



De-Confused-Ex

Reversing ConfuserEx Obfuscated Binaries

Presenter: Shannon Lucas

Agenda

- Confused
- Dumping Modules
- Header Reconstruction
- De-obfuscate
- Reflective Decryption
- Perseverance
- Issues

Confused



Confused

Big shout out to Matt Graeber for explaining this black magic to me



Dumping Modules

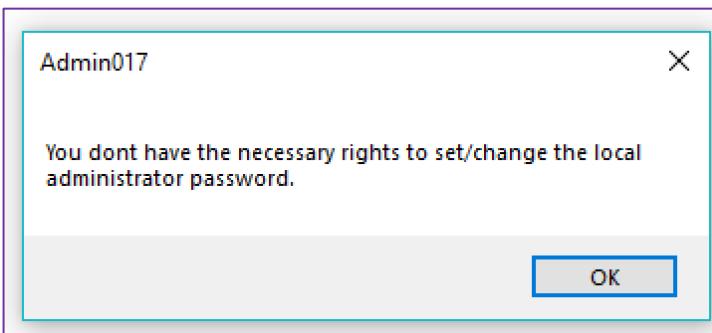
- Use Procmon to determine dlls that get loaded
- Load up executable with Windbg(x86) (Binary is x86)
 - Or load binary with PowerShell and attach to PowerShell process with Windbg
- Break on a function that executes after Confuser has unpacked
- Dump module from memory
- Header Reconstruction

Dumping Modules

Time of Day	Process Name	PID	Operation	Path	Result	Detail
1:19:20:7260007 PM	Admin017.exe	1280	CreateFile	C:\Windows\SysWOW64\CoreUIComponents.dll	SUCCESS	Desired Access: Read Data/List Directory, Execute/Traverse, Synchronize, Disposition: Open, Options: Synchronous IO Non-Alert, Non-Directory File, ...
1:19:20:7260428 PM	Admin017.exe	1280	CreateFileMapping	C:\Windows\SysWOW64\CoreUIComponents.dll	FILE LOCKED WITH O...	SyncType: SyncTypeCreateSection, PageProtection: PAGE_EXECUTE_READ
1:19:20:7260579 PM	Admin017.exe	1280	CreateFileMapping	C:\Windows\SysWOW64\CoreUIComponents.dll	SUCCESS	SyncType: SyncTypeOther
1:19:20:7261211 PM	Admin017.exe	1280	CreateFile	C:\Windows\SysWOW64\CoreMessaging.dll	SUCCESS	Desired Access: Read Data/List Directory, Execute/Traverse, Synchronize, Disposition: Open, Options: Synchronous IO Non-Alert, Non-Directory File, ...
1:19:20:7261607 PM	Admin017.exe	1280	CreateFileMapping	C:\Windows\SysWOW64\CoreMessaging.dll	FILE LOCKED WITH O...	SyncType: SyncTypeCreateSection, PageProtection: PAGE_EXECUTE_READPAGE_NOCACHE
1:19:20:7261831 PM	Admin017.exe	1280	CreateFileMapping	C:\Windows\SysWOW64\CoreMessaging.dll	SUCCESS	SyncType: SyncTypeOther
1:19:20:7264426 PM	Admin017.exe	1280	CloseFile	C:\Windows\SysWOW64\CoreUIComponents.dll	SUCCESS	
1:19:20:7264654 PM	Admin017.exe	1280	CloseFile	C:\Windows\SysWOW64\CoreMessaging.dll	SUCCESS	
1:19:20:7273361 PM	Admin017.exe	1280	CreateFile	C:\Windows\SysWOW64\WinTypes.dll	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, ShareMode: Read, Write, Delete, AllocationSize: n/...
1:19:20:7273410 PM	Admin017.exe	1280	CreateFile	C:\Windows\SysWOW64\vtmarta.dll	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, ShareMode: Read, Write, Delete, AllocationSize: n/...
1:19:20:7287161 PM	Admin017.exe	1280	QueryBasicInformationFile	C:\Windows\SysWOW64\vtmarta.dll	SUCCESS	CreationTime: 4/11/2018 7:34:50 PM, LastAccessTime: 9/14/2019 1:18:40 PM, LastWriteTime: 4/11/2018 7:34:50 PM, ChangeTime: 12/4/2018 1:4...
1:19:20:7287330 PM	Admin017.exe	1280	CloseFile	C:\Windows\SysWOW64\vtmarta.dll	SUCCESS	
1:19:20:7295165 PM	Admin017.exe	1280	CreateFile	C:\Windows\SysWOW64\WinTypes.dll	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, ShareMode: Read, Write, Delete, AllocationSize: n/...
1:19:20:7295952 PM	Admin017.exe	1280	CreateFile	C:\Windows\SysWOW64\WinTypes.dll	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, ShareMode: Read, Write, Delete, AllocationSize: n/...
1:19:20:7296178 PM	Admin017.exe	1280	CreateFile	C:\Windows\SysWOW64\vtmarta.dll	SUCCESS	Desired Access: Read Data/List Directory, Execute/Traverse, Synchronize, Disposition: Open, Options: Synchronous IO Non-Alert, Non-Directory File, ...
1:19:20:7296628 PM	Admin017.exe	1280	QueryBasicInformationFile	C:\Windows\SysWOW64\WinTypes.dll	SUCCESS	CreationTime: 2/25/2019 1:36:21 PM, LastAccessTime: 9/14/2019 1:18:41 PM, LastWriteTime: 1/1/2019 2:37:58 AM, ChangeTime: 8/14/2019 10:2...
1:19:20:7296738 PM	Admin017.exe	1280	CloseFile	C:\Windows\SysWOW64\vtmarta.dll	SUCCESS	
1:19:20:7296889 PM	Admin017.exe	1280	CreateFileMapping	C:\Windows\SysWOW64\vtmarta.dll	FILE LOCKED WITH O...	SyncType: SyncTypeCreateSection, PageProtection: PAGE_EXECUTE_READ
1:19:20:7296934 PM	Admin017.exe	1280	QueryBasicInformationFile	C:\Windows\SysWOW64\vtmarta.dll	SUCCESS	CreationTime: 2/25/2019 1:36:21 PM, LastAccessTime: 9/14/2019 1:18:41 PM, LastWriteTime: 1/1/2019 2:37:58 AM, ChangeTime: 8/14/2019 10:2...
1:19:20:7296974 PM	Admin017.exe	1280	CloseFile	C:\Windows\SysWOW64\vtmarta.dll	SUCCESS	
1:19:20:7297043 PM	Admin017.exe	1280	CreateFileMapping	C:\Windows\SysWOW64\vtmarta.dll	SUCCESS	SyncType: SyncTypeOther
1:19:20:7298396 PM	Admin017.exe	1280	CreateFile	C:\Windows\SysWOW64\WinTypes.dll	SUCCESS	Desired Access: Read Data/List Directory, Execute/Traverse, Synchronize, Disposition: Open, Options: Synchronous IO Non-Alert, Non-Directory File, ...
1:19:20:7299285 PM	Admin017.exe	1280	CreateFileMapping	C:\Windows\SysWOW64\WinTypes.dll	FILE LOCKED WITH O...	SyncType: SyncTypeCreateSection, PageProtection: PAGE_EXECUTE_READPAGE_NOCACHE
1:19:20:7299635 PM	Admin017.exe	1280	CloseFile	C:\Windows\SysWOW64\vtmarta.dll	SUCCESS	SyncType: SyncTypeOther
1:19:20:7299672 PM	Admin017.exe	1280	CreateFileMapping	C:\Windows\SysWOW64\WinTypes.dll	SUCCESS	SyncType: SyncTypeOther
1:19:20:7302974 PM	Admin017.exe	1280	QueryBasicInformationFile	C:\Windows\SysWOW64\WinTypes.dll	SUCCESS	CreationTime: 2/25/2019 1:36:21 PM, LastAccessTime: 9/14/2019 1:18:41 PM, LastWriteTime: 1/1/2019 2:37:58 AM, ChangeTime: 8/14/2019 10:2...
1:19:20:7303256 PM	Admin017.exe	1280	CloseFile	C:\Windows\SysWOW64\WinTypes.dll	SUCCESS	
1:19:20:7310012 PM	Admin017.exe	1280	CreateFile	C:\Windows\SysWOW64\vtmarta.dll	SUCCESS	
1:19:20:7318201 PM	Admin017.exe	1280	CreateFile	C:\Windows\SysWOW64\vtmarta.dll	SUCCESS	Desired Access: Read Control, Disposition: Open, Options: , Attributes: n/a, ShareMode: Read, Delete, AllocationSize: n/a, OpenResult: Opened
1:19:20:7320950 PM	Admin017.exe	1280	QuerySecurityFile	C:\Windows\SysWOW64\vtmarta.dll	BUFFER OVERFLOW	Information: Owner
1:19:20:7321380 PM	Admin017.exe	1280	QuerySecurityFile	C:\Windows\SysWOW64\vtmarta.dll	SUCCESS	Information: Owner
1:19:20:7321687 PM	Admin017.exe	1280	CloseFile	C:\Windows\SysWOW64\vtmarta.dll	SUCCESS	
1:19:20:7329563 PM	Admin017.exe	1280	CreateFile	C:\Windows\SysWOW64\CoreMessaging.dll	SUCCESS	Desired Access: Read Control, Disposition: Open, Options: , Attributes: n/a, ShareMode: Read, Delete, AllocationSize: n/a, OpenResult: Opened
1:19:20:7331032 PM	Admin017.exe	1280	QuerySecurityFile	C:\Windows\SysWOW64\CoreMessaging.dll	BUFFER OVERFLOW	Information: Owner
1:19:20:7331318 PM	Admin017.exe	1280	QuerySecurityFile	C:\Windows\SysWOW64\CoreMessaging.dll	SUCCESS	Information: Owner
1:19:20:7331504 PM	Admin017.exe	1280	CloseFile	C:\Windows\SysWOW64\CoreMessaging.dll	SUCCESS	
1:19:20:7337002 PM	Admin017.exe	1280	CreateFile	C:\Windows\SysWOW64\WinTypes.dll	SUCCESS	Desired Access: Read Control, Disposition: Open, Options: , Attributes: n/a, ShareMode: Read, Delete, AllocationSize: n/a, OpenResult: Opened
1:19:20:7338578 PM	Admin017.exe	1280	QuerySecurityFile	C:\Windows\SysWOW64\WinTypes.dll	BUFFER OVERFLOW	Information: Owner
1:19:20:7338729 PM	Admin017.exe	1280	QuerySecurityFile	C:\Windows\SysWOW64\WinTypes.dll	SUCCESS	Information: Owner
1:19:20:7338864 PM	Admin017.exe	1280	CloseFile	C:\Windows\SysWOW64\WinTypes.dll	SUCCESS	
1:19:20:7341793 PM	Admin017.exe	1280	CreateFile	C:\Windows\SysWOW64\CoreUIComponents.dll	SUCCESS	Desired Access: Read Control, Disposition: Open, Options: , Attributes: n/a, ShareMode: Read, Delete, AllocationSize: n/a, OpenResult: Opened
1:19:20:7342195 PM	Admin017.exe	1280	QuerySecurityFile	C:\Windows\SysWOW64\CoreUIComponents.dll	BUFFER OVERFLOW	Information: Owner
1:19:20:7342316 PM	Admin017.exe	1280	QuerySecurityFile	C:\Windows\SysWOW64\CoreUIComponents.dll	SUCCESS	Information: Owner
1:19:20:7342428 PM	Admin017.exe	1280	CloseFile	C:\Windows\SysWOW64\CoreUIComponents.dll	SUCCESS	
1:19:20:7344258 PM	Admin017.exe	1280	CreateFile	C:\Windows\SysWOW64\TextInputFramework.dll	SUCCESS	Desired Access: Read Control, Disposition: Open, Options: , Attributes: n/a, ShareMode: Read, Delete, AllocationSize: n/a, OpenResult: Opened
1:19:20:7344710 PM	Admin017.exe	1280	QuerySecurityFile	C:\Windows\SysWOW64\TextInputFramework.dll	BUFFER OVERFLOW	Information: Owner
1:19:20:7344830 PM	Admin017.exe	1280	QuerySecurityFile	C:\Windows\SysWOW64\TextInputFramework.dll	SUCCESS	Information: Owner
1:19:20:7344943 PM	Admin017.exe	1280	CloseFile	C:\Windows\SysWOW64\TextInputFramework.dll	SUCCESS	
1:19:20:7349217 PM	Admin017.exe	1280	CreateFile	C:\Windows\SysWOW64\KernelBase.dll	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, ShareMode: Read, Write, Delete, AllocationSize: n/...
1:19:20:7349757 PM	Admin017.exe	1280	QueryBasicInformationFile	C:\Windows\SysWOW64\KernelBase.dll	SUCCESS	CreationTime: 7/9/2019 6:32:03 PM, LastAccessTime: 9/5/2019 3:36:45 PM, LastWriteTime: 7/4/2019 12:42:15 AM, ChangeTime: 8/14/2019 10:23...
1:19:20:7349916 PM	Admin017.exe	1280	CloseFile	C:\Windows\SysWOW64\KernelBase.dll	SUCCESS	
1:19:20:7352263 PM	Admin017.exe	1280	QueryNameInformationFile	C:\Windows\SysWOW64\WinTypes.dll	SUCCESS	Name: \Windows\SysWOW64\WinTypes.dll
1:19:20:7353959 PM	Admin017.exe	1280	QueryNameInformationFile	C:\Windows\SysWOW64\CoreUIComponents.dll	SUCCESS	Name: \Windows\SysWOW64\CoreUIComponents.dll
1:19:20:7356358 PM	Admin017.exe	1280	QueryNameInformationFile	C:\Windows\SysWOW64\TextInputFramework.dll	SUCCESS	Name: \Windows\SysWOW64\TextInputFramework.dll
1:19:23:5524030 PM	Admin017.exe	1280	CreateFile	C:\Windows\SysWOW64\user32.dll	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, ShareMode: Read, Write, Delete, AllocationSize: n/...
1:19:23:5524270 PM	Admin017.exe	1280	QueryBasicInformationFile	C:\Windows\SysWOW64\user32.dll	SUCCESS	CreationTime: 12/3/2018 12:06:50 PM, LastAccessTime: 9/5/2019 3:36:45 PM, LastWriteTime: 10/21/2018 7:37:42 AM, ChangeTime: 8/14/2019 1...
1:19:23:5524868 PM	Admin017.exe	1280	CloseFile	C:\Windows\SysWOW64\user32.dll	SUCCESS	

Dumping Modules

- Binary pops MessageBox if it's not run as admin.
- Creates Logs folder and log file
- Log file specifies App Process Name, Computer Name, and whether it successfully set the Admin password.



The screenshot shows a Windows File Explorer window with the path 'System (C:) > CMDMGMT > LOGS' highlighted. Inside the 'LOGS' folder, there is a file named 'Admin007.log'. This file is opened in a Notepad window. The log content is as follows:

```
2019-Sep-16 11:42:59: Running in set mode.  
2019-Sep-16 11:42:59: Set administrator password successful  
2019-Sep-16 02:44:02:  
2019-Sep-16 02:44:02: [Begin Admin017]  
2019-Sep-16 02:44:02:  
2019-Sep-16 02:44:02: App Source Path: C:\Users\tester\Desktop\TradeCraft  
2019-Sep-16 02:44:02: App Process Name: Admin017  
2019-Sep-16 02:44:02: App Process Name Length: 1  
2019-Sep-16 02:44:02: Computer Name: WIN-PICKLE  
2019-Sep-16 02:44:02: App engine version: 1.0.0.0  
2019-Sep-16 02:45:16:  
2019-Sep-16 02:45:16: [Begin Admin017]  
2019-Sep-16 02:45:16:  
2019-Sep-16 02:45:16: App Source Path: C:\Users\tester\Desktop\TradeCraft  
2019-Sep-16 02:45:16: App Process Name: Admin017  
2019-Sep-16 02:45:16: App Process Name Length: 1  
2019-Sep-16 02:45:16: Computer Name: WIN-PICKLE  
2019-Sep-16 02:45:16: App engine version: 1.0.0.0  
2019-Sep-16 02:54:22:  
2019-Sep-16 02:54:22: [Begin Admin017]  
2019-Sep-16 02:54:22:  
2019-Sep-16 02:54:22: App Source Path: C:\Users\tester\Desktop\TradeCraft  
2019-Sep-16 02:54:22: App Process Name: ORIGINAL_Admin017  
2019-Sep-16 02:54:22: App Process Name Length: 1  
2019-Sep-16 02:54:22: Computer Name: WIN-PICKLE  
2019-Sep-16 02:54:22: App engine version: 1.0.0.0  
2019-Sep-16 02:54:50: Insufficient privileges to set the local admin password  
2019-Sep-16 02:54:50: [End]  
2019-Sep-16 02:54:50:
```

Dumping Modules

- Break on user32 module load
- Continue
- Set breakpoint for MessageBoxW call from user32 module
- Continue

```
Command - C:\Users\tester\Desktop\TradeCraft\SetADAttributes.exe - WinDbg:6.3.9600.17298 X86

Microsoft (R) Windows Debugger Version 6.3.9600.17298 X86
Copyright (c) Microsoft Corporation. All rights reserved.

CommandLine: C:\Users\tester\Desktop\TradeCraft\SetADAttributes.exe
Symbol search path is: *** Invalid ***
*****
* Symbol loading may be unreliable without a symbol search path. *
* Use .sympath to have the debugger choose a symbol path. *
* After setting your symbol path, use .reload to refresh symbol locations. *
*****
Executable search path is:
ModLoad: 00d50000 00d6c000  image00d50000
ModLoad: 77d50000 77ee0000  ntdll.dll
ModLoad: 71db0000 71e05000  C:\WINDOWS\SysWOW64\MSCOREE.DLL
ModLoad: 77870000 77950000  C:\WINDOWS\SysWOW64\KERNEL32.dll
ModLoad: 76040000 76224000  C:\WINDOWS\SysWOW64\KERNELBASE.dll
(194c.2264): Break instruction exception - code 80000003 (first chance)
*** ERROR: Symbol file could not be found. Defaulted to export symbols for ntdll.dll -
eax=00000000 ebx=00000010 ecx=aecd0000 edx=00000000 esi=00f54000 edi=77d5671c
eip=77df7d89 esp=010ff488 ebp=010ff4b4 iopl=0 nv up ei pl nz na pe nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00000246
ntdll!LdrInitShimEngineDynamic+0x6e9:
77df7d89 cc int 3
0:000> sxe ld user32
0:000> g
ModLoad: 747e0000 7496d000  C:\WINDOWS\SysWOW64\USER32.dll
eax=00000000 ebx=00800000 ecx=00000000 edx=00000000 esi=0128b758 edi=0128b6b8
eip=77dbab5c esp=010fd574 ebp=010fd5c0 iopl=0 nv up ei pl nz na pe nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00000206
ntdll!NtMapViewOfSection+0xc:
77dbab5c c22800 ret 28h
0:000> bp user32!MessageBoxW
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\SysWOW64\USER32.dll -
0:000> g
(194c.2264): Unknown exception - code 04242420 (first chance)
(194c.2264): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=015ab960 ebx=012d61e8 ecx=00000000 edx=0322542c esi=5f105135 edi=010fe924
eip=015ab964 esp=010fe714 ebp=010fe934 iopl=0 nv up ei pl nz na pe nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00210246
015ab964 3909 cmp dword ptr [ecx],ecx ds:002b:00000000=???????
```

Dumping Modules

- Load clr
- DumpDomain (list loaded assemblies in AppDomain with object address)

```
Command - C:\Users\tester\Desktop\TradeCraft\SetADAttributes.exe - WinDbg:6.3.9600.17298 X86
PDB symbol for clr.dll not loaded
c0000005 Exception in C:\Windows\Microsoft.NET\Framework\v4.0.30319\sos.DumpDomain debugger extension.
  PC: 689612a3  VA: 00000000  R/W: 0  Parameter: 00000000
0:000> .load C:\Windows\Microsoft.NET\Framework\v4.0.30319\clr
0:000> !DumpDomain

System Domain: 70f26740
LowFrequencyHeap: 70f26a64
HighFrequencyHeap: 70f26ab0
StubHeap: 70f26afc
Stage: OPEN
Name: None

Shared Domain: 70f263f0
LowFrequencyHeap: 70f26a64
HighFrequencyHeap: 70f26ab0
StubHeap: 70f26afc
Stage: OPEN
Name: None
Assembly: 012ef660 [C:\WINDOWS\Microsoft.Net\assembly\GAC_32\mscorlib\v4.0_4.0.0.0_b77a5c561934e089\mscorlib.dll]
ClassLoader: 012ef728
  Module Name
6bcd1000 C:\WINDOWS\Microsoft.Net\assembly\GAC_32\mscorlib\v4.0_4.0.0.0_b77a5c561934e089\mscorlib.dll

Domain 1: 0129f4e8
LowFrequencyHeap: 0129f954
HighFrequencyHeap: 0129f9a0
StubHeap: 0129f9ec
Stage: OPEN
SecurityDescriptor: 012a0440
Name: SetADAttributes.exe
Assembly: 012ef660 [C:\WINDOWS\Microsoft.Net\assembly\GAC_32\mscorlib\v4.0_4.0.0.0_b77a5c561934e089\mscorlib.dll]
ClassLoader: 012ef728
SecurityDescriptor: 012ebf30
  Module Name
6bcd1000 C:\WINDOWS\Microsoft.Net\assembly\GAC_32\mscorlib\v4.0_4.0.0.0_b77a5c561934e089\mscorlib.dll

Assembly: 012f7410 [C:\Users\tester\Desktop\TradeCraft\SetADAttributes.exe]
ClassLoader: 012f8e58
SecurityDescriptor: 012f8d50
  Module Name
014c403c C:\Users\tester\Desktop\TradeCraft\SetADAttributes.exe

Assembly: 01300238 [dseLyPnkElHJchtABiqILQioasJC, Version=0.0.0.0, Culture=neutral, PublicKeyToken=null]
ClassLoader: 01300300
SecurityDescriptor: 012feef0
  Module Name
014c5fd8 dseLyPnkElHJchtABiqILQioasJC, Version=0.0.0.0, Culture=neutral, PublicKeyToken=null

Assembly: 01306f78 [C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\Microsoft.VisualBasic\v4.0_10.0.0.0_b03f5f7f11d50a3a\Microsoft.VisualBasic.dll]
ClassLoader: 01307040
SecurityDescriptor: 013056d0
```

Dumping Modules

```
Assembly: 0131e8d0 [C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.Xml\v4.0_4.0.0.0__b77a5c561934e089\System.Xml.dll]
ClassLoader: 0131e960
SecurityDescriptor: 0131eed0
Module Name
65d31000 C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.Xml\v4.0_4.0.0.0__b77a5c561934e089\System.Xml.dll

Assembly: 0132cfid0 [C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.Runtime.Remoting\v4.0_4.0.0.0__b77a5c561934e089\System.Runtime.Remot
ClassLoader: 013292d0
SecurityDescriptor: 01328e98
Module Name
5ddd1000 C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.Runtime.Remoting\v4.0_4.0.0.0__b77a5c561934e089\System.Runtime.Remoting.dll

Assembly: 01338d60 [C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.DirectoryServices.AccountManagement\v4.0_4.0.0.0__b77a5c561934e089\S
ClassLoader: 01338e28
SecurityDescriptor: 01338888
Module Name
014cbc0c C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.DirectoryServices.AccountManagement\v4.0_4.0.0.0__b77a5c561934e089\System.DirectorySer
Assembly: 0133c308 [C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.DirectoryServices\v4.0_4.0.0.0__b03f5f7f11d50a3a\System.DirectorySer
ClassLoader: 0133c3d0
SecurityDescriptor: 0133b0b0
Module Name
5f0c1000 C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.DirectoryServices\v4.0_4.0.0.0__b03f5f7f11d50a3a\System.DirectoryServices.dll

0:000> !SaveModule 014c403c C:\Users\tester\Desktop\TradeCraft\SetADAttributes_dumped.exe
5 sections in file
section 0 - VA=2000, VASize=7e78, FileAddr=400, FileSize=8000
section 1 - VA=a000, VASize=bb50, FileAddr=8400, FileSize=bc00
section 2 - VA=16000, VASize=600, FileAddr=14000, FileSize=600
section 3 - VA=18000, VASize=c, FileAddr=14600, FileSize=200
section 4 - VA=1a000, VASize=10, FileAddr=14800, FileSize=200
```

- SaveModule
 - Writes an image, which is loaded in memory at the specified address.
 - !SaveModule <object address> C:\path\to\new\binary.exe

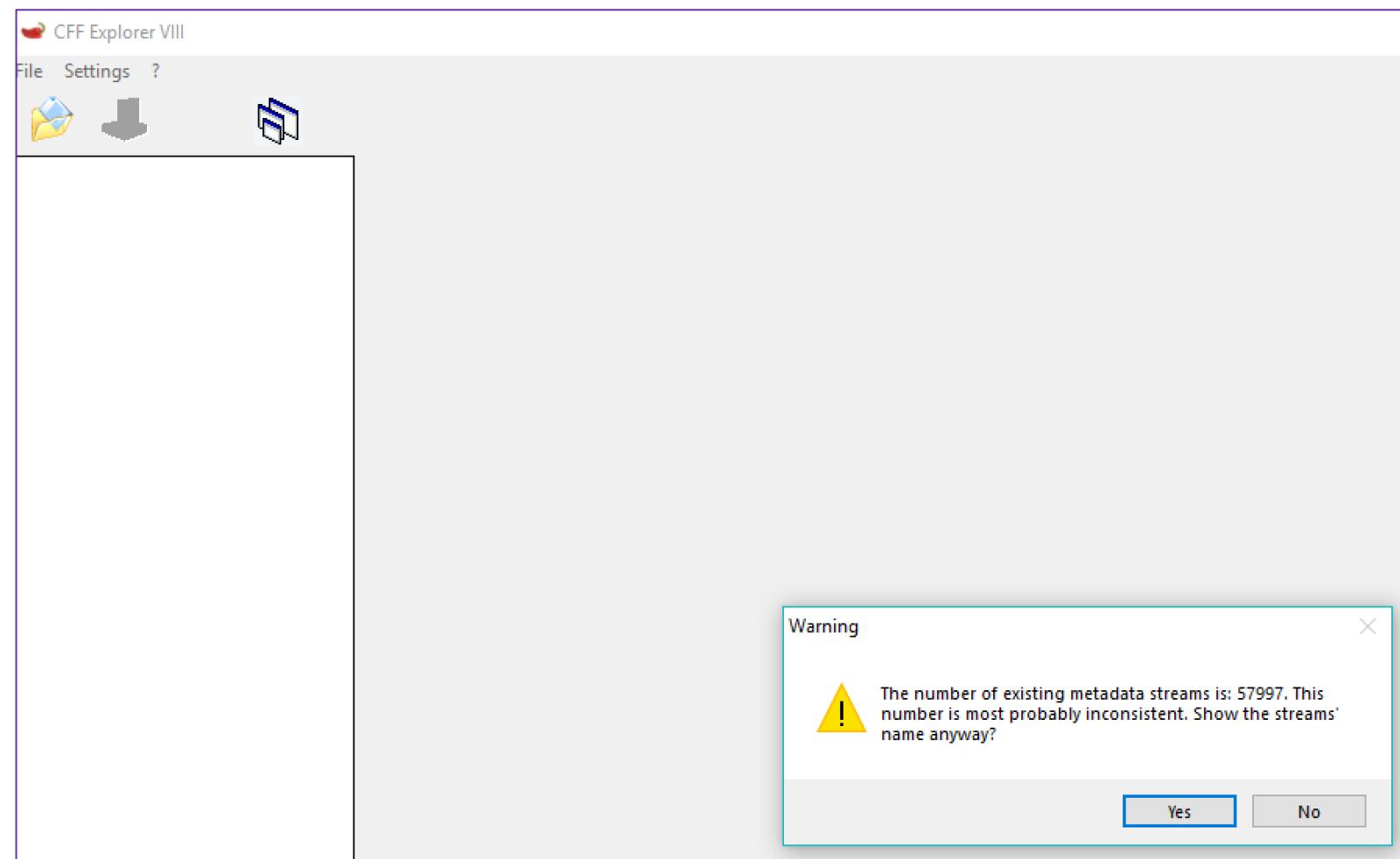
Header Reconstruction

```
C:\Users\tester\Desktop\de4dot-net35>de4dot.exe C:\Users\tester\Desktop\TradeCraft\SetADAttributes_dumped1.exe  
de4dot v3.1.41592.3405 Copyright (C) 2011-2015 de4dot@gmail.com  
Latest version and source code: https://github.com/0xd4d/de4dot  
  
WARNING: The file isn't a .NET PE file: C:\Users\tester\Desktop\TradeCraft\SetADAttributes_dumped1.exe
```



Header Reconstruction

Issue with metadata?!



Header Reconstruction

Missing MetaData blob

Member	Offset	Size	Value	Meaning
cb	00008400	Dword	00000000	
MajorRuntimeVersion	00008404	Word	0000	
MinorRuntimeVersion	00008406	Word	0000	
MetaData RVA	00008408	Dword	00000000	
MetaData Size	0000840C	Dword	00000000	
Flags	00008410	Dword	00020003	Click here
EntryPointToken	00008414	Dword	0600005D	
Resources RVA	00008418	Dword	00000000	
Resources Size	0000841C	Dword	00000000	
StrongNameSignature RVA	00008420	Dword	00000000	
StrongNameSignature Size	00008424	Dword	00000000	
CodeManagerTable RVA	00008428	Dword	00000000	
CodeManagerTable Size	0000842C	Dword	00000000	
VTableFixups RVA	00008430	Dword	00000000	
VTableFixups Size	00008434	Dword	00000000	
ExportAddressTableJumps RVA	00008438	Dword	00000000	
ExportAddressTableJumps Size	0000843C	Dword	00000000	
ManagedNativeHeader RVA	00008440	Dword	00000000	
ManagedNativeHeader Size	00008444	Dword	00000000	

Member	Offset	Size	Value	Meaning
cb	00008400	Dword	00000048	
MajorRuntimeVersion	00008404	Word	0002	
MinorRuntimeVersion	00008406	Word	0005	
MetaData RVA	00008408	Dword	0000A8BC	
MetaData Size	0000840C	Dword	0000B294	
Flags	00008410	Dword	00020003	Click here
EntryPointToken	00008414	Dword	0600005D	
Resources RVA	00008418	Dword	00000000	
Resources Size	0000841C	Dword	00000000	
StrongNameSignature RVA	00008420	Dword	00000000	
StrongNameSignature Size	00008424	Dword	00000000	
CodeManagerTable RVA	00008428	Dword	00000000	
CodeManagerTable Size	0000842C	Dword	00000000	
VTableFixups RVA	00008430	Dword	00000000	
VTableFixups Size	00008434	Dword	00000000	
ExportAddressTableJumps RVA	00008438	Dword	00000000	
ExportAddressTableJumps Size	0000843C	Dword	00000000	
ManagedNativeHeader RVA	00008440	Dword	00000000	
ManagedNativeHeader Size	00008444	Dword	00000000	

Header Reconstruction

CFF Explorer VIII - [SetADAttributes_dumped1.exe]

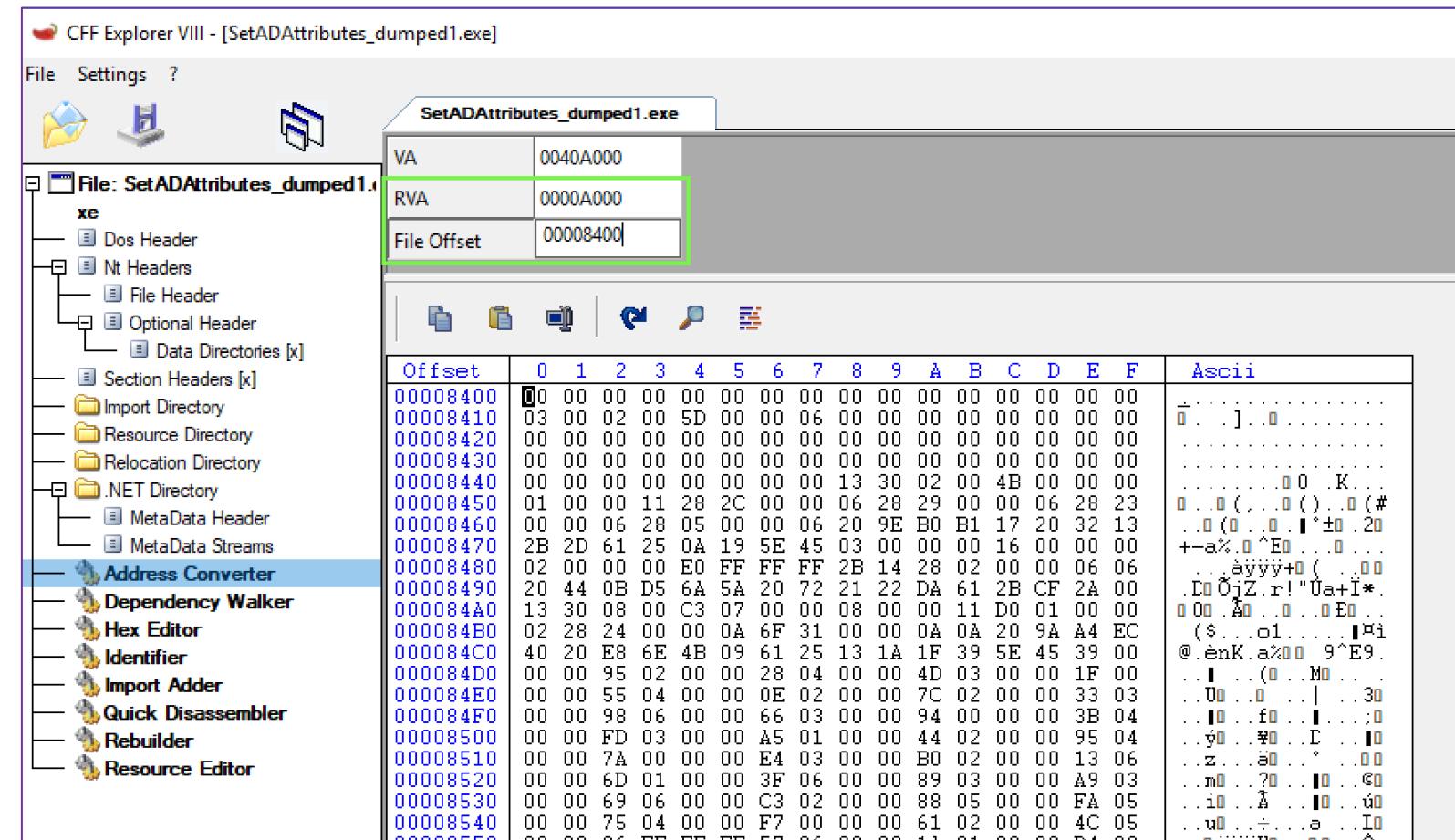
Member	Offset	Size	Value	Section
Export Directory RVA	000000F8	Dword	00000000	
Export Directory Size	000000FC	Dword	00000000	
Import Directory RVA	00000100	Dword	0000A870	
Import Directory Size	00000104	Dword	0000004B	
Resource Directory RVA	00000108	Dword	00016000	
Resource Directory Size	0000010C	Dword	00000600	
Exception Directory RVA	00000110	Dword	00000000	
Exception Directory Size	00000114	Dword	00000000	
Security Directory RVA	00000118	Dword	00014A00	
Security Directory Size	0000011C	Dword	00001C78	
Relocation Directory RVA	00000120	Dword	00018000	
Relocation Directory Size	00000124	Dword	0000000C	
Debug Directory RVA	00000128	Dword	00000000	
Debug Directory Size	0000012C	Dword	00000000	
Architecture Directory RVA	00000130	Dword	00000000	
Architecture Directory Size	00000134	Dword	00000000	
Reserved	00000138	Dword	00000000	
Reserved	0000013C	Dword	00000000	
TLS Directory RVA	00000140	Dword	00000000	
TLS Directory Size	00000144	Dword	00000000	
Configuration Directory RVA	00000148	Dword	00000000	
Configuration Directory Size	0000014C	Dword	00000000	
Bound Import Directory RVA	00000150	Dword	00000000	
Bound Import Directory Size	00000154	Dword	00000000	
Import Address Table Directory ...	00000158	Dword	0001A000	
Import Address Table Directory ...	0000015C	Dword	00000008	
Delay Import Directory RVA	00000160	Dword	00000000	
Delay Import Directory Size	00000164	Dword	00000000	
.NET MetaData Directory RVA	00000168	Dword	0000A000	
.NET MetaData Directory Size	0000016C	Dword	00000048	

CFF Explorer VIII - [SetADAttributes.exe]

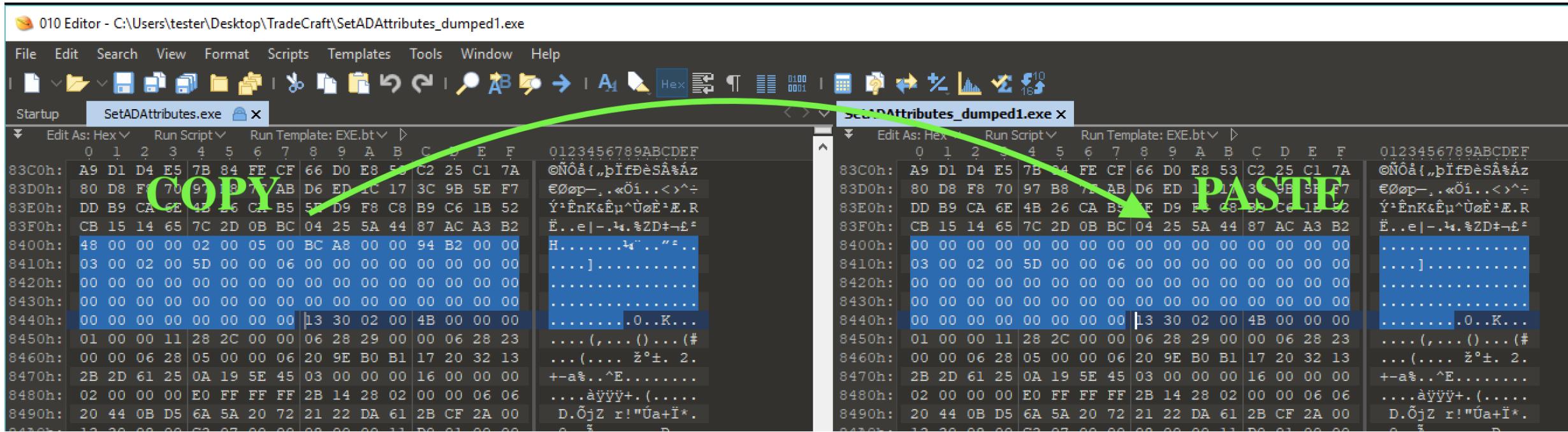
Member	Offset	Size	Value	Section
Export Directory RVA	000000F8	Dword	00000000	
Export Directory Size	000000FC	Dword	00000000	
Import Directory RVA	00000100	Dword	0000A870	.text
Import Directory Size	00000104	Dword	0000004B	
Resource Directory RVA	00000108	Dword	00016000	.rsrc
Resource Directory Size	0000010C	Dword	00000600	
Exception Directory RVA	00000110	Dword	00000000	
Exception Directory Size	00000114	Dword	00000000	
Security Directory RVA	00000118	Dword	00014A00	.text
Security Directory Size	0000011C	Dword	00001C78	
Relocation Directory RVA	00000120	Dword	00018000	.reloc
Relocation Directory Size	00000124	Dword	0000000C	
Debug Directory RVA	00000128	Dword	00000000	
Debug Directory Size	0000012C	Dword	00000000	
Architecture Directory RVA	00000130	Dword	00000000	
Architecture Directory Size	00000134	Dword	00000000	
Reserved	00000138	Dword	00000000	
Reserved	0000013C	Dword	00000000	
TLS Directory RVA	00000140	Dword	00000000	
TLS Directory Size	00000144	Dword	00000000	
Configuration Directory RVA	00000148	Dword	00000000	
Configuration Directory Size	0000014C	Dword	00000000	
Bound Import Directory RVA	00000150	Dword	00000000	
Bound Import Directory Size	00000154	Dword	00000000	
Import Address Table Directory ...	00000158	Dword	0001A000	
Import Address Table Directory ...	0000015C	Dword	00000008	
Delay Import Directory RVA	00000160	Dword	00000000	
Delay Import Directory Size	00000164	Dword	00000000	
.NET MetaData Directory RVA	00000168	Dword	0000A000	text
.NET MetaData Directory Size	0000016C	Dword	00000048	

Header Reconstruction

Use CFF Address Converter to convert RVA to file offset



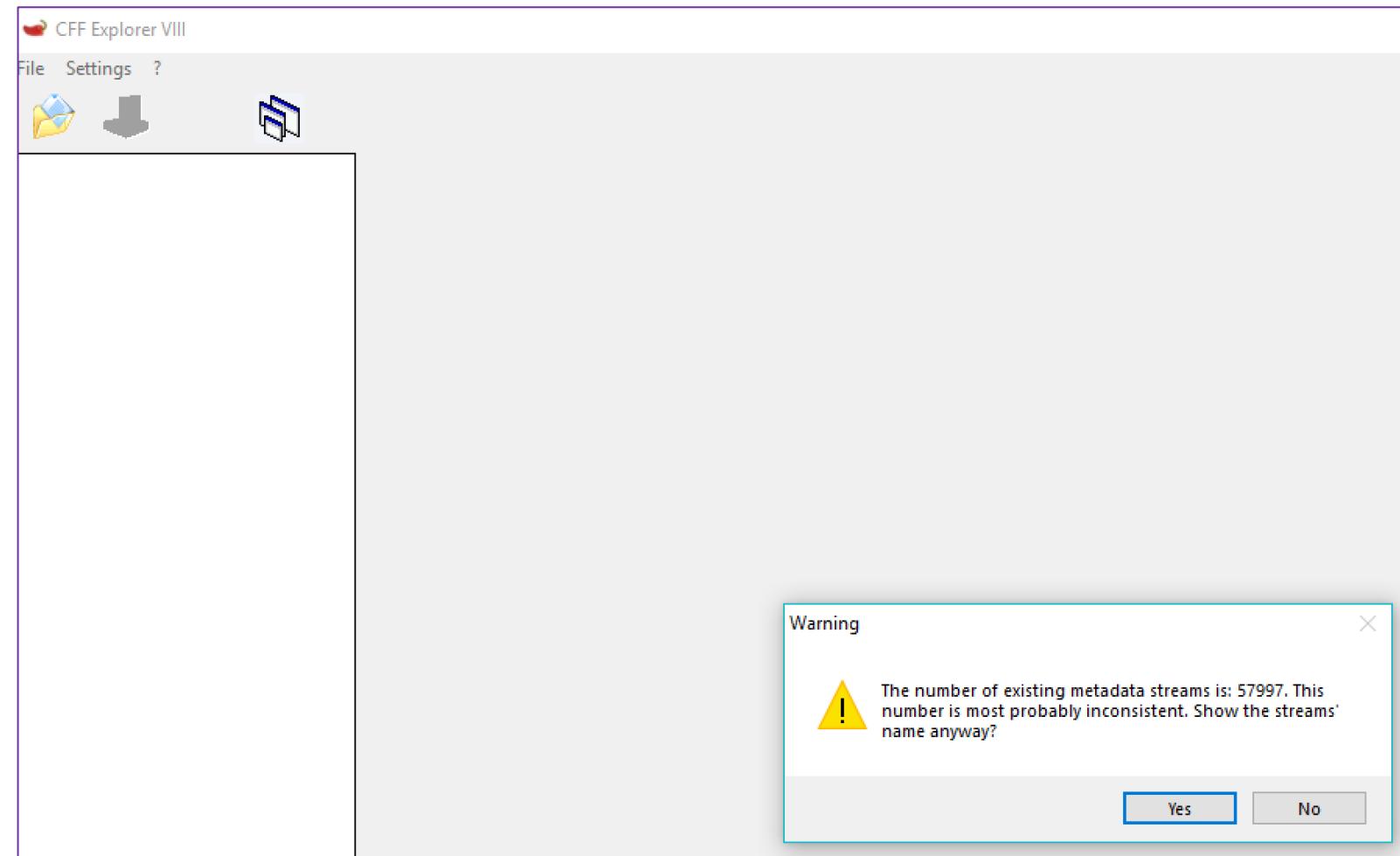
Header Reconstruction



Copy MetaData header and paste in newly dumped binary

Header Reconstruction

Still missing the metadata blob



Header Reconstruction

Member	Offset	Size	Value	Meaning
cb	00008400	Dword	00000048	
MajorRuntimeVersion	00008404	Word	0002	
MinorRuntimeVersion	00008406	Word	0005	
MetaData RVA	00008408	Dword	0000A8BC	
MetaData Size	0000840C	Dword	0000B294	
Flags	00008410	Dword	00020003	Click here
EntryPointToken	00008414	Dword	0600005D	
Resources RVA	00008418	Dword	00000000	
Resources Size	0000841C	Dword	00000000	
StrongNameSignature RVA	00008420	Dword	00000000	
StrongNameSignature Size	00008424	Dword	00000000	
CodeManagerTable RVA	00008428	Dword	00000000	
CodeManagerTable Size	0000842C	Dword	00000000	
VTableFixups RVA	00008430	Dword	00000000	
VTableFixups Size	00008434	Dword	00000000	
ExportAddressTableJumps RVA	00008438	Dword	00000000	
ExportAddressTableJumps Size	0000843C	Dword	00000000	
ManagedNativeHeader RVA	00008440	Dword	00000000	
ManagedNativeHeader Size	00008444	Dword	00000000	

The screenshot shows a debugger interface with two windows. The left window displays a table of file header members. The right window shows the file structure of 'SetADAttributes.exe' with the '.NET Directory' expanded. The 'Tables' section under '.NET Directory' is selected. A context menu is open over the 'Tables' entry, listing tools: Address Converter, Dependency Walker, Hex Editor, Identifier, Import Adder, Quick Disassembler, Rebuilder, and Resource Editor. The 'Tables' entry is highlighted in blue.

Member	Offset	Size	Value	Meaning
cb	00008400	Dword	00000048	
MajorRuntimeVersion	00008404	Word	0002	
MinorRuntimeVersion	00008406	Word	0005	
MetaData RVA	00008408	Dword	0000A8BC	
MetaData Size	0000840C	Dword	0000B294	
Flags	00008410	Dword	00020003	Click here
EntryPointToken	00008414	Dword	0600005D	
Resources RVA	00008418	Dword	00000000	
Resources Size	0000841C	Dword	00000000	
StrongNameSignature RVA	00008420	Dword	00000000	
StrongNameSignature Size	00008424	Dword	00000000	
CodeManagerTable RVA	00008428	Dword	00000000	
CodeManagerTable Size	0000842C	Dword	00000000	
VTableFixups RVA	00008430	Dword	00000000	
VTableFixups Size	00008434	Dword	00000000	
ExportAddressTableJumps RVA	00008438	Dword	00000000	
ExportAddressTableJumps Size	0000843C	Dword	00000000	
ManagedNativeHeader RVA	00008440	Dword	00000000	
ManagedNativeHeader Size	00008444	Dword	00000000	

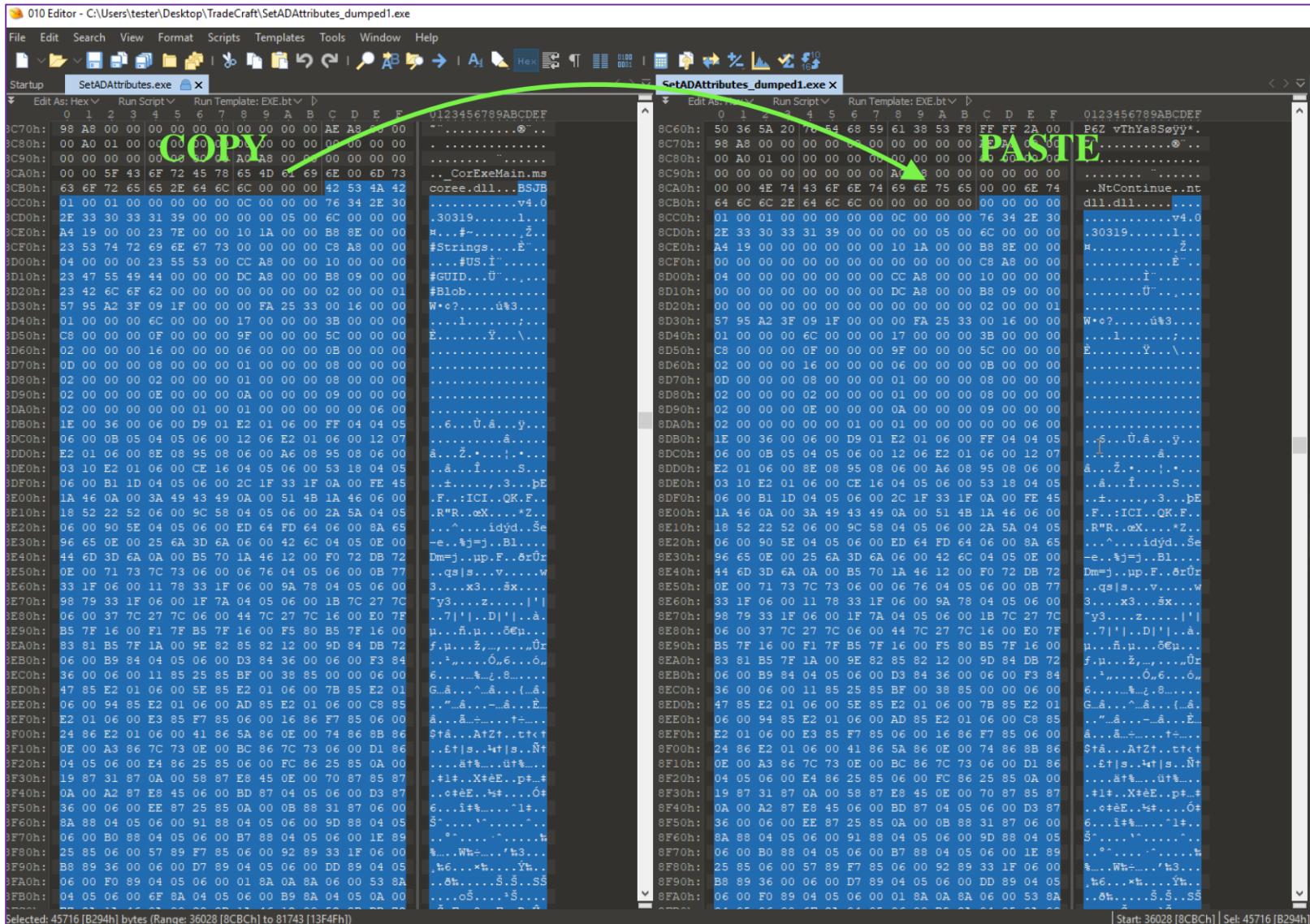
Use MetaData RVA offset and MetaData size to copy MetaData blob

Header Reconstruction

0530h:	00 00 69 06 00 00 C3 02	00 00 00 05 00 00 FA 05	.I....A.... .u.	0530h:	00 00 69 06
8540h:	00 00 75 04 00 00 F7 00	00 00 61 02 00 00 4C 05	.u....÷....a...L.	8540h:	00 00 75 04
8550h:	00 00 06 FF FF FF 57 06	00 00 1A 01 00 00 D4 00	...ÿÿÿW.....Ô.	8550h:	00 00 06 FF
8560h:	00 00 BD 04 00 00 A2 05	00 00 05 00 00 00 E3 05	..‰....¢.....ä.	8560h:	00 00 BD 04
8570h:	00 00 E9 04 00 00 CA 01	00 00 21 02 00 00 AD 04	..é....È....!....-	8570h:	00 00 E9 04
A580h:	00 00 51 00 00 00 F1 01	00 00 BA 00 00 00 6C 05	o ñ ° 1	A580h:	00 00 51 00

In hex editor select range by specifying start offset and size of MetaData blob

Header Reconstruction



Header Reconstruction

The image displays two instances of the CFF Explorer VIII application side-by-side, illustrating the reconstruction of executable file headers.

Left Window (Dumped File):

- File Tree:** Shows the structure of the dumped file, including the main file entry and various header and section components.
- Table:** A grid showing metadata streams with columns: Offset, Size, and Name. The entries are:
 - Dword Dword szAlignedAnsi
 - 0000006C 000019A4 #~
 - 00001A10 00008EB8 #Strings
 - 0000A8C8 00000004 #US
 - 0000A8CC 00000010 #GUID
 - 0000A8DC 000009B8 #Blob

Right Window (Original File):

- File Tree:** Shows the structure of the original file, including the main file entry and detailed headers like Dos Header, Nt Headers, File Header, Optional Header, Data Directories, Section Headers, Import Directory, Resource Directory, Relocation Directory, .NET Directory, and MetaData Header.
- Table:** A grid showing metadata streams with columns: Offset, Size, and Name. The entries are:
 - Dword Dword szAlignedAnsi
 - 0000006C 000019A4 #~
 - 00001A10 00008EB8 #Strings
 - 0000A8C8 00000004 #US
 - 0000A8CC 00000010 #GUID
 - 0000A8DC 000009B8 #Blob
- Tool Bar:** A list of tools available in the application, including Address Converter, Dependency Walker, Hex Editor, Identifier, Import Adder, Quick Disassembler, Rebuilder, and Resource Editor.

De-Obfuscate

```
C:\Users\tester\Desktop\de4dot-net35>de4dot.exe C:\Users\tester\Desktop\TradeCraft\SetADAttributes_dumped.exe

de4dot v3.1.41592.3405 Copyright (C) 2011-2015 de4dot@gmail.com
Latest version and source code: https://github.com/0xd4d/de4dot

Detected Unknown Obfuscator (C:\Users\tester\Desktop\TradeCraft\SetADAttributes_dumped.exe)
Cleaning C:\Users\tester\Desktop\TradeCraft\SetADAttributes_dumped.exe
Renaming all obfuscated symbols
Saving C:\Users\tester\Desktop\TradeCraft\SetADAttributes_dumped-cleaned.exe
ERROR: Error calculating max stack value. If the method's obfuscated, set CilBody.KeepOldMaxStack or MetadataOptions.Flags (KeepOldMaxStack, global option) to ignore this error. Otherwise fix your generated CIL code so it conforms to the E
MA standard.
```

de4dot renames obfuscated symbols and helps with clean up



De-Obfuscate

The screenshot shows the Microsoft Visual Studio interface with the Assembly Explorer and Module views open. The Assembly Explorer on the left lists the project 'SetADAttributes' (1.0.0.0) and its components: PE, References, Resources, and a module named {} containing a class <Module>. The Module view on the right displays the decompiled C# code for this class. The code includes numerous obfuscated method names like smethod_0 through smethod_40, and various numeric offsets and file offsets. The code is heavily annotated with assembly labels (IL_45, IL_129) and switch statements. At the bottom, there are tabs for Locals, Name, Value, and Time.

```
1 using System;
2 using System.Diagnostics;
3 using System.IO;
4 using System.Reflection;
5 using System.Runtime.InteropServices;
6 using System.Text;
7 using System.Threading;
8
9 // Token: 0x02000001 RID: 1
10 internal class <Module>
11 {
12     // Token: 0x06000001 RID: 1 RVA: 0x00002690 File Offset: 0x00000890
13     static <Module>()
14     {
15         <Module>.smethod_40();
16         <Module>.smethod_38();
17         <Module>.smethod_32();
18         <Module>.smethod_2();
19         for (;;)
20         {
21             IL_45:
22             uint num = 397521054u;
23             for (;;)
24             {
25                 uint num2;
26                 switch ((num2 = (num ^ 757797682u)) % 3u)
27                 {
28                     case 1u:
29                         <Module>.smethod_0();
30                         num = (num2 * 1792346948u ^ 3659669874u);
31                         continue;
32                     case 2u:
33                         goto IL_45;
34                 }
35             }
36         }
37     }
38 }
39
40 // Token: 0x06000002 RID: 2 RVA: 0x000026EC File Offset: 0x000008EC
41 private static void smethod_0()
42 {
43     string string_ = <Module>.smethod_33<string>(3901519397u);
44     for (;;)
45     {
46         IL_129:
47         uint num = 3370746283u;
48         for (;;)
49         {
50             uint num2;
51             switch ((num2 = (num ^ 3564460950u)) % 7u)
52             {
53                 case 0u:
54                     smethod_1();
55                     num = (num2 * 1792346948u ^ 3659669874u);
56                     continue;
57                 case 1u:
58                     smethod_2();
59                     num = (num2 * 1792346948u ^ 3659669874u);
60                     continue;
61                 case 2u:
62                     smethod_3();
63                     num = (num2 * 1792346948u ^ 3659669874u);
64                     continue;
65                 case 3u:
66                     smethod_4();
67                     num = (num2 * 1792346948u ^ 3659669874u);
68                     continue;
69                 case 4u:
70                     smethod_5();
71                     num = (num2 * 1792346948u ^ 3659669874u);
72                     continue;
73                 case 5u:
74                     smethod_6();
75                     num = (num2 * 1792346948u ^ 3659669874u);
76                     continue;
77                 case 6u:
78                     smethod_7();
79                     num = (num2 * 1792346948u ^ 3659669874u);
80                     continue;
81             }
82         }
83     }
84 }
```

Reflective Decryption

- Use PowerShell to reflectively load the reconstructed binary
- Reflection examines metadata and can invoke methods in the binary
- Modules.ResolveMethod() returns the method identified by the metadata token ID

The screenshot shows the dnSpy interface with two panes. The left pane, 'Assembly Explorer', lists numerous methods (smethod_16 through smethod_38) with their assembly addresses. The right pane, 'GForm0', displays C# code for a class with string manipulation logic. Below the code is a 'Windows PowerShell' window showing a series of PowerShell commands used to resolve methods and invoke them.

```
dnSpy v5.0.10 (x64)
File Edit View Debug Window Help | G# | Start | 
Assembly Explorer
smethod_16(Thread, object) : void @06000013
smethod_17(int) : void @06000014
smethod_18() : bool @06000015
smethod_19() : bool @06000016
smethod_20(void) : void @06000005
smethod_20(string) : void @06000017
smethod_21(Thread) : bool @06000018
smethod_22(string) : void @06000019
smethod_23(int) : void @0600001A
smethod_24(RuntimeTypeHandle) : Type @0600001B
smethod_25(Type) : Module @0600001C
smethod_26(Module) : IntPtr @0600001D
smethod_27(Module) : string @0600001E
smethod_28(string, int) : char @0600001F
smethod_29(byte[], int, IntPtr, int) : void @06000020
smethod_30(RuntimeTypeHandle) : Type @06000006
smethod_30(byte[], int, IntPtr, int) : void @06000021
smethod_31(byte[]) : byte[] @06000022
smethod_32(void) : void @06000023
smethod_33(uint) : T @06000024
smethod_34(uint) : T @06000025
smethod_35(uint) : T @06000026
smethod_36(uint) : T @06000027
smethod_37(uint) : T @06000028
smethod_38(void) : void @06000029
Windows PowerShell
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly = [System.Reflection.Assembly]::Load([IO.File]::ReadAllBytes('C:\Users\tester\Desktop\TradeCraft\SetADAttributes_dumped-cleaned.exe'))
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000025).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 4116165899))
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000028).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 1887629692))
GetEnvironmentVariable
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000028).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2003513117))
c:\CMDMGMT\LOGS
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000026).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 1532378290))
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000025).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2572441635))
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000024).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2267448765))
cmdSetADAttributes.log
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000024).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2134964137))
.com
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000024).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2134964137))
.com
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000024).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 3070689446))
[START]
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000026).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2429311005))
UI
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000024).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 3412964790))
PS C:\Users\tester\Desktop\TradeCraft>
```

Reflective Decryption

Trace smethod_XX to obtain Token ID for that decryption routine

The screenshot shows the dnSpy interface with the Assembly Explorer and Module tabs open. The Module tab displays assembly code for a module at address 0x02000001. A specific section of the code is highlighted with a green box:

```
// Token: 0x06000024 RID: 36 RVA: 0x000046D8 File Offset: 0x000028D8
internal static T smethod_33<T>(uint uint_0)
{
    uint_0 = (uint_0 * 3498425777u ^ 1838153877u);
    T result;
    for (;;)
    {
        IL_2FE:
        uint num = 906850024u;
        for (;;)
        {
            uint num2;
            switch ((num2 = (num ^ 772608802u)) % 16u)
            {
                case 0u:
                    {
                        T[] array;
                        Buffer.BlockCopy(<Module>.byte_0, (int)uint_0, array, 0, sizeof(T));
                        result = array[0];
                        num = (num2 * 4294117222u ^ 553947038u);
                        continue;
                    }
                case 2u:
                    result = default(T);
                    uint_0 &= 1073741823u;
            }
        }
    }
}
```

A green arrow points from the highlighted assembly code down to a corresponding line in a Windows PowerShell window titled "Windows PowerShell". The PowerShell session shows the deobfuscation of a file named "SetADAttributes-dumped-cleaned.exe". The assembly code is being executed within the PowerShell environment to decrypt the file.

```
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly = [System.Reflection.Assembly]::Load([IO.File]::ReadAllBytes('C:\Users\tester\Desktop\TradeCraft\SetADAttributes-dumped-cleaned.exe'))
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000025).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 4116165899))
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000028).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 1887629692))
GetEnvironmentVariable
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000028).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2003513171))
c:\CMDMGT\LOGS
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000026).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 1532378290))
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000025).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2572441635))
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000024).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2267448765))
cmdSetADAttributes.log
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000024).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2134964137))
W
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000024).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2134964137))
W
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000024).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 3070689446))
[START]
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000026).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2429311005))
W
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000024).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 3412964790))
W
PS C:\Users\tester\Desktop\TradeCraft>
```

Reflective Decryption

Call decryption routine on protected string resource

The screenshot shows the dnSpy interface with the Assembly Explorer and GForm0 tabs open. The assembly code lists various methods, and the GForm0 tab displays C# code for a decryption routine. A green arrow points from the PowerShell window at the bottom to the decryption code in GForm0.

dnSpy v5.0.10 (x64)

File Edit View Debug Window Help | Start | GForm0

Assembly Explorer

GForm0

```
115     string string_ = <Module>.smethod_33<string>(3412964790u);
116     string text;
117     string string_2;
118     string string_3;
119     string string_4;
120     string text2;
121     string string3;
122     string string4;
123     for (;;)
124     {
125         IL_01:
126         uint num = 105361974u;
127         for (;;)
128         {
129             uint num2;
130             switch ((num2 = (num ^ 1693222786u)) % 5u)
131             {
132                 case 1u:
133                     text = "";
134                     num = (num2 * 2246993651u ^ 3806510851u);
135                     continue;
136                 case 2u:
137                     string_2 = <Module>.smethod_35<string>(2429311005u);
138                     string_3 = GForm0.smethod_11();
139                     string_4 = <Module>.smethod_33<string>(2134964137u);
140                     text2 = <Module>.smethod_33<string>(2134964137u);
141                     num = (num2 * 2705050580u ^ 1260127859u);
142             }
143         }
144     }
145 }
```

Windows PowerShell

```
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly = [System.Reflection.Assembly]::Load([IO.File]::ReadAllBytes('C:\Users\tester\Desktop\TradeCraft\SetADAttributes_dumped-cleaned.exe'))
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000025).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 4116165899))
1
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000028).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 1887629692))
GetEnvironmentVariable
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000028).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2003513117))
c:\CMDMGMT\LOGS
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000026).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 1532378290))
\ 
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000025).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2572441635))
:
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000024).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2287448765))
cmdSetADAttributes.log
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000024).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2134964137))
w
.com
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000024).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2134964137))
w
.com
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000024).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 3070689446))
[START]
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000026).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2429311005))
w
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000024).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 3412964790))
F0
GForm0
```

Reflective Decryption

Valid domain creds!!

The screenshot shows the dnSpy interface with the Assembly Explorer and GForm0 panes visible. The assembly code lists various smethod_ methods. The GForm0 pane contains C# code for string manipulation and a switch statement. Below the application window is a Windows PowerShell session. The PowerShell session shows the deobfuscation of assembly code, specifically resolving methods like smethod_11, smethod_33, and smethod_35, and setting environment variables such as \$CMDMGMT and \$LOGS. The code being deobfuscated includes logic for generating strings based on numerical calculations and switching between them.

```
dnSpy v5.0.10 (x64)
File Edit View Debug Window Help | Start | Search
Assembly Explorer
smethod_16(Thread, object) : void @06000013
smethod_17(int) : void @06000014
smethod_18() : bool @06000015
smethod_19() : bool @06000016
smethod_20() : void @06000005
smethod_20(string) : void @06000017
smethod_21(Thread) : bool @06000018
smethod_22(string) : void @06000019
smethod_23(int) : void @0600001A
smethod_24(RuntimeTypeHandle) : Type @0600001B
smethod_25(Type) : Module @0600001C
smethod_26(Module) : IntPtr @0600001D
smethod_27(Module) : string @0600001E
smethod_28(string, int) : char @0600000F
smethod_29(byte[], int, IntPtr, int) : void @06000020
smethod_3(RuntimeTypeHandle) : Type @06000006
smethod_30(byte[], int, IntPtr, int) : void @06000021
smethod_31(byte[]) : byte[] @06000022
smethod_32() : void @06000023
smethod_33(uint) : T @06000024
smethod_34(uint) : T @06000025
smethod_35(uint) : T @06000026
smethod_36(uint) : T @06000027
smethod_37(uint) : T @06000028
smethod_38() : void @06000029
GForm0
string string_ = <Module>.smethod_33<string>(3412964790u);
string text;
string string_2;
string string_3;
string string_4;
string text2;
string text3;
string text4;
for (;;)
{
    IL_00A1:
    uint num = 105361974u;
    for (;;)
    {
        uint num2;
        switch ((num2 = (num ^ 1693222786u)) % 5u)
        {
            case 1u:
                text = "";
                num = (num2 * 2246993651u ^ 3806510851u);
                continue;
            case 2u:
                string_2 = <Module>.smethod_35<string>(2429311005u);
                string_3 = GForm0.smethod_11();
                string_4 = <Module>.smethod_33<string>(2134964137u);
                text2 = <Module>.smethod_33<string>(2134964137u);
                num = (num2 * 2705050580u ^ 1260127859u);
        }
    }
}
Windows PowerShell
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly = [System.Reflection.Assembly]::Load([IO.File]::ReadAllBytes('C:\Users\tester\Desktop\TradeCraft\SetADAttributes_dumped-cleaned.exe'))
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000025).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 4116165899))
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000028).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 1887629692))
GetEnvironmentVariable
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000028).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2003513117))
$c : \CMDMGMT\LOGS
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000026).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 1532378290))
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000025).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2572441635))
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000024).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2267448765))
$cmdSetADAttributes.log
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000024).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2134964137))
$W : \Windows\system32\cmdSetADAttributes.log
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000024).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2134964137))
$W : \Windows\system32\cmdSetADAttributes.log
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000024).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 3070689446))
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000026).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 2429311005))
$UI : \Windows\system32\cmdSetADAttributes.log
PS C:\Users\tester\Desktop\TradeCraft> $DeobfuscatedAssembly.Modules[0].ResolveMethod(0x06000024).MakeGenericMethod(@([String])).Invoke($null, @([UInt32] 3412964790))
$O : \Windows\system32\cmdSetADAttributes.log
PS C:\Users\tester\Desktop\TradeCraft>
```

Perseverance

The screenshot shows a debugger interface with assembly code on the left and a tooltip on the right. The assembly code is annotated with line numbers from 506 to 552. The tooltip provides details about the `ValidateCredentials` method:

`PrincipalContext.ValidateCredentials(string userName, string password)` (+ 1 overload)
Creates the connections to the server and returns a Boolean value that specifies whether the specified username and password are valid.
userName: The username that is validated on the server. See the Remarks section for information on the format of **userName**.
password: The password that is validated on the server.
Returns: true if the credentials are valid; otherwise false.

```
506      ; 506
507      ; 507
508      ; 508
509      ; 509
510      ; 510
511      ; 511
512      ; 512
513      ; 513
514      ; 514
515      ; 515
516      ; 516
517      ; 517
518      ; 518
519      ; 519
520      ; 520
521      ; 521
522      ; 522
523      ; 523
524      ; 524
525      ; 525
526      ; 526
527      ; 527
528      ; 528
529      ; 529
530      ; 530
531      ; 531
532      ; 532
533      ; 533
534      ; 534
535      ; 535
536      ; 536
537      ; 537
538      ; 538
539      ; 539
540      ; 540
541      ; 541
542      ; 542
543      ; 543
544      ; 544
545      ; 545
546      ; 546
547      ; 547
548      ; 548
549      ; 549
550      ; 550
551      ; 551
552      ; 552
```

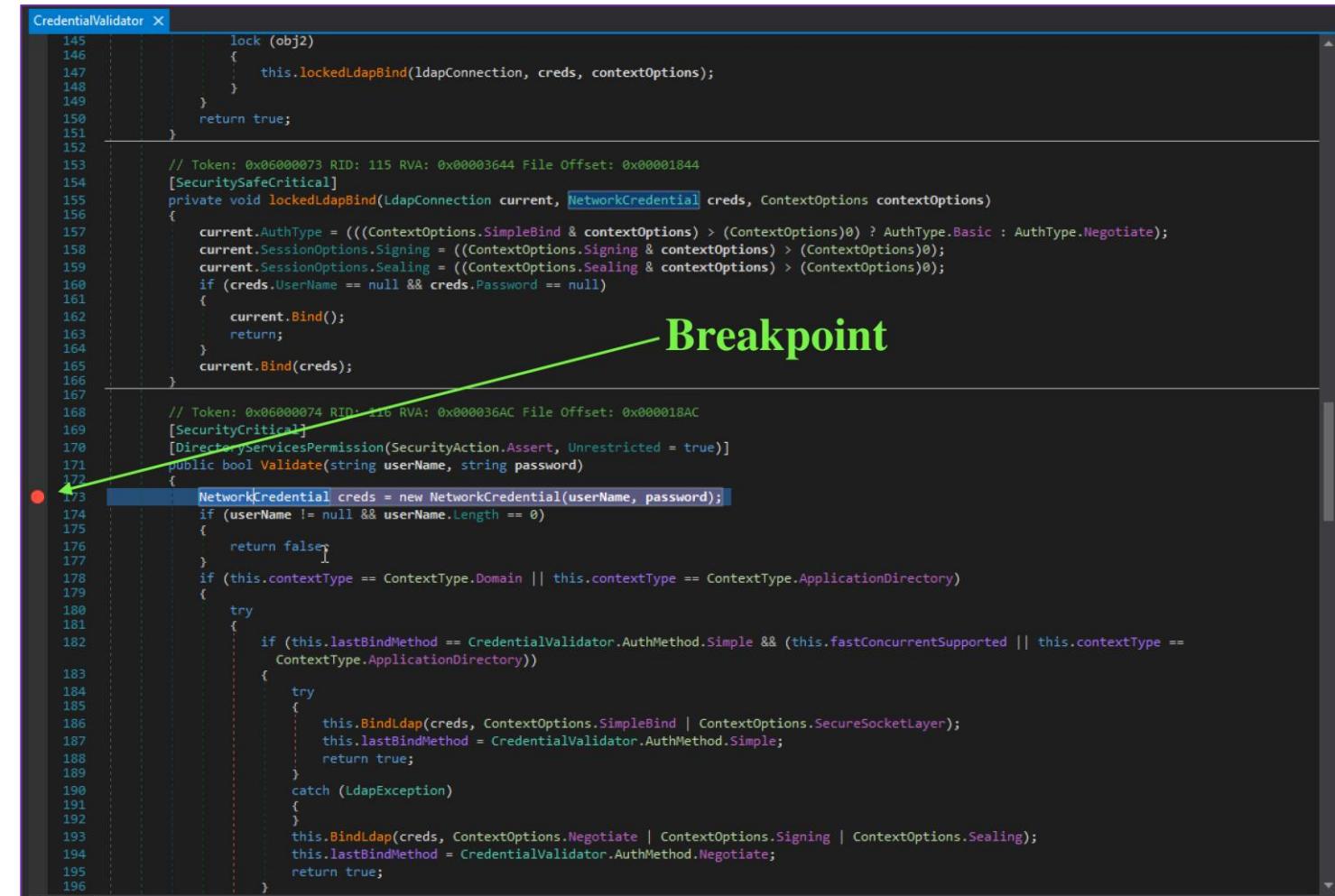
3401872413u : 28

Perseverance

```
PrincipalContext X
204 }
205
206     /// <summary>Gets the name of the server to which the principal context is connected. </summary>
207     /// <returns>The name of the server to which the principal context is connected or null if the principal context is not connected to a server. </returns>
208     // Token: 0x1700000D RID: 13
209     // (get) Token: 0x06000082 RID: 130 RVA: 0x000039FD File Offset: 0x00001BFD
210     public string ConnectedServer
211     {
212         get
213         {
214             this.CheckDisposed();
215             this.Initialize();
216             return this.connectedServer;
217         }
218     }
219
220     /// <summary>Creates the connections to the server and returns a Boolean value that specifies whether the specified username and password are valid. </summary>
221     /// <param name="userName">The username that is validated on the server. See the Remarks section for information on the format of <paramref name="userName" />. </param>
222     /// <param name="password">The password that is validated on the server.</param>
223     /// <returns>
224     ///     <see langword="true" /> if the credentials are valid; otherwise <see langword="false" />. </returns>
225     // Token: 0x06000083 RID: 131 RVA: 0x00003A11 File Offset: 0x00001C11
226     public bool ValidateCredentials(string userName, string password)
227     {
228         this.CheckDisposed();
229         if ((userName == null && password != null) || (userName != null && password == null))
230         {
231             throw new ArgumentException(StringResources.ContextBadUserPwdCombo);
232         }
233         return this.credValidate.Validate(userName, password);
234     }
235     @ bool CredentialValidator.Validate(string userName, string password) (+ 1 overload)
236
237     /// <summary>Creates the connections to the server and returns a Boolean value that specifies whether the specified user name and password are valid. This method performs fast credential validation of the username and password. </summary>
238     /// <param name="userName">The username that is validated on the server. See the Remarks section for information on the format of <paramref name="userName" />. </param>
239     /// <param name="password">The password that is validated on the server.</param>
240     /// <param name="options">A combination of one or more <see cref="T:System.DirectoryServices.AccountManagement.ContextOptions" /> enumeration values that specify the options used to bind to the server. This parameter can only specify Simple bind with or without SSL, or Negotiate bind. </param>
241     /// <returns>
242     ///     <see langword="true" /> if the credentials are valid; otherwise <see langword="false" />. </returns>
243     /// <exception cref="T:System.ArgumentException">The <paramref name="options" /> parameter must specify <see cref="F:System.DirectoryServices.AccountManagement.ContextOptions.Negotiate" /> when the context type is <see cref="F:System.DirectoryServices.AccountManagement.ContextType.Machine" />. </exception>
244     // Token: 0x06000084 RID: 132 RVA: 0x00003A40 File Offset: 0x00001C40
245     public bool ValidateCredentials(string userName, string password, ContextOptions options)
246     {
247         this.CheckDisposed();
```

- After password creation, the credentials are tested against a domain server
- Validate Credentials method looks to send username and password as string args
- Trace Validate down one more level

Perseverance



A screenshot of a debugger interface showing a C# code editor. A green arrow points from the word 'Breakpoint' to the line of code where a breakpoint is set. The line of code is highlighted in blue.

```
145     lock (obj2)
146     {
147         this.lockedLdapBind(ldapConnection, creds, contextOptions);
148     }
149     return true;
150 }
151 }
152
153 // Token: 0x06000073 RID: 115 RVA: 0x00003644 File Offset: 0x00001844
154 [SecuritySafeCritical]
155 private void lockedLdapBind(LdapConnection current, NetworkCredential creds, ContextOptions contextOptions)
156 {
157     current.AuthType = (((ContextOptions.SimpleBind & contextOptions) > (ContextOptions)0) ? AuthType.Basic : AuthType.Negotiate);
158     current.SessionOptions.Signing = (((ContextOptions.Signing & contextOptions) > (ContextOptions)0));
159     current.SessionOptions.Sealing = (((ContextOptions.Sealing & contextOptions) > (ContextOptions)0));
160     if (creds.UserName == null && creds.Password == null)
161     {
162         current.Bind();
163         return;
164     }
165     current.Bind(creds);
166 }
167
168 // Token: 0x06000074 RID: 116 RVA: 0x000036AC File Offset: 0x000018AC
169 [SecurityCritical]
170 [DirectoryServicesPermission(SecurityAction.Assert, Unrestricted = true)]
171 public bool Validate(string userName, string password)
172 {
173     NetworkCredential creds = new NetworkCredential(userName, password);
174     if (userName != null && userName.Length == 0)
175     {
176         return false;
177     }
178     if (this.contextType == ContextType.Domain || this.contextType == ContextType.ApplicationDirectory)
179     {
180         try
181         {
182             if (this.lastBindMethod == CredentialValidator.AuthMethod.Simple && (this.fastConcurrentSupported || this.contextType ==
183             ContextType.ApplicationDirectory))
184             {
185                 try
186                 {
187                     this.BindLdap(creds, ContextOptions.SimpleBind | ContextOptions.SecureSocketLayer);
188                     this.lastBindMethod = CredentialValidator.AuthMethod.Simple;
189                     return true;
190                 }
191                 catch (LdapException)
192                 {
193                     this.BindLdap(creds, ContextOptions.Negotiate | ContextOptions.Signing | ContextOptions.Sealing);
194                     this.lastBindMethod = CredentialValidator.AuthMethod.Negotiate;
195                     return true;
196                 }
197             }
198         }
199     }
200 }
```

- NetworkCredential variable being set by calling NetworkCredential method with username and password string variables
- Set a breakpoint on this line
- DnsSpy will need to be x86 version and run as Admin to reach this part of the code.
- Run it!!

Perseverance

dnSpy v5.0.10 (x86, Administrator, Debugging)

File Edit View Debug Window Help | C# | Continue | Back | Stop | Step | Break | Run | Assembly Explorer CredentialValidator

Assembly Explorer: Admin017 (1.0.0.0) - System.DirectoryServices.AccountManagement (4.0.0.0) - System.DirectoryServices.AccountManagement.dll

C# CredentialValidator

```
144     object obj2 = this.cacheLock;
145     lock (obj2)
146     {
147         this.lockedLdapBind(ldapConnection, creds, contextOptions);
148     }
149     return true;
150 }
151 }
152 }

// Token: 0x06000073 RID: 115 RVA: 0x00003644 File Offset: 0x00001844
[SecuritySafeCritical]
private void lockedLdapBind(LdapConnection current, NetworkCredential creds, ContextOptions contextOptions)
{
    current.AuthType = (((ContextOptions.SimpleBind & contextOptions) > (ContextOptions)0) ? A
    current.SessionOptions.Signing = ((ContextOptions.Signing & contextOptions) > (ContextOpti
    current.SessionOptions.Sealing = ((ContextOptions.Sealing & contextOptions) > (ContextOpti
    if (creds.UserName == null && creds.Password == null)
    {
        current.Bind();
        return;
    }
    current.Bind(creds);
}

// Token: 0x06000074 RID: 116 RVA: 0x000036AC File Offset: 0x000018AC
[SecurityCritical]
[DirectoryServicesPermission(SecurityAction.Assert, Unrestricted = true)]
public bool Validate(string userName, string password)
{
    NetworkCredential creds = new NetworkCredential(userName, password);
    if (userName != null && userName.Length == 0)
    {
        return false;
    }
}

100 %
```

Locals

Name	Value
this	System.DirectoryServices.AccountManagement.CredentialValidator
userName	"Administrator"
password	"6[REDACTED]"
creds	null
ex	null

- Change VM hostname to any hostname in the environment
- Run this de-obfuscated binary in dnspy and hit this breakpoint
- Gain Local Admin credentials to any machine in the domain
- Confirmed by using PoC hostname and confirming password.



Issues

- ConfuserEx on max settings adds anti-dumping protections that this workflow was not able to bypass
- ConfuserEx obfuscated binaries have been recently getting caught and removed by Windows Defender likely due to the unpacking method.
- ConfuserEx is used by actual threat actors in the wild



www.pickles.xyz



@_P1CKLES_



slucas@specterops.io