Using PowerShell for Security Testing (3)

We can use PowerShell to access domain controllers and to list and create users within a Windows Domain. To list out all of the users with a domain we can execute to the following command.

```
PS C:\> Import-Module ActiveDirectory
PS C:\> Get-ADUser -Server WS2012-01 -Filter * -Properties samAccountName
```

We can create users using the following command. The Read-Host parameter will ask you to input new password. Note that the password should meet the length, complexity and history requirements of your domain.

```
PS C:\> New-ADUser -Server WS2012-01 -Name "Peter Pan" -GivenName "Peter" -
Surname "Pann" -SamAccountName "P.Pann" -UserPrincipalName
"peter.pann@merimetso.net" -Path "OU=users,DC=merimetso,DC=com" -
AccountPassword(Read-Host -AsSecureString "Input Password") -Enabled $true
```

We can also disable an account within a domain using power shell. In the following we will disable the account with the **`UserPrincipalName`** of **`peter.pann@merimetso.net`**.

```
PS C:\> Disable-ADAcount -Server WS2012-01 -UserPrincipalName
"peter.pann@merimetso.net"
```

Using PowerShell, we can create a group with a domain using the following command. In the following we will create a new group called **`"My New Group"`**.

```
PS C:\> New-ADGroup "My New Group" -Server WS2012-01 -Path
"OU=Users,DC=merimetso,dc=com"
```

Once we have created a group, we can add people to it. So, in the following we will add the user **`P.Pann`** to the group **`"My New Group"`**. Of course, the ability to add a user to a group creates the ability to add a user to the **`"Domain Admins"`** group, thus promoting a user to an administrator within a domain.

```
PS C:\> Add-AdGroupmember -Server WS2012-01 -Identify "My New Group" -Members
P.Pann
```

Once we have added a user to a group within a domain then we may wish to examine the group to check that it has worked and to examine other users in group. We can do this via the following PowerShell command.

```
PS C:\> Get-GroupMember -Server WS2012-01 -Identify "My New Group"
```

Using PowerShell, we can reset a user's password as follows. In the following we will reset the username's **`P.Pann`** password to **`Password@123`**. This is achieved via defining a parameter and then using that parameter in the **`Set-ADAccountPassword`** PowerShell function.

```
PS C:\> $Pass = ConvertTo-SecureString "Password@123" -AsPlainText -Force
PS C:\> Set-ADAccountPassword -Identity P.Pann -NewPassword $pass -Reset
```

The other way to re-set a user's password is via the application and utilization of the ADSI interface as follows. While the ADSI method can be used on any Windows or PowerShell version, it has more disadvantages. First, it does not support the use of different credentials. That means you would need to run the PowerShell session under an account with the proper privileges to reset passwords. Second, it only supports the user's distinguished name (**`CN=P.Pann,OU=Uesr,DC=Merimetso,DC=com`**). This limits you to targeting only users in a specific organizational unit (OU), so you would need to know in advance where the user account is located and change the code appropriately.

```
PS C:\> $userid = [ADSI]"LDAP://CN=P.Pann,OU=Uesr,DC=Merimetso,DC=com"
PS C:\> $userid.psbase.invoke("SetPassword",'Password@123')
PS C:\> $userid.psbase.CommitChanges()
```