SysAdmin
MAGAZINE

Power Move: Master More PowerShell Commands

SysAdmin
Magazine

№ 57
March '20

SysAdmin Magazine is a free source of knowledge for IT Pros who are eager to keep a tight grip on network security and do the job faster.

The Sysadmin Magazine team
sysadmin.magazine@netwrix.com

Contents

New Sysadmin Magazine Page
on Facebook

Now on Facebook

SysAdmin
Magazine

Get the best articles from the magazine, along with the freshest IT news and top tips from the IT community.

Follow the page to streamline your workload and stay on top of what's going on in IT.

Follow Us

Top 10 Group Policy PowerShell Commands

Russell Smith
Windows Security Expert, IT Consultant, Writer, and MCSE

In addition to the Group Policy Management Console (GPMC), Microsoft provides a set of Windows PowerShell cmdlets you can use to manage Group Policy. To use the Group Policy PowerShell cmdlets, you must have GPMC installed on the device where you will run the cmdlets. To check if the Group Policy PowerShell module is installed on a device, run the command below, which will display all the available Group Policy cmdlets available if the module is installed.

```
Get-Command -Module GroupPolicy
```

Creating a new Group Policy Object

Let's start by creating a new Group Policy object (GPO). The command below creates a new GPO called 'Netwrix PCs' and adds a comment to describe the its purpose:

```
New-GPO -Name "Netwrix PCs" -Comment "Clie-
nt settings for Netwrix PCs"
```

The command creates an empty GPO with no settings. If you have starter GPOs configured in your Active Directory domain, you can create a new GPO based on their settings. The following command creates a new GPO called 'Netwrix PCs' based on the 'Windows 10 MS Security Settings' GPO:

```
New-GPO -Name "Netwrix PCs" -StarterGPOName
"Windows 10 MS Security Settings"
```

You can optionally link the GPO to a domain, domain controller's organizational unit (OU) or site using piping. The command below creates a new GPO and links it to the Clients OU in the ad.contoso.com domain:

```
New-GPO -Name "Netwrix PCs" | New-GPLink
-Target "ou=clients,dc=ad,dc=contoso,dc=com"
```

To unlink a GPO, use the Remove-GPLink cmdlet:

```
Remove-GPLink -Name "Netwrix PCs" -Target
"ou=clients,dc=ad,dc=contoso,dc=com"
```

```
[dc1.mshome.net]: PS C:\Users\Administrator\Documents> New-GPO -Name "Netwrix PCs" | New-GPLink -Target "ou=clients,dc=ad,dc=contoso,dc=com"

GpoId       : 78e271c3-78b3-4234-94a1-b413a9b477c0
DisplayName : Netwrix PCs
Enabled     : True
Enforced    : False
Target      : OU=Clients,DC=ad,DC=contoso,DC=com
Order       : 1
```

```
[dc1.mshome.net]: PS C:\Users\Administrator\Documents> Remove-GPLin
Link -Name "Netwrix PCs" -Target "ou=clients,dc=ad,dc=contoso,dc=com"
DisplayName      : Netwrix PCs
DomainName       : ad.contoso.com
Owner            : AD\Domain Admins
Id               : 78e271c3-78b3-4234-94a1-b413a9b477c0
GpoStatus        : AllSettingsEnabled
Description      :
CreationTime     : 14/03/2019 12:52:20
ModificationTime : 14/03/2019 12:52:20
UserVersion      : AD Version: 0, SysVol Version: 0
ComputerVersion  : AD Version: 0, SysVol Version: 0
WmiFilter        :


[dc1.mshome.net]: PS C:\Users\Administrator\Documents>
```

Figure 1. How to link and unlink a GPO

Getting information about a GPO

Once a GPO is created, you can use Get-GPO to return information like GPO status, creation time and last modification time:

```
Get-GPO -Name "Netwrix PCs"
```

If you want more information, pipe the object created by Get-GPO to Get-GPOReport. The script below creates an HTML report that gives information about the GPO similar to what you might see in the Group Policy Management Console:

```
Get-GPO -Name "Netwrix PCs" | Get-GPOReport
-ReportType HTML -Path c:\temp\report.html
```

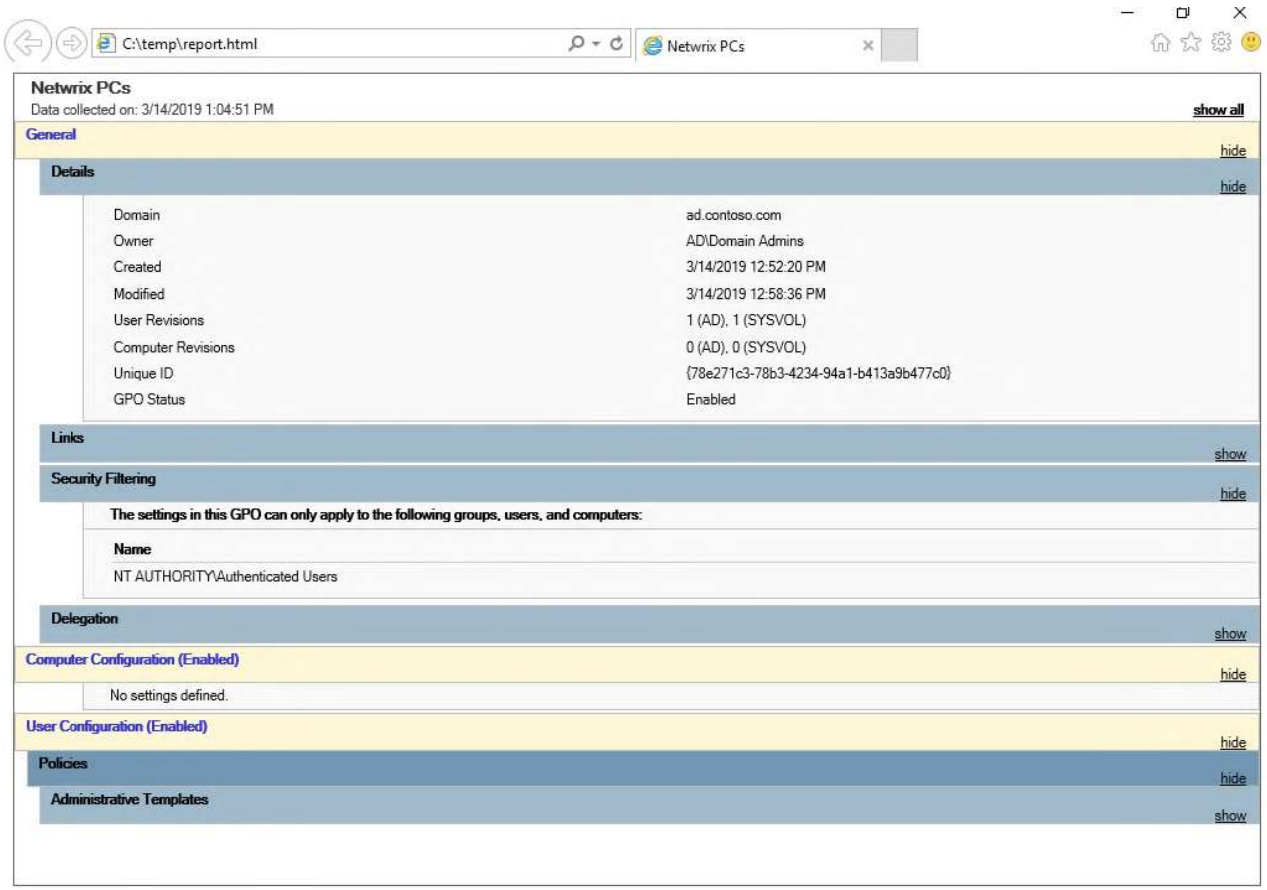


Figure 2. HTML report with detailed data about a specific GPO

Configuring Group Policy settings

If you know the location for a registry-based Group Policy setting, you can use the Set-GPRegistryValue cmdlet to configure it. Registry-based Group Policy settings are those that appear under Administrative Templates in GPMC. Set-GPRegistryValue can also be used to set registry values that are not covered by Group Policy settings. For example, if you want to configure registry settings for third-party applications that don't have an ADMX file for Group Policy, Set-GPRegistryValue is a quick way to configure the settings you need. The following command sets a screensaver timeout of 300 seconds for the logged-in user:

```
Set-GPRegistryValue -Name "Netwrix PCs" -Key
"HKCU\Software\Policies\Microsoft\Windows\Con-
trol Panel\Desktop" -ValueName ScreenSaveTime-
Out -Type DWord -Value 300
```

You can specify either computer configuration or user configuration settings using Set- GPRegistryValue The registry path in the -Key parameter below starts with "HKCU" (which stands for "HKEY_CURRENT_USER"). If you want to configure a computer setting instead, replace "HKCU" with "HKLM" (which expands to HKEY_LOCAL_MACHINE).

To get detailed information about a registry key configured in a GPO, use Get-GPRegistryValue:

```
Get-GPRegistryValue -Name "Netwrix PCs" -Key "HKCU\Software\Policies\Microsoft\Windows\Control Panel\Desktop"
```

```
[dc1.mshome.net]: PS C:\Users\Administrator\Documents> Get-GPRegistryValue -Name "Netwrix PCs" -Key "HKCU\Software\Policies\Microsoft\Windows\Control Panel\Deskto
p"


KeyPath     : Software\Policies\Microsoft\Windows\Control Panel\Desktop
FullKeyPath : HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\Control Panel\Desktop
Hive        : CurrentUser
PolicyState : Set
Value       : 200
Type        : DWord
ValueName   : ScreenSaveTimeOut
HasValue    : True


[dc1.mshome.net]: PS C:\Users\Administrator\Documents>
```

Figure 3. How to get detailed information about a registry key configured in a GPO

To remove a registry setting from a GPO, use Remove-GPRegistryValue:

```
Remove-GPRegistryValue -Name "Netwrix PCs" -Key "HKCU\Software\Policies\Microsoft\Windows\Control Panel\
Desktop" -ValueName ScreenSaveTimeOut
```

The three cmdlets above have Group Policy Preference equivalents if you decide to use Preferences instead of Policies to set registry keys: Set-GPPrefRegistryValue, Get-GPPrefRegistryValue, and Remove-GPPrefRegistryValue.

Applying Group Policy settings

Provided that your GPO is linked to a domain, OU or site, it will apply to user and computer objects below where it is linked. But if you want to force a Group Policy update on a remote server or other device, you can use Invoke-GPUpdate. Running Invoke-GPUpdate without any parameters will force an update of user and computer configuration settings on the local computer. The command below forces a Group Policy update on server1 for user configuration settings only:

```
Invoke-GPUpdate -Computer "ad\server1" -Target
"User"
```

Reviewing which GPOs are applied to a user or computer

To get information about which GPOs are applied to a user or computer, you can generate a Resultant Set of Policy (RSoP) report using the Get-GPResultantSetOfPolicy cmdlet. The command below generates a report for the computer called "dc1" and writes the results to the c:\temp directory:

```
Get-GPResultantSetOfPolicy -Computer dc1 -ReportType HTML -Path c:\temp\dc1rsop.html
```

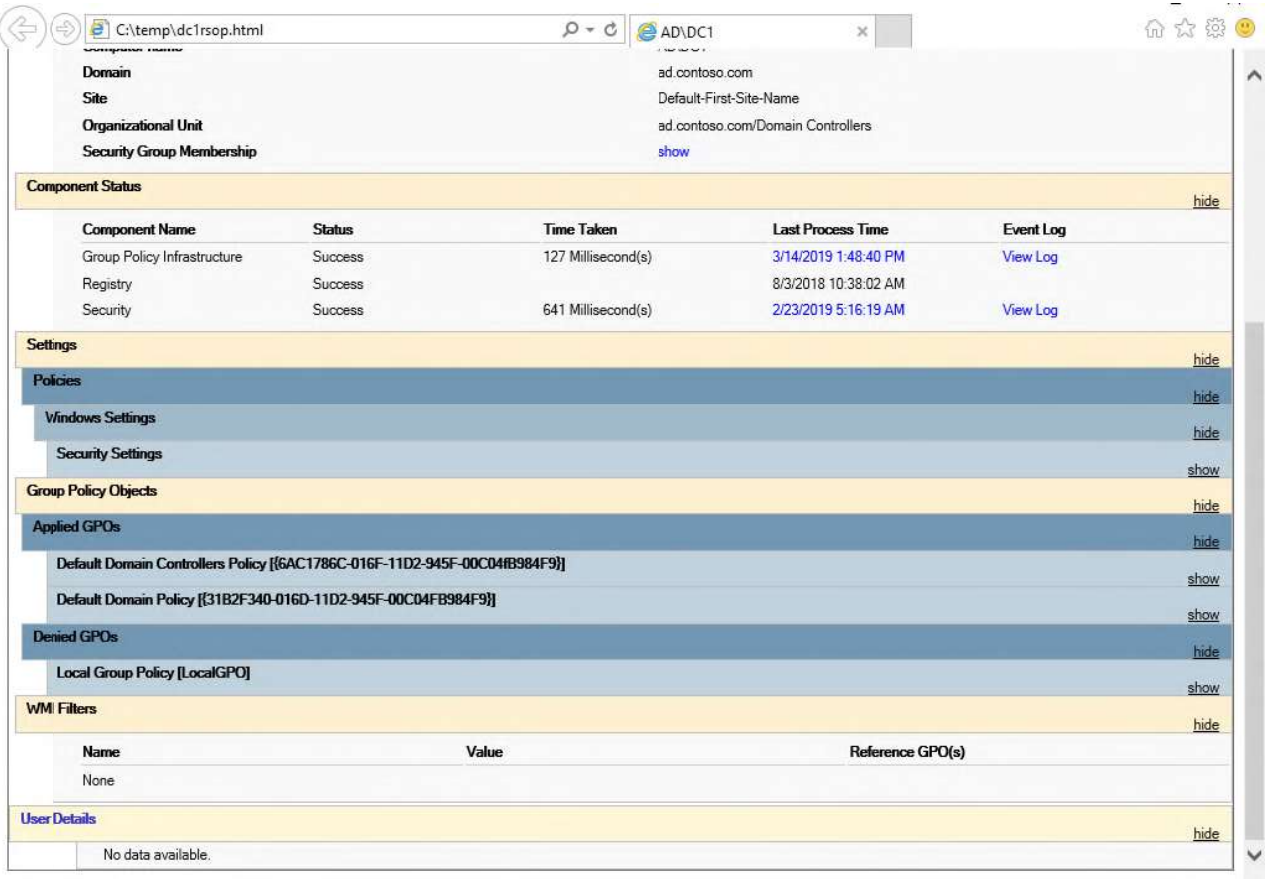


Figure 4. How to get information about which GPOs are applied to a user or computer

Most useful SharePoint PowerShell Commands

Jeff Melnick
IT Security Expert, Blogger

SharePoint Server is a web-based collaborative platform that integrates with Microsoft Office. To configure SharePoint site settings, system administrators often use the SharePoint Management Shell that is installed with the SharePoint product. Running the SharePoint Management Shell calls the Windows PowerShell runtime environment and executes a script file named sharepoint.ps1, which loads the Windows PowerShell snap-in for SharePoint and runs a few commands. These commands are not very important; they include choosing C:\Users\Username as the home location for command execution and running the latest version of PowerShell console.

A better option is to use the PowerShell ISE. Not only does it include many cmdlets created especially for managing SharePoint, it also offers color-highlighted code, a debug engine and a cmdlet search engine.

To load the SharePoint snap-in, we need to run the following command:

```
Add-PSSnapin Microsoft.SharePoint.PowerShell
```

After that, it is prudent to update PowerShell help in order get the latest information about the Microsoft SharePoint PowerShell cmdlets:

```
Update-Help
```

So, what cmdlets are available in the SharePoint snap-in? Here is the command that will list all of them for you:

```
Get-Command -Module "Microsoft.SharePoint.PowerShell"
```

Getting SharePoint site information

Get-SPSite cmdlet is the main cmdlet for getting information about your SharePoint site collections. It lists the URLs of your SharePoint sites and their compatibility levels (SharePoint versions).

```
PS C:\Windows\system32> Get-SPSite
                                  CompatibilityLevel
Url                               ------------------
---                               15
http://sharepoint/my              15
http://sharepoint/sites/ent
```

Using the **Select-Object** parameter with this cmdlet, we can get specific properties about a site, such as the site owner, storage usage, maximum quota level and last content modified date:MAC address filtering.

```
Get-SPSite "http://sharepoint/sites/ent" | Select-Object url, owner, @{Expression={$_.Usage.Storage}}, @{Expression={$_.Audit.AuditFlags}}, readonly, LastContentModifiedDate, @{Express={$_.QuotaStorageMaximumLevel}}
```

```
Url                          : http://sharepoint/sites/ent
Owner                        : i:0#.w|enterprise\i.scur
$_.Usage.Storage             : 1916840
$_.Audit.AuditFlags          : SecurityChange
ReadOnly                     : False
LastContentModifiedDate      : 1/5/2018 2:55:38 PM
$_.QuotaStorageMaximumLevel :
```

In addition, we can export information about all sites in our SharePoint farm to a csv file:

```
Get-SPWebApplication http://sharepoint/ |
Get-SPSite -Limit All | Get-SPWeb -Limit
All | Select Title, URL, ID, ParentWebID
| Export-CSV C:\root\sharepointinventory.
csv -NoTypeInformation
```

We can also use the **Get-SPSite** cmdlet to create a SharePoint PowerShell script that lists all the groups and their members for a particular SharePoint site:

```
$site = Get-SPSite http://sharepoint/si-
tes/ent/
$groups = $site.RootWeb.sitegroups
foreach ($grp in $groups) {"Group: " +
$grp.name; foreach ($user in $grp.users)
{"  User: " + $user.name} }
$site.Dispose()
```

```
Group: Enterprise Members
  User: Tom Simpson
Group: Enterprise Owners
  User: Ian Scur
Group: Enterprise Visitors
Group: Site Designers
  User: Elena Anderson
  User: System Account
```

If you need a complete permissions report for a SharePoint site, run the following code, specifying the SharePoint site URL ($SPSiteURL) and the path to export the data to a csv file ($ExportFile) :

```
[void][System.Reflection.Assembly]::Load-
WithPartialName("Microsoft.SharePoint")
$SPSiteUrl = "http://sharepoint/sites/
ent"
$SPSite = New-Object Microsoft.Share-
Point.SPSite($SPSiteUrl);
$ExportFile = "C:\root\Permissions.csv"

"Web Title,Web URL,List Title,User or
Group,Role,Inherited" | out-file $Export-
File

foreach ($WebPath in $SPSite.AllWebs)
{
   if ($WebPath.HasUniqueRoleAssignments)
        {
          $SPRoles = $WebPath.RoleAssign-
ments;
          foreach ($SPRole in $SPRoles)
          {
            foreach ($SPRoleDefinition in
$SPRole.RoleDefinitionBindings)
            {
                $WebPath.Title + "," + $We-
bPath.Url + "," + "N/A" + "," + $SPRole.
Member.Name + "," + $SPRoleDefinition.Name +
"," + $WebPath.HasUniqueRoleAssignments |
out-file $ExportFile -append
            }
          }
        }

        foreach ($List in $WebPath.Lists)
        {
           if ($List.HasUniqueRoleAssign-
ments)
            {
              $SPRoles = $List.RoleAssign-
ments;
              foreach ($SPRole in $SPRoles)
              {
                foreach ($SPRoleDefinition in
$SPRole.RoleDefinitionBindings)
                {
                     $WebPath.Title + "," +
```

```
$WebPath.Url + "," + $List.Title + "," +
$SPRole.Member.Name + "," + $SPRoleDefiniti-
on.Name | out-file $ExportFile -append
                }
              }
            }
          }
}
$SPSite.Dispose();
```

To find a certain file on a SharePoint site, we need to use the **Get-SPWeb** cmdlet. Here is a script that searches for a file whose name contains the word "readme" in the "http://sharepoint/sites/ent" site:

```
Get-SPWeb http://sharepoint/sites/ent |
   Select -ExpandProperty Lists |
   Where { $_.GetType().Name -eq "SPDocu-
mentLibrary" -and
           -not $_.Hidden } |
   Select -ExpandProperty Items |
   Where { $_.Name -like "*readme*" } |
   Select Name, {$_.File.Length}, url
```

Now let's make a report that will output all files created by a certain user. This script can be helpful, for example, when an employee leaves the company and you need to transfer their data to other people.

```
Get-SPWeb http://sharepoint/sites/ent |
   Select -ExpandProperty Lists |
   Where { $_.GetType().Name -eq "SPDocu-
mentLibrary" -and
           -not $_.Hidden } |
   Select -ExpandProperty Items |
   Where { $_["Created By"] -like
"*system*" } |
   Select Name, url, {$_["Created By"]}
```

Our last script using the **Get-SPWeb** cmdlet reports on all files with a specified extension:

```
Get-SPWeb http://sharepoint/sites/ent |
Select -ExpandProperty Lists |
   Where { $_.GetType().Name -eq "SPDocu-
mentLibrary" -and
           -not $_.Hidden } |
   Select -ExpandProperty Items |
   Where { $_.Name -Like "*.rtf" } |
   Select Name,
         @{Name="URL";
          Expression={$_.ParentList.Par-
entWeb.Url + "/" + $_.Url}}
```

Using PowerShell to manage SharePoint sites and objects

New SharePoint sites are typically created using a template. To get a list of all site templates, run the **Get-SPWebTemplate** cmdlet with no parameters.

We use the Get-SPWebTemplate cmdlet with the New-SPSite cmdlet to create a new SharePoint site based on a template. Here is an example of a script for creating a site using the "Team Site" template (STS#0):

```
$template = Get-SPWebTemplate "STS#0"
New-SPSite -Url "http://sharepoint/sites/
netwrixteamsite" -OwnerAlias "enterprise\t.
simpson" -Template $template
```

To delete a site, we use the **Remove-SPSite** cmdlet:

```
Remove-SPSite -Identity "http://sharepoint/
sites/netwrixteamsite" -GradualDelete
```

Sometimes, you might need to change the site collection administrator. Execute the following script to add admin rights to the specified user:

```
Set-SPSite -Identity "http://sharepoint/si-
tes/netwrixteamsite" -SecondaryOwnerAlias "
i:0#.w|enterprise\i.scur"
```

Now, let's see how to manage permissions to our site collections. First, let's add certain access rights to a particular Active Directory user account. In this case, the user "enterprise\t.simpson" will be given "Contributor" rights to the site "http://sharepoint/sites/ent". Note that before a regular user account name like "enterprise\t.simpson", you need to use the prefix "i:0#.w|". Otherwise, the execution will fail and the script will generate an error.

```
Set-SPUser -Identity "i:0#.w|enterprise\t.
simpson" -Web http://sharepoint/sites/ent
-AddPermissionLevel "Contributor"
```

To add permissions to a certain AD security group, use the same script but type the name of the group instead of the name of the user account.

To add a user to a group, use this command:

```
Set-SPUser -Identity "i:0#.w|enterprise\j.
carter" -Web http://sharepoint/sites/ent
-Group "Enterprise Owners"
```

As you can see, managing and reporting on SharePoint sites using SharePoint PowerShell scripts is not as hard as it may seem at first. In fact, in some cases, it is much faster to run a script rather than generate a report from the admin audit log.

Most Useful Office 365 PowerShell Commands

Adam Stetson
Systems Engineer, Security Expert

Using Windows PowerShell to manage Office 365 may seem odd at first. After all, cloud solutions promise simplicity and ease of use — adjectives rarely used in connection with Windows PowerShell. But bear with me. In this article, I'll show you the ten most useful Office 365 PowerShell cmdlets for system administrators. Perhaps after reading these instructions, you'll agree that PowerShell can be a valuable tool, even for cloud-based systems.

1. Connecting to an Office 365 instance with PowerShell

First, we need to install the Office 365 module for Windows PowerShell and connect to the Office 365 instance. Take the following steps:

- Download and install the Microsoft Online Services Sign-In Assistant for IT Professionals RTW.
- Import the Online Services PowerShell module for Microsoft Azure Active Directory and Office 365:

```
Install-Module -Name AzureAD
Install-Module -Name MSOnline
```

- Enter your Office 365 admin credentials:

```
$Cred = Get-Credential
```

- Create a remote PowerShell session:

```
$O365 = New-PSSession -ConfigurationName
Microsoft.Exchange -ConnectionUri https://
outlook.office365.com/powershell-liveid/
-Credential $Cred -Authentication Basic
-AllowRedirection
```

- Connect to all Office 365 services:

```
Connect-MsolService -Credential $O365
```

Once we have imported the modules for Windows PowerShell, we are ready to manage our Office 365 instance.

2. Connecting to Exchange Online and SharePoint Online with PowerShell

We can also connect to Microsoft Exchange Online and Microsoft SharePoint Online separately. Connecting to Exchange Online with PowerShell is basically the same as connecting to Office 365:

```
$Cred = Get-Credential
$Session = New-PSSession -ConfigurationName
Microsoft.Exchange -ConnectionUri https://
outlook.office365.com/powershell-liveid/
-Credential $Cred -Authentication Basic -
AllowRedirection
```

Connecting to SharePoint Online is a little bit different In order to manage your SharePoint Online tenant, you first need to download and install the SharePoint Online Management Shell feature. Then run the following PowerShell script:

```
$admin="Admin@enterprise.onmicrosoft.com"
$orgname="enterprise"
$userCred = Get-Credential -UserName $admin
-Message "Type the password."
Connect-SPOService -Url https://$orgname-
admin.sharepoint.com -Credential $userCred
```

3. Getting a list of available Office 365 PowerShell cmdlets

To get a list of all available Office 365 PowerShell commands, we need to run the Get-Command cmdlet:

```
Get-Command -module MSOnline
```

```
PS C:\Windows\system32> Get-Command -module MSOnline

CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Cmdlet          Add-MsolAdministrativeUnitMember                   1.1.183.17 MSOnline
Cmdlet          Add-MsolForeignGroupToRole                         1.1.183.17 MSOnline
Cmdlet          Add-MsolGroupMember                                1.1.183.17 MSOnline
Cmdlet          Add-MsolRoleMember                                 1.1.183.17 MSOnline
Cmdlet          Add-MsolScopedRoleMember                           1.1.183.17 MSOnline
Cmdlet          Confirm-MsolDomain                                 1.1.183.17 MSOnline
Cmdlet          Confirm-MsolEmailVerifiedDomain                    1.1.183.17 MSOnline
Cmdlet          Connect-MsolService                                1.1.183.17 MSOnline
Cmdlet          Convert-MsolDomainToFederated                      1.1.183.17 MSOnline
Cmdlet          Convert-MsolDomainToStandard                       1.1.183.17 MSOnline
Cmdlet          Convert-MsolFederatedUser                          1.1.183.17 MSOnline
Cmdlet          Disable-MsolDevice                                 1.1.183.17 MSOnline
Cmdlet          Enable-MsolDevice                                  1.1.183.17 MSOnline
Cmdlet          Get-MsolAccountSku                                 1.1.183.17 MSOnline
Cmdlet          Get-MsolAdministrativeUnit                         1.1.183.17 MSOnline
Cmdlet          Get-MsolAdministrativeUnitMember                   1.1.183.17 MSOnline
Cmdlet          Get-MsolCompanyAllowedDataLocation                 1.1.183.17 MSOnline
Cmdlet          Get-MsolCompanyInformation                         1.1.183.17 MSOnline
Cmdlet          Get-MsolContact                                    1.1.183.17 MSOnline
```

We can also get the list of cmdlets for Azure Active Directory:

```
Get-Command -module AzureAD
```

4. Getting a list of all Office 365 users with PowerShell

If you need to provide a list of Office 365 users and licenses, use the Get-MsolUser cmdlet. It'll retrieve all users with a valid license in the Office 365 tenant, along with the DisplayName, City, Department and ObjectID parameters.

```
Get-MsolUser | Select DisplayName, City,
Department, ObjectID
```

To see the number of account licenses, you need to run the following cmdlet:

```
Get-MsolAccountSku
```

To list the available services, run the following script:

```
Get-MsolAccountSku | select -ExpandProperty
ServiceStatus
```

```
ServicePlan                    ProvisioningStatus
-----------                    ------------------
STREAM_O365_SMB                Success
OFFICEMOBILE_SUBSCRIPTION      Success
BPOS_S_TODO_1                  Success
FORMS_PLAN_E1                  Success
FLOW_O365_P1                   Success
POWERAPPS_O365_P1              Success
TEAMS1                         Success
PROJECTWORKMANAGEMENT          Success
```

5. Creating a new user in Office 365 with PowerShell

To create a new user, we use the New-MsolUser command:

```
New-MsolUser -UserPrincipalName JSmith@
enterprise.onmicrosoft.com -DisplayName "John
Smith"  -FirstName "John" -LastName "Smith"
```

The system will output the user's password and license status data.

6. Removing a user from all sites with PowerShell

To remove a user from all sites at once, we use the following command:

```
Get-SPOSite | ForEach {Remove-SPOUser -Site
$_.Url -LoginName " JSmith@enterprise.
onmicrosoft.com"}
```

7. Changing a password in Office 365 with PowerShell

If you need to change the password for an account, use the Set-MsolUserPassword cmdlet. You can either specify a new password as in the example below, or omit the -NewPassword parameter to have the system automatically generate a random password.

```
Set-MsolUserPassword -UserPrincipalName
JSmith@netwrixqcspa.onmicrosoft.com
-NewPassword P@SSw0rd!
```

8. Managing group membership in Office 365 with PowerShell

We can also manage Office 365 groups using PowerShell cmdlets. To retrieve a list of all groups in Office 365, simply use the command **Get-MsolGroup**. To add users to a group, use the **Add-MsolGroupMember** command:

```
Add-MsolGroupMember -GroupObjectId
5b61d9e1-a13f-4a2d-b5ba-773cebc08eec
-GroupMemberObjectId a56cae92-a8b9-4fd0-
acfc-6773a5c1c767 -GroupMembertype user
```

GroupObjectId is the hexadecimal ID of the group, which you can get from the Get-MsolGroup command. GroupMemberObejctId is the user object ID, which you can find by running this command:

```
Get-MsolUser | Select ObjectID.
```

9. Creating a SharePoint site collection with PowerShell

We can also create a SharePoint site collection using PowerShell:

```
New-SPOSite -Url "https://enterprise.
sharepoint.com/sites/NewSite" -Owner
"JSmith@enterprise.onmicrosoft.com"
-StorageQuota "100" -Title "New Site"
```

10. Creating reports in Office 365 with PowerShell

PowerShell is a great tool for making different reports. Here are some useful Office 365 reports done via PowerShell:

- Details about all mailboxes:

```
Get-mailbox | get-MailboxStatistics
```

- A list of all mailboxes that haven't been logged into during the last 30 days:

```
Get-Mailbox -RecipientType 'UserMailbox'
| Get-MailboxStatistics | Sort-Object
LastLogonTime | Where {$_.LastLogonTime -lt
([System.DateTime]::Now).AddDays(-30) } |
Format-Table DisplayName, LastLogonTime
```

- A report on the highest volume senders and recipients:

```
Get-MailTrafficTopReport
```

PowerShell Commands for Effective Password Management

Russell Smith
Windows Security Expert, IT Consultant, Writer, and MCSE

Automation is the key to streamlining Active Directory management tasks. In this article, I'll show you how to create, change and test user passwords with PowerShell scripts.

Installing the AD PowerShell module

Before you can use PowerShell to manage Active Directory, you need to install the Active Directory PowerShell module. If you are using Windows 10 to manage AD, first install the Remote Server Administration Tools (RSAT).

Windows 10 Version 1809

If you are using Windows 10 version 1809, RSAT is included as a Feature On Demand, so you don't need to download the RSAT package. To enable RSAT in Windows 10 version 1809, run the following command in an elevated PowerShell console:

```
Add-WindowsCapability -Online -Name Rsat.
ActiveDirectory.DS-LDS.Tools~~~~0.0.1.0
```

Earlier Versions of Windows 10

If you are using an earlier version of Windows 10, download the appropriate RSAT package from Microsoft's website:

- If you are managing Windows Server version 1803 or 1709, download and install the WS_1803 package.
- If you are managing Windows Server 2016 or earlier versions of Windows Server, download and install the

Once RSAT is installed, start the PowerShell console as a local administrator and enable the AD PowerShell module using this PowerShell command:

```
Enable-WindowsOptionalFeature -Online -Fea-
tureName RSATClient-Roles-AD-Powershell
```

Create credential with password using PowerShell

To create a new user account, use the New-ADUser cmdlet. In the example below, I have hardcoded the ad.contoso.com domain in the $UPN variable. You should change this to match the UPN suffix you want to assign to users.

Provide the user's first name and last name. The UPN and SamAccountName will then be created by adding a period between the first and last name. Use the following PowerShell script:

```
$GivenName = (Read-Host -Prompt "First
Name")
$Surname = (Read-Host -Prompt "Last Name")
$User = $GivenName+"."+$Surname
$UPN = $User+"@ad.contoso.com"
$Password = (Read-Host -Prompt "Password"
-AsSecureString)
New-ADUser -Name $User -SamAccountName
$User -UserPrincipalName $UPN -AccountPass-
```

```
word $Password -GivenName $GivenName -Surname
$Surname -Enabled $True
```

Create new AD user password using PowerShell

The following code will prompt you to specify a username and password. You must enter a username that already exists in AD and a password that meets the domain's password complexity requirements.

```
$User = (Read-Host -Prompt "Username")
$NewPassword = (Read-Host -Prompt "New
Password" -AsSecureString)
Set-ADAccountPassword -Identity $User -New-
Password $NewPassword -Reset
```

Change password using PowerShell

- Change a local user's password

To change a local user's password, you need to use the Get-LocalUser and Set-LocalUser cmdlets:

```
$Password = (Read-Host -Prompt "New Pass-
word" -AsSecureString)
$User = (Read-Host -Prompt "Username")
$UserAccount = Get-LocalUser -Name $User
$UserAccount | Set-LocalUser -Password
$Password
```

- Change an AD user's password

To create a new AD user password using PowerShell, use the following script. You will be prompted to specify the username of an existing AD account and then a new password, which must meet the domain's password complexity requirements.

```
$User = (Read-Host -Prompt "Username")
$NewPassword = (Read-Host -Prompt "New Pass-
word" -AsSecureString)
Set-ADAccountPassword -Identity $User -New-
Password $NewPassword -Reset
```

- Force a user to change their password at next logon

The Set-LocalUser cmdlet doesn't support setting a local user account to force a password change at next logon. However, you can achieve the same goal by forcing the password to expire:

```
$User = (Read-Host -Prompt "Username")
$Usrstring = "WinNT://localhost/"+$User
$usr=[ADSI] $Usrstring
$usr.passwordExpired = 1
$usr.setinfo()
```

But you can force users to change their AD account passwords using Set-ADAccountPassword:

```
$User = (Read-Host -Prompt "Username")
Set-Aduser -Identity $User -ChangePassword-
AtLogon $true
```

- Change an administrator password

To change the AD administrator password, type administrator when you are prompted for a username using the code below:

```
$User = (Read-Host -Prompt "Username")
$NewPassword = (Read-Host -Prompt "New
Password" -AsSecureString)
Set-ADAccountPassword -Identity $User -New-
Password $NewPassword -Reset
```

To change a local administrator password, type administrator when prompted for a username:

```
$Password = (Read-Host -Prompt "New Pass-
word" -AsSecureString)
$User = (Read-Host -Prompt "Username")
$UserAccount = Get-LocalUser -Name $User
$UserAccount | Set-LocalUser -Password
$Password
```

- Change the "password never expires" attribute

To set the "password never expires" attribute on a local user account, use Set-LocalUser:

```
$User = (Read-Host -Prompt "Username")
Set-LocalUser -Name $User -PasswordNeverEx-
pires $true
```

To set the "password never expires" attribute on an Active Directory user account, use Set-ADUser:

```
$User = (Read-Host -Prompt "Username")
Set-ADUser -Identity $User -PasswordNever-
Expires $true
```

- Change the service account password

To change the logon properties of a service, use the Get-Credential and Set-Service cmdlets. The following code changes the AppReadiness service from using the Local System account to using the username and password that are entered when prompted. Note that the Set-Service -Credential parameter is supported only in PowerShell 6 and later.

```
$credential = Get-Credential
Set-Service -Name "AppReadiness" -Credenti-
al $credential
```

- Change a password's expiration date in Active Directory

If you need to extend the time a user can keep their current password, set the pwsLastSet attribute to the current date, giving them extra time until Active Directory forces them to change their password. Clearing the attribute and then setting it to -1 will set it to the current date and time.

```
$Username = (Read-Host -Prompt "Username")
$User = Get-ADUser $Username -Properties
pwdlastset
$User.pwdlastset = 0
Set-ADUser -Instance $User
$User.pwdlastset = -1
Set-ADUser -Instance $User
```

- Bulk password reset

The best way to get users to change their AD passwords is to force a password reset. You can do this in bulk by combining the Get-ADUser and Set-ADUser cmdlets. The command below uses a filter to get users in the "Accounts" organizational unit (OU) and pipes the results to the Set-ADUser cmdlet to force all users in the OU to change their password at next logon.

```
Get-ADUser -Filter * -SearchScope Subtree
-SearchBase "OU=Accounts,DC=ad,DC=conto-
so,DC=com" |
Set-ADUser -ChangePasswordAtLogon $true
```

Testing a user's credentials

If you want to test if a user's credentials are working, all you need to do is start a process using their username and password. The code below starts cmd.exe using the credentials entered when prompted.

```
Start-Process -FilePath cmd.exe /c -Credential (Get-Credential)
```

How-to for IT Pro

HOW TO GET LOCAL ADMINISTRATORS

1. Open the Powershell ISE → Create new script with the following code and run it, specifying the computer list and the path for export:

```
invoke-command {
    $members = net localgroup
administrators |
     where {$_ -AND $_ -notmatch
"command completed successfully"} |
     select -skip 4
    New-Object PSObject -Property @{
     Computername = $env:COMPUTERNAME
     Group = "Administrators"
     Members=$members
     }
    } -computer fs1,sp01,ncnad
-HideComputerName |
    Select * -ExcludeProperty RunspaceID
| Export-CSV c:\data\local_admins.csv
-NoTypeInformation
```

2. Open the file produced by the script in MS Excel. Sample report:

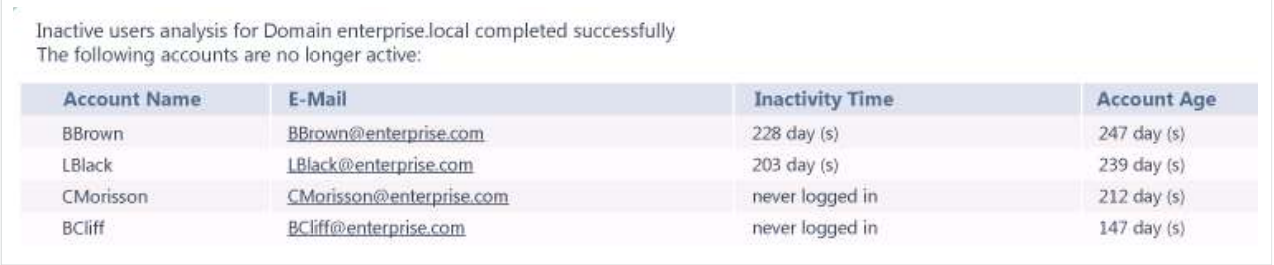
	A	B	C
1	Group	Computername	Members
2	Administrators	NCNAD	Administrator ENTERPRISE\Domain Admins
3	Administrators	FS1	Administrator ENTERPRISE\Domain Admins J.Carter
4	Administrators	SP01	Administrator ENTERPRISE\Domain Admins service_01

FREE TOOL OF THE MONTH

Inactive User Tracker

Download Free Tool

Freeware tool that tracks down inactive user accounts, so you can harden your Active Directory security and mitigate the risk of breaches.



Inactive users analysis for Domain enterprise.local completed successfully
The following accounts are no longer active:

Account Name	E-Mail	Inactivity Time	Account Age
BBrown	BBrown@enterprise.com	228 day (s)	247 day (s)
LBlack	LBlack@enterprise.com	203 day (s)	239 day (s)
CMorisson	CMorisson@enterprise.com	never logged in	212 day (s)
BCliff	BCliff@enterprise.com	never logged in	147 day (s)

[On-Demand Webinar]

Managing SharePoint Online and Exchange Online with PowerShell

Leam Cleary
Security expert, Microsoft MVP

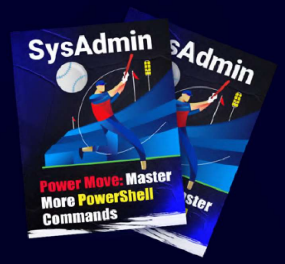
Jeff Melnick
IT Security Expert, Blogger

Managing SharePoint Online and Exchange Online can be a painful task — you have to constantly switch between multiple administration centers and it's hard to know where to go for a specific setting. Luckily, you can accomplish many tasks with PowerShell.

In this webinar, Liam Cleary and Jeff Melnick will walk you through how to use PowerShell to:

- Connect to Office 365
- Perform basic management tasks like user and mailbox administration
- Modify permissions and retrieve log data when auditing permissions

Watch Now



What did you think of this issue?

About Netwrix

Netwrix is a software company that enables information security and governance professionals to reclaim control over sensitive, regulated and business-critical data, regardless of where it resides.

Over 10,000 organizations worldwide rely on Netwrix solutions to secure sensitive data, realize the full business value of enterprise content, pass compliance audits with less effort and expense, and increase the productivity of IT teams and knowledge workers.

For more information visit www.netwrix.com

CORPORATE HEADQUARTER:

300 Spectrum Center Drive
Suite 200 Irvine, CA 92618

565 Metro Place S, Suite 400
Dublin, OH 43017

5 New Street Square
London EC4A 3TW

PHONES:

1-949-407-5125
Toll-free (USA): 888-638-9749

1-201-490-8840

+44 (0) 203 588 3023

OTHER LOCATIONS:

Spain:	+34 911 982608	Switzerland:	+41 43 508 3472	Hong Kong:	+852 5808 1306
Netherlands:	+31 858 887 804	France:	+33 9 75 18 11 19	Italy:	+39 02 947 53539
Sweden:	+46 8 525 03487	Germany:	+49 711 899 89 187		

SOCIAL: netwrix.com/social