Konfiguracja i zarządzanie polityką AppLocker przy użyciu PowerShell'a.

Cmdlety AppLocker zazwyczaj są dostępne w wersjach systemu Windows Server oraz w edycjach Enterprise systemu Windows.

Możesz sprawdzić dostępność cmdletów AppLocker w PowerShell, wykonując następujące kroki:

Uruchom PowerShell z uprawnieniami administratorskimi. W tym celu kliknij prawym przyciskiem myszy ikonę PowerShell na pasku zadań lub w menu Start i wybierz "Uruchom jako administrator".

Po uruchomieniu PowerShell, wykonaj następujące polecenie:

```
Get-Command -Module AppLocker
```

To polecenie wyświetli listę wszystkich cmdletów dostępnych w module AppLocker. Jeśli zwróci listę cmdletów, oznacza to, że moduł AppLocker jest załadowany i dostępny w bieżącej sesji PowerShell.

Jeśli moduł AppLocker nie jest dostępny, możliwe przyczyny mogą obejmować:

Brak odpowiednich uprawnień. Upewnij się, że uruchomiłeś PowerShell jako administrator.

Brak zainstalowanego i skonfigurowanego AppLocker na Twoim systemie operacyjnym. AppLocker jest dostępny w niektórych wersjach systemu Windows, takich jak Windows Server oraz w edycjach Enterprise i Education systemu Windows.

Jeśli cmdlety AppLocker nie są dostępne, a chcesz skorzystać z funkcji zarządzania zasadami, możliwe jest, że AppLocker nie jest skonfigurowany lub nie jest dostępny w Twoim środowisku. W takim przypadku możesz skorzystać z innych narzędzi do zarządzania zasadami, takich jak Group Policy lub inne rozwiązania dostępne w Twoim środowisku.

Przygotowanie

```
New-Item -Path "C:\MDOP\" -ItemType Directory

$Username = "User1"

$Password = ConvertTo-SecureString "zaq1@WSX" -AsPlainText -Force

$PasswordNeverExpires = $true

New-LocalUser -Name $Username -Password $Password -FullName "Pełna Nazwa Użytkownika" -Description "Opis Użytkownika"

Set-LocalUser -Name $Username -PasswordNeverExpires $PasswordNeverExpires

$stream = New-Object System.IO.FileStream('C:\MDOP\test3.pdf', [System.IO.FileMode]::Create)

$writer = New-Object System.IO.BinaryWriter($stream)

$writer.Write((New-Object byte[] 1971500))

$writer.Close()

$stream2 = New-Object System.IO.FileStream('C:\MDOP\test2.pdf', [System.IO.FileMode]::Create)

$writer2 = New-Object System.IO.BinaryWriter($stream2)
```

```
$writer2.Write((New-Object byte[] 1871500))

$writer2.Close()

New-LocalUser -Name "User2" -FullName "Pełna Nazwa Użytkownika" -Description "Opis użytkownika" -Password (ConvertTo-SecureString "zaq1@WSX" -AsPlainText -Force)

New-LocalUser -Name "User3" -FullName "Pełna Nazwa Użytkownika" -Description "Opis Użytkownika" -Password (ConvertTo-SecureString "zaq1@WSX" -AsPlainText -Force)

New-LocalUser -Name "User4" -FullName "Pełna Nazwa Użytkownika" -Description "Opis Użytkownika" -Password (ConvertTo-SecureString "zaq1@WSX" -AsPlainText -Force)


# Ścieżka do pliku źródłowego w języku C#

$sourceFilePath = "C:\MDOP\Aplikacja.cs"

# Zawartość pliku źródłowego w języku C#

$sourceCode = @"

using System;

class Program

{

    static void Main(string[] args)

    {

        Console.WriteLine("ProductName: Nazwa Mojej Aplikacji");

    }

}

"@

# Zapisanie zawartości do pliku źródłowego

$sourceCode | Set-Content -Path $sourceFilePath -Encoding UTF8

# Kompilacja pliku źródłowego do pliku wykonywalnego

$exeFilePath = "C:\MDOP\Aplikacja.exe"

& "C:\Windows\Microsoft.NET\Framework\v4.0.30319\csc.exe" /out:$exeFilePath $sourceFilePath

# Wyświetlenie ścieżki do nowo utworzonego pliku wykonywalnego

$exeFilePath

# Ustaw adresy URL do plików

$urlMsi = "https://isobczak.zsl.gda.pl/sso_l/9 Przygotowanie do pracy serwera telnet i ssh/inne/putty-64bit-0.78-installer.msi"
```

```
$urlExe = "https://isobczak.zsl.gda.pl/sso_l/9 Przygotowanie do pracy serwera telnet i ssh/inne/putty.exe"

# Pobranie plikow

# Określ ścieżki docelowe

$destinationFolder = "C:\MDOP"

$destinationMsi = Join-Path -Path $destinationFolder -ChildPath "putty-64bit-0.78-installer.msi"

$destinationExe = Join-Path -Path $destinationFolder -ChildPath "putty.exe"

# Komunikat informujący o konieczności podania danych uwierzytelniających

Write-Host "Wprowadź dane uwierzytelniające, aby pobrać pliki:"

# Pobierz login

$username = Read-Host -Prompt "Podaj login"

# Pobierz hasło

$password = Read-Host -Prompt "Podaj hasło" -AsSecureString

# Tworzenie obiektu PSCredential

$credential = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList $username, $password

# Pobierz plik MSI

Invoke-WebRequest -Uri $urlMsi -OutFile $destinationMsi -Credential $credential

# Pobierz plik exe

Invoke-WebRequest -Uri $urlExe -OutFile $destinationExe -Credential $credential

Write-Host "Pliki zostały pobrane i zapisane w folderze $destinationFolder."
```

A. Zmiana reguły dotyczącej wykonywania plików dla użytkownika "User1":

1. # Pobierz bieżące zasady AppLocker

```
$existingPolicy = Get-AppLockerPolicy -Local
```

2. # Dodaj użytkownika do zasad AppLocker

```
$existingPolicy.RuleCollections | ForEach-Object {
    $_.UserSids += 'User1'
}
```

3. # Ustaw zaktualizowane zasady

```
Set-AppLockerPolicy -PolicyObject $existingPolicy
```

4. # Pobierz bieżące zasady AppLocker następnie utwórz zasady AppLocker na podstawie tych informacji, wybierz typ reguł, użyj parametrów-RuleType Publisher, Hash , co oznacza, że chcemy utworzyć reguły polityki oparte na wydawcy oraz na sumie kontrolnej (hashu) pliku Parametr -User określa, dla którego użytkownika ma zostać utworzona polityka. Parametr -RuleNamePrefix określa prefiks nazwy reguły. Każda reguła polityki utworzona przez to polecenie będzie miała nazwę rozpoczynającą się od MDOPu1.

Get-AppLockerFileInformation C:\MDOP* | New-AppLockerPolicy -RuleType Publisher, Hash -User User1 -RuleNamePrefix MDOPu1

5. # Pobierz informacje o plikach

$files = Get-AppLockerFileInformation -Path 'C:\MDOP*.pdf'

6. # Utwórz zasady AppLocker z regułami typu Publisher

$policy = $files | New-AppLockerPolicy -RuleType Publisher -User User1 -RuleNamePrefix MDOPeu1

7. # Ustaw zasady

Set-AppLockerPolicy -PolicyObject $policy

8. Polecenie Get-AppLockerFileInformation zbiera informacje o plikach, a polecenie New-AppLockerPolicy tworzy nową politykę AppLocker na podstawie tych informacji, aby zapewnić kontrolę nad plikami znajdującymi się w określonym katalogu dla użytkownika User1. Reguły polityki będą oparte na ścieżkach plików.

Get-AppLockerFileInformation C:\MDOP* | New-AppLockerPolicy -RuleType Path -User User1 -RuleNamePrefix MDOPeu1

9. # Pobierz informacje o plikach

$files = Get-AppLockerFileInformation -Path 'C:\MDOP*.pdf'

10. # Utwórz zasady AppLocker z regułami typu Publisher

$policy = $files | New-AppLockerPolicy:

Polecenie New-AppLockerPolicy tworzy nową politykę AppLocker na podstawie danych dostarczonych przez zmienną $files.

Polecenie | (pipeline) przekazuje dane z zmiennej $files do polecenia New-AppLockerPolicy.

Parametr -RuleType Publisher określa, że reguły polityki będą oparte na wydawcy plików.

Parametr -User User1 określa, dla którego użytkownika ma zostać utworzona polityka. W tym przypadku jest to User1.

Parametr -RuleNamePrefix MDOPeu1 określa prefiks nazwy reguły. Wszystkie reguły polityki utworzone przez to polecenie będą miały nazwę rozpoczynającą się od MDOPeu1.

Całość jest przypisywana do zmiennej $policy, co sugeruje, że wynik operacji, czyli utworzona polityka, jest przechowywany w tej zmiennej dla dalszego wykorzystania w skrypcie.

$policy = $files | New-AppLockerPolicy -RuleType Publisher -User User1 -RuleNamePrefix MDOPeu1

11. # Ustaw zasady

```
Set-AppLockerPolicy -PolicyObject $policy
```

12. # Sprawdź zasady

```
Get-AppLockerFileInformation -Path 'C:\MDOP\*'
```

B. Pobierz SID użytkownika lokalnego, utwórz obiektu zawierający reguły AppLocker w formacie XML, a następnie zapisz ten obiektu do pliku XML na pulpicie.

1 Użyj cmdletu Get-WmiObject do uzyskania SID dla użytkownika lokalnego.

```
$objUser = Get-WmiObject -Class Win32_UserAccount -Filter "Name='User1'"
$objUser.SID
```

Upewnij się, że zastąpisz "UserOrGroupSid" SID użytkownika "S-1-5-21-638117130-2762957075-1857722085-1045".

2 Utwórz nowy obiekt zawierający reguły dotyczące AppLocker:

```
$policy = @"
<?xml version="1.0" encoding="UTF-8"?>
<AppLockerPolicy Version="1">
  <RuleCollection Type="Executable">
    <FilePathRule Id="1" Name="Allow User1 to run all executables" Description="Allow User1 to run all executables" UserOrGroupSid="S-1-5-21-638117130-2762957075-1857722085-1045" Action="Allow" >
      <Conditions>
        <FilePublisherCondition Type="NotEquals" ComparisonValue="*"/>
      </Conditions>
      <Exceptions>
        <FilePublisherCondition Type="Equals" ComparisonValue="CN=Microsoft Corporation, O=Microsoft Corporation, L=Redmond, S=Washington, C=US"/>
      </Exceptions>
    </FilePathRule>
  </RuleCollection>
</AppLockerPolicy>
"@
```

3 Utwórz nowy plik XML na pulpicie i zapisz w nim utworzony obiekt:

$policy | Set-Content "$env:UserProfile\Desktop\AppLockerPolicy1.xml"

$policy | Out-File -FilePath "$env:UserProfile\Desktop\AppLockerPolicy1.xml"

Powyższe polecenie zapisuje zawartość zmiennej $policy do pliku AppLockerPolicy.xml na pulpicie użytkownika bieżącej sesji.

C. Dodanie nowej reguły blokującej wykonywanie plików z konkretnego folderu dla użytkownika "User2":

1. # Pobierz informacje o plikach

$files = Get-AppLockerFileInformation -Path 'C:\MDOP*'

2. # Utwórz zasady AppLocker z regułami typu Publisher

$policy = $files | New-AppLockerPolicy -RuleType Publisher -User User2 -RuleNamePrefix MDOPu1

3. # Ustaw zasady

Set-AppLockerPolicy -PolicyObject $policy

4. # Sprawdź zasady

Get-AppLockerFileInformation -Path 'C:\MDOP*'

Wyjaśnij efekt:

```
PS C:\Users\admin> $policy = $files | New-AppLockerPolicy -RuleType Publisher -User User2 -RuleNamePrefix MDOPu1
New-AppLockerPolicy : Nie można utworzyć reguł. Wymaganych informacji o pliku nie ma w następującym pliku: %OSDRIVE%\MD
OP\APLIKACJA.EXE
At line:1 char:20
+ ...  = $files | New-AppLockerPolicy -RuleType Publisher -User User2 -Rule ...
+                 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [New-AppLockerPolicy], MissingFileInformationException
    + FullyQualifiedErrorId : Microsoft.Security.ApplicationId.PolicyManagement.MissingFileInformationException,Micros
   oft.Security.ApplicationId.PolicyManagement.Cmdlets.NewAppLockerPolicyCmdlet

PS C:\Users\admin> # Ustaw zasady
PS C:\Users\admin> Set-AppLockerPolicy -PolicyObject $policy
Set-AppLockerPolicy : Cannot bind parameter 'PolicyObject'. Cannot convert the "<?xml version="1.0" encoding="UTF-8"?>
<AppLockerPolicy Version="1">
  <RuleCollection Type="Executable">
    <FilePathRule Id="1" Name="Allow User1 to run all executables" Description="Allow User1 to run all executables" Use
rOrGroupSid="S-1-5-21-638117130-2762957075-1857722085-1045" Action="Allow" >
      <Conditions>
        <FilePublisherCondition Type="NotEquals" ComparisonValue="*"/>
      </Conditions>
      <Exceptions>
        <FilePublisherCondition Type="Equals" ComparisonValue="CN=Microsoft Corporation, O=Microsoft Corporation, L=Red
mond, S=Washington, C=US"/>
      </Exceptions>
    </FilePathRule>
  </RuleCollection>
</AppLockerPolicy>" value of type "System.String" to type "Microsoft.Security.ApplicationId.PolicyManagement.PolicyMode
l.AppLockerPolicy".
At line:1 char:35
+ Set-AppLockerPolicy -PolicyObject $policy
+                                   ~~~~~~~
    + CategoryInfo          : InvalidArgument: (:) [Set-AppLockerPolicy], ParameterBindingException
    + FullyQualifiedErrorId : CannotConvertArgumentNoMessage,Microsoft.Security.ApplicationId.PolicyManagement.Cmdlets
   .SetAppLockerPolicyCmdlet
```

D. Utworzenie polityki AppLocker w PowerShell: zdobądź SID, utwórz XML, zapisz na Pulpicie:

1 Użyj cmdletu Get-WmiObject do uzyskania SID dla użytkownika lokalnego.

```
$objUser = Get-WmiObject -Class Win32_UserAccount -Filter "Name='User2'"
$objUser.SID
```

Upewnij się, że zastąpisz "UserOrGroupSid" SID użytkownika "S-1-5-21-638117130-2762957075-1857722085-1046", któremu chcesz zablokować dostęp, do folderu który chcesz zablokować.

2 Utwórz nowy obiekt zawierający reguły dotyczące AppLocker:

```
$policyXml = @"
<AppLockerPolicy Version="1">
  <RuleCollection Type="Executable">
    <FilePathRule Id="1" Name="Allow User2 to run all executables" Description="Allow User2 to run all executables" UserOrGroupSid="S-1-5-21-638117130-2762957075-1857722085-1046" Action="Allow">
      <Conditions>
        <FilePublisherCondition Type="NotEquals" ComparisonValue="*"/>
      </Conditions>
      <Exceptions>
        <FilePublisherCondition Type="Equals" ComparisonValue="CN=Microsoft Corporation, O=Microsoft Corporation, L=Redmond, S=Washington, C=US"/>
      </Exceptions>
    </FilePathRule>
  </RuleCollection>
</AppLockerPolicy>
"@
```

Ten skrypt PowerShell tworzy ciąg tekstowy zawierający definicję polityki AppLocker w formacie XML. Opis jego działania:

1.

$policyXml = @" ... "@:

Ten fragment kodu tworzy ciąg wieloliniowy, który zawiera definicję polityki AppLocker w formacie XML. Definicja ta jest otoczona znacznikami @" i "@, co pozwala na tworzenie ciągów tekstowych obejmujących wiele linii w PowerShell.

2.

<AppLockerPolicy Version="1"> ... </AppLockerPolicy>:

Początek i koniec definicji polityki AppLocker. Wskazuje, że to jest polityka AppLocker w wersji 1.

3.

<RuleCollection Type="Executable"> ... </RuleCollection>:

Określa typ kolekcji reguł. W tym przypadku typ to "Executable", co oznacza, że polityka dotyczy plików wykonywalnych.

4.

<FilePathRule Id="1" Name="Allow User2 to run all executables" Description="Allow User2 to run all executables" UserOrGroupSid="S-1-5-21-638117130-2762957075-1857722085-1048" Action="Allow"> ... </FilePathRule>:

Definicja reguły dla ścieżki pliku. Reguła ta ma identyfikator "1" i jest przeznaczona dla użytkownika o SID "S-1-5-21-638117130-2762957075-1857722085-1048" (w tym przypadku dla "User2"). Reguła ma nazwę "Allow User2 to run all executables" oraz opis "Allow User2 to run all executables". Działanie reguły jest ustawione na "Allow", co oznacza, że użytkownik ma zezwolenie na uruchamianie plików zgodnych z tą regułą.

5.

<Conditions> ... </Conditions>:

Warunki określające, kiedy reguła ma być stosowana. W tym przypadku jest warunek "NotEquals", który oznacza, że reguła będzie miała zastosowanie, gdy plik nie będzie pochodził od określonego wydawcy (w tym przypadku warunek jest ustawiony na "*").

6.

<Exceptions> ... </Exceptions>:

Wyjątki odnoszące się do reguły. W tym przypadku jest wyjątek "Equals", który oznacza, że pliki pochodzące od określonego wydawcy (w tym przypadku Microsoft Corporation) będą wyłączone z działania tej reguły.

Ten skrypt jest przydatny, gdy chcemy zdefiniować politykę AppLocker w formacie XML, aby można ją było łatwo zastosować lub przechowywać w pliku.

3 Utwórz nowy plik XML na pulpicie i zapisz w nim utworzony obiekt:

```
# Ścieżka do pliku XML na pulpicie
$filePath = "$env:UserProfile\Desktop\AppLockerPolicy2.xml"

# Zapis polityki do pliku XML
$policyXml | Set-Content -Path $filePath
```

E. Dodanie nowej reguły zezwalającej na wykonywanie podpisanych cyfrowo plików dla użytkownika "User3":

1.
```
# Pobierz informacje o plikach
$files = Get-AppLockerFileInformation -Path 'C:\MDOP\putty.exe'
```

2. # Utwórz zasady AppLocker z regułami typu Publisher

$policy = $files | New-AppLockerPolicy -RuleType Publisher -User User3 -RuleNamePrefix PublisherName

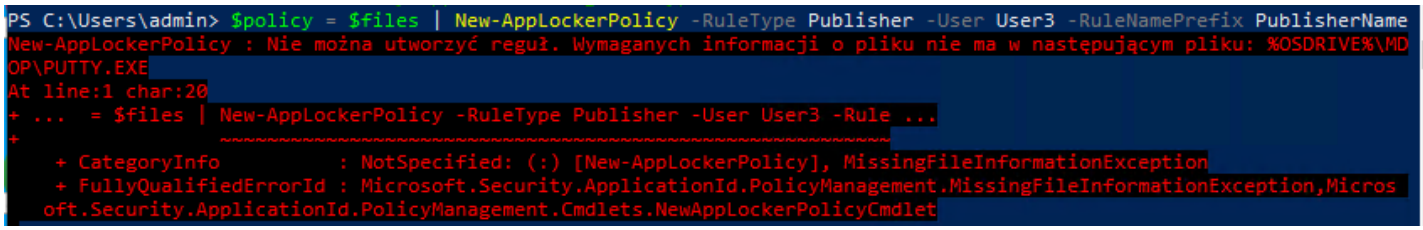
3. # Ustaw zasady

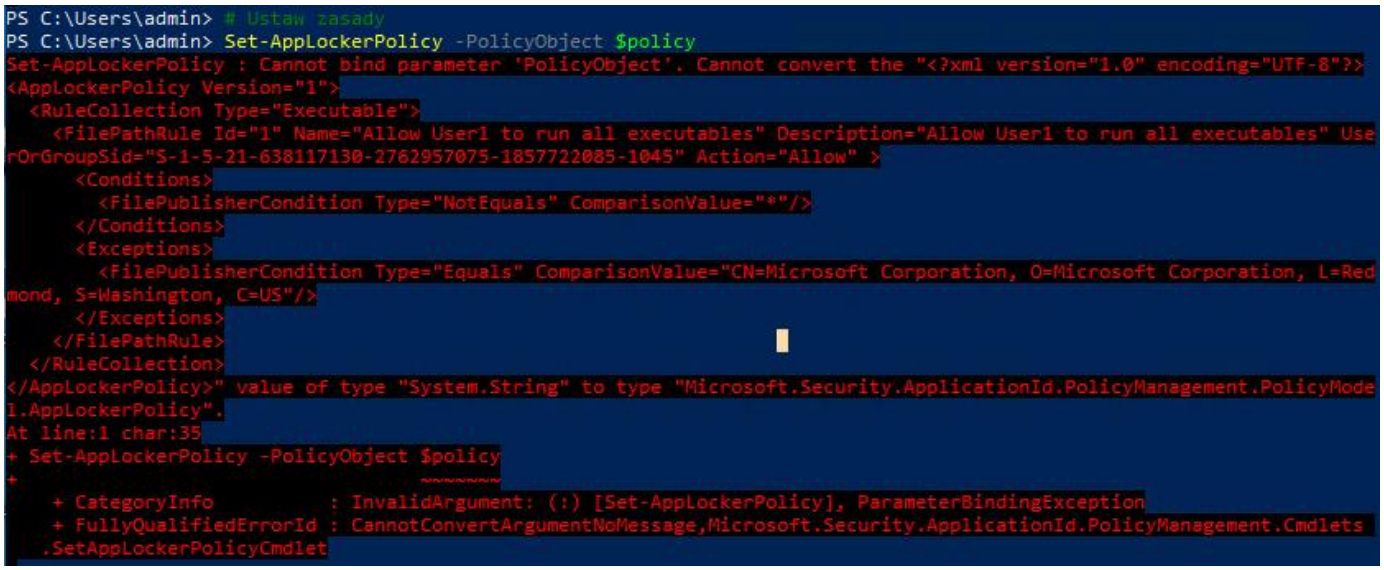
Set-AppLockerPolicy -PolicyObject $policy

4. # Pobierz informacje o plikach

Get-AppLockerFileInformation -Path 'C:\MDOP\putty.exe'

Wyjaśnij efekt:





F. Utwórz nowy obiekt zawierający reguły dotyczące AppLocker, które zezwalają na wykonywanie podpisanych cyfrowo plików dla użytkownika "User3":

1 Użyj cmdletu Get-WmiObject do uzyskania SID dla użytkownika lokalnego.

$objUser = Get-WmiObject -Class Win32_UserAccount -Filter "Name='User3'"

$User3SID = $objUser.SID

2 Utwórz nowy obiekt zawierający reguły dotyczące AppLocker:

Definicja polityki AppLocker w formacie XML

$policyXml = @"

```
<AppLockerPolicy Version="1">
 <RuleCollection Type="Executable">
  <FilePathRule Id="1" Name="Allow digitally signed files for User3" Description="Allow digitally signed files for User3" UserOrGroupSid="$User3SID" Action="Allow">
   <Conditions>
    <FilePublisherCondition Type="Equals" ComparisonValue="*"/>
    <FilePublisherCondition Type="Signed" />
   </Conditions>
  </FilePathRule>
 </RuleCollection>
</AppLockerPolicy>
"@
```

3 Utwórz nowy plik XML na pulpicie i zapisz w nim utworzony obiekt:

```
# Ścieżka do pliku XML na pulpicie
$filePath = "$env:UserProfile\Desktop\AppLockerPolicy_User3.xml"
# Zapis polityki do pliku XML
$policyXml | Set-Content -Path $filePath
```

G. Dodanie nowej reguły blokującej wykonywanie plików z określonym wydawcą dla wszystkich użytkowników:

```
1. # Pobierz informacje o plikach
$files = Get-AppLockerFileInformation -Path 'C:\MDOP\*.pdf'
2. # Utwórz zasady AppLocker z regułami typu Publisher
$policy = $files | New-AppLockerPolicy -RuleType Publisher -User User3 -RuleNamePrefix MkrSoft
3. # Ustaw zasady
Set-AppLockerPolicy -PolicyObject $policy
4. # Ścieżka do pliku XML na pulpicie
$filePath = "$env:UserProfile\Desktop\AppLockerPolicyg.xml"
5. # Zapis polityki do pliku XML
$policyXml | Set-Content -Path $filePath
```

H. Tworzenie i zapisywanie polityki AppLocker blokującej wydawcę MkrSoft.ltd.

1 Utwórz nowy obiekt zawierający reguły dotyczące AppLocker:

```
$policyXml = @"
<?xml version="1.0" encoding="UTF-8"?>
-<AppLockerPolicy Version="1">
-<RuleCollection Type="Publisher">
-<FilePublisherRule Action="Deny" Name="Block MkrSoft.ltd publisher" Id="1">
<Description>Deny MkrSoft.ltd publisher.</Description>
-<Conditions>
<PublisherCondition PublisherName="MkrSoft.ltd"/>
</Conditions>
</FilePublisherRule>
</RuleCollection>
</AppLockerPolicy>
"@
```

2 Utwórz nowy plik XML na pulpicie i zapisz w nim utworzony obiekt:

```
# Ścieżka do pliku XML na pulpicie
$filePath = "$env:UserProfile\Desktop\AppLockerPolicyh.xml"
# Zapis polityki do pliku XML
$policyXml | Set-Content -Path $filePath
```

I. Dodanie nowej reguły zezwalającej na wykonywanie plików z określonymi rozszerzeniami dla użytkownika "User4":

1 Użyj cmdletu Get-WmiObject do uzyskania SID dla użytkownika lokalnego.

```
$objUser = Get-WmiObject -Class Win32_UserAccount -Filter "Name='User4'"
$objUser.SID
```

Wartość $newGuid jest generowana przy użyciu metody NewGuid() z klasy System.Guid. Jest to unikatowy identyfikator (GUID - Globally Unique Identifier), który jest wykorzystywany jako identyfikator reguły w polityce AppLocker.

```
$newGuid = [System.Guid]::NewGuid().ToString()
```

Sprawdźmy, czy możemy uzyskać wersję pliku, którą możemy użyć jako zakres wersji binarnej.

```
$fileVersionInfo = (Get-Command $exeFilePath).FileVersionInfo
```

$fileVersion = $fileVersionInfo.FileVersion

Upewnij się, że zastąpisz "UserOrGroupSid" SID użytkownika " S-1-5-21-638117130-2762957075-1857722085-1048"

2 Utwórz nowy obiekt zawierający reguły dotyczące AppLocker:

Define the AppLocker policy

$policyXml = @"

<AppLockerPolicy Version="1">

<RuleCollection Type="Publisher">

<FilePublisherRule Id="$newGuid" Name="Zezwalaj na pliki exe w folderze MDOP" Description="Zezwalaj na pliki exe w folderze MDOP dla Uytkownika 4." UserOrGroupSid="S-1-5-21-638117130-2762957075-1857722085-1048" Action="Allow">

<Conditions>

<FilePublisherCondition PublisherName="*" ProductName="$productName" BinaryName="$binaryName" BinaryVersionRange="$fileVersion" />

</Conditions>

</FilePublisherRule>

</RuleCollection>

</AppLockerPolicy>

"@

3 Utwórz nowy plik XML na pulpicie i zapisz w nim utworzony obiekt:

Define the file path for the XML policy

$filePath = "$env:UserProfile\Desktop\AppLockerPolicy_User4.xml"

Save the policy to a file

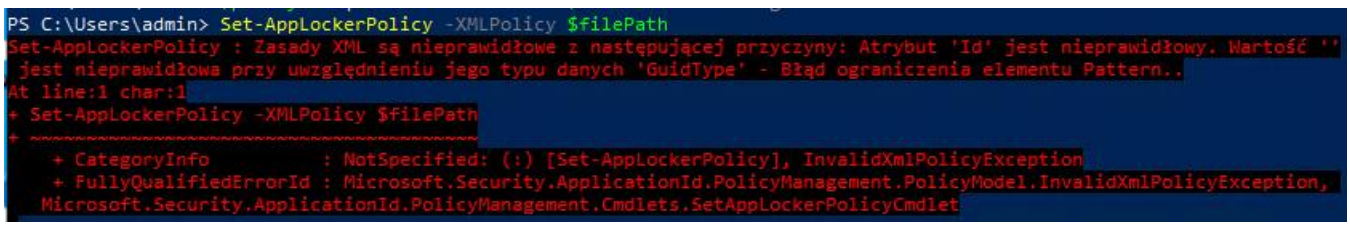
$policyXml | Set-Content -Path $filePath -Encoding UTF8

Nie działa

Set the AppLocker policy from the XML file

Set-AppLockerPolicy -XMLPolicy $filePath

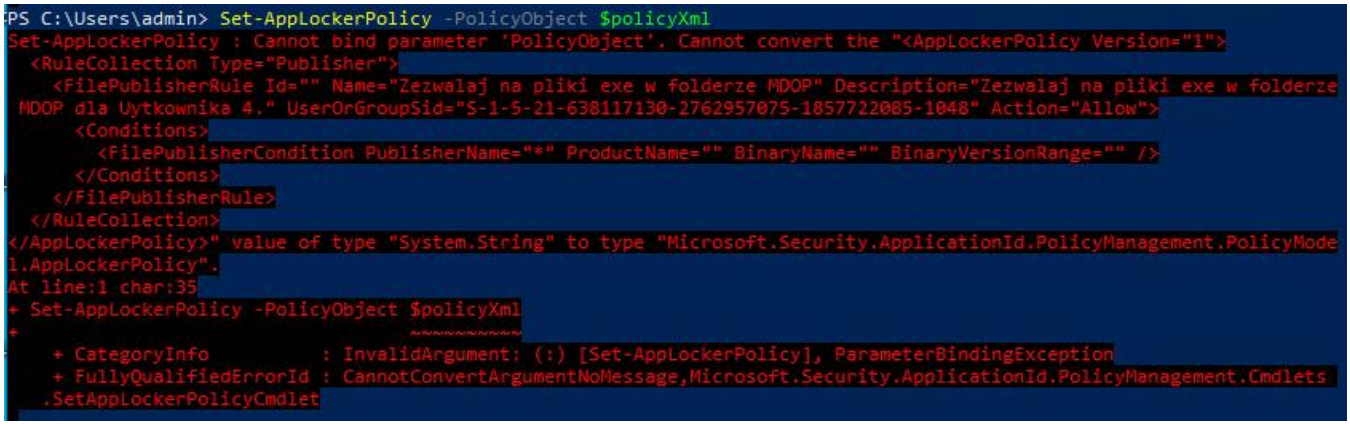
Wyjaśnij efekt:



Ustaw zasady

Set-AppLockerPolicy -PolicyObject $policyXml

Wyjaśnij efekt:



J. Zmiana decyzji polityki dla konkretnego użytkownika i typu reguły:

1 Metoda pierwsza to:

Set-AppLockerPolicy -UserOrGroup "User5" -RuleType MSI -PolicyDecision "Audited"

Set-AppLockerPolicy -RuleType MSI -PolicyDecision "Audited" -FilePath "C:\MDOP\putty-64bit-0.78-installer.msi"

Jeżeli nie zadziała to:

2 Użyj cmdletu Get-WmiObject do uzyskania SID dla użytkownika lokalnego.

$objUser = Get-WmiObject -Class Win32_UserAccount -Filter "Name='User5'"

$objUser.SID

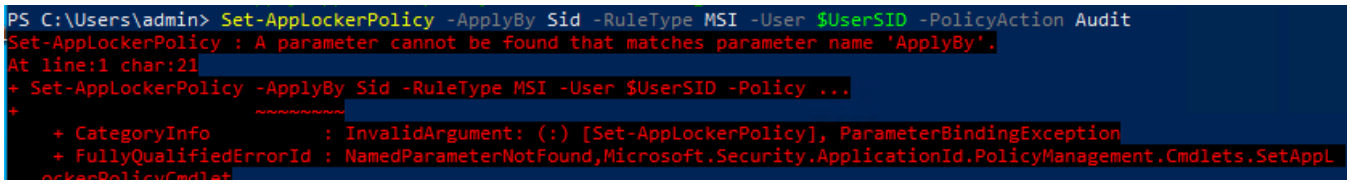
3 Utwórz nowy obiekt zawierający reguły dotyczące AppLocker:

$UserSID = $objUser.SID

Ustaw politykę AppLocker dla użytkownika o określonym SID

Set-AppLockerPolicy -UserOrGroup $UserSID -RuleType MSI -PolicyAction Audit

Set-AppLockerPolicy -ApplyBy Sid -RuleType MSI -User $UserSID -PolicyAction Audit



4 Jeżeli nie działa to żaden z parametrów, których próbowaliśmy użyć (-UserOrGroup, -UserPath, -ApplyBy), nie jest rozpoznawany przez polecenie Set-AppLockerPolicy w Twoim środowisku.

W tej sytuacji spróbuj podejścia alternatywnego. Możemy użyć zasad grupy do stosowania polityk AppLocker dla określonych użytkowników. Oto jak to zrobić:

Otwórz konsolę Zasady zabezpieczeń lokalnych na komputerze.

1. Przejdź do Obiektu zasad grupy (GPO), który chcesz edytować, lub utwórz nowy.
2. Edytuj Obiekt zasad grupy i przejdź do Konfiguracja komputera -> Ustawienia systemu Windows -> Ustawienia zabezpieczeń -> Polityki kontrolowania aplikacji -> AppLocker.
3. Kliknij prawym przyciskiem myszy na AppLocker i wybierz "Utwórz nową regułę..." aby utworzyć nową regułę.
4. Postępuj zgodnie z kreatorami, aby utworzyć regułę AppLocker dla plików MSI i określić użytkownika lub grupę, do której ma być stosowana reguła.
5. Po utworzeniu reguły zamknij Edytor zarządzania zasadami.
6. Pozwól na chwilę na propagację zasad grupy.

Korzystając z zasad grupy do stosowania reguł AppLocker, łatwo możesz skierować konkretne użytkowników lub grupy, bez konieczności bezpośredniego określania SID w poleceniach PowerShell. Ponadto, korzystanie z zasad grupy zapewnia scentralizowane i skalowalne podejście do zarządzania politykami kontroli aplikacji w środowisku Aktywnego Katalogu.

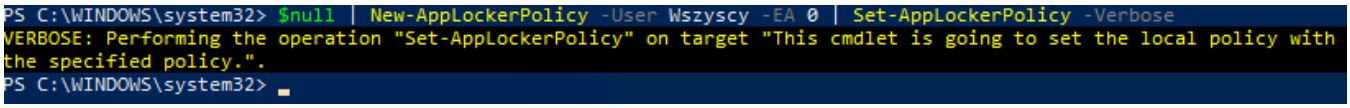
K. Importowanie polityki AppLocker z pliku XML:

```
Set-AppLockerPolicy -XmlPolicy "env:UserProfile\Desktop\AppLockerPolicy4.xml" -Merge
```

Jeżeli nie działa to spróbuj podejścia alternatywnego otwórz konsolę Zasady zabezpieczeń lokalnych na komputerze.

L. Jak wyczyścić lokalną politykę Applocker

```
$null | New-AppLockerPolicy -User Wszyscy -EA 0 | Set-AppLockerPolicy -Verbose
```



EA alias ErrorAction, a 0 oznacza cichą kontynuację.

M. Włączenie szczegółowego logowania dla AppLocker:

```
Set-AppLockerPolicy -UseVerboseLogging
```

Jeżeli nie działa to spróbuj podejścia alternatywnego otwórz konsolę Zasady zabezpieczeń lokalnych na komputerze.

N. Łączenie dwóch polityk AppLocker w jedną:

```
Set-AppLockerPolicy -Merge
"env:UserProfile\Desktop\AppLockerPolicy3.xml","env:UserProfile\Desktop\AppLockerPolicy4.xml"
```

Jeżeli nie działa to spróbuj podejścia alternatywnego otwórz konsolę Zasady zabezpieczeń lokalnych na komputerze.

O. Optymalizacja polityki AppLocker w celu zmniejszenia jej rozmiaru:

Set-AppLockerPolicy -Optimize

Jeżeli nie działa to spróbuj podejścia alternatywnego otwórz konsolę Zasady zabezpieczeń lokalnych na komputerze.

P. Uruchomienie trybu audytowania dla polityki AppLocker:

Set-AppLockerPolicy -PolicyDecision "Audited"

Jeżeli nie działa to spróbuj podejścia alternatywnego otwórz konsolę Zasady zabezpieczeń lokalnych na komputerze.

Warto zauważyć, że nie wszystkie parametry są wymagane w każdym przypadku i zależą od rodzaju modyfikacji, jaką chcemy wprowadzić w polityce.