25.1 Tworzenie użytkowników z odpowiednimi uprawnieniami

A. Wykrywanie poleceń administracyjnych użytkownika lokalnego.

Najpierw upewnij się, że w Twoim systemie działa PowerShell 5.1. Otwórz PowerShell i uruchom

(Get-Host).Version

```
Major  Minor  Build  Revision
-----  -----  -----  --------
5      1      22621  3810
```

Polecenia można znaleźć, uruchamiając

Get-Command -Module Microsoft.PowerShell.LocalAccounts

Jeśli brak to należy zainstalować i zaimportować moduł.

install-module -name LocalAccount

Import-Module -name LocalAccount

Jeśli wpiszesz jak poniżej to oznacza, że moduł już jest w systemie:

Get-Command -Module Microsoft.PowerShell.LocalAccounts

```
CommandType     Name                               Version    Source
-----------     ----                               -------    ------
Cmdlet          Add-LocalGroupMember               1.0.0.0    Microsoft.PowerShell.LocalAccounts
Cmdlet          Disable-LocalUser                  1.0.0.0    Microsoft.PowerShell.LocalAccounts
Cmdlet          Enable-LocalUser                   1.0.0.0    Microsoft.PowerShell.LocalAccounts
Cmdlet          Get-LocalGroup                     1.0.0.0    Microsoft.PowerShell.LocalAccounts
Cmdlet          Get-LocalGroupMember               1.0.0.0    Microsoft.PowerShell.LocalAccounts
Cmdlet          Get-LocalUser                      1.0.0.0    Microsoft.PowerShell.LocalAccounts
Cmdlet          New-LocalGroup                     1.0.0.0    Microsoft.PowerShell.LocalAccounts
Cmdlet          New-LocalUser                      1.0.0.0    Microsoft.PowerShell.LocalAccounts
Cmdlet          Remove-LocalGroup                  1.0.0.0    Microsoft.PowerShell.LocalAccounts
Cmdlet          Remove-LocalGroupMember            1.0.0.0    Microsoft.PowerShell.LocalAccounts
Cmdlet          Remove-LocalUser                   1.0.0.0    Microsoft.PowerShell.LocalAccounts
Cmdlet          Rename-LocalGroup                  1.0.0.0    Microsoft.PowerShell.LocalAccounts
Cmdlet          Rename-LocalUser                   1.0.0.0    Microsoft.PowerShell.LocalAccounts
Cmdlet          Set-LocalGroup                     1.0.0.0    Microsoft.PowerShell.LocalAccounts
Cmdlet          Set-LocalUser                      1.0.0.0    Microsoft.PowerShell.LocalAccounts
```

*B. Polecenie cmdlet **Get-LocalUser** pobiera konta użytkowników lokalnych.*

Za pomocą poleceń poniżej

1. Wyświetli konta użytkowników

Get-LocalUser

```
Name               Enabled Description
----               ------- -----------
admin              True
Administrator      False   Wbudowane konto do administrowania komputerem/domeną
Gość               False   Wbudowane konto do dostępu do komputera/domeny dla gościa
Konto domyślne     False   Konto użytkownika zarządzane przez system.
WDAGUtilityAccount False   Konto użytkownika zarządzane i używane przez system na potrzeby
```

2. Jeśli chcesz zobaczyć wszystkie dostępne parametry, prześlij wyniki do polecenia cmdlet Select:

a) Uruchomienie polecenia cmdlet bez żadnych parametrów zwraca wszystkie konta.

Get-LocalUser | Select *

b) można dodać parametry -Name lub -SID, aby zwrócić informacje o określonym koncie.

Get-LocalUser -Name "admin"

```
Name  Enabled Description
----  ------- -----------
admin True
```

c) Poniższe polecenie zwraca konto użytkownika z identyfikatorem bezpieczeństwa

Get-LocalUser -Name "admin" | select *

```
AccountExpires         :
Description            :
Enabled                : True
FullName               :
PasswordChangeableDate : 26.06.2024 10:14:17
PasswordExpires        :
UserMayChangePassword  : True
PasswordRequired       : False
PasswordLastSet        : 26.06.2024 10:14:17
LastLogon              : 01.08.2024 12:55:15
Name                   : admin
SID                    : S-1-5-21-3664158698-3187143441-3837179366-1001
PrincipalSource        : Local
ObjectClass            : User
```

3. Wyświetli czas ostatniego logowania użytkowników

a) Aby znaleźć wszystkich użytkowników, którzy zalogowali się w ciągu ostatnich 10 dni, uruchom

Get-LocalUser | Where-Object {$_.Lastlogon -ge (Get-Date).AddDays(-10)} | Select-Object Name,Enabled,SID,Lastlogon | Format-List

```
Name      : admin
Enabled   : True
SID       : S-1-5-21-3664158698-3187143441-3837179366-1001
LastLogon : 01.08.2024 12:55:15
```

b) Aby wyszukać użytkowników, którzy nie logowali się w ciągu ostatnich 30 dni, uruchom

Get-LocalUser | Where-Object {$_.Lastlogon -le (Get-Date).AddDays(-30)} | Select-Object Name,Enabled,SID,Lastlogon | Format-List

W pole wyszukiwania wpisz Zarządzanie komputerem, otwórz > Użytkownicy i grupy lokalne > Użytkownicy. Odszukaj i przedstaw zmiany.

Zgłoszenie 1

*C. Polecenie cmdlet **New-LocalUser** tworzy lokalne konto użytkownika.*

Za pomocą poleceń poniżej

1. Utwórz konto użytkownika

New-LocalUser -Name "User02" -Description "Opis konta." -NoPassword

Polecenie tworzy lokalne konto użytkownika i nie określa parametrów AccountExpires (określa, kiedy konto użytkownika wygasa) ani Password (określa hasło do konta użytkownika). Dlatego konto nie wygasa ani nie ma domyślnie hasła.

```
Name   Enabled Description
----   ------- -----------
User02 True    Opis konta.
```

2. Utwórz konto użytkownika z hasłem

```
$Password = Read-Host -AsSecureString
```

Pierwsze polecenie monituje o hasło za pomocą polecenia cmdlet Read-Host. Polecenie przechowuje hasło jako bezpieczny ciąg w zmiennej $Password.

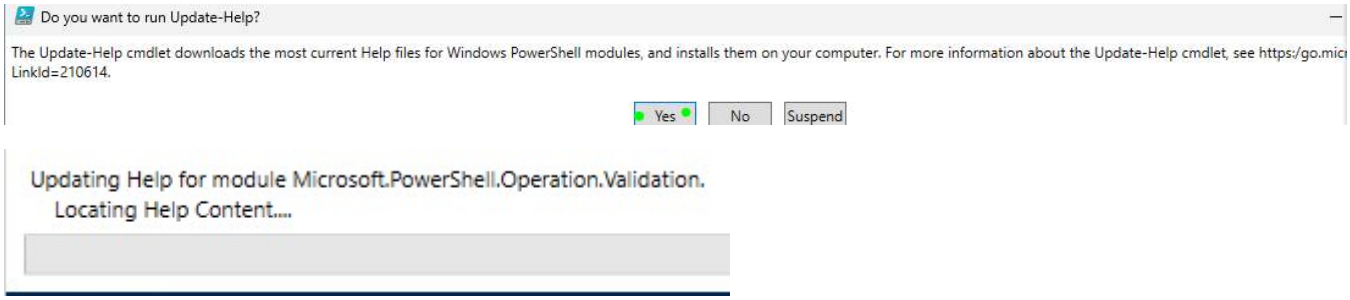

```
New-LocalUser "User03" -Password $Password -FullName "Trzeci uzytkownik" -Description "Opis konta."
```

Drugie polecenie tworzy konto użytkownika lokalnego przy użyciu hasła przechowywanego w zmiennej $Password. Polecenie określa nazwę użytkownika, pełną nazwę i opis konta użytkownika.

```
Name   Enabled Description
----   ------- -----------
User03 True    Opis konta.
```

3. Wyświetl składnie polecenia New-LocalUser i zinterpretuj w zeszycie

```
Get-Help New-LocalUser
```




```
NAME
    New-LocalUser

SYNOPSIS
    Creates a local user account.


SYNTAX
    New-LocalUser [-Name] <System.String> [-AccountExpires <System.DateTime>] [-AccountNeverExpires] [-Description <System.String>] [-Disabled]
    [-FullName <System.String>] -NoPassword [-UserMayNotChangePassword] [-Confirm] [-WhatIf] [<CommonParameters>]

    New-LocalUser [-Name] <System.String> [-AccountExpires <System.DateTime>] [-AccountNeverExpires] [-Description <System.String>] [-Disabled]
    [-FullName <System.String>] -Password <System.Security.SecureString> [-PasswordNeverExpires] [-UserMayNotChangePassword] [-Confirm] [-WhatIf
    ] [<CommonParameters>]
```

4. Wyświetl przykłady w składni polecenia New-LocalUser i zinterpretuj te przykłady w zeszycie

```
NAME
    New-LocalUser

SYNOPSIS
    Creates a local user account.


    -------------- Example 1: Create a user account --------------

    New-LocalUser -Name 'User02' -Description 'Description of this account.' -NoPassword

    Name    Enabled  Description
    ----    -------  -----------
    User02  True     Description of this account.

    This command creates a local user account and doesn't specify the AccountExpires or Password parameters. The account doesn't expire or have
    a password.
    ----- Example 2: Create a user account that has a password -----

    $Password = Read-Host -AsSecureString
    $params = @{
        Name        = 'User03'
        Password    = $Password
        FullName    = 'Third User'
        Description = 'Description of this account.'
    }
    New-LocalUser @params

    Name    Enabled  Description
    ----    -------  -----------
    User03  True     Description of this account.

    The first command uses the `Read-Host` cmdlet to prompts you for a password. The command stores the password as a secure string in the `$Pas
    sword` variable.

    The second command creates a local user account and sets the new account's password to the secure string stored in `$Password`. The command
    specifies a user name, full name, and description for the user account.
```

5. Ustawianie datę wygaśnięcia

Domyślnie nowe konto użytkownika nie wygaśnie, ale za pomocą cmdletu New-LocalUser możemy ustawić datę wygaśnięcia konta. Dla daty będziemy musieli użyć obiektu PowerShell DateTime:

```
$date = Get-Date -Year 2032 -Month 06 -Day 10
```

```
New-LocalUser -Name "User04" -Password $password -AccountExpires $date -FullName "User04" -Description "Opis konta."
```

```
Name   Enabled Description
----   ------- -----------
User04 True    Opis konta.
```

```
Get-LocalUser -Name "User04" | Select-Object AccountExpires
```

```
AccountExpires
--------------
10.06.2032 13:09:40
```

6. Tworzenie użytkowników:

a) z wskazanym hasłem w poleceniu

```
New-LocalUser -Name "User05" -Password (ConvertTo-SecureString "Pa$$w0rd" -AsPlainText -Force)
```

```
Name   Enabled Description
----   ------- -----------
User05 True
```

b) konfiguracja hasła nigdy nie wygasa ustawienie PasswordNeverExpires:

```
Set-LocalUser -Name "User05" -PasswordNeverExpires $true -Description "Opis konta5”
```

```
Get-LocalUser -Name "User05" | Select-Object PasswordNeverExpires,Description
```

```
PasswordNeverExpires Description
-------------------- -----------
                     Opis konta5
```

W pole wyszukiwania wpisz Zarządzanie komputerem, otwórz > Użytkownicy i grupy lokalne > Użytkownicy. Odszukaj i przedstaw zmiany.

Zgłoszenie 2

*D. Polecenie cmdlet **Set-LocalUser -** ustawienie użytkownika lokalnego*

1. Zmieni opis konta użytkownika

```
Set-LocalUser -Name "User04" -Description "Opis tego 4 konta."
```

```
Get-LocalUser -Name "User04" | Select-Object Description
```

```
Description
-----------
Opis tego 4 konta.
```

2. Modyfikuje lokalne konto użytkownika. To polecenie cmdlet zresetuje hasło konta użytkownika lokalnego.

```
$Password = Read-Host -AsSecureString
```

```
$UserAccount = Get-LocalUser -Name "User02"
```

```
$UserAccount | Set-LocalUser -Password $Password
```

Aby sprawdzić, czy hasło zostało zresetowane w PowerShell, możesz użyć poniższego polecenia, które wyświetli datę i czas ostatniego ustawienia hasła dla konta użytkownika:

```
$UserAccount = Get-LocalUser -Name "User02"
```

```
$UserAccount.PasswordLastSet
```

```
czwartek, 1 sierpnia 2024 13:31:06
```

3. Modyfikuje ustawienie lokalnego konto użytkownika hasło nigdy nie wygasa

```
Set-LocalUser -Name "User04" -PasswordNeverExpires $true
```

Aby sprawdzić, czy dla lokalnego konta użytkownika User04 zostało ustawione, że hasło nigdy nie wygasa, możesz użyć następującego polecenia w PowerShell:

```
(Get-LocalUser -Name "User04").PasswordNeverExpires
```

Jeśli po wpisaniu polecenia (Get-LocalUser -Name "User04").PasswordNeverExpires w PowerShell nie otrzymałeś żadnej odpowiedzi, to może oznaczać, że polecenie zostało wykonane poprawnie, ale nie zwróciło żadnego wyniku w konsoli.

Jeśli chcesz mieć dodatkowe potwierdzenie, możesz uruchomić następujące polecenie, które zwróci bardziej szczegółowe informacje o koncie, w tym status opcji PasswordNeverExpires:

```
Get-LocalUser -Name "User04" | Select-Object Name, PasswordNeverExpires
```

```
Name   PasswordNeverExpires
----   --------------------
User04
```

4. Modyfikuje ustawienie lokalnego konto użytkownika które wskazuje, że użytkownik nie może zmienić hasło na koncie użytkownika.

```
Set-LocalUser -Name "User03" -UserMayChangePassword $False
```

Aby sprawdzić, czy dla lokalnego konta użytkownika User03 została ustawiona opcja, że użytkownik nie może zmieniać hasła (UserMayChangePassword $False), użyj:

```
Write-Output (Get-LocalUser -Name "User03").UserMayChangePassword
```

```
False
```

*E. Polecenie cmdlet **New-LocalGroup** tworzy lokalną grupę zabezpieczeń w menedżerze kont zabezpieczeń.*

1: Utwórz grupę bezpieczeństwa o nazwie SecurityGroup04.

```
New-LocalGroup -Name "SecurityGroup04"
```

```
Name            Description
----            -----------
SecurityGroup04
```

*F. Polecenie cmdlet **Set-LocalGroup** zmienia lokalną grupę zabezpieczeń.*

1: Zmień opis grupy

```
Set-LocalGroup -Name "SecurityGroup04" -Description "To jest przykładowy opis."
```

Aby sprawdzić, czy opis dla lokalnej grupy bezpieczeństwa SecurityGroup04 został ustawiony, użyj:

```
Get-LocalGroup -Name "SecurityGroup04" | Select-Object Name, Description
```

```
Name            Description
----            -----------
SecurityGroup04 To jest przykładowy opis.
```

W pole wyszukiwania wpisz Zarządzanie komputerem, otwórz > Użytkownicy i grupy lokalne > Użytkownicy. Odszukaj i przedstaw zmiany.

Zgłoszenie 3

*G. Polecenie cmdlet **Add-LocalGroupMember -** ustawie użytkownika członkiem grupy*

Po utworzeniu użytkownika musisz uczynić go członkiem lokalnej grupy. Bez tego użytkownik nie będzie mógł się zalogować.

1. Aby uczynić użytkownika członkiem grupy, użyj polecenia cmdlet Add-LocalGroupMember.

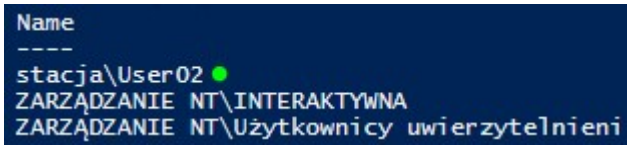
Add-LocalGroupMember wymaga tylko nazwy grupy i członka, którego chcesz dodać:

Add-LocalGroupMember -Group Użytkownicy -Member User02 (problem z interpretacją ż) dlatego użyj Get-LocalGroup | select * i zidentyfikuj SID dla grupy Użytkownicy a następnie użyj polecenia jak niżej:

Add-LocalGroupMember -SID "S-1-5-32-545" -Member User02

Aby sprawdzić, czy użytkownik User02 został dodany do lokalnej grupy o identyfikatorze SID S-1-5-32-545, użyj:

Get-LocalGroupMember -SID "S-1-5-32-545" | Select-Object Name



Polecenie cmdlet nie daje żadnych danych wyjściowych w przypadku powodzenia, a jedynie błąd, gdy nie znaleziono nazwy grupy lub członka.

2. Dodaj wielu użytkowników do grupy lokalnej za pomocą PowerShell. Po prostu rozdziel przecinkami członków w cmdlecie:

Add-LocalGroupMember -SID "S-1-5-32-545" -Member "User04", "User03"

Sprawdź:

Get-LocalGroupMember -SID "S-1-5-32-545" | Select-Object Name

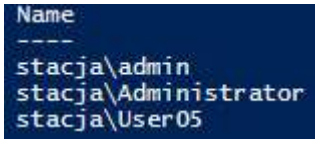
3. Dodaj członka do grupy Administratorzy.

To polecenie dodaje członka do lokalnej grupy Administratorzy.

Add-LocalGroupMember -Group "Administratorzy" -Member "User05"

Sprawdź:

Get-LocalGroupMember -Group "Administratorzy" | Select-Object Name



4. Dodaj członka do grupy „Użytkownicy pulpitu zdalnego”.

To polecenie dodaje członka do lokalnej grupy Użytkownicy pulpitu zdalnego.

Add-LocalGroupMember -Group „Użytkownicy pulpitu zdalnego” -Member User02 (problem z interpretacją ż) dlatego użyj Get-LocalGroup | select * i zidentyfikuj SID dla grupy „Użytkownicy pulpitu zdalnego” a następnie użyj polecenia jak niżej:

Add-LocalGroupMember -SID "S-1-5-32-555" -Member User03

Sprawdź:

Get-LocalGroupMember -SID "S-1-5-32-555" | Select-Object Name

```
Name
----
stacja\User03
```

5. Dodaj członka do grupy „SecurityGroup04”.

To polecenie dodaje członka do lokalnej grupy SecurityGroup04.

Add-LocalGroupMember -Group "SecurityGroup04" -Member "User04"

Sprawdź:

Get-LocalGroupMember -Group "SecurityGroup04" | Select-Object

```
ObjectClass Name          PrincipalSource
----------- ----          ---------------
User        stacja\User04 Local
```

W pole wyszukiwania wpisz Zarządzanie komputerem, otwórz > Użytkownicy i grupy lokalne > Użytkownicy. Odszukaj i przedstaw zmiany.

Zgłoszenie 4

H. Zmień nazwę

1. Polecenie cmdlet Rename-LocalGroup zmienia nazwę lokalnej grupy zabezpieczeń.

Rename-LocalGroup -Name "SecurityGroup04" -NewName "SecurityGroup"

Sprawdź:

Get-LocalGroup -Name "SecurityGroup"

```
Name          Description
----          -----------
SecurityGroup To jest przykładowy opis.
```

2. Polecenie cmdlet Rename-LocalUser zmienia nazwę lokalnego konta użytkownika.

Rename-LocalUser -Name "User05" -NewName "User01"

Sprawdź:

Get-LocalUser -Name "User01"

```
Name   Enabled Description
----   ------- -----------
User01 True    Opis konta5
```

I. Weryfikacja istnienia kont

1. Polecenie cmdlet Get-LocalUser pobiera konta użytkowników lokalnych.

```
Name               Enabled Description
----               ------- -----------
admin              True
Administrator      False   Wbudowane konto do administrowania komputerem/domeną
Gość               False   Wbudowane konto do dostępu do komputera/domeny dla gościa
Konto domyślne     False   Konto użytkownika zarządzane przez system.
User01             True    Opis konta5
User02             True    Opis konta.
User03             True    Opis konta.
User04             True    Opis tego 4 konta.
WDAGUtilityAccount False   Konto użytkownika zarządzane i używane przez system na potrzeby
```

2. Polecenie cmdlet Get-LocalGroup pobiera lokalne grupy zabezpieczeń

```
Get-LocalGroup
```

```
PS C:\Windows\system32> Get-LocalGroup

Name                                Description
----                                -----------
SecurityGroup                       To jest przykładowy opis.
Administratorzy                     Administratorzy mają pełny i nieograniczony dostęp do komputera/domeny
Administratorzy funkcji Hyper-V     Członkowie tej grupy mają pełny i nieograniczony dostęp do wszystkich
Czytelnicy dzienników zdarzeń       Członkowie tej grupy mogą odczytywać dzienniki zdarzeń z komputera lok
Goście                              Goście mają domyślnie takie same prawa dostępu jak członkowie grupy Uż
Grupa kont zarządzana przez system Członkowie tej grupy są zarządzani przez system.
IIS_IUSRS                           Grupa wbudowana używana przez program Internetowe usługi informacyjne.
Operatorzy konfiguracji sieci       Członkowie tej grupy mogą mieć niektóre uprawnienia administracyjne w
Operatorzy kopii zapasowych         Operatorzy kopii zapasowych mogą zastępować ograniczenia zabezpieczają
Operatorzy kryptograficzni          Członkowie mają autoryzację do wykonywania operacji kryptograficznych.
Operatorzy pomocy kontroli dostępu Członkowie tej grupy mogą zdalnie badać atrybuty autoryzacji i uprawni
Replikator                          Obsługuje replikację plików w domenie
Użytkownicy                         Użytkownicy nie mogą przeprowadzać przypadkowych ani celowych zmian na
Użytkownicy DCOM                    Członkowie mogą uruchamiać, aktywować i używać obiektów Distributed CO
Użytkownicy dzienników wydajności   Członkowie tej grupy mogą planować rejestrowanie liczników wydajności,
Użytkownicy monitora wydajności     Członkowie tej grupy mogą zdalnie i lokalnie uzyskiwać dostęp do danyc
Użytkownicy pulpitu zdalnego        Członkom tej grupy udziela się prawa do logowania zdalnego
Użytkownicy zaawansowani            Użytkownicy zaawansowani są uwzględnieni w celu zachowania zgodności z
Użytkownicy zarządzania zdalnego    Członkowie tej grupy mogą uzyskiwać dostęp do zasobów WMI za pomocą pr
Właściciele urządzenia              Członkowie tej grupy mogą zmieniać ustawienia dotyczące całego systemu
```

3. Polecenie cmdlet Get-LocalGroupMember pobiera członków z grupy lokalnej.

```
Get-LocalGroupMember -Group SecurityGroup
```

```
ObjectClass Name          PrincipalSource
----------- ----          ---------------
User        stacja\User04 Local
```

4. Polecenie cmdlet Remove-LocalGroup usuwa lokalne grupy zabezpieczeń. To polecenie cmdlet usuwa tylko grupę lokalną. Nie usuwa kont użytkowników, kont komputerów ani kont grupowych należących do tej grupy. Usuniętej grupy nie można odzyskać.

```
Remove-LocalGroup -Name "SecurityGroup"
```

Sprawdź:

```
Get-LocalGroup -Name "SecurityGroup"
```

```
Get-LocalGroup : Group SecurityGroup was not found.
```

Jeśli otrzymasz komunikat o błędzie, że grupa nie istnieje, oznacza to, że grupa została pomyślnie usunięta.

5. Polecenie cmdlet Remove-LocalGroupMember usuwa użytkowników lub grupy z grupy lokalnej.

```
$members = "User01"

Remove-LocalGroupMember -Group "Administratorzy" -Member $members
```

Sprawdź:

```
Get-LocalGroupMember -Group "Administratorzy" | Select-Object Name
```

To polecenie wyświetli listę członków grupy. Jeśli User01 nie pojawia się na liście, oznacza to, że został pomyślnie usunięty z grupy.

```
Name
----
stacja\admin
stacja\Administrator
```

6. Polecenie Remove-LocalUsercmdlet usuwa konta użytkowników lokalnych.

```
Remove-LocalUser -Name "User01", "User02", "User03", "User04"
```

Sprawdź:

```
$users = @("User01", "User02", "User03", "User04")

foreach ($user in $users) {

    try {

        $userInfo = Get-LocalUser -Name $user

        Write-Output "Użytkownik $user usunięty."

    } catch {

        Write-Output "Użytkownik $user istnieje."

    }

}
```

```
Użytkownik User01 usunięty.
Get-LocalUser : User User02
At line:4 char:21
+         $userInfo = Get-L
+                     ~~~~~~
    + CategoryInfo
    + FullyQualifiedErrorId

Użytkownik User02 usunięty.
Get-LocalUser : User User03
At line:4 char:21
+         $userInfo = Get-L
+                     ~~~~~~
    + CategoryInfo
    + FullyQualifiedErrorId

Użytkownik User03 usunięty.
Get-LocalUser : User User04
At line:4 char:21
+         $userInfo = Get-L
+                     ~~~~~~
    + CategoryInfo
    + FullyQualifiedErrorId

Użytkownik User04 usunięty.
```

W pole wyszukiwania wpisz Zarządzanie komputerem, otwórz > Użytkownicy i grupy lokalne > Użytkownicy. Odszukaj i przedstaw zmiany.

Zgłoszenie 5

J. Skrypty dla tworzenia nowego lokalnego użytkownika w PowerShell

Dwa skrypty, które pomogą Ci w utworzeniu lokalnego konta użytkownika w PowerShell.

W obu skryptach jest opcja zapisu pliku dziennika. Ten plik dziennika jest przechowywany w udziale sieciowym, dzięki czemu można łatwo sprawdzić, czy tworzenie na komputerze powiodło się. Dostosuj ten udział.

1. Pierwszy skrypt ma ustawione w skrypcie hasło, więc możesz po prostu uruchomić skrypt na komputerze. Pamiętaj, że aby utworzyć lokalne konto użytkownika, musisz mieć uprawnienia administratora!

W tym przypadku skrypt uczyni użytkownika członkiem grupy Administratorzy.
Możesz to zmienić na dowolną inną grupę. Upewnij się, że w pierwszej części skryptu ustawiłeś nazwę użytkownika, hasło i ścieżkę do pliku dziennika.

Pobierz cały skrypt tutaj z repozytorium Github.

2. Skrypt konta użytkownika lokalnego

Drugi skrypt tworzy lokalne konto użytkownika, które jest członkiem grup użytkowników.
Różnica w stosunku do pierwszego skryptu polega na tym, że ten skrypt poprosi o hasło.

Pobierz cały skrypt tutaj z repozytorium Github.

Nowy użytkownik lokalny powinien również mieć możliwość utworzenia konta lokalnego połączonego z kontem Microsoft. Ale nazwa użytkownika jest nadal ograniczona do 20 znaków i nie akceptuje symbolu @. Ograniczamy się tylko do kont lokalnych.

W pole wyszukiwania wpisz Zarządzanie komputerem, otwórz > Użytkownicy i grupy lokalne > Użytkownicy. Odszukaj i przedstaw zmiany.

Zgłoszenie 6