

Zarządzanie uprawnieniami NTFS z użyciem PowerShell

PowerShell umożliwia zarządzanie uprawnieniami NTFS (czyli systemem uprawnień do plików i folderów w systemach Windows) poprzez cmdlety i klasy dostępne w .NET. Do odczytywania i ustawiania uprawnień NTFS można wykorzystać moduły takie jak Get-Acl, Set-Acl oraz System.Security.AccessControl. Poniżej znajduje się szczegółowy opis i przykłady.

1. Odczytywanie uprawnień NTFS

Do odczytania bieżących uprawnień NTFS używa się cmdletu Get-Acl.

```
$acl = Get-Acl -Path "C:\MojeDane"  
$acl
```

2. Ustawianie uprawnień NTFS

Do ustawiania uprawnień NTFS używa się cmdletu Set-Acl.

Aby to zrobić, należy najpierw pobrać bieżący ACL, zmodyfikować go, a następnie zapisać zmiany.

```
$acl = Get-Acl -Path "C:\MojeDane"  
$accessRule = New-Object  
System.Security.AccessControl.FileSystemAccessRule(  
    "JanKowalski",  
    "FullControl",  
    "ContainerInherit, ObjectInherit",  
    "None",  
    "Allow"  
)  
$acl.AddAccessRule($accessRule)  
Set-Acl -Path "C:\MojeDane" -AclObject $acl
```

Parametry:

1. Identity:

- Nazwa użytkownika lub grupy, której reguła dotyczy.
- Przykłady:
 - "JanKowalski" – lokalny użytkownik.
 - "DOMAIN\Sales" – grupa domenowa.

2. **Rights:**

- Określa poziom uprawnień.
- Możliwe wartości:
 - FullControl – Pełna kontrola.
 - Modify – Modyfikacja (zmienianie, usuwanie).
 - ReadAndExecute – Odczyt i wykonywanie.
 - Read – Tylko odczyt.
 - Write – Tylko zapis.

3. **InheritanceFlags:**

- Określa, czy uprawnienia są dziedziczone przez podfoldery i pliki.
- Możliwe wartości:
 - None – Brak dziedziczenia.
 - ContainerInherit – Dziedziczenie przez foldery.
 - ObjectInherit – Dziedziczenie przez pliki.

4. **PropagationFlags:**

- Dotyczy propagacji reguły, jeśli dziedziczenie jest włączone.
- Możliwe wartości:
 - None – Brak propagacji.
 - InheritOnly – Reguła dotyczy tylko obiektów dziedziczonych.
 - NoPropagateInherit – Reguła nie jest propagowana dalej.

5. **AccessControlType:**

- Określa, czy reguła umożliwia (Allow) czy zabrania (Deny) dostępu.
- Możliwe wartości:
 - Allow
 - Deny

Przykład 2: Modyfikacja dla grupy DOMAIN\Sales

```
$accessRule = New-Object
System.Security.AccessControl.FileSystemAccessRule(

    "DOMAIN\\Sales",
    "Modify",
    "ContainerInherit, ObjectInherit",
    "None",
    "Allow"
)
```

Przykład 3: Odczyt i wykonywanie dla użytkownika Public

```
$accessRule = New-Object
System.Security.AccessControl.FileSystemAccessRule(
    "Public",
    "ReadAndExecute",
    "ContainerInherit, ObjectInherit",
    "None",
    "Allow"
)
```

3. Usuwanie uprawnień NTFS

Do usuwania uprawnień NTFS można wykorzystać metodę `RemoveAccessRule`.

Przykład:

```
$acl = Get-Acl -Path "C:\MojeDane"
$accessRule = New-Object
System.Security.AccessControl.FileSystemAccessRule(
    "JanKowalski",
    "FullControl",
    "ContainerInherit, ObjectInherit",
    "None",
    "Allow"
)
$acl.RemoveAccessRule($accessRule)
Set-Acl -Path "C:\MojeDane" -AclObject $acl
```

4. Zmiana właściciela pliku lub folderu

Do zmiany właściciela można użyć klasy `System.Security.AccessControl`.

Przykład:

```
$acl = Get-Acl -Path "C:\MojeDane"  
$acl.SetOwner([System.Security.Principal.NTAccount] "NowyWlasciciel")  
Set-Acl -Path "C:\MojeDane" -AclObject $acl
```

5. Wyświetlenie szczegółowych informacji o uprawnieniach

Przykład:

```
$acl = Get-Acl -Path "C:\MojeDane"  
$acl.Access | ForEach-Object {  
    Write-Output "Użytkownik/Grupa: $($_.IdentityReference)"  
    Write-Output "Typ: $($_.AccessControlType)"  
    Write-Output "Uprawnienia: $($_.FileSystemRights)"  
    Write-Output "Dziedziczenie: $($_.InheritanceFlags)"  
    Write-Output "-----"  
}
```

Dobre praktyki

1. **Testowanie:** Przed zastosowaniem reguł testuj je na katalogach testowych.
2. **Dziedziczenie:** Staraj się precyzyjnie definiować dziedziczenie, aby uniknąć niezamierzonych efektów.
3. **Backup ACL:** Przed wprowadzeniem zmian zapisz istniejące ACL:

```
$acl = Get-Acl -Path "C:\\Folder"  
  
$acl | Export-Clixml -Path "C:\\Backup\\folder_acl.xml"
```