


<b>POLITECHNIKA WROCŁAWSKA</b>  Wydział Informatyki i Telekomunikacji	Wydział: Informatyki i Telekomunikacji Kierunek: Cyberbezpieczeństwo Rok Akademicki: 2024/2025 Rok studiów, semestr: I, 2 Grupa: 2 Termin: <i>poniedziałek, godz. 15.15</i>
<b>Programowanie skryptowe – Laboratorium 7</b>	
Prowadzący: mgr inż. Karolina Pfajfer Data wykonania ćwiczenia: <i>14.04.2025</i> Data oddania sprawozdania: <i>23.04.2025</i>	Autor: <i>Adam Dąbrowski, 283832</i>

<https://learn.microsoft.com/en-us/powershell/scripting/developer/help/examples-of-comment-based-help?view=powershell-7.5>

kolokwium koncowe

zalozyc wirtualke z Windowsem

1. Założyć konto na GitHubie + dołączyć skrypty
2. Przećwiczyć gita - <https://learnkitbranching.js.org/?locale=pl>
3. Założyć maszynę wirtualną z Windowsem
4. Pobrać Visual Studio Code
5. Dodać Get-Help do skryptów
6. Pamiętać o nazwie skryptów, przykładowo: Monitor-Process

## Część Praktyczna

1. Napisz skrypt PowerShell, który monitoruje określony folder i automatycznie przenosi nowo dodane pliki .txt do innej lokalizacji.

a) Jeżeli folder docelowy nie istnieje, skrypt musi go utworzyć.

b) Skrypt powinien działać ciągle, aż do jego ręcznego wyłączenia.

```
1  # Ścieżki folderów
2  $watchPath = "C:\Users\PC\Desktop\Prog-Skryptowe\monitorowany"
3  $destinationPath = "C:\Users\PC\Desktop\Prog-Skryptowe\kopiowane"
4
5  # tworzenie pliku docelowego jeśli istnieje
6  if (-not (Test-Path $destinationPath)) {
7      New-Item -Path $destinationPath -ItemType Directory
8      Write-Host "Utworzono folder docelowy: $destinationPath"
9      Write-Host "1"
10 }
11
12 # tworzymy obiekt watcher
13 $swatcher = New-Object System.IO.FileSystemWatcher
14 $swatcher.Path = $watchPath
15 $swatcher.Filter = "*.txt"
16 $swatcher.EnableRaisingEvents = $true
17 $swatcher.IncludeSubdirectories = $false
18
19 # Funkcja do obsługi zdarzenia dodania pliku
20
21 write-host "$destinationPath"
22
23 $onCreated = Register-ObjectEvent $swatcher Created -Action {
24     Start-Sleep -Seconds 1
25     Write-Host "3"
26     Write-Host "a"
27     write-host "$using"
28     Write-Host "a $using:destinationPath"
29
30     $sourcePath = $Event.SourceEventArgs.FullPath
31     Write-Host "$sourcePath"
32     $fileName = $Event.SourceEventArgs.Name
33     write-host "$fileName"
34     $destFile = Join-Path -Path $using:destinationPath -ChildPath $fileName
35     write-host "$destinationPath"
36     write-host $destFile
37     Move-Item -Path $sourcePath -Destination $destFile
38     Write-Host "Przeniesiono plik: $fileName"
39 }
40
41 Write-Host "Monitorowanie folderu: $watchPath. Naciśnij Ctrl+C, aby zatrzymać."
42
43 # Nieskończona pętla
44 while ($true) {
45     Start-Sleep -Seconds 1
46 }
47
48
49
50 |
```

Niestety program nie działa ponieważ nie jestem w stanie przekazać zmiennej `destinationpath` do środka `onCreated`

2. Napisz skrypt w PowerShell który:

a) Obliczy sumę kontrolną pliku (MD5 lub SHA256).

b) Wyśle zapytanie do API VirusTotal

c) Zinterpretuje odpowiedź API i wyświetli informację, czy plik jest bezpieczny, czy nie.

```
PS C:\> .\Users\PC\Desktop\Prog-Skryptowe\skan-plikow.ps1
Get-FileHash : The file 'C:\Users\PC\Desktop\Prog-Skryptowe\monitorowany\eicar.txt' cannot be read: Operacja nie zakończyła się pomyślnie, ponieważ plik zawiera wirusa lub potencjalnie niechciane oprogramowanie.
At C:\Users\PC\Desktop\Prog-Skryptowe\Skan-plikow.ps1:8 char:11
+ $sha256 = Get-FileHash -Path $filePath -Algorithm SHA256
+ ~~~~~
+ CategoryInfo          : ReadError: (C:\Users\PC\Des...owany\eicar.txt:PS
  Object) [Write-Error], WriteErrorException
+ FullyQualifiedErrorId : FileReadError,Get-FileHash

SHA256:
Invoke-RestMethod : {"error":{"code":"NotFoundError","message":"Resource not found."}}
At C:\Users\PC\Desktop\Prog-Skryptowe\Skan-plikow.ps1:19 char:13
+ $response = Invoke-RestMethod -Uri $vtUrl -Headers $headers -Method G ...
+ ~~~~~
+ CategoryInfo          : InvalidOperation: (System.Net.HttpWebRequest:Http
  pWebRequest) [Invoke-RestMethod], WebException
+ FullyQualifiedErrorId : WebCmdletWebResponseException,Microsoft.PowerShe
  ll.Commands.InvokeRestMethodCommand
```

W przypadku stworzenia pliku `EICAR` windows defender od razu usuwa plik a nawet jeśli zdążymy wykonać program przed jego usunięciem nie będziemy mogli tego zrobić.

```
PS C:\> .\Users\PC\Desktop\Prog-Skryptowe\skan-plikow.ps1
SHA256: E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
Wykrycia złośliwego kodu: 0 z 60
Plik wygląda na bezpieczny.

1 # klucz aApi
2 $apiKey = "83df948511c8fd060677f4035f11ed6e79335c990ef852592638e1956f7aa0d4"
3
4 # Ścieżka do pliku
5 $filePath = "C:\Users\PC\Desktop\Prog-Skryptowe\monitorowany\nowy.txt"
6
7 # obliczamy sumę kontrolną SHA256
8 $sha256 = Get-FileHash -Path $filePath -Algorithm SHA256
9 $hashValue = $sha256.Hash
10 Write-Host "SHA256: $hashValue"
11
12 # wysyłamy zapytanie do virustotal
13 $url = "https://www.virustotal.com/api/v3/files/$hashValue"
14 $headers = @{"x-apikey" = $apiKey}
15
16 # wysłanie zapytania
17 $response = Invoke-RestMethod -Uri $url -Headers $headers -Method GET
18
19 # sprawdzamy liczbę detekcji
20 $malicious = $response.data.attributes.last_analysis_stats.malicious
21 $total = $response.data.attributes.last_analysis_stats.harmless + $malicious + $response.data.attributes.last_analysis_stats.undetected
22
23 #generujemy odpowiedź dla użytkownika
24 Write-Host "Wykrycia złośliwego kodu: $malicious z $total"
25
26 if ($malicious -gt 0) {
27     Write-Host "Plik jest niebezpieczny"
28 } else {
29     Write-Host "Plik wygląda na bezpieczny."
30 }
```

Sprawdzić, czy wyniki są zgodne z oczekiwaniami - pobrać plik `EICAR` oraz

utworzyć nowy plik testowy. Każdy etap skryptu powinien być opatrzony komentarzem.