Building an Active Directory Lab - Part 2

blog.spookysec.net/ad-lab-2

29 May 2022

Hello Everyone! Welcome back to the “Building an Active Directory Lab” series - In this post we’re going to continue off of the foundation set in part 1. It’s been a while and I’ve had to rebuild my AD Lab, but it’s important to note, all the same principles still apply.

In this post, we’ll be exploring the following topics

- Creating, Managing, and Organizing User Accounts
- Creating Service Accounts
- Delegating Specific Rights to Users
- Joining PCs to the Domain
- Group Policy Objects (or GPO) Basics

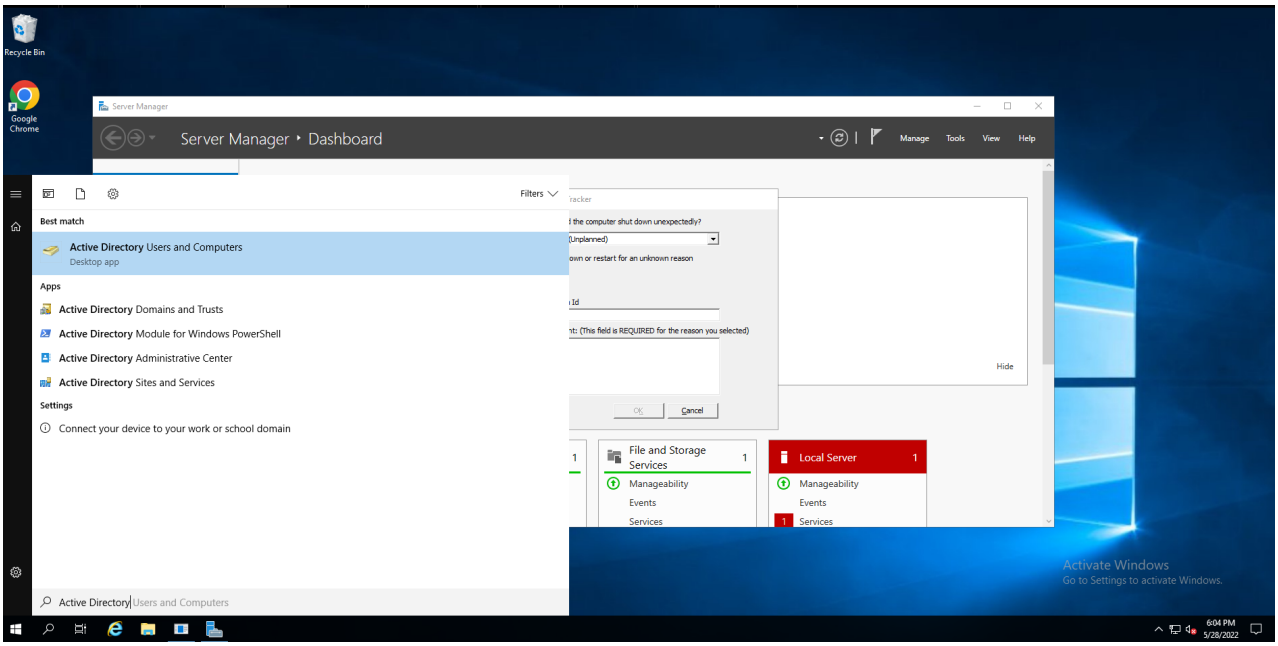
Let’s dive into it.

Creating, Managing, and Organizing User Accounts

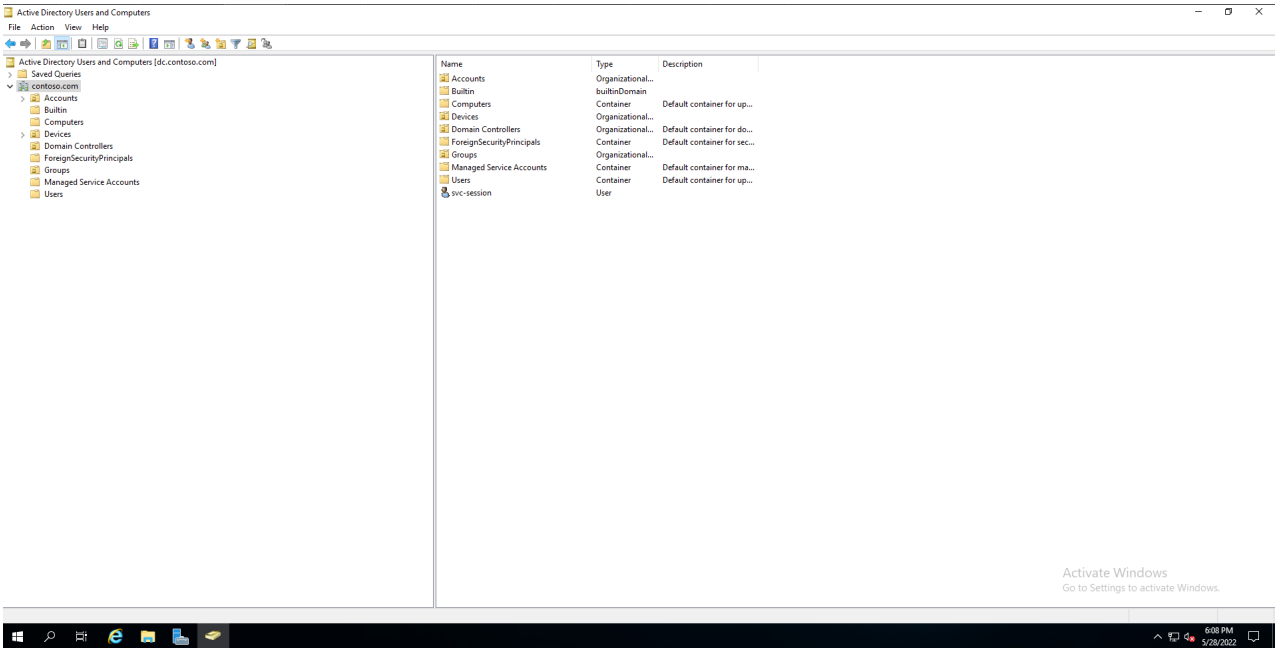
Organization is one of the most important aspects to managing an Active Directory Domain - Everything should be structured in a specific way, for a specific reason. **Organizational Units** are exactly as they sound - They allow you to organize Users, Groups, and Computers within an Active Directory domain.

Creating Active Directory Objects

Before we can create an Object in Active Directory, first we must open the “Active Directory Users and Computers” program. You can find this by typing the program name into the search bar *or* by clicking “Tools” and “Active Directory Users and Computers” in the “Server Manager” console.



After opening it, you will see something that looks like so.

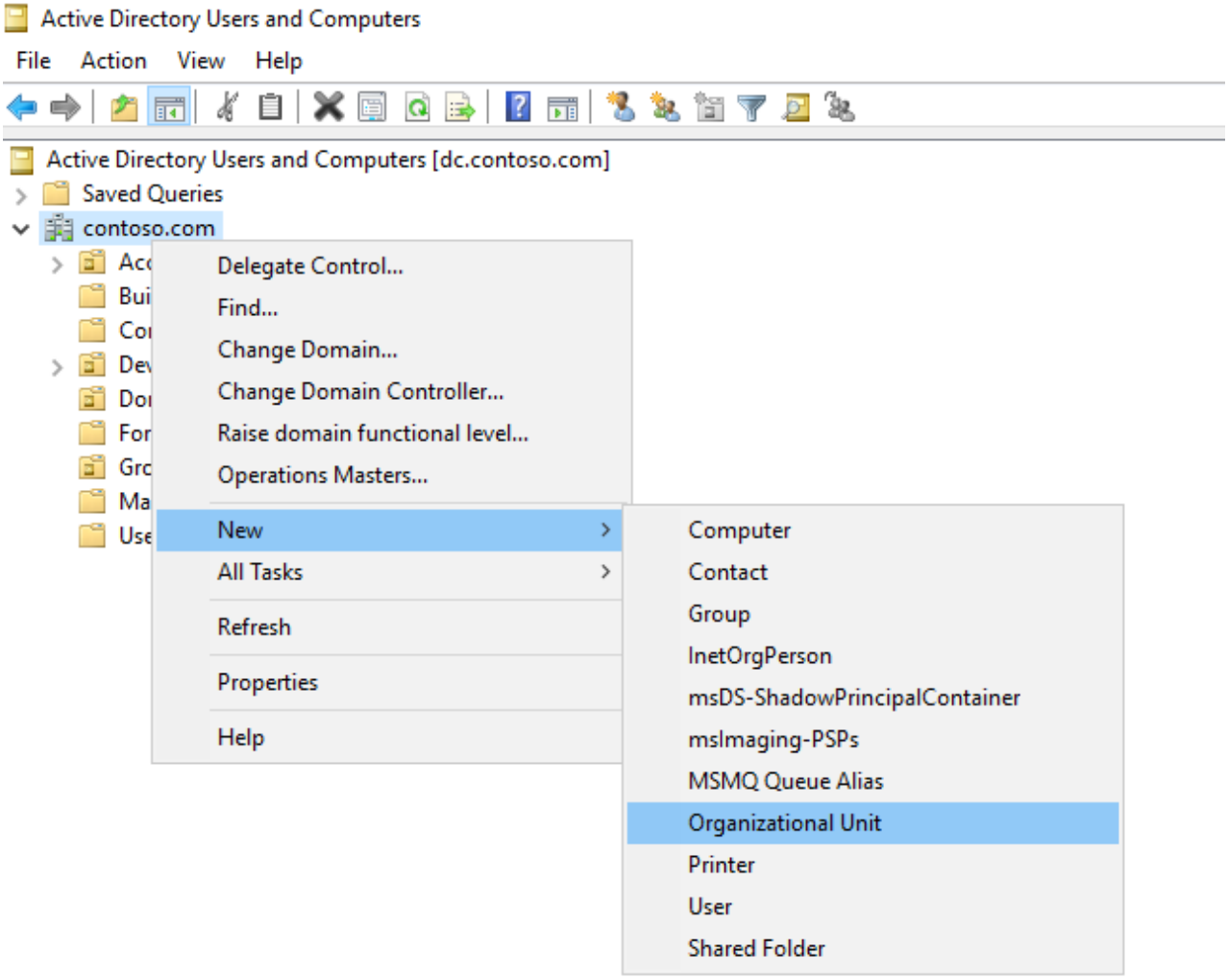


It's important to note that "Accounts", "Devices", and "Groups" are custom OUs that I have created. By default, they will not exist. By default there are a handful of OUs (Users, Managed Service Accounts, Foreign Security Principals, Domain Controllers, BuiltIn, and Computers) that have some pre-populated accounts and or groups. There are a ton of built in Groups and Users, that I can't explain the purpose of all of them. Here are some key Users and Groups that you should know about:

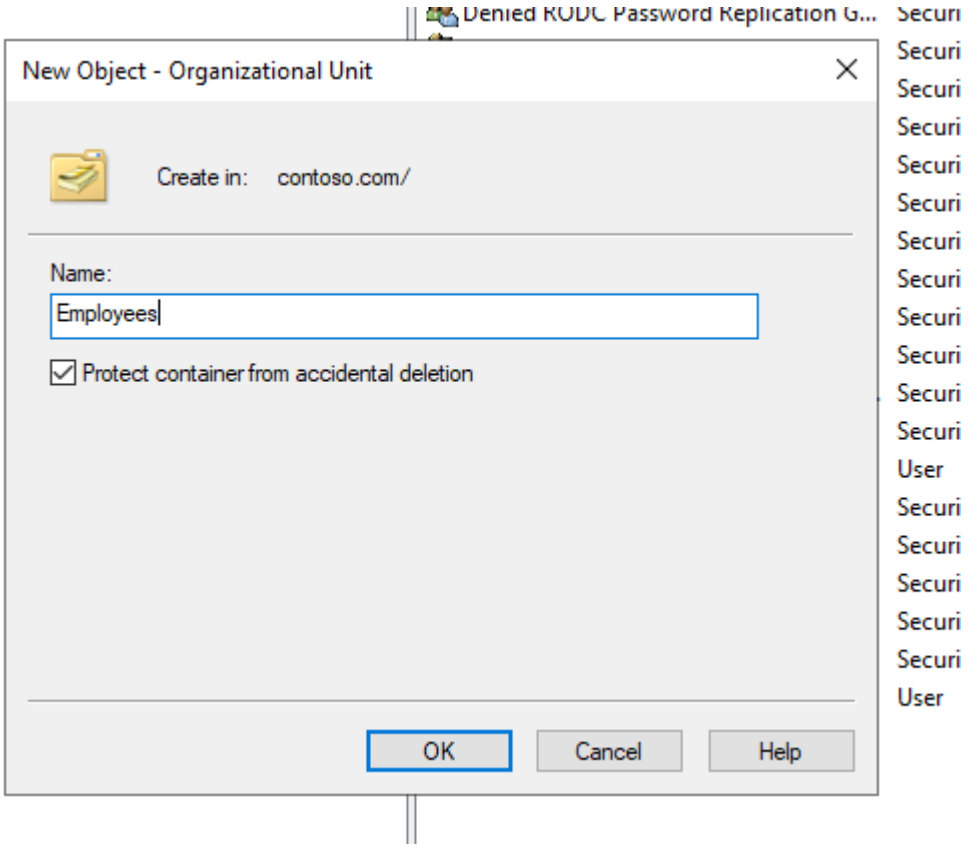
Users/Groups	Purpose
User - Administrator	The default User created when setting up an Active Directory Domain.

Users/Groups	Purpose
Group - Account Operators	This is a privileged Group that can reset non privileged User Accounts passwords
Group - Domain Admins	This is a privileged Group that allows for management of an Active Directory Domain.
Group - Enterprise Admins	This is an incredibly highly privileged Group that allows you to manage any Domains in the Forest.
Group - Schema Admins	This is another privileged Group that can allow editing of the LDAP schema. This could cause irreversible damage if you are an inexperienced user.

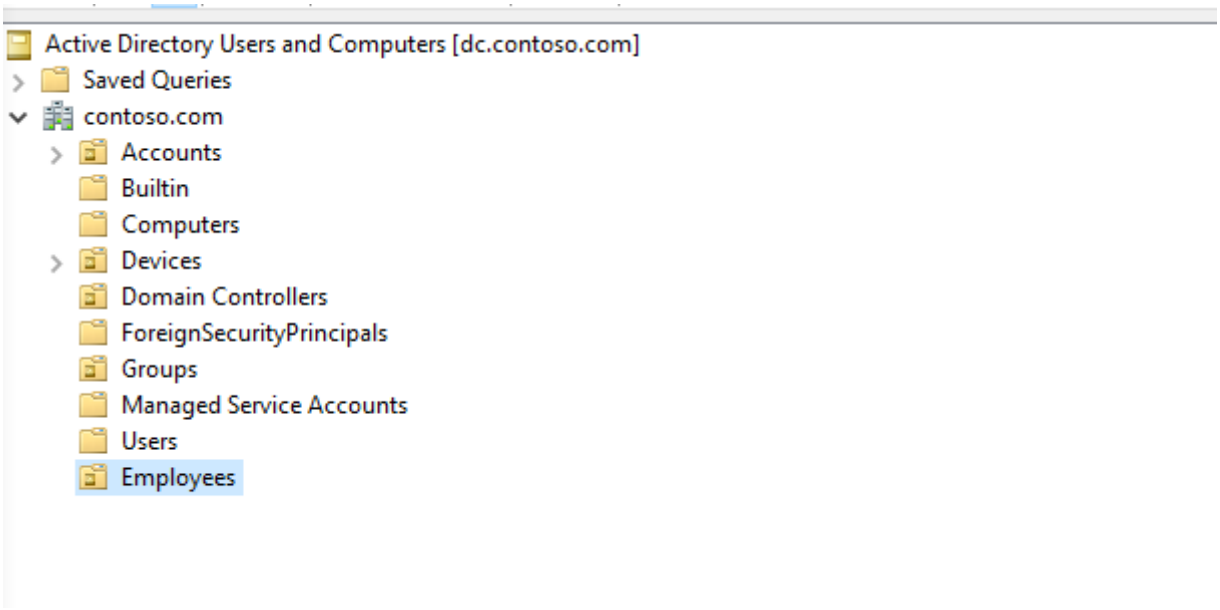
Now that we know a little bit more about some sensitive User and Groups, let's dive into creating a new Active Directory Object. Let's begin with a basic object - an Organizational Unit. To do so, select the root of the domain "yourdomain.com", then hover over "New". You should see a number of objects, some of them are interesting, some are not. In this Blog Post, we will primarily be focusing on Users, Organizational Units, and Groups.



After selecting Organizational Unit, you will be prompted to enter a new name. I will call the OU “Employees”. Within the “Employees” OU, I will create three Sub OUs - One for FTE (Full Time Employees), one for Contractors, and one for Vendors.

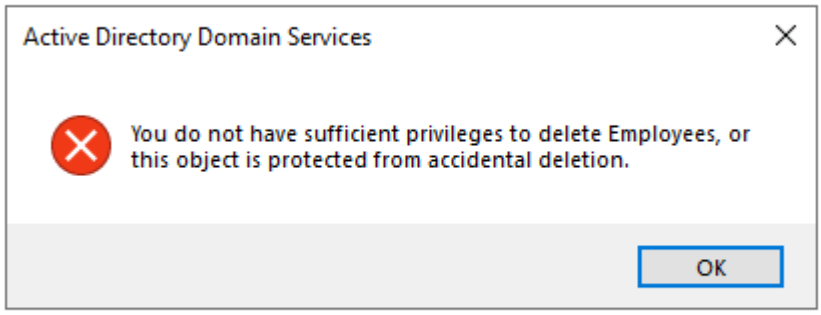


Once you are happy with your name, select “OK” and your new OU is created!

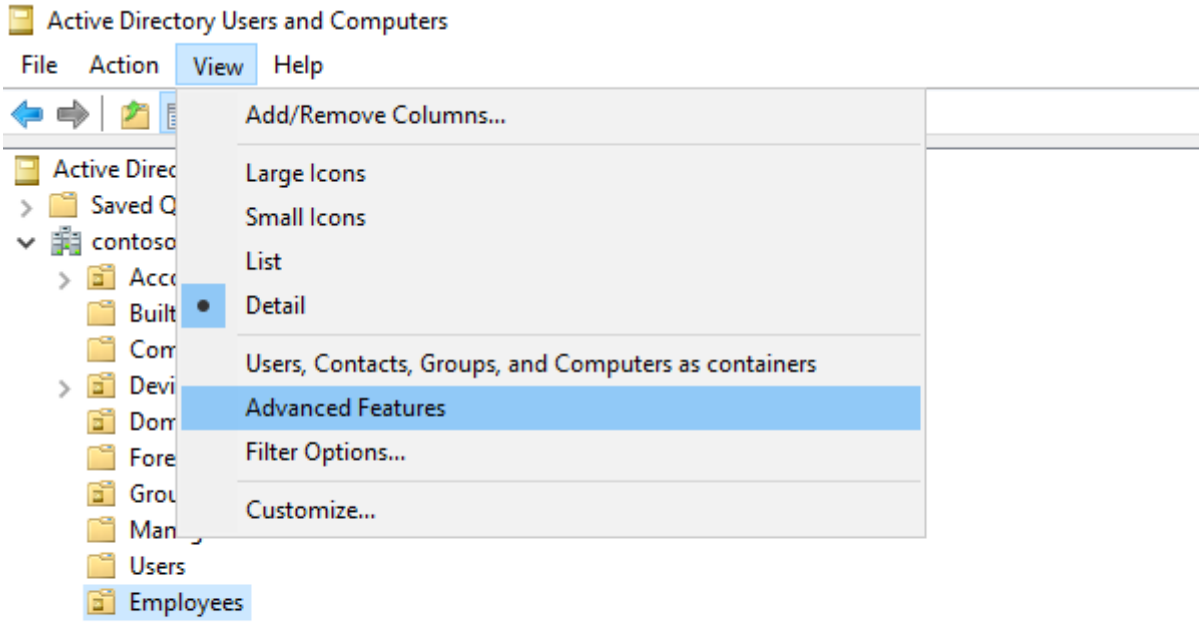


Preventing Accidental OU Deletion

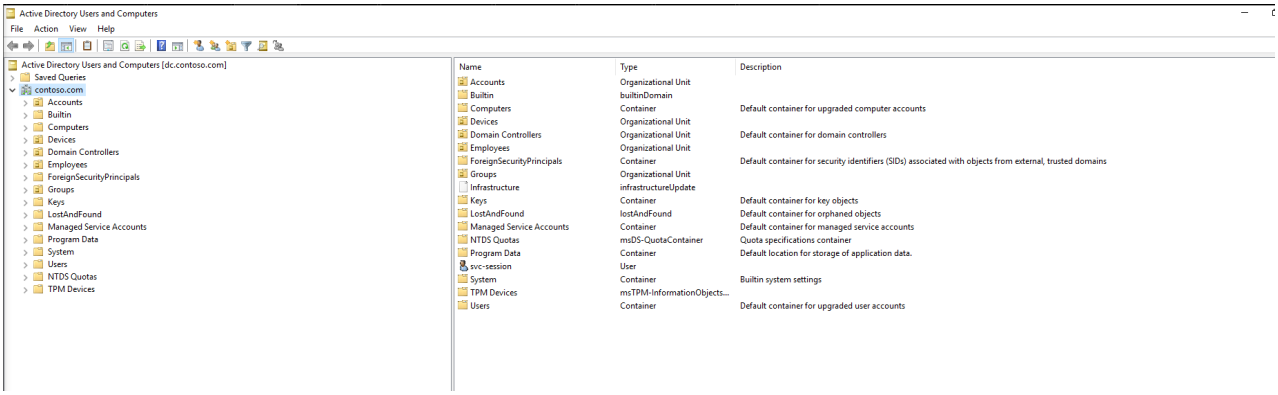
You may have noticed the setting ticked called “Protect container from accidental deletion”. This is a nice little setting that will prevent you from deleting the object. When attempting to delete an object without disabling this setting, you will be prompted with an error:



To bypass this, you must go to “View” and select “Advanced Features”.



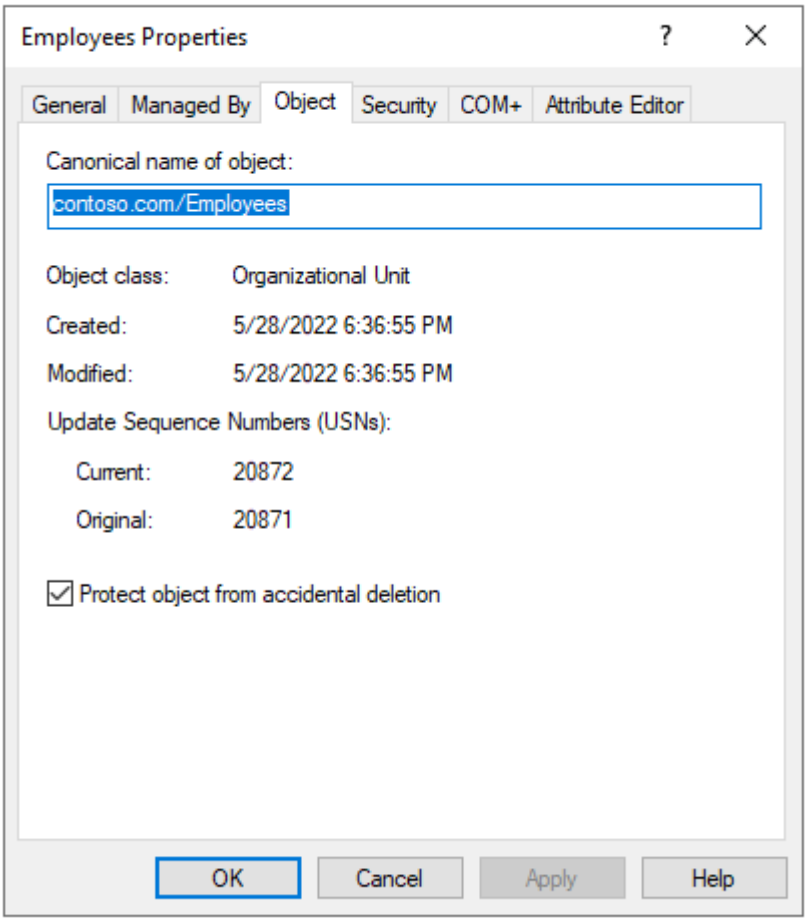
After selecting this, you will notice a large number of hidden objects will become visible:



I won’t spend too much time on this but there are a ton of really advanced settings here. One important one to know is the “AdminSDHolder” group, which can allow “Shadow Admins” to be created that may have administrative privileges over the domain that you

might not be able to find very easily!

Moving on - Now that the “Advanced Features” setting is enabled - if you find the object you want to delete, right click it and select “Properties”. If you navigate to the “Object” tab, you should see the tickbox that allows you to unselect “Protect Object from accidental deletion”.

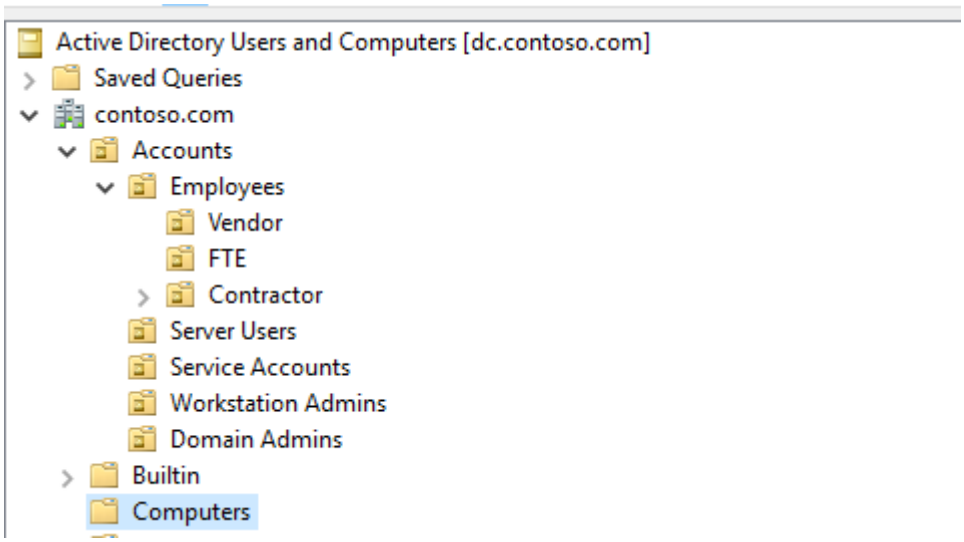


Unselecting that will then allow you to delete the object and make other modifications, like the moving the object.

Now that you have a better idea of how to create and structure OUs -Create the following in your lab:

```
Parent OU - Accounts
        Sub OU - Service Accounts
        Sub OU - Server Users
        Sub OU - Workstation Admins
        Sub OU - Domain Admins
        Sub OU - Employees
                Sub-Sub OU - Contractor
                Sub-Sub OU - FTE
                Sub-Sub OU - Vendor
Parent OU - Groups
Parent OU - Devices
        Sub OU - Servers
        Sub OU - Workstations
```

When finished, it should look like so:



Creating a Naming Scheme

In our last article, we talked about how critical it is to have a naming scheme for devices - the same applies for User Accounts and Groups as well. They should clearly define *what* the account is and *what* permissions it has, and who it belongs to. Some of these attributes can be specified in the Username field, while others can be specified in the Description fields.

Let's start developing a naming scheme, first we will start with generic users. There are many formats that you could follow, in our Lab, we will be using the first name **only**. This is a horrible idea in a production environment, but will work fine for our lab. Here are some example names we will be using:

- Jack
- Jill
- Eric

Now that we have a base naming scheme setup for our user accounts, let's move into our Workstation Admin accounts. I like to prefix the user accounts with a title like **wadm-**. This stands for Workstation Admin. You can have this be whatever you like, **wksadm-**,

wsadm, etc. In the Lab, I will be using **wadm-**. Some of our Workstation Admin user accounts will look like so:

- wadm-Morgan
- wadm-Jeff
- wadm-Tom

Once again, we'll setup another naming scheme for Service Accounts. Following the same methodology we have specified before, we could go with **svc-**, **srvc-**, etc. In the Lab, I will be using the **svc-** prefix. Some service accounts will look like so:

- svc-mssql
- svc-iis
- svc-pcjoiner

We also need to develop a naming scheme for our Server users, we can follow the prefix naming scheme again, some examples may be **srvusr-**, **srv-**, **su-**, etc. In the lab, I will be using the **su-** prefix. Some examples may look like so:

- su-emily
- su-jared
- su-hue

Let's start developing a naming scheme, first we will start with generic users. There are many formats that you could follow, first.last, finit Lastly, we will want to setup a naming scheme for Domain Admins. Since Enterprise Admins may be out of your control, we will skip them for this post. Though, I believe I've drilled the principal into your head by this point. Some examples that you could follow may be **da-**. **dadm-**, etc. We will be using **da-** in the lab. Some examples may look like so:

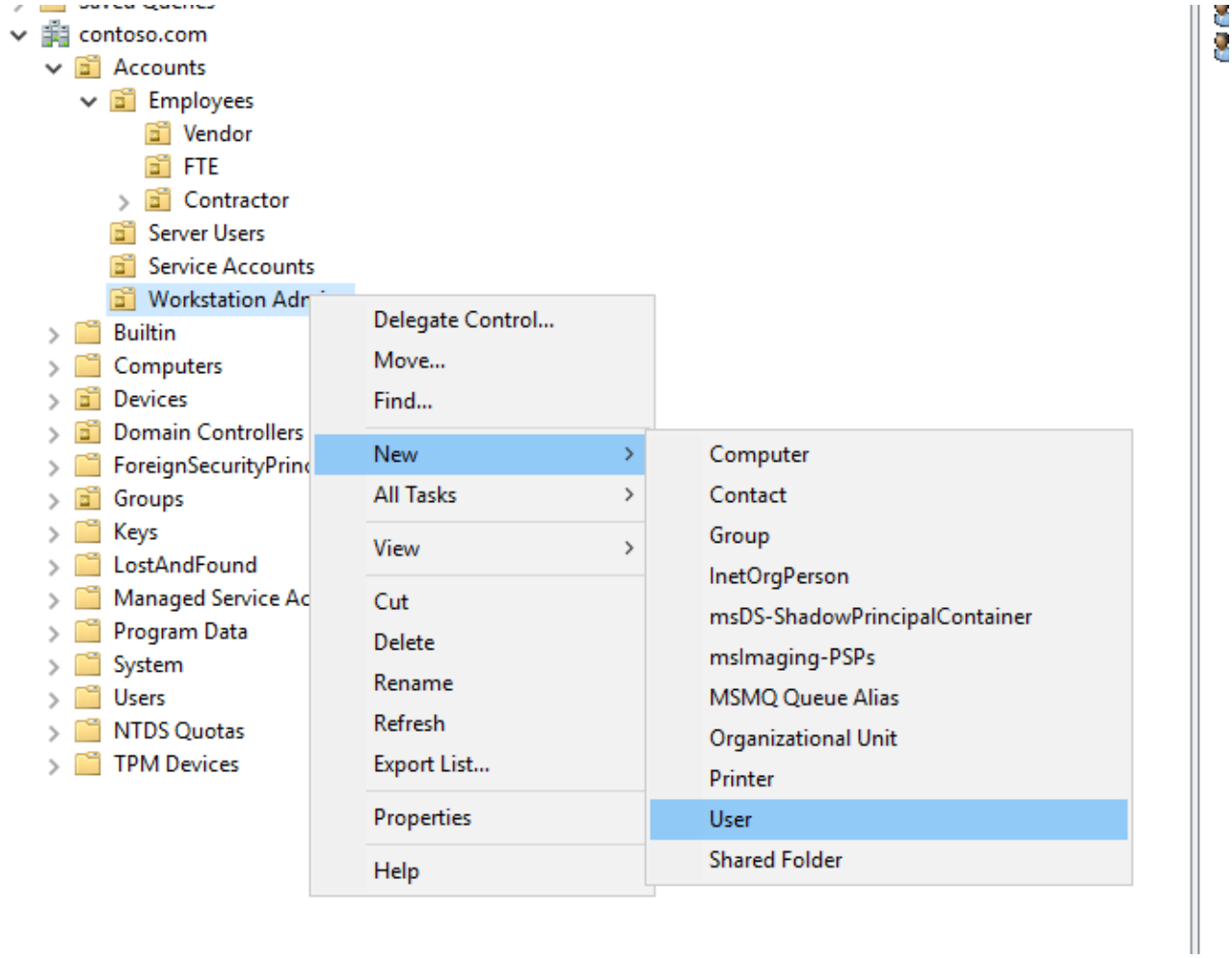
- da-serena
- da-eliot
- da-hilda

Last thing to note: For every different type of account (This could be FTE, Vendors, Contractors, Subcontractors), you should develop a naming scheme. In an enterprise, it's incredibly important to be able to quickly identify the type of account your dealing with in the event of an incident. So if you take one piece of advice from this, let this be to develop a proper, well planned naming scheme!

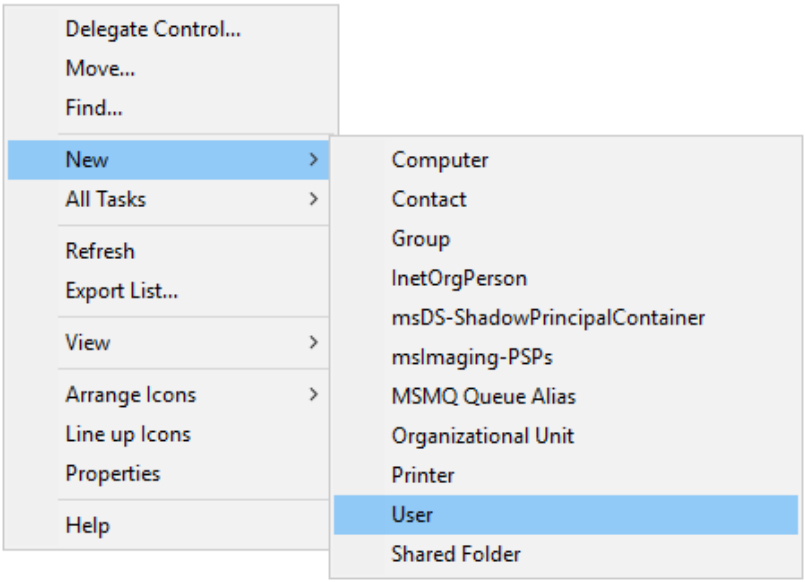
Creating User Accounts

Now that I have drilled a series of naming schemes into your head, let's get into building the actual user accounts.

To create a new user account, either right click on the OU and select "New", then "User":

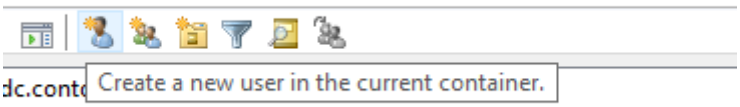


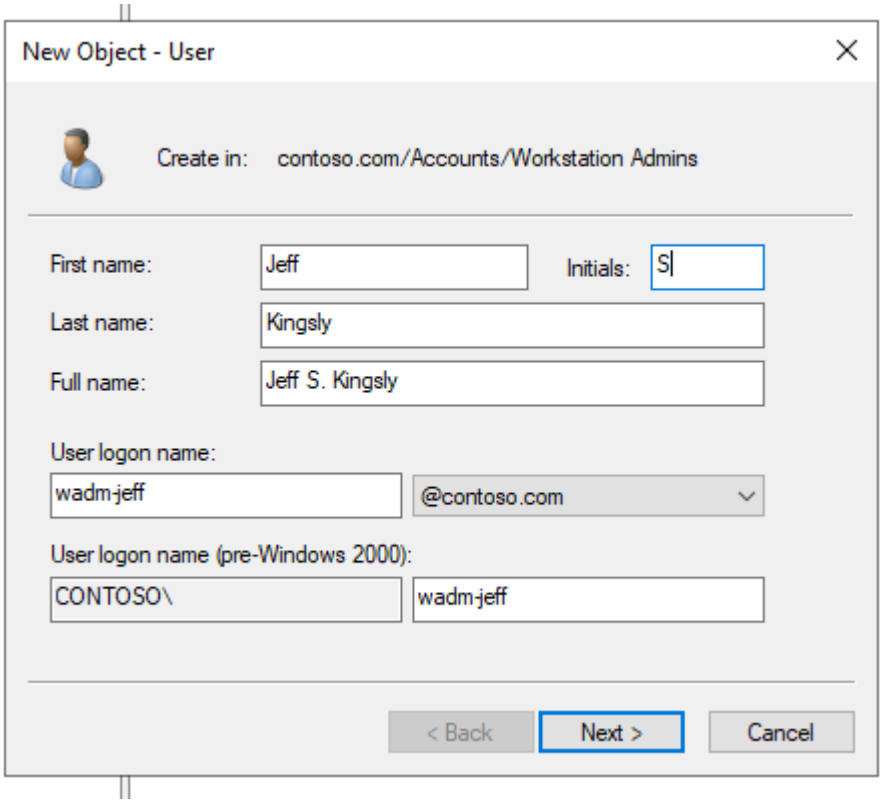
Navigate to the OU you would like to create a new user in, right click on the pane on the right side of your screen:



Navigate to the OU you would like to create a new user in, and select the “Create New User in the Current Container” button.

We will start out by creating a new Workstation Admin named “wadm-jeff”:

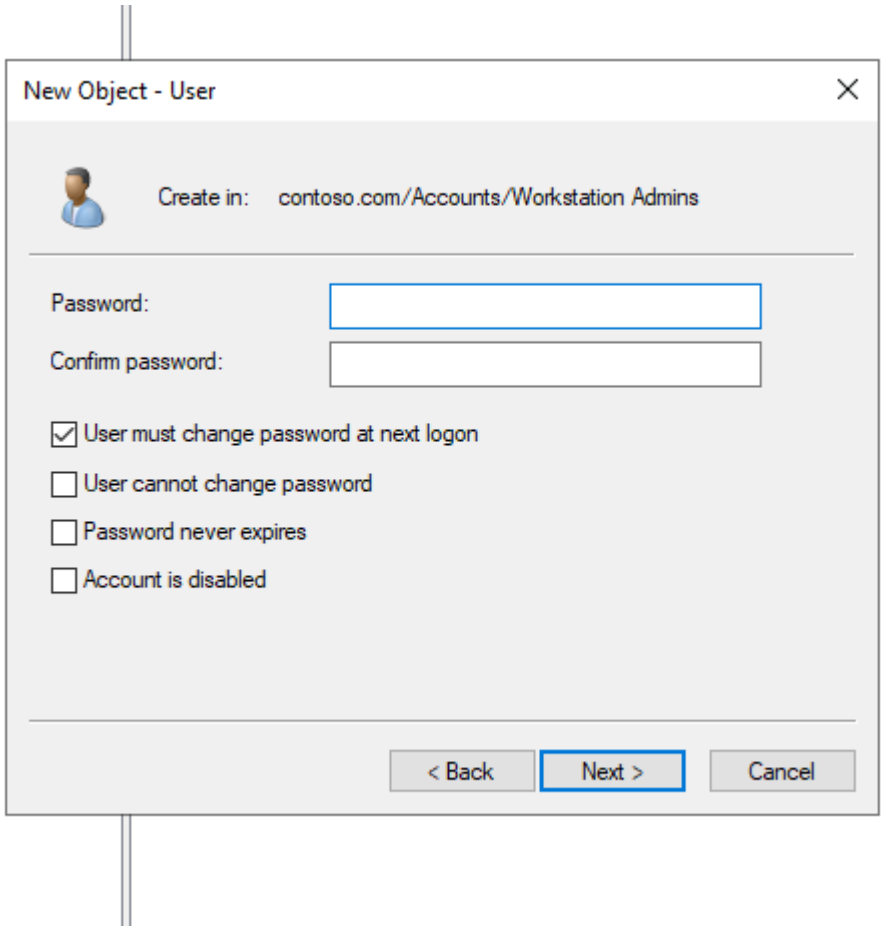




Jeff's full name is Jeff Steven Kingsly, so we will assign the first name to "Jeff", the middle initial to "S", and the Last name to "Kingsly", and his User Logon Name to wadm-jeff that we outlined in our naming scheme.

After selecting "Next", you will be prompted to give Jeff a Password, there are several important options to note:

- User Must Change Password at Next Logon - This is often used to give an employee a generic password for them to get started (perhaps this is given during the onboarding process). When they logon, they will then need to change it.
- User cannot change password. This is common in Service Accounts, where passwords shouldn't be managed by the end user, rather a Privileged Identity Management system, like CyberArk.
- Password Never Expires. This is commonly used in accounts that are used to join PCs to the domain, or are embedded in other misc scripts.
- Account is Disabled. This is generally done when an employee leaves the company, or an account is involved in an incident and needs to be locked out, but not deleted.

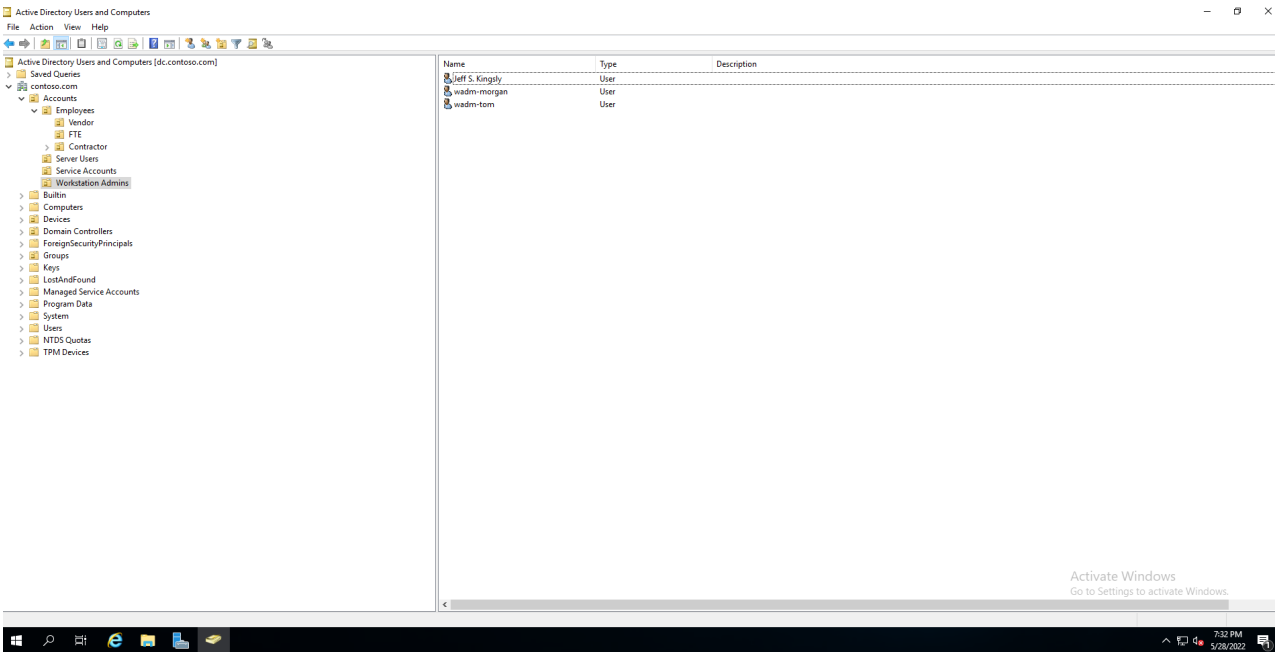


Since no actual users exist in the Domain, I will be enabling “Password Never Expires” and disabling “User must change password at next logon”.

Remember to add the credentials to your spreadsheet! At this point, it shoud look like so:

User	Password
Administrator	S3cur3P@ssw0rd123!
DSRM Password	DSRMRestoreP@ssw0rd123!
wadm-jeff	WADM-P@ssw0rd123

After successfully adding Jeff’s Workstation Admin user account, you should see something that looks like so:



Using the naming scheme we outlined in the previous section, populate the OUs we have created with the usernames we previously outlined. When it's done, you should have the following users in the following OUs:

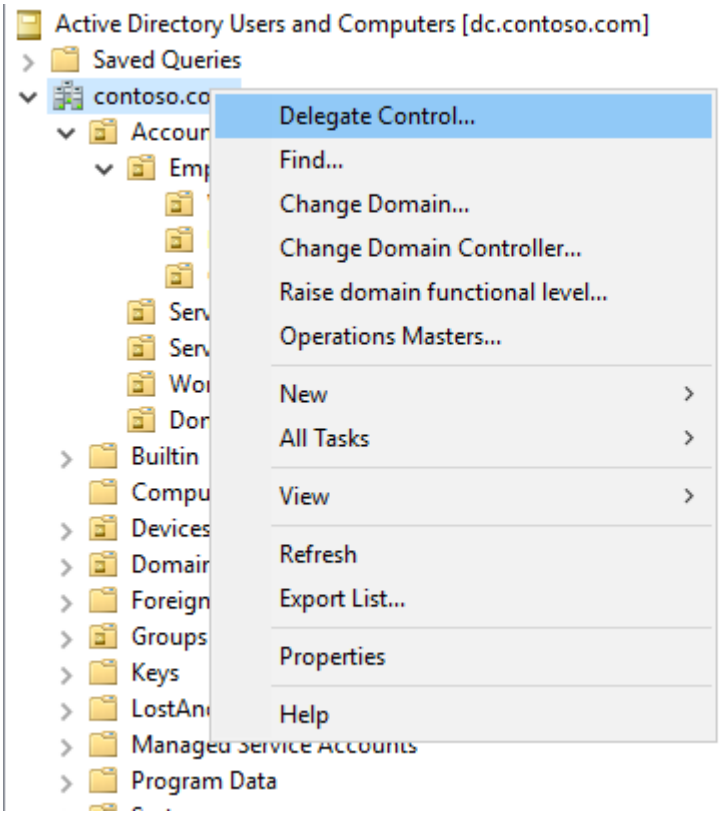
OU Name	User Accounts
Domain Admins	da-serena, da-eliot and da-hilda
Server Users	su-emily, su-jared and su-hue
Workstation Admins	wadm-tom, wadm-morgan and wadm-jeff
Service Accounts	svc-mssql, svc-iis, svc-pcjoiner
FTE	jack, jill, eric

Delegation Permissions to Users and Groups

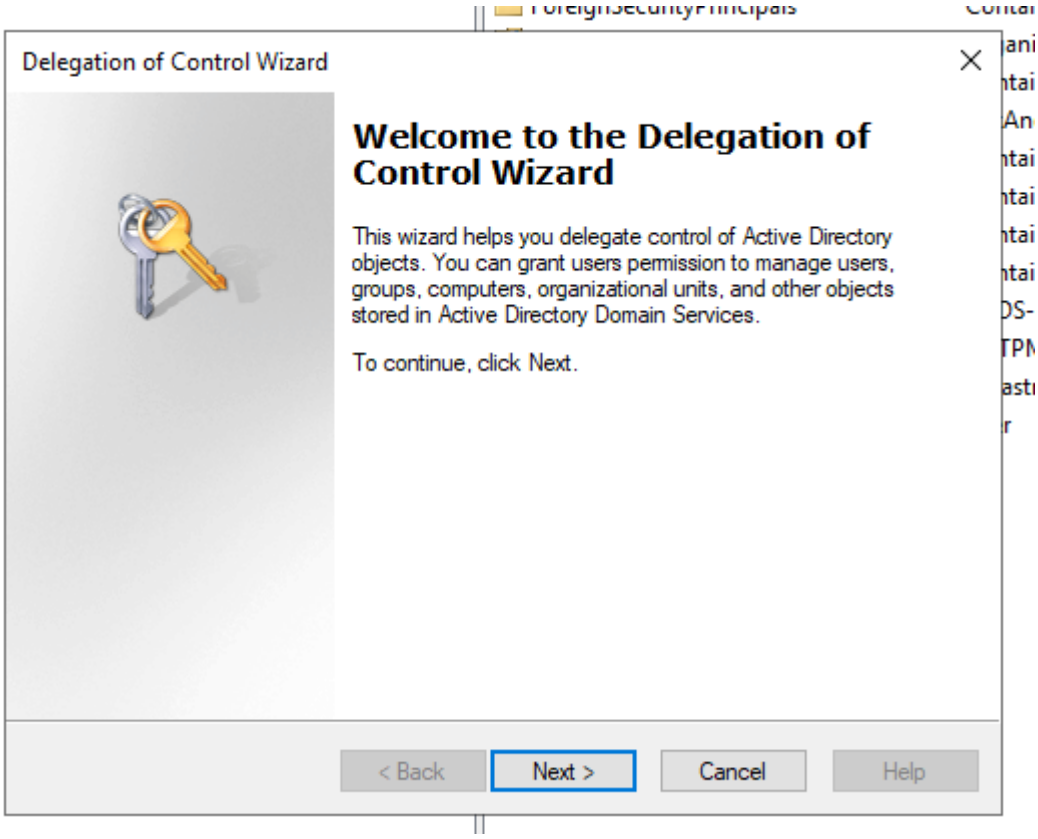
One important thing to learn how to do in Active Directory is to delegate specific permissions to Users and Groups. One classic example of this is delegating permissions to a user account to join PCs to the Active Directory Domain. By default, all users can join 10 PCs to the domain, this quota *should* be zero and a dedicated user account should be used for this task. So, we will be setting up a user account to do so!

We will be utilizing the svc-pcjoiner Service Account that you created to allow this account to do so. This can all be accomplished through the "Active Directory Users and Computers" application that we have already been working with.

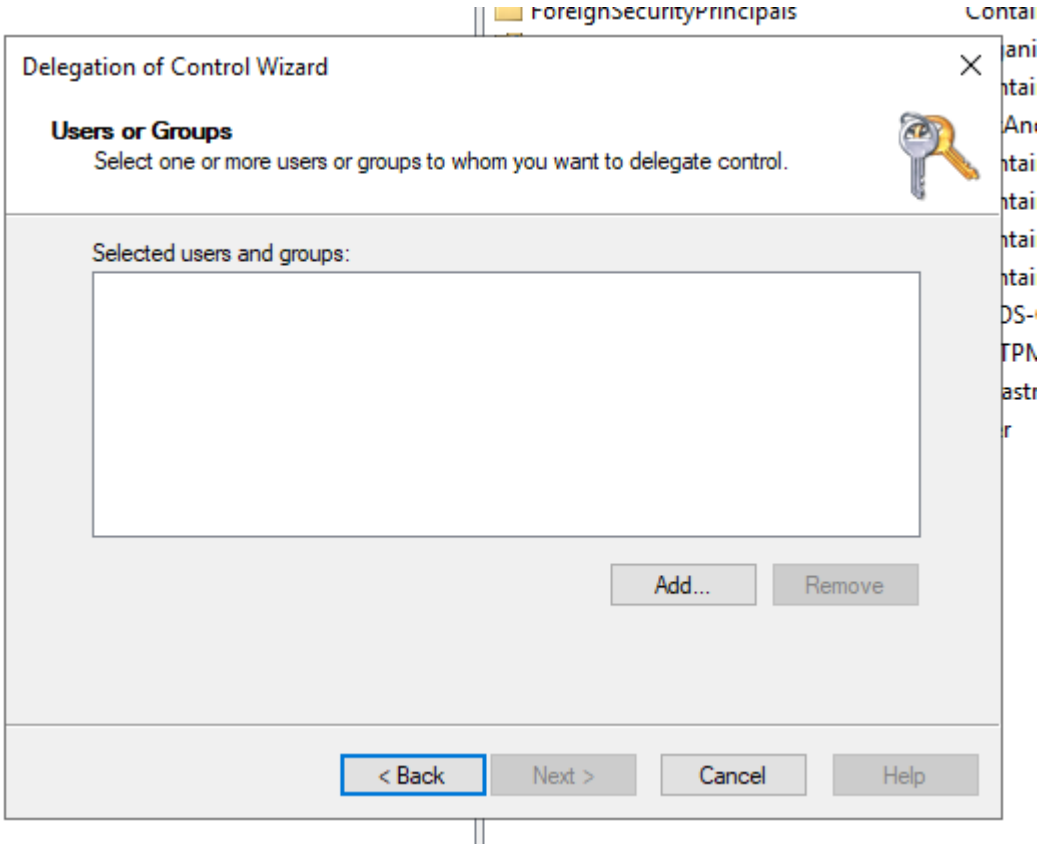
To open up the "Delegate Control Wizard", right click on the root of the domain and select "Delegate Control":



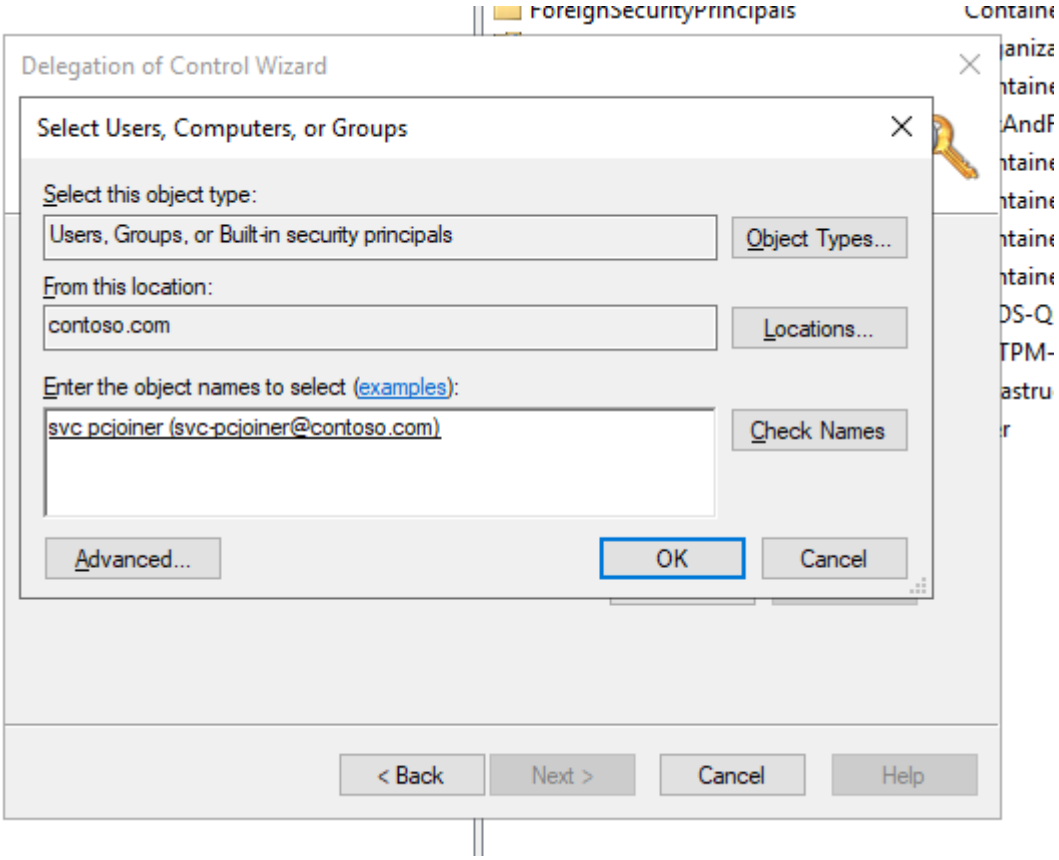
After doing so, you should see a menu that welcomes you to the Delegation Control Wizard!



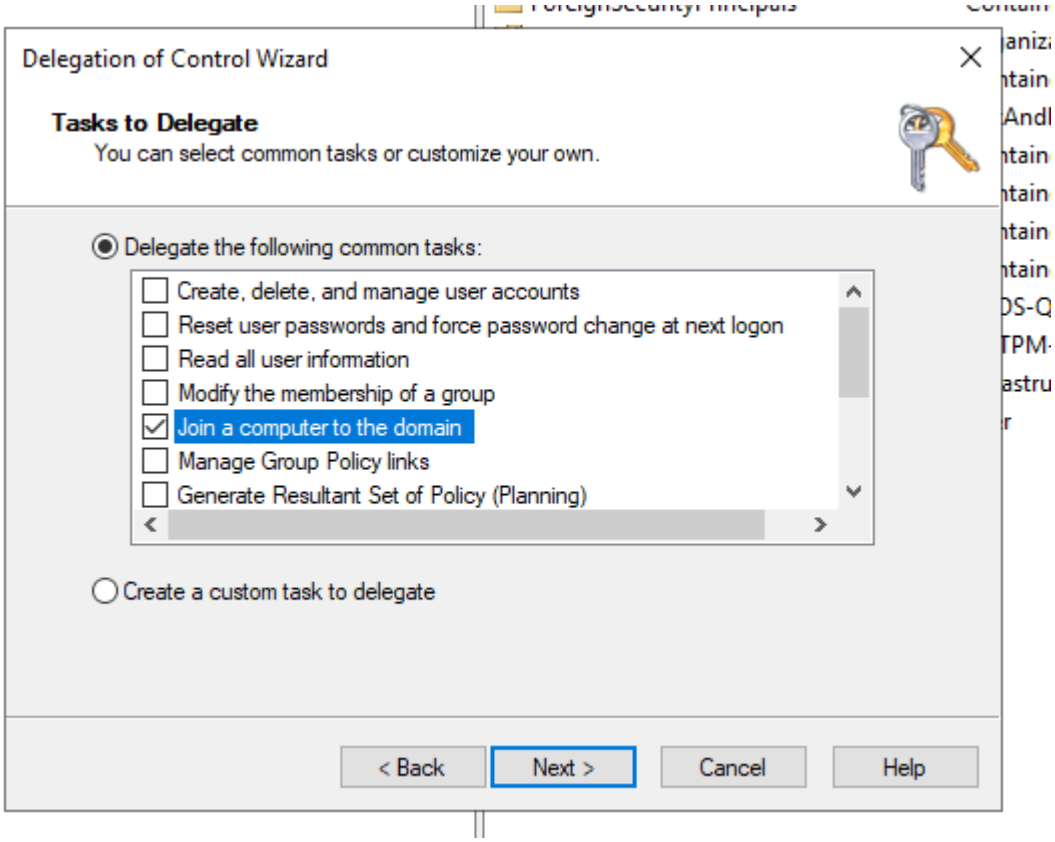
Select “Next”, and you will be prompted to specify a User or Group to delegate control to:



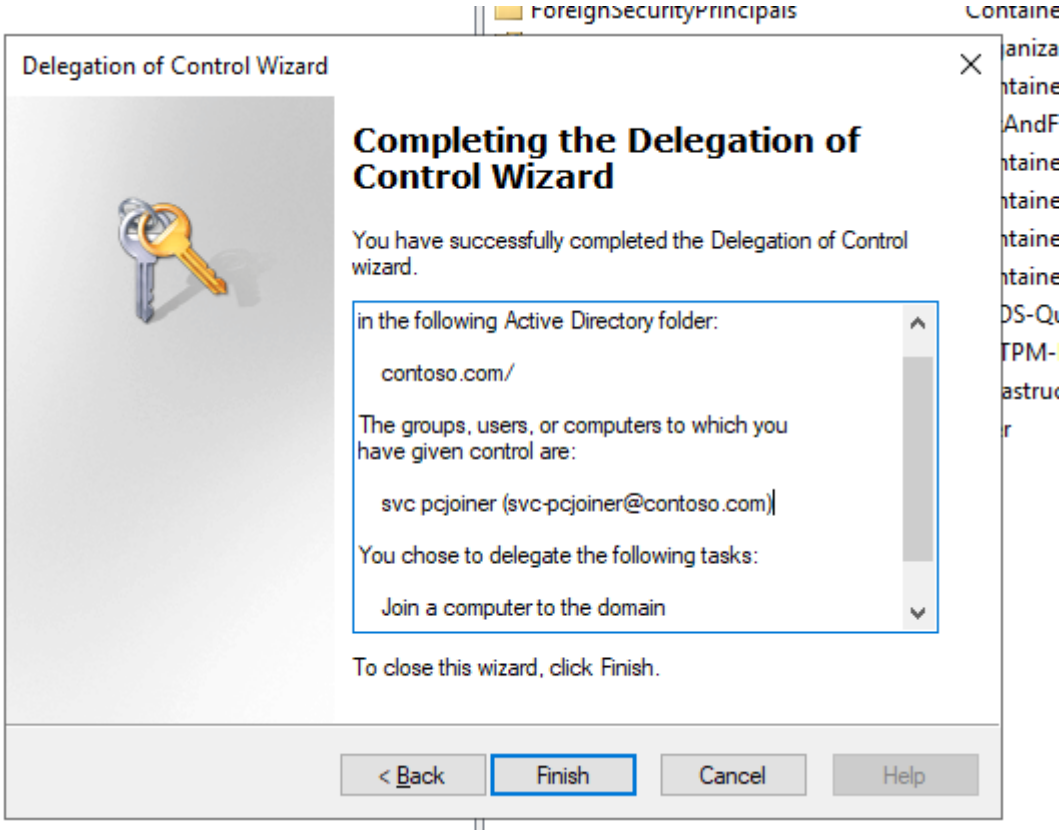
Click “Add” and a new window will open that allows you to search the AD Domain for Users.



Search for “svc-pcjoiner”, click “Check Names”. You should then see AD autofill in the rest of the username/domain. Click “OK”. You should then see “svc-pcjoiner” as one of the users. Click “Next”.



This brings up the “Tasks to Delegate” window, which allows you to delegate permissions to the selected user or group. In this window, we are going to select “Join a Computer to the Domain”. After the box is ticked, click “Next”.



This brings us to the summary window which displays all of our changes. Pressing “Finish” will apply the changes to Active Directory and will grant the specified user account the permissions you granted it.

While we are on the topic of delegating permissions to accounts - It's important to know that not *all* service accounts may be used in services. Some may be used in *processes*, such as joining a PC to the domain, thus not all service accounts may have a SPN issued to them and may not all be Kerberoastable.

Joining a PC To the Domain

Now that we have learned how to delegate permissions on how to join PCs to the Domain, it's time to explore *how* we can join a PC to the domain. In order to get this one, there are several things we must make sure we have before starting:

1. The credentials to svc-pcjoiner
2. A Non-Duplicate machine SID

The first of the two things is trivial to obtain, the second one is slightly more difficult. We will explain in the next section why this is important.

Prepping the Workstation

In a Windows environment, each computer has something called a SID - or Security Identifier that cannot be duplicate if is joined to an Active Directory domain.

So how might you get a duplicate SID? Great question. In Virtualization, we may often create a base image to build from. For my lab, to avoid installing Windows 4-5 times, I created a "Gold OVA". Unfortunately, when you do this, the machine SID is the same. Fortunately, Microsoft has recognized people do this, so they built a tool called "SysPrep" to revert a machine to its "Default State" and re-randomize the SID of the workstation.

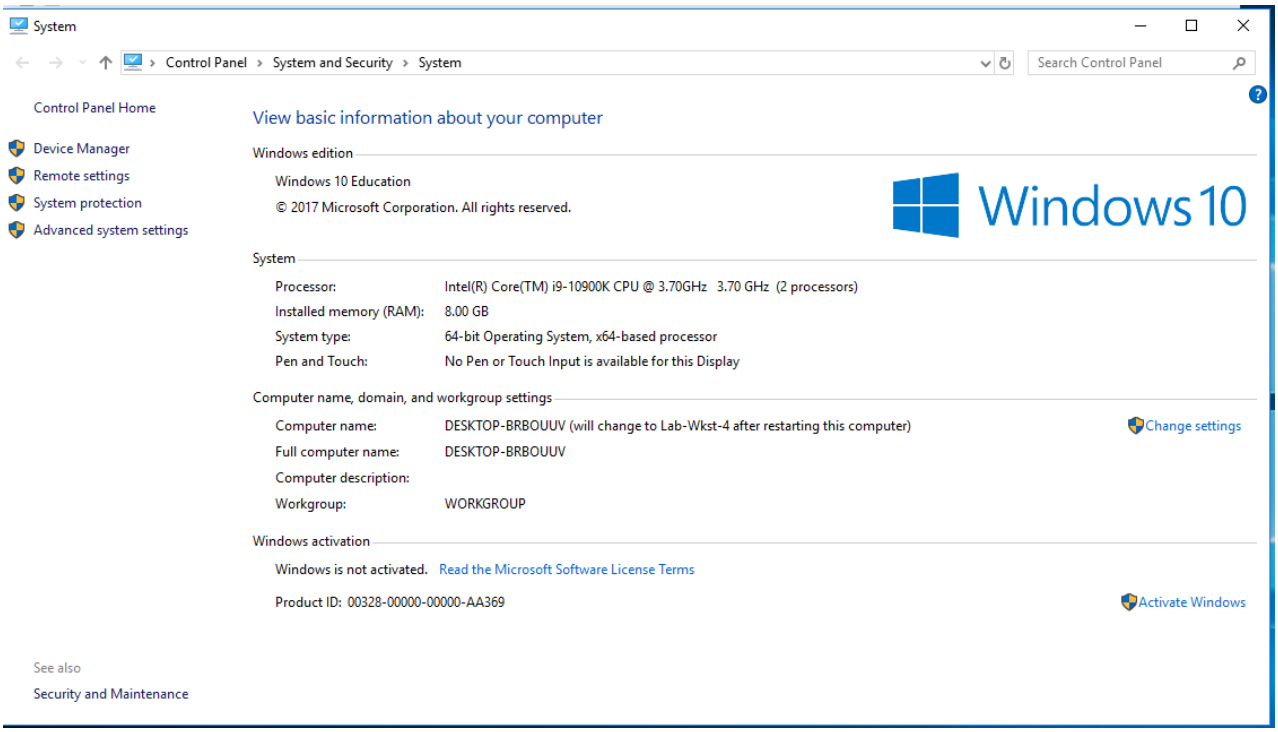
To execute SysPrep, open up CMD as an Administrator and navigate to C:\Windows\System32\SysPrep\, then execute the following command:

```
.\sysprep.exe /quiet /generalize /oobe
```

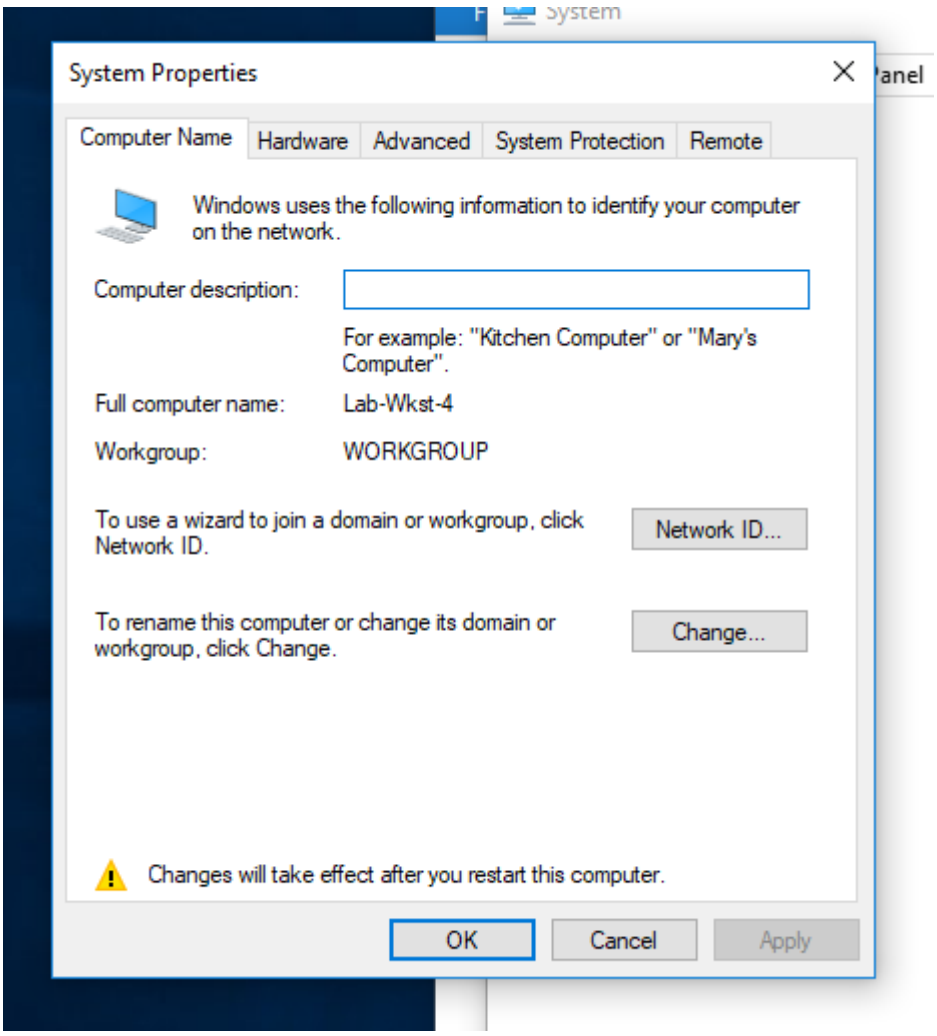
After executing this, the machine should reboot and your SID should be re-randomized. One thing to note before you attempt to join the PC to the Domain, make sure that your DNS server is set to the Domain Controller if you are *not* using DHCP.

Setting your Hostname

As I mentioned before, coming up with a naming convention is crucial. For the Lab, I am using a naming scheme of Lab-Wkst-#. In order to set the Hostname, open up Explorer.exe, right click on "This PC" and select "Properties".

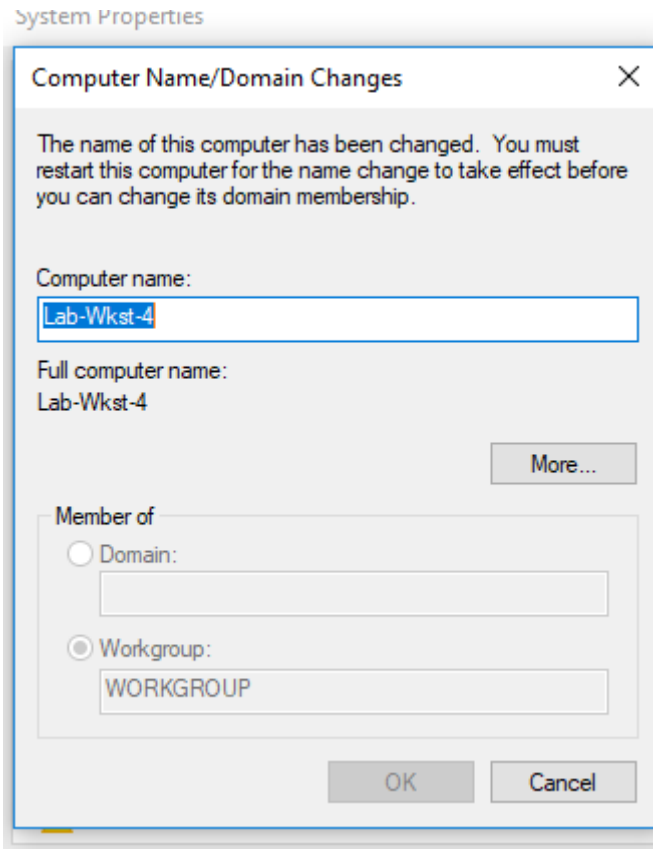


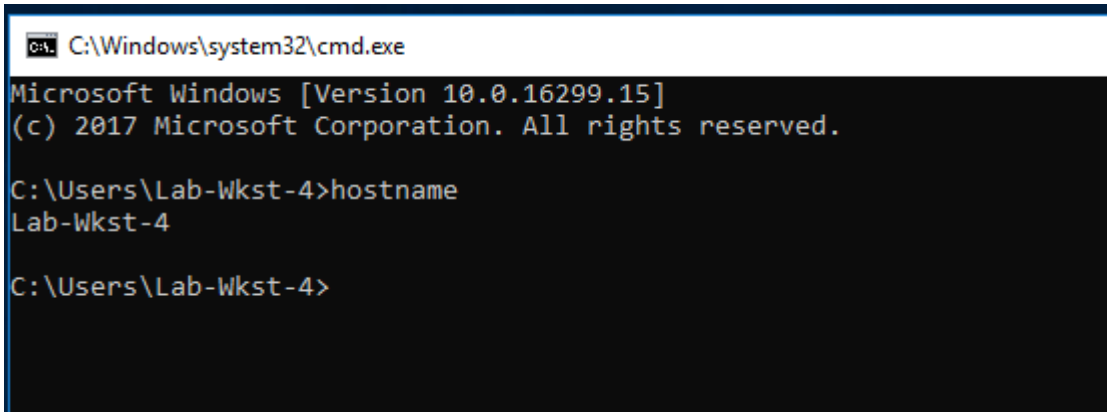
After this is done, click “Change Settings”, or “Advanced System Settings”. Then select “Computer Name”.



Select “Change” to change the Workstation’s name, and rename it to whatever you desire!

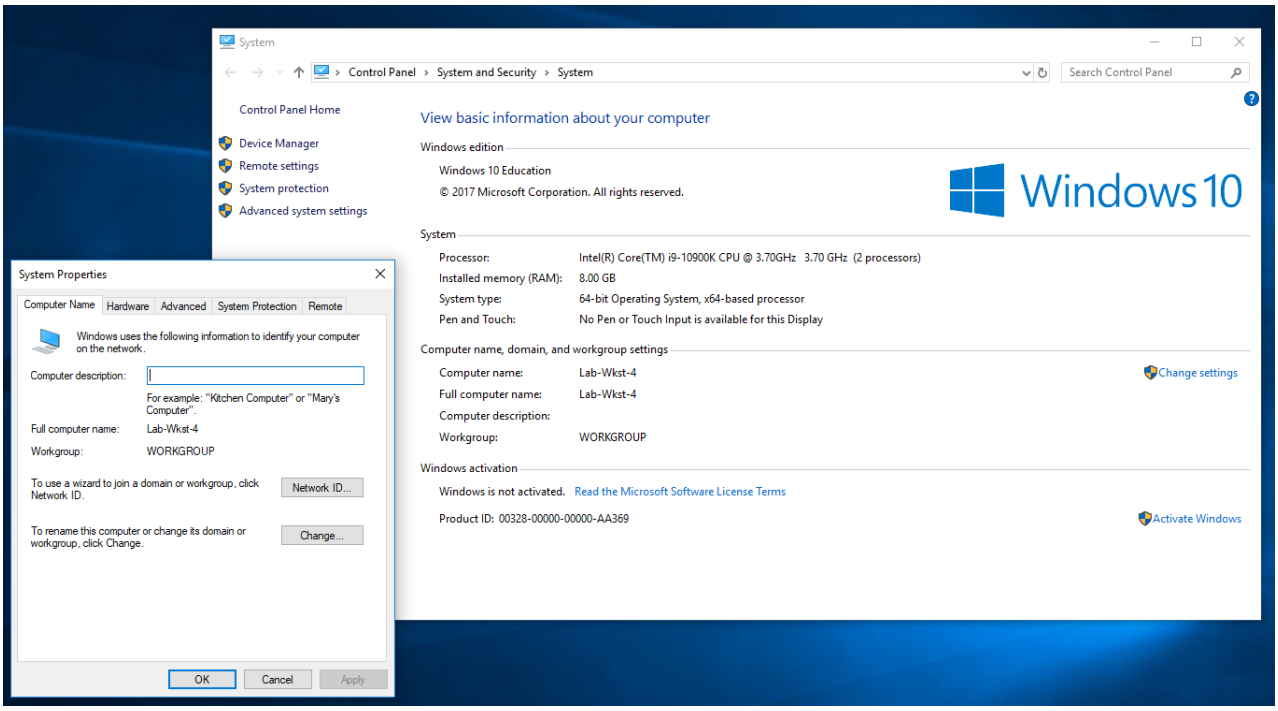
After clicking “OK”, you will be prompted to reboot your PC. For the name change to take affect, you must do this, so after this is done, give it a quick reboot. To verify the hostname change worked, open up cmd.exe and type “hostname”.



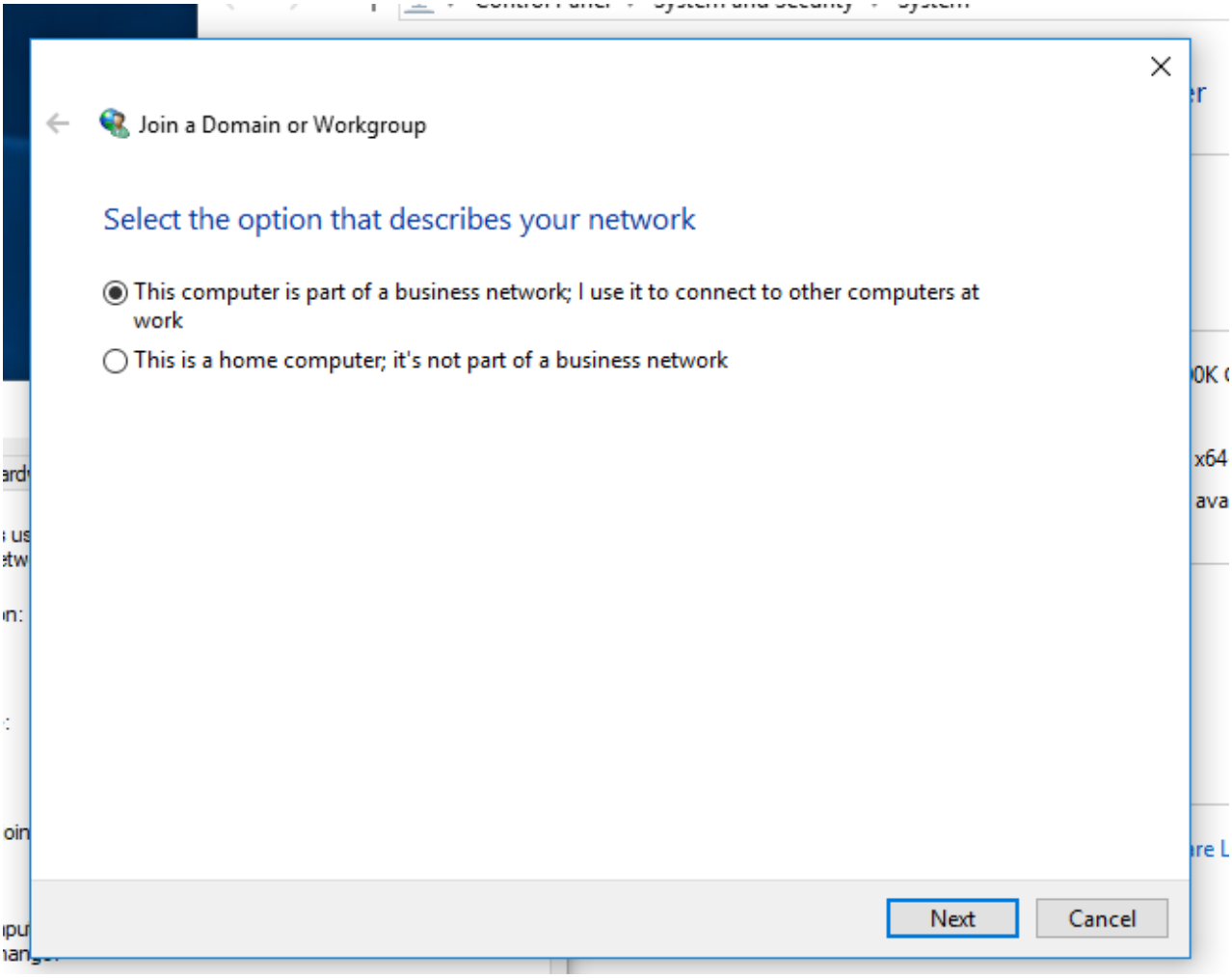


Joining the PC to the Domain

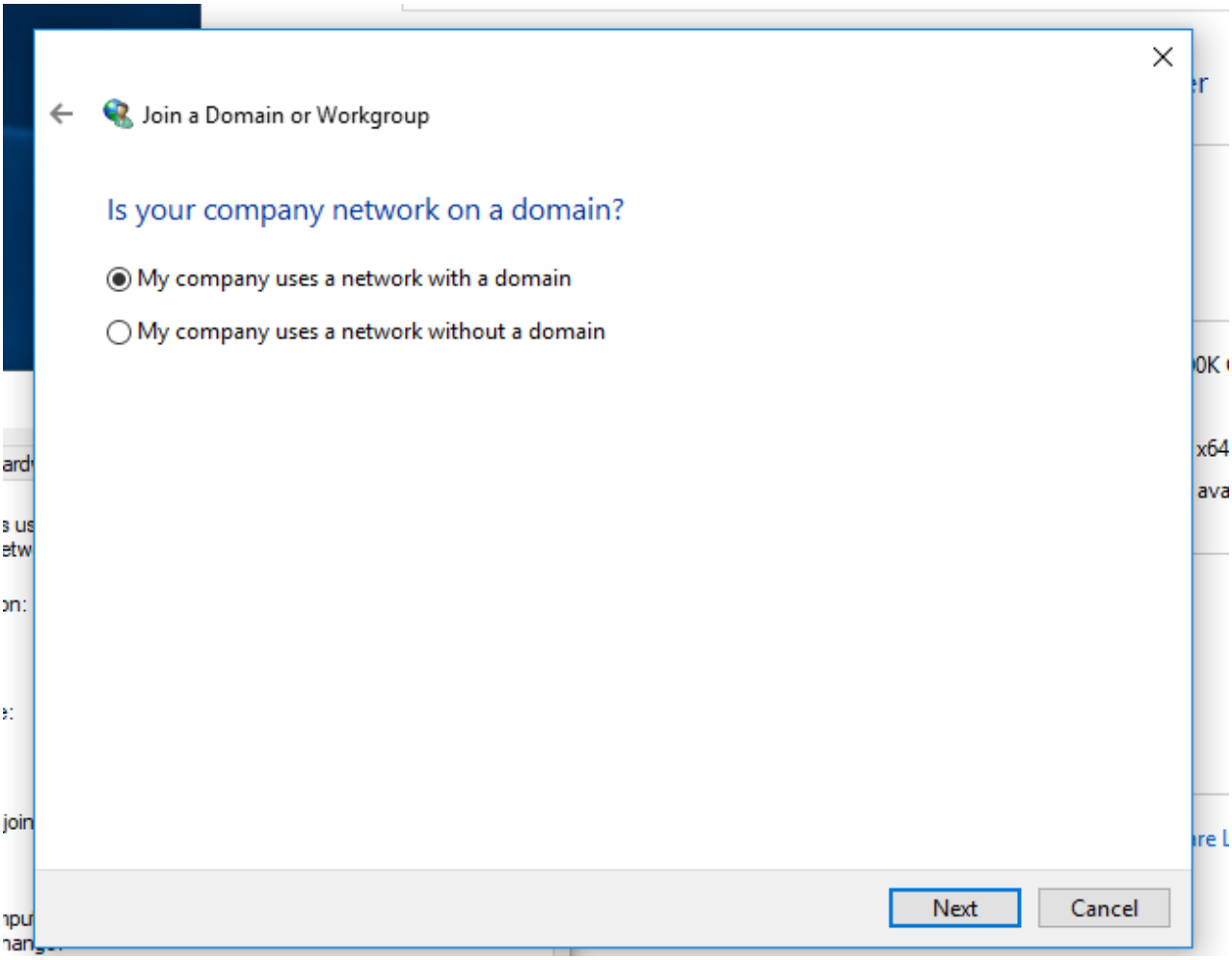
Now that we have updated our PCs hostname, we can now join the PC to the domain! To kick the process off, open up Explorer, right click on “This PC” and select “Properties”.



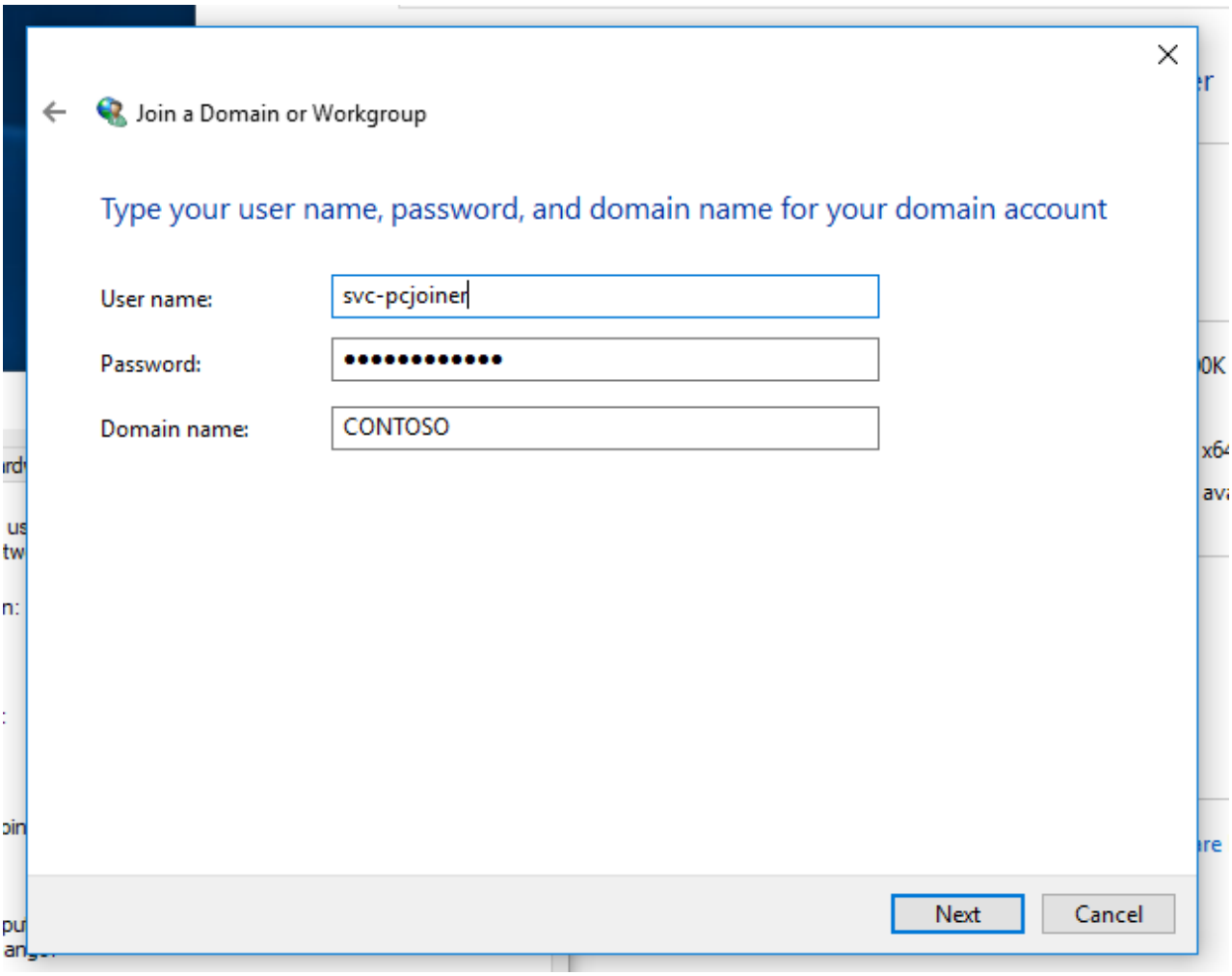
This time, we will be selecting “Network ID” instead of “Change”. This will start a “Join me to the Domain” wizard. The first question is to determine if we must join the PC to a Workgroup or a Domain.



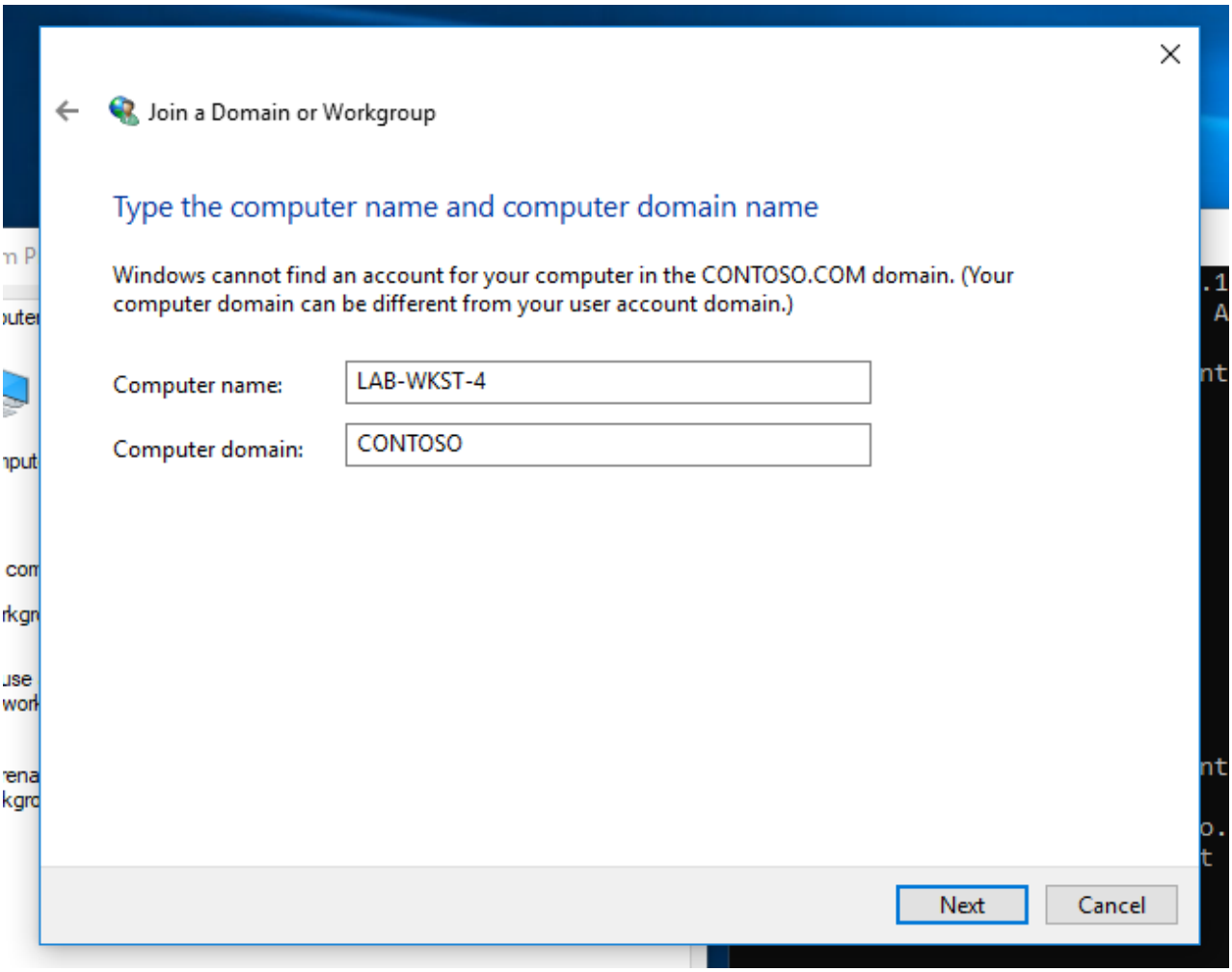
This is indeed part of a business network, so select that and click “Next’. The next question will ask if we use a Domain or not (This is another Workgroup question). We do use a domain, so select that and click “Next”.



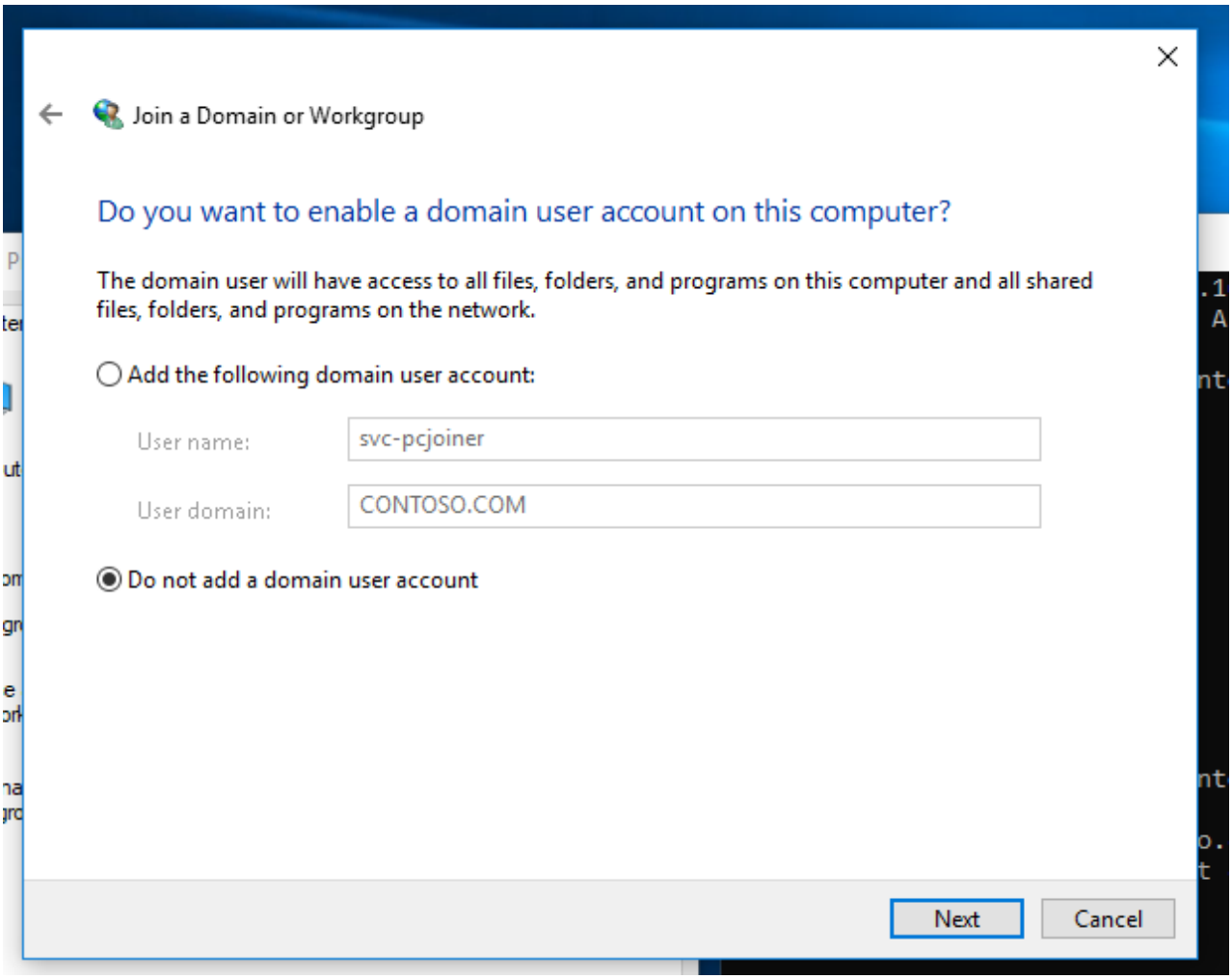
Windows will then inform us we need a Username, Password, and a Domain Name, so click “Next”. Then we will populate the next window with our svc-pcjoiner credentials and our Domain Name.



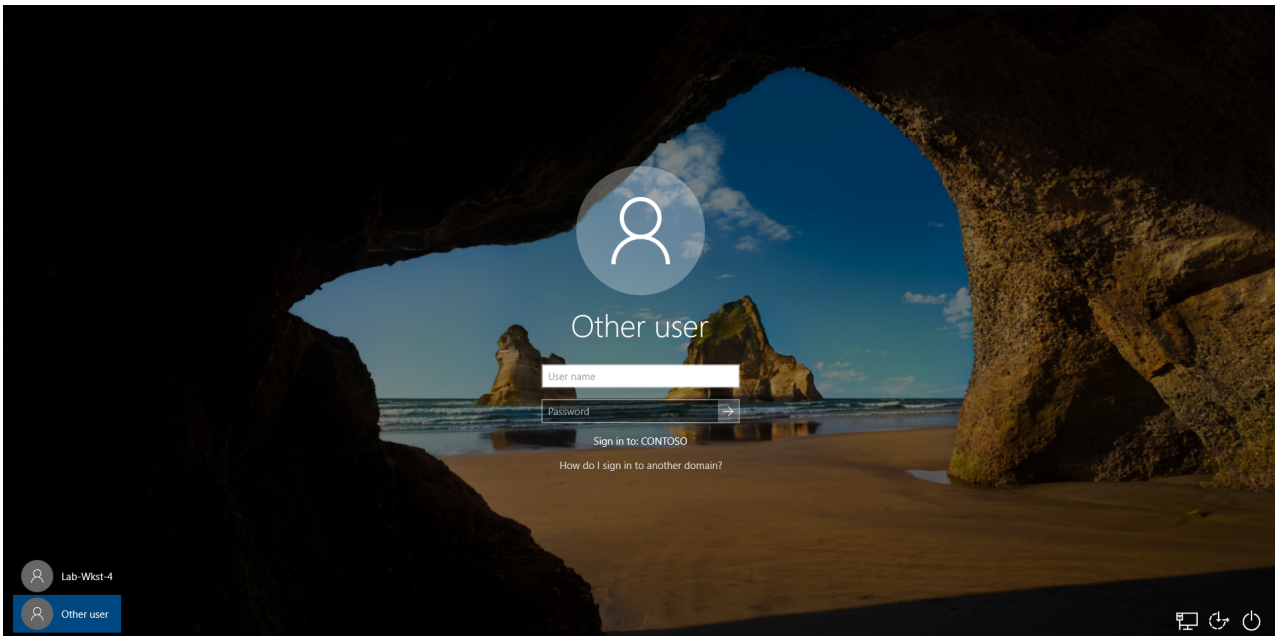
Afterwards, click “Next”, Windows will then try to reach out to Active Directory and validate the credentials we provided. If you did not create a Computer Object for this workstation (which is likely the case), Active Directory may come back and say “Hey, a Computer Object doesn’t exist. What do you want to do?”. If this is the case, supply the Workstation Name and the Domain again and then press enter.



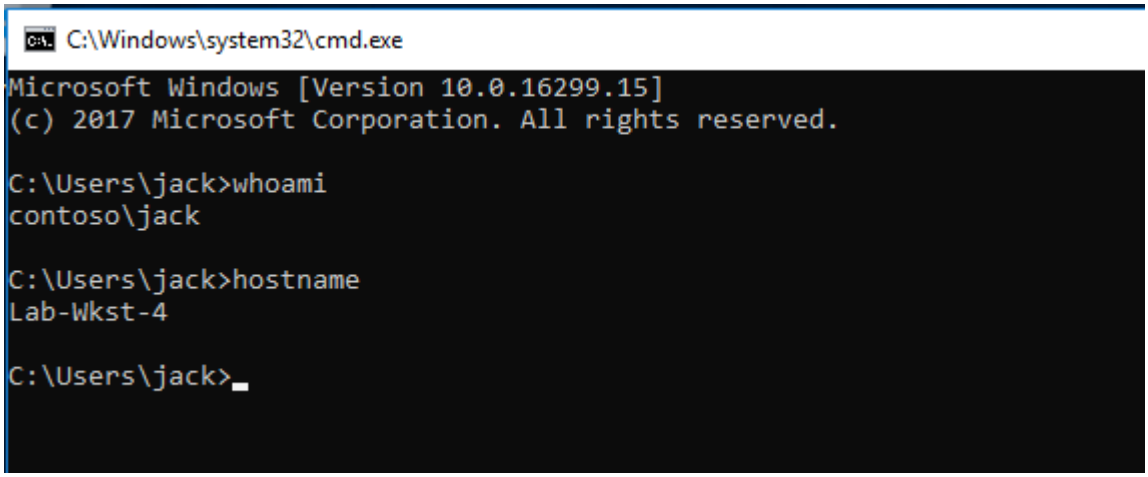
Afterwards, Windows will ask you if you want to create a user profile for the user previously mentioned. The answer is **no**. This is a Service account and has one explicit purpose - To join PCs to the Domain.



Afterwards, click “Next”, “Next”, then OK. Your computer will then reboot and you will be brought to a Domain Logon screen when the PC restarts.

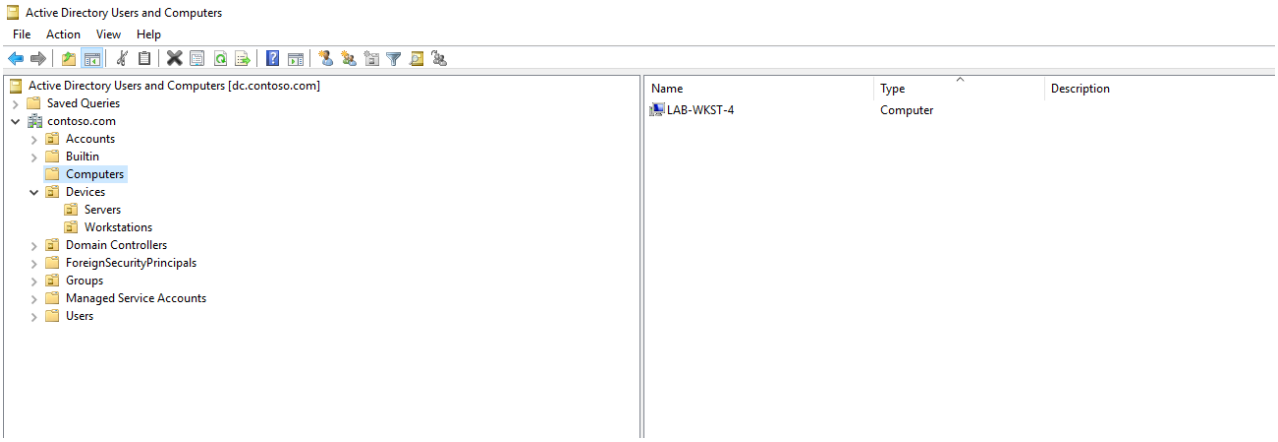


At this point, you can then sign in with **unprivileged** user credentials to ensure the PC has joined itself to the domain properly. After we log on, we can determine our user with `whoami` and print the `hostname`.

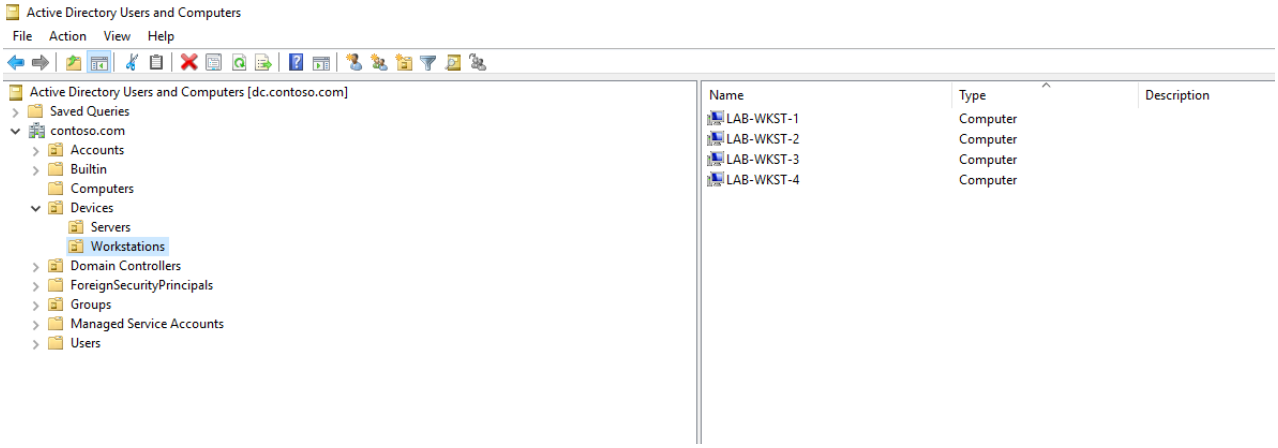


Verifying on the Domain Controller

Flipping back over to the Domain Controller, navigate back to the “Active Directory Users and Computers” program and head over to the “Computers” OU. We want to move this to our custom “Devices -> Workstations” OU for better management. We can do this by dragging and dropping “LAB-WKST-4” to the desired OU.



Moving over to the “Devices -> Workstations” OU, we can see that our Workstation was successfully moved over!

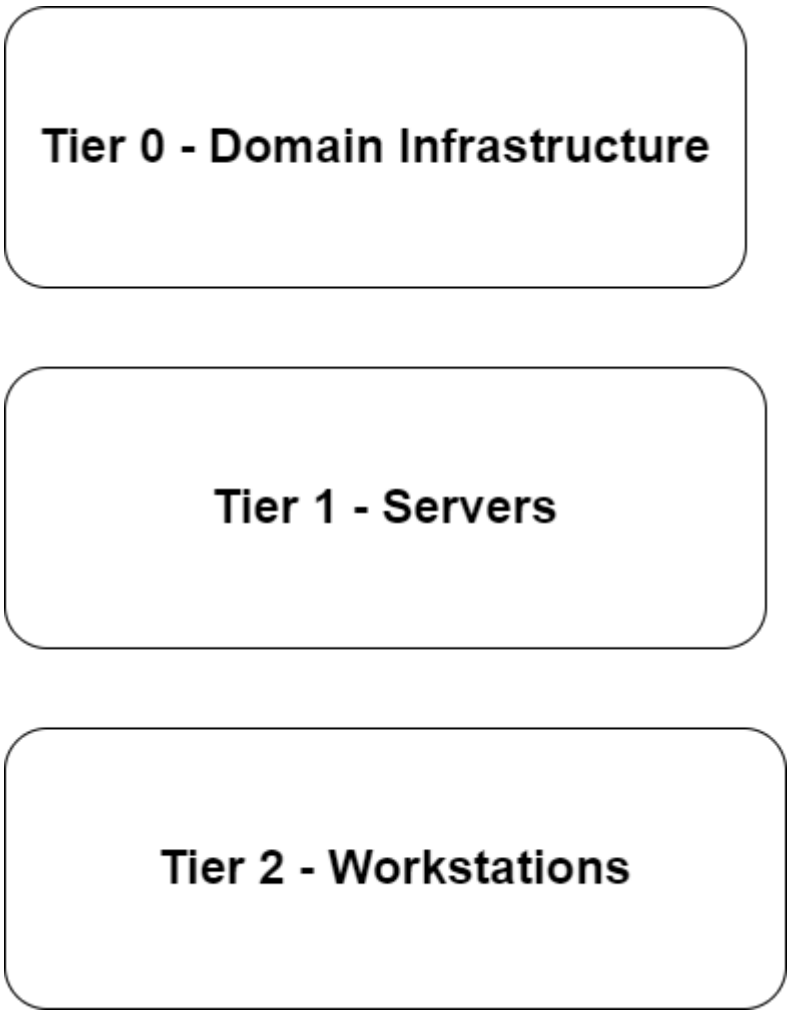


And with all of that, we have finished the process of setting up a User with privileges to join a PC to the Domain, and then joining a PC to the Domain. Its as simple as that!

Generally, in an enterprise you will prepare one master image which is compatible with various different physical PCs, with company supplied software. Like EDR, AV, Office365, etc. This would be called a Gold Image. It's suppose to help automate the deployment process for PCs in the Domain. Beware that this image should be secured because it often contains a *script* that contains credentials to join PCs to the Domain.

Group Policy Objects

The last topic we're going to talk about today is Group Policy Objects. These are Objects within Active Directory that directly affect Users and Computers within the Domain. It's important to know GPOs affect all objects within a Container (or OU). We have laid the foundation for a tiered infrastructure.



We are now going to work on applying Group Policy Objects to conform around our tiered infrastructure model we have setup. So, what kind of Group Policy Objects are going to implement? That's a great question - Here's what we're going to start with:

1. Deny Generic Users to sign into Servers
2. Deny Server Users to sign into Workstations
3. Deny Workstation Admins to sign into Servers
4. Allow Generic Users to sign into Workstations
5. Allow Workstation Admins into Workstations

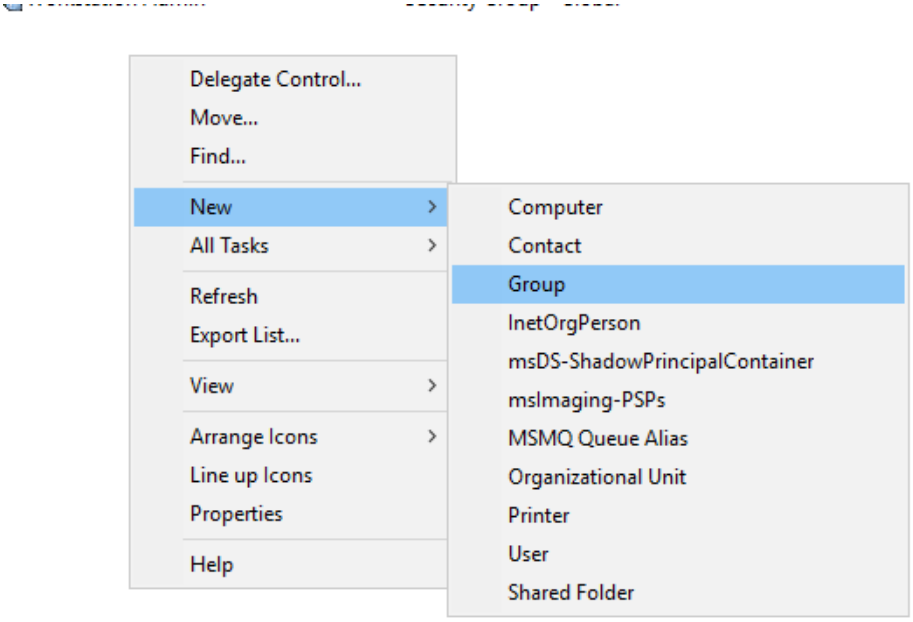
6. Allow Server Users to Signing Into Servers

We can acomplish this with a mix of Groups and Group Policy Objects. In order to execute this we are going to create the following groups:

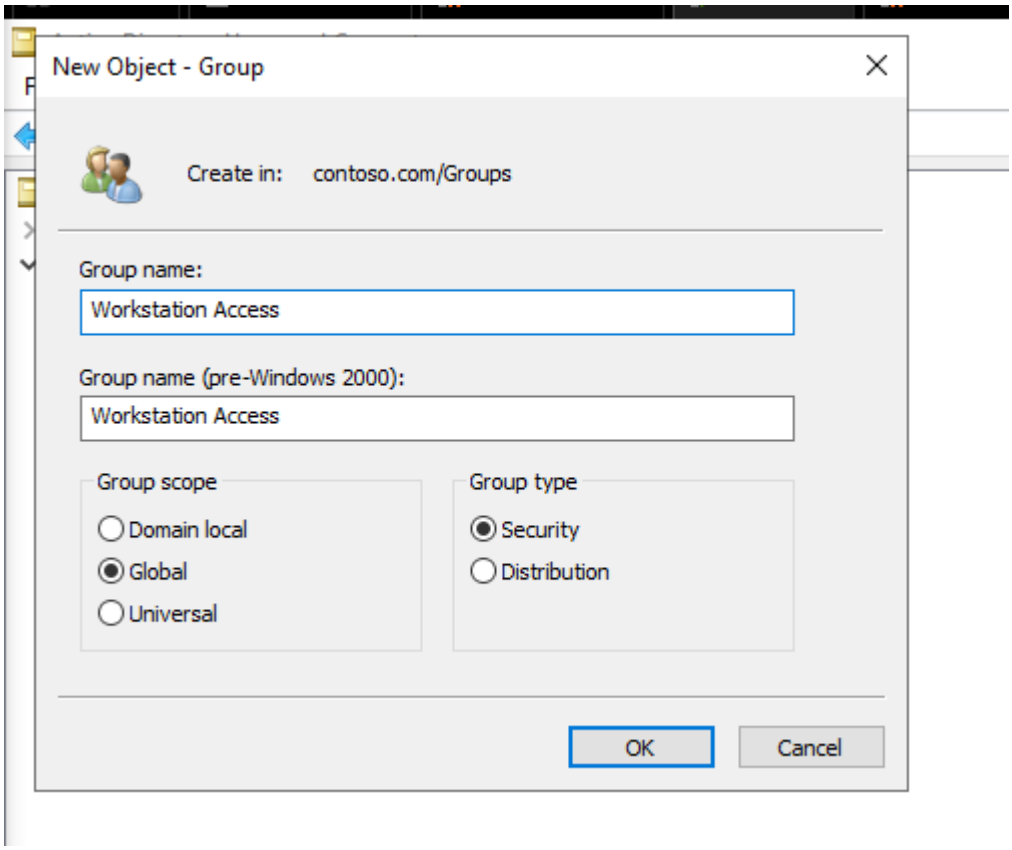
1. Server Access
2. Workstation Access
3. Workstation Admins

Creating a Group

We're going to dive back into "Active Directory Users and Computers". Hop on into the "Groups" OU we created a while back, right click and select New -> Group.

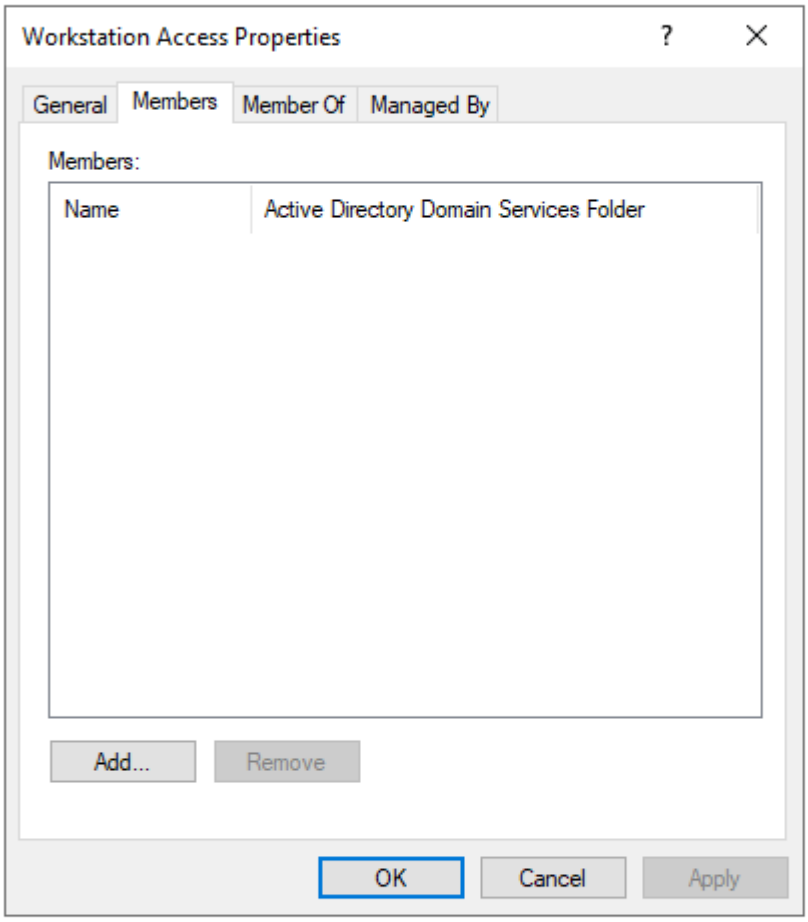


We will be creating a group for "Server Access", "Workstation Access", and lastly "Workstation Admins". Pressing "OK" will create our new group.

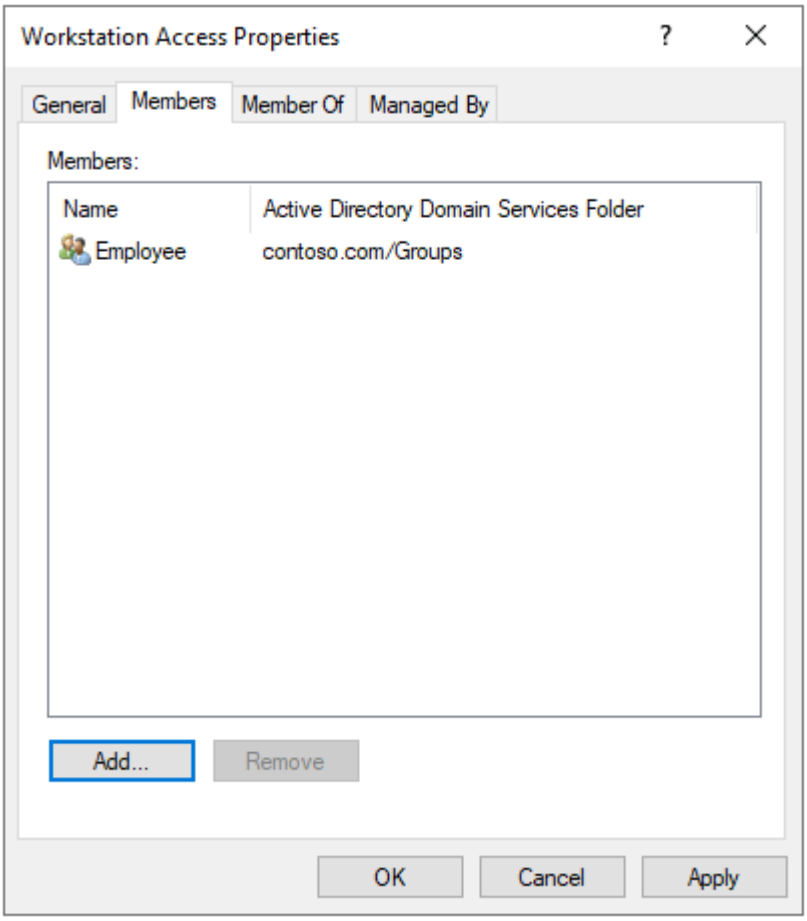


It would be a good idea to create a group for Active Employees, and Inactive Employees as well. This group will only belong to primary users accounts, and not any speciality accounts (like Workstation Admins), since users *should* have both a primary account for their every day tasks as well as their speciality tasks. So go ahead and create the groups mentioned above, as well as “Employee”.

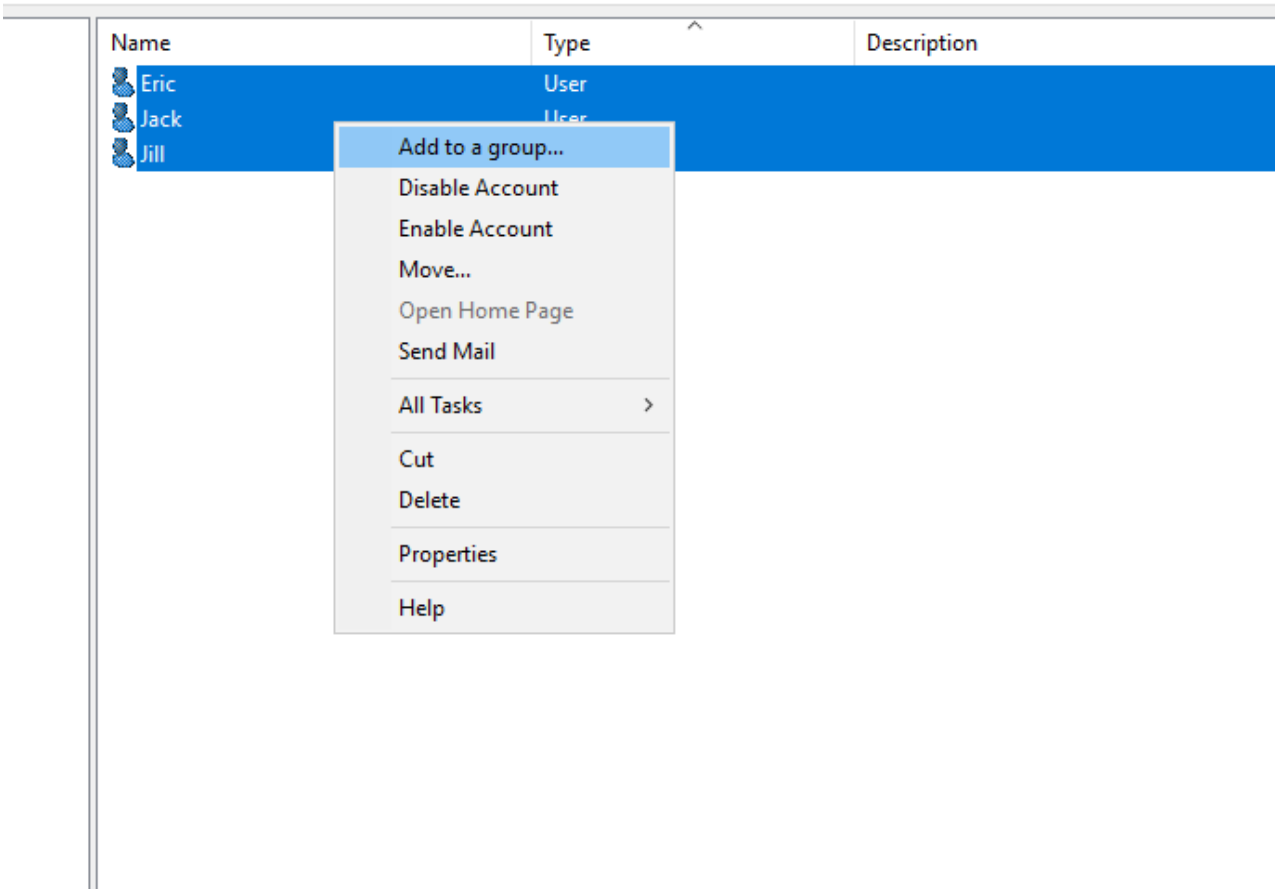
Once you have created the groups, right click and select properties on “Workstation Access”. Then select the “Members” tab.



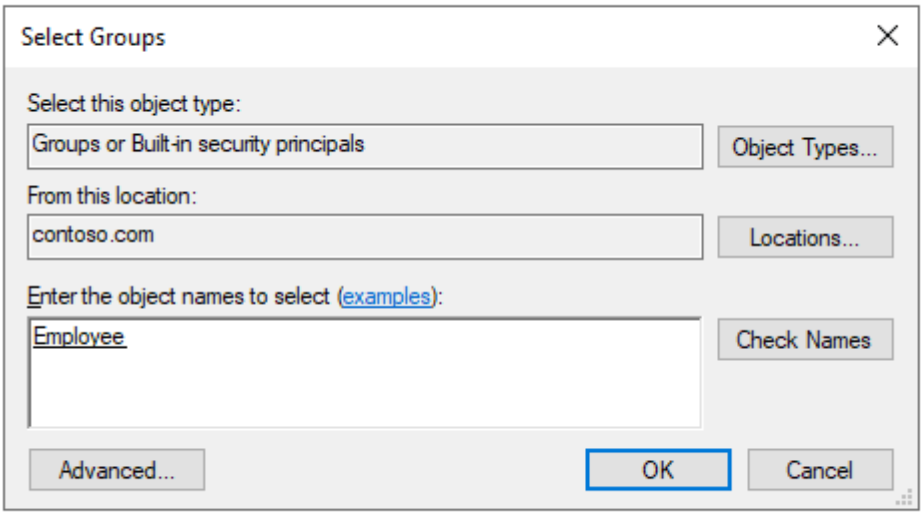
Select “Add” and search for “Employee” and select “Apply” and “OK”.



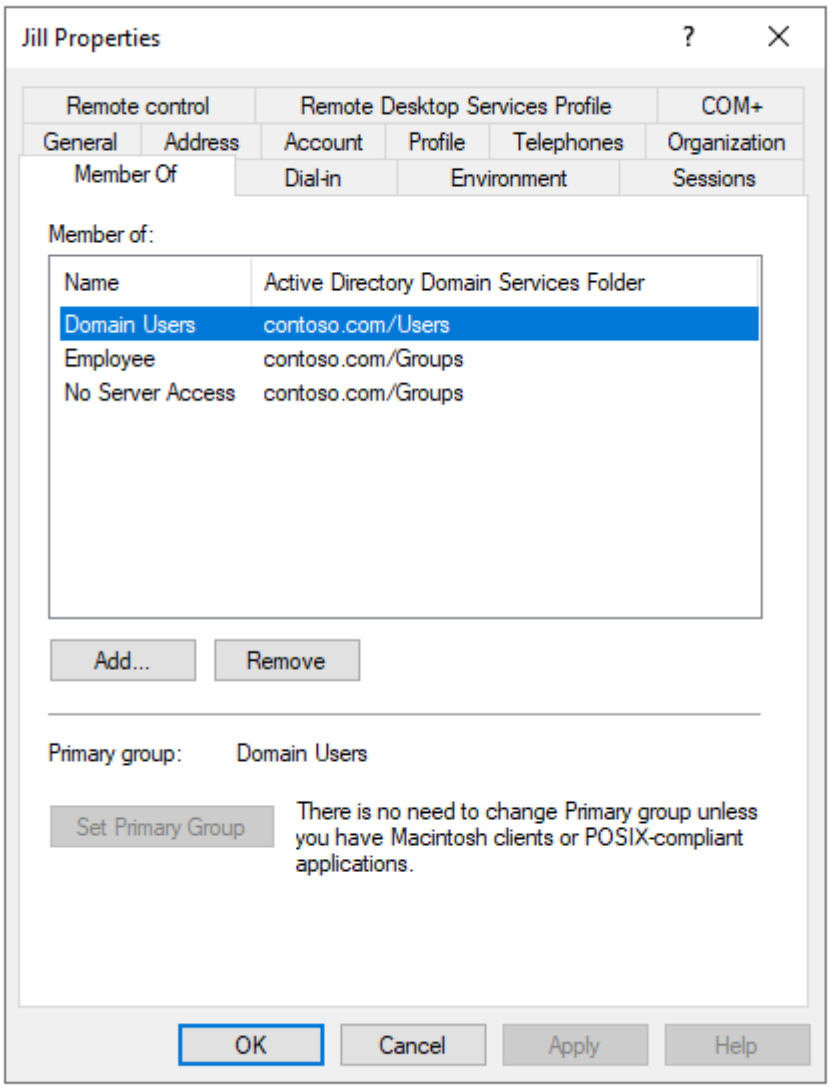
Then navigate to the “Accounts -> Employees -> FTE” OU and select all the user accounts with Ctrl+A, and right click and select “Add to Group”.



Then search for “Employee”, and select “OK”.



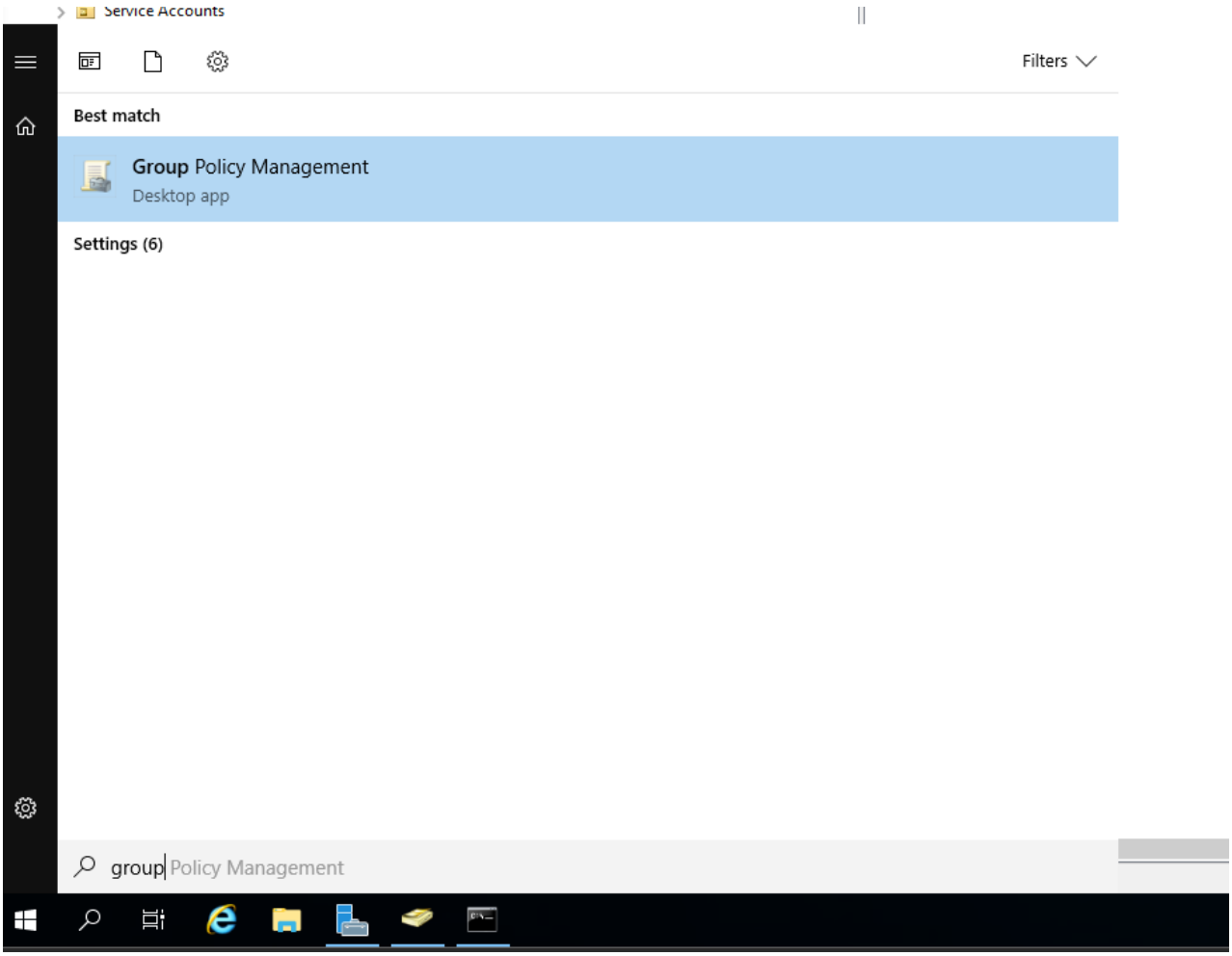
The “Add to Group” operation should then be successful. Opening up their User account and selecting “Members Of”, you should then see “Employees” as an added group member.



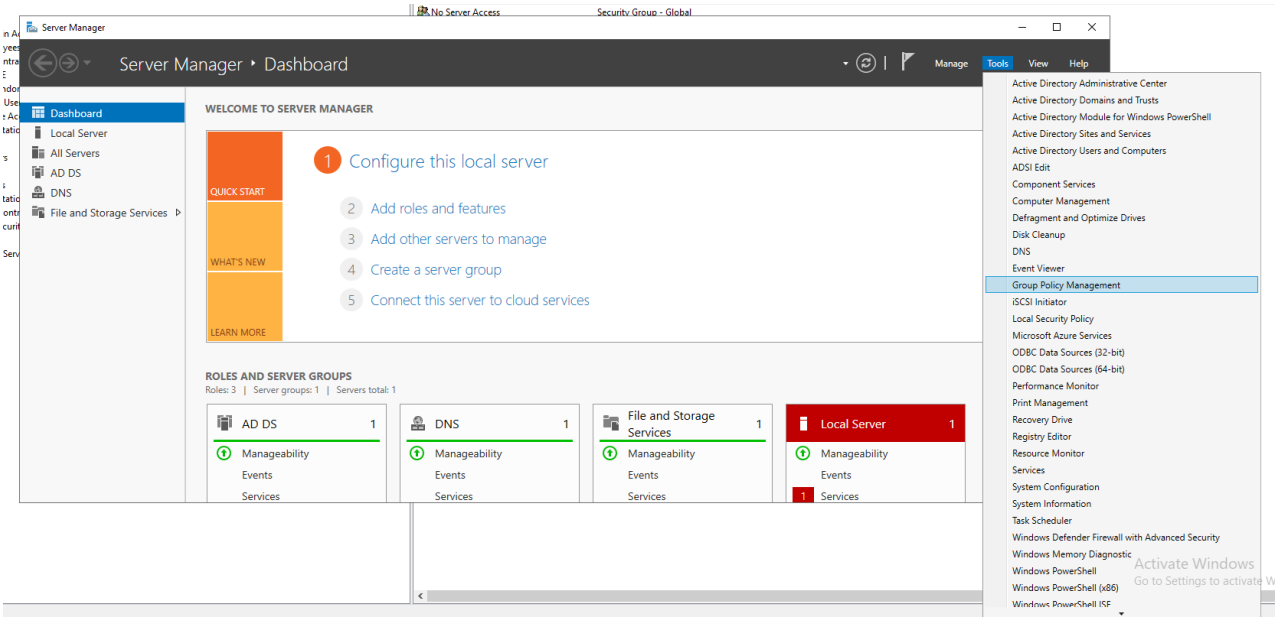
Now that we have added the “Employees” group and assigned all the employees as members, this would be a good candidate to *deny* access to sensitive devices.

Exploring Group Policy Objects

Let’s create a Group Policy Object that denies Employees access to Servers. To access the Group Policy Management settings, search for “Group Policy Management” in the search bar.

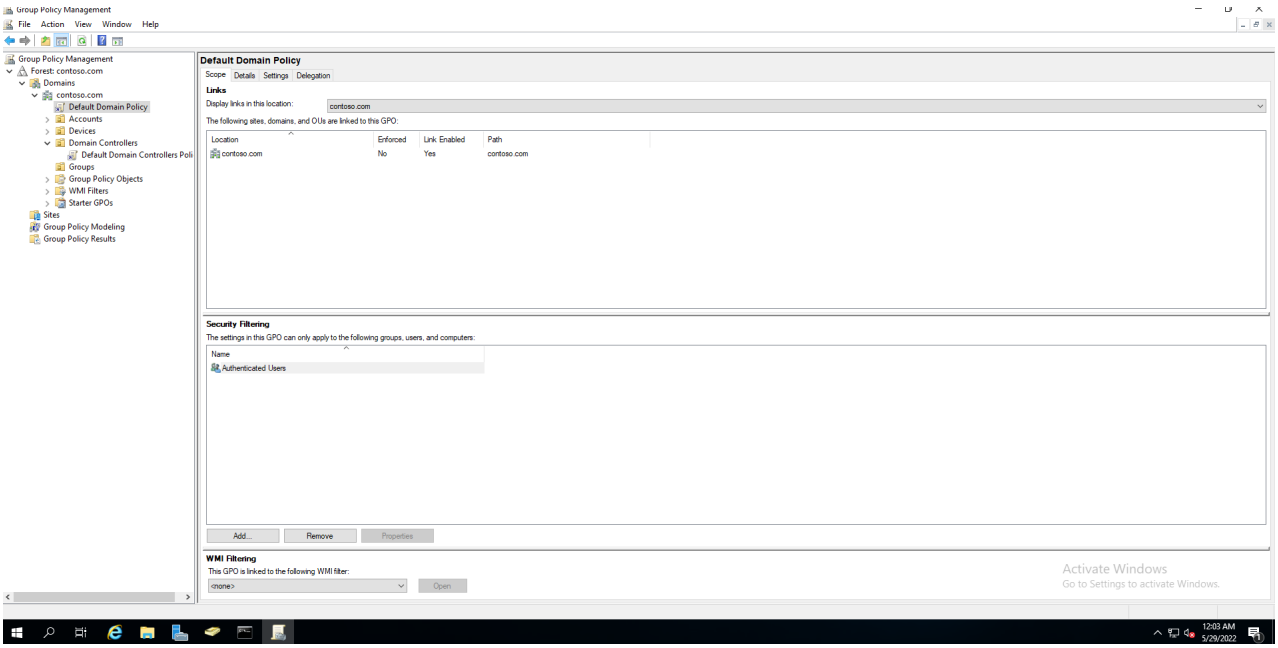


Alternatively, you can open up “Server Manager” and select “Tools” and select “Group Policy Management”.



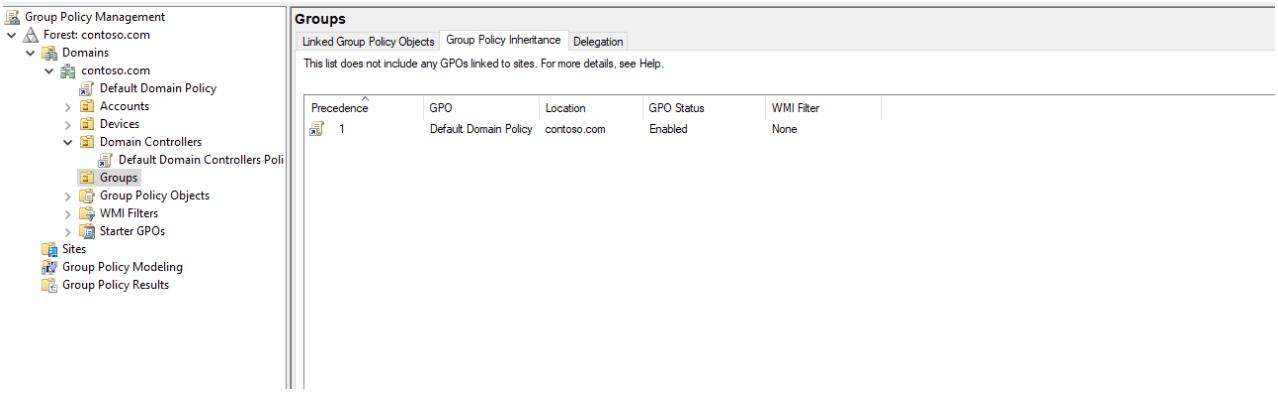
By default there are two group policy objects. They are the following:

- Default Domain Controller Policy
- Default Domain Policy



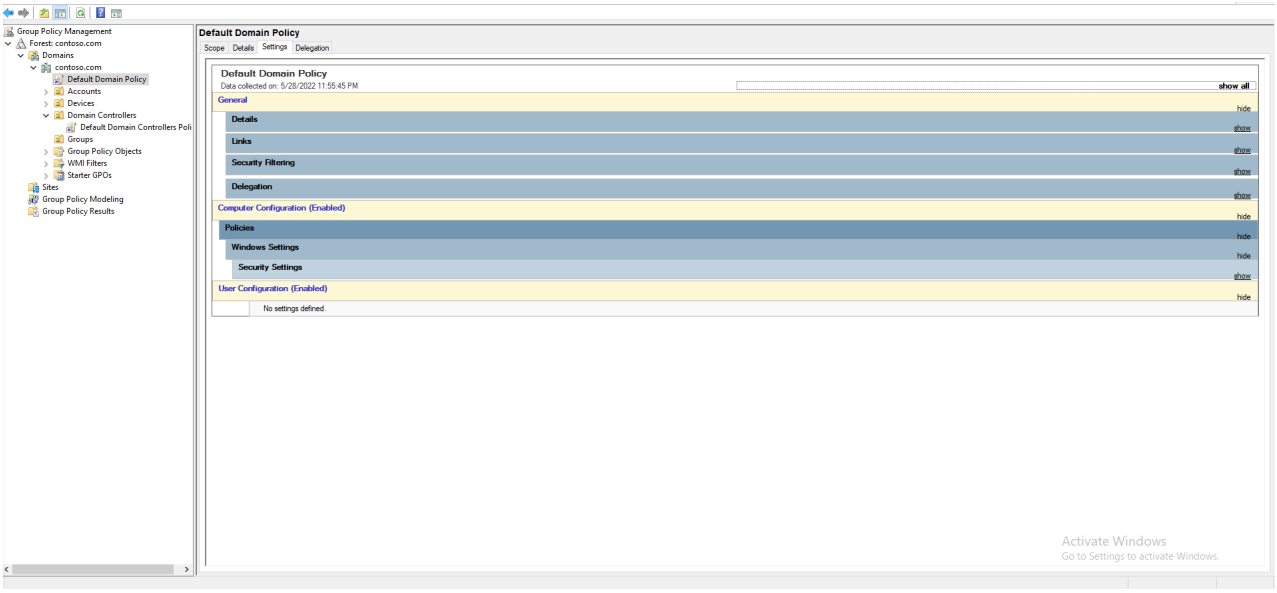
One important thing to note about GPOs is they can be found in the “Group Policy Objects” OU, this will display all the GPOs in the domain. Every GPO lives within “Group Policy Objects” OU, and is essentially sym-linked to the OU we would like to apply it to.

Note that the “Default Domain Policy” is linked to the root of the domain - This means it will recursively apply to all the other OUs in the Domain. We can verify that it is recrusively applying by selecting another OU. For Example, looking within the “Groups” OU and selecting “Group Policy Inheritance”, we can see the Default Domain Policy is applied, even though the Default Domain Policy is not directly linked to this OU.

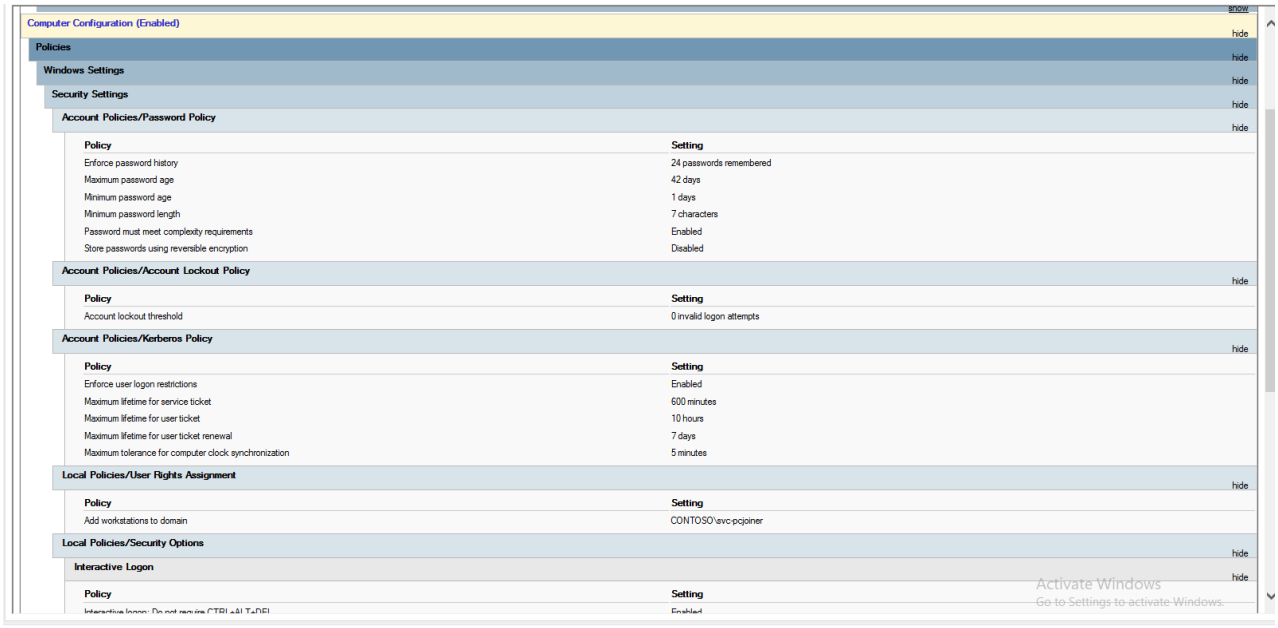


Viewing GPO Settings

When stepping into a new environment, exploring what settings a Group Policy Object applies is important to know, especially for Security Professionals. In order to view the settings, select the Group Policy Object. In this example, we will be using the “Default Domain Policy”. Once that is selected, click the “Settings” tab.



Then select the “Show All” dropdown and scroll down to “Computer Configuration” or “User Configuration”.

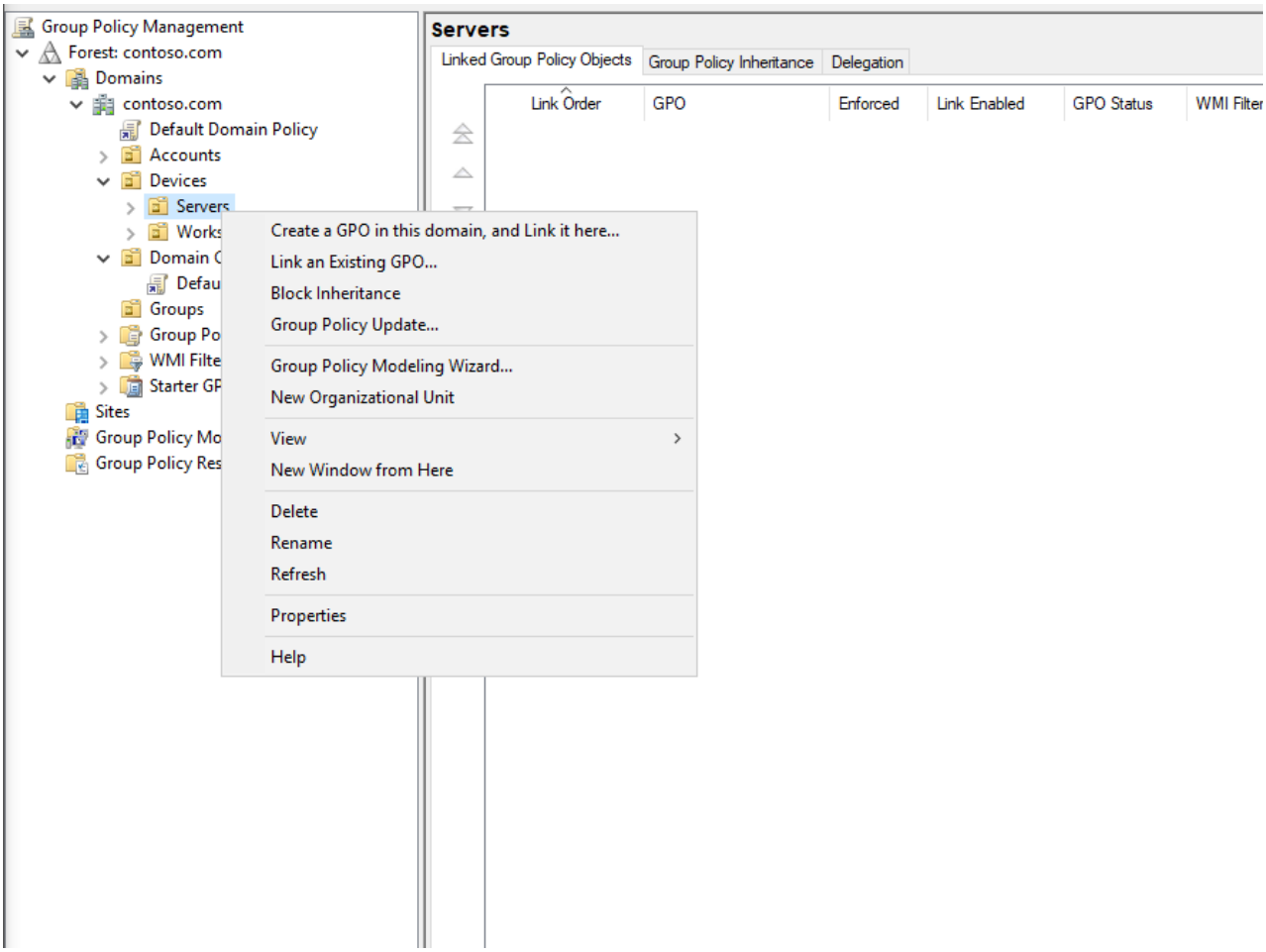


In this specific case, the Default Domain Policy applies password policy settings, Account Lockout Settings, Interactive Logon Settings for the Workstations in the Domain and a whole bunch others!

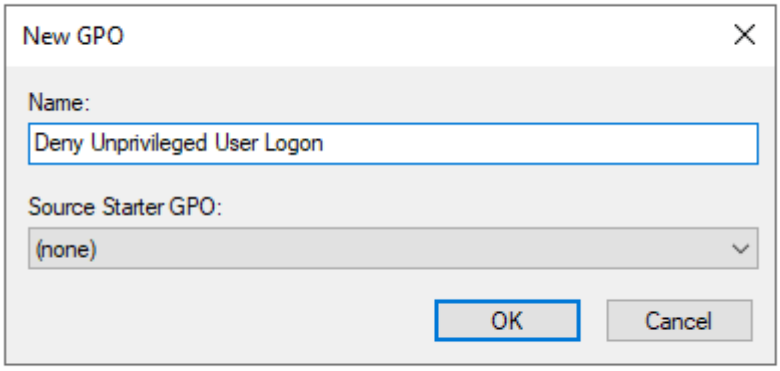
Creating Group Policy Objects

Now that we have a general idea of some of the things Group Policy Objects can influence, we can work on creating our own GPOs.

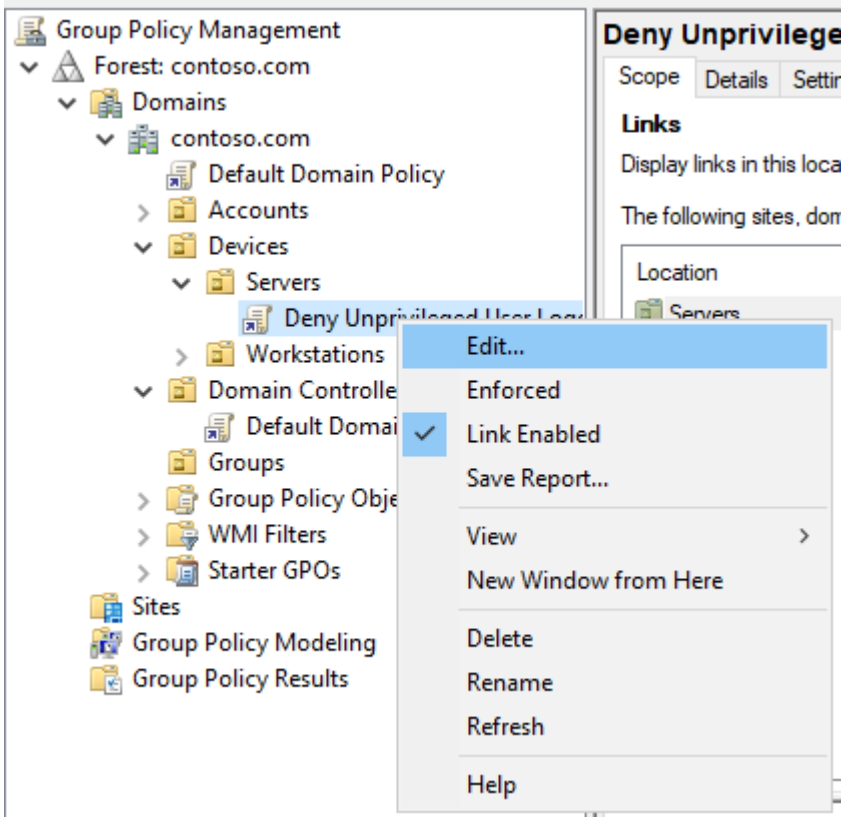
To start, we’re going to select the “Servers” OU. Right click it and select “Create a GPO in this Domain, and link it here…”



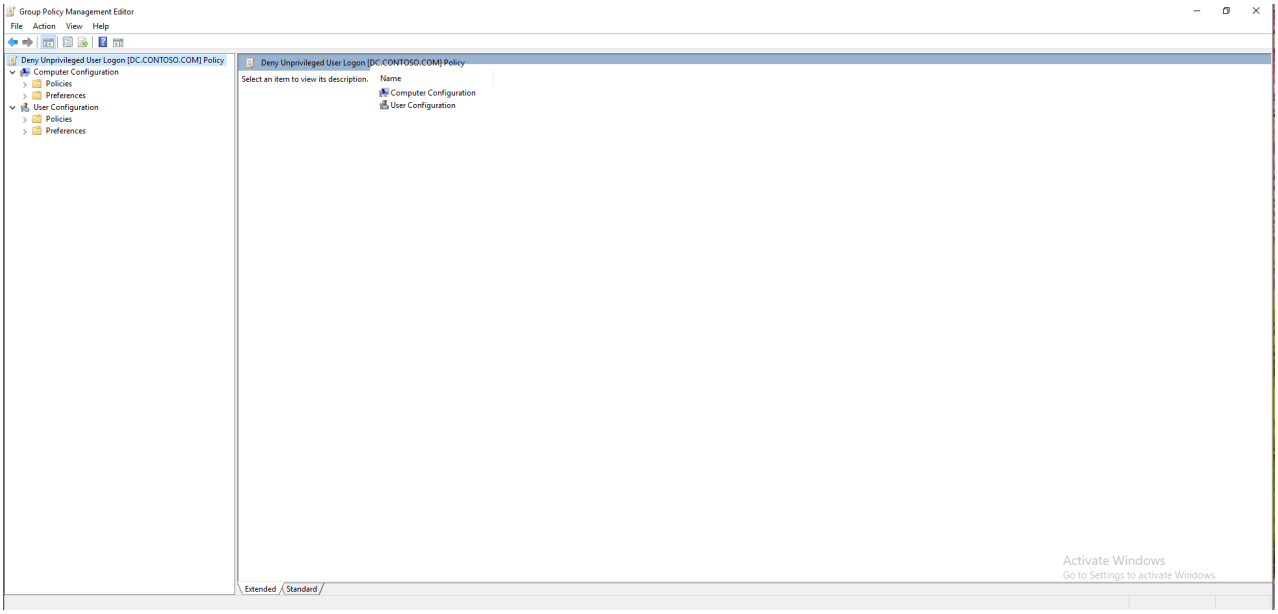
We will name this GPO, “Deny Unprivileged User Logon” and select “OK”.



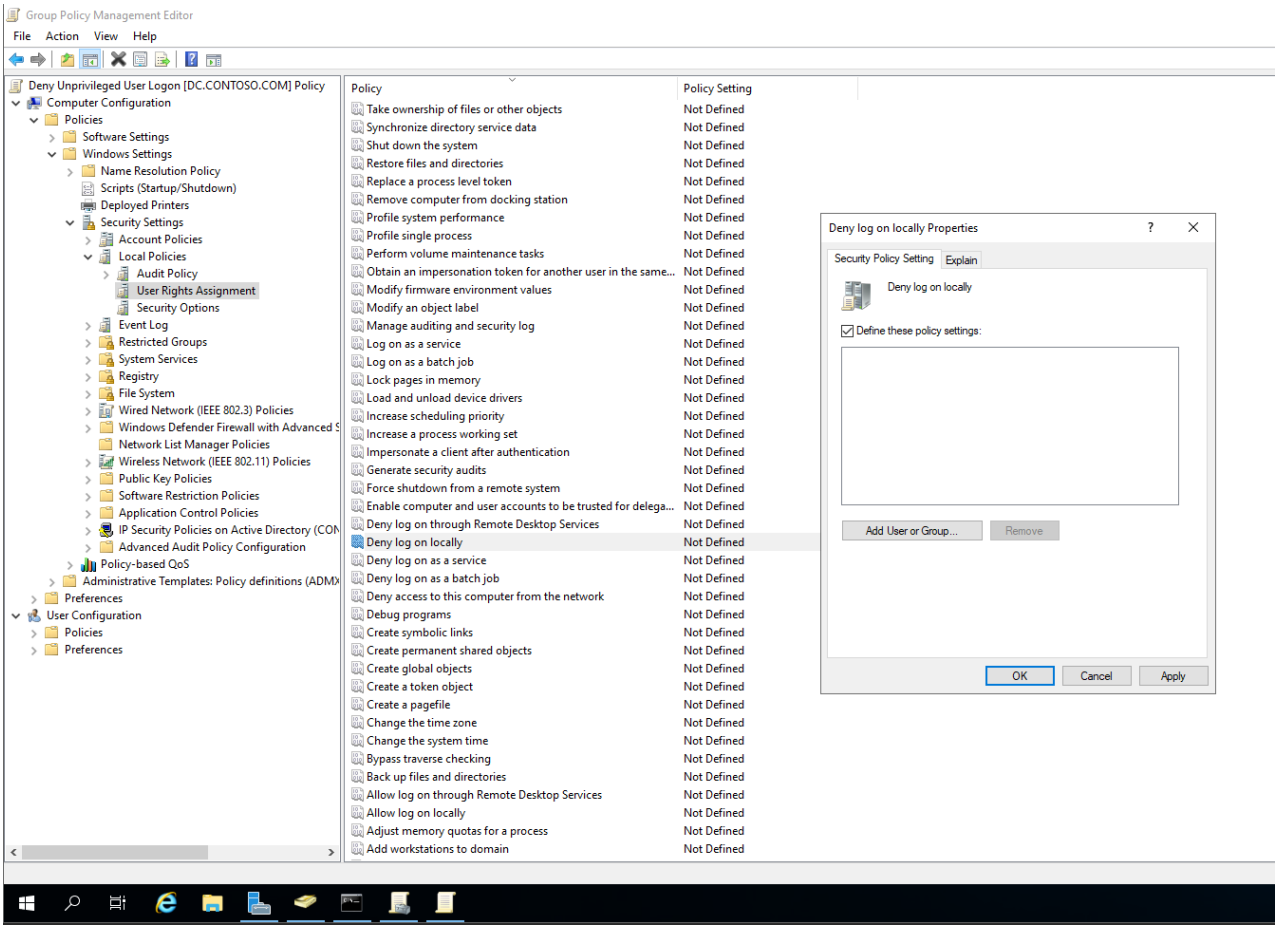
Active Directory will then create the GPO and link it to the specific OU. Now right click on the Group Policy Object and select “Edit”.



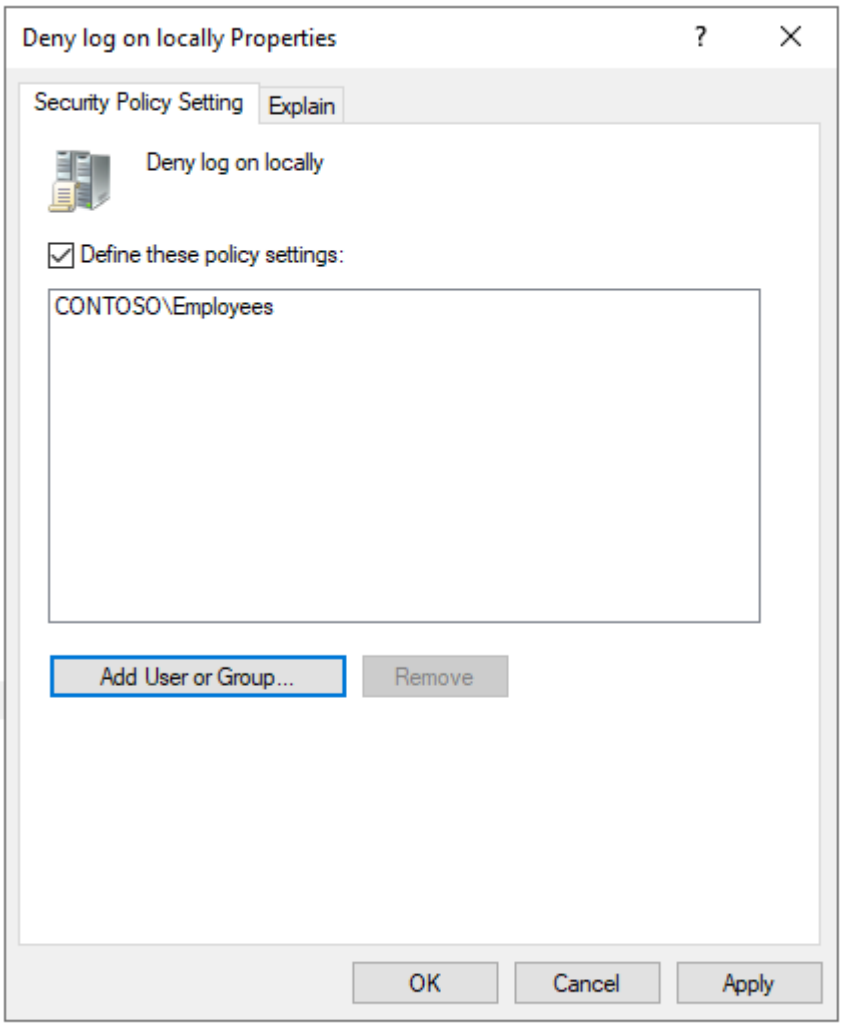
Welcome to the “Group Policy Management Editor”! We finally made it!



We can now work on creating a rule in the Computer Configuration User Rights option that states “Deny Local Log on to X Users”. To do this, expand Computer Configuration -> Policies -> Security Settings -> Local Policies -> User Rights Assignment -> Deny Log on Locally.

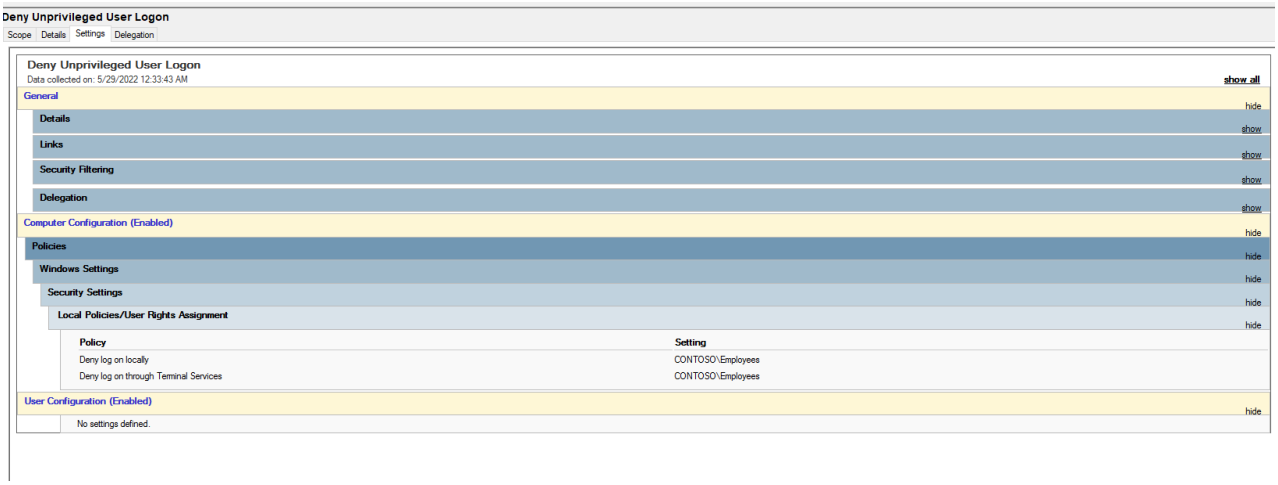


Double click the policy object to define it, then click on “Add User or Group”, then search for “Employees”, select “Apply”, then “OK”.



Repeat this process for “Deny log on through Remote Desktop Services” as well.

After you finished this, close out of the GPO Editor and go to the “Settings” tab and make sure it was applied properly.



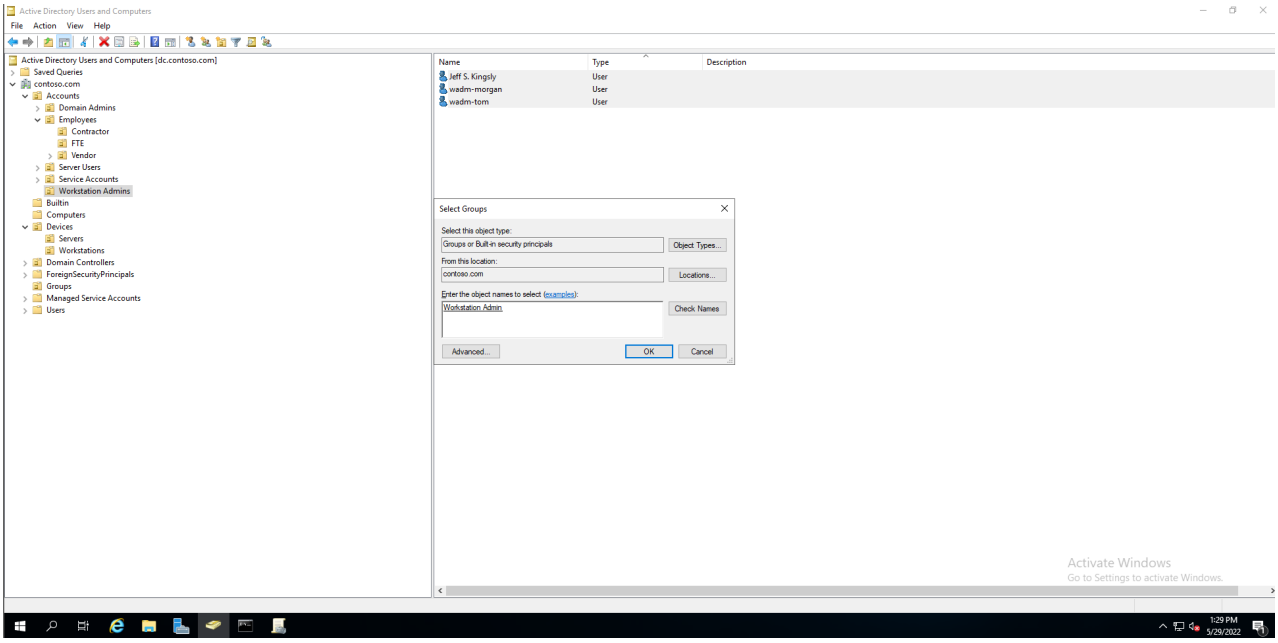
We can see that Employees are denied logon rights to Servers. If you have a Sever object in the OU, you may still be able to log on. If you can, you would then need to execute a `gpupdate /force` command for the Server to pull the latest Group Policy

Objects. It would also be a good idea to create another object that explicitly **allows** log on rights for Server Users as well as granting them local admin rights.

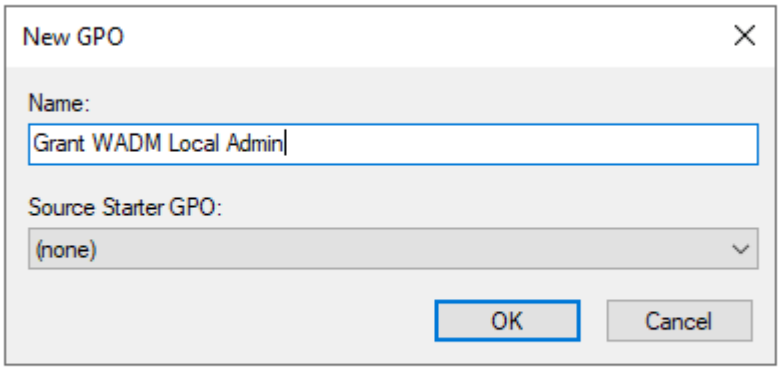
Because we have only gone over setting up a Workstation, let's create a Group Policy Object that targets and affects Workstations. A good candidate for this is creating a Workstation Admin -> Local Admin GPO.

Creating Workstation Admin GPO

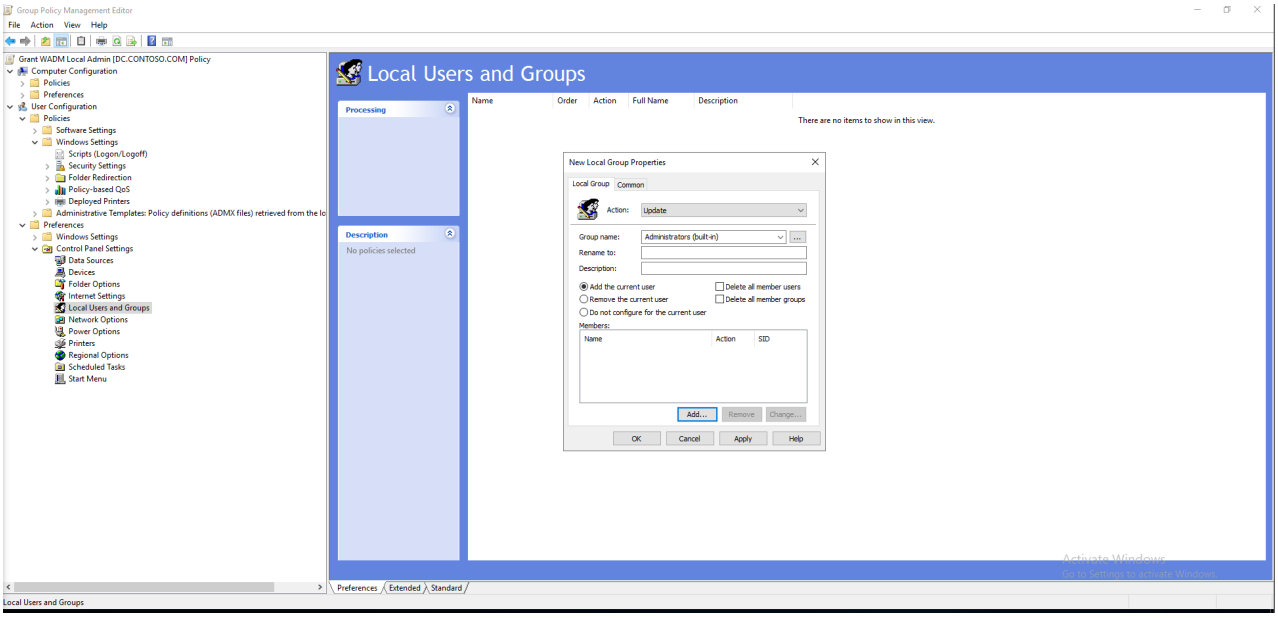
Before we get started, hop over to the "Active Directory Users and Computers" application and go to the "Accounts->Workstation Admins" OU. Highlight all the users in the OU and add them to the "Workstation Admins" group.



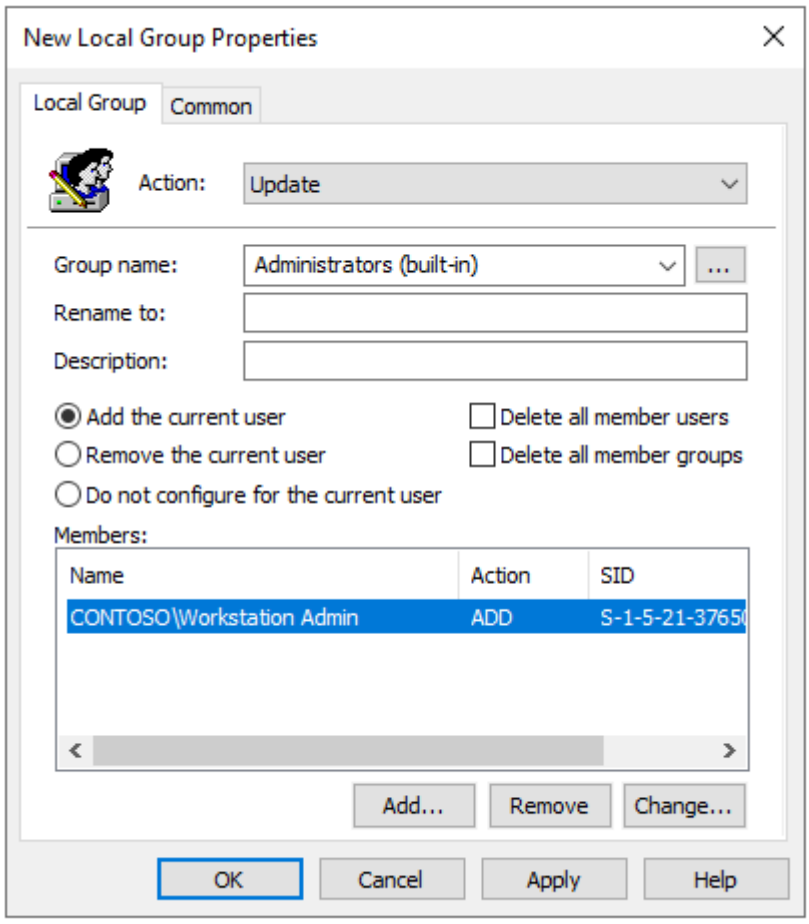
Afterwards, select "OK" and head over to the "Group Policy Management" application and find the "Devices->Workstations" OU. Right click on the OU and create a new Group Policy Object named "Grant WADM Local Admin"



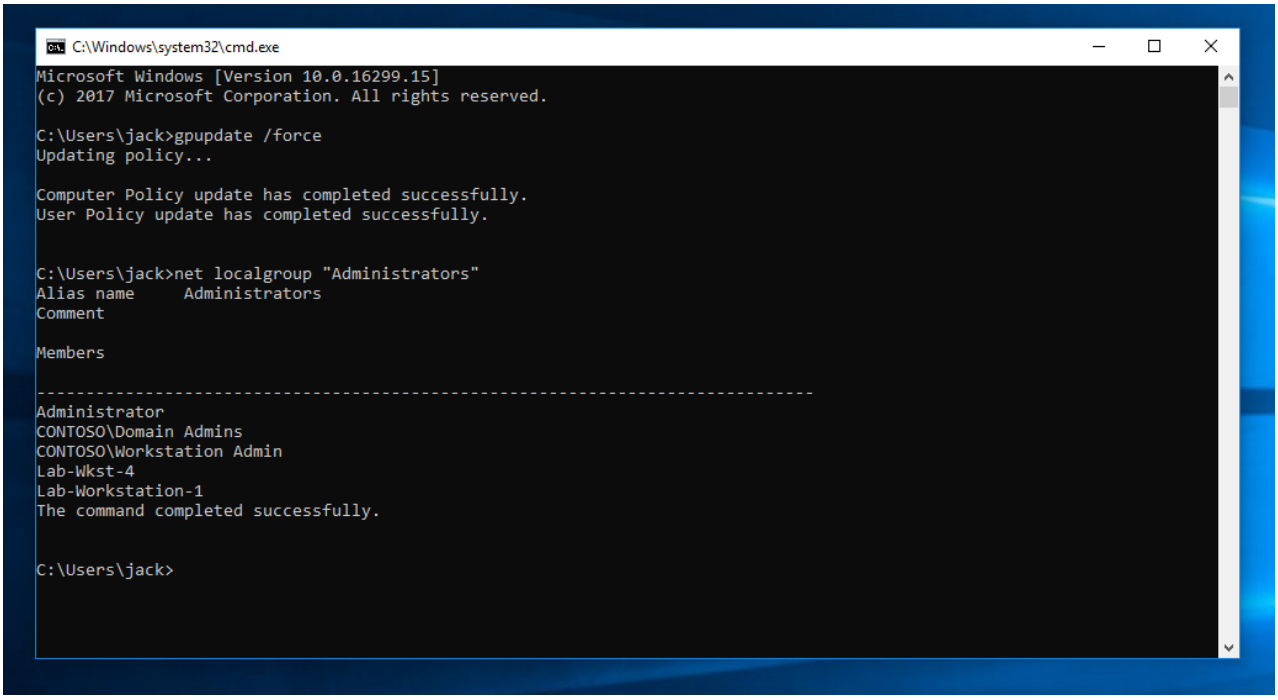
After it is created, right click and edit the object. Expand the “User Configuration” dropdown, select “Preferences”, then “Control Panel Settings”. Then select “Local Users and Groups”. Right click and then select “New Local Group”.



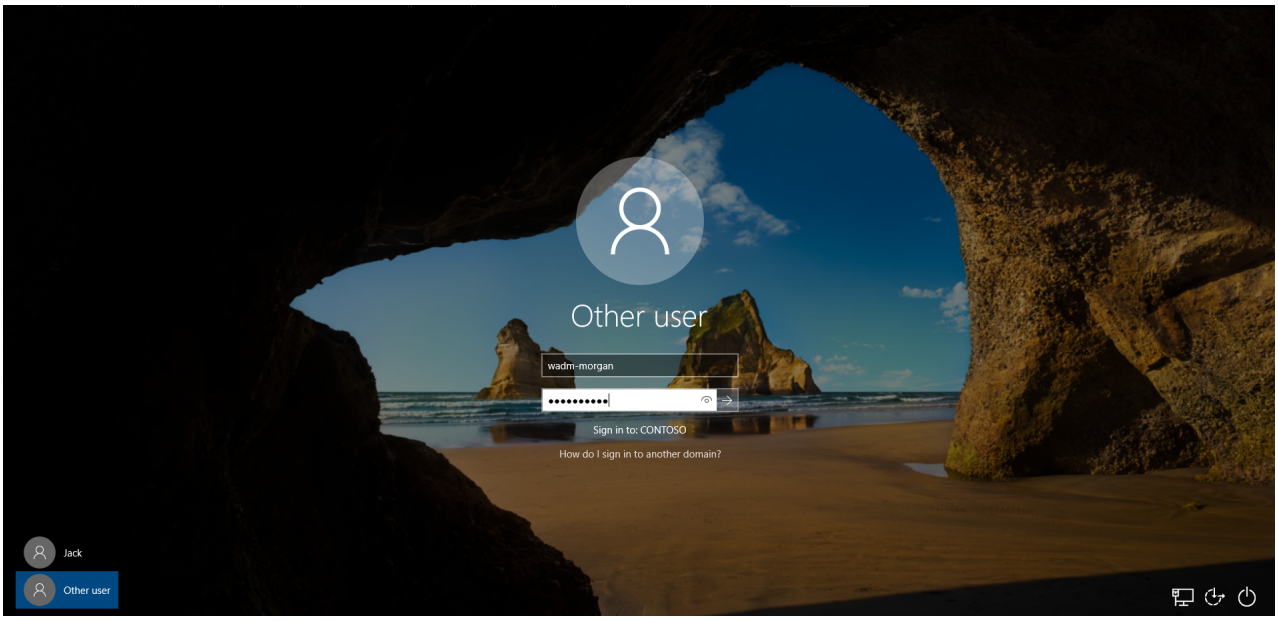
Once the menu is open, select the “Group Name” to “Administrators (Built In)”, then select the “Add the current user” radio button. Select the “Add” button, then add the Workstation Admins group.



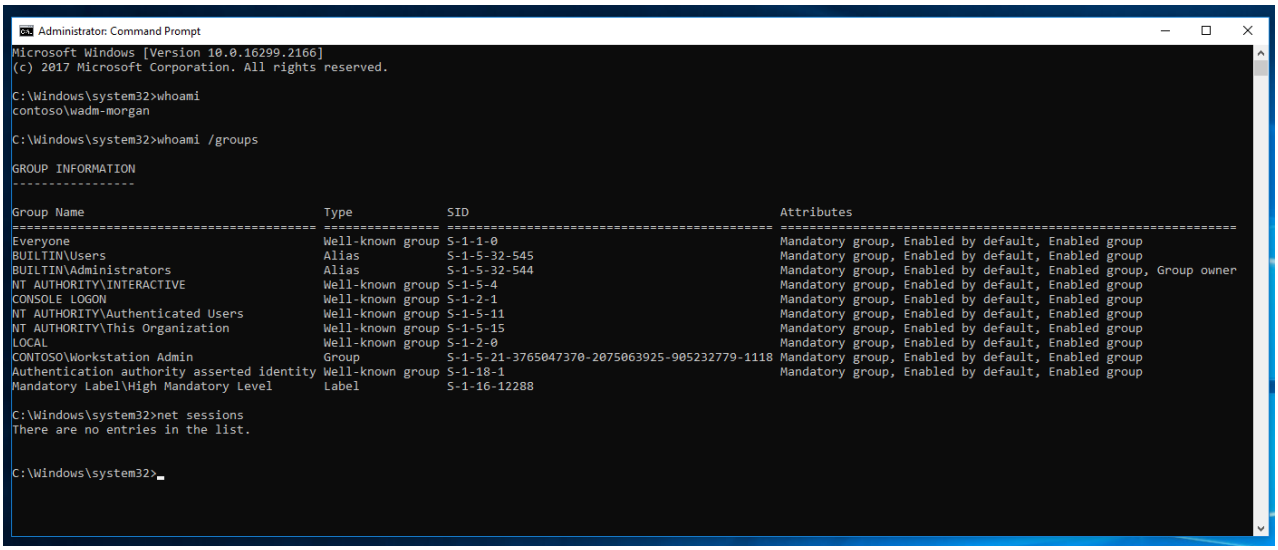
Then select “Apply”, “OK”. The Group Policy Object should now be applied to the Workstations within the domain. Now, log onto a Workstation within the Domain, open up cmd.exe and run a `gpupdate /force`. After the command has completed, run a `net localgroup "Administrators"` and inspect the members of the Administrators group. Does “Workstation Admin” show up? If not, give the Workstation a restart, and execute `gpupdate /force` on the domain controller as well.



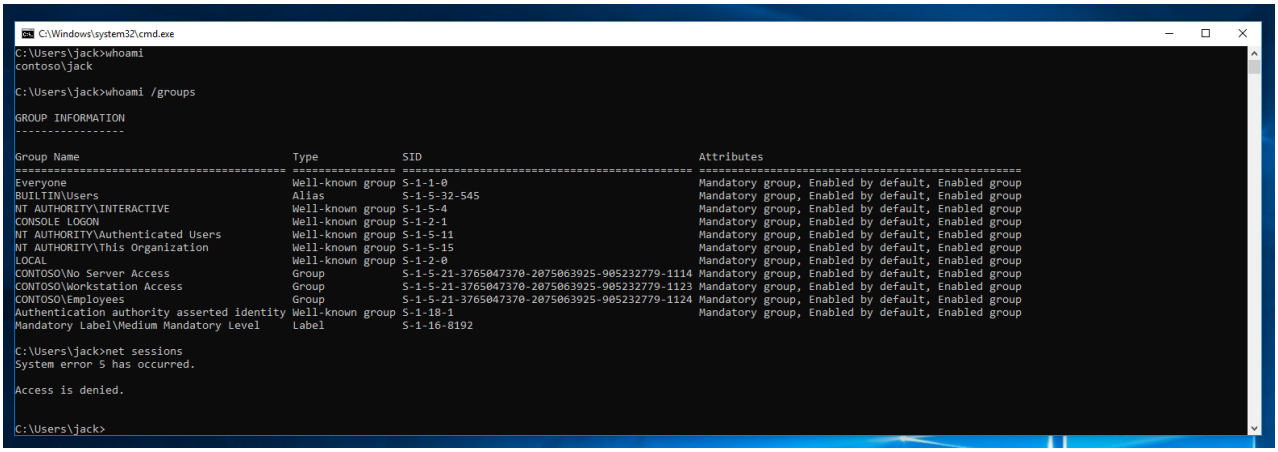
If so, log on with a member of the Workstation Admin group.



We can check this by running a few commands - `whoami`, `whoami /groups`, and `net sessions`. The first command will identify our current user, the second command will verify our current groups, and the last command is a command used to identify SMB Sessions on the device, which can **only** be run by a Local Admin. Once logged in, right click CMD, and select “Run as Administrator” to spawn a high integrity level process.



We can see we are indeed wadm-morgan, we are a member of the BUILTIN\Administrators group on the Workstation, and there are no sessions on the workstation. A failed attempt would look something like so:

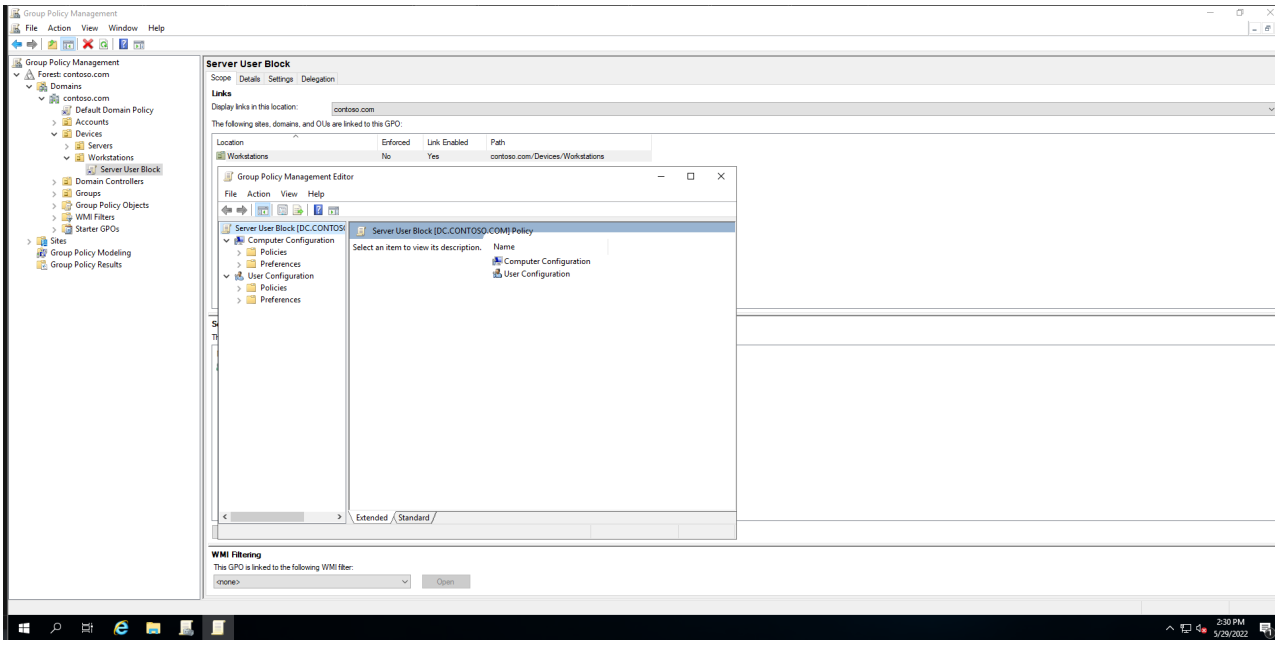


Its worth noting, to inverse the operation you will have to create a GPO to *remove* the Workstation Admins group from the Workstations. That process can be relatively tricky, and is out of the scope of this blog post.

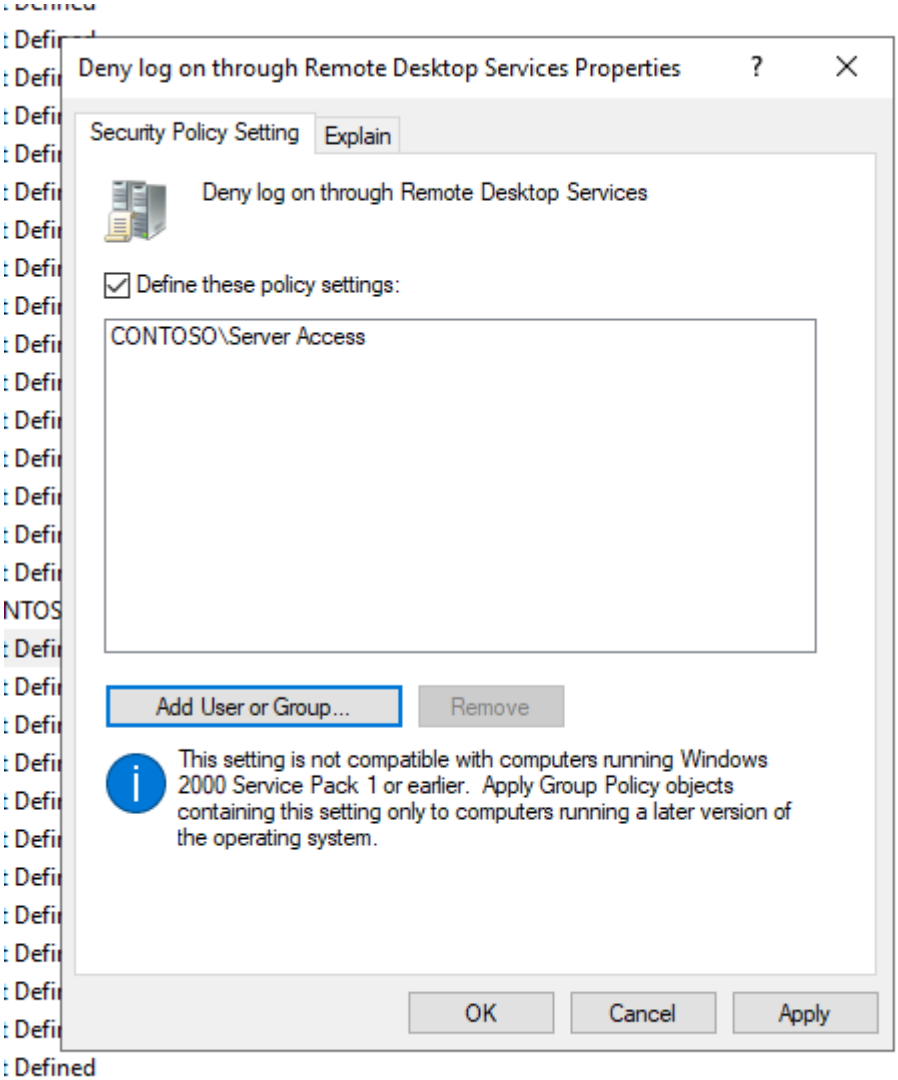
Creating a Server User Block GPO

One thing we may choose to do is restrict tier access, we could do this by blocking sign-in rights to Server Users on Workstations. This process is the same as setting up our Block policy for employees - just swapping the OU and substituting Employees for Server Users. Before we begin, make sure all SU- accounts are in the Server Access group.

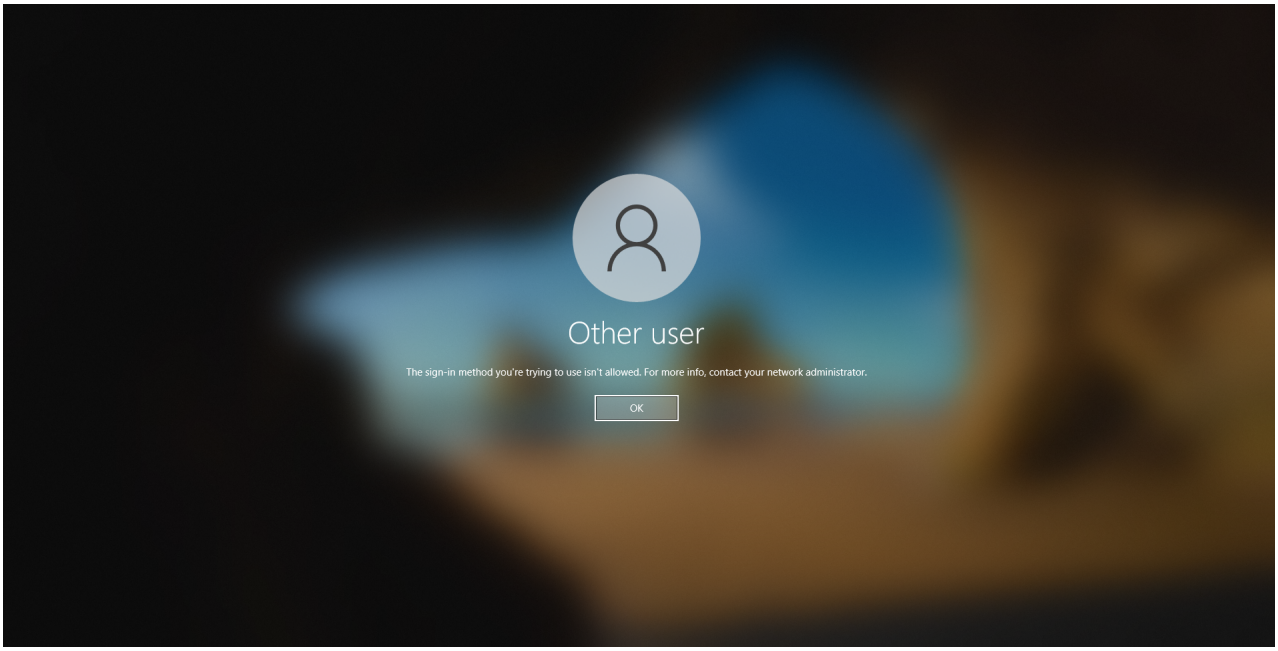
To start, create a new GPO in the Workstation OU named “Server User Block” and edit the Group Policy Object.



Navigate to “Computer Configuration” -> “Policies” -> “Windows Settings” -> “Security Settings” -> “Local Policies” -> “User Rights Assignment” and look for “Deny Log On Locally” and “Deny Log On Through RDP Services”. Define the policy and apply it.



Now we can hop on over to the Workstations and issue a `gpupdate`. Restart the workstation, then try logging in with an SU- account. When attempting to log on, we are presented with an error stating “The sign-in method you’re trying to use isn’t allowed.”



This block is being caused by our newly implemented GPO. The trick is with GPOs is knowing where the policy you want is. The Microsoft MSDN wiki is an excellent resource for identifying if a GPO exists for the thing you want.

With that, it brings us to the end of this blog post. We have laid the foundations for joining other devices to the domain, configuring policies, and delegating rights to accounts. In the next post, we will learn how to install LAPS, learn more about delegating rights to users, creating Service accounts, and much more!

Comments