Scripts para Bloqueio Permanente de Aplicativos Indesejados no Windows

Este arquivo contém os scripts mencionados no guia completo para implementação da solução em camadas para bloqueio permanente de aplicativos indesejados no Windows.

EnhancedPolicyEnforcer.ps1

```
 # Script de Políticas de Registro Aprimoradas para Bloqueio de Apps Indesejados
# Salvar como EnhancedPolicyEnforcer.ps1

# Executar como administrador
if (-NOT ([Security.Principal.WindowsPrincipal]
[Security.Principal.WindowsIdentity]::GetCurrent()).IsInRole([Security.Principal.WindowsBuiltInR
"Administrator")) {
   Write-Warning "Por favor, execute como administrador!"
   Break
}

# Registrar data e hora da execução
$logDate = Get-Date -Format "yyyy-MM-dd HH:mm:ss"
Write-Output "Aplicando políticas em: $logDate" | Out-File -Append -FilePath
"$env:USERPROFILE\AppData\Local\PolicyEnforcer_log.txt"

# === Políticas existentes ===
# Bloquear Microsoft Store
New-Item -Path "HKLM:\Software\Policies\Microsoft\WindowsStore" -Force | Out-
Null
Set-ItemProperty -Path "HKLM:\Software\Policies\Microsoft\WindowsStore" -Name
"RemoveWindowsStore" -Type DWord -Value 1
Set-ItemProperty -Path "HKLM:\Software\Policies\Microsoft\WindowsStore" -Name
"AutoDownload" -Type DWord -Value 2

# Impedir sugestões de apps no Menu Iniciar
New-Item -Path "HKLM:\Software\Policies\Microsoft\Windows\CloudContent" -
Force | Out-Null
Set-ItemProperty -Path "HKLM:
\Software\Policies\Microsoft\Windows\CloudContent" -Name
"DisableWindowsConsumerFeatures" -Type DWord -Value 1

# Impedir reinstalação de apps automáticos
New-Item -Path "HKLM:
\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer" -Force | Out-Null
```

```
Set-ItemProperty -Path "HKLM:
\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer" -Name
"NoAutoInstallProgram" -Type DWord -Value 1

# Desativar reinstalação de apps recomendados silenciosos
New-Item -Path "HKLM:
\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Force |
Out-Null
Set-ItemProperty -Path "HKLM:
\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"SilentInstalledAppsEnabled" -Type DWord -Value 0
Set-ItemProperty -Path "HKLM:
\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"SubscribedContent-338387Enabled" -Type DWord -Value 0
Set-ItemProperty -Path "HKLM:
\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"SubscribedContent-338388Enabled" -Type DWord -Value 0
Set-ItemProperty -Path "HKLM:
\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"SubscribedContent-353694Enabled" -Type DWord -Value 0
Set-ItemProperty -Path "HKLM:
\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"SubscribedContent-353696Enabled" -Type DWord -Value 0

# Bloquear OneDrive
New-Item -Path "HKLM:\Software\Policies\Microsoft\Windows\OneDrive" -Force |
Out-Null
Set-ItemProperty -Path "HKLM:\Software\Policies\Microsoft\Windows\OneDrive" -
Name "DisableFileSyncNGSC" -Type DWord -Value 1

# Minimizar envio de dados para a Microsoft
New-Item -Path "HKLM:
\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\DataCollection" -Force |
Out-Null
Set-ItemProperty -Path "HKLM:
\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\DataCollection" -Name
"AllowTelemetry" -Type DWord -Value 0

# === Políticas adicionais ===
# Desativar dicas e sugestões do Windows
New-Item -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\CloudContent" -
Force | Out-Null
Set-ItemProperty -Path "HKLM:
\SOFTWARE\Policies\Microsoft\Windows\CloudContent" -Name
"DisableSoftLanding" -Type DWord -Value 1
Set-ItemProperty -Path "HKLM:
\SOFTWARE\Policies\Microsoft\Windows\CloudContent" -Name
"DisableWindowsSpotlightFeatures" -Type DWord -Value 1
Set-ItemProperty -Path "HKLM:
\SOFTWARE\Policies\Microsoft\Windows\CloudContent" -Name
"DisableWindowsConsumerFeatures" -Type DWord -Value 1
```

```
# Desativar instalação automática de aplicativos sugeridos
New-Item -Path "HKLM:
\SOFTWARE\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Force |
Out-Null
Set-ItemProperty -Path "HKLM:
\SOFTWARE\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"PreInstalledAppsEnabled" -Type DWord -Value 0
Set-ItemProperty -Path "HKLM:
\SOFTWARE\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"PreInstalledAppsEverEnabled" -Type DWord -Value 0
Set-ItemProperty -Path "HKLM:
\SOFTWARE\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"OEMPreInstalledAppsEnabled" -Type DWord -Value 0

# Desativar reinstalação automática de aplicativos removidos
Set-ItemProperty -Path "HKLM:
\SOFTWARE\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"ContentDeliveryAllowed" -Type DWord -Value 0
Set-ItemProperty -Path "HKLM:
\SOFTWARE\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"SystemPaneSuggestionsEnabled" -Type DWord -Value 0

# Desativar sugestões na tela de bloqueio
Set-ItemProperty -Path "HKLM:
\SOFTWARE\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"RotatingLockScreenEnabled" -Type DWord -Value 0
Set-ItemProperty -Path "HKLM:
\SOFTWARE\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"RotatingLockScreenOverlayEnabled" -Type DWord -Value 0

# Desativar sugestões no menu Iniciar
Set-ItemProperty -Path "HKLM:
\SOFTWARE\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"SubscribedContent-338393Enabled" -Type DWord -Value 0
Set-ItemProperty -Path "HKLM:
\SOFTWARE\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"SubscribedContent-353694Enabled" -Type DWord -Value 0
Set-ItemProperty -Path "HKLM:
\SOFTWARE\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"SubscribedContent-353696Enabled" -Type DWord -Value 0

# Desativar sugestões de aplicativos no menu Iniciar
Set-ItemProperty -Path "HKLM:
\SOFTWARE\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"SubscribedContent-314559Enabled" -Type DWord -Value 0
Set-ItemProperty -Path "HKLM:
\SOFTWARE\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"SubscribedContent-314563Enabled" -Type DWord -Value 0

# Aplicar as mesmas configurações para o usuário atual
$registryPaths = @(
    "HKCU:\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager",
```

```
    "HKCU:\Software\Policies\Microsoft\Windows\CloudContent"
)

foreach ($path in $registryPaths) {
    if (-not (Test-Path $path)) {
        New-Item -Path $path -Force | Out-Null
    }
}

# Aplicar configurações para o usuário atual
Set-ItemProperty -Path "HKCU:
\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"SilentInstalledAppsEnabled" -Type DWord -Value 0
Set-ItemProperty -Path "HKCU:
\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"ContentDeliveryAllowed" -Type DWord -Value 0
Set-ItemProperty -Path "HKCU:
\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"PreInstalledAppsEnabled" -Type DWord -Value 0
Set-ItemProperty -Path "HKCU:
\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"PreInstalledAppsEverEnabled" -Type DWord -Value 0
Set-ItemProperty -Path "HKCU:
\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"OEMPreInstalledAppsEnabled" -Type DWord -Value 0
Set-ItemProperty -Path "HKCU:
\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"SubscribedContent-338388Enabled" -Type DWord -Value 0
Set-ItemProperty -Path "HKCU:
\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"SubscribedContent-338389Enabled" -Type DWord -Value 0
Set-ItemProperty -Path "HKCU:
\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -Name
"SystemPaneSuggestionsEnabled" -Type DWord -Value 0

Write-Output "Políticas aplicadas com sucesso!"
```

PolicyMonitor.ps1

```
 # Script de Monitoramento e Reaplicação de Políticas
# Salvar como PolicyMonitor.ps1

# Função para verificar se as políticas estão ativas
function Test-Policies {
    $policies = @(
        @{Path="HKLM:\Software\Policies\Microsoft\WindowsStore";
Name="RemoveWindowsStore"; ExpectedValue=1},
        @{Path="HKLM:\Software\Policies\Microsoft\WindowsStore";
Name="AutoDownload"; ExpectedValue=2},
```

```
        @{Path="HKLM:\Software\Policies\Microsoft\Windows\CloudContent";
Name="DisableWindowsConsumerFeatures"; ExpectedValue=1},
        @{Path="HKLM:
\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer";
Name="NoAutoInstallProgram"; ExpectedValue=1},
        @{Path="HKLM:
\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager";
Name="SilentInstalledAppsEnabled"; ExpectedValue=0},
        @{Path="HKLM:\Software\Policies\Microsoft\Windows\OneDrive";
Name="DisableFileSyncNGSC"; ExpectedValue=1}
    )

    $failedPolicies = @()

    foreach ($policy in $policies) {
        if (-not (Test-Path $policy.Path)) {
            $failedPolicies += "$($policy.Path)\$($policy.Name)"
            continue
        }

        $value = Get-ItemProperty -Path $policy.Path -Name $policy.Name -
ErrorAction SilentlyContinue
        if ($null -eq $value -or $value.$($policy.Name) -ne $policy.ExpectedValue) {
            $failedPolicies += "$($policy.Path)\$($policy.Name)"
        }
    }

    return $failedPolicies
}

# Verificar políticas
$failedPolicies = Test-Policies

# Registrar resultados
$logDate = Get-Date -Format "yyyy-MM-dd HH:mm:ss"
$logPath = "$env:USERPROFILE\AppData\Local\PolicyMonitor_log.txt"

if ($failedPolicies.Count -gt 0) {
    Write-Output "[$logDate] Políticas alteradas ou ausentes detectadas:" | Out-File -
Append -FilePath $logPath
    $failedPolicies | ForEach-Object { Write-Output "- $_" | Out-File -Append -
FilePath $logPath }

    # Reaplicar políticas
    Write-Output "[$logDate] Reaplicando políticas..." | Out-File -Append -FilePath
$logPath

    # Executar o script de políticas
    $policyScript = "$PSScriptRoot\EnhancedPolicyEnforcer.ps1"
    if (Test-Path $policyScript) {
        Start-Process powershell.exe -ArgumentList "-ExecutionPolicy Bypass -File
`"$policyScript`"" -Verb RunAs -Wait
```

```
        Write-Output "[$logDate] Políticas reaplicadas com sucesso." | Out-File -
Append -FilePath $logPath
    } else {
        Write-Output "[$logDate] ERRO: Script de políticas não encontrado em
$policyScript" | Out-File -Append -FilePath $logPath
    }
} else {
    Write-Output "[$logDate] Verificação concluída: Todas as políticas estão ativas." |
Out-File -Append -FilePath $logPath
}
```

SetupScheduledTask.ps1

```
 # Script para Configurar Tarefa Agendada
# Salvar como SetupScheduledTask.ps1

# Executar como administrador
if (-NOT ([Security.Principal.WindowsPrincipal]
[Security.Principal.WindowsIdentity]::GetCurrent()).IsInRole([Security.Principal.WindowsBuiltInR
"Administrator")) {
    Write-Warning "Por favor, execute como administrador!"
    Break
}

# Definir caminhos
$scriptDir = "$env:USERPROFILE\Documents\WindowsPolicies"
$monitorScript = "$scriptDir\PolicyMonitor.ps1"
$enforcerScript = "$scriptDir\EnhancedPolicyEnforcer.ps1"

# Criar diretório se não existir
if (-not (Test-Path $scriptDir)) {
    New-Item -Path $scriptDir -ItemType Directory -Force | Out-Null
}

# Verificar se os scripts existem no diretório atual
if (Test-Path ".\PolicyMonitor.ps1") {
    Copy-Item ".\PolicyMonitor.ps1" -Destination $monitorScript -Force
}

if (Test-Path ".\EnhancedPolicyEnforcer.ps1") {
    Copy-Item ".\EnhancedPolicyEnforcer.ps1" -Destination $enforcerScript -Force
}

# Criar tarefa agendada para verificação diária
$taskName = "VerificarPoliticasWindows"
$taskDescription = "Verifica e reaplicar políticas de bloqueio de aplicativos
indesejados"

# Remover tarefa existente, se houver
```

```powershell
Unregister-ScheduledTask -TaskName $taskName -Confirm:$false -ErrorAction SilentlyContinue

# Criar ação
$action = New-ScheduledTaskAction -Execute "powershell.exe" -Argument "-ExecutionPolicy Bypass -File `"$monitorScript`""

# Criar gatilhos
# 1. Diariamente
$triggerDaily = New-ScheduledTaskTrigger -Daily -At 9AM

# 2. Na inicialização
$triggerStartup = New-ScheduledTaskTrigger -AtStartup

# 3. Após instalação de atualizações
$triggerUpdate = New-ScheduledTaskTrigger -AtLogOn

# Configurar principal (usuário que executa a tarefa)
$principal = New-ScheduledTaskPrincipal -UserId "SYSTEM" -LogonType ServiceAccount -RunLevel Highest

# Criar configurações
$settings = New-ScheduledTaskSettingsSet -AllowStartIfOnBatteries -DontStopIfGoingOnBatteries -StartWhenAvailable -RunOnlyIfNetworkAvailable

# Registrar tarefa
Register-ScheduledTask -TaskName $taskName -Description $taskDescription -Trigger @($triggerDaily, $triggerStartup, $triggerUpdate) -Action $action -Principal $principal -Settings $settings

# Executar a tarefa imediatamente para verificar
Start-ScheduledTask -TaskName $taskName

Write-Output "Tarefa agendada configurada com sucesso!"
Write-Output "A verificação de políticas será executada diariamente às 9h, na inicialização do sistema e após a instalação de atualizações."
```