black hat EUROPE 2018

DECEMBER 3-6, 2018

EXCEL LONDON / UNITED KINGDOM

Network Defender Archeology

An NSM Case Study in Lateral Movement with DCOM

Gigamon® | ATR

RESEARCH PROTOTYPE DETECT

Justin Warner
@sixdub

Current: Principal Security Engineer at Gigamon Applied Threat Research (ATR)

Former:

- Computer Science, USAF Academy c/o '10
- Cyber Counter-Intelligence, USAF
- Red Team Lead – Veris Group ATD
- ICEBRG

Gigamon® | ATR

RESEARCH PROTOTYPE DETECT

Alex Sirr @DarkAl3x1s

Current: Security Engineer at Gigamon Applied Threat Research (ATR)

Former:

- ICEBRG intern and Security Engineer
- Informatics @ University of Washington

Acknowledgements

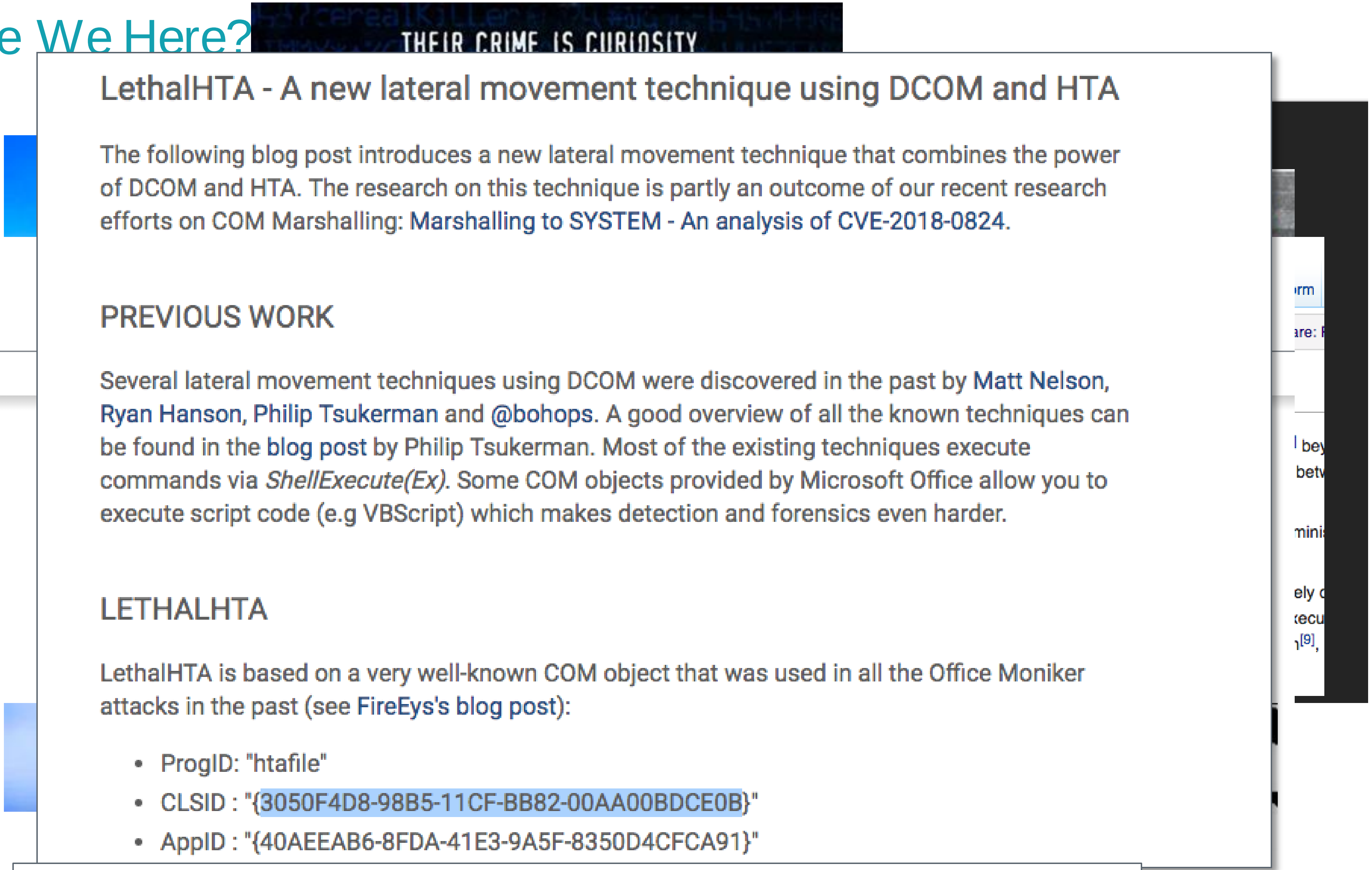
Things are rarely ever “new” anymore, but rather a continuous expansion of others’ previous works that branches into many directions…

A special thanks to those who inspired, influenced, or shaped this presentation!

Shout out to Casey Smith, Matt Nelson, Philip Tsukerman, Joe Johnson

Let’s Tell a Story

Why Are We Here?



LethalHTA - A new lateral movement technique using DCOM and HTA

The following blog post introduces a new lateral movement technique that combines the power of DCOM and HTA. The research on this technique is partly an outcome of our recent research efforts on COM Marshalling: Marshalling to SYSTEM - An analysis of CVE-2018-0824.

PREVIOUS WORK

Several lateral movement techniques using DCOM were discovered in the past by Matt Nelson, Ryan Hanson, Philip Tsukerman and @bohops. A good overview of all the known techniques can be found in the blog post by Philip Tsukerman. Most of the existing techniques execute commands via *ShellExecute(Ex)*. Some COM objects provided by Microsoft Office allow you to execute script code (e.g VBScript) which makes detection and forensics even harder.

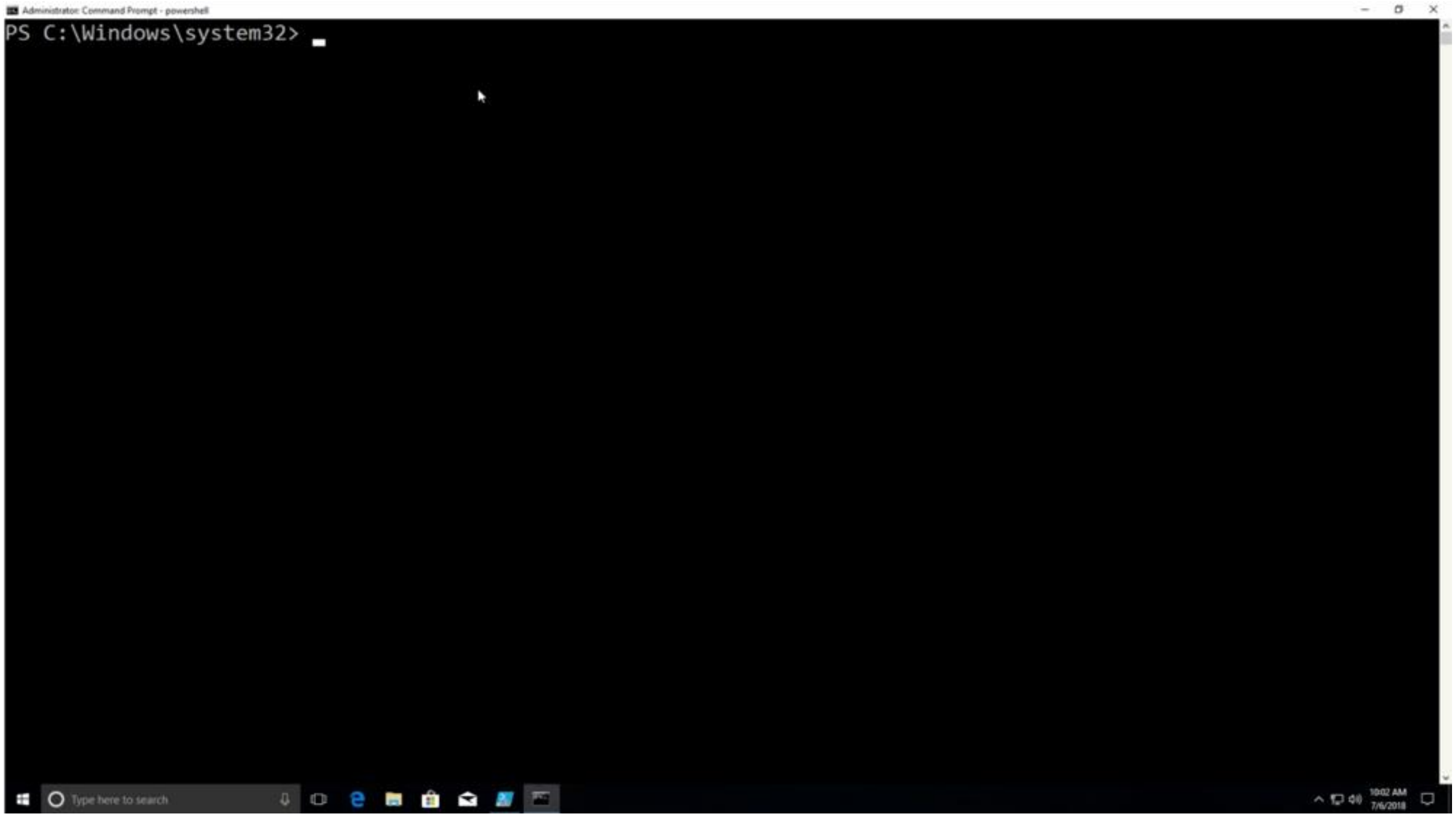
LETHALHTA

LethalHTA is based on a very well-known COM object that was used in all the Office Moniker attacks in the past (see FireEys's blog post):

- ProgID: "htafile"
- CLSID : "{3050F4D8-98B5-11CF-BB82-00AA00BDCE0B}"
- AppID : "{40AEEAB6-8FDA-41E3-9A5F-8350D4CFCA91}"

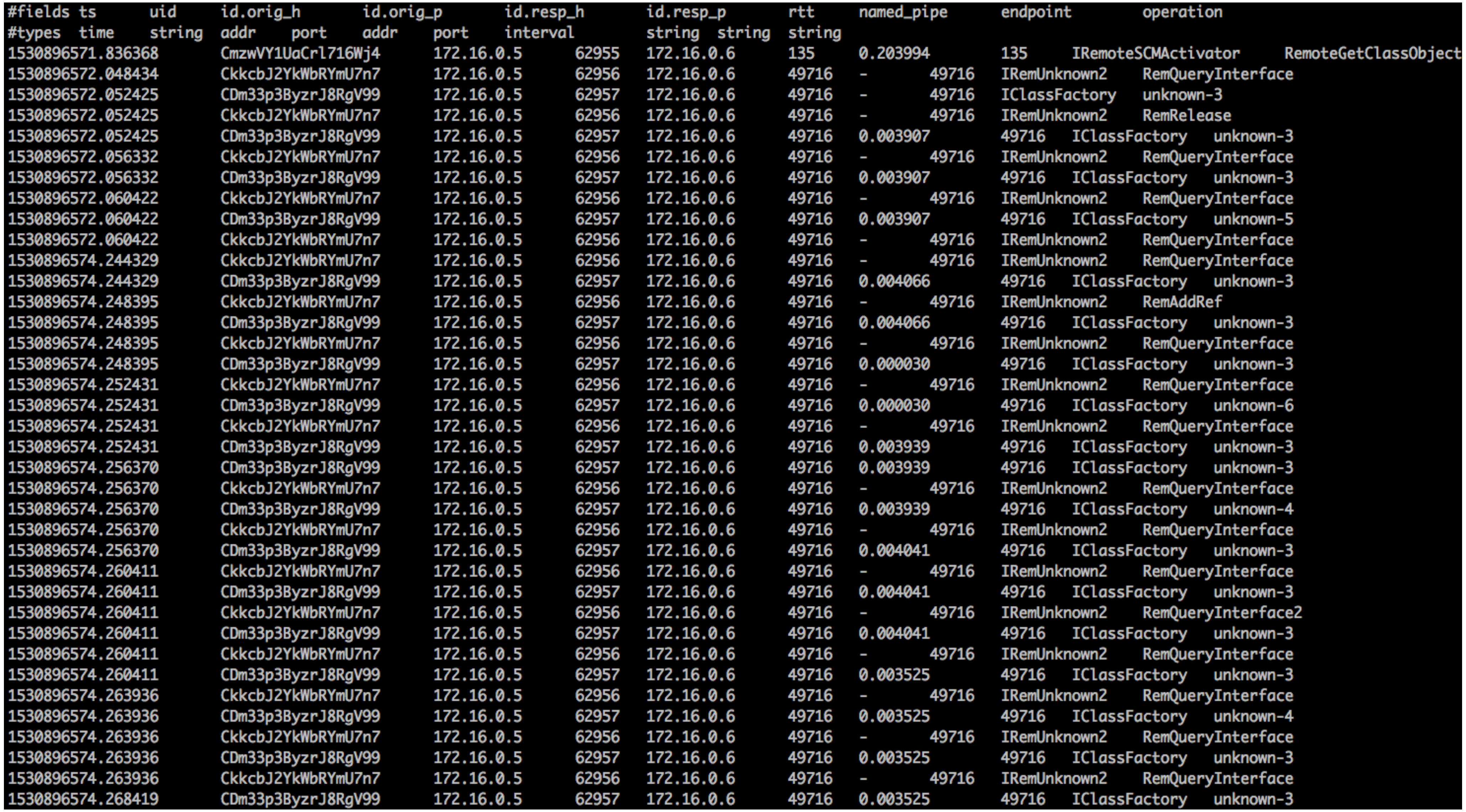
https://codewhitesec.blogspot.com/2018/07/lethalhta.html

DCOM Lateral Movement



```
PS C:\Windows\system32> _
```

Problem Here



```
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h       id.resp_p       rtt     named_pipe      endpoint        operation
#types  time    string  addr    port    addr    port    interval        string  string  string
1530896571.836368       CmzwVY1UaCrl716Wj4      172.16.0.5      62955   172.16.0.6      135     0.203994        135     IRemoteSCMActivator     RemoteGetClassObject
1530896572.048434       CkkcbJ2YkWbRYmU7n7      172.16.0.5      62956   172.16.0.6      49716   -       49716   IRemUnknown2    RemQueryInterface
1530896572.052425       CDm33p3ByzrJ8RgV99      172.16.0.5      62957   172.16.0.6      49716   -       49716   IClassFactory   unknown-3
1530896572.052425       CkkcbJ2YkWbRYmU7n7      172.16.0.5      62956   172.16.0.6      49716   -       49716   IRemUnknown2    RemRelease
1530896572.052425       CDm33p3ByzrJ8RgV99      172.16.0.5      62957   172.16.0.6      49716   0.003907        49716   IClassFactory   unknown-3
1530896572.056332       CkkcbJ2YkWbRYmU7n7      172.16.0.5      62956   172.16.0.6      49716   -       49716   IRemUnknown2    RemQueryInterface
1530896572.056332       CDm33p3ByzrJ8RgV99      172.16.0.5      62957   172.16.0.6      49716   0.003907        49716   IClassFactory   unknown-3
1530896572.060422       CkkcbJ2YkWbRYmU7n7      172.16.0.5      62956   172.16.0.6      49716   -       49716   IRemUnknown2    RemQueryInterface
1530896572.060422       CDm33p3ByzrJ8RgV99      172.16.0.5      62957   172.16.0.6      49716   0.003907        49716   IClassFactory   unknown-5
1530896572.060422       CkkcbJ2YkWbRYmU7n7      172.16.0.5      62956   172.16.0.6      49716   -       49716   IRemUnknown2    RemQueryInterface
1530896574.244329       CkkcbJ2YkWbRYmU7n7      172.16.0.5      62956   172.16.0.6      49716   -       49716   IRemUnknown2    RemQueryInterface
1530896574.244329       CDm33p3ByzrJ8RgV99      172.16.0.5      62957   172.16.0.6      49716   0.004066        49716   IClassFactory   unknown-3
1530896574.248395       CkkcbJ2YkWbRYmU7n7      172.16.0.5      62956   172.16.0.6      49716   -       49716   IRemUnknown2    RemAddRef
1530896574.248395       CDm33p3ByzrJ8RgV99      172.16.0.5      62957   172.16.0.6      49716   0.004066        49716   IClassFactory   unknown-3
1530896574.248395       CkkcbJ2YkWbRYmU7n7      172.16.0.5      62956   172.16.0.6      49716   -       49716   IRemUnknown2    RemQueryInterface
1530896574.248395       CDm33p3ByzrJ8RgV99      172.16.0.5      62957   172.16.0.6      49716   0.000030        49716   IClassFactory   unknown-3
1530896574.252431       CkkcbJ2YkWbRYmU7n7      172.16.0.5      62956   172.16.0.6      49716   -       49716   IRemUnknown2    RemQueryInterface
1530896574.252431       CDm33p3ByzrJ8RgV99      172.16.0.5      62957   172.16.0.6      49716   0.000030        49716   IClassFactory   unknown-6
1530896574.252431       CkkcbJ2YkWbRYmU7n7      172.16.0.5      62956   172.16.0.6      49716   -       49716   IRemUnknown2    RemQueryInterface
1530896574.252431       CDm33p3ByzrJ8RgV99      172.16.0.5      62957   172.16.0.6      49716   0.003939        49716   IClassFactory   unknown-3
1530896574.256370       CDm33p3ByzrJ8RgV99      172.16.0.5      62957   172.16.0.6      49716   0.003939        49716   IClassFactory   unknown-3
1530896574.256370       CkkcbJ2YkWbRYmU7n7      172.16.0.5      62956   172.16.0.6      49716   -       49716   IRemUnknown2    RemQueryInterface
1530896574.256370       CDm33p3ByzrJ8RgV99      172.16.0.5      62957   172.16.0.6      49716   0.003939        49716   IClassFactory   unknown-4
1530896574.256370       CkkcbJ2YkWbRYmU7n7      172.16.0.5      62956   172.16.0.6      49716   -       49716   IRemUnknown2    RemQueryInterface
1530896574.256370       CDm33p3ByzrJ8RgV99      172.16.0.5      62957   172.16.0.6      49716   0.004041        49716   IClassFactory   unknown-3
1530896574.260411       CkkcbJ2YkWbRYmU7n7      172.16.0.5      62956   172.16.0.6      49716   -       49716   IRemUnknown2    RemQueryInterface
1530896574.260411       CDm33p3ByzrJ8RgV99      172.16.0.5      62957   172.16.0.6      49716   0.004041        49716   IClassFactory   unknown-3
1530896574.260411       CkkcbJ2YkWbRYmU7n7      172.16.0.5      62956   172.16.0.6      49716   -       49716   IRemUnknown2    RemQueryInterface2
1530896574.260411       CDm33p3ByzrJ8RgV99      172.16.0.5      62957   172.16.0.6      49716   0.004041        49716   IClassFactory   unknown-3
1530896574.260411       CkkcbJ2YkWbRYmU7n7      172.16.0.5      62956   172.16.0.6      49716   -       49716   IRemUnknown2    RemQueryInterface
1530896574.260411       CDm33p3ByzrJ8RgV99      172.16.0.5      62957   172.16.0.6      49716   0.003525        49716   IClassFactory   unknown-3
1530896574.263936       CkkcbJ2YkWbRYmU7n7      172.16.0.5      62956   172.16.0.6      49716   -       49716   IRemUnknown2    RemQueryInterface
1530896574.263936       CDm33p3ByzrJ8RgV99      172.16.0.5      62957   172.16.0.6      49716   0.003525        49716   IClassFactory   unknown-4
1530896574.263936       CkkcbJ2YkWbRYmU7n7      172.16.0.5      62956   172.16.0.6      49716   -       49716   IRemUnknown2    RemQueryInterface
1530896574.263936       CDm33p3ByzrJ8RgV99      172.16.0.5      62957   172.16.0.6      49716   0.003525        49716   IClassFactory   unknown-3
1530896574.263936       CkkcbJ2YkWbRYmU7n7      172.16.0.5      62956   172.16.0.6      49716   -       49716   IRemUnknown2    RemQueryInterface
1530896574.268419       CDm33p3ByzrJ8RgV99      172.16.0.5      62957   172.16.0.6      49716   0.003525        49716   IClassFactory   unknown-3
```

Real Talk

Simple fact: DCOM is abused by adversaries yet NSM techniques for DCOM are rarely discussed.

We must rise to the occasion to:

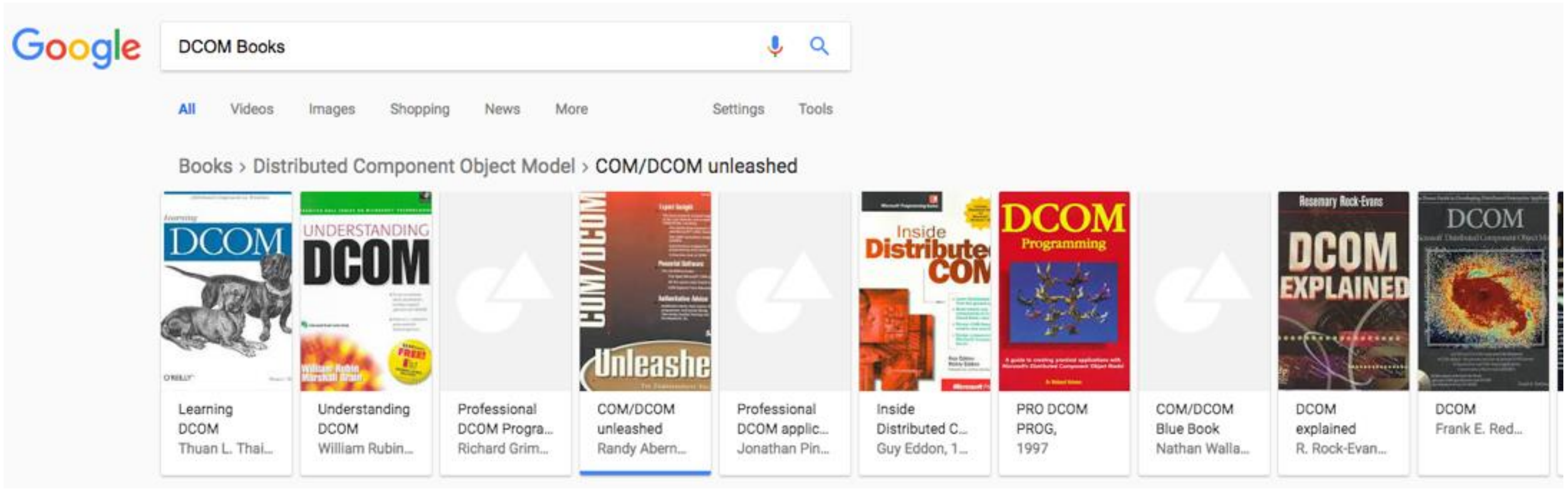
- Understand the “normal” behaviors of DCOM
- Recognize malicious indications of DCOM abuse
- Adapt our tools to empower us to detect DCOM abuse

Let’s deep dive into DCOM and look at NSM techniques.

Let’s Dive In

Where Did We Learn This?

A sample of some of the books out there on DCOM. We used 1 or 2 of these.


Google
DCOM Books
All
Videos
Images
Shopping
News
More
Settings
Tools
Books › Distributed Component Object Model › COM/DCOM unleashed
Learning DCOM Thuan L. Thai...
Understanding DCOM William Rubin...
Professional DCOM Progra... Richard Grim...
COM/DCOM unleashed Randy Abern...
Professional DCOM applic... Jonathan Pin...
Inside Distributed C... Guy Eddon, 1...
PRO DCOM PROG, 1997
COM/DCOM Blue Book Nathan Walla...
DCOM explained R. Rock-Evan...
DCOM Frank E. Red...

The books we read are older than Alex…

Component Object Model (COM)

COM is a language independent model that allows applications to expose objects with functionality

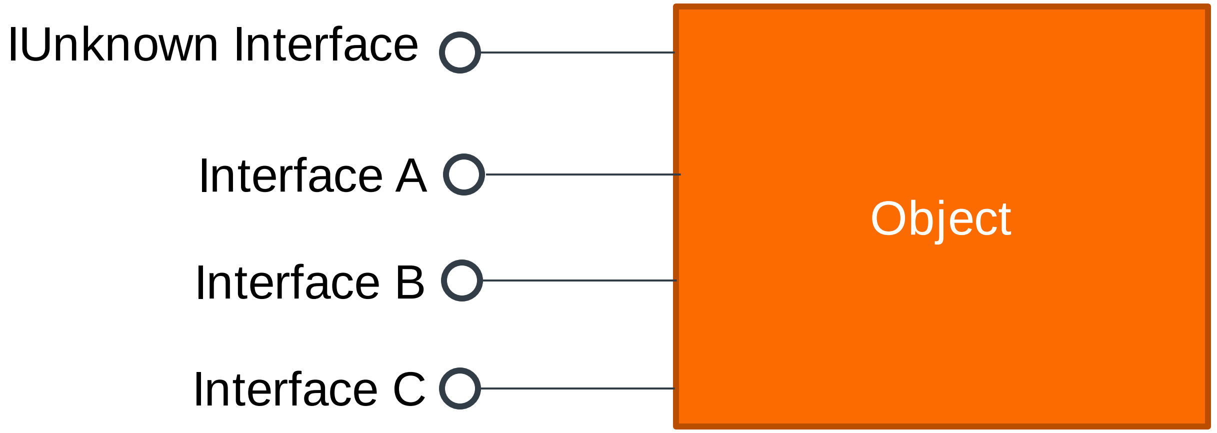
COM Overview

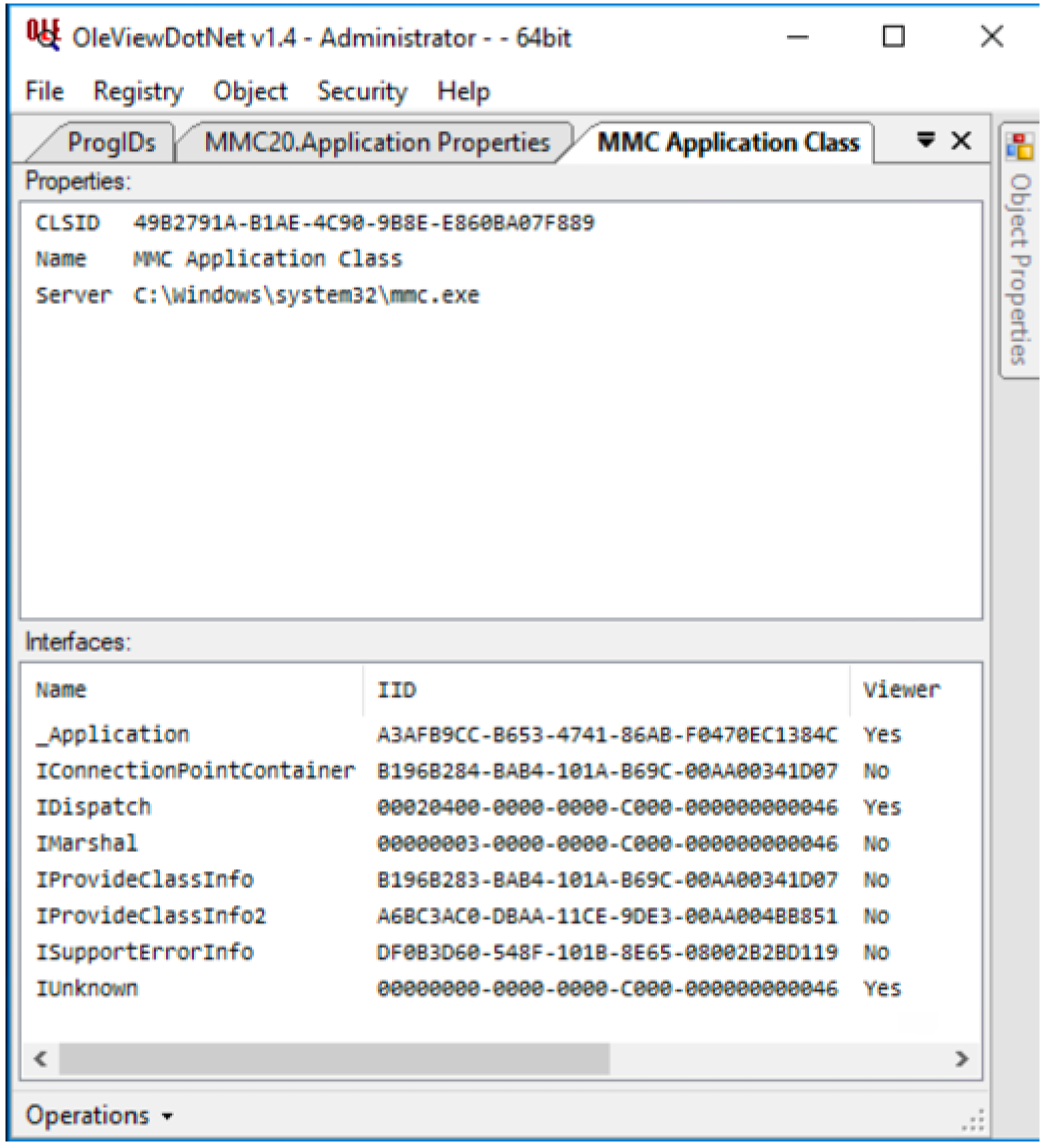
- Components are compiled code that provide functionality to the system
- Class IDs (CLSID) are used on system to uniquely identify a component
- Components are **typically** registered on a system, sometimes automatically
- Components implement one or more interfaces

COM Interfaces

Components implement interfaces for interoperability

- Interfaces define and expose common functionality
- All objects implement the IUnknown interface
- The IDispatch interface handles “automation”
- The ISystemActivator interfaces handle instantiation (direct or through IClassFactory)




OleViewDotNet v1.4 - Administrator - - 64bit
File Registry Object Security Help
ProgIDs
MMC20.Application Properties
MMC Application Class
Object Properties
Operations

Properties:

```
CLSID   49B2791A-B1AE-4C90-9B8E-E860BA07F889
Name    MMC Application Class
Server  C:\Windows\system32\mmc.exe
```

Interfaces:

Name	IID	Viewer
_Application	A3AFB9CC-B653-4741-86AB-F0470EC1384C	Yes
IConnectionPointContainer	B196B284-BAB4-101A-B69C-00AA00341D07	No
IDispatch	00020400-0000-0000-C000-000000000046	Yes
IMarshal	00000003-0000-0000-C000-000000000046	No
IProvideClassInfo	B196B283-BAB4-101A-B69C-00AA00341D07	No
IProvideClassInfo2	A6BC3AC0-DBAA-11CE-9DE3-00AA004BB851	No
ISupportErrorInfo	DF0B3D60-548F-101B-8E65-08002B2BD119	No
IUnknown	00000000-0000-0000-C000-000000000046	Yes

Distributed COM (DCOM)

COM objects are exposed/callable over a network in a distributed manner, hence Distributed COM (DCOM)

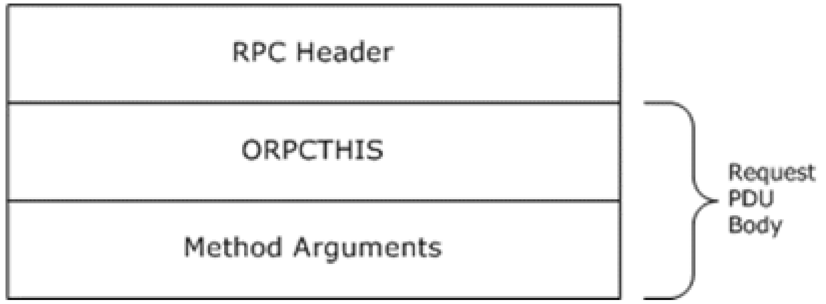
DCOM Overview

- Uses DCERPC as a transport mechanism
- Client/server model
- Uses request / response structures to communicate
- Code is executed on the server and output is returned to client

DCOM Primer

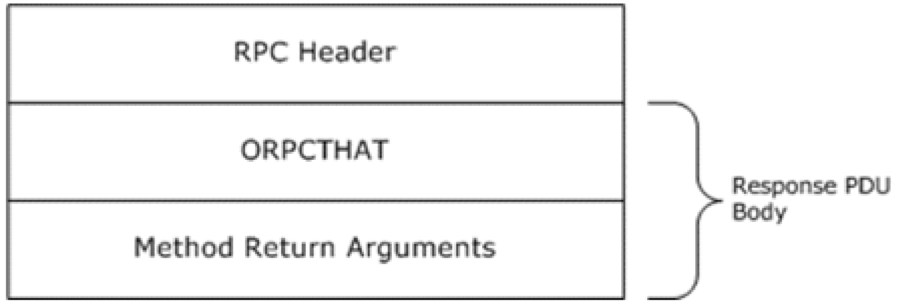
- DCERPC “Bind” is used to attach to a specific COM interface using the Interface ID (IID) and “Alter_Context” can be used to move between bound interfaces.
- After binding, messages will be passed using specific structures:

DCOM Requests



```
typedef struct tagORPCTHIS {
  COMVERSION version;
  unsigned long flags;
  unsigned long reserved1;
  CID cid;
  [unique] ORPC_EXTENT_ARRAY* extensions;
} ORPCTHIS;
```

DCOM Response



```
typedef struct tagORPCTHAT {
  unsigned long flags;
  [unique] ORPC_EXTENT_ARRAY* extensions;
} ORPCTHAT;
```

https://msdn.microsoft.com/en-us/library/cc226811.aspx

Adversarial COM/DCOM & Lateral Movement

Adversary Techniques Using COM

Numerous benefits of abusing COM functionality:

- Legacy technique that is not well documented or understood
- Can be used for multiple stages in attack lifecycle
- Host / network indicators vary. Visibility often insufficient.

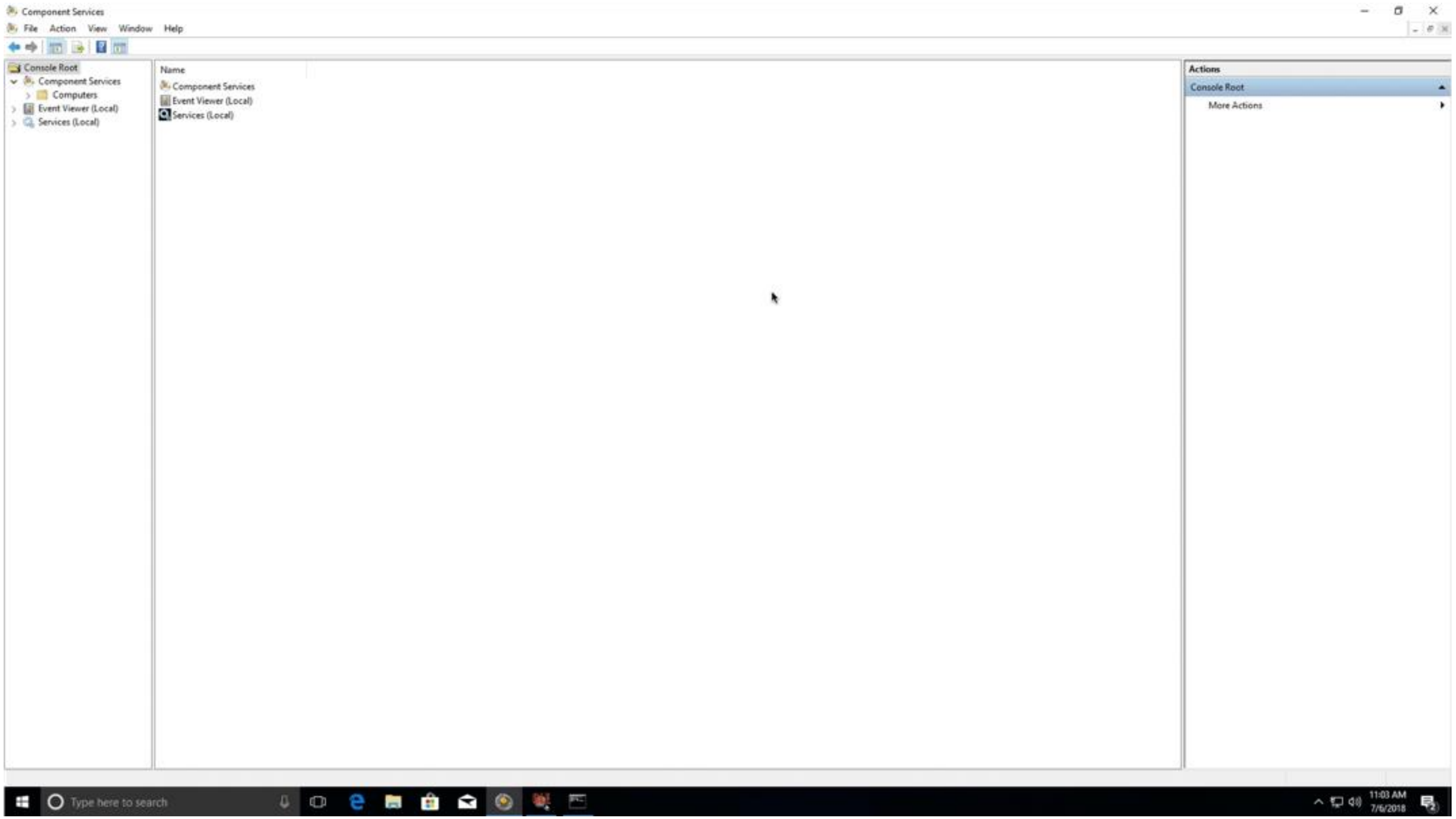
Attack Lifecycle Highlights

Code Execution	Lateral Movement	Persistence
• COM Scriptlets • WSH Injection (pubprn.vbs) • AMSI Bypass	• Many common objects allow for execution	• COM Hijacking • Malicious Office Add-ins

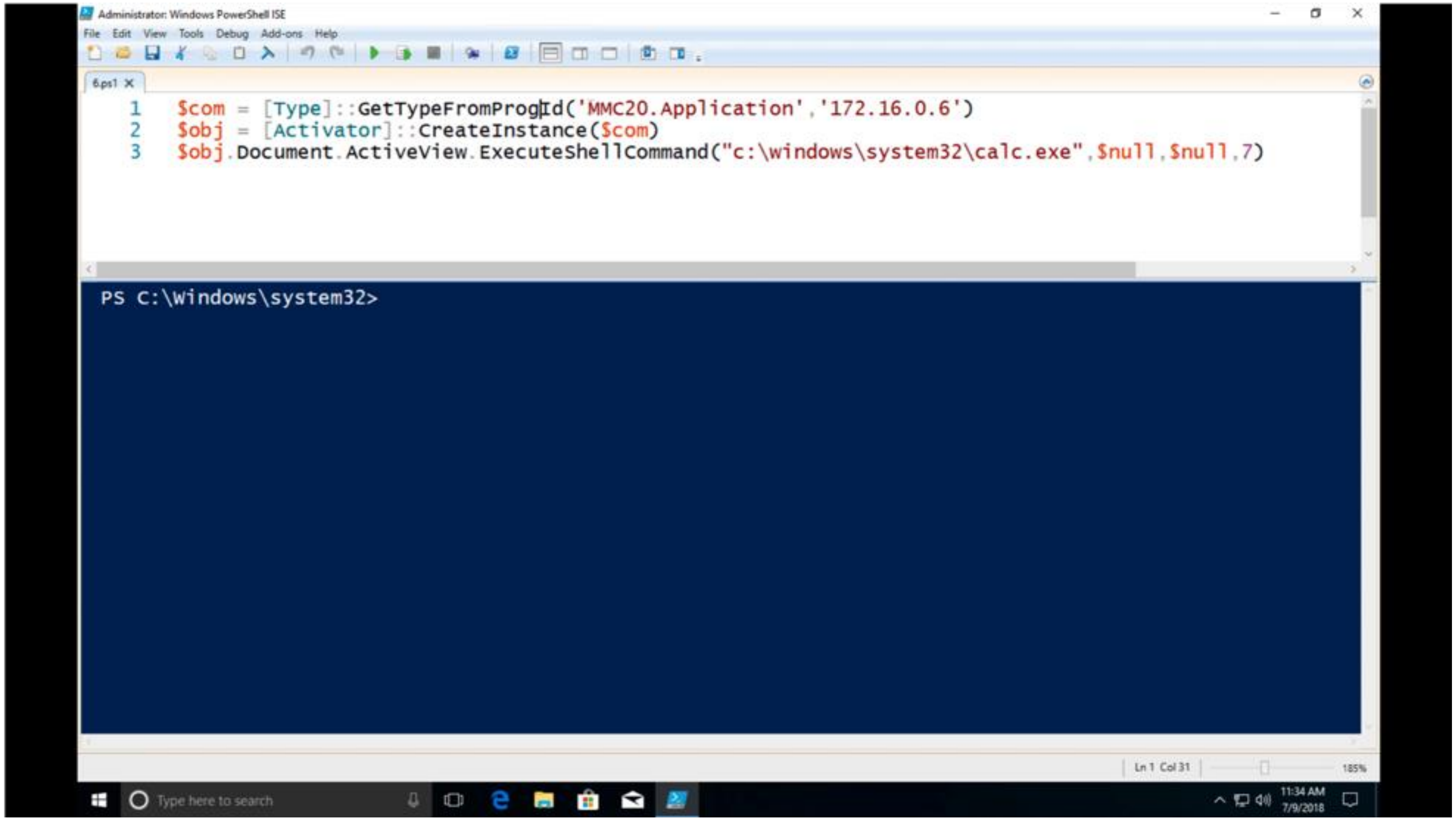
GO WATCH THIS TALK! Casey and Matt document it all.
https://www.slideshare.net/enigma0x3/windows-operating-system-archaeology

All of These

Finding Object - DEMO



Fun Technique #1 – MMC2.0 Application

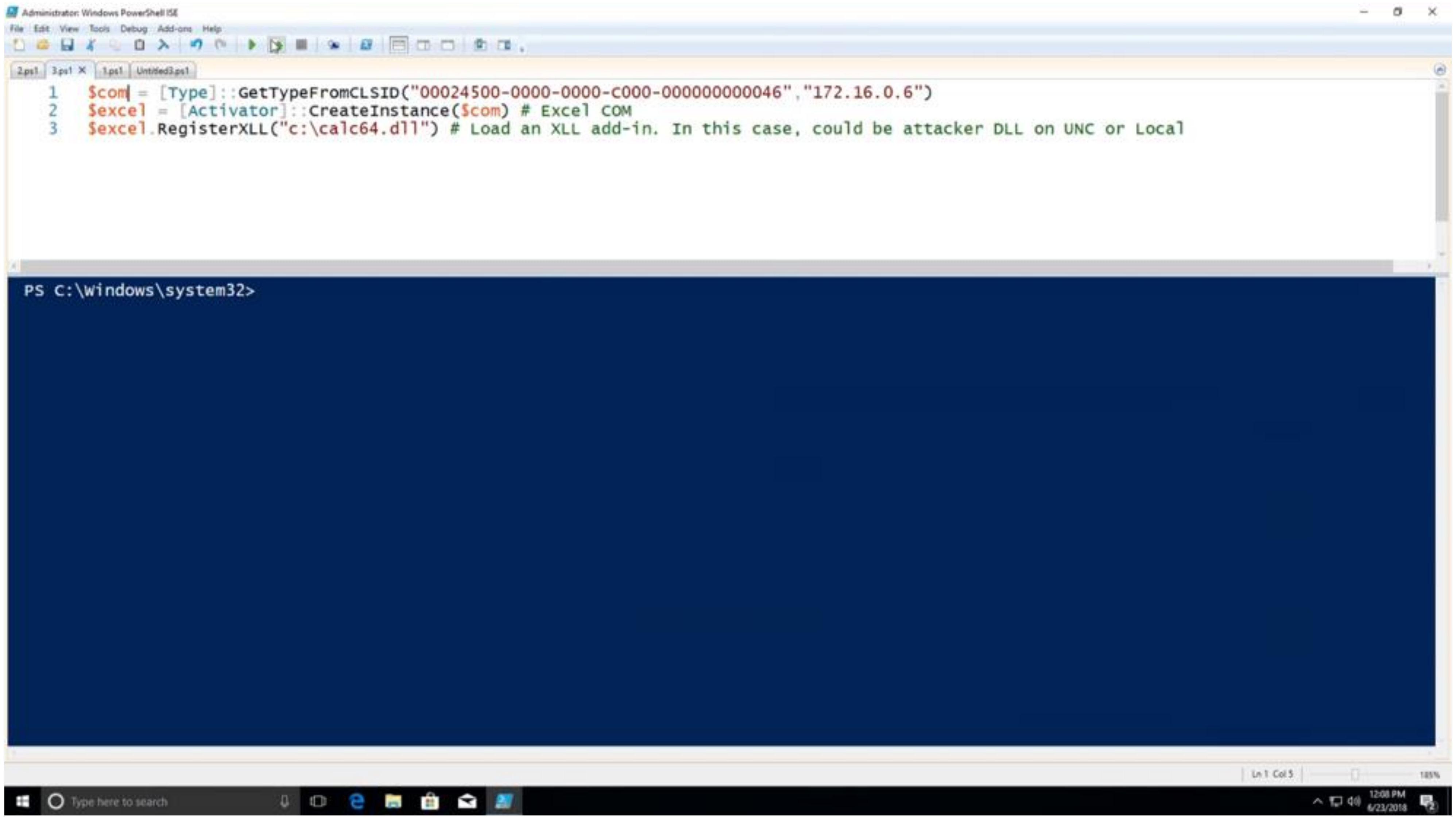


Administrator: Windows PowerShell ISE

```
$com = [Type]::GetTypeFromProgId('MMC20.Application','172.16.0.6')
$obj = [Activator]::CreateInstance($com)
$obj.Document.ActiveView.ExecuteShellCommand("c:\windows\system32\calc.exe",$null,$null,7)
```

```
PS C:\Windows\system32>
```

Fun Technique #2 — Excel XLL Registration



```
$com = [Type]::GetTypeFromCLSID("00024500-0000-0000-C000-000000000046","172.16.0.6")
$excel = [Activator]::CreateInstance($com) # Excel COM
$excel.RegisterXLL("c:\calc64.dll") # Load an XLL add-in. In this case, could be attacker DLL on UNC or Local
```

```
PS C:\Windows\system32>
```

https://gist.github.com/ryhanson/227229866af52e2d963cf941af135a52

Behavior Analysis

Finding Abuse Objects

Which objects of are particular interest:

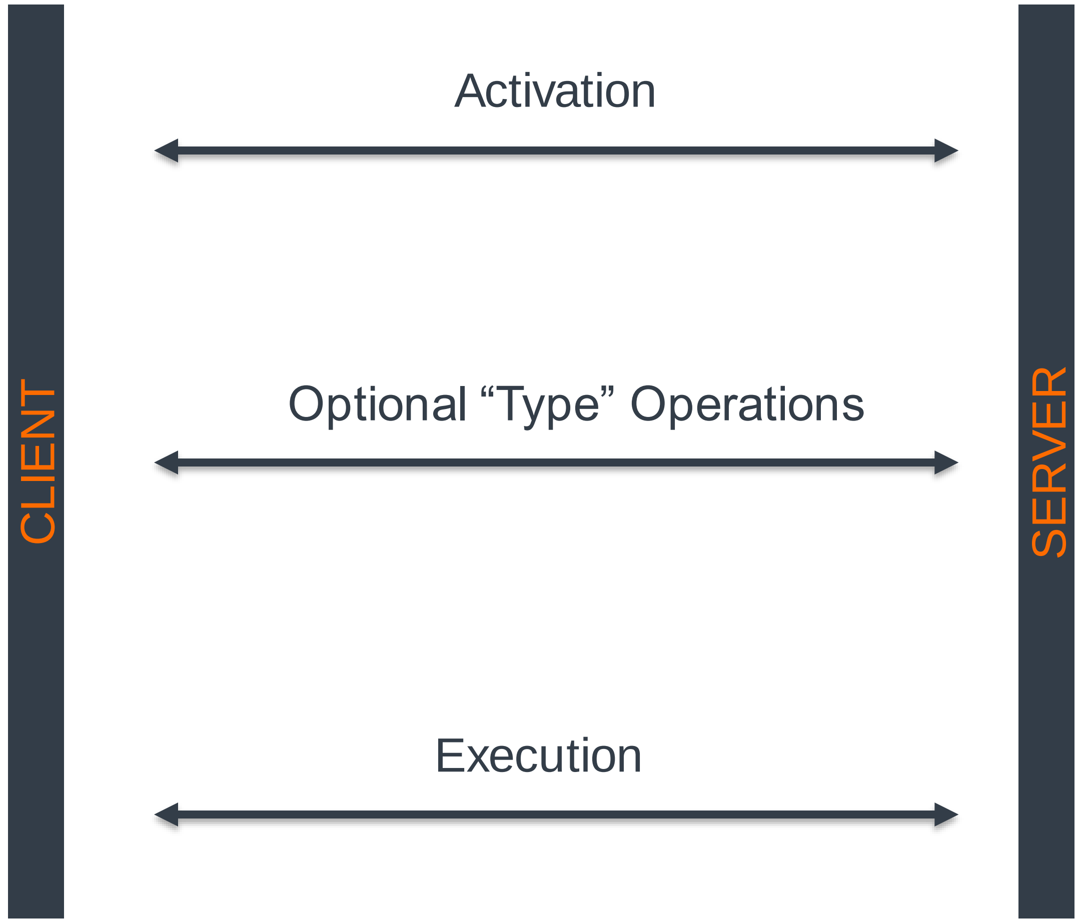
- **Objects that have functionality to load or execute code**
- Objects with no explicit “Launch Permission”
- Objects that implement IDispatch

Methods of exploring COM classes:

- Registry (HKEY Classes Root)
- OleView .NET
- Stumbling around system directories

DCOM objects available on a system can be enumerated via Win32_DCOMApplication WMI Class!

DCOM Behavior Overview



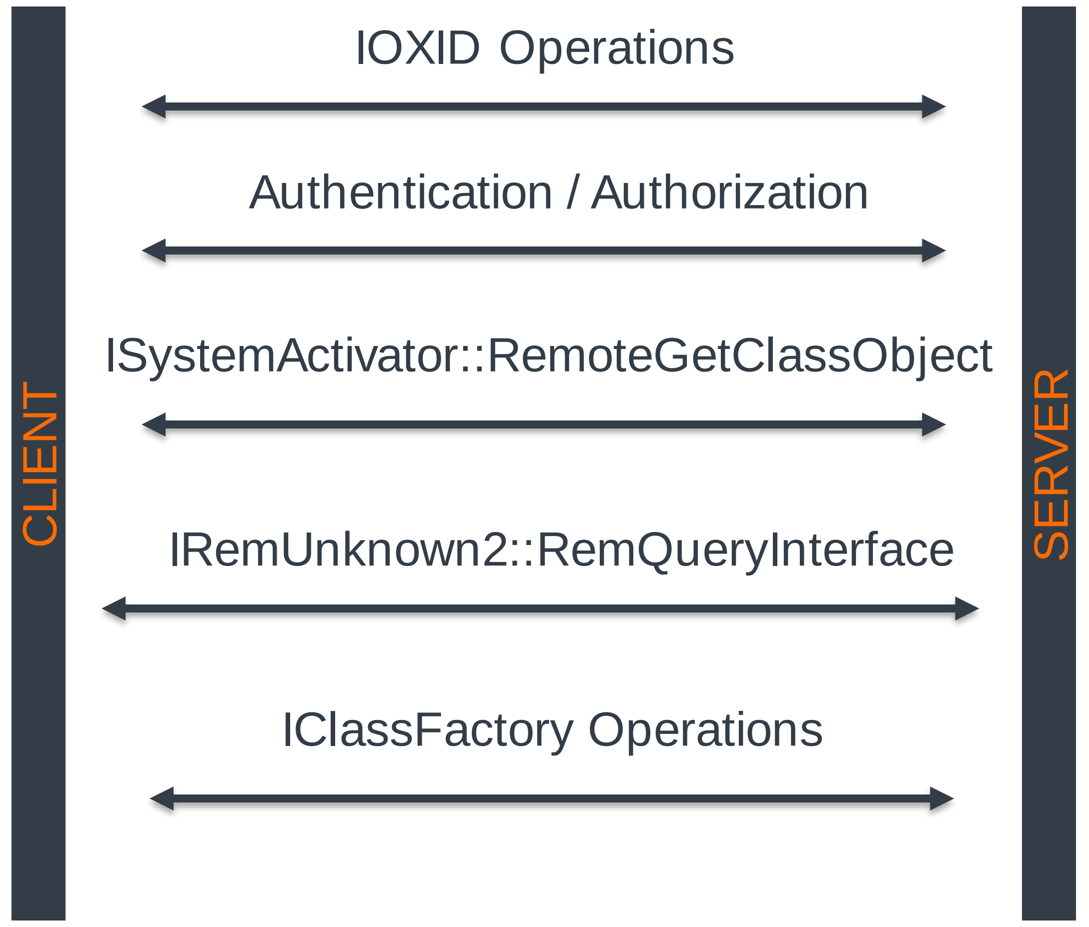
Object is instantiated on remote host

Function information is resolved

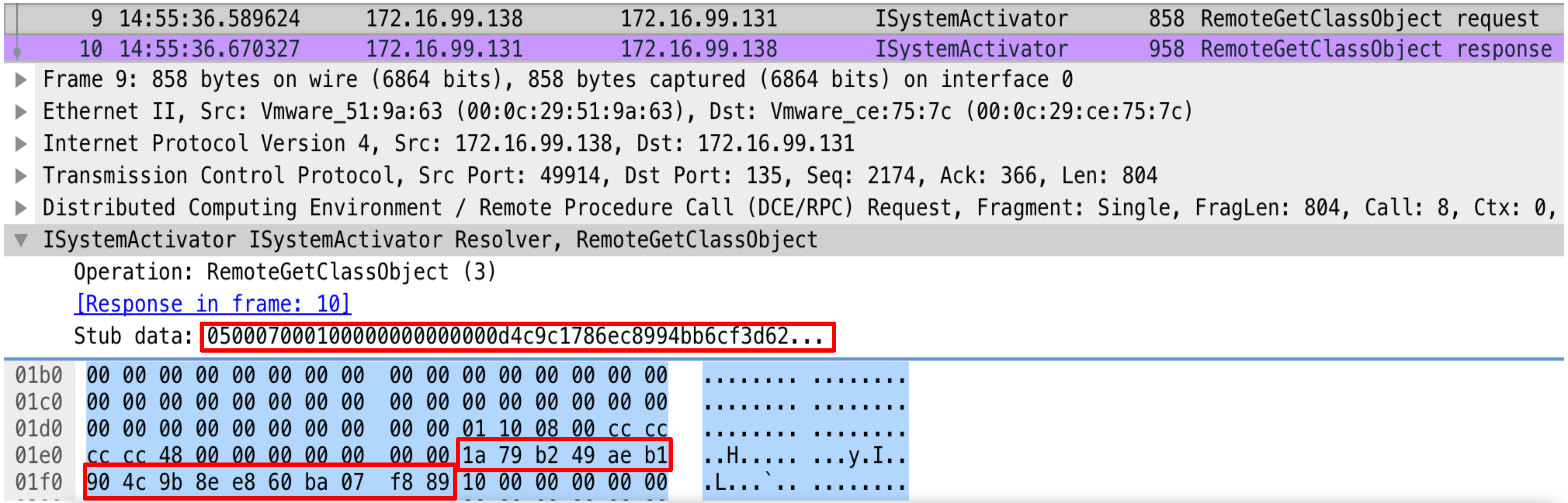
Selected function is executed

Breakdown Part 1 – Activation of MMC20.Application Using PowerShell

The activation phase was the first observed interactions on the network.

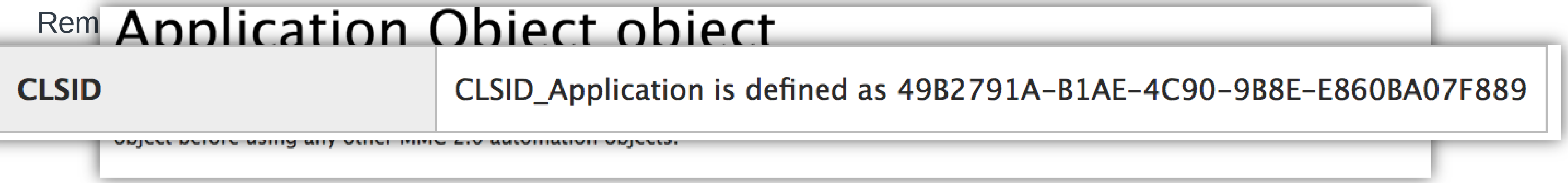

CLIENT
SERVER
IOXID Operations
Authentication / Authorization
ISystemActivator::RemoteGetClassObject
IRemUnknown2::RemQueryInterface
IClassFactory Operations

Breakdown Part 2 - RemoteGetClassObject



```
  9 14:55:36.589624   172.16.99.138   172.16.99.131   ISystemActivator   858 RemoteGetClassObject request
 10 14:55:36.670327   172.16.99.131   172.16.99.138   ISystemActivator   958 RemoteGetClassObject response
▶ Frame 9: 858 bytes on wire (6864 bits), 858 bytes captured (6864 bits) on interface 0
▶ Ethernet II, Src: Vmware_51:9a:63 (00:0c:29:51:9a:63), Dst: Vmware_ce:75:7c (00:0c:29:ce:75:7c)
▶ Internet Protocol Version 4, Src: 172.16.99.138, Dst: 172.16.99.131
▶ Transmission Control Protocol, Src Port: 49914, Dst Port: 135, Seq: 2174, Ack: 366, Len: 804
▶ Distributed Computing Environment / Remote Procedure Call (DCE/RPC) Request, Fragment: Single, FragLen: 804, Call: 8, Ctx: 0,
▼ ISystemActivator ISystemActivator Resolver, RemoteGetClassObject
    Operation: RemoteGetClassObject (3)
    [Response in frame: 10]
    Stub data: 05000700010000000000000000d4c9c1786ec8994bb6cf3d62...

01b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01d0  00 00 00 00 00 00 00 00  00 00 01 10 08 00 cc cc   ........ ........
01e0  cc cc 48 00 00 00 00 00  00 00 1a 79 b2 49 ae b1   ..H..... ...y.I..
01f0  90 4c 9b 8e e8 60 ba 07  f8 89 10 00 00 00 00 00   .L...`.. ........
```



Rem

Application Object object

CLSID	CLSID_Application is defined as 49B2791A-B1AE-4C90-9B8E-E860BA07F889

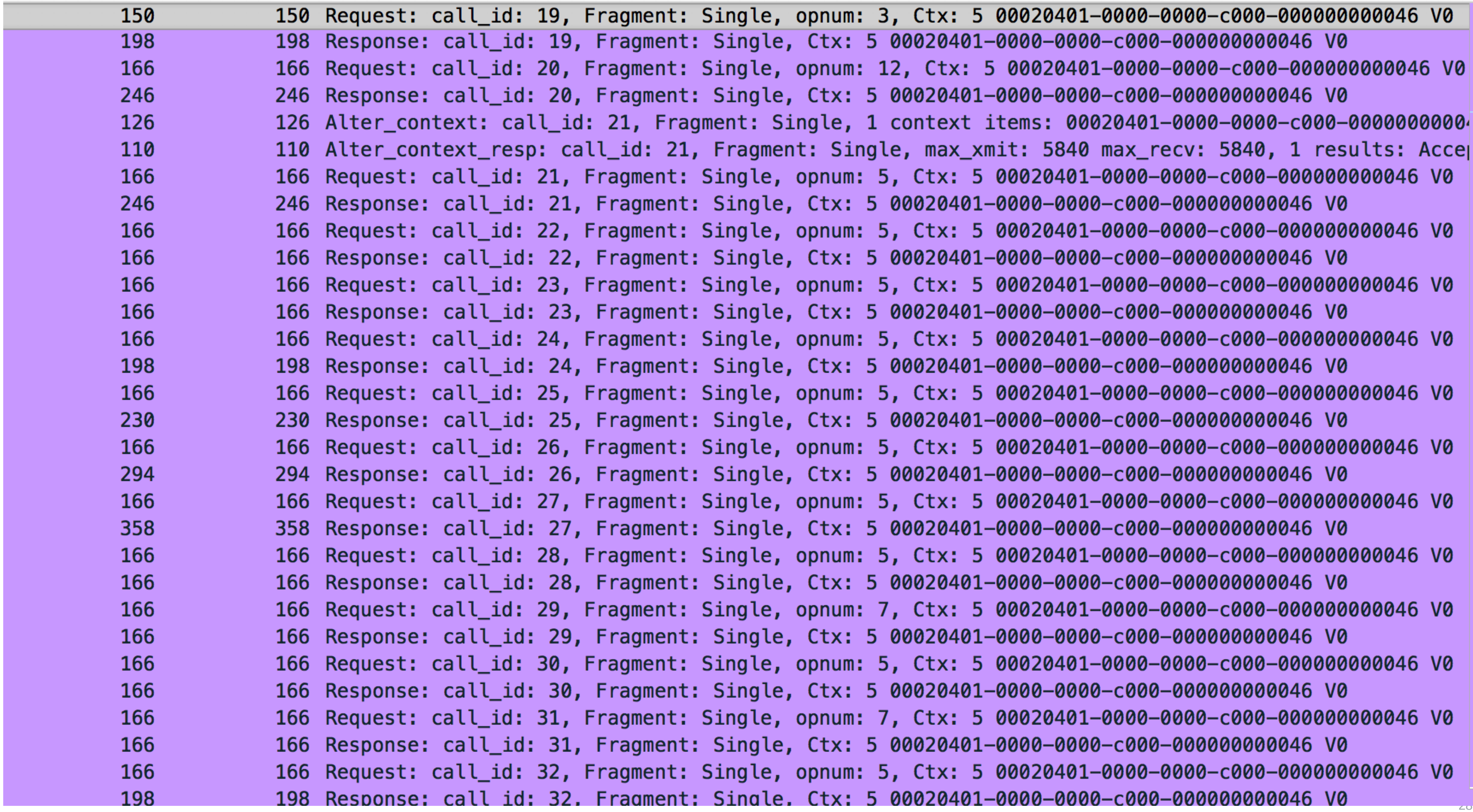
object before using any other MMC 2.0 automation objects.

https://msdn.microsoft.com/en-us/library/aa814545(v=vs.85).aspx

Breakdown Part 2 – ITypeInfo Operations

ITypeInfo - Interface that provides function and type information to the user. It is part of COM "automation". Automatically provided through IDispatch.

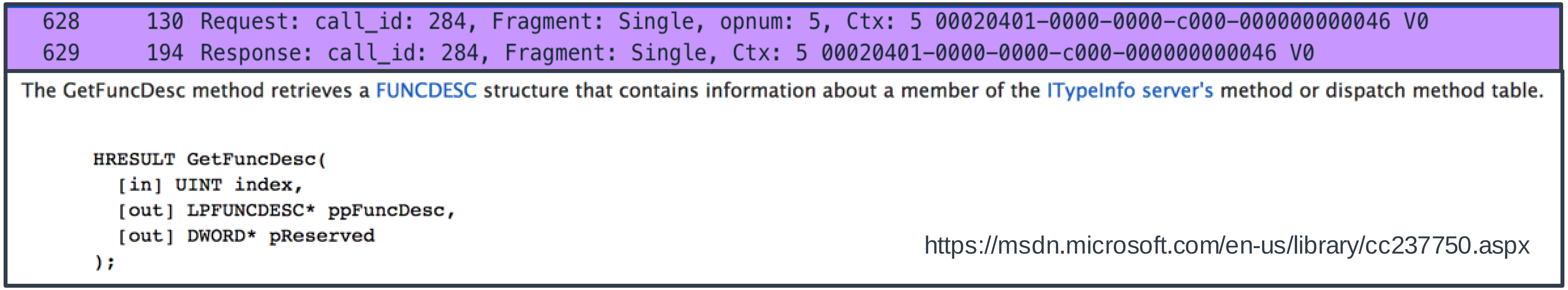
Common ITypeInfo Ops	**Function Name**	**Purpose**
	GetFuncDesc	Function Resolution
	GetNames	Property / Function name resolution
	GetDocumentation	Help Documentation retrieval



```
150      150 Request: call_id: 19, Fragment: Single, opnum: 3, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
198      198 Response: call_id: 19, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
166      166 Request: call_id: 20, Fragment: Single, opnum: 12, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
246      246 Response: call_id: 20, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
126      126 Alter_context: call_id: 21, Fragment: Single, 1 context items: 00020401-0000-0000-c000-0000000000
110      110 Alter_context_resp: call_id: 21, Fragment: Single, max_xmit: 5840 max_recv: 5840, 1 results: Acce
166      166 Request: call_id: 21, Fragment: Single, opnum: 5, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
246      246 Response: call_id: 21, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
166      166 Request: call_id: 22, Fragment: Single, opnum: 5, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
166      166 Response: call_id: 22, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
166      166 Request: call_id: 23, Fragment: Single, opnum: 5, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
166      166 Response: call_id: 23, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
166      166 Request: call_id: 24, Fragment: Single, opnum: 5, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
198      198 Response: call_id: 24, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
166      166 Request: call_id: 25, Fragment: Single, opnum: 5, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
230      230 Response: call_id: 25, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
166      166 Request: call_id: 26, Fragment: Single, opnum: 5, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
294      294 Response: call_id: 26, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
166      166 Request: call_id: 27, Fragment: Single, opnum: 5, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
358      358 Response: call_id: 27, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
166      166 Request: call_id: 28, Fragment: Single, opnum: 5, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
166      166 Response: call_id: 28, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
166      166 Request: call_id: 29, Fragment: Single, opnum: 7, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
166      166 Response: call_id: 29, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
166      166 Request: call_id: 30, Fragment: Single, opnum: 5, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
166      166 Response: call_id: 30, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
166      166 Request: call_id: 31, Fragment: Single, opnum: 7, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
166      166 Response: call_id: 31, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
166      166 Request: call_id: 32, Fragment: Single, opnum: 5, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
198      198 Response: call_id: 32, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
```

Breakdown Part 2 – ITypeInfo Operations

ITypeInfo:GetFuncDesc (5)



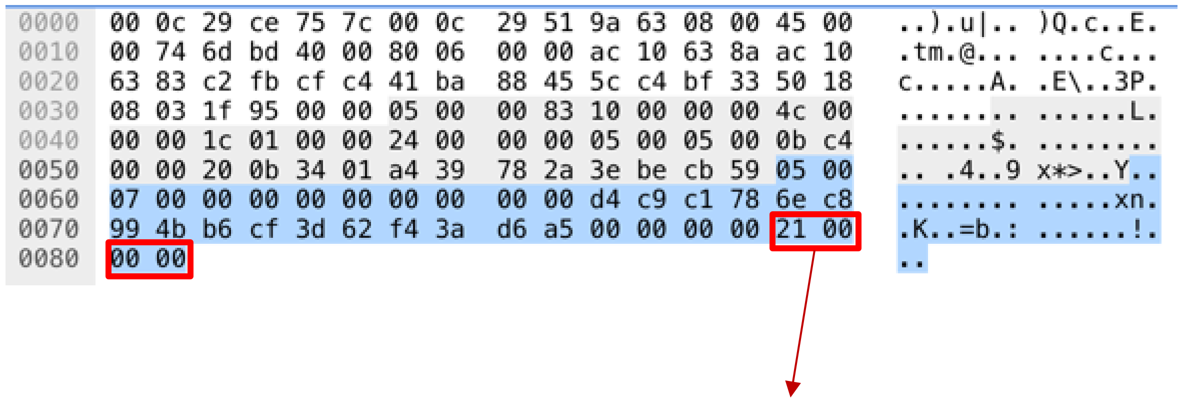
```
628     130 Request: call_id: 284, Fragment: Single, opnum: 5, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
629     194 Response: call_id: 284, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
```

The GetFuncDesc method retrieves a FUNCDESC structure that contains information about a member of the ITypeInfo server's method or dispatch method table.

```
HRESULT GetFuncDesc(
  [in] UINT index,
  [out] LPFUNCDESC* ppFuncDesc,
  [out] DWORD* pReserved
);
```

https://msdn.microsoft.com/en-us/library/cc237750.aspx

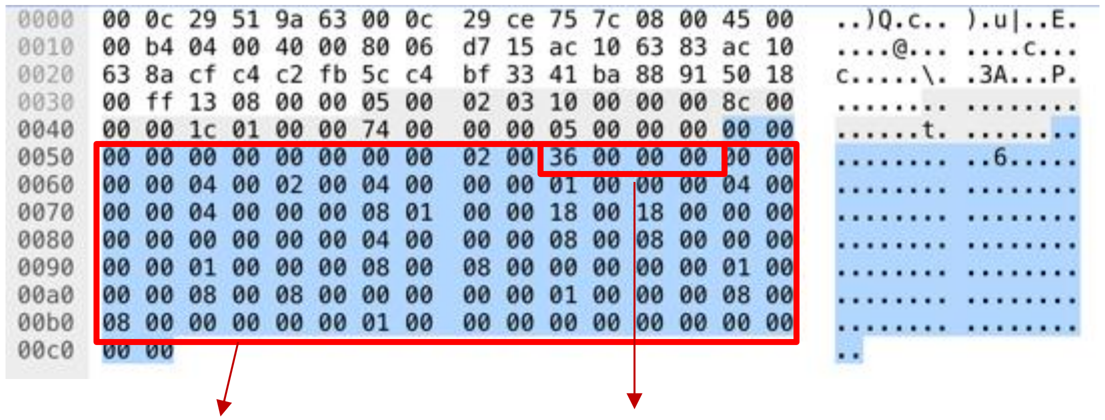
Request



```
0000  00 0c 29 ce 75 7c 00 0c  29 51 9a 63 08 00 45 00   ..).u|.. )Q.c..E.
0010  00 74 6d bd 40 00 80 06  00 00 ac 10 63 8a ac 10   .tm.@... ....c...
0020  63 83 c2 fb cf c4 41 ba  88 45 5c c4 bf 33 50 18   c.....A. .E\..3P.
0030  08 03 1f 95 00 00 05 00  00 83 10 00 00 00 4c 00   ........ ......L.
0040  00 00 1c 01 00 00 24 00  00 00 05 00 05 00 0b c4   ......$. ........
0050  00 00 20 0b 34 01 a4 39  78 2a 3e be cb 59 05 00   .. .4..9 x*>..Y..
0060  07 00 00 00 00 00 00 00  00 00 d4 c9 c1 78 6e c8   ........ ......xn.
0070  99 4b b6 cf 3d 62 f4 3a  d6 a5 00 00 00 00 21 00   .K..=b.: ......!.
0080  00 00                                              ..
```

Function Index / Ordinal

Response



```
0000  00 0c 29 51 9a 63 00 0c  29 ce 75 7c 08 00 45 00   ..)Q.c.. ).u|..E.
0010  00 b4 04 00 40 00 80 06  d7 15 ac 10 63 83 ac 10   ....@... ....c...
0020  63 8a cf c4 c2 fb 5c c4  bf 33 41 ba 88 91 50 18   c.....\. .3A...P.
0030  00 ff 13 08 00 00 05 00  02 03 10 00 00 00 8c 00   ........ ........
0040  00 00 1c 01 00 00 74 00  00 00 05 00 00 00 00 00   ......t. ........
0050  00 00 00 00 00 00 00 00  02 00 36 00 00 00 00 00   ........ ..6.....
0060  00 00 04 00 02 00 04 00  00 00 01 00 00 00 04 00   ........ ........
0070  00 00 04 00 00 00 08 01  00 00 18 00 18 00 00 00   ........ ........
0080  00 00 00 00 00 00 04 00  00 00 08 00 08 00 00 00   ........ ........
0090  00 00 01 00 00 00 08 00  08 00 00 00 00 00 01 00   ........ ........
00a0  00 00 08 00 08 00 00 00  00 00 01 00 00 00 08 00   ........ ........
00b0  08 00 00 00 00 00 01 00  00 00 00 00 00 00 00 00   ........ ........
00c0  00 00                                              ..
```

FUNDESC

MEMBERID

Breakdown Part 2 – ITypeInfo Operations

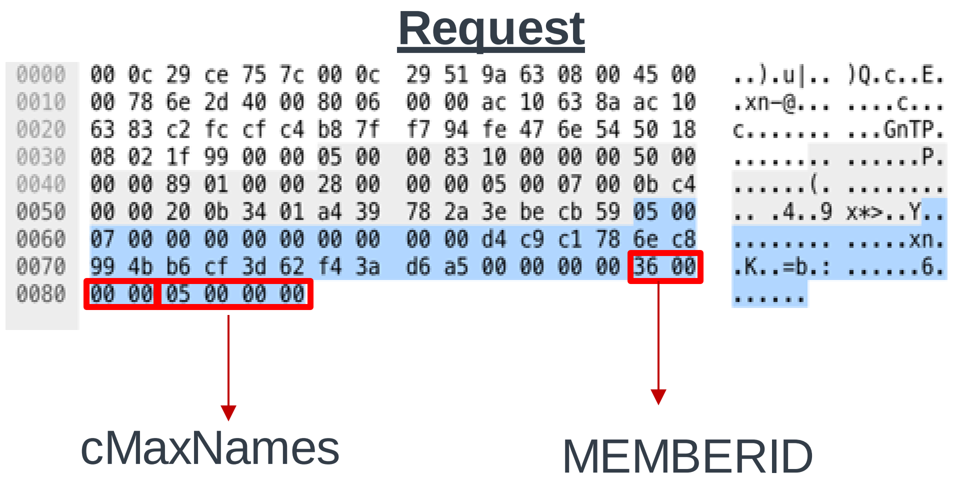
<u>ITypeInfo:GetNames (7)</u>

```
857      134 Request: call_id: 393, Fragment: Single, opnum: 7, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
858      306 Response: call_id: 393, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
```

The GetNames method retrieves the data member name or the method and parameter names associated with a specified MEMBERID.

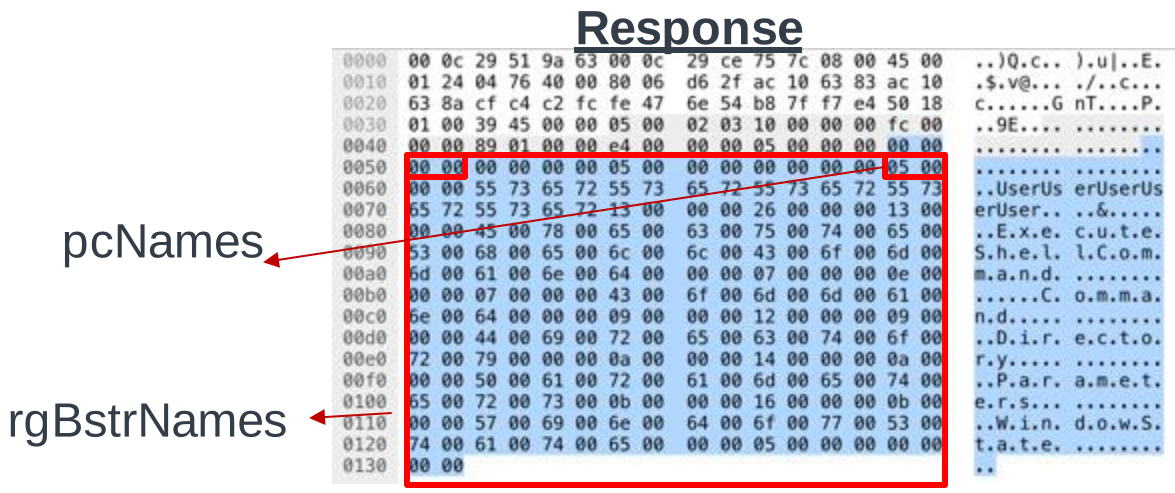
```
HRESULT GetNames(
  [in] MEMBERID memid,
  [out, size_is(cMaxNames), length_is(*pcNames)]
    BSTR* rgBstrNames,
  [in] UINT cMaxNames,
  [out] UINT* pcNames
);
```

https://msdn.microsoft.com/en-us/library/cc237753.aspx



Request

```
0000  00 0c 29 ce 75 7c 00 0c  29 51 9a 63 08 00 45 00   ..).u|.. )Q.c..E.
0010  00 78 6e 2d 40 00 80 06  00 00 ac 10 63 8a ac 10   .xn-@... ....c...
0020  63 83 c2 fc cf c4 b8 7f  f7 94 fe 47 6e 54 50 18   c....... ...GnTP.
0030  08 02 1f 99 00 00 05 00  00 83 10 00 00 00 50 00   ........ ......P.
0040  00 00 89 01 00 00 28 00  00 00 05 00 07 00 0b c4   ......(. ........
0050  00 00 20 0b 34 01 a4 39  78 2a 3e be cb 59 05 00   .. .4..9 x*>..Y..
0060  07 00 00 00 00 00 00 00  00 00 d4 c9 c1 78 6e c8   ........ ......xn.
0070  99 4b b6 cf 3d 62 f4 3a  d6 a5 00 00 00 00 36 00   .K..=b.: ......6.
0080  00 00 05 00 00 00                                   ......
```

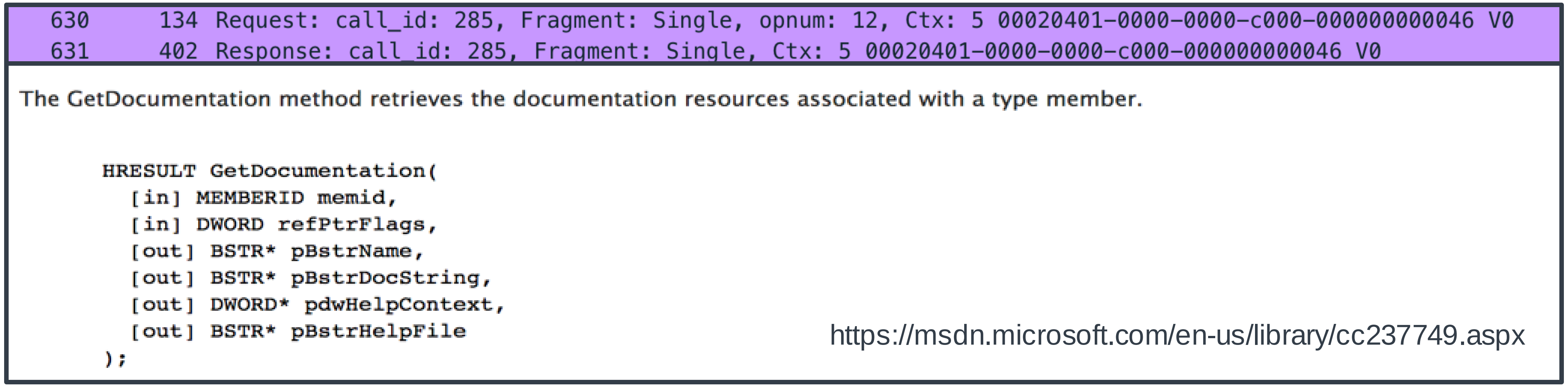


Response

```
0000  00 0c 29 51 9a 63 00 0c  29 ce 75 7c 08 00 45 00   ..)Q.c.. ).u|..E.
0010  01 24 04 76 40 00 80 06  d6 2f ac 10 63 83 ac 10   .$.v@... ./..c...
0020  63 8a cf c4 c2 fc fe 47  6e 54 b8 7f f7 e4 50 18   c......G nT....P.
0030  01 00 39 45 00 00 05 00  02 03 10 00 00 00 fc 00   ..9E.... ........
0040  00 00 89 01 00 00 e4 00  00 00 05 00 00 00 00 00   ........ ........
0050  00 00 00 00 00 00 05 00  00 00 00 00 00 00 05 00   ........ ........
0060  00 00 55 73 65 72 55 73  65 72 55 73 65 72 55 73   ..UserUs erUserUs
0070  65 72 55 73 65 72 13 00  00 00 26 00 00 00 13 00   erUser.. ..&.....
0080  00 00 45 00 78 00 65 00  63 00 75 00 74 00 65 00   ..E.x.e. c.u.t.e.
0090  53 00 68 00 65 00 6c 00  6c 00 43 00 6f 00 6d 00   S.h.e.l. l.C.o.m.
00a0  6d 00 61 00 6e 00 64 00  00 00 07 00 00 00 0e 00   m.a.n.d. ........
00b0  00 00 07 00 00 00 43 00  6f 00 6d 00 6d 00 61 00   ......C. o.m.m.a.
00c0  6e 00 64 00 00 00 09 00  00 00 12 00 00 00 09 00   n.d..... ........
00d0  00 00 44 00 69 00 72 00  65 00 63 00 74 00 6f 00   ..D.i.r. e.c.t.o.
00e0  72 00 79 00 00 00 0a 00  00 00 14 00 00 00 0a 00   r.y..... ........
00f0  00 00 50 00 61 00 72 00  61 00 6d 00 65 00 74 00   ..P.a.r. a.m.e.t.
0100  65 00 72 00 73 00 0b 00  00 00 16 00 00 00 0b 00   e.r.s... ........
0110  00 00 57 00 69 00 6e 00  64 00 6f 00 77 00 53 00   ..W.i.n. d.o.w.S.
0120  74 00 61 00 74 00 65 00  00 00 05 00 00 00 00 00   t.a.t.e. ........
0130  00 00                                               ..
```

Breakdown Part 2 – ITypeInfo Operations

ITypeInfo:GetDocumentation (12)



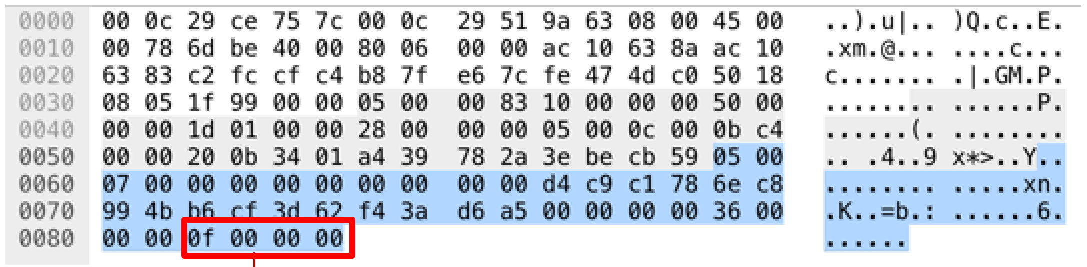
```
630      134 Request: call_id: 285, Fragment: Single, opnum: 12, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
631      402 Response: call_id: 285, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
```

The GetDocumentation method retrieves the documentation resources associated with a type member.

```
HRESULT GetDocumentation(
  [in] MEMBERID memid,
  [in] DWORD refPtrFlags,
  [out] BSTR* pBstrName,
  [out] BSTR* pBstrDocString,
  [out] DWORD* pdwHelpContext,
  [out] BSTR* pBstrHelpFile
);
```

https://msdn.microsoft.com/en-us/library/cc237749.aspx

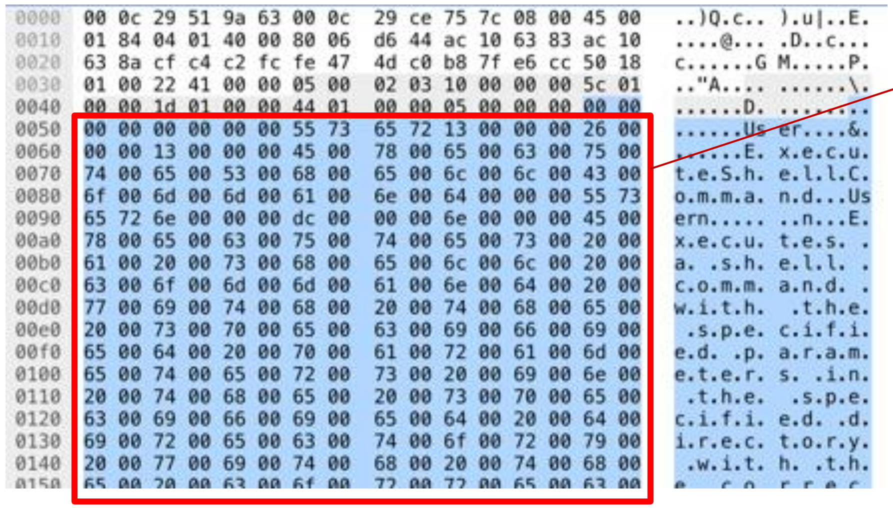
Request



```
0000  00 0c 29 ce 75 7c 00 0c  29 51 9a 63 08 00 45 00   ..).u|.. )Q.c..E.
0010  00 78 6d be 40 00 80 06  00 00 ac 10 63 8a ac 10   .xm.@... ....c...
0020  63 83 c2 fc cf c4 b8 7f  e6 7c fe 47 4d c0 50 18   c....... .|.GM.P.
0030  08 05 1f 99 00 00 05 00  00 83 10 00 00 00 50 00   ........ ......P.
0040  00 00 1d 01 00 00 28 00  00 00 05 00 0c 00 0b c4   ......(. ........
0050  00 00 20 0b 34 01 a4 39  78 2a 3e be cb 59 05 00   .. .4..9 x*>..Y..
0060  07 00 00 00 00 00 00 00  00 00 d4 c9 c1 78 6e c8   ........ .....xn.
0070  99 4b b6 cf 3d 62 f4 3a  d6 a5 00 00 00 00 36 00   .K..=b.: ......6.
0080  00 00 0f 00 00 00                                  ......
```

RefPtrFlags = 0x0000000f = HelpFileArg | DocStringArg

Response



```
0000  00 0c 29 51 9a 63 00 0c  29 ce 75 7c 08 00 45 00   ..)Q.c.. ).u|..E.
0010  01 84 04 01 40 00 80 06  d6 44 ac 10 63 83 ac 10   ....@... .D..c...
0020  63 8a cf c4 c2 fc fe 47  4d c0 b8 7f e6 cc 50 18   c......G M.....P.
0030  01 00 22 41 00 00 05 00  02 03 10 00 00 00 5c 01   .."A.... ......\.
0040  00 00 1d 01 00 00 44 01  00 00 05 00 00 00 00 00   ......D. ........
0050  00 00 00 00 00 00 55 73  65 72 13 00 00 00 26 00   ......Us er....&.
0060  00 00 13 00 00 00 45 00  78 00 65 00 63 00 75 00   ......E. x.e.c.u.
0070  74 00 65 00 53 00 68 00  65 00 6c 00 6c 00 43 00   t.e.S.h. e.l.l.C.
0080  6f 00 6d 00 6d 00 61 00  6e 00 64 00 00 00 55 73   o.m.m.a. n.d...Us
0090  65 72 6e 00 00 00 dc 00  00 00 6e 00 00 00 45 00   ern..... ..n...E.
00a0  78 00 65 00 63 00 75 00  74 00 65 00 73 00 20 00   x.e.c.u. t.e.s. .
00b0  61 00 20 00 73 00 68 00  65 00 6c 00 6c 00 20 00   a. .s.h. e.l.l. .
00c0  63 00 6f 00 6d 00 6d 00  61 00 6e 00 64 00 20 00   c.o.m.m. a.n.d. .
00d0  77 00 69 00 74 00 68 00  20 00 74 00 68 00 65 00   w.i.t.h.  .t.h.e.
00e0  20 00 73 00 70 00 65 00  63 00 69 00 66 00 69 00    .s.p.e. c.i.f.i.
00f0  65 00 64 00 20 00 70 00  61 00 72 00 61 00 6d 00   e.d. .p. a.r.a.m.
0100  65 00 74 00 65 00 72 00  73 00 20 00 69 00 6e 00   e.t.e.r. s. .i.n.
0110  20 00 74 00 68 00 65 00  20 00 73 00 70 00 65 00    .t.h.e.  .s.p.e.
0120  63 00 69 00 66 00 69 00  65 00 64 00 20 00 64 00   c.i.f.i. e.d. .d.
0130  69 00 72 00 65 00 63 00  74 00 6f 00 72 00 79 00   i.r.e.c. t.o.r.y.
0140  20 00 77 00 69 00 74 00  68 00 20 00 74 00 68 00    .w.i.t. h. .t.h.
0150  65 00 20 00 63 00 6f 00  72 00 72 00 65 00 63 00   e. .c.o. r.r.e.c.
```

Help Docs Output

Breakdown Part 3 - Execution

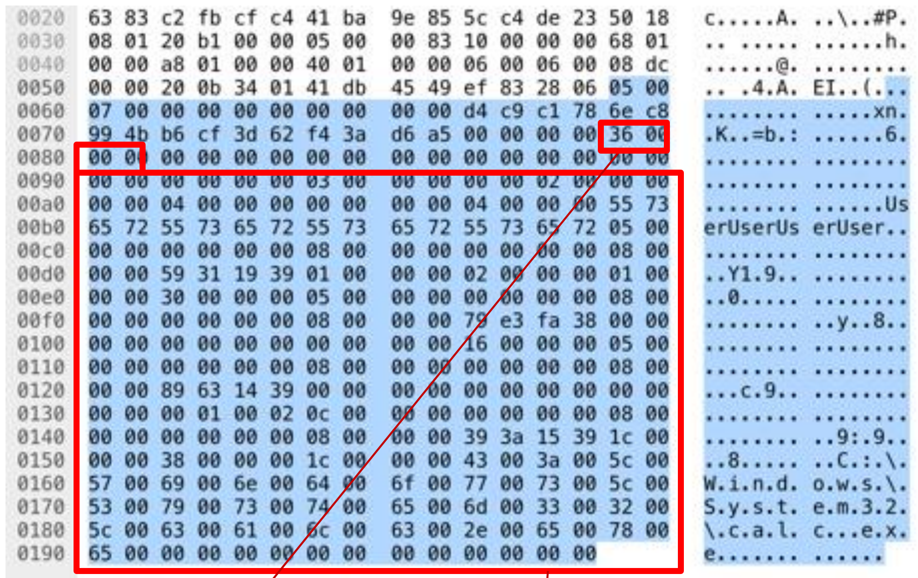
IDispatch:Invoke (6)

```
919     414  Invoke request ID=0x36 Method PropertyGet Args=4 NamedArgs=0 VarRef=0
924     194  Invoke response SCode=S_OK VarRef=0 -> S_OK
```

The Invoke method provides access to properties and methods exposed by the automation server.

```
HRESULT Invoke(
  [in] DISPID dispIdMember,
  [in] REFIID riid,
  [in] LCID lcid,
  [in] DWORD dwFlags,
  [in] DISPPARAMS* pDispParams,
  [out] VARIANT* pVarResult,
  [out] EXCEPINFO* pExcepInfo,
  [out] UINT* pArgErr,
  [in] UINT cVarRef,
  [in, size_is(cVarRef)] UINT* rgVarRefIdx,
  [in, out, size_is(cVarRef)] VARIANT* rgVarRef
);
```

Request

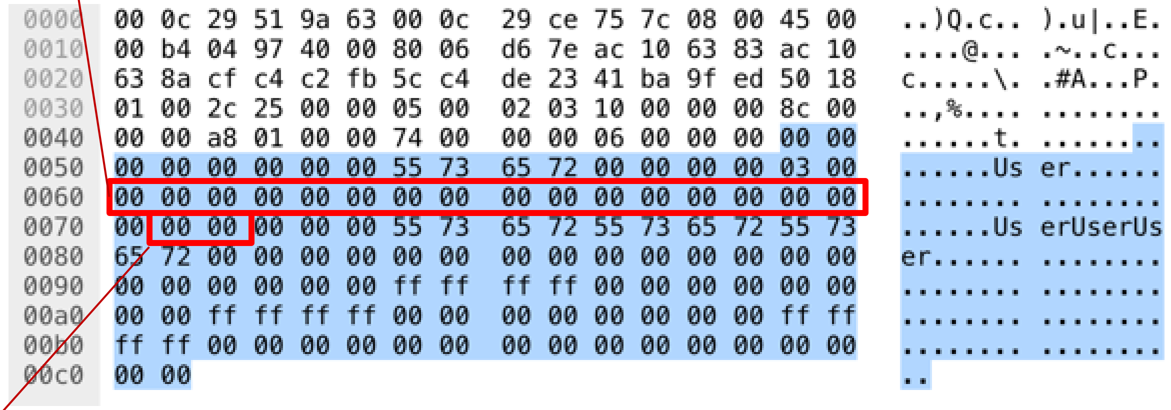


```
0020  63 83 c2 fb cf c4 41 ba  9e 85 5c c4 de 23 50 18   c.....A. ..\..#P.
0030  08 01 20 b1 00 00 05 00  00 83 10 00 00 00 68 01   .. ..... ......h.
0040  00 00 a8 01 00 00 40 01  00 00 06 00 06 00 08 dc   ......@. ........
0050  00 00 20 0b 34 01 41 db  45 49 ef 83 28 06 05 00   .. .4.A. EI..(...
0060  07 00 00 00 00 00 00 00  00 00 d4 c9 c1 78 6e c8   ........ ......xn.
0070  99 4b b6 cf 3d 62 f4 3a  d6 a5 00 00 00 00 36 00   .K..=b.: ......6.
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0090  00 00 00 00 00 00 03 00  00 00 00 00 02 00 00 00   ........ ........
00a0  00 00 04 00 00 00 00 00  00 00 04 00 00 00 55 73   ........ ......Us
00b0  65 72 55 73 65 72 55 73  65 72 55 73 65 72 05 00   erUserUs erUser..
00c0  00 00 00 00 00 00 08 00  00 00 00 00 00 00 08 00   ........ ........
00d0  00 00 59 31 19 39 01 00  00 00 02 00 00 00 01 00   ..Y1.9.. ........
00e0  00 00 30 00 00 00 05 00  00 00 00 00 00 00 08 00   ..0..... ........
00f0  00 00 00 00 00 00 08 00  00 00 79 e3 fa 38 00 00   ........ ..y..8..
0100  00 00 00 00 00 00 00 00  00 00 16 00 00 00 05 00   ........ ........
0110  00 00 00 00 00 00 08 00  00 00 00 00 00 00 08 00   ........ ........
0120  00 00 89 63 14 39 00 00  00 00 00 00 00 00 00 00   ...c.9.. ........
0130  00 00 00 01 00 02 0c 00  00 00 00 00 00 00 08 00   ........ ........
0140  00 00 00 00 00 00 08 00  00 00 39 3a 15 39 1c 00   ........ ..9:.9..
0150  00 00 38 00 00 00 1c 00  00 00 43 00 3a 00 5c 00   ..8..... ..C.:.\.
0160  57 00 69 00 6e 00 64 00  6f 00 77 00 73 00 5c 00   W.i.n.d. o.w.s.\.
0170  53 00 79 00 73 00 74 00  65 00 6d 00 33 00 32 00   S.y.s.t. e.m.3.2.
0180  5c 00 63 00 61 00 6c 00  63 00 2e 00 65 00 78 00   \.c.a.l. c...e.x.
0190  65 00 00 00 00 00 00 00  00 00 00 00 00 00         e....... ......
```

dispIdMember

pDispParams

pvarResult - Null



```
0000  00 0c 29 51 9a 63 00 0c  29 ce 75 7c 08 00 45 00   ..)Q.c.. ).u|..E.
0010  00 b4 04 97 40 00 80 06  d6 7e ac 10 63 83 ac 10   ....@... .~..c...
0020  63 8a cf c4 c2 fb 5c c4  de 23 41 ba 9f ed 50 18   c.....\. .#A...P.
0030  01 00 2c 25 00 00 05 00  02 03 10 00 00 00 8c 00   ..,%.... ........
0040  00 00 a8 01 00 00 74 00  00 00 06 00 00 00 00 00   ......t. ........
0050  00 00 00 00 00 00 55 73  65 72 00 00 00 00 03 00   ......Us er......
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0070  00 00 00 00 00 00 55 73  65 72 55 73 65 72 55 73   ......Us erUserUs
0080  65 72 00 00 00 00 00 00  00 00 00 00 00 00 00 00   er...... ........
0090  00 00 00 00 00 00 ff ff  ff ff 00 00 00 00 00 00   ........ ........
00a0  00 00 ff ff ff ff 00 00  00 00 00 00 00 00 ff ff   ........ ........
00b0  ff ff 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00c0  00 00                                              ..
```

pExecInfo – S_OK

Defining Loose & Strict Criteria

Strict Criteria

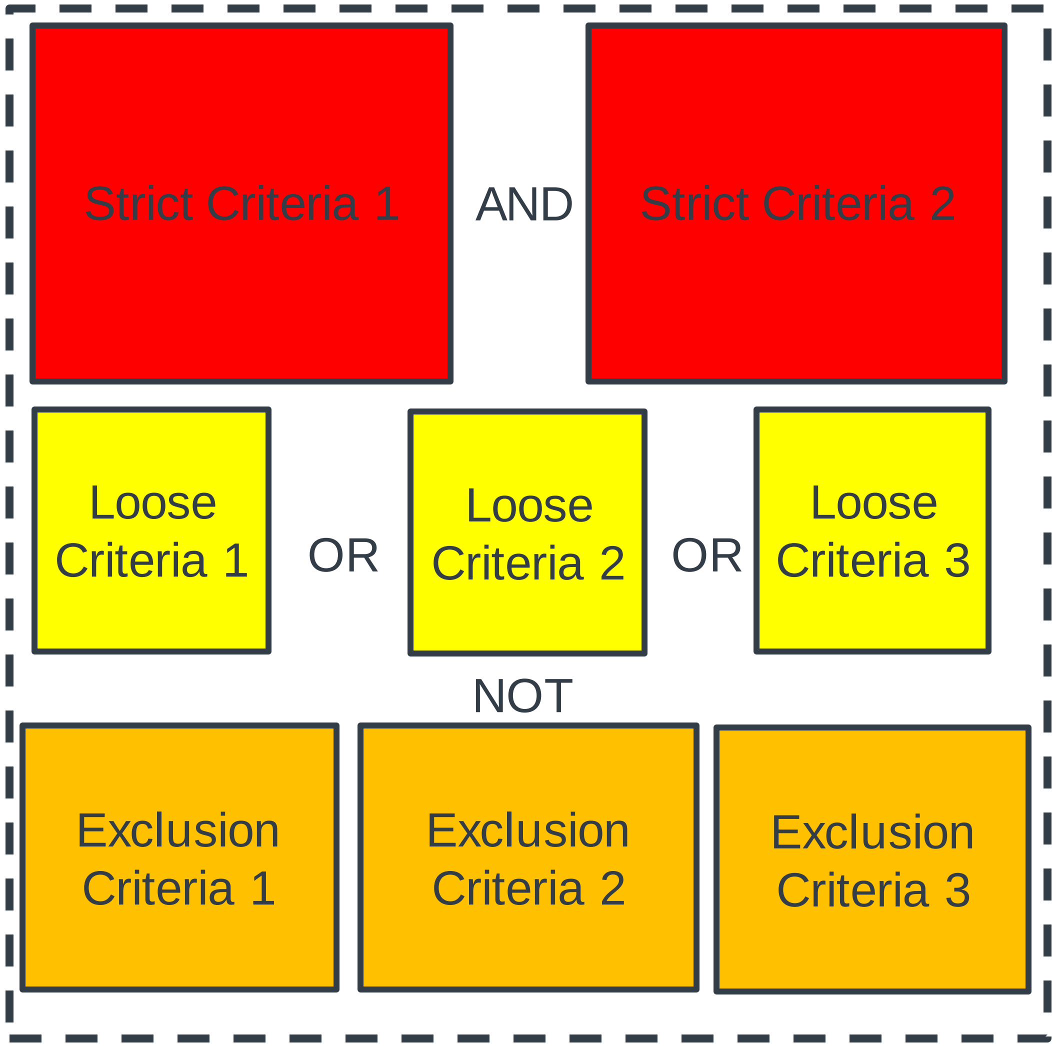
Components of a particular attack chain that are required to be present for the chain to exist.

Loose Criteria

Components of a particular attack chain that will commonly be present in the attack chain. Generally, at least one of these will be present. Also includes attacker behavior choices.

Exclusion Criteria

Values or components that appear to be part of an attack chain but are benign in nature and should be excluded from detection logic (aided by large data corpus).



Criteria Analysis

Strict

- Activation with relevant GUID / CLSID
- Some method of initiating execution

Exclusions

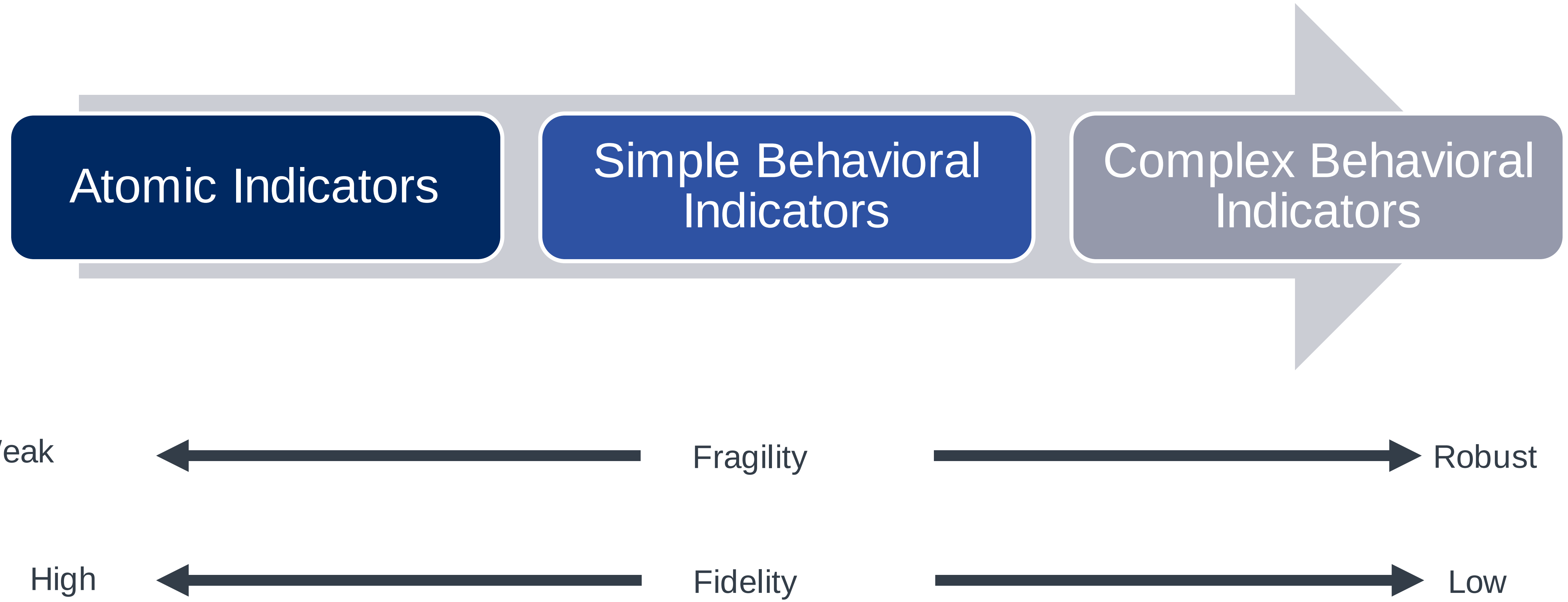
- Systems that normally utilize DCOM for operations. This might be common in certain development environments or on line-of-business applications.

Loose

- The RPC object used to execute code can vary (CLSID)
- There might be ITypeInfo operations if the COM object is explored
- There might be an IDispatch::Invoke with the Dispatch ID / MEMBERID of relevant function
- The object instantiation can vary between IClassFactor::CreateInstance or ISystemActivator::RemoteGetClassObject

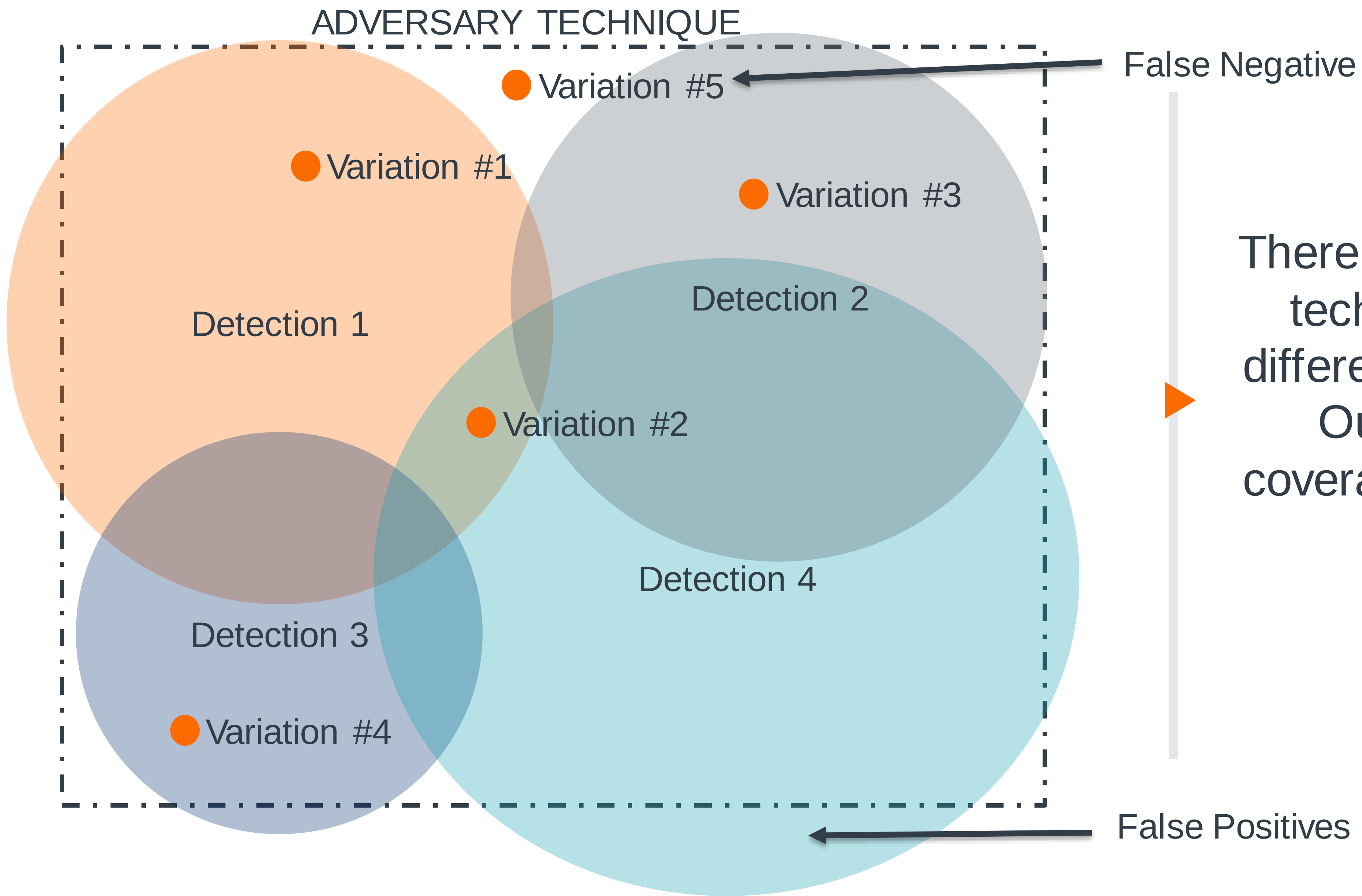
Network Detection & Analysis

Detection Spectrum



*Specificity could vary within each indicator type

Detection Coverage



There are many variations of techniques that result in different detectable artifacts. Our goal is to optimal coverage over the technique space.

NSM Collection Techniques

Rule Based IDS

- Inspects traffic, looks for rule matches, generates alert
- Only identifies “known bad”
- Lacks context but useful and easy step #1

Metadata Extraction

- Inspects traffic, extracts event metadata, logs events
- Limited by protocol parsers and fields available
- Enables time-series traffic analysis and detection

Full Content and PCAP

- Can be very tough to scale
- Can be very tough to use. Often paired with tool to “index” metadata
- Complete forensic data available (sort-of)

IDS Signature – Identifying DCOM Objects

```
alert tcp any any -> any 135 (msg:"Lateral Movement: MMC DCOM Object Created with RemoteGetClassObject"; flow:to_server, established; content:"|03|"; offset:22; depth:1; content:"MEOW"; offset:68; depth:4; content:"|1a 79 b2 49 ae b1 90 4c 9b 8e e8 60 ba 07 f8 89|"; offset:436; depth: 16; classtype: misc-attack; sid: XXXXXX; rev: XXXXXX;)
```

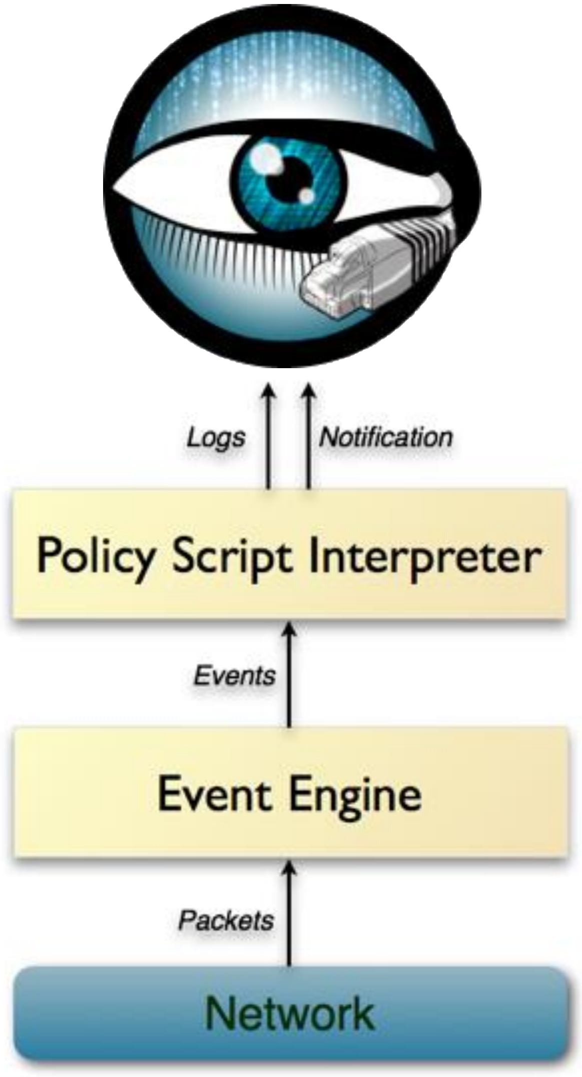
Let's look at the content blocks (red):

- content:"|03|" – Looking for DCERPC Opnum 3
- content:"MEOW" – Looking for characteristic MEOW signature present in these packets
- content:"|1a 79 b2 49 ae b1 90 4c 9b 8e e8 60 ba 07 f8 89|" – Looking for the GUID of the particular class I am interested in (MMC20.Application)

Metadata Extraction

Zeek – Open source network IDS that focuses on protocol parsing and extraction of relevant info

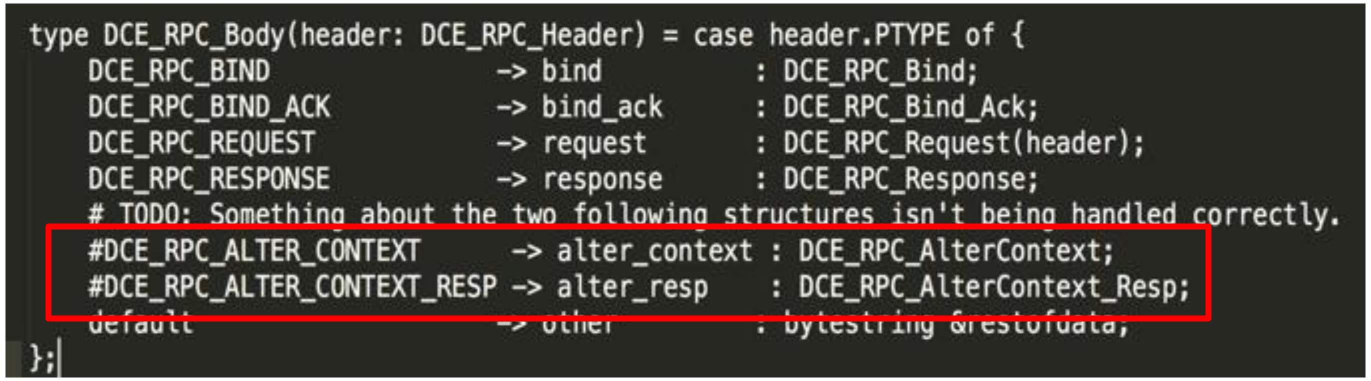
- **BinPAC** — High level language for protocol parsers.
- **Events** — Reduces traffic into “high level events”. Generated by parsers. Passes variables defined in BinPAC to event handlers.
- **Scripts** — Execute and handle event handlers to consume parser events.



https://www.bro.org/sphinx/intro/index.html

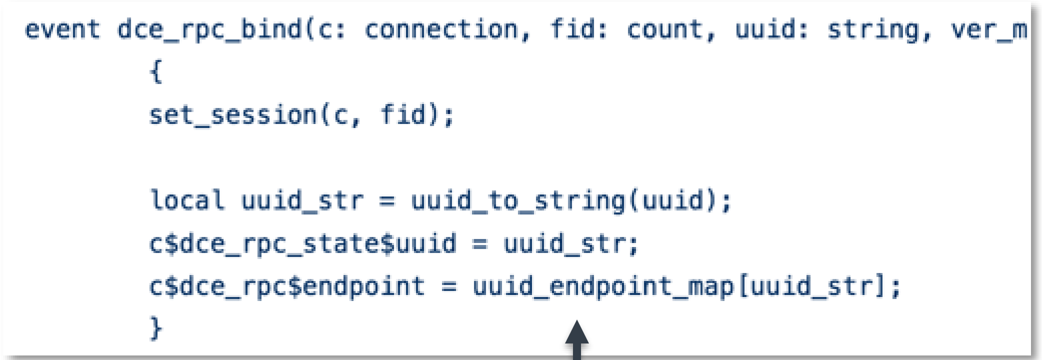
Current weaknesses in Zeek’s DCERPC parser?

- DCERPC metadata doesn’t identify object being called
- The parser will mislabel PDUs that come after an alter_context



```
type DCE_RPC_Body(header: DCE_RPC_Header) = case header.PTYPE of {
    DCE_RPC_BIND                -> bind          : DCE_RPC_Bind;
    DCE_RPC_BIND_ACK            -> bind_ack      : DCE_RPC_Bind_Ack;
    DCE_RPC_REQUEST             -> request       : DCE_RPC_Request(header);
    DCE_RPC_RESPONSE            -> response      : DCE_RPC_Response;
    # TODO: Something about the two following structures isn't being handled correctly.
    #DCE_RPC_ALTER_CONTEXT      -> alter_context : DCE_RPC_AlterContext;
    #DCE_RPC_ALTER_CONTEXT_RESP -> alter_resp    : DCE_RPC_AlterContext_Resp;
    default                     -> other         : bytestring &restofdata;
};
```

https://github.com/bro/bro/blob/master/src/analyzer/protocol/dce-rpc/dce_rpc-protocol.pac



```
event dce_rpc_bind(c: connection, fid: count, uuid: string, ver_m
        {
        set_session(c, fid);

        local uuid_str = uuid_to_string(uuid);
        c$dce_rpc_state$uuid = uuid_str;
        c$dce_rpc$endpoint = uuid_endpoint_map[uuid_str];
        }
```

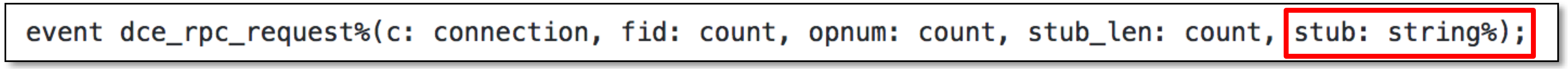
https://github.com/bro/bro/blob/master/scripts/base/protocols/dce-rpc/main.bro

Operation & Endpoint only set during BIND

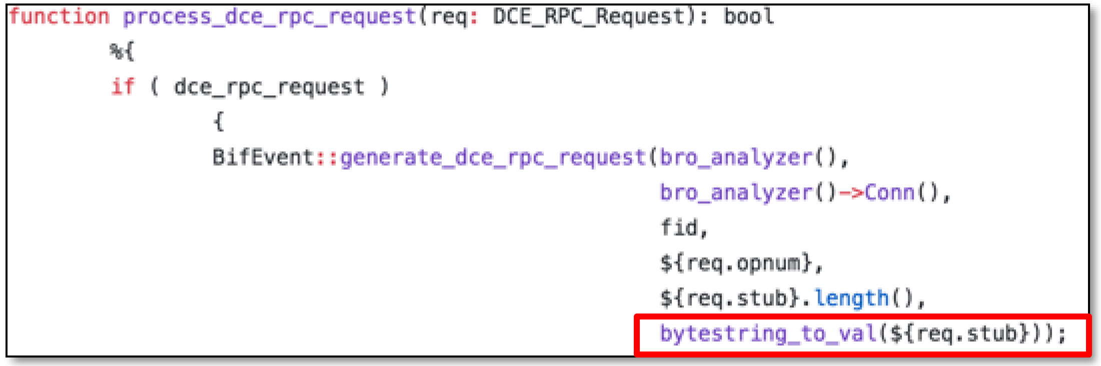
Metadata Extraction – Short Term Fix

Modify Zeek to parse CLSIDs during instantiation

- Add state tracking on DCE_RPC Requests
- Pass full “stub” during ***dce_rpc_request*** to the event handler
- Create Bro script to handle events during RemoteGetClassObject or RemoteCreateInstance operations and parse CLSID



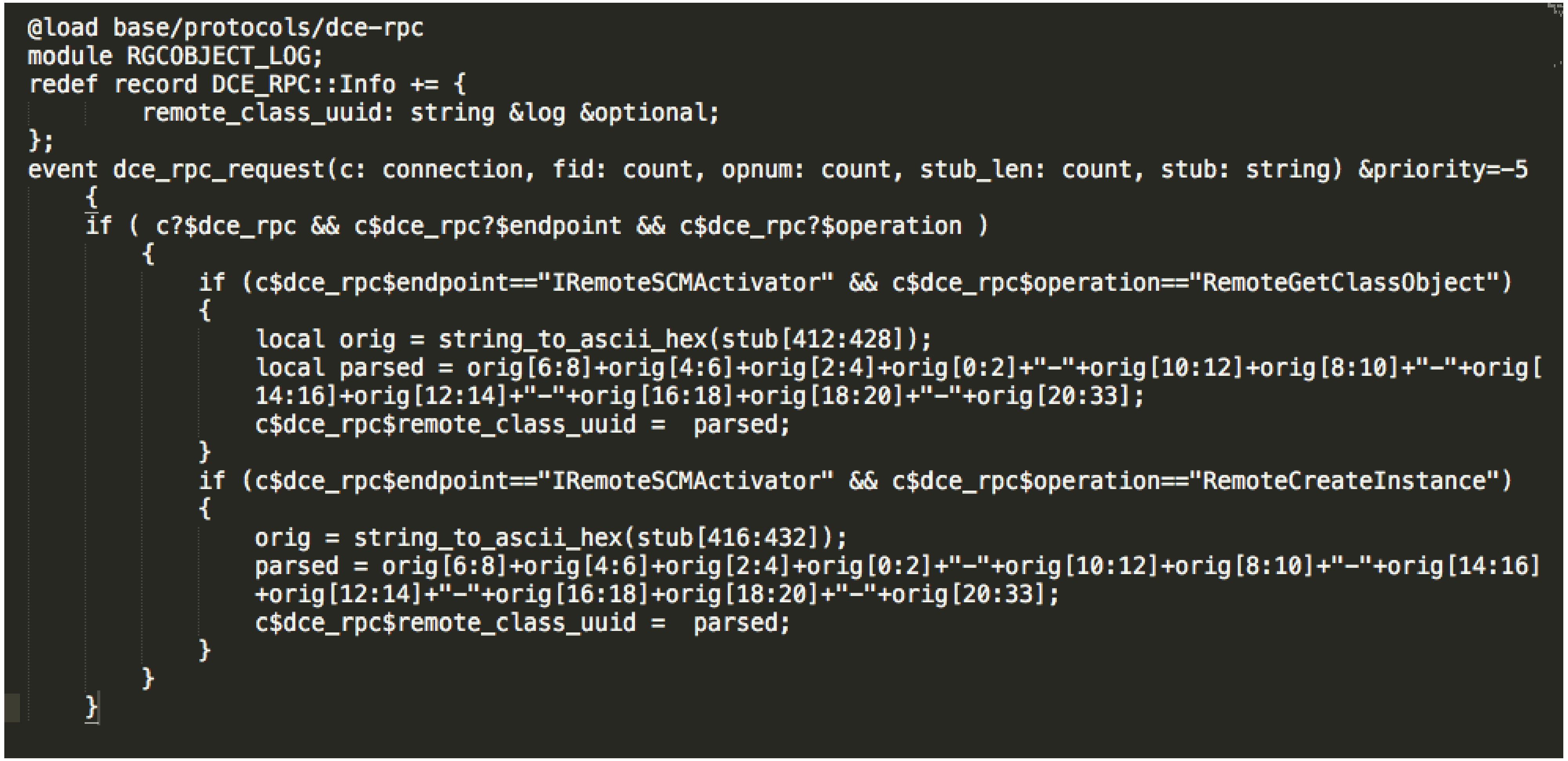
```
event dce_rpc_request%(c: connection, fid: count, opnum: count, stub_len: count, stub: string%);
```



```
function process_dce_rpc_request(req: DCE_RPC_Request): bool
	%{
	if ( dce_rpc_request )
		{
		BifEvent::generate_dce_rpc_request(bro_analyzer(),
		                                   bro_analyzer()->Conn(),
		                                   fid,
		                                   ${req.opnum},
		                                   ${req.stub}.length(),
		                                   bytestring_to_val(${req.stub}));
```

*Disclaimer: While I have tested this, it is a beta POC on top of already beta code. Undergoing further testing before submitting to Zeek.

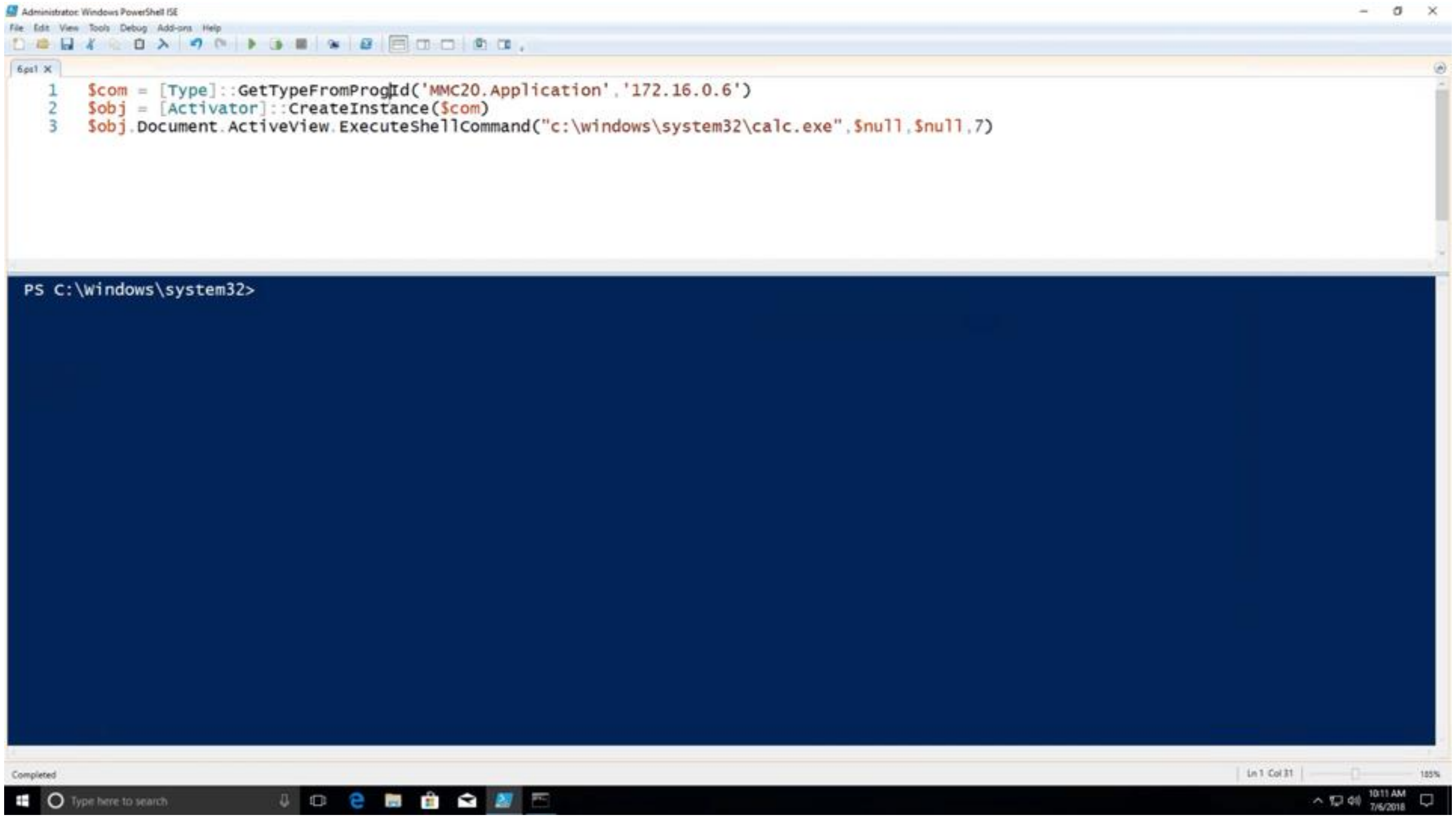
Metadata Extraction – Zeek Script Example



```
@load base/protocols/dce-rpc
module RGCOBJECT_LOG;
redef record DCE_RPC::Info += {
        remote_class_uuid: string &log &optional;
};
event dce_rpc_request(c: connection, fid: count, opnum: count, stub_len: count, stub: string) &priority=-5
    {
    if ( c?$dce_rpc && c$dce_rpc?$endpoint && c$dce_rpc?$operation )
        {
            if (c$dce_rpc$endpoint=="IRemoteSCMActivator" && c$dce_rpc$operation=="RemoteGetClassObject")
            {
                local orig = string_to_ascii_hex(stub[412:428]);
                local parsed = orig[6:8]+orig[4:6]+orig[2:4]+orig[0:2]+"-"+orig[10:12]+orig[8:10]+"-"+orig[
                14:16]+orig[12:14]+"-"+orig[16:18]+orig[18:20]+"-"+orig[20:33];
                c$dce_rpc$remote_class_uuid =  parsed;
            }
            if (c$dce_rpc$endpoint=="IRemoteSCMActivator" && c$dce_rpc$operation=="RemoteCreateInstance")
            {
                orig = string_to_ascii_hex(stub[416:432]);
                parsed = orig[6:8]+orig[4:6]+orig[2:4]+orig[0:2]+"-"+orig[10:12]+orig[8:10]+"-"+orig[14:16]
                +orig[12:14]+"-"+orig[16:18]+orig[18:20]+"-"+orig[20:33];
                c$dce_rpc$remote_class_uuid =  parsed;
            }
        }
    }
```

*Disclaimer: While I have tested this, it is a beta POC on top of already beta code. Undergoing further testing before submitting to Zeek.

Checking New Telemetry



```
$com = [Type]::GetTypeFromProgId('MMC20.Application','172.16.0.6')
$obj = [Activator]::CreateInstance($com)
$obj.Document.ActiveView.ExecuteShellCommand("c:\windows\system32\calc.exe",$null,$null,7)
```

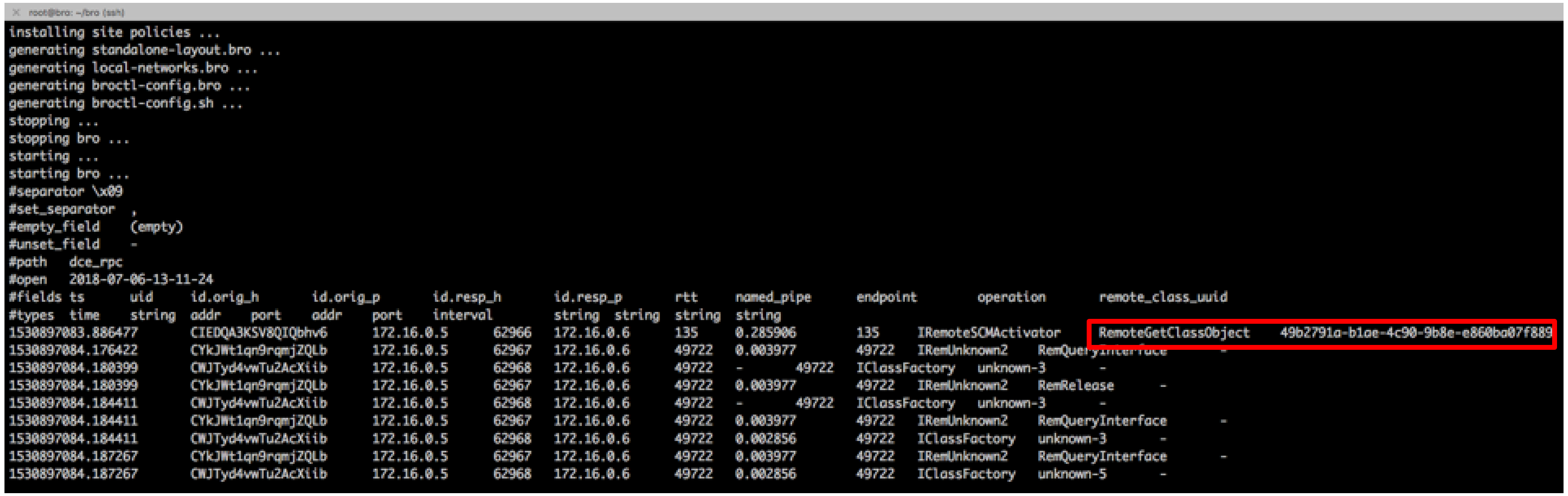
```
PS C:\Windows\system32>
```

Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
6.ps1
Completed
Ln 1 Col 31
185%
Type here to search
10:11 AM
7/6/2018

A Bit Better

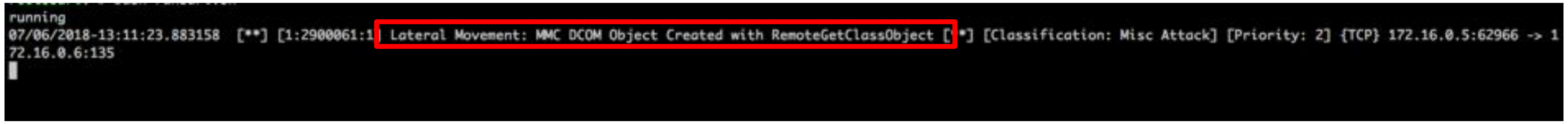
With these modifications, we have:

- Telemetry with CLSID of object getting mapped
- IDS indication of known suspicious / bad



root@bro: ~/bro (ssh)

```
installing site policies ...
generating standalone-layout.bro ...
generating local-networks.bro ...
generating broctl-config.bro ...
generating broctl-config.sh ...
stopping ...
stopping bro ...
starting ...
starting bro ...
#separator \x09
#set_separator  ,
#empty_field    (empty)
#unset_field    -
#path   dce_rpc
#open   2018-07-06-13-11-24
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h       id.resp_p       rtt     named_pipe      endpoint        operation       remote_class_uuid
#types  time    string  addr    port    addr    port    interval        string  string  string  string
1530897083.886477       CIEDQA3KSV8QIQbhv6      172.16.0.5      62966   172.16.0.6      135     0.285906        135     IRemoteSCMActivator     RemoteGetClassObject    49b2791a-b1ae-4c90-9b8e-e860ba07f889
1530897084.176422       CYkJWt1qn9rqmjZQLb      172.16.0.5      62967   172.16.0.6      49722   0.003977        49722   IRemUnknown2    RemQueryInterface       -
1530897084.180399       CWJTyd4vwTu2AcXiib      172.16.0.5      62968   172.16.0.6      49722   -       49722   IClassFactory   unknown-3       -
1530897084.180399       CYkJWt1qn9rqmjZQLb      172.16.0.5      62967   172.16.0.6      49722   0.003977        49722   IRemUnknown2    RemRelease      -
1530897084.184411       CWJTyd4vwTu2AcXiib      172.16.0.5      62968   172.16.0.6      49722   -       49722   IClassFactory   unknown-3       -
1530897084.184411       CYkJWt1qn9rqmjZQLb      172.16.0.5      62967   172.16.0.6      49722   0.003977        49722   IRemUnknown2    RemQueryInterface       -
1530897084.184411       CWJTyd4vwTu2AcXiib      172.16.0.5      62968   172.16.0.6      49722   0.002856        49722   IClassFactory   unknown-3       -
1530897084.187267       CYkJWt1qn9rqmjZQLb      172.16.0.5      62967   172.16.0.6      49722   0.003977        49722   IRemUnknown2    RemQueryInterface       -
1530897084.187267       CWJTyd4vwTu2AcXiib      172.16.0.5      62968   172.16.0.6      49722   0.002856        49722   IClassFactory   unknown-5       -
```



```
running
07/06/2018-13:11:23.883158  [**] [1:2900061:1] Lateral Movement: MMC DCOM Object Created with RemoteGetClassObject [**] [Classification: Misc Attack] [Priority: 2] {TCP} 172.16.0.5:62966 -> 1
72.16.0.6:135
```

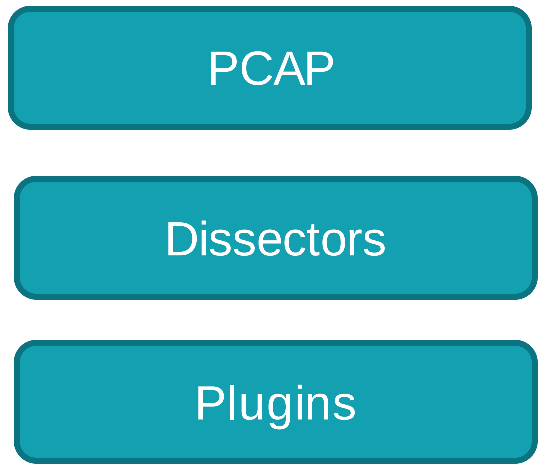
Empowering Detection

Detection & Analysis	
	Build detections around known bad DCOM objects
	Build behavioral indicators of when a DCOM object is instantiated followed by the IDispatch Invoke
	Implement anomaly detection on DCOM to show when objects might be getting used abnormally or when suspicious objects appear
	Model the DCOM ITypeInfo interactions to show “normal” programmatic use vs interactive human use

PCAP Analysis

Wireshark – Open source and widely used network protocol analyzer. Used extensively in forensics and response.

- **Dissectors** — These iteratively analyze and parse protocols, usually subsequently handing off to sub-dissectors.
- **Plugins** — External components that extend the functionality of Wireshark through various methods, to include protocol dissection. Written in Lua.

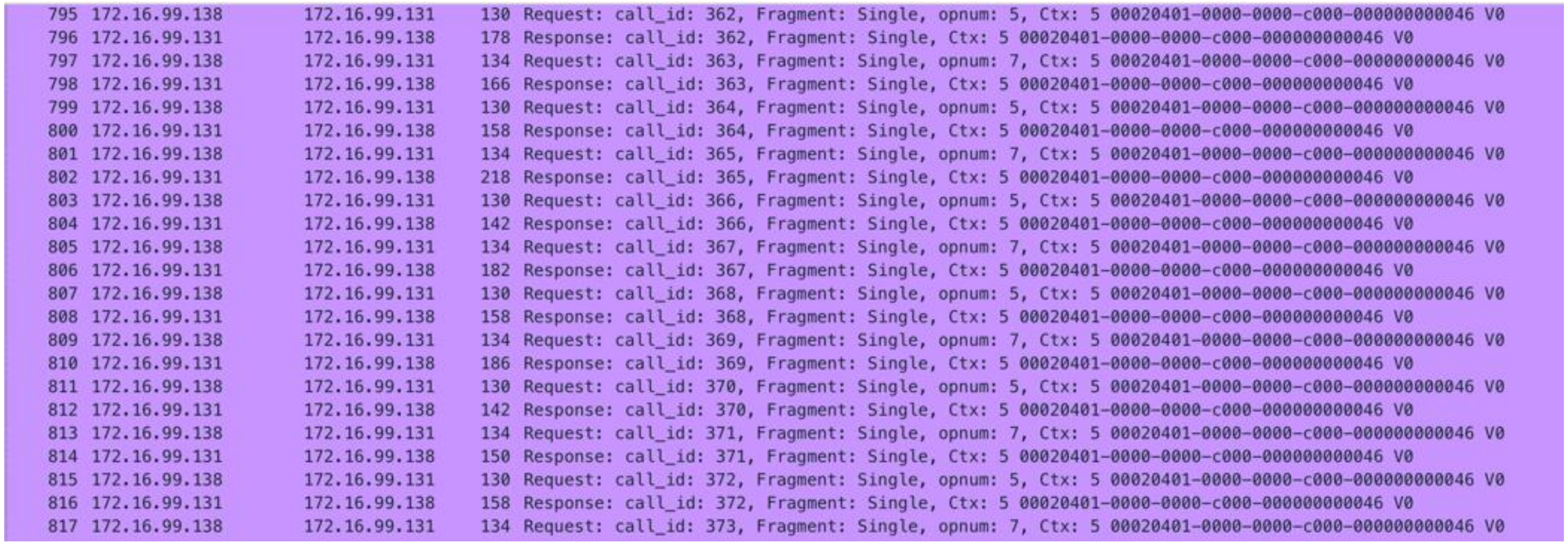


https://www.wireshark.org/docs/wsdg_html_chunked/ChDissectAdd.html

PCAP Analysis — What Is This In Wireshark?

Wireshark mostly handles the classic DCOM operations

- We realized that ITypeInfo interface was not implemented
- Several other endpoints / operations lacked dissection



```
795 172.16.99.138        172.16.99.131     130 Request: call_id: 362, Fragment: Single, opnum: 5, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
796 172.16.99.131        172.16.99.138     178 Response: call_id: 362, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
797 172.16.99.138        172.16.99.131     134 Request: call_id: 363, Fragment: Single, opnum: 7, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
798 172.16.99.131        172.16.99.138     166 Response: call_id: 363, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
799 172.16.99.138        172.16.99.131     130 Request: call_id: 364, Fragment: Single, opnum: 5, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
800 172.16.99.131        172.16.99.138     158 Response: call_id: 364, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
801 172.16.99.138        172.16.99.131     134 Request: call_id: 365, Fragment: Single, opnum: 7, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
802 172.16.99.131        172.16.99.138     218 Response: call_id: 365, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
803 172.16.99.138        172.16.99.131     130 Request: call_id: 366, Fragment: Single, opnum: 5, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
804 172.16.99.131        172.16.99.138     142 Response: call_id: 366, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
805 172.16.99.138        172.16.99.131     134 Request: call_id: 367, Fragment: Single, opnum: 7, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
806 172.16.99.131        172.16.99.138     182 Response: call_id: 367, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
807 172.16.99.138        172.16.99.131     130 Request: call_id: 368, Fragment: Single, opnum: 5, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
808 172.16.99.131        172.16.99.138     158 Response: call_id: 368, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
809 172.16.99.138        172.16.99.131     134 Request: call_id: 369, Fragment: Single, opnum: 7, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
810 172.16.99.131        172.16.99.138     186 Response: call_id: 369, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
811 172.16.99.138        172.16.99.131     130 Request: call_id: 370, Fragment: Single, opnum: 5, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
812 172.16.99.131        172.16.99.138     142 Response: call_id: 370, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
813 172.16.99.138        172.16.99.131     134 Request: call_id: 371, Fragment: Single, opnum: 7, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
814 172.16.99.131        172.16.99.138     150 Response: call_id: 371, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
815 172.16.99.138        172.16.99.131     130 Request: call_id: 372, Fragment: Single, opnum: 5, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
816 172.16.99.131        172.16.99.138     158 Response: call_id: 372, Fragment: Single, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
817 172.16.99.138        172.16.99.131     134 Request: call_id: 373, Fragment: Single, opnum: 7, Ctx: 5 00020401-0000-0000-c000-000000000046 V0
```

PCAP Analysis — Contributing To Wireshark

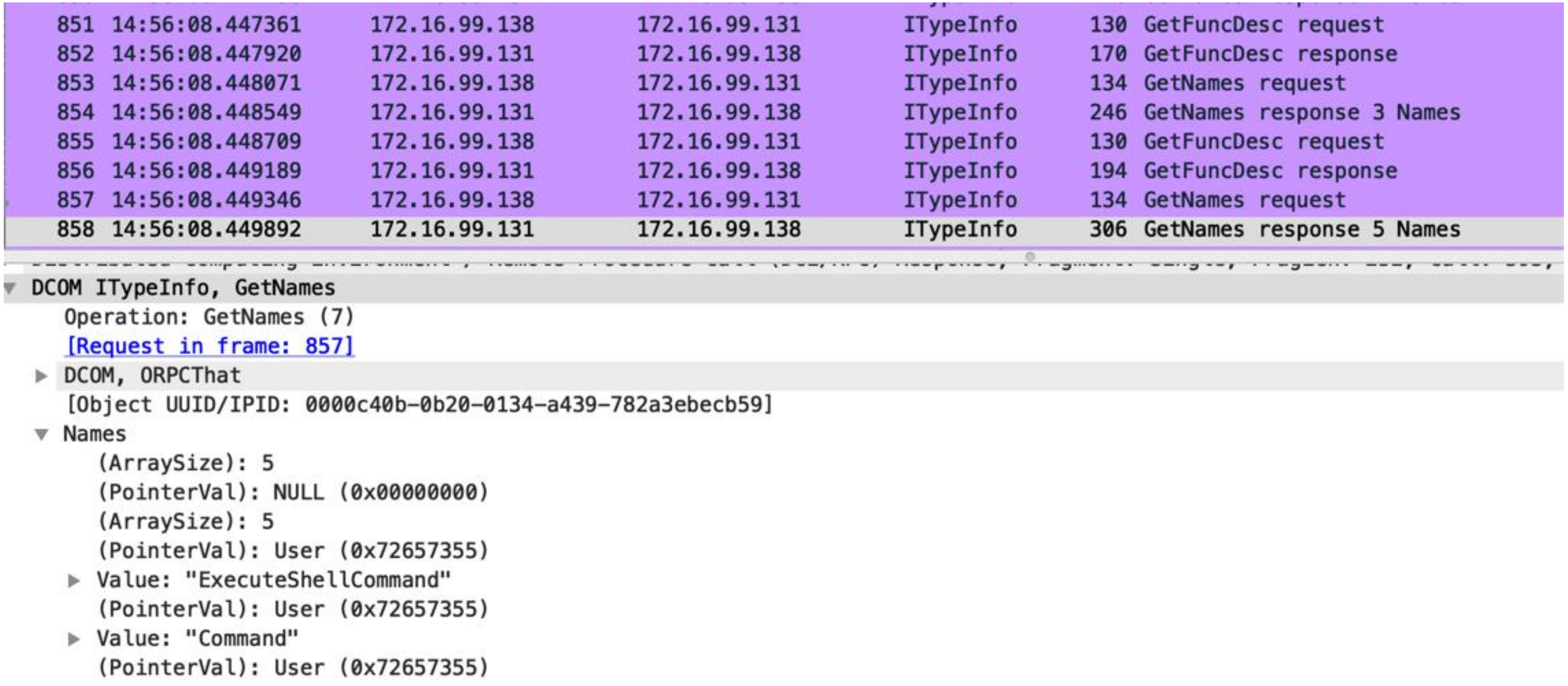
We built upon existing Wireshark capabilities

- Added references to our dissector in packet-dcom.c
- Built out packet-dcom-typeinfo.c
- Added to packet-dcom-sysact.c

Tips for protocol RE and contributions:

- Leverage existing dissectors and plugins for understanding
- Use official documentation on the protocols (RFCs, MSDN, etc)
- Utilize multiple variations of attack (multiple PCAPs)
- Identify high confidence “markers” and fill in around
- No guarantee other people did it right
- For DCOM: https://wiki.wireshark.org/Pidl

PCAP Analysis — Contributing To Wireshark


```
851 14:56:08.447361   172.16.99.138   172.16.99.131   ITypeInfo   130 GetFuncDesc request
852 14:56:08.447920   172.16.99.131   172.16.99.138   ITypeInfo   170 GetFuncDesc response
853 14:56:08.448071   172.16.99.138   172.16.99.131   ITypeInfo   134 GetNames request
854 14:56:08.448549   172.16.99.131   172.16.99.138   ITypeInfo   246 GetNames response 3 Names
855 14:56:08.448709   172.16.99.138   172.16.99.131   ITypeInfo   130 GetFuncDesc request
856 14:56:08.449189   172.16.99.131   172.16.99.138   ITypeInfo   194 GetFuncDesc response
857 14:56:08.449346   172.16.99.138   172.16.99.131   ITypeInfo   134 GetNames request
858 14:56:08.449892   172.16.99.131   172.16.99.138   ITypeInfo   306 GetNames response 5 Names
```

```
▼ DCOM ITypeInfo, GetNames
    Operation: GetNames (7)
    [Request in frame: 857]
  ▶ DCOM, ORPCThat
    [Object UUID/IPID: 0000c40b-0b20-0134-a439-782a3ebecb59]
  ▼ Names
      (ArraySize): 5
      (PointerVal): NULL (0x00000000)
      (ArraySize): 5
      (PointerVal): User (0x72657355)
    ▶ Value: "ExecuteShellCommand"
      (PointerVal): User (0x72657355)
    ▶ Value: "Command"
      (PointerVal): User (0x72657355)
```

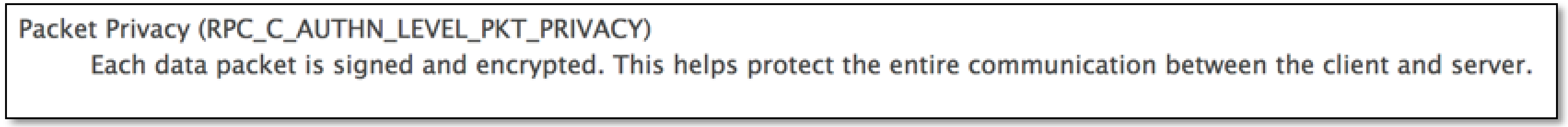
Endpoint Detection Tease

Multiple Strategies Needed

The best solution is one that incorporates resilient telemetry of multiple forms to strengthen confidence on a detection!

What if they specifically attempted to evade on the network?

- Multiple BINDs or alter_context calls to confuse context
- PKT_PRIVACY

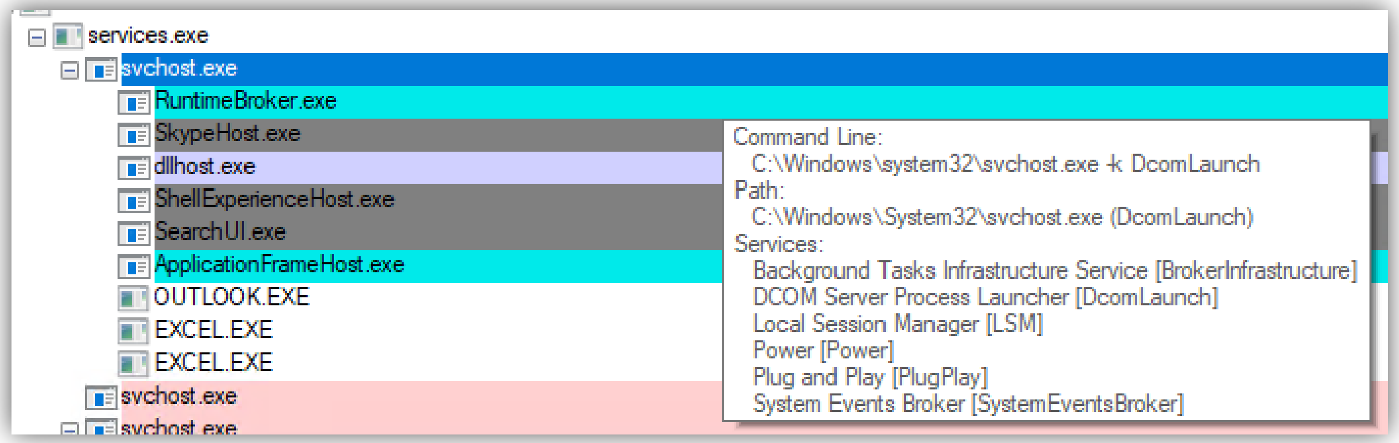

Packet Privacy (RPC_C_AUTHN_LEVEL_PKT_PRIVACY)
Each data packet is signed and encrypted. This helps protect the entire communication between the client and server.

https://msdn.microsoft.com/en-us/library/windows/desktop/ms678509(v=vs.85).aspx

Don’t forget… PKT_PRIVACY could be an indicator on its own!

Some Things To Explore

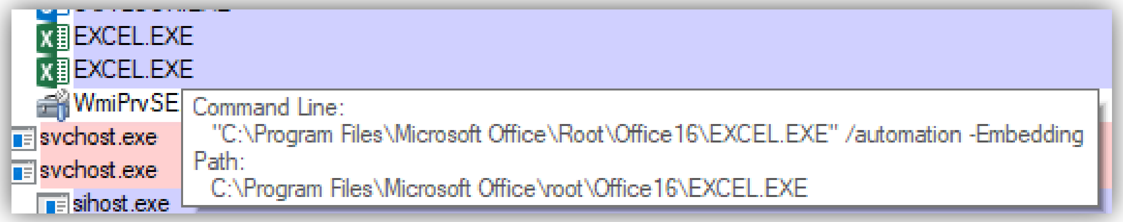
- DCOM processes spawn underneath the DCOMLaunch service (svchost.exe –k DcomLaunch)



```
services.exe
  svchost.exe
    RuntimeBroker.exe
    SkypeHost.exe
    dllhost.exe
    ShellExperienceHost.exe
    SearchUI.exe
    ApplicationFrameHost.exe
    OUTLOOK.EXE
    EXCEL.EXE
    EXCEL.EXE
  svchost.exe
  svchost.exe
```

```
Command Line:
  C:\Windows\system32\svchost.exe -k DcomLaunch
Path:
  C:\Windows\System32\svchost.exe (DcomLaunch)
Services:
  Background Tasks Infrastructure Service [BrokerInfrastructure]
  DCOM Server Process Launcher [DcomLaunch]
  Local Session Manager [LSM]
  Power [Power]
  Plug and Play [PlugPlay]
  System Events Broker [SystemEventsBroker]
```

- Many objects don't expose a "quit" or "exit" method leaving lagging processes. They make good indicators.
- Office applications started via COM have the "-Embedding" or "/automation" options



```
EXCEL.EXE
EXCEL.EXE
WmiPrvSE
svchost.exe
svchost.exe
sihost.exe
```

```
Command Line:
  "C:\Program Files\Microsoft Office\Root\Office16\EXCEL.EXE" /automation -Embedding
Path:
  C:\Program Files\Microsoft Office\root\Office16\EXCEL.EXE
```

Wrapping Up

Wrapping Up

This process is repeatable for DCOM lateral movement techniques as they come out:

- Emulate the technique and collect telemetry
- Identify the activation and execution components on network and host
- Conduct analysis on the components to extract artifacts and behaviors
- Leverage identified artifacts or behaviors to build detection criteria
- Use criteria to build a detection across systems

Non-DCOM Specific Takeaways:

- Detecting threats is our mission — but often the knowledge and capabilities required to do so is lacking
- Using a structured process to mock up threat behavior, study the behavior, evaluate indicators, and author detections helps identify gaps
- Sometimes we must increase visibility and fill these gaps

Questions?