



 main ▾


 


Go to file

<> Code ▾

 v1

 v2

 README.md

 README

SysmonEoP


Proof of Concept for arbitrary file delete/write in Sysmon (CVE-2022-41120/CVE-2022-44704)


Vulnerability


Vulnerability is in code responsible for ClipboardChange event that can be reached through RPC. Local users can send data to RPC server which will then be written in C:\Sysmon directory (default ArchiveDirectory) and deleted afterwards. In version before 14.11 Sysmon would not check if directory was created by low privilege user or if it's a junction which can be abused to perform arbitrary file delete/write (kinda limited as you can


About


No description, website, or topics provided.

 Readme

 Activity

 181 stars

 5 watching

 31 forks

Report repository

Releases

No releases published

Packages

No packages published

Languages

C 62.1%

C++ 37.9%

Page 1 of 4

only write strings) in context of NT AUTHORITY\SYSTEM user. In version 14.11/14.12, after initial fix, Sysmon would check if directory exists and would refuse to write/delete files if directory exists. This patch was bypassed by letting Sysmon create C:\Sysmon directory first (using CreateDirectory API) and opening handle on it before SetFileSecurity is called and change DACL's on C:\Sysmon directory.

Exploitation

All testing was done on Windows 10.

In my PoC I have chained arbitrary file delete/write to first delete setup information file of printer driver and then write modified .INF file (as spooler service is enabled by default and low privilege users can re-install printer drivers on windows clients). Setup information files can be abused to perform all kind of operations such service creation, registry modification, file copy etc. I choose to copy some of printer default DLL's in c:\windows\system32 and set permissions on it so that low privilege users can modify it, this is done using CopyFiles directive (<https://learn.microsoft.com/en-us/windows-hardware/drivers/install/inf-copyfiles-directive>). Once file is copied it is overwritten with DLL that will spawn elevated cmd.exe process. It is possible to abuse just arbitrary file delete for LPE by abusing windows installer behavior (trick found by [@KLINIX5](#) and is documented by ZDI here <https://www.zerodayinitiative.com/blog/2022/3/16/abusing-arbitrary-file-deletes-to-escalate-privilege-and-other-great-tricks>).

Vulnerable versions and pre-requirements

All testing was done on versions 13.34-14.12. I don't know exactly lowest version that is vulnerable, but I believe that

versions 12.0 - 14.12 are vulnerable as ClipboardChange event was introduced in version 12.0. In order to exploit this vulnerability events that use ArchiveDirectory should not be enabled (ClipboardChange and FileDelete I believe) as if those two are used then ArchiveDirectory will be created and have secure permissions.

Workaround

If you are using vulnerable version and cannot update you can create ArchiveDirectory (C:\Sysmon by default) and set permissions that will only allow access to NT AUTHORITY\SYSTEM account.

Timeline

- 2022/06/13 - Vulnerability reported to Microsoft
- 2022/06/16 - Vulnerability confirmed.
- 2022/11/08 - Patch and CVE released.
- 2022/11/08 - Bypass reported to Microsoft.
- 2022/11/11 - Microsoft cannot reproduce vulnerability, asks for different PoC.
- 2022/11/11 - I send same PoC and suggest that sysmon is either not installed on testing VM or installation was corrupted.
- 2022/11/15 - Microsoft confirmed bypass.
- 2022/11/28 - Microsoft release v14.13 that patched vulnerability (CVE will be released in December Patch Tuesday)

Links & Resources

- <https://itm4n.github.io/fuzzing-windows-rpc-rpcview/>

- <https://www.zerodayinitiative.com/blog/2022/3/16/abusing-arbitrary-file-deletes-to-escalate-privilege-and-other->

[Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact](#) [Manage cookies](#) [Do not share my personal information](#)



© 2024 GitHub, Inc.