Empire for Pentester: Active Directory Enumeration

hackingarticles.in/empire-for-pentester-active-directory-enumeration

Raj April 28, 2021

In this article, we take a look inside Active Directory through PowerShell Empire. PowerShell Empire consists of some post-exploitation modules inside the situational awareness section. PowerView is integrated inside the Empire to extract data from a Domain.

Table of Contents

Introduction

In our previous article focused on **Active Directory Enumeration: PowerView**, we discussed a ton of options some of those are also present in Empire so those can seem to be repeating the similar approach but there are some more interactive modules here that are worth looking into. We will be using the same Active Directory Lab configuration from the PowerView Article mentioned above. In this Article/Demonstration, we are focused on our ability to Enumerate Information that can be then further be used to elevate privileges or be able to help with Lateral Movement. A tool by the name of PowerView was developed and integrated **by Will Schroeder (a.k.a harmj0y)** for

PowerSploit. It soon became an integral toolkit to perform Active Directory Attacks and Enumeration. We will be using PowerShell Empire to demonstrate the various Enumeration Tactics by PowerView.

What is Situational Awareness?

Situational Awareness is defined as: “Within a volume of time and space, the perception of an enterprise’s security posture and its threat environment; the comprehension/meaning of both taken together (risk); and the projection of their status into the near future.” In simpler terms learning and understanding the structure of any enterprise or network in a particular set of time while making a note of potential risks and making a plan of action is called Situational Awareness.

Get User

In our Active Directory Lab Setup, we created 8 users with different roles and privileges. Then when we emulate the attack on the AD from PowerShell Empire using Kali Linux as demonstrated, we generate the following result.

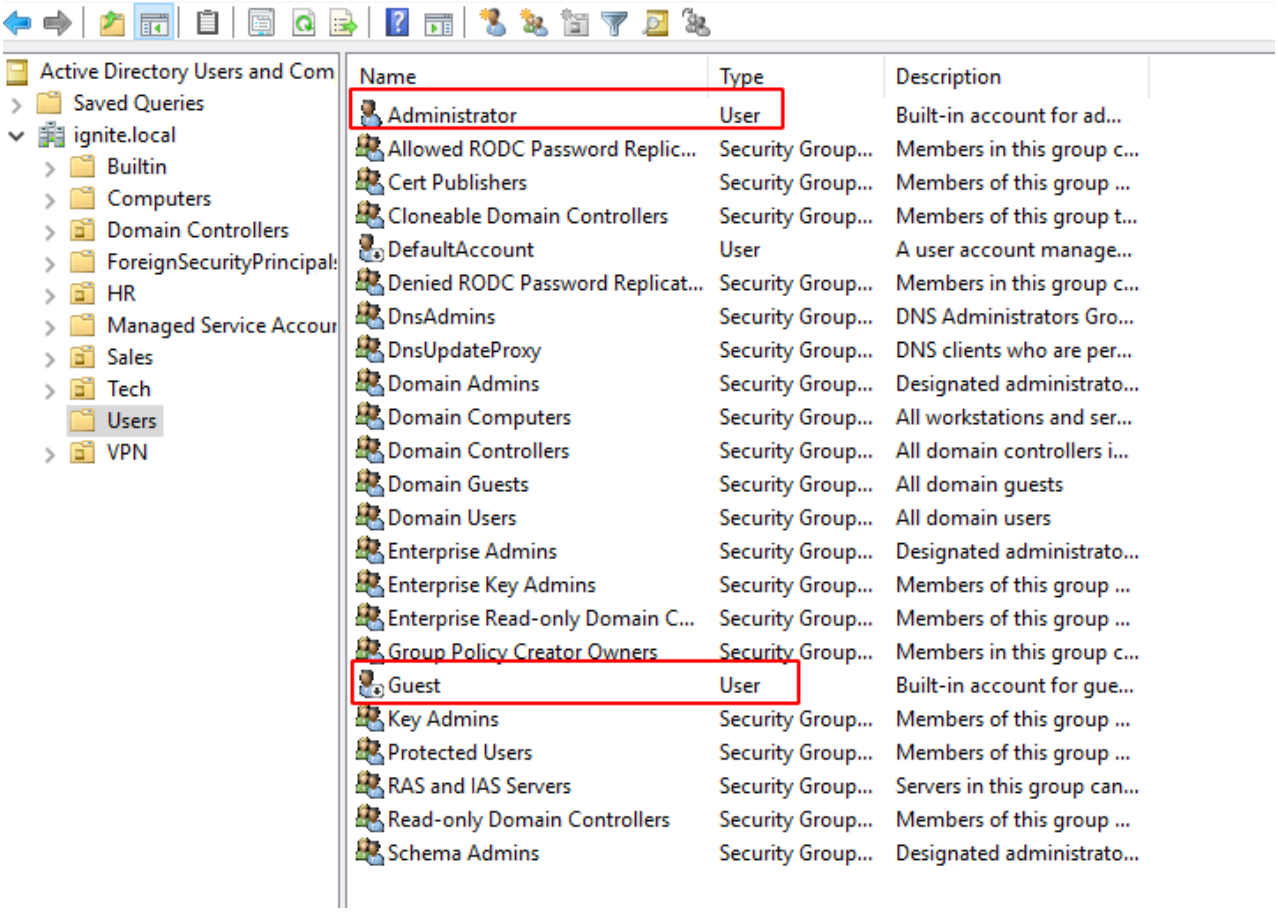
```
usemodule situational_awareness/network/powerview/get_user
execute
```

```
(Empire: TC4UKELM) > usemodule situational_awareness/network/powerview/get_user
(Empire: powershell/situational_awareness/network/powerview/get_user) > execute
[*] Tasked TC4UKELM to run TASK_CMD_JOB
[*] Agent TC4UKELM tasked with task ID 1
[*] Tasked agent TC4UKELM to run module powershell/situational_awareness/network/powerview/get_
(Empire: powershell/situational_awareness/network/powerview/get_user) >
Job started: TBLVC2



logoncount             : 60
badpasswordtime        : 4/2/2021 8:39:44 AM
description            : Built-in account for administering the computer/domain
distinguishedname      : CN=Administrator,CN=Users,DC=ignite,DC=local
objectclass            : {top, person, organizationalPerson, user}
lastlogontimestamp     : 4/2/2021 1:34:59 PM
name                   : Administrator
objectsid              : S-1-5-21-501555289-2168925624-2051597760-500
samaccountname         : Administrator
admincount             : 1
codepage               : 0
samaccounttype         : USER_OBJECT
accountexpires         : NEVER
countrycode            : 0
whenchanged            : 4/2/2021 8:34:59 PM
instancetype           : 4
objectguid             : c00f6d7e-69c7-44cf-ba81-0a513e8aaac4
lastlogon              : 4/7/2021 5:32:02 AM
lastlogoff             : 12/31/1600 4:00:00 PM
objectcategory         : CN=Person,CN=Schema,CN=Configuration,DC=ignite,DC=local
dscorepropagationdata  : {7/6/2020 5:39:37 PM, 7/6/2020 5:39:37 PM, 6/29/2020 4:54:43 PM, 1/1/1
memberof               : {CN=Group Policy Creator Owners,CN=Users,DC=ignite,DC=local, CN=Domain
                         Admins,CN=Users,DC=ignite,DC=local, CN=Enterprise Admins,CN=Users,DC=i
                         CN=Schema Admins,CN=Users,DC=ignite,DC=local ... }
whencreated            : 6/29/2020 4:54:05 PM
iscriticalsystemobject : True
badpwdcount            : 0
cn                     : Administrator
useraccountcontrol     : NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD
usncreated             : 8196
primarygroupid         : 513
pwdlastset             : 6/29/2020 9:40:26 AM
usnchanged             : 106631

pwdlastset             : 12/31/1600 4:00:00 PM
logoncount             : 0
badpasswordtime        : 12/31/1600 4:00:00 PM
description            : Built-in account for guest access to the computer/domain
distinguishedname      : CN=Guest,CN=Users,DC=ignite,DC=local
objectclass            : {top, person, organizationalPerson, user}
name                   : Guest
objectsid              : S-1-5-21-501555289-2168925624-2051597760-501
samaccountname         : Guest
codepage               : 0
samaccounttype         : USER_OBJECT
accountexpires         : NEVER
```

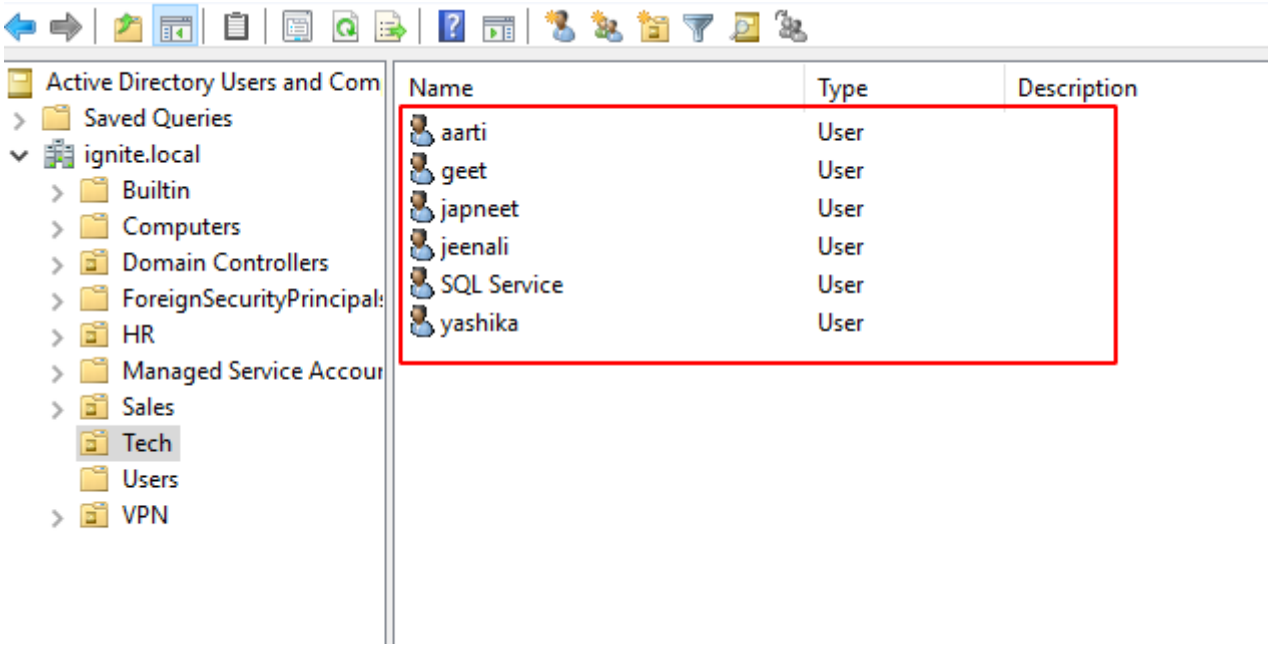
Users that are enumerated are not just restricted to Usernames. Data collected consist of logoncount that can give an idea of an active or inactive user in the network. Next, there is a badpasswordtime which tells the last time and date that an attempt to log on was made with an invalid password on this account. Then a small description of the user with the names of groups that this particular user is part of. At last, 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 Behavior.

```
logoncount            : 41
badpasswordtime       : 4/3/2021 9:55:35 AM
distinguishedname     : CN=yashika,OU=Tech,DC=ignite,DC=local
objectclass           : {top, person, organizationalPerson, user}
displayname           : yashika
lastlogontimestamp    : 3/26/2021 11:24:23 AM
userprincipalname     : yashika@ignite.local
name                  : yashika
objectsid             : S-1-5-21-501555289-2168925624-2051597760-1103
samaccountname        : yashika
admincount            : 1
codepage              : 0
samaccounttype        : USER_OBJECT
accountexpires        : NEVER
countrycode           : 0
whenchanged           : 3/26/2021 6:37:49 PM
instancetype          : 4
usncreated            : 16577
objectguid            : d2ff2fb0-5f92-471b-b94c-a1bc5be262f2
lastlogoff            : 12/31/1600 4:00:00 PM
objectcategory        : CN=Person,CN=Schema,CN=Configuration,DC=ignite,DC=local
dscorepropagationdata : {3/26/2021 6:37:49 PM, 1/1/1601 12:00:00 AM}
givenname             : yashika
lastlogon             : 4/4/2021 9:19:23 AM
badpwdcount           : 0
cn                    : yashika
useraccountcontrol    : NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD
whencreated           : 6/29/2020 5:08:49 PM
primarygroupid        : 513
pwdlastset            : 6/29/2020 10:08:49 AM
usnchanged            : 81982

logoncount            : 0
badpasswordtime       : 12/31/1600 4:00:00 PM
distinguishedname     : CN=geet,OU=Tech,DC=ignite,DC=local
objectclass           : {top, person, organizationalPerson, user}
displayname           : geet
userprincipalname     : geet@ignite.local
name                  : geet
objectsid             : S-1-5-21-501555289-2168925624-2051597760-1104
samaccountname        : geet
codepage              : 0
samaccounttype        : USER_OBJECT
accountexpires        : NEVER
countrycode           : 0
whenchanged           : 6/29/2020 5:09:17 PM
instancetype          : 4
usncreated            : 16584
objectguid            : 944569dc-bae7-400b-8ba3-68bd6849a8ef
lastlogoff            : 12/31/1600 4:00:00 PM
objectcategory        : CN=Person,CN=Schema,CN=Configuration,DC=ignite,DC=local
dscorepropagationdata : 1/1/1601 12:00:00 AM
givenname             : geet
lastlogon             : 12/31/1600 4:00:00 PM
badpwdcount           : 0
cn                    : geet
useraccountcontrol    : NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD
```

Users Extracted are Administrator, Guest, Yashika, Geet. It is clear from the output that the user's Administrator and Guest are the part of Users Group. This can be verified using our Active Directory Setup as shown below.



And the users Yashika, Geet, etc are part of Tech OU. More data will be extracted on OU later.



Get Computer

The next module that the attacker can use against the target server is the Get Computer module. The information this module target is primarily the Computer Name. It also extracts other information as demonstrated.

```
situational_awareness/network/powerview/get_computer
execute
```

```
(Empire: TC4UKELM) > usemodule situational_awareness/network/powerview/get_computer
(Empire: powershell/situational_awareness/network/powerview/get_computer) > execute
[*] Tasked TC4UKELM to run TASK_CMD_JOB
[*] Agent TC4UKELM tasked with task ID 3
[*] Tasked agent TC4UKELM to run module powershell/situational_awareness/network/powerview/get_
(Empire: powershell/situational_awareness/network/powerview/get_computer) >
Job started: NBSG1A



pwdlastset                    : 4/7/2021 5:30:23 AM
logoncount                    : 100
msds-generationid             : {168, 207, 198, 26 ... }
serverreferencebl             : CN=DC1,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Config
                                l
badpasswordtime               : 12/31/1600 4:00:00 PM
distinguishedname             : CN=DC1,OU=Domain Controllers,DC=ignite,DC=local
objectclass                   : {top, person, organizationalPerson, user ... }
lastlogontimestamp            : 4/2/2021 8:36:12 AM
name                          : DC1
objectsid                     : S-1-5-21-501555289-2168925624-2051597760-1000
samaccountname                : DC1$
localpolicyflags              : 0
codepage                      : 0
samaccounttype                : MACHINE_ACCOUNT
whenchanged                   : 4/7/2021 12:30:23 PM
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=ignite,DC=local
objectguid                    : de681d91-bd3c-45df-8285-c9ceb8eb7c37
operatingsystemversion        : 10.0 (14393)
lastlogoff                    : 12/31/1600 4:00:00 PM
objectcategory                : CN=Computer,CN=Schema,CN=Configuration,DC=ignite,DC=local
dscorepropagationdata         : {6/29/2020 4:54:43 PM, 1/1/1601 12:00:01 AM}
serviceprincipalname          : {TERMSRV/DC1, TERMSRV/DC1.ignite.local,
                                Dfsr-12F9A27C-BF97-4787-9364-D31B6C55EB04/DC1.ignite.local,
                                ldap/DC1.ignite.local/ForestDnsZones.ignite.local ... }
usncreated                    : 12293
memberof                      : CN=RAS and IAS Servers,CN=Users,DC=ignite,DC=local
lastlogon                     : 4/7/2021 5:31:32 AM
badpwdcount                   : 0
cn                            : DC1
useraccountcontrol            : SERVER_TRUST_ACCOUNT, TRUSTED_FOR_DELEGATION
whencreated                   : 6/29/2020 4:54:43 PM
primarygroupid                : 516
iscriticalsystemobject        : True
msds-supportedencryptiontypes : 28
usnchanged                    : 147496
ridsetreferences              : CN=RID Set,CN=DC1,OU=Domain Controllers,DC=ignite,DC=local
dnshostname                   : DC1.ignite.local
```

The output of the result that is generated by the module starts with information like pwdlastset information. This is the date and time when the user has reset their password. As discussed earlier it can help the attacker distinguish between active and inactive users. It can also help the user distinguish between the users that use proper security mechanisms and change passwords regularly and those who don't. Moving on, it also prints the username that is logged in on the Computer. Then it informs the attacker about the Operating System that is running on the target machine.

```
useraccountcontrol              : SERVER_TRUST_ACCOUNT, TRUSTED_FOR_DELEGATION
whencreated                     : 6/29/2020 4:54:43 PM
primarygroupid                  : 516
iscriticalsystemobject          : True
msds-supportedencryptiontypes   : 28
usnchanged                      : 147496
ridsetreferences                : CN=RID Set,CN=DC1,OU=Domain Controllers,DC=ignite,DC=
dnshostname                     : DC1.ignite.local

logoncount                      : 8
badpasswordtime                 : 12/31/1600 4:00:00 PM
distinguishedname               : CN=CLIENT,CN=Computers,DC=ignite,DC=local
objectclass                     : {top, person, organizationalPerson, user ... }
badpwdcount                     : 0
lastlogontimestamp              : 9/23/2020 10:11:02 AM
objectsid                       : S-1-5-21-501555289-2168925624-2051597760-2101
samaccountname                  : CLIENT$
localpolicyflags                : 0
codepage                        : 0
samaccounttype                  : MACHINE_ACCOUNT
countrycode                     : 0
cn                              : CLIENT
accountexpires                  : NEVER
whenchanged                     : 9/23/2020 5:11:32 PM
instancetype                    : 4
usncreated                      : 45103
objectguid                      : eb45051d-ae46-4e52-a86a-2ddbcdffa213
operatingsystem                 : Windows 10 Pro
operatingsystemversion          : 10.0 (18362)
lastlogoff                      : 12/31/1600 4:00:00 PM
objectcategory                  : CN=Computer,CN=Schema,CN=Configuration,DC=ignite,DC=
dscorepropagationdata           : 1/1/1601 12:00:00 AM
serviceprincipalname            : {RestrictedKrbHost/CLIENT, HOST/CLIENT, RestrictedKr
                                  HOST/client.ignite.local}
lastlogon                       : 9/23/2020 10:18:49 AM
iscriticalsystemobject          : False
usnchanged                      : 45122
useraccountcontrol              : WORKSTATION_TRUST_ACCOUNT
whencreated                     : 9/23/2020 5:11:01 PM
primarygroupid                  : 515
pwdlastset                      : 9/23/2020 10:11:32 AM
msds-supportedencryptiontypes   : 28
name                            : CLIENT
dnshostname                     : client.ignite.local

logoncount                      : 56
badpasswordtime                 : 12/31/1600 4:00:00 PM
distinguishedname               : CN=DESKTOP-ATNONJ9,CN=Computers,DC=ignite,DC=local
objectclass                     : {top, person, organizationalPerson, user ... }
badpwdcount                     : 0
lastlogontimestamp              : 3/26/2021 11:24:23 AM
objectsid                       : S-1-5-21-501555289-2168925624-2051597760-2102
samaccountname                  : DESKTOP-ATNONJ9$
localpolicyflags                : 0
codepage                        : 0
samaccounttype                  : MACHINE_ACCOUNT
countrycode                     : 0
```

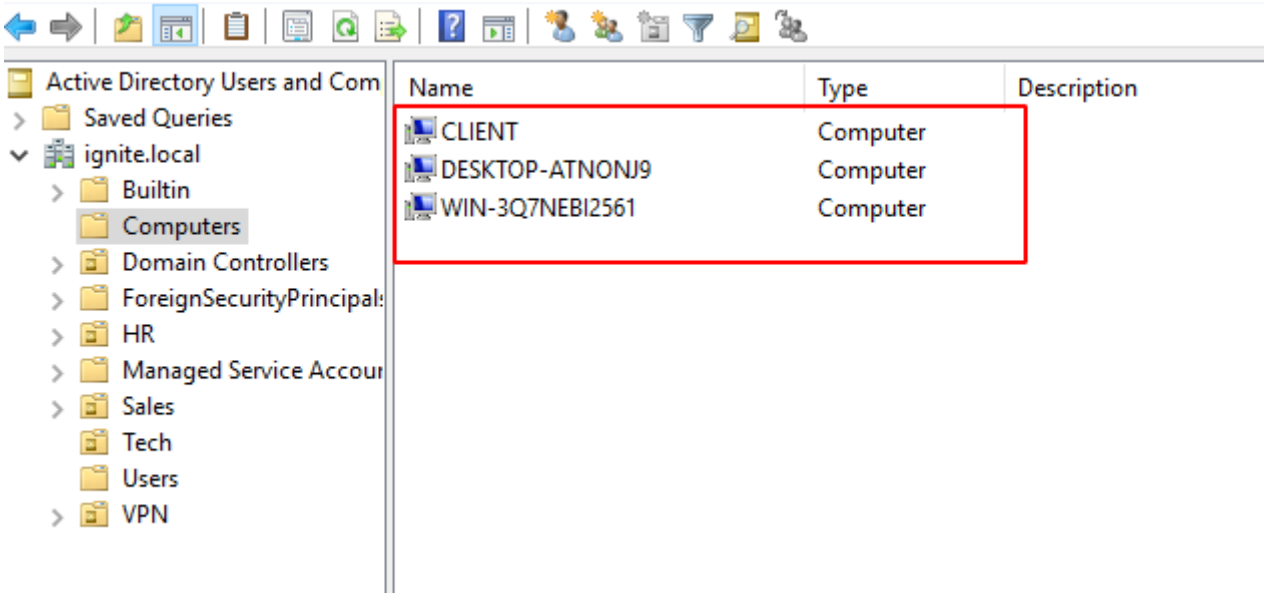
The output also tells the attacker the last time when the target machine was logged off. This can also help differentiate among users. Some other information that is extracted contains the badpwdcount that tells the number of times an incorrect password was attempted on that particular machine. Then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

```
logoncount                    : 56
badpasswordtime               : 12/31/1600 4:00:00 PM
distinguishedname             : CN=DESKTOP-ATNONJ9,CN=Computers,DC=ignite,DC=local
objectclass                   : {top, person, organizationalPerson, user ... }
badpwdcount                   : 0
lastlogontimestamp            : 3/26/2021 11:24:23 AM
objectsid                     : S-1-5-21-501555289-2168925624-2051597760-2102
samaccountname                : DESKTOP-ATNONJ9$
localpolicyflags              : 0
codepage                      : 0
samaccounttype                : MACHINE_ACCOUNT
countrycode                   : 0
cn                            : DESKTOP-ATNONJ9
accountexpires                : NEVER
whenchanged                   : 4/2/2021 8:34:59 PM
instancetype                  : 4
usncreated                    : 57378
objectguid                    : 87e76131-3cbb-4f64-8ed5-e6a3952194e0
operatingsystem               : Windows 10 Pro
operatingsystemversion        : 10.0 (18362)
lastlogoff                    : 12/31/1600 4:00:00 PM
objectcategory                : CN=Computer,CN=Schema,CN=Configuration,DC=ignite,DC=local
dscorepropagationdata         : 1/1/1601 12:00:00 AM
serviceprincipalname          : {TERMSRV/DESKTOP-ATNONJ9, TERMSRV/DESKTOP-ATNONJ9.ignite.l
                                RestrictedKrbHost/DESKTOP-ATNONJ9, HOST/DESKTOP-ATNONJ9 ...
lastlogon                     : 4/4/2021 1:27:16 PM
iscriticalsystemobject        : False
usnchanged                    : 106635
useraccountcontrol            : WORKSTATION_TRUST_ACCOUNT
whencreated                   : 3/6/2021 4:17:59 PM
primarygroupid                : 515
pwdlastset                    : 4/2/2021 1:34:59 PM
msds-supportedencryptiontypes : 28
name                          : DESKTOP-ATNONJ9
dnshostname                   : DESKTOP-ATNONJ9.ignite.local

logoncount                    : 4
badpasswordtime               : 12/31/1600 4:00:00 PM
distinguishedname             : CN=WIN-3Q7NEBI2561,CN=Computers,DC=ignite,DC=local
objectclass                   : {top, person, organizationalPerson, user ... }
badpwdcount                   : 0
lastlogontimestamp            : 3/27/2021 11:12:00 AM
objectsid                     : S-1-5-21-501555289-2168925624-2051597760-2103
samaccountname                : WIN-3Q7NEBI2561$
localpolicyflags              : 0
codepage                      : 0
samaccounttype                : MACHINE_ACCOUNT
```

Moreover, the attacker can also enumerate the SID of the user and OU of that particular user that is logged in on the machine. This can also tell the attacker if a particular user is about to be expired or is it set to never expire. Then we have the Group Details of the user as well.

```
logoncount                    : 4
badpasswordtime               : 12/31/1600 4:00:00 PM
distinguishedname             : CN=WIN-3Q7NEBI2561,CN=Computers,DC=ignite,DC=loc
objectclass                   : {top, person, organizationalPerson, user ... }
badpwdcount                   : 0
lastlogontimestamp            : 3/27/2021 11:12:00 AM
objectsid                     : S-1-5-21-501555289-2168925624-2051597760-2103
samaccountname                : WIN-3Q7NEBI2561$
localpolicyflags              : 0
codepage                      : 0
samaccounttype                : MACHINE_ACCOUNT
operatingsystemservicepack    : Service Pack 1
countrycode                   : 0
cn                            : WIN-3Q7NEBI2561
accountexpires                : NEVER
whenchanged                   : 3/27/2021 6:12:31 PM
instancetype                  : 4
usncreated                    : 90157
objectguid                    : 90179f2d-ed05-4e3e-9a7f-d6933b527f54
operatingsystem               : Windows 7 Ultimate
operatingsystemversion        : 6.1 (7601)
lastlogoff                    : 12/31/1600 4:00:00 PM
objectcategory                : CN=Computer,CN=Schema,CN=Configuration,DC=ignite
dscorepropagationdata         : 1/1/1601 12:00:00 AM
serviceprincipalname          : {RestrictedKrbHost/WIN-3Q7NEBI2561, HOST/WIN-3Q7
                                RestrictedKrbHost/WIN-3Q7NEBI2561.ignite.local,
lastlogon                     : 3/27/2021 11:12:38 AM
iscriticalsystemobject        : False
usnchanged                    : 90167
useraccountcontrol            : WORKSTATION_TRUST_ACCOUNT
whencreated                   : 3/27/2021 6:11:59 PM
primarygroupid                : 515
pwdlastset                    : 3/27/2021 11:11:59 AM
msds-supportedencryptiontypes : 28
name                          : WIN-3Q7NEBI2561
dnshostname                   : WIN-3Q7NEBI2561.ignite.local
```

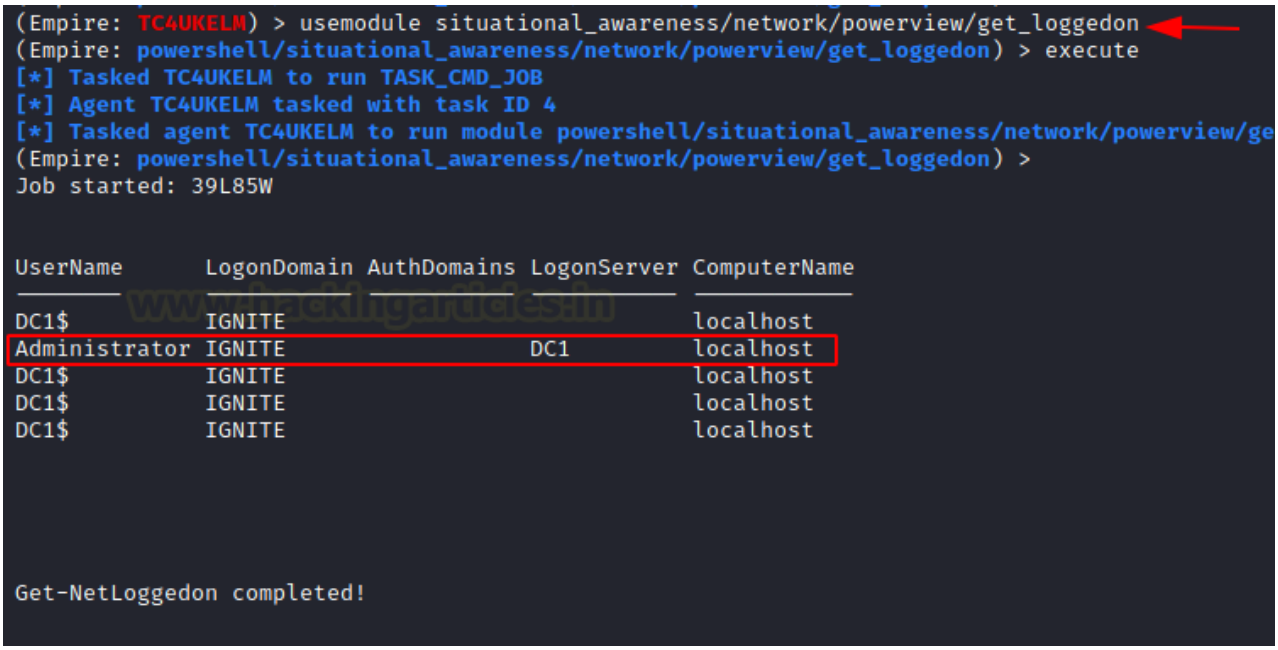
We can see that the output suggests that there are 3 machines in the Domain. Named as CLIENT, DESKTOP-ATNONJ9, and WIN-3Q7NEB12561. This can be verified from the Domain Controller as shown in the image below.



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 executes the NetWkstaUserEnum Win32API call to extract the users that are currently logged on. It can be observed the module has extracted the users that are logged in.

```
situational_awareness/network/powerview/get_loggedon
execute
```



Process Hunter

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 tells the Process ID of the process so if the attacker has access to that process, they can tinker around with it such as stopping or restarting such process.

```
situational_awareness/network/powerview/process_hunter
execute
```

```
(Empire: TC4UKELM) > usemodule situational_awareness/network/powerview/process_hunter
(Empire: powershell/situational_awareness/network/powerview/process_hunter) > execute
[*] Tasked TC4UKELM to run TASK_CMD_JOB
[*] Agent TC4UKELM tasked with task ID 7
[*] Tasked agent TC4UKELM to run module powershell/situational_awareness/network/powerview/p
(Empire: powershell/situational_awareness/network/powerview/process_hunter) >
Job started: X13PHU


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

ComputerName : DC1.ignite.local
ProcessName  : RuntimeBroker.exe
ProcessID    : 3680
Domain       : IGNITE
User         : Administrator


ComputerName : DC1.ignite.local
ProcessName  : sihost.exe
ProcessID    : 3812
Domain       : IGNITE
User         : Administrator


ComputerName : DC1.ignite.local
ProcessName  : svchost.exe
ProcessID    : 1148
Domain       : IGNITE
User         : Administrator


ComputerName : DC1.ignite.local
ProcessName  : taskhostw.exe
ProcessID    : 3476
Domain       : IGNITE
User         : Administrator


ComputerName : DC1.ignite.local
ProcessName  : explorer.exe
ProcessID    : 380
Domain       : IGNITE
User         : Administrator


ComputerName : DC1.ignite.local
ProcessName  : ShellExperienceHost.exe
ProcessID    : 4256
Domain       : IGNITE
User         : Administrator
```

The correlation can be done between the extracted data from Process Hunter and the actual tasks running on the machine by listing the process on the target machine. It has been demonstrated below using the tasklist command. The PID can be used to verify the process status.

```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>tasklist

Image Name                     PID Session Name        Session#    Mem Usage
========================= ======== ================ =========== ============
System Idle Process              0 Services                   0          4 K
System                           4 Services                   0        136 K
smss.exe                       280 Services                   0      1,192 K
csrss.exe                      384 Services                   0      4,328 K
wininit.exe                    500 Services                   0      4,872 K
csrss.exe                      508 Console                    1     10,912 K
winlogon.exe                   564 Console                    1      8,648 K
services.exe                   636 Services                   0     10,624 K
lsass.exe                      652 Services                   0     62,196 K
svchost.exe                    836 Services                   0     19,524 K
svchost.exe                    892 Services                   0     11,964 K
svchost.exe                   1020 Services                   0     11,828 K
dwm.exe                        332 Console                    1    109,800 K
svchost.exe                    632 Services                   0     33,364 K
svchost.exe                    324 Services                   0     26,224 K
svchost.exe                    912 Services                   0     23,892 K
svchost.exe                    936 Services                   0     22,880 K
svchost.exe                   1184 Services                   0     19,836 K
svchost.exe                   1236 Services                   0     68,728 K
svchost.exe                   1408 Services                   0      7,120 K
WmiPrvSE.exe                  2056 Services                   0     18,776 K
svchost.exe                   2180 Services                   0     10,764 K
spoolsv.exe                   2352 Services                   0     15,796 K
svchost.exe                   2444 Services                   0     23,188 K
vmtoolsd.exe                  2460 Services                   0     21,724 K
dfsrs.exe                     2484 Services                   0     21,312 K
Microsoft.ActiveDirectory     2512 Services                   0     40,244 K
wlms.exe                      2528 Services                   0      3,144 K
svchost.exe                   2536 Services                   0     10,544 K
svchost.exe                   2544 Services                   0     17,964 K
svchost.exe                   2552 Services                   0     11,484 K
ismserv.exe                   2560 Services                   0      5,576 K
dns.exe                       2592 Services                   0    125,460 K
VGAuthService.exe             2676 Services                   0     10,392 K
dfssvc.exe                    2728 Services                   0      7,232 K
sppsvc.exe                    2772 Services                   0     18,500 K
vds.exe                       3124 Services                   0     10,688 K
dllhost.exe                   3356 Services                   0     12,520 K
SppExtComObj.Exe              3428 Services                   0     11,596 K
msdtc.exe                     3500 Services                   0      9,644 K
RuntimeBroker.exe             3680 Console                    1     22,736 K
sihost.exe                    3812 Console                    1     21,588 K
svchost.exe                   1148 Console                    1     19,752 K
taskhostw.exe                 3476 Console                    1     16,372 K
explorer.exe                   380 Console                    1     47,800 K
ShellExperienceHost.exe       4256 Console                    1     64,352 K
SearchUI.exe                  4352 Console                    1    113,648 K
```

Get OU

OUs are the smallest unit in the Active Directory system. OU is abbreviated from is Organizational Unit. OUs are containers for users, groups, and computers, and they exist within a domain. OUs are useful when an administrator wants to deploy Group Policy

settings to a subset of users, groups, and computers within your 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 run execute the command.

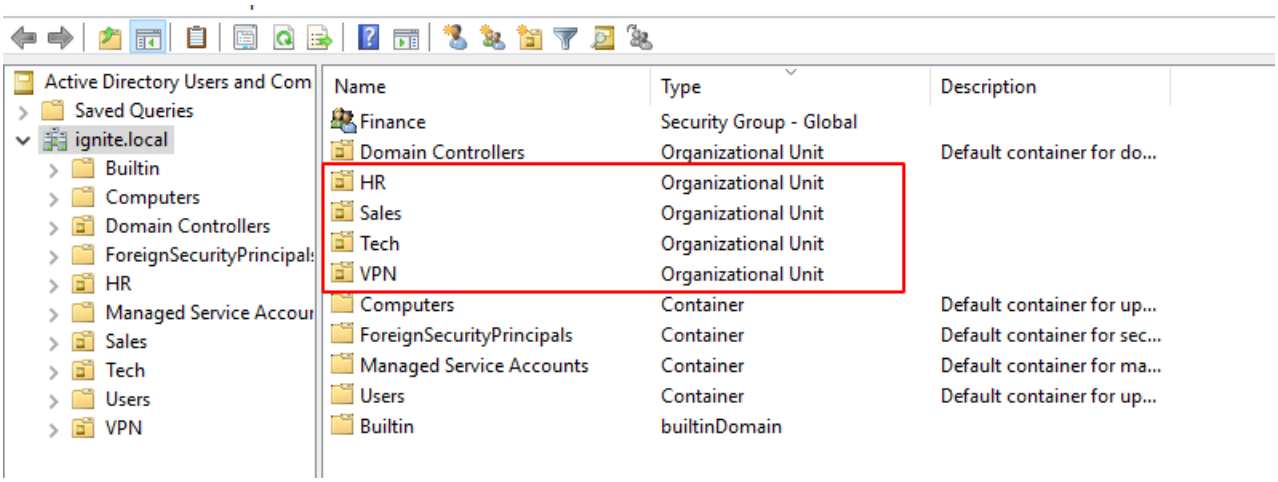
```
usemodule situational_awareness/network/powerview/get_ou
execute
```

```
(Empire: TC4UKELM) > usemodule situational_awareness/network/powerview/get_ou
(Empire: powershell/situational_awareness/network/powerview/get_ou) > execute
[*] Tasked TC4UKELM to run TASK_CMD_JOB
[*] Agent TC4UKELM tasked with task ID 8
[*] Tasked agent TC4UKELM to run module powershell/situational_awareness/network/powerview/get_
(Empire: powershell/situational_awareness/network/powerview/get_ou) >
Job started: 4R9LHD



usncreated             : 6031
systemflags            : -1946157056
iscriticalsystemobject : True
gplink                 : [LDAP://CN={6AC1786C-016F-11D2-945F-00C04fB984F9},CN=Policies,CN=Syste
whenchanged            : 6/29/2020 4:54:05 PM
objectclass            : {top, organizationalUnit}
showinadvancedviewonly : False
usnchanged             : 6031
dscorepropagationdata  : {6/29/2020 5:08:18 PM, 6/29/2020 4:54:43 PM, 1/1/1601 12:04:16 AM}
name                   : Domain Controllers
description            : Default container for domain controllers
distinguishedname      : OU=Domain Controllers,DC=ignite,DC=local
ou                     : Domain Controllers
whencreated            : 6/29/2020 4:54:05 PM
instancetype           : 4
```

As soon as the module is executed, it contacts the Target Server and extracts the requested information and then PowerShell Empire starts to print the response. Information such as gplink, object class, name of OUs, Date and Time of Creation, etc is printed for each OUs.

```
whencreated           : 6/29/2020 5:08:18 PM
instancetype          : 4
objectcategory        : CN=Organizational-Unit,CN=Schema,CN=Configuration,DC=ignite,DC=local
ou                    : Tech
objectguid            : 07ed228d-f71e-47d6-abcb-21013bb355a6
whenchanged           : 6/29/2020 5:08:18 PM
name                  : Tech
distinguishedname     : OU=Tech,DC=ignite,DC=local
usnchanged            : 16574
objectclass           : {top, organizationalUnit}
usncreated            : 16573
dscorepropagationdata : {6/29/2020 5:08:18 PM, 6/29/2020 5:08:18 PM, 1/1/1601 12:00:00 AM}

whencreated           : 7/6/2020 5:32:25 PM
instancetype          : 4
objectcategory        : CN=Organizational-Unit,CN=Schema,CN=Configuration,DC=ignite,DC=local
ou                    : VPN
objectguid            : f7b098e9-0ad0-4c60-866d-3148419c21a5
whenchanged           : 7/6/2020 7:45:48 PM
name                  : VPN
distinguishedname     : OU=VPN,DC=ignite,DC=local
usnchanged            : 28733
objectclass           : {top, organizationalUnit}
usncreated            : 20507
dscorepropagationdata : {7/6/2020 5:32:25 PM, 1/1/1601 12:00:00 AM}

whencreated           : 4/3/2021 7:49:14 PM
instancetype          : 4
objectcategory        : CN=Organizational-Unit,CN=Schema,CN=Configuration,DC=ignite,DC=local
ou                    : Sales
objectguid            : 36b91099-4b69-4fb6-ad33-caf22759b0ad
whenchanged           : 4/3/2021 7:49:14 PM
name                  : Sales
distinguishedname     : OU=Sales,DC=ignite,DC=local
usnchanged            : 118855
objectclass           : {top, organizationalUnit}
usncreated            : 118854
dscorepropagationdata : {4/3/2021 7:49:14 PM, 1/1/1601 12:00:00 AM}

whencreated           : 4/3/2021 7:49:34 PM
instancetype          : 4
objectcategory        : CN=Organizational-Unit,CN=Schema,CN=Configuration,DC=ignite,DC=local
ou                    : HR
objectguid            : b7791e4e-d5b2-438b-b53b-cd5d4f5d8c04
whenchanged           : 4/3/2021 7:49:34 PM
name                  : HR
distinguishedname     : OU=HR,DC=ignite,DC=local
usnchanged            : 118857
objectclass           : {top, organizationalUnit}
usncreated            : 118856
dscorepropagationdata : {4/3/2021 7:49:34 PM, 1/1/1601 12:00:00 AM}
```

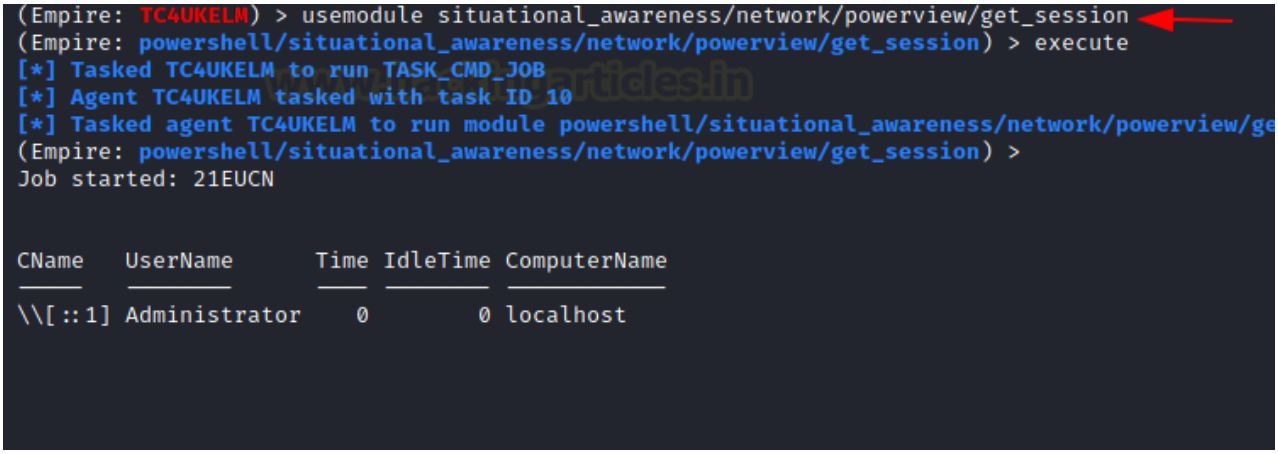
It can be observed that there are 4 OUs on the Target Server. Namely, Tech, VPN, Sales, and HR. To verify, we can take a look at the OUs directly from the Server. There are 4 OUs listed. This means that our module worked accurately.



Get Session

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. This function executes the NetSessionEnum Win32API call for extracting the session information.

```
situational_awareness/network/powerview/get_session
execute
```



Get Domain Controller

Next on the lineup, we have the Get DomainController. This provides the information of the particular server device instead of the domain. When an attacker wants to extract the data about the Domain Controller Machine then this tool can be used. It extracts the Forest Information, with the Time and Date configured on the Server. It tells the OS Version that can help constraint the search for Kernel Exploits for the attacker. Then the attacker has the IP Addressing data with the Inbound and Outbound connections.

```
situational_awareness/network/powerview/get_domain_controller
execute
```

```
(Empire: TC4UKELM) > usemodule situational_awareness/network/powerview/get_domain_controller
(Empire: powershell/situational_awareness/network/powerview/get_domain_controller) > execute
[*] Tasked TC4UKELM to run TASK_CMD_JOB
[*] Agent TC4UKELM tasked with task ID 11
[*] Tasked agent TC4UKELM to run module powershell/situational_awareness/network/powerview/get_domain_controll
(Empire: powershell/situational_awareness/network/powerview/get_domain_controller) >
Job started: 2LPRU9



Forest                     : ignite.local
CurrentTime                : 4/7/2021 1:26:54 PM
HighestCommittedUsn        : 147517
OSVersion                  : Windows Server 2016 Standard Evaluation
Roles                      : {SchemaRole, NamingRole, PdcRole, RidRole ...}
Domain                     : ignite.local
IPAddress                  : ::1
SiteName                   : Default-First-Site-Name
SyncFromAllServersCallback :
InboundConnections         : {}
OutboundConnections        : {}
Name                       : DC1.ignite.local
Partitions                 : {DC=ignite,DC=local, CN=Configuration,DC=ignite,DC=local,
                             CN=Schema,CN=Configuration,DC=ignite,DC=local, DC=DomainDnsZones,DC=ignite,DC=loc
```

Get Group

Enumerating group information is one of the most important pieces of information an attacker should enumerate on its target. Group Information categories the uses and helps understand the users that have the high privilege or they might be the one that has the access to a particular database. This can be performed using the get group module as demonstrated.

```
situational_awareness/network/powerview/get_group
execute
```

```
(Empire: SGENTK7Z) > usemodule situational_awareness/network/powerview/get_group
(Empire: powershell/situational_awareness/network/powerview/get_group) > execute
[*] Tasked SGENTK7Z to run TASK_CMD_JOB
[*] Agent SGENTK7Z tasked with task ID 7
[*] Tasked agent SGENTK7Z to run module powershell/situational_awareness/network/powervie
(Empire: powershell/situational_awareness/network/powerview/get_group) >
Job started: L1P8X4
```

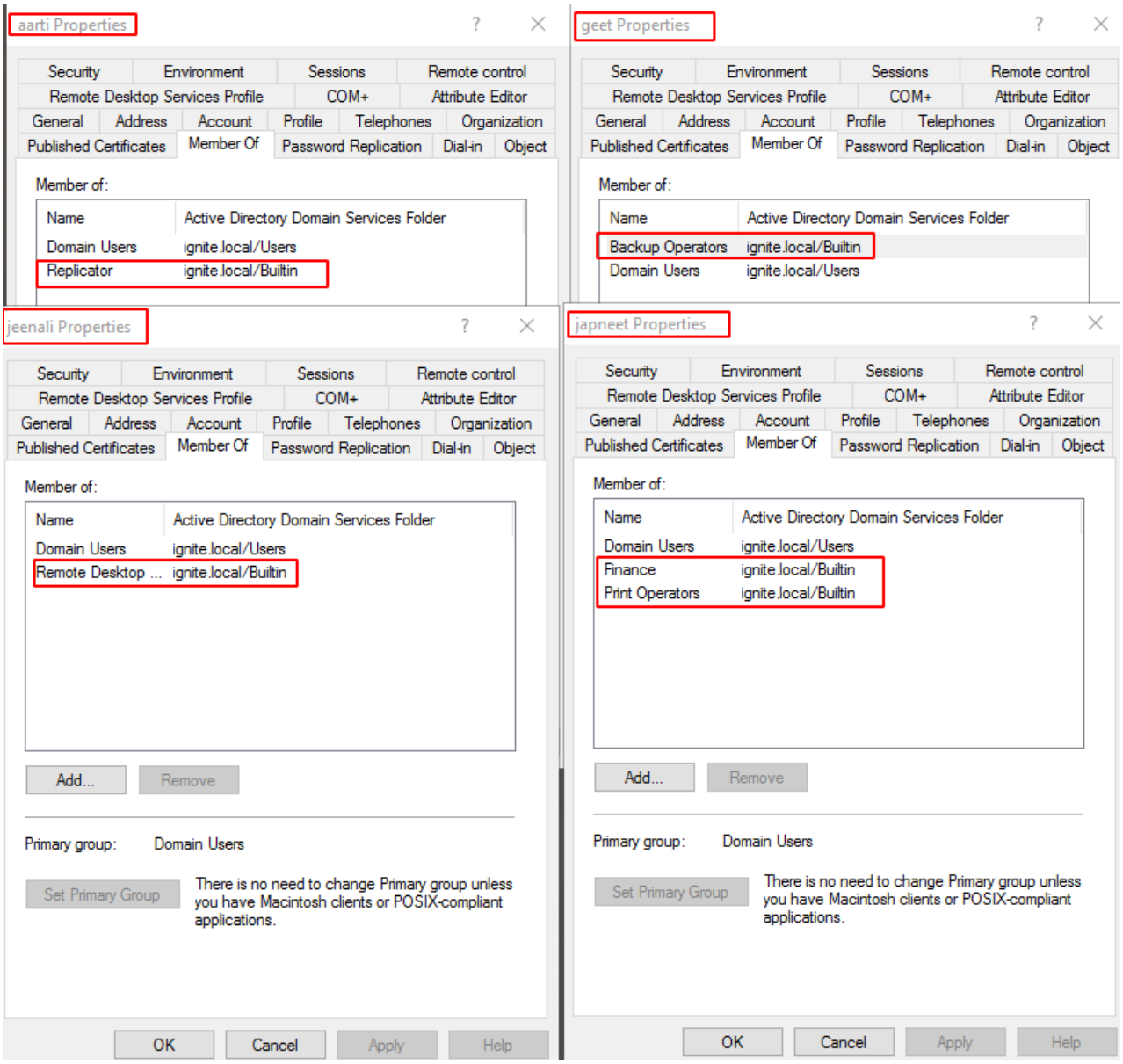
Upon analyzing the output of the module that we just discussed, we can see that we get a group by the name of Print Operators. To find the user inside that particular group there is a parameter named member. It can be seen that user Japneet is a part of the Print Operators group. Similarly, the Backup Operators group has the user geet. The interesting part about the backup operators is that they can read almost all the files on the system as you cannot make a backup of a file that you don't have permission to read. Hence it is worth trying to take over the user that is a part of the Backup Operators group.

```
grouptype               : CREATED_BY_SYSTEM, DOMAIN_LOCAL_SCOPE, SECURITY
admincount              : 1
iscriticalsystemobject  : True
samaccounttype          : ALIAS_OBJECT
samaccountname          : Print Operators
whenchanged             : 4/7/2021 1:45:55 PM
objectsid               : S-1-5-32-550
objectclass             : {top, group}
cn                      : Print Operators
usnchanged              : 151629
systemflags             : -1946157056
name                    : Print Operators
dscorepropagationdata   : {7/6/2020 5:39:37 PM, 6/29/2020 4:54:43 PM, 1/1/1601 12:04:16 A
description             : Members can administer printers installed on domain controller
distinguishedname       : CN=Print Operators,CN=Builtin,DC=ignite,DC=local
member                  : CN=japneet,OU=Tech,DC=ignite,DC=local
usncreated              : 8212
whencreated             : 6/29/2020 4:54:05 PM
instancetype            : 4
objectguid              : 2cda2d0f-0716-44dd-8ea8-1447d8da4ec6
objectcategory          : CN=Group,CN=Schema,CN=Configuration,DC=ignite,DC=local

grouptype               : CREATED_BY_SYSTEM, DOMAIN_LOCAL_SCOPE, SECURITY
admincount              : 1
iscriticalsystemobject  : True
samaccounttype          : ALIAS_OBJECT
samaccountname          : Backup Operators
whenchanged             : 4/7/2021 1:46:15 PM
objectsid               : S-1-5-32-551
objectclass             : {top, group}
cn                      : Backup Operators
usnchanged              : 151633
systemflags             : -1946157056
name                    : Backup Operators
dscorepropagationdata   : {7/6/2020 5:39:37 PM, 6/29/2020 4:54:43 PM, 1/1/1601 12:04:16 A
description             : Backup Operators can override security restrictions for the so
                          restoring files
distinguishedname       : CN=Backup Operators,CN=Builtin,DC=ignite,DC=local
member                  : CN=geet,OU=Tech,DC=ignite,DC=local
usncreated              : 8213
whencreated             : 6/29/2020 4:54:05 PM
instancetype            : 4
objectguid              : f2d07966-5803-493b-b7ef-3b77edc0fe15
objectcategory          : CN=Group,CN=Schema,CN=Configuration,DC=ignite,DC=local
```

Moving down the output we can see that there is a group by the name of Replicator. The member of Replicator is an aarti user. The members of this group can replicate the Active Directory Architecture. Next, we have the Remote Desktop Users group. This is also a group if compromised can pose disastrous consequences. This a group of users that have the privilege to access the desktop users. As can be observed from the screenshot the Jeenali user is a member of the Remote Desktop Users group.

```
grouptype               : CREATED_BY_SYSTEM, DOMAIN_LOCAL_SCOPE, SECURITY
admincount              : 1
iscriticalsystemobject  : True
samaccounttype          : ALIAS_OBJECT
samaccountname          : Replicator
whenchanged             : 4/7/2021 1:53:05 PM
objectsid               : S-1-5-32-552
objectclass             : {top, group}
cn                      : Replicator
usnchanged              : 151645
systemflags             : -1946157056
name                    : Replicator
dscorepropagationdata   : {7/6/2020 5:39:37 PM, 6/29/2020 4:54:43 PM, 1/1/1601 12:0
description             : Supports file replication in a domain
distinguishedname       : CN=Replicator,CN=Builtin,DC=ignite,DC=local
member                  : CN=aarti,OU=Tech,DC=ignite,DC=local
usncreated              : 8214
whencreated             : 6/29/2020 4:54:05 PM
instancetype            : 4
objectguid              : 602a5047-5246-44ef-8863-8a6e25f7010b
objectcategory          : CN=Group,CN=Schema,CN=Configuration,DC=ignite,DC=local

grouptype               : CREATED_BY_SYSTEM, DOMAIN_LOCAL_SCOPE, SECURITY
systemflags             : -1946157056
iscriticalsystemobject  : True
samaccounttype          : ALIAS_OBJECT
samaccountname          : Remote Desktop Users
whenchanged             : 4/7/2021 1:43:06 PM
objectsid               : S-1-5-32-555
objectclass             : {top, group}
cn                      : Remote Desktop Users
usnchanged              : 151625
dscorepropagationdata   : {6/29/2020 4:54:43 PM, 1/1/1601 12:00:01 AM}
name                    : Remote Desktop Users
description             : Members in this group are granted the right to logon remo
distinguishedname       : CN=Remote Desktop Users,CN=Builtin,DC=ignite,DC=local
member                  : CN=jeenali,OU=Tech,DC=ignite,DC=local
usncreated              : 8215
whencreated             : 6/29/2020 4:54:05 PM
instancetype            : 4
objectguid              : e7cbb628-0f6f-40aa-8da9-53b762bd0fc3
objectcategory          : CN=Group,CN=Schema,CN=Configuration,DC=ignite,DC=local
```

All the information that we extracted using the PowerView Module can be directly verified from the Domain Controller by checking the Properties of users. The properties will have a tab named Member Of. It will contain the name of the group that the user is part of.



Get Group Member

In the previous stage, we extracted the groups from usernames but this next module named get group member does the exact opposite. It requires the attacker to provide a group name and then it works to extract all the members of that particular user. In the demonstration below, we try to enumerate the users of the Domain Admin group. The module tells us that the Yashika user a member of the Domain Admin Group.

```
situational_awareness/network/powerview/get_group_member
set Recursive "Domain Admins"
execute
```

```
(Empire: SGENTK7Z) > usemodule situational_awareness/network/powerview/get_group_member
(Empire: powershell/situational_awareness/network/powerview/get_group_member) > set Recurse "Domain Admins"
(Empire: powershell/situational_awareness/network/powerview/get_group_member) > execute
[*] Tasked SGENTK7Z to run TASK_CMD_JOB
[*] Agent SGENTK7Z tasked with task ID 12
[*] Tasked agent SGENTK7Z to run module powershell/situational_awareness/network/powerview/get_group_member
(Empire: powershell/situational_awareness/network/powerview/get_group_member) >
Job started: 2TGZ7H



GroupDomain             : ignite.local
GroupName               : Domain Admins
GroupDistinguishedName  : CN=Domain Admins,CN=Users,DC=ignite,DC=local
MemberDomain            : ignite.local
MemberName              : yashika
MemberDistinguishedName : CN=yashika,OU=Tech,DC=ignite,DC=local
MemberObjectClass       : user
MemberSID               : S-1-5-21-501555289-2168925624-2051597760-1103

GroupDomain             : ignite.local
GroupName               : Domain Admins
GroupDistinguishedName  : CN=Domain Admins,CN=Users,DC=ignite,DC=local
MemberDomain            : ignite.local
MemberName              : Administrator
MemberDistinguishedName : CN=Administrator,CN=Users,DC=ignite,DC=local
MemberObjectClass       : user
MemberSID               : S-1-5-21-501555289-2168925624-2051597760-500
```

As always this can be simply verified on the Domain Controller by running the net group command with the group whose member you are trying to enumerate.

```
C:\Users\Administrator>net group "Domain Admins"
Group name     Domain Admins
Comment        Designated administrators of the domain

Members

-------------------------------------------------------------------------------
Administrator            yashika
The command completed successfully.


C:\Users\Administrator>
```

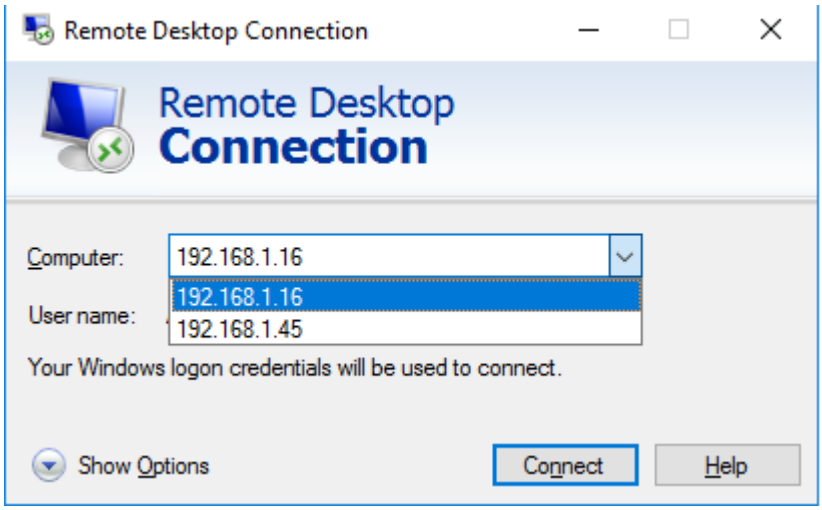
Get Cached RDP Connection

RDP or Remote Desktop Connections are one of the most used functionalities that are used in an enterprise. The work of induvial employees is also heavily dependent on the Remote Desktop connections while working from home. Windows can cache the devices that the user is trying to connect to using RDP. The get cached RDP connection uses remote registry functionality to query all entries for the “Windows Remote Desktop Connection Client” on the local (or a remote) machine

```
situational_awareness/network/powerview/get_cached_rdpconnection
execute
```

```
(Empire: SGENTK7Z) > usemodule situational_awareness/network/powerview/get_cached_rdpconnection
(Empire: powershell/situational_awareness/network/powerview/get_cached_rdpconnection) > execute
[*] Tasked SGENTK7Z to run TASK_CMD_JOB
[*] Agent SGENTK7Z tasked with task ID 30
[*] Tasked agent SGENTK7Z to run module powershell/situational_awareness/network/powerview/get_cached_rd
(Empire: powershell/situational_awareness/network/powerview/get_cached_rdpconnection) >
Job started: 81UNLH



ComputerName : localhost
UserName     : IGNITE\Administrator
UserSID      : S-1-5-21-501555289-2168925624-2051597760-500
TargetServer : 192.168.1.45
UsernameHint :

ComputerName : localhost
UserName     : IGNITE\Administrator
UserSID      : S-1-5-21-501555289-2168925624-2051597760-500
TargetServer : 192.168.1.16
UsernameHint :

ComputerName : localhost
UserName     : IGNITE\Administrator
UserSID      : S-1-5-21-501555289-2168925624-2051597760-500
TargetServer : 192.168.1.45
UsernameHint :





Get-WMIRegCachedRDPConnection completed!
```

As can be observed from the above image that the module has extracted 2 users that are supposed to be cached in the registry of the target machine. This can be verified from the RDP Connection Windows as shown below. The IP Address 192.168.1.16 and 192.168.1.45 are the devices that are controlled using RDP. This can help the attacker map other machines in the network and it also informs the attacker that RDP is enabled on these machines.



Find Local Administered Access

This next module helps that attacker to enumerate where the current user has local administration access. In simpler terms, it enumerates all machines on the current domain and for each machine, it checks if the current users have local administrator access. From the demonstration, it can be concluded that DC1 user has local administration access on this machine only.

```
situational_awareness/network/powerview/find_localadmin_access
execute
```

```
(Empire: D8G5H4Y6) > usemodule situational_awareness/network/powerview/find_localadmin_access
(Empire: powershell/situational_awareness/network/powerview/find_localadmin_access) > execute
[*] Tasked D8G5H4Y6 to run TASK_CMD_JOB
[*] Agent D8G5H4Y6 tasked with task ID 5
[*] Tasked agent D8G5H4Y6 to run module powershell/situational_awareness/network/powerview/find_local
(Empire: powershell/situational_awareness/network/powerview/find_localadmin_access) >
Job started: 3WGKH1

DC1.ignite.local



Find-LocalAdminAccess completed!
```

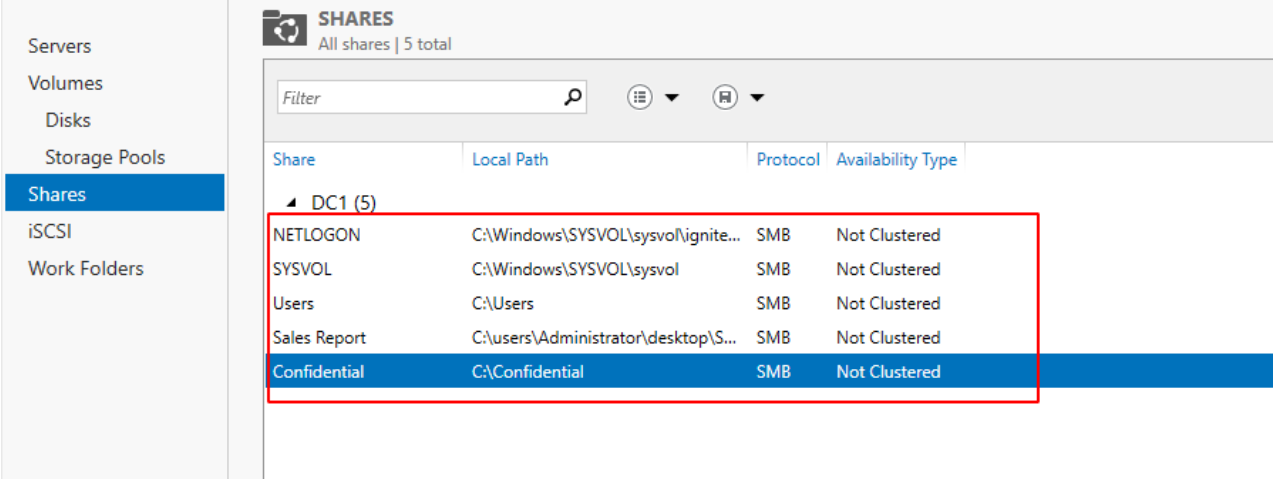
Share Finder

As the name suggests that this module can help the attacker extract shares hosted on the network. Any inexperienced attacker can tell that why is there a need for enumerating the shares when that can be done externally using the SMB enumeration. But an experienced attacker will know that some shares are not visible for all. It can be configured as to if that particular share is visible and accessible to all or some specific user.

```
situational_awareness/network/powerview/share_finder
execute
```

```
(Empire: D8G5H4Y6) > usemodule situational_awareness/network/powerview/share_finder
(Empire: powershell/situational_awareness/network/powerview/share_finder) > execute
[*] Tasked D8G5H4Y6 to run TASK_CMD_JOB
[*] Agent D8G5H4Y6 tasked with task ID 7
[*] Tasked agent D8G5H4Y6 to run module powershell/situational_awareness/network/powerview/sha
(Empire: powershell/situational_awareness/network/powerview/share_finder) >
Job started: SGNEL3


Name                Type Remark              ComputerName
----                ---- ------              ------------
ADMIN$        2147483648 Remote Admin        DC1.ignite.local
C$            2147483648 Default share       DC1.ignite.local
Confidential           0                     DC1.ignite.local
IPC$          2147483651 Remote IPC          DC1.ignite.local
NETLOGON               0 Logon server share  DC1.ignite.local
Sales Report           0                     DC1.ignite.local
SYSVOL                 0 Logon server share  DC1.ignite.local
Users                  0                     DC1.ignite.local
```

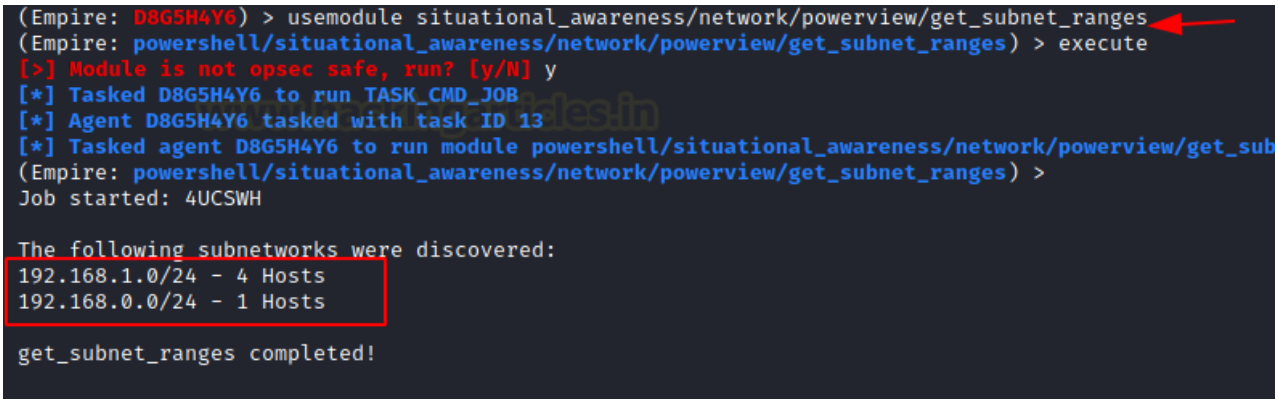
From the module above and the image of Server Manager below it can be seen that there shares by the name of Confidential and Sales Report in the network.



Get Subnet Ranges

Enumerating Subnets may seem like not a useful idea but there is something that could help the attacker to understand how the domain is laid out. Several hosts are connected to this particular subnet. It can also inform the attacker of other subnets in which the network is divided. In the demonstration below, there are 4 hosts connected to this particular subnet. That would probably split into 3 clients.

```
situational_awareness/network/powerview/get_subnet_ranges
execute
```



Get Forest

Apart from the domain information and the user information, the attacker can also gain information about the forests and there can be multiple forests inside a domain. To procure information about the forest in the current user's domain is to use the get forest module.

```
situational_awareness/network/powerview/get_forest
execute
```

```
(Empire: D8G5H4Y6) > usemodule situational_awareness/network/powerview/get_forest
(Empire: powershell/situational_awareness/network/powerview/get_forest) > execute
[*] Tasked D8G5H4Y6 to run TASK_CMD_JOB
[*] Agent D8G5H4Y6 tasked with task ID 17
[*] Tasked agent D8G5H4Y6 to run module powershell/situational_awareness/network/powerview/get_forest
(Empire: powershell/situational_awareness/network/powerview/get_forest) >
Job started: EKX5W3



RootDomainSid         : S-1-5-21-501555289-2168925624-2051597760
Name                  : ignite.local
Sites                 : {Default-First-Site-Name}
Domains               : {ignite.local}
GlobalCatalogs        : {DC1.ignite.local}
ApplicationPartitions : {DC=ForestDnsZones,DC=ignite,DC=local, DC=DomainDnsZones,DC=ignite,DC=local}
ForestModeLevel       : 7
ForestMode            : Unknown
RootDomain            : ignite.local
Schema                : CN=Schema,CN=Configuration,DC=ignite,DC=local
SchemaRoleOwner       : DC1.ignite.local
NamingRoleOwner       : DC1.ignite.local
```

Get Forest Domain

In simpler terms, a domain is a set of computers inside a boundary, which have a particular rule for accessing data and administering data values. Domains are situated inside trees. It can be said that a tree is a group or collection of domains that are arranged systematically bearing the same namespace. To enumerate the Forest Domain details including the name of the forest with its children and Domain Level then the attacker can use the get forest domain module.

```
situational_awareness/network/powerview/get_forest_domain
execute
```

```
(Empire: D8G5H4Y6) > usemodule situational_awareness/network/powerview/get_forest_domain
(Empire: powershell/situational_awareness/network/powerview/get_forest_domain) > execute
[*] Tasked D8G5H4Y6 to run TASK_CMD_JOB
[*] Agent D8G5H4Y6 tasked with task ID 18
[*] Tasked agent D8G5H4Y6 to run module powershell/situational_awareness/network/powerview/get_fo
(Empire: powershell/situational_awareness/network/powerview/get_forest_domain) >
Job started: EN7T6M



Forest                  : ignite.local
DomainControllers       : {DC1.ignite.local}
Children                : {}
DomainMode              : Unknown
DomainModeLevel         : 7
Parent                  :
PdcRoleOwner            : DC1.ignite.local
RidRoleOwner            : DC1.ignite.local
InfrastructureRoleOwner : DC1.ignite.local
Name                    : ignite.local
```

Get GPO

A Group Policy is created to figure out how the Domain is set up and what set of rules and policies are designed by the Administrator to govern the Domain. This can be enumerated using this module. It will extract all the information regarding Group Policies that are configured on the Target System.

```
situational_awareness/network/powerview/get_gpo
execute
```

```
(Empire: D8G5H4Y6) > usemodule situational_awareness/network/powerview/get_gpo
(Empire: powershell/situational_awareness/network/powerview/get_gpo) > execute
[*] Tasked D8G5H4Y6 to run TASK_CMD_JOB
[*] Agent D8G5H4Y6 tasked with task ID 23
[*] Tasked agent D8G5H4Y6 to run module powershell/situational_awareness/network/powerview/get_gpo
(Empire: powershell/situational_awareness/network/powerview/get_gpo) >
Job started: UGD65B



usncreated               : 5900
systemflags              : -1946157056
displayname              : Default Domain Policy
gpcmachineextensionnames : [{35378EAC-683F-11D2-A89A-00C04FBBCFA2}{53D6AB1B-2488-11D1-A28C-00C04FB9
                           -11D2-A4EA-00C04F79F83A}{803E14A0-B4FB-11D0-A0D0-00A0C90F574B}][{B1BE8D7
                           4F79F83A}{53D6AB1B-2488-11D1-A28C-00C04FB94F17}]
whenchanged              : 6/29/2020 5:04:39 PM
objectclass              : {top, container, groupPolicyContainer}
gpcfunctionalityversion  : 2
showinadvancedviewonly   : True
usnchanged               : 16421
dscorepropagationdata    : {6/29/2020 4:54:43 PM, 1/1/1601 12:00:00 AM}
name                     : {31B2F340-016D-11D2-945F-00C04FB984F9}
flags                    : 0
cn                       : {31B2F340-016D-11D2-945F-00C04FB984F9}
iscriticalsystemobject   : True
gpcfilesyspath           : \\ignite.local\sysvol\ignite.local\Policies\{31B2F340-016D-11D2-945F-00C
distinguishedname        : CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=ignit
whencreated              : 6/29/2020 4:54:05 PM
versionnumber            : 3
instancetype             : 4
objectguid               : 4aaf7089-5629-4f93-b6cc-0ecc1c4dba1e
objectcategory           : CN=Group-Policy-Container,CN=Schema,CN=Configuration,DC=ignite,DC=local

usncreated               : 5903
systemflags              : -1946157056
displayname              : Default Domain Controllers Policy
gpcmachineextensionnames : [{35378EAC-683F-11D2-A89A-00C04FBBCFA2}{D02B1F72-3407-48AE-BA88-E8213C67
                           -11D2-A4EA-00C04F79F83A}{803E14A0-B4FB-11D0-A0D0-00A0C90F574B}]
whenchanged              : 4/7/2021 4:46:25 PM
objectclass              : {top, container, groupPolicyContainer}
gpcfunctionalityversion  : 2
showinadvancedviewonly   : True
usnchanged               : 155719
dscorepropagationdata    : {6/29/2020 4:54:43 PM, 1/1/1601 12:00:00 AM}
name                     : {6AC1786C-016F-11D2-945F-00C04fB984F9}
flags                    : 0
cn                       : {6AC1786C-016F-11D2-945F-00C04fB984F9}
iscriticalsystemobject   : True
gpcfilesyspath           : \\ignite.local\sysvol\ignite.local\Policies\{6AC1786C-016F-11D2-945F-00C
distinguishedname        : CN={6AC1786C-016F-11D2-945F-00C04fB984F9},CN=Policies,CN=System,DC=ignit
whencreated              : 6/29/2020 4:54:05 PM
versionnumber            : 6
instancetype             : 4
objectguid               : f852ef84-af95-4083-ba7c-8eabfa710587
objectcategory           : CN=Group-Policy-Container,CN=Schema,CN=Configuration,DC=ignite,DC=local

usncreated               : 155735
```

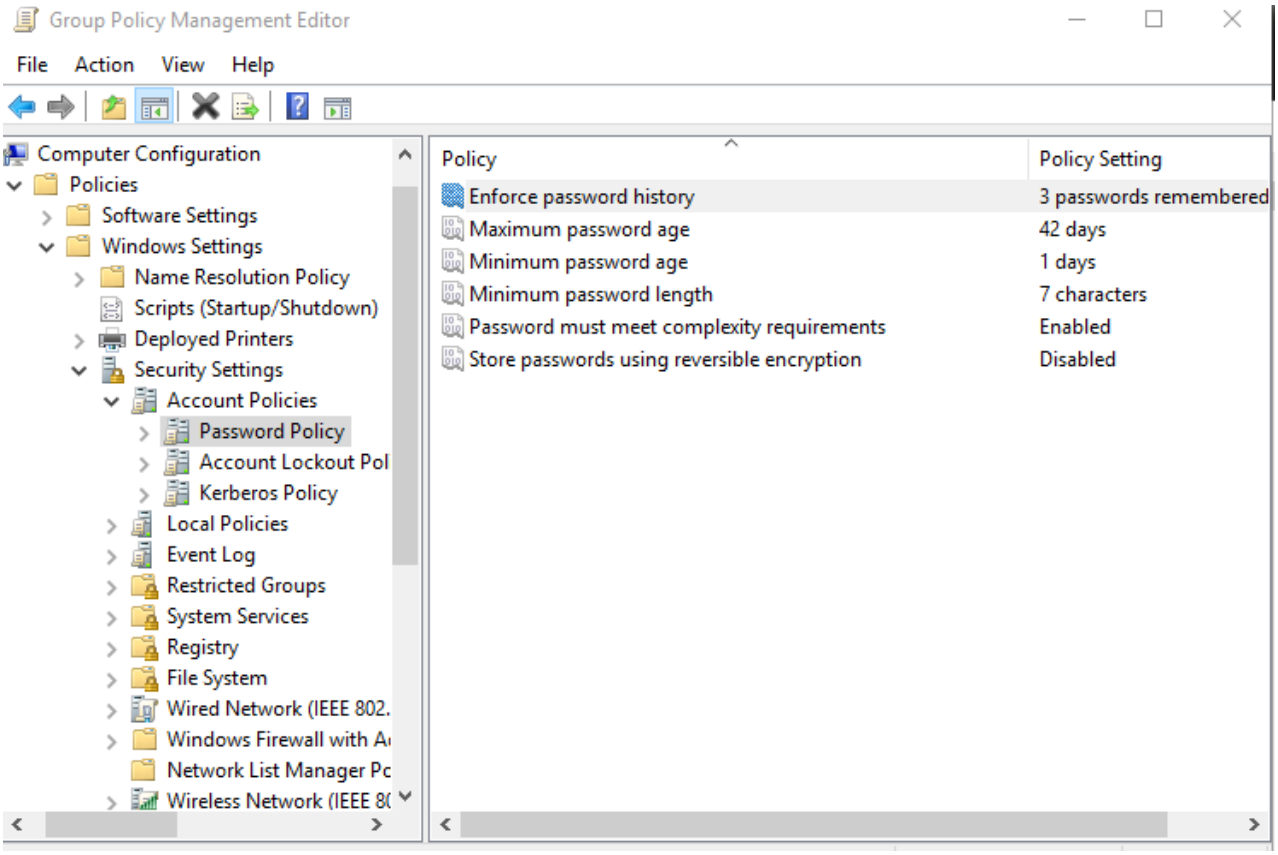
Get Domain Policy

The Domain Policy of a Domain can reveal some information such as extracting the policy of the current domain. It reads the default domain policy or the domain controller policy for the current domain or a specified domain/domain controller. In the demonstration, it can be observer a set of System Access Policy defined which include the Password Expiration Time and Minimum Password Length.

```
situational_awareness/network/powerview/get_domain_policy
execute
```

```
(Empire: UL2MCR1X) > usemodule situational_awareness/network/powerview/get_domain_policy
(Empire: powershell/situational_awareness/network/powerview/get_domain_policy) > execute
[*] Tasked UL2MCR1X to run TASK_CMD_JOB
[*] Agent UL2MCR1X tasked with task ID 8
[*] Tasked agent UL2MCR1X to run module powershell/situational_awareness/network/powerview/get_domain_policy
(Empire: powershell/situational_awareness/network/powerview/get_domain_policy) >
Job started: S5Y1AL



Unicode        : @{Unicode=yes}
SystemAccess   : @{MinimumPasswordAge=1; MaximumPasswordAge=42; MinimumPasswordLength=7; PasswordComplexity=1;
                 PasswordHistorySize=3; LockoutBadCount=0; RequireLogonToChangePassword=0; ForceLogoffWhenHourExpire=0;
                 ClearTextPassword=0; LSAAnonymousNameLookup=0}
KerberosPolicy : @{MaxTicketAge=10; MaxRenewAge=7; MaxServiceAge=600; MaxClockSkew=5; TicketValidateClient=1}
Version        : @{signature="$CHICAGO$"; Revision=1}
RegistryValues : @{MACHINE\System\CurrentControlSet\Control\Lsa\NoLMHash=System.Object[]}
Path           : \\ignite.local\sysvol\ignite.local\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Microsoft\Wi
                 ndows NT\SecEdit\GptTmpl.inf
GPOName        : {31B2F340-016D-11D2-945F-00C04FB984F9}
GPODisplayName : Default Domain Policy
```

This can be verified from the Group Policy Management Editor on the Domain Controller. You can create more policies and just configure other policies.



Get RDP Session

This module enumerates the remote (or local) RDP sessions on a remote machine that the Administrator has access to. It also pulls in the originating IP of the connection as well. In the demonstration, it can be observed that there are 3 connections one of them is the Active with an IP of 192.168.1.45. The attacker can also provide the ComputerName option to get refined results.

```
situational_awareness/network/powerview/get_rdp_session
set ComputerName DC1
execute
```

```
(Empire: UL2MCR1X) > usemodule situational_awareness/network/powerview/get_rdp_session
(Empire: powershell/situational_awareness/network/powerview/get_rdp_session) > set ComputerName DC1
(Empire: powershell/situational_awareness/network/powerview/get_rdp_session) > execute
[*] Tasked UL2MCR1X to run TASK_CMD_JOB
[*] Agent UL2MCR1X tasked with task ID 15
[*] Tasked agent UL2MCR1X to run module powershell/situational_awareness/network/powerview/get_rdp_session
(Empire: powershell/situational_awareness/network/powerview/get_rdp_session) >
Job started: VPDG9U



ComputerName : DC1
SessionName  : Services
UserName     :
ID           : 0
State        : Disconnected
SourceIP     :

ComputerName : DC1
SessionName  : RDP-Tcp#1
UserName     : IGNITE\Administrator
ID           : 1
State        : Active
SourceIP     : 192.168.1.45

ComputerName : DC1
SessionName  : Console
UserName     :
ID           : 3
State        : Connected
SourceIP     :
```

Get Site

Finally, this module enumerates and provides the attacker with a list of all the sites in the current domain. This can help the attacker to get details about the sites and their location. Coupled with other vulnerabilities this kind of information can lead to big attacks.

```
situational_awareness/network/powerview/get_site
execute
```

```
(Empire: VU2BZS9T) > usemodule situational_awareness/network/powerview/get_site
(Empire: powershell/situational_awareness/network/powerview/get_site) > execute
[*] Tasked VU2BZS9T to run TASK_CMD_JOB
[*] Agent VU2BZS9T tasked with task ID 1
[*] Tasked agent VU2BZS9T to run module powershell/situational_awareness/network/powerview/
(Empire: powershell/situational_awareness/network/powerview/get_site) >
Job started: P76EXG



usncreated              : 4113
systemflags             : 1107296256
name                    : Default-First-Site-Name
whenchanged             : 6/29/2020 4:53:59 PM
objectclass             : {top, site}
showinadvancedviewonly  : True
usnchanged              : 4113
dscorepropagationdata   : 1/1/1601 12:00:00 AM
cn                      : Default-First-Site-Name
distinguishedname       : CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=ignite,DC=
whencreated             : 6/29/2020 4:53:59 PM
instancetype            : 4
objectguid              : c400439e-7a75-415f-949d-2bce60af487e
objectcategory          : CN=Site,CN=Schema,CN=Configuration,DC=ignite,DC=local
```

Conclusion

This concludes our second article on Active Directory. It is still a very extensive topic. We provide this detailed resource so that you can enumerate your Active Directory Deployment from Kali and with the help of PowerShell Empire and understand the information that an attacker can extract. If you want a direct PowerShell-based enumeration, check out this **article**.

Author: Pavandeep Singh is a Technical Writer, Researcher, and Penetration Tester. Can be Contacted on **Twitter** and **LinkedIn**