```
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]
                                                                                  push
                                                                                         es1
                                                                                         [ebp+arg_0], eax
                                                                                         sub 31486A
                                                                                         eax, eax
             Forensics Subversion
                                                                                         short loc_31306D
                            Covert Channels
                                                                                         [ebp+arg_4]
                                                                                         sub 314623
                                                                                         eax, eax
                                                                                         short loc_31306D
                                                                                         [ebp+arg_0], esi
                                                                                         short loc 31308F
                                                                                                      ; CODE XREF: sub 312FD8
                                      Aaron Sedlacek
                                                                                                      : sub 312FD8+55
                                                                                  push
                                                                                         sub 31411B
                                       Branden Clark
                                                                                                      ; CODE XREF: sub 312FD8
                                                                                                      ; sub 312FD8+49
                                                                                  call
                                                                                         sub 3140F3
                                                                                  test
                                                                                         eax, eax
                                                                                         short loc 31307D
                                                                                  call
                                                                                         sub_3140F3
                                                                                         short loc_31308C
                                                                                  THE
                                                                     loc_31307D:
                                                                                                      ; CODE XREF: sub 312FD8
                                                                                         sub_3140F3
                                                                                         eax, 80070000h
                                         Forensics Subversion and Covert
Malware - 12/01/2015
                                                    Channels
                                                                     loc_31308C:
                                                                                                      ; CODE XREF: sub 312FD8
                                                                                         [ebp+var 4], eax
```

push

edi

Agenda

- Access Token Manipulation
- IRP Handling and Filter Drivers
- Forensics Countermeasures
- Covert Channels
- Out-Of-Band

```
eax, [ebp+var_84]
                         short loc 313066
                         eax, [ebp+var_84]
                         [ebp+arg_0], eax
                 call
                         sub 31486A
                 test
                         eax, eax
                         short loc 31306D
                 lea
                         eax, [ebp+arg_0]
                         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
                         sub 3140F3
                         eax, eax
                         short loc 31307D
                         sub_3140F3
                         short loc_31308C
loc_31307D:
                                          ; CODE XREF: sub 312FD8
                call
                         sub_3140F3
                         eax, 80070000h
loc_31308C:
                                          ; CODE XREF: sub 312FD8
```

[ebp+var 4], eax

push

call

test

edi

sub_314623

short loc_31306D [ebp+arg_0], ebx

short loc_313066 eax, [ebp+var 70]

eax, eax

Windows Authorization

test eax, eax

jz short loc_31306D

[ep+arg_0], ebx

TOK [C17]S

[ebp+arg_0], eax

eax, [ebp+arg_0]

- User logs on
 - -OS generates a "Primary" access token based on:
 - The user's account
 - The user's security groups
 - Privileges granted by the administrator
- All processes launched on behalf of the user will be given a copy of this token

[ebp+var 4], eax

Windows Authorization

- test cax, cax

 jz short loc_31306D

 [eip+arg_0], ebx

 [eip+arg_0], ebx

 cax, [ebp+arg_0], ebx

 jb short loc_313066

 sub cax, [ebp+var_84]

 push esi

 push esi
- The OS uses the access token to authorize a process when it attempts to
 - Perform an action that requires special privileges
 - Access a securable object
- Securable Object Any object that has a security descriptor
- Security Descriptor
 - Determines an object's owner, primary security
 group, and discretionary access control list (DACL)

; code XREF: sub_312FD: call sub_3140F3

Windows Authorization



- List of access control entries (ACES)
 - each ACE identifies a user, or security group, and the operations that they're allowed to perform on an object
- 34 different privileges
 - SeDebugPrivilege
 - SeShutdownPrivilege
 - SeLoadDriverPrivilege

```
- . . .
```

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

[ebp+var 4], eax

: CODE XREF: sub 312FD

sub 314623

Modifying Tokens

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]

eax, eax

esi, 1D0h

[ebp+arg_4]

sub 314623

eax, eax

test

short loc 31306D

eax, [ebp+arg_0]

short loc 31306D [ebp+arg_0], esi

- Two ways to modify access tokens: a. Windows API:
 - OpenProcessToken()
 - AdjustTokenPrivileges()
 - AdjustTokenGroups()

b.DKOM

 EPROCESS struct contains a pointer to the token object

```
call
                         sub 31411B
loc_31306D:
                                          ; CODE XREF: sub 312FD8
                                          ; sub 312FD8+49
                         sub 3140F3
                         eax, eax
                         short loc 31307D
                         sub 3140F3
                         short loc 31308C
loc_31307D:
                                          ; CODE XREF: sub 312FD8
                         sub_3140F3
```

eax, 80070000h

[ebp+var 4], eax

: CODE XREF: sub 312FD8

Agenda

- Access Token Manipulation
- IRP Handling and Filter Drivers
- Forensics Countermeasures
- Covert Channels
- Out-Of-Band

```
eax, [ebp+var 84]
                         short loc 313066
                         eax, [ebp+var_84]
                         [ebp+arg_0], eax
                 call
                         sub 31486A
                 test
                         eax, eax
                         short loc 31306D
                 lea
                         eax, [ebp+arg_0]
                         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
                         sub 3140F3
                         eax, eax
                         short loc 31307D
                         sub_3140F3
                         short loc_31308C
loc_31307D:
                                          ; CODE XREF: sub 312FD8
                call
                         sub_3140F3
                         eax, 80070000h
loc_31308C:
                                          ; CODE XREF: sub 312FD8
```

[ebp+var 4], eax

push

call

test

edi

sub_314623

short loc_31306D [ebp+arg_0], ebx

short loc_313066 eax, [ebp+var 70]

eax, eax

Drivers and Device Stack

- Drivers add themselves to the chain of drivers
 - Create a device
 - loCreateDevice()
 - Insert device into group of devices
 - IoAttachDevice()

```
eax, [ebp+arg_0]
        esi, 1D0h
        [ebp+arg_4]
call
        sub 314623
test
        eax, eax
        short loc_31306D
        [ebp+arg 0], esi
        short loc 31308F
                         ; CODE XREF: sub 312FD8
                         ; sub 312FD8+55
push
call
        sub 31411B
                         ; CODE XREF: sub 312FD8
                         ; sub 312FD8+49
        sub 3140F3
```

eax, eax

sub_3140F3

short loc_31307D sub_3140F3 short loc_31308C

[ebp+var 4], eax

push

loc 313066:

loc_31306D:

loc_31307D:

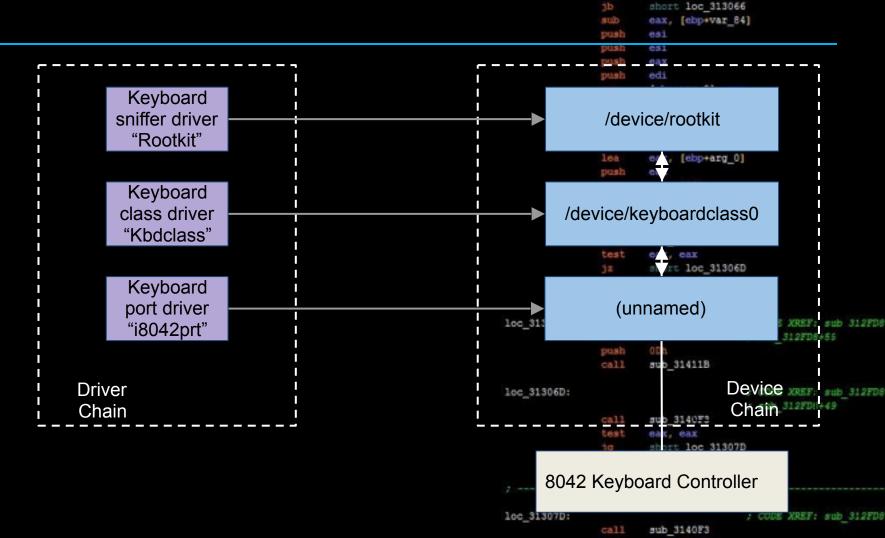
edi sub 314623

short loc_31306D [ebp+arg_0], ebx

short loc_313066 eax, [ebp+var_84]

; CODE XREF: sub 312FD8

Drivers and Device Stack



push

sub 314623

short loc 31306D

[ebp+arg_0], ebx short loc 313066

eax, 80070000h

[ebp+var 4], eax

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]

push

 Read request made to read keystroke

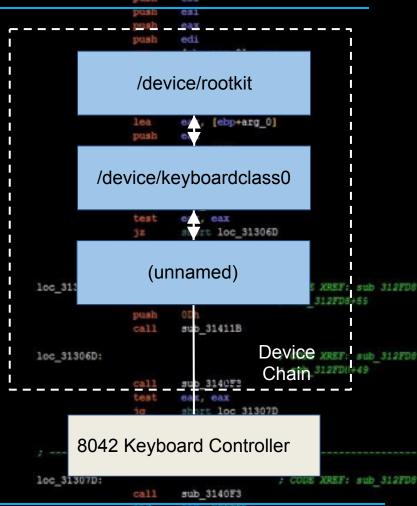
IRP

IRP Header

IO STACK LOCATION (1)

IO STACK LOCATION (2)

IO STACK LOCATION (3)



eax, 80070000h

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

• IRP Makes its way down the device stack

IRP

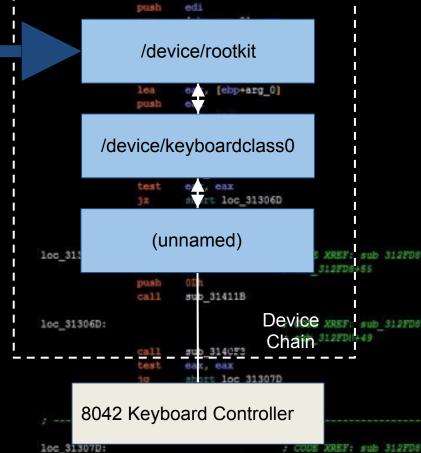
IRP Header

Stack Location Pointer

IO_STACK_LOCATION (1)

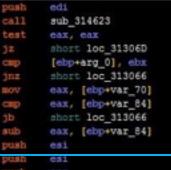
IO STACK LOCATION (2)

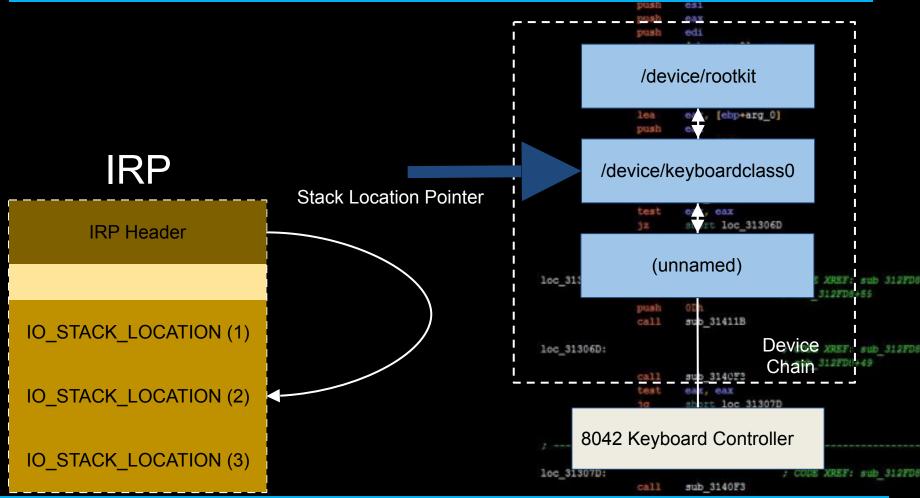
IO_STACK_LOCATION(3)



sub 3140F3

eax, 80070000h





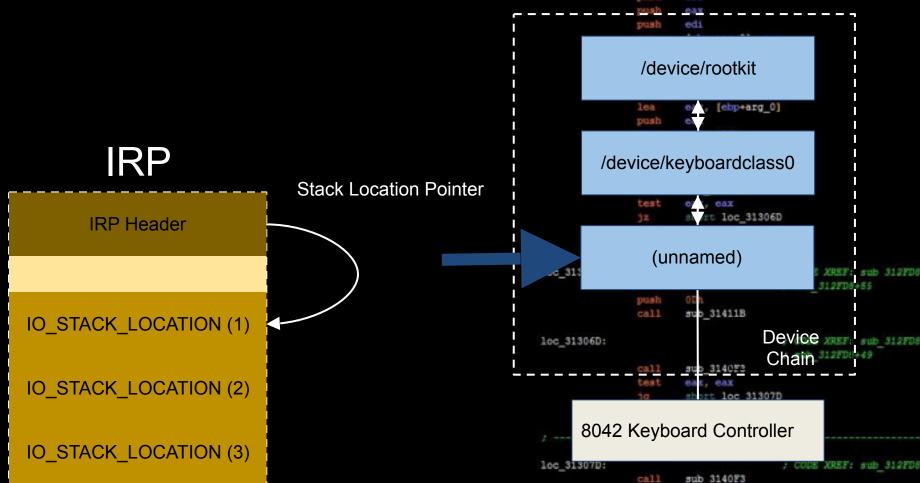
eax, 80070000h

[ebp+var 4], eax

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

eax, 80070000h

[ebp+var 4], eax



Malware - 12/01/2015

IRP Forwarding and Completion

- IoCallDriver(gNextDevice, theIRP)
 - -Calls the next device in the stack
 - IoSkipCurrentStackLocation(theIRP)
 - —If this function is called, IoCallDriver() calls the next driver with the IO_STACK_LOCATION of the current driver
 - This trick allows the lower-level driver to use any arguments or completion routines supplied
 - IoCompleteRequest(theIRP)
 - Returns the given IRP to the I/O Control Manager

loc_31307D:

call sub_3140F3
and eax, Offfin
or eax, 80070000h

[ebp+arg 0], eax

Completion Routines

```
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
```

short loc 31306D

sub 314623

- IoSetCompletionRoutine(theIRP, co outine, ...)
 - Registers a routine to be executed when the next lowest driver has completed the IRP

```
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
                         sub 3140F3
                         eax, eax
                         short loc 31307D
                         sub 3140F3
                         short loc 31308C
loc_31307D:
                                          ; CODE XREF: sub 312FD8
                         sub_3140F3
```

eax, 80070000h

[ebp+var 4], eax

; CODE XREF: sub 312FD8

Agenda

- Access Token Manipulation
- IRP Handling and Filter Drivers
- Forensics Countermeasures
- Covert Channels
- Out-Of-Band

```
eax, [ebp+var_84]
                         short loc 313066
                         eax, [ebp+var_84]
                         [ebp+arg_0], eax
                 call
                         sub 31486A
                 test
                         eax, eax
                         short loc 31306D
                 lea
                         eax, [ebp+arg_0]
                         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
                         sub 3140F3
                         eax, eax
                         short loc 31307D
                         sub_3140F3
                         short loc_31308C
loc_31307D:
                                          ; CODE XREF: sub 312FD8
                call
                         sub_3140F3
                         eax, 80070000h
loc_31308C:
                                          ; CODE XREF: sub 312FD8
```

push

call

test

edi

sub_314623

short loc_31306D [ebp+arg_0], ebx

short loc 313066 eax, [ebp+var 70]

eax, eax

short loc 31306D [ebp+arg_0], ebx Hooking Countermeasures short loc 313066 eax, [ebp+var 84]

 There are certain ranges of addresses that most call table entries should contain

		push	[ebp+arg_4]	
Call Table	Red Flag Condition	call test jz	sub_314623 eax, eax short loc_3130	6D
IAT	An entry lies outside of its designated DLL address rang	e ^{1z}	[ebp+arg_0], es short loc_3130	
IDT	The address of the 0x2E handler lies outside of ntoskrnl	.exe	0Dh	; sub_312FD8+55
MSR	The contents of IA32_SYSENTER_EIP lie outside of nto	skrnl	.exe	; CODE XREF: su
SSDT	Pointers to the Nt*() routines lie outside of ntoskrnl.exe	call test	sub_3140F3 eax, eax short loc 3130	70
IRP	The addresses of dispatch routines lie outside of the driv	ver's	address rang	ge

; CODE XREF: sub 312FD8

loc_31307D:

edi sub 314623

eax, [ebp+arg_0]

esi, 1D0h

sub 3140F3

[ebp+var 4], eax

Patching Countermeasure

- Byte Scanning
 - Scan the first few and last few lines of for a telltale jump instruction
 - Checksum Calculations

```
short loc 313066
                        eax, [ebp+var_84]
                        [ebp+arg_0], eax
                call
                        sub 31486A
                       each routine
                        eax, [ebp+arg_0]
                        esi, 1D0h
                        [ebp+arg_4]
                        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
                        sub 3140F3
                        eax, eax
                        short loc 31307D
                        sub 3140F3
                        short loc 31308C
loc_31307D:
                                        ; CODE XREF: sub 312FD8
                        sub_3140F3
                        eax, 80070000h
```

[ebp+var 4], eax

; CODE XREF: sub 312FD8

push

edi sub 314623

short loc_31306D [ebp+arg_0], ebx

DKOM Countermeasures

Cross-View Detection

- -There are usually multiple ways to collect the same information
- —Some may be under the control of the rootkit, but not all
- Typically, Cross-view Detection uses high-level and low-level methods

```
push oDh
call sub_314118

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
```

[ebp+var 4], eax

push

sub 314623

short loc_31306D [ebp+arg_0], ebx

short loc 313066

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

[ebp+arg_0], eax

DKOM Countermeasures

- Cross-View Enumerations
 - High-Level Enumeration
 - CreateToolHelp32Snapshot()
 - PID Bruteforce
 - Low-level Enumeration
 - Walk the Process list
 - Walk the Thread list
 - Rootkit Detection Tools
 - RootkitRevealer (Sysinternals tool)
 - HackerDefender, Vanquish, AFX, Blacklight, Volatility

loc 313066:

```
[ebp+arg_0], eax
        short loc 31306D
             [ebp+arg_0]
        esi, 1D0h
        [ebp+arg_4]
        sub 314623
test
        eax, eax
        short loc 31306D
        [ebp+arg 0], esi
        short loc 31308F
        sub 31411B
        sub 3140F3
```

push

edi

short loc 31306D

[ebp+arg 0], ebx

[ebp+var 84]

Counter-Countermeasure

- Authors have three options:
 - 1. Starve the Opposition
 - a. The Grugq's 'Data Contraception' philosophy
 - 2. Stepping out of the two-ring model
 - a.Rogue Hypervisors (Ring -1)
 - b.System Management Mode Rootkits (Ring -2)
 - c.Lights-Out Management Facilities (Ring -3)
 - d.Bootkits
 - e.Onboard Flash Storage and Circuit-Level

loc_31307D:

3. Covert Channels

[ebp+arg_0], eax

eax, [ebp+arg_0]

Agenda

- Access Token Manipulation
- IRP Handling and Filter Drivers
- Forensics Countermeasures
- Covert Channels
- Out-Of-Band

```
eax, [ebp+var_84]
                         [ebp+arg_0], eax
                 call
                         sub 31486A
                 test
                         eax, eax
                         short loc 31306D
                 lea
                         eax, [ebp+arg_0]
                         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
                         sub 3140F3
                         eax, eax
                         short loc 31307D
                         sub_3140F3
                         short loc_31308C
loc_31307D:
                                          ; CODE XREF: sub 312FD8
                call
                         sub_3140F3
                         eax, 80070000h
loc_31308C:
                                          ; CODE XREF: sub 312FD8
                         [ebp+var 4], eax
```

push

call

test

edi

sub_314623

short loc_31306D [ebp+arg_0], ebx

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

eax, eax

Covert Channels

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

short loc 31306D

[ebp+arg_0], esi short loc 31308F

sub_3140F3 short loc 31308C

• Why?

- If an analyst can't find a rootkit on the victim machine, they can still watch all network traffic
- What good is a rootkit if you can't talk to it or use it to retrieve data?

How?

- Covert Channel A network connection that disguises its byte stream as normal traffic
- Doesn't need to be fancy, just needs to be unanticipated

```
; CODE XREF: sub_312FD0
```

; sub 312FD8+49

loc_31307D:

Common Malware Chann

- 1980's Listen on an obscure port
 - Firewalls
- Late 1980's IRC
 - Malware connects back to the attacker, does not leave an obscure port hanging around
 - -Now stands out like a sore thumb
- 1990's Peer-to-peer
 - -P2P protocols aren't necessarily inconspicuous

[ebp+arg_0], eax

short loc 31306D

eax, [ebp+arg_0]

[ebp+arg_0], esi short loc 31308F

sub 312FD8+55

Common Malware Chann

- Now HTTP
 - Web traffic is everywhere, supports a higher-level protocols
 - -with TCP, can be considered reliable
 - Malware will stick to strictly legitimate
 URLs
 - Malware has started using things like Google Groups as C2 servers
 - Difficult to filter, lends itself to steganography

[ebp+arg_0], eax

Common Malware Chann

- HTTP provides an endless variety of place to hide data
 - Particularly, mixed into the content of legitimate HTTP requests/replies
 - Using HTTPS can be useful, but it that some data is being hidden
 - Rootkits will commonly encrypt segments of the HTTP content instead

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

/ CODE XREF: sub_312FD
call sub_3140F3
```

eax, [ebp+arg_0]

ebp+arg 4

loc_31307D:

- 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
- Stash our data in the nooks and crannies of an otherwise mundane protocol
 - The goal is to bore the forensic analyst death
 - Covert data stream must:
 - -Make it past the perimeter firewall
 - -Blend in with existing traffic 100_313

```
eax, eax
        short loc 31306D
        [ebp+arg 0], esi
        short loc 31308F
                         sub 312FD8+55
call
        sub 31411B
                         ; CODE XREF: sub 312FD8
                         7 sub 312FD8+49
        sub 3140F3
        eax, eax
        short loc 31307D
        sub 3140F3
        short loc 31308C
                         : CODE XREF: sub 312FU
        sub 3140F3
```

ebp+var 41.

loc_31307D:

HTTP

- Reliable and flexible, but noisy
- Typically used for Data exfiltration

DNS

- Not suited for data transfer
- Typically used for WAN-based C2
- Ubiquitous

ICMP

- Often blocked at the perimeter
- Typically used for LAN-based C2
- Often Ignored by forensic analysts

```
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]
        [ebp+arg_0], eax
        sub 31486A
        short loc 31306D
lea
        eax, [ebp+arg_0]
        esi, 1D0h
push
push
        [ebp+arg_4]
        sub 314623
test
        eax, eax
        short loc 31306D
        [ebp+arg 0], esi
        short loc 31308F
                          sub 312FD8+55
push
call
        sub 31411B
                          : CODE XREF: sub 312FD6
                         ; sub 312FD8+49
        sub 3140F3
test
        eax, eax
        short loc 31307D
call
        sub 3140F3
        short loc 31308C
```

push

call

edi

sub 314623

```
loc_31307D:
```

loc 313066:

```
; CODE XREF: sub_312FD8
```

[ebp+var 4], eax

- ICMP example
 - RC3 CTF Forensics 400

```
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
                         esi
                         C51
                push
                MOV
                         [ebp+arg_0], eax
                call
                         sub_31486A
                test
                         eax, eax
                         short loc_31306D
                push
                         esi
                lea
                         eax, [ebp+arg_0]
                         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
                         0 Dh
                push
                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
                THE
loc_31307D:
                                          ; CODE XREF: sub 312FD8
                         sub_3140F3
                         eax, 80070000h
```

push

call

edi

sub_314623

- ICMP example
 - RC3 CTF Forensics 400
 - scapy example
 - Kit kappa icmp transfer

```
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]
                push
                         C51
                push
                         [ebp+arg_0], eax
                call
                         sub 31486A
                         eax, eax
                         short loc 31306D
                push
                         681
                lea
                         eax, [ebp+arg_0]
                         esi, 1D0h
                push
                         681
                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
                         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
                THE
loc_31307D:
                                          ; CODE XREF: sub 312FD8
                         sub_3140F3
                         eax, 80070000h
                                                      30
loc_31308C:
                                          ; CODE XREF: sub 312FD8
```

[ebp+var 4], eax

push

edi

IPv4

test	eax, eax
jz	short loc_31306D
CMD	[ebp+arg_0], ebx
jnz	short loc_313066
MOV	eax, [ebp+var_70]
CMD	eax, [ebp+var_64]
)b	short loc_313066
sub	eax, [ebp+var_84]
push	
push	es1
push	eax
push	edi
MOV	[ebp+arg_0], eax
call	sub_31486A
test	eax, eax
JZ	short loc_31306D
push	esi

push

edi sub 314623

IPv4 Header Format																																			
Octet				0)				1									2									3								
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31			
0	Version IHL DSCP											CP			Е	CN		Total Length																	
32	Identification														Flags Fragment Offset																				
64			Tir	me T	o Liv	e						Pro	tocol				Header Checksum																		
96															Soi	urce	P Ad	dres	s																
128															esti	natio	n IP /	Addre	ess																
160															Opt	tions	(if IH	. > 5	5)																

- IPv4
 - ID field 16 bpp (bits per packet)
 - -DF flag 1 bpp
 - -TTL 8 bpp
 - many more

```
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
                         C51
                push
                         [ebp+arg_0], eax
                call
                         sub 31486A
                         eax, eax
                         short loc 31306D
                 push
                         681
                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
                 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

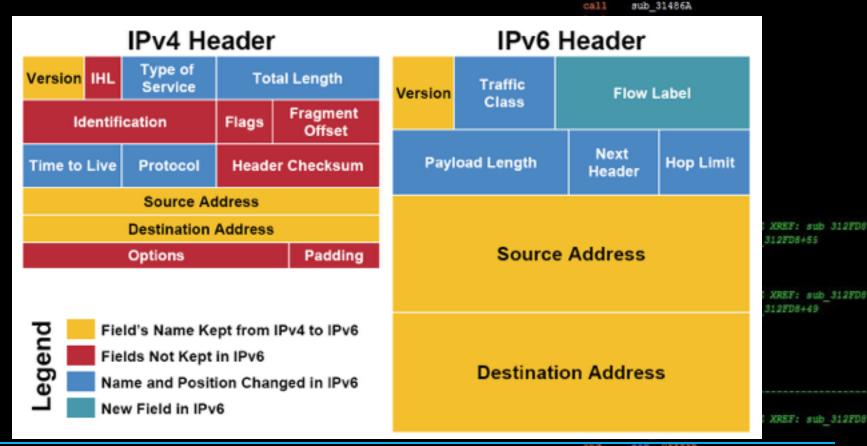
; CODE XREF: sub 312FD8

push

edi sub_314623

push edi call sub 314623 eax, eax test short loc_31306D [ebp+arg 0], ebx short loc 313066 eax, [ebp+var 70] eax, [ebp+var_64] short loc 313066 sub eax, [ebp+var_84] push eax push MOV [ebp+arg_0], eax

IPv6



[ebp+var 4], eax

- IPv6
 - Much simpler...but...

```
CID
                         [ebp+arg_0], ebx
                         short loc 313066
                         eax, [ebp+var_70]
                         eax, [ebp+var_84]
                         short loc 313066
                 sub
                         eax, [ebp+var_84]
                push
                         esi
                         C51
                push
                MOV
                         [ebp+arg_0], eax
                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, Offffn
                         eax, 80070000h
                                                      34
```

push

test

edi sub_314623

eax, eax

short loc_31306D

- IPv6
 - Much simpler…but…
 - Has extension headers!

		mov esi, 100
Extension Header	Type	Description
Hop-by-Hop Options	0	Options that need to be examined by all devices on the path.
Destination Options (before routing header)	60	Options that need to be examined only by the destination of the packet.
Routing	43	Methods to specify the route for a datagram (used with Mobile IPv6).
Fragment	44	Contains parameters for fragmentation of datagrams.
Authentication Header (AH)	51	Contains information used to verify the authenticity of most parts of the packet.
Encapsulating Security Payload (ESP)	50	Carries encrypted data for secure communication.
Destination Options (before upper- layer header)	60	Options that need to be examined only by the destination of the packet.
Mobility (currently without upper- layer header)	135	Parameters used with Mobile IPv6.

```
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]
        [ebp+arg_0], eax
call
        sub 31486A
        eax, eax
        short loc 31306D
        eax, [ebp+arg_0]
```

- IPv6
 - Much simpler...but...
 - Has extension headers!

	jz	short loc_31306D
	CHIP	[ebp+arg_0], ebx
	inz	short loc_313066
	mov	eax, [ebp+var_70]
	cmp	eax, [ebp+var_84]
	jb.	short loc_313066
	sub	eax, [ebp+var_84]
	push	esi
Ī	push	es1
	push	eax
	push	edi
	mov	[ebp+arg_0], eax
	call	sub_31486A
	test	eax, eax
)Z	short loc_31306D
	push	esi
	lea	eax, [ebp+arg_0]
	push	eax
	MOV	esi, 1D0h
	push	esi

Hop-by-Hop Options and Destination Options extension header format

trop by trop options and bestination options enterision neader format																																		
Offsets	Octet					0								1				2									3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	1	0 11	12	13	14	15	16	17	18	19	20	21	22	23	3 24	25	26	27	28	3 29	30	31	
0	0			Ν	ext I	Head	<i>ier</i>				Hdr Ext Len							Options and Padding																
4	32															Opti	ns a	nd P	addir	g														
8	64													Onti	ional	mai	e Opt	tinno	and	Dade	lina													
12	96													Opu	uriai	. 11101	e opi	CIUID	ariu	rauc	mig.													

Next Header (8 bits)

Specifies the type of the next header.

Hdr Ext Len (8 bits)

Length of this header in 8-octet units, not including the first 8 octets.

Options (variable)

Contains one or more options, and optional padding fields to align options and to make the total header length a multiple of 8 octets. Options are TLV-coded.

Protocol Tunneling

- IPv6
 - Much simpler…but…
 - Has extension headers!
 - The extension headers add a LOT of value
 each packet
 - Plenty of opportunities for crazy channels

```
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]
        [ebp+arg_0], eax
call
        sub 31486A
        short loc 31306D
        eax, [ebp+arg_0]
        esi, 1D0h
```

push

edi

of Variability to

```
test eax, eax
jz short loc_31306D
cmp [ebp+arg_0], esi
```

; CODE XREF: sub 3121 ; sub 312FDS+55 push ODh call sub 31411B

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

eax, 80070000h

loc_31306D:

loc_31307D:

```
; CODE XREF: sub_312FD8
```

; CODE XREF: sub 312FD8

Protocol Tunneling

- Finding covert channels
 - Read the RFCs
 - -...that's about it

```
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
                         C51
                push
                         [ebp+arg_0], eax
                call
                         sub 31486A
                 test
                         eax, eax
                         short loc 31306D
                 push
                         esi
                 lea
                         eax, [ebp+arg_0]
                         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
                         eax, eax
                         short loc 31307D
                call
                         sub_3140F3
                         short loc_31308C
                 THE C
                                          ; CODE XREF: sub 312FD8
loc_31307D:
                         sub_3140F3
                         eax, 80070000h
```

push

call

edi

sub_314623

Tunneling Data From a

Rootkit

[ebp+arg_0], eax

[ebp+arg 4]

- User-mode
 - Windows Sockets 2 (Winsock) API
 - Simple, well documented, but easy to detect
- Kernel-mode
 - Winsock Kernel (WSK) API
 - More obscure, complicated, protocol-specific details
 - Network Driver Interface Specification (NDIS)
 - High host-based stealth, low network-based stealth

short loc 31308C

Passive Covert Channels

- You're still generating new packets
 - -That may be enough for an analyst to find a rootkit
- Instead, make subtle modifications to existing packets.
 - -Steganography at the packet level
- Extra STEALTH but additional EFF
 - Requires a foothold on a device that has access to all network traffic (like a router)

short loc_31307D sub_3140F3

sub_3140F3

short loc 31306D

short loc 31306D

Agenda

- Access Token Manipulation
- IRP Handling and Filter Drivers
- Forensics Countermeasures
- Covert Channels
- Out-Of-Band

Malware - 12/01/2015

```
short loc 313066
                         eax, [ebp+var_84]
                         [ebp+arg_0], eax
                 call
                         sub 31486A
                 test
                         eax, eax
                         short loc 31306D
                 lea
                         eax, [ebp+arg_0]
                         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
                         sub 3140F3
                         eax, eax
                         short loc 31307D
                         sub_3140F3
                         short loc_31308C
loc_31307D:
                                          ; CODE XREF: sub 312FD8
                call
                         sub_3140F3
                         eax, 80070000h
loc_31308C:
                                          ; CODE XREF: sub 312FD8
```

push

call

test

edi

sub_314623

short loc_31306D [ebp+arg 0], ebx

short loc 313066 eax, [ebp+var 70] eax, [ebp+var 84]

eax, eax

Additional Processor Mod

- The IA-32 architecture supports 3
 - Real mode (kernel mode)
 - Protected mode (user mode)
 - System management mode
- Intel processors that support virtualiza have 2 more modes:
 - Root mode
 - Non-root mode

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

[ebp+arg 0],

short loc 31306D

[ebp+arg_4]

[ebp+arg_0]

System Management Mod

- Operating mode provided for handling:
 - Power management
 - System hardware control
 - Proprietary OEM designed code
- SMM provides a way to execute instructions in a manner that's hidden from the loc 313066: targeted OS!

```
[ebp+arg_4]
```

short loc 31306D [ebp+arg_0], ebx

short loc 313066

short loc 31306D

[ebp+arg_0]

```
/ sub 312FD8+55
                         sub 31411B
loc_31306D:
                                          ; CODE XREF: sub 312FD8
                                          ; sub 312FD8+49
                         sub 3140F3
```

```
short loc 31307D
sub 3140F3
short loc 31308C
                ; CODE XREF: sub 312FD8
```

loc_31307D:

[ebp+var 4], eax

eax, eax

Switching to SMM

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] edi [ebp+arg 0], eax 2. Processor saves sub 31486A 1. System Management eax, eax current state to short loc 31306D Interrupt (SMI) Occurs System Management eax, [ebp+arg_0] RAM (SMRAM) esi, 1D0h [ebp+arg_4] 3. Processor executes SMI handler 5. Processor loads saved (Also in SMRAM) loc 313066: state information and resumes normal operation sub_31411B loc_31306D : CODE XREF: sub 312FD8 ; sub 312FD8+49 call sub 3140F3 test eax, eax 4. Processor short loc 31307D encounters RSM call sub_3140F3 short loc 31308C THE C instruction (resume) ; CODE XREF: sub 312FD8 call sub_3140F3

push call

test

sub 314623

eax, eax

eax, 80070000h

Switching to SMM

- push edi
 call sub_314623
 test eax, eax
 jz short loc_31306D
 cmp [ebp+arg_0], ebx
 jnz short loc_313066
 mov eax, [ebp+var_70]
 cmp eax, [ebp+var_84]
 jb short loc_313066
 sub eax, [ebp+var_84]
 push esi
 push esi
- System Management Interrupt (SMI)
 - Normal Hardware Interrupt
 - No hardware-agnostic way to generate an SMI"
 - Most researchers generate an SMI by using Programmed Input/Output (PIO)
 - Talk to the hardware and get it to do something to cause an SMI

loc_31307D:

- SMRAM
 - —64 KB in size, all code and data the SMI Handler is composed of is here

SMI Handlers sub_312FD8
call sub_3140F3
test eax, eax
jg short loc_31307D
call sub_3140F3
jmp short loc_31308C

/ CODE XREF: sub_312FD8
call sub_3140F3

sub 31411B

SMM Operations

- 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
- No such things as privilege levels or address mappings
 - Just the handler and memory
 - The handler can arbitrarily modify anything in physical memory
 - SMM is controlled by flags in the SMRAMC Register
 - the SMRAMC register can be manipulated by PIO
 - Contains the D_LCK and D_OPEN flags

SMM Operations

- D_LCK Flag
 - When set, SMRAMC is read-only and D_OPEN is cleared
- D_OPEN Flag
 - When set, SMRAM can be accessed by non-SMM code
 - Until 2008, vendors used to leave this flag set by accident
 - -Bypass found in 2009

```
http://invisiblethingslab.com/sresources/misc09/ssf.sub_312FD8+49
smm_cache_fun.pdf

smm_cache_fun.pdf
```

loc 313066:

push

edi sub 314623

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

eax, [ebp+arg_0]

```
loc_31307D: ; CODE XREF: sub_312FD8
call sub_3140F3
and cax, Offern
```

sub 312FD8+55

eax, 80070000b

short loc 31308C

Rogue Hypervisors

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

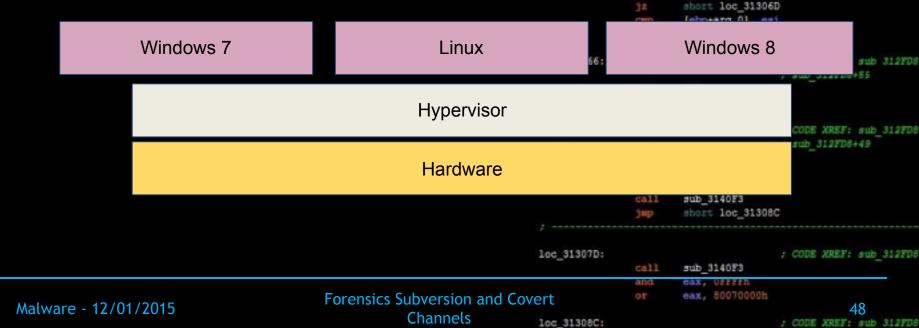
sub 314623

[ebp+var 4], eax

eax, eax

test

- Hypervisor/Virtual Machine Monitor (VMM)
 - Low-level layer of software that allows a computer to share its resources between multiple operating systems



Rogue Hypervisors

- Blue Pill proof-of-concept
 - Developed by Joanna Rutkowska, Alexander
 Tereshkin, and Rong Fan

```
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
                call
                         sub 31486A
                test
                         eax, [ebp+arg_0]
                         esi, 1D0h
                 push
                         881
                 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
                         0 Dh
                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
                call
                         sub_3140F3
                         eax, 80070000h
```

[ebp+var 4], eax

; CODE XREF: sub 312FD8

push

call

test

edi

sub_314623

eax, eax

Lights-Out Management

- OEM's have put more and more functionality at the motherboard level
 - Machines can be accessed and managed without an OS or without even being turned on
- Intel's Active Management Technology (AMT)
 - Has an independent processor to execute AMT code
 - Even supports a stand-alone web server
 - Researchers have demonstrated how to modify the code that AMT executes
 - Essentially construct a ring -3 rootkit

Bootkits

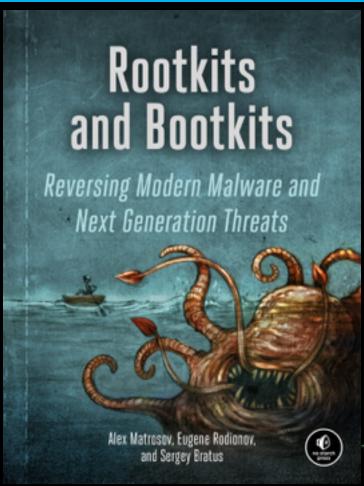
- push edi
 call sub_314623
 test eax, eax
 jz short loc_31306D
 cmp [ebp+arg_0], ebx
 jnz short loc_313066
 mov eax, [ebp+var_70]
 cmp eax, [ebp+var_84]
 jb short loc_313066
 sub eax, [ebp+var_84]
 push esi
 push esi
- Subvert the boot process so that an attacker's code is executed first
- The next major development in malware
- Mostly used to bypass Microsoft Patchguard
- Unfortunately, outside of the scope of this course

```
loc 313066:
                                          ; sub 312FD8+55
                push
                call
                         sub 31411B
loc_31306D:
                                          ; CODE XREF: sub 312FD8
                                          ; sub 312FD8+49
                         sub 3140F3
                         eax, eax
                         short loc 31307D
                         sub 3140F3
                         short loc 31308C
loc_31307D:
                                          ; CODE XREF: sub 312FD8
                         sub 3140F3
```

[ebp+var 4], eax

: CODE XREF: sub 312FD8

Bootkits



```
inz
        short loc 313066
        eax, [ebp+var 70]
        eax, [ebp+var_84]
        short loc 313066
        eax, [ebp+var_84]
sub
push
        eax
push
        [ebp+arg_0], eax
MOV
call
        sub 31486A
test
        eax, eax
        short loc 31306D
push
        esi
lea
        eax, [ebp+arg_0]
push
        esi, 1D0h
MOV
push
        681
push
        [ebp+arg_4]
push
        edi
call
        sub 314623
        eax, eax
test
        short loc_31306D
        [ebp+arg_0], esi
        short loc 31308F
                         ; CODE XREF: sub 312FD8
                         ; sub 312FD8+55
push
call
        sub 31411B
                         ; CODE XREF: sub 312FD8
                         ; sub 312FD8+49
call
        sub 3140F3
test
        eax, eax
        short loc 31307D
call.
        sub_3140F3
THE
        short loc 31308C
                         ; CODE XREF: sub 312FD8
        sub_3140F3
call
```

push

call

test

edi

sub 314623

short loc_31306D [ebp+arg_0], ebx

eax, eax

eax, 80070000h

		push	edi	
		call	sub_314623	
		test	eax, eax	
		jz	short loc_31306D	
		Camp	[ebp+arg_0], ebx	
		jnz	short loc_313066	
		MOV	eax, [ebp+var_70]	
		Cmp	eax, [ebp+var_84]	
)b	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]	
	Questions?	push	eax	
	CHESTIONS:	mov	esi, 1D0h	
	Questions.	push	esi Ishnang 41	
		push	[ebp+arg_4] edi	
		push		
		test	sub_314623 eax, eax	
		jz	short loc_31306D	
		cmp	[ebp+arg_0], esi	
		jz	short loc_31308F	
	loc_313066:		; CODE XREF: su	b 312FD8
			; sub_312FD8+55	
		push	ODh	
		call	sub_31411B	
			-	
	loc_31306D:		; CODE XREF: su	
			/ sub_312FD8+49	
		call	sub_3140F3	
		test	eax, eax	
		30	short loc_31307D	
		call	aub_3140F3	
		jmp	short loc_31308C	
	lec_31307D:		; CODE XREF: su	b 312FD8
		call	sub_3140F3	
		and	eax, Offffn	
Malware - 12/01/2015	Forensics Subversion and Covert		eax, 80070000h	
	Characala		53	
	Channels 100_31308C:		; CODE XREF: su	0_312FD8
		MOV	[ebp+var_4], eax	

References

```
push edi
call sub_314623
test eax, eax
jz short loc_313060
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
```

- Blunden, Bill. The Rootkit Arsenal Escape and Evasion in the Dark Corners of the System, Second Edition. 2nd ed. Burlington, Mass.: Jones & Bartlett Learning, 2013. Print.
- 1. Hoglund, Greg, and James Butler. Rootkits: Subverting the Windows Kernel. Upper Saddle River, NJ: Addison-Wesley, 2006. Print.

```
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
                call
                         sub_3140F3
                         eax, 80070000h
```