**Writeup GKSK #4**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**bemrdo**

*Note: I’m using Hackintosh Mojave, so there are some different codes with Linux’s terminal*

**LIST OF LYFE**

**JOY 1**

Hack The Game v0.0.1 1

Hack The Game v0.0.2 2

**JAIL 7**

Counting 7

**REVERSE ENGINEERING 9**

Decimal 1.0 9

Serial Code 10

Decimal 2.0 12

**CRYPTOGRAPHY 14**

GKSK Crypto Service? 14

**JOY**

Hack The Game v0.0.1

50 pts

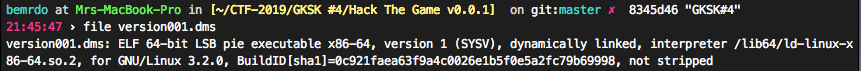
I just tried to make a simple game, please help me to test it!

Author: Leonardo

**Attachment :** [version001](https://gksk.stikom-bali.ac.id/files/2b3c904b6933b33f6eefab5922c445c1/version001)

**POC :**

Diberikan sebuah file binary ELF 64-bit



Ketika dijalankan pada terminal, program ini hanya mengeluarkan output dan kemudian exit



Dari output tersebut, saya meramal bahwa dilakukan compare string untuk mendapatkan flag, tapi ternyata ramalan saya gagal… File ini saya disassembly menggunakan IDAPro 64, terlihat bahwa file ini menggunakan bahasa C++ dengan kode yang sangat panjang ketika diubah menjadi pseudocode. Setelah menganalisis rangkaian kode yang seperti koran, saya menemukan sebuah function checkBetaTest(v3)



Memang terdapat compare string di dalam function tersebut antara v7 dengan v9, namun flag bisa saya dapatkan dengan menggabungkan beberapa string berikut



v6 = "4re\_Y0u\_53ri0usly\_checking\_f0r\_b3t4\_t3sT?"



v8 = "GKSK{" + v6

v9 = v8 + "}"

Sehingga didapatkan flag **GKSK{4re\_Y0u\_53ri0usly\_checking\_f0r\_b3t4\_t3sT?}**

## Hack The Game v0.0.2

### 150 pts

nc 180.250.7.183 20202

The game is progessing quite nicely, now you can fight some monster.

And maybe if you level up enough you could defeat the Final Boss “The Flag Guardian”

Author: Leonardo

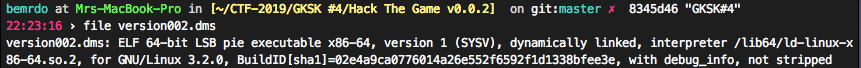
**Hint 1 :** Are you sure it's just regular base64?

**Hint 2 :** Do you realise you can save your progress when resting? Maybe try to make your own story from there...

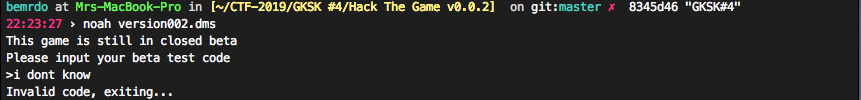
**Attachment :** [version002](https://gksk.stikom-bali.ac.id/files/e11044c934c42d61871e4c7c45508659/version002)

**POC :**

Diberikan sebuah file binary ELF 64-bit, dari Namanya terlihat bahwa ini lanjutan dari Hack The Game v0.0.1

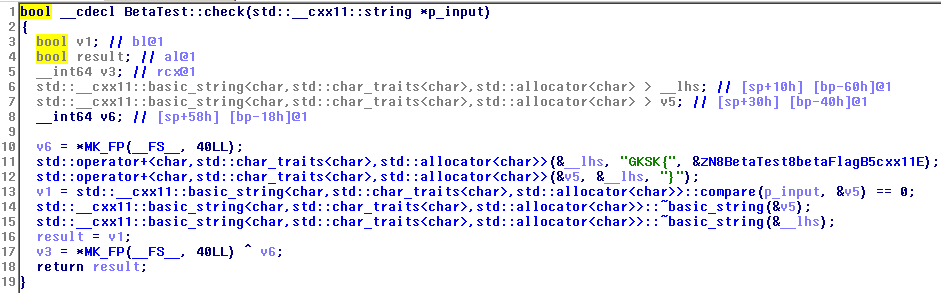


Ketika dijalankan, program meminta input berupa beta test code..

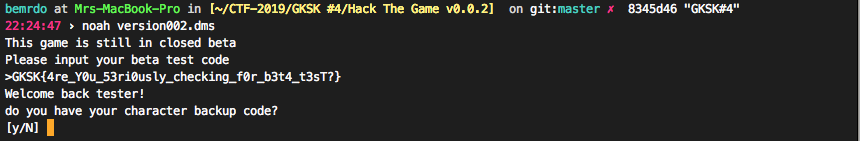


Oke next disassemble aja yaa…

Setelah ditelaah lebih dalam, setelah memasukkan beta test code, terdapat function check(&p\_input) yg mengecheck input beta test code dengan code yang benar



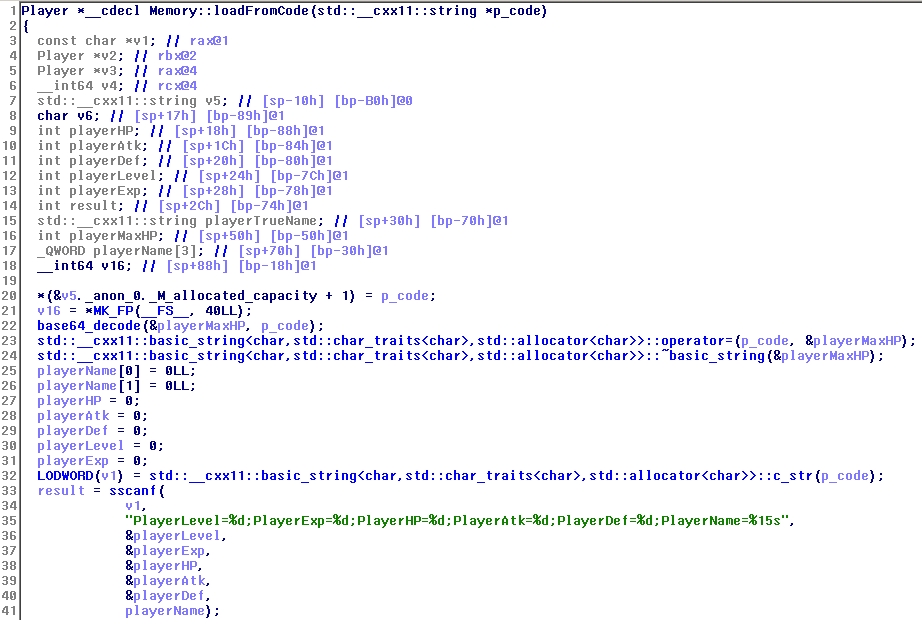
Pada program terlihat bahwa input beta test code di compare dengan "GKSK{" + "GKSK{", &ZN8BetaTest8betaFlagB5cxx11E + "}". Dari nama variable itu saya menyimpulkan bahwa input beta test code merupakan flag dari challenge Hack The Game v0.0.1 yaitu GKSK{4re\_Y0u\_53ri0usly\_checking\_f0r\_b3t4\_t3sT?}



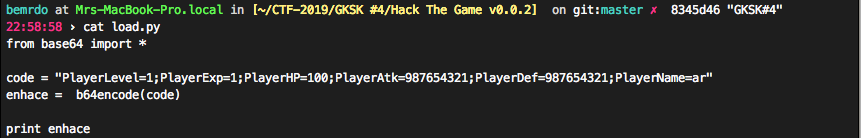
Setelah memasukan beta code test yang benar, terdapat pilihan untuk character backup… Jika dibaca dari program terdapat function loadFromCode(&p\_input) apabila saya memasukan ‘y’ atau ‘Y’ pada pilihan tersebut.

Dalam function tersebut, terdapat proses base64 decode dari input yang masuk ke function. Kemudian hasil base64 decode di parsing dengan pattern

"PlayerLevel=%d;PlayerExp=%d;PlayerHP=%d;PlayerAtk=%d;PlayerDef=%d;PlayerName=%15s"

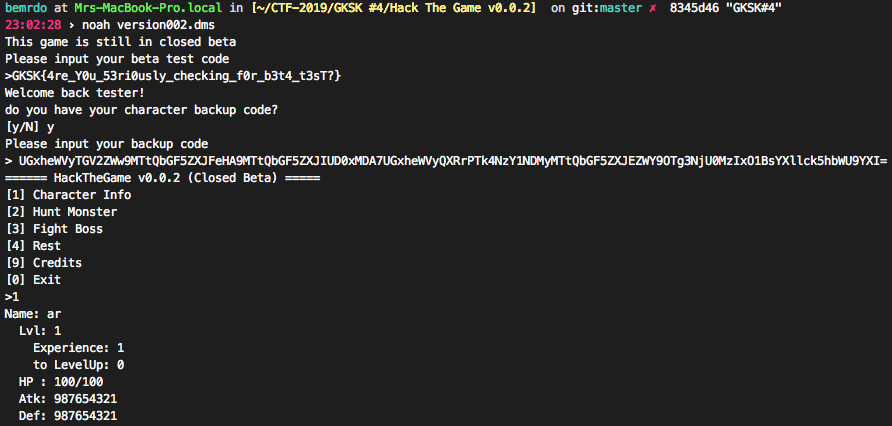


Jadi saya buatkan pattern tersebut sesuai keinginan saya dan kemudian di encode dengan base64

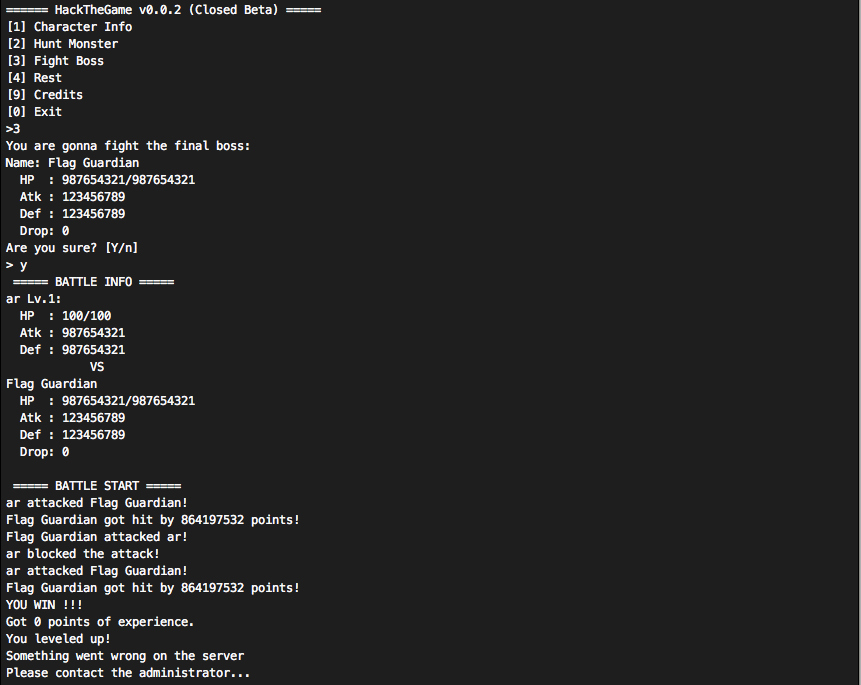




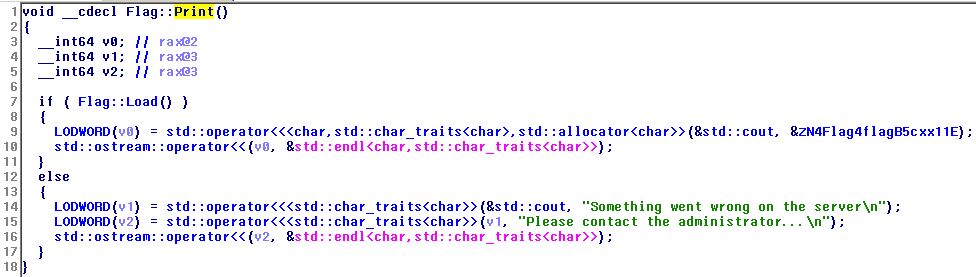
Kemudian saya coba masukan pada program



Ok, sesuai keinginan… langsung saja saya lawan Boss pada game ini



Aahhhsiiyyaappp… pesan di akhir muncul karena program tidak menemukan file flag.txt, seperti yang dijelaskan pada function Print()



Langsung aja dicoba ke server yang diberikan… Tapi sebuah bencana terjadi :(

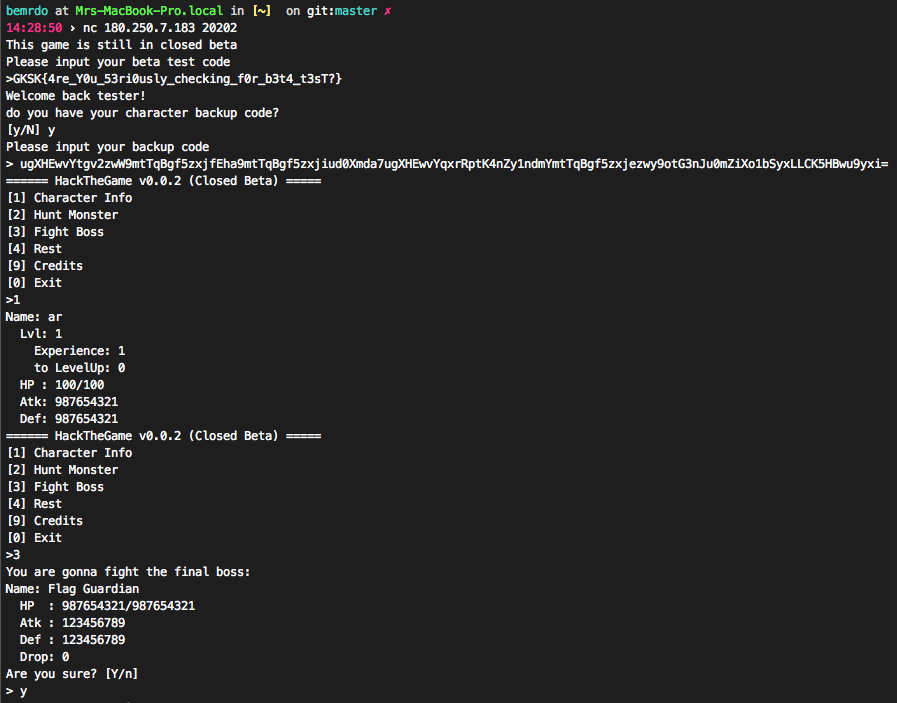
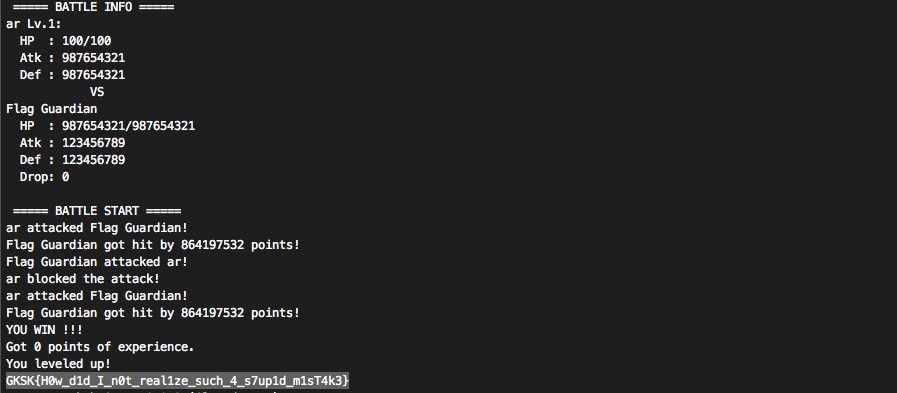
Backup code yang dimasukkan dikatakan invalid, saya jadi sedih… kemudian saya memutuskan untuk bermain jujur dengan membuat akun baru pada game tersebut. Sesuatu muncul ketika saya memilih menu [4] Rest pada game (setelah lelah bermain jujur), game memunculkan semacam kode yang sudah di encode dengan base64…



Terlihat mirip seperti backup code yang sebelumnya saya buat, hanya saja terdapat penukaran huruf kapital dengan huruf kecil dan juga sebaliknya. Saya kemudian membuat backup code baru dengan penukaran huruf capital dan huruf kecil tersebut.

Kembali bermain curang… Ternyata backup code yang baru bias digunakan pada server

Setelah mengalahkan Flag Guardian, dapatlah flagnya **GKSK{H0w\_d1d\_I\_n0t\_real1ze\_such\_4\_s7up1d\_m1sT4k3}**

**JAIL**

## Counting

### 75 pts

Counting!! Counting!! 1,2,3 to 3,2,1 yeeaaayy!1!1

**nc 180.250.7.183 13337**

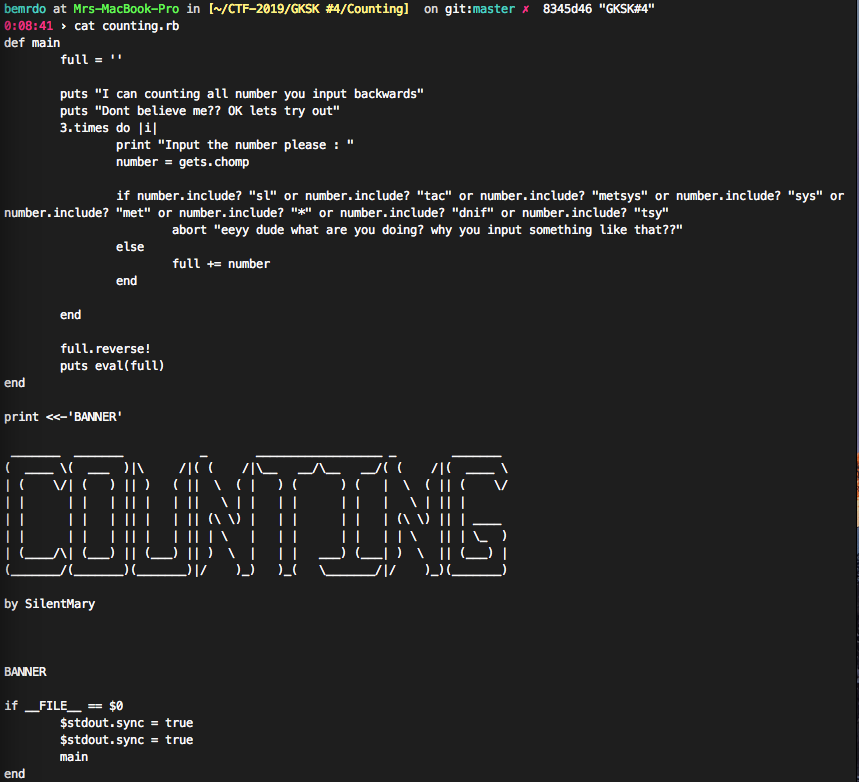
Author: **SilentMary**

**Hint :** no ls and cat allowed on my Ruby stone

**Attachment :** [counting.rb](https://gksk.stikom-bali.ac.id/files/773366c4e5f10620170a399820489166/counting.rb)

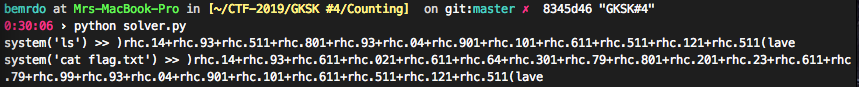
**POC :**

Diberikan source code ruby seperti berikut



Target saya adalah syntax puts eval(full), function eval() dapat digunakan untuk memproses argument apapun menjadi syntax aktif (case in ruby). Sehingga saya dapat menggunakan system(‘ls’) dan system(‘cat <filename>’) untuk mendapatkan flag. Namun kendalanya terdapat filter pada program sehingga input yang saya inginkan tidak bisa dijalankan.

Cara yang saya gunakan adalah dengan memecah setiap karakter dari input menjadi bentuk desimalnya yang kemudian diubah kembali dalam function eval() menjadi string. Tapi untuk membuat string tersebut aktif maka diperlukan eval() lagi yang berjalan dalam fungsi eval(). Saya membuat input dengan program python seperti berikut

Pertama masukkan input untuk system(‘ls’) yaitu

)rhc.14+rhc.93+rhc.511+rhc.801+rhc.93+rhc.04+rhc.901+rhc.101+rhc.611+rhc.511+rhc.121+rhc.511(lave

Kemudian tekan tombol Enter 2 kali (input kosong)

Ternyata terdapat file flag.txt yang sepertinya menyimpan flag.. Selanjutnya jalankan lagi counting.rb dan masukkan input untuk system(‘cat flag.txt’) yaitu

)rhc.14+rhc.93+rhc.611+rhc.021+rhc.611+rhc.64+rhc.301+rhc.79+rhc.801+rhc.201+rhc.23+rhc.611+rhc.79+rhc.99+rhc.93+rhc.04+rhc.901+rhc.101+rhc.611+rhc.511+rhc.121+rhc.511(lave

Kemudian tekan tombol Enter 2 kali



Ternyata memang ada flagnya **GKSK{R\_4\_r3VersE\_E\_4\_3v4L}**

**REVERSE ENGINEERING**

## Decimal 1.0

### 50 pts

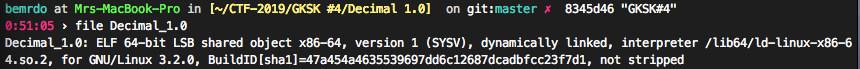
Si Jono mendapatkan sebuah pesan singkat dari orang yang tak di kenal, tetapi pesan tersebut membutuhkan **Key Yang Benar** untuk dapat membacanya, bisakan kalian membantunya??

Author: **Wayan**

**Attachment :** [Decimal\_1.0](https://gksk.stikom-bali.ac.id/files/d48bedfea511ea8d68bfc474889f30a6/Decimal_1.0)

**POC :**

Diberikan file ELF 64-bit



Oke langsung aja pake IDAPro… Setelah menyusuri kode demi kode, akhirnya sauya menemukan ide dari program ini. Program ini akan meminta key sepanjang 6 karakter yang akan di check dengan key yang telah dimodifikasi. Jika semua karakter key benar maka array geser[] diisi dengan nilai 5 pada index 0 hingga 5, nilai index inilah yang digunakan untuk menambah nilai ascii dari sebuah string "BFNFv@/nTZ@0.FZ.n@fZ?.>,h/Gx" sehingga string tersebut akan menjadi flag



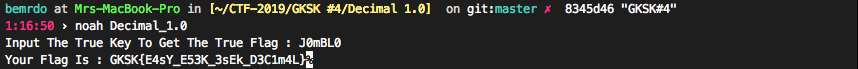
Kemudian saya membuat program python yang langsung merubah string tersebut menjadi flag

Cara lainnya bisa dengan mencari key yang sesuai, pada program ini key yang digunakan adalah string dari hasil penambahan nilai ascii dari string "@&c8B&" senilai 10, sehingga menjadi seperti berikut



Kemudian tinggal jalankan program dan masukkan key nya



Flagnya esek esek banget **GKSK{E4sY\_E53K\_3sEk\_D3C1m4L}**

## Serial Code

### 50 pts

Si Cantique diberikan misi untuk memecahkan sebuah serial code sehingga ia bisa mendapatkan flagnya. Bisakah kamu membantu memecahkannya?

Akses ke:

* **nc 180.250.7.183 13369**
* **nc 103.200.7.150 13368**

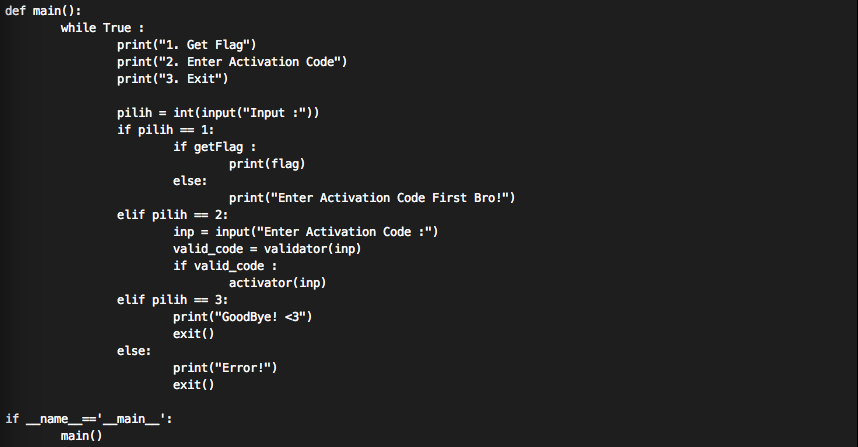
Author: **bad\_girl, I love you.**

**Attachment :** [reverse.py](https://gksk.stikom-bali.ac.id/files/f87569eb691bdfbcf604bac2f1c7f190/reverse.py)

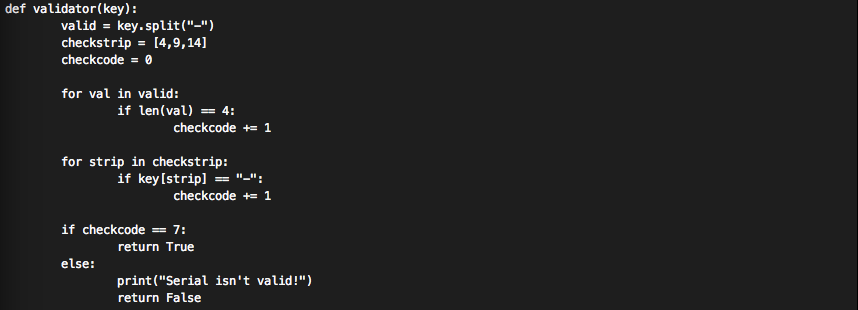
**POC :**

Diberikan python code yang sangat panjang… mari bahas pelan-pelan :v

Ketika menjalankan program ini, maka akan langsung masuk ke fungsi main(). Dalam fungsi ini terdapat menu Get Flag yang menjadi incaran saya, namun sebelum itu saya harus mengaktivasi code nya pada menu Enter Activation Code. Jika code sudah diaktivasi maka bisa menggunakan menu Get Flag



Pada menu Enter Activation Code, terdapat 2 segment yaitu validator(key) dan activator(key). validator(key) hanya mengecheck format key yang diinputkan, jika sudah benar maka masuk ke fungsi activator(key)



Pada fungsi activator(key), setiap karakter key di check agar sesuai kondisi yang terdapat pada program. Sebenarnya bias menggunakan z3-solver untuk mencari key, berhubung key tidak terlalu Panjang dan masih lebih cepat berpikir daripada berpikir + ngetik. Saya memutuskan hitung manual, hehehe…

Key dibagi menjadi 4 bagian (dipisahkan tanda ‘-‘) yang mana setiap bagian terdiri dari 4 karakter

check\_1. Nilai setiap karakter jika di mod 2 menghasilkan 0. Langsung saja semua karakter bernilai 0. Diperlukan 4 kali benar dalam kondisi ini

0000



check\_2. Nilai dari bagian ini (4 karakter) di convert menjadi sebuah integer dengan kondisi berikut.

Jika di looping dari nilai 2 hingga nilai integer-1, maka integer di mod dengan nilai looping tersebut harus tidak sama dengan 0. Diperlukan 1 kali benar dalam kondisi ini

0003



check\_3. Nilai dari bagian ini di convert menjadi sebuah integer dengna kondisi jika integer di mod 4 harus bernilai 0. Diperlukan 1 kali benar dalam kondisi ini

0000 atau 0004



check\_4. Setiap karakter harus bernilai ascii dari 71, 75, 83, 75. Diperlukan 1 kali benar dalam kondisi ini

GKSK



Jika semua kondisi benar, maka serial key di aktivasi



Key : 0000-0003-0004-GKSK

Tinggal masukkan pada server yang disediakan dan pilih menu Get Flag



Nice code flagnya **GKSK{b4D\_c0D3\_M0rE\_eA5i3r}**

Decimal 2.0

75 pts

Si Jono kembali mendapatkan sebuah pesan singkat dari orang yang tak di kenal, tetapi pesan tersebut membutuhkan **Key Yang Benar** untuk dapat membacanya, bisakan kalian membantunya Lagi??

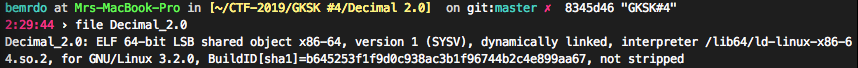
Author: **Wayan**

**Hint :** sepertinya beberapa Decimal sudah di geser

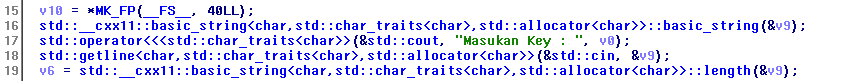
**Attachment :** [Decimal\_2.0](https://gksk.stikom-bali.ac.id/files/520e683e5d87c2fa688120703c7e6eea/Decimal_2.0)

**POC :**

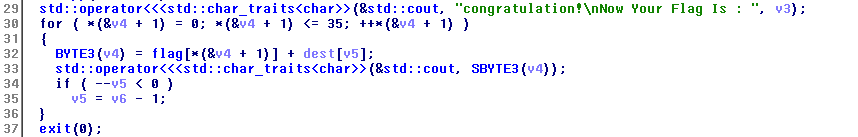
Diberikan file ELF 64-bit



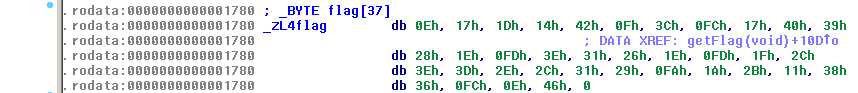
Tanpa menghayal yang aneh aneh, saya langsung disassemble dengan IDAPro 64. Dan perhatian saya langsung tertuju pada function getFlag(). Dalam function tersebut terdapat input key yang digunakan untuk anu anu flag nya :v.



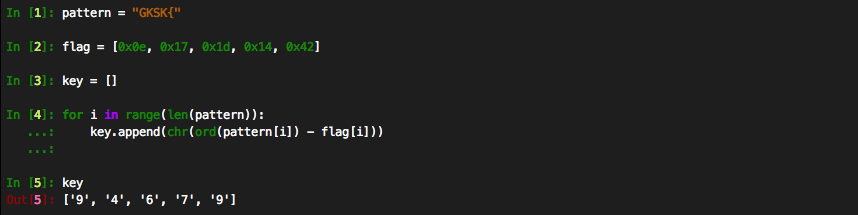
Oke saya skip dulu karena memang gk tau key yg benar itu apa. Pada proses print flag terlihat bahwa terjadi penambahan nilai ascii pada setiap karakter dalam array flag sebelum akhirnya ditampilkan.



Kita check dulu isi array flag nya



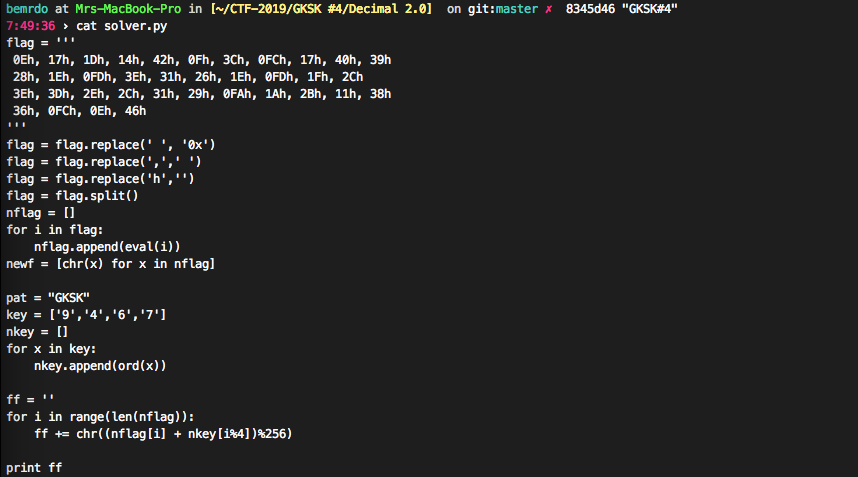
Kemudian data tersebut saya coba anu anu juga menggunakan python, setelah mencoba berbagai cara untuk mendapatkan si dia. Akhirnya saya menemukan bahwa ini adalah repetition key, terlihat dalam percobaan saya berikut



Terlihat bahwa nilai pertama dan kelima adalah 9, jika benar repetition key maka seharusnya pada karakter terakhir dalam array flag menggunakan key ‘7’ (length flag = 36)



Berikut solver yang saya buat menggunakan python



Okayyy… Thank you, next…



Ini flag anu anu nya **GKSK{Cr3Pto\_W1th\_R3Verseee\_1S\_Goo0D}**

**CRYPTOGRAPHY**

## GKSK Crypto Service?

### 75 pts

Someone use **GKSK CRYPTO SERVICE** to encrypt malicious message Please decrypt it and find the secret message!

md5sum:  
27bf40195ff473a85e7efff1ac14be28  [**GKSK\_CRYPTO\_SERVICE.zip**](https://drive.google.com/open?id=1Ka8nqU6k5-neGpiy27A1LIL5Psjy7zRk)

Author: **ιнρи¢**

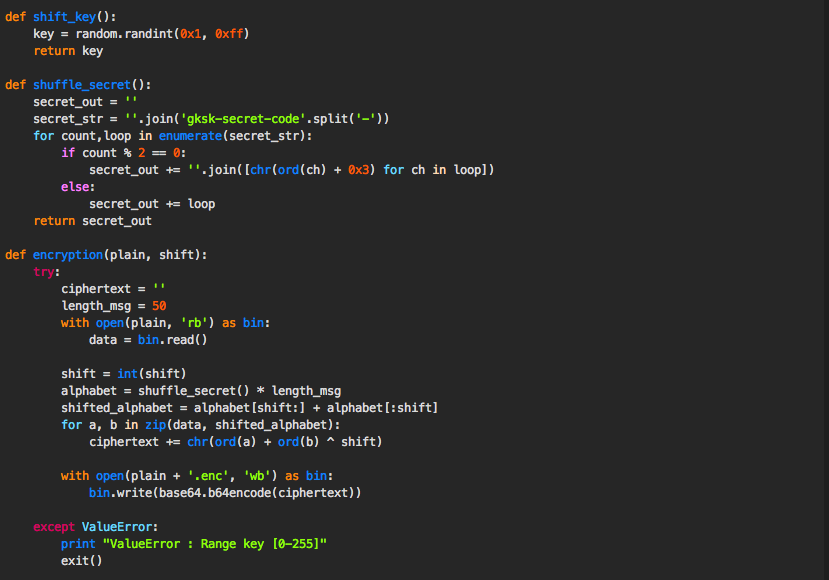
**Attachment :** [GKSK\_CRYPTO\_SERVICE.zip](https://drive.google.com/open?id=1Ka8nqU6k5-neGpiy27A1LIL5Psjy7zRk)

**POC :**

Diberikan program dalam python yang jika dibuka dapat menyebabkan pusing, sakit kepala, bahkan depresi berkepanjangan :v

Okay, ide dari program ini adalah digunakan untuk enkripsi dan dekripsi pesan (flag) namun fungsi dekripisi belum terdapat pada program (embel-embelnya harus upgrade ke premium)

Jadi saya hanya perlu membuat fungsi dekripsi dengan membalik proses enkripsi berikut



Enkripsi tersebut menggunakan secret code yang bisa kita ambil langsung dari fungsi suffle\_secret()

Kemudian flag d proses setiap karakternya seperti berikut



Hasil ciphertext kemudian di encode ke base 64 lalu di write pada file



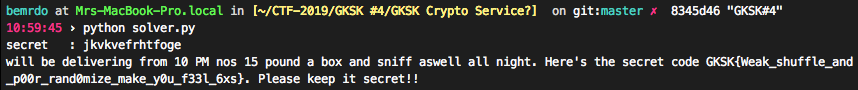
Yang tidak bisa kita tentukan adalah nilai variable key karena mungkin pembuat pesan mengambil nilai random pada fungsi shift\_key().



Program solver yang saya buat menggunakan brute force nilai shift dari angka 1 hingga saya tidak tau berapa :v … program saya dalam python seperti berikut



Hasilnya seperti ini



Ingat aja Try Harder eeaaa **GKSK{Weak\_shuffle\_and\_p00r\_rand0mize\_make\_y0u\_f33l\_6xs}**