Execution Policy w PowerShell – Przegląd i Przykłady

Dostępne Execution Policy i ich efekty

Restricted

❌ Domyślna polityka – nie pozwala uruchamiać żadnych skryptów. Tylko komendy interaktywne.

AllSigned

✅ Tylko skrypty podpisane cyfrowo przez zaufanego wydawcę mogą być uruchamiane. Wymaga podpisu także lokalnych skryptów.

RemoteSigned

✅ Skrypty lokalne mogą być uruchamiane bez podpisu. Skrypty pobrane z internetu muszą być podpisane lub odblokowane.

Unrestricted

✅ Umożliwia uruchamianie wszystkich skryptów. Wyświetla ostrzeżenie przy uruchamianiu skryptów z internetu (Zone.Identifier).

Bypass

✅ Brak jakichkolwiek ostrzeżeń czy blokad. Wszystkie skrypty uruchamiają się natychmiast – używane głównie w automatyzacji.

Undefined

⚙️ Brak jawnie ustawionej polityki na danym poziomie. PowerShell dziedziczy z wyższego poziomu lub stosuje domyślne ustawienia (Restricted).

RemoteSigned – jak PowerShell rozpoznaje pochodzenie skryptu

PowerShell używa alternatywnego strumienia danych (ADS) o nazwie Zone.Identifier, aby określić, czy plik pochodzi z Internetu. Gdy plik jest pobierany (np. z przeglądarki), Windows dodaje do niego ukryty strumień danych:

```
[ZoneTransfer]
ZoneId=3
```

ZoneId=3 oznacza Internet, ZoneId=2 – Intranet, ZoneId=0 – komputer lokalny.

Możesz sprawdzić strumień danych:

```
Get-Content .\skrypt.ps1 -Stream Zone.Identifier
```

Możesz usunąć tę informację:

```
Remove-Item -Path .\skrypt.ps1 -Stream Zone.Identifier
```

Albo użyć wygodnej komendy:

```
Unblock-File .\skrypt.ps1
```

Unrestricted vs Bypass – różnice

Unrestricted: pozwala uruchamiać wszystkie skrypty, ale ostrzega przy uruchamianiu skryptów z internetu.
Bypass: całkowicie ignoruje polityki i ostrzeżenia. Używany do automatyzacji (np. CI/CD).

Przykład użycia Unrestricted:

```
Set-ExecutionPolicy Unrestricted -Scope Process
```

Przykład użycia Bypass:

```
Set-ExecutionPolicy Bypass -Scope Process
```

Undefined – efekt praktyczny

Jeśli Execution Policy na danym poziomie jest ustawiona jako Undefined, PowerShell dziedziczy ustawienie z wyższego poziomu. Można to sprawdzić komendą:

```
Get-ExecutionPolicy -List
```

Jeśli wszystkie poziomy są Undefined, stosowana jest domyślna polityka Restricted.

Hierarchia poziomów:
1. Process
2. CurrentUser
3. LocalMachine
4. Group Policy (MachinePolicy, UserPolicy)

Jeśli żadna polityka nie jest ustawiona, PowerShell domyślnie użyje Restricted.

Praktyczne przykłady sprawdzania Zone.Identifier

Aby sprawdzić, czy plik posiada oznaczenie strefy (ZoneId), użyj:

```
Get-Content -Path .\skrypt.ps1 -Stream Zone.Identifier
```

Jeśli plik nie ma strumienia Zone.Identifier, pojawi się błąd FileNotFoundException – oznacza to, że plik nie pochodzi z Internetu lub nie został oznaczony jako taki.

Aby sprawdzić wszystkie strumienie pliku:

```
Get-Item -Path .\skrypt.ps1 | Get-Item -Stream *
```

Jeśli Zone.Identifier nie istnieje – strumień nie zostanie wyświetlony.

Pliki pobrane za pomocą Invoke-WebRequest, curl, wget itp. NIE otrzymują strumienia Zone.Identifier, ponieważ te narzędzia nie korzystają z mechanizmu Attachment Execution Service (AES).

Aby zasymulować plik z pochodzeniem z Internetu (ZoneId=3), dodaj ręcznie strumień:

```
Set-Content -Path "D:\az\test.ps1" -Stream Zone.Identifier -Value "[ZoneTransfer]`nZoneId=3"
```

Następnie sprawdź go:

```
Get-Content -Path "D:\az\test.ps1" -Stream Zone.Identifier
```

Aby usunąć ten strumień i odblokować plik:

```
Remove-Item -Path "D:\az\test.ps1" -Stream Zone.Identifier
```

lub

```
Unblock-File -Path "D:\az\test.ps1"
```