Script surpervison v3

Ajout de tentative de redemarage de services critiques

<#

.SYNOPSIS

Supervision des événements, processus et services critiques avec export des logs et alerte Teams.

Inclut planification et tentative de redémarrage de services critiques en cas de panne.

#>

# === Paramètres utilisateur ===

$Interval = Read-Host "Entrez l'intervalle en minutes entre chaque vérification"

$CriticalServices = @("w32time","wuauserv","lanmanserver","Dnscache","Dhcp")

$TeamsWebhook = "http://teams/webhoook-entrant/fictif"

# === Création du dossier logs ===

$LogDir = "$env:USERPROFILE\Desktop\logs"

if (-not (Test-Path $LogDir)) { New-Item -ItemType Directory -Path $LogDir }

# === Boucle principale ===

while ($true) {

$Timestamp = (Get-Date).ToString("ddMMyyyy\_HHmmss")

$LogFile = "$LogDir\Supervision\_$Timestamp.csv"

$Errors = @()

# 1. Gestion des événements : 10 dernières erreurs critiques

$EventErrors = Get-WinEvent -LogName System -ErrorAction SilentlyContinue |

Where-Object { $\_.LevelDisplayName -eq "Critical" } |

Select-Object TimeCreated, Id, Message -First 10

# 2. Gestion des processus : top 5 en mémoire

$TopProcesses = Get-Process | Sort-Object -Property WorkingSet -Descending |

Select-Object -First 5 Name, Id, CPU, WS

# 3. Gestion des services : vérification + redémarrage

$ServiceStatus = foreach ($svc in $CriticalServices) {

$service = Get-Service -Name $svc -ErrorAction SilentlyContinue

if ($service.Status -ne "Running") {

$attempt = 0

do {

try {

Start-Service -Name $svc -ErrorAction Stop

Start-Sleep -Seconds 3

} catch {}

$service.Refresh()

$attempt++

} while ($service.Status -ne "Running" -and $attempt -lt 3)

$msg = if ($service.Status -eq "Running") {

"Service $svc redémarré avec succès après $attempt tentative(s)."

} else {

"Échec du redémarrage du service $svc après 3 tentatives."

}

$Errors += $msg

}

[PSCustomObject]@{

ServiceName = $svc

Status = $service.Status

}

}

# Export CSV

$Report = [PSCustomObject]@{

Timestamp = $Timestamp

CriticalEvents = $EventErrors.Count

RunningServices = ($ServiceStatus | Where-Object {$\_.Status -eq "Running"}).Count

TopProcesses = ($TopProcesses | ForEach-Object { $\_.Name }) -join ", "

}

$Report | Export-Csv -Path $LogFile -NoTypeInformation -Append

# Suppression doublons

$Lines = Get-Content $LogFile | Select-Object -Unique

$Lines | Set-Content $LogFile

# Envoi alerte Teams si erreur

if ($Errors.Count -gt 0 -or $EventErrors.Count -gt 0) {

$Body = @{

title = "Alerte Supervision"

text = "Problème détecté : $($Errors -join '; ')"

} | ConvertTo-Json -Depth 3

Invoke-RestMethod -Uri $TeamsWebhook -Method Post -Body $Body -ContentType 'application/json'

}

# Pause avant prochaine vérification

Start-Sleep -Seconds ($Interval \* 60)

}