Diagnostyka i utrzymanie

Liczniki wydajności, WMI, dziennik zdarzeń, kosz i profile.

- Diagnostyka systemu Windows
- Kosz i profile użytkowników

Diagnostyka systemu Windows

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

Efekt: naprawiamy zepsute liczniki wydajności i repozytorium WMI, wiemy, w których dziennikach zdarzeń i po jakich identyfikatorach szukać przyczyny problemu, i zbieramy dane potrzebne do zgłoszenia.

Skróty użyte na tej stronie (4)

Skrót	Co znaczy
CIM	Common Information Model: standard opisu obiektów zarządzania w Windows, na nim stoi WMI
CPU	Central Processing Unit: procesor
MOF	Microsoft Operations Framework: rama Microsoftu do prowadzenia eksploatacji IT, oparta na ITIL; wersja 4.0 z 2008 roku
WMI	Windows Management Instrumentation: interfejs do odczytu stanu Windowsa

Fiszka

TYLKO ODCZYT

Przebudowanie liczników wydajności: lodctr /R

Komendy uruchamiamy w wierszu polecenia jako administrator, osobno dla wersji 64- i 32-bitowej:

```
cd C:\Windows\system32
lodctr /R
```

```
cd C:\Windows\SysWOW64
lodctr /R
```

`/R` odbudowuje definicje liczników wydajności z plików rejestru (`.ini`) zainstalowanych komponentów — naprawia sytuację, w której konkretny licznik zniknął z listy albo pokazuje same zera.

cd = change directory: przejdź do katalogu

TYLKO ODCZYT

Resynchronizacja liczników z WMI: winmgmt /resyncperf

```
WINMGMT.EXE /RESYNCPERF
```

Wykonujemy po `lodctr /R`, jeśli liczniki są poprawnie zarejestrowane, ale wciąż niewidoczne przez zapytania WMI (na przykład przez narzędzia monitoringu korzystające z klas `Win32_PerfFormattedData_*`) — synchronizuje bazę liczników z repozytorium WMI.

WMI = Windows Management Instrumentation: interfejs do odczytu stanu Windowsa

TYLKO ODCZYT

Weryfikacja spójności repozytorium WMI: winmgmt /verifyrepository

```
winmgmt /verifyrepository
```

Pierwszy krok przy podejrzeniu problemów z WMI — sprawdza spójność repozytorium bez wprowadzania żadnych zmian.

Przykład:

```
WMI repository is inconsistent.
```

WMI = Windows Management Instrumentation: interfejs do odczytu stanu Windowsa

TYLKO ODCZYT

Naprawa repozytorium bez utraty danych: winmgmt /salvagerepository

```
winmgmt /salvagerepository
```

Próbuje naprawić wykryte niespójności, zachowując istniejące dane repozytorium — sięgamy po to w pierwszej kolejności, zanim rozważymy pełny reset.

OSTROŻNIE · NIEODWRACALNE

Reset repozytorium do stanu fabrycznego: winmgmt /resetrepository

```
winmgmt /resetrepository
```

Ostateczność. Polecenie usuwa całe repozytorium WMI i odbudowuje je od zera wyłącznie z domyślnych definicji MOF systemu — wszystko, co zarejestrowały w WMI zainstalowane aplikacje i agenci (własne klasy monitoringu, integracje backupu, niektóre sterowniki), znika i musi zostać zarejestrowane ponownie przez reinstalację tych komponentów. Sięgamy po `/resetrepository` dopiero, gdy `/salvagerepository` nie usunął problemu, i mamy świadomość, że część oprogramowania na serwerze będzie wymagać ponownej rejestracji swoich klas WMI.

WMI = Windows Management Instrumentation: interfejs do odczytu stanu Windowsa · **MOF** = Microsoft Operations Framework: rama Microsoftu do prowadzenia eksploatacji IT, oparta na ITIL; wersja 4.0 z 2008 roku

Z życia wzięte

Z ŻYCIA WZIĘTE

Zabbix przestaje pokazywać wykorzystanie CPU dla fs01, mimo że sam host odpowiada normalnie na wszystkie inne zapytania. Sprawdzamy licznik ręcznie przez `Get-Counter` i dostajemy błąd o nieznanej kategorii, jakby licznik w ogóle nie istniał w systemie. Po `lodctr /R` i `winmgmt /resyncperf` na fs01 Zabbix znów widzi metryki, bez potrzeby restartu całego serwera.

Liczniki wydajności i repozytorium WMI

Pojęcie: liczniki wydajności (performance counters). Mechanizm, przez który procesy systemowe i aplikacje publikują swoje metryki (zużycie CPU, kolejki dysku, liczba połączeń) w formie odczytywalnej przez Podgląd wydajności, `Get-Counter` czy zewnętrzny system monitoringu. Definicje liczników są rejestrowane w systemie przy instalacji komponentu — kiedy ten rejestr się rozjedzie (uszkodzona instalacja, częściowo wycofana aktualizacja), monitoring zaczyna zgłaszać braki danych mimo że sam proces działa poprawnie.

Pojęcie: repozytorium WMI. Baza danych (CIM), w której Windows Management Instrumentation trzyma definicje klas obiektów zarządzania i ich instancje. Z tego repozytorium korzysta ogromna część narzędzi administracyjnych — `Get-CimInstance`, skrypty logon, agenci monitoringu, sam Menedżer zadań. Uszkodzone repozytorium objawia się błędami w rodzaju „Invalid class" albo „Not found" przy zapytaniach, które wcześniej działały bez problemu.

Dziennik zdarzeń

Pojęcie: dziennik zdarzeń. Uporządkowany zapis zdarzeń systemowych, aplikacyjnych i bezpieczeństwa, przeglądany przez Podgląd zdarzeń albo `Get-WinEvent`. Każdy wpis ma dziennik (log), źródło, poziom istotności i identyfikator zdarzenia (Event ID) — to właśnie po identyfikatorze najszybciej filtrujemy konkretny typ zdarzenia spośród tysięcy innych.

Sprawdzenie, kto zrestartował system: Get-WinEvent -FilterHashtable

```
Get-WinEvent -FilterHashtable @{logname = 'System'; id = 1074} | Format-Table -Property * -Wrap
```

Przykład:

```
TimeCreated          Message
-----------          -------
2024-11-03 22:14:02  The process C:\Windows\system32\svchost.exe (FS01) has
                      initiated the restart of computer FS01 on behalf of
                      user DOMENA\jan.kowalski for the following reason:
```

```
Operating System: Service pack (Planned)
Reason code: 0x80020003
Shutdown Type: restart
```

Zdarzenie `1074` jest pierwszym miejscem, w którym szukamy odpowiedzi na pytanie „kto i dlaczego zrestartował ten serwer" — pokazuje proces inicjujący, konto użytkownika i deklarowany powód.

Najważniejsze dzienniki i identyfikatory, których szukamy najczęściej

Dziennik	Czego szukamy	Przykładowe Event ID
System	restarty, awarie usług, problemy sterowników	`1074` (zainicjowany restart), `6008` (nieoczekiwane zamknięcie), `41` (Kernel-Power — utrata zasilania/reset bez czystego zamknięcia), `7031` / `7034` (usługa nieoczekiwanie zakończyła działanie)
Application	błędy aplikacji, awarie procesów	`1000` (awaria aplikacji), `1026` (nieobsłużony wyjątek .NET)
Setup	przebieg instalacji aktualizacji i komponentów	zdarzenia CBS/DISM bez ustalonego jednego ID — przeglądamy chronologicznie wokół czasu instalacji
Security	logowania, zmiany uprawnień, blokady kont	`4624` (udane logowanie), `4625` (nieudane logowanie), `4740` (zablokowane konto)

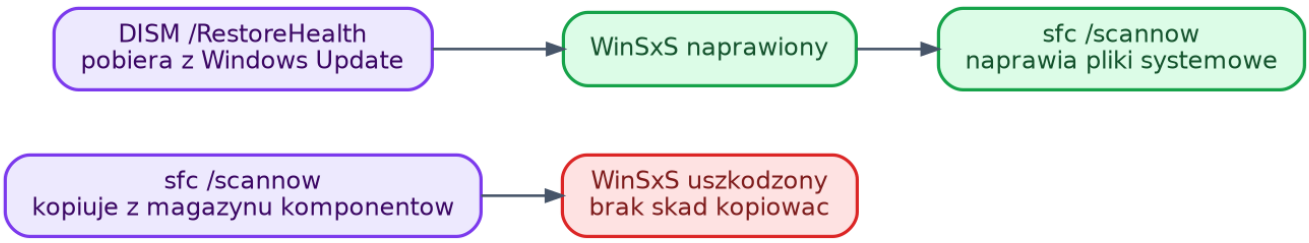
Filtrowanie po zakresie czasu i poziomie istotności

```
Get-WinEvent -FilterHashtable @{LogName='System'; Level=1,2; StartTime=(Get-Date).AddHours(-24)} | Format-Table TimeCreated, Id, LevelDisplayName, Message -Wrap
```

`Level=1,2` ogranicza wynik do zdarzeń krytycznych i błędów, pomijając ostrzeżenia i informacje — przydatne, kiedy przeglądamy dziennik po awarii i nie chcemy przekopywać się przez setki nieistotnych wpisów.

Weryfikacja integralności systemu: sfc i DISM

Kolejność ma znaczenie. `sfc /scannow` naprawia uszkodzone pliki systemowe, ale robi to, kopiując poprawne wersje z lokalnego magazynu komponentów opisanego na stronie [WinSxS: czyszczenie magazynu komponentów]. Jeśli sam ten magazyn jest uszkodzony, `sfc` nie ma skąd wziąć poprawnej kopii i zwróci błąd o niemożności naprawy części plików.



Uszkodzony WinSxS blokuje sfc, DISM /RestoreHealth naprawia magazyn zanim sfc sproboje ponownie (kliknięcie otwiera pełny rozmiar)

```
sfc /scannow
```

Jeżeli `sfc` zgłasza pliki, których nie potrafi naprawić, naprawiamy najpierw magazyn komponentów przez DISM — pobierze brakujące poprawne wersje z Windows Update albo z podanego źródła:

```
DISM /Online /Cleanup-Image /RestoreHealth
```

Dopiero po zakończonym `/RestoreHealth` uruchamiamy `sfc /scannow` ponownie — tym razem ma z czego kopiować.

Zbieranie danych do zgłoszenia

Podsumowanie konfiguracji systemu: Get-ComputerInfo

```
Get-ComputerInfo | Select-Object CsName, WindowsProductName, WindowsVersion,
OsHardwareAbstractionLayer, OsLastBootUpTime
```

Klasyczny odpowiednik: systeminfo

```
systeminfo
```

Przydatny, kiedy zbieramy dane ze skryptu wspólnego dla wielu serwerów albo gdy PowerShell nie jest pod ręką — `Get-ComputerInfo` bywa zauważalnie wolniejszy na starszych systemach.

Częste błędy

Sięganie od razu po `winmgmt /resetrepository` bez wcześniejszej próby `/salvagerepository` — resetujemy wtedy więcej, niż było konieczne, i musimy ręcznie odtwarzać rejestrację klas WMI zainstalowanych aplikacji.

Uruchamianie `sfc /scannow` w kółko bez sprawdzenia, czy problemem nie jest sam magazyn komponentów — bez naprawy przez DISM kolejne przebiegi `sfc` kończą się tym samym błędem.

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

Źródła: notatki własne, dokumentacja Microsoft (`winmgmt`, `lodctr`, DISM Servicing)

Kosz i profile użytkowników

Typ: fiszka **Dotyczy: Windows Server 2012 R2+ (w tym role Remote Desktop Services)**

Efekt: czyścimy kosz wszystkich użytkowników na serwerze, sprawdzamy i usuwamy profile użytkowników, i wiemy, gdzie jeszcze szukać zajętego miejsca poza samym koszem.

Skróty użyte na tej stronie (2)

Skrót	Co znaczy
NTFS	New Technology File System: system plików Windows z wbudowanymi uprawnieniami ACL na poziomie pliku i folderu
RDS	Remote Desktop Services: zdalne pulpity i aplikacje Windows Server

Fiszka

OSTROŻNIE · NIEODWRACALNE

Czyszczenie kosza wszystkich userów

Komendę uruchamiamy jako administrator w cmd/PowerShell:

```
rd /s /q C:/$Recycle.bin
```

`/s` oznacza usunięcie podfolderów, `/q` — tryb cichy, bez pytania o potwierdzenie. Docelowy dysk `C:` zmieniamy na odpowiedni, jeśli kosz do wyczyszczenia jest na innym woluminie.

ZMIENIA SYSTEM

Czyszczenie kosza poleceniem PowerShell: Clear-RecycleBin

```
Clear-RecycleBin -DriveLetter LiteraDysku -Force
```

Odpowiednik natywny dla PowerShella — działa identycznie jak polecenie `rd` wyżej, ale bez wychodzenia poza jeden ekosystem narzędzi, jeśli reszta skryptu jest już w PowerShellu.

◫◫ OSTROŻNIE · NIEODWRACALNE

Czyszczenie kosza dla jednego konkretnego użytkownika

```
rd /s /q "C:\$Recycle.Bin\SidUzytkownika"
```

Przydatne, gdy chcemy posprzątać tylko po jednym koncie (np. po zwolnionym pracowniku, przed usunięciem jego profilu), nie ruszając kosza pozostałych zalogowanych aktualnie użytkowników.

Z życia wzięte

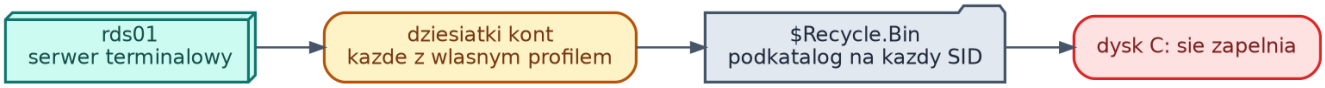
◫◫ Z ŻYCIA WZIĘTE

Dysk systemowy na rds01 spada poniżej pięciu procent wolnego miejsca w piątek po południu, akurat gdy Kasia i trzy inne handlowczynie próbują wydrukować faktury z terminala. Sprawdzamy `Get-CimInstance Win32_UserProfile` i widzimy kilkadziesiąt profili osób, które nie logowały się od miesięcy, każdy z własnym pełnym `$Recycle.Bin`. Po usunięciu nieaktywnych profili przez `Remove-CimInstance` i wyczyszczeniu kosza dysk wraca do normy bez restartu serwera.

$Recycle.Bin na serwerze terminalowym

> **Pojęcie: `$Recycle.Bin`.** Ukryty katalog na każdym woluminie NTFS, wewnątrz którego każdy użytkownik ma własny podkatalog nazwany swoim SID-em — to tam trafiają pliki skasowane „na miękko", zanim zostaną skasowane trwale albo przywrócone.

Na zwykłej stacji roboczej ten katalog rzadko robi się problemem — jeden użytkownik, ograniczona ilość kasowanych danych. Na serwerze terminalowym (RDS) sytuacja jest inna: dziesiątki albo setki kont logują się na ten sam serwer, każde z własnym podkatalogiem w `$Recycle.Bin`, i nikt indywidualnie tego kosza nie czyści, bo z perspektywy użytkownika to nie jego prywatny komputer, o który dba na co dzień. Po miesiącach eksploatacji `$Recycle.Bin` na dysku systemowym serwera RDS potrafi się rozrosnąć do dziesiątek gigabajtów, zanim ktokolwiek to zauważy.



Dziesiatki kont na rds01, kazde z wlasnym podkatalogiem kosza, zapelniaja dysk systemowy (kliknięcie otwiera pełny rozmiar)

Profile użytkowników

> **Pojęcie: profil użytkownika.** Zestaw katalogów i gałęzi rejestru (`NTUSER.DAT`) przechowujący ustawienia i dane danego konta na danym komputerze — pulpit, dokumenty, ustawienia aplikacji, pliki tymczasowe profilu. Na serwerze RDS z profilami lokalnymi (nie roamującymi) każdy zalogowany kiedykolwiek użytkownik zostawia po sobie pełny katalog profilu, nawet jeśli od miesięcy się nie loguje.

Wylistowanie profili użytkowników: Get-CimInstance Win32_UserProfile

```
Get-CimInstance -ClassName Win32_UserProfile | Select-Object LocalPath, LastUseTime, Loaded, Special
```

Przykład:

```
LocalPath                       LastUseTime          Loaded Special
---------                       -----------          ------ -------
C:\Users\jan.kowalski            2024-09-02 08:14:11   False   False
C:\Users\anna.nowak              2023-11-14 16:02:47   False   False
C:\Users\Administrator           2024-11-03 09:00:02    True   False
```

`Special` oznacza profile systemowe (`Administrator`, `Public`, konta usługowe) — te pomijamy przy porządkach, `Loaded` pokazuje, czy profil jest akurat załadowany (użytkownik zalogowany albo trwa jeszcze wygaszanie sesji).

Usunięcie profilu użytkownika

```
Get-CimInstance -ClassName Win32_UserProfile | Where-Object { $_.LocalPath -like
"*NazwaUzytkownika*" -and -not $_.Special -and -not $_.Loaded } | Remove-CimInstance
```

Usunięcie przez `Win32_UserProfile`, a nie ręczne kasowanie katalogu w `C:\Users`, jest ważne — ta klasa czyści przy okazji też powiązany wpis w rejestrze (`ProfileList`). Ręczne skasowanie samego katalogu zostawia sierocy wpis rejestru, który przy następnym logowaniu tego konta na ten sam serwer bywa źródłem błędu o uszkodzonym profilu.

Katalogi tymczasowe

Kosz i profile nie są jedynym miejscem, gdzie na serwerze terminalowym gromadzą się dane bez właściciela. Warto regularnie sprawdzać też:

```
Get-ChildItem "C:\Windows\Temp" -Recurse | Measure-Object -Property Length -Sum
```

oraz katalogi `%TEMP%` w profilach poszczególnych użytkowników — instalatory, tymczasowe pliki aplikacji i logi diagnostyczne aplikacji potrafią tam zalegać miesiącami, zwłaszcza po nieudanych instalacjach czy aktualizacjach.

Częste błędy

Ręczne usuwanie katalogu profilu z `C:\Users` zamiast przez `Win32_UserProfile` — zostawia sierocy wpis w rejestrze i psuje kolejne logowanie tego konta.

Czyszczenie kosza konta, które jest akurat zalogowane (`Loaded = True`) — operacja może się nie udać albo skasować pliki, które użytkownik w tej samej chwili świadomie trzyma w koszu do przywrócenia.

Powiązane: Diagnostyka systemu · RemoteApps w Remote Desktop Services

Źródła: notatki własne