Tools and Basic Reverse Engineering – Part 2

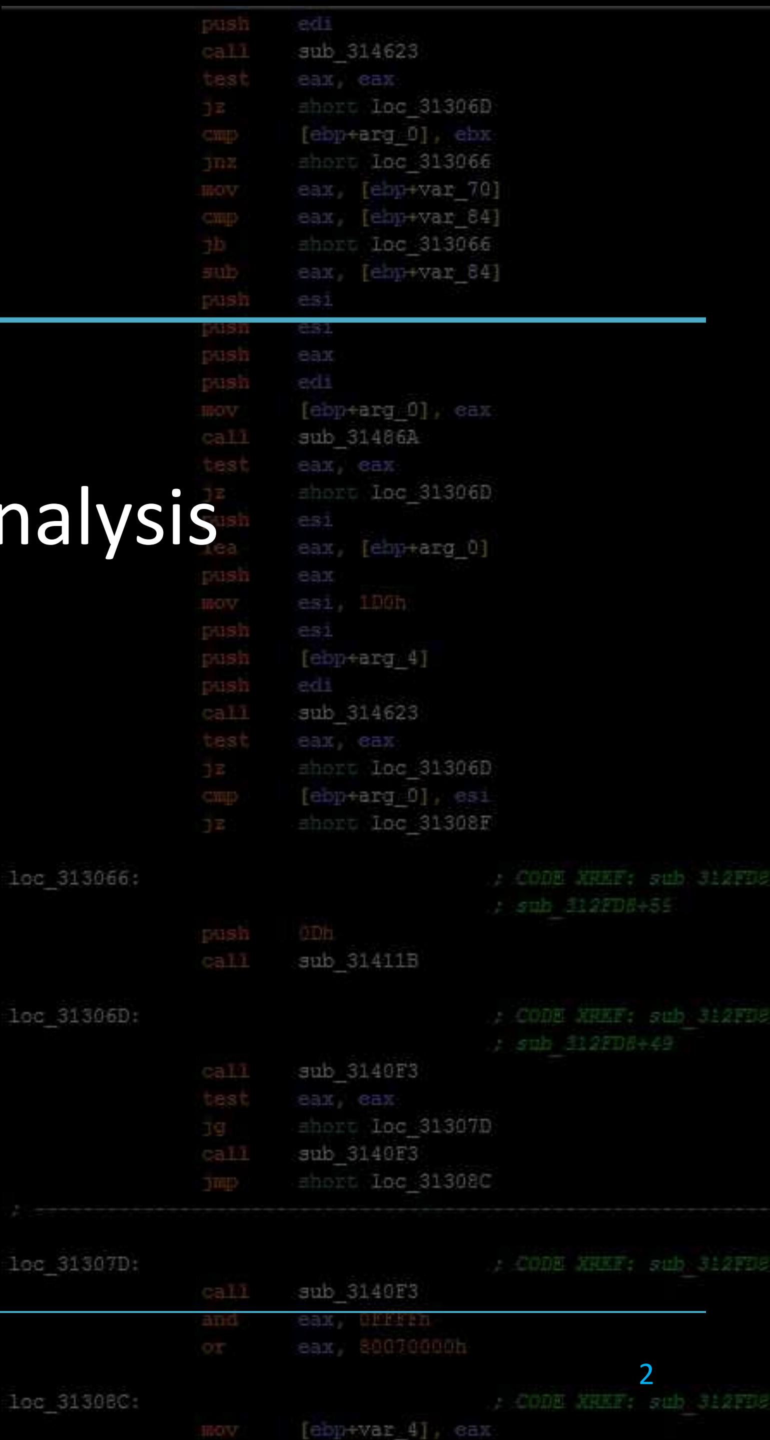
Modern Binary Exploitation

CSCI 4968 – Spring 2015

Jeremy Blackthorne

Lecture Overview

1. Review of Last Lecture
2. Introduction to Dynamic Analysis
3. Tools!
4. Resources

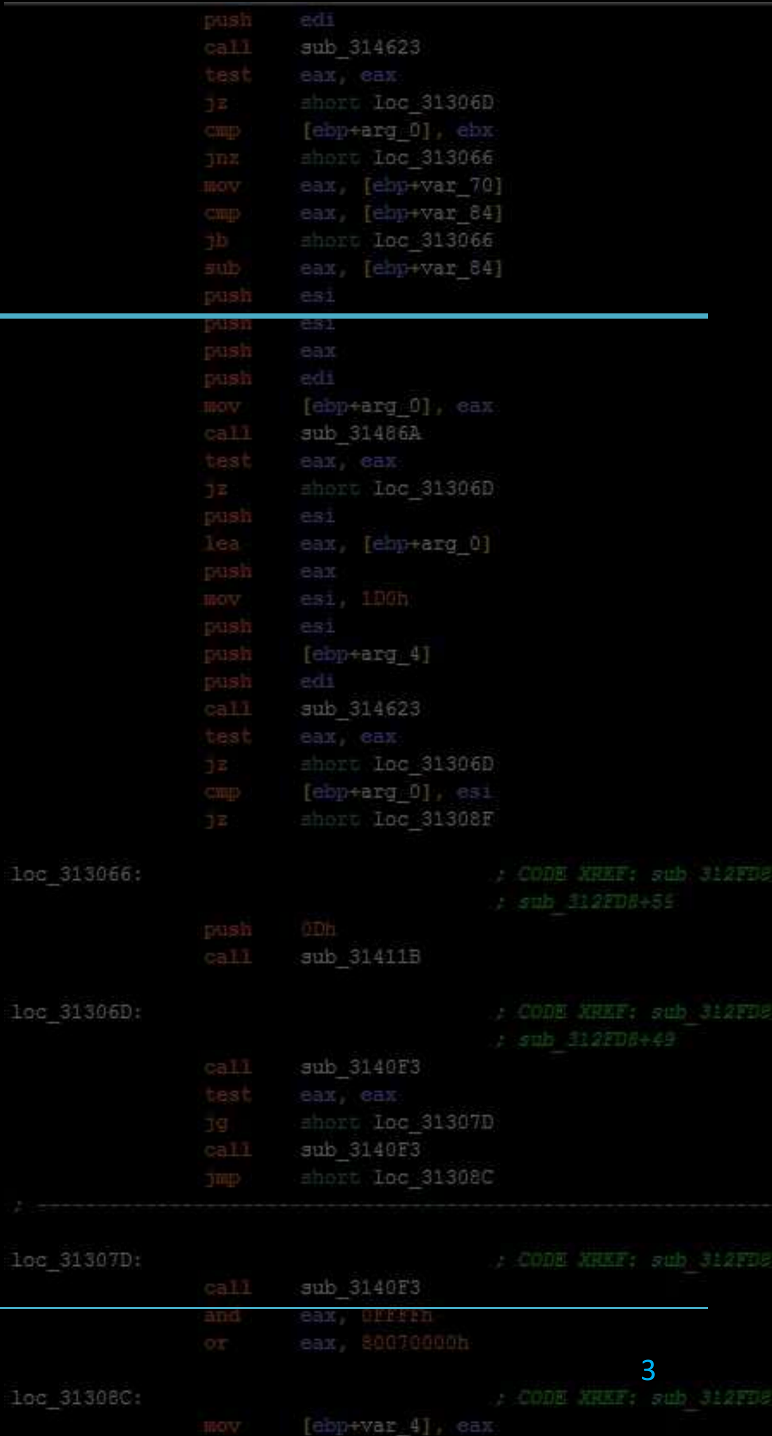


```
                push    edi
                call    sub_314623
                test    eax, eax
                jz      short loc_31306D
                cmp     [ebp+arg_0], ebx
                jnz     short loc_313066
                mov     eax, [ebp+var_70]
                cmp     eax, [ebp+var_84]
                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, 0FFFFh
                or      eax, 80070000h

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

Review

Reversing Concepts:

- Static vs dynamic
- Diffing
- patching



```
                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, 0FFFFh
                or      eax, 80070000h

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

Review

Tools:

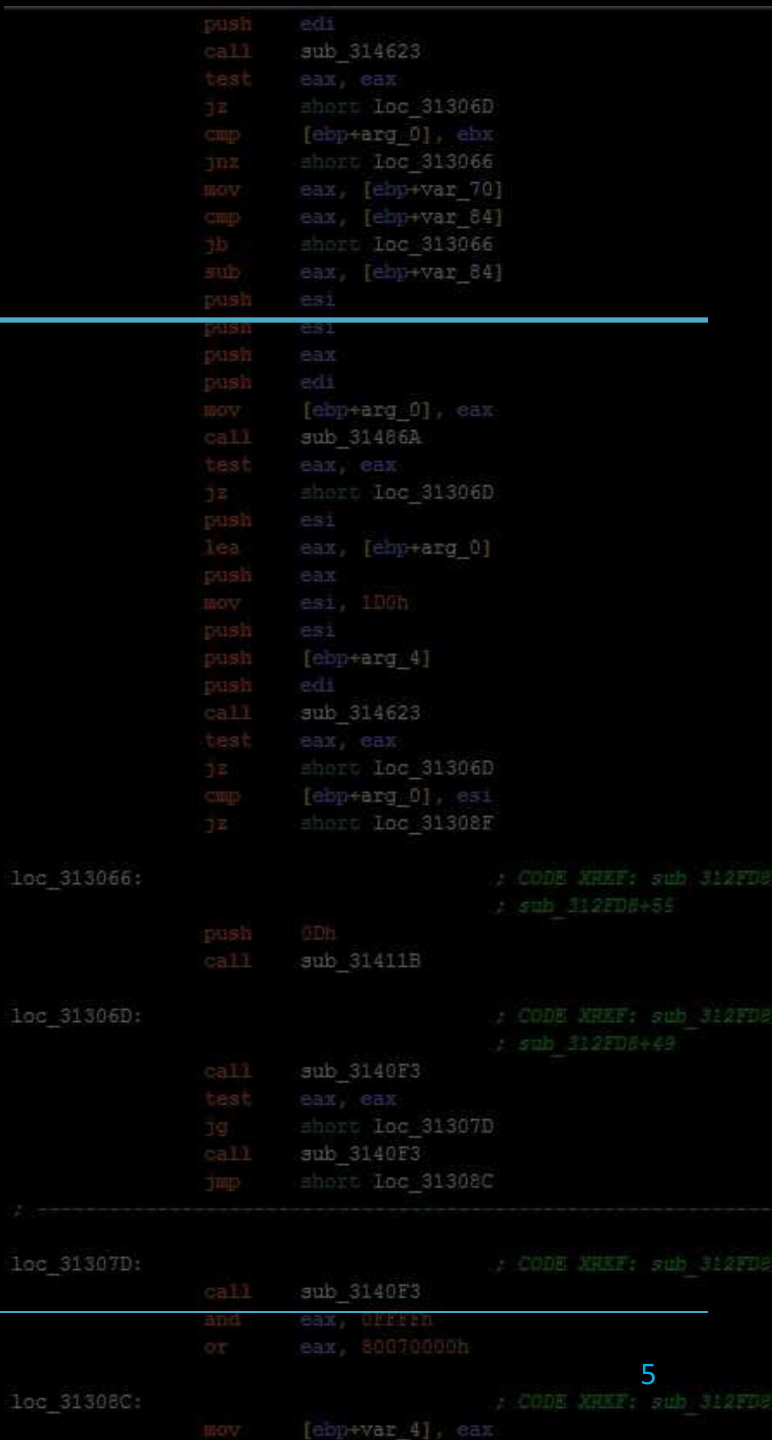
- –file
- –md5sum
- –ssdeep
- –strings
- –readelf
- –objdump
- –IDA Pro.exe

```
                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, 0FFFFh
                or      eax, 80070000h

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

Review

IDA Pro:

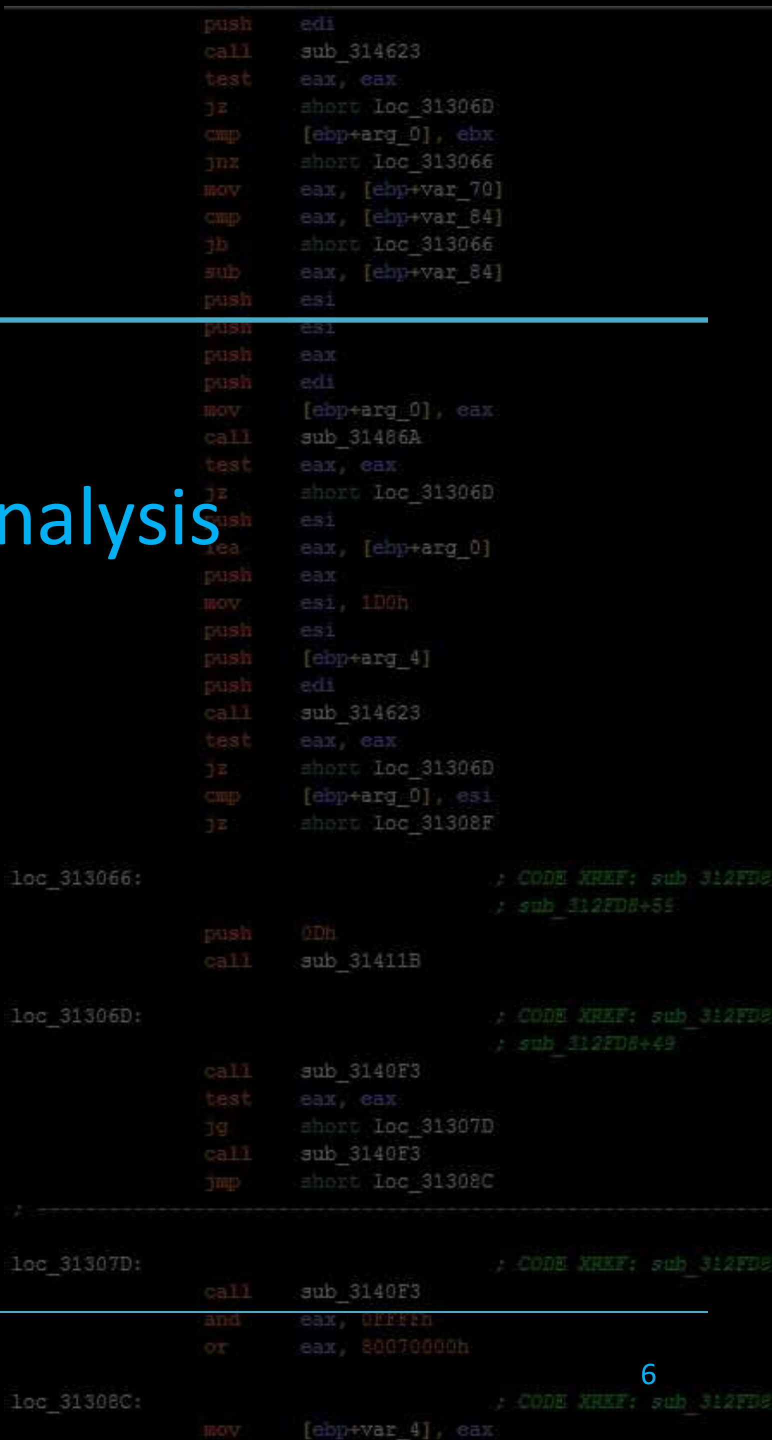
- –Rename variables
- –Insert comments
- –Recognize structures
- –Cross reference
- –Stack usage in assembly



```
                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, 0FFFFh
                or      eax, 80070000h

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

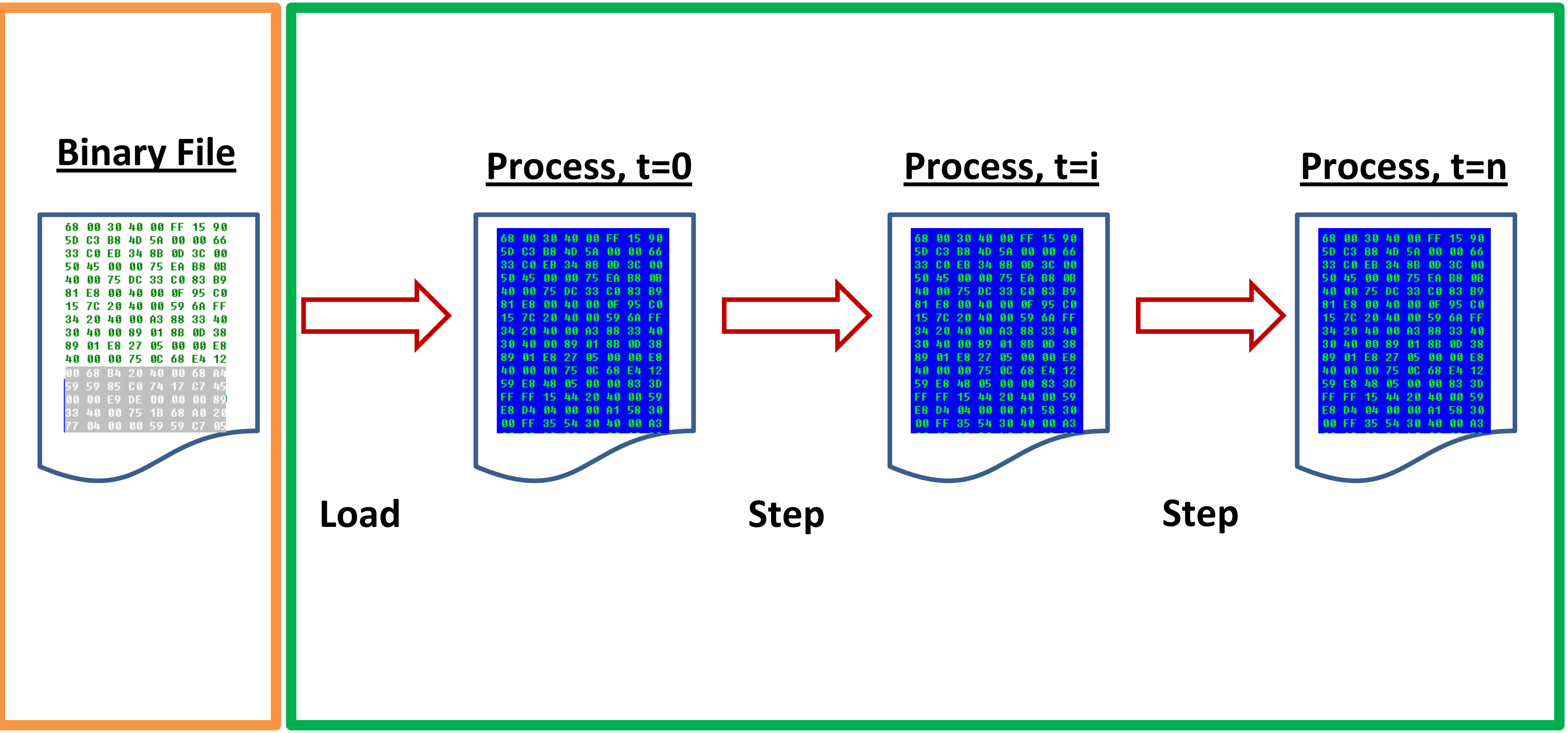
Lecture Overview

1. Review of Last Lecture
2. Introduction to Dynamic Analysis
3. Tools!
4. Resources



```
                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, 100h
                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, 0FFFFh
                or      eax, 80070000h

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

RE Domain



Binary File

```
68 00 30 40 00 FF 15 90
5D C3 B8 4D 5A 00 00 66
33 C0 EB 34 8B 0D 3C 00
50 45 00 00 75 EA B8 0B
40 00 75 DC 33 C0 83 B9
81 E8 00 40 00 0F 95 C0
15 7C 20 40 00 59 6A FF
34 20 40 00 A3 88 33 40
30 40 00 89 01 8B 0D 38
89 01 E8 27 05 00 00 E8
40 00 00 75 0C 68 E4 12
00 68 B4 20 40 00 68 A4
59 59 85 C0 74 17 C7 45
00 00 E9 DE 00 00 00 89
33 40 00 75 1B 68 A0 20
77 04 00 00 59 59 C7 05
```

Load

Process, t=0

```
68 00 30 40 00 FF 15 90
5D C3 B8 4D 5A 00 00 66
33 C0 EB 34 8B 0D 3C 00
50 45 00 00 75 EA B8 0B
40 00 75 DC 33 C0 83 B9
81 E8 00 40 00 0F 95 C0
15 7C 20 40 00 59 6A FF
34 20 40 00 A3 88 33 40
30 40 00 89 01 8B 0D 38
89 01 E8 27 05 00 00 E8
40 00 00 75 0C 68 E4 12
59 E8 48 05 00 00 83 3D
FF FF 15 44 20 40 00 59
E8 D4 04 00 00 A1 58 30
00 FF 35 54 30 40 00 A3
```

Step

Process, t=i

```
68 00 30 40 00 FF 15 90
5D C3 B8 4D 5A 00 00 66
33 C0 EB 34 8B 0D 3C 00
50 45 00 00 75 EA B8 0B
40 00 75 DC 33 C0 83 B9
81 E8 00 40 00 0F 95 C0
15 7C 20 40 00 59 6A FF
34 20 40 00 A3 88 33 40
30 40 00 89 01 8B 0D 38
89 01 E8 27 05 00 00 E8
40 00 00 75 0C 68 E4 12
59 E8 48 05 00 00 83 3D
FF FF 15 44 20 40 00 59
E8 D4 04 00 00 A1 58 30
00 FF 35 54 30 40 00 A3
```

Step

Process, t=n

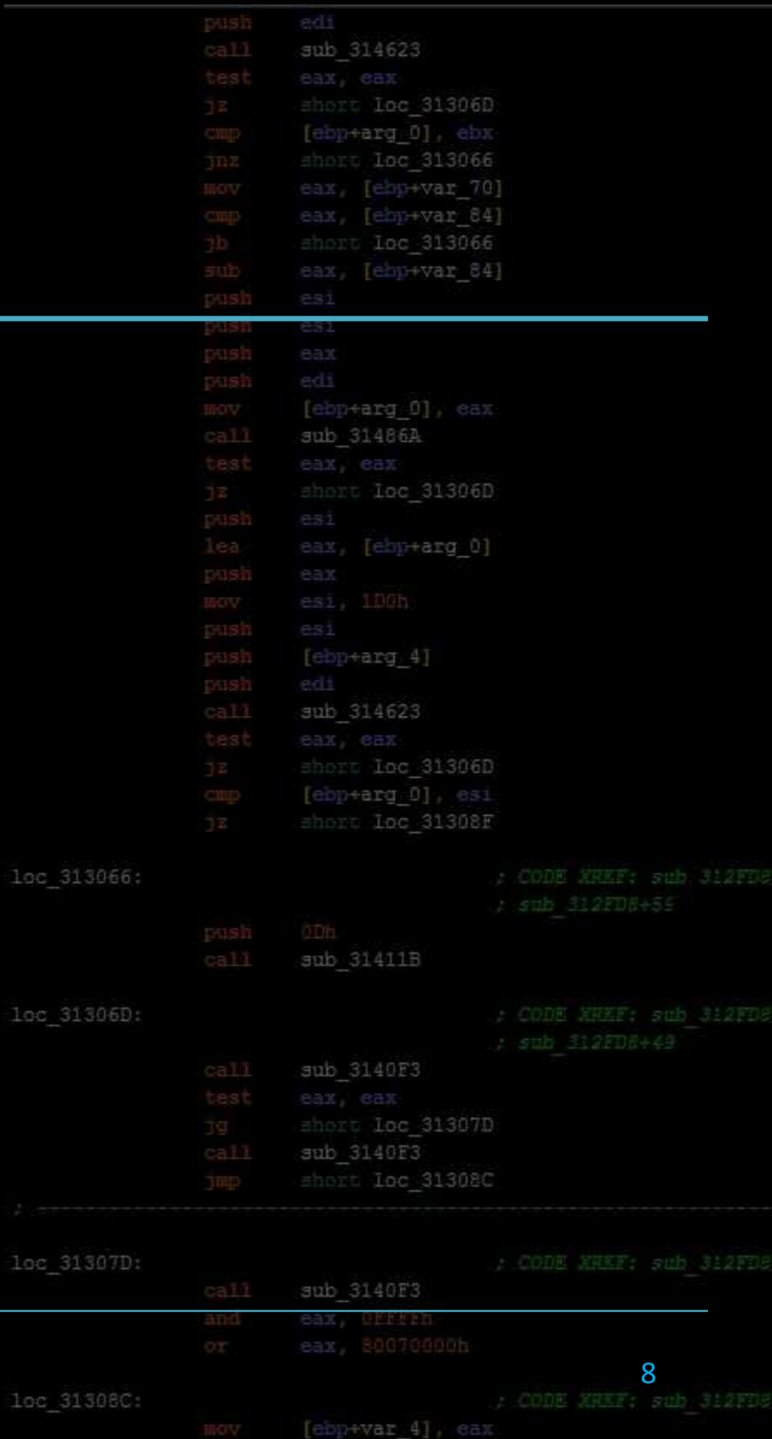
```
68 00 30 40 00 FF 15 90
5D C3 B8 4D 5A 00 00 66
33 C0 EB 34 8B 0D 3C 00
50 45 00 00 75 EA B8 0B
40 00 75 DC 33 C0 83 B9
81 E8 00 40 00 0F 95 C0
15 7C 20 40 00 59 6A FF
34 20 40 00 A3 88 33 40
30 40 00 89 01 8B 0D 38
89 01 E8 27 05 00 00 E8
40 00 00 75 0C 68 E4 12
59 E8 48 05 00 00 83 3D
FF FF 15 44 20 40 00 59
E8 D4 04 00 00 A1 58 30
00 FF 35 54 30 40 00 A3
```

Static

Dynamic

Slide Colors

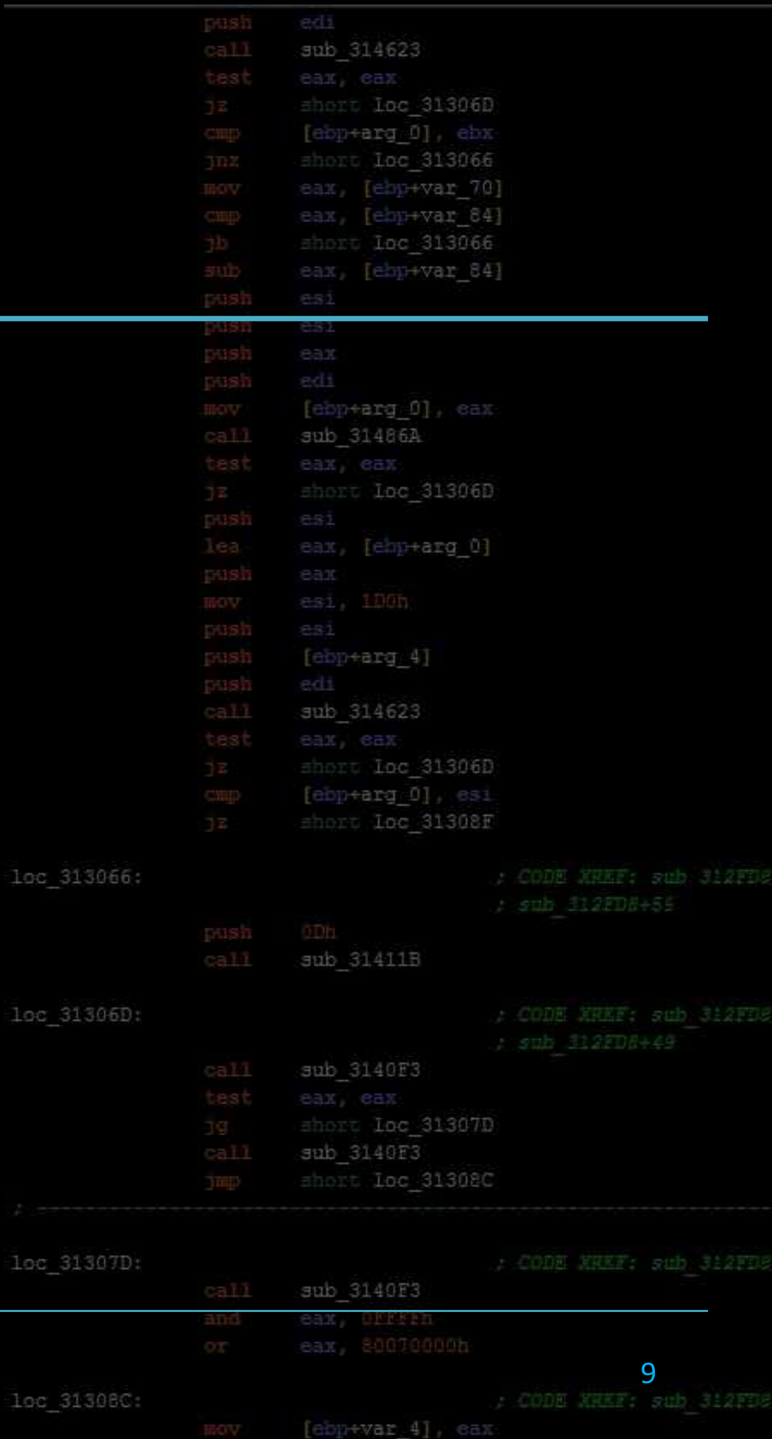
- Linux Tool
  - Command
- Windows Tool
  - ToolName.exe
- Associated Challenges:
  - ChallengeName



```
                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, 0FFFFh
                or      eax, 80070000h

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

Debugger – IDA Pro

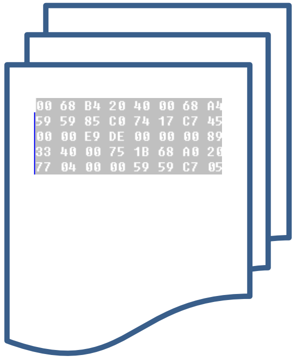
- crackme0x04_win.exe
- IDA Pro.exe



```
                        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, 100h
                        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_312FD
                                                ; sub_312FD8+55
                        push    0Dh
                        call    sub_31411B

loc_31306D:                                     ; CODE XREF: sub_312FD
                                                ; 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_312FD
                        call    sub_3140F3
                        and     eax, 0FFFFh
                        or      eax, 80070000h

loc_31308C:                                     ; CODE XREF: sub_312FD
                        mov     [ebp+var_4], eax
```

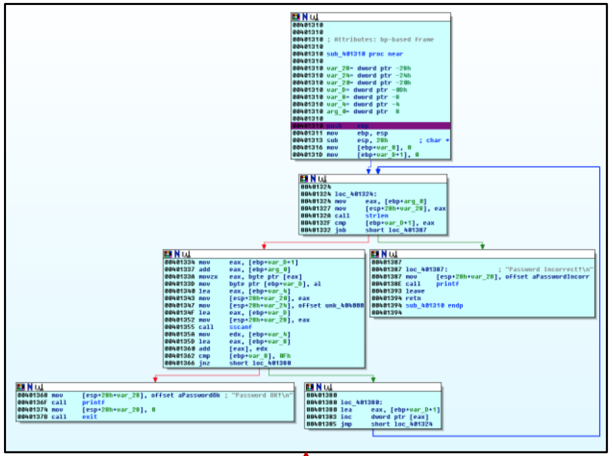
RE Domain

Libraries



```
00 68 B4 20 40 00 68 A4
59 59 85 C0 74 17 C7 45
00 00 E9 DE 00 00 00 89
33 40 00 75 1B 68 A0 20
77 04 00 00 59 59 C7 05
```

Code



```
00401310
00401310
00401310 ; Attributes: bp-based frame
00401310
00401310 sub_401310 proc near
00401310
00401310 var_28= dword ptr -28h
00401310 var_24= dword ptr -24h
00401310 var_20= dword ptr -20h
00401310 var_D= dword ptr -0Dh
00401310 var_8= dword ptr -8
00401310 var_4= dword ptr -4
00401310 arg_0= dword ptr  8
00401310
00401310 push    ebp
00401311 mov     ebp, esp
00401313 sub     esp, 28h        ; char *
00401316 mov     [ebp+var_8], 0
0040131D mov     [ebp+var_D+1], 0
```

```
00401324
00401324 loc_401324:
00401324 mov     eax, [ebp+arg_0]
00401327 mov     [esp+28h+var_28], eax
0040132A call    strlen
0040132F cmp     [ebp+var_D+1], eax
00401332 jnb     short loc_401387
```

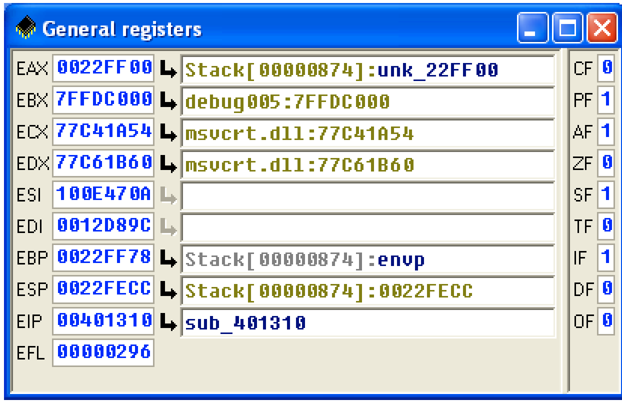
```
00401334 mov     eax, [ebp+var_D+1]
00401337 add     eax, [ebp+arg_0]
0040133A movzx   eax, byte ptr [eax]
0040133D mov     byte ptr [ebp+var_D], al
00401340 lea     eax, [ebp+var_4]
00401343 mov     [esp+28h+var_20], eax
00401347 mov     [esp+28h+var_24], offset unk_404000
0040134F lea     eax, [ebp+var_D]
00401352 mov     [esp+28h+var_28], eax
00401355 call    sscanf
0040135A mov     edx, [ebp+var_4]
0040135D lea     eax, [ebp+var_8]
00401360 add     [eax], edx
00401362 cmp     [ebp+var_8], 0Fh
00401366 jnz     short loc_401380
```

```
00401387
00401387 loc_401387:                 ; "Password Incorrect!\n"
00401387 mov     [esp+28h+var_28], offset aPasswordIncorr
0040138E call    printf
00401393 leave
00401394 retn
00401394 sub_401310 endp
00401394
```

```
00401368 mov     [esp+28h+var_28], offset aPasswordOk ; "Password OK!\n"
0040136F call    printf
00401374 mov     [esp+28h+var_28], 0
0040137B call    exit
```

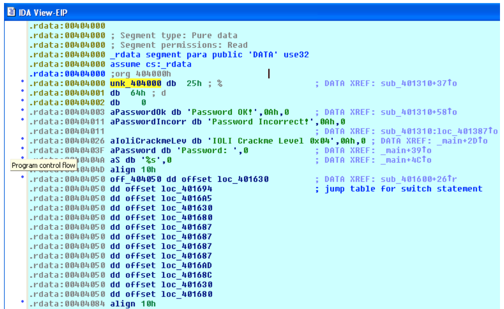
```
00401380
00401380 loc_401380:
00401380 lea     eax, [ebp+var_D+1]
00401383 inc     dword ptr [eax]
00401385 jmp     short loc_401324
```

Registers



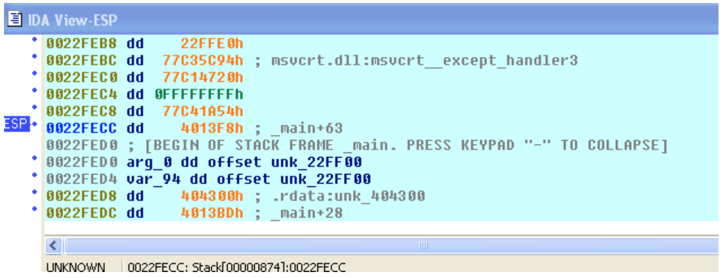
```
EAX 0022FF00 ↳ Stack[00000874]:unk_22FF00      CF 0
EBX 7FFDC000 ↳ debug005:7FFDC000                PF 1
ECX 77C41A54 ↳ msvcrt.dll:77C41A54              AF 1
EDX 77C61B60 ↳ msvcrt.dll:77C61B60              ZF 0
ESI 100E470A ↳                                  SF 1
EDI 0012D89C ↳                                  TF 0
EBP 0022FF78 ↳ Stack[00000874]:envp             IF 1
ESP 0022FECC ↳ Stack[00000874]:0022FECC         DF 0
EIP 00401310 ↳ sub_401310                       OF 0
EFL 00000296
```

Other Memory



```
.rdata:00404000
.rdata:00404000 ; Segment type: Pure data
.rdata:00404000 ; Segment permissions: Read
.rdata:00404000 _rdata segment para public 'DATA' use32
.rdata:00404000 assume cs:_rdata
.rdata:00404000 ;org 404000h
.rdata:00404000 unk_404000 db  25h ; %               ; DATA XREF: sub_401310+37↑o
.rdata:00404001 db  64h ; d
.rdata:00404002 db    0
.rdata:00404003 aPasswordOk db 'Password OK!',0Ah,0   ; DATA XREF: sub_401310+58↑o
.rdata:00404011 aPasswordIncorr db 'Password Incorrect!',0Ah,0
.rdata:00404011                                       ; DATA XREF: sub_401310:loc_401387↑o
.rdata:00404026 aIoliCrackmeLev db 'IOLI Crackme Level 0x04',0Ah,0 ; DATA XREF: _main+2D↑o
.rdata:0040403F aPassword db 'Password: ',0           ; DATA XREF: _main+39↑o
           0404A aS db '%s',0                         ; DATA XREF: _main+4C↑o
           0404D align 10h
.rdata:00404050 off_404050 dd offset loc_401630       ; DATA XREF: sub_401600+26↑r
.rdata:00404050 dd offset loc_401694                  ; jump table for switch statement
.rdata:00404050 dd offset loc_4016A5
.rdata:00404050 dd offset loc_401630
.rdata:00404050 dd offset loc_401680
.rdata:00404050 dd offset loc_401687
.rdata:00404050 dd offset loc_401687
.rdata:00404050 dd offset loc_401687
.rdata:00404050 dd offset loc_401687
.rdata:00404050 dd offset loc_4016AD
.rdata:00404050 dd offset loc_40168C
.rdata:00404050 dd offset loc_401630
.rdata:00404050 dd offset loc_401680
.rdata:00404084 align 10h
```

Stack

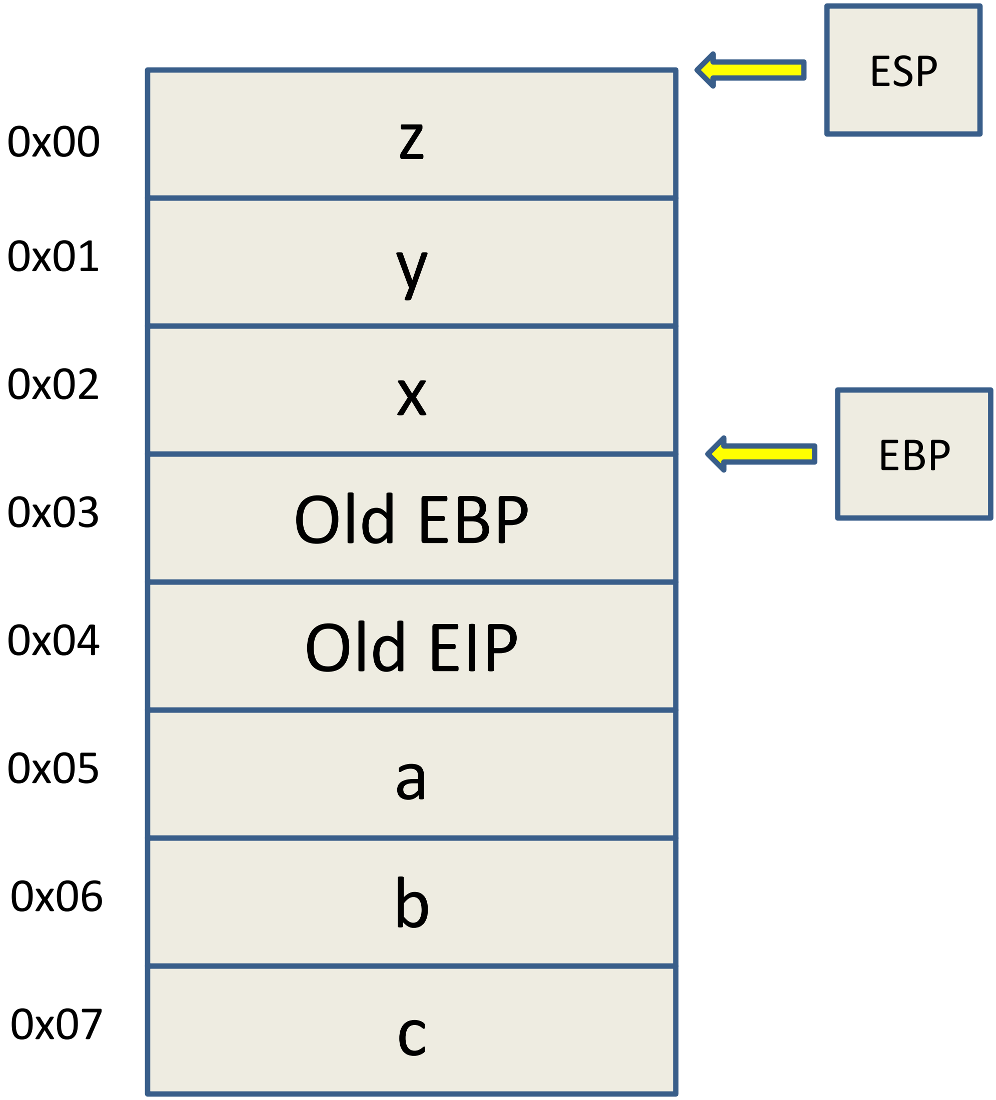


```
0022FEB8 dd    22FFE0h
0022FEBC dd  77C35C94h ; msvcrt.dll:msvcrt__except_handler3
0022FEC0 dd  77C14720h
0022FEC4 dd 0FFFFFFFFh
0022FEC8 dd  77C41A54h
0022FECC dd    4013F8h ; _main+63
0022FED0 ; [BEGIN OF STACK FRAME _main. PRESS KEYPAD "-" TO COLLAPSE]
0022FED0 arg_0 dd offset unk_22FF00
0022FED4 var_94 dd offset unk_22FF00
0022FED8 dd    404300h ; .rdata:unk_404300
0022FEDC dd    4013BDh ; _main+28
```

```
UNKNOWN    0022FECC: Stack[00000874]:0022FECC
```

Stack

```
int foo(int a, int b, int c)
{
    int x;
    int y;
    int z;

    x=y=z=0;
    z=x+y+a+b+c;
    return z;


}
int main(int argc, char **argv) {


        foo(1,2,3);
}
```



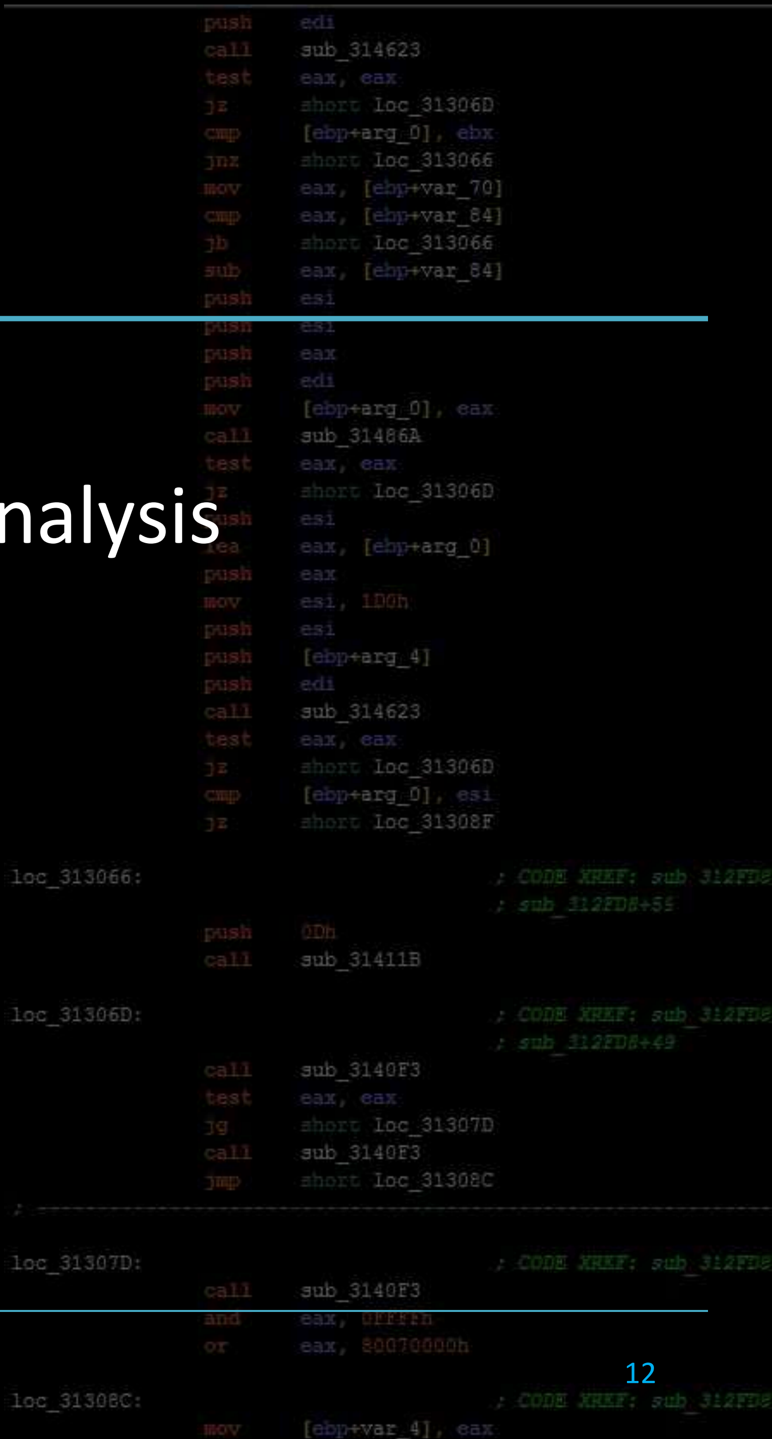
ESP

EBP

0x00	z
0x01	y
0x02	x
0x03	Old EBP
0x04	Old EIP
0x05	a
0x06	b
0x07	c

Lecture Overview

1. Review of Last Lecture
2. Introduction to Dynamic Analysis
3. Tools!
4. Resources



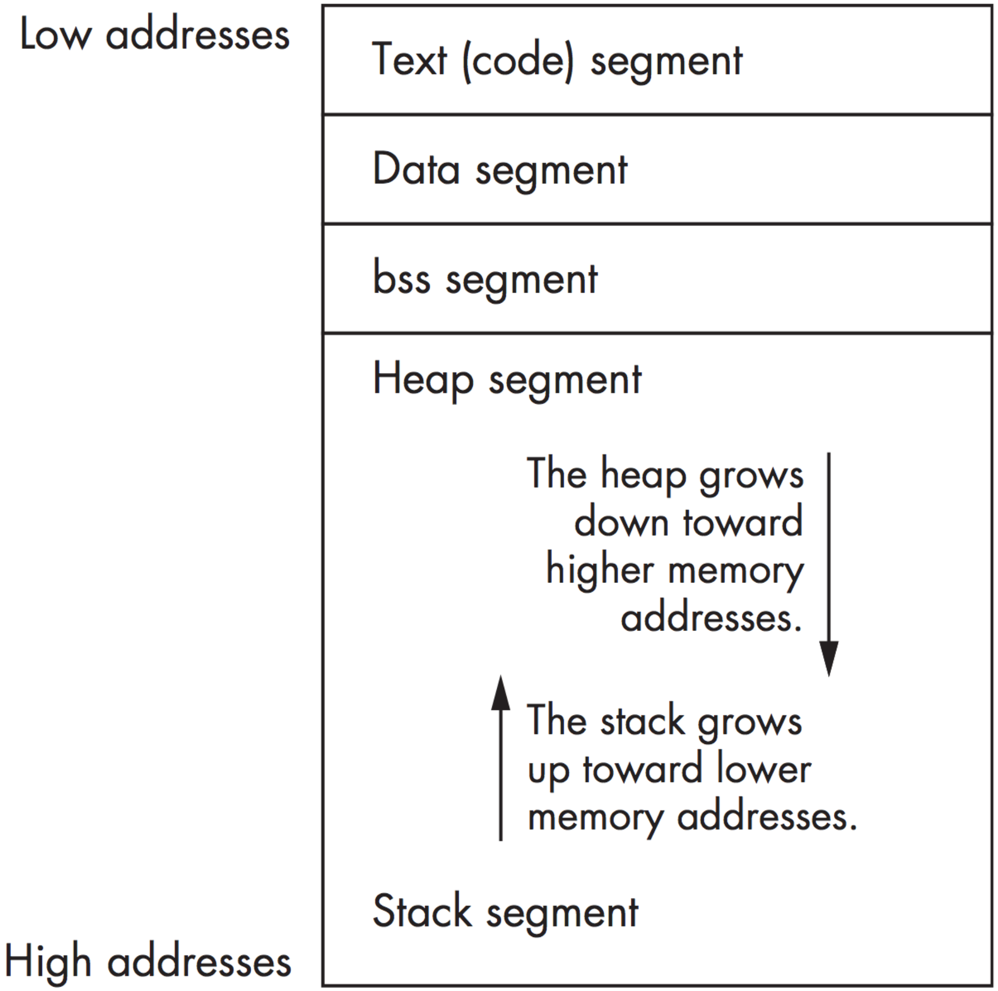
```
            push    edi
            call    sub_314623
            test    eax, eax
            jz      short loc_31306D
            cmp     [ebp+arg_0], ebx
            jnz     short loc_313066
            mov     eax, [ebp+var_70]
            cmp     eax, [ebp+var_84]
            jb      short 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, 100h
            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, 0FFFFh
            or      eax, 80070000h

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

Debugger – Evan’s Debugger

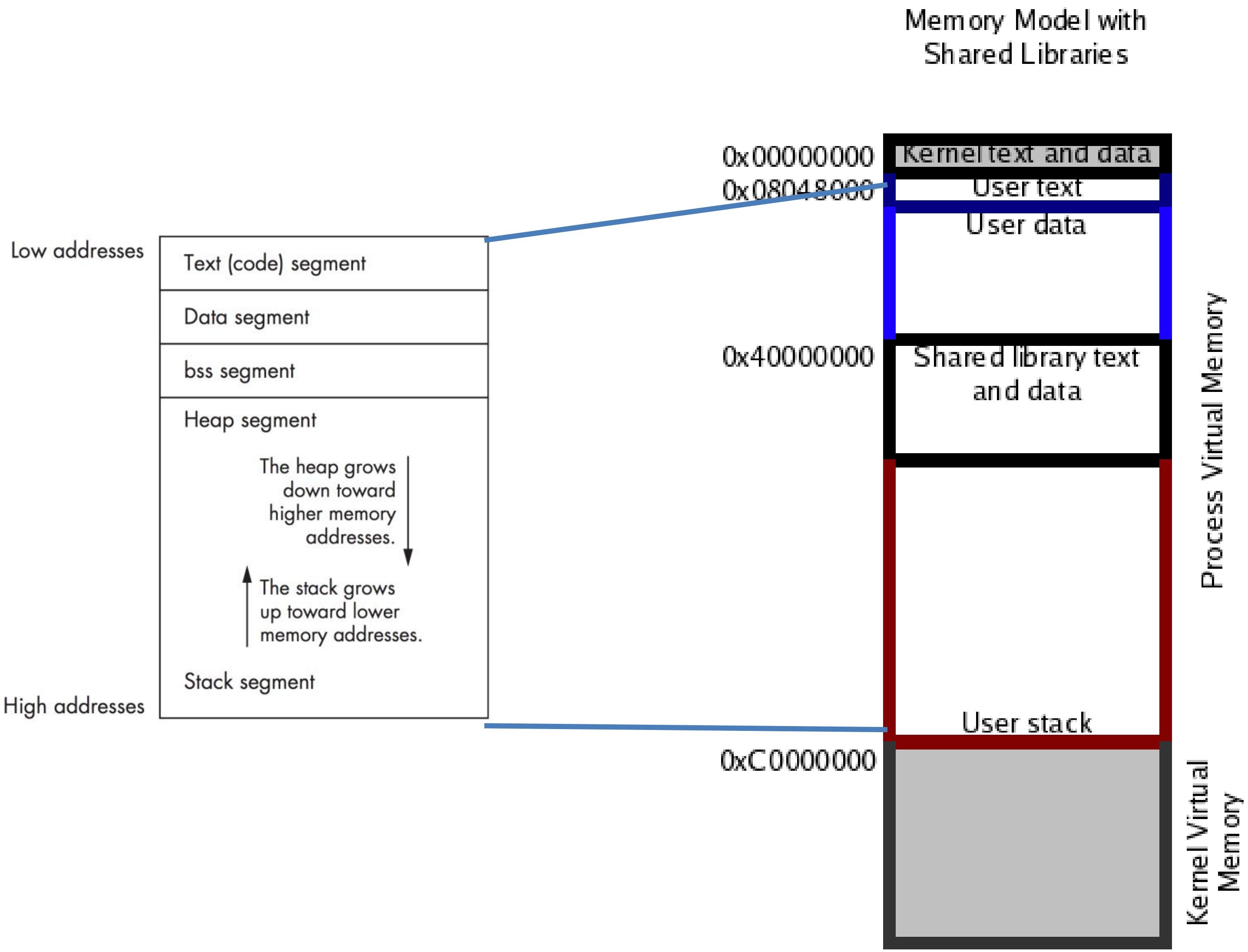
- crackme0x00a.exe
- edb
  - edb->options->Preferences->Appearance

```
                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    0Dh
                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, 0FFFFh
                or      eax, 80070000h

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

ELF Memory Layout


Low addresses
Text (code) segment
Data segment
bss segment
Heap segment
The heap grows down toward higher memory addresses.
The stack grows up toward lower memory addresses.
Stack segment
High addresses

Virtual Memory Layout



Low addresses

Text (code) segment

Data segment

bss segment

Heap segment

The heap grows down toward higher memory addresses.

The stack grows up toward lower memory addresses.

Stack segment

High addresses

Memory Model with Shared Libraries

0x00000000 Kernel text and data

0x08048000 User text

User data

0x40000000 Shared library text and data

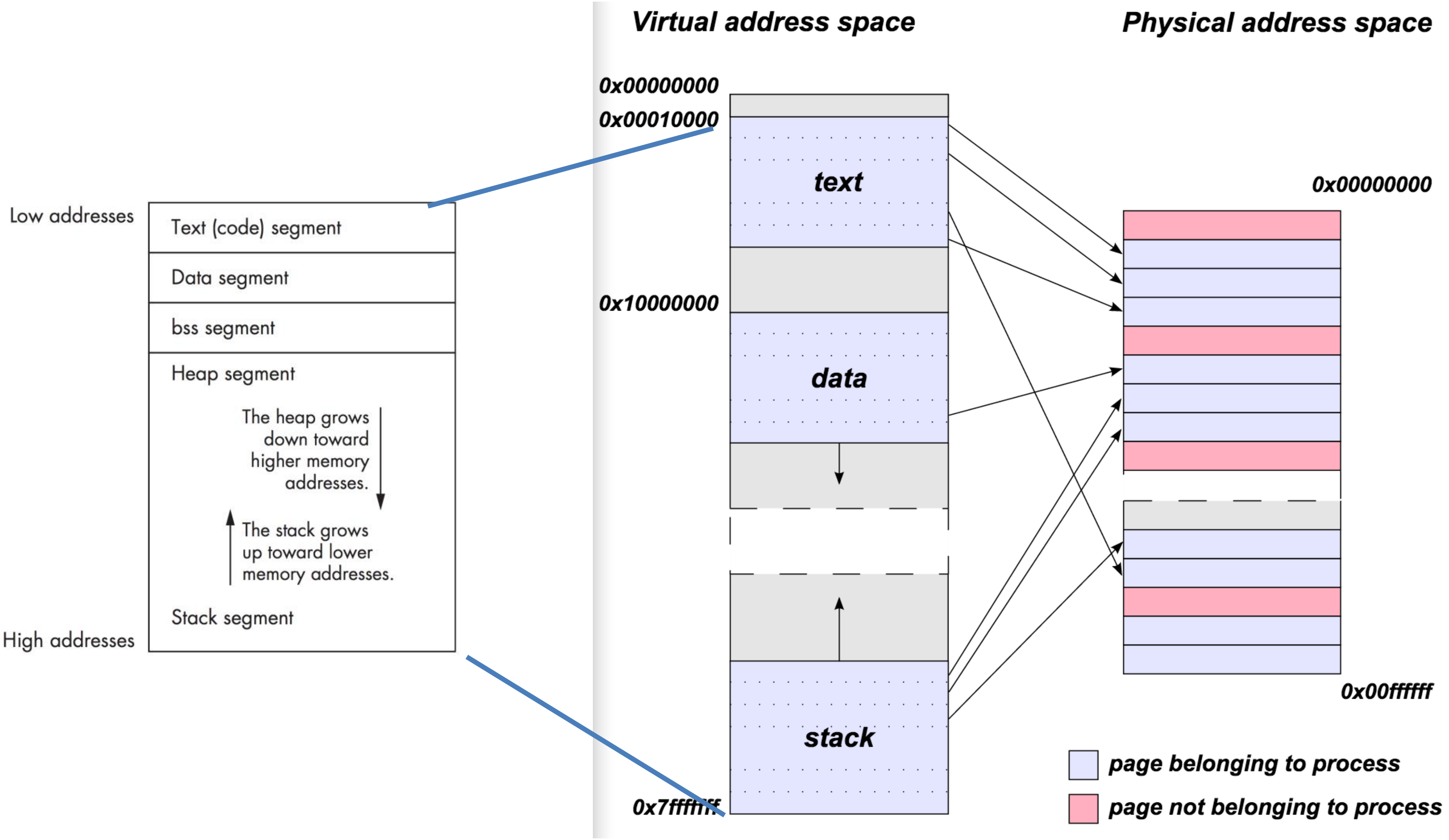
User stack

0xC0000000

Process Virtual Memory

Kernel Virtual Memory

Physical Memory Layout



Low addresses

Text (code) segment

Data segment

bss segment

Heap segment

The heap grows down toward higher memory addresses.

The stack grows up toward lower memory addresses.

Stack segment

High addresses

Virtual address space

0x00000000

0x00010000

text

0x10000000

data

stack

0x7fffffff

Physical address space

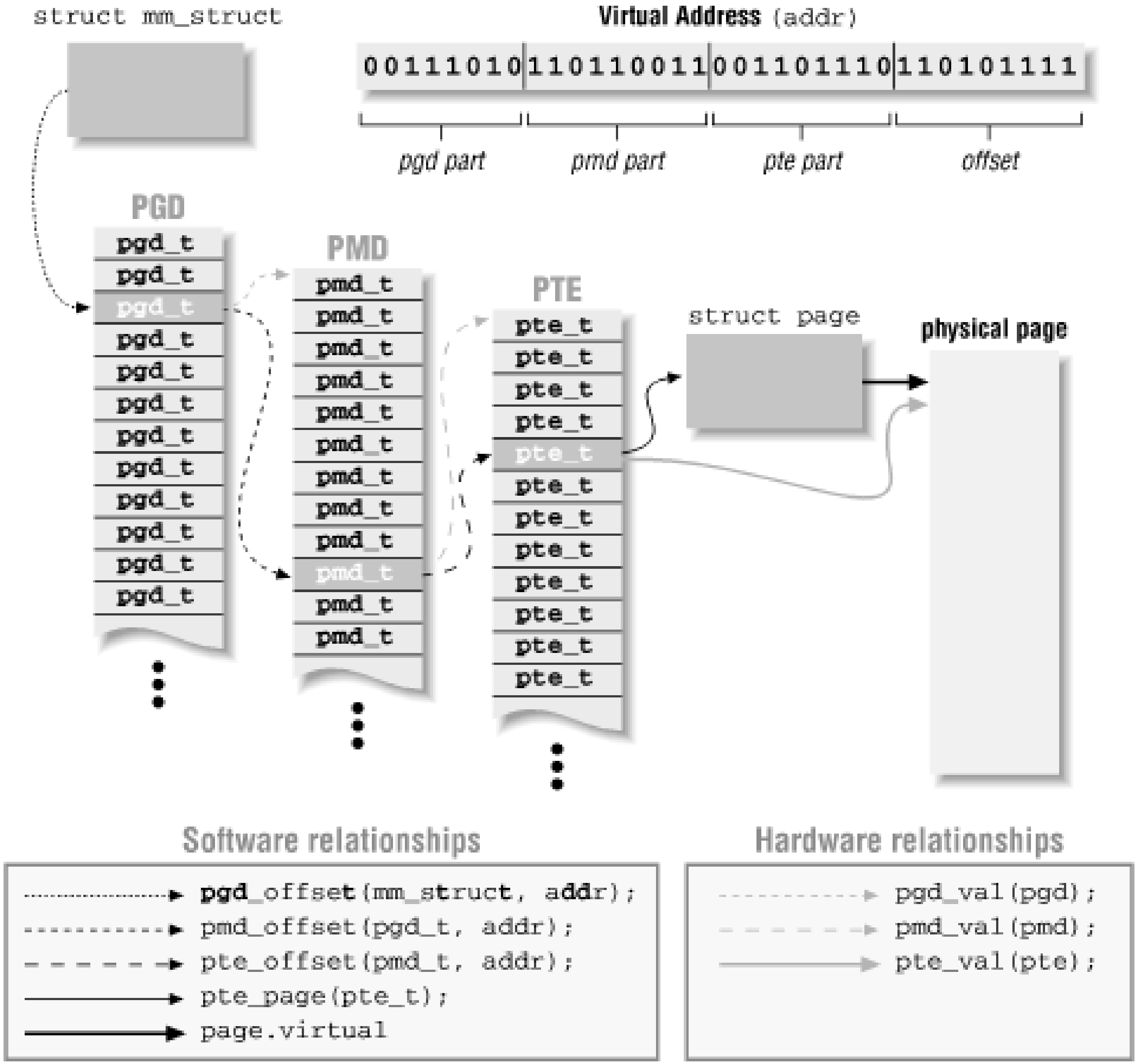
0x00000000

0x00ffffff

☐ page belonging to process

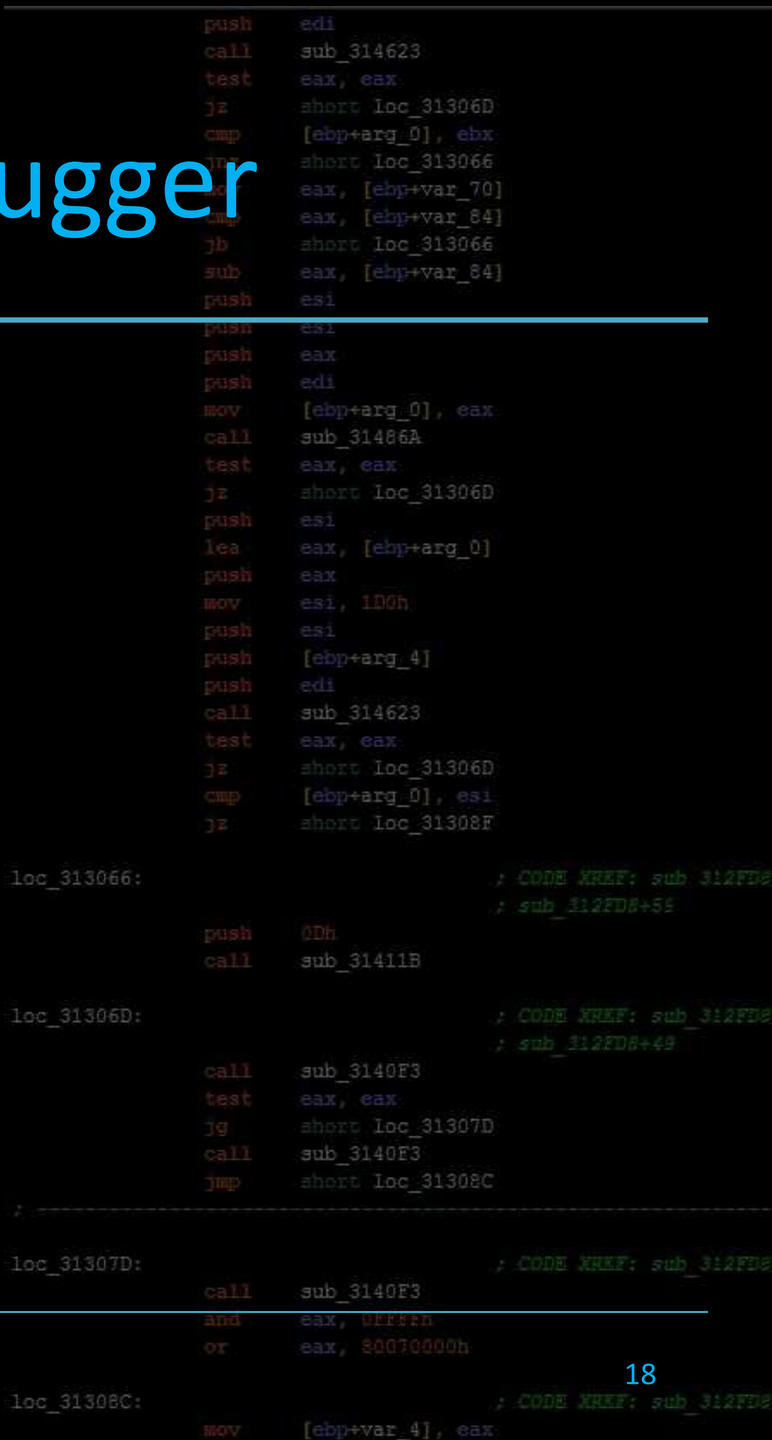
🟥 page not belonging to process

Physical Memory Layout



Debugger – GNU Debugger

- crackme0x00a
- gdb



```
        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, 0FFFFh
        or      eax, 80070000h

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

GNU Debugger - Basics

- crackme0x00a
- gdb
  - – disassemble main (disas main)
  - – set disassembly-flavor intel
  - – break main (b main)
  - – run
  - – stepi (s), step into
  - – nexti (n), step over

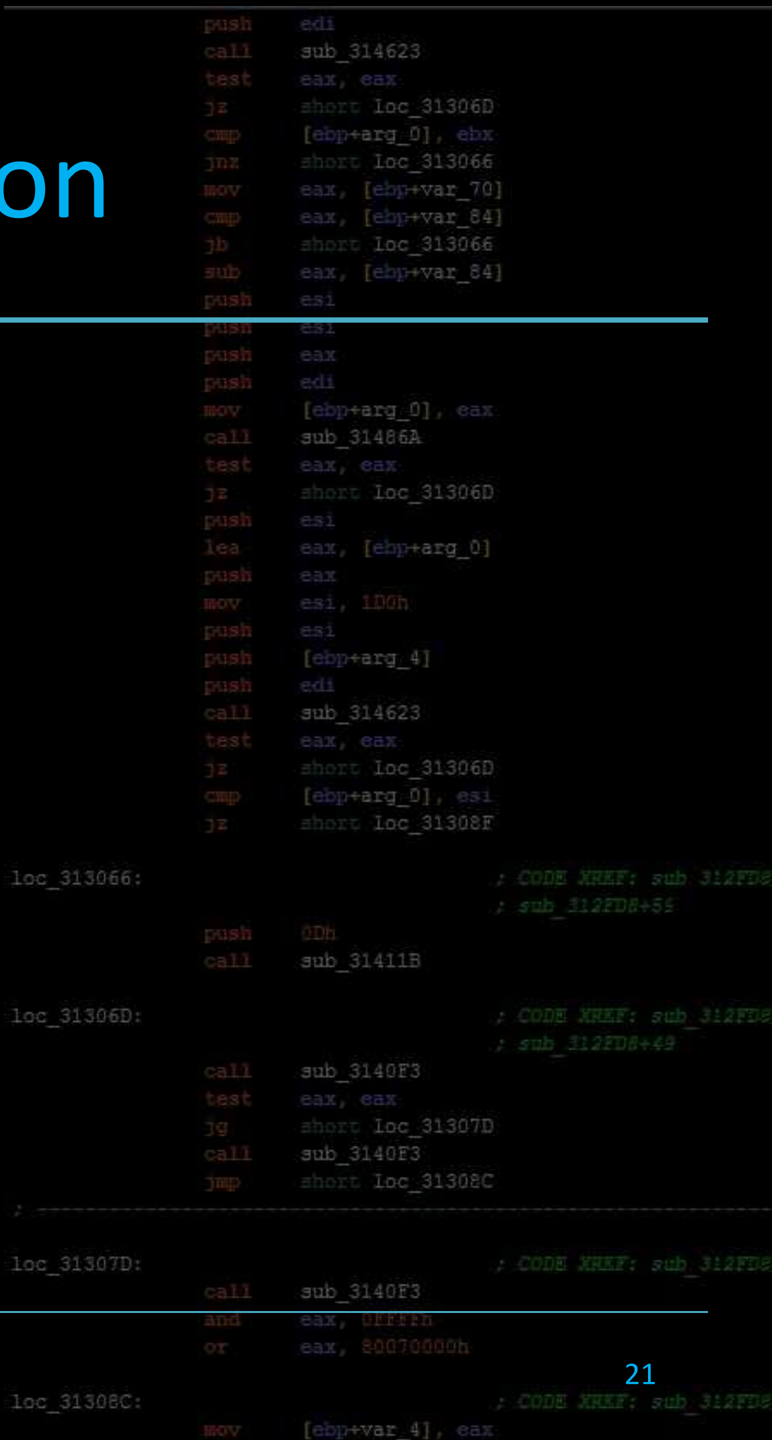
```
                 push    edi
                 call    sub_314623
                 test    eax, eax
                 jz      short loc_31306D
                 cmp     [ebp+arg_0], ebx
                 jnz     short loc_313066
                 mov     eax, [ebp+var_70]
                 cmp     eax, [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, 0FFFFh
                 or      eax, 80070000h

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

GNU Debugger – Examine Memory

- gdb
  - Examine memory: x/NFU address
  - N = number
  - F = format
  - U = unit
- Examples
  - x/10xb 0xdeadbeef, examine 10 bytes in hex
  - x/xw 0xdeadbeef, examine 1 word in hex
  - x/s 0xdeadbeef, examine null terminated string

GNU Debugger - python

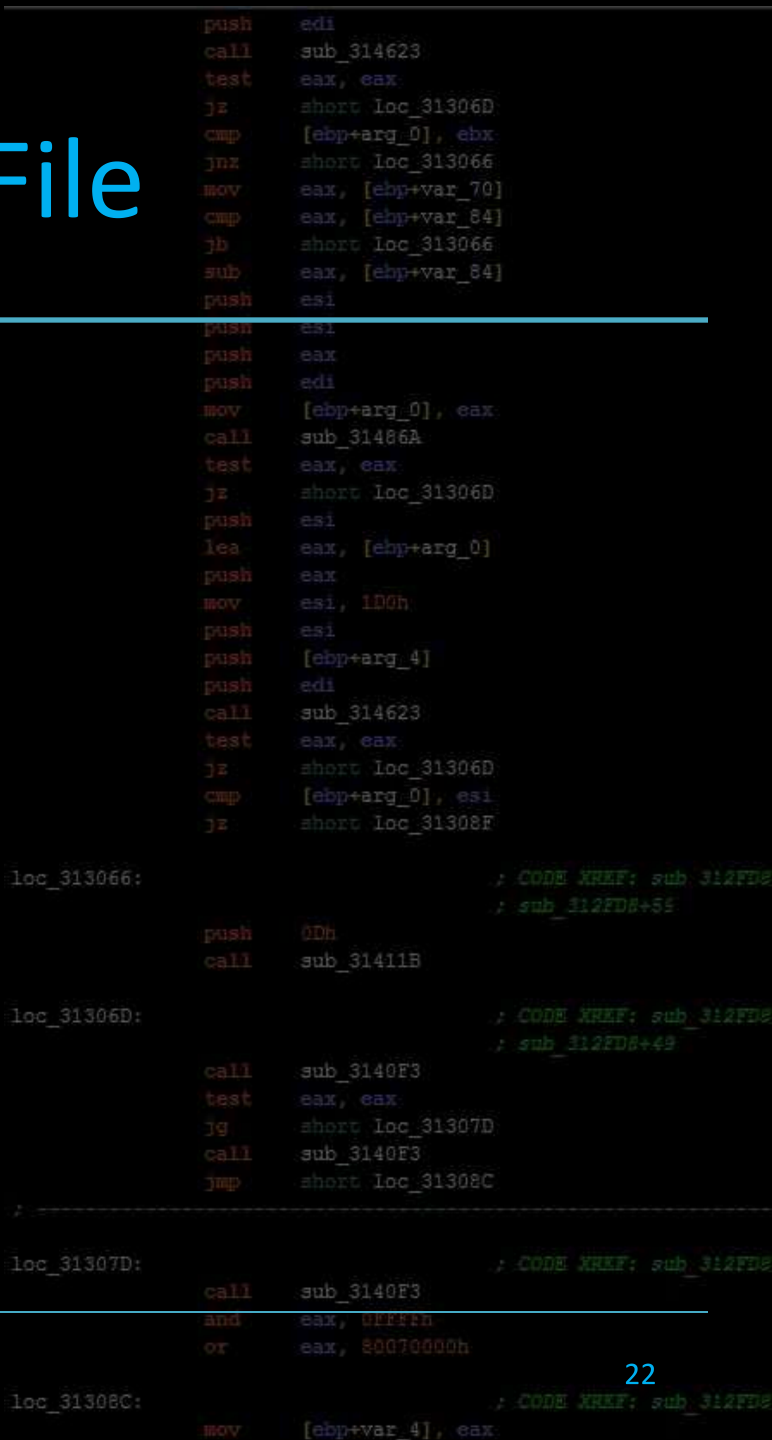
- gdb
  - python print ‘A’ *10



```
                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, 0FFFFh
                or      eax, 80070000h

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

GNU Debugger – Init File

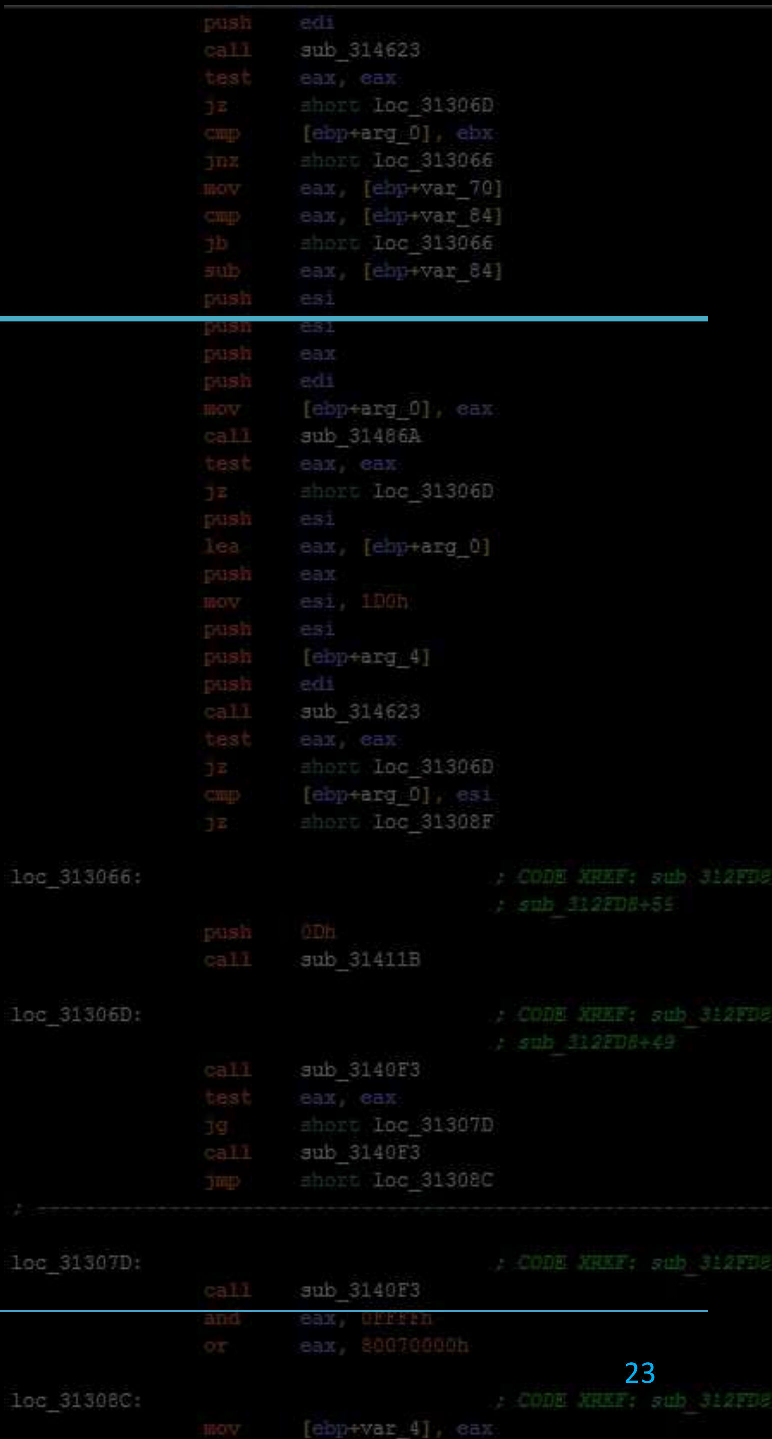
- mv special ~/.gdbinit
- gdb
  - help user
  - hexdump



```
                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, 0FFFFh
                or      eax, 80070000h

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

Tracing

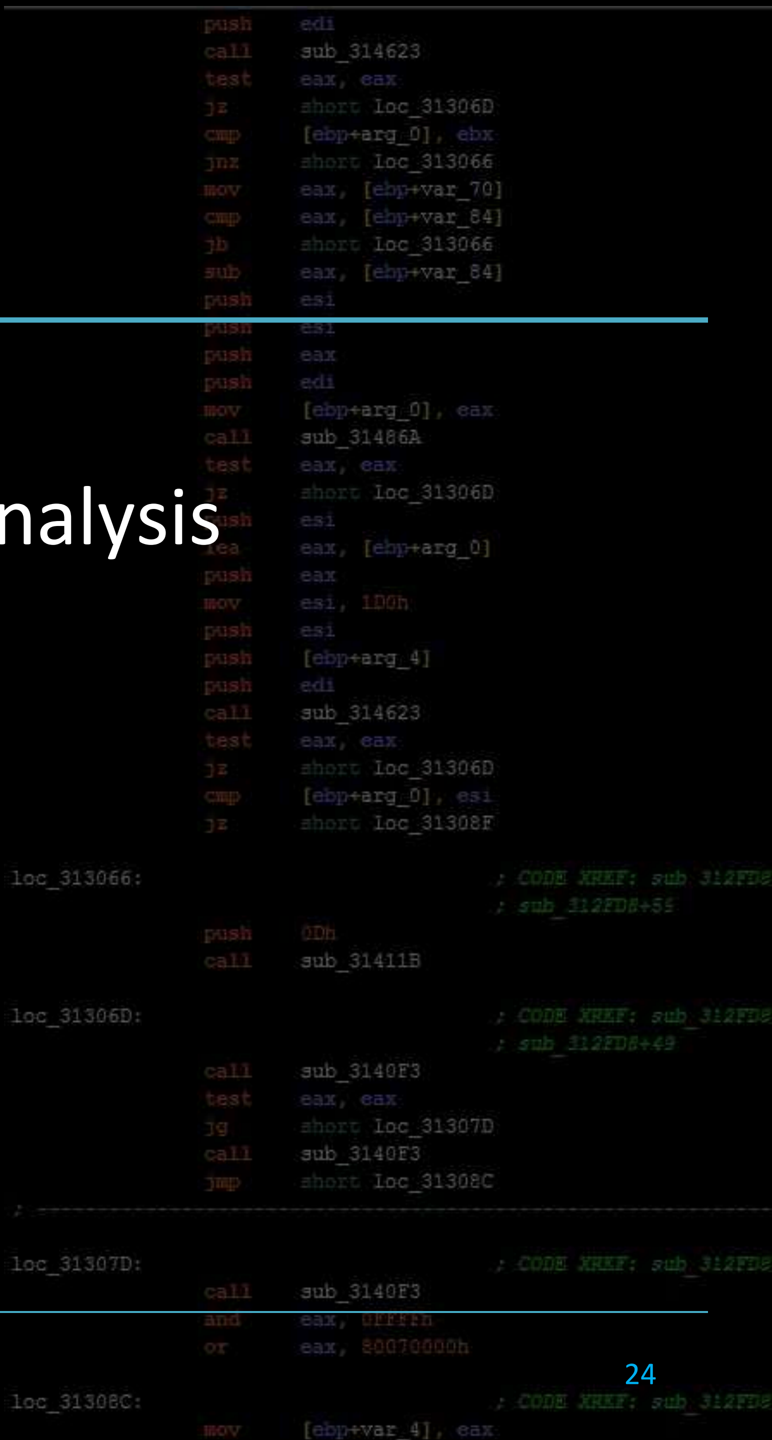
- ltrace, library calls
- strace, system calls



```
                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, 0FFFFh
                or      eax, 80070000h

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

Lecture Overview

1. Review of Last Lecture
2. Introduction to Dynamic Analysis
3. Tools!
4. Resources



```
                    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, 100h
                    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, 0FFFFh
                    or      eax, 80070000h

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

Additional Resources

- Gdb customizations
  - http://reverse.put.as/gdbinit/
  - https://github.com/dholm/voidwalker
  - http://stackoverflow.com/questions/209534/prettify-my-gdb
  - https://github.com/longld/peda
- Ring security
  - http://duartes.org/gustavo/blog/post/cpu-rings-privilege-and-protection/
  - http://www.amazon.com/The-Rootkit-Arsenal-Evasion-Corners/dp/1598220616