Rad Omens: The Good and Evil Uses of WMI, CIM

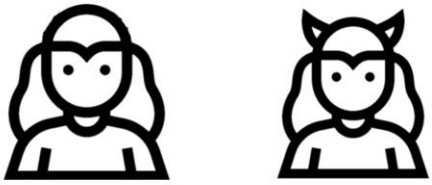
Katherine Scrupa
Network Technology CCNA, Hons; GIAC-GCWN
MUUG.ca board member
The Long Con - November 2, 2019

1

Apologies to Neil Gaiman and Terry Pratchett
Good Omens

Goal:

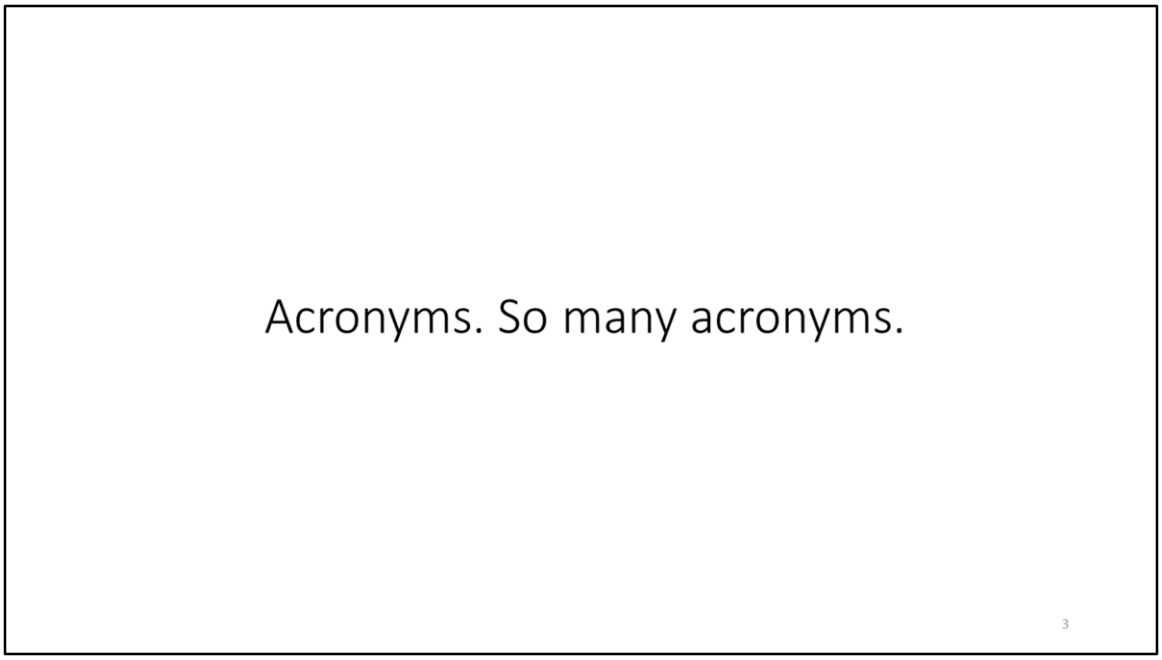
Get you familiar with
the structure and capabilities
of WMI and CIM
so you can experiment
and bend it to your own purposes



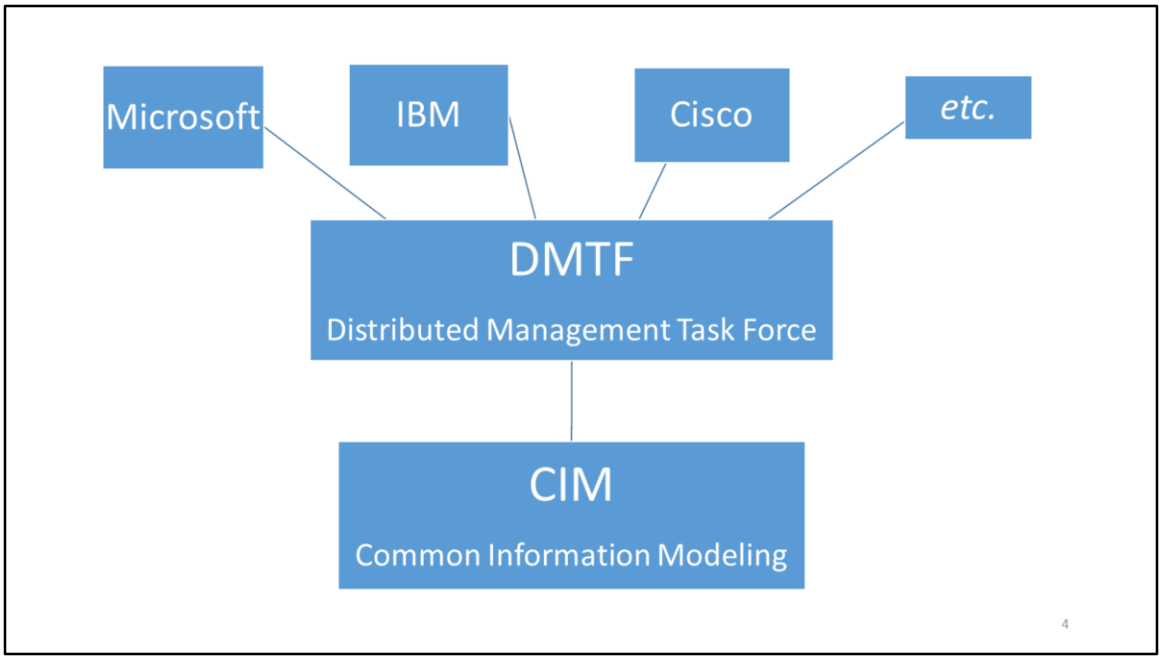
2

Matt Graeber has a good talk if you are more curious about evil uses
https://www.youtube.com/watch?v=0SjMgnGwpq8
Abusing Windows Management Instrumentation (WMI)
@mattifestation

Lots of good infosec people on twitter – not quite as much as a dumpster fire as reddit or hacker news



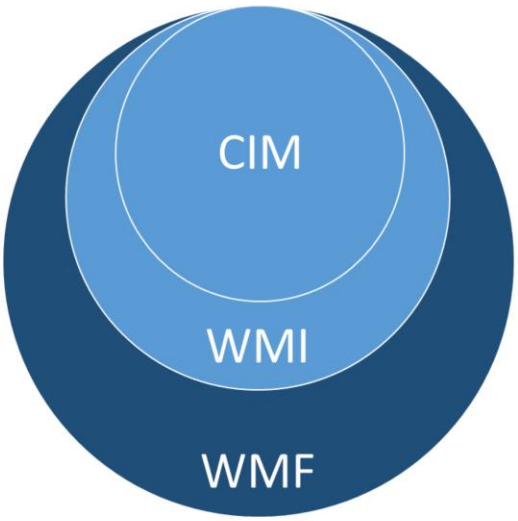
You may have seen many of these, but are unsure was to what exactly they are, or how they're related



Orgs: “We need to define standard of how to access computer info”

Is likely to become more important as vendors move to cloud stuff and are playing a little nicer with each other (Microsoft, Github, Linux on Windows, Powershell on Linux, etc.)

- Common Information Modeling (CIM)
  - **Core** and **Common** data models as specified by the DMTF
  - OS-agnostic standard
- Windows Management Instrumentation (WMI)
  - **Extended**, Microsoft-specific data models
- Windows Management Framework (WMF)
  - Provides multiple services to manage Windows, including CIM & WMI queries



5

Wmi's been around since NT 4.0 days as a separate download. Mid 90's.

And WBEM, WS-MAN, WinRM?

- Web-based Enterprise Management (WBEM)
  - Previous slides show what is accessible. WBEM describes how to access it.
  - WBEM is based on Internet & DMTF open standards
  - Covers Mappings, Protocols, Discovery, and Query Languages

- Web Services Management (WS-MAN) is a Protocol used for WBEM.
  - WinRM is the Microsoft Implementation of WS-MAN

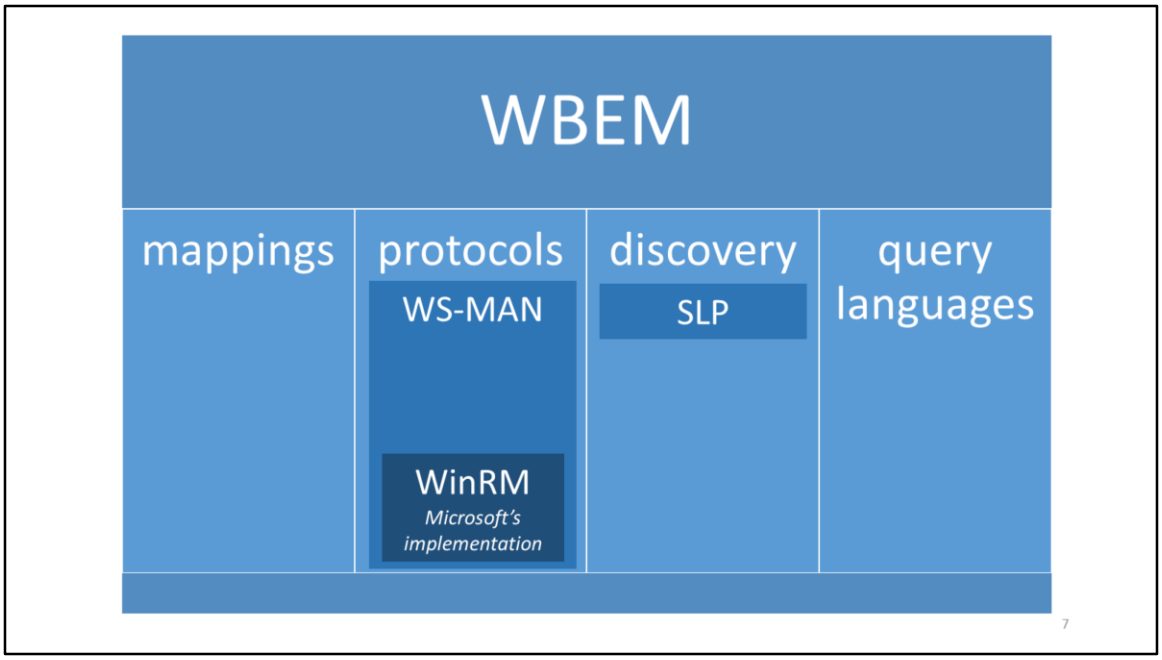
WBEM & WS-MAN are OS-agnostic.

6

People think since it begins with W, it's a Windows thing. It's not, necessarily.

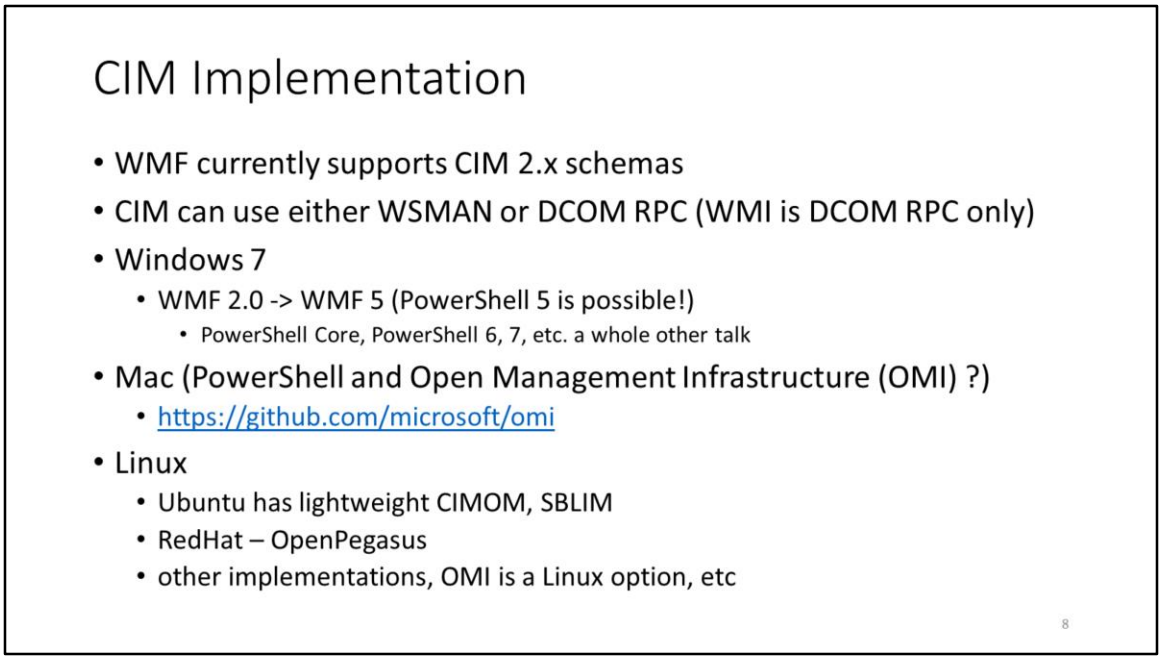
Wikipedia page a good resource for investigating particular OSes' use of WBEM and WS-MAN
https://en.wikipedia.org/wiki/Web-Based_Enterprise_Management
https://en.wikipedia.org/wiki/WS-Management

WBEM is REST-based, WS-MAN is SOAP-based.
https://docs.microsoft.com/en-us/windows/win32/winrm/portal



simplified version of last slide (some stuff missing)

Query langes (very SQL-like)
WMI Query Language
CIM Query Language
Filter Query Language



You'll see a lot of people say PowerShell 5 and its goodies are not available on Windows 7. That's not true. WMF 5.1 is a whole 63 MB download.
Get rid of Windows 7 if you can, but if not, update WMF to 5.1
Server 2008 and 2012 with WMF 5.1 are possible too! (but 2008 is also approaching EOL)
https://www.microsoft.com/en-us/download/details.aspx?id=54616

Linux still prefers files for everything (files… files EVERYWHERE). Adoption is slower. Didn't have time to play with Linux stuff, but other people have. Go nuts.

https://www.opslogix.com/part-3-apple-osx-mp-configuring-mac-omi/

Query Methods and Discovery – WMI (1)

- PowerShell directly (local or remote targeting)
  - PowerShell via PS-Remoting
- .NET System.Management namespace
- cmd (wmic.exe) 
  - old-skool
  - just don't
  - stop it

9

Wmic can be unwieldy - such as using it to uninstall stuff and trying to tell it to not reboot (not possible in some cases)

Have your IT department use PS-Remoting with certs for WMI and other PowerShell queries. Then a SIEM can let you you know people using wmic is likely sketchy traffic.

For those who don't know, PowerShell , PowerShell Core, .NET are all tightly integrated. Are also very object-oriented. A whole other talk.

If you come from the Linux world, it may break your brain a bit. Just remember pretty much every result you get back is likely an object, even if it just looks like a string (an object may hold data that is of type string, but there are built-in methods on it such as .length() and maniuplations). Don't think of the text result on your screen as "just text".

Query Methods and Discovery – WMI (2)

- C:\windows\system32\wbemtest.exe
  - Very clunky but ok for testing WMI queries
- MMC – either local or remote pc
  - kinda sucks for browsing
- "JFGI" - Google / StackExchange / GitHub, etc.
- Hey Scripting Guy
  - https://blogs.technet.com/heyscriptingguy/author/the-scripting-guys
- www.powershell.org

10

If you've thought of a problem you need to solve, so has someone else. Probably something close to your needs already on StackExchange, Github, etc.

Query Methods and Discovery – WMI (3)

Third-party Tools:

- WMI Explorer
  - https://www.github.com/vinaypamnani/wmie2/releases
  - Free, small, stand-alone exe
  - Browseable interface like Windows Explorer
- CIM Studio
  - Microsoft MSDN download
  - part of the Windows Management Instrumentation (WMI) SDK 1.5.
  - disappeared from MSDN?

11

I make good use of WMI Explorer, makes discovery a breeze. Interesting to browse through.

CIM Studio looks similar to WMI Explorer. Can find references on TechNet, but download links seem to have disappeared

Query Methods and Discovery – WMI (4)

- Other Goodies on Github
  - Open Asset Logger
  - Impacket
  - WMI-Shell

Check out https://www.fireeye.com/content/dam/fireeye-www/services/pdfs/sans-dfir-2015.pdf

12

Others too

CIM & WMI Structure

Kinda like a database. Kinda like an instance of OOP stuff.

It's a collection of essentially anything and everything installed, attached to, or associated with your computer.

- CIM Providers
  - CIM Classes
    - Instances of the class
    - Properties (data)
    - Methods
    - extensible (WMI classes)
- WQL, CQL, FQL

13

Twisted love child of an instance of an OOP construct and a database

Don't be pendantic (thought that makes us good at our jobs)

All very SQL-like queries
WMI Query Language
CIM Query Language
Filter Query Language

Example - List WMI Objects

```
Get-WmiObject –list
NameSpace: ROOT\cimv2
Name                                Methods       Properties
CIM_Indication                      {}            {CorrelatedIndications,
IndicationFilterName...}
CIM_ClassIndication                 {}            {ClassDefinition, CorrelatedInd...}
CIM_ClassDeletion                   {}            {ClassDefinition, CorrelatedInd...}
CIM_ClassCreation                   {}            {ClassDefinition, CorrelatedInd...}
CIM_ClassModification               {}            {ClassDefinition, CorrelatedInd...}
Win32_Trustee                       {}            {Domain, Name, SID, SidLength...}
Win32_ACE                           {}            {AccessMask, AceFlags, AceType...}
Win32_SecurityDescriptor            {}            {ControlFlags, DACL, Group, Owner...}
Win32_NetworkAdapterConfiguration   {EnableDHCP, Renew...
                                                  {ArpAlwaysSourceRoute, ArpUseEther...}
...
```

14

Root\cimv2 is default namespace
Can specify other namespaces with a switch (command option)
Though you're going to be living in root\cimv2 pretty much all the time

Note there are methods for some classes

Example – List Instance of computersystem

```
$c = Get-WMIObject -class Win32_ComputerSystem
$c | format-list *
CreationClassName           : Win32_ComputerSystem
CurrentTimeZone             : -300
DaylightInEffect            : True
DNSHostName                 : HOSER
EnableDaylightSavingsTime   : True
HypervisorPresent           : False
Manufacturer                : LENOVO
Model                       : 20N40026US
PrimaryOwnerName            : kat
Roles                       : {LM_Workstation, LM_Server, NT}
SystemFamily                : ThinkPad T590
$c.DNSHostName
```

15

"| format-list *" Because powershell tends to show you a few values it expects you might want, doesn't show all values of the object
Use * when you need to, but in a script, minimize the data you grab to optimize performance (especially if you're piping several things in one statement)

Example – Determine CIM Equivalent of WMI

```
$c.__DERIVATION
__CLASS                     : Win32_ComputerSystem
__SUPERCLASS                : CIM_UnitaryComputerSystem
__DYNASTY                   : CIM_ManagedSystemElement
__RELPATH                   : Win32_ComputerSystem.Name="HOSER"
__PROPERTY_COUNT            : 64
__DERIVATION                : {CIM_UnitaryComputerSystem, CIM_ComputerSystem,
CIM_System, CIM_LogicalElement...}
__SERVER                    : HOSER
__NAMESPACE                 : root\cimv2
__PATH                      :
\\HOSER\root\cimv2:Win32_ComputerSystem.Name="HOSER"
...
CreationClassName           : Win32_ComputerSystem
CurrentTimeZone             : -300
DaylightInEffect            : True
```

16

You can do
$c | select-object –expandproperty __derivation
What you're looking for is one of those in that list

Example – CIM with WQL Query

```
Get-CimInstance -Query "SELECT * from Win32_Process WHERE name LIKE 'p%'"
Get-CimInstance -ClassName Win32_Process -Filter "Name like 'p%'"
```

ProcessId	Name	HandleCount	WorkingSetSize	VirtualSize
3832	PresentationFontCache.exe	232	16805888	4873068544
10476	PowerMgr.exe	394	5697536	118558720
15740	powershell_ise.exe	1093	258007040	5669785600
10096	PaintStudio.View.exe	606	430080	2203847753728
15624	putty.exe	292	21245952	4501696512
3864	POWERPNT.EXE	1838	138481664	773525504

17

Directly run the SQL-like queries with either of these commands

https://docs.microsoft.com/en-us/powershell/module/cimcmdlets/get-ciminstance?view=powershell-6

MOF files?

- Managed Object Format (MOF) files
  - The structure of CIM classes
    - describes data and events
- MOFs are also used with Desired State Configuration (DSC), which is a whole other topic…
  - If you manage Windows, start learning about DSC!
  - PowerShell too, if you haven't already.

18

You may have seen MOF files floating around and wondered what the heck they are

DSC = Automation, like Chef, Ansible, Puppet

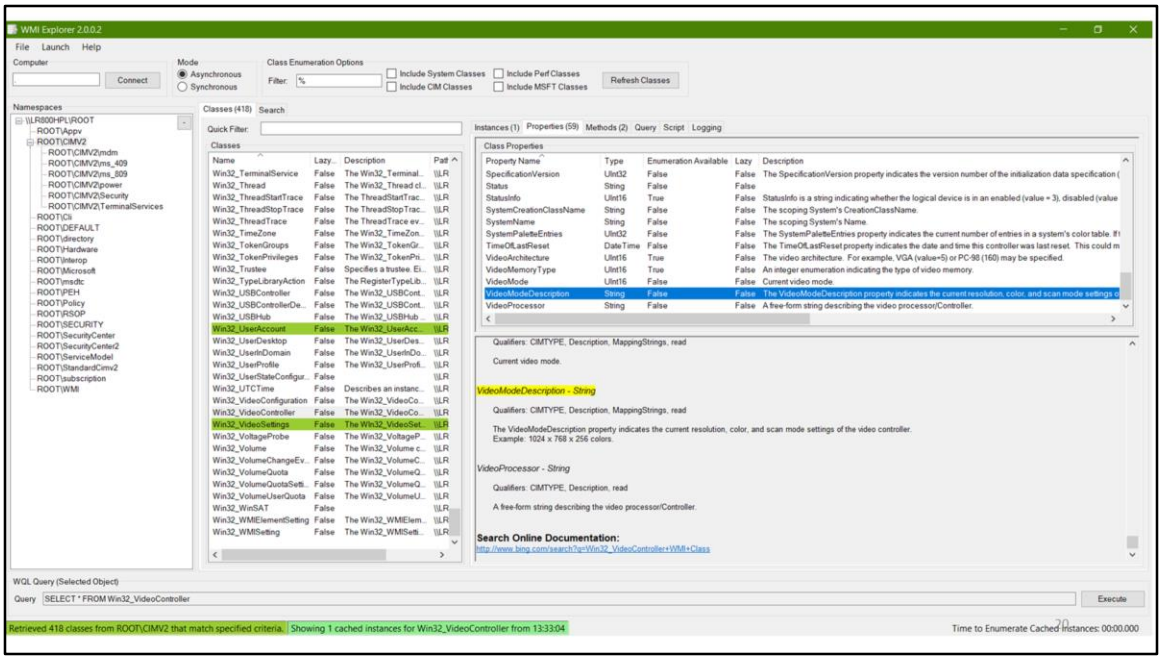
Story Time – WMI Good

Deploy wallpaper to your org. What resolution(s) do you target?

In your entire org, who has multiple screens, and what resolutions are they using?

19

Mishmash of equipment over time
Inventory may not give the exact info you need



Columns: Namespaces, classes, instances, properties and their values

Red – unable to enumerate or empty
Green – have recently enumerated (double-clicked) – in memory

Slow?
WMI uses a COM Client-Server architecture (DCOM is slow)
and uses Marshaling for communications (mainly IPSFactoryBuffer interface)
https://social.msdn.microsoft.com/Forums/aspnet/en-US/6c13d669-389c-47e5-8d61-dce3c8ac30f7/why-is-wmi-so-slow?forum=vcgeneral
Not my forte, but apparently that's why.

Also, minimize the calls you make and filter as soon as possible for only the data you need

Someone suggested to not use aliases in scripts. Theoretically may help, but perhaps not much. There are other reasons to not include aliases in scripts (might get hijacked in a different environment, etc)

Note query at the bottom - **browsing gives you the exact WQL code that you can copy in to a WQL query**

Quite fun to browser this

Get-ScreenResolution.ps1 //code snippets

```
$users = Get-WmiObject
         -Class win32_computersystem
         -Property username
         -ComputerName $_
$sizes = Get-WmiObject
         -Class win32_videocontroller
         -ComputerName $_
         | Select-Object VideoModeDescription
```

21

These are two separate commands used in my script, with stuff in between

Note that PowerShell and PowerShell ISE has tab completion for commands

```
for($i=0; $i -lt $screens; $i++){
  $resolution = $sizes[$i].VideoModeDescription
    if ($resolution -ne $null) {
      $values = $resolution.Split(" ")
      $x = $values[0]
      $y = $values[2]
      Out-File -FilePath $outfile -Append
               -InputObject “up,$_,$user,$x,$y"
      Write-host "up,$_,$user,$x,$y“
```

22

Used to get the x, y values of all screens attached
Look for win32_videocontroller class and videomodedescription property in WMI explorer to get a full description of what the values are

Results

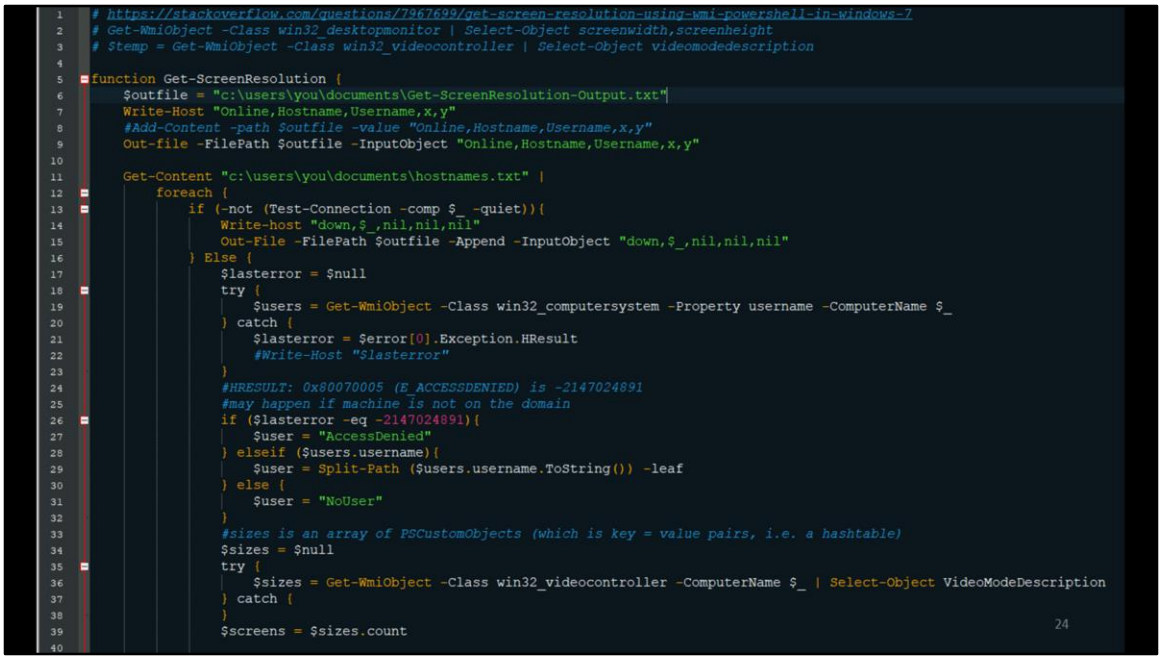
```
up,comp5321,asmith,1920,1080
up,comp5321,asmith,1920,1080
up,comp3333,jblack,1600,900
up,comp3333,jblack,1920,1080
down,comp4413,nil,nil,nil
up,comp9032,cjackson,1920,1080
up,comp0153,rmosby,1920,1080
```

23

Opened the results in excel and could use formulas to sort and count frequencies

Script can be used for identifying users with special needs, people who should know about CTRL + or CTRL – in web browsers, those due for an upgrade, or hoarders doing naughty things like poaching equipment (too many monitors, laptops/desktops, etc)

You could easily write your own custom inventory system using WMI & CIM (*for relative values of easy)



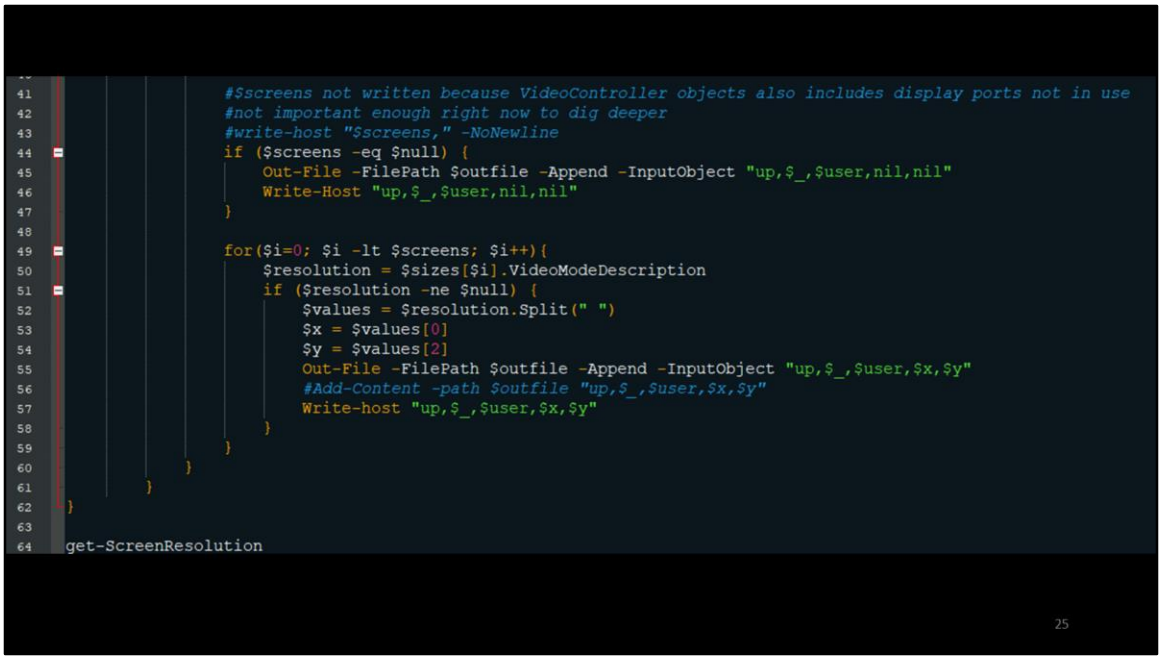
```powershell
# https://stackoverflow.com/questions/7967699/get-screen-resolution-using-wmi-powershell-in-windows-7
# Get-WmiObject -Class win32_desktopmonitor | Select-Object screenwidth,screenheight
# $temp = Get-WmiObject -Class win32_videocontroller | Select-Object videomodedescription

function Get-ScreenResolution {
    $outfile = "c:\users\you\documents\Get-ScreenResolution-Output.txt"
    Write-Host "Online,Hostname,Username,x,y"
    #Add-Content -path $outfile -value "Online,Hostname,Username,x,y"
    Out-file -FilePath $outfile -InputObject "Online,Hostname,Username,x,y"

    Get-Content "c:\users\you\documents\hostnames.txt" |
        foreach {
            if (-not (Test-Connection -comp $_ -quiet)){
                Write-host "down,$_,nil,nil,nil"
                Out-File -FilePath $outfile -Append -InputObject "down,$_,nil,nil,nil"
            } Else {
                $lasterror = $null
                try {
                    $users = Get-WmiObject -Class win32_computersystem -Property username -ComputerName $_
                } catch {
                    $lasterror = $error[0].Exception.HResult
                    #Write-Host "$lasterror"
                }
                #HRESULT: 0x80070005 (E_ACCESSDENIED) is -2147024891
                #may happen if machine is not on the domain
                if ($lasterror -eq -2147024891){
                    $user = "AccessDenied"
                } elseif ($users.username){
                    $user = Split-Path ($users.username.ToString()) -leaf
                } else {
                    $user = "NoUser"
                }
                #sizes is an array of PSCustomObjects (which is key = value pairs, i.e. a hashtable)
                $sizes = $null
                try {
                    $sizes = Get-WmiObject -Class win32_videocontroller -ComputerName $_ | Select-Object VideoModeDescription
                } catch {
                }
                $screens = $sizes.count
```

Full script ½

(Screenshot from Notepad ++ with Plastic Code Wrap styling)



```
                #$screens not written because VideoController objects also includes display ports not in use
                #not important enough right now to dig deeper
                #write-host "$screens," -NoNewline
                if ($screens -eq $null) {
                    Out-File -FilePath $outfile -Append -InputObject "up,$_,$user,nil,nil"
                    Write-Host "up,$_,$user,nil,nil"
                }

                for($i=0; $i -lt $screens; $i++){
                    $resolution = $sizes[$i].VideoModeDescription
                    if ($resolution -ne $null) {
                        $values = $resolution.Split(" ")
                        $x = $values[0]
                        $y = $values[2]
                        Out-File -FilePath $outfile -Append -InputObject "up,$_,$user,$x,$y"
                        #Add-Content -path $outfile "up,$_,$user,$x,$y"
                        Write-host "up,$_,$user,$x,$y"
                    }
                }
            }
        }
}

get-ScreenResolution
```


Full script 2/2

The method of creating this csv probably should have been done with Export-CSV so the document gets tagged as a Microsoft CSV file, but whatever, this also works
I was just learning ☺

Story: A Case of Mistaken, Assumed Identity

- Common GPO WQL query for determining laptop vs desktop:

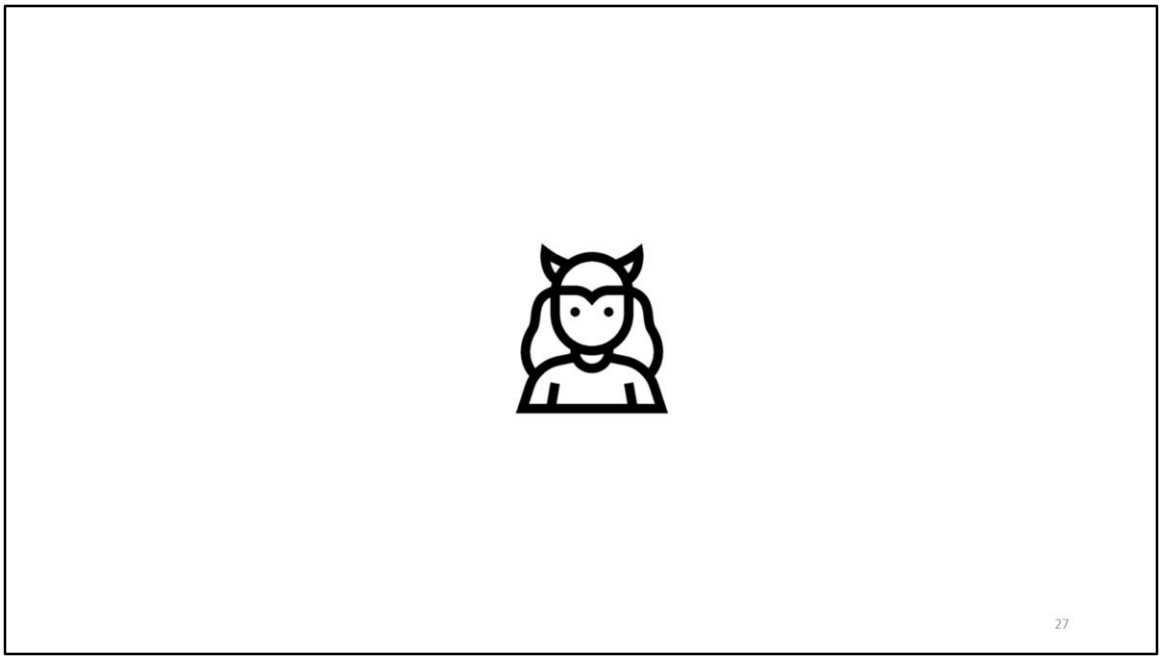
```
Select * from Win32_PhysicalMemory
    where (formfactor != 12)
    ###should give desktops
```

- Except some computers (Dell - both laptops and desktops) report this as 8.
- This is why you need to investigate queries on your fleet, and test in your own environment.
  - Maybe there is a better query? How else do you tell a laptop from desktop?

26

Food for thought…

Is there a query for chassis?
Even if you aren't using Dell, some mini pc's may use laptop-style RAM.
Laptops often have a removable battery… maybe a look for a battery class?



Uh oh.
You need to think about what what I said earlier about what WMI is

WMI is a collection of essentially
anything and everything
Installed, attached, or associated with
your computer.

28

Remember there is data, but methods as well

Often including .delete or create

Hackers *love* WMI.

29

Recon (1)

```
Get-WmiObject win32_service | ft name,state


$p=Get-WmiObject win32_process
     | where {$_.name -imatch "iexplore.exe"}
$p[0].GetOwner().User
```

30

WMI literally built for recon

Get services
Get processes and their owners…

Recon (2)

```
Get-WmiObject win32_StartupCommand

Get-WmiObject –class win32_share
     –ComputerName <hostname>

Get-WmiObject -Class win32_computersystem
     -Property username –ComputerName <hostname>
```

31

Leverage one system to access another

https://docs.microsoft.com/en-us/windows/win32/cimwin32prov/create-method-in-class-win32-process

Execution (1)

```
$Session = New-CimSession -ComputerName $Comp
$CimParams = @{
    ClassName  = 'Win32_Process'
    MethodName = 'Create'
    Arguments  = @{CommandLine = 'powershell.exe
-command "Start-Process notepad.exe"'}
    CimSession = $Session
}
Invoke-CimMethod @CimParams
```

32

Can do similar things as above with Invoke-WmiMethod
-computername available for Invoke-CIMmethod – so you can run this remotely too.
Lateral movement.
Methods to invoke, create, delete…

Execution (2)

```
$process = ([WMICLASS]"\\SERVER\ROOT\CIMV2:win32_process")
        .Create("cmd.exe /c c:\commands.bat")
```

33

Other ways to invoke processes (though I am not familiar with this method)

“Tweak” A CPU Fan 

```
//if motherboard has WMI interface
[Dynamic, Provider("CIMWin32"), UUID("{464FFAB5-
946F-11d2-AAE2-006008C78BC7}"), AMENDMENT]
class Win32_Fan : CIM_Fan
{
boolean  ActiveCooling;
uint64   DesiredSpeed;
uint16   PowerManagementCapabilities[];
uint16   StatusInfo;
…
```

34

The spec for some classes (the MOF) can show you really interesting things. This is why I like browsing with WMI Explorer
Setting a fan speed to zero would be unfortunate…

Lots more examples online

Halp!

35

Hints on how to protect yourself (not exhaustive)
We need to do something about this

Halp (1)

- Use PowerShell remoting
  - `Enter-PsSession <hostname>`
- Take care with username & authentication
  - use local admin account (with individual / LAPS password)
  - use `-credential $creds` with `New-CimSession`
  - Pops-up windows dialog to get your credentials
    - `secure…ish (can be hacked with .NET Runtime calls)`
- destroy $creds as soon as possible
  - `$creds = $null`
  - `Remove-Variable $creds`

36

Avoid using global / domain accounts for queries

Halp (2)

- Authenticate with Kerberos
  - `New-CimSession … -Authentication Kerberos`
- Use AES Encryption
  - Right-click the user account, Account tab,
    - "*This account supports Kerberos AES zzz Bit Encryption*"
  - … if you can (doesn't work with older Windows versions)

37

NTLM and NTMLv1 suck
DES sucks

Halp (3)

- Use SSL when connecting with WSMAN
  - `$option = New-CimSessionOption –UseSSL`
  - `New-CimSession … -SessionOption $option`
- When using DCOM RPC instead (i.e. WMI, not CIM queries), RPC is encrypted by default with `-PacketPrivacy`, but it can be explicitly required.

38

Explicitly use –PacketPrivacy if you're ultra-paranoid

Halp (4)

- Set additional Security Settings for the provider
  - Enable auditing for WMI and monitor with a SIEM
  - See guides at https://docs.microsoft.com/en-us/windows/win32/wmisdk/maintaining-wmi-security
- Verify correct users and groups have access
  - Use the MMC console for WMI

39

WMI Auditing is not enabled by default, requires a little setup
Exercise left for the reader…

Audit WMI access regularly

"An angel who did not so much fall,
as saunter vaguely downwards"

Neil Gaiman & Terry Pratchett
Good Omens: The Nice and Accurate Prophecies of Agnes Nutter, Witch

40

Yeah. This quote kinda describes WMI.

Please use and maintain WMI for good, not evil.

<demo of WMI Explorer>