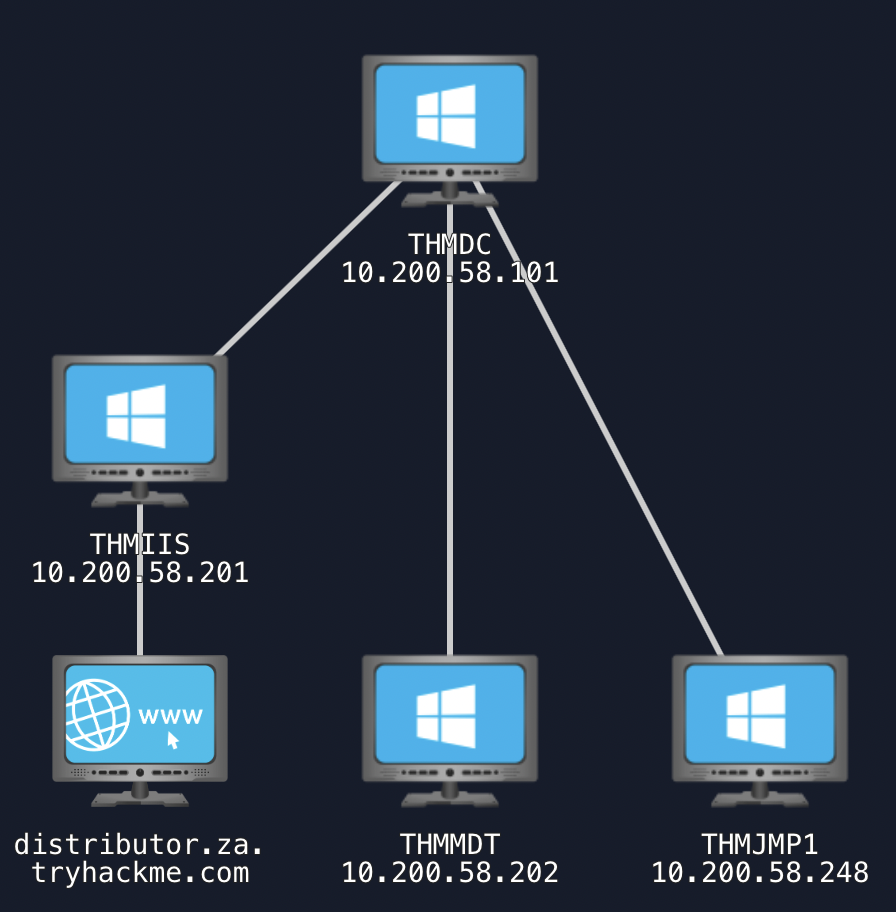
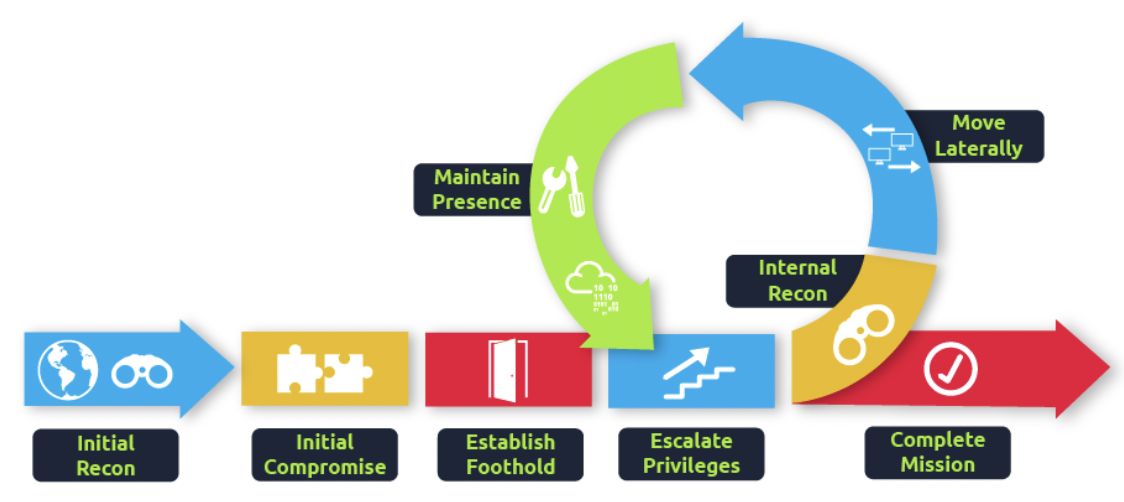
**Liệt kê Active Directory**



**Tại sao phải liệt kê Active Directory?**

Khi chúng ta có tài khoản AD đầu tiên và xác thực trên mạng, một thế giới mới với nhiều cơ hội được mở ra. Chúng ta bắt đầu liệt kê các chi tiết khác nhau về thiết lập và cấu trúc AD.

Trong quá trình kiểm thử của đội đỏ, điều này sẽ giúp chúng ta leo thang đặc quyền hoặc lateral movement để giành thêm quyền truy cập cho đến khi chúng ta có đủ đặc quyền để thực thi và đạt được mục tiêu cuối cùng. Trong hầu hết các trường hợp, việc liệt kê và khai thác có mối liên hệ chặt chẽ với nhau. Khi quá trình liệt kê tìm được con đường tấn công, lại tiếp tục liệt kê ở vị trí mới.



Để mô phỏng AD breach, bạn sẽ được cung cấp 1 tài khoản AD. Khi thiết lập mạng hoàn tất, chuyển hướng đến trang http://distributor.za.tryhackme.com/creds để nhận tài khoản. Trên attackbox, kết nối RDP hoặc SSH đến THMJMP1.za.tryhackme.com

ssh za.tryhackme.com\\<AD Username>@thmjmp1.za.tryhackme.com

xfreerdp /u:username /p:password /v:thmjmp1.za.tryhackme.com

**Credential Injection**

**Windows vs Linux**

Bạn có thể liệt kê AD từ xa bằng máy ảo Kali. Nếu bạn muốn liệt kê sâu hơn, thậm chí là khai thác, bạn cần hiểu và bắt chước kẻ thù. Do đó, bạn cần 1 máy ảo Windows. Điều này cho phép chúng ta sử dụng các công cụ có sẵn để liệt kê và khai thác. Trong mạng này, chúng ta sẽ khám phá một trong những công cụ được dựng sẵn: runas.exe

**Giải thích Runas**

Bạn đã bao giờ tìm được thông tin đăng nhập nhưng không biết cách dùng chúng? Runas có thể là câu trả lời bạn đang tìm kiếm.

Trong kiểm thử bảo mật, bạn sẽ có quyền truy cập mạng và khám phá được thông tin đăng nhập trên AD nhưng không có phương tiện hoặc đặc quyền để tạo một máy mới tham gia domain. Vì vậy chúng ta cần có khả năng dùng thông tin đăng nhập đó trên máy Windows chúng ta kiểm soát.

Nếu chúng ta có tài khoản AD ở định dạng username:password, chúng ta có thể dùng Runas, để tiêm thông tin đăng nhập trong bộ nhớ.

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

* /netonly: vì chúng ta không tham gia domain, chúng ta muốn tải thông tin đăng nhập để xác thực mạng nhưng không xác thực với domain controller. Vì vậy lệnh được thực thi cục bộ trên máy tính sẽ chạy trong bối cảnh của tài khoản Windows, nhưng mọi kết nối mạng sẽ xảy ra với tài khoản được chỉ định
* /user: chúng ta cung cấp chi tiết domain và tên người dùng. Luôn an toàn khi chúng ta sử dụng Fully Qualified Domain Name (FQDN) thay vì tên NetBIOS của domain vì điều này sẽ giúp phân giải.
* cmd.exe: chương trình chúng ta muốn thực thi khi tiêm thông tin đăng nhập.

Khi bạn chạy lệnh này, bạn sẽ được hỏi cung cấp mật khẩu. Khi chúng ta thêm tham số /netonly, thông tin đăng nhập không được xác thực bởi domain controller nên nó sẽ chấp nhận mọi mật khẩu. Chúng ta vẫn xác nhận thông tin đăng nhập trên mạng đã được tải thành công chưa.

**Lưu ý:** Nếu bạn dùng máy Windows của riêng mình, bạn phải chạy command prompt dưới tư cách là Administrator. Điều này sẽ tiêm token của Administrator vào trong CMD. Nếu bạn chạy công cụ yêu cầu đặc quyền của local Administrator từ CMD Runas sinh ra, token sẽ khả thi. Điều này không cho bạn đặc quyền của quản trị viên nhưng sẽ đảm bảo bất cứ lệnh nào bạn thực thi sẽ chạy dưới đặc quyền của admin.

**Luôn là DNS**

Bước tiếp theo bạn chỉ cần thực hiện nếu bạn dùng máy ảo Windows của riêng mình.

Sau khi cung cấp mật khẩu, một command prompt mới sẽ mở. Bây giờ chúng ta cần xác minh thông tin đăng nhập có đang hoạt động. Cách chắc chắn nhất là liệt kê SYSVOL. Bất cứ tài khoản AD nào, không quan trọng đặc quyền thấp, có thể đọc nội dung thư mục SYSVOL.

SYSVOL là thư mục tồn tại trong tất cả domain controllers. Nó là 1 thư mục chia sẻ lưu trữ GPOs và thông tin cùng với bất kỳ tập lệnh liên quan đến tên miền nào khác. Nó là 1 thành phần thiết yếu cho Active Directory vì nó cung cấp các GPOs cho tất cả máy tính trên domain. Sau đó, các máy tính tham gia domain có thể đọc GPOs và áp dụng các GPOs phù hợp, thực hiện các thay đổi cấu hình trên toàn miền.

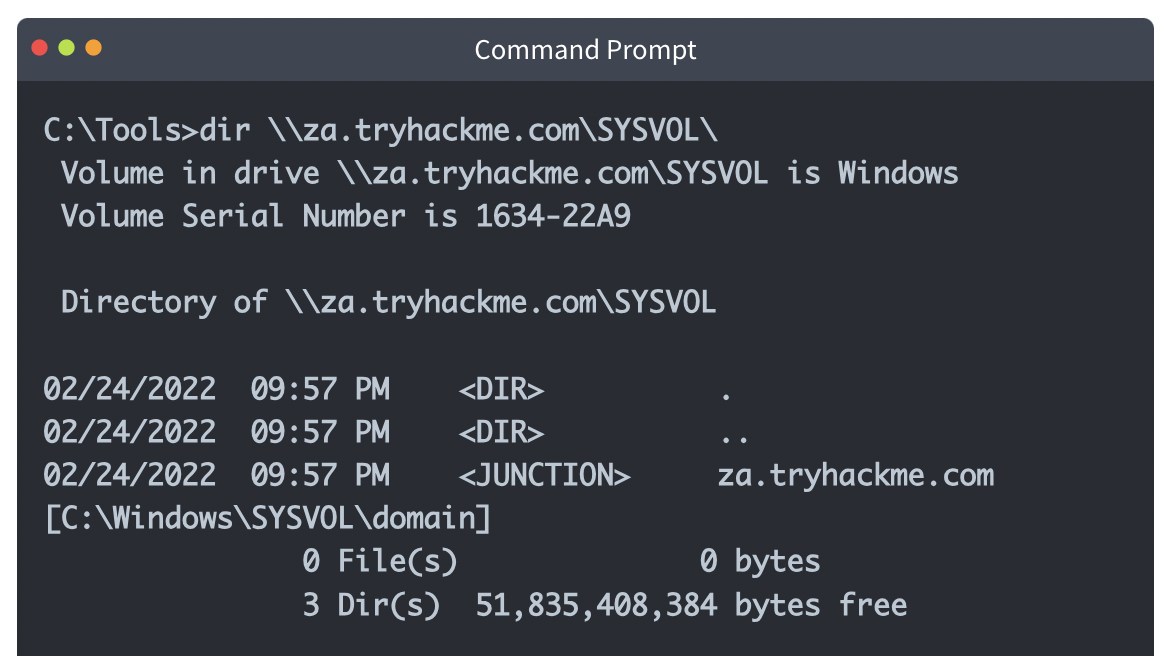
Trước khi liệt kê SYSVOL, chúng ta cần cấu hình DNS. Đôi khi, DNS nội bộ tự cấu hình cho bạn qua DHCP hoặc kết nối VPN. Cấu hình bằng tay vẫn tốt hơn. DNS server thường là domain controller. Dùng IP của domain controller, chúng ta có thể chạy lệnh sau trong Powershell:



Tất nhiên, Ethernet sẽ là bất kỳ giao diện nào được kết nối với mạng của Tryhackme. Chúng ta có thể xác minh DNS đang hoạt động bằng lệnh sau:

nslookup za.tryhackme.com

DNS đang hoạt động, cuối cùng chúng ta kiểm tra thông tin đăng nhập:



**IP vs Hostnames**

Điểm khác nhau giữa dir \\za.tryhackme.com\SYSVOL và dir \\<DC IP>\SYSVOL

Có khá nhiều sự khác biệt và nó phụ thuộc vào phương thức xác thực đang được sử dụng. Khi chúng ta cung cấp hostname, xác thực mạng là Kerberos. Vì xác thực Kerberos dùng hostname nhúng trong các tickets. Nếu chúng ta cung cấp ip, loại xác thực là NTLM. Trong một số trường hợp, các tổ chức sẽ theo dõi các cuộc tấn công OverPass và Pass-The-Hash. Xác thực NTLM là một lựa chọn tốt để tránh bị phát hiện trong trường hợp này.

**Dùng thông tin đăng nhập được tiêm**

Bây giờ chúng ta đã tiêm tài khoản AD trong bộ nhớ. Với tùy chọn /netonly, các ứng dụng trên mạng sẽ dùng thông tin đăng nhập đó để xác thực, bao gồm các ứng dụng thực thi từ command prompt. Bạn đã bao giờ gặp trường hợp cơ sở dữ liệu MS SQL sử dụng xác thực Windows Authentication và bạn chưa tham gia domain? Chạy MS SQL Studio từ command prompt Runas; mặc dù nó hiển thị tên người dùng cục bộ của bạn, hãy nhấp vào Đăng nhập và nó sẽ sử dụng tài khoản AD để xác thực. Chúng ta có thể dùng điều này để xác thực các ứng dụng web bằng giao thức NTLM.

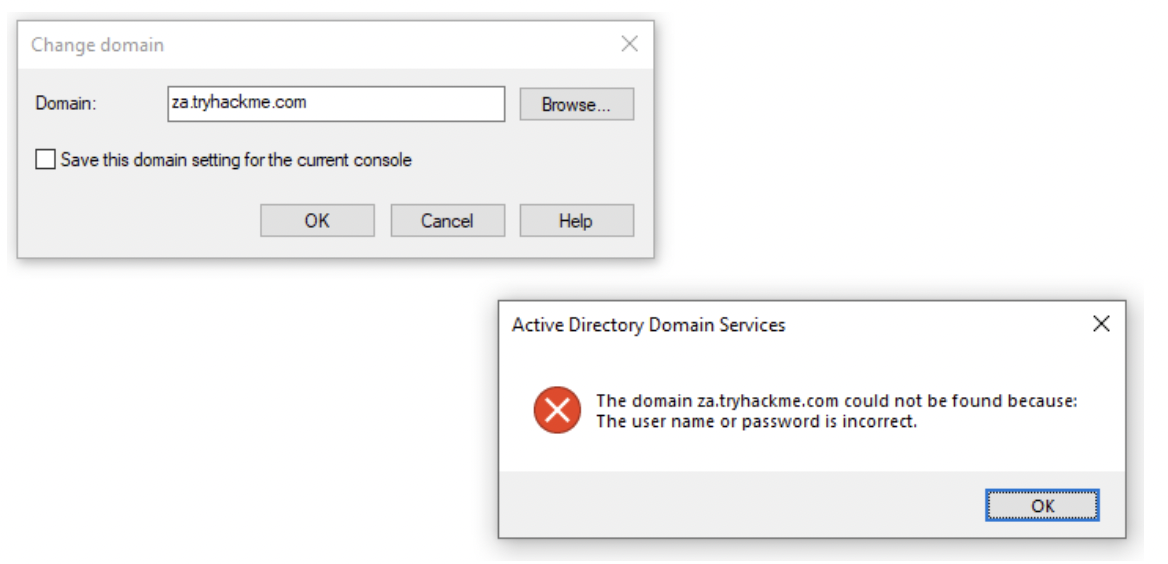
**Liệt kê bằng Microsoft Management Console**

**Microsoft Management Console**

Trong bài này, chúng ta sẽ khám phá phương pháp liệt kê đầu tiên, đây là phương pháp duy nhất sử dụng GUI cho đến bài cuối cùng. Chúng ta sẽ dùng Microsoft Management Console (MMC) với Remote Server Administration Tools (RSAT) AD Snap-Ins. Nếu bạn dùng máy ảo Windows được cung cấp (THMJMP1), nó đã được cài đặt sẵn cho bạn. Tuy nhiên, nếu bạn đang dùng máy Windows của riêng mình, bạn có thể làm theo các bước sau để cài đặt 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

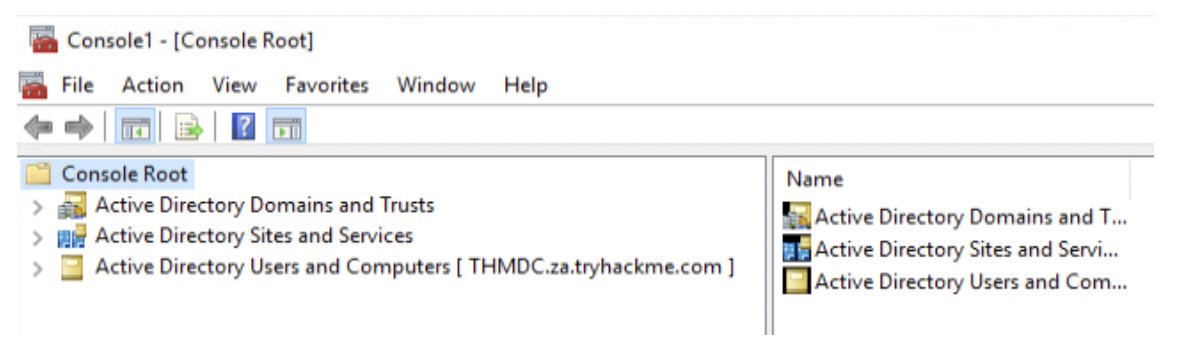
Bạn có thể bắt đầu MMC bằng cách dùng Windows Start, chọn run và nhập MMC. Nếu chúng ta chỉ chạy MMC bình thường, nó sẽ không hoạt động vì máy tính của chúng ta chưa tham gia domain và tài khoản cục bộ không xác thực được với domain.



Đây là lúc cửa sổ Runas phát huy tác dụng. Trong cửa sổ đó, chúng ta có thể bắt đầu MMC, đảm bảo kết nối mạng MMC dùng tài khoản AD được tiêm.

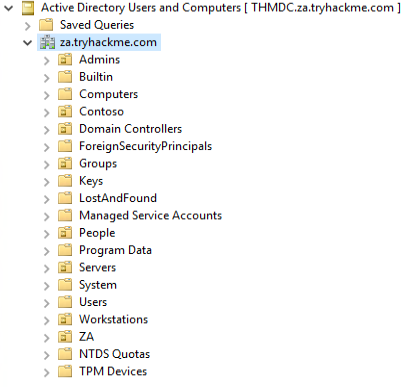
Trong MMC, chúng ta có thể đính kém AD RSAT Snap-In:

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

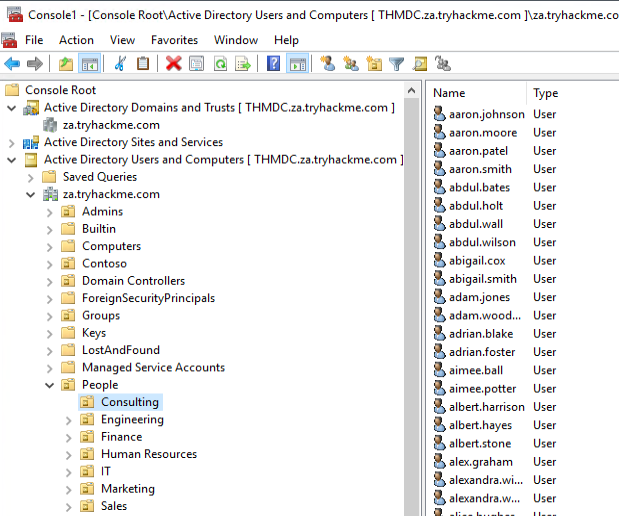


**Users and Computers**

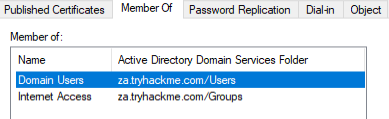
Let's take a look at the Active Directory structure. For this task, we will focus on AD Users and Computers. Expand that snap-in and expand the za domain to see the initial Organisational Unit (OU) structure:



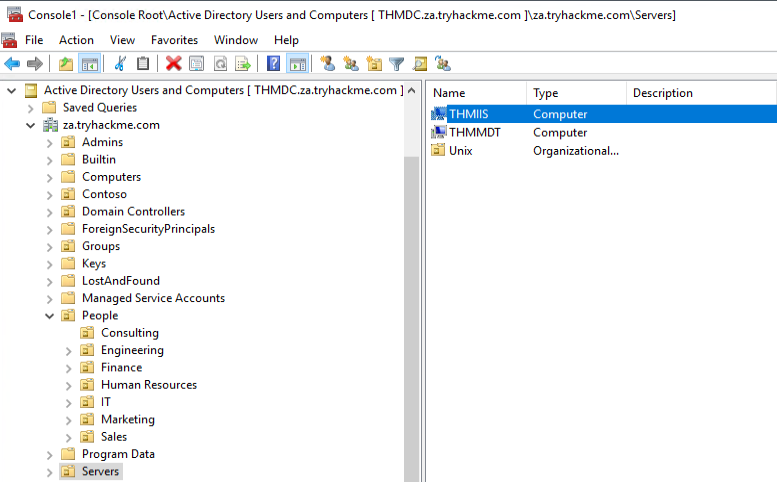
Let's take a look at the People directory. Here we see that the users are divided according to department OUs. Clicking on each of these OUs will show the users that belong to that department.



Clicking on any of these users will allow us to review all of their properties and attributes. We can also see what groups they are a member of:



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.



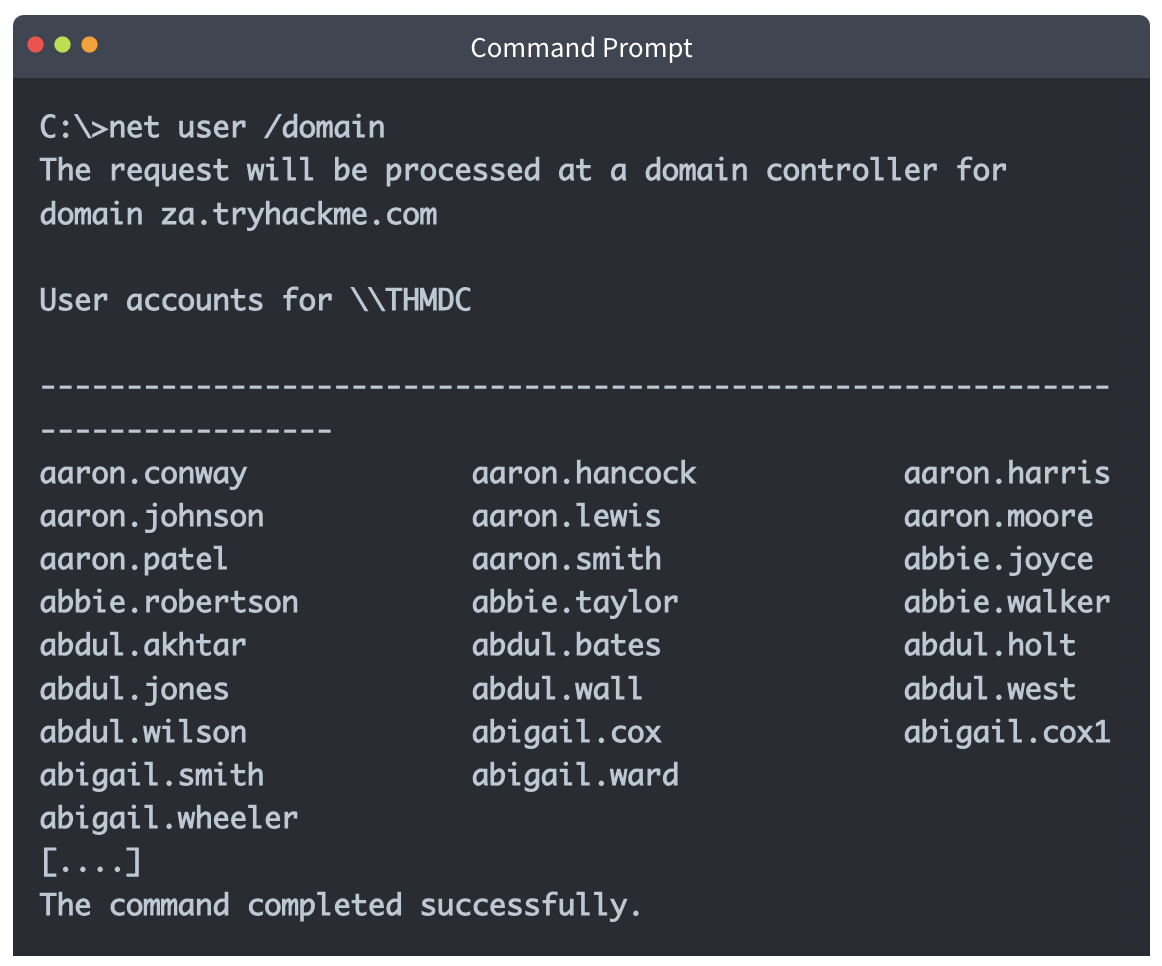
If we had the relevant permissions, we could also use MMC to directly make changes to AD, such as changing the user's password or adding an account to a specific group. Play around with MMC to better understand the AD domain structure. Make use of the search feature to look for objects.

**Liệt kê bằng Command Prompt**

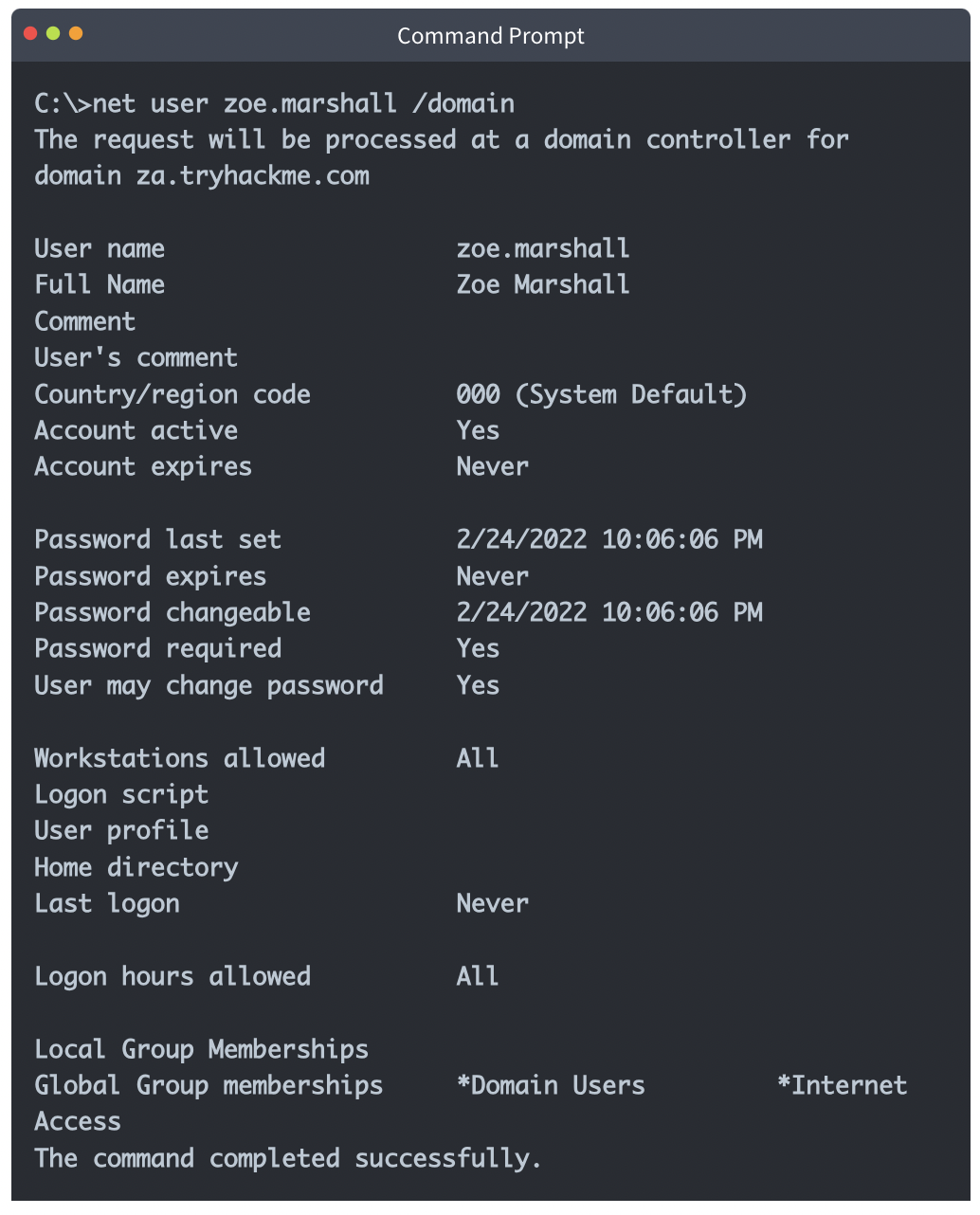
CMD có một công cụ được dựng sẵn để liệt kê AD: net. Lệnh net liệt kê thông tin về hệ thống cục bộ và AD.

**Users**

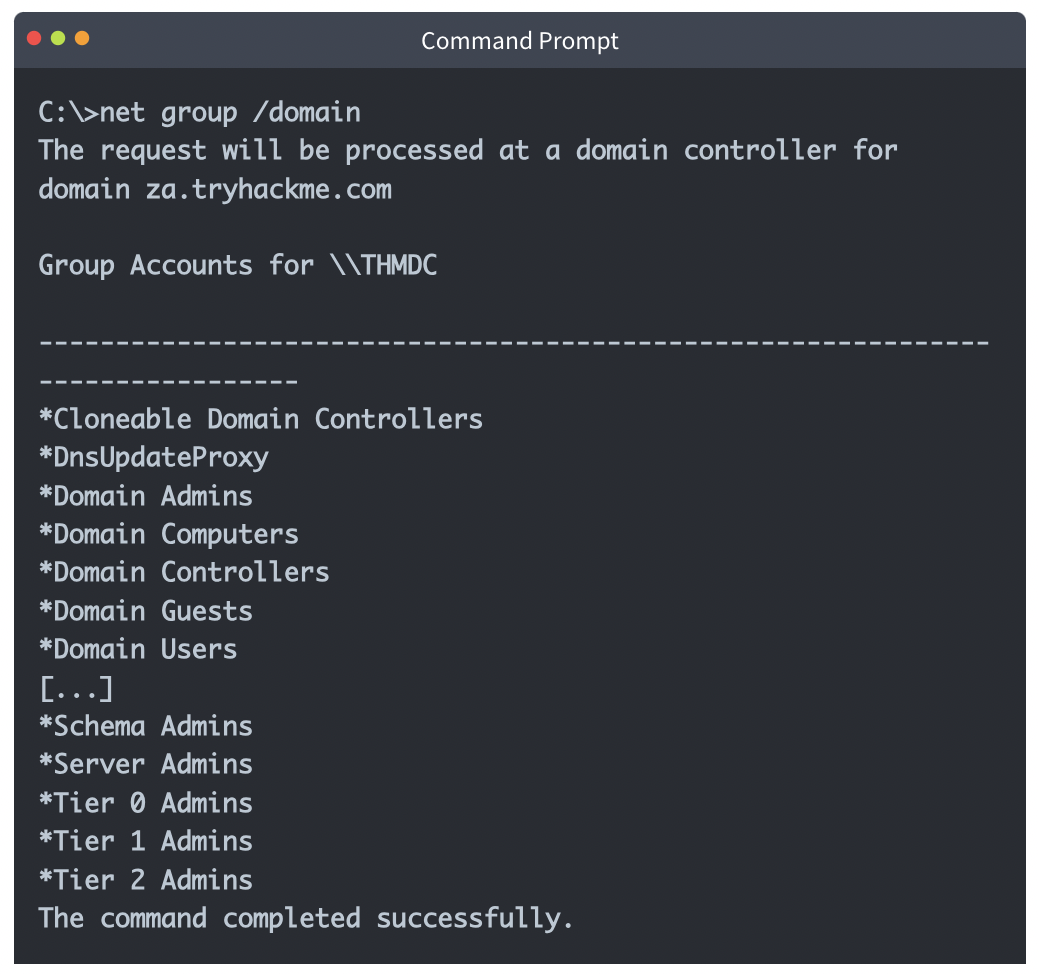
Chúng ta có thể dùng lệnh net để liệt kê tất cả người dùng trong miền AD:

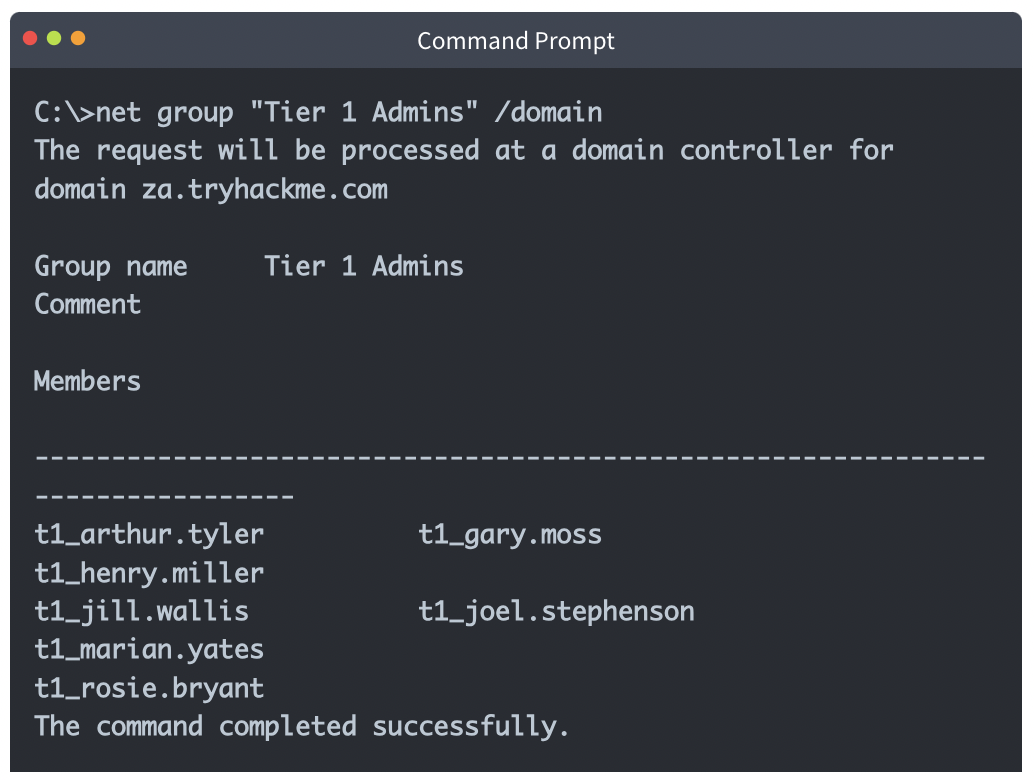


Liệt kê chi tiết về một người dùng

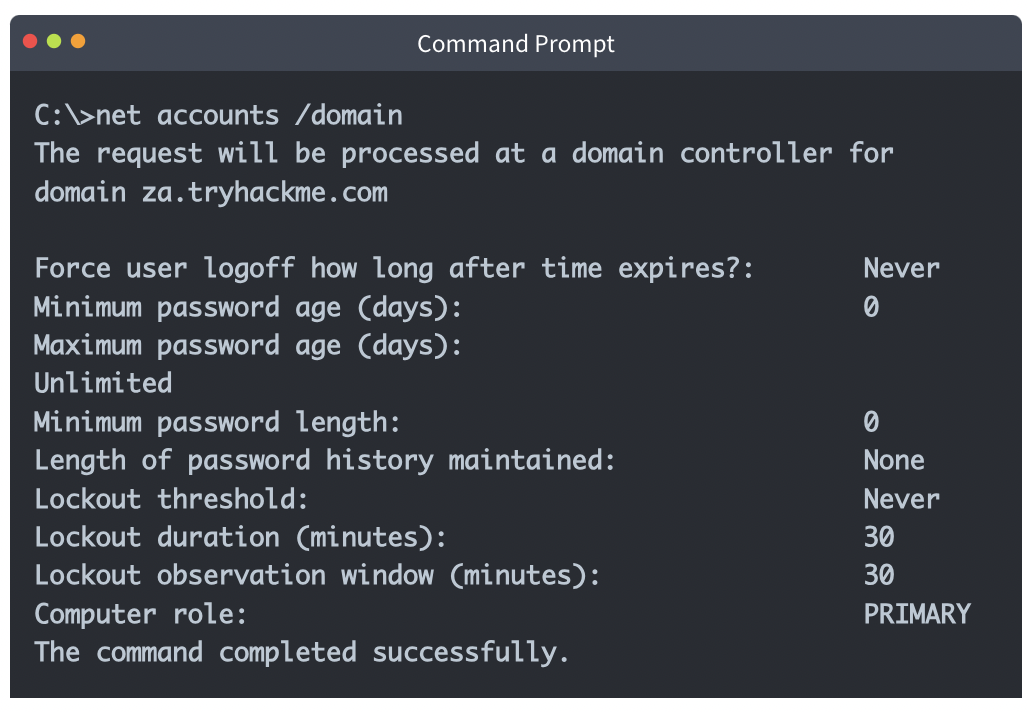


**Groups**





**Password Policy**



* Lệnh net phải được thực thi từ máy tham gia domain. Nếu máy không tham gia domain, nó sẽ mặc định là WORKGROUP domain.
* Lệnh net không hiển thị tất cả thông tin. Ví dụ, nếu một người dùng là thành viên của hơn mười nhóm, sẽ không hiển thị hết các nhóm.

Đối với bài này, bạn phải dùng THMJMP1 và không thể sử dụng máy ảo Windows của riêng mình.

**Liệt kê bằng Powershell**

**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](https://github.com/PowerShellEmpire/PowerTools/tree/master/PowerView) did, we can already get very far using the built-in ones.

Since we installed the AD-RSAT tooling in Task 3, it automatically installed the associated cmdlets for us. There are 50+ cmdlets installed. We will be looking at some of these, but refer to [this list for the complete list of cmdlets.](https://docs.microsoft.com/en-us/powershell/module/activedirectory/?view=windowsserver2022-ps)

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

**Users**

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

SSHPowerShell

**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**

**[...]**

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

For most of these cmdlets, 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:

SSHPowerShell

**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**

**Groups**

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

SSHPowerShell

**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:

SSHPowerShell

**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**

**AD Objects**

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:

SSHPowerShell

**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:

SSHPowerShell

**PS C:\> Get-ADObject -Filter 'badPwdCount -gt 0' -Server za.tryhackme.com**

**PS C:\>**

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

**Domains**

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

SSHPowerShell

**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**

**Altering AD Objects**

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:

SSHPowerShell

**PS C:\> Set-ADAccountPassword -Identity gordon.stevens -Server za.tryhackme.com -OldPassword (ConvertTo-SecureString -AsPlaintext "old" -force) -NewPassword (ConvertTo-SecureString -AsPlainText "new" -Force)**

Remember to change the identity value and password for the account you were provided with for enumeration on the distributor webpage in Task 1.

**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.Top of Form

ccjalfjladsjkffadsjklfalksd

**Liệt kê bằng Bloodhound**