

# ROP Lab

## Modern Binary Exploitation CSCI 4968 - Spring 2015 Markus Gaasedelen

```
push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], ebx
jnz short loc_313066
mov eax, [ebp+var_70]
cmp eax, [ebp+var_84]
jb short loc_313066
sub eax, [ebp+var_84]
push esi
push esi
push eax
push edi
mov [ebp+arg_0], eax
call sub_31486A
test eax, eax
jz short loc_31306D
push esi
lea eax, [ebp+arg_0]
push eax
mov esi, 1D0h
push esi
push [ebp+arg_4]
push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], esi
jz short loc_31308F

loc_313066: ; CODE XREF: sub_312FD8
; sub_312FD8+56
push 0Dh
call sub_31411B

loc_31306D: ; CODE XREF: sub_312FD8
; sub_312FD8+49
call sub_3140F3
test eax, eax
jg short loc_31307D
call sub_3140F3
jmp short loc_31308C

loc_31307D: ; CODE XREF: sub_312FD8
call sub_3140F3
and eax, 0FFFFFFh
or eax, 80070000h

loc_31308C: ; CODE XREF: sub_312FD8
mov [ebp+var_4], eax
```

# Final notes on DEP

- One last concept
  - ret2libc

```
push    edi
call   sub_314623
test   eax, eax
jz     short loc_31306D
cmp    [ebp+arg_0], ebx
jnz   short loc_313066
mov    eax, [ebp+var_70]
cmp    eax, [ebp+var_84]
jb     short loc_313066
sub    eax, [ebp+var_84]
push   esi
push   esi
push   eax
push   edi
mov    [ebp+arg_0], eax
call   sub_31486A
test   eax, eax
jz     short loc_31306D
push   esi
lea   eax, [ebp+arg_0]
push   eax
mov    esi, 1D0h
push   esi
push   [ebp+arg_4]
push   edi
call   sub_314623
test   eax, eax
jz     short loc_31306D
cmp    [ebp+arg_0], esi
jz     short loc_31308F

loc_313066:                                     ; CODE XREF: sub_312FD8
                                                ; sub_312FD8+56
push   0Dh
call   sub_31411B

loc_31306D:                                     ; CODE XREF: sub_312FD8
                                                ; sub_312FD8+49
call   sub_3140F3
test   eax, eax
jg     short loc_31307D
call   sub_3140F3
jmp    short loc_31308C
; -----

loc_31307D:                                     ; CODE XREF: sub_312FD8
call   sub_3140F3
and    eax, 0FFFFFFh
or     eax, 80070000h

loc_31308C:                                     ; CODE XREF: sub_312FD8
mov    [ebp+var_4], eax
```

# ret2libc

- 'ret2libc' is a technique of **ROP** where you return to functions in standard libraries (libc), rather than using **gadgets**
- If you know the addresses of the functions you want to **ROP** through in libc (assuming libc exists), ret2libc is easier than making a **ROP chain** with **gadgets**

```
push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], ebx
jnz short loc_313066
mov eax, [ebp+var_70]
cmp eax, [ebp+var_84]
jb short loc_313066
sub eax, [ebp+var_84]
push esi
```

```
push esi
push eax
push edi
mov [ebp+var_70], esi
lea esi, [ebp+var_84]
test eax, eax
jz short loc_31306D
lea eax, [ebp+arg_0]
push eax
mov esi, 1D0h
push esi
push [ebp+arg_4]
push edi
call sub_314623
test eax, eax
jz short loc_31306D
mov [ebp+arg_0], esi
jnz short loc_313066
```

```
loc_313066:                                     ; CODE XREF: sub_312FD8
; sub_312FD8+56
```

```
push 1D0h
call sub_31411B
```

```
loc_31306F:                                     ; CODE XREF: sub_312FD8
; sub_312FD8+49
```

```
call sub_3140F3
test eax, eax
jg short loc_31307D
call sub_3140F3
jmp short loc_31308C
```

```
;
```

```
loc_31307D:                                     ; CODE XREF: sub_312FD8
```

```
call sub_3140F3
and eax, 0FFFFFFh
or eax, 80070000h
```

```
loc_31308C:                                     ; CODE XREF: sub_312FD8
```

```
mov [ebp+var_4], eax
```

# Common ret2libc Targets

- **system()**
  - Executes something on the command line
  - **system("cat flag.txt");**
- **(f) open() / read() / write()**
  - Open/Read/Write a file contents

```
push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], ebx
jnz short loc_313066
mov eax, [ebp+var_70]
cmp eax, [ebp+var_84]
jb short loc_313066
sub eax, [ebp+var_84]
push esi
push esi
push eax
push edi
mov [ebp+arg_0], eax
call sub_31486A
test eax, eax
jz short loc_31306D
lea eax, [ebp+arg_0]
push eax
mov esi, 1D0h
push esi
push [ebp+arg_4]
push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], esi
jz short loc_31308F
```

```
loc_313066: ; CODE XREF: sub_312FD8
; sub_312FD8+56
push 0Dh
call sub_31411B
```

```
loc_31306D: ; CODE XREF: sub_312FD8
; sub_312FD8+49
```

```
call sub_3140F3
test eax, eax
jg short loc_31307D
call sub_3140F3
jmp short loc_31308C
```

```
loc_31307D: ; CODE XREF: sub_312FD8
```

```
call sub_3140F3
and eax, 0FFFFFFh
or eax, 80070000h
```

```
loc_31308C: ; CODE XREF: sub_312FD8
```

# ret2libc example

```

push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], ebx
jnz short loc_313066
mov eax, [ebp+var_70]
cmp eax, [ebp+var_84]
jb short loc_313066
sub eax, [ebp+var_84]
push esi
push esi

```

system() --->

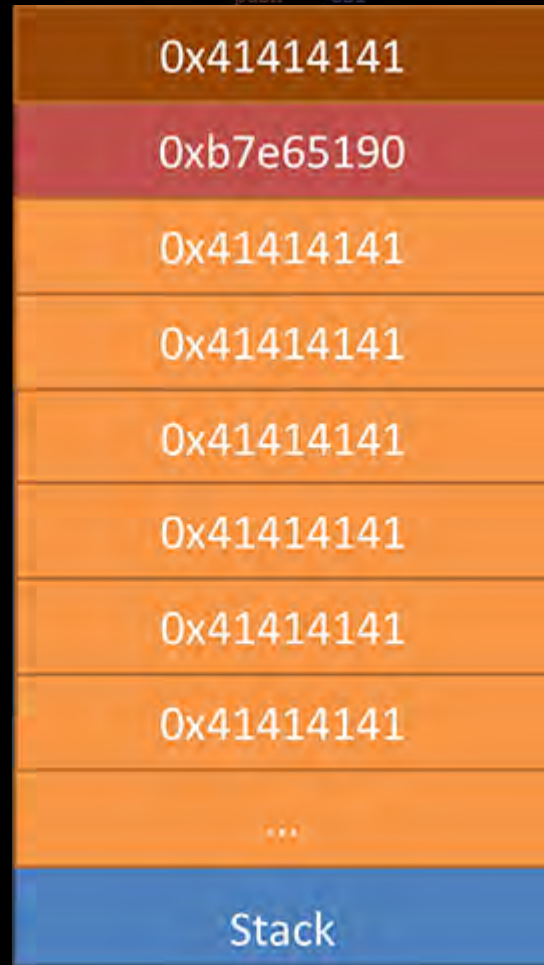
0x08045430: ret ← EIP

.....  
system()

```

0xb7e65190: push ebx
0xb7e65191: sub esp, 8
0xb7e65194: mov eax, DWORD PTR
[esp+0x10]
...

```



CODE XREF: sub\_312FD8  
sub\_312FD8+6

CODE XREF: sub\_312FD8  
sub\_312FD8+49

CODE XREF: sub\_312FD8

```

and eax, 0FFFFFFh
or eax, 80070000h

```

```

loc_31308C: mov [ebp+var_4], eax
; CODE XREF: sub_312FD8

```

# Returning to System

- We want to call `system("cat flag.txt");`
- Because we are **ROPing** into `system` rather than calling it, you have to think about setting up the stack (to pass arguments) a little bit differently

```
push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], ebx
jnz short loc_313066
mov eax, [ebp+var_70]
cmp eax, [ebp+var_84]
jb short loc_313066
sub eax, [ebp+var_84]
push esi
push esi
push eax
push edi
call [ebp+arg_0], eax
test eax, eax
jz short loc_31306D
push esi
lea eax, [ebp+arg_0]
push eax
mov [ebp+var_1D0], eax
push [ebp+arg_4]
push edi
sub [ebp+var_62], 6
test eax, eax
jz short loc_31306D
cmp [ebp+var_70], ebx
jz short loc_313066
; -----
loc_313066: ; CODE XREF: sub_312FD8
; sub_312FD8+56
push 0Dh
call sub_31411B
loc_31306D: ; CODE XREF: sub_312FD8
; sub_312FD8+49
call sub_3140F3
test eax, eax
jg short loc_31307D
call sub_3140F3
jmp short loc_31308C
; -----
loc_31307D: ; CODE XREF: sub_312FD8
call sub_3140F3
and eax, 0FFFFFFh
or eax, 80070000h
loc_31308C: ; CODE XREF: sub_312FD8
mov [ebp+var_4], eax
```

# ret2libc example

```

push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], ebx
jnz short loc_313066
mov eax, [ebp+var_70]
cmp eax, [ebp+var_84]
jb short loc_313066
sub eax, [ebp+var_84]
push esi
push esi

```

system() --->

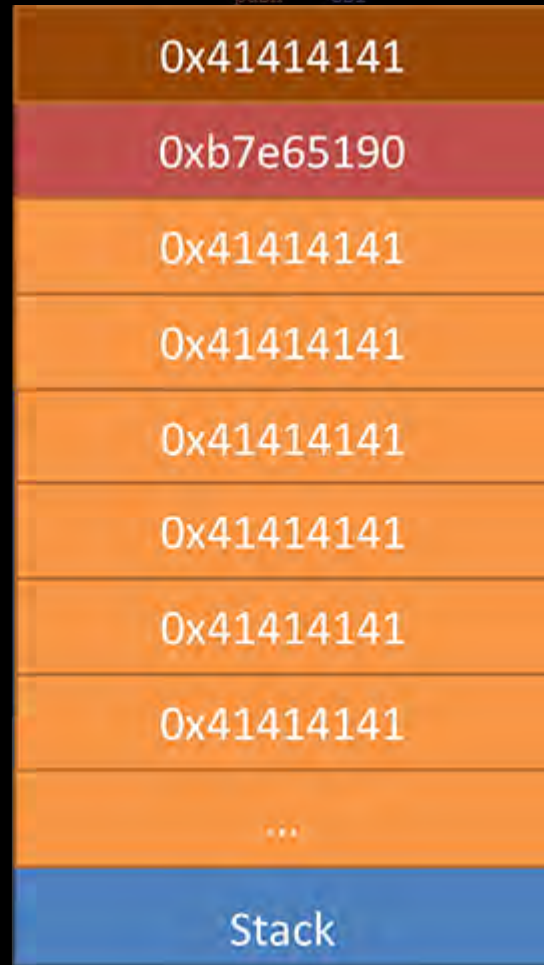
0x08045430: ret ← EIP

.....  
system()

```

0xb7e65190: push ebx
0xb7e65191: sub esp, 8
0xb7e65194: mov eax, DWORD PTR
[esp+0x10]
...

```



CODE XREF: sub\_312FD8  
sub\_312FD8+6  
CODE XREF: sub\_312FD8  
sub\_312FD8+49  
CODE XREF: sub\_312FD8

```

and eax, 0FFFFFFh
or eax, 80070000h
loc_31308C:
mov [ebp+var_4], eax

```

# ret2libc example

```

push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], ebx
jnz short loc_313066
mov eax, [ebp+var_70]
cmp eax, [ebp+var_84]
jb short loc_313066
sub eax, [ebp+var_84]
push esi
push esi

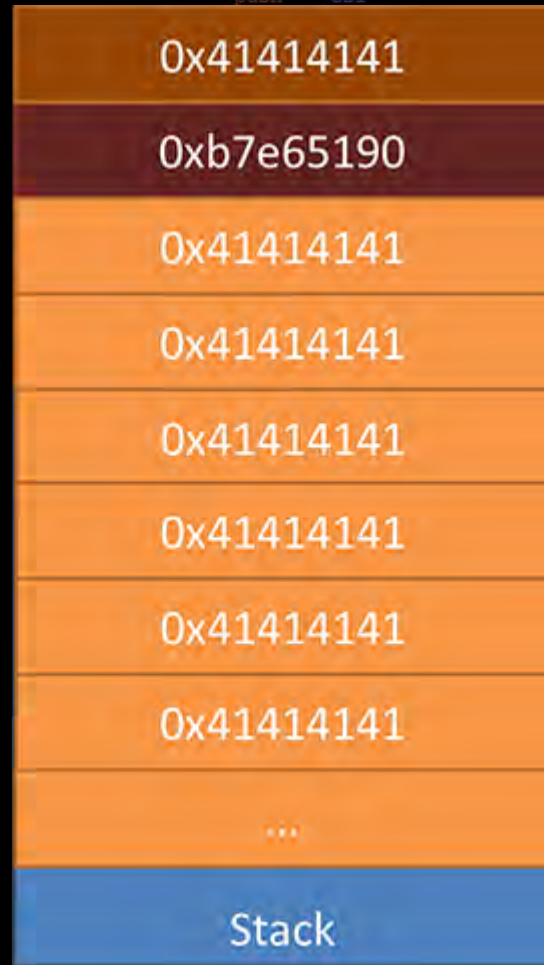
```

**system()** --->  
 ?????? --->  
 ?????? --->

0x08045430: ret

.....  
 system()

0xb7e65190: push ebx  
 0xb7e65191: sub esp, 8  
 0xb7e65194: mov eax, DWORD PTR  
 [esp+0x10]  
 ...



← ESP  
 Stack Growth  
 CODE XREF: sub\_312FD8  
 sub\_312FD8+6  
 CODE XREF: sub\_312FD8  
 sub\_312FD8+49  
 CODE XREF: sub\_312FD8

```

and eax, 0FFFFFFh
or eax, 80070000h
loc_31308C:
mov [ebp+var_4], eax
; CODE XREF: sub_312FD8

```



# ret2libc example

```

push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], ebx
jnz short loc_313066
mov eax, [ebp+var_70]
cmp eax, [ebp+var_84]
jb short loc_313066
sub eax, [ebp+var_84]
push esi
push esi

```

system() --->

ret address --->

first arg --->

0x08045430: ret

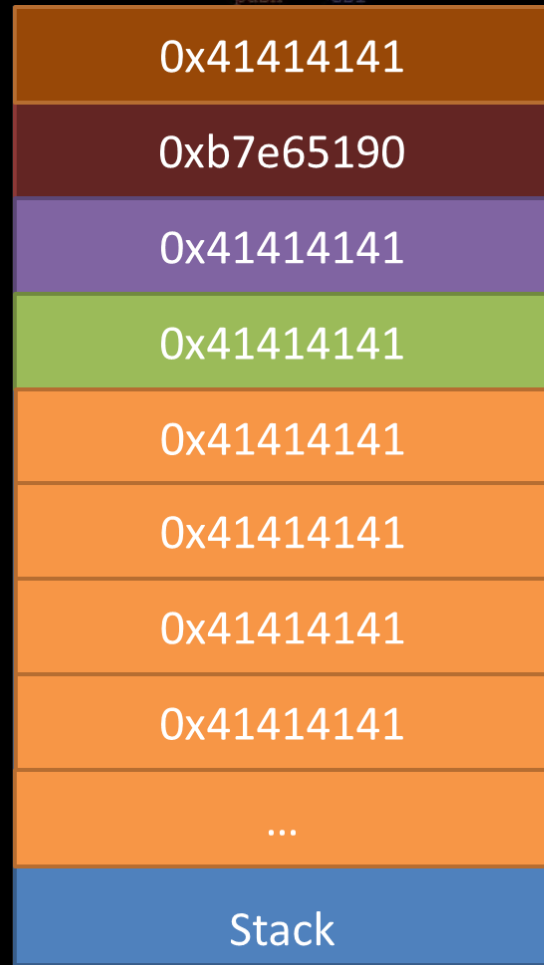
.....  
system()

0xb7e65190: push ebx

0xb7e65191: sub esp, 8

0xb7e65194: mov eax, DWORD PTR [esp+0x10]

...



```

CODE XREF: sub_312FD8
sub_312FD8+6
CODE XREF: sub_312FD8
sub_312FD8+49
CODE XREF: sub_312FD8

```

```

and eax, 0FFFFFFh
or eax, 80070000h
loc_31308C:
mov [ebp+var_4], eax

```

# ret2libc example

system()

0xb7e65190: push ebx

0xb7e65191: sub esp, 8

0xb7e65194: mov eax, DWORD PTR [esp+0x10]

...

```
push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], ebx
jnz short loc_313066
mov eax, [ebp+var_70]
cmp eax, [ebp+var_84]
jb short loc_313066
sub eax, [ebp+var_84]
push esi
push esi
push eax
push edi
mov [ebp+arg_0], eax
call sub_31486A
test eax, eax
jz short loc_31306D
push esi
lea eax, [ebp+arg_0]
push eax
mov esi, 1D0h
push esi
push [ebp+arg_4]
push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], esi
jz short loc_31308F

loc_313066: ; CODE XREF: sub_312FD8+55
call sub_31411B

loc_31306D: ; CODE XREF: sub_312FD8+49
; sub_312FD8+49
call sub_3140F3
test eax, eax
jg short loc_31307D
call sub_3140F3
jmp short loc_31308C

loc_31307D: ; CODE XREF: sub_312FD8+55
call sub_3140F3
and eax, 0FFFFFFh
or eax, 80070000h

loc_31308C: ; CODE XREF: sub_312FD8+55
mov [ebp+var_4], eax
```

# ret2libc example

```

push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], ebx
jnz short loc_313066
mov eax, [ebp+var_70]
cmp eax, [ebp+var_84]
jb short loc_313066
sub eax, [ebp+var_84]
push esi
push esi

```

system() --->

ret address --->

first arg --->

0x08045430: ret

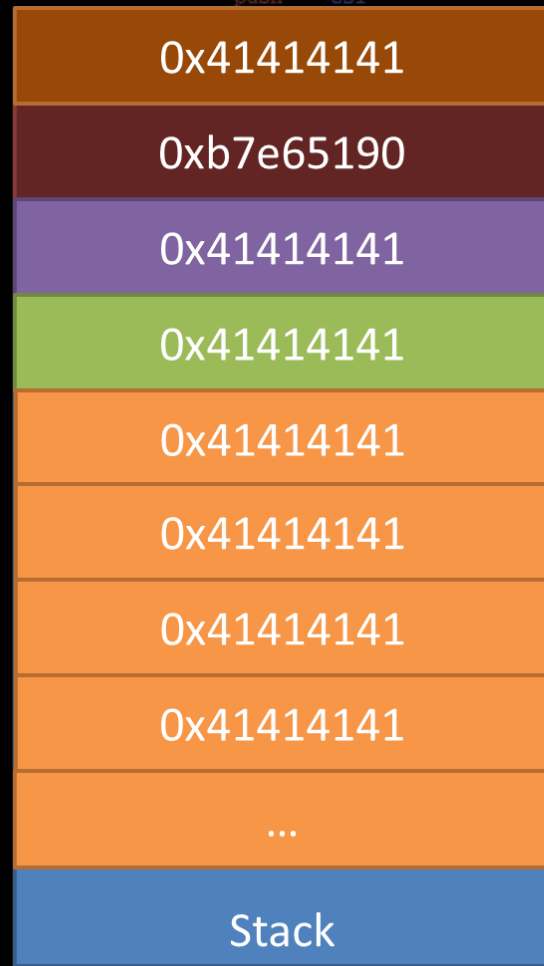
.....  
system()

0xb7e65190: push ebx

0xb7e65191: sub esp, 8

0xb7e65194: mov eax, DWORD PTR [esp+0x10]

...



CODE XREF: sub\_312FD8  
sub\_312FD8+6  
CODE XREF: sub\_312FD8  
sub\_312FD8+49  
CODE XREF: sub\_312FD8

```

and eax, 0FFFFFFh
or eax, 80070000h
loc_31308C:
mov [ebp+var_4], eax

```

# ret2libc example

```

push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], ebx
jnz short loc_313066
mov eax, [ebp+var_70]
cmp eax, [ebp+var_84]
jb short loc_313066
sub eax, [ebp+var_84]
push esi
push esi

```

← ESP

system()'s stack frame

ret address --->

first arg --->

```
0x08045430: ret
```

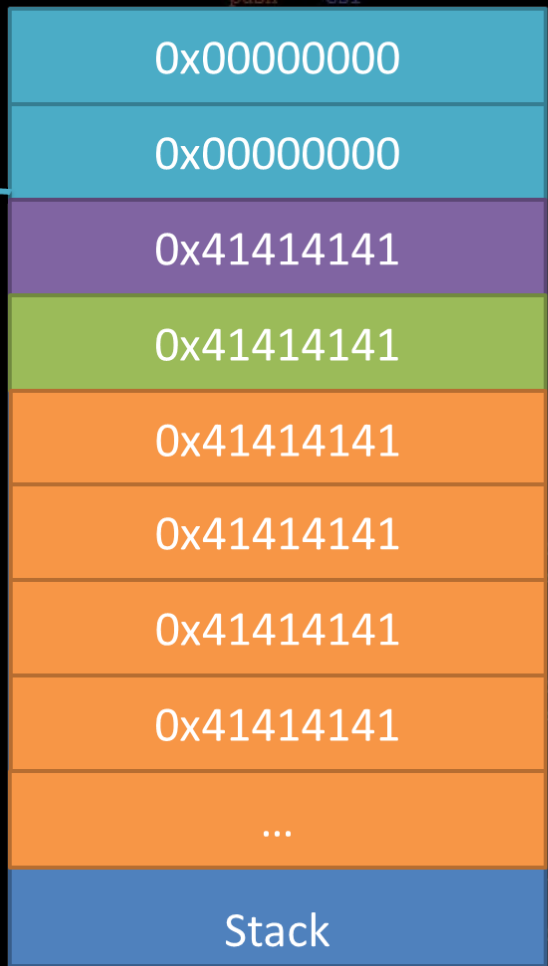
.....  
system()

```

0xb7e65190: push ebx
0xb7e65191: sub esp, 8
0xb7e65194: mov eax, DWORD PTR
[esp+0x10]

```

← EIP



```

CODE XREF: sub_312FD8
sub_312FD8+6
CODE XREF: sub_312FD8
sub_312FD8+49
CODE XREF: sub_312FD8

```

```

and eax, 0FFFFFFh
or eax, 80070000h
loc_31308C:
mov [ebp+var_4], eax

```

# ret2libc example

```

push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], ebx
jnz short loc_313066
mov eax, [ebp+var_70]
cmp eax, [ebp+var_84]
jb short loc_313066
sub eax, [ebp+var_84]
push esi
push esi

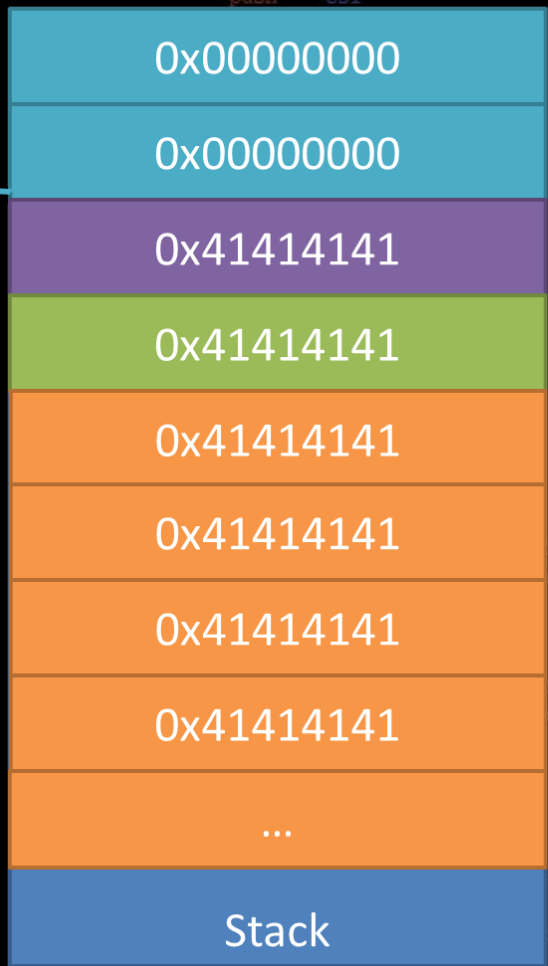
```



system()'s stack frame

ret address --->

first arg --->



Stack Growth

```
0x08045430: ret
```

.....  
system()

```

0xb7e65190: push ebx
0xb7e65191: sub esp, 8
0xb7e65194: mov eax, DWORD PTR
[esp+0x10]

```



...

```

and eax, 0FFFFFFh
or eax, 80070000h

```

```

loc_31308C: mov [ebp+var_4], eax
; CODE XREF: sub_312FD8

```

# REWIND

```
push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], ebx
jnz short loc_313066
mov eax, [ebp+var_70]
cmp eax, [ebp+var_84]
jb short loc_313066
sub eax, [ebp+var_84]
push esi
push esi
push eax
push edi
mov [ebp+arg_0], eax
call sub_31486A
test eax, eax
jz short loc_31306D
push esi
lea eax, [ebp+arg_0]
push eax
mov esi, 1D0h
push esi
push [ebp+arg_4]
push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], esi
jz short loc_31308F

loc_313066: ; CODE XREF: sub_312FD8
; sub_312FD8+56
push 0Dh
call sub_31411B

loc_31306D: ; CODE XREF: sub_312FD8
; sub_312FD8+49
call sub_3140F3
test eax, eax
jg short loc_31307D
call sub_3140F3
jmp short loc_31308C
; -----

loc_31307D: ; CODE XREF: sub_312FD8
call sub_3140F3
and eax, 0FFFFFFh
or eax, 80070000h

loc_31308C: ; CODE XREF: sub_312FD8
mov [ebp+var_4], eax
```

# ret2libc example

```

push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], ebx
jnz short loc_313066
mov eax, [ebp+var_70]
cmp eax, [ebp+var_84]
jb short loc_313066
sub eax, [ebp+var_84]
push esi
push esi

```

system() --->

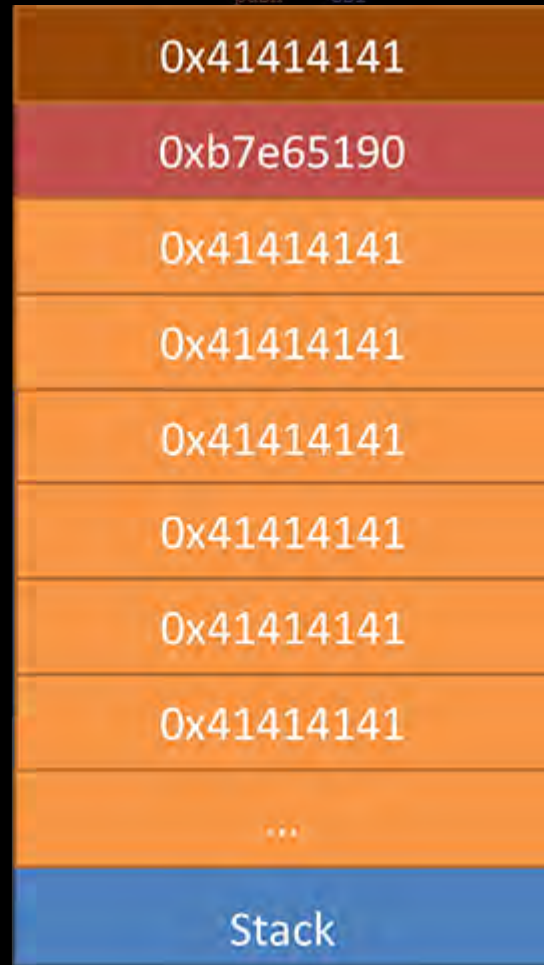
0x08045430: ret ← EIP

.....  
system()

```

0xb7e65190: push ebx
0xb7e65191: sub esp, 8
0xb7e65194: mov eax, DWORD PTR
[esp+0x10]
...

```



CODE XREF: sub\_312FD8  
sub\_312FD8+6

CODE XREF: sub\_312FD8  
sub\_312FD8+49

CODE XREF: sub\_312FD8

```

and eax, 0FFFFFFh
or eax, 80070000h

```

```

loc_31308C: mov [ebp+var_4], eax
; CODE XREF: sub_312FD8

```

# ret2libc example

```

push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], ebx
jnz short loc_313066
mov eax, [ebp+var_70]
cmp eax, [ebp+var_84]
jb short loc_313066
sub eax, [ebp+var_84]
push esi
push esi

```

system() --->

ret address --->

first arg --->  
"cat flag.txt"

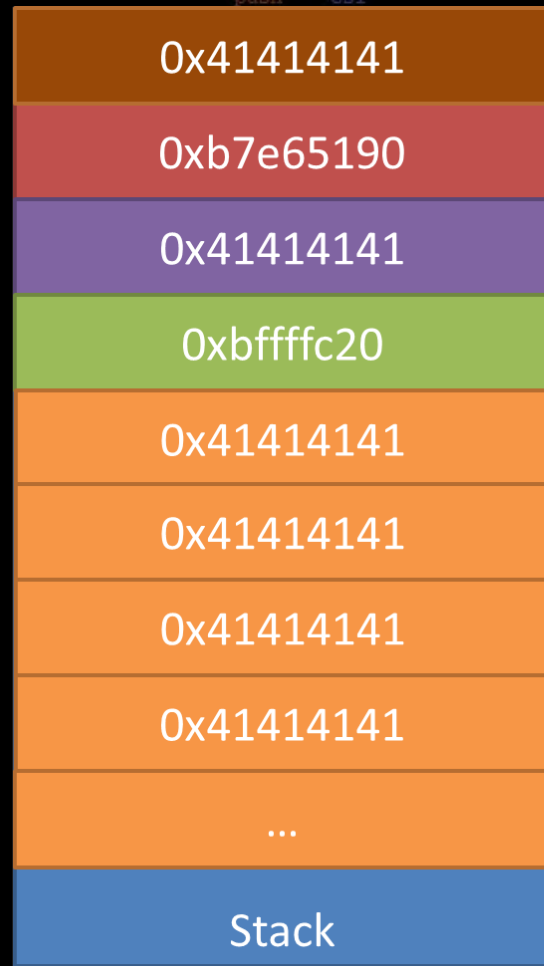
0x08045430: ret ← EIP

.....  
system()

```

0xb7e65190: push ebx
0xb7e65191: sub esp, 8
0xb7e65194: mov eax, DWORD PTR
[esp+0x10]
...

```



CODE XREF: sub\_312FD8  
sub\_312FD8+6  
CODE XREF: sub\_312FD8  
sub\_312FD8+49  
CODE XREF: sub\_312FD8

```

and eax, 0FFFFFFh
or eax, 80070000h
loc_31308C:
mov [ebp+var_4], eax

```



# ret2libc example

```

push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], ebx
jnz short loc_313066
mov eax, [ebp+var_70]
cmp eax, [ebp+var_84]
jb short loc_313066
sub eax, [ebp+var_84]
push esi
push esi

```

system() --->

ret address --->

first arg --->  
"cat flag.txt"

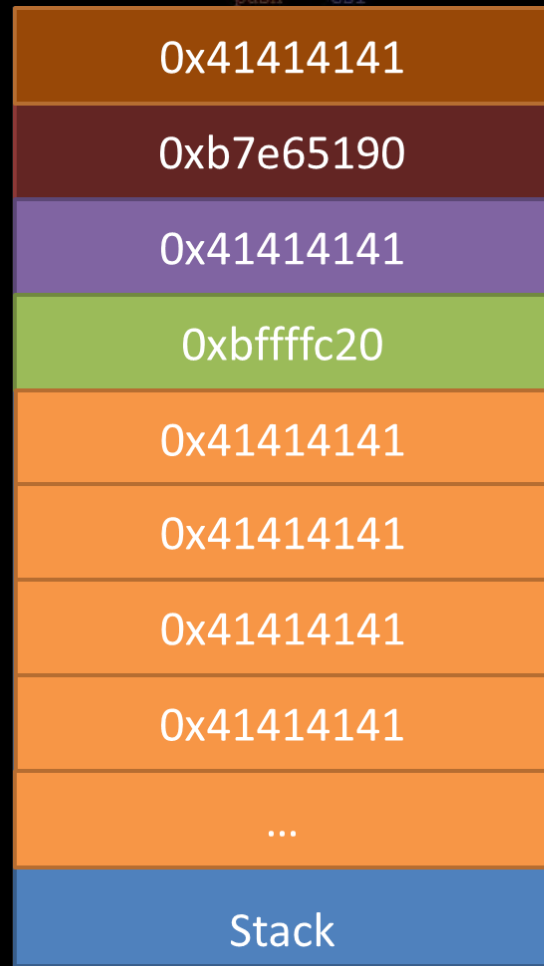
```
0x08045430: ret
```

.....  
system()

```

0xb7e65190: push ebx
0xb7e65191: sub esp, 8
0xb7e65194: mov eax, DWORD PTR [esp+0x10]
...

```



```

CODE XREF: sub_312FD8
sub_312FD8+6
CODE XREF: sub_312FD8
sub_312FD8+49
CODE XREF: sub_312FD8

```

```

and eax, 0FFFFFFh
or eax, 80070000h
loc_31308C:
mov [ebp+var_4], eax

```

# ret2libc example

```

push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], ebx
jnz short loc_313066
mov eax, [ebp+var_70]
cmp eax, [ebp+var_84]
jb short loc_313066
sub eax, [ebp+var_84]
push esi
push esi

```

← ESP

system()'s stack frame

ret address --->

first arg --->  
"cat flag.txt"

```
0x08045430: ret
```

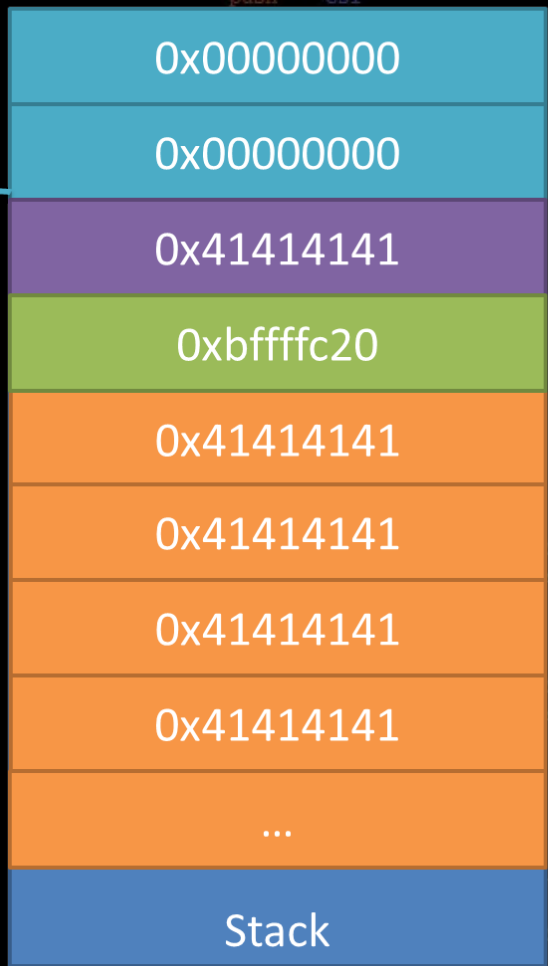
.....  
system()

```

0xb7e65190: push ebx
0xb7e65191: sub esp, 8
0xb7e65194: mov eax, DWORD PTR
[esp+0x10]

```

← EIP



↑ Stack Growth

```

CODE XREF: sub_312FD8
sub_312FD8+6
CODE XREF: sub_312FD8
sub_312FD8+49
CODE XREF: sub_312FD8

```

```

and eax, 0FFFFFFh
or eax, 80070000h
loc_31308C:
mov [ebp+var_4], eax

```

# ret2libc example

```

push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], ebx
jnz short loc_313066
mov eax, [ebp+var_70]
cmp eax, [ebp+var_84]
jb short loc_313066
sub eax, [ebp+var_84]
push esi
push esi

```

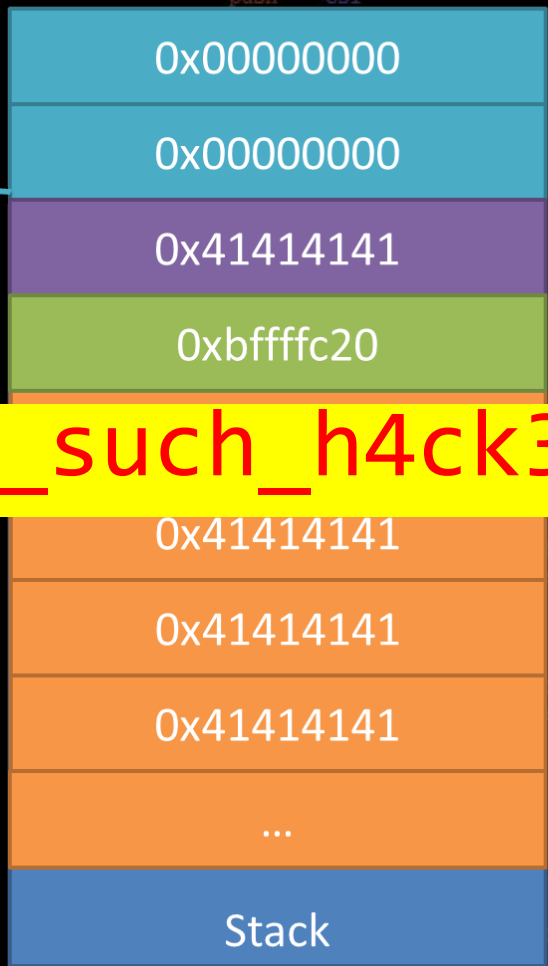
← ESP

system()'s stack frame

ret address --->

first arg --->

"cat flag.txt"



0x08 w0w\_u\_g0t\_th3\_f14g\_such\_h4ck3r

```

system()
0xb7e65190: push ebx
0xb7e65191: sub esp, 8
0xb7e65194: mov eax, DWORD PTR
[esp+0x10]
...

```

← EIP

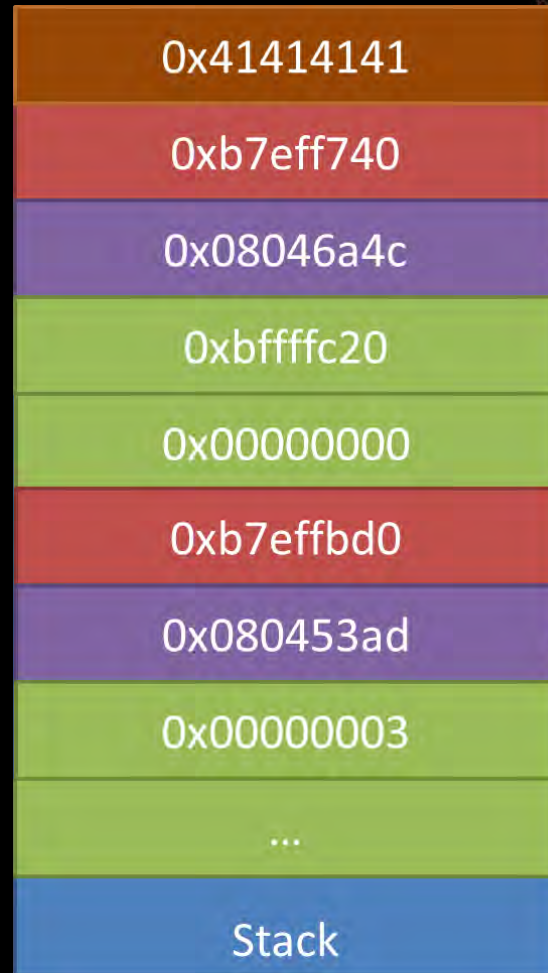
```

and eax, 0FFFFFFh
or eax, 80070000h
loc_31308C:
mov [ebp+var_4], eax

```

# Chaining Calls

**open()** --->  
 pivot --->  
 first arg --->  
 second arg --->  
**read()** --->  
 ret address --->  
 first arg --->



```

push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], ebx
jnz short loc_313066
mov eax, [ebp+var_70]
cmp eax, [ebp+var_84]
jb short loc_313066
sub eax, [ebp+var_84]
push esi
sh esi
sh eax
sh si
v [ebp+arg_0], eax
ll sub_31486A
st eax
short loc_31306D
sh esi
a eax, [ebp+arg_0]
sh eax
v esi, 1D0h
sh esi
sh [ebp+arg_4]
sh edi
ll sub_314623
st eax
short loc_31306D
p [ebp+arg_0], esi
short loc_31308F
; CODE XREF: sub_312FD8
; sub_312FD8+56
sh 1Dh
sub_31411B
; CODE XREF: sub_312FD8
; sub_312FD8+49
ll sub_3140F3
st eax, eax
short loc_31307D
ll sub_3140F3
p short loc_31308C
; CODE XREF: sub_312FD8
ll sub_3140F3
and eax, 0FFFFFFh
or eax, 80070000h

```

# Lab 5

- lab5C - ret2libc
- lab5B - Basic ROP
- lab5A - More involved ROP

```
push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], ebx
jnz short loc_313066
mov eax, [ebp+var_70]
cmp eax, [ebp+var_84]
jb short loc_313066
sub eax, [ebp+var_84]
push esi
push esi
push eax
push edi
mov [ebp+arg_0], eax
call sub_31486A
test eax, eax
jz short loc_31306D
push esi
lea eax, [ebp+arg_0]
push eax
mov esi, 1D0h
push esi
push [ebp+arg_4]
push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_4], esi
jz short loc_31308F
```

```
loc_313066: ; CODE XREF: sub_312FD8
; sub_312FD8+56
```

```
push 0Dh
call sub_31411B
```

```
loc_31306D: ; CODE XREF: sub_312FD8
; sub_312FD8+49
```

```
call sub_3140F3
test eax, eax
jg short loc_31307D
call sub_3140F3
jmp short loc_31308C
```

```
loc_31307D: ; CODE XREF: sub_312FD8
```

```
call sub_3140F3
and eax, 0FFFFFFh
or eax, 80070000h
```

```
loc_31308C: ; CODE XREF: sub_312FD8
```

```
mov [ebp+var_4], eax
```