Active Directory Enumeration: PowerView

🌐 **hackingarticles.in**/active-directory-enumeration-powerview

Raj   April 26, 2021

Active Directory Enumeration is a challenge for even some of the seasoned attackers and it is easy to miss some key components and lose the change to elevate that initial foothold that you might receive. In this article, we bring you methods that you can use to enumerate AD using PowerShell.

Table of Contents

Introduction

We have configured an Active Directory Lab that mimics a Real-Life Environment with a bunch of Users, Machines, and Vulnerabilities. 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)**. It soon became an integral toolkit to perform Active Directory Attacks and Enumeration. For this demonstration, we will assume that we have gained the initial foothold. Now we will use PowerShell with PowerView to enumerate the machine and the Domain. In case you run into difficulties running any of the commands depicted use the Official **GitHub** for the Installation Process.

Get-NetUser

In our Active Directory Lab Setup, we created 7 users with different roles and privileges. We can confirm this by Viewing the Active Directory Users and Computers as shown in the image.
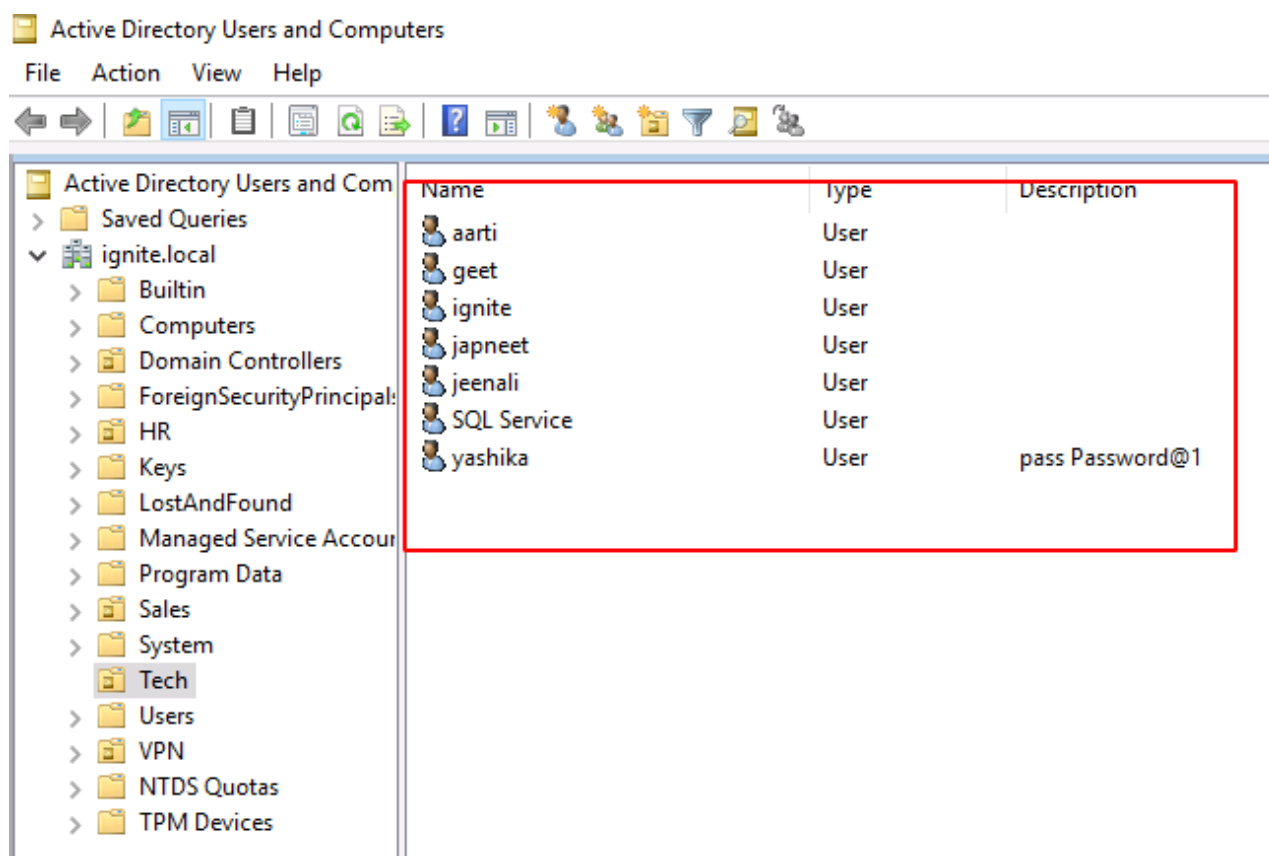
This was to show and co-relate the information that we are about to enumerate using PowerShell. The attacker has transferred the PowerView to the Target System. To run the PowerShell Script on the System, the Execution Policy must be set to Bypass as shown in the image. Next, Importing the Modules from the PowerView Script. This was a one-time process. After this, the attacker can directly use the Modules to perform Enumeration. To get the Users that are active on the Network the attacker ran the following command.

Get-NetUser

```
PS C:\Users\Administrator> cd .\Desktop\
PS C:\Users\Administrator\Desktop> powershell -ep bypass    <---
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator\Desktop> Import-Module .\powerview.ps1    <---
PS C:\Users\Administrator\Desktop> Get-NetUser


logoncount           : 90
badpasswordtime      : 4/7/2021 7:25:25 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       : 805306368
whenchanged          : 4/2/2021 8:34:59 PM
accountexpires       : 9223372036854775807
countrycode          : 0
adspath              : LDAP://CN=Administrator,CN=Users,DC=ignite,DC=local
instancetype         : 4
objectguid           : c00f6d7e-69c7-44cf-ba81-0a513e8aaac4
lastlogon            : 4/11/2021 3:32:09 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 4:54:4
memberof             : {CN=Group Policy Creator Owners,CN=Users,DC=ignite,DC=local
whencreated          : 6/29/2020 4:54:05 PM
iscriticalsystemobject : True
badpwdcount          : 0
cn                   : Administrator
useraccountcontrol   : 66048
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
```

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             : 60
badpasswordtime        : 4/7/2021 7:12:41 AM
description            : pass Password@1
distinguishedname      : CN=yashika,OU=Tech,DC=ignite,DC=local
objectclass            : {top, person, organizationalPerson, user}
displayname            : yashika
lastlogontimestamp     : 4/7/2021 7:12:47 AM
userprincipalname      : yashika@ignite.local
name                   : yashika
objectsid              : S-1-5-21-501555289-2168925624-2051597760-1103
samaccountname         : yashika
admincount             : 1
codepage               : 0
samaccounttype         : 805306368
whenchanged            : 4/10/2021 2:08:59 PM
accountexpires         : 9223372036854775807
countrycode            : 0
adspath                : LDAP://CN=yashika,OU=Tech,DC=ignite,DC=local
instancetype           : 4
objectguid             : d2ff2fb0-5f92-471b-b94c-a1bc5be262f2
lastlogon              : 4/10/2021 7:26:55 AM
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
memberof               : CN=Domain Admins,CN=Users,DC=ignite,DC=local
whencreated            : 6/29/2020 5:08:49 PM
badpwdcount            : 0
cn                     : yashika
useraccountcontrol     : 66048
usncreated             : 16577
primarygroupid         : 513
pwdlastset             : 6/29/2020 10:08:49 AM
usnchanged             : 200768

logoncount             : 1
badpasswordtime        : 12/31/1600 4:00:00 PM
distinguishedname      : CN=geet,OU=Tech,DC=ignite,DC=local
objectclass            : {top, person, organizationalPerson, user}
displayname            : geet
lastlogontimestamp     : 4/7/2021 7:23:57 AM
userprincipalname      : geet@ignite.local
name                   : geet
objectsid              : S-1-5-21-501555289-2168925624-2051597760-1104
samaccountname         : geet
admincount             : 1
codepage               : 0
samaccounttype         : 805306368
whenchanged            : 4/7/2021 2:23:57 PM
accountexpires         : 9223372036854775807
countrycode            : 0
adspath                : LDAP://CN=geet,OU=Tech,DC=ignite,DC=local
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  : {4/7/2021 1:47:03 PM, 1/1/1601 12:00:00 AM}
givenname              : geet
memberof               : CN=Backup Operators,CN=Builtin,DC=ignite,DC=local
lastlogon              : 4/7/2021 7:23:57 AM
badpwdcount            : 0
cn                     : geet
```

Similar Information is available for the users Yashika and Geet.

To get an abstract list of users created on the Network, Grab the Common Name by using the select command on the output of the Get-NetUser Module.

Get-NetUser | select cn

Administrator, Yashika, Geet, Aarti, Raj, Pavan, Jeenali, Japneet, etc. are the various users in this Network Environment.

Similarly to gather information about a particular user. For example, after the attacker extracted users in the previous section, a specific user is chosen to be targeted. Now, more information about a particular user is required. This can be done using a flag -Username with the username that the attacker wants to target. In this case, the attacker chose Yashika User.

Get-NetUser -UserName yashika

```
PS C:\Users\Administrator\Desktop> Get-NetUser -UserName yashika    ←

logoncount            : 60
badpasswordtime       : 4/7/2021 7:12:41 AM
description           : pass Password@1
distinguishedname     : CN=yashika,OU=Tech,DC=ignite,DC=local
objectclass           : {top, person, organizationalPerson, user}
displayname           : yashika
lastlogontimestamp    : 4/7/2021 7:12:47 AM
userprincipalname     : yashika@ignite.local
name                  : yashika
objectsid             : S-1-5-21-501555289-2168925624-2051597760-1103
samaccountname        : yashika
admincount            : 1
codepage              : 0
samaccounttype        : 805306368
whenchanged           : 4/10/2021 2:08:59 PM
accountexpires        : 9223372036854775807
countrycode           : 0
adspath               : LDAP://CN=yashika,OU=Tech,DC=ignite,DC=local
instancetype          : 4
objectguid            : d2ff2fb0-5f92-471b-b94c-a1bc5be262f2
lastlogon             : 4/10/2021 7:26:55 AM
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
memberof              : CN=Domain Admins,CN=Users,DC=ignite,DC=local
whencreated           : 6/29/2020 5:08:49 PM
badpwdcount           : 0
cn                    : yashika
useraccountcontrol    : 66048
usncreated            : 16577
primarygroupid        : 513
pwdlastset            : 6/29/2020 10:08:49 AM
usnchanged            : 200768
```

A streamlined but detailed output regarding the Yashika User is extracted by the attacker.

Get-UserProperty

When working with the Users and their properties, we see that there is a variable by the name pwdlastset. We can use this to check which user is reluctant to change their passwords. This can be configured to any of the property that was extracted in the previous. For this demonstration, we will be extracting the password last set property of all the users.

Get-UserProperty -Properties pwdlastset

```
PS C:\Users\Administrator\Desktop> Get-UserProperty -Properties pwdlastset    ←

name            pwdlastset
----            ----------
Administrator   6/29/2020 9:40:26 AM
Guest           12/31/1600 4:00:00 PM
DefaultAccount  12/31/1600 4:00:00 PM
krbtgt          6/29/2020 9:54:43 AM
yashika         6/29/2020 10:08:49 AM
geet            6/29/2020 10:09:17 AM
aarti           6/29/2020 10:10:52 AM
Raj             7/6/2020 10:33:10 AM
pavan           7/6/2020 12:24:15 PM
SQL Service     4/3/2021 9:17:09 AM
jeenali         4/5/2021 12:31:09 PM
japneet         4/5/2021 12:32:28 PM
ignite          4/9/2021 8:43:37 AM
```
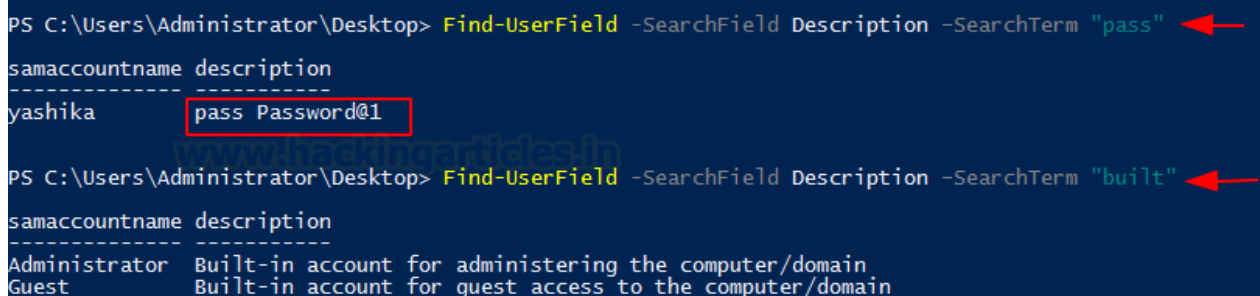
Find-UserField

There are times where there are so many users in the network that it becomes very difficult for the Domain Administrator to keep track of all users or their credentials. This is where they resort to some of the risky techniques to save the credentials information. A good example that I have seen more than ever in the real environment is saving the credentials or important information about the user in their description. This can be extracted by the use of Find-UserField with a search term. In this demonstration, we used the term pass to search for potential passwords. The user Yashika has their password written and saved in their description. This is not limited to this type of information. Lots of different data can also be extracted by using the right set of keywords such as built. This will extract the attacker from the accounts that are Built-in Accounts.

Find-UserField -SearchField Description -SearchTerm "pass"
Find-UserField -SearchField Description -SearchTerm "built"

The information that is extracted using UserField is the information stored at the Properties of that user. While on the Server this can be viewed by opening the list of users and then right-clicking on any particular user. Then choose Properties. This will lead to a window similar to the one shown in the image below. Here, we can see that the Administrator has provided the password in their Description Field. This goes without saying that this should not be done at all. From the attacker's point of view, always check for such descriptions as they will contain some clue that can help you get further.

Invoke-UserHunter

While enumerating the Domain, the attacker that has a targeted approach will be able to extract more data and that faster. The setup at home servers that we practice on doesn't have a time constrain that attackers have to adhere to. In real-life red teaming assessments, if the attacker is taking their sweet time to extract data, they pose a risk to be detected and get thrown out of their initial access or even get captured. This is where some Reconnaissance comes in handy. During the Recon, the attacker can have a list of specific users that take priority to enumerate first and it is possible that those users will help the attacker to elevate access and they won't need to enumerate other users. This reduces the time as well as the noise and logs that will be created when the attacker enumerates users. This is solved using the Invoke-UserHunter. It helps the attacker search or as its name suggests "hunt" for those specific users. It will accept usernames and if the attacker has a handy list of usernames, it will graciously accept it as well. It accepts the domain group and host lists as well. It uses a mix of Get-NetSessions and Get-NetLoggedon against every server and then compares the result against the target user set. Then again it raises the question of the amount of noise it will generate. But giving it a smaller number of usernames in the list or even giving it a single username will help the attacker reduce the noise significantly. It is worth noting that Invoke-UserHunter

will run without any Administrator Privileges. Moving to the demonstration, the attacker runs the Invoke-UserHunter directly without any users or options. It will run against all users that it could find that usually is the Administrator. It can be observed that the information extracted is pretty basic but useful in the case of profiling a user.

Invoke-UserHunter

```
PS C:\Users\Administrator\Desktop> Invoke-UserHunter

UserDomain   : IGNITE
UserName     : Administrator
ComputerName : DC1.ignite.local
IP           : 192.168.1.172
SessionFrom  :
LocalAdmin   :
```

A pretty nifty feature that was interesting enough to add was the CheckAcess function. This feature allows the attacker to check for the Local Administrator Access of that particular user or list of users that they provided. In the demonstration, the attacker tested the Access of the Administrator which without surprise comes to be True.

Invoke-UserHunter -CheckAccess

```
PS C:\Users\Administrator\Desktop> Invoke-UserHunter -CheckAccess

UserDomain   : IGNITE
UserName     : Administrator
ComputerName : DC1.ignite.local
IP           : 192.168.1.172
SessionFrom  :
LocalAdmin   : True
```

Get-NetDomain

When the attacker needs to extract the domain-related information directly from the target server, Get-NetDomain got them covered. It pretty much extracts the Domain data that includes the Forest Name, Domain Controllers with Children (that might be configured in a real environment server). Then there is the Name of the Parents with the RidRoleOwner which is a DC Object that holds the relative identifier (RID) master role and PdcRoleOwner another DC Object that holds the PDC emulator role for that specific Domain.

Get-NetDomain

In case the attacker wanted to go against a specific domain, they can use a domain option by providing the name of the exact domain that they are looking for and Get-NetDomain will extract the data for that particular domain.
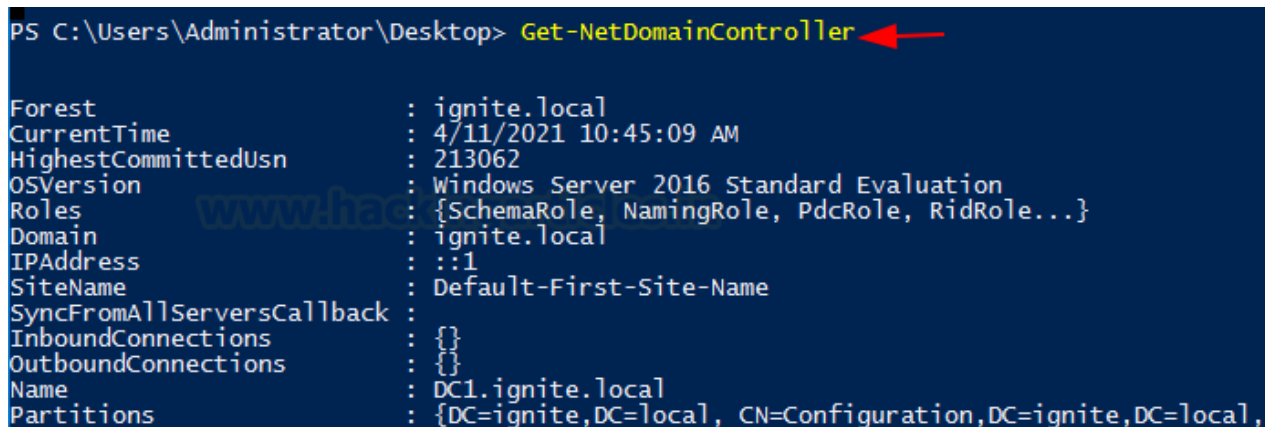
Get-NetDomain -domain "ignite.local"

Get-NetDomainController

Next on the lineup, we have the Get-NetDomainController. 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.

Get-NetDomainController
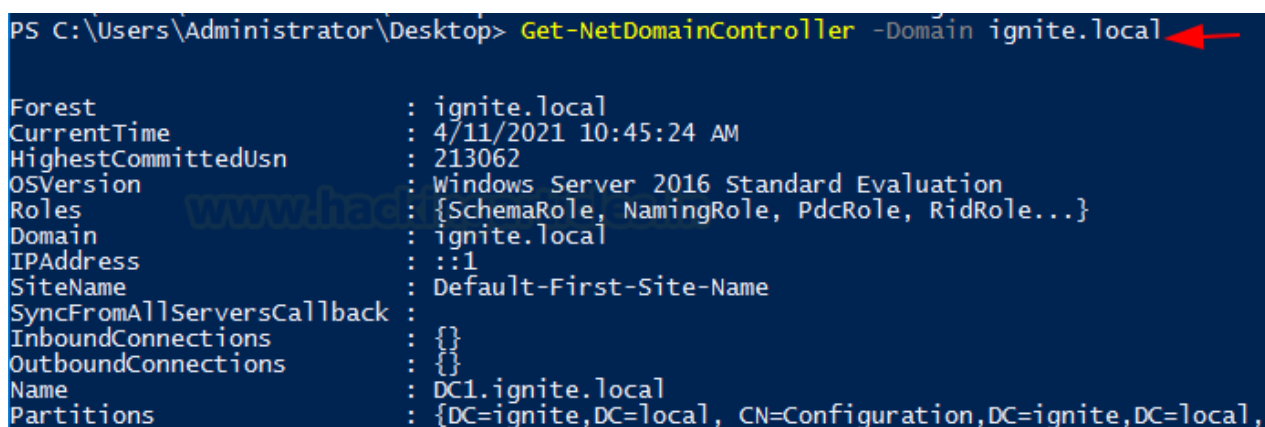
Similar to the Get-NetDomain the attacker can configure Get-NetDomainController to be targeted to a specific domain. The scenario that the attacker might be looking at multiple domains set up with multiple server setup so the attacker can use the -Domain option to target that specific Domain Controller inside the Domain.
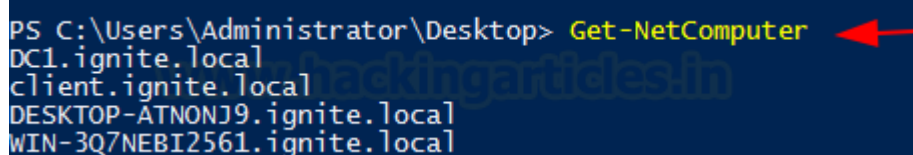
Get-NetDomainController -Domain ignite.local

Get-NetComputer

What seems to be a pretty simple option can turn out to be one of the most used tools to extract a huge amount of data from either the Domain Controller or even a single device. If the attacker runs the Get-NetComputer directly on the Domain Controller machine as demonstrated, it will reveal the Computer Names of all the devices connected in the Domain.

Get-NetComputer

Moving on, if the attacker decides to use -Ping Option then they can get the list of all the devices that can be pinged from the machine they are running the Get-NetComputer from.

Get-NetComputer -Ping

```
PS C:\Users\Administrator\Desktop> Get-NetComputer -Ping
DC1.ignite.local
```

If the attacker doesn't want to extract the data one parameter at a time there is an option to extract all the data from the Machine. This can be done with the FullData option, but keep in mind that a large amount of data extraction leads to large chances of getting detected.

Get-NetComputer -FullData

```
PS C:\Users\Administrator\Desktop> Get-NetComputer -FullData

pwdlastset                     : 4/7/2021 5:30:23 AM
logoncount                     : 147
msds-generationid              : {168, 207, 198, 26...}
serverreferencebl              : CN=DC1,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Conf
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                 : 805306369
whenchanged                    : 4/7/2021 12:30:23 PM
accountexpires                 : 9223372036854775807
countrycode                    : 0
adspath                        : LDAP://CN=DC1,OU=Domain Controllers,DC=ignite,DC=local
instancetype                   : 4
msdfsr-computerreferencebl     : CN=DC1,CN=Topology,CN=Domain System Volume,CN=DFSR-GlobalSett
objectguid                     : de681d91-bd3c-45df-8285-c9ceb8eb7c37
operatingsystem                : Windows Server 2016 Standard Evaluation
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-BF97-47
usncreated                     : 12293
memberof                       : CN=RAS and IAS Servers,CN=Users,DC=ignite,DC=local
lastlogon                      : 4/11/2021 3:31:14 AM
badpwdcount                    : 0
cn                             : DC1
useraccountcontrol             : 532480
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

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
```

Moreover, if the attacker decides to use the -OperatingSystem option with the Get-NetComputer and provide the Name of the OS as a parameter then they can extract all the machines that are running that specific Operating System.

Get-NetComputer -Operatingsystem "Windows Server 2016 Standard Evaluation"

Get-UserProperty

Next on the list is the UserProperty. Up until now, the attacker can extract the users and very little information about them. This was limited but this problem is solved using UserProperty. With it, the attacker can aim to those niche details about any particular property. Some of the information extractable is check for Administrator Level Access, Password Time, Password Change Date, Description of the User, check what group the different users are a part of, and much more.

Get-UserProperty

To target a specific Property, the attacker can use the Properties option and specify the property they want to inquire about. For the demonstration, the property that was inquired here was badpwdcount. This tells the attacker about the unsuccessful attempts that were made against all the users.

Get-UserProperty -Properties badpwdcount

```
PS C:\Users\Administrator\Desktop> Get-UserProperty -Properties badpwdcount
name              badpwdcount
----              -----------
Administrator              0
Guest                      0
DefaultAccount             0
krbtgt                     0
yashika                    0
geet                       0
aarti                      0
Raj                        0
pavan                      2
SQL Service                0
jeenali                    0
japneet                    0
ignite                     0
```

The attacker can focus on the logoncount property to get an understanding as to which of the users are dormant and which among them are active. In a real-life scenario, inactive users might be the users in a network of ex-employees that have been overlooked by the Administrator. This can create a problem as firstly these accounts would not adhere to change their password also the attack mounted on these accounts won't raise flags being these users are legit.

Get-UserProperty -Properties logoncount

```
PS C:\Users\Administrator\Desktop> Get-UserProperty -Properties logoncount
name              logoncount
----              ----------
Administrator             92
Guest                      0
DefaultAccount             0
krbtgt                     0
yashika                   60
geet                       1
aarti                      0
Raj                        0
pavan                      0
SQL Service                0
jeenali                    0
japneet                    0
ignite                     0
```

Get-NetForest

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 Get-NetForest.

Get-NetForest

Get-NetForestCatalog

Forests typically have different global catalogs that can help the attacker to get some precarious information about the domain. This can be observed from the following demonstration of extracting all the global catalogs of the current forest using the Get-NetForestCatalog.
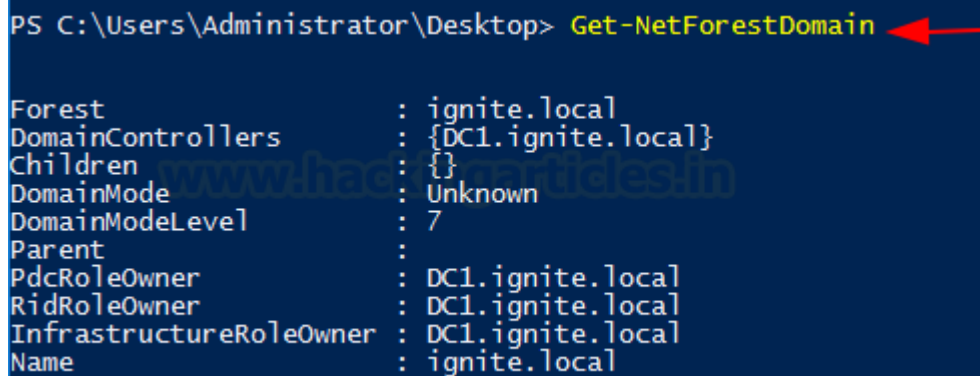
Get-NetForestCatalog

Get-NetForestDomain

Moving on from the catalogs, the attacker can also work on extracting the various domains of the forest the current user is located in. This can be done by running Get-NetForestDomain as shown in the demonstration.

Get-NetForestDomain

```
PS C:\Users\Administrator\Desktop> Get-NetForestDomain

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
```
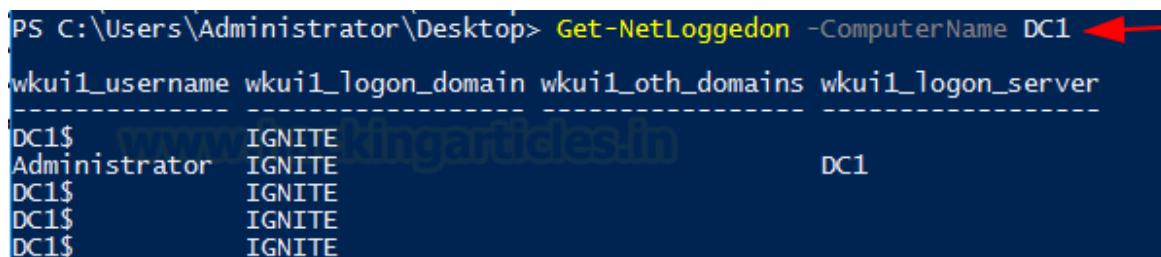
Get-NetLoggedon

That's enough Forest, getting back to the users on the local or remote machine the attacker can take advantage of the NetLoggedon module. It should be noted that Administrative Rights are required to use this module. This module executes NetWkstaUserEnum Win32API call to extract the users that are currently logged on. If the attacker is in a bit of a hurry, they can enumerate all the uses that are logged on for all the machines in the domain by using the Get-DomainComputer and then running the Get-NetLoggedon on that data. This can be concatenated using a pipe.

Get-DomainComputer | Get-NetLoggedon
In this demonstration, however, it is shown how to enumerate users that are loggedon on a particular machine with the help of the ComputerName option and providing the Name.

Get-NetLoggedon -ComputerName DC1

```
PS C:\Users\Administrator\Desktop> Get-NetLoggedon -ComputerName DC1

wkui1_username wkui1_logon_domain wkui1_oth_domains wkui1_logon_server
-------------- ------------------ ----------------- ------------------
DC1$           IGNITE
Administrator  IGNITE                                DC1
DC1$           IGNITE
DC1$           IGNITE
DC1$           IGNITE
```

Get-DomainPolicy

Amongst other information, the Domain Policy of a Domain can also reveal some pretty good information. The attacker can use the Get-Domain to extract 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. To get more focused on a particular domain the Domain option. To extract Domain or Domain Controller using the Source Option or Server option to bind to a particular Active Directory server.

Get-DomainPolicy

To enumerate Kerberos details, the attacker can try and go after the Kerberos Policy which contains data such as the Max Ticket Age, Max Renew Age, and several Ticket Validation Client. This kind of information can come in handy if the attacker is trying to perform a ticket forging attack or similar attack.

(Get-DomainPolicy)."KerberosPolicy"

To extract the data regarding the System Access such as the password data that we extracted earlier like Password Age, Password Complexity and Password Length, etc.

(Get-DomainPolicy)."SystemAccess"

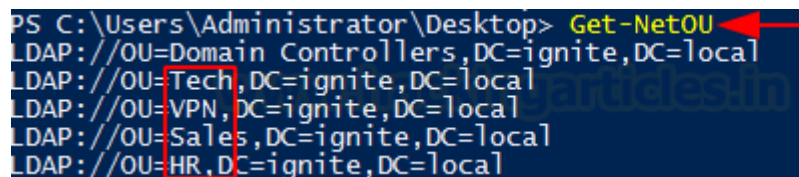## Get-NetOU

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, run the following command on PowerShell.

Get-NetOU

It can be observed that there are 4 OUs on the Target Server. Namely, Tech, VPN, Sales, and HR.

Get-NetGroup

During enumeration that the attacker is trying to perform extracting the Group, information is one of the most important that the attacker can enumerate. To get all the groups in the current domain, the attacker can use the Get-NetGroup as demonstrated.

Get-NetGroup

```
PS C:\Users\Administrator\Desktop> Get-NetGroup    <---
Administrators
Users
Guests
Print Operators
Backup Operators
Replicator
Remote Desktop Users
Network Configuration Operators
Performance Monitor Users
Performance Log Users
Distributed COM Users
IIS_IUSRS
Cryptographic Operators
Event Log Readers
Certificate Service DCOM Access
RDS Remote Access Servers
RDS Endpoint Servers
RDS Management Servers
Hyper-V Administrators
Access Control Assistance Operators
Remote Management Users
System Managed Accounts Group
Storage Replica Administrators
Domain Computers
Domain Controllers
Schema Admins
Enterprise Admins
Cert Publishers
Domain Admins
Domain Users
Domain Guests
Group Policy Creator Owners
RAS and IAS Servers
Server Operators
Account Operators
Pre-Windows 2000 Compatible Access
Incoming Forest Trust Builders
Windows Authorization Access Group
Terminal Server License Servers
Allowed RODC Password Replication Group
Denied RODC Password Replication Group
Read-only Domain Controllers
Enterprise Read-only Domain Controllers
Cloneable Domain Controllers
Protected Users
Key Admins
Enterprise Key Admins
DnsAdmins
DnsUpdateProxy
Finance
```

When the attacker requires to extract the groups that consist of the admin keyword as those might be important or might contain some information regarding the administrator as this would give all kinds of administrator groups as demonstrated.

Get-NetGroup *admin*

```
PS C:\Users\Administrator\Desktop> Get-NetGroup *admin*    <---
Administrators
Hyper-V Administrators
Storage Replica Administrators
Schema Admins
Enterprise Admins
Domain Admins
Key Admins
Enterprise Key Admins
DnsAdmins
```

Suppose the attacker wanted to check for the membership of a particular user then they can use the UserName option. This also can be checked as shown in the image below. The attacker extracted the information for the Yashika User.

Get-NetGroup -UserName yashika

```
PS C:\Users\Administrator\Desktop> Get-NetGroup -UserName yashika
BUILTIN\Administrators
IGNITE\Denied RODC Password Replication Group
IGNITE\Domain Admins
```

To target a specific domain the attacker can use the Domain option with the domain name provided against as shown in the demonstration.

Get-NetGroup -Domain ignite.local

```
PS C:\Users\Administrator\Desktop> Get-NetGroup -Domain ignite.local
Administrators
Users
Guests
Print Operators
Backup Operators
Replicator
Remote Desktop Users
Network Configuration Operators
Performance Monitor Users
Performance Log Users
Distributed COM Users
IIS_IUSRS
Cryptographic Operators
Event Log Readers
Certificate Service DCOM Access
RDS Remote Access Servers
RDS Endpoint Servers
RDS Management Servers
Hyper-V Administrators
Access Control Assistance Operators
Remote Management Users
System Managed Accounts Group
Storage Replica Administrators
Domain Computers
Domain Controllers
Schema Admins
Enterprise Admins
Cert Publishers
Domain Admins
Domain Users
Domain Guests
Group Policy Creator Owners
RAS and IAS Servers
Server Operators
Account Operators
Pre-Windows 2000 Compatible Access
Incoming Forest Trust Builders
Windows Authorization Access Group
Terminal Server License Servers
Allowed RODC Password Replication Group
Denied RODC Password Replication Group
Read-only Domain Controllers
Enterprise Read-only Domain Controllers
Cloneable Domain Controllers
Protected Users
Key Admins
```

Furthermore, if the attacker wanted to extract all the data regarding the groups working on the Domain, they can use the FullData option and extract all the users with their group details. In the demonstration, it can be observed that information enumerated such as there is an Admin in this domain which is a part of Administrator Group and then other User Groups.

Get-NetGroup -FullData

There is a Member name Japneet that is a member of the Tech Group and looking for more information about the user groups, it can be observed that the is a user by the name of geet that is a part of the Tech group as well.

```
grouptype               : -2147483643
admincount              : 1
iscriticalsystemobject  : True
samaccounttype          : 536870912
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
adspath                 : LDAP://CN=Print Operators,CN=Builtin,DC=ignite,DC=local
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 controllers
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               : -2147483643
admincount              : 1
iscriticalsystemobject  : True
samaccounttype          : 536870912
samaccountname          : Backup Operators
whenchanged             : 4/9/2021 5:30:20 PM
objectsid               : S-1-5-32-551
objectclass             : {top, group}
cn                      : Backup Operators
usnchanged              : 192583
systemflags             : -1946157056
name                    : Backup Operators
adspath                 : LDAP://CN=Backup Operators,CN=Builtin,DC=ignite,DC=local
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
distinguishedname       : CN=Backup Operators,CN=Builtin,DC=ignite,DC=local
member                  : {CN=ignite,OU=Tech,DC=ignite,DC=local, CN=geet,OU=Tech,DC=ignit
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 on from the user-based Group Enumeration to the Group Based Enumeration by providing the Group Name as shown in the image below.

Get-NetGroup "Domain Admins"

```
PS C:\Users\Administrator\Desktop> Get-NetGroup "Domain Admins"
Domain Admins
```

The attacker can also use multiple options to target a particular Group and enumerate all data about that group as shown in the demonstration.

Get-NetGroup "Domain Admins" -FullData

There are more possible solutions for the attacker to streamline their enumeration process by providing a bunch of options and parameters to target the exact information. This includes a particular Group Name option and a Domain option.

Get-NetGroup -GroupName *admin* -Domain ignite.local

Get-NetGroupMember

In the enumeration, if the attacker gets to a stage where they have successfully enumerated the group names then they can use that in collaboration with the Get-NetGroupMember to extract the members of that group. In the demonstration, we extracted the members of the group Domain Admins.

Get-NetGroupMember -GroupName "Domain Admins"

As discussed earlier Get-NetGroupMember also supports some options to run along such as the Recurse. It helps the attacker extracts significant amounts of data about all the users of the group they provided. As it can be observed from the screenshots of running Get-NetGroupMember with and without Recurse there is some significant difference between them both.

Get-NetGroupMember -GroupName "Administrators" -Recurse

```
PS C:\Users\Administrator\Desktop> Get-NetGroupMember -GroupName 'Administrators' -Recurse

GroupDomain  : ignite.local
GroupName    : Administrators
MemberDomain : ignite.local
MemberName   : Domain Admins
MemberSid    : S-1-5-21-501555289-2168925624-2051597760-512
IsGroup      : True
MemberDN     : CN=Domain Admins,CN=Users,DC=ignite,DC=local

Cannot index into a null array. :
logonCount              : 64
badPasswordTime         : 4/7/2021 7:12:41 AM
description             : pass Password@1
distinguishedName       : CN=yashika,OU=Tech,DC=ignite,DC=local
objectClass             : {top, person, organizationalPerson, user}
displayName             : yashika
lastLogonTimestamp      : 4/7/2021 7:12:47 AM
userPrincipalName       : yashika@ignite.local
objectSid               : S-1-5-21-501555289-2168925624-2051597760-1103
adminCount              : 1
codePage                : 0
sAMAccountType          : 805306368
countryCode             : 0
whenChanged             : 4/10/2021 2:08:59 PM
instanceType            : 4
objectGUID              : d2ff2fb0-5f92-471b-b94c-a1bc5be262f2
lastLogoff              : 12/31/1600 4:00:00 PM
sAMAccountName          : yashika
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
memberOf                : CN=Domain Admins,CN=Users,DC=ignite,DC=local
lastLogon               : 4/11/2021 4:02:06 AM
badPwdCount             : 0
cn                      : yashika
userAccountControl      : 66048
whenCreated             : 6/29/2020 5:08:49 PM
primaryGroupID          : 513
pwdLastSet              : 6/29/2020 10:08:49 AM
name                    : yashika
GroupDomain             : ignite.local
GroupName               : Domain Admins
MemberDomain            : ignite.local
MemberName              : yashika
MemberSid               : S-1-5-21-501555289-2168925624-2051597760-1103
IsGroup                 : False
MemberDN                : CN=yashika,OU=Tech,DC=ignite,DC=local
```

Get-NetGPO

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

Get-NetGPO

```
PS C:\Users\Administrator\Desktop> Get-NetGPO  ←

usncreated                : 5900
systemflags               : -1946157056
displayname               : Default Domain Policy
gpcmachineextensionnames  : [{35378EAC-683F-11D2-A89A-00C04FBBCFA2}{53D6AB1B-2488-11D1-A28
                            -11D1-A28C-00C04FB94F17}]
whenchanged               : 4/8/2021 1:58:58 PM
objectclass               : {top, container, groupPolicyContainer}
gpcfunctionalityversion   : 2
showinadvancedviewonly    : True
usnchanged                : 163911
dscorepropagationdata     : {6/29/2020 4:54:43 PM, 1/1/1601 12:00:00 AM}
name                      : {31B2F340-016D-11D2-945F-00C04FB984F9}
adspath                   : LDAP://CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,C
flags                     : 0
cn                        : {31B2F340-016D-11D2-945F-00C04FB984F9}
iscriticalsystemobject    : True
gpcfilesyspath            : \\ignite.local\sysvol\ignite.local\Policies\{31B2F340-016D-11D
distinguishedname         : CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=Syste
whencreated               : 6/29/2020 4:54:05 PM
versionnumber             : 7
instancetype              : 4
objectguid                : 4aaf7089-5629-4f93-b6cc-0ecc1c4dba1e
objectcategory            : CN=Group-Policy-Container,CN=Schema,CN=Configuration,DC=ignite

usncreated                : 5903
systemflags               : -1946157056
displayname               : Default Domain Controllers Policy
gpcmachineextensionnames  : [{35378EAC-683F-11D2-A89A-00C04FBBCFA2}{D02B1F72-3407-48AE-BA8
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}
adspath                   : LDAP://CN={6AC1786C-016F-11D2-945F-00C04fB984F9},CN=Policies,C
flags                     : 0
cn                        : {6AC1786C-016F-11D2-945F-00C04fB984F9}
iscriticalsystemobject    : True
gpcfilesyspath            : \\ignite.local\sysvol\ignite.local\Policies\{6AC1786C-016F-11D
distinguishedname         : CN={6AC1786C-016F-11D2-945F-00C04fB984F9},CN=Policies,CN=Syste
whencreated               : 6/29/2020 4:54:05 PM
versionnumber             : 6
instancetype              : 4
objectguid                : f852ef84-af95-4083-ba7c-8eabfa710587
```

As it can be observed from the previous iteration of running the Get-NetGPO, the amount of information is overwhelming. Hence to get a clean and easy-to-understand output selection can be used to get those specific names of the policies.

Get-NetGPO | select displayname


```
PS C:\Users\Administrator\Desktop> Get-NetGPO | select displayname  ←

displayname
-----------
Default Domain Policy
Default Domain Controllers Policy
New Group Policy Object
```

Find-GPOLocation

Getting the GPO location is a good way to map the abilities of a specific user. It takes the username that is provided to it and checks for the permissions for that users. This means that it will return the locations that are accessible for that user. In this demonstration, we

use the Yashika user and we choose the verbose option as well to elaborate the result to get the most out of it.

Find-GPOLocation -UserName yashika -verbose

Invoke-EnumerateLocalAdmin

Invoke-EnumerateLocalAdmin does exactly what the names say. It searched for the Local Administrators for the domain. In our demonstration, we see that we have extracted the Administrator, Enterprise Admins and Domain Admins for our domain ignite. local.

Invoke-EnumerateLocalAdmin

Get-NetProcess

Enumerating the running process is one of the most things that the attacker should. It can tell so much about the target machine. 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.

Get-NetProcess

```
PS C:\Users\Administrator\Desktop> Get-NetProcess    ←

ComputerName : DC1
ProcessName  : System Idle Process
ProcessID    : 0
Domain       :
User         :

ComputerName : DC1
ProcessName  : System
ProcessID    : 4
Domain       :
User         :

ComputerName : DC1
ProcessName  : smss.exe
ProcessID    : 324
Domain       : NT AUTHORITY
User         : SYSTEM

ComputerName : DC1
ProcessName  : csrss.exe
ProcessID    : 452
Domain       : NT AUTHORITY
User         : SYSTEM

ComputerName : DC1
ProcessName  : wininit.exe
ProcessID    : 564
Domain       : NT AUTHORITY
User         : SYSTEM

ComputerName : DC1
ProcessName  : csrss.exe
ProcessID    : 572
Domain       : NT AUTHORITY
User         : SYSTEM

ComputerName : DC1
ProcessName  : winlogon.exe
ProcessID    : 656
Domain       : NT AUTHORITY
User         : SYSTEM
```

Invoke-ShareFinder

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. Hence, to enumerate the shares in a domain use Invoke-ShareFinder.

Invoke-ShareFinder

```
PS C:\Users\Administrator\Desktop> Invoke-ShareFinder    ←
\\DC1.ignite.local\ADMIN$          - Remote Admin
\\DC1.ignite.local\C$       - Default share
\\DC1.ignite.local\Confidential      -
\\DC1.ignite.local\IPC$            - Remote IPC
\\DC1.ignite.local\NETLOGON        - Logon server share
\\DC1.ignite.local\Sales Report      -
\\DC1.ignite.local\SYSVOL          - Logon server share
\\DC1.ignite.local\Users           -
```

Invoke-FileFinder

Searching on the machine that the attacker has an initial foothold is not that difficult task. But to search a specific file across the network in the domain can be done using the Invoke FileFinder. It will search for sensitive files such as the Credentials files and other files that can lead to a serious compromise.

Invoke-FileFinder

```
PS C:\Users\Administrator\Desktop> Invoke-FileFinder ←

FullName     : \\DC1.ignite.local\Users\Administrator
Owner        : NT AUTHORITY\SYSTEM
LastAccessTime : 4/10/2021 8:01:42 AM
LastWriteTime  : 4/10/2021 8:01:42 AM
CreationTime   : 6/29/2020 9:40:36 AM
Length       :

FullName     : \\DC1.ignite.local\Users\Administrator\AppData\Local\Microsoft\Credentials
Owner        : BUILTIN\Administrators
LastAccessTime : 3/6/2021 8:12:12 AM
LastWriteTime  : 3/6/2021 8:12:12 AM
CreationTime   : 6/29/2020 9:40:37 AM
Length       :

FullName     : \\DC1.ignite.local\Users\Administrator\AppData\Local\Microsoft_Corporation\
Owner        : BUILTIN\Administrators
LastAccessTime : 4/11/2021 4:40:14 AM
LastWriteTime  : 4/11/2021 4:40:14 AM
CreationTime   : 6/29/2020 9:41:09 AM
Length       : 152966

FullName     : \\DC1.ignite.local\Users\Administrator\AppData\Local\Packages\windows.immer
Owner        : BUILTIN\Administrators
LastAccessTime : 6/29/2020 9:40:54 AM
LastWriteTime  : 7/16/2016 6:18:57 AM
CreationTime   : 6/29/2020 9:40:54 AM
Length       : 1309
```

Invoke-ACLScanner

ACL or Access Control Lists can be scanned on a domain that will return the weak permissions on the files. Bear in mind that Domain Permission can be a bit challenging to wrap your head around and the permission that you might find using Invoke-ACLScanner can be difficult to exploit. However, this does not mean that any attacker should not check for those. In simpler terms, Invoke-ACLScanner finds the permissions that the users and group have which are possible subject to exploitation. It determines this by separating the default permission and showing the list of permissions that do not default or new defined by the Administrator.

Invoke-ACLScanner -ResolveGUIDs

```
PS C:\Users\Administrator\Desktop> Invoke-ACLScanner -ResolveGUIDs  ←

InheritedObjectType    : All
ObjectDN               : CN=MicrosoftDNS,CN=System,DC=ignite,DC=local
ObjectType             : All
IdentityReference      : IGNITE\DnsAdmins
IsInherited            : False
ActiveDirectoryRights  : CreateChild, DeleteChild, ListChildren, ReadProperty, DeleteTree, ExtendedR
PropagationFlags       : None
ObjectFlags            : None
InheritanceFlags       : ContainerInherit
InheritanceType        : All
AccessControlType      : Allow
ObjectSID              :
IdentitySID            : S-1-5-21-501555289-2168925624-2051597760-1101

InheritedObjectType    : All
ObjectDN               : DC=RootDNSServers,CN=MicrosoftDNS,CN=System,DC=ignite,DC=local
ObjectType             : All
IdentityReference      : IGNITE\DnsAdmins
IsInherited            : True
ActiveDirectoryRights  : CreateChild, DeleteChild, ListChildren, ReadProperty, DeleteTree, ExtendedR
PropagationFlags       : None
ObjectFlags            : None
InheritanceFlags       : ContainerInherit
InheritanceType        : All
AccessControlType      : Allow
ObjectSID              :
IdentitySID            : S-1-5-21-501555289-2168925624-2051597760-1101

InheritedObjectType    : All
ObjectDN               : DC=@,DC=RootDNSServers,CN=MicrosoftDNS,CN=System,DC=ignite,DC=local
ObjectType             : All
IdentityReference      : IGNITE\DnsAdmins
IsInherited            : True
ActiveDirectoryRights  : CreateChild, DeleteChild, ListChildren, ReadProperty, DeleteTree, ExtendedR
PropagationFlags       : None
ObjectFlags            : None
InheritanceFlags       : ContainerInherit
InheritanceType        : All
AccessControlType      : Allow
ObjectSID              :
IdentitySID            : S-1-5-21-501555289-2168925624-2051597760-1101
```

Find-LocalAdminAccess

Find-LocalAdminAccess also is pretty self-defined. It enumerated for machines on the local domain that have the users who have the local administrator access. It checks if the user has local administrator access using Test-AdminAccess. Then it checks for the Credential option. If passed, then it uses Invoke-UserImpersonation to impersonate the specified user before enumeration.

Find-LocalAdminAccess

```
PS C:\Users\Administrator\Desktop> Find-LocalAdminAccess  ←
DC1.ignite.local
```

Get-NetSession

At last, it's time to shine some light on the Sessions that are generated inside a Domain. This can be enumerated with the help of the Get-NetSession tool. Upon running this 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. It can be used bare as demonstrated or it can be used with a ComputerName Option to target a specific host.

Get-NetSession

```
PS C:\Users\Administrator\Desktop> Get-NetSession  ⬅—

sesi10_cname sesi10_username sesi10_time sesi10_idle_time
------------ --------------- ----------- ----------------
\\[::1]      Administrator             0                0
```

Conclusion

Active Directory is extensive and can be confusing for novice security professionals. We provide this detailed resource so that you can enumerate your Active Directory Deployment and understand the information that an attacker can extract. It will also help our Blue Teamers to understand how this kind of information can be extracted and what kind of alerts they need to set up to restrict the attacker.