SCRIPT #05 · CYBER SECURITY

Eventi di login falliti (rilevamento bruteforce)

Analizza il log di sicurezza per individuare pattern di tentativi di accesso falliti ripetuti.

AUTORE	COPYRIGHT	PREREQUISITI
Francesco Larossa	© 2026 SysAdmin Italia	Permessi di lettura sul log di sicurezza

Cosa fa questo script

Lo script interroga il log di sicurezza di Windows (Event ID 4625, accesso fallito) nelle ultime N ore, raggruppa i tentativi per account e per indirizzo IP di origine, e segnala le combinazioni con un numero di tentativi superiore alla soglia configurata, un pattern tipico di attacchi bruteforce.

Come si usa

Esegui lo script specificando quante ore indietro analizzare e la soglia di tentativi da considerare sospetta.

```
.\Report-LoginFalliti.ps1 -OreIndietro 24 -SogliaTentativi 10
```

In quale contesto usarlo

Da schedulare come controllo ricorrente, idealmente ogni poche ore su sistemi esposti: un attacco bruteforce lascia tracce chiarissime nel log di sicurezza molto prima che abbia successo, e intercettarlo per tempo permette di bloccare l'IP o l'account coinvolto prima che il tentativo vada a buon fine.

Codice completo

Il codice è lo stesso del file Report-LoginFalliti.ps1: per usarlo scarica il file dalla pagina invece di copiarlo da questo PDF.

```
<#
.SYNOPSIS
    Analizza il log di sicurezza per individuare pattern di tentativi di accesso falliti ripetuti.
#>
param(
    [int]$OreIndietro = 24,
    [int]$SogliaTentativi = 10,
    [string]$PercorsoCSV = "C:\Report\LoginFalliti.csv"
)

$dataLimite = (Get-Date).AddHours(-$OreIndietro)

$eventi = Get-WinEvent -FilterHashtable @{LogName='Security'; Id=4625; StartTime=$dataLimite} -ErrorAction SilentlyContinue

$raggruppati = $eventi | ForEach-Object {
    $xml = [xml]$_.ToXml()
    $dati = $xml.Event.EventData.Data
    [PSCustomObject]@{
        Account = ($dati | Where-Object {$_.Name -eq 'TargetUserName'}).'#text'
        IPOrigine = ($dati | Where-Object {$_.Name -eq 'IpAddress'}).'#text'
        Orario = $_.TimeCreated
    }
} | Group-Object Account, IPOrigine | Where-Object { $_.Count -ge $SogliaTentativi }
```

```
$risultati = $raggruppati | Select-Object @{N="AccountEIP";E={$_.Name}}, Count,
    @{N="PrimoTentativo";E={($_.Group | Sort-Object Orario | Select-Object -First 1).Orario}},
    @{N="UltimoTentativo";E={($_.Group | Sort-Object Orario -Descending | Select-Object -First 1).Orario}}

$risultati | Export-Csv -Path $PercorsoCSV -NoTypeInformation -Encoding UTF8 -Delimiter ";"
Write-Host "Trovate $($risultati.Count) combinazioni account/IP con oltre $SogliaTentativi tentativi falliti." `
    -ForegroundColor $(if($risultati.Count -gt 0){"Red"}else{"Green"})
```