Unlock AD accounts with PowerShell

https://4sysops.com/archives/unlock-ad-accounts-with-powershell/

IT admins or helpdesk teams often use graphical tools, such as Active Directory Users and Computers (ADUC), to unlock AD accounts. However, it is usually easier to unlock multiple AD accounts with PowerShell.

Contents

1. Prerequisites
2. Install RSAT tools
3. Search-ADAccount cmdlet
4. Unlock-ADAccount cmdlet
5. Unlock multiple AD accounts

Prerequisites

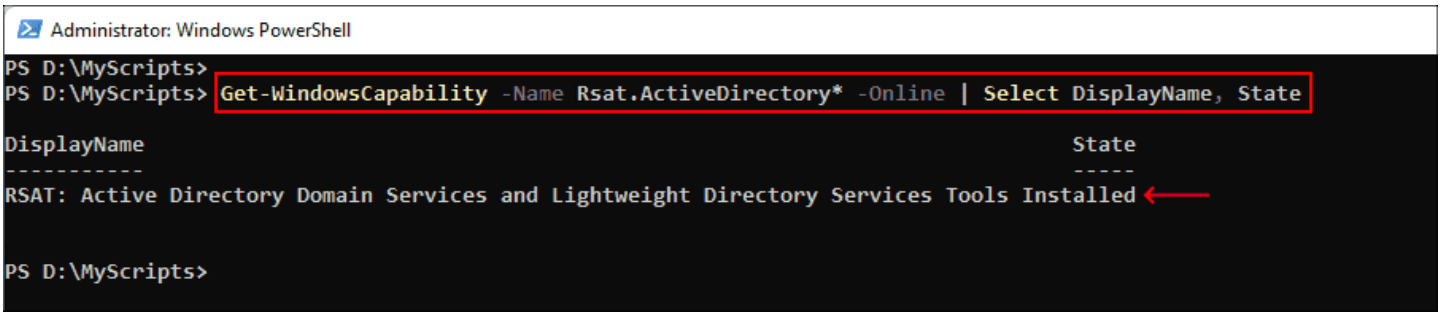
- A domain-joined computer running Professional or Enterprise edition of Windows 10 or Windows 11.
- Active Directory PowerShell module installed on the computer.

Install RSAT tools

The remote server administration tools (RSAT) help admins remotely manage Windows Server roles from a computer running Windows 10/11. RSAT is a large collection of admin tools used to manage most server roles. However, for the purpose of this post, we are only interested in the Active Directory (AD) PowerShell module.

To check whether the AD PowerShell module is installed, launch an elevated PowerShell console, and run the following command:

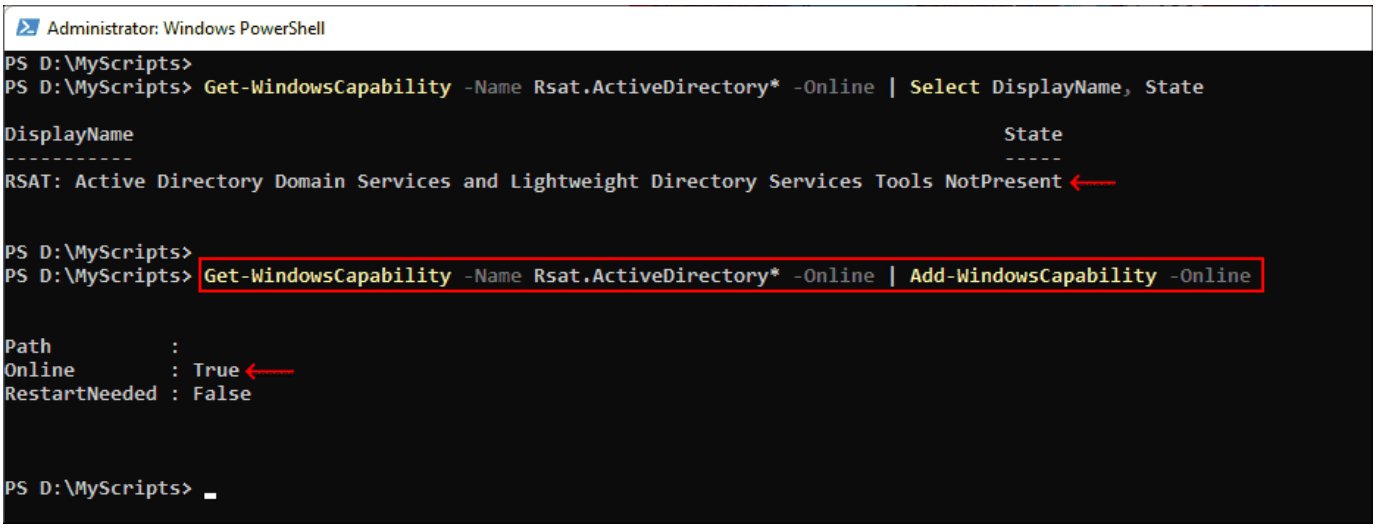
```
Get-WindowsCapability -Name Rsat.ActiveDirectory* -Online | Select DisplayName, State
```



Checking whether the Active Directory PowerShell module is installed

If you see *State = Installed*, you're all set. If you see *State = NotPresent*, run the following command to install the module:

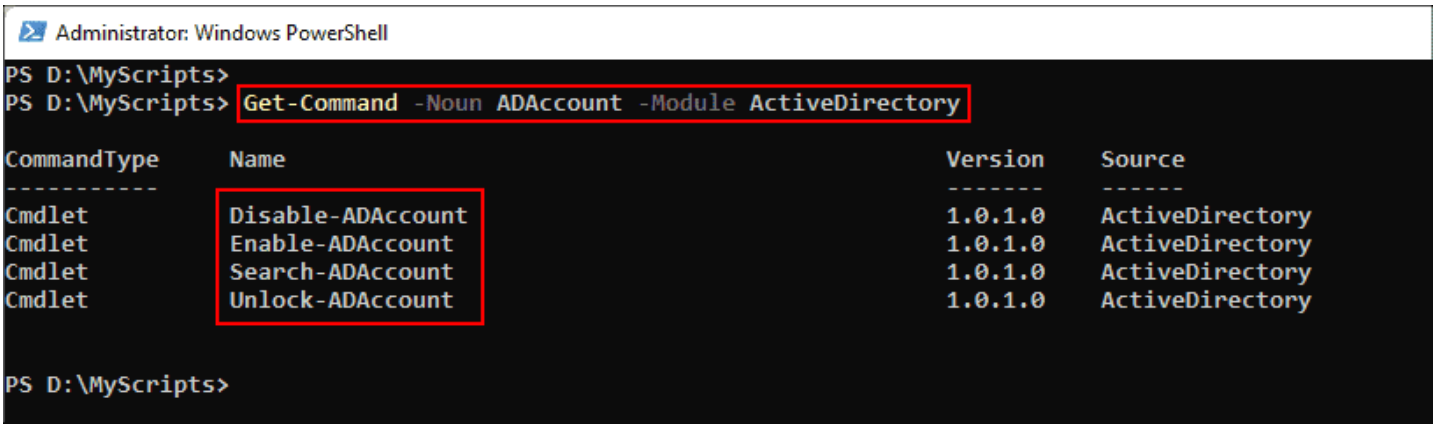
```
Get-WindowsCapability -Name Rsat.ActiveDirectory* -Online | Add-WindowsCapability –Online
```



Installing the Active Directory PowerShell module

If you just installed the module, it is a good idea to run the *Update-Help -Module ActiveDirectory -Force* command to update the help files. Once you have the AD PowerShell module, you are ready to manage AD accounts remotely. To identify suitable cmdlets to work with AD accounts, use this command:

```
Get-Command -Noun ADAccount -Module ActiveDirectory
```

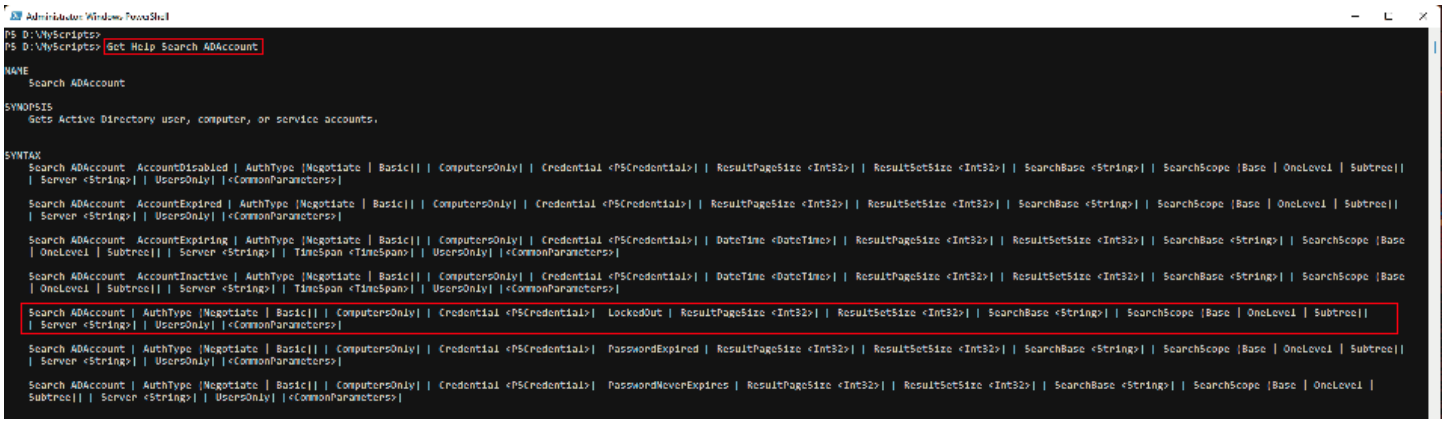


Finding the cmdlets related to ADAccount

As you can see in the screenshot, there are four cmdlets. The cmdlet verb gives a fair idea of what each command does. For this post, we will use only the *Search-ADAccount* and *Unlock-ADAccount* cmdlets.

Search-ADAccount cmdlet

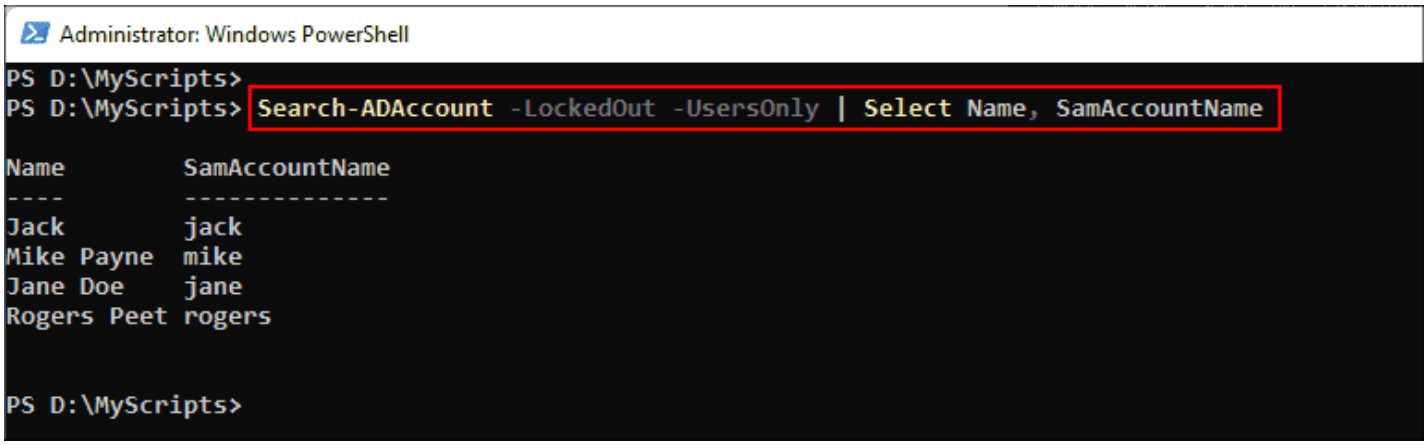
The *Search-ADAccount* cmdlet allows you to search for AD accounts based on various criteria. If you take a look at the command help, you will notice that it can search accounts that are disabled, expired, expiring, inactive, locked out, and that have a password that expired or that never expires. For now, we are only interested in locked-out accounts. By default, the credentials of the currently logged-on user are used on the remote domain controller, but you can use the *-Credential* parameter to supply alternate credentials.



Viewing the help of the Search ADAccount cmdlet

Now, we can use the *Search-ADAccount* cmdlet, as shown below, to search for locked-out AD accounts:

```
Search-ADAccount -LockedOut -UsersOnly | Select Name, SamAccountName
```



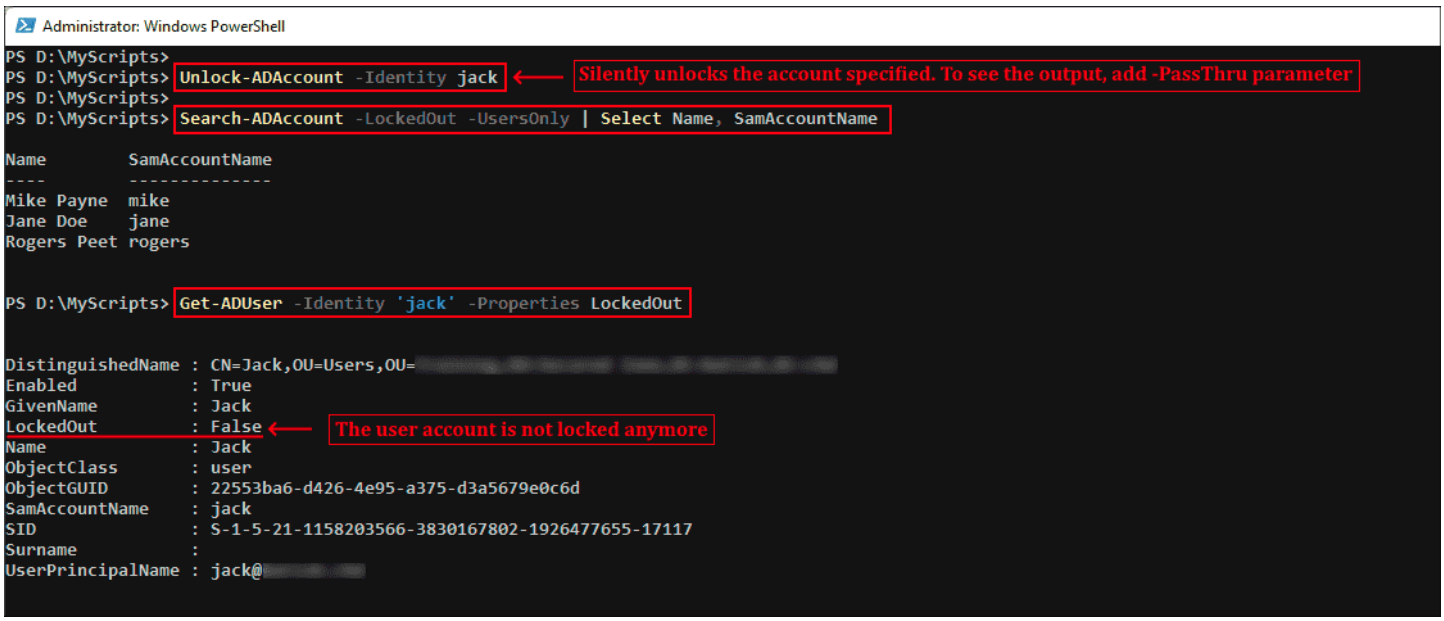
Searching locked out AD user accounts with PowerShell

This command lists all AD users that are currently locked out.

Unlock-ADAccount cmdlet

The *Unlock-ADAccount* cmdlet can be used to unlock AD accounts. If you take a look at the help section, you will notice that it accepts the *-Identity* parameter, which allows you to specify the SAM account name, the security identifier (SID), the globally unique identifier (GUID), or the distinguished name. To unlock a user account named *Jack*, simply run the command, as shown below:

```
Unlock-ADAccount -Identity jack
```



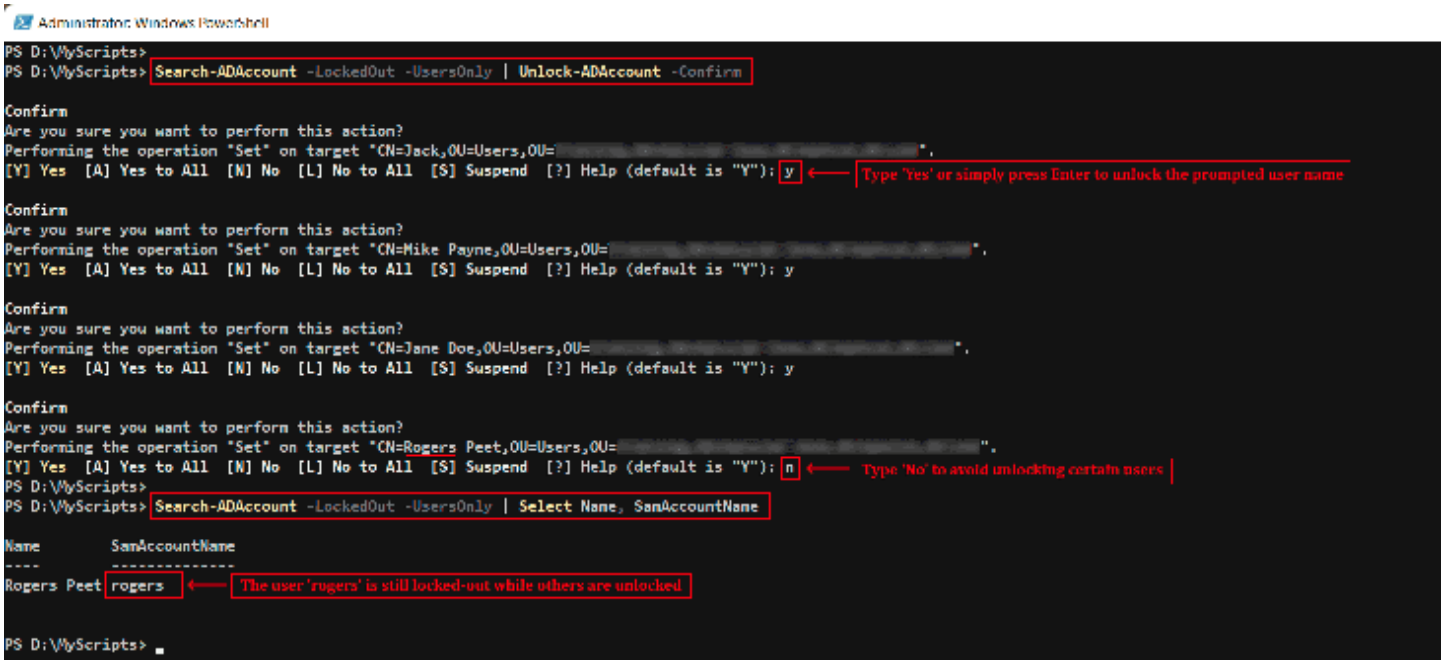
Unlock a particular AD user account with PowerShell

The command above does not produce any output, but the user account is silently unlocked. You could specify the *-PassThru* parameter, but that won't exactly show the *LockedOut* property for that user. Use the *Get-ADUser -Identity 'jack' -Properties LockedOut* command to see the updated state of the LockedOut property.

Unlock multiple AD accounts

If you want to quickly unlock multiple AD accounts that are currently locked out, use the following command:

```
Search-ADAccount -LockedOut -UsersOnly | Unlock-ADAccount –Confirm
```



Unlock all the locked out AD users at once with PowerShell

Note that the *-Confirm* switch plays an important role here. It requires your confirmation for each account before unlocking it. This approach allows you to skip certain accounts that you don't want to unlock. If you omit the *-Confirm* parameter, PowerShell will unlock all the locked-out user accounts without any confirmation, which could be a really bad idea. The AD accounts are locked out for a reason; when incorrect password attempts exceed the maximum number allowed in the account lockout policy, the user account is locked out, and access to AD services is restricted. If you unlock all users without even thinking, it defeats the purpose of the lockout policy.