Malware Unpacking Workshop

"Ah, finished unpacking."
"Time to decorate!"

Lilly Chalupowski
August 28, 2019

whois lilly.chalupowski

Table: *who.is results*

Name	Lilly Chalupowski
Status	Employed
Creation Date	1986
Expiry	A Long Time from Now (Hopefully)
Registrant Name	GoSecure
Administrative Contact	Travis Barlow
Job	TITAN Malware Research Lead

Agenda

What will we cover?

- Disclaimer
- Reverse Engineering
  - Registers
  - Stack
  - Heap
  - Assembly
  - Calling Conventions
- Tools
  - x64dbg
  - Cutter
  - Radare2
  - Detect it Easy
  - HxD
- Injection Techniques
  - DLL Injection
  - PE Injection
  - Process Hollowing
  - Atom Bombing
- Workshop

Disclaimer

Don't be a Criminal

disclaimer.log

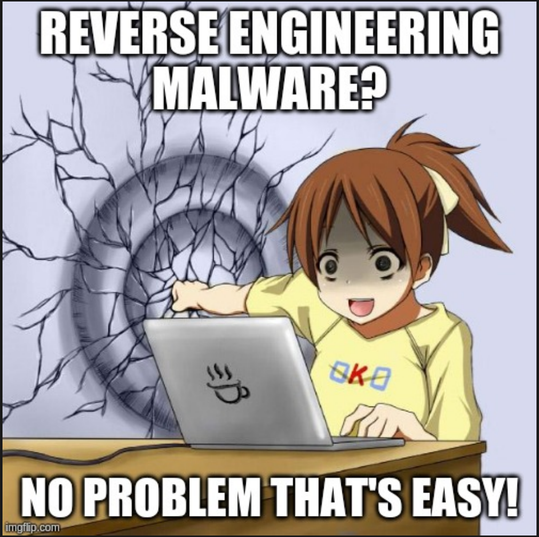
The tools and techniques covered in this presentation can be dangerous and are being shown for educational purposes.

It is a violation of Federal laws to attempt gaining unauthorized access to information, assets or systems belonging to others, or to exceed authorization on systems for which you have not been granted.

Only use these tools with/on systems you own or have written permission from the owner. I (the speaker) do not assume any responsibility and shall not be held liable for any illegal use of these tools.

Reverse Engineering

It's easy don't worry!


REVERSE ENGINEERING MALWARE?
OKO
NO PROBLEM THAT'S EASY!
imgflip.com

Registers

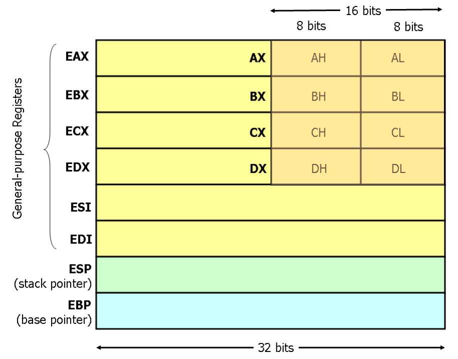
Not this one!

We Rent Tools
Professional Tools For Professional Results
Tool Rental
WE RENT
Windows XP Professional

Registers

Not the kind with money in them

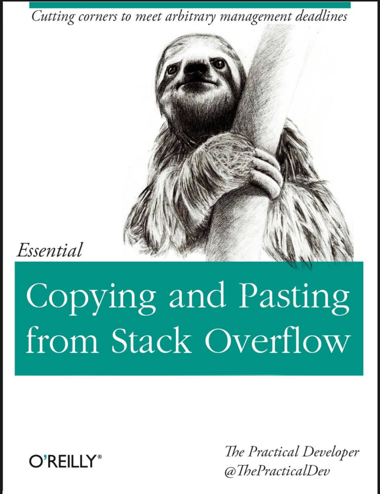
- EAX - Return Value of Functions
- EBX - Base Index (for use with arrays)
- ECX - Counter in Loops
- EDI - Destination Memory Operations
- ESI - Source Memory Operations
- ESP - Stack Pointer
- EBP - Base Frame Pointer


16 bits
8 bits
8 bits
General-purpose Registers
EAX
AX
AH
AL
EBX
BX
BH
BL
ECX
CX
CH
CL
EDX
DX
DH
DL
ESI
EDI
ESP
(stack pointer)
EBP
(base pointer)
32 bits

Did You Know: In computer architecture, a processor register is a quickly accessible location available to a computer's central processing unit (CPU).

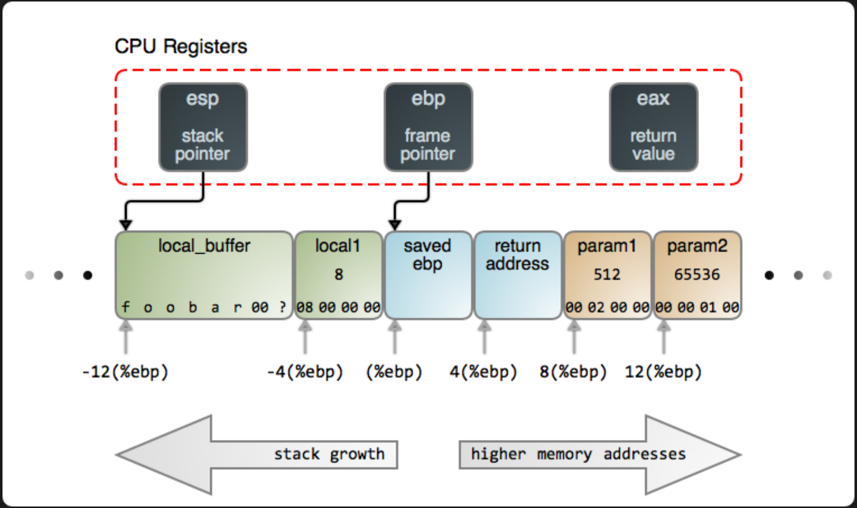
The Stack

- Last-In First-Out
- Downward Growth
- Function Local Variables
- ESP
- Increment / Decrement = 4
    - Double-Word Aligned


Cutting corners to meet arbitrary management deadlines
Essential
Copying and Pasting from Stack Overflow
O'REILLY®
The Practical Developer
@ThePracticalDev

Stack

The stack



CPU Registers
esp
stack pointer
ebp
frame pointer
eax
return value
local_buffer
f o o b a r 00 ?
local1
8
08 00 00 00
saved ebp
return address
param1
512
00 02 00 00
param2
65536
00 00 01 00
-12(%ebp)
-4(%ebp)
(%ebp)
4(%ebp)
8(%ebp)
12(%ebp)
stack growth
higher memory addresses

Control Flow

Keeping it under control

- Conditionals
  - CMP
  - TEST
  - JMP
  - JCC
- EFLAGS
  - ZF / Zero Flag
  - SF / Sign Flag
  - CF / Cary Flag
  - OF/Overflow Flag

zf = 0
zf = 1
JAKE-CLARK.TUMBLR
imgflip.com

Calling Conventions

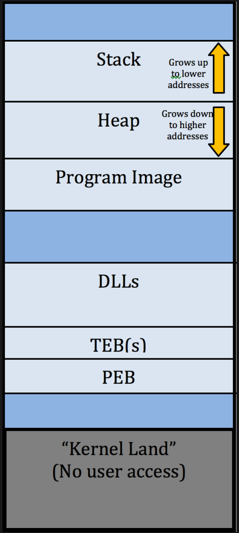
Subtitle goes here

- CDECL
  - Arguments Right-to-Left
  - Return Values in EAX
  - Calling Function Cleans the Stack
- STDCALL
  - Used in Windows Win32API
  - Arguments Right-to-Left
  - Return Values in EAX
  - The called function cleans the stack, unlike CDECL
  - Does not support variable arguments
- FASTCALL
  - Uses registers as arguments
  - Useful for shellcode

LINUX
WINDOWS
CDECL
imgflip.com

Windows Memory Structure
subtitle

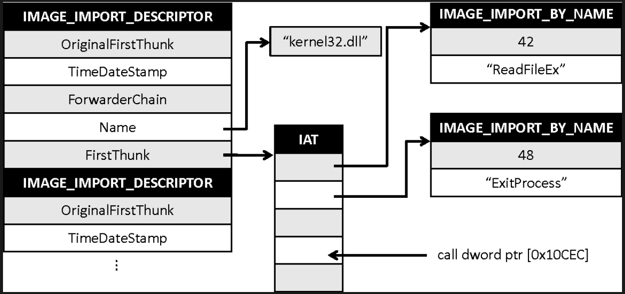
- Stack - Grows up to lower addresses
- Heap - Grows down to higher addresses
- Program Image
- TEB - Thread Environment Block
  - GetLastError()
  - GetVersion()
  - Pointer to the PEB
- PEB - Process Environment Block
  - Image Name
  - Global Context
  - Startup Parameters
  - Image Base Address
  - IAT (Import Address Table)


Stack
Grows up to lower addresses
Heap
Grows down to higher addresses
Program Image
DLLs
TEB(s)
PEB
"Kernel Land" (No user access)

IAT (Import Address Table) and IDT (Import Lookup Table)

subtitle

- Identical to the IDT (Import Directory Table)
- Binding - The process of where functions are mapped to their virtual addresses overwriting the IAT
- Often the IDT and IAT must be rebuilt when packing and unpacking malware


IMAGE_IMPORT_DESCRIPTOR
OriginalFirstThunk
TimeDateStamp
ForwarderChain
Name
FirstThunk
IMAGE_IMPORT_DESCRIPTOR
OriginalFirstThunk
TimeDateStamp
⋮
"kernel32.dll"
IAT
IMAGE_IMPORT_BY_NAME
42
"ReadFileEx"
IMAGE_IMPORT_BY_NAME
48
"ExitProcess"
call dword ptr [0x10CEC]

Assembly
Instructions

- Common Instructions
  - MOV
  - XOR
  - PUSH
  - POP

ASSEMBLY
HE FINALLY LEARNED IT
imgflip.com
thruF.com

Assembly CDECL (Linux)

subtitle

cdecl.c

```c
__cdecl int add_cdecl(int a, int b){
  return a + b;
}
int x = add_cdecl(2, 3);
```

Assembly CDECL (Linux)

subtitle

cdecl.asm

```
_add_cdecl:
  push ebp
  mov ebp, esp
  mov eax, [ebp + 8]  ; get 3 from the stack
  mov edx, [ebp + 12] ; get 2 from the stack
  add eax, edx        ; add values to eax
  pop ebp
  ret
_start:
  push 3              ; second argument
  push 2              ; first argument
  call _add_cdecl
  add esp, 8
```

Assembly STDCALL (Windows)

subtitle

stdcall.c

```c
__stdcall int add_stdcall(int a, int b){
  return a + b;
}
int x = add_stdcall(2, 3);
```

Assembly STDCALL (Windows)

subtitle

stdcall.asm

```
_add_stdcall:
  push ebp
  mov ebp, esp
  mov eax, [ebp + 8]  ; set eax to 3
  mov edx, [ebp + 12] ; set edx to 2
  add eax, edx
  pop ebp
  ret 8               ; how many bytes to pop
_start:               ; main function
  push 3              ; second argument
  push 2              ; first argument
  call _add_stdcall
```

Assembly FASTCALL

subtitle

cdecl.c

```c
__fastcall int add_fastcall(int a, int b){
  return a + b;
}
int x = add_fastcall(2, 3);
```

Assembly FASTCALL

subtitle

fastcall.asm

```
_add_fastcall:
  push ebp
  mov ebp, esp
  add eax, edx          ; add and save result in eax
  pop ebp
  ret
_start:
  mov eax, 2            ; first argument
  mov edx, 3            ; second argument
  call _add_fastcall
```

Guess the Calling Convention

Hello World Intel Syntax

hello.asm

```
section     .text                   ; the code section
global      _start                  ; tell linker entrypoint
_start:
  mov     edx,len                   ; message length
  mov     ecx,msg                   ; message to write
  mov     ebx,1                     ; file descriptor stdout
  mov     eax,4                     ; syscall number for write
  int     0x80                      ; linux x86 interrupt
  mov     eax,1                     ; syscall number for exit
  int     0x80                      ; linux x86 interrupt
section     .data                   ; the data section
  msg     db  'Hello, world!',0x0   ; null terminated string
  len     equ \$ - msg              ; message length
```

Assembler and Linking

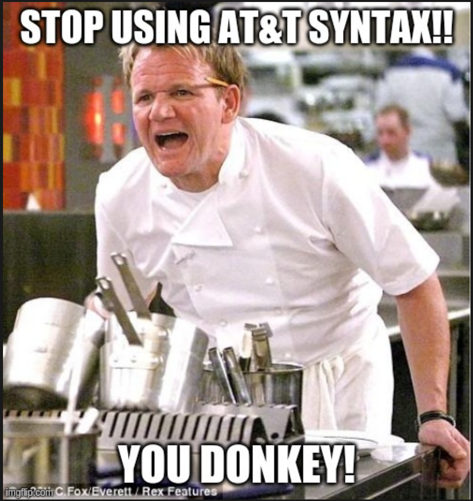
subtitle

terminal

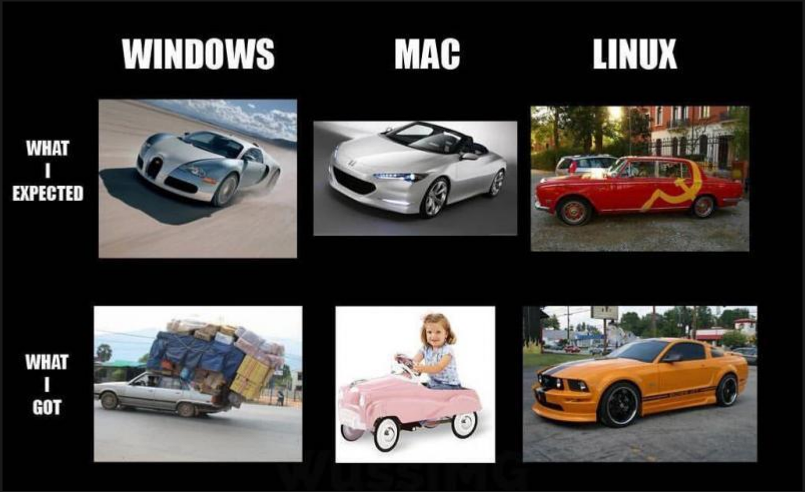
```
malware@work ~$ nasm -f elf32 -o hello.o hello.asm
malware@work ~$ ld -m elf_i386 -o hello hello.o
malware@work ~$ ./hello
Hello, World!
malware@work ~$
```

Assembly Flavors

I know you were thinking it!

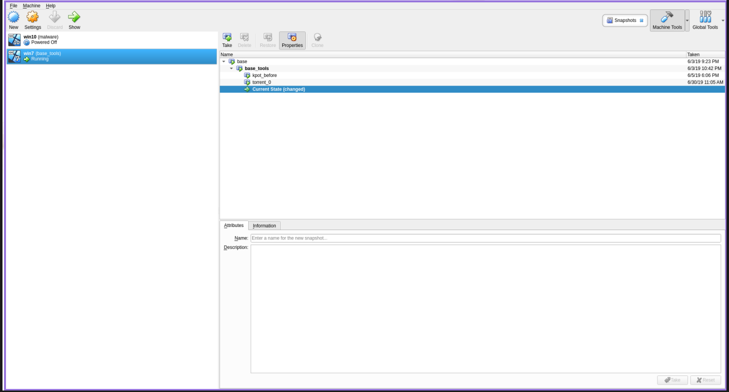

STOP USING AT&T SYNTAX!!
YOU DONKEY!
imgflip.com C.Fox/Everett / Rex Features

Tools of the Trade


WINDOWS
MAC
LINUX
WHAT I EXPECTED
WHAT I GOT

VirtualBox
Purpose

- Snapshots
- Security Layer
- Multiple Systems


File
Machine
Help
New
Settings
Discard
Show
Snapshots
Machine Tools
Global Tools
Take
Delete
Restore
Properties
Clone
Attributes
Information
Take
Reset

win10 (malware)
Powered Off

win7 (base_tools)
Running

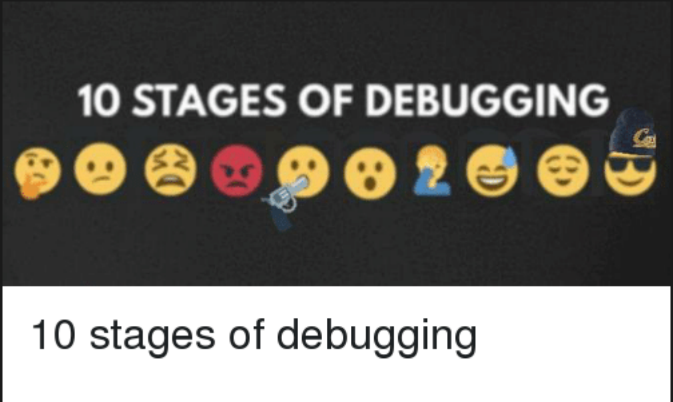
Name	Taken
base	6/3/19 9:23 PM
base_tools	6/3/19 10:42 PM
kpot_before	6/5/19 6:06 PM
torrent_0	6/30/19 11:05 AM
Current State (changed)	

Name: Enter a name for the new snapshot...
Description:

x64dbg

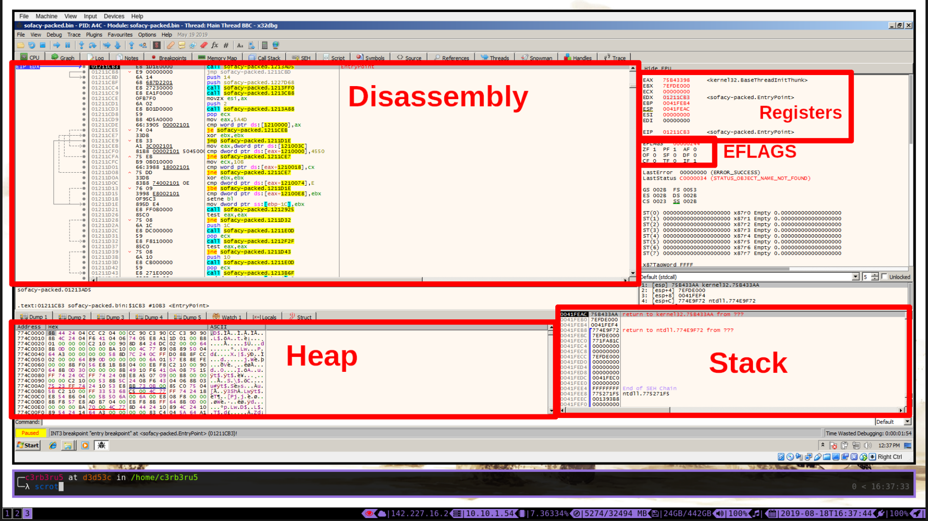
Purpose

- Resolving APIs
- Dumping Memory
- Modify Control Flow
- Identify Key Behaviors


10 STAGES OF DEBUGGING
10 stages of debugging

x64dbg

Overview


Disassembly
Registers
EFLAGS
Heap
Stack
File Machine View Input Devices Help
File View Debug Trace Plugins Favourites Options Help
CPU Graph Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Snowman Handles Trace
Dump 1 Dump 2 Dump 3 Dump 4 Dump 5 Watch 1 Locals Struct
Start

sofacy-packed.bin - PID: A4C - Module: sofacy-packed.bin - Thread: Main Thread BBC - x32dbg

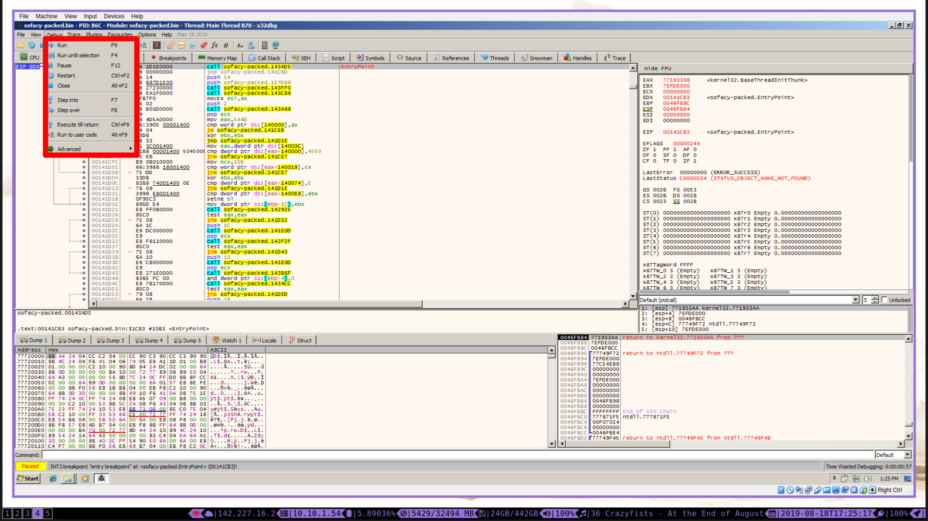
```
sofacy-packed.01213AD5

.text:01211CB3 sofacy-packed.bin:$1CB3 #10B3 <EntryPoint>
```

```
Paused   INT3 breakpoint "entry breakpoint" at <sofacy-packed.EntryPoint> (01211CB3)!
```

```
┌─c3rb3ru5 at d3d53c in /home/c3rb3ru5
└─λ scrot
```

x64dbg

Navigation



File Machine View Input Devices Help
sofacy-packed.bin - PID: B6C - Module: sofacy-packed.bin - Thread: Main Thread B70 - x32dbg
File View Debug Trace Plugins Favourites Options Help May 19 2019
Run F9
Run until selection F4
Pause F12
Restart Ctrl+F2
Close Alt+F2
Step into F7
Step over F8
Execute till return Ctrl+F9
Run to user code Alt+F9
Advanced
CPU
Breakpoints
Memory Map
Call Stack
SEH
Script
Symbols
Source
References
Threads
Snowman
Handles
Trace
Hide FPU
Dump 1
Dump 2
Dump 3
Dump 4
Dump 5
Watch 1
Locals
Struct
Command:
Default
Start
Right Ctrl

```
call sofacy-packed.143AD5                          EntryPoint
jmp sofacy-packed.141CBD
push 14
push sofacy-packed.157D68
call sofacy-packed.143FF0
call sofacy-packed.143CB8
movzx esi,ax
push 2
call sofacy-packed.143A88
pop ecx
mov eax,5A4D
cmp word ptr ds:[140000],ax
je sofacy-packed.141CEB
xor ebx,ebx
jmp sofacy-packed.141D1E
mov eax,dword ptr ds:[14003C]
cmp dword ptr ds:[eax+140000],4550
jne sofacy-packed.141CE7
00141CFC     B9 0B010000       mov ecx,10B
00141D01     66:3988 18001400  cmp word ptr ds:[eax+140018],cx
00141D08   ^ 75 DD             jne sofacy-packed.141CE7
00141D0A     33DB              xor ebx,ebx
00141D0C     83B8 74001400 0E  cmp dword ptr ds:[eax+140074],E
00141D13   v 76 09             jbe sofacy-packed.141D1E
00141D15     3998 E8001400     cmp dword ptr ds:[eax+1400E8],ebx
00141D1B     0F95C3            setne bl
00141D1E     895D E4           mov dword ptr ss:[ebp-1C],ebx
00141D21     E8 FF0B0000       call sofacy-packed.142925
00141D26     85C0              test eax,eax
00141D28   v 75 08             jne sofacy-packed.141D32
00141D2A     6A 1C             push 1C
00141D2C     E8 DC000000       call sofacy-packed.141E0D
00141D31     59                pop ecx
00141D32     E8 F8110000       call sofacy-packed.142F2F
00141D37     85C0              test eax,eax
00141D39   v 75 08             jne sofacy-packed.141D43
00141D3B     6A 10             push 10
00141D3D     E8 CB000000       call sofacy-packed.141E0D
00141D42     59                pop ecx
00141D43     E8 271E0000       call sofacy-packed.143B6F
00141D48     8365 FC 00        and dword ptr ss:[ebp-4],0
00141D4C     E8 7B170000       call sofacy-packed.1434CC
00141D51     85C0              test eax,eax
00141D53   v 79 08             jns sofacy-packed.141D5D
00141D55     6A 1B             push 1B

sofacy-packed.00143AD5

.text:00141CB3 sofacy-packed.bin:$1CB3 #10B3 <EntryPoint>
```

```
EAX   77193398    <kernel32.BaseThreadInitThunk>
EBX   7EFDE000
ECX   00000000
EDX   00141CB3    <sofacy-packed.EntryPoint>
EBP   0046FB8C
ESP   0046FB84
ESI   00000000
EDI   00000000

EIP   00141CB3    <sofacy-packed.EntryPoint>

EFLAGS   00000244
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000000 (ERROR_SUCCESS)
LastStatus C0000034 (STATUS_OBJECT_NAME_NOT_FOUND)

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B

x87TagWord FFFF
x87TW_0 3 (Empty)   x87TW_1 3 (Empty)
x87TW_2 3 (Empty)   x87TW_3 3 (Empty)
x87TW_4 3 (Empty)   x87TW_5 3 (Empty)
x87TW_6 3 (Empty)   x87TW_7 3 (Empty)
```

Default (stdcall) 5 ☐ Unlocked

```
1: [esp] 771933AA kernel32.771933AA
2: [esp+4] 7EFDE000
3: [esp+8] 0046FBCC
4: [esp+C] 77749F72 ntdll.77749F72
5: [esp+10] 7EFDE000
```

```
0046FB84  771933AA  return to kernel32.771933AA from ???
0046FB88  7EFDE000
0046FB8C  0046FBCC
0046FB90  77749F72  return to ntdll.77749F72 from ???
0046FB94  7EFDE000
0046FB98  77C54EB8
```

Paused INT3 breakpoint "entry breakpoint" at <sofacy-packed.EntryPoint> (00141CB3)!

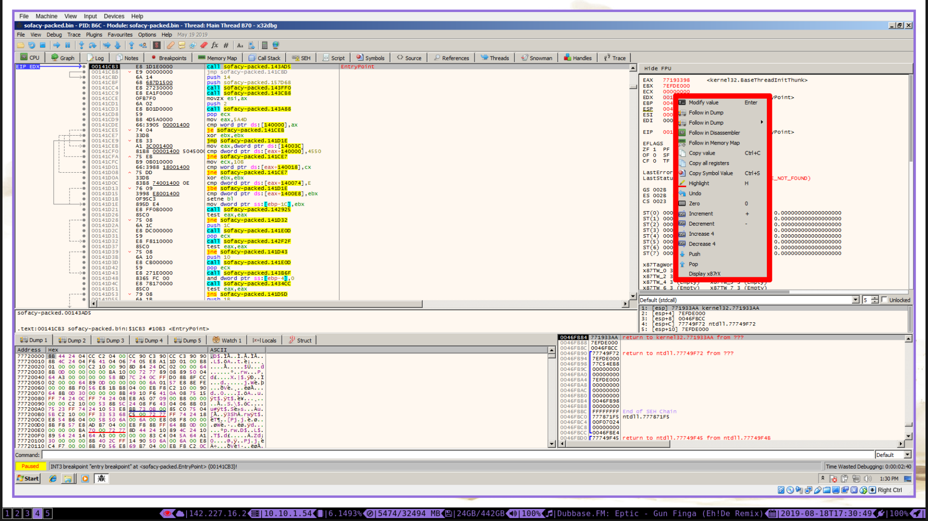
Time Wasted Debugging: 0:00:00:57

1:25 PM

1 2 3 4 5 |142.227.16.2|10.10.1.54|5.89036%|5429/32494 MB|24GB/442GB|100%|36 Crazyfists - At the End of August|2019-08-18T17:25:17|100%

x64dbg

Context Menus


sofacy-packed.bin - PID: B6C - Module: sofacy-packed.bin - Thread: Main Thread B70 - x32dbg

File Machine View Input Devices Help
File View Debug Trace Plugins Favourites Options Help May 19 2019
CPU Graph Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Snowman Handles Trace
Hide FPU
Modify value Enter
Follow in Dump
Follow in Dump
Follow in Disassembler
Follow in Memory Map
Copy value Ctrl+C
Copy all registers
Copy Symbol Value Ctrl+S
Highlight H
Undo
Zero 0
Increment +
Decrement -
Increase 4
Decrease 4
Push
Pop
Display x87rX
Dump 1 Dump 2 Dump 3 Dump 4 Dump 5 Watch 1 Locals Struct
Default (stdcall)
Unlocked
Command:
Default
Paused
Start
Right Ctrl

```
sofacy-packed.001430A5
.text:00141CB3 sofacy-packed.bin:$1CB3 #10B3 <EntryPoint>
```

```
EAX 77193398 <kernel32.BaseThreadInitThunk>
EBX 7EFDE000
```

```
1: [esp] 771933AA kernel32.771933AA
2: [esp+4] 7EFDE000
3: [esp+8] 0046FBCC
4: [esp+C] 77749F72 ntdll.77749F72
5: [esp+10] 7EFDE000
```

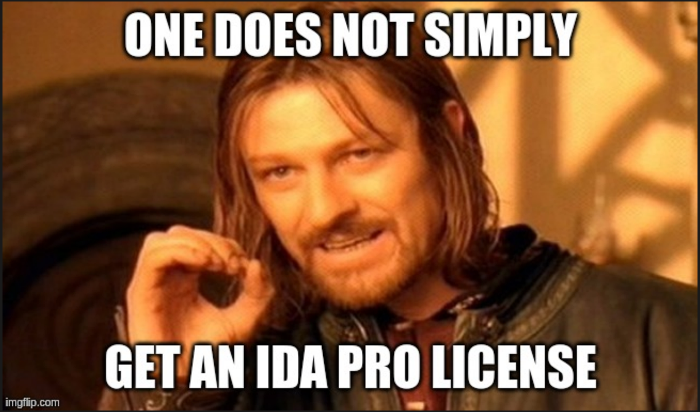
INT3 breakpoint "entry breakpoint" at <sofacy-packed.EntryPoint> (00141CB3)!

Time Wasted Debugging: 0:00:02:40

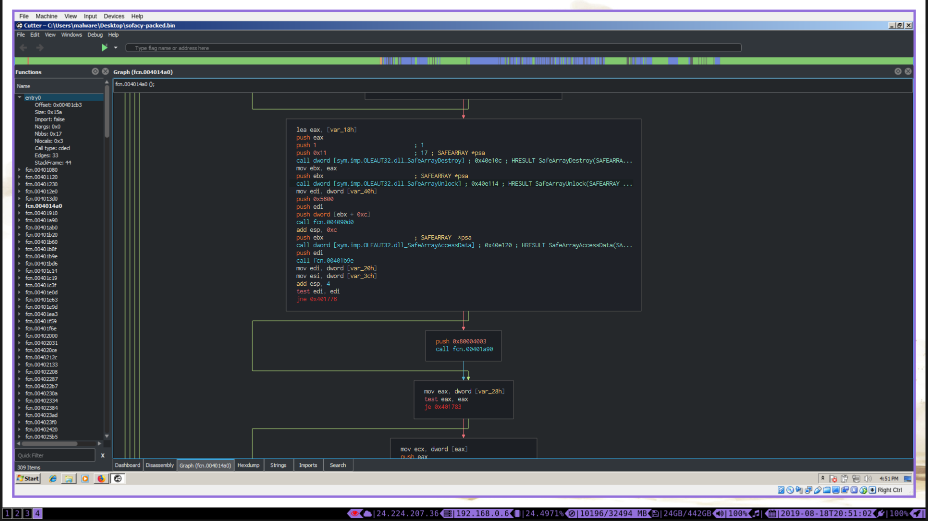
Cutter

Purpose

- Markup Reverse Engineered Code
- Control Flow Navigation
- Pseudo Code


ONE DOES NOT SIMPLY
GET AN IDA PRO LICENSE
imgflip.com

Cutter
Graph View


File Machine View Input Devices Help
Cutter – C:\Users\malware\Desktop\sofacy-packed.bin
File Edit View Windows Debug Help
Type flag name or address here
Functions
Graph (fcn.004014a0)
Name
Quick Filter
309 Items
Dashboard Disassembly Graph (fcn.004014a0) Hexdump Strings Imports Search
Start
4:51 PM
Right Ctrl
1 2 3 4
24.224.207.36 | 192.168.0.6 | 24.4971% | 10196/32494 MB | 24GB/442GB | 100% | 2019-08-18T20:51:02 | 100%

```
entry0
    Offset: 0x00401cb3
    Size: 0x15a
    Import: false
    Nargs: 0x0
    Nbbs: 0x17
    Nlocals: 0x3
    Call type: cded
    Edges: 33
    StackFrame: 44
fcn.00401080
fcn.00401120
fcn.00401230
fcn.004012e0
fcn.004013d0
fcn.004014a0
fcn.00401910
fcn.00401a90
fcn.00401ab0
fcn.00401b20
fcn.00401b60
fcn.00401b8f
fcn.00401b9e
fcn.00401bd6
fcn.00401c14
fcn.00401c19
fcn.00401c3f
fcn.00401e0d
fcn.00401e63
fcn.00401e9d
fcn.00401ea3
fcn.00401f59
fcn.00401f6e
fcn.00402000
fcn.00402031
fcn.004020ce
fcn.0040212c
fcn.00402133
fcn.00402208
fcn.00402287
fcn.004022b7
fcn.0040230a
fcn.00402334
fcn.00402384
fcn.004023ad
fcn.004023f0
fcn.00402420
fcn.004025b5
```

```
fcn.004014a0 ();
```

```
lea eax, [var_18h]
push eax
push 1                          ; 1
push 0x11                       ; 17 ; SAFEARRAY *psa
call dword [sym.imp.OLEAUT32.dll_SafeArrayDestroy] ; 0x40e10c ; HRESULT SafeArrayDestroy(SAFEARRA...
mov ebx, eax
push ebx                        ; SAFEARRAY *psa
call dword [sym.imp.OLEAUT32.dll_SafeArrayUnlock] ; 0x40e114 ; HRESULT SafeArrayUnlock(SAFEARRAY ...
mov edi, dword [var_40h]
push 0x5600
push edi
push dword [ebx + 0xc]
call fcn.004090d0
add esp, 0xc
push ebx                        ; SAFEARRAY  *psa
call dword [sym.imp.OLEAUT32.dll_SafeArrayAccessData] ; 0x40e120 ; HRESULT SafeArrayAccessData(SA...
push edi
call fcn.00401b9e
mov edi, dword [var_20h]
mov esi, dword [var_3ch]
add esp, 4
test edi, edi
jne 0x401776
```

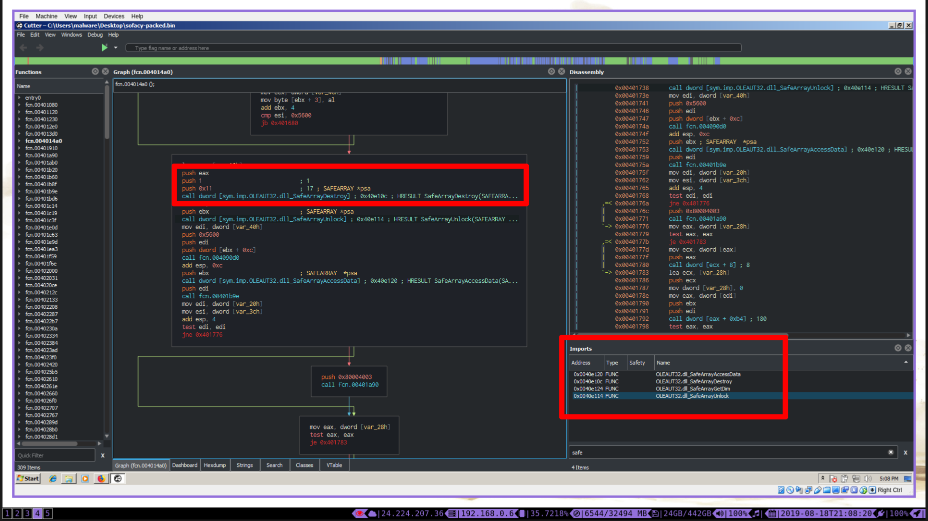
```
push 0x80004003
call fcn.00401a90
```

```
mov eax, dword [var_28h]
test eax, eax
je 0x401783
```

```
mov ecx, dword [eax]
push eax
```

Cutter

Navigation



File Machine View Input Devices Help
Cutter – C:\Users\malware\Desktop\sofacy-packed.bin
File Edit View Windows Debug Help
Type flag name or address here
Functions
Name
Graph (fcn.004014a0)
Disassembly
Imports
Quick Filter
309 Items
Graph (fcn.004014a0)
Dashboard
Hexdump
Strings
Search
Classes
VTable
safe
4 Items
Start
5:08 PM
Right Ctrl
24.224.207.36 | 192.168.0.6 | 35.7218% | 6544/32494 MB | 24GB/442GB | 100% | 2019-08-18T21:08:20 | 100%

```
entry0
fcn.00401080
fcn.00401120
fcn.00401230
fcn.004012e0
fcn.004013d0
fcn.004014a0
fcn.00401910
fcn.00401a90
fcn.00401ab0
fcn.00401b20
fcn.00401b60
fcn.00401b8f
fcn.00401b9e
fcn.00401bd6
fcn.00401c14
fcn.00401c19
fcn.00401c3f
fcn.00401e0d
fcn.00401e63
fcn.00401e9d
fcn.00401ea3
fcn.00401f59
fcn.00401f6e
fcn.00402000
fcn.00402031
fcn.004020ce
fcn.0040212c
fcn.00402133
fcn.00402208
fcn.00402287
fcn.004022b7
fcn.0040230a
fcn.00402334
fcn.00402384
fcn.004023ad
fcn.004023f0
fcn.00402420
fcn.004025b5
fcn.00402610
fcn.0040261e
fcn.00402660
fcn.004026f0
fcn.00402707
fcn.00402767
fcn.0040289d
fcn.004028b0
fcn.004028d1
```

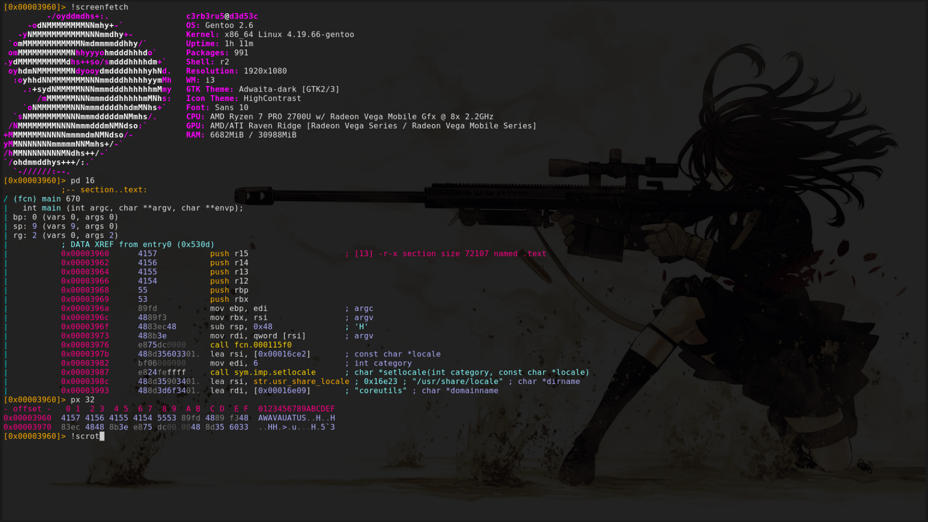
```
fcn.004014a0 ();

mov byte [ebx + 3], al
add ebx, 4
cmp esi, 0x5600
jb 0x401680

push eax
push 1                              ; 1
push 0x11                           ; 17 ; SAFEARRAY *psa
call dword [sym.imp.OLEAUT32.dll_SafeArrayDestroy] ; 0x40e10c ; HRESULT SafeArrayDestroy(SAFEARRA...
push ebx                            ; SAFEARRAY *psa
call dword [sym.imp.OLEAUT32.dll_SafeArrayUnlock] ; 0x40e114 ; HRESULT SafeArrayUnlock(SAFEARRAY ...
mov edi, dword [var_40h]
push 0x5600
push edi
push dword [ebx + 0xc]
call fcn.004090d0
add esp, 0xc
push ebx                            ; SAFEARRAY  *psa
call dword [sym.imp.OLEAUT32.dll_SafeArrayAccessData] ; 0x40e120 ; HRESULT SafeArrayAccessData(SA...
push edi
call fcn.00401b9e
mov edi, dword [var_20h]
mov esi, dword [var_3ch]
add esp, 4
test edi, edi
jne 0x401776

push 0x80004003
call fcn.00401a90

mov eax, dword [var_28h]
test eax, eax
je 0x401783
```

```
|          0x00401738      call dword [sym.imp.OLEAUT32.dll_SafeArrayUnlock] ; 0x40e114 ; HRESULT Sa
|          0x0040173e      mov edi, dword [var_40h]
|          0x00401741      push 0x5600
|          0x00401746      push edi
|          0x00401747      push dword [ebx + 0xc]
|          0x0040174a      call fcn.004090d0
|          0x0040174f      add esp, 0xc
|          0x00401752      push ebx ; SAFEARRAY  *psa
|          0x00401753      call dword [sym.imp.OLEAUT32.dll_SafeArrayAccessData] ; 0x40e120 ; HRESUL
|          0x00401759      push edi
|          0x0040175a      call fcn.00401b9e
|          0x0040175f      mov edi, dword [var_20h]
|          0x00401762      mov esi, dword [var_3ch]
|          0x00401765      add esp, 4
|          0x00401768      test edi, edi
|      ,=< 0x0040176a      jne 0x401776
|      |   0x0040176c      push 0x80004003
|      |   0x00401771      call fcn.00401a90
|      `-> 0x00401776      mov eax, dword [var_28h]
|          0x00401779      test eax, eax
|      ,=< 0x0040177b      je 0x401783
|      |   0x0040177d      mov ecx, dword [eax]
|      |   0x0040177f      push eax
|      |   0x00401780      call dword [ecx + 8] ; 8
|      `-> 0x00401783      lea ecx, [var_28h]
|          0x00401786      push ecx
|          0x00401787      mov dword [var_28h], 0
|          0x0040178e      mov eax, dword [edi]
|          0x00401790      push ebx
|          0x00401791      push edi
|          0x00401792      call dword [eax + 0xb4] ; 180
|          0x00401798      test eax, eax
```

Address	Type	Safety	Name
0x0040e120	FUNC		OLEAUT32.dll_SafeArrayAccessData
0x0040e10c	FUNC		OLEAUT32.dll_SafeArrayDestroy
0x0040e124	FUNC		OLEAUT32.dll_SafeArrayGetDim
0x0040e114	FUNC		OLEAUT32.dll_SafeArrayUnlock

Radare2

Backend of Cutter

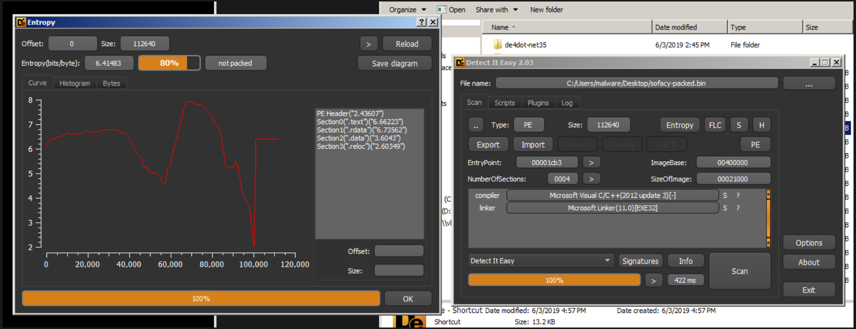


```
[0x00003960]> !screenfetch
         -/oyddmdhs+:.                c3rb3ru5@d3d53c
     -odNMMMMMMMMNNmhy+-`             OS: Gentoo 2.6
   -yNMMMMMMMMMMMNNNmmdhy+-           Kernel: x86_64 Linux 4.19.66-gentoo
 `omMMMMMMMMMMMMNmdmmmmddhhy/`        Uptime: 1h 11m
 omMMMMMMMMMMMNhhyyyohmdddhhhdo`      Packages: 991
.ydMMMMMMMMMMdhs++so/smdddhhhhdm+`    Shell: r2
 oyhdmNMMMMMMMNdyooydmddddhhhhyhNd.   Resolution: 1920x1080
  :oyhhdNNMMMMMMMNNNmmdddhhhhhyymMh   WM: i3
    .:+sydNMMMMMNNNmmmdddhhhhhhmMmy   GTK Theme: Adwaita-dark [GTK2/3]
       /mMMMMMMNNNmmmdddhhhhhmMNhs:   Icon Theme: HighContrast
    `oNMMMMMMMNNNmmmddddhhdmMNhs+`    Font: Sans 10
  `sNMMMMMMMMNNNmmmdddddmNMmhs/.      CPU: AMD Ryzen 7 PRO 2700U w/ Radeon Vega Mobile Gfx @ 8x 2.2GHz
 /NMMMMMMMMNNNNmmmdddmNMNdso:`        GPU: AMD/ATI Raven Ridge [Radeon Vega Series / Radeon Vega Mobile Series]
+MMMMMMMNNNNNmmmmdmNMNdso/-           RAM: 6682MiB / 30988MiB
yMMNNNNNNNmmmmmNNMmhs+/-`
/hMMNNNNNNNNMNdhs++/-`
`/ohdmmddhys+++/:.`
  `-//////:--.
[0x00003960]> pd 16
            ;-- section..text:
/ (fcn) main 670
|   int main (int argc, char **argv, char **envp);
| bp: 0 (vars 0, args 0)
| sp: 9 (vars 9, args 0)
| rg: 2 (vars 0, args 2)
|           ; DATA XREF from entry0 (0x530d)
|           0x00003960      4157           push r15                    ; [13] -r-x section size 72107 named .text
|           0x00003962      4156           push r14
|           0x00003964      4155           push r13
|           0x00003966      4154           push r12
|           0x00003968      55             push rbp
|           0x00003969      53             push rbx
|           0x0000396a      89fd           mov ebp, edi                ; argc
|           0x0000396c      4889f3         mov rbx, rsi                ; argv
|           0x0000396f      4883ec48       sub rsp, 0x48               ; 'H'
|           0x00003973      488b3e         mov rdi, qword [rsi]        ; argv
|           0x00003976      e875dc0000     call fcn.000115f0
|           0x0000397b      488d35603301.  lea rsi, [0x00016ce2]       ; const char *locale
|           0x00003982      bf06000000     mov edi, 6                  ; int category
|           0x00003987      e824feffff     call sym.imp.setlocale      ; char *setlocale(int category, const char *locale)
|           0x0000398c      488d35903401.  lea rsi, str.usr_share_locale ; 0x16e23 ; "/usr/share/locale" ; char *dirname
|           0x00003993      488d3d6f3401.  lea rdi, [0x00016e09]       ; "coreutils" ; char *domainname
[0x00003960]> px 32
- offset -   0 1  2 3  4 5  6 7  8 9  A B  C D  E F  0123456789ABCDEF
0x00003960  4157 4156 4155 4154 5553 89fd 4889 f348  AWAVAUATUS..H..H
0x00003970  83ec 4848 8b3e e875 dc00 0048 8d35 6033  ..HH.>.u...H.5`3
[0x00003960]> !scrot
```

Detect it Easy

Purpose

- Type
- Packer
- Linker
- Entropy


Entropy
Offset: 0
Size: 112640
Reload
Save diagram
Curve
Histogram
Bytes
8
7
6
5
4
3
2
0
20,000
40,000
60,000
80,000
100,000
120,000
Offset:
Size:
OK
Organize
Open
Share with
New folder
Name
Date modified
Type
Size
Detect It Easy 2.03
...
Scan
Scripts
Plugins
Log
Entropy
FLC
S
H
Export
Import
Resource
Overlay
.NET
PE
Options
About
Exit
Signatures
Info
Scan

Entropy(bits/byte): 6.41483 | 80% | not packed

PE Header("2.43607")
Section0(".text")("6.66223")
Section1(".rdata")("6.73562")
Section2(".data")("3.6043")
Section3(".reloc")("2.60349")

100%

de4dot-net35 | 6/3/2019 2:45 PM | File folder

File name: C:/Users/malware/Desktop/sofacy-packed.bin

Type: PE | Size: 112640

EntryPoint: 00001cb3 | ImageBase: 00400000

NumberOfSections: 0004 | SizeOfImage: 00021000

compiler: Microsoft Visual C/C++(2012 update 3)[-] S ?
linker: Microsoft Linker(11.0)[EXE32] S ?

Detect It Easy

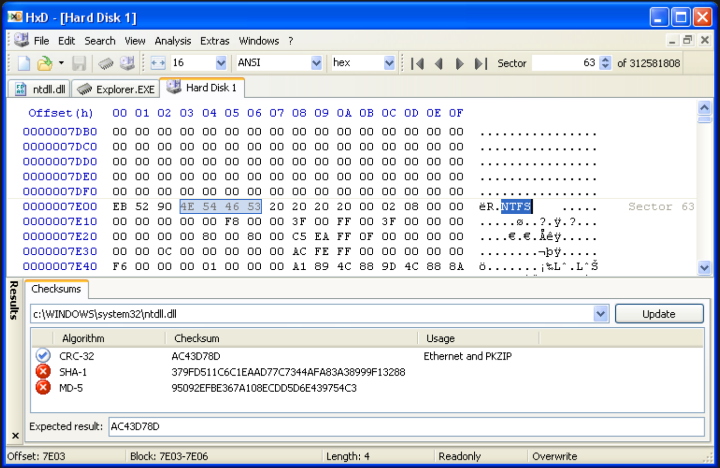
100% | 422 ms

Shortcut Date modified: 6/3/2019 4:57 PM | Date created: 6/3/2019 4:57 PM
Shortcut | Size: 13.2 KB

HxD

Purpose

- Modify Dumps
- Read Memory
- Determine File Type


HxD - [Hard Disk 1]
File Edit Search View Analysis Extras Windows ?
16
ANSI
hex
Sector
63
of 312581808
ntdll.dll
Explorer.EXE
Hard Disk 1
Results
Checksums
Update

```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

0000007DB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000007DC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000007DD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000007DE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000007DF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000007E00 EB 52 90 4E 54 46 53 20 20 20 20 00 02 08 00 00  ëR.NTFS    .....      Sector 63
0000007E10 00 00 00 00 00 F8 00 00 3F 00 FF 00 3F 00 00 00  .....ø..?.ÿ.?...
0000007E20 00 00 00 00 80 00 80 00 C5 EA FF 0F 00 00 00 00  ....€.€.Åêÿ.....
0000007E30 00 00 0C 00 00 00 00 00 AC FE FF 00 00 00 00 00  ........¬þÿ.....
0000007E40 F6 00 00 00 01 00 00 00 A1 89 4C 88 9D 4C 88 8A  ö.......¡‰Lˆ.LˆŠ
```

c:\WINDOWS\system32\ntdll.dll

	Algorithm	Checksum	Usage
✓	CRC-32	AC43D78D	Ethernet and PKZIP
✗	SHA-1	379FD511C6C1EAAD77C7344AFA83A38999F13288	
✗	MD-5	95092EFBE367A108ECDD5D6E439754C3	

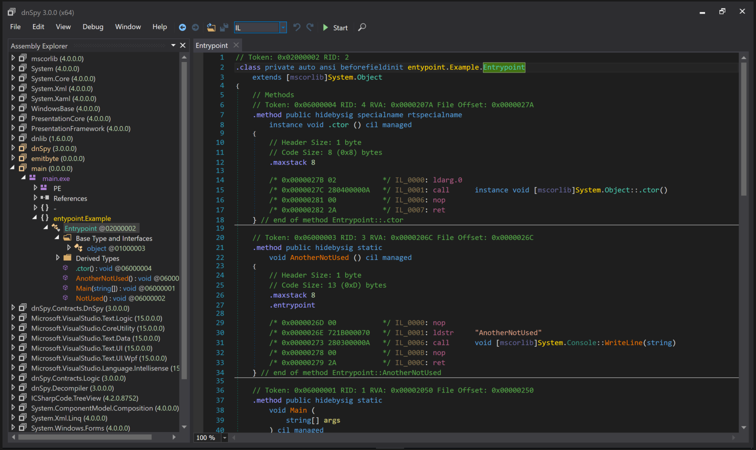
Expected result: AC43D78D

Offset: 7E03 | Block: 7E03-7E06 | Length: 4 | Readonly | Overwrite

DnSpy

Purpose

- Code View
- Debugging
- Unpacking


dnSpy 3.0.0 (x64)
File
Edit
View
Debug
Window
Help
IL
Start
Assembly Explorer
mscorlib (4.0.0.0)
System (4.0.0.0)
System.Core (4.0.0.0)
System.Xml (4.0.0.0)
System.Xaml (4.0.0.0)
WindowsBase (4.0.0.0)
PresentationCore (4.0.0.0)
PresentationFramework (4.0.0.0)
dnlib (1.6.0.0)
dnSpy (3.0.0.0)
emitbyte (0.0.0.0)
main (0.0.0.0)
main.exe
PE
References
{ } -
{ } entypoint.Example
Entrypoint @02000002
Base Type and Interfaces
object @01000003
Derived Types
.ctor() : void @06000004
AnotherNotUsed() : void @06000
Main(string[]) : void @06000001
NotUsed() : void @06000002
dnSpy.Contracts.DnSpy (3.0.0.0)
Microsoft.VisualStudio.Text.Logic (15.0.0.0)
Microsoft.VisualStudio.CoreUtility (15.0.0.0)
Microsoft.VisualStudio.Text.Data (15.0.0.0)
Microsoft.VisualStudio.Text.UI (15.0.0.0)
Microsoft.VisualStudio.Text.UI.Wpf (15.0.0.0)
Microsoft.VisualStudio.Language.Intellisense (15
dnSpy.Contracts.Logic (3.0.0.0)
dnSpy.Decompiler (3.0.0.0)
ICSharpCode.TreeView (4.2.0.8752)
System.ComponentModel.Composition (4.0.0.0)
System.Xml.Linq (4.0.0.0)
System.Windows.Forms (4.0.0.0)
Entrypoint
100 %

```
// Token: 0x02000002 RID: 2
.class private auto ansi beforefieldinit entypoint.Example.Entrypoint
    extends [mscorlib]System.Object
{
    // Methods
    // Token: 0x06000004 RID: 4 RVA: 0x0000207A File Offset: 0x0000027A
    .method public hidebysig specialname rtspecialname
        instance void .ctor () cil managed
    {
        // Header Size: 1 byte
        // Code Size: 8 (0x8) bytes
        .maxstack 8

        /* 0x0000027B 02           */ IL_0000: ldarg.0
        /* 0x0000027C 280400000A   */ IL_0001: call       instance void [mscorlib]System.Object::.ctor()
        /* 0x00000281 00           */ IL_0006: nop
        /* 0x00000282 2A           */ IL_0007: ret
    } // end of method Entrypoint::.ctor

    // Token: 0x06000003 RID: 3 RVA: 0x0000206C File Offset: 0x0000026C
    .method public hidebysig static
        void AnotherNotUsed () cil managed
    {
        // Header Size: 1 byte
        // Code Size: 13 (0xD) bytes
        .maxstack 8
        .entrypoint

        /* 0x0000026D 00           */ IL_0000: nop
        /* 0x0000026E 721B000070   */ IL_0001: ldstr      "AnotherNotUsed"
        /* 0x00000273 280300000A   */ IL_0006: call       void [mscorlib]System.Console::WriteLine(string)
        /* 0x00000278 00           */ IL_000B: nop
        /* 0x00000279 2A           */ IL_000C: ret
    } // end of method Entrypoint::AnotherNotUsed

    // Token: 0x06000001 RID: 1 RVA: 0x00002050 File Offset: 0x00000250
    .method public hidebysig static
        void Main (
            string[] args
        ) cil managed
```

Useful Linux Commads

subtitle

terminal

```
malware@work ~$ file sample.bin
sample.bin: PE32 executable (GUI) Intel 80386, for MS Windows
malware@work ~$ exiftool sample.bin > metadata.log
malware@work ~$ hexdump -C -n 128 sample.bin | less
malware@work ~$ VBoxManage list vms
”win10” {53014b4f-4c94-49b0-9036-818b84a192c9}
”win7” {942cde2e-6a84-4edc-b98a-d7326b4662ee}
malware@work ~$ VBoxManage startvm win7
malware@work ~$
```

Injection Techniques

When you're afraid of **Injection** 😬

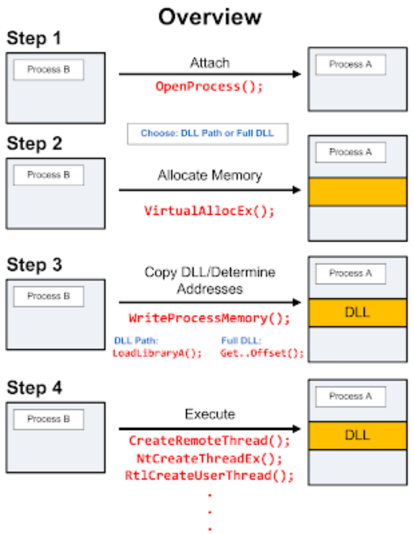
But your **crush** is watching 😂

That relatable moment when your crush comes to watch your doctor's appointment 😂😂😂😂😂😂😂

DLL Injection

Test

- Get Handle to Target Process
- Allocate Memory
- Write Memory
- Execute by use of Remote Thread


Overview
Step 1
Process B
Attach
OpenProcess();
Process A
Choose: DLL Path or Full DLL
Step 2
Process B
Allocate Memory
VirtualAllocEx();
Process A
Step 3
Process B
Copy DLL/Determine Addresses
WriteProcessMemory();
DLL Path:
LoadLibraryA();
Full DLL:
Get..Offset();
Process A
DLL
Step 4
Process B
Execute
CreateRemoteThread();
NtCreateThreadEx();
RtlCreateUserThread();
Process A
DLL

PE (Portable Executable) Injection

Placeholder