Lab 4: User Management

Server System Management - Windows Server Labs

<STUDENT NAME>

Academic Year: 2022-2023

Table of Contents

[Introduction 3](#_Toc128100666)

[Learning Goals 3](#_Toc128100667)

[Knowledge (what you need to know) 3](#_Toc128100668)

[Abilities (what you need to be able to do) 3](#_Toc128100669)

[Requirements 3](#_Toc128100670)

[Accessing Active Directory 4](#_Toc128100671)

[Creating & Managing Users & Groups 5](#_Toc128100672)

[Extensions - Optional Assignments 12](#_Toc128100673)

## Introduction

In this lab we will populate our AD environment with users and groups. In the first part we try and enumerate all information in our Windows environment. Afterwards we will add more users (both manually and with PowerShell automation).

## Learning Goals

# Knowledge (what you need to know)

* What (user & computer) information can you enumerate in an AD domain.
* What attributes are required when creating users & groups

# Abilities (what you need to be able to do)

* CRUD operations on users & groups through the GUI and with PowerShell
* Enumerating information when having access to an AD domain

## Requirements

The three virtual machines, fully configured as they should be at the end of Lab 3 – installing an AD domain.

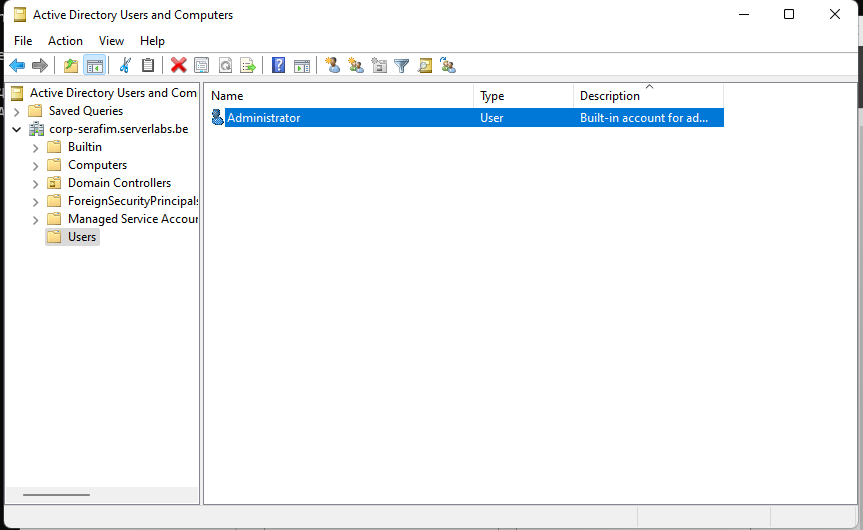
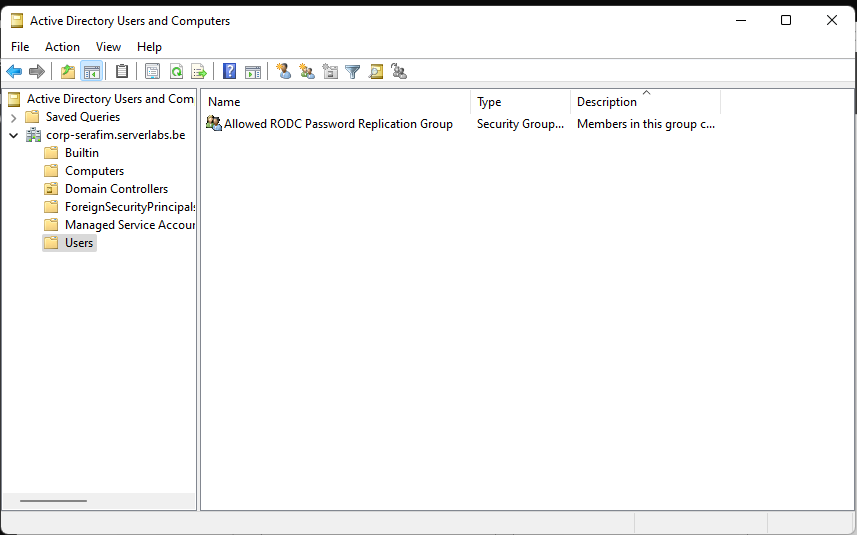
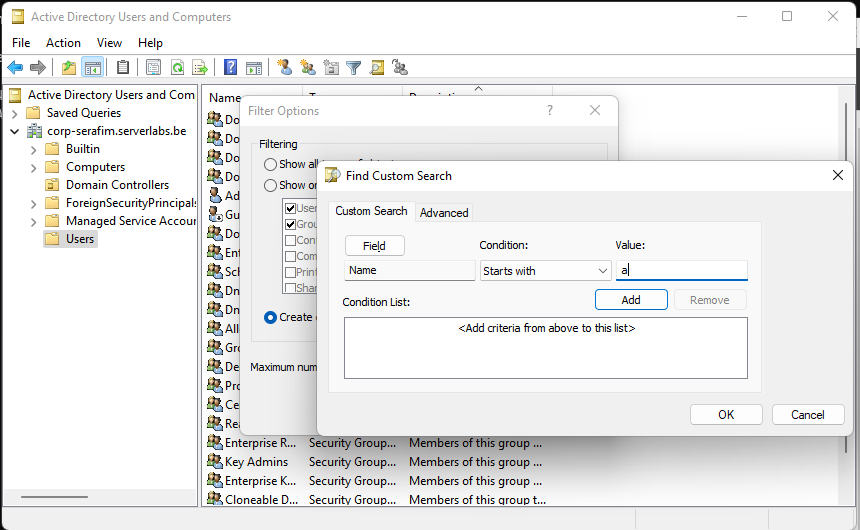
## Accessing Active Directory

In this part of the lab you will access and consult objects from Active Directory using the Active Directory Users and Computers (ADU&C) tool and PS-cmdlets. The best practice is to do everything from your Windows 11 client and logged in as the domain administrator, but you are free to do this first part of the lab on your GUI server as well to save resources (the other virtual machines are not required for now).

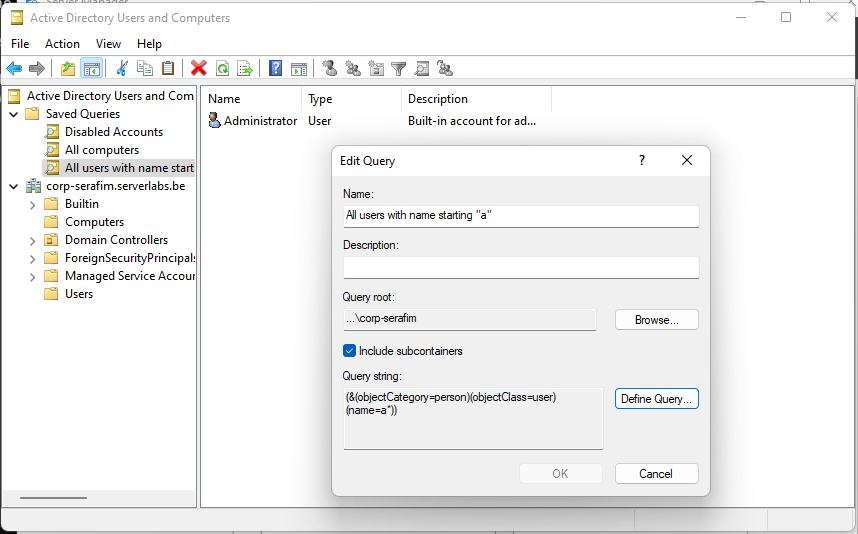
1. We will start by **enumerating** information from Active Directory through a GUI tool. Open the ADU&C tool and search for the following objects:

Server Manager > Tools > Active Directory Users and Computers

* 1. All user(groups) whose name starts with the letter “a”

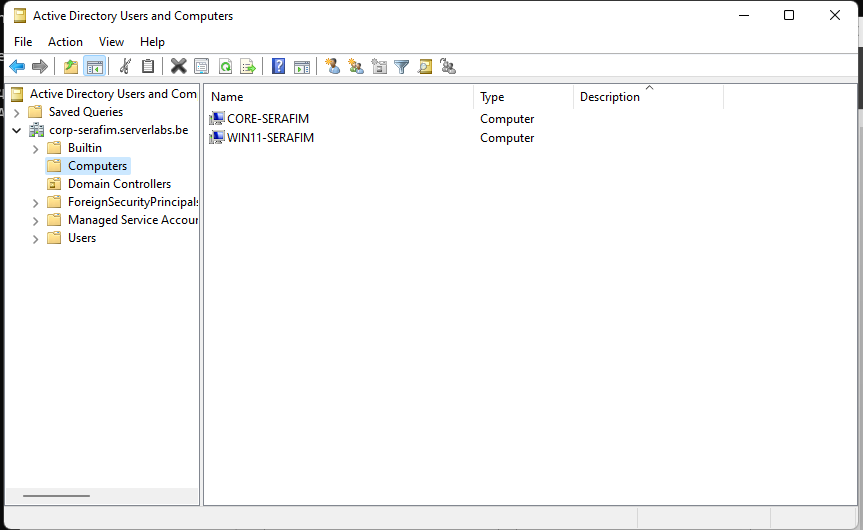


I filtered on Group - Name, and also on User - Name

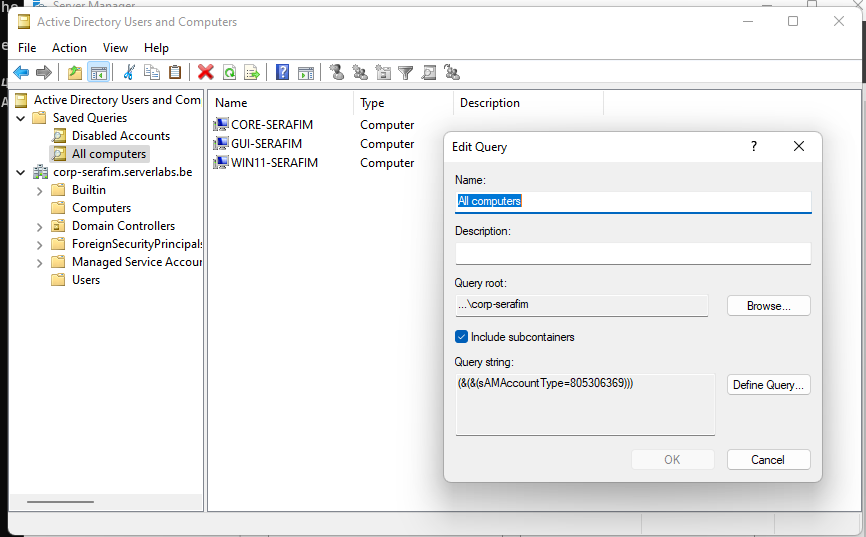


Queries also work good.

* 1. All computers

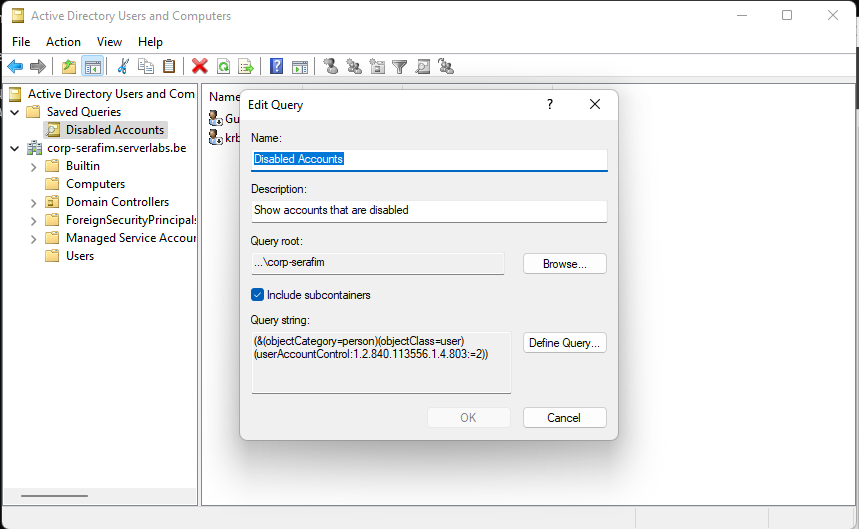
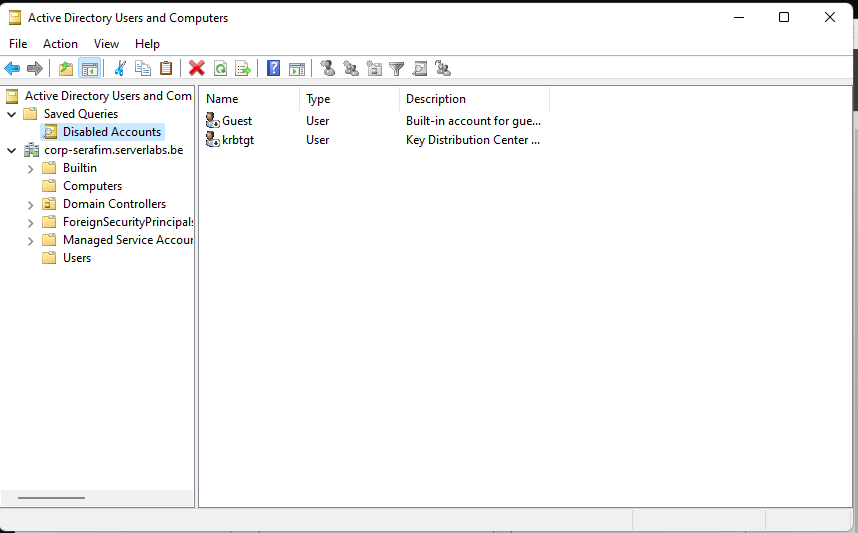


It does not show the Domain Controller in this one, but I believe it is normal (it is in another category)



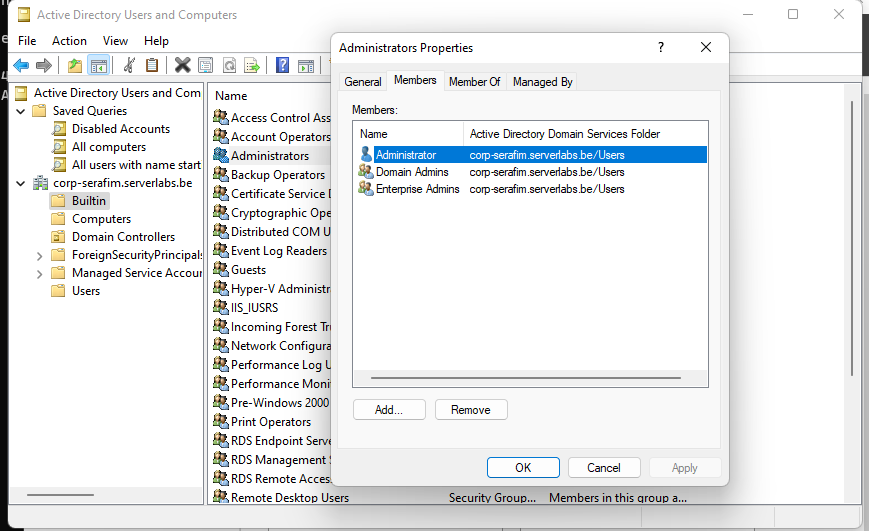
Queries also work better

* 1. All disabled accounts



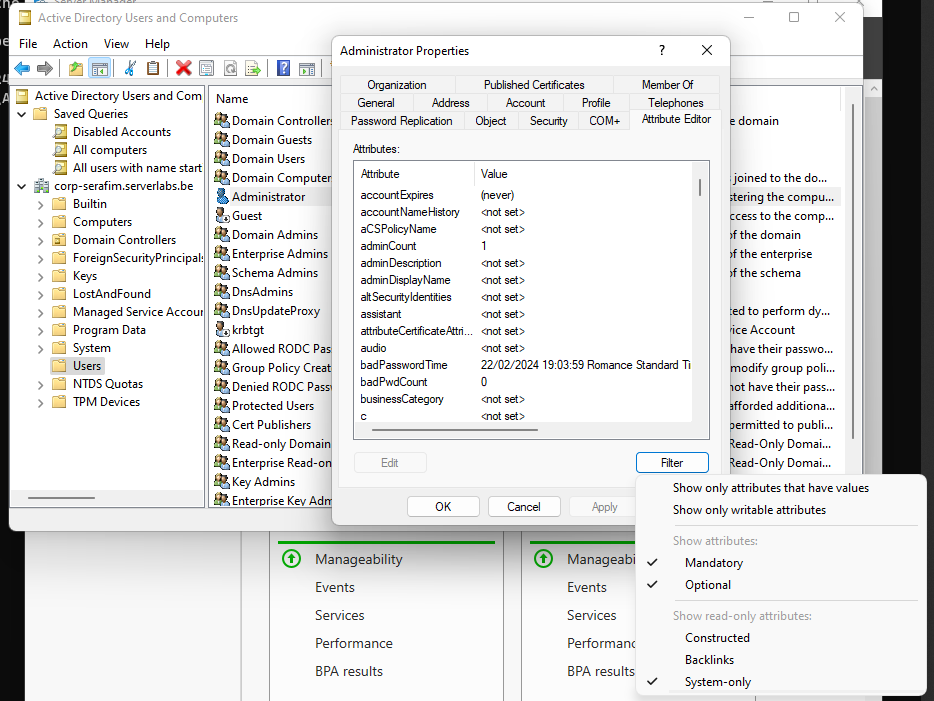
I tried to filter manually, but this also works, with queries

* 1. All members of the Administrators Group

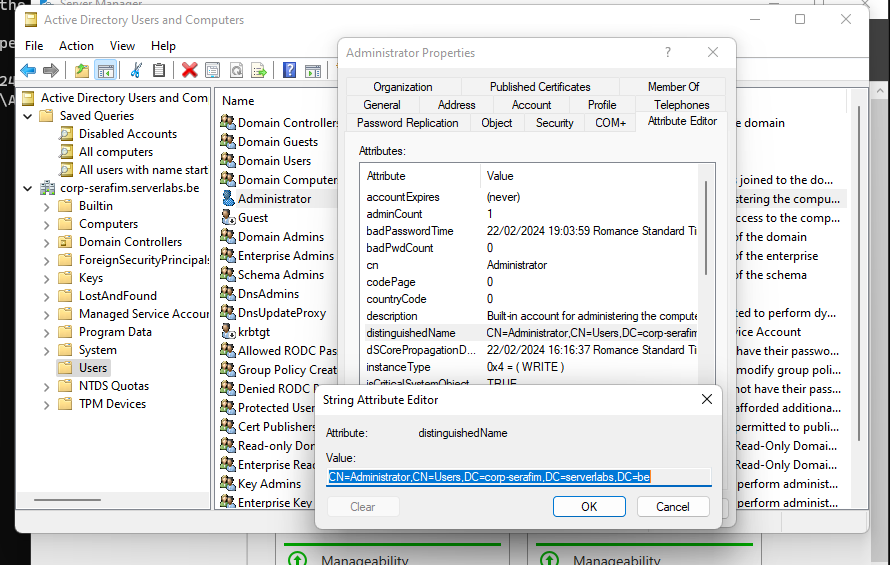


Tried it with the propetries, if that is fine

1. Activate the “advanced features” view in the AD U&C window to enumerate **all attributes** of an object and open the properties window for the **domain administrator** account. In this window the most frequently used attributes of your account are shown. In the “**attribute editor**” tab you can also perform a filter. Try to create a filter that only shows attributes for fields that have a value. Write down the distinguished name of the domain administrator account.



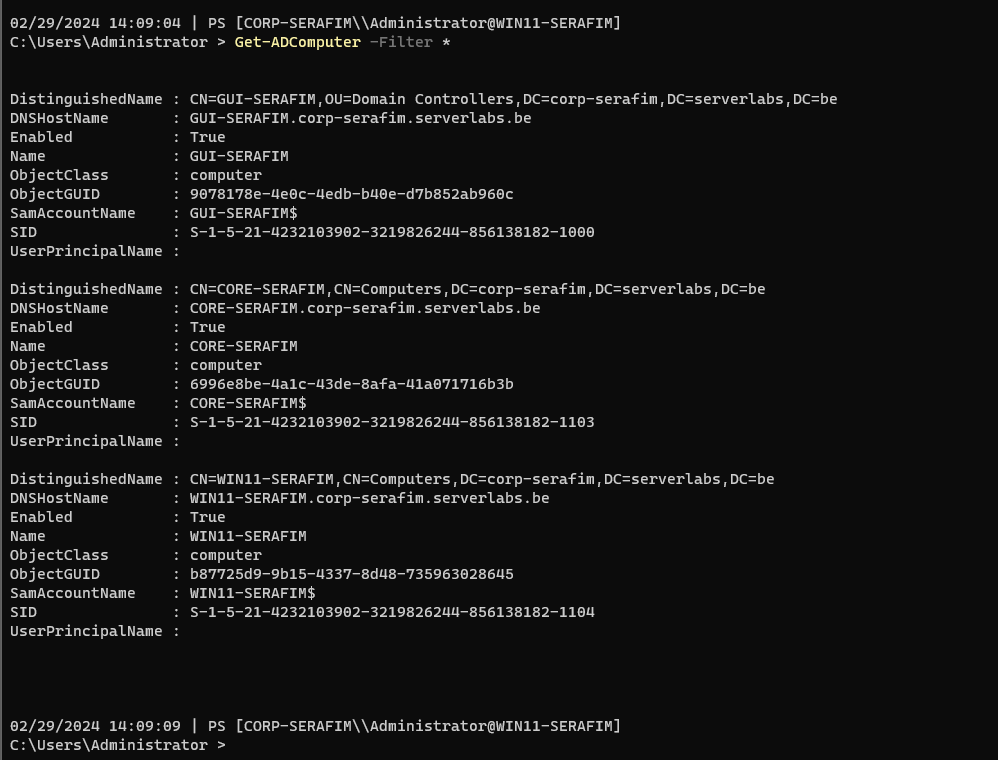
**CN=Administrator,CN=Users,DC=corp-serafim,DC=serverlabs,DC=be**



1. Use PowerShell cmdlets to perform the following exercises. Write down the cmdlets below the questions.
   1. Show all computers in your domain in a non-interactive way (i.e. without asking for additional information). In the cmdlet, use only the parameters that are strictly necessary to do this.

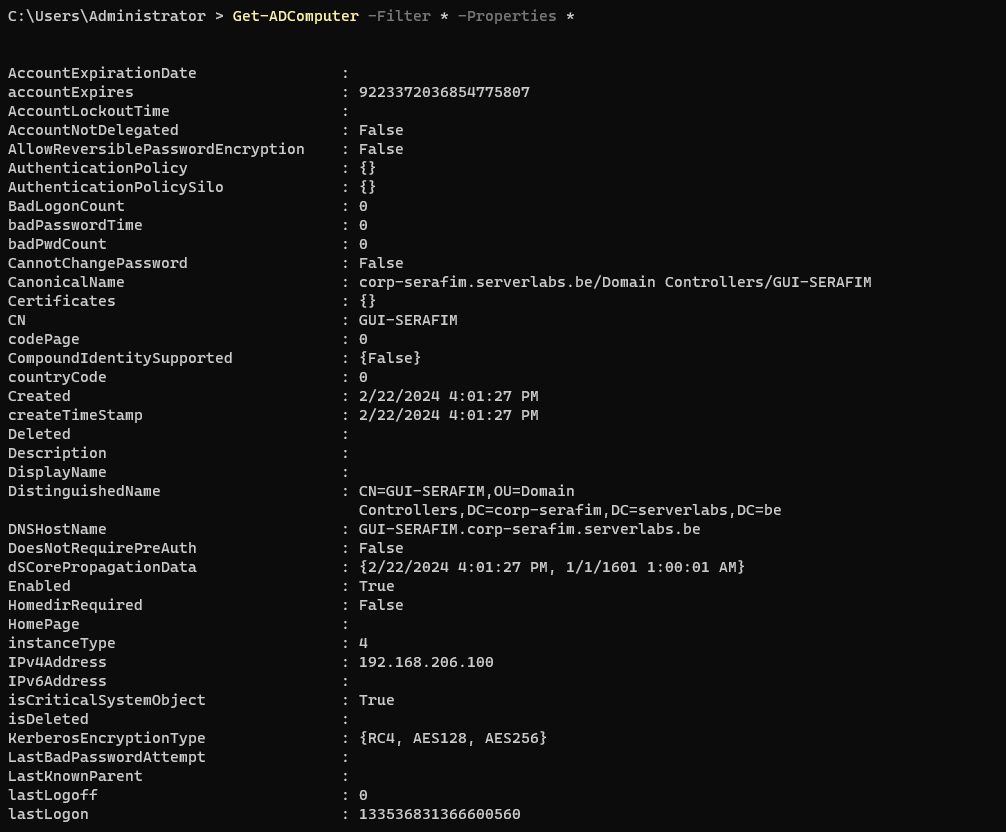
`Get-ADComputer -Filter \*`

Internet also said I would need –SearchBase, but it worked without it



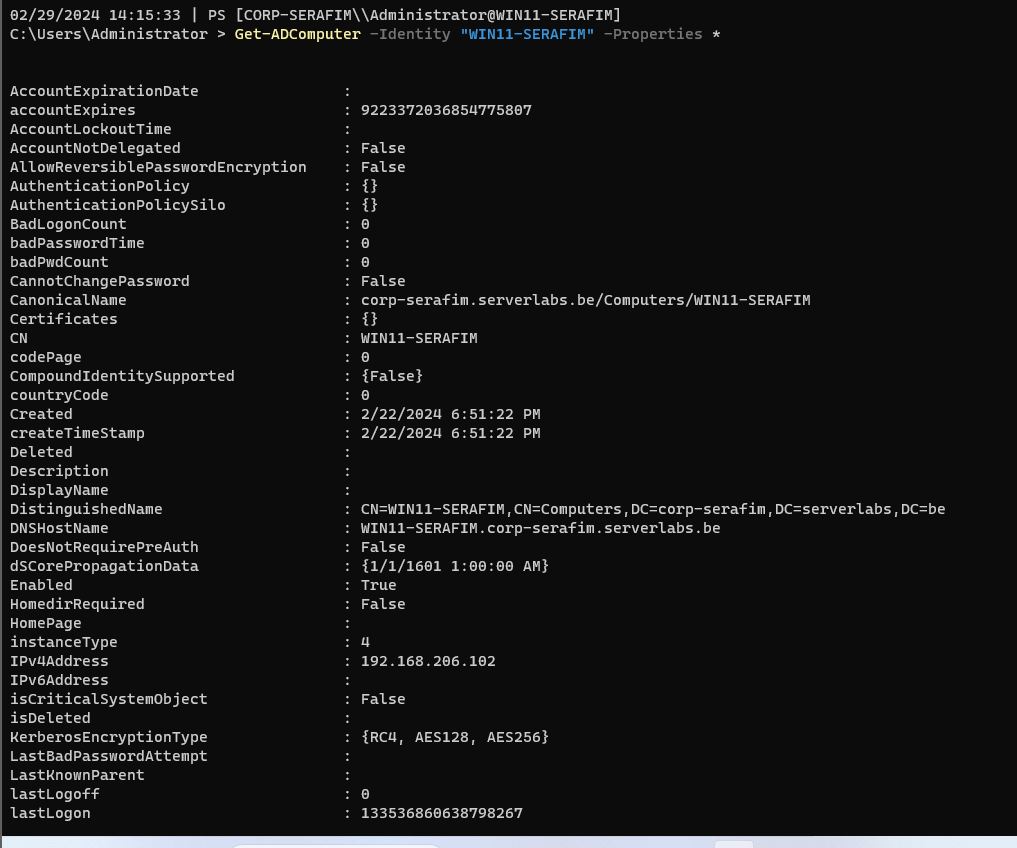
* 1. Like most cmdlets, the cmdlet used in the previous question only shows a limited number of properties/attributes of the desired object. Adjust your previous cmdlet in such a way that **all properties** of **all computers** are displayed.

`Get-ADComputer -Filter \* -Properties \*`



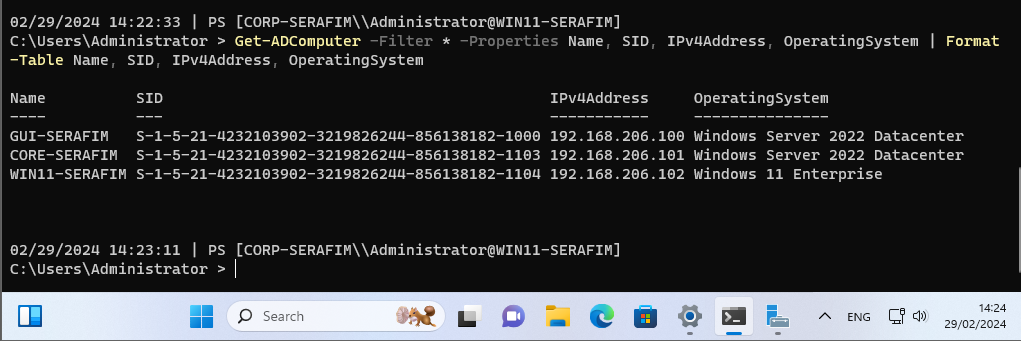
* 1. Show all attributes of the Windows 11 computer.

` Get-ADComputer -Identity "WIN11-SERAFIM" -Properties \*`



* 1. Show only the name, SID, the IPv4 address and OS of all computers in your domain. Make sure that these properties are shown in a column form (the output). **Paste a screenshot of the output below (make sure everything is clearly visible: prompt & required properties)**

`Get-ADComputer -Filter \* -Properties Name, SID, IPv4Address, OperatingSystem | Format-Table Name, SID, IPv4Address, OperatingSystem`

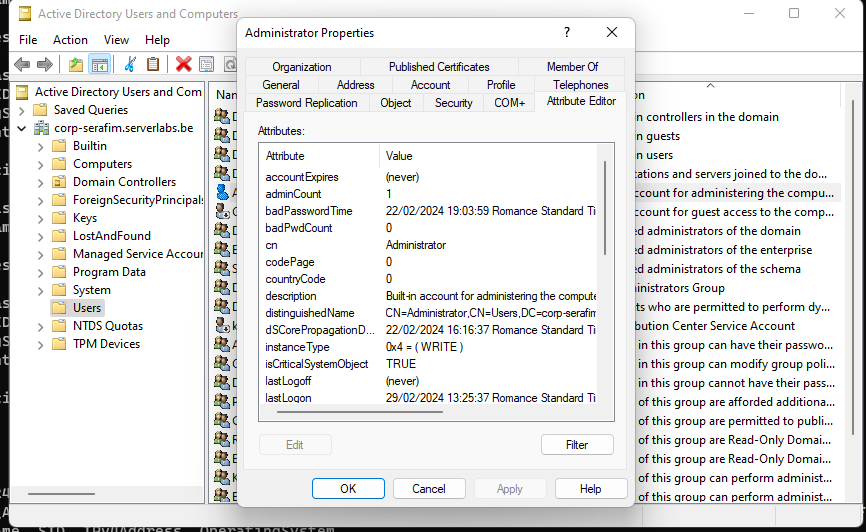


## Creating & Managing Users & Groups

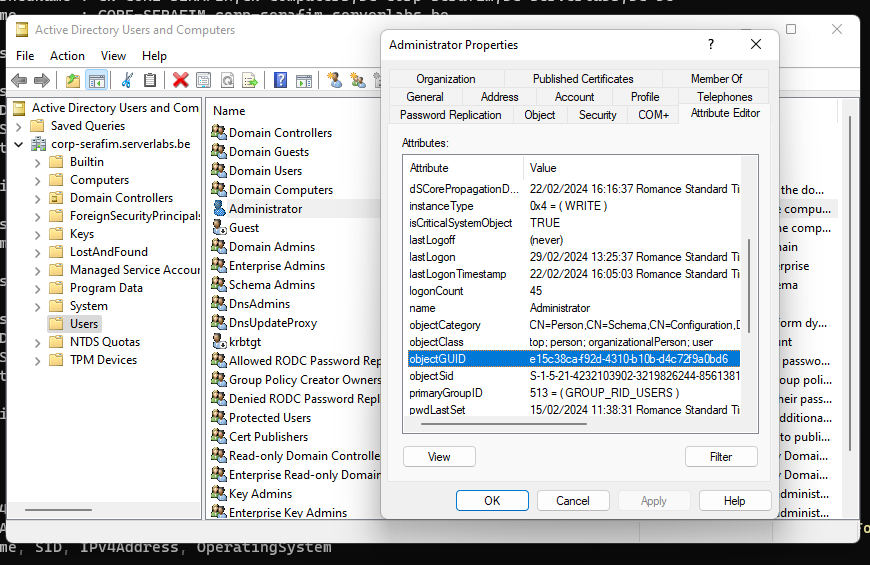
From now on boot at least the GUI server and the Windows 11 client machine and perform the following tasks from Win11. The core server is – for now - not needed.

1. (Recap question as it is important ) Open the ADU&C console (fastest method is to use **run** > **dsa.msc**) and choose the “advanced features view”. Select the built-in domain administrator account. The custom view gives you a few extra tabs. Which tab should you click to see all attributes of the account?

Attribute Editor

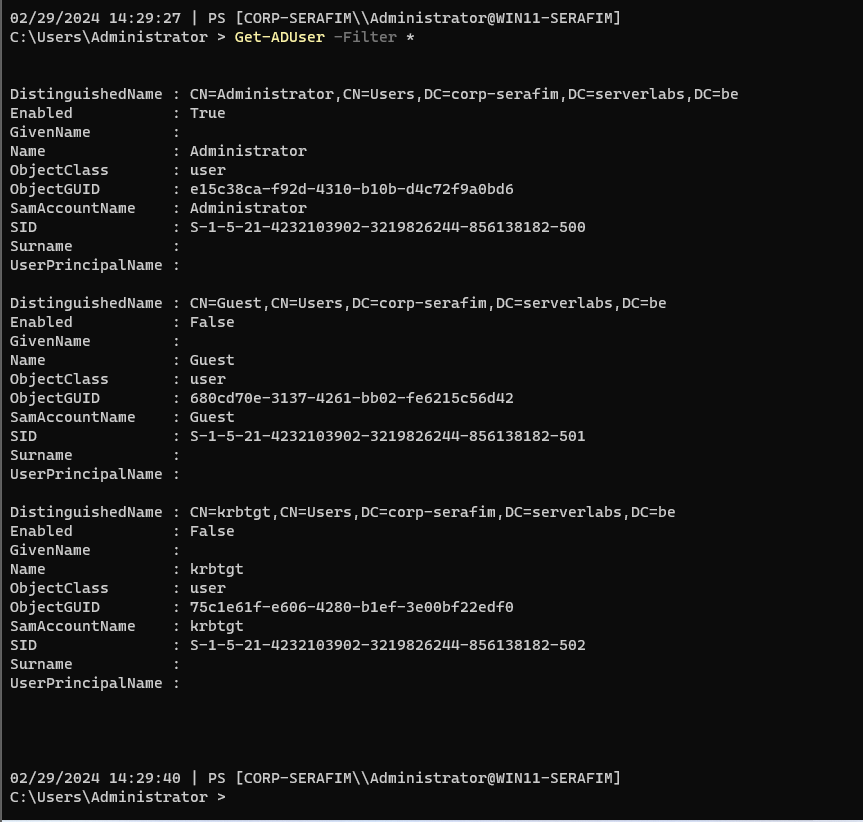


1. Click on this tab. You will find that many attributes have no value. Make sure that only see the attributes that have a filled-in value. Put a screenshot here of the GUID of the built-in domain administrator account.



1. Use a PS-cmdlet to request a list of all domain users. Make sure that your cmdlet does not require any extra input.

Get-ADUser -Filter \*

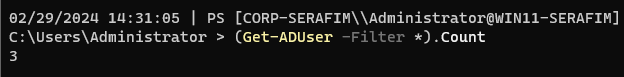


1. If you have used the correct cmdlet, you will notice that (unless specified otherwise) a specific subset of so-called **default properties** are displayed from each domain user. Is “the number of times a domain user logged in” a default property?

No, it is not

1. Use a PS-cmdlet to show the **number** of domain users.

(Get-ADUser -Filter \*).Count



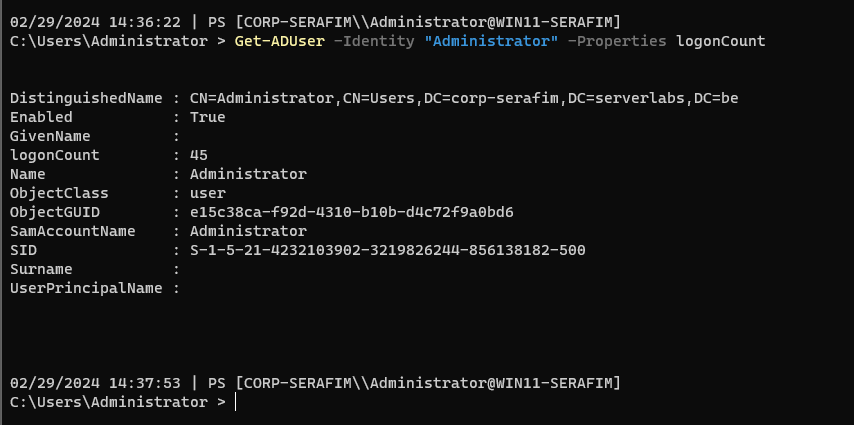
1. Use a PS-cmdlet to retrieve **all** properties (default and extended properties) of the built-in domain administrator account.

`Get-ADUser -Identity "Administrator" -Properties \*`



1. Edit your previous command and find out how many times the built-in domain administrator account has already logged in (this may take a moment).

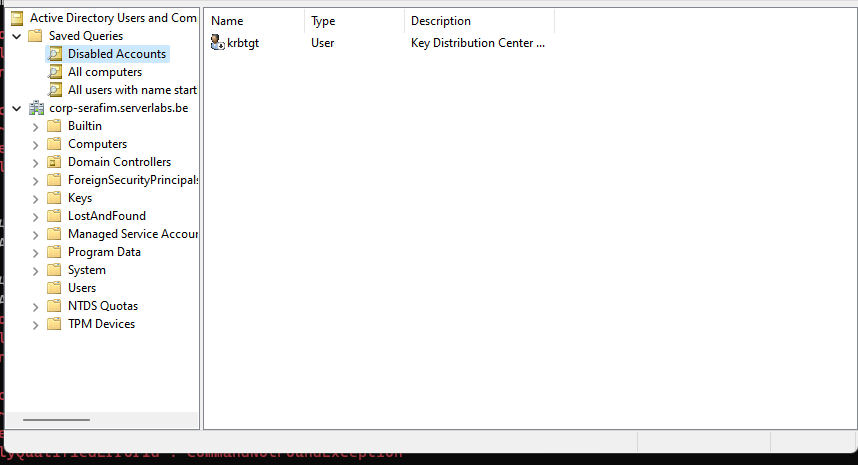
`Get-ADUser -Identity "Administrator" -Properties logonCount`



1. As you may know, the guest account is disabled by default. Activate this account using a PS-cmdlet.

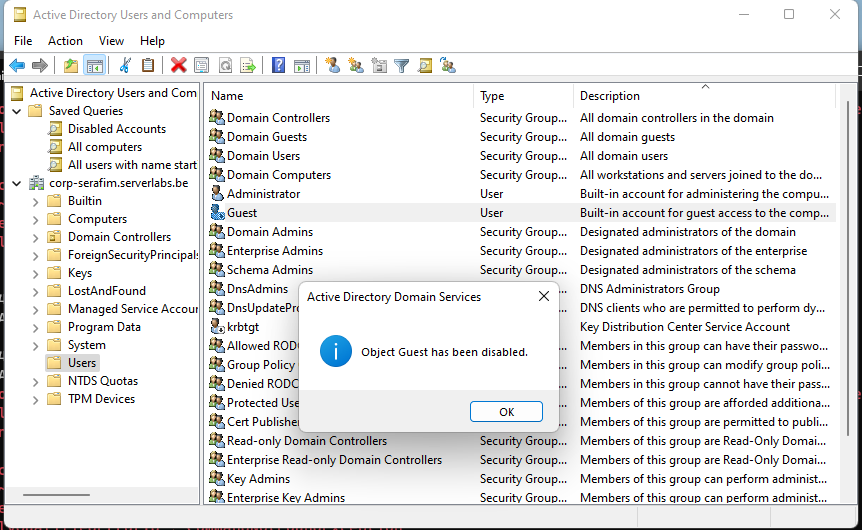
`Enable-ADAccount -Identity "Guest"`

1. Using the GUI, verify that your command from the previous question was successful. Afterwards disable the guest account in the GUI

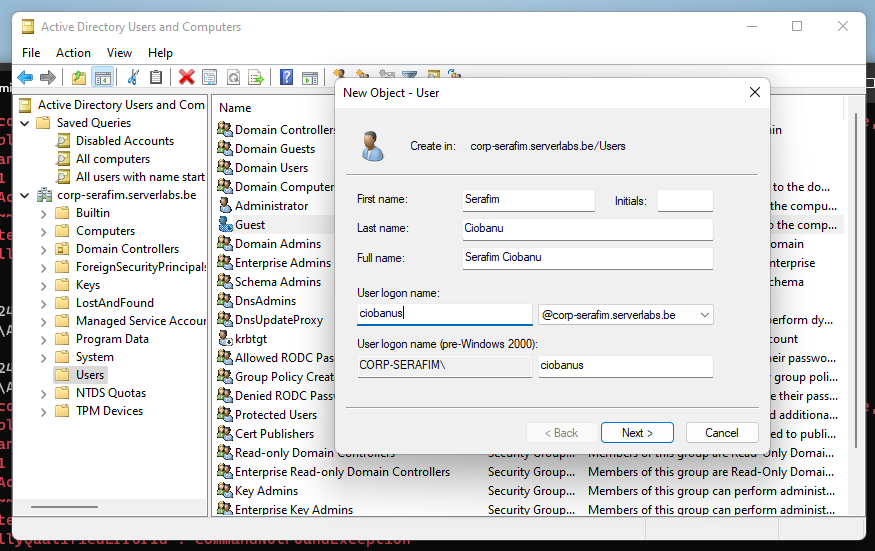


Users > Guest > Right-click > Disable user

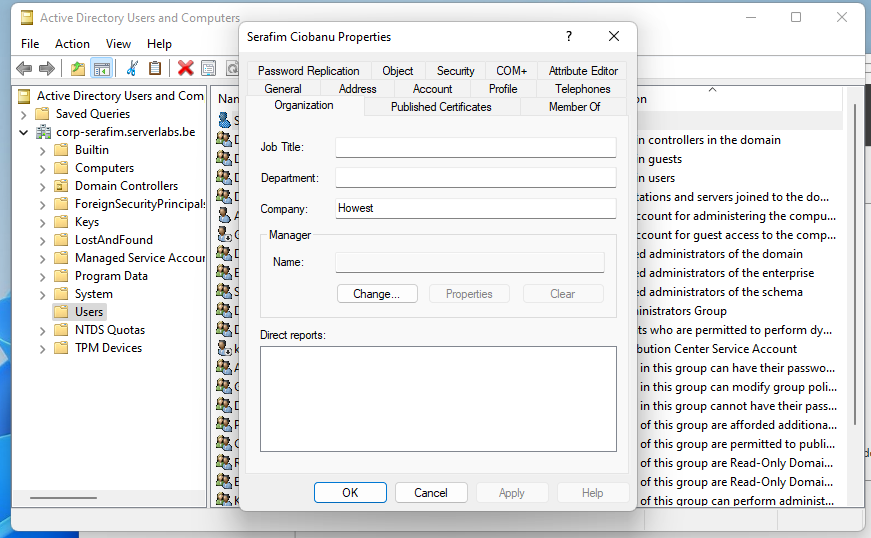
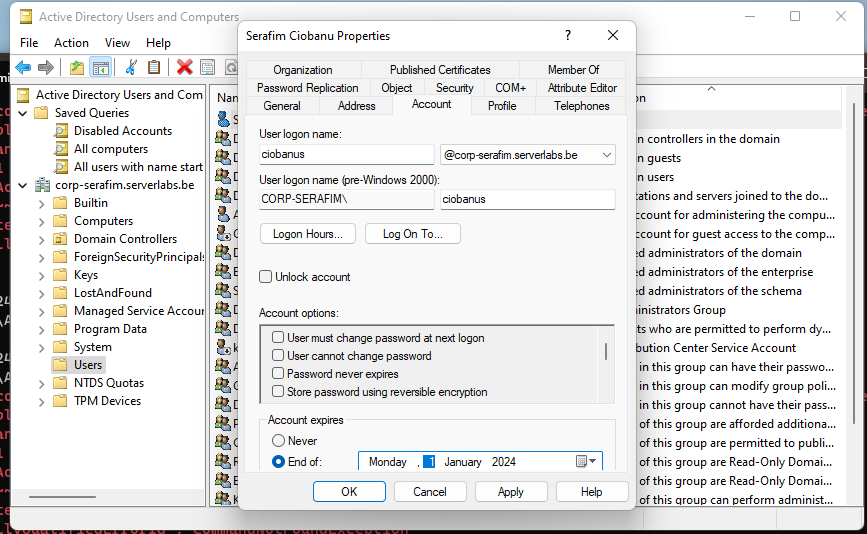
or also - `Disable-ADAccount -Identity "Guest"`



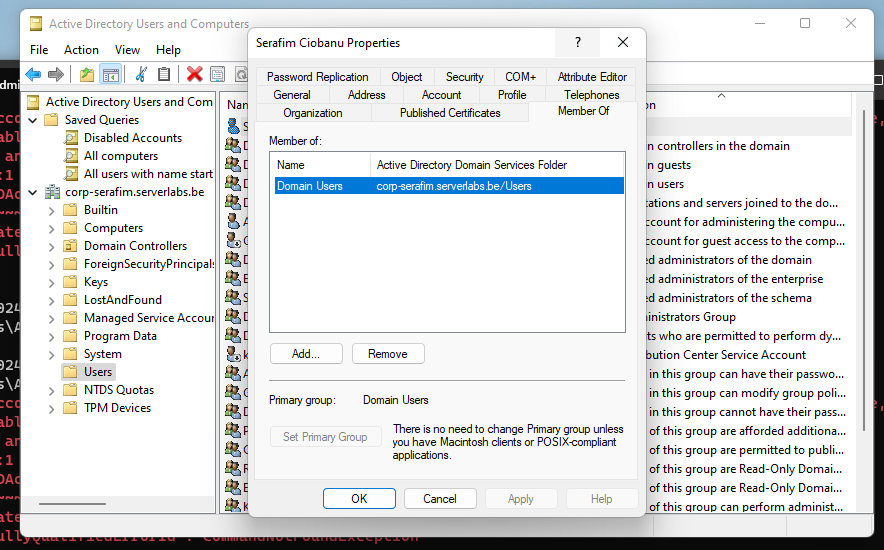
1. Using the GUI, create the following personal domain user with your own name:
   1. First name: <firstname> (ex: Thomas)
   2. Last name: <lastname> (ex: Clauwaert)
   3. Don’t use initials
   4. User logon name: your lastname + the first letter of your first name, no spaces or special characters (ex clauwaertt)
   5. Password: Friday13th!
   6. No need to reset password
   7. Make sure the account expires on the first day of 2024
   8. Company: Howest
   9. (Tip: You might get errors from your win11. Verify/do the magic on your GUI-server if needed)



I did not specify that the password will not expire.



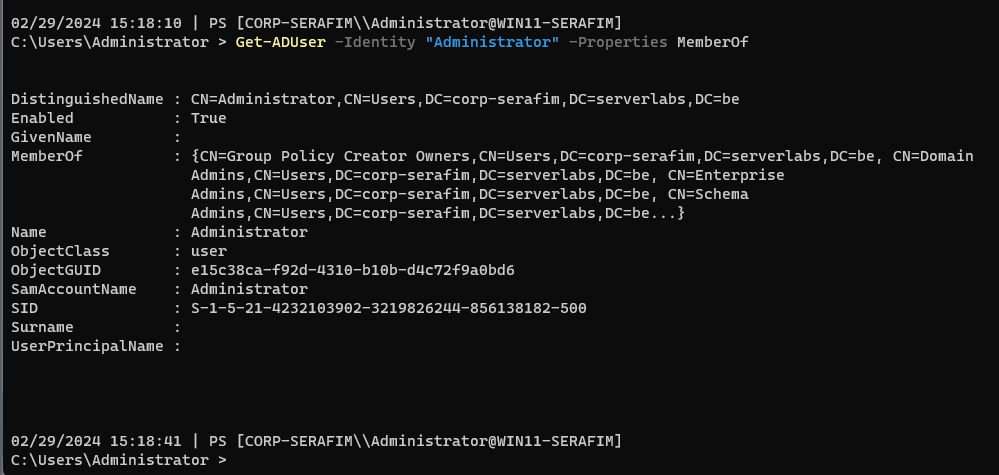
1. Use the GUI to verify your new user. Which groups does (s)he belongs to?



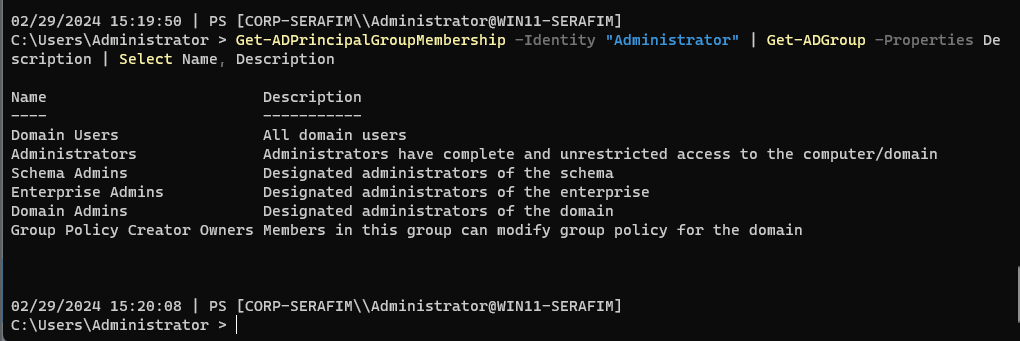
Domain Users

1. Show, using a PS-cmdlet, all groups of which the domain administrator account is a member.

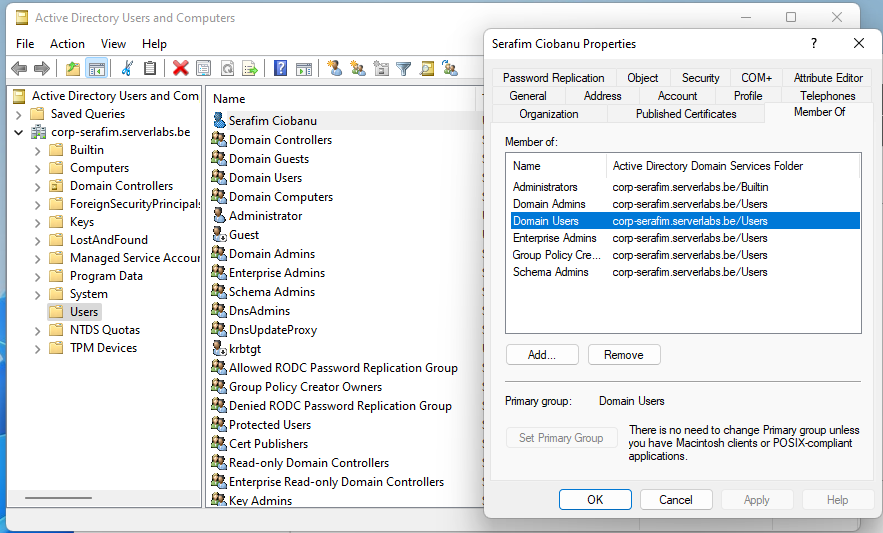
`Get-ADUser -Identity "Administrator" -Properties MemberOf`



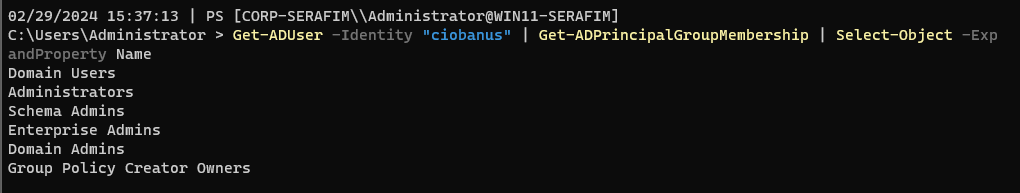
Or a better version - `Get-ADPrincipalGroupMembership -Identity "Administrator" | Get-ADGroup -Properties Description | Select Name, Description`



1. Use the GUI to make the new domain user part of all the groups you found in the previous question.



1. Use a PS-cmdlet to print all groups of the new domain user. Make sure the format is the full DN name of every Group.



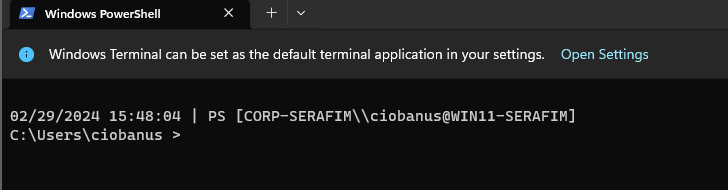
`Get-ADUser -Identity "ciobanus" | Get-ADPrincipalGroupMembership | Select-Object -ExpandProperty Name`

1. Sign out of your windows 11 client and sign in using your new personal domain account. Configure your custom prompt (refer to lab 3, question 18).

`New-Item -Path $profile -Type File -Force`

`notepad $PROFILE`

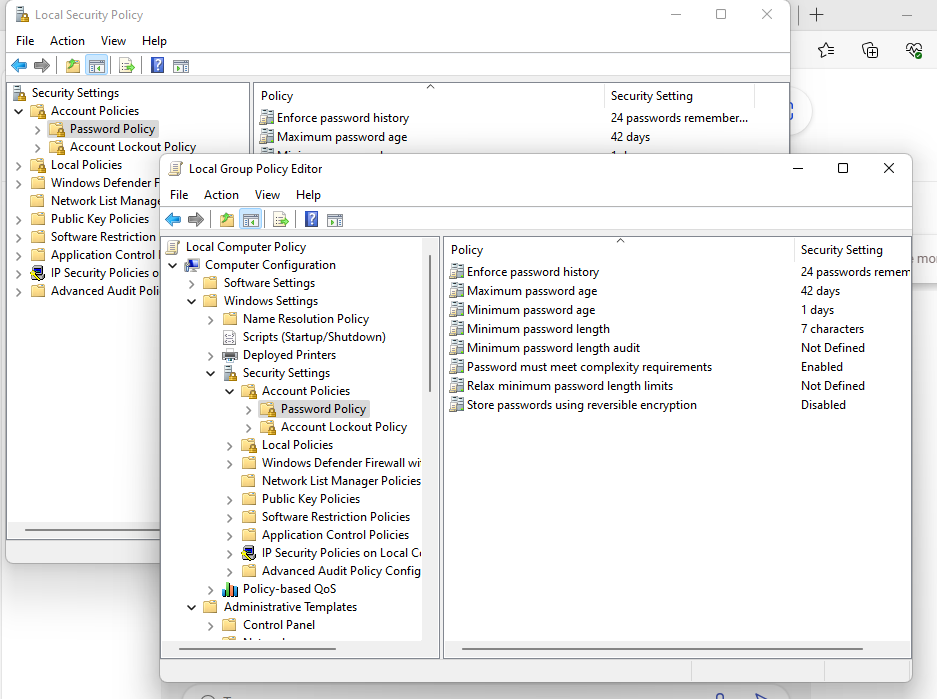
function prompt {"`n$(get-date) | PS [$Env:userdomain\\$Env:username@$Env:computername] `n$($PWD.ProviderPath) > "}



1. One of the advantages of using a domain is that you can also manage (group) policies, such as password policies for all domain users. On your windows 11 client there are 2 tools to manage group policies:
   1. The **local group policy editor** – this one is typically not used in a domain setting
   2. **Group Policy Management** – use this one to manage group policies in the domain

Open (a) the local group policy editor and search for a policy that indicates the minimum password length of local users on the machine. Paste a screenshot of the window where you found this policy.

Win + R > gpedit.msc

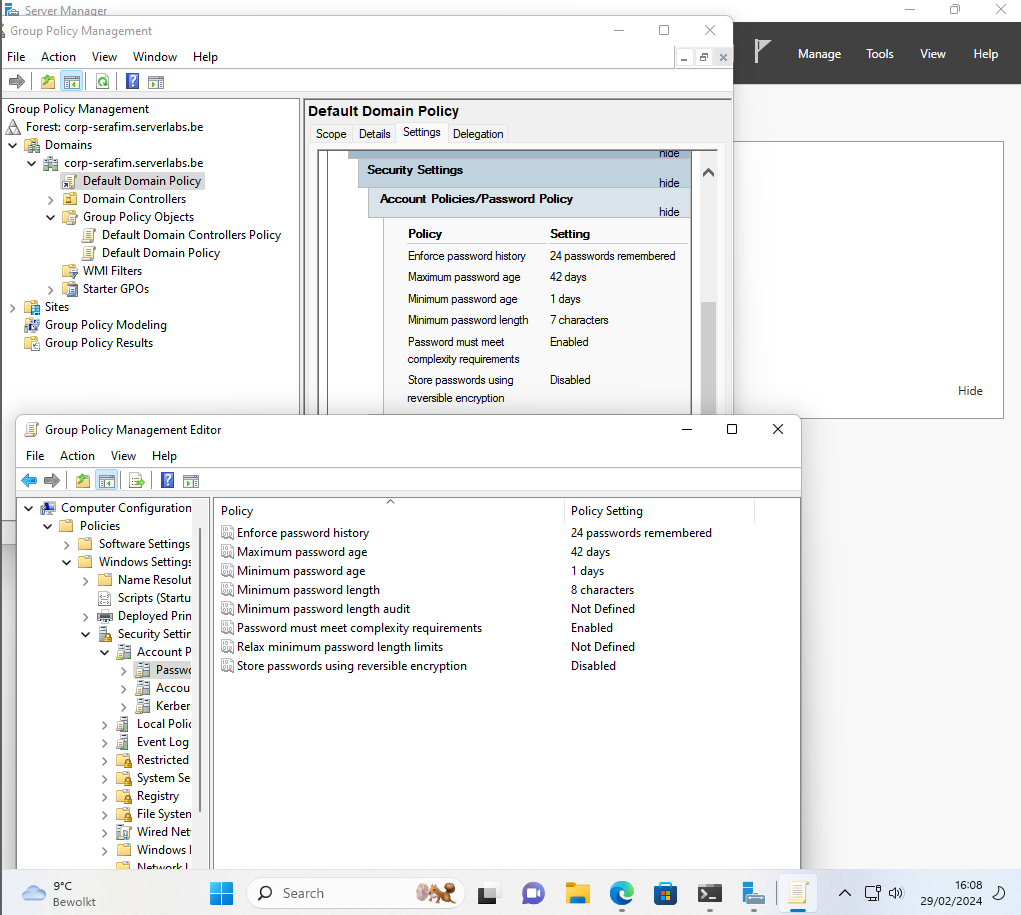
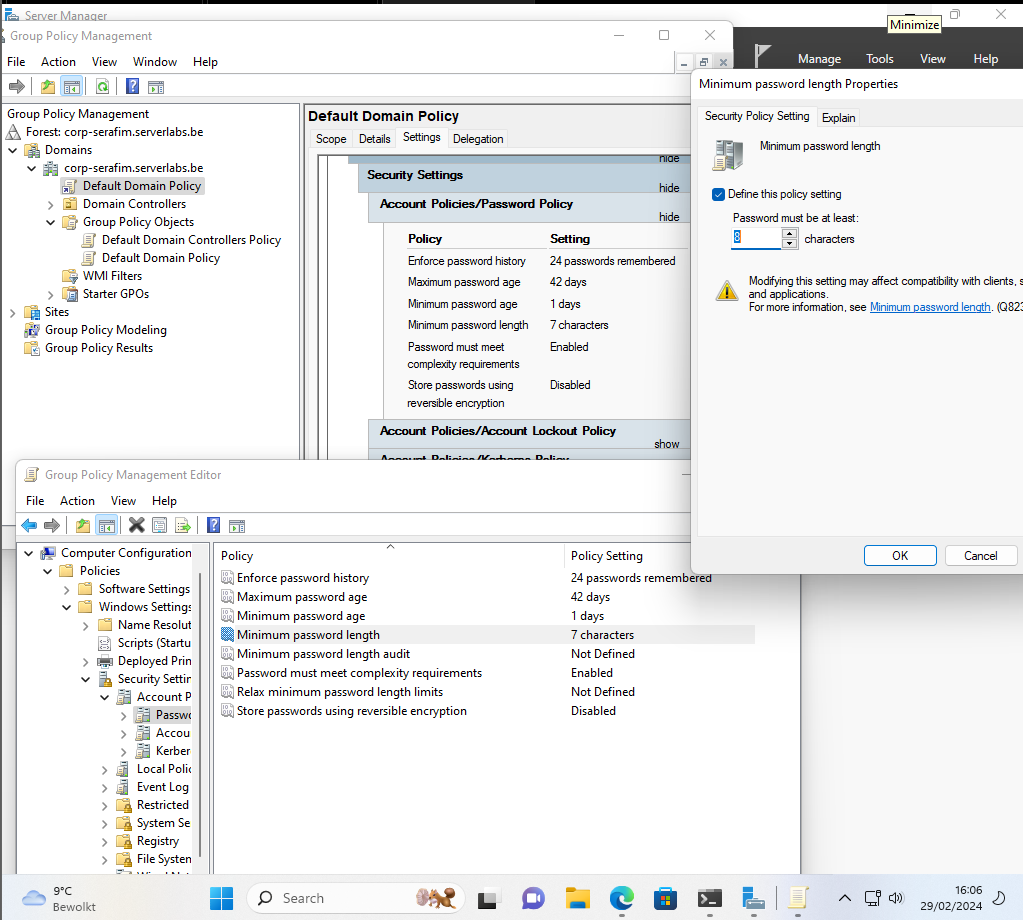


1. Try to adjust the minimum password length in this tool. Does this work?

Nope, does not work

1. Now open the correct tool and try to change the minimum password length from 7 to 8. Paste a screenshot below to show where you changed this and that the length has changed to 8.

Server Manager > Tools > Group Policy Management > Domains > my domain > Default Domain Policy > Settings > Lots of unfolding > Right-click > Edit > Windows Settings > Security > Account and etc.

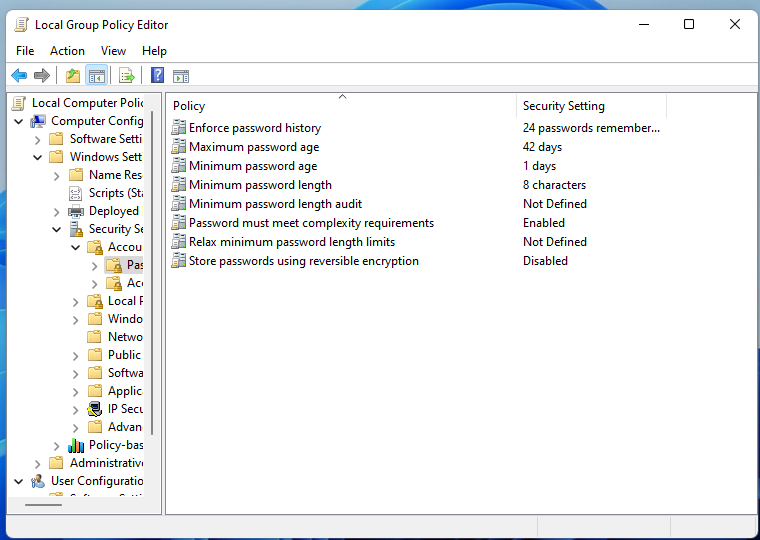
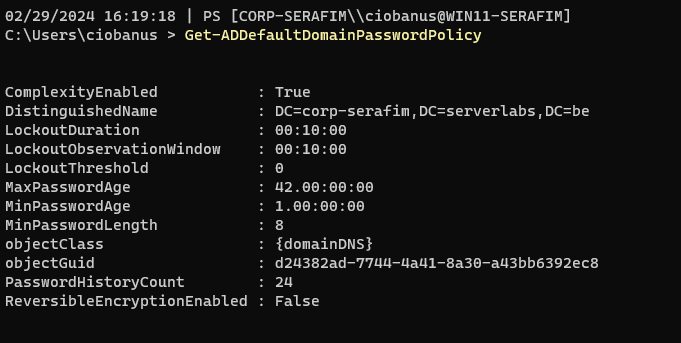


1. Since this policy is a computer configuration setting, it will not be applied until you restart your computer. However you can do it immediately by running the following legacy command

**gpupdate /force**

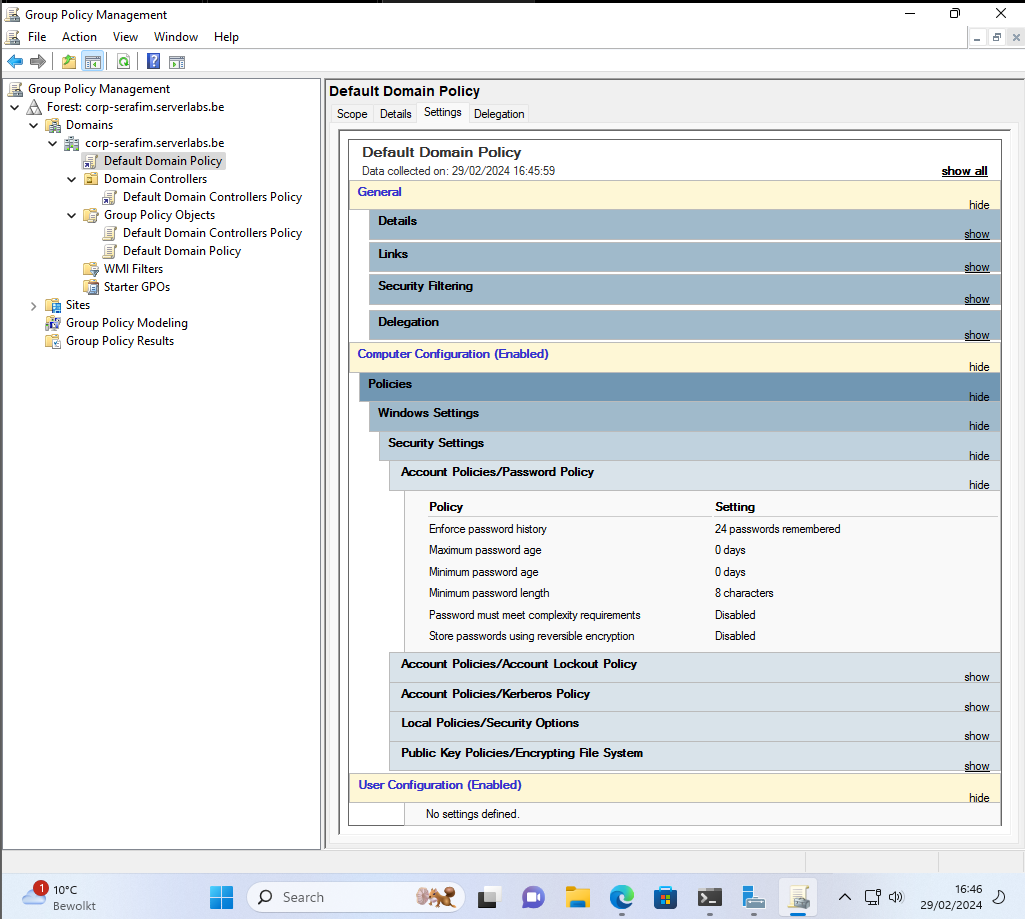
Now open the local group policy editor and check if the minimum length is set to 8 there as well. If not, restart your windows 11 virtual machine and test again. Finally, use a PS-cmdlet to request the default domain password policy. The result of this cmdlet should also show the desired 8. This might take some time and/or reboots.

`Get-ADDefaultDomainPasswordPolicy`

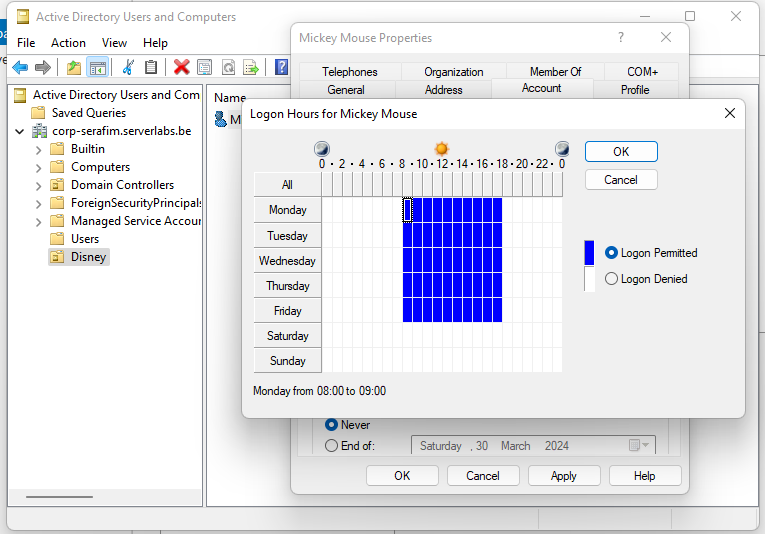
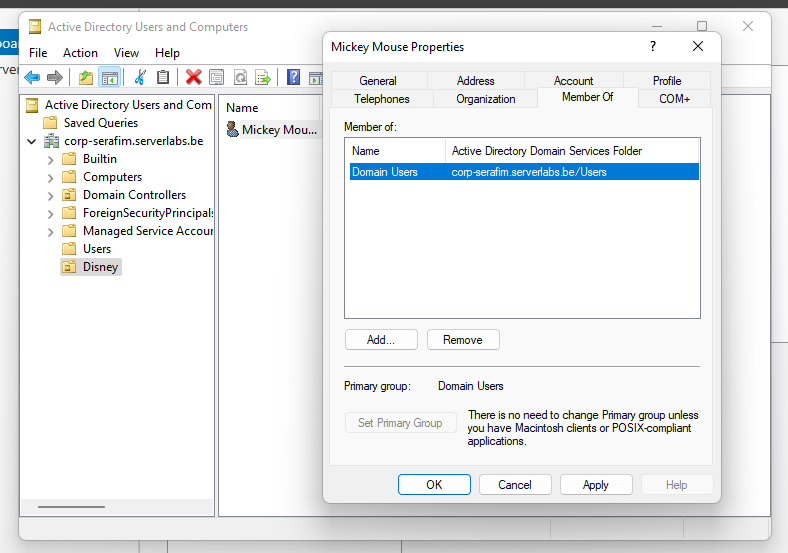
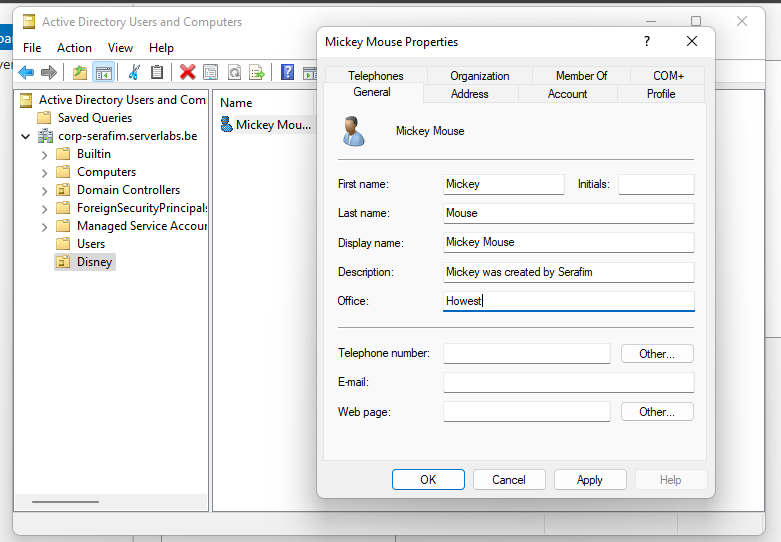
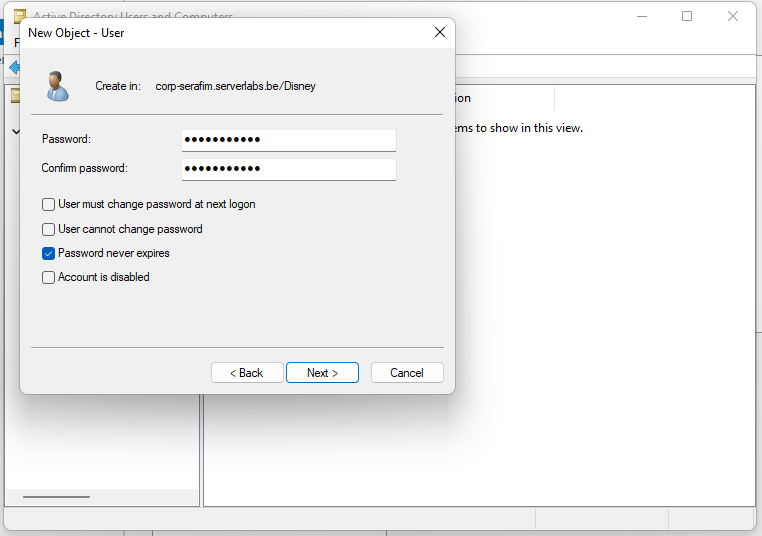
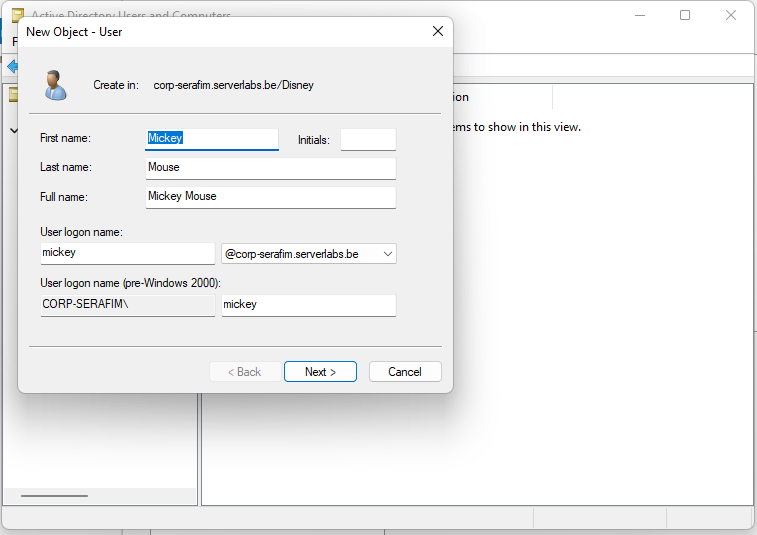


1. Because this is a test environment we will make things easier for us by reducing the complexity of passwords. This is obviously not a best practice, and should not be done in a production environment/real life scenario! In other words use a PS-cmdlet to change the default password policy in such a way that complex passwords are not needed and the minimum and maximum age is not applicable/defined.

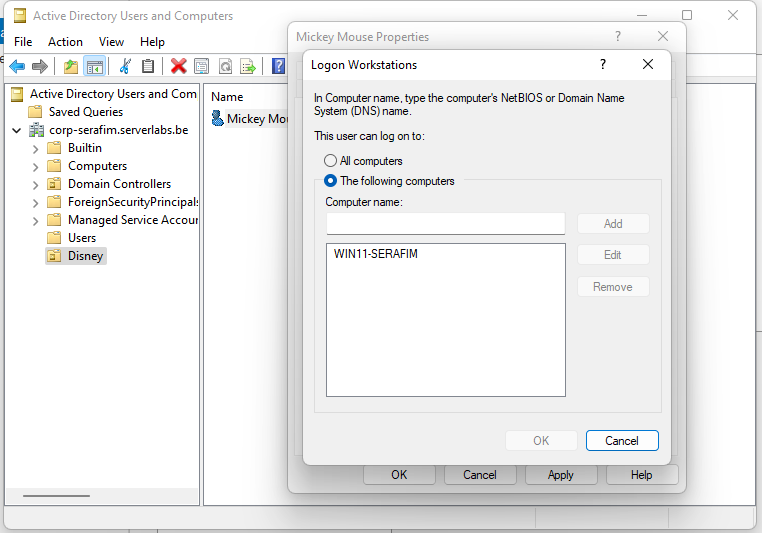
`Set-ADDefaultDomainPasswordPolicy -Identity "corp-serafim.serverlabs.be" -MinPasswordAge 0 -MaxPasswordAge 0 -ComplexityEnabled $false`



1. Use the AD U&C tool (right click on your corp-<name>.serverlabs.be) to create a new Organizational Unit (OU) with the name **Disney** and create a new domain user account with the following properties in this OU:
   1. First name: Mickey
   2. Last name: Mouse
   3. No initials
   4. User logon name: mickey
   5. Password: Friday13th!
   6. No need to change the password
   7. Password never expires
   8. Description: “Mickey was created by <firstname>” (replace <firstname> by your own name)
   9. Office: Howest
   10. Make sure Mickey is only allowed to log in on the windows 11 client and on workdays from 8h till 18h.
   11. Make sure he is a member of the domain users group.



*I do not know if* Saturday and Sunday is considered a word day, so I put it from Monday till Friday.

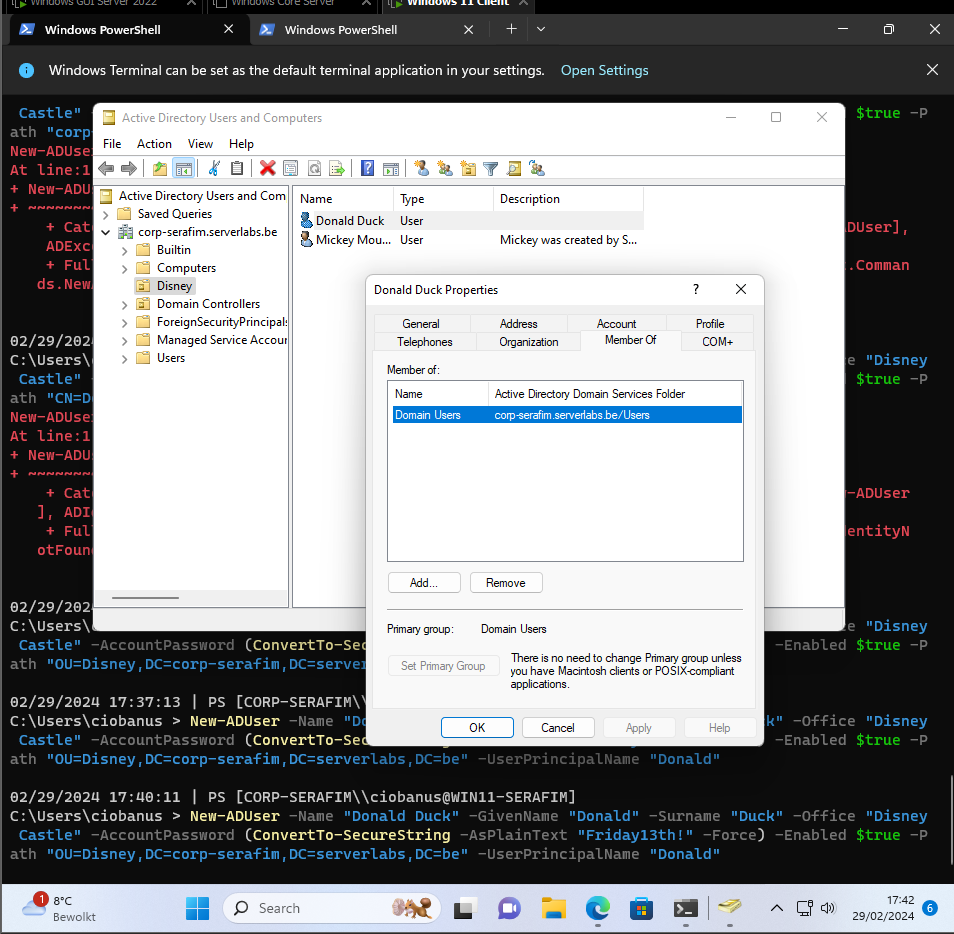
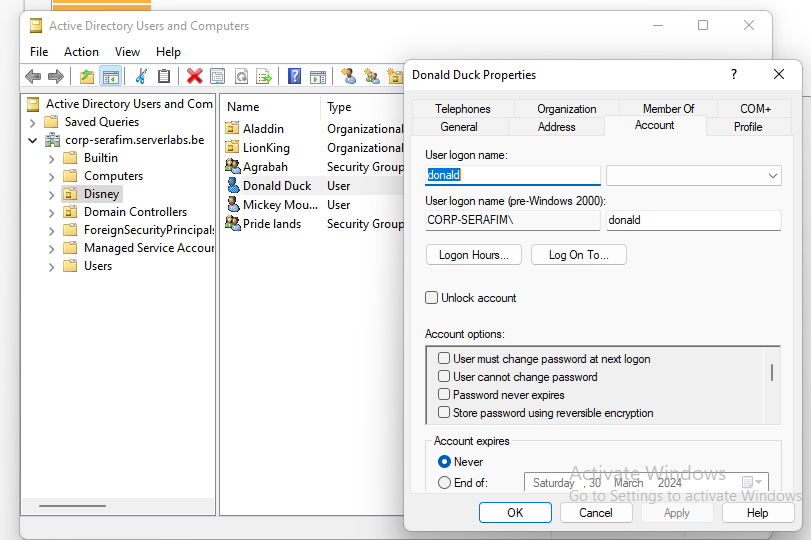
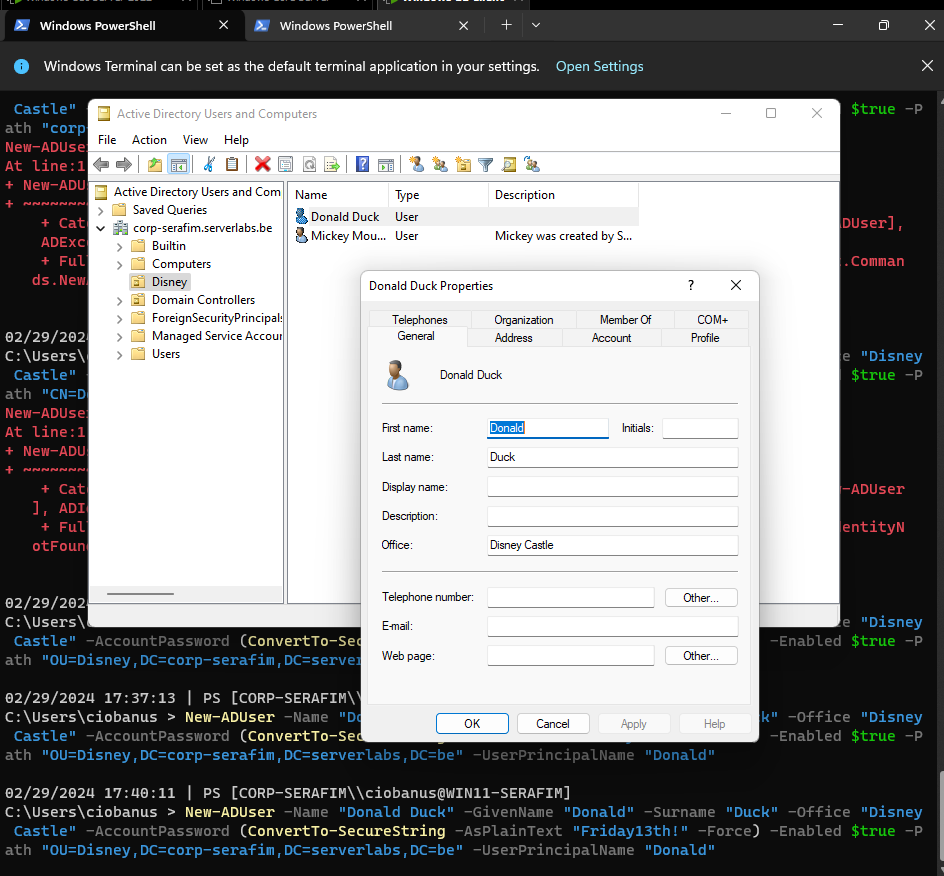


1. Change the following settings for the mickey user using a single PS-cmdlet.
   1. Change the Office to “Disney Castle”
   2. Add a title: “King”

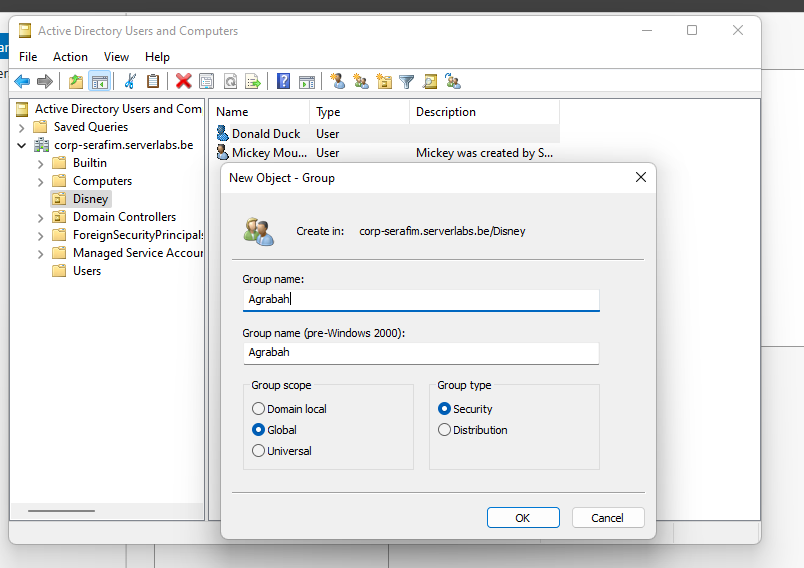
`Set-ADUser -Identity "mickey" -Office "Disney Castle" -Title "King"`

1. Create a new domain user account with the following attributes, using a single (non-interactive) PS-cmdlet:
   1. First name: Donald
   2. Last name: Duck
   3. Logon name: Donald
   4. Office: Disney Castle
   5. OU: Disney
   6. Account: enabled
   7. Part of the domain users group
   8. You can use the following parameter to set the password correctly:   
      **“-AccountPassword (ConvertTo-SecureString -AsPlainText “Friday13th!” -Force)”**

New-ADUser -Name "Donald Duck" -GivenName "Donald" -Surname "Duck" -Office "Disney Castle" -AccountPassword (ConvertTo-SecureString -AsPlainText "Friday13th!" -Force) -Enabled $true -Path "OU=Disney,DC=corp-serafim,DC=serverlabs,DC=be" -UserPrincipalName "donald" -SamAccountName "donald"



1. Create a new global security group called “Agrabah” in the Disney OU using the GUI.



1. Create a new global security group called “Pride Lands” in the Disney OU using a PS-cmdlet (so “Pride lands” and “Agrabah” should be in the same container/at the same level).

`New-ADGroup -Name "Pride lands" -Path "OU=Disney,DC=corp-serafim,DC=serverlabs,DC=be" -GroupScope Global -GroupCategory Security -DisplayName "Pride lands"`



1. Write a PowerShell script that automatically creates the users, specified in the “disneyusers.csv” file, and adds them to the correct containers (OU “Aladdin” and/or OU “LionKing”, which are part of the “Disney” OU). You are free to pimp your script as you want (for example also the creation of the OU’s) but at the very least it should:
   1. Import the csv
   2. Pipe it to a foreach
   3. Fill in the correct attributes of the new-aduser cmdlet

Take a screenshot of the “Aladdin” OU with its users clearly visible, and paste it here

$users = Import-Csv -Path "C:\Users\ciobanus\Documents\lab04\disneyusers.csv" -Delimiter ";" # Adjust delimiter if necessary

if (-not (Get-ADOrganizationalUnit -Filter "Name -eq 'Disney'")) {

New-ADOrganizationalUnit -Name "Disney" -Path "DC=corp-serafim,DC=serverlabs,DC=be"

}

if (-not (Get-ADOrganizationalUnit -Filter "Name -eq 'Aladdin'")) {

New-ADOrganizationalUnit -Name "Aladdin" -Path "OU=Disney,DC=corp-serafim,DC=serverlabs,DC=be"

}

if (-not (Get-ADOrganizationalUnit -Filter "Name -eq 'LionKing'")) {

New-ADOrganizationalUnit -Name "LionKing" -Path "OU=Disney,DC=corp-serafim,DC=serverlabs,DC=be"

}

foreach ($user in $users) {

try {

if ($user.ParentOU -match "Aladdin") {

$ouPath = "OU=Aladdin,OU=Disney,DC=corp-serafim,DC=serverlabs,DC=be"

} else {

$ouPath = "OU=LionKing,OU=Disney,DC=corp-serafim,DC=serverlabs,DC=be"

}

# Convert string representations to boolean

$isEnabled = $user.Enabled -eq '$true'

$changePassword = $user.mustChangePassword -eq '$true'

New-ADUser -Name $user.Name -GivenName $user.Givenname -Surname $user.sn -SamAccountName $user.samAccountName -UserPrincipalName $user.'E-mail' -EmailAddress $user.'E-mail' -Path $ouPath -AccountPassword (ConvertTo-SecureString $user.Password -AsPlainText -Force) -Enabled $isEnabled -ChangePasswordAtLogon $changePassword -Description $user.Description

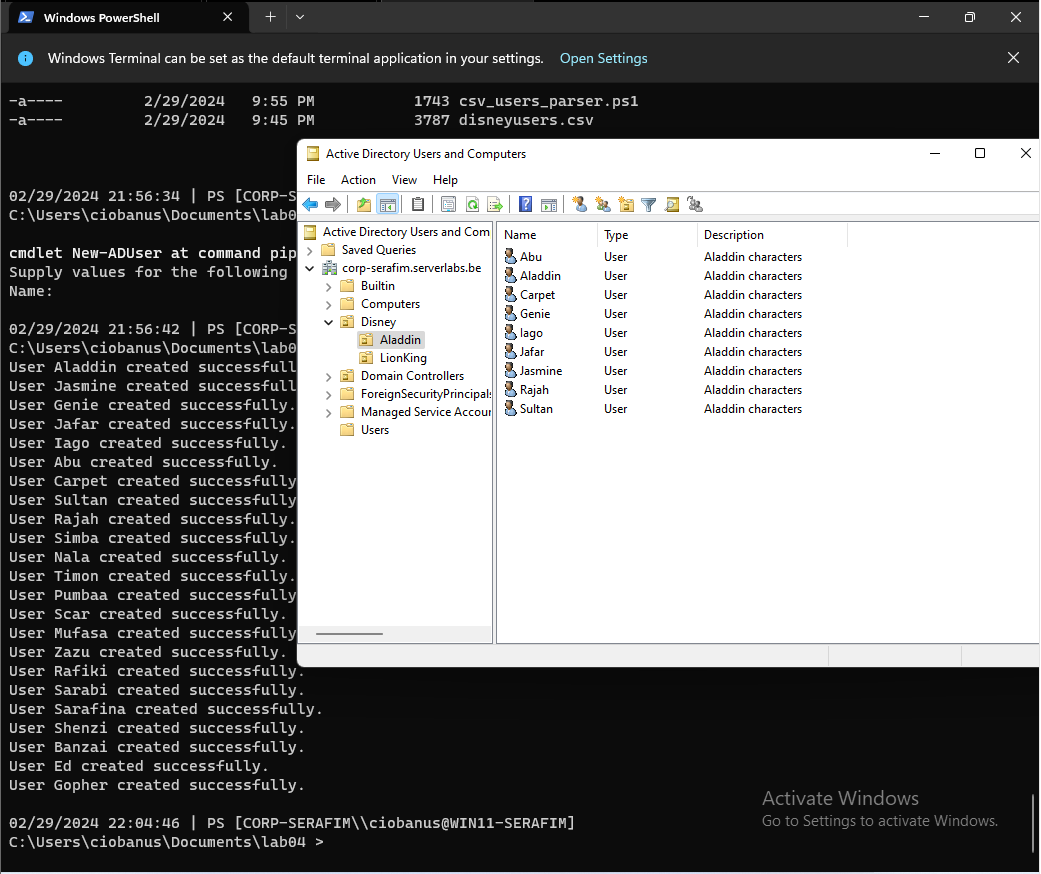
Write-Host "User $($user.Name) created successfully."

} catch {

Write-Warning "Failed to create user $($user.Name): $\_"

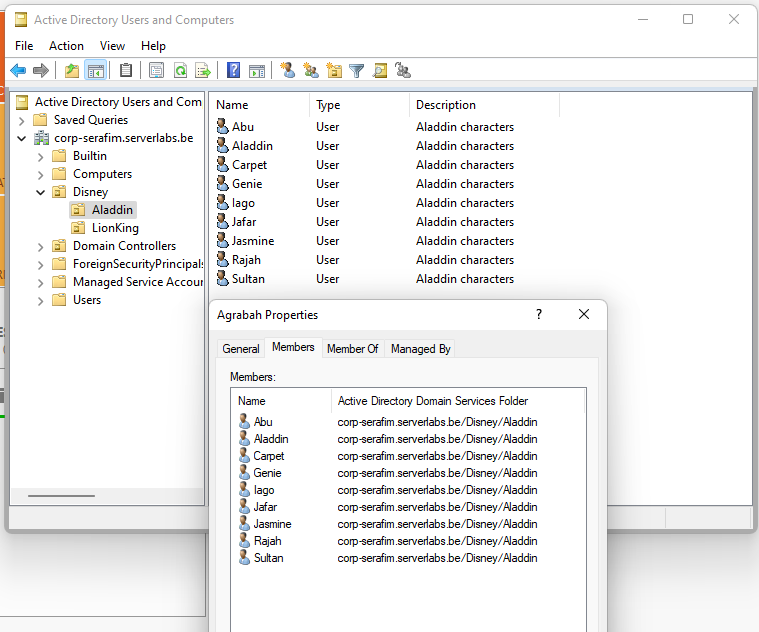
}

}

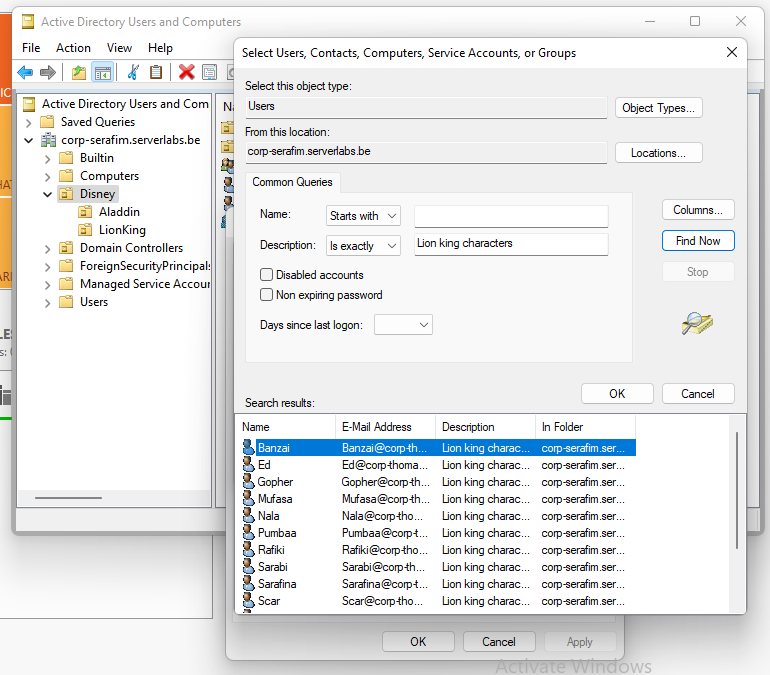


1. Use a PS-cmdlet (and piping) to add all users in the Aladdin OU to the Agrabah Security group. Tip: A possible solution is to enumerate and filter with get-aduser and pipe it to a foreach loop followed by adding the user to the group.

`Get-ADUser -Filter \* -SearchBase "OU=Aladdin,OU=Disney,DC=corp-serafim,DC=serverlabs,DC=be" | ForEach-Object { Add-ADGroupMember -Identity "CN=Agrabah,OU=Disney,DC=corp-serafim,DC=serverlabs,DC=be" -Members $\_ }`



1. Now add all the users from the “LionKing” OU to the “Pride Lands” security group, this time **using the GUI**. Do this for all LionKing users at once, not one by one! Tip: use the advanced option and search on the description (the value in the description field is the same for all LionKing users).

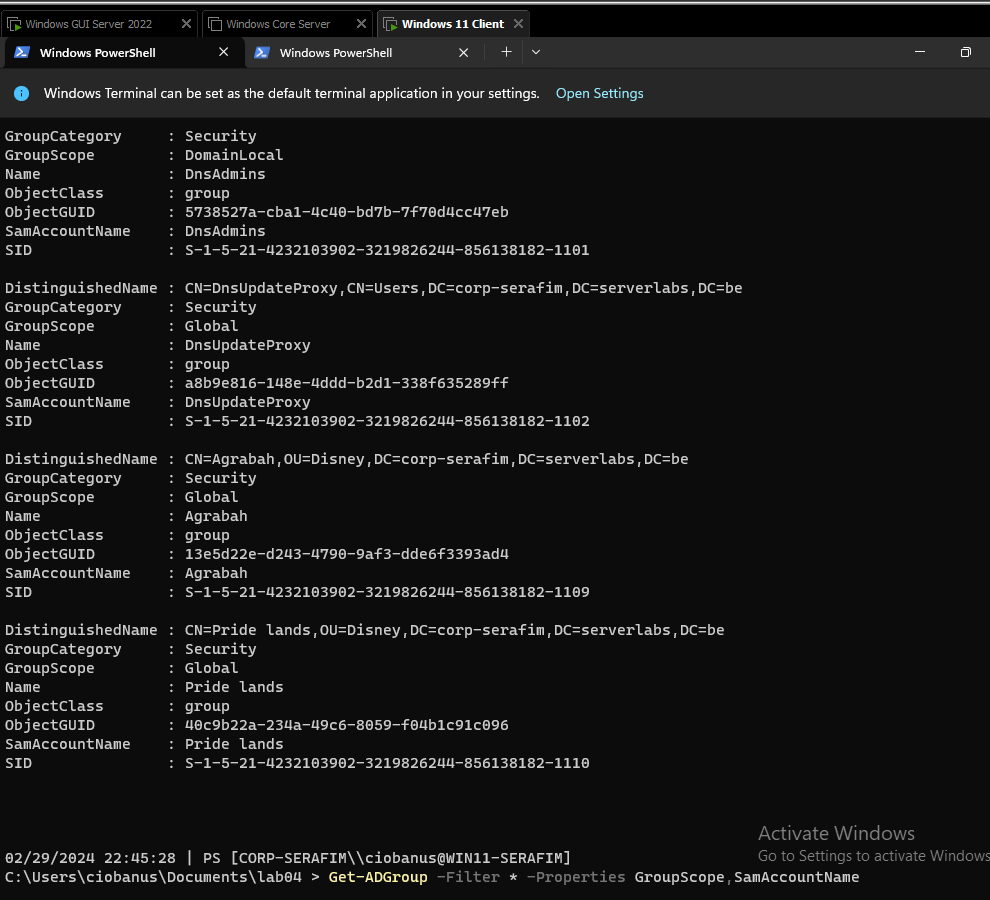


1. Use a PS-cmdlet to display the following items:
   1. All groups in your domain

`Get-ADGroup -Filter \*`

* 1. The sam account names and the scopes of all groups in your domain

`Get-ADGroup -Filter \* -Properties GroupScope,SamAccountName`



* 1. List below the 3 types of “group scopes” used by Windows and investigate the differences between these scopes (search online for their meaning)

There are 3 scopes – Universal, Global, Domain

Universal:

* Group can include as members accounts from any domain within the forest in which the universal group resides
* Group can be assigned permissions in any domain or forest.

Global:

* Group can include as members accounts from the same domain as the parent global group
* Member permissions can be assigned in any domain

Domain:

* Group can include as members accounts from any domain, global groups from any domain, Universal groups from any domain
* Member permissions can be assigned only within the same domain as the parent domain local group
  1. The sam account names and categories of all groups in your domain, what is the difference between the categories? Search online if needed.

`Get-ADGroup -Filter \* -Properties GroupCategory,SamAccountName`

There are 2 groups:

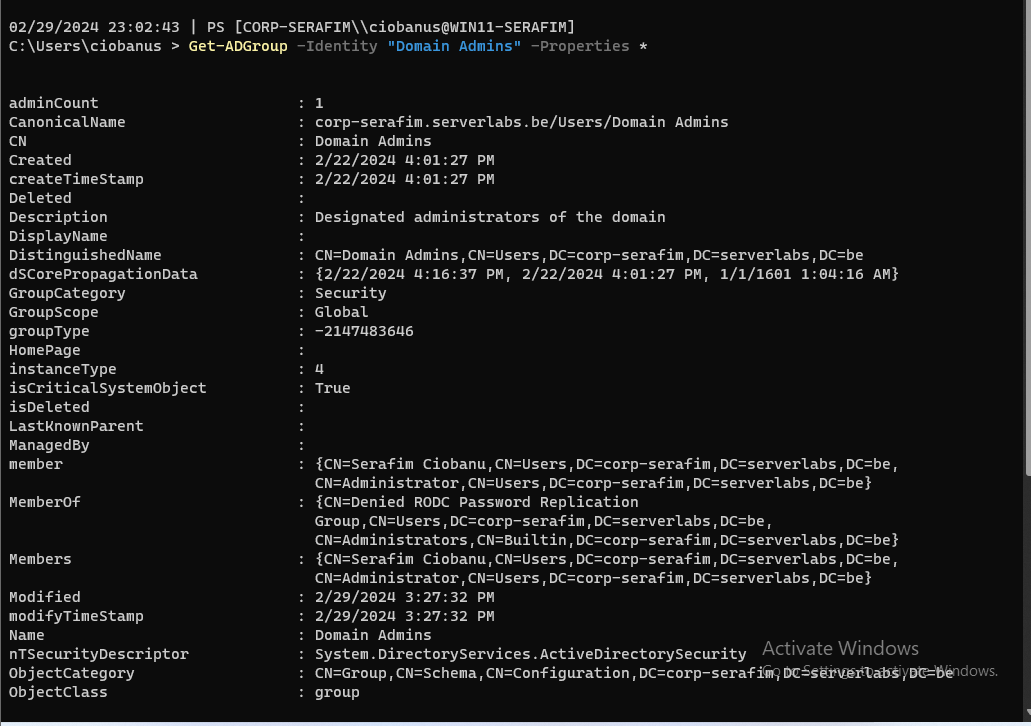
Security groups:

* Use to assign permissions to shared resources
* Assign user rights to security groups in AD
* Assign permissions to security groups for resources
* You can use distribution groups only to send email to collections of users by using an email application
* Distribution groups aren't security enabled

Distribution groups:

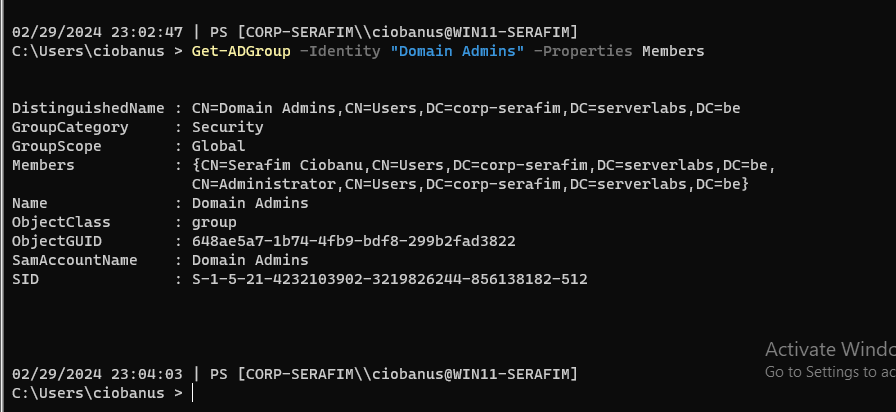
* Use to create email distribution lists
  1. All properties of the domain admins group

`Get-ADGroup -Identity "Domain Admins" -Properties \*`



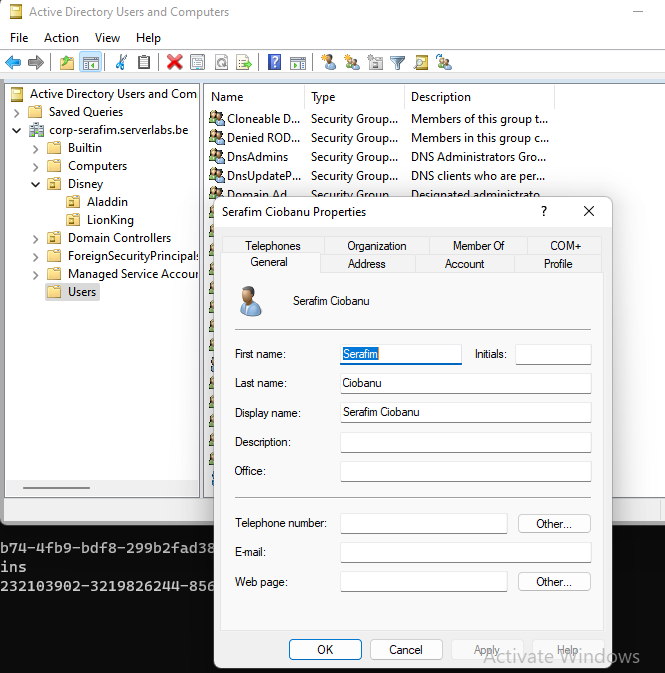
* 1. All members of the domain admins group

`Get-ADGroup -Identity "Domain Admins" -Properties Members`



* 1. What kind of names are typically shown of these members?

The display name? I do not really understand the questions, but the CN notation looks like the Display Name of a user



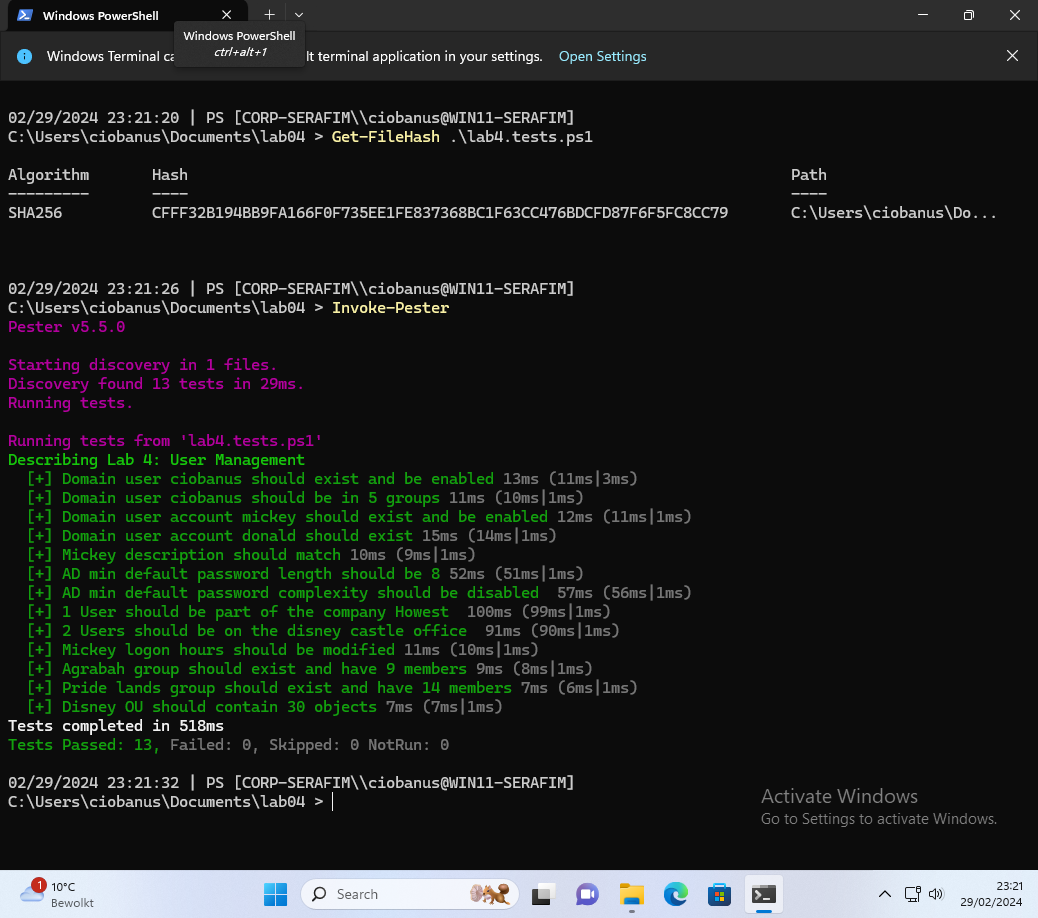
Otherwise, my answer can also be that you are showing the name (CN), the “organizational unit” (in this case – CN Users), and then the DC, meaning the domain which those users are part of.

1. Run the Pester Script and put a screenshot (or multiple ones if needed) below. To get a list of variables you can always open the script and read the first few lines. Update your $PROFILE file with a domainusername variable and the other ones as well if you didn’t already.

$PesterPreference = New-PesterConfiguration;

$PesterPreference.output.verbosity = 'Detailed';

$username="<firstname>" ; $domain="corp-<firstname>"; $domainusername="<lastname+firstletteroffirstname>"



## Extensions - Optional Assignments

* Redo the first part of the lab after adding all other uses & groups.
* Is it possible to create a local user on the Core server remotely?
* Explore ADSI Edit under tools in server manager.
* Try to perform an operation that enumerates AD information and sniff it with Wireshark. Can you find sensitive information?
* Create a script that shows the domain administrators which users are working the hardest on the network. You are free to determine how.