-Nasional/Day 2/easy]
└─$ nc 68.183.188.198 11101
You have old laptop, then you want to sell it to buy flag
How much do you want to sell your laptop ?
```

   Ketika dijalankan, akan muncul pertanyaan seperti di atas. Dari sini, kita bisa

mendapatkan flag nya dengan cara memasukkan sejumlah angka negatif, yang dimana saya memasukkan angka -1000 untuk mendapatkan flag-nya.

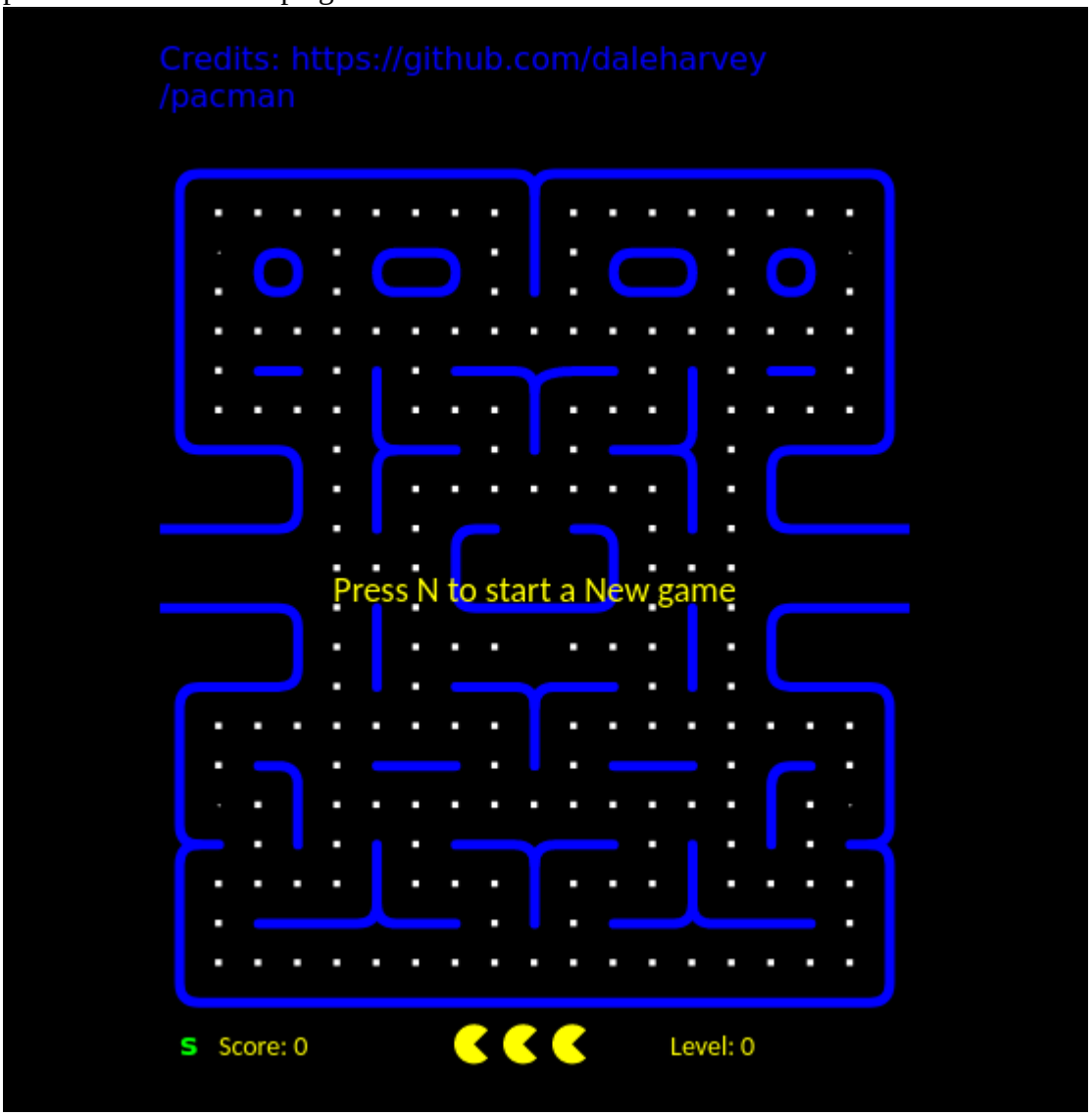
```
┌──(kali㉿kali)-[~/LKS-Nasional/Day 2/easy]
└─$ nc 68.183.188.198 11101
You have old laptop, then you want to sell it to buy flag
How much do you want to sell your laptop ? -1000
OK you have -1000$
Ok you will get bonus : LKSN{basic_integer_overflow}
```

Flag : LKSN{basic_integer_overflow}

B. Web Exploitation

1. Arcade Sidescreen

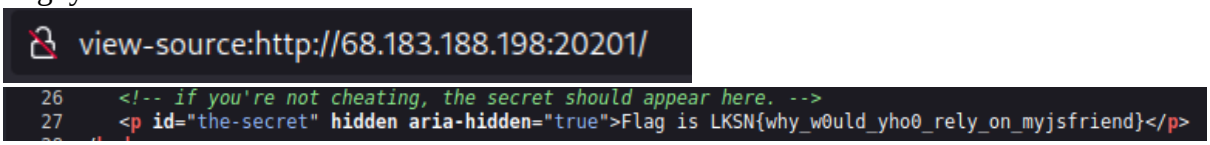
   Di challenge ini, terdapat sebuah link yang jika dibuka, akan mengarahkan kita pada sebuah web berupa game *PacMan*.



   Untuk mendapatkan flag-nya, kita perlu mengakses Page Source dari web ini. Biasanya untuk mengakses web page source itu bisa dilakukan dengan klik kanan

lalu klik View Page Source. Akan tetapi, jika kita melakukan itu, web akan memuncul notifikasi bahwa kita melakukan kecurangan atau *cheating*.

Untuk mengakali ini, kita akan melakukan ketik manual pada search browser yang kita gunakan. Ketik view-source:http://68.183.188.198:20201/. Dan akan muncul flagnya.

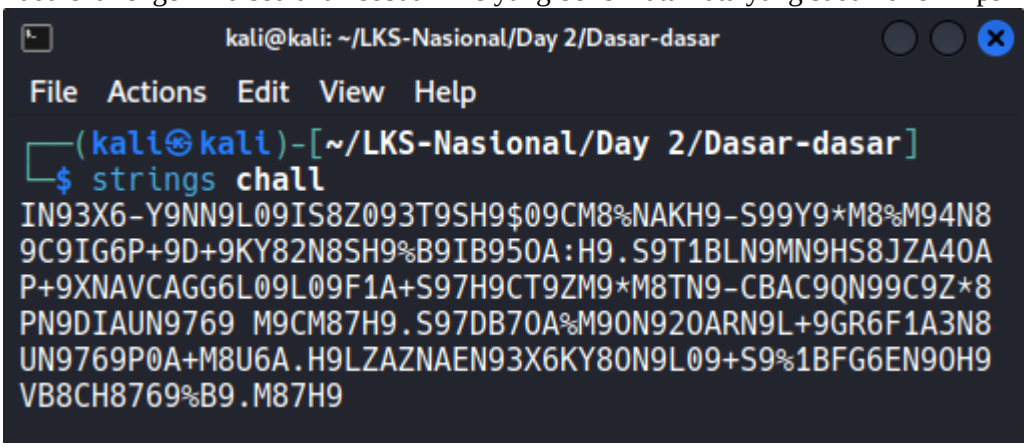


Flag : LKSN {why_w0uld_yho0_rely_on_myjsfriend}

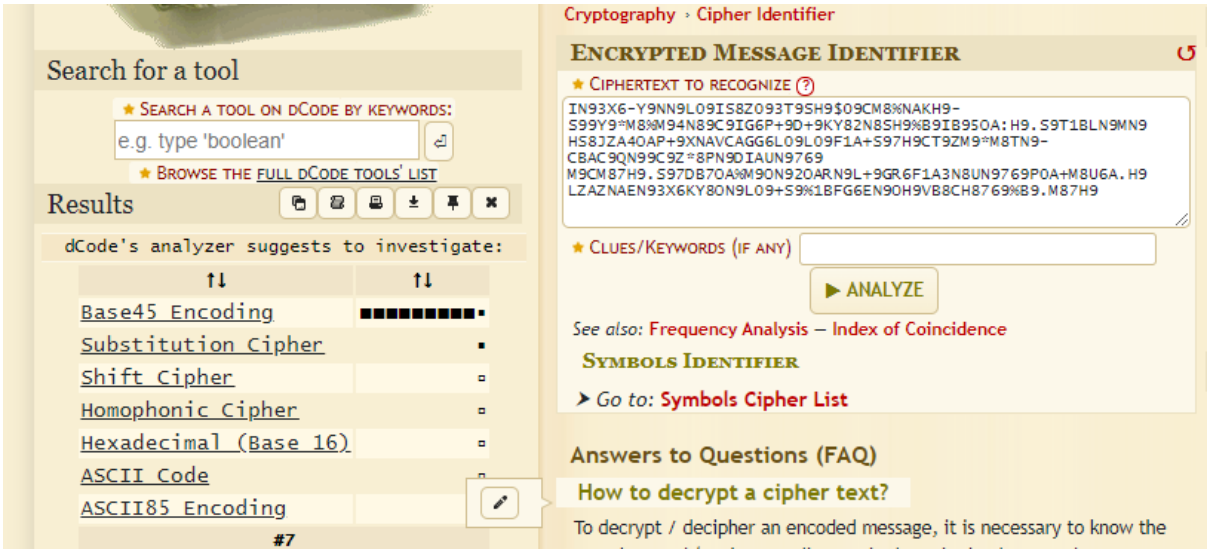
C. Cryptography

1. Dasar-dasar

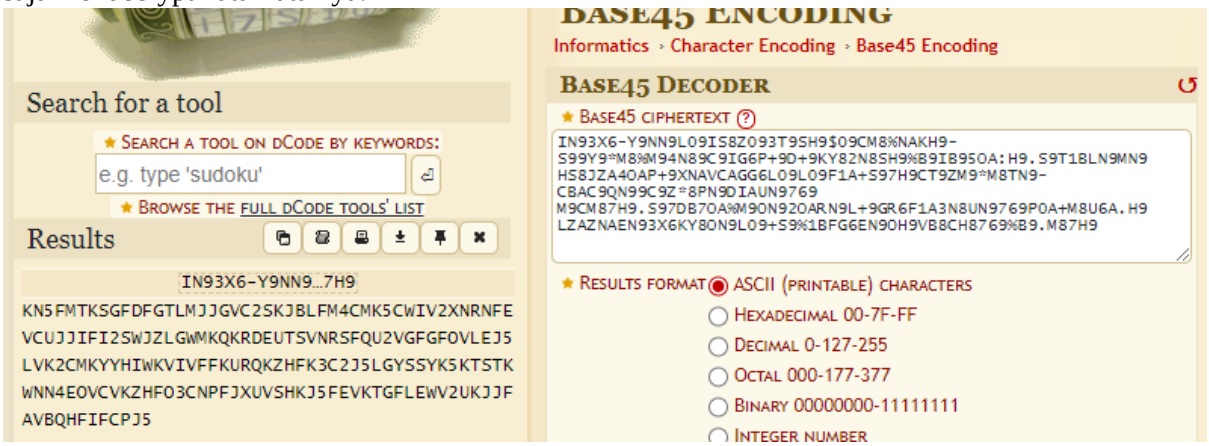
Pada challenge ini disediakan sebuah file yang berisi kata-kata yang sudah di enkripsi



Awalnya kami tidak mengetahui jenis enkripsi apa kata-kata di atas, akhirnya kami memakai online tool yang Bernama *Cipher Identifier* pada link berikut https://www.dcode.fr/cipher-identifier .



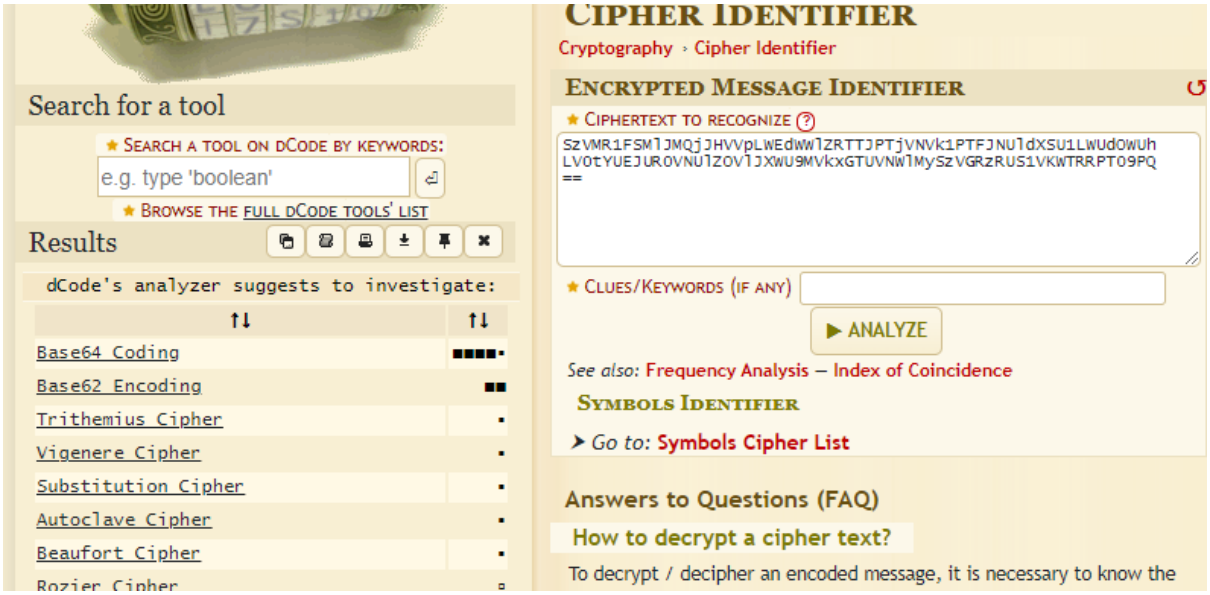
Dengan tool online tersebut, kita dapat mengetahui jenis enkripsi apa yang digunakan, setelah mengetahui bahwa enkripsi yang digunakan adalah Base45 maka kita langsung saja mendecrypt kata-kata nya.



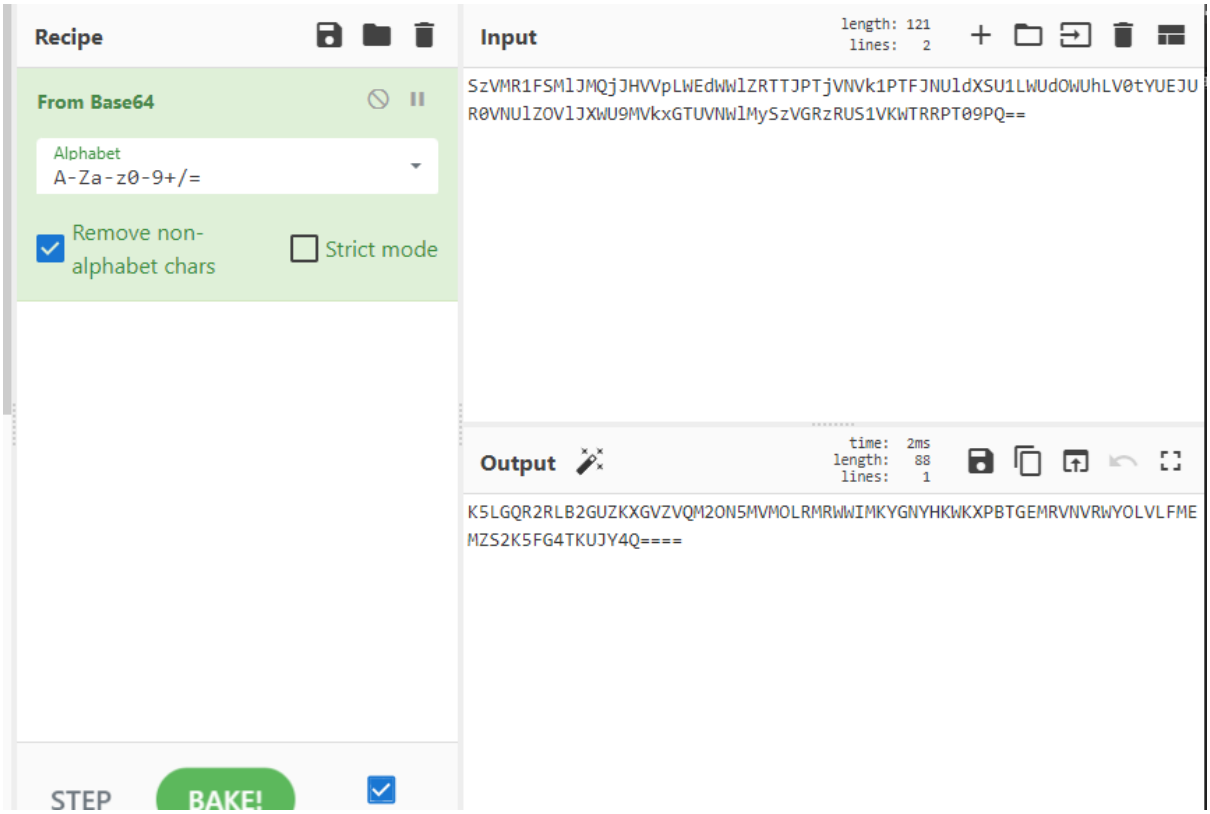
Terlihat terdapat hasil yang masih berupa kata-kata enkripsi, maka kita identifikasi lagi jenis enkripsi tersebut dengan tool *Cipher Identifier* sebelumnya.

Dan didapatkan hasil bahwa kata-kata tersebut merupakan Base32, maka kita decrypt lagi kata-katanya.

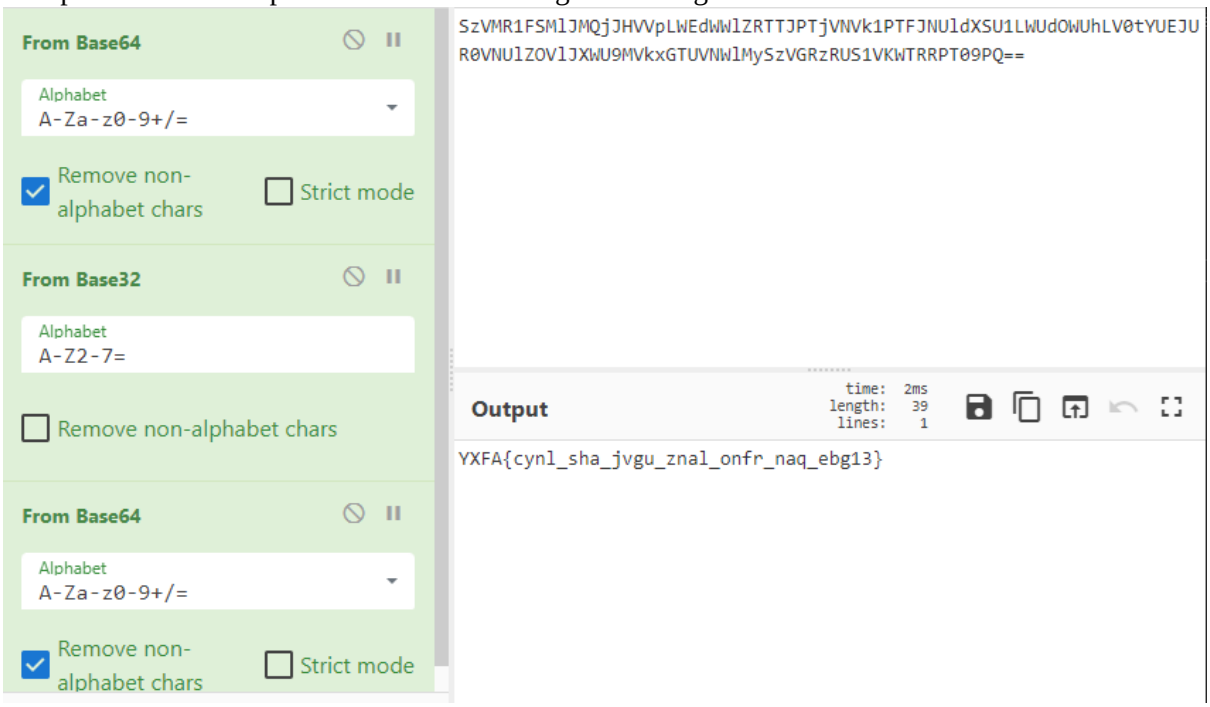
Dan lagiiii, kita mendapatkan hasil yang masih terinkripsi, maka kita ulangi lagi langkah-langkah sebelumnya sampai kita mendapatkan hasil yang benar-benar sudah terdecrypt dan bisa dibaca oleh manusia.



Setelah mengetahui bahwa enkripsi tersebut Base64 maka kita akan menggunakan online tools yang lain yaitu *Cyber Chef* .

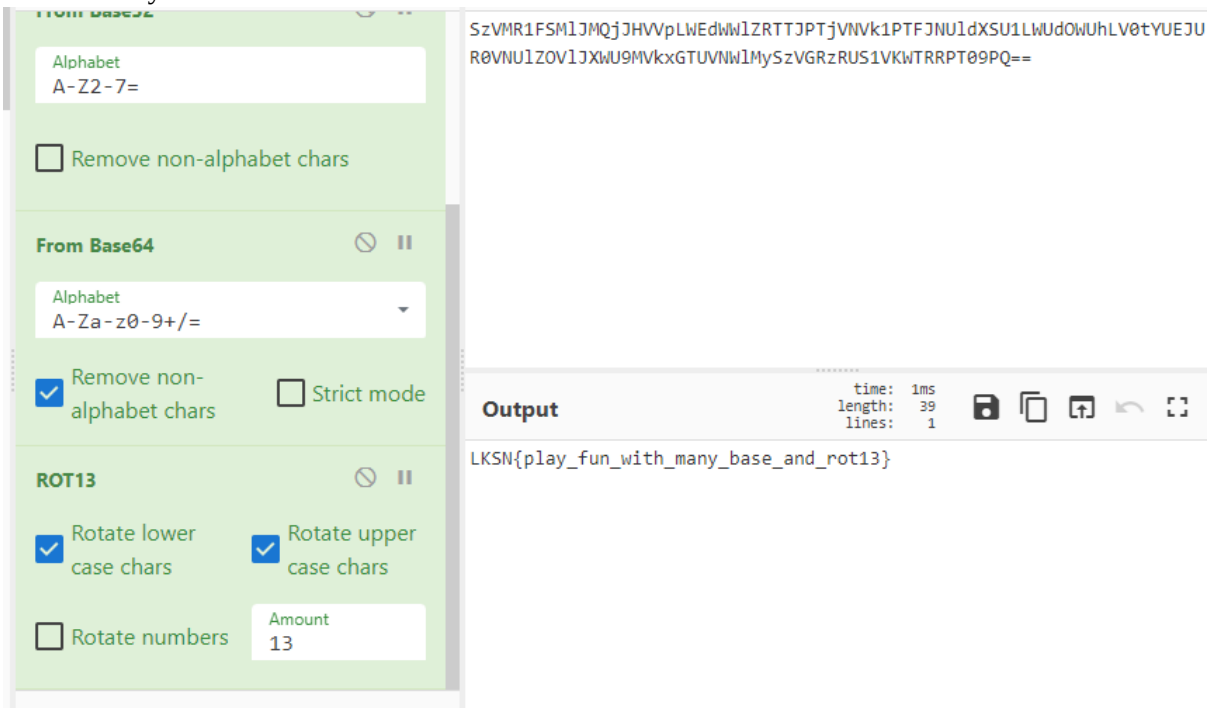


Ternyata hasilnya masih berupa kata-kata terenkripsi, maka kita lakukan decrypt lagi dengan cara mengeklik tombol seperti tongkat sihir yang berada di samping kata "Output" kita klik sampai tidak ada tombol tongkat sihir lagi.



Nah, sampai di sini, kami mengalami kesusahan karena ketika kami menggunakan tool *Cipher Identifier* tidak menemukan jenis enkripsi yang cocok sama sekali, kemudian kami menyadari bahwa kata akhir yang berupa angka 13 tersebut bisa jadi jenis enkripsi

ROT13, maka tanpa pikir panjang lagi, kami mendekripsi nya menggunakan ROT13 dan inilah hasilnya.



Flag: LKSN{play_fun_with_many_base_and_rot13}

D. Reverse Engineering

1. Bahasa Ular

   Pada challenge ini, kami diberikan challenge dengan deskripsi yang disertai file berupa file bahasaular.pyc dan flag.txt.malware yang terenkripsi.

Kami pikir, file .pyc itu sama saja dengan format file python biasa yaitu .py, akan tetapi saat kami berusaha untuk menjalankannya, terjadi error. Akhirnya kami mencari cara bagaimana mengeksekusi file .pyc. Dan ternyata untuk menjalankan

file .pyc, kita harus mendecompile atau mengubah file .pyc menjadi file .py. Kita menggunakan tool *pycdc* yang didapat dari internet untuk melakukannya.

```
┌──(kali㉿kali)-[~/Downloads/LKS-Nasional/pycdc]
└─$ ./pycdc /home/kali/Downloads/LKS-Nasional/reveng/bahasaular.pyc
# Source Generated with Decompyle++
# File: bahasaular.pyc (Python 3.8)

Unsupported opcode: BEGIN_FINALLY
import this
input_peserta = input('Yuk belajar bahasa ular yang sudah di-"goreng"!\nMasukkan nama file kamu dan kami mengubahnya menjadi format yang k
eren! >>')
f_handler = open(input_peserta, 'rb').read()
f_transform = []
for karakter in f_handler:
    f_transform.append(karakter ^ 2 ^ 3 ^ 7 ^ 9 ^ 11 ^ 13)
f_final = ''
for karakterlagi in range(len(f_transform)):
    f_final += chr(f_transform[karakterlagi] ^ ord(this.s[karakterlagi % len(f_transform)]))
# WARNING: Decompyle incomplete
```

Command bisa berbeda-beda tergantung dimana kita menaruh file bahasaular.pyc kita.

Setelah kita decompile menggunakan pycdc, kita copy hasil decompile nya dan kita masukkan ke text editor.

```
1  # Source Generated with Decompyle++
2  # File: bahasaular.pyc (Python 3.8)
3
4  Unsupported opcode: BEGIN_FINALLY
5  import this
6  input_peserta = input('Yuk belajar bahasa ular yang sudah di-"goreng"!\nMasukkan nama file kamu dan kami mengubahnya menjadi format yang keren! >>')
7  f_handler = open(input_peserta, 'rb').read()
8  f_transform = []
9  for karakter in f_handler:
10     f_transform.append(karakter ^ 2 ^ 3 ^ 7 ^ 9 ^ 11 ^ 13)
11 f_final = ''
12 for karakterlagi in range(len(f_transform)):
13     f_final += chr(f_transform[karakterlagi] ^ ord(this.s[karakterlagi % len(f_transform)]))
14 # WARNING: Decompyle incomplete
```

Setelah kami baca code di atas, kami menyadari bahwa variable f_final tidak pernah di print, maka dari itu kami tambahkan print(f_final) seperti pada gambar di bawah ini.

```
1  # Source Generated with Decompyle++
2  # File: bahasaular.pyc (Python 3.8)
3
4  import this
5  input_peserta = input('Yuk belajar bahasa ular yang sudah di-"goreng"!\nMasukkan nama file kamu dan kami mengubahnya menjadi format yang keren! >>')
6  f_handler = open(input_peserta, 'rb').read()
7  f_transform = []
8  for karakter in f_handler:
9      f_transform.append(karakter ^ 2 ^ 3 ^ 7 ^ 9 ^ 11 ^ 13)
10 f_final = ''
11 for karakterlagi in range(len(f_transform)):
12     f_final += chr(f_transform[karakterlagi] ^ ord(this.s[karakterlagi % len(f_transform)]))
13     print(f_final)
14 # WARNING: Decompyle incomplete
```

Setelah itu kami coba jalankan script nya dan kami masukkan flag.txt.malware dan inilah hasilnya.

Flag: LKSN(w4h_kamU_b!sa_84hasa_uLar_yang_diG0reng_a.k.a_PYC}

Sekian writeup CTF dari kami, kurang lebihnya mohon maaf. Sekian dan terimakasih.