Ten skrypt PowerShell, o nazwie "Unprotect-FilesOnLogicalDisk_v3.ps1," ma na celu znalezienie i zdekodowanie plików i folderów zaszyfrowanych za pomocą EFS (Encrypting File System) na wszystkich logicznych dyskach komputera. Poniżej znajduje się wyjaśnienie jego działania:

1. **Wersja PowerShell:** Skrypt wymaga co najmniej PowerShell w wersji 3.0. Jeśli używana jest wersja 2.0, zaleca się skorzystanie z wcześniejszej wersji skryptu dostępnej na stronie GitHub autora.

```
#Requires -version 3.0
```

2. **Sprawdzenie katalogu dziennika (`Logs`):** Skrypt rozpoczyna sprawdzanie, czy istnieje katalog `C:\Logs`. Jeśli go nie ma, zostaje utworzony.

```
If (-not(Test-Path -Path C:\Logs))
{
    New-Item -Path C: -Name Logs -ItemType directory
}
```

3. **Pobranie daty i utworzenie pliku dziennika (`EfsDecryptLog_YYYY-MM-DD.log`):** Skrypt pobiera aktualną datę i tworzy plik dziennika w katalogu `C:\Logs` z nazwą zawierającą aktualną datę.

```
$today = Get-Date
$filename = $today.ToString("yyyy-MM-dd")
$logFile = "C:\Logs\EfsDecryptLog_$filename.log"
```

4. **Pobranie wszystkich logicznych dysków:** Skrypt używa WMI (Windows Management Instrumentation), aby pobrać informacje o wszystkich logicznych dyskach i zapisuje je do zmiennej `$drive`.

```
$drive = Get-WmiObject Win32_logicaldisk | Select-Object -ExpandProperty deviceID
Add-Content $logFile "$today Found the following drives: $drive"
```

5. **Znalezienie zaszyfrowanych plików:** Skrypt przeszukuje wszystkie dyski w poszukiwaniu plików o atrybucie "Encrypted" i zapisuje ich pełne ścieżki do zmiennej `$encryptedfiles`.

```
$encryptedfiles += foreach ($d in $drive) {
    Get-ChildItem $d -File -Recurse -Force -ErrorAction SilentlyContinue |
    Where-Object { $_.Attributes -match "Encrypted" } |
    Select-Object -ExpandProperty FullName
```

```
}
```

6. **Dekodowanie znalezionych plików:** Skrypt próbuje zdekodować każdy znaleziony plik przy użyciu metody `.Decrypt()` i zapisuje informacje o każdym pliku w pliku dziennika.

```
foreach ($file in $encryptedfiles) {
   try {
      (Get-Item -Force -LiteralPath $file).Decrypt()
      Add-Content $logFile "$file decrypted"
   }
   catch [Exception] {
      Add-Content $logFile "ERROR: Decrypting $file failed. Error message: $_.Exception.ToString()"
   }
}
```

7. **Znalezienie i dekodowanie zaszyfrowanych folderów:** Skrypt przeszukuje wszystkie dyski w poszukiwaniu folderów o atrybucie "Encrypted" i używa narzędzia `cipher.exe` do ich dekodowania. Informacje o każdym zdekodowanym folderze są zapisywane w pliku dziennika.

```
$encryptedfolders += foreach ($d in $drive) {
   Get-ChildItem $d -Directory -Recurse -Force -ErrorAction SilentlyContinue |
   Where-Object { $_.Attributes -match "Encrypted" } |
   Select-Object -ExpandProperty FullName
}

foreach ($folder in $encryptedfolders) {
   try {
      cipher.exe /d /i $folder
      Add-Content $logFile "$folder decrypted"
   }
   catch [Exception] {
      Add-Content $logFile "ERROR: Decrypting $folder failed. Error message:
$_.Exception.ToString()"
   }
```

}

8. **Zapisanie podsumowania w pliku dziennika:** Na końcu skrypt dodaje podsumowanie operacji do pliku dziennika.

Add-Content $logFile ""

Add-Content $logFile "Finished decrypting folders"

Write-Host "Finished decrypting folders"

Add-Content $logFile ""

Add-Content $logFile ""

Add-Content $logFile "===END of script==="

Write-Host "===End of script==="

Skrypt informuje użytkownika o postępie operacji i zapisuje wszystkie komunikaty w pliku dziennika, który znajduje się w katalogu `C:\Logs`. W przypadku wystąpienia błędów podczas dekodowania, informacje o błędzie są również rejestrowane w pliku dziennika.