35.2 Zabezpieczenie oprogramowanie stosując AppLocker w PowerShell

1. Opracuj wykaz kolejnych czynności, aby zabezpieczyć oprogramowanie używając funkcji AppLocker w PowerShell.
2. Przedstaw opracowany wykaz kolejnych czynności, aby zabezpieczyć oprogramowanie używając funkcji AppLocker w PowerShell.
3. Wykonaj notatkę zawierająca wykaz kolejnych czynności, aby zabezpieczyć oprogramowanie używając funkcji AppLocker w PowerShell.

Co to jest funkcja AppLocker?

AppLocker to biała i czarna lista aplikacji wbudowana w system Windows. Pozwala na pisanie reguł w zasadach grupy, dla których aplikacje, skrypty i instalatory Windows mogą być uruchamiane (a które nie), które są wymuszane na komputerze klienckim przez usługę identyfikacji aplikacji (AppIDSvc).

Planowanie

Najpierw musisz zdecydować, co chcesz osiągnąć, wdrażając funkcję AppLocker. Jest to ważne, ponieważ określi, w jaki sposób zamierzasz napisać swoje reguły funkcji AppLocker. W mojej sytuacji chciałem zablokować uruchamianie złośliwego oprogramowania w profilach użytkowników, a także uniemożliwić instalowanie lub uruchamianie nieautoryzowanego oprogramowania z nośnika USB. Istnieją dwa sposoby wdrażania reguł: umieszczanie na czarnej liście i umieszczanie na białej liście.

Czarna lista

Czarna lista w funkcji AppLocker pozwala na wszystko, ale blokuje określone aplikacje, skrypty i instalatory systemu Windows, których nie chcesz dopuścić na swoich komputerach. (Microsoft niedawno opublikował białą księgę, w jaki sposób Microsoft IT zrobił to wewnętrznie. Ta metoda najprawdopodobniej spowoduje najmniej bólów głowy, jeśli dokładnie wiesz, co chcesz zablokować. Minusem jest to, że będziesz musiał wygenerować listę tego, co chcesz blokować i aktualizować listę. Ta metoda jest również łatwiejsza do obejścia, jeśli używasz ścieżek plików do identyfikowania aplikacji lub skrótów plików, które nie zawierają wszystkich wersji aplikacji.

Biała lista

Biała lista w AppLocker pozwala odmówić wszystkiego oprócz określonych aplikacji, skryptów i instalatorów systemu Windows, na które chcesz zezwolić. Wszystko, co nie znajduje się na Twojej liście, zostanie zablokowane. Ta metoda będzie wymagała znacznie więcej pracy z góry, aby upewnić się, że czegoś przypadkowo nie zablokujesz, ale na dłuższą metę zatrzyma uruchamianie więcej nieautoryzowanych aplikacji.

Praca detektywa

Teraz, gdy już zdecydowałeś, w jaki sposób chcesz wdrożyć funkcję AppLocker, musisz zidentyfikować pliki wykonywalne, na które będziesz musiał zezwolić lub odmówić.

Przygotowanie

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

New-LocalUser "User1" -Password $Password

New-LocalUser "User2" -Password $Password

New-LocalUser "User3" -Password $Password

New-LocalUser "User4" -Password $Password

New-LocalUser "User5" -Password $Password

$stream = New-Object System.IO.FileStream('C:\Users\admin\Desktop\file.exe',
[System.IO.FileMode]::Create)

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

$writer.Write((New-Object byte[] 209715200))

$writer.Close()

# Tworzenie folderu "MDOP", jeśli nie istnieje

New-Item -ItemType Directory -Force -Path 'C:\MDOP'

$stream = New-Object System.IO.FileStream('C:\MDOP\test1.exe', [System.IO.FileMode]::Create)

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

$writer.Write((New-Object byte[] 109715000))

$writer.Close()

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

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

$writer.Write((New-Object byte[] 19715000))

$writer.Close()

Get-AppLockerFileInformation C:\Users\admin\Desktop\*.exe | New-AppLockerPolicy -RuleType
Publisher, Hash -User admin -RuleNamePrefix DesktopEXE

Get-AppLockerPolicy -Effective | Test-AppLockerPolicy -Path C:\Users\admin\Desktop\*.exe -User
admin

Get-AppLockerPolicy -Effective | Test-AppLockerPolicy -Path C:\MDOP\*.exe -User admin
```

A. Zarządzanie funkcją AppLocker za pomocą PowerShell

Wyobraź sobie, że nigdy nie musisz usuwać wirusa z komputera ani bez wysiłku blokować aplikacji na podstawie wersji, wydawcy lub jakiegokolwiek innego ważnego atrybutu. AppLocker umożliwia to zarządzanie. PowerShell ułatwia zarządzanie AppLockerem.

Moduł AppLocker dla programu PowerShell zawiera pięć poleceń cmdlet. Nie daj się zwieść małej liczbie poleceń! Z wyjątkiem polecenia usuwania, są one więcej niż wystarczające do obsłużenia całego cyklu życia polityki. Aby rozpocząć eksplorację PowerShell, otwórz PowerShell ISE i wpisz Get-Command -Module AppLocker

Polecenie PowerShell "Get-Command -Module AppLocker" służy do wyświetlenia listy wszystkich dostępnych poleceń związanych z modułem AppLocker w bieżącej sesji PowerShell.

```
PS C:\WINDOWS\system32> Get-Command -Module AppLocker

CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Cmdlet          Get-AppLockerFileInformation                       2.0.0.0    AppLocker
Cmdlet          Get-AppLockerPolicy                                2.0.0.0    AppLocker
Cmdlet          New-AppLockerPolicy                                2.0.0.0    AppLocker
Cmdlet          Set-AppLockerPolicy                                2.0.0.0    AppLocker
Cmdlet          Test-AppLockerPolicy                               2.0.0.0    AppLocker
```

Jak pokazano powyżej, obecnie dostępnych jest 5 różnych poleceń cmdlet PowerShell do interakcji z funkcją AppLocker, wyjaśnimy na przykładach każde z nich.

1. Get-AppLockerFileInformation:

a) to polecenie cmdlet pobiera informacje o plikach dla wszystkich plików exe i skryptów w obszarze %windir%\system32.

Get-AppLockerFileInformation -Directory C:\WINDOWS\system32\ -Recurse -FileType exe, script

```
PS C:\WINDOWS\system32> Get-AppLockerFileInformation -Directory C:\Windows\system32\ -Recurse -FileType exe, script
WARNING: The following directories cannot be searched: C:\Windows\system32\LogFiles\WMI\RtBackup,
C:\Windows\system32\config\systemprofile\AppData\Local\Microsoft\Windows\INetCache\Content.IE5

Path                                    Publisher
----                                    ---------
%SYSTEM32%\CHCP.COM                     O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US\MICROSOFT® WINDO...
%SYSTEM32%\FORMAT.COM                   O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US\MICROSOFT® WINDO...
%SYSTEM32%\MODE.COM                     O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US\MICROSOFT® WINDO...
```

b) to polecenie cmdlet pobiera informacje o pliku dla pliku określonego przez ścieżkę.

Get-AppLockerFileInformation -Path "C:\Program Files (x86)\Internet Explorer\iexplore.exe" | Format-List

```
PS C:\WINDOWS\system32> Get-AppLockerFileInformation -Path "C:\Program Files (x86)\Internet Explorer\iexplore.exe" | Format-List


Path      : %PROGRAMFILES%\INTERNET EXPLORER\IEXPLORE.EXE
Publisher : O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US\INTERNET EXPLORER\IEXPLORE.EXE,11.0.19041.1
Hash      : SHA256 0x71145DE5C265B21C9BCC66DA8D21F608BA64D3343A44FD49CFB9E4F9606B050B
AppX      : False
```

Get-AppLockerFileInformation -Path "C:\Program Files\Internet Explorer\iexplore.exe" | Format-List

```
PS C:\WINDOWS\system32> Get-AppLockerFileInformation -Path "C:\Program Files\Internet Explorer\iexplore.exe" | Format-List


Path      : %PROGRAMFILES%\INTERNET EXPLORER\IEXPLORE.EXE
Publisher : O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US\INTERNET EXPLORER\IEXPLORE.EXE,11.0.19041.1
Hash      : SHA256 0x49AF5EF7E46F9C7CBD9969222611AF8E088331268C74734149B72BB62C2E5AB7
AppX      : False
```

Odpowiedz w zeszycie na pytanie na podstawie efektu powyżej. Czy to są różne pliki czy kopia tego samego? Odpowiedz uzasadnij.

c) to polecenie cmdlet wyświetlane są informacje o plikach dla wszystkich spakowanych aplikacji zainstalowanych na tym komputerze dla wszystkich użytkowników.

Get-AppXPackage -AllUsers | Get-AppLockerFileInformation

```
PS C:\WINDOWS\system32> Get-AppXPackage -AllUsers | Get-AppLockerFileInformation

Path                                                                                Publisher
----                                                                                ---------
windows.immersivecontrolpanel_10.0.2.1000_neutral_neutral_cw5n1h2txyewy.appx        CN=Microsoft Windows, O=Microsoft ...
Windows.PrintDialog_6.2.1.0_neutral_neutral_cw5n1h2txyewy.appx                      CN=Microsoft Windows, O=Microsoft ...
```

d) to polecenie cmdlet wyświetlane są informacje o pliku dla wszystkich kontrolowanych zdarzeń w lokalnym dzienniku zdarzeń. Audytowane zdarzenia odpowiadają zdarzeniu Ostrzeżenie w dzienniku audytu funkcji AppLocker.

Get-AppLockerFileInformation -EventLog -EventType Audited

```
PS C:\WINDOWS\system32> Get-AppLockerFileInformation -EventLog -EventType Audited
PS C:\WINDOWS\system32>
```

e) to polecenie cmdlet wyświetla statystyki dla wszystkich dozwolonych zdarzeń w lokalnym dzienniku zdarzeń. Dla każdego pliku w dzienniku zdarzeń polecenie cmdlet zsumuje liczbę wystąpień danego typu zdarzenia

Get-AppLockerFileInformation -EventLog -EventType Allow -Statistics

```
PS C:\WINDOWS\system32> Get-AppLockerFileInformation -EventLog -EventType Allow -Statistics
PS C:\WINDOWS\system32>
```

f) to polecenie cmdlet pobiera wymagane informacje potrzebne do utworzenia reguł funkcji AppLocker z listy plików. W poniższym przykładzie widzimy ścieżkę, wydawcę i skrót pliku .exe w określonej lokalizacji.

Get-AppLockerFileInformation -Path C:\Users\admin\Desktop*.exe | Format-List

```
PS C:\WINDOWS\system32> Get-AppLockerFileInformation -Path C:\Users\admin\Desktop\*.exe |Format-List


Path      : %OSDRIVE%\USERS\ADMIN\DESKTOP\FILE.EXE
Publisher :
Hash      : SHA256 0xE5B844CC57F57094EA4585E235F36C78C1CD222262BB89D53C94DCB4D6B3E55D
AppX      : False
```

2. Get-AppLockerPolicy:

a) to polecenie cmdlet służy do pobierania lokalnych lub domenowych zasad funkcji AppLocker. Możemy określić politykę lokalną z opcją -Local, politykę domeny z opcją -Domain, po której następuje ścieżka LDAP do polityki lub -Effective, aby wyświetlić obowiązującą i zastosowaną politykę. Możemy również określić -Xml, aby wyprowadzić wyniki jako wartość XML. Poniższy przykład pokazuje to, co pozwala nam zobaczyć, co robi nasza zasada AppLocker za pośrednictwem PowerShell.

Get-AppLockerPolicy -local

Get-AppLockerPolicy -Effective -Xml

```
PS C:\WINDOWS\system32> Get-AppLockerPolicy -local

Version RuleCollections RuleCollectionTypes
------- --------------- -------------------
      1 {}              {}


PS C:\WINDOWS\system32> Get-AppLockerPolicy -Effective -Xml
<AppLockerPolicy Version="1" />
```

b) to polecenie cmdlet pobiera skuteczną politykę na komputer, a następnie wysyła ją w formacie XML do określonego pliku w istniejącej ścieżce.

Get-AppLockerPolicy -Effective -Xml | Set-Content ('c:\curr.xml')

```
PS C:\WINDOWS\system32> Get-AppLockerPolicy -Effective -Xml | Set-Content ('c:\curr.xml')
```

3. New-AppLockerPolicy:

a) to polecenie cmdlet tworzy nową zasadę funkcji AppLocker na podstawie listy określonych informacji. Informacje można wyświetlić, uruchamiając Get-AppLockerFileInformation.

Get-AppLockerFileInformation C:\Users\admin\Desktop*.exe | New-AppLockerPolicy -RuleType Publisher, Hash -User Wszyscy -RuleNamePrefix DesktopEXE

```
PS C:\WINDOWS\system32> Get-AppLockerFileInformation C:\Users\admin\Desktop\*.exe | New-AppLockerPolicy -RuleType Publisher, Hash -User Wszyscy -RuleNamePrefix DesktopEXE

Version RuleCollections RuleCollectionTypes
------- --------------- -------------------
      1 {0}             {Exe}
```

b) to polecenie cmdlet tworzy zasady funkcji AppLocker, które zawierają reguły zezwalające dla wszystkich plików wykonywalnych w C:\Windows\System32. Zasady zawierają reguły wydawcy dla tych plików, które zawierają informacje o wydawcy, oraz reguły mieszania dla tych, które ich nie zawierają. Reguły są poprzedzone prefiksem System32: i mają zastosowanie do grupy Wszyscy.

Get-ChildItem C:\Windows\System32*.exe | Get-AppLockerFileInformation | New-AppLockerPolicy -RuleType Publisher, Hash -User Wszyscy -RuleNamePrefix System32

```
PS C:\WINDOWS\system32> Get-ChildItem C:\Windows\System32\*.exe | Get-AppLockerFileInformation | New-AppLockerPolicy -Ru
leType Publisher, Hash -User Wszyscy -RuleNamePrefix System32

Version RuleCollections RuleCollectionTypes
------- --------------- -------------------
      1 {0}             {Exe}
```

c) to polecenie cmdlet tworzy zasady funkcji AppLocker w formacie XML dla wszystkich plików wykonywalnych w programie C:\Windows\System32. Zasady zawierają tylko reguły ścieżek. Reguły są stosowane do grupy Wszyscy. Parametr Optimize wskazuje, że podobne reguły są zgrupowane tam, gdzie to możliwe. Zasady funkcji AppLocker ufają wszystkim lokalnym składnikom systemu Windows.

Get-ChildItem C:\Windows\System32*.exe | Get-AppLockerFileInformation | New-AppLockerPolicy -AllowWindows -RuleType Path -User Wszyscy -Optimize -XML

```
PS C:\WINDOWS\system32> Get-ChildItem C:\Windows\System32\*.exe | Get-AppLockerFileInformation | New-AppLockerPolicy -Al
lowWindows -RuleType Path -User Wszyscy -Optimize -XML
<AppLockerPolicy Version="1"><RuleCollection Type="Exe" EnforcementMode="NotConfigured"><FilePathRule Id="7340f615-34ae-
42d5-8f3a-34255afa9999" Name="%SYSTEM32%\*" Description="" UserOrGroupSid="S-1-1-0" Action="Allow"><Conditions><FilePath
Condition Path="%SYSTEM32%\*" /></Conditions></FilePathRule><RuleCollectionExtensions><ThresholdExtensions /><RedstoneEx
tensions><SystemApps Allow="Enabled" /></RedstoneExtensions></RuleCollectionExtensions></RuleCollection></AppLockerPolic
y>
```

4. Set-AppLockerPolicy:

a) to polecenie cmdlet ustawia zasady AppLocker dla określonego obiektu zasad grupy. Wcześniej zauważyliśmy, jak utworzyć plik XML za pomocą polecenia cmdlet Get-AppLockerPolicy, jest to rodzaj pliku, który można wysłać do Set-AppLockerPolicy.

Set-AppLockerPolicy -XMLPolicy C:\curr.xml

```
PS C:\WINDOWS\system32> Set-AppLockerPolicy -XMLPolicy C:\curr.xml
```

5. Test-AppLockerPolicy:

a) to polecenie cmdlet służy do określania, czy określony użytkownik lub grupa użytkowników będzie w stanie wykonać akcję na podstawie zasad, co zasadniczo umożliwia nam przetestowanie zasad funkcji AppLocker. Na przykład w poniższym przykładzie możemy sprawdzić, czy skuteczne polityki zastosowane na tym komputerze umożliwiają każdemu użytkownikowi dostęp do plików .exe na pulpicie konta admin, w którym to przypadku jest to domyślnie zabronione.

Get-AppLockerPolicy -Effective | Test-AppLockerPolicy -Path C:\Users\admin\Desktop*.exe -User Wszyscy

```
PS C:\WINDOWS\system32> Get-AppLockerPolicy -Effective | Test-AppLockerPolicy -Path C:\Users\admin\Desktop\*.exe -User Wszyscy

FilePath                        PolicyDecision MatchingRule
--------                        -------------- ------------
C:\Users\admin\Desktop\file.exe AllowedByDefault
```

Pamiętaj, że jeśli zapomnisz, co robią te polecenia cmdlet, zawsze możesz użyć polecenia cmdlet „Get-Help", aby wyświetlić przydatne informacje. Możemy również dodać -Full, -Examples lub -Detailed, aby uzyskać jeszcze więcej dokumentacji.

B. Ścieżka, wydawca lub skrót

AppLocker może zezwalać lub blokować aplikacje na podstawie trzech typów kryteriów. Pliki można pogrupować według ścieżki, wydawcy lub skrótu. W Windows AppLocker, poszczególne reguły powinny być budowane w tej kolejności następujących powodów:

- Wydawca: zużywa najmniej pracy administracyjnej i jest najbardziej elastyczny.
- Hash: bezpieczniejszy niż reguła ścieżki, nieelastyczny podczas aktualizacji programu.
- Ścieżka: najmniej pożądana. Lokalizacje ścieżek nie powinny umożliwiać standardowym użytkownikom dostępu do zapisu.

1. Korzystając z Get-AppLockerFileInformation, możemy skanować pliki lub katalogi, aby zobaczyć, jakie typy reguł będą obsługiwane. To polecenie będzie rekurencyjnie przeszukiwać katalog:

Get-AppLockerFileInformation -Directory "C:\Users" -Recurse

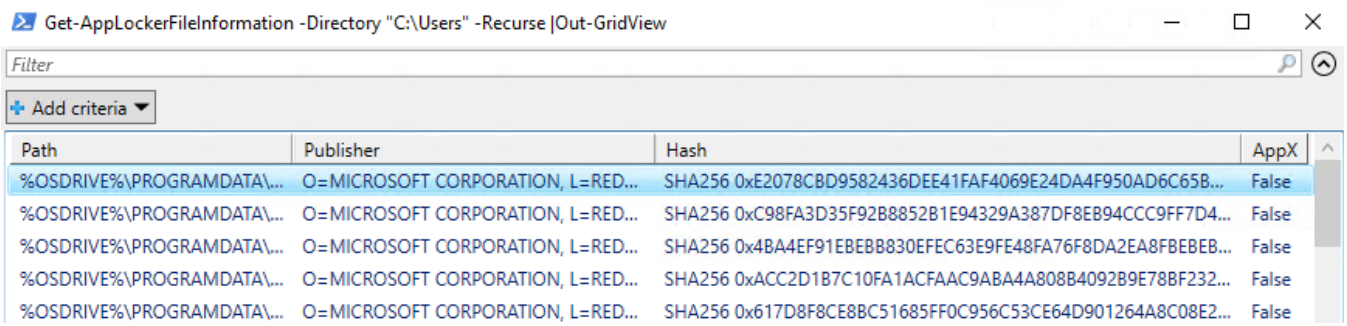
```
PS C:\WINDOWS\system32> Get-AppLockerFileInformation -Directory "C:\Users" -Recurse
WARNING: The following directories cannot be searched: C:\Users\Public\Documents\Moje wideo, C:\Users\Public\Documents\Moje
obrazy, C:\Users\Public\Documents\Moja muzyka, C:\Users\Default User, C:\Users\Default\Ustawienia lokalne,
C:\Users\Default\Szablony, C:\Users\Default\SendTo, C:\Users\Default\Recent, C:\Users\Default\PrintHood,
%OSDRIVE%\USERS\ADMIN\APPDATA\LOCAL\MICROSOFT\ONEDRIVE\20.169.0823.0006\AMD64\FILESYNCSHELL64... O=MICROSOFT CORPORATION, L=RE...


PS C:\WINDOWS\system32>
```

2. Chociaż część ścieżki katalogu została usunięta, nadal możesz zobaczyć, jak przydatne jest to polecenie cmdlet do planowania. Aby ułatwić sortowanie, możemy przekazać dowolne dane wyjściowe do polecenia cmdlet Out-GridView.

Get-AppLockerFileInformation -Directory "C:\Users" -Recurse |Out-GridView

```
PS C:\WINDOWS\system32> Get-AppLockerFileInformation -Directory "C:\Users" -Recurse |Out-GridView
```



W prawym górnym rogu zwróć uwagę na możliwość filtrowania wyników!

C. Tworzenie i testowanie zasad AppLocker

Możemy szybko tworzyć reguły, używając Get-AppLockerFileInformation i przesyłając dane wyjściowe do New-AppLockerPolicy.

1. Polecenie przeszukuje katalog "C:\Program Files" (oraz podkatalogi) i zwraca informacje o plikach zgodnych z zasadami AppLocker następnie tworzy nową zasadę AppLocker z zasadami "Publisher" i "Hash" i zastosowana do grupy użytkowników "Wszyscy". Wszystkie reguły zostaną nazwane zgodnie z prefiksem "Programy" i przekierowany do polecenia Out-File C:\Programy.xml, które zapisuje wynik do pliku "C:\Programy.xml".

```
Get-AppLockerFileInformation -Directory "C:\Program Files" -Recurse | New-AppLockerPolicy -RuleType Publisher,Hash -User Wszyscy -RuleNamePrefix Programy -XML | Out-File C:\Programy.xml
```

```
PS C:\WINDOWS\system32> Get-AppLockerFileInformation -Directory "C:\Program Files" -Recurse | New-AppLockerPolicy -RuleType Publis
her,Hash -User Wszyscy -RuleNamePrefix Programy -XML | Out-File C:\Programy.xml
WARNING: The following directories cannot be searched: C:\Program Files\Windows NT\Akcesoria, C:\Program Files\Windows Defender
Advanced Threat Protection\Classification\Configuration
```

2. Przed zastosowaniem naszych reguł funkcji AppLocker na komputerze (lub w obiekcie zasad grupy), najpierw będziemy chcieli je przetestować. Testowanie pozwala nam poprawić błędy, zanim przypadkowo zablokujemy potrzebny plik. Testowanie można wykonać, uruchamiając Test-AppLockerPolicy na określonych plikach. W poniższym przykładzie testuję plik Programy.xml z plikiem C:\Program Files\Windows Defender\MpAsDesc.dll.
W powyższym przykładzie Test-AppLockerPolicy jest używane do przetestowania polityki zdefiniowanej w pliku C:\Programy.xml na pliku MpAsDesc.dll znajdującym się w katalogu C:\Program Files\Windows Defender. Jeśli plik MpAsDesc.dll jest dozwolony na podstawie polityki, to zostanie wyświetlona informacja o zgodności z polityką AppLocker, a jeśli plik jest niedozwolony, to zostanie wyświetlona informacja o niezgodności.
Wynik testu może pomóc w diagnozowaniu problemów związanych z działaniem polityki AppLocker i w wykryciu nieautoryzowanych aplikacji i skryptów w systemie.

```
Test-AppLockerPolicy -XMLPolicy C:\Programy.xml -Path "C:\Program Files\Windows Defender\MpAsDesc.dll"
```

```
PS C:\WINDOWS\system32> Test-AppLockerPolicy -XMLPolicy C:\Programy.xml -Path "C:\Program Files\Windows Defender\MpAsDesc.dll"

FilePath                                    PolicyDecision MatchingRule
--------                                    -------------- ------------
C:\Program Files\Windows Defender\MpAsDesc.dll       Allowed Programy: MPASDESC.DLL wersja dokładnie 4.18.1907.16384 w MICROS...
```

D. Ustawianie reguł AppLocker

Generowanie pliku XML nie powoduje zastosowania naszych reguł funkcji AppLocker.
Aby zastosować tę zasadę, możemy zaimportować reguły do lokalnego zestawu reguł funkcji AppLocker lub zaimportować reguły do określonego obiektu zasad grupy.
Obie te metody są obsługiwane za pomocą polecenia cmdlet Set-AppLockerPolicy.

1. Aby zastosować reguły Programy.xml lokalnie, uruchom:

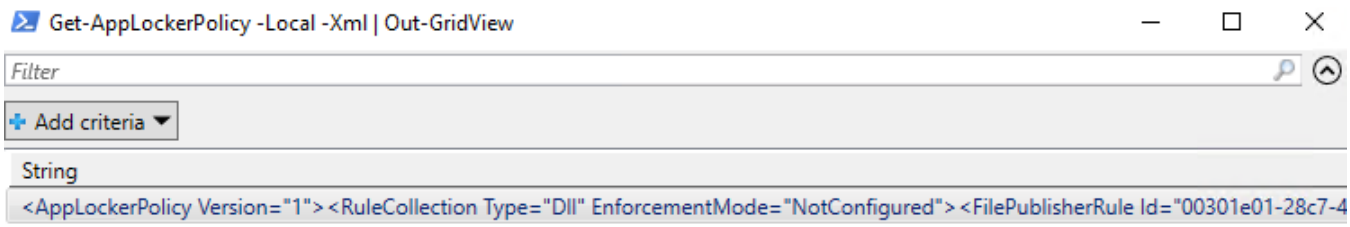
Set-AppLockerPolicy -XMLPolicy C:\Programy.xml

```
PS C:\WINDOWS\system32> Set-AppLockerPolicy -XMLPolicy C:\Programy.xml
```

2. Zweryfikuj, czy nasza zasada została zaimportowana, uruchamiając:

Get-AppLockerPolicy -Local -Xml | Out-GridView

```
PS C:\WINDOWS\system32> Get-AppLockerPolicy -Local -Xml | Out-GridView
```



Filtruj za pomocą polecenia cmdlet Out-GridView

E. Importowanie reguł do obiektu zasad grupy

1. Importuj zasady funkcji AppLocker z określonego pliku XML. Zastąp istniejące zasady, dodaj parametr -Merge.

Set-AppLockerPolicy -XMLPolicy C:\Programy.xml -Merge

```
PS C:\WINDOWS\system32> Set-AppLockerPolicy -XMLPolicy C:\Programy.xml -Merge
PS C:\WINDOWS\system32>
```

2. Sprawdź informacje o plikach zgodnych z zasadami AppLocker

Get-AppLockerFileInformation -Directory "C:\Program Files" -Recurse

3. Podaj wnioski.