Enumerating Active Directory

- Credential Injection

"If you know the enemy and know yourself, you need not fear the results of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer defeat." - Sun Tzu, Art of War.

if you genuinely want to do in-depth enumeration and even exploitation, you need to understand and mimic your enemy. Thus, you need a Windows machine. This will allow us to use several built-in methods to stage our enumeration and exploits. In this network, we will explore one of these built-in tools, called the `runas.exe` binary.

If we have the AD credentials in the format of :, we can use Runas, a legitimate Windows binary, to inject the credentials into memory. The usual Runas command would look something like this:

```
runas.exe /netonly /user:<domain>\<username> cmd.exe
```

- **/netonly** - Since we are not domain-joined, we want to load the credentials for network authentication but not authenticate against a domain controller. So commands executed locally on the computer will run in the context of your standard Windows account, but any network connections will occur using the account specified here.
- **/user** - Here, we provide the details of the domain and the username. It is always a safe bet to use the Fully Qualified Domain Name (FQDN) instead of just the NetBIOS name of the domain since this will help with resolution.
- **cmd.exe** - This is the program we want to execute once the credentials are injected. This can be changed to anything, but the safest bet is cmd.exe since you can then use that to launch whatever you want, with the credentials injected.

Once you run this command, you will be prompted to supply a password. Note that since we added the /netonly parameter, the credentials will not be verified directly by a domain controller so that it will accept any password. We still need to confirm that the network credentials are loaded successfully and correctly.

Note: If you use your own Windows machine, you should make sure that you run your first Command Prompt as Administrator. This will inject an Administrator token into CMD. If you run tools that require local Administrative privileges from your Runas spawned CMD, the token will already be available. This does not give you administrative privileges on the network, but will ensure that any local commands you execute, will execute with administrative privileges.

After providing the password, a new command prompt window will open. Now we still need to verify that our credentials are working. The most surefire way to do this is to list **SYSVOL**. Any AD account, no matter how low-privileged, can read the contents of the **SYSVOL** directory.

SYSVOL is a folder that exists on all domain controllers. It is a shared folder storing the Group Policy Objects (GPOs) and information along with any other domain related scripts. It is an essential component for Active Directory since it delivers these GPOs to all computers on the domain. Domain-joined computers can then read these GPOs and apply the applicable ones, making domain-wide configuration changes from a central location.

we need to configure our DNS. Sometimes you are lucky, and internal DNS will be configured for you automatically through DHCP or the VPN connection, but not always (like this TryHackMe network). It is good to understand how to do it manually. Your safest bet for a DNS server is usually a domain controller. Using the IP of the domain controller, we can execute the following commands in a PowerShell window:

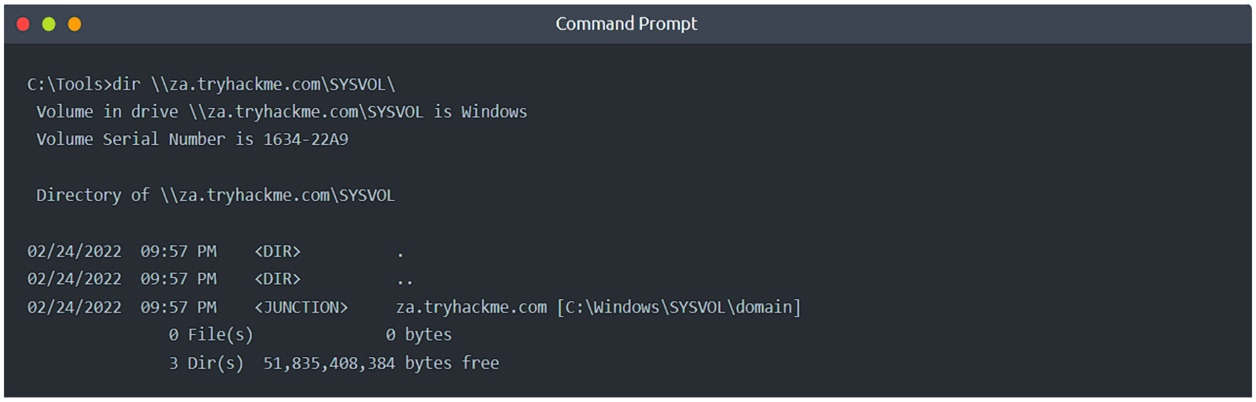
```
$dnsip = "<DC IP>"

$index = Get-NetAdapter -Name 'Ethernet' | Select-Object -
ExpandProperty 'ifIndex'

Set-DnsClientServerAddress -InterfaceIndex $index -ServerAddresses
$dnsip
```

Then try to test the connection by this command,

```
nslookup za.tryhackme.com
```

we can finally test our credentials. We can use the following command to force a network-based listing of the SYSVOL directory

```
dir \\za.tryhackme.com\SYSVOL\
```



```
C:\Tools>dir \\za.tryhackme.com\SYSVOL\
 Volume in drive \\za.tryhackme.com\SYSVOL is Windows
 Volume Serial Number is 1634-22A9

 Directory of \\za.tryhackme.com\SYSVOL

02/24/2022  09:57 PM    <DIR>          .
02/24/2022  09:57 PM    <DIR>          ..
02/24/2022  09:57 PM    <JUNCTION>     za.tryhackme.com [C:\Windows\SYSVOL\domain]
               0 File(s)              0 bytes
               3 Dir(s)  51,835,408,384 bytes free
```

IP vs Hostnames

There is quite a difference, and it boils down to the authentication method being used. When we provide the hostname, network authentication will attempt first to perform Kerberos authentication. Since Kerberos authentication uses hostnames embedded in the tickets, if we provide the IP instead, we can force the authentication type to be NTLM.

- **Microsoft Management Console**

the only method that makes use of a GUI.

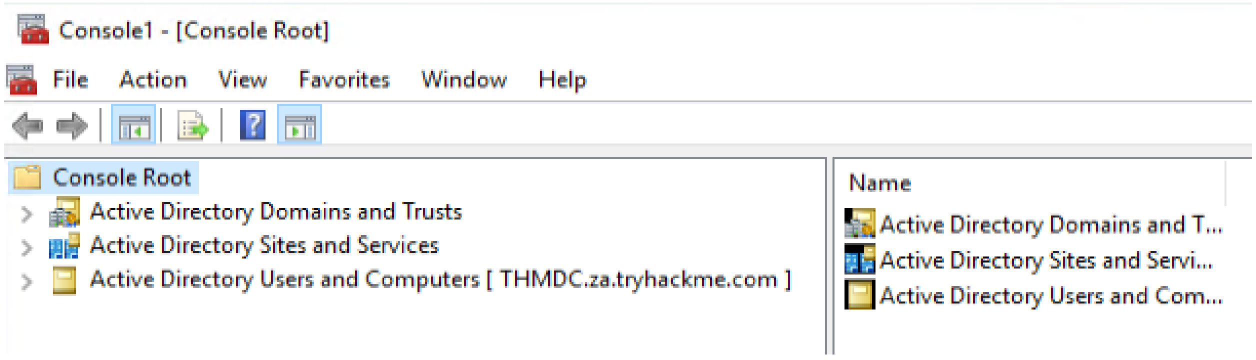
if you are using your own Windows machine, you can perform the following steps to install the Snap-Ins:

1. Press **Start**
2. Search **"Apps & Features"** and press enter
3. Click **Manage Optional Features**
4. Click **Add a feature**
5. Search for **"RSAT"**
6. Select "**RSAT: Active Directory Domain Services and Lightweight Directory Tools"** and click **Install**

we can start MMC, which will ensure that all MMC network connections will use our injected AD credentials.

In MMC, we can now attach the AD RSAT Snap-In:

1. Click **File** -> **Add/Remove Snap-in**
2. Select and **Add** all three Active Directory Snap-ins
3. Click through any errors and warnings
4. Right-click on **Active Directory Domains and Trusts** and select **Change Forest**
5. Enter *za.tryhackme.com* as the **Root domain** and Click **OK**
6. Right-click on **Active Directory Sites and Services** and select **Change Forest**
7. Enter *za.tryhackme.com* as the **Root domain** and Click OK
8. Right-click on **Active Directory Users and Computers** and select **Change Domain**
9. Enter *za.tryhackme.com* as the **Domain** and Click **OK**
10. Right-click on **Active Directory Users and Computers** in the left-hand pane
11. Click on **View** -> **Advanced Features**



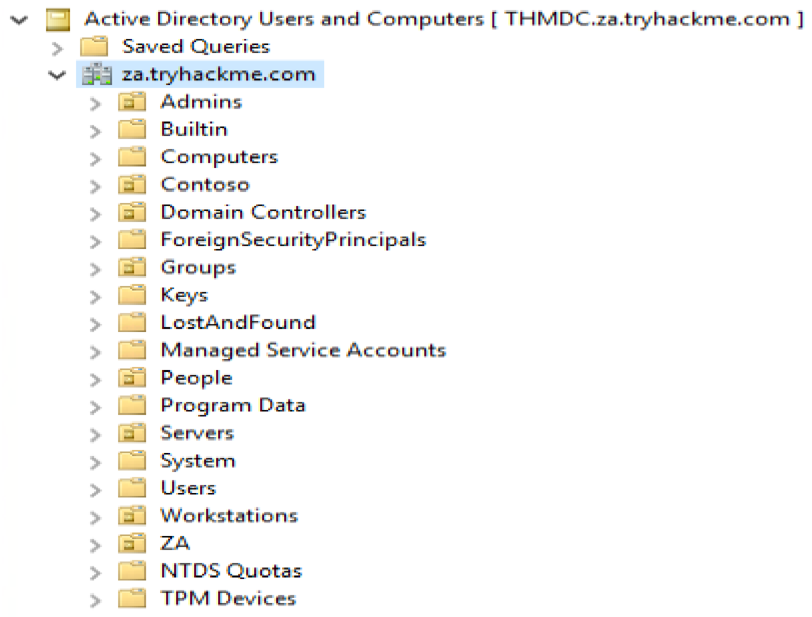
Active Directory Domains and Trusts
Active Directory Sites and Services
Active Directory Users and Computers [THMDC.za.tryhackme.com]

Active Directory Domains and T...
Active Directory Sites and Servi...
Active Directory Users and Com...

We can now start enumerating information about the AD structure here.

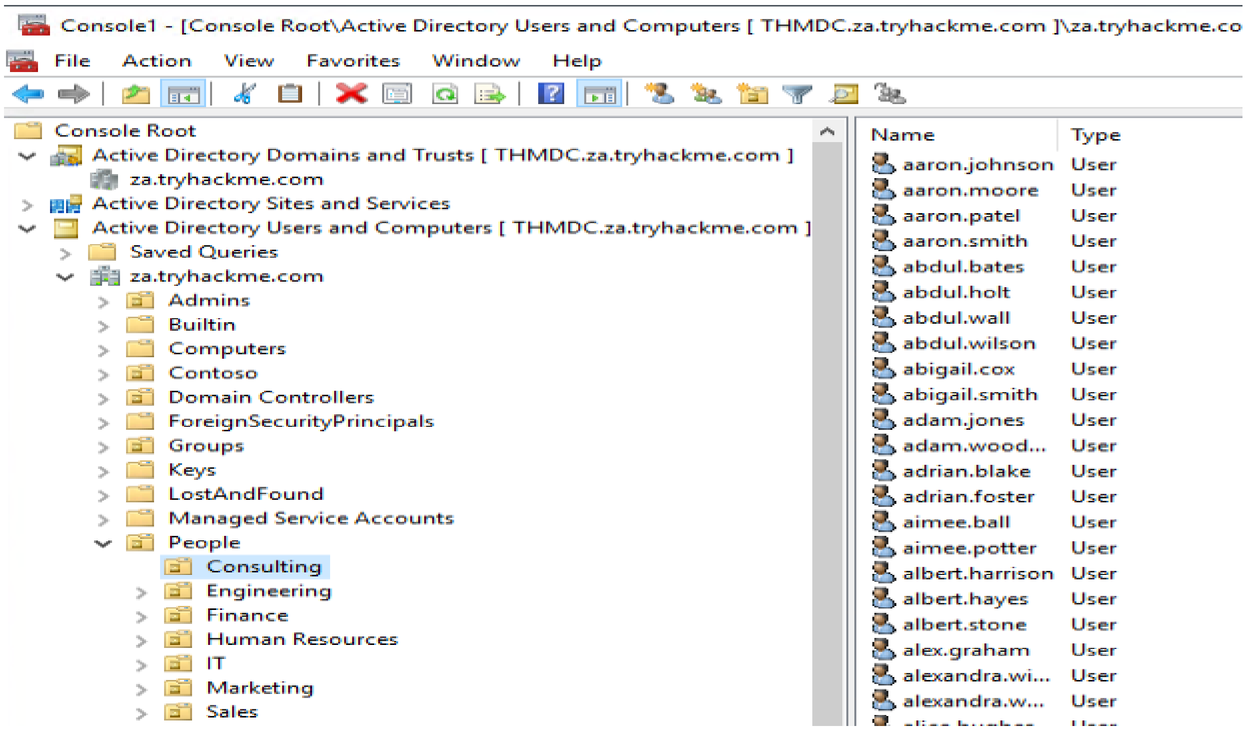
Users and Computers

we will focus on AD Users and Computers. Expand that snap-in and expand the za domain to see the initial Organisational Unit (OU) structure



```
Active Directory Users and Computers [ THMDC.za.tryhackme.com ]
  Saved Queries
  za.tryhackme.com
    Admins
    Builtin
    Computers
    Contoso
    Domain Controllers
    ForeignSecurityPrincipals
    Groups
    Keys
    LostAndFound
    Managed Service Accounts
    People
    Program Data
    Servers
    System
    Users
    Workstations
    ZA
    NTDS Quotas
    TPM Devices
```

Let's take a look at the People directory. Here we see that the users are divided according to department OUs. Clicking on each of these OUs will show the users that belong to that department.

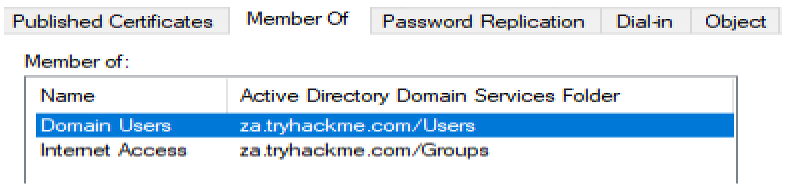


Console1 - [Console Root\Active Directory Users and Computers [THMDC.za.tryhackme.com]\za.tryhackme.co

```
Console Root
  Active Directory Domains and Trusts [ THMDC.za.tryhackme.com ]
    za.tryhackme.com
  Active Directory Sites and Services
  Active Directory Users and Computers [ THMDC.za.tryhackme.com ]
    Saved Queries
    za.tryhackme.com
      Admins
      Builtin
      Computers
      Contoso
      Domain Controllers
      ForeignSecurityPrincipals
      Groups
      Keys
      LostAndFound
      Managed Service Accounts
      People
        Consulting
        Engineering
        Finance
        Human Resources
        IT
        Marketing
        Sales
```

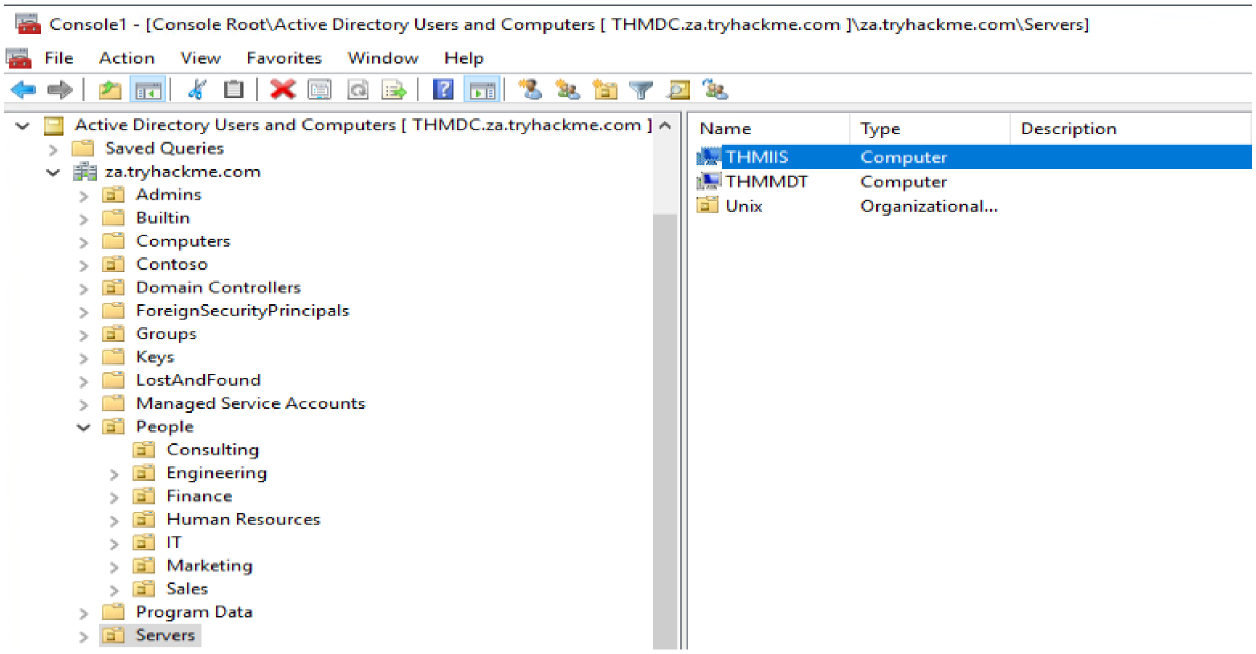
Name	Type
aaron.johnson	User
aaron.moore	User
aaron.patel	User
aaron.smith	User
abdul.bates	User
abdul.holt	User
abdul.wall	User
abdul.wilson	User
abigail.cox	User
abigail.smith	User
adam.jones	User
adam.wood...	User
adrian.blake	User
adrian.foster	User
aimee.ball	User
aimee.potter	User
albert.harrison	User
albert.hayes	User
albert.stone	User
alex.graham	User
alexandra.wi...	User
alexandra.w...	User

Clicking on any of these users will allow us to review all of their properties and attributes.



Name	Active Directory Domain Services Folder
Domain Users	za.tryhackme.com/Users
Internet Access	za.tryhackme.com/Groups

We can also use MMC to find hosts in the environment. If we click on either Servers or Workstations, the list of domain-joined machines will be displayed.



Console1 - [Console Root\Active Directory Users and Computers [THMDC.za.tryhackme.com]\za.tryhackme.com\Servers]

```
Active Directory Users and Computers [ THMDC.za.tryhackme.com ]
  Saved Queries
  za.tryhackme.com
    Admins
    Builtin
    Computers
    Contoso
    Domain Controllers
    ForeignSecurityPrincipals
    Groups
    Keys
    LostAndFound
    Managed Service Accounts
    People
      Consulting
      Engineering
      Finance
      Human Resources
      IT
      Marketing
      Sales
    Program Data
    Servers
```

Name	Type	Description
THMIIS	Computer	
THMMDT	Computer	
Unix	Organizational...	

Benefits

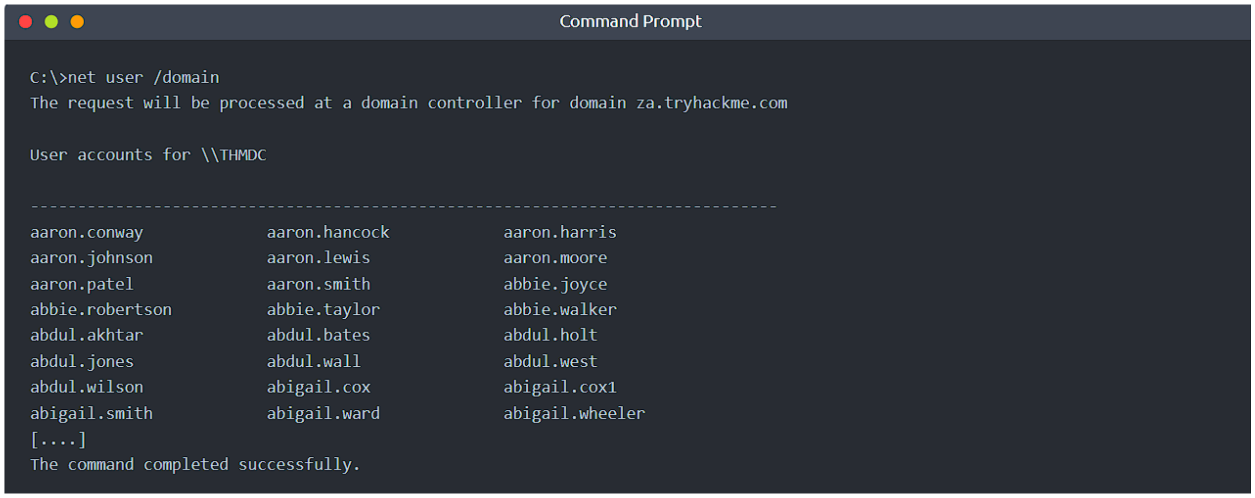
- The GUI provides an excellent method to gain a holistic view of the AD environment.
- Rapid searching of different AD objects can be performed.
- It provides a direct method to view specific updates of AD objects.
- If we have sufficient privileges, we can directly update existing AD objects or add new ones.

- **Command Prompt**

There are times when you just need to perform a quick and dirty AD lookup, and Command Prompt has your back. Good ol' reliable CMD is handy when you perhaps don't have RDP access to a system, defenders are monitoring for PowerShell use, and you need to perform your AD Enumeration through a Remote Access Trojan (RAT). It can even be helpful to embed a couple of simple AD enumeration commands in your phishing payload to help you gain the vital information that can help you stage the final attack.

We can use the `net` command to list all users in the AD domain by using the `user` sub-option

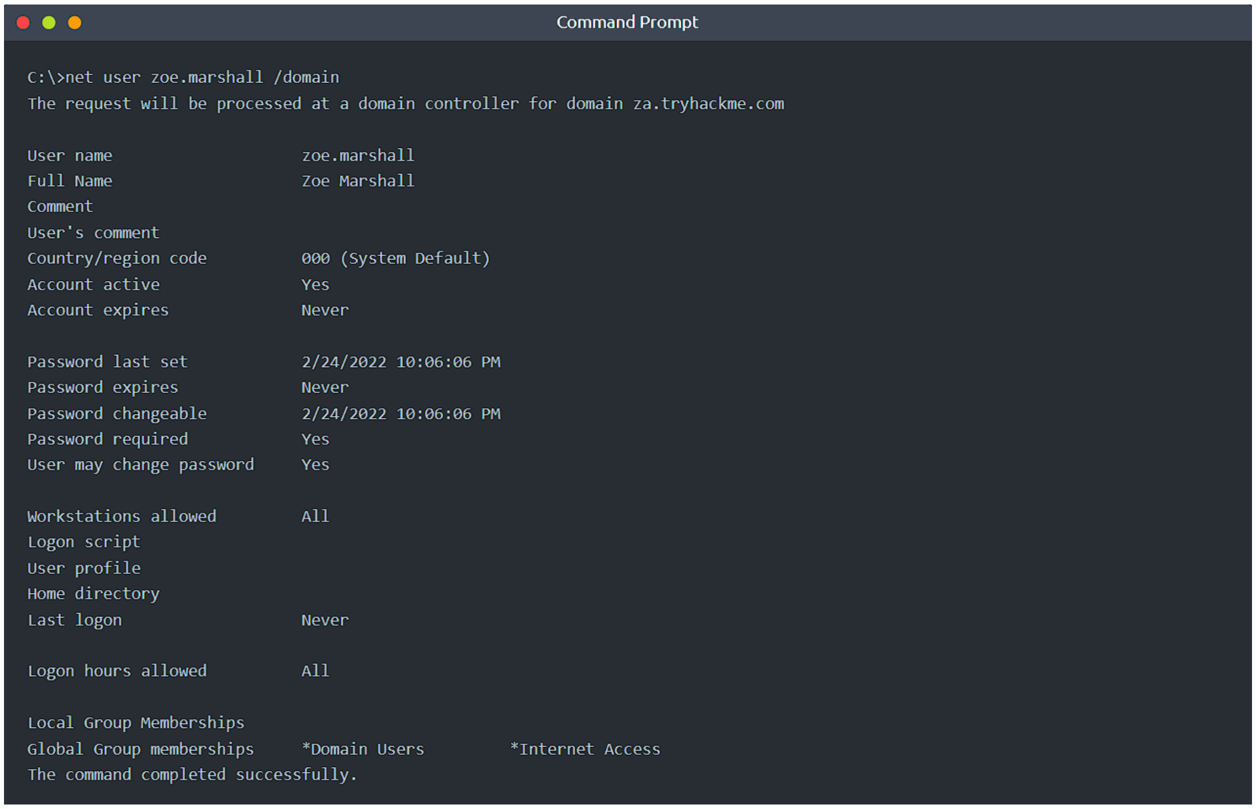
`net user /domain`



```
C:\>net user /domain
The request will be processed at a domain controller for domain za.tryhackme.com

User accounts for \\THMDC

-------------------------------------------------------------------------------
aaron.conway             aaron.hancock            aaron.harris
aaron.johnson            aaron.lewis              aaron.moore
aaron.patel              aaron.smith              abbie.joyce
abbie.robertson          abbie.taylor             abbie.walker
abdul.akhtar             abdul.bates              abdul.holt
abdul.jones              abdul.wall               abdul.west
abdul.wilson             abigail.cox              abigail.cox1
abigail.smith            abigail.ward             abigail.wheeler
[....]
The command completed successfully.
```

We can also use this sub-option to enumerate more detailed information about a single user account

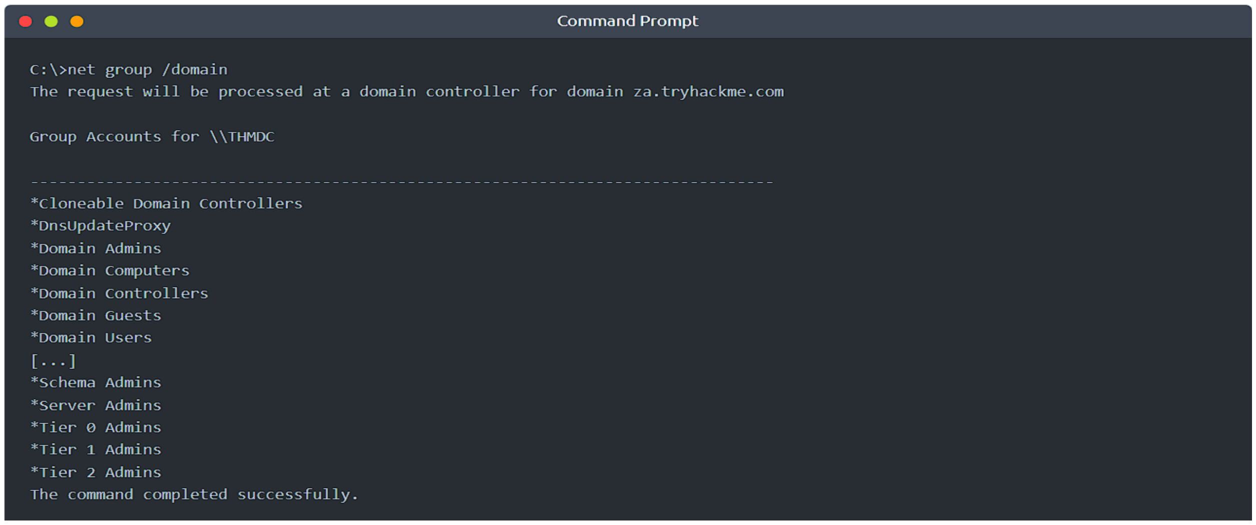
`net user zoe.marshall /domain`



```
C:\>net user zoe.marshall /domain
The request will be processed at a domain controller for domain za.tryhackme.com

User name                    zoe.marshall
Full Name                    Zoe Marshall
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            2/24/2022 10:06:06 PM
Password expires             Never
Password changeable          2/24/2022 10:06:06 PM
Password required            Yes
User may change password     Yes

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

Logon hours allowed          All

Local Group Memberships
Global Group memberships     *Domain Users         *Internet Access
The command completed successfully.
```

We can use the `net` command to enumerate the groups of the domain by using the `group` sub-option

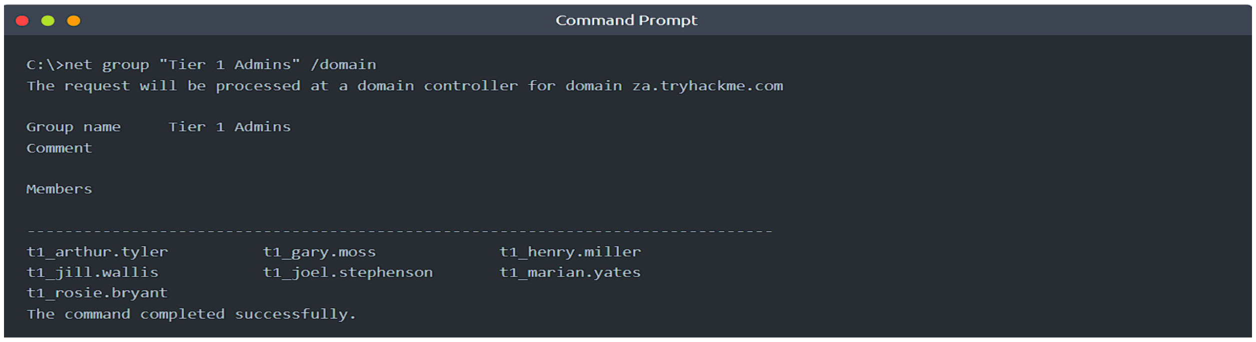
`net group /domain`



```
C:\>net group /domain
The request will be processed at a domain controller for domain za.tryhackme.com

Group Accounts for \\THMDC

-------------------------------------------------------------------------------
*Cloneable Domain Controllers
*DnsUpdateProxy
*Domain Admins
*Domain Computers
*Domain Controllers
*Domain Guests
*Domain Users
[...]
*Schema Admins
*Server Admins
*Tier 0 Admins
*Tier 1 Admins
*Tier 2 Admins
The command completed successfully.
```

We could also enumerate more details such as membership to a group by specifying the group in the same command

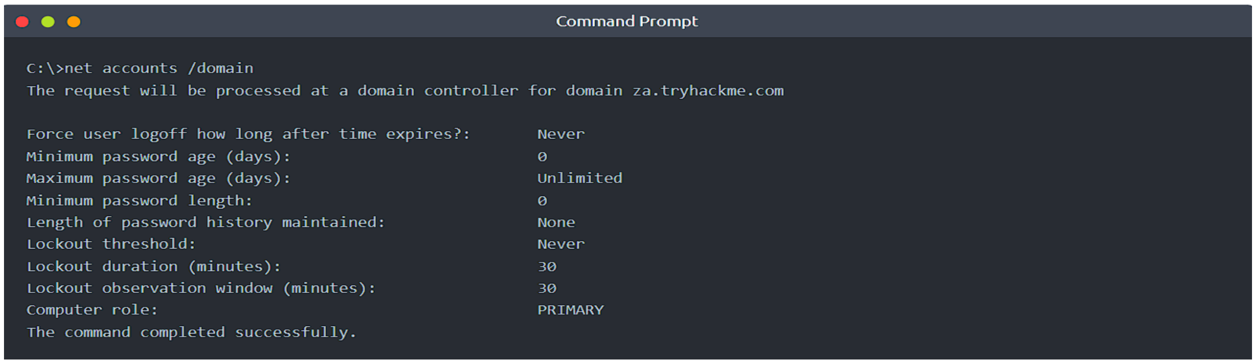
`net group "Tier 1 Admins" /domain`



```
C:\>net group "Tier 1 Admins" /domain
The request will be processed at a domain controller for domain za.tryhackme.com

Group name     Tier 1 Admins
Comment

Members

-------------------------------------------------------------------------------
t1_arthur.tyler          t1_gary.moss             t1_henry.miller
t1_jill.wallis           t1_joel.stephenson       t1_marian.yates
t1_rosie.bryant
The command completed successfully.
```

We can use the `net` command to enumerate the password policy of the domain by using the `accounts` sub-option

`net accounts /domain`



```
C:\>net accounts /domain
The request will be processed at a domain controller for domain za.tryhackme.com

Force user logoff how long after time expires?:       Never
Minimum password age (days):                          0
Maximum password age (days):                          Unlimited
Minimum password length:                              0
Length of password history maintained:                None
Lockout threshold:                                    Never
Lockout duration (minutes):                           30
Lockout observation window (minutes):                 30
Computer role:                                        PRIMARY
The command completed successfully.
```

Benefits

- No additional or external tooling is required, and these simple commands are often not monitored for by the Blue team.
- We do not need a GUI to do this enumeration.
- VBScript and other macro languages that are often used for phishing payloads support these commands natively so they can be used to enumerate initial information regarding the AD domain before more specific payloads are crafted.

Drawbacks

- The `net` commands must be executed from a domain-joined machine. If the machine is not domain-joined, it will default to the WORKGROUP domain.
- The `net` commands may not show all information. For example, if a user is a member of more than ten groups, not all of these groups will be shown in the output.

- **PowerShell**

PowerShell is the upgrade of Command Prompt. Microsoft first released it in 2006. While PowerShell has all the standard functionality Command Prompt provides, it also provides access to cmdlets (pronounced command-lets), which are .NET classes to perform specific functions. While we can write our own cmdlets, like the creators of PowerView did, we can already get very far using the built-in ones.

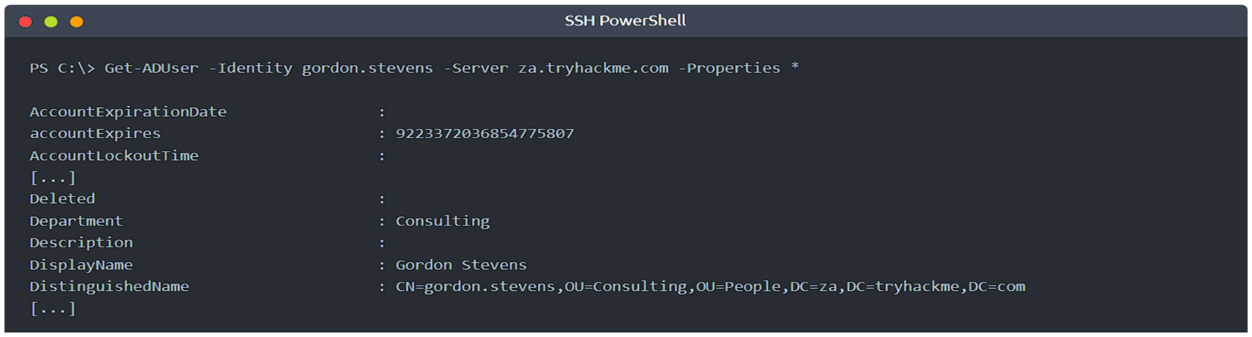
Using our SSH terminal, we can upgrade it to a PowerShell terminal using the following command: `powershell`

We can use the `Get-ADUser` cmdlet to enumerate AD users

```
Get-ADUser -Identity username -Server za.tryhackme.com -Properties *
```

The parameters are used for the following:

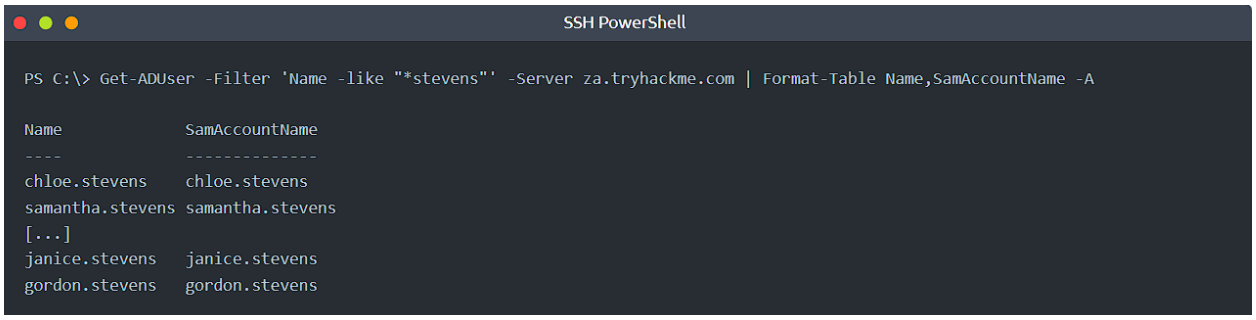
- -Identity - The account name that we are enumerating
- -Properties - Which properties associated with the account will be shown, * will show all properties
- -Server - Since we are not domain-joined, we have to use this parameter to point it to our domain controller



```
PS C:\> Get-ADUser -Identity gordon.stevens -Server za.tryhackme.com -Properties *

AccountExpirationDate                  :
accountExpires                         : 9223372036854775807
AccountLockoutTime                     :
[...]
Deleted                                :
Department                             : Consulting
Description                            :
DisplayName                            : Gordon Stevens
DistinguishedName                      : CN=gordon.stevens,OU=Consulting,OU=People,DC=za,DC=tryhackme,DC=com
[...]
```

we can also use the `-Filter` parameter that allows more control over enumeration and use the `Format-Table` cmdlet to display the results such as the following neatly

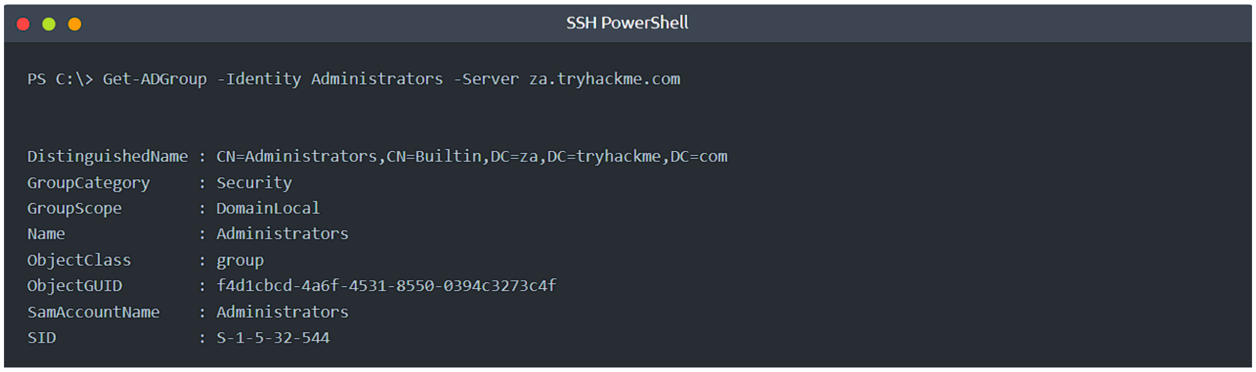
```
Get-ADUser -Filter 'Name -like "*stevens"' -Server za.tryhackme.com | Format-Table Name,SamAccountName -A
```



```
PS C:\> Get-ADUser -Filter 'Name -like "*stevens"' -Server za.tryhackme.com | Format-Table Name,SamAccountName -A

Name             SamAccountName
----             --------------
chloe.stevens    chloe.stevens
samantha.stevens samantha.stevens
[...]
janice.stevens   janice.stevens
gordon.stevens   gordon.stevens
```

We can use the `Get-ADGroup` cmdlet to enumerate AD groups

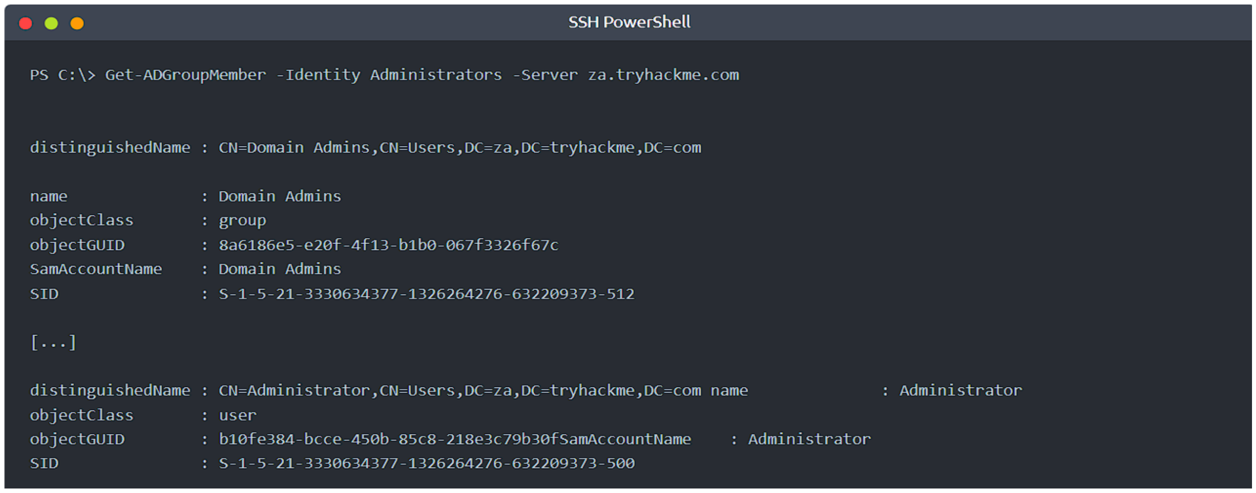
```
Get-ADGroup -Identity Administrators -Server za.tryhackme.com
```



```
PS C:\> Get-ADGroup -Identity Administrators -Server za.tryhackme.com


DistinguishedName : CN=Administrators,CN=Builtin,DC=za,DC=tryhackme,DC=com
GroupCategory     : Security
GroupScope        : DomainLocal
Name              : Administrators
ObjectClass       : group
ObjectGUID        : f4d1cbcd-4a6f-4531-8550-0394c3273c4f
SamAccountName    : Administrators
SID               : S-1-5-32-544
```

We can also enumerate group membership using the `Get-ADGroupMember` cmdlet

```
Get-ADGroupMember -Identity Administrators -Server za.tryhackme.com
```

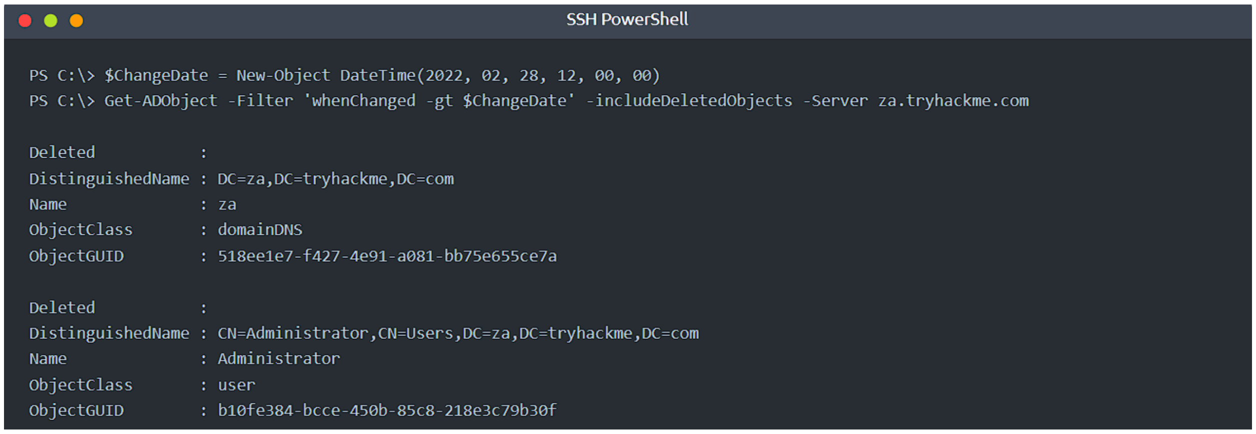


```
PS C:\> Get-ADGroupMember -Identity Administrators -Server za.tryhackme.com


distinguishedName : CN=Domain Admins,CN=Users,DC=za,DC=tryhackme,DC=com

name              : Domain Admins
objectClass       : group
objectGUID        : 8a6186e5-e20f-4f13-b1b0-067f3326f67c
SamAccountName    : Domain Admins
SID               : S-1-5-21-3330634377-1326264276-632209373-512

[...]

distinguishedName : CN=Administrator,CN=Users,DC=za,DC=tryhackme,DC=com name              : Administrator
objectClass       : user
objectGUID        : b10fe384-bcce-450b-85c8-218e3c79b30fSamAccountName    : Administrator
SID               : S-1-5-21-3330634377-1326264276-632209373-500
```

A more generic search for any AD objects can be performed using the `Get-ADObject` cmdlet. For example, if we are looking for all AD objects that were changed after a specific date

`$ChangeDate = New-Object DateTime(2022, 02, 28, 12, 00, 00)`

`Get-ADObject -Filter 'whenChanged -gt $ChangeDate' -includeDeletedObjects -Server za.tryhackme.com`



```
PS C:\> $ChangeDate = New-Object DateTime(2022, 02, 28, 12, 00, 00)
PS C:\> Get-ADObject -Filter 'whenChanged -gt $ChangeDate' -includeDeletedObjects -Server za.tryhackme.com

Deleted           :
DistinguishedName : DC=za,DC=tryhackme,DC=com
Name              : za
ObjectClass       : domainDNS
ObjectGUID        : 518ee1e7-f427-4e91-a081-bb75e655ce7a

Deleted           :
DistinguishedName : CN=Administrator,CN=Users,DC=za,DC=tryhackme,DC=com
Name              : Administrator
ObjectClass       : user
ObjectGUID        : b10fe384-bcce-450b-85c8-218e3c79b30f
```

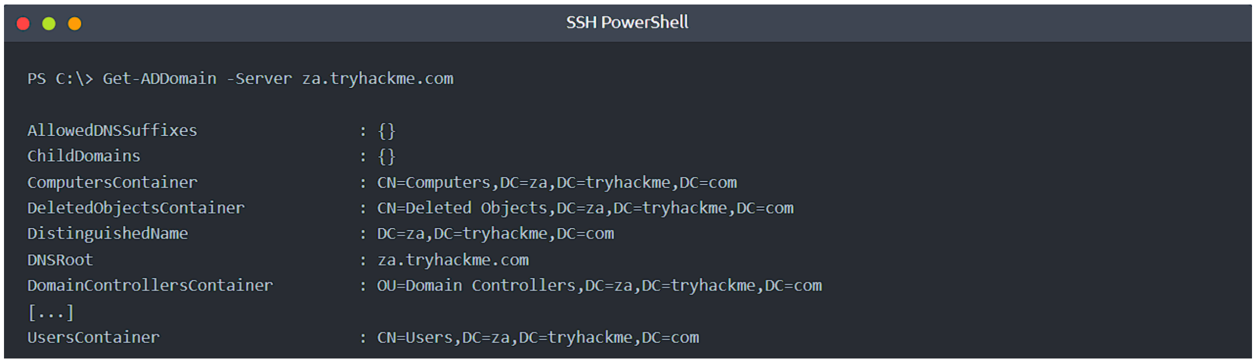
If we wanted to, for example, perform a password spraying attack without locking out accounts, we can use this to enumerate accounts that have a badPwdCount that is greater than 0, to avoid these accounts in our attack

`Get-ADObject -Filter 'badPwdCount -gt 0' -Server za.tryhackme.com`

This will only show results if one of the users in the network mistyped their password a couple of times.

We can use `Get-ADDomain` to retrieve additional information about the specific domain

`Get-ADDomain -Server za.tryhackme.com`



```
PS C:\> Get-ADDomain -Server za.tryhackme.com

AllowedDNSSuffixes                 : {}
ChildDomains                       : {}
ComputersContainer                 : CN=Computers,DC=za,DC=tryhackme,DC=com
DeletedObjectsContainer            : CN=Deleted Objects,DC=za,DC=tryhackme,DC=com
DistinguishedName                  : DC=za,DC=tryhackme,DC=com
DNSRoot                            : za.tryhackme.com
DomainControllersContainer         : OU=Domain Controllers,DC=za,DC=tryhackme,DC=com
[...]
UsersContainer                     : CN=Users,DC=za,DC=tryhackme,DC=com
```

The great thing about the AD-RSAT cmdlets is that some even allow you to create new or alter existing AD objects. However, our focus for this network is on enumeration. Creating new objects or altering existing ones would be considered AD exploitation, which is covered later in the AD module.

However, we will show an example of this by force changing the password of our AD user by using the `Set-ADAccountPassword` cmdlet

```
Set-ADAccountPassword -Identity username -Server za.tryhackme.com -OldPassword (ConvertTo-SecureString -AsPlaintext "old" -force) -NewPassword (ConvertTo-SecureString -AsPlainText "new" -Force)
```

Benefits

- The PowerShell cmdlets can enumerate significantly more information than the net commands from Command Prompt.
- We can specify the server and domain to execute these commands using runas from a non-domain-joined machine.
- We can create our own cmdlets to enumerate specific information.
- We can use the AD-RSAT cmdlets to directly change AD objects, such as resetting passwords or adding a user to a specific group.

Drawbacks

- PowerShell is often monitored more by the blue teams than Command Prompt.
- We have to install the AD-RSAT tooling or use other, potentially detectable, scripts for PowerShell enumeration.

- **BloodHound**

"Defenders think in lists, Attackers think in graphs."

Bloodhound allowed attackers (and by now defenders too) to visualise the AD environment in a graph format with interconnected nodes. Each connection is a possible path that could be exploited to reach a goal. In contrast, the defenders used lists, like a list of Domain Admins or a list of all the hosts in the environment.

Sharphound

You will often hear users refer to Sharphound and Bloodhound interchangeably. However, they are not the same. Sharphound is the enumeration tool of Bloodhound. It is used to enumerate the AD information that can then be visually displayed in Bloodhound. Bloodhound is the actual GUI used to display the AD attack graphs.

Note: Your Bloodhound and Sharphound versions must match for the best results. Usually there are updates made to Bloodhound which means old Sharphound results cannot be ingested.

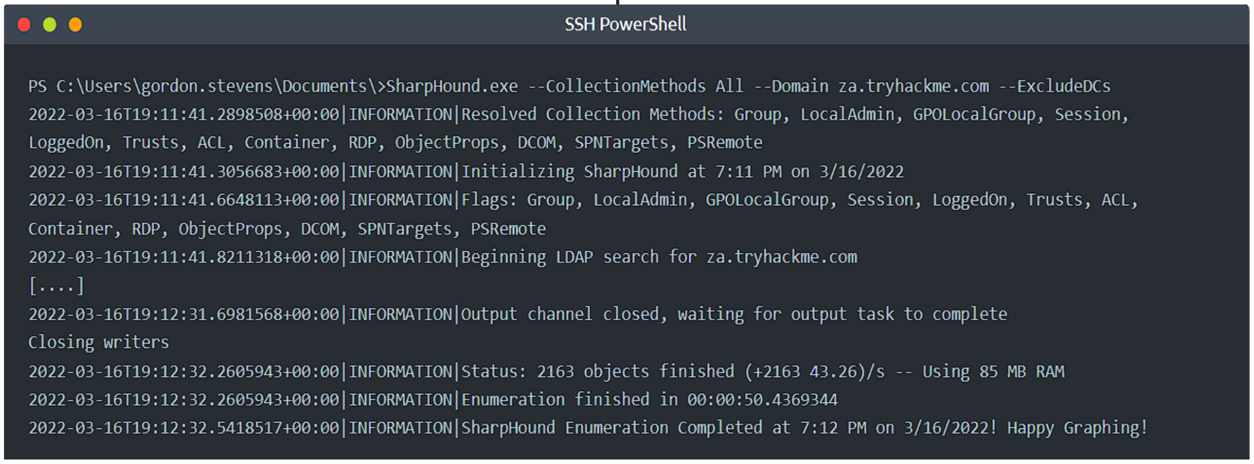
When using these collector scripts on an assessment, there is a high likelihood that these files will be detected as malware and raise an alert to the blue team. This is again where our Windows machine that is non-domain-joined can assist. We can use the `runas` command to inject the AD credentials and point Sharphound to a Domain Controller. Since we control this Windows machine, we can either disable the AV or create exceptions for specific files or folders

Sharphound command,

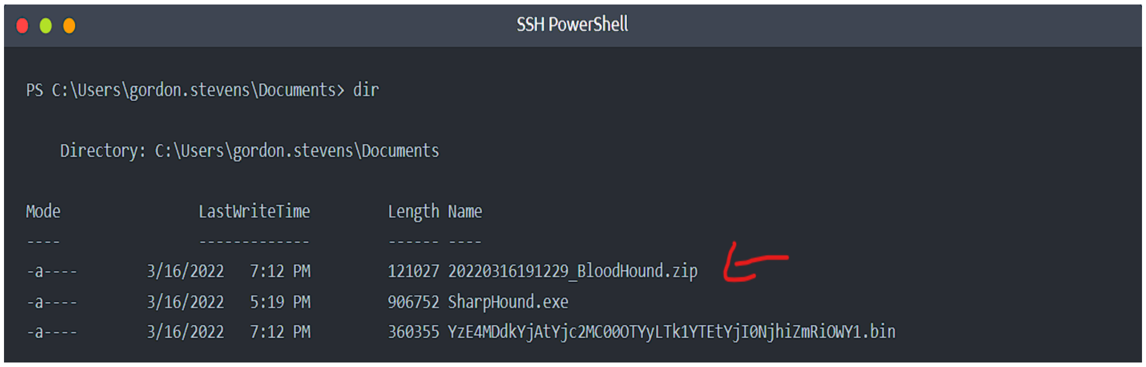
```
Sharphound.exe --CollectionMethods <Methods> --Domain za.tryhackme.com --ExcludeDCs
```

Parameters explained:

- **CollectionMethods** - Determines what kind of data Sharphound would collect. The most common options are Default or All. Also, since Sharphound caches information, once the first run has been completed, you can only use the Session collection method to retrieve new user sessions to speed up the process.
- **Domain** - Here, we specify the domain we want to enumerate. In some instances, you may want to enumerate a parent or other domain that has trust with your existing domain. You can tell Sharphound which domain should be enumerated by altering this parameter.
- **ExcludeDCs** -This will instruct Sharphound not to touch domain controllers, which reduces the likelihood that the Sharphound run will raise an alert.



```
PS C:\Users\gordon.stevens\Documents\>SharpHound.exe --CollectionMethods All --Domain za.tryhackme.com --ExcludeDCs
2022-03-16T19:11:41.2898508+00:00|INFORMATION|Resolved Collection Methods: Group, LocalAdmin, GPOLocalGroup, Session,
LoggedOn, Trusts, ACL, Container, RDP, ObjectProps, DCOM, SPNTargets, PSRemote
2022-03-16T19:11:41.3056683+00:00|INFORMATION|Initializing SharpHound at 7:11 PM on 3/16/2022
2022-03-16T19:11:41.6648113+00:00|INFORMATION|Flags: Group, LocalAdmin, GPOLocalGroup, Session, LoggedOn, Trusts, ACL,
Container, RDP, ObjectProps, DCOM, SPNTargets, PSRemote
2022-03-16T19:11:41.8211318+00:00|INFORMATION|Beginning LDAP search for za.tryhackme.com
[....]
2022-03-16T19:12:31.6981568+00:00|INFORMATION|Output channel closed, waiting for output task to complete
Closing writers
2022-03-16T19:12:32.2605943+00:00|INFORMATION|Status: 2163 objects finished (+2163 43.26)/s -- Using 85 MB RAM
2022-03-16T19:12:32.2605943+00:00|INFORMATION|Enumeration finished in 00:00:50.4369344
2022-03-16T19:12:32.5418517+00:00|INFORMATION|SharpHound Enumeration Completed at 7:12 PM on 3/16/2022! Happy Graphing!
```



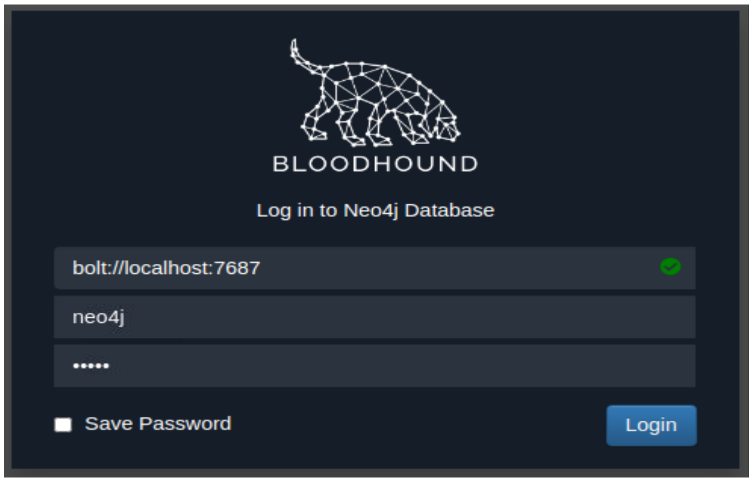
```
PS C:\Users\gordon.stevens\Documents> dir

    Directory: C:\Users\gordon.stevens\Documents

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         3/16/2022   7:12 PM         121027 20220316191229_BloodHound.zip
-a----         3/16/2022   5:19 PM         906752 SharpHound.exe
-a----         3/16/2022   7:12 PM         360355 YzE4MDdkYjAtYjc2MC00OTYyLTk1YTEtYjI0NjhiZmRiOWY1.bin
```

Bloodhound is the GUI that allows us to import data captured by Sharphound and visualise it into attack paths. Bloodhound uses Neo4j as its backend database and graphing system. Neo4j is a graph database management system

Use this command to stard neo4j console,

`neo4j console`

In another Terminal tab, run `bloodhound --no-sandbox`. This will show you the authentication GUI



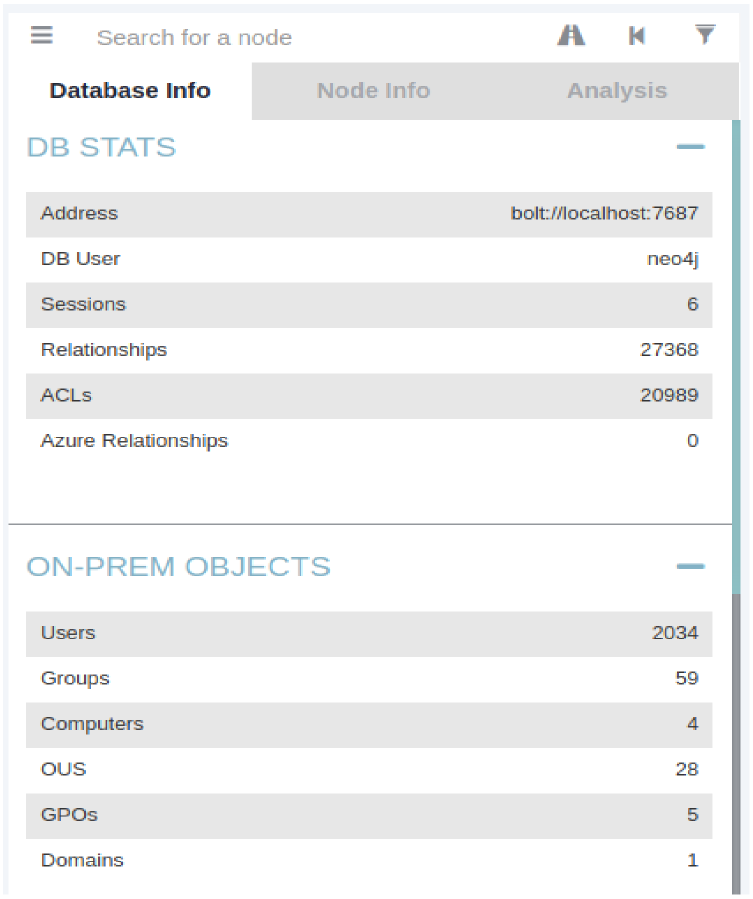
The default credentials for the neo4j database will be `neo4j:neo4j`

you will need to recover the ZIP file from the Windows host. The simplest way is to use SCP command on your AttackBox

```
scp <AD Username>@THMJMP1.za.tryhackme.com:C:/Users/<AD Username>/Documents/<Sharphound ZIP>
```

Drag and drop the ZIP file onto the Bloodhound GUI to import into Bloodhound. It will show that it is extracting the files and initiating the import.

Once all JSON files have been imported, we can start using Bloodhound to enumerate attack paths for this specific domain.



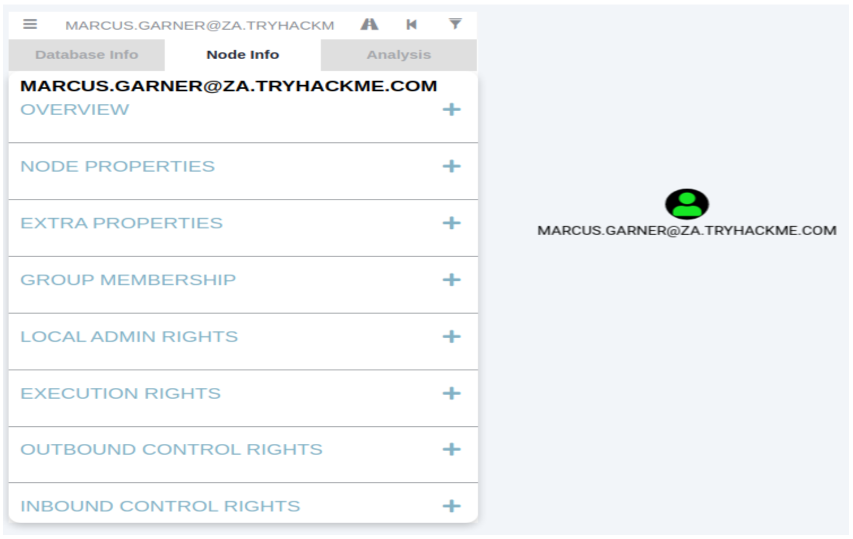
DB STATS

Address	bolt://localhost:7687
DB User	neo4j
Sessions	6
Relationships	27368
ACLs	20989
Azure Relationships	0

ON-PREM OBJECTS

Users	2034
Groups	59
Computers	4
OUS	28
GPOs	5
Domains	1

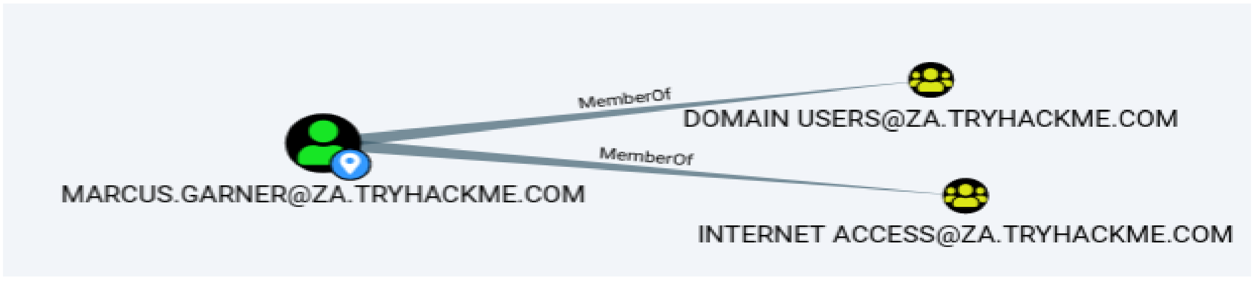
Note that if you import a new run of Sharphound, it would cumulatively increase these counts. First, we will look at Node Info. Let's search for our AD account in Bloodhound. You must click on the node to refresh the view



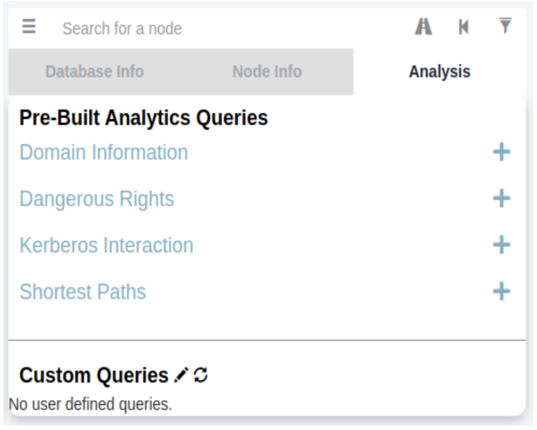
Each of the categories provides the following information:

- **Overview** - Provides summaries information such as the number of active sessions the account has and if it can reach high-value targets.
- **Node Properties** - Shows information regarding the AD account, such as the display name and the title.
- **Extra Properties** - Provides more detailed AD information such as the distinguished name and when the account was created.
- **Group Membership** - Shows information regarding the groups that the account is a member of.
- **Local Admin Rights** - Provides information on domain-joined hosts where the account has administrative privileges.
- **Execution Rights** - Provides information on special privileges such as the ability to RDP into a machine.
- **Outbound Control Rights** - Shows information regarding AD objects where this account has permissions to modify their attributes.
- **Inbound Control Rights** - Provides information regarding AD objects that can modify the attributes of this account.

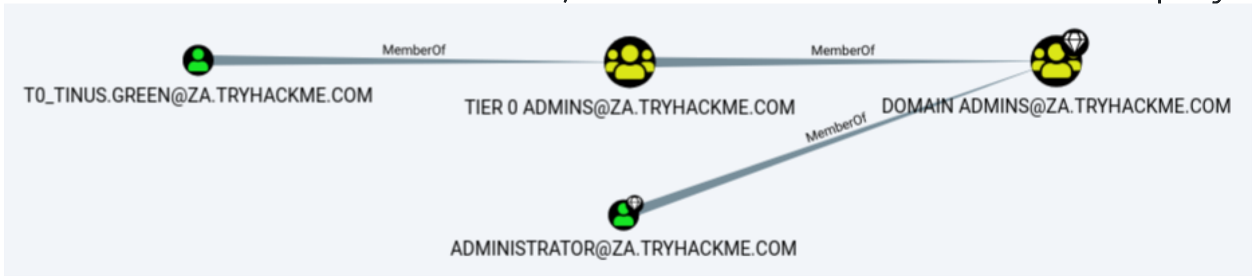
If you want more information in each of these categories, you can press the number next to the information query. For instance, let's look at the group membership associated with our account. By pressing the number next to "First Degree Group Membership", we can see that our account is a member of two groups.


MemberOf
DOMAIN USERS@ZA.TRYHACKME.COM
MemberOf
MARCUS.GARNER@ZA.TRYHACKME.COM
INTERNET ACCESS@ZA.TRYHACKME.COM

Next, we will be looking at the Analysis queries. These are queries that the creators of Bloodhound have written themselves to enumerate helpful information.



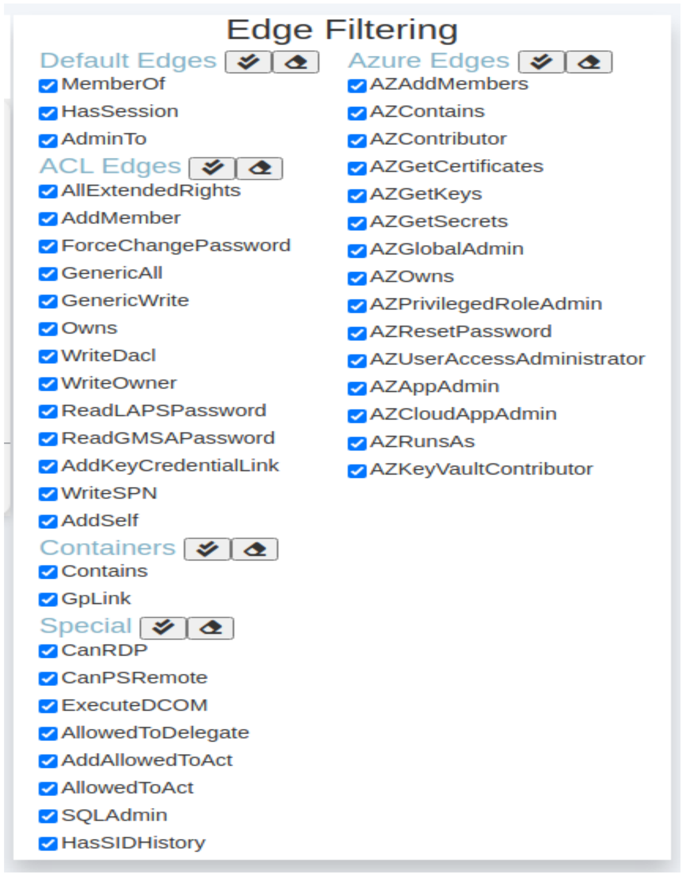
Under the Domain Information section, we can run the Find all Domain Admins query.


T0_TINUS.GREEN@ZA.TRYHACKME.COM
MemberOf
TIER 0 ADMINS@ZA.TRYHACKME.COM
MemberOf
DOMAIN ADMINS@ZA.TRYHACKME.COM
MemberOf
ADMINISTRATOR@ZA.TRYHACKME.COM

The icons are called nodes, and the lines are called edges. Let's take a deeper dive into what Bloodhound is showing us. There is an AD user account with the username of **T0_TINUS.GREEN**, that is a member of the group **Tier 0 ADMINS**. But, this group is a nested group into the **DOMAIN ADMINS** group, meaning all users that are part of the **Tier 0 ADMINS** group are effectively DAs.

Furthermore, there is an additional AD account with the username of **ADMINISTRATOR** that is part of the **DOMAIN ADMINS** group. Hence, there are two accounts in our attack surface that we can probably attempt to compromise if we want to gain DA rights. Since the **ADMINISTRATOR** account is a built-in account, we would likely focus on the user account instead.

Bloodhound has various available edges that can be accessed by the filter icon



Edge Filtering

Default Edges
- [✓] MemberOf
- [✓] HasSession
- [✓] AdminTo

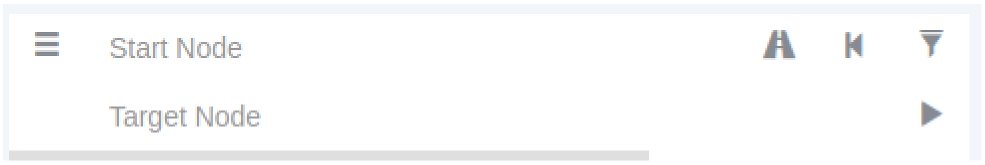
ACL Edges
- [✓] AllExtendedRights
- [✓] AddMember
- [✓] ForceChangePassword
- [✓] GenericAll
- [✓] GenericWrite
- [✓] Owns
- [✓] WriteDacl
- [✓] WriteOwner
- [✓] ReadLAPSPassword
- [✓] ReadGMSAPassword
- [✓] AddKeyCredentialLink
- [✓] WriteSPN
- [✓] AddSelf

Containers
- [✓] Contains
- [✓] GpLink

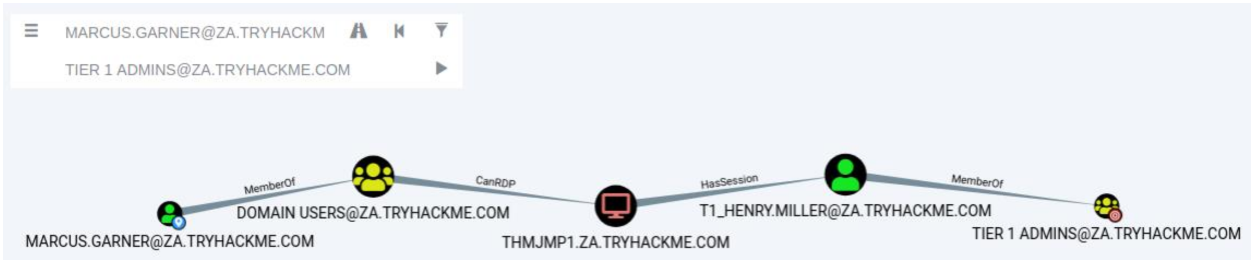
Special
- [✓] CanRDP
- [✓] CanPSRemote
- [✓] ExecuteDCOM
- [✓] AllowedToDelegate
- [✓] AddAllowedToAct
- [✓] AllowedToAct
- [✓] SQLAdmin
- [✓] HasSIDHistory

Azure Edges
- [✓] AZAddMembers
- [✓] AZContains
- [✓] AZContributor
- [✓] AZGetCertificates
- [✓] AZGetKeys
- [✓] AZGetSecrets
- [✓] AZGlobalAdmin
- [✓] AZOwns
- [✓] AZPrivilegedRoleAdmin
- [✓] AZResetPassword
- [✓] AZUserAccessAdministrator
- [✓] AZAppAdmin
- [✓] AZCloudAppAdmin
- [✓] AZRunsAs
- [✓] AZKeyVaultContributor

We will run a search in Bloodhound to enumerate the attack path. Press the path icon to allow for path searching.



Our Start Node would be our AD username, and our End Node will be the **Tier 1 ADMINS** group since this group has administrative privileges over servers.


MARCUS.GARNER@ZA.TRYHACKM
TIER 1 ADMINS@ZA.TRYHACKME.COM
MemberOf
CanRDP
HasSession
MemberOf
MARCUS.GARNER@ZA.TRYHACKME.COM
DOMAIN USERS@ZA.TRYHACKME.COM
THMJMP1.ZA.TRYHACKME.COM
T1_HENRY.MILLER@ZA.TRYHACKME.COM
TIER 1 ADMINS@ZA.TRYHACKME.COM

We could do something like the following to exploit this path:

1. Use our AD credentials to RDP into **THMJMP1**.
2. Look for a privilege escalation vector on the host that would provide us with Administrative access.
3. Using Administrative access, we can use credential harvesting techniques and tools such as Mimikatz.
4. Since the T1 Admin has an active session on **THMJMP1**, our credential harvesting would provide us with the NTLM hash of the associated account.

Benefits

- Provides a GUI for AD enumeration.
- Has the ability to show attack paths for the enumerated AD information.
- Provides more profound insights into AD objects that usually require several manual queries to recover.

Drawbacks

- Requires the execution of Sharphound, which is noisy and can often be detected by AV or EDR solutions.