26 Konfiguracja użytkownikom dostępu do zasobów

Wykonaj poniższe polecenia według przykładów:

0. Przygotowanie

Enable-PSRemoting -Force -SkipNetworkProfileCheck

New-Item -Path 'C:\udo\New Folder' -ItemType Directory

New-Item -Path 'C:\udo\New Folder\file.txt' -ItemType File -f -v "nnn"

New-Item -Path 'C:\udo\cars.xml' -ItemType File -f -v "xml_nnn"

New-Item -Path 'C:\udo\kealtcheck.html' -ItemType File -f -v "html_nnn"

New-Item -Path 'C:\udo\servicereport.html' -ItemType File -f -v "service_html_nnn"

New-Item -Path 'C:\udo\style.css' -ItemType File -f -v "css_nnn"

New-Item -Path 'C:\udo\test.xlsx' -ItemType File -f -v "xlsx_nnn"

New-Item -Path 'C:\udo\testcsv.csv' -ItemType File -f -v "csv_nnn"

New-Item -Path 'C:\udo\Profile' -ItemType Directory

New-Item -Path 'C:\udo\Profile\info.txt' -ItemType File -f -v "info o profilach"

New-Item -Path 'C:\udo\Pliki dyskow' -ItemType Directory

New-Item -Path 'C:\udo\Pliki dyskow\mapuj.txt' -ItemType File -f -v "mapowane dyski"

New-Item -Path 'C:\Udo0' -ItemType Directory

New-Item -Path 'C:\Udo1' -ItemType Directory

New-Item -Path 'C:\Udo1\kealtcheck.html' -ItemType File -f -v "html_nnn"

New-Item -Path 'C:\Udo1\test.xlsx' -ItemType File -f -v "xlsx_nnn"

New-Item -Path 'C:\Udo1\testcsv.csv' -ItemType File -f -v "csv_nnn"

New-Item -Path 'C:\Udo1\cbt.txt' -ItemType File -f -v "cbt_nnn"

New-Item -Path 'C:\Udo1\New' -ItemType Directory

New-Item -Path 'C:\Udo1\New\file.txt' -ItemType File -f -v "nnn"

New-Item -Path 'C:\Udo1\New\about.html' -ItemType File -f -v "about_html"

New-Item -Path 'C:\Udo2' -ItemType Directory

New-LocalUser -Name "User01" -Password (ConvertTo-SecureString "Pa$$w0rd" -AsPlainText -Force)

Set-LocalUser -Name "User01" -PasswordNeverExpires $true -Description "Opis konta1”

Add-LocalGroupMember -SID "S-1-5-32-545" -Member User01

New-LocalUser -Name "User02" -Password (ConvertTo-SecureString "Pa$$w0rd" -AsPlainText -Force)

Set-LocalUser -Name "User02" -PasswordNeverExpires $true -Description "Opis konta1"

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

A. Udostępnianie folderu

1. Jak udostępnić folder Windows za pomocą PowerShell?

Polecenie cmdlet New-SmbShare udostępnia folder systemu plików klientom zdalnym jako udział bloku komunikatów serwera (SMB).

Aby usunąć udział utworzony przez to polecenie cmdlet, używa się polecenia cmdlet Remove-SmbShare.

a) Udostępnij w Windows za pomocą PowerShell, folder o nazwie „udo" pod nazwą „Udostepniony" użyj polecenia New-SmbShare które jest częścią modułu SmbShare.

New-SmbShare -Path C:\udo\ -Name "Udostepniony"

Folder udo będzie domyślnie udostępniany pod nazwą „Udostepniony" z uprawnieniami do odczytu wszystkich, ponieważ nie określiliśmy zakresu.

```
Name         ScopeName Path   Description
----         --------- ----   -----------
Udostepniony *         C:\udo
```

b) Przypisz uprawnienie Pełny dostęp do konkretnego użytkownika, użyj parametru -FullAccess, po którym następuje nazwa użytkownika, aby zapewnić pełny dostęp.

New-SmbShare -Name "Udostepniony1" -Path "C:\udo\" -FullAccess "$env:ComputerName\admin","$env:ComputerName\Administrator"

Pełny dostęp otrzymają użytkownicy admin i Administrator komputera lokalnego.

```
Name          ScopeName Path   Description
----          --------- ----   -----------
Udostepniony1 *         C:\udo
```

Dla uprawnień tylko do odczytu możemy użyć parametru -Readonly.

Aby uzyskać uprawnienia do zmiany i odczytu, użyj parametru -ChangeAccess.

c) Przypisz uprawnienia dla wielu użytkowników,

New-SmbShare -Name "Udostepniony2" -Path "C:\udo\" -ChangeAccess "$env:ComputerName\User01" -FullAccess "$env:ComputerName\User02"

Powyższy przykład zapewni dostęp do zmian użytkownikowi User01 i pełny dostęp użytkownikowi User02.

```
Name          ScopeName Path   Description
----          --------- ----   -----------
Udostepniony2 *         C:\udo
```

d) Utwórz zaszyfrowany folder współdzielony, użyj parametr -Encrypted.

New-SmbShare -Name "Udostepniony3"-Path "C:\udo\"-EncryptData $true

```
Name          ScopeName Path   Description
----          --------- ----   -----------
Udostepniony3 *         C:\udo
```

e) Utwórz ukryty folder współdzielony Profile dla grupy Wszyscy, użyj parametr - FullAccess.

New-SmbShare -Name "Profile$"-Path "C:\udo\Profile"-FullAccess Wszyscy

```
Name     ScopeName Path           Description
----     --------- ----           -----------
Profile$ *         C:\udo\Profile
```

f) Utwórz ukryty „Pliki dyskow” folder współdzielony dla grupy Wszyscy, użyj parametr -FullAccess.

New-SmbShare -Name "Pliki dyskow$"-Path "C:\udo\Pliki dyskow"-FullAccess Wszyscy

```
Name          ScopeName Path                Description
----          --------- ----                -----------
Pliki dyskow$ *         C:\udo\Pliki dyskow
```

Polecenie cmdlet New-SmbMapping tworzy mapowanie bloku komunikatów serwera (SMB) na kliencie SMB na udział SMB.

g) Utwórz mapowanie SMB

New-SmbMapping -LocalPath 'X:' -RemotePath '\\stacja\Pliki dyskow$'

```
Status Local Path Remote Path
------ ---------- -----------
OK     X:         \\stacja\Pliki dyskow$
```

Polecenie cmdlet Get-SmbMapping pobiera mapowania katalogów klienta bloku komunikatów serwera (SMB) utworzone dla serwera. Może to być mapowanie z litery dysku lokalnego na zdalny folder współdzielony lub mapowanie bez ścieżki lokalnej.

h) Uzyskaj mapowania katalogów klientów SMB

Get-SmbMapping

```
Status Local Path Remote Path
------ ---------- -----------
OK     X:         \\stacja\Pliki dyskow$
```

Polecenie cmdlet Get-SmbShareAccess pobiera obiekty reprezentujące prawa, które zostały przyznane zasadom zabezpieczeń, aby uzyskać dostęp do udziału bloku komunikatów serwera (SMB).

i) Pobierz listę ACL udziału

Get-SmbShareAccess Profile$, "Pliki dyskow$", Udostepniony, Udostepniony1, Udostepniony2, Udostepniony3

```
Name          ScopeName AccountName           AccessControlType AccessRight
----          --------- -----------           ----------------- -----------
Udostepniony2 *         stacja\User02         Allow             Full
Udostepniony2 *         stacja\User01         Allow             Change
Profile$      *         Wszyscy               Allow             Full
Udostepniony3 *         Wszyscy               Allow             Read
Udostepniony  *         Wszyscy               Allow             Read
Pliki dyskow$ *         Wszyscy               Allow             Full
Udostepniony1 *         stacja\admin          Allow             Full
Udostepniony1 *         stacja\Administrator Allow             Full
```

Polecenie cmdlet Get-SmbShare pobiera obiekty, które reprezentują udziały Server Message Block (SMB) wyświetlane przez komputer.

j) Pokaż udziały SMB, wpisz: Get-SMBShare

```
Name          ScopeName Path                 Description
----          --------- ----                 -----------
ADMIN$        *         C:\Windows           Administracja zdalna
C$            *         C:\                  Domyślny udział
IPC$          *                              Zdalne wywołanie IPC
Pliki dyskow$ *         C:\udo\Pliki dyskow
Profile$      *         C:\udo\Profile
Udostepniony  *         C:\udo
Udostepniony1 *         C:\udo
Udostepniony2 *         C:\udo
Udostepniony3 *         C:\udo
```

k) Aby wyświetlić informacje o udostępnionych zasobach na komputerze lokalnym, wpisz: net share

```
Udzia       Zas˜b                           Uwaga

------------------------------------------------------------------------------
ADMIN$       C:\Windows                      Administracja zdalna
C$           C:\                             Domylny udzia
IPC$                                         Zdalne wywoanie IPC
Pliki dyskow$
             C:\udo\Pliki dyskow
Profile$     C:\udo\Profile
Udostepniony C:\udo
Udostepniony1
             C:\udo
Udostepniony2
             C:\udo
Udostepniony3
             C:\udo
Polecenie zostao wykonane pomylnie.
```

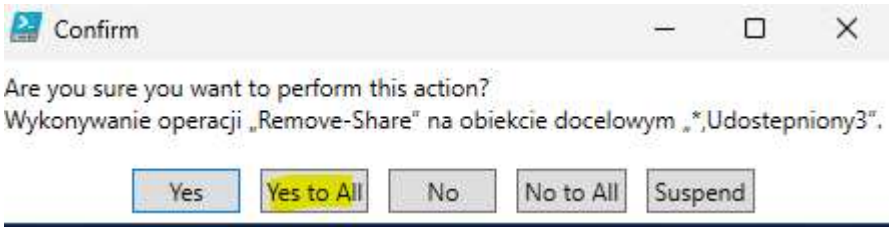
Polecenie cmdlet Remove-SmbShare usuwa jeden lub więcej udziałów bloku komunikatów serwera (SMB). Usunięcie udziału SMB wymusza rozłączenie wszystkich istniejących połączeń z udziałem. Używaj tego polecenia cmdlet z rozwagą. Klienci, którzy są odłączeni od udziału, nie mogą opróżnić danych z pamięci podręcznej lokalnie. Może to spowodować utratę danych.

Użyj polecenia cmdlet Get-SmbSession, aby określić, czy użytkownicy są połączeni z udziałem.

l) Usuń udział SMB

Remove-SmbShare -Name "Udostepniony3"

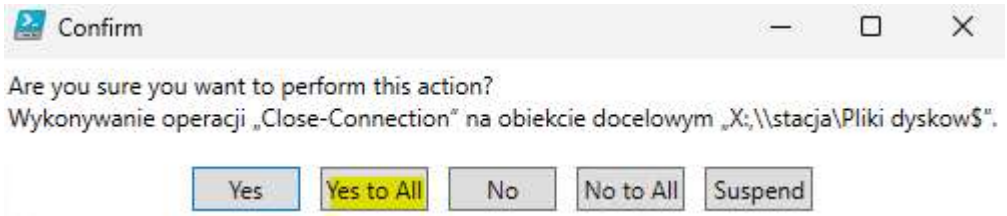
To polecenie usuwa udział SMB o nazwie Udostepniony3.



Polecenie cmdlet Remove-SmbMapping usuwa mapowanie bloku komunikatów serwera (SMB) do udziału SMB.

m)Usuń mapowanie SMB do udziału SMB

Remove-SmbMapping -LocalPath "X:"



n) Usuń udział SMB bez potwierdzenia

Remove-SmbShare -Name "Pliki dyskow$" -Force

To polecenie usuwa udział SMB o nazwie "Pliki dyskow$" bez potwierdzenia użytkownika.

Sprawdz, czy polecenie Remove-SmbShare -Name "Pliki dyskow$" -Force zostało wykonane, użyj polecenia, aby wylistować wszystkie udziały SMB i sprawdzić, czy udział “Pliki dyskow$” został usunięty:

Get-SmbShare | Where-Object { $_.Name -eq "Pliki dyskow$" }

Zgłoszenie 1

Więcej informacji o poleceniach cmdlet dla wszystkich poleceń cmdlet bloku komunikatów serwera (SMB) specyficznych dla udziału poniżej link.

https://docs.microsoft.com/en-us/powershell/module/smbshare/?view=windowsserver2022-ps

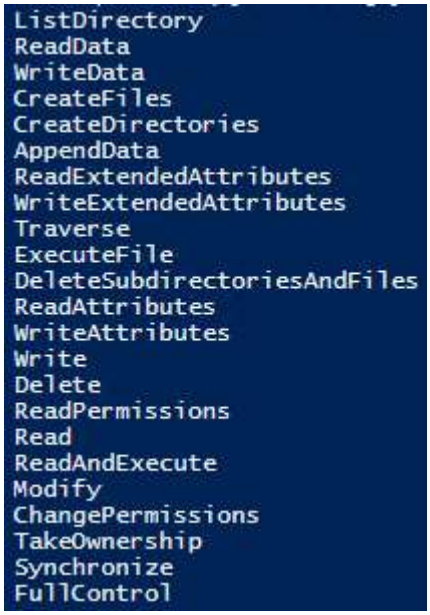
B. Uprawnienia NTFS do plików i folderów

2. Poznawanie uprawnień NTFS do plików i folderów

NTFS ma dużą liczbę uprawnień, które można ustawić w różnych kombinacjach dla plików i folderów.

a) Wyświetl wszystkie dostępne uprawnienia, wyprowadź wyliczenie System.Security.AccessControl.FileSystemRights.

[System.Enum]::GetNames([System.Security.AccessControl.FileSystemRights])



Nie zawsze jest jasne, co te uprawnienia mogą zrobić i są one podzielone na uprawnienia podstawowe i uprawnienia zaawansowane.

Uprawnienia podstawowe:

Pełna kontrola (Full Control): użytkownicy mogą modyfikować, dodawać, przenosić i usuwać pliki i katalogi, a także powiązane z nimi właściwości. Ponadto użytkownicy mogą zmieniać ustawienia uprawnień dla wszystkich plików i podkatalogów.

Modyfikuj (Modify): Użytkownicy mogą przeglądać i modyfikować pliki i właściwości plików, w tym usuwać i dodawać pliki do katalogu lub właściwości pliku do pliku.

Odczyt i wykonanie (Read & Execute): użytkownicy mogą uruchamiać pliki wykonywalne, w tym skrypt

Czytaj (Read): Użytkownicy mogą przeglądać pliki, właściwości plików i katalogi.

Zapis (Write): Użytkownicy mogą pisać do pliku i dodawać pliki do katalogów.

Uprawnienia zaawansowane:

Przejdź przez folder/wykonaj plik (Traverse Folder/Execute File): Zezwalaj na nawigację po folderach, nawet jeśli użytkownik nie ma wyraźnych uprawnień do tych plików lub folderów. Dodatkowo użytkownicy mogą uruchamiać pliki wykonywalne.

Lista folderów/odczyt danych (List Folder/Read Data): Możliwość przeglądania listy plików i podfolderów w folderze, a także przeglądania zawartości plików w nim zawartych.

Odczyt atrybutów (Read Attributes): wyświetla atrybuty pliku lub folderu.

Zapis Atrybutów (Write Attributes): Zmień atrybuty pliku lub folderu.

Odczyt rozszerzonych atrybutów (Read Extended Attributes): wyświetla rozszerzone atrybuty pliku lub folderu.

Zapis rozszerzonych atrybutów (Write Extended Attributes): Zmień rozszerzone atrybuty pliku lub folderu.

Utwórz pliki/zapisz dane (Create Files/Write Data): Zezwalaj na tworzenie plików w folderze, podczas gdy dane zapisu umożliwiają zmiany w plikach w folderze.

Utwórz foldery/Dołącz dane (Create Folders/Append Data): Twórz foldery w istniejącym folderze i zezwalaj na dodawanie danych do pliku, ale nie zmieniaj, nie usuwaj ani nie nadpisuj istniejących danych w pliku.

Usuń (Delete): Możliwość usunięcia pliku lub folderu.

Uprawnienia do odczytu (Read Permissions): użytkownicy mogą czytać uprawnienia do pliku lub folderu.

Zmień uprawnienia (Change Permissions): Użytkownicy mogą zmieniać uprawnienia pliku lub folderu.

Przejmij na własność (Take Ownership): użytkownicy mogą przejąć na własność plik lub folder.

Synchronizuj (Synchronize): Użyj pliku lub folderu do synchronizacji. Dzięki temu wątek może czekać, aż obiekt znajdzie się w stanie zasygnalizowanym.

3. Pobieranie uprawnień dostępu do pliku i folderu

Teraz, gdy wiemy, jakie są uprawnienia, możemy spojrzeć na dany folder i zobaczyć, jakie są przypisane uprawnienia. Używając cmdletu Get-ACL pobierz reguły dostępu do obiektu.

a) lokalnie: Get-ACL -Path "C:\udo"

```
    Directory: C:\


Path Owner                    Access
---- -----                    ------
udo  BUILTIN\Administratorzy BUILTIN\Administratorzy Allow  FullControl...
```

b) sieciowo: Get-Acl \\stacja\Udostepniony

```
    Directory:


Path Owner                    Access
---- -----                    ------
     BUILTIN\Administratorzy BUILTIN\Administratorzy Allow  FullControl...
```

c) przez zmienne: $aclsf = Get-Acl \\stacja\Udostepniony
Domyślny widok nie dostarcza nam mnóstwa informacji, więc rozszerzmy go.

d) Zobacz szczegółowo z wykorzystaniem, jakie uprawnienia są ustawione w tym folderze.

(Get-ACL -Path "C:\udo").Access | Format-Table IdentityReference,FileSystemRights,AccessControlType,IsInherited,InheritanceFlags -AutoSize

```
IdentityReference                          FileSystemRights AccessControlType IsInherited InheritanceFlags
-----------------                          ---------------- ----------------- ----------- ----------------
BUILTIN\Administratorzy                         FullControl             Allow        True ...erInherit, ObjectInherit
ZARZĄDZANIE NT\SYSTEM                           FullControl             Allow        True ...erInherit, ObjectInherit
BUILTIN\Użytkownicy                ReadAndExecute, Synchronize             Allow        True ...erInherit, ObjectInherit
ZARZĄDZANIE NT\Użytkownicy uwierzytelnieni       Modify, Synchronize             Allow        True                 None
ZARZĄDZANIE NT\Użytkownicy uwierzytelnieni                -536805376             Allow        True ...erInherit, ObjectInherit
```

Jak widać, jest tu o wiele więcej uprawnień. To, co widać powyżej, to typowe uprawnienia użytkownika do nowo utworzonego folderu.

e) zobacz uprawnienia do pliku C:\udo\test.xlsx

```
(Get-ACL -Path "C:\udo\test.xlsx").Access | Format-Table
IdentityReference,FileSystemRights,AccessControlType,IsInherited,InheritanceFlags -AutoSize
```

```
IdentityReference                          FileSystemRights AccessControlType IsInherited InheritanceFlags
-----------------                          ---------------- ----------------- ----------- ----------------
BUILTIN\Administratorzy                         FullControl             Allow        True             None
ZARZĄDZANIE NT\SYSTEM                           FullControl             Allow        True             None
BUILTIN\Użytkownicy                ReadAndExecute, Synchronize             Allow        True             None
ZARZĄDZANIE NT\Użytkownicy uwierzytelnieni         Modify, Synchronize             Allow        True             None
```

Jak można się domyślić, są one nieco inne i nie zawierają większości reguł dostępu, których potrzebują foldery.

4. Modyfikowanie plików i uprawnień do folderów

Jak w takim razie postąpimy z aktualizacją uprawnień do plików i folderów?
Co by było, gdybyśmy chcieli przyznać nowemu użytkownikowi prawa do odczytu pliku?

a) Aby to zrobić w PowerShell, wykonaj czteroetapowy proces.

- Pobierz istniejące reguły ACL
- Utwórz nowy FileSystemAccessRule do zastosowania
- Dodaj nową regułę ACL do istniejącej polisy uprawnień
- Zastosuj nową listę ACL do istniejącego pliku lub folderu za pomocą Set-ACLAby stworzyć samą regułę, musimy utworzyć taką FileSystemAccessRule, która ma konstruktor taki jak: Identity String, FileSystemRights, AccessControlType .

```
$ACL = Get-ACL -Path "C:\udo\test.xlsx"

$AccessRule = New-Object
System.Security.AccessControl.FileSystemAccessRule("User01","Read","Allow")

$ACL.SetAccessRule($AccessRule)

$ACL | Set-Acl -Path "C:\udo\test.xlsx"

(Get-ACL -Path "C:\udo\test.xlsx").Access | Format-Table
IdentityReference,FileSystemRights,AccessControlType,IsInherited,InheritanceFlags -AutoSize
```

```
IdentityReference                          FileSystemRights AccessControlType IsInherited InheritanceFlags
-----------------                          ---------------- ----------------- ----------- ----------------
stacja\User01                                   Read, Synchronize             Allow       False             None
BUILTIN\Administratorzy                         FullControl             Allow        True             None
ZARZĄDZANIE NT\SYSTEM                           FullControl             Allow        True             None
BUILTIN\Użytkownicy                ReadAndExecute, Synchronize             Allow        True             None
ZARZĄDZANIE NT\Użytkownicy uwierzytelnieni         Modify, Synchronize             Allow        True             None
```

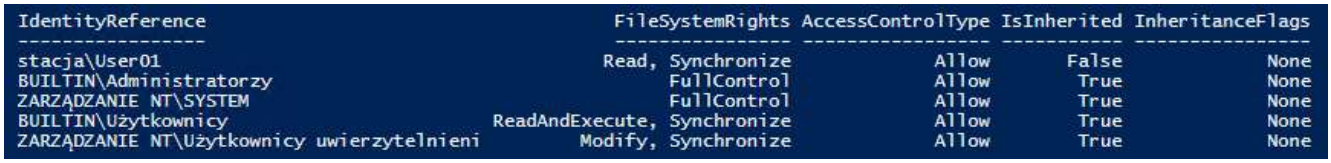
Jak widać w powyższym procesie, zmiana uprawnień jest szybka i łatwa, a konstruktory FileSystemAccessRule obiektu są proste.

5. Kopiowanie uprawnień do nowego obiektu

Skoro ustawiliśmy User01 dostęp do odczytu do naszego pliku, co by było, gdybyśmy chcieli skopiować ten sam zestaw uprawnień do innego pliku?

a) Użyj możliwości potoku PowerShell, aby przenieść uprawnienia z jednego obiektu do drugiego.

```
Get-ACL -Path "C:\udo\test.xlsx" | Set-ACL -Path "C:\udo\testcsv.csv"

(Get-ACL -Path "C:\udo\testcsv.csv").Access | Format-Table
IdentityReference,FileSystemRights,AccessControlType,IsInherited,InheritanceFlags -AutoSize
```



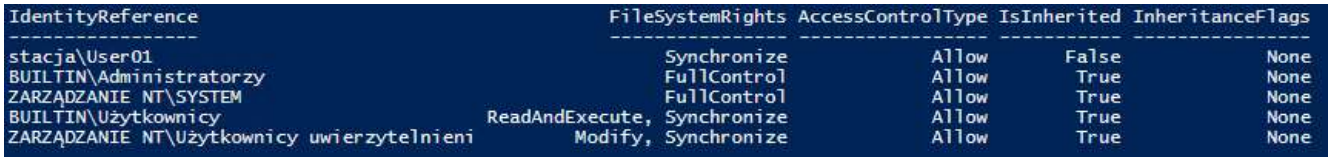
IdentityReference	FileSystemRights	AccessControlType	IsInherited	InheritanceFlags
stacja\User01	Read, Synchronize	Allow	False	None
BUILTIN\Administratorzy	FullControl	Allow	True	None
ZARZĄDZANIE NT\SYSTEM	FullControl	Allow	True	None
BUILTIN\Użytkownicy	ReadAndExecute, Synchronize	Allow	True	None
ZARZĄDZANIE NT\Użytkownicy uwierzytelnieni	Modify, Synchronize	Allow	True	None

6. Usuwanie uprawnień do plików lub folderów

Po dodaniu tych uprawnień zdecydowaliśmy, że User01 nie powinno mieć uprawnień do pliku C:\udo\test.xlsx. Różnica w usuwaniu reguły polega na tym, że musimy odtworzyć dokładnie FileSystemAccessRule, który chcemy usunąć. Jest to sposób usuwania uprawnień, który usuwa niejasności dotyczące tego, jakie uprawnienia należy usunąć.

a) Usuń uprawnienia w sposób jak wyżej opisałem

```
$ACL = Get-ACL -Path "C:\udo\test.xlsx"

$AccessRule = New-Object System.Security.AccessControl.FileSystemAccessRule("User01","Read","Allow")

$ACL.RemoveAccessRule($AccessRule)

$ACL | Set-Acl -Path "C:\udo\test.xlsx"

(Get-ACL -Path "C:\udo\test.xlsx").Access | Format-Table IdentityReference,FileSystemRights,AccessControlType,IsInherited,InheritanceFlags -AutoSize
```



IdentityReference	FileSystemRights	AccessControlType	IsInherited	InheritanceFlags
stacja\User01	Synchronize	Allow	False	None
BUILTIN\Administratorzy	FullControl	Allow	True	None
ZARZĄDZANIE NT\SYSTEM	FullControl	Allow	True	None
BUILTIN\Użytkownicy	ReadAndExecute, Synchronize	Allow	True	None
ZARZĄDZANIE NT\Użytkownicy uwierzytelnieni	Modify, Synchronize	Allow	True	None

Jak widać powyżej, usunęliśmy uprawnienia z tego obiektu. Uprawnienie synchronize to specjalne uprawnienie używane przez system operacyjny do utrzymania właściwej kontroli nad uprawnieniami do plików i folderów.

7. Modyfikowanie dziedziczenia i własności

Wreszcie dwa dodatkowe zadania związane z systemem plików, które są bardzo przydatne, to włączanie i wyłączanie dziedziczenia w folderze oraz zmiana właściciela plików.

Wyłącz/włącz dziedziczenie uprawnień

Aby zmodyfikować właściwości dziedziczenia obiektu, musimy użyć SetAccessRuleProtection. Pierwsza wartość określa, czy folder dziedziczy uprawnienia dostępu, czy nie. Ustawienie tej wartości na $true wyłączy dziedziczenie, jak pokazano w poniższym przykładzie.

Właściwość druga zachowaj dziedziczenie pozwala skopiować istniejące odziedziczone uprawnienia do obiektu, jeśli usuwamy dziedziczenie. Może to być bardzo ważne, abyśmy nie stracili dostępu do obiektu, ale nie jest to pożądane.

a) Wykonaj

```
$ACL = Get-Acl -Path "C:\udo"

$ACL.SetAccessRuleProtection($true,$false)

$ACL | Set-Acl -Path "C:\udo"
```

Sprawdz, czy polecenia dotyczące listy kontroli dostępu (ACL) zostały wykonane poprawnie, użyj polecenia Get-Acl dla tej samej ścieżki i przeanalizuj wyniki.:

```
$ACL = Get-Acl -Path "C:\udo"
```

```
$ACL.Access
```

```
Path Owner                      Access
---- -----                      ------
udo  BUILTIN\Administratorzy
```

Możesz otrzymać błąd, Set-Acl: The process does not possess the 'SeSecurityPrivilege' privilege which is required for this operation. co oznacza, że powinieneś uruchomić ten proces na koncie Administratora.

Zwróć uwagę, że uprawnienia nie są już prawdziwe w ramach IsInherited. Oznacza to, że pomyślnie skopiowaliśmy uprawnienia i złamaliśmy dziedziczenie w tym folderze.

Zmień właściciela

Jeśli chcesz zmienić właściciela pliku, możesz to zrobić po prostu za pomocą metody SetOwner.
Po uruchomieniu polecenia Get-ACL widzimy, że właściciel zmienił się na nowego użytkownika.

b) Wykonaj zmianę właściciela

```
$ACL = Get-Acl -Path "C:\udo"
```

```
$User = New-Object System.Security.Principal.Ntaccount("User01")
```

```
$ACL.SetOwner($User)
```

```
$ACL | Set-Acl -Path "C:\udo"
```

```
Get-ACL -Path "C:\udo"
```

```
    Directory: C:\

Path Owner          Access
---- -----          ------
udo  stacja\User01
```

c) Uzyskaj więcej informacji jak pokazano poniżej:

```
Get-Acl -Path C:\udo | Format-Table -Wrap
```

```
    Directory: C:\

Path Owner          Access
---- -----          ------
udo  stacja\User01
```

d) Uzyskaj więcej informacji, użyj listy formatów:

```
Get-ACL -Path "C:\udo" | Format-List
```

```
Path   : Microsoft.PowerShell.Core\FileSystem::C:\udo
Owner  : stacja\User01
Group  : stacja\Brak
Access :
Audit  :
Sddl   : O:S-1-5-21-3664158698-3187143441-3837179366-1002G:S-1-5-21-3664158698-3187143441-3837179366-513D:PAI
```

e) Uzyskaj więcej informacji jak pokazano poniżej:

(Get-Acl -Path C:\Udo1).Access

```
FileSystemRights  : FullControl
AccessControlType : Allow
IdentityReference : BUILTIN\Administratorzy
IsInherited       : True
InheritanceFlags  : ContainerInherit, ObjectInherit
PropagationFlags  : None

FileSystemRights  : FullControl
AccessControlType : Allow
IdentityReference : ZARZĄDZANIE NT\SYSTEM
IsInherited       : True
InheritanceFlags  : ContainerInherit, ObjectInherit
PropagationFlags  : None

FileSystemRights  : ReadAndExecute, Synchronize
AccessControlType : Allow
IdentityReference : BUILTIN\Użytkownicy
IsInherited       : True
InheritanceFlags  : ContainerInherit, ObjectInherit
PropagationFlags  : None

FileSystemRights  : Modify, Synchronize
AccessControlType : Allow
IdentityReference : ZARZĄDZANIE NT\Użytkownicy uwierzytelnieni
IsInherited       : True
InheritanceFlags  : None
PropagationFlags  : None

FileSystemRights  : -536805376
AccessControlType : Allow
IdentityReference : ZARZĄDZANIE NT\Użytkownicy uwierzytelnieni
IsInherited       : True
InheritanceFlags  : ContainerInherit, ObjectInherit
PropagationFlags  : InheritOnly
```

f) Uzyskaj więcej informacji jak pokazano poniżej:

(Get-Acl -Path C:\Udo1).Access.IdentityReference

```
Value
-----
BUILTIN\Administratorzy
ZARZĄDZANIE NT\SYSTEM
BUILTIN\Użytkownicy
ZARZĄDZANIE NT\Użytkownicy uwierzytelnieni
ZARZĄDZANIE NT\Użytkownicy uwierzytelnieni
```

g) Aby dowiedzieć się, jakich parametrów można użyć, naciśnij TAB w oknie PowerShell po wpisaniu kropki. Na przykład wpisując (Get-Acl C:\Udo1). a następnie naciśnięcie klawisza TAB doda Dostęp do polecenia. Wielokrotne naciskanie TAB spowoduje przewinięcie wszystkich opcji.

(Get-Acl -Path C:\Udo1).[TAB]

```
(Get-Acl -Path C:\Udo1).Access
(Get-Acl -Path C:\Udo1).Access.IdentityReference
(Get-Acl -Path C:\Udo1).Access
```

Efekt:

```
FileSystemRights  : FullControl
AccessControlType : Allow
IdentityReference : BUILTIN\Administratorzy
IsInherited       : True
InheritanceFlags  : ContainerInherit, ObjectInherit
PropagationFlags  : None

FileSystemRights  : FullControl
AccessControlType : Allow
IdentityReference : ZARZĄDZANIE NT\SYSTEM
IsInherited       : True
InheritanceFlags  : ContainerInherit, ObjectInherit
PropagationFlags  : None

FileSystemRights  : ReadAndExecute, Synchronize
AccessControlType : Allow
IdentityReference : BUILTIN\Użytkownicy
IsInherited       : True
InheritanceFlags  : ContainerInherit, ObjectInherit
PropagationFlags  : None

FileSystemRights  : Modify, Synchronize
AccessControlType : Allow
IdentityReference : ZARZĄDZANIE NT\Użytkownicy uwierzytelnieni
IsInherited       : True
InheritanceFlags  : None
PropagationFlags  : None

FileSystemRights  : -536805376
AccessControlType : Allow
IdentityReference : ZARZĄDZANIE NT\Użytkownicy uwierzytelnieni
IsInherited       : True
InheritanceFlags  : ContainerInherit, ObjectInherit
PropagationFlags  : InheritOnly
```

Gdy jest używany samodzielnie, Get-Acl może jednocześnie raportować tylko jeden plik lub katalog. Jeśli chcesz wygenerować raport w hierarchii folderów, musisz przekazać każdy folder do Get-Acl za pomocą pętli ForEach.

h) Użyj polecenia cmdlet Get-ChildItem, aby utworzyć obiekt przechowujący hierarchię folderów, którą chcę przekazać do Get-Acl.

```
$FolderPath = Get-ChildItem -Directory -Path "C:\Udo1" -Recurse -Force
```

```
$FolderPath
```

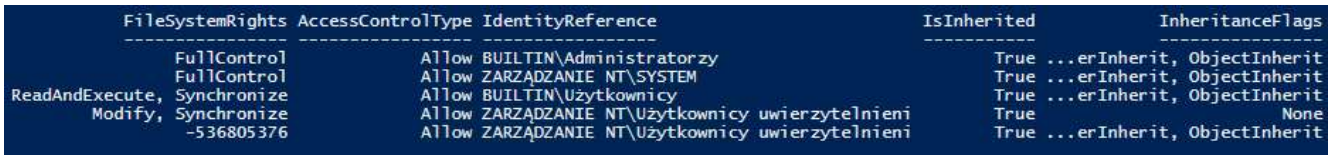
```
    Directory: C:\Udo1


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        01.08.2024     14:53                New
```

Jeśli polecenie zwróci listę folderów, oznacza to, że zostało wykonane poprawnie. Możesz również sprawdzić szczegóły każdego obiektu w zmiennej, aby upewnić się, że zawiera ona wszystkie potrzebne informacje, takie jak ścieżka, nazwa i inne atrybuty folderu.

Jeśli chcesz sprawdzić, czy wszystkie foldery mają ustawione odpowiednie ACL, możesz użyć zmiennej $FolderPath w połączeniu z poleceniem Get-Acl w następujący sposób:

```
foreach ($Folder in $FolderPath) {
    $Acl = Get-Acl -Path $Folder.FullName
    $Acl.Access | Format-Table -AutoSize
```

}



To polecenie przejdzie przez każdy folder w zmiennej $FolderPath, pobierze jego ACL i wyświetli reguły dostępu w formie tabeli, co pozwoli Ci na łatwą weryfikację ustawień zabezpieczeń.

i) Pierwsza pętla przechodzi przez każdy folder w hierarchii. Dla każdego folderu uruchamiam inną pętlę ForEach, która wyświetla wpisy (ACE) w jego ACL, tworząc zmienną ($Properties), która formatuje dane wyjściowe w celu wyświetlenia nazwy folderu, grupy lub użytkownika w ACE, przyznanych uprawnień i czy są dziedziczone. Na koniec tworzę nowy obiekt za pomocą zmiennej $Properties, która jest wyświetlana w danych wyjściowych w oknie PowerShell.

```
ForEach ($Folder in $FolderPath) {
  $Acl = Get-Acl -Path $Folder.FullName
  ForEach ($Access in $Acl.Access) {
$Properties = [ordered]@{'Folder Name'=$Folder.FullName;'Group/User'=$Access.IdentityReference;'Permissions'=$Access.FileSystemRights;'Inherited'=$Access.IsInherited}
New-Object -TypeName PSObject -Property $Properties
        }
}
```

Możesz zobaczyć dane wyjściowe tylko listy folderów. W wynikach nie ma plików. Możesz również utworzyć tablicę ($Output) i potoku wyniki do Out-GridView lub pliku .csv.

j) Użyj PowerShell, aby uzyskać uprawnienia do plików NTFS

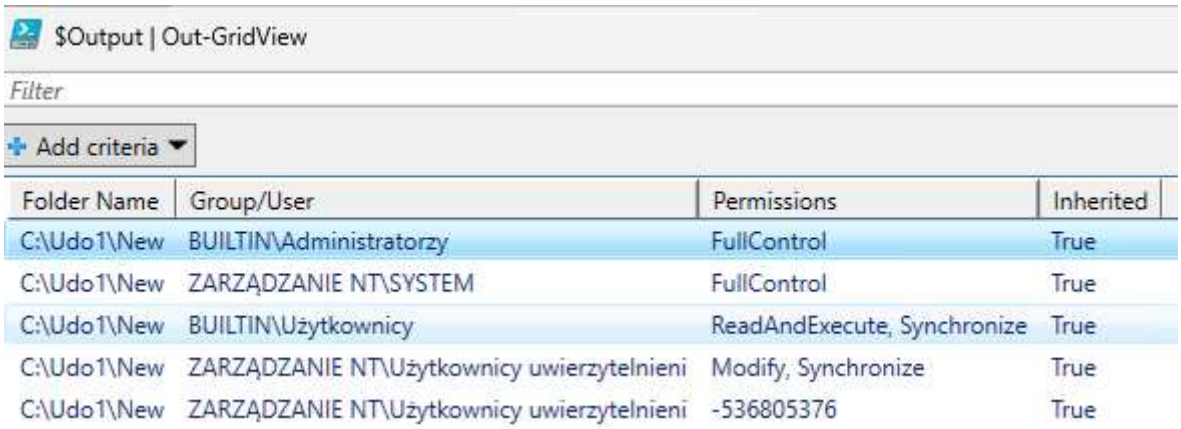
```
$FolderPath = Get-ChildItem -Directory -Path "C:\Udo1" -Recurse -Force
$Output = @()
ForEach ($Folder in $FolderPath) {
  $Acl = Get-Acl -Path $Folder.FullName
  ForEach ($Access in $Acl.Access) {
$Properties = [ordered]@{'Folder Name'=$Folder.FullName;'Group/User'=$Access.IdentityReference;'Permissions'=$Access.FileSystemRights;'Inherited'=$Access.IsInherited}
$Output += New-Object -TypeName PSObject -Property $Properties
```

}

}

$Output | Out-GridView

Skrypt i polecenia, które przedstawiłem, powinny pomóc w rozpoczęciu korzystania z programu PowerShell do raportowania uprawnień NTFS.



Folder Name	Group/User	Permissions	Inherited
C:\Udo1\New	BUILTIN\Administratorzy	FullControl	True
C:\Udo1\New	ZARZĄDZANIE NT\SYSTEM	FullControl	True
C:\Udo1\New	BUILTIN\Użytkownicy	ReadAndExecute, Synchronize	True
C:\Udo1\New	ZARZĄDZANIE NT\Użytkownicy uwierzytelnieni	Modify, Synchronize	True
C:\Udo1\New	ZARZĄDZANIE NT\Użytkownicy uwierzytelnieni	-536805376	True

Zgłoszenie 2

https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.security/get-acl?view=powershell-7.2 – więcej o Get-Acl

https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.security/set-acl?view=powershell-7.2 - więcej o Set-Acl

C. NTFSSecurity

Ustawienie uprawnień zabezpieczeń NTFS z Eksploratora plików Windows jest w porządku, gdy masz do czynienia z pojedynczym serwerem. Jest to jednak zupełnie inna sytuacja, w której musisz zmodyfikować zabezpieczenia NTFS na 100 folderach rozmieszczonych na 20 serwerach.
W przypadku tych zadań administracyjnych polegamy na programie Windows PowerShell, aby wykonać zadanie szybko, dokładnie i łatwo.

Zarządzanie uprawnieniami za pomocą PowerShell jest tylko nieco łatwiejsze niż w VBS lub wierszu poleceń, ponieważ nie ma poleceń cmdlet dla większości codziennych zadań, takich jak uzyskiwanie raportu o uprawnieniach lub dodawanie uprawnień do elementu. PowerShell oferuje tylko Get-Acl i Set-Acl, ale brakuje wszystkiego między pobieraniem a ustawianiem listy ACL.

Ten moduł wypełnia lukę.

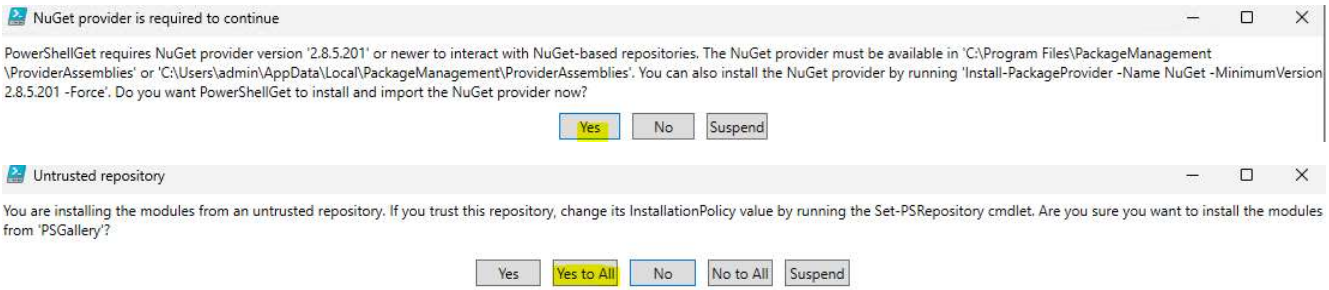
1. Przeglądanie uprawnień NTFS
a) Sprawdź, czy automatyczne ładowanie modułu PowerShell prawidłowo rozpoznaje moduł NTFSSecurity. Użyj Get-Command, aby sprawdzić obecność modułu NTFSSecurity:

Get-Command -Module NTFSSecurity

brak

b) Zainstaluj moduł NTFSSecurity:

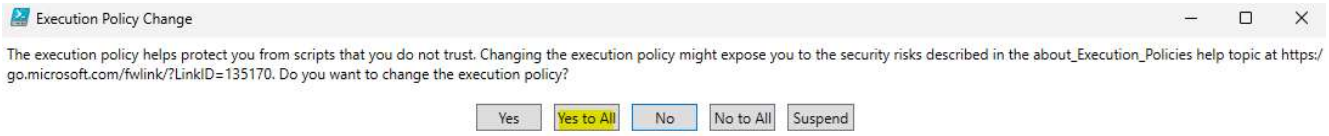
Install-Module -Name NTFSSecurity



W systemie jest włączona polityka wykonania skryptów, która uniemożliwia uruchamianie skryptów PowerShell. Jest to częsta praktyka zabezpieczeń, aby zapobiec nieautoryzowanemu lub potencjalnie szkodliwemu kodowi przed uruchomieniem.

c) Tymczasowo zmień politykę wykonania skryptów, aby umożliwić załadowanie modułu, wpisz następujące polecenie:

Set-ExecutionPolicy -ExecutionPolicy RemoteSigned



d) Załaduj moduł do bieżącej sesji PowerShell, użyj polecenia Import-Module.:

Import-Module NTFSSecurity

e) Sprawdź, czy automatyczne ładowanie modułu PowerShell prawidłowo rozpoznaje moduł NTFSSecurity. Użyj Get-Command, aby sprawdzić obecność modułu NTFSSecurity i wyliczy jego polecenia:

Get-Command -Module NTFSSecurity

CommandType	Name	Version	Source
Cmdlet	Add-NTFSAccess	4.2.6	NTFSSecurity
Cmdlet	Add-NTFSAudit	4.2.6	NTFSSecurity
Cmdlet	Clear-NTFSAccess	4.2.6	NTFSSecurity
Cmdlet	Clear-NTFSAudit	4.2.6	NTFSSecurity
Cmdlet	Copy-Item2	4.2.6	NTFSSecurity
Cmdlet	Disable-NTFSAccessInheritance	4.2.6	NTFSSecurity
Cmdlet	Disable-NTFSAuditInheritance	4.2.6	NTFSSecurity
Cmdlet	Disable-Privileges	4.2.6	NTFSSecurity
Cmdlet	Enable-NTFSAccessInheritance	4.2.6	NTFSSecurity
Cmdlet	Enable-NTFSAuditInheritance	4.2.6	NTFSSecurity
Cmdlet	Enable-Privileges	4.2.6	NTFSSecurity
Cmdlet	Get-ChildItem2	4.2.6	NTFSSecurity
Cmdlet	Get-DiskSpace	4.2.6	NTFSSecurity
Cmdlet	Get-FileHash2	4.2.6	NTFSSecurity
Cmdlet	Get-Item2	4.2.6	NTFSSecurity
Cmdlet	Get-NTFSAccess	4.2.6	NTFSSecurity
Cmdlet	Get-NTFSAudit	4.2.6	NTFSSecurity
Cmdlet	Get-NTFSEffectiveAccess	4.2.6	NTFSSecurity
Cmdlet	Get-NTFSHardLink	4.2.6	NTFSSecurity
Cmdlet	Get-NTFSInheritance	4.2.6	NTFSSecurity
Cmdlet	Get-NTFSOrphanedAccess	4.2.6	NTFSSecurity
Cmdlet	Get-NTFSOrphanedAudit	4.2.6	NTFSSecurity
Cmdlet	Get-NTFSOwner	4.2.6	NTFSSecurity
Cmdlet	Get-NTFSSecurityDescriptor	4.2.6	NTFSSecurity
Cmdlet	Get-NTFSSimpleAccess	4.2.6	NTFSSecurity
Cmdlet	Get-Privileges	4.2.6	NTFSSecurity
Cmdlet	Move-Item2	4.2.6	NTFSSecurity
Cmdlet	New-NTFSHardLink	4.2.6	NTFSSecurity
Cmdlet	New-NTFSSymbolicLink	4.2.6	NTFSSecurity
Cmdlet	Remove-Item2	4.2.6	NTFSSecurity
Cmdlet	Remove-NTFSAccess	4.2.6	NTFSSecurity
Cmdlet	Remove-NTFSAudit	4.2.6	NTFSSecurity
Cmdlet	Set-NTFSInheritance	4.2.6	NTFSSecurity
Cmdlet	Set-NTFSOwner	4.2.6	NTFSSecurity
Cmdlet	Set-NTFSSecurityDescriptor	4.2.6	NTFSSecurity
Cmdlet	Test-Path2	4.2.6	NTFSSecurity

Brak – patrz https://github.com/raandree/NTFSSecurity/wiki/How-to-install i wykonaj samodzielnie po zapoznaniu z instrukcją lub wykonaj to co jest poniżej

Instalowanie modułu PowerShell oznacza po prostu skopiowanie plików do jednego z nich

C:\Users<nazwa użytkownika>\Documents\WindowsPowerShell\Modules

C:\Program Files\WindowsPowerShell\Modules (dostępne od wersji 4.0)

C:\WINDOWS\system32\WindowsPowerShell\v1.0\Modules

Po prostu utwórz folder „NTFSSecurity" w jednej z wcześniej wymienionych ścieżek i skopiuj tam pliki z folderu NTFSSecurity.

Jeśli to zrobiłeś, moduł powinien być wymieniony w „Get-Module -ListAvailable" i można go zaimportować za pomocą „Import-Module NTFSSecurity"

Wykonaj

Get-Module -ListAvailable

set-executionpolicy UnRestricted

Import-Module NTFSSecurity

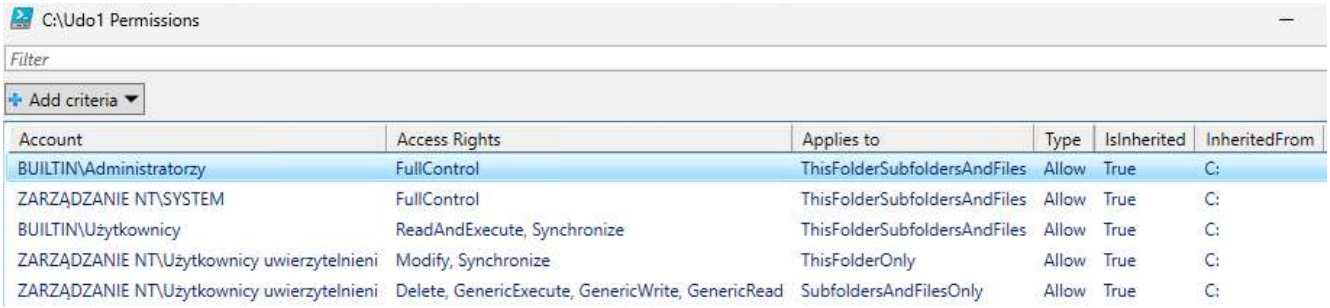
Get-Command -Module NTFSSecurity

Na woluminie NTFS foldery i pliki zawierają arbitralną listę kontroli dostępu (DACL), która z kolei składa się z co najmniej jednego wpisu kontroli dostępu (ACE). Te pozycje ACE definiują konta grup, użytkowników i komputerów, którym jawnie przyznano lub odmówiono dostępu do tych plików i folderów.

f) Wylicz wpisy NTFS ACE dla C:\Udo1 i wyświetl wyniki w widoku siatki:

Get-NTFSAccess –Path 'C:\Udo1' | Out-GridView –Title 'C:\Udo1 Permissions'

```
PS C:\WINDOWS\system32> Get-NTFSAccess -Path 'C:\Udo1' | Out-GridView -Title 'C:\Udo1 Permissions'
```



To jest widok pozycji ACE katalogu.

g) To samo możemy zrób dla poszczególnych plików:

Get-NTFSAccess -Path 'C:\Udo1\cbt.txt' | Select-Object –Property Account, AccessRights

```
Account                                                 AccessRights
-------                                                 ------------
BUILTIN\Administratorzy                                  FullControl
ZARZĄDZANIE NT\SYSTEM                                    FullControl
BUILTIN\Użytkownicy                     ReadAndExecute, Synchronize
ZARZĄDZANIE NT\Użytkownicy uwierzytelnieni       Modify, Synchronize
```

h) Wyświetl rekurencyjnie uprawnienia dla całej zawartości folderu, połą Get-ChildItem, Get-NTFSAccess i potok PowerShell:

```
Get-ChildItem –Path 'C:\Udo1' –Recurse | Get-NTFSAccess
```

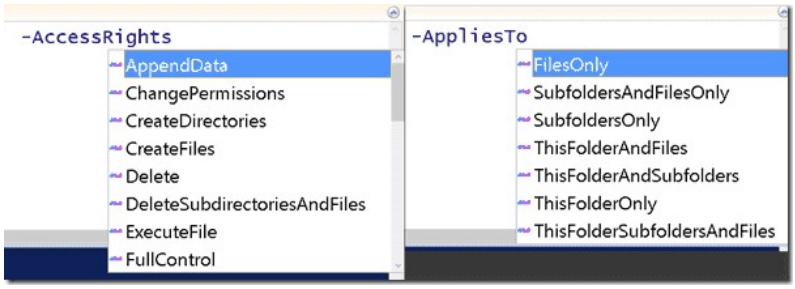
```
    Path: C:\Udo1\test.xlsx (Inheritance enabled)

Account                          Access Rights    Applies to           Type  IsInherited InheritedFrom
-------                          -------------    ----------           ----  ----------- -------------
BUILTIN\Administratorzy           FullControl      ThisFolderOnly       Allow True        C:
ZARZĄDZANIE NT\SYSTEM             FullControl      ThisFolderOnly       Allow True        C:
BUILTIN\Użytkownicy               ReadAndExecut... ThisFolderOnly       Allow True        C:
ZARZĄDZANIE NT\Użytkownicy uwier... Modify, Synch... ThisFolderOnly     Allow True        C:


    Path: C:\Udo1\testcsv.csv (Inheritance enabled)

Account                          Access Rights    Applies to           Type  IsInherited InheritedFrom
-------                          -------------    ----------           ----  ----------- -------------
BUILTIN\Administratorzy           FullControl      ThisFolderOnly       Allow True        C:
ZARZĄDZANIE NT\SYSTEM             FullControl      ThisFolderOnly       Allow True        C:
BUILTIN\Użytkownicy               ReadAndExecut... ThisFolderOnly       Allow True        C:
ZARZĄDZANIE NT\Użytkownicy uwier... Modify, Synch... ThisFolderOnly     Allow True        C:


    Path: C:\Udo1\New\about.html (Inheritance enabled)

Account                          Access Rights    Applies to           Type  IsInherited InheritedFrom
-------                          -------------    ----------           ----  ----------- -------------
BUILTIN\Administratorzy           FullControl      ThisFolderOnly       Allow True        C:
ZARZĄDZANIE NT\SYSTEM             FullControl      ThisFolderOnly       Allow True        C:
BUILTIN\Użytkownicy               ReadAndExecut... ThisFolderOnly       Allow True        C:
ZARZĄDZANIE NT\Użytkownicy uwier... Modify, Synch... ThisFolderOnly     Allow True        C:


    Path: C:\Udo1\New\file.txt (Inheritance enabled)

Account                          Access Rights    Applies to           Type  IsInherited InheritedFrom
-------                          -------------    ----------           ----  ----------- -------------
BUILTIN\Administratorzy           FullControl      ThisFolderOnly       Allow True        C:
ZARZĄDZANIE NT\SYSTEM             FullControl      ThisFolderOnly       Allow True        C:
BUILTIN\Użytkownicy               ReadAndExecut... ThisFolderOnly       Allow True        C:
ZARZĄDZANIE NT\Użytkownicy uwier... Modify, Synch... ThisFolderOnly     Allow True        C:
```

2. Modyfikowanie uprawnień NTFS

a) Przyznaj użytkownikowi o nazwie User02 i User02 uprawnienia tylko do odczytu do całej zawartości folderu Udo1. Zwróć uwagę, że polecenie cmdlet *Add-NTFSAccess* akceptuje listę rozdzielonych przecinkami identyfikatorów zabezpieczeń (SID) lub nazw kont:

```
Add-NTFSAccess –Path 'C:\Udo1' –Account 'stacja\User01', 'stacja\User02' –AccessRights Read
```

Możemy dodać parametr i wartość –AppliesTo ThisFolderOnly, jeśli chcemy zablokować dziedziczenie uprawnień NTFS do zawartości folderu. W rzeczywistości spójrz na następujący zrzut ekranu Windows PowerShell ISE; Chcę pokazać zawartość wyliczenia dla obu –AccessRights i –AppliesTo:



Sprawdź efekt polecenia Add-NTFSAccess dla konkretnych kont, użyj polecenia Get-NTFSAccess z modułu NTFSSecurity, wyświetl aktualne wpisy ACL dla ścieżki C:\Udo1. Filtruj wyniki dla określonych użytkowników.:

```
Get-NTFSAccess -Path 'C:\Udo1' | Where-Object { $_.Account -eq 'stacja\User01' -or $_.Account -eq 'stacja\User02' }
```

```
    Path: C:\Udo1 (Inheritance enabled)

Account                  Access Rights    Applies to               Type  IsInherited InheritedFrom
-------                  -------------    ----------               ----  ----------- -------------
stacja\User01            Read, Synchro... ThisFolderSubfoldersAn... Allow False
stacja\User02            Read, Synchro... ThisFolderSubfoldersAn... Allow False
```

b) Ustaw -AccessRights AppendData -AppliesTo FilesOnly

```
Add-NTFSAccess -Path 'C:\Udo1' -Account 'stacja\User01', 'stacja\User02' -AccessRights AppendData -AppliesTo FilesOnly
```

Sprawdź efekt polecenia Add-NTFSAccess dla wskazanych kont, użyj polecenia Get-NTFSAccess z modułu NTFSSecurity. Pozwoli Ci to na wyświetlenie aktualnych wpisów ACL dla ścieżki C:\Udo1 i sprawdzenie czy prawa dostępu AppendData zostały dodane tylko do plików dla użytkowników stacja\User01 i stacja\User02.:

```
Get-NTFSAccess -Path 'C:\Udo1' | Where-Object { ($_.Account -match 'stacja\\User01' -or $_.Account -match 'stacja\\User02') -and $_.AppliesTo -eq 'FilesOnly' }
```

To polecenie zwróci wpisy ACL dla użytkowników stacja\User01 i stacja\User02 z prawem AppendData, które mają zastosowanie tylko do plików. Jeśli wpisy te zostaną wyświetlone, oznacza to, że polecenie Add-NTFSAccess zostało wykonane poprawnie.

```
    Path: C:\Udo1 (Inheritance enabled)

Account                  Access Rights    Applies to               Type  IsInherited InheritedFrom
-------                  -------------    ----------               ----  ----------- -------------
stacja\User01            Read, Synchro... ThisFolderSubfoldersAn... Allow False
stacja\User01            CreateDirecto... FilesOnly                Allow False
stacja\User02            Read, Synchro... ThisFolderSubfoldersAn... Allow False
stacja\User02            CreateDirecto... FilesOnly                Allow False
```

c) Sprawdź, czy C:\Udo1 ACL pobrał naszą aktualizację:

```
Get-NTFSAccess -Path 'C:\Udo1' | Where-Object -FilterScript { $_.AccessRights -eq 'Read, Synchronize' } | Format-Table -AutoSize
```

```
    Path: C:\Udo1 (Inheritance enabled)

Account       Access Rights     Applies to                  Type  IsInherited InheritedFrom
-------       -------------     ----------                  ----  ----------- -------------
stacja\User01 Read, Synchronize ThisFolderSubfoldersAndFiles Allow False
stacja\User02 Read, Synchronize ThisFolderSubfoldersAndFiles Allow False
```

3. Określanie efektywnych uprawnień

Użyj modułu NTFSSecurity, aby wykorzystać oszczędność czasu.

a) Zobacz efektywne uprawnienia User02 do pliku C:\Udo1\New\about.html. Użyj Get-NTFSEffectiveAccess

```
Get-Item -Path 'C:\Udo1\New\about.html' | Get-NTFSEffectiveAccess -Account 'stacja\User02' | Format-List
```

```
Name                : about.html
FullName            : C:\Udo1\New\about.html
InheritanceEnabled  : False
InheritedFrom       :
AccessControlType   : Allow
AccessRights        : Modify, Synchronize
Account             : stacja\User02
InheritanceFlags    : None
IsInherited         : False
PropagationFlags    : None
AccountType         :
```

Teraz nie panikuj. Prawdopodobnie zastanawiasz się: „Dlaczego User02 ma uprawnienia do modyfikacji, gdy ustawiamy Odczyt na poziomie folderu nadrzędnego?" Pamiętaj, że oryginalna lista ACL zawierała wpis Modyfikuj uprawnienia dla uwierzytelnionych użytkowników; Uprawnienia NTFS kumulują się. W związku z tym, aby rozwiązać ten problem, musimy albo zmodyfikować lub usunąć uwierzytelnionych użytkowników, albo dodać uprawnienia Odmów do User02 (nie jest to uważane za najlepszą praktykę, ponieważ może powodować dalsze problemy z rozwiązywaniem problemów).

Ponadto dane wyjściowe Format-List, które ci przekazałem, wprowadzają inne pytania, takie jak „Dlaczego właściwość Dziedziczona z jest pusta?" Odpowiedź na to pytanie jest taka, że moduł NTFSSecurity jest projektem społeczności open-source. W związku z tym na pewno znajdziesz błędy i inne niespójne zachowanie. Dobrą wiadomością jest to, że możesz swobodnie rozwijać projekt i samodzielnie wprowadzać zmiany!

4. Usuwanie uprawnień NTFS

a) Na koniec usuńmy te dwa nowe wpisy z naszej C:\Udo1 ACL:

```
Remove-NTFSAccess -Path 'C:\Udo1' -Account stacja\User01, stacja\User02 -AccessRights Read -PassThru
```

```
    Path: C:\Udo1 (Inheritance enabled)

Account                          Access Rights    Applies to              Type  IsInherited InheritedFrom
-------                          -------------    ----------              ----  ----------- -------------
stacja\User01                    CreateDirecto... FilesOnly               Allow False       |
stacja\User02                    CreateDirecto... FilesOnly               Allow False
BUILTIN\Administratorzy          FullControl      ThisFolderSubfoldersAn... Allow True
ZARZĄDZANIE NT\SYSTEM            FullControl      ThisFolderSubfoldersAn... Allow True
BUILTIN\Użytkownicy              ReadAndExecut... ThisFolderSubfoldersAn... Allow True
ZARZĄDZANIE NT\Użytkownicy uwier... Modify, Synch... ThisFolderOnly       Allow True
ZARZĄDZANIE NT\Użytkownicy uwier... Delete, Gener... SubfoldersAndFilesOnly Allow True
```

Parametr przełącznika –PassThru jest przydatny, gdy chcesz zobaczyć wyniki potoku, gdy takie dane wyjściowe są zwykle pomijane.

Wniosek: PowerShell jest w stanie szybko tworzyć, modyfikować i usuwać uprawnienia do plików i folderów w systemie plików Windows NTFS. Wielu administratorów systemów polega na skryptach do modyfikowania uprawnień do dużej liczby plików, a PowerShell sprawia, że ten proces jest szybki i łatwy, oszczędzając setki godzin operacji GUI!

Zgłoszenie 3

Posprzątaj

wykonać następujące kroki:

1. **Przejęcie własności folderu**: Uruchom PowerShell jako administrator i użyj polecenia takeown w cmd.exe, aby przejąć własność folderu:

```
takeown /f "C:\udo" /r /d y
```

Opcja /r powoduje rekursywne przejęcie własności, a /d y odpowiada ‘tak’ na wszystkie monity.

2. **Zmiana uprawnień**: Następnie zmień uprawnienia, aby nadać sobie pełny dostęp do folderu:

icacls "C:\udo" /grant "Użytkownicy:(OI)(CI)F" /t

W powyższym poleceniu, “Użytkownicy” to nazwa grupy lub użytkownika, któremu chcesz nadać uprawnienia. (OI) oznacza obiekty dziedziczone, (CI) oznacza kontener dziedziczony, a F oznacza pełny dostęp. Opcja /t stosuje zmiany rekursywnie.

3. **Usunięcie folderu**: Usuń folder/y:

Remove-Item -Path "C:\udo","C:\Udo2","C:\Udo1","C:\Udo0" -Recurse -Force

lub/i

Remove-Item -Path "C:\udo" -Recurse -Force

Zgłoszenie 4

Koniec