

PowerShell — automatyzacja diagnostyki

Czas: 2 × 45 min

praca indywidualna

Cel: Napisanie skryptu, który codziennie sprawdza stan komputera (miejsce na dyskach, stan dysków, błędy w dziennikach, czas pracy, usługi) i zapisuje wynik do logu z datą, uruchamianego automatycznie przez Harmonogram zadań.

Po zajęciach potrafisz:

- sprawdza w skrypcie wolne miejsce, stan SMART i błędy systemowe
- stosuje instrukcję warunkową do generowania ostrzeżeń
- zapisuje wyniki do pliku logu z datą i godziną
- tworzy i testuje zadanie w Harmonogramie zadań, a po zajęciach je usuwa

Sprzęt i materiały

- komputer stacjonarny z Windows 11 na drugim dysku (z zadania 13, rok 1) — tam masz konto administratora; alternatywnie maszyna wirtualna Windows 11 (z zadania 26, rok 1)
- pendrive ze skrypcem z zadania 2

Oprogramowanie

- **Windows PowerShell 5.1 (wbudowany)** — część poleceń (stan SMART) wymaga uruchomienia jako administrator
- **Harmonogram zadań (taskschd.msc, wbudowany)**
- **Visual Studio Code (opcjonalnie)** — lub PowerShell ISE / Notatnik
code.visualstudio.com/download

Przygotowanie przed zajęciami

- Przy uruchamianiu komputera wybierz Windows z drugiego dysku. Sprawdź, czy możesz uruchomić PowerShell jako administrator.

Bezpieczeństwo

- Skrypt ma tylko odczytywać dane — nie umieszczaj w nim poleceń usuwających pliki ani zatrzymujących usług.
- Zadanie w Harmonogramie twórz wyłącznie w systemie z drugiego dysku lub w maszynie wirtualnej. Przed końcem zajęć usuń je (krok 5), chyba że nauczyciel poleci inaczej.
- Zasady wykonywania skryptów ustawiaj tylko dla procesu (`-ExecutionPolicy Bypass` w argumentach zadania), nie dla całego komputera.

Przebieg zadania

1 Polecenia diagnostyczne

Uruchom PowerShell jako administrator i przetestuj polecenia.

1. Wolne miejsce: `Get-CimInstance Win32_LogicalDisk -Filter 'DriveType=3' | Select-Object DeviceID, @{n='WolneGB';e={[math]::Round($_.FreeSpace/1GB,1)}}, @{n='Procent';e={[math]::Round($_.FreeSpace/$_ .Size*100,1)}}`.
2. Stan dysków: `Get-PhysicalDisk | Select-Object FriendlyName, HealthStatus, OperationalStatus` oraz liczniki: `Get-PhysicalDisk | Get-StorageReliabilityCounter | Select-Object DeviceId, Temperature, Wear, ReadErrorsTotal, PowerOnHours`.
3. Błędy z ostatnich 24 godzin: `Get-WinEvent -FilterHashtable @{LogName='System'; Level=1,2; StartTime=(Get-Date).AddDays(-1)} -MaxEvents 10 -ErrorAction SilentlyContinue`.
4. Czas pracy: `(Get-Date) - (Get-CimInstance Win32_OperatingSystem).LastBootUpTime` oraz stan usług: `Get-Service -Name WinDefend, wuauserv, Spooler, EventLog | Select-Object Name, Status, StartType`.

2 Skrypt diagnostyka.ps1

Zapisz skrypt w `C:\Diag\diagnostyka.ps1`.

1. Utwórz ścieżkę logu z datą: `$log = "C:\Diag\logi\diag_$(Get-Date -Format 'yyyy-MM-dd_HH-mm').log"` i folder `C:\Diag\logi`.
2. Napisz funkcję zapisu: `function Zapisz($tekst) { "$(Get-Date -Format 'HH:mm:ss') $tekst" | Out-File $log -Append -Encoding UTF8 }`.
3. Dla każdego dysku logicznego dodaj warunek: jeśli procent wolnego miejsca jest mniejszy niż 15, zapisz `OSTRZEŻENIE: mało miejsca na C:`, w przeciwnym razie `OK`.
4. Zapisz stan dysków fizycznych; jeśli `HealthStatus` jest inny niż `Healthy`, zapisz ostrzeżenie. Dopisz temperaturę i liczbę błędów odczytu (gdy są dostępne).
5. Zapisz liczbę błędów z dziennika System z ostatnich 24 godzin i treść trzech najnowszych (`TimeCreated`, `ProviderName`, `Id`). Dopisz czas pracy komputera w dniach i godzinach oraz usługi, które nie są uruchomione.
6. Na końcu dopisz podsumowanie: liczbę ostrzeżeń (licznik zwiększany w każdym warunku).

3 Test skryptu

1. Uruchom: `powershell.exe -NoProfile -ExecutionPolicy Bypass -File C:\Diag\diagnostyka.ps1` i otwórz log.
2. Przetestuj ostrzeżenie: tymczasowo zmień próg z 15% na 99% i sprawdź, czy w logu pojawiają się ostrzeżenia. Przywróć próg 15%.

4 Harmonogram zadań

1. Otwórz `taskschd.msc` → Biblioteka Harmonogramu zadań → Utwórz zadanie. Nazwa: `Diagnostyka_<nazwisko>`, zaznacz „Uruchom z najwyższymi uprawnieniami”.
2. Wyzwalacz: codziennie o wybranej godzinie oraz drugi — „Przy logowaniu”. Akcja: program `powershell.exe`, argumenty: `-NoProfile -ExecutionPolicy Bypass -File C:\Diag\diagnostyka.ps1`.
3. Uruchom zadanie ręcznie (prawy przycisk → Uruchom). Sprawdź kolumnę „Wynik ostatniego uruchomienia” (0x0 = sukces) i czy powstał nowy log.
4. Wyloguj się i zaloguj ponownie — sprawdź, czy powstał kolejny log.

5 Porządki

1. Skopiuj skrypt i logi na pendrive.
2. Usuń zadanie z Harmonogramu (lub poleceniem `Unregister-ScheduledTask -TaskName Diagnostyka_<nazwisko>`) i pokaż nauczycielowi, że go nie ma.

W sprawozdaniu umieść

- ☐ pełny kod skryptu `diagnostyka.ps1` z komentarzami
- ☐ przykładowy log z prawidłowym działaniem i log z testem ostrzeżeń
- ☐ zrzut właściwości zadania w Harmonogramie (wyzwalacze i akcja) oraz wyniku ostatniego uruchomienia
- ☐ interpretacja wyników: stan dysków, znalezione błędy systemowe
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Dlaczego `Get-StorageReliabilityCounter` wymaga uprawnień administratora i dlaczego część jego pól bywa pusta?
2. Po co w nazwie pliku logu umieszcza się datę i godzinę w formacie rok-miesiąc-dzień?
3. Co oznacza wynik 0x0 w Harmonogramie zadań i jak sprawdzisz przyczynę, gdy wynik jest inny?
4. Jakie inne sprawdzenia warto dodać do takiego skryptu w szkolnej pracowni?

Kryteria oceny

Kryterium	Punkty
Skrypt sprawdza wszystkie wymagane elementy	6
Warunki i ostrzeżenia działają (test progu)	3
Log z datą, czytelny format	3
Zadanie w Harmonogramie uruchomione i usunięte po zajęciach	4
Interpretacja wyników i odpowiedzi na pytania	4
Razem	20

Sprawozdanie przygotuj na wzorze dołączonym do zadania (plik .docx o tym samym numerze).