Aktualizacje

Windows Update, WSUS, WinSxS i runbook aktualizacji produkcji.

- Windows Update z wiersza poleceń
- WSUS: czyszczenie i utrzymanie
- WinSxS: czyszczenie magazynu komponentów
- Standardowa aktualizacja systemu produkcyjnego (Windows)

Windows Update z wiersza poleceń

Typ: fiszka **Dotyczy: Windows Server 2016+ / Windows 10+, moduł PSWindowsUpdate**

Efekt: czyścimy zablokowany cache Windows Update, sprawdzamy i instalujemy poprawki z poziomu PowerShella, i weryfikujemy, co faktycznie się zainstalowało.

Skróty użyte na tej stronie (4)

Skrót	Co znaczy
CLI	Command Line Interface: praca przez wpisywanie komend w terminalu
GUI	Graphical User Interface: okienka i myszka zamiast komend
WinRE	Windows Recovery Environment: środowisko odzyskiwania Windows na osobnej partycji, potrzebne między innymi do szyfrowania BitLockerem
WSUS	Windows Server Update Services: firmowy serwer aktualizacji Windows

Fiszka

ZMIENIA SYSTEM

Czyszczenie cache Windows Update: zatrzymanie usług, zmiana nazw katalogów, ponowne uruchomienie

```
# Zatrzymujemy usługi trzymające otwarte pliki w SoftwareDistribution i catroot2
Stop-Service -Name wuauserv -Force
Stop-Service -Name cryptSvc -Force
Stop-Service -Name bits -Force
Stop-Service -Name msiserver -Force

# Zmieniamy nazwy katalogów zamiast je kasować — jeśli coś pójdzie nie tak, łatwo wrócić
```

```
do poprzedniego stanu
Rename-Item -Path "$env:windir\SoftwareDistribution" -NewName "SoftwareDistribution.old"
Rename-Item -Path "$env:windir\System32\catroot2" -NewName "catroot2.old"

# Uruchamiamy usługi ponownie — Windows sam odtworzy oba katalogi od nowa, puste
Start-Service -Name wuauserv
Start-Service -Name cryptSvc
Start-Service -Name bits
Start-Service -Name msiserver

# Wymuszamy natychmiastowe wykrycie dostępnych aktualizacji, zamiast czekać na kolejny
automatyczny cykl
usoclient StartScan
```

Uruchamiamy PowerShell jako administrator. Operację wykonujemy, gdy Windows Update faktycznie się zacina — rutynowe czyszczenie cache przy każdej aktualizacji nie jest potrzebne.

□□ TYLKO ODCZYT

Weryfikacja stanu usług po czyszczeniu cache: Get-Service

```
Get-Service wuauserv, bits
```

Obie usługi powinny mieć status `Running`. Kolejne uruchomienie Windows Update (przez GUI albo `Get-WindowsUpdate` niżej) powinno od nowa pobrać pełną listę dostępnych aktualizacji.

□□ **GUI** = Graphical User Interface: okienka i myszka zamiast komend

□□ OSTROŻNIE · NIEODWRACALNE

Czyszczenie podkatalogu Download: Remove-Item

```
Remove-Item -Path "$env:windir\SoftwareDistribution\Download\*" -Recurse -Force
```

Samo czyszczenie cache samo w sobie nie usuwa niczego z folderu `Download`, jeśli akurat trwa pobieranie — w razie potrzeby czyścimy ten podkatalog osobno. Robimy to też wtedy, gdy po pomyślnie wgranych aktualizacjach i restarcie zabrakło miejsca na dysku systemowym — pobrane pliki instalacyjne po udanej instalacji są już zbędne.

□□ ZMIENIA SYSTEM

Instalacja modułu: Install-Module

```
Install-Module -Name PSWindowsUpdate -force
```

□□ ZMIENIA SYSTEM

Zaimportowanie modułu do sesji: Import-Module

```
Import-Module PSWindowsUpdate
```

□□ TYLKO ODCZYT

Sprawdzenie dostępnych aktualizacji: Get-WindowsUpdate

```
Get-WindowsUpdate
```

□□ TYLKO ODCZYT

Pobranie aktualizacji bez instalacji: Get-WindowsUpdate -Download

```
Get-WindowsUpdate -AcceptAll -Download
```

Przydatne, gdy chcemy pobrać aktualizacje przed umówionym oknem serwisowym — w samym oknie zostaje już tylko instalacja i restart, bez czekania na pobieranie.

□□ ZMIENIA SYSTEM

Instalacja aktualizacji bez automatycznego restartu: Install-WindowsUpdate

```
Install-WindowsUpdate -MicrosoftUpdate -AcceptAll -IgnoreReboot
```

□□ ZMIENIA SYSTEM

Instalacja z automatycznym restartem: Install-WindowsUpdate -AutoReboot

```
Install-WindowsUpdate -MicrosoftUpdate -AcceptAll -AutoReboot
```

□□ ZMIENIA SYSTEM

Instalacja tylko jednej kategorii aktualizacji

```
Get-WindowsUpdate -Category "SecurityUpdates" | Install-WindowsUpdate
```

□□ TYLKO ODCZYT

Ukrycie aktualizacji, żeby nigdy nie została zainstalowana: Hide-WindowsUpdate

```
Hide-WindowsUpdate -KBArticleID "KB5034439"
```

Sięgamy po to, gdy konkretna poprawka powoduje znany problem na danej konfiguracji (klasyczny przykład — aktualizacje dotykające partycji WinRE na dyskach z za małą partycją odzyskiwania).

□□ **WinRE** = Windows Recovery Environment: środowisko odzyskiwania Windows na osobnej partycji, potrzebne między innymi do szyfrowania BitLockerem

□□ TYLKO ODCZYT

Sprawdzenie ukrytych aktualizacji: Get-WindowsUpdate -IsHidden

```
Get-WindowsUpdate -IsHidden
```

□□ OSTROŻNIE · NIEODWRACALNE

Odinstalowanie zainstalowanej aktualizacji: Remove-WindowsUpdate

```
Remove-WindowsUpdate -KBArticleID KB4489873 -NoRestart
```

□□ TYLKO ODCZYT

Sprawdzenie historii aktualizacji: Get-WUHistory

```
Get-WUHistory
```

□□ TYLKO ODCZYT

Sprawdzenie, czy wymagany jest restart: Get-WURebootStatus

```
Get-WURebootStatus
```

Przykład:

```
RebootRequired
--------------
          True
```

□□ TYLKO ODCZYT

Wylistowanie zainstalowanych poprawek: Get-HotFix

```
Get-HotFix
```

Przykład:

```
Source      Description      HotFixID      InstalledBy          InstalledOn
------      -----------      --------      -----------          -----------
RDS01       Update           KB5034123     NT AUTHORITY\SYSTEM   2024-10-08 00:00:00
RDS01       Security Update  KB5031354     NT AUTHORITY\SYSTEM   2024-09-10 00:00:00
```

□□ TYLKO ODCZYT

Dziennik zdarzeń instalacji: Get-WinEvent -LogName Setup

```
Get-WinEvent -LogName Setup | Select-Object TimeCreated, Id, Message -First 20
```

Dziennik `Setup` zawiera zdarzenia samego procesu serwicowania (CBS/DISM), niezależnie od tego, czy aktualizację zainstalowaliśmy przez GUI, `Install-WindowsUpdate`, czy WSUS — to tu szukamy śladu, jeśli instalacja poprawki zakończyła się błędem, którego nie widać w historii `Get-WUHistory`.

GUI = Graphical User Interface: okienka i myszka zamiast komend · **WSUS** = Windows Server Update Services: firmowy serwer aktualizacji Windows

Z życia wzięte

Z ŻYCIA WZIĘTE

Rds01 wisi we wtorek po Patch Tuesday na etapie „Pobieranie aktualizacji" od trzech godzin, a `Get-WindowsUpdate` w kółko zwraca tę samą listę bez postępu. Zatrzymujemy `wuauserv`, `bits` i `cryptSvc`, zmieniamy nazwę katalogu `SoftwareDistribution` na `SoftwareDistribution.old` i uruchamiamy usługi ponownie. Po `usoclient StartScan` Windows sam odtwarza pusty katalog i tym razem pobieranie kończy się w kilka minut.

Usługa wuauserv i katalog SoftwareDistribution

Pojęcie: wuauserv. Usługa Windows Update — proces w tle, który odpytuje serwer aktualizacji (Windows Update online albo lokalny WSUS), pobiera metadane i same pliki poprawek, i zleca ich instalację. Kiedy Windows Update „się zawiesza", zawiesza się właśnie ta usługa albo jej dane robocze.

Pojęcie: SoftwareDistribution. Katalog `%windir%\SoftwareDistribution`, w którym usługa trzyma pobrane pliki aktualizacji (`Download`) oraz bazę danych stanu aktualizacji. Uszkodzona zawartość tego katalogu jest najczęstszą przyczyną sytuacji, w której Windows Update nie widzi nowych aktualizacji, wisi na etapie pobierania, albo ciągle próbuje zainstalować tę samą poprawkę.



Zatrzymanie uslug, zmiana nazwy uszkodzonego cache i nowy pusty katalog po restarcie uslug (kliknięcie otwiera pełny rozmiar)

PSWindowsUpdate — wygodniejsza droga niż GUI

Moduł `PSWindowsUpdate` daje pełną kontrolę nad Windows Update z poziomu PowerShella, łącznie z operacjami, których nie ma w GUI (ukrywanie konkretnej poprawki, instalacja tylko wybranej kategorii).

Częste błędy

Instalacja przez CLI, a sprawdzanie efektu przez GUI. Panel Windows Update w ustawieniach systemu potrafi nie odświeżyć swojego stanu nawet po restarcie serwera i mylnie pokazywać, że aktualizacje nie zostały zainstalowane, mimo że `Get-HotFix` i `Get-WUHistory` potwierdzają instalację. Wystarczy raz kliknąć przycisk *Sprawdź aktualizacje* w panelu, żeby dane się odświeżyły — ale to dodatkowy, mylący krok, o którym warto pamiętać przed alarmowaniem, że coś nie zadziałało.

Czyszczenie cache jako rutynowy krok przed każdą aktualizacją. Bez powodu wydłuża to tylko czas potrzebny na ponowne wykrycie i pobranie pełnej listy aktualizacji.

Powiązane: Standardowa aktualizacja systemu produkcyjnego (Windows) · WSUS: czyszczenie i utrzymanie · WinSxS: czyszczenie magazynu komponentów

Źródła: notatki własne, dokumentacja modułu PSWindowsUpdate (PowerShell Gallery)

WSUS: czyszczenie i utrzymanie

Typ: fiszka **Dotyczy: Windows Server Update Services (WSUS)**

Efekt: odrzucamy zastąpione aktualizacje, czyścimy bazę WSUS z niepotrzebnych metadanych i plików, i ustawiamy to jako zadanie cykliczne zamiast robić to ręcznie za każdym razem.

Skróty użyte na tej stronie (4)

Skrót	Co znaczy
HTTP	Hypertext Transfer Protocol: protokół stron WWW
HTTPS	HTTP zabezpieczony szyfrowaniem TLS
SQL	Structured Query Language: język zapytań do relacyjnych baz danych
WSUS	Windows Server Update Services: firmowy serwer aktualizacji Windows

Fiszka

TYLKO ODCZYT

Uruchomienie skryptu odrzucającego zastąpione aktualizacje: Decline-Updates.ps1

```
.\Decline-Updates.ps1 -WSUSServer WSUSServer.Company.com -WSUSPort 8530
```

`WSUSServer.Company.com` to adres naszego serwera WSUS, `8530` to numer portu WSUS (domyślnie `8530` dla HTTP, `8531` dla HTTPS). Efekty działania: skrypt analizuje listę dostępnych aktualizacji, odrzuca te spełniające kryteria, a jeśli odrzucił cokolwiek — od razu uruchamia czyszczenie WSUS, żeby zwolnić miejsce.

WSUS = Windows Server Update Services: firmowy serwer aktualizacji Windows · **HTTP** = Hypertext Transfer Protocol: protokół stron WWW · **HTTPS** = HTTP zabezpieczony szyfrowaniem TLS

⚠ ZMIENIA SYSTEM

Treść skryptu

```
Param(
    [Parameter(Mandatory=$false,
    ValueFromPipeline=$true,
    ValueFromPipelineByPropertyName=$true,
    ValueFromRemainingArguments=$false,
    Position=0)]
    [string]$WSUSServer = "Localhost", #default to localhost
    [int]$WSUSPort=8530,
    [switch]$reportonly
    )

Function Decline-Updates{
    Param(
        [string]$WsusServer,
        [int]$WSUSPort,
        [switch]$ReportOnly
    )
    write-host "Connecting to WSUS Server $WSUSServer and getting list of updates"
    $Wsus = Get-WSUSserver -Name $WSUSServer -PortNumber $WSUSPort
    if($WSUS -eq $Null){
        write-error "unable to contact WSUSServer $WSUSServer"
    }else{
        $Updates = $wsus.GetUpdates()
        write-host "$(($Updates | where {$_.IsDeclined -eq $false} | measure).Count)
Updates before cleanup"
        $updatesToDecline = $updates | where {$_.IsDeclined -eq $false -and (
        $_.IsSuperseded -eq $true -or   #remove superseded updates
        $_.PublicationState -eq "Expired" -or #remove updates that have been pulled by
Microsoft
        $_.LegacyName -imatch "ia64" -or #remove updates for itanium computers (1/2)
        $_.LegacyName -imatch "x86" -or  #remove updates for 32-bit computers
        $_.LegacyName -match "XP" -or   #remove Windows XP updates (1/2)
        $_.producttitles -match "XP" -or #remove Windows XP updates (1/2)
        $_.Title -match "Itanium" -or   #remove updates for itanium computers (2/2)
        $_.Title -match "ARM64" -or   #remove updates for itanium computers (2/2)
```

```
        ($_.Title -imatch "language" -and $_.Title -notmatch "(PL|ENG)") -or  #remove
langauge packs
        $_.Title -match "Sprachpaket" -or
        $_.title -match "Internet Explorer 7" -or #remove updates for old versions of IE
        $_.title -match "Internet Explorer 8" -or
        $_.title -match "Internet Explorer 9" -or
        $_.title -match "Japanese" -or #some non-english updates are not filtered by
WSUS language filtering
        $_.title -match "Korean" -or
        $_.title -match "Taiwan" -or
        $_.Title -match "Beta" -or     #Beta products and beta updates
        $_.title -match "Embedded"     #Embedded version of Windows
        )}

        write-host "$(($updatesToDecline | measure).Count) Updates to decline"
        $changemade = $false
        if($reportonly){
            write-host "ReportOnly was set to true, so not making any changes"
        }else{
            $changemade = $true
            $updatesToDecline | %{$_.Decline()}
        }

        #Decline updates released more then 3 months prior to the release of an included
service pack
        # - service packs updates don't appear to contain the supersedance information.
        Foreach($SP in $($updates | where title -match "^Windows Server \d{4} .* Service
Pack \d")){
            if(($SP.ProductTitles |measure ).count -eq 1){
                $updatesToDecline = $updates | where {$_.IsDeclined -eq $false -and
$_.ProductTitles -contains $SP.ProductTitles -and $_.CreationDate -lt
$SP.CreationDate.Addmonths(-3)}
                if($updatesToDecline -ne $null){
                    write-host "$(($updatesToDecline | measure).Count) Updates to
decline (superseded by $($SP.Title))"
                    if(-not $reportonly){
                        $changemade = $true
                        $updatesToDecline | %{$_.Decline()}
                    }
```

```
                }
            }
        }

        #if changes were made, run a WSUS cleanup to recover disk space
        if($changemade -eq $true -and $reportonly -eq $false){
            $Updates = $wsus.GetUpdates()
            write-host "$(($Updates | where {$_.IsDeclined -eq $false} | measure).Count)
Updates remaining, running WSUS cleanup"
            Invoke-WsusServerCleanup -updateServer $WSUS -CleanupObsoleteComputers -
CleanupUnneededContentFiles -CleanupObsoleteUpdates -CompressUpdates -
DeclineExpiredUpdates -DeclineSupersededUpdates
        }

    }
}

Decline-Updates -WSUSServer $WSUSServer -WSUSPort $WSUSPort -reportonly:$reportonly
```

Uruchomienie z `-reportonly` pokazuje, co skrypt by odrzucił, bez wykonywania żadnej zmiany — wygodne przy pierwszym uruchomieniu na nieznanym serwerze, zanim zaufamy mu na produkcji.

WSUS = Windows Server Update Services: firmowy serwer aktualizacji Windows · **ARM64** = 64-bitowa architektura procesorów ARM (aarch64): inna niż x86-64 w zwykłych serwerach i komputerach; serwery P4 i Lore mają dla niej osobne wydania · **SP** = Special Publication: oznaczenie serii dokumentów NIST, tu wytycznych o hasłach SP 800-63B

TYLKO ODCZYT

Uruchomienie czyszczenia WSUS ręcznie: Invoke-WsusServerCleanup

```
Invoke-WsusServerCleanup -CleanupObsoleteComputers -CleanupUnneededContentFiles -
CleanupObsoleteUpdates -CompressUpdates -DeclineExpiredUpdates -DeclineSupersededUpdates
```

Skrypt wyżej wywołuje czyszczenie WSUS automatycznie, ale te same parametry można uruchomić osobno, ręcznie.

WSUS = Windows Server Update Services: firmowy serwer aktualizacji Windows

◫ TYLKO ODCZYT

Reindeksacja bazy SUSDB: sqlcmd

```
sqlcmd -S NazwaSerwera\NazwaInstancji -d SUSDB -i .\WsusDBMaintenance.sql
```

> (gdzie `NazwaSerwera\NazwaInstancji` to instancja SQL Server albo Windows Internal Database trzymająca `SUSDB`, a `WsusDBMaintenance.sql` to skrypt reindeksujący udostępniany przez Microsoft dla WSUS)

Po dużym czyszczeniu warto zreindeksować tabele w `SUSDB` — usunięcie tysięcy wierszy zostawia mocno pofragmentowane indeksy, które spowalniają samą konsolę WSUS. Uruchamiamy standardowy skrypt reindeksacji Microsoftu przeciwko bazie `SUSDB` (na WID albo na pełnym SQL Server, w zależności od instalacji).

◫ **SQL** = Structured Query Language: język zapytań do relacyjnych baz danych · **WSUS** = Windows Server Update Services: firmowy serwer aktualizacji Windows · **sqlcmd** = SQL command: klient bazy MSSQL

◫ ZMIENIA SYSTEM

Utworzenie zadania cyklicznego: New-ScheduledTaskAction + Register-ScheduledTask

```
$action = New-ScheduledTaskAction -Execute "powershell.exe" -Argument "-File
C:\Skrypty\Decline-Updates.ps1 -WSUSServer WSUSServer.Company.com -WSUSPort 8530"
$trigger = New-ScheduledTaskTrigger -Weekly -DaysOfWeek Sunday -At 2am
Register-ScheduledTask -TaskName "WSUS - odrzucanie i czyszczenie" -Action $action -
Trigger $trigger -RunLevel Highest -User "NazwaDomeny\NazwaKonta"
```

Zarówno odrzucanie starych paczek, jak i reindeksację warto uruchamiać automatycznie zamiast pamiętać o nich ręcznie.

◫ **WSUS** = Windows Server Update Services: firmowy serwer aktualizacji Windows

Z życia wzięte

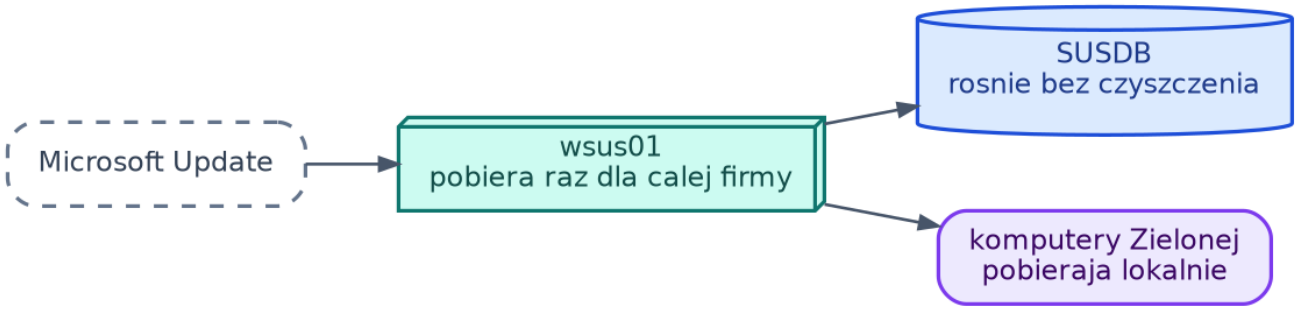
Z ŻYCIA WZIĘTE

We wtorek po Patch Tuesday wsus01 rozdaje najnowsze poprawki na wszystkie stacje w Pamiątkowie i Swarzędzu naraz, a konsola WSUS reaguje z zauważalnym opóźnieniem przy każdym kliknięciu. Sprawdzamy `SUSDB` i baza od dwóch lat nie była czyszczona, bo nikt nie ustawił tego jako zadanie cykliczne. Po uruchomieniu skryptu odrzucającego zastąpione aktualizacje i `Invoke-WsusServerCleanup` konsola znów odpowiada normalnie.

Czym jest WSUS i dlaczego jego baza puchnie

Pojęcie: WSUS. Windows Server Update Services — rola serwerowa, która pobiera aktualizacje z Microsoft Update raz, do jednego miejsca w sieci lokalnej, i rozprowadza je do klientów przez wewnętrzny serwer, zamiast każda maszyna pobierała te same pliki niezależnie z internetu. Metadane aktualizacji (tytuły, zależności, informacje o zastępowaniu) trzyma w bazie `SUSDB`, a same pliki instalacyjne — w katalogu na dysku.

Baza puchnie, bo WSUS domyślnie nie usuwa niczego sam z siebie. Każda aktualizacja, która została zastąpiona nowszą wersją (superseded), każdy wygasły biuletyn, każdy język, którego nie używamy — wszystko to zostaje w bazie i na dysku, dopóki ktoś ręcznie nie każe WSUS-owi to posprzątać. Po kilku latach eksploatacji bez czyszczenia baza `SUSDB` potrafi urosnąć do rozmiaru, przy którym konsola WSUS zaczyna reagować z zauważalnym opóźnieniem.



Aktualizacje ida od Microsoftu przez wsus01 do komputerow Zielonej, a SUSDB rosnie bez czyszczenia (kliknięcie otwiera pełny rozmiar)

Odrzucenie zastąpionych aktualizacji

Ręczne przeglądanie tysięcy aktualizacji w konsoli i odrzucanie ich pojedynczo nie ma sensu — poniższy skrypt robi to automatycznie, według zestawu reguł.

Wymagania: uprawnienia administracyjne do serwera WSUS, zainstalowany moduł PowerShell dla WSUS, połączenie z serwerem WSUS.

Zakres działania. Skrypt odrzuca aktualizacje, jeśli spełniają którykolwiek z poniższych warunków: są zastąpione przez nowsze aktualizacje (superseded); są oznaczone jako wygasłe (expired) według definicji Microsoft; dotyczą systemów operacyjnych x86 lub Itanium; są przeznaczone dla Windows XP; są pakietami językowymi (poza polskim i angielskim); dotyczą starszych wersji Internet Explorer (7, 8, 9); są aktualizacjami specyficznymi dla regionów; są aktualizacjami beta; są przeznaczone dla systemów wbudowanych (embedded).

Jeżeli jedna aktualizacja obejmuje wiele systemów operacyjnych i tylko część z nich spełnia powyższe kryteria, odrzucone zostają wyłącznie te wersje, które kryteria spełniają.

Invoke-WsusServerCleanup — parametry

Parametr	Co robi
`-CleanupObsoleteComputers`	usuwa z bazy komputery, które nie łączyły się z WSUS od dawna (odinstalowane, wymienione)
`-CleanupUnneededContentFiles`	usuwa z dysku pliki aktualizacji, których żadna zatwierdzona aktualizacja już nie potrzebuje
`-CleanupObsoleteUpdates`	usuwa z bazy metadane aktualizacji odrzuconych i wygasłych
`-CompressUpdates`	kompresuje zrewidowane aktualizacje, zmniejszając rozmiar bazy
`-DeclineExpiredUpdates`	odrzuca aktualizacje oznaczone przez Microsoft jako wygasłe
`-DeclineSupersededUpdates`	odrzuca aktualizacje zastąpione nowszymi

Częste błędy

Uruchamianie czyszczenia w środku dnia roboczego. `Invoke-WsusServerCleanup` na dużej, dawno nieczyszczonej bazie potrafi trwać wiele godzin i mocno obciąża serwer WSUS — konsola w tym czasie reaguje wolno albo w ogóle się nie otwiera. Zadanie planujemy na noc albo weekend.

Czyszczenie bez wcześniejszego odrzucenia starych aktualizacji. `Invoke-WsusServerCleanup` bez `-DeclineSupersededUpdates` / `-DeclineExpiredUpdates` posprząta tylko to, co już odrzucone — bez skryptu odrzucającego większość zbędnych aktualizacji nigdy nie trafi do sprzątania.

Powiązane: Windows Update z wiersza poleceń · WinSxS: czyszczenie magazynu komponentów

Źródła: notatki własne, skrypt bazowany na przykładzie Nicka Ealesa (Microsoft), dokumentacja `Invoke-WsusServerCleanup`

WinSxS: czyszczenie magazynu komponentów

Typ: fiszka **Dotyczy: Windows Server 2012 R2+ / Windows 10+**

Efekt: czyścimy nieużywane wersje komponentów systemowych, rozumiejąc, ile miejsca realnie da się odzyskać i czym grozi trwałe zatwierdzenie zmian przez `/ResetBase`*.*

Skróty użyte na tej stronie (2)

Skrót	Co znaczy
GB	Gigabyte: miliard bajtów, jednostka pojemności pamięci i plików, tu najczęściej rozmiar modelu albo pamięci karty graficznej
GUI	Graphical User Interface: okienka i myszka zamiast komend

Fiszka

TYLKO ODCZYT

Analiza magazynu komponentów: Dism /AnalyzeComponentStore

```
Dism.exe /online /Cleanup-Image /AnalyzeComponentStore
```

Sprawdzamy tym poleceniem, zanim cokolwiek ruszymy, czy magazyn komponentów w ogóle kwalifikuje się do czyszczenia — DISM sam ocenia, czy operacja przyniesie realny efekt, i podaje to w polu `Component Store Cleanup Recommended`.

Przykład:

```
Component Store (WinSxS) information:

Windows Explorer Reported Size of Component Store : 8.14 GB
Actual Size of Component Store                    : 6.02 GB
```

```
Shared with Windows folder                    : 3.11 GB
Backups and Disabled Features                 : 1.44 GB
Cache and Temporary Data                      : 0.02 GB
Date of Last Cleanup                          : 2024-06-11
Number of Reclaimable Packages                : 6
Component Store Cleanup Recommended           : Yes
```

◫◫ TYLKO ODCZYT

Czyszczenie magazynu komponentów: Dism /StartComponentCleanup

```
Dism.exe /online /Cleanup-Image /StartComponentCleanup
```

Usuwa nadmiarowe pliki dla komponentów, które nadal mogą zostać odinstalowane (poprzednie wersje aktualizacji sprzed 30 dni), ale zachowuje możliwość ich cofnięcia.

◫◫ OSTROŻNIE · NIEODWRACALNE

Czyszczenie z trwałym zatwierdzeniem: Dism /StartComponentCleanup /ResetBase

```
Dism.exe /online /Cleanup-Image /StartComponentCleanup /ResetBase
```

`/ResetBase` idzie o krok dalej niż samo `/StartComponentCleanup` — usuwa też poprzednie wersje aktualnie zainstalowanych komponentów, których normalnie `/StartComponentCleanup` nie rusza, bo wciąż dałoby się do nich wrócić.

Tu leży haczyk: po `/ResetBase` nie da się już odinstalować żadnej wcześniej zainstalowanej aktualizacji, której komponenty zostały właśnie oczyszczone — `Get-HotFix`/`wusa /uninstall` na taką poprawkę zwróci błąd, bo dane potrzebne do przywrócenia poprzedniej wersji plików już nie istnieją. Sięgamy po `/ResetBase` dopiero wtedy, gdy mamy pewność, że nie będziemy chcieli cofnąć żadnej z zainstalowanych dotąd aktualizacji — typowo po potwierdzeniu stabilności systemu przez dłuższy czas po ostatnim patchowaniu.

Z życia wzięte

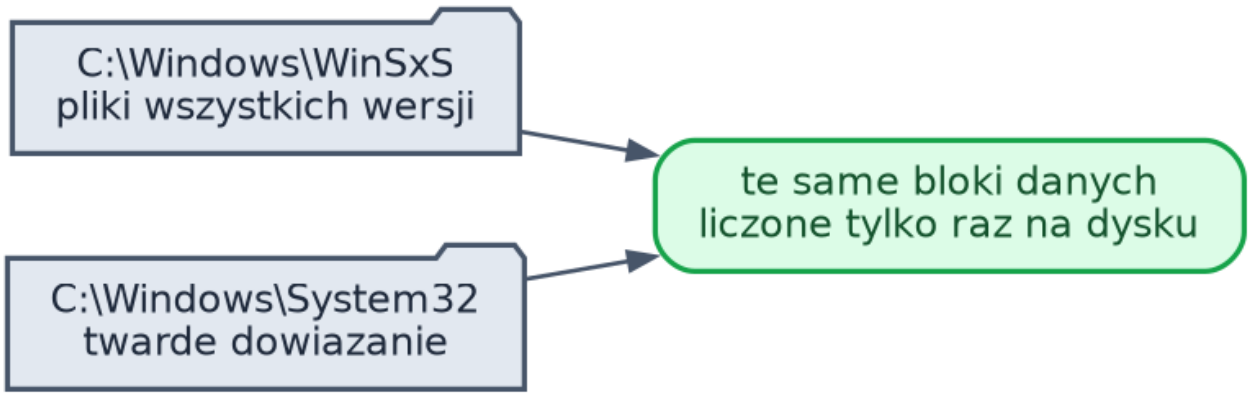
Z ŻYCIA WZIĘTE

Pan Zbigniew wraca z urlopu z nowym laptopem i prosi o sprawdzenie, czemu na starym dysk C: pokazuje tylko trzy gigabajty wolnego miejsca, mimo że nic nowego nie instalował. Eksplorator pokazuje folder `WinSxS` jako osiem gigabajtów, więc na pierwszy rzut oka to on jest winny. `Dism /AnalyzeComponentStore` pokazuje, że realny narzut to półtora gigabajta, a resztę Eksplorator liczy podwójnie przez twarde dowiązania z `System32`.

Magazyn komponentów i mylący rozmiar

Pojęcie: magazyn komponentów (WinSxS). Katalog `C:\Windows\WinSxS`, w którym Windows trzyma wszystkie zainstalowane wersje komponentów systemowych — nie tylko bieżącą, ale też poprzednie, które mogą być potrzebne do odinstalowania aktualizacji albo do naprawy systemu. Nazwa „side-by-side" odnosi się właśnie do tego, że wiele wersji tej samej biblioteki systemowej współistnieje obok siebie.

Rozmiar folderu `WinSxS` raportowany przez Eksplorator czy `dir` jest systematycznie mylący. Pliki w `System32` i innych katalogach systemowych nie są kopiami plików z `WinSxS` — są twardymi dowiązaniami (hard links) do tych samych bloków danych na dysku. Narzędzie liczące rozmiar folderu, które nie rozumie twardych dowiązań, liczy te same dane wielokrotnie — raz jako plik w `WinSxS`, drugi raz jako „kopię" w `System32`. Realny narzut miejsca, jaki `WinSxS` faktycznie zajmuje ponad to, co i tak byłoby potrzebne do działania systemu, jest znacznie mniejszy niż suma wyświetlana przez Eksplorator.



WinSxS i System32 wskazuja na te same bloki danych przez twarde dowiazania, Eksplorator liczy je podwojnie (kliknięcie otwiera pełny rozmiar)

Cleanmgr z profilem sageset

Alternatywna droga przez GUI, wygodna do jednorazowego, ręcznego doboru kategorii do wyczyszczenia (nie tylko WinSxS, ale też pliki tymczasowe, stare raporty błędów, kosz):

```
cleanmgr /sageset:1
```

Otwiera okno konfiguracji, w którym zaznaczamy kategorie do sprzątania (w tym „Czyszczenie systemu Windows Update" — odpowiednik `/StartComponentCleanup`). Zapisany profil `1` uruchamiamy bez ponownego wybierania opcji:

```
cleanmgr /sagerun:1
```

Ile realnie można odzyskać

Na systemie regularnie aktualizowanym i regularnie czyszczonym `/StartComponentCleanup` bez `/ResetBase` odzyskuje zwykle od kilkuset megabajtów do 1-2 GB. Na systemie, który nigdy nie był czyszczony i zebrał kilka lat aktualizacji kumulatywnych, `/ResetBase` potrafi odzyskać kilka do kilkunastu gigabajtów — ale kosztem utraty możliwości cofnięcia którejkolwiek z wcześniej zainstalowanych poprawek.

Częste błędy

Uruchamianie `/ResetBase` rutynowo, przy każdym czyszczeniu, bez zastanowienia. Zamiast tego zaczynamy od samego `/StartComponentCleanup`, a `/ResetBase` traktujemy jako świadomą, rzadszą decyzję.

Oczekiwanie, że wynik z Eksploratora („rozmiar folderu WinSxS to X GB") odpowiada temu, ile miejsca DISM faktycznie odzyska — jak wyżej, spora część tego rozmiaru to twarde dowiązania współdzielone z resztą systemu, nie da się ich „odzyskać" bez usunięcia samych komponentów systemowych.

Powiązane: Windows Update z wiersza poleceń · Diagnostyka systemu

Źródła: notatki własne, dokumentacja Microsoft (`Dism.exe`, Component Store Servicing)

Standardowa aktualizacja systemu produkcyjnego (Windows)

□□ Typ: procedura **□□ Dotyczy: Windows Server 2016+**

□ Czas: ~45–90 min na serwer, plus czas trwania restartu

Efekt: serwer ma zainstalowane bieżące aktualizacje, jest po restarcie, wszystkie usługi wróciły do stanu sprzed prac, a ewidencja serwerów odzwierciedla nową datę patchowania. Odpowiednik dla Linuksa opisuje strona Standardowa aktualizacja systemu produkcyjnego.

□□ Skróty użyte na tej stronie (2)

Skrót	Co znaczy
CMDB	Configuration Management Database: baza konfiguracji, czyli spis elementów, z których składają się usługi, i ich powiązań; w Zielonej rolę tę pełni NetBox
GUI	Graphical User Interface: okienka i myszka zamiast komend

Zanim zaczniemy

> **□□ Z ŻYCIA WZIĘTE**
>
> Zgłoszenie na wtorek po Patch Tuesday obejmuje sql01, bo księgowość kończy akurat miesiąc i pani Halina prosiła o przesunięcie okna serwisowego na później niż zwykle. Robimy snapshot maszyny bez zrzutu pamięci, sprawdzamy wolne miejsce na dysku systemowym i dopiero wtedy zaczynamy pobieranie aktualizacji, żeby samo okno ograniczyć do instalacji i restartu.

Sprawdzamy zgłoszenie i okno serwisowe, w którym wykonujemy prace — patchowanie produkcyjnego serwera poza umówionym oknem jest wyjątkiem, nie regułą.

Wykonujemy snapshot maszyny wirtualnej przed jakąkolwiek zmianą. Przy tworzeniu snapshotu odznaczamy opcję zrzutu pamięci maszyny (*include virtual machine memory*), chyba że mamy konkretny powód, żeby ją zostawić — domyślnie zostaje odznaczona, bo istotnie wydłuża czas tworzenia i kasowania snapshotu bez realnej korzyści dla samego patchowania.

Sprawdzamy ilość wolnego miejsca na dysku systemowym — pobrane pliki aktualizacji w `SoftwareDistribution\Download` potrafią zająć kilka gigabajtów, a instalacja dużej aktualizacji kumulatywnej wymaga dodatkowego zapasu na dysku na czas samej operacji.

Spisujemy listę usług, które przed patchowaniem trzeba zatrzymać ręcznie — zwykle agent antywirusowy/EDR (część z nich blokuje pliki systemowe w trakcie instalacji poprawek) oraz usługi własnej aplikacji, jeśli aplikacja nie radzi sobie z restartem w trakcie swojej pracy. Włączamy tryb konserwacji w systemie monitoringu na patchowany serwer, żeby restart nie wygenerował fałszywych alertów.

Sprawdzamy, czy przy okazji restartu nie warto podnieść wersji VMware Tools albo wirtualnego sprzętu (compatibility/hardware version) maszyny — restart i tak nastąpi, więc to naturalna okazja, żeby zrobić to bez dodatkowego przestoju.

Przebieg



Piec etapow standardowej aktualizacji: snapshot, pobranie, instalacja w oknie, restart, weryfikacja (kliknięcie otwiera pełny rozmiar)

1. Instalacja i import modułu PSWindowsUpdate

Uruchamiamy okno PowerShell jako administrator. Jeśli moduł nie jest jeszcze zainstalowany:

```
Set-ExecutionPolicy –ExecutionPolicy RemoteSigned -Force
Install-Module -Name PSWindowsUpdate -Force
```

Instalujemy moduł nawet wtedy, gdy wygląda na już zainstalowany — polecenie przy okazji dociąga jego nowszą wersję, jeśli jest dostępna.

Na starszych wersjach Windows Server automatyczne pobranie modułu z PowerShell Gallery bywa niedostępne (brak zaufanego źródła, brak dostępu do internetu z serwera). W takiej sytuacji instalujemy moduł ręcznie: pobieramy go na innej maszynie z dostępem do internetu, kopiujemy katalog modułu na docelowy serwer i importujemy z lokalnej ścieżki.

```
Save-Module –Name PSWindowsUpdate –Path C:\ps\
```

Skopiowaną zawartość umieszczamy w `%WINDIR%\System32\WindowsPowerShell\v1.0\Modules` na serwerze docelowym, a następnie:

```
Import-Module PSWindowsUpdate
```

2. Pobranie aktualizacji przed oknem serwisowym

Żeby skrócić samo okno serwisowe, aktualizacje warto pobrać wcześniej — w oknie zostaje wtedy tylko instalacja i restart.

Przed pobraniem czyścimy katalog `Download`, żeby nie kumulować starych, częściowo pobranych plików, i sprawdzamy wolne miejsce na dysku C:

```
Remove-Item -Path "$env:windir\SoftwareDistribution\Download\*" -Recurse -Force
```

Pobieramy aktualizacje bez instalowania ich od razu:

```
Get-WindowsUpdate -AcceptAll -Download
```

3. Instalacja w oknie serwisowym

W umówionym oknie serwisowym instalujemy wcześniej pobrane aktualizacje. Instalację można wykonać przez GUI albo przez PowerShell — z moich obserwacji aktualizacje instalują się przez PowerShell nieco szybciej, ale GUI ma tę przewagę, że rzetelnie odświeża swój stan po restarcie (więcej w sekcji *Gdy coś nie zadziała* niżej).

```
Install-WindowsUpdate -MicrosoftUpdate -AcceptAll -IgnoreReboot
```

4. Restart serwera

```
Restart-Computer -Force
```

Weryfikacja

Sprawdzamy, czy usługi, które zatrzymywaliśmy ręcznie przed pracami (agent antywirusowy/EDR, usługi aplikacji), wróciły do działania po restarcie — część z nich trzeba włączyć ręcznie, jeśli nie były skonfigurowane na automatyczny start.

```
Get-Service NazwaUslugi
```

Sprawdzamy, czy aplikacja działająca na serwerze odpowiada — w zależności od aplikacji jest to test end-to-end, sprawdzenie portu nasłuchu, albo zapytanie do endpointu health-check.

Przeglądamy dziennik zdarzeń w poszukiwaniu błędów, które pojawiły się dopiero po restarcie:

```
Get-WinEvent -FilterHashtable @{LogName='System','Application'; Level=1,2; StartTime=(Get-Date).AddHours(-2)} | Format-Table TimeCreated, LogName, Id, LevelDisplayName, Message -Wrap
```

Sprawdzamy, czy dalej nie jest wymagany kolejny restart (bywa, że jedna aktualizacja kumulatywna wymaga dwóch cykli restartu):

```
Get-WURebootStatus
```

Jeśli po pomyślnej instalacji i restarcie zabrakło miejsca na dysku, czyścimy ponownie katalog pobranych plików — są już zbędne:

```
Remove-Item -Path "$env:windir\SoftwareDistribution\Download\*" -Recurse -Force
```

Aktualizujemy datę wykonania patchowania i status serwera w ewidencji (CMDB / arkusz utrzymania), zdejmujemy tryb konserwacji w systemie monitoringu.

Wycofanie zmian

Jeśli konkretna aktualizacja jest przyczyną problemu, odinstalowujemy tylko ją, bez przywracania całego serwera:

```
Remove-WindowsUpdate -KBArticleID NumerKB -NoRestart
```

Jeżeli problem jest głębszy niż jedna poprawka — usługa nie startuje, aplikacja nie działa mimo poprawnie zainstalowanych aktualizacji — wracamy do snapshotu wykonanego przed rozpoczęciem prac, zamiast szukać przyczyny na produkcji pod presją czasu.

Gdy coś nie zadziała

Objaw	Przyczyna	Naprawa
GUI Windows Update pokazuje, że aktualizacje nie zostały zainstalowane, mimo że instalacja przez PowerShell zakończyła się sukcesem	znany błąd odświeżania panelu GUI po instalacji przez CLI	sprawdzamy stan przez `Get-HotFix` / `Get-WUHistory`, a w GUI wystarczy raz kliknąć *Sprawdź aktualizacje*, żeby dane się odświeżyły
Instalacja wisi na etapie pobierania	uszkodzony cache Windows Update	czyścimy cache zgodnie z procedurą na stronie Windows Update z wiersza poleceń
Po restarcie usługa aplikacji się nie uruchamia	usługa nie była ustawiona na automatyczny start, albo zależy od usługi wciąż się podnoszącej	sprawdzamy typ uruchomienia (`Get-Service` / `Set-Service -StartupType Automatic`), w razie potrzeby startujemy ręcznie po odczekaniu na zależności
Brakuje miejsca na dysku w trakcie instalacji	nie wyczyszczono katalogu `Download` przed pobraniem, albo aktualizacja kumulatywna wymaga większego zapasu niż zwykle	czyścimy `SoftwareDistribution\Download`, w razie potrzeby tymczasowo podłączamy dodatkowy dysk pod hypervisorem
Po aktualizacji system działa niestabilnie, przyczyna niejasna	aktualizacja wprowadziła regresję niewidoczną od razu w dzienniku zdarzeń	przywracamy snapshot wykonany przed pracami, zgłaszamy problem, nie próbujemy diagnozować pod presją czasu na aktywnej produkcji

Powiązane: Windows Update z wiersza poleceń · WSUS: czyszczenie i utrzymanie · Standardowa aktualizacja systemu produkcyjnego (odpowiednik dla Linuksa)

Źródła: notatki własne