```
push
        edi
        sub_314623
test
        eax, eax
        short loc_31306D
        [ebp+arg 0], ebx
        short loc 313066
        eax, [ebp+var_70]
        eax, [ebp+var_84]
        short loc 313066
        eax, [ebp+var_84]
push
push
        es1
        [ebp+arg_0], eax
call
        sub 31486A
        eax, eax
        short loc 31306D
lea
        eax, [ebp+arg_0]
```

Analyzing Windows Pr

Malware Analysis CSCI 4976 - Fall 2015 Branden Clark

; CODE XREF: sub 312FD8 : sub 312FD8+55 0 Dh call sub_31411B ; CODE XREF: sub 312FD8

; sub 312FD8+49

; CODE XREF: sub 312FD8

call sub 3140F3 test eax, eax short loc 31307D call sub_3140F3 short loc_31308C THE

sub 314623

short loc_31306D [ebp+arg_0], esi short loc_31308F

eax, eax

loc_31307D:

loc_31308C:

test

sub_3140F3

eax, 80070000h ; CODE XREF: sub 312FD8

[ebp+var 4], eax

Side note

- 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 eax
- If you're ever bored...check out http://security.cs.rpi.edu/courses/malware-spring2013/
- Lot's of links, articles, forums, and tutorials are posted there

CrackMe's,

[ebp+arg 0], esi

short loc_31307D sub_3140F3 short loc_31308C

loc_31307D:

loc_31308C:

; CODE XREF: sub 312FD8

Windows

```
push
        edi
call
        sub_314623
test
        eax, eax
        short loc_31306D
        [ebp+arg 0], ebx
        short loc 313066
        eax, [ebp+var 70]
        eax, [ebp+var_84]
        short loc 313066
        eax, [ebp+var_84]
```

short loc 31306D

eax, [ebp+arg_0]

short loc_31306D

[ebp+arg 0], esi short loc 31308F

esi, 1D0h

[ebp+arg_4]

sub 314623

sub 31411B

sub 3140F3

eax, eax

881

lea

push

call

test

push

call

loc 313066:

loc_31306D:

loc_31307D:

loc_31308C:

- You've probably heard us throw arour terms
 - Handle
 - Process
 - Thread
 - Registry
 - DLL
 - Service
- And you're probably wondering what short loc 31308C mean

```
; CODE XREF: sub 312FD8
sub_3140F3
eax, 80070000h
                 ; CODE XREF: sub 312FD8
[ebp+var 4], eax
```

; CODE XREF: sub 312FD8

; CODE XREF: sub 312FD8

; sub 312FD8+55

; sub 312FD8+49

Windows API

```
push
        sub 314623
        short loc_31306D
        [ebp+arg_0], ebx
        short loc 313066
        eax, [ebp+var 70]
        eax, [ebp+var 84]
        short loc 313066
             [ebp+var 84]
```

eax, [ebp+arg_0]

[ebp+arg 4]

sub 31411B

- The standard way of interacting with Windows
 - Implemented through a broad set of DLLs... (Dynamic Link Library)
 - kernel32.dll & ntdll.dll
 - Low level, interact with kernel (think syscalls)
 - kernel32.dll and ntdll.dll are always loaded

loc 313066:

loc_31306D:

loc_31307D:

- wininet.dll / ws2_32.dll
 - Networking
- advapi32.dll
 - "Advanced" API
 - Services, processes, permissions, crypto, etc
- Lots more

```
; CODE XREF; sub 312FD
sub 3140F3
ebb+var 4].
```

short loc 31308C

/ sub 312FD8+55

; sub 312FD8+49

: CODE XREF: sub 312FD8

Processes

- A resource container
- Each process has its own
 - virtual address space
 - threads
 - Windows bookkeeping information
 - Usually stored in structs and linked list
 - Diagram by Jeremy Blackthorne

• Some API: CreateProcess, CreateProcessAsUser, EnumProcesses

```
short loc 313066
eax, [ebp+var 70]
eax, [ebp+var 84]
short loc 313066
eax, [ebp+var_84]
[ebp+arg 0], eax
short loc 31306D
eax, [ebp+arg_0]
esi, 1D0h
[ebp+arg_4]
                 / sub 312FD8+55
sub 31411B
```

push

sub 314623

short loc_31306D [ebp+arg_0], ebx

```
; CODE XREF: sub_312FD
call sub_3140F3
and eax, Offern
or eax, 80070000h
; CODE XREF: sub_312FD
mov [ebp+var 4], eax
```

; sub 312FD8+49

loc_31307D:

Threads

- Scheduled and executed by the OS
- Belong to a single process, and share its address space
- Have their own thread context
- Some API: CreateThread,
 CreateRemoteThread

```
short loc 31306D
                         [ebp+arg 0], esi
                         short loc 31308F
loc 313066:
                                          sub 312FD8+55
                push
                         sub 31411B
loc_31306D:
                                          ; CODE XREF: sub 312FD6
                                          ; sub 312FD8+49
                         sub 3140F3
                         eax, eax
                         short loc 31307D
                         sub 3140F3
                         short loc 31308C
loc_31307D:
                                          ; CODE XREF: sub 312FD8
                         sub 3140F3
```

push

edi sub_314623 eax_eax

short loc_31306D [ebp+arg_0], ebx

short loc_313066 eax, [ebp+var_70] eax, [ebp+var_84] short loc_313066 eax, [ebp+var_84]

Thread Context

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

[ebp+arg_4]

sub 314623

short loc_31306D [ebp+arg_0], esi short loc_31308F

eax, eax

loc 313066:

loc_31307D:

- Keeps track of the state of a thread
 - necessary when there are multiple threads on a system
 - State is defined by register values
- MSDN: "See CONTEXT in WinNT.h"

 Some API: GetThreadContext, SetThreadContext

DLLs

```
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
```

- All processes share loaded DLLs
 - -Saves space, only loaded into memory once
 - Must map into your own virtual address space (LoadLibrary)
 - Changes only affect your processes address space
- Exports functions for use by processes
- DllMain is automatically called when loaded

loc_31306D:

Some API: LoadLibrary, GetProcAddress

```
jg short loc_31307D
call sub_3140F3
jmp short loc_31308C

;

loc_31307D: ; CODE XREF: sub_312FD8
call sub_3140F3
and cax, Offer
or eax, 80070000h
```

[ebp+var 4], eax

; CODE XREF: sub 312FD

; sub 312FD8+49

Registry

- Big bookkeeping mess, "Organized" in hierarchical tree
- Used to store configuration information for everything
 - -Key: like a folder
 - Value entry: like a file
- Use regedit to navigate it
- Buzzwords: Autorun, Load Order Hijacking, Trusted DLLs
- Some API: RegOpenKey, RegSetValue, RegEnumKey

sub 312FD8+55

; sub 312FD8+49

; CODE XREF: sub 312FD6

push

edi sub_314623 eax_eax

short loc_31306D [ebp+arg_0], ebx

short loc_313066 eax, [ebp+var_70] eax, [ebp+var_84] short loc_313066 eax, [ebp+var_84]

short loc 31306D

sub 314623

short loc_31306D

[ebp+arg_0], esi short loc 31308F

short loc_31307D sub_3140F3 short loc_31308C

eax, eax

test

loc 313066:

loc_31307D:

Service

- Similar to a process
- Gets "Installed"
 - onto disk/registry
- Can interact with them via the manager (services.exe)
 - -Start, stop, suspend, schedule, autostart
- Some API: CreateService,
 OpenSCServiceManager,
 EnumDependentServices

```
short loc 313066
eax, [ebp+var_84]
[ebp+arg 0], eax
sub 31486A
short loc 31306D
eax, [ebp+arg_0]
short loc 31306D
[ebp+arg 0], esi
short loc 31308F
                 / sub 312FD8+55
sub 31411B
                 ; CODE XREF: sub 312FD8
                 7 sub 312FD8+49
sub 3140F3
short loc 31307D
sub 3140F3
short loc 31308C
                 ; CODE XREF: sub 312FD:
```

push

edi sub_314623 eax_eax

short loc_31306D [ebp+arg_0], ebx

short loc_313066 eax, [ebp+var_70] eax, [ebp+var_84]

loc_31307D:

sub_3140F3

Handle

push edi sub 314623 eax, eax short loc_31306D [ebp+arg_0], ebx short loc 313066 eax, [ebp+var 70] eax, [ebp+var 84] short loc 313066 eax, [ebp+var_84] [ebp+arg_0], eax

[ebp+arg_4]

- Abstract pointer to something
 - A specific process, file, registry key, service, etc.
 - Is sometimes an actual (memory) pointer

```
Some things that return Handles
                                     Some things that use open handles
OpenFile, CreateFile
                                     WriteFile, ReadFile
                                     OpenService, ControlService
OpenService, OpenSCManager
                                     RegCreateKey, RegReplaceKey
RegCreateKey
```

sub 3140F3 short loc 31308C ; CODE XREF: sub 312FD8

sub 3140F3

short loc 31307D

sub 312FD8+55

sub 312FD8+49

CODE XREF: sub 312FD8

loc_31307D:

Lecture sample

```
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
```

http://ark.rpis.ec/Malware/Lectures/
 03_Analyzing_Windows_Programs/

```
MOV
                         esi, 1D0h
                push
                         681
                 push
                         [ebp+arg_4]
                 push
                call
                         sub 314623
                test
                         eax, eax
                         short loc_31306D
                         [ebp+arg 0], esi
                         short loc_31308F
loc 313066:
                                          ; CODE XREF: sub 312FD8
                                          : sub 312FD8+55
                push
                         0 Dh
                call
                         sub_31411B
loc_31306D:
                                          ; CODE XREF: sub 312FD8
                                          ; sub 312FD8+49
                call
                         sub 3140F3
                 test
                         eax, eax
                         short loc 31307D
                call
                         sub_3140F3
                         short loc_31308C
                 ij
                                          ; CODE XREF: sub 312FD8
loc_31307D:
```

sub_3140F3

[ebp+var 4], eax

eax, [ebp+arg_0]

COM (Component Object

- Code sharing / interoperability giant by MS
- COM servers offer up implementation
 - Identified by GUIDs, CLSIDs, and IIDs
- COM Clients use the interface to exec the implementation
- Similar to the #include/DLL method but...

 - Client doesn't need to know who is implementing it or where

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

loc_31307D:

all sub_3140F3

or eax, 8007

13

Lecture sample

• PMA Lab 07-02

```
push
                         edi
                call
                         sub_314623
                test
                         eax, eax
                         short loc_31306D
                CID
                         [ebp+arg_0], ebx
                         short loc_313066
                         eax, [ebp+var_70]
                         eax, [ebp+var_84]
                         short loc 313066
                         eax, [ebp+var_84]
                push
                         esi
                         C51
                push
                         edi
                         [ebp+arg_0], eax
                MOV
                call
                         sub_31486A
                test
                         eax, eax
                         short loc_31306D
                push
                         esi
                lea
                         eax, [ebp+arg_0]
                         cax
                         esi, 1D0h
                push
                         681
                push
                         [ebp+arg_4]
                push
                         sub_314623
                test
                         eax, eax
                         short loc_31306D
                         [ebp+arg 0], esi
                         short loc_31308F
loc_313066:
                                          ; CODE XREF: sub 312FD8
                                          ; sub 312FD8+55
                         0 Dh
                push
                call
                         sub_31411B
loc_31306D:
                                          ; CODE XREF: sub 312FD8
                                          ; sub 312FD8+49
                call
                         sub_3140F3
                test
                         eax, eax
                         short loc 31307D
                call
                         sub_3140F3
                         short loc_31308C
                 THE C
loc_31307D:
                                          ; CODE XREF: sub 312FD8
                         sub_3140F3
                         eax, 80070000h
```

[ebp+var 4], eax

Lab this Friday

sub_314623 test eax, eax short loc_31306D [ebp+arg 0], ebx short loc 313066 eax, [ebp+var 70] eax, [ebp+var_84] short loc 313066 eax, [ebp+var_84]

short loc 31306D

eax, [ebp+arg_0]

push

lea

edi

- Jeremy Blackthorne from MIT Linc Laboratory is coming to speak
- No lab if he's still coming

```
esi, 1D0h
                 push
                 push
                         [ebp+arg_4]
                 call
                         sub 314623
                test
                         eax, eax
                         short loc_31306D
                         [ebp+arg 0], esi
                         short loc 31308F
loc 313066:
                                          ; CODE XREF: sub 312FD8
                                          : sub 312FD8+55
                push
                call
                         sub 31411B
loc_31306D:
                                          ; CODE XREF: sub 312FD8
                                          ; sub 312FD8+49
                call
                         sub 3140F3
                         eax, eax
                         short loc 31307D
                         sub_3140F3
                         short loc_31308C
loc_31307D:
                                          ; CODE XREF: sub 312FD8
                         sub_3140F3
                         eax, 80070000h
```

Fairgame & CSAW

- Fairgame solutions this Friday!!
 - fairgame.rpis.ec
- CSAW CTF this weekend!

```
[ebp+arg 0], ebx
                         short loc 313066
                         eax, [ebp+var_70]
                         eax, [ebp+var_84]
                         short loc 313066
                         eax, [ebp+var_84]
                         C51
                         [ebp+arg_0], eax
                         sub 31486A
                 test
                         eax, eax
                         short loc 31306D
                 lea
                         eax, [ebp+arg_0]
                         esi, 1D0h
                 push
                         881
                 push
                         [ebp+arg_4]
                 push
                call
                         sub 314623
                test
                         eax, eax
                         short loc_31306D
                         [ebp+arg 0], esi
                         short loc_31308F
loc 313066:
                                          ; CODE XREF: sub 312FD8
                                          : sub 312FD8+55
                push
                         0 Dh
                call
                         sub 31411B
loc_31306D:
                                          ; CODE XREF: sub 312FD8
                                          ; sub 312FD8+49
                call
                         sub 3140F3
                         eax, eax
                         short loc 31307D
                call
                         sub_3140F3
                         short loc_31308C
loc_31307D:
                                          ; CODE XREF: sub 312FD8
                         sub_3140F3
                         eax, 80070000h
                                                       16
loc_31308C:
                                          ; CODE XREF: sub 312FD8
```

push

call

test

edi

sub_314623

short loc_31306D

eax, eax

References

```
push
        edi
        sub_314623
test
        eax, eax
        short loc_31306D
        [ebp+arg_0], ebx
        short loc 313066
        eax, [ebp+var_70]
        eax, [ebp+var_84]
        short loc 313066
        eax, [ebp+var_84]
```

1. Sikorski, Michael, and Andrew Honig. Practical Malware Analysis the Hands-on Guide to Dissecting Malicious Software. San Francisco: No Starch, 2012. Print. short loc_31306D

```
cax
                         esi, 1D0h
                push
                         881
                push
                         [ebp+arg_4]
                 push
                call
                         sub 314623
                test
                         eax, eax
                         short loc_31306D
                         [ebp+arg 0], esi
                         short loc 31308F
loc 313066:
                                          ; CODE XREF: sub 312FD8
                                          : sub 312FD8+55
                push
                         0 Dh
                call
                         sub_31411B
loc_31306D:
                                          ; CODE XREF: sub 312FD8
                                          ; sub 312FD8+49
                call
                         sub 3140F3
                 test
                         eax, eax
                         short loc 31307D
                call
                         sub_3140F3
                         short loc_31308C
                                          ; CODE XREF: sub 312FD8
loc_31307D:
                call
                         sub_3140F3
```

eax, Offffh eax, 80070000h

push

lea

esi

eax, [ebp+arg_0]

loc_31308C: