

Active Directory Automation: User Management & GPO Security

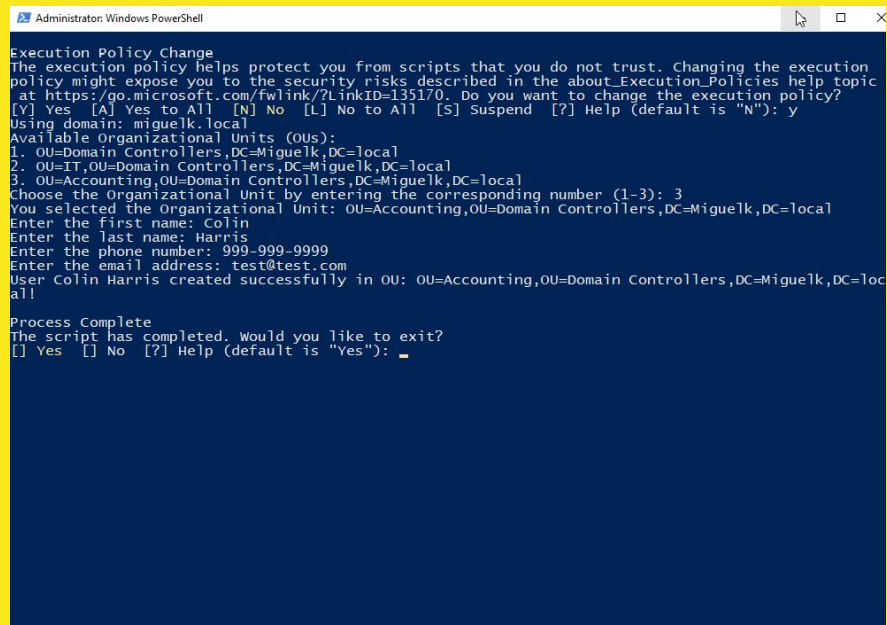
The Active Directory series: Part 2

Onboarding, Offboarding, and Security Policies for OUs with PowerShell

By Miguel K.

Automating User Creation with PowerShell

- First, I made sure the **Active Directory module** was imported using the command:
`Import-Module ActiveDirectory`.
- I developed a PowerShell script to automate the creation of Active Directory users.
- The script prompts me for specific user details, including (name ,organizational unit, email,etc.) and other necessary information.
- Based on the input, users will be automatically assigned to the correct **Organizational Unit (OU)**, ensuring proper categorization, and other details are entered as well.



```
Administrator: Windows PowerShell

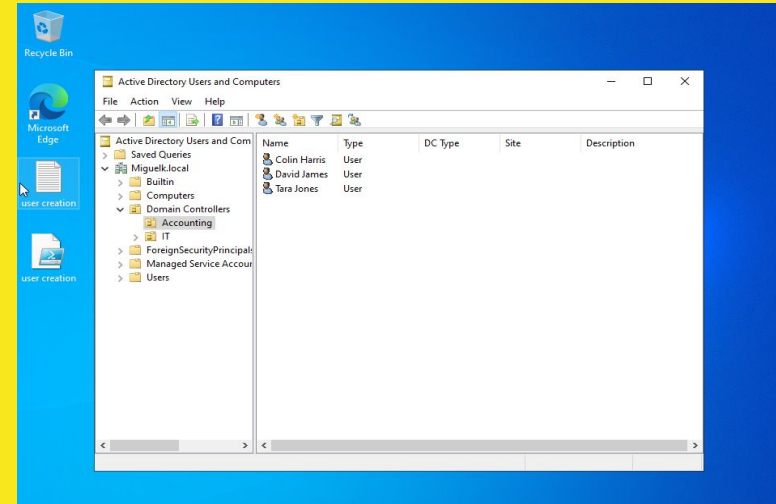
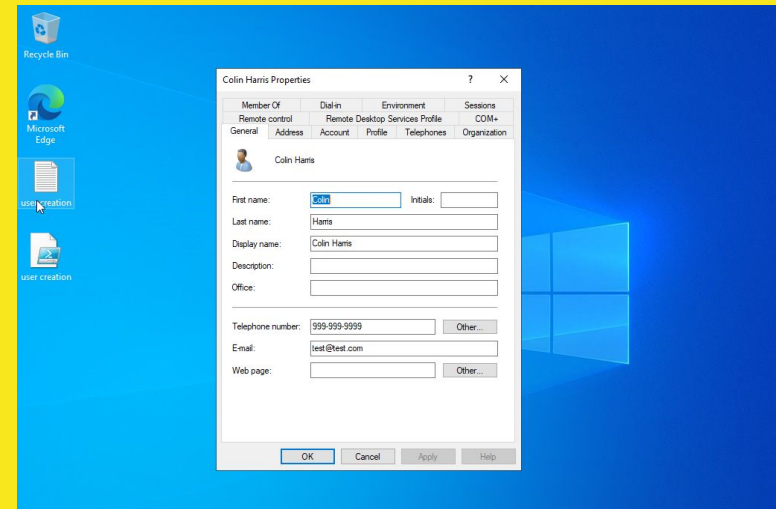
Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose you to the security risks described in the about_Execution_Policies help topic at https://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): y

Using domain: miguelk.local
Available Organizational Units (OUs):
1. OU=Domain Controllers,DC=Miguelk,DC=local
2. OU=IT,OU=Domain Controllers,DC=Miguelk,DC=local
3. OU=Accounting,OU=Domain Controllers,DC=Miguelk,DC=local
Choose the Organizational Unit by entering the corresponding number (1-3): 3
You selected the Organizational Unit: OU=Accounting,OU=Domain Controllers,DC=Miguelk,DC=local
Enter the first name: Colin
Enter the last name: Harris
Enter the phone number: 999-999-9999
Enter the email address: test@test.com
User Colin Harris created successfully in OU: OU=Accounting,OU=Domain Controllers,DC=Miguelk,DC=local
all!

Process Complete
The script has completed. Would you like to exit?
[] Yes [] No [?] Help (default is "Yes"): _
```

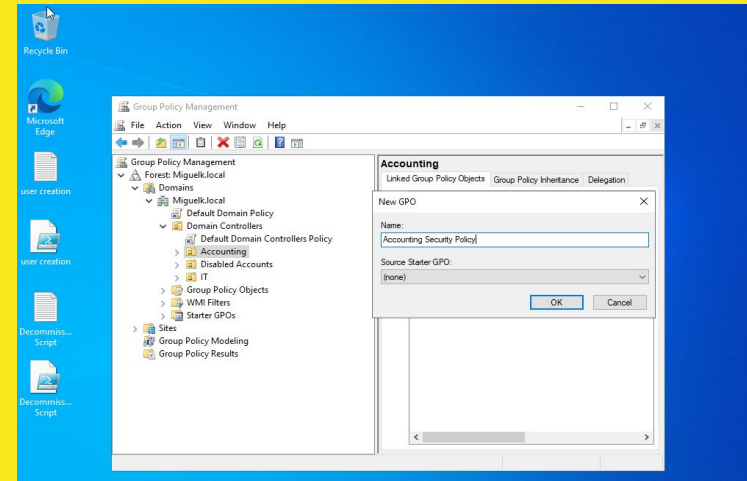
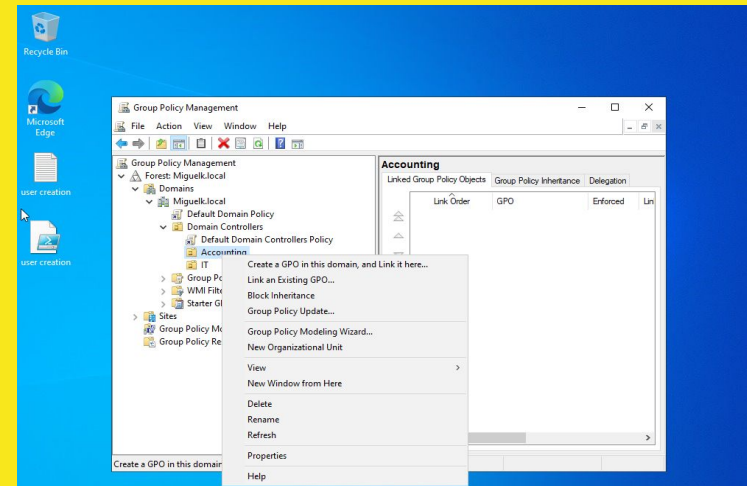
Active Directory Onboarding Results

- After running the PowerShell script, the newly created user appears in Active Directory.
- The user is placed in the correct **Organizational Unit (OU)**, which in this case is **Accounting**.
- All the information entered during the script execution, such as user details, is reflected in the user's Active Directory profile.



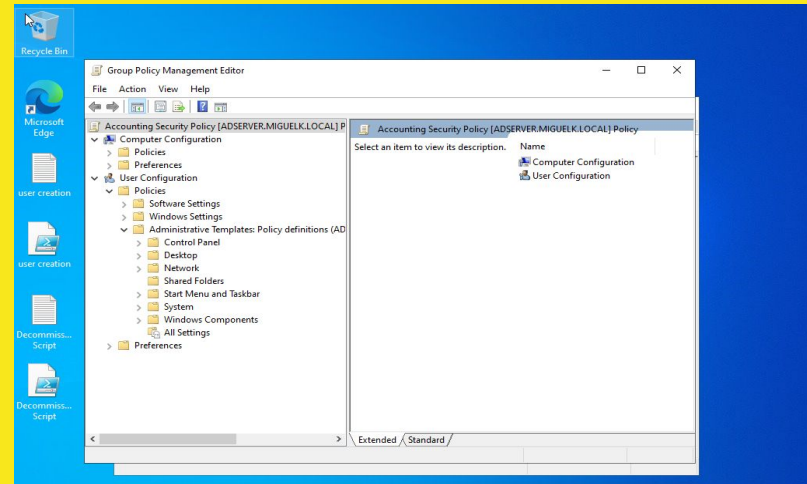
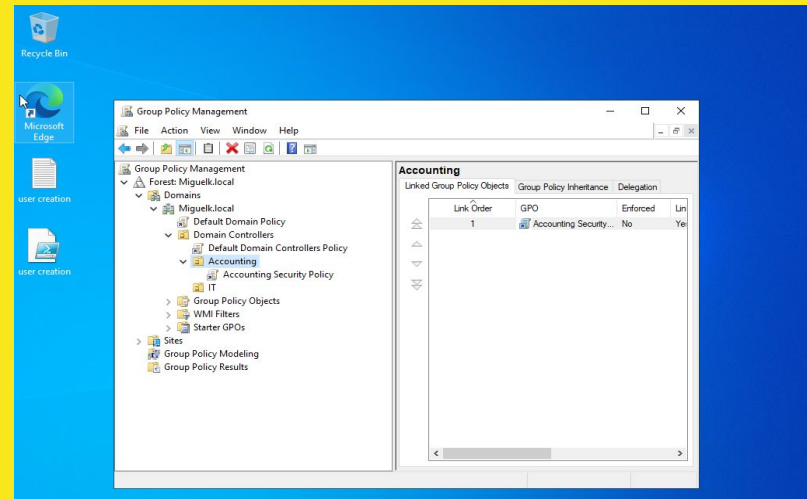
Creating a GPO for the Accounting OU

- At this stage, we are creating a Group Policy Object (GPO) specifically for the **Accounting** Organizational Unit (OU).
- The GPO is created and given a descriptive name to reflect its purpose, ensuring that security policies and settings are applied appropriately to the users in the Accounting OU.
- The GPO is named **Accounting Security Policy** to reflect its purpose of applying security settings to users within the Accounting OU.



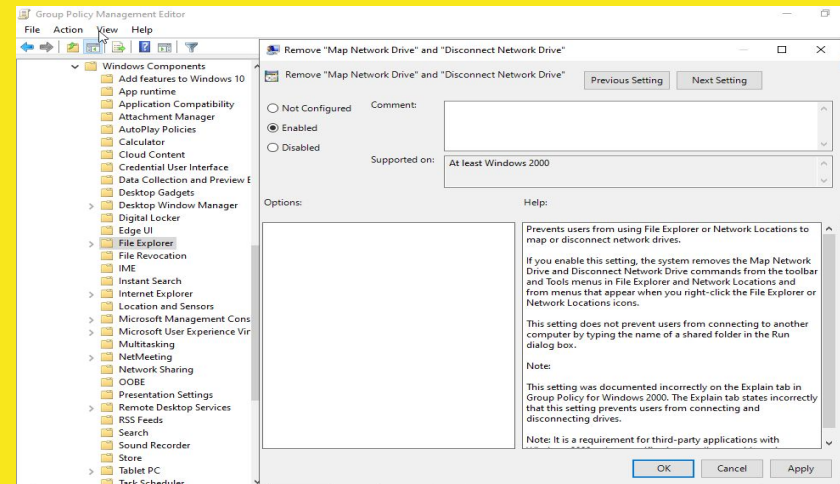
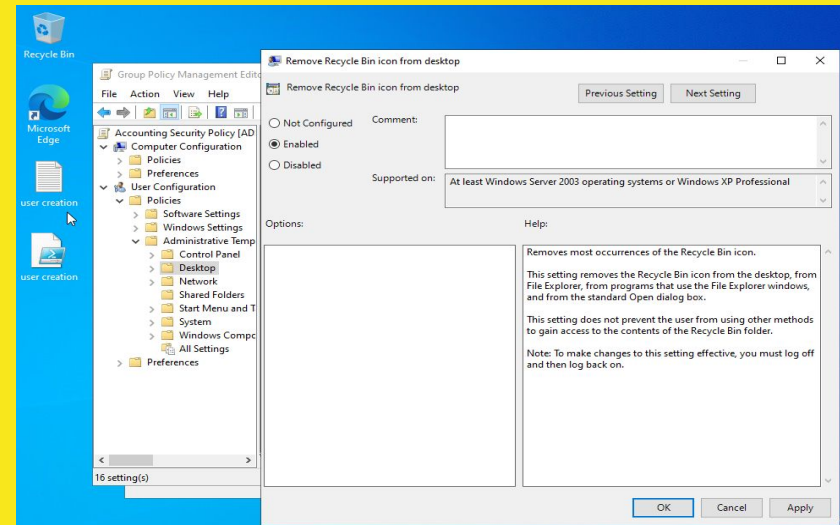
Accessing the Group Policy Management Editor

- Now that the **Accounting Security Policy** GPO has been created, right-clicking on it and selecting **Edit** opens the **Group Policy Management Editor**.
- In the editor, we can configure policies. I selected **User Configuration**, then **Policies**, followed by **Administrative Templates** to set specific user policies.
- You also have the option to edit **Computer Configuration** policies if needed, giving flexibility to apply settings based on either users or computers.



Security Policy Implementation for Accounting OU

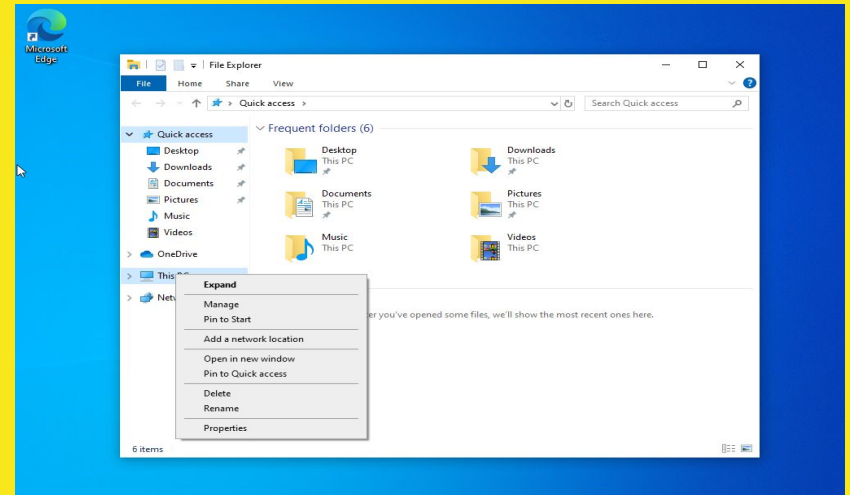
- I implemented a security policy that **removes the Recycle Bin** from users' desktops in the **Accounting Organizational Unit (OU)** to prevent the accidental or unauthorized deletion of **sensitive files**.
- I also configured a policy to **disable the ability to map network drives** from the **Tools menu in File Explorer**, restricting access to network resources and ensuring only authorized users can connect to shared drives.



GPO Security Settings

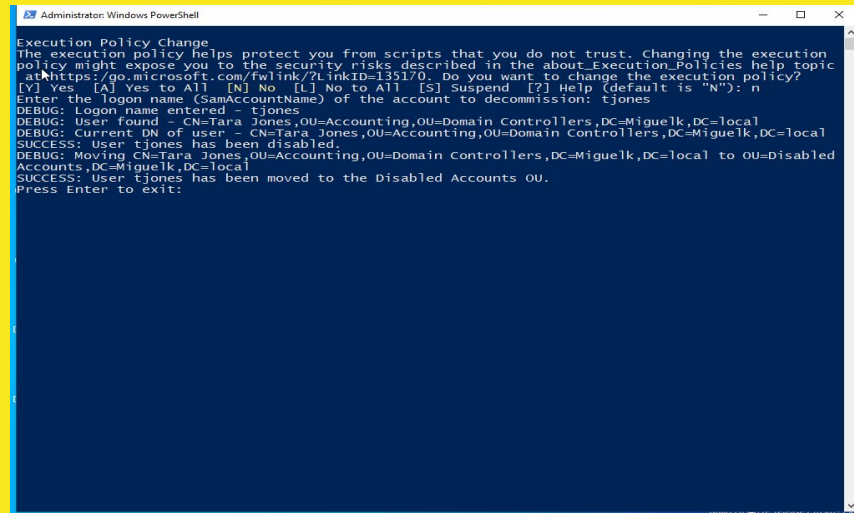
for Accounting OU

- The first image highlights the **absence of the Recycle Bin** from the desktop, showcasing the policy that prevents users from deleting files without proper authorization.
- The second image illustrates that users can no longer **right-click the "This PC" icon to map network drives**, demonstrating the restriction on accessing network resources.
- These settings are **applied only to users in the Accounting OU**, ensuring that the security policies are specific to this group while leaving other users unaffected.



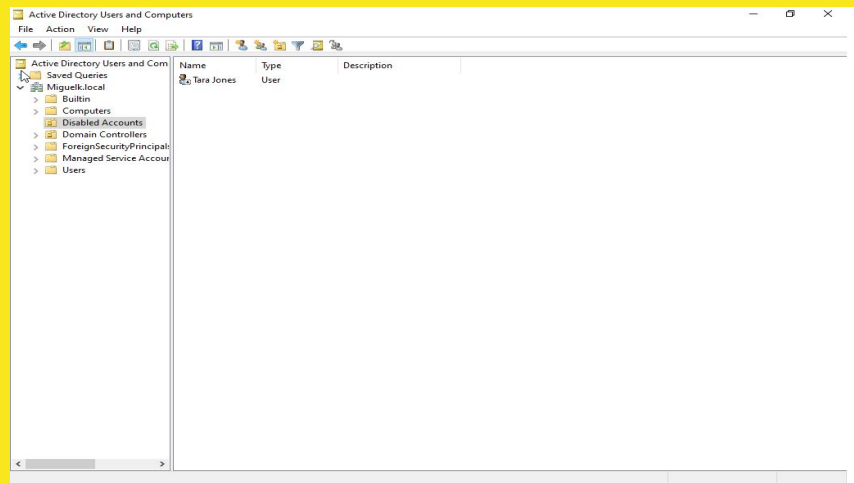
Disabling a User Account in Active Directory

- This script automates the process of disabling the **test user** account in **Active Directory**. For this example, the test user is **Tara Jones**, who will no longer be able to log in or access resources.
- Once disabled, the script moves **Tara Jones'** account to the "**Disabled Accounts**" **Organizational Unit (OU)**. This ensures that inactive accounts are properly organized, making **user management** more efficient.
- The script also keeps a record of the disabled account for future reactivation. This allows administrators to easily re-enable the account when necessary, ensuring proper tracking and management.



```
Administrator: Windows PowerShell

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution
policy might expose you to the security risks described in the about Execution Policies help topic
at https://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): n
Enter the logon name (SamAccountName) of the account to decommision: tjones
DEBUG: Logon name entered - tjones
DEBUG: User found - CN=Tara Jones,OU=Accounting,OU=Domain Controllers,DC=Miguelk,DC=local
DEBUG: Current DN of user - CN=Tara Jones,OU=Accounting,OU=Domain Controllers,DC=Miguelk,DC=local
SUCCESS: User tjones has been disabled.
DEBUG: Moving CN=Tara Jones,OU=Accounting,OU=Domain Controllers,DC=Miguelk,DC=local to OU=Disabled
Accounts,DC=Miguelk,DC=local
SUCCESS: User tjones has been moved to the Disabled Accounts OU.
Press Enter to exit:
```



Project Summary

In this project, I utilized PowerShell to automate user management tasks in Active Directory, ensuring a streamlined approach to both user creation and security policy implementation.

First, I ensured the Active Directory module was imported and developed a script to automate the creation of user accounts. The script prompts for essential user details, assigning them to the correct Organizational Unit (OU), and creating a new user in Active Directory. I also created a Group Policy Object (GPO) specifically for the Accounting OU, applying security policies such as removing the Recycle Bin and restricting network drive mapping to secure sensitive resources.

Additionally, I automated the process of disabling user accounts and moving them to the "Disabled Accounts" OU. This helps keep inactive accounts organized and simplifies user management, allowing for future reactivation if needed.

This project gave me hands-on experience in managing Active Directory, creating user accounts, and applying security policies. I learned how to automate tasks like user creation and account disabling using PowerShell, as well as how to organize users into the correct Organizational Units. Additionally, I gained practical experience working with Group Policy Objects (GPOs) to apply security settings specific to certain groups.

Overall, this project helped me strengthen my skills in Active Directory management, automation, and security, providing me with valuable experience for future IT tasks.