hfiref0x / UACME

Defeating Windows User Account Control

uac-bypass uac dll-hijack bypass-uac verifier

147 commits 1 branch 4 releases 1 contributor BSD-2-Clause

Branch: master New pull request Find file Clone or download

hfiref0x v 2.8.7 ... Latest commit 37163d6 7 days ago

Compiled	v 2.8.7	7 days ago

Source	v 2.8.7	7 days ago
LICENSE.md	License and Readme update	4 months ago
README.md	v 2.8.7	7 days ago
UACME.sha256	v 2.8.7	7 days ago

README.md

UACMe

- Defeating Windows User Account Control by abusing built-in Windows AutoElevate backdoor.

System Requirements

- x86-32/x64 Windows 7/8/8.1/10 (client, some methods however works on server version too).
- Admin account with UAC set on default settings required.

Usage

Run executable from command line: akagi32 [Key] [Param] or akagi64 [Key] [Param]. See "Run examples" below for more info.

First param is number of method to use, second is optional command (executable file name including full path) to run. Second param can be empty - in this case program will execute elevated cmd.exe from system32 folder.

Keys (watch debug output with dbgview or similar for more info):

1. Author: Leo Davidson
    - Type: Dll Hijack
    - Method: IFileOperation
    - Target(s): \system32\sysprep\sysprep.exe
    - Component(s): cryptbase.dll
    - Implementation: ucmStandardAutoElevation
    - Works from: Windows 7 (7600)
    - Fixed in: Windows 8.1 (9600)
        - How: sysprep.exe hardened LoadFrom manifest elements
2. Author: Leo Davidson derivative
    - Type: Dll Hijack
    - Method: IFileOperation
    - Target(s): \system32\sysprep\sysprep.exe
    - Component(s): ShCore.dll
    - Implementation: ucmStandardAutoElevation
    - Works from: Windows 8.1 (9600)
    - Fixed in: Windows 10 TP (> 9600)
        - How: Side effect of ShCore.dll moving to \KnownDlls
3. Author: Leo Davidson derivative by WinNT/Pitou
    - Type: Dll Hijack
    - Method: IFileOperation
    - Target(s): \system32\oobe\setupsqm.exe
    - Component(s): WdsCore.dll

- Implementation: ucmStandardAutoElevation
- Works from: Windows 7 (7600)
- Fixed in: Windows 10 TH2 (10558)
  - How: side effect of OOBE redesign

4. Author: Jon Ericson, WinNT/Gootkit, mzH
   - Type: AppCompat
   - Method: RedirectEXE Shim
   - Target(s): \system32\cliconfg.exe
   - Component(s): -
   - Implementation: ucmShimRedirectEXE
   - Works from: Windows 7 (7600)
   - Fixed in: Windows 10 TP (> 9600)
     - How: Sdbinst.exe autoelevation removed, KB3045645/KB3048097 for rest Windows versions
5. Author: WinNT/Simda
   - Type: Elevated COM interface
   - Method: ISecurityEditor
   - Target(s): HKLM registry keys
   - Component(s): -
   - Implementation: ucmSimdaTurnOffUac
   - Works from: Windows 7 (7600)
   - Fixed in: Windows 10 TH1 (10147)
     - How: ISecurityEditor interface method changed
6. Author: Win32/Carberp
   - Type: Dll Hijack
   - Method: WUSA

- Target(s): \ehome\mcx2prov.exe, \system32\migwiz\migwiz.exe
- Component(s): WdsCore.dll, CryptBase.dll, CryptSP.dll
- Implementation: ucmWusaMethod
- Works from: Windows 7 (7600)
- Fixed in: Windows 10 TH1 (10147)
  - How: WUSA /extract option removed

7. Author: Win32/Carberp derivative
   - Type: Dll Hijack
   - Method: WUSA
   - Target(s): \system32\cliconfg.exe
   - Component(s): ntwdblib.dll
   - Implementation: ucmWusaMethod
   - Works from: Windows 7 (7600)
   - Fixed in: Windows 10 TH1 (10147)
     - How: WUSA /extract option removed
8. Author: Leo Davidson derivative by Win32/Tilon
   - Type: Dll Hijack
   - Method: IFileOperation
   - Target(s): \system32\sysprep\sysprep.exe
   - Component(s): Actionqueue.dll
   - Implementation: ucmStandardAutoElevation
   - Works from: Windows 7 (7600)
   - Fixed in: Windows 8.1 (9600)
     - How: sysprep.exe hardened LoadFrom manifest
9. Author: Leo Davidson, WinNT/Simda, Win32/Carberp derivative

- Type: Dll Hijack
- Method: IFileOperation, ISecurityEditor, WUSA
- Target(s): IFEO registry keys, \system32\cliconfg.exe
- Component(s): Attacker defined Application Verifier Dll
- Implementation: ucmAvrfMethod
- Works from: Windows 7 (7600)
- Fixed in: Windows 10 TH1 (10147)
  - How: WUSA /extract option removed, ISecurityEditor interface method changed

10. Author: WinNT/Pitou, Win32/Carberp derivative
    - Type: Dll Hijack
    - Method: IFileOperation, WUSA
    - Target(s): \system32\{New}or{Existing}\{autoelevated}.exe, e.g. winsat.exe
    - Component(s): Attacker defined dll, e.g. PowProf.dll, DevObj.dll
    - Implementation: ucmWinSATMethod
    - Works from: Windows 7 (7600)
    - Fixed in: Windows 10 TH2 (10548)
      - How: AppInfo elevated application path control hardening
11. Author: Jon Ericson, WinNT/Gootkit, mzH
    - Type: AppCompat
    - Method: Shim Memory Patch
    - Target(s): \system32\iscsicli.exe
    - Component(s): Attacker prepared shellcode
    - Implementation: ucmShimPatch
    - Works from: Windows 7 (7600)

- Fixed in: Windows 8.1 (9600)
  - How: Sdbinst.exe autoelevation removed, KB3045645/KB3048097 for rest Windows versions

12. Author: Leo Davidson derivative
    - Type: Dll Hijack
    - Method: IFileOperation
    - Target(s): \system32\sysprep\sysprep.exe
    - Component(s): dbgcore.dll
    - Implementation: ucmStandardAutoElevation
    - Works from: Windows 10 TH1 (10240)
    - Fixed in: Windows 10 TH2 (10565)
      - How: sysprep.exe manifest updated
13. Author: Leo Davidson derivative
    - Type: Dll Hijack
    - Method: IFileOperation
    - Target(s): \system32\mmc.exe EventVwr.msc
    - Component(s): elsext.dll
    - Implementation: ucmMMCMethod
    - Works from: Windows 7 (7600)
    - Fixed in: Windows 10 RS1 (14316)
      - How: Missing dependency removed
14. Author: Leo Davidson, WinNT/Sirefef derivative
    - Type: Dll Hijack
    - Method: IFileOperation
    - Target(s): \system\credwiz.exe, \system32\wbem\oobe.exe
    - Component(s): netutils.dll

- Implementation: ucmSirefefMethod
- Works from: Windows 7 (7600)
- Fixed in: Windows 10 TH2 (10548)
  - How: AppInfo elevated application path control hardening

15. Author: Leo Davidson, Win32/Addrop, Metasploit derivative
    - Type: Dll Hijack
    - Method: IFileOperation
    - Target(s): \system32\cliconfg.exe
    - Component(s): ntwdblib.dll
    - Implementation: ucmGenericAutoelevation
    - Works from: Windows 7 (7600)
    - Fixed in: Windows 10 RS1 (14316)
      - How: Cliconfg.exe autoelevation removed
16. Author: Leo Davidson derivative
    - Type: Dll Hijack
    - Method: IFileOperation
    - Target(s): \system32\GWX\GWXUXWorker.exe, \system32\inetsrv\inetmgr.exe
    - Component(s): SLC.dll
    - Implementation: ucmGWX
    - Works from: Windows 7 (7600)
    - Fixed in: Windows 10 RS1 (14316)
      - How: AppInfo elevated application path control and inetmgr executable hardening
17. Author: Leo Davidson derivative
    - Type: Dll Hijack (Import forwarding)
    - Method: IFileOperation

- Target(s): \system32\sysprep\sysprep.exe
- Component(s): unbcl.dll
- Implementation: ucmStandardAutoElevation2
- Works from: Windows 8.1 (9600)
- Fixed in: Windows 10 RS1 (14371)
  - How: sysprep.exe manifest updated

18. Author: Leo Davidson derivative
    - Type: Dll Hijack (Manifest)
    - Method: IFileOperation
    - Target(s): \system32\taskhost.exe, \system32\tzsync.exe (any ms exe without manifest)
    - Component(s): Attacker defined
    - Implementation: ucmAutoElevateManifest
    - Works from: Windows 7 (7600)
    - Fixed in: Windows 10 RS1 (14371)
      - How: Manifest parsing logic reviewed
19. Author: Leo Davidson derivative
    - Type: Dll Hijack
    - Method: IFileOperation
    - Target(s): \system32\inetsrv\inetmgr.exe
    - Component(s): MsCoree.dll
    - Implementation: ucmInetMgrMethod
    - Works from: Windows 7 (7600)
    - Fixed in: Windows 10 RS1 (14376)
      - How: inetmgr.exe executable manifest hardening, MitigationPolicy->ProcessImageLoadPolicy->PreferSystem32Images

20. Author: Leo Davidson derivative
    - Type: Dll Hijack
    - Method: IFileOperation
    - Target(s): \system32\mmc.exe, Rsop.msc
    - Component(s): WbemComn.dll
    - Implementation: ucmMMCMethod
    - Works from: Windows 7 (7600)
    - Fixed in: Windows 10 RS3 (16232)
        - How: Target requires wbemcomn.dll to be signed by MS
21. Author: Leo Davidson derivative
    - Type: Dll Hijack
    - Method: IFileOperation, SxS DotLocal
    - Target(s): \system32\sysprep\sysprep.exe
    - Component(s): comctl32.dll
    - Implementation: ucmSXSMethod
    - Works from: Windows 7 (7600)
    - Fixed in: Windows 10 RS3 (16232)
        - How: sysprep.exe requires MS signed modules to load
22. Author: Leo Davidson derivative
    - Type: Dll Hijack
    - Method: IFileOperation, SxS DotLocal
    - Target(s): \system32\consent.exe
    - Component(s): comctl32.dll
    - Implementation: ucmSXSMethod
    - Works from: Windows 7 (7600)

- Fixed in: unfixed 🙊
  - How: -

23. Author: Leo Davidson derivative
    - Type: Dll Hijack
    - Method: IFileOperation
    - Target(s): \system32\pkgmgr.exe
    - Component(s): DismCore.dll
    - Implementation: ucmDismMethod
    - Works from: Windows 7 (7600)
    - Fixed in: unfixed 🙊
      - How: -
24. Author: BreakingMalware
    - Type: Shell API
    - Method: Environment variables expansion
    - Target(s): \system32\CompMgmtLauncher.exe
    - Component(s): Attacker defined
    - Implementation: ucmCometMethod
    - Works from: Windows 7 (7600)
    - Fixed in: Windows 10 RS2 (15031)
      - How: CompMgmtLauncher.exe autoelevation removed
25. Author: Enigma0x3
    - Type: Shell API
    - Method: Registry key manipulation
    - Target(s): \system32\EventVwr.exe, \system32\CompMgmtLauncher.exe
    - Component(s): Attacker defined

- Implementation: ucmHijackShellCommandMethod
- Works from: Windows 7 (7600)
- Fixed in: Windows 10 RS2 (15031)
  - How: EventVwr.exe redesigned, CompMgmtLauncher.exe autoelevation removed

26. Author: Enigma0x3
    - Type: Race Condition
    - Method: File overwrite
    - Target(s): %temp%\GUID\dismhost.exe
    - Component(s): LogProvider.dll
    - Implementation: ucmDiskCleanupRaceCondition
    - Works from: Windows 10 TH1 (10240)
    - AlwaysNotify compatible
    - Fixed in: Windows 10 RS2 (15031)
      - How: File security permissions altered
27. Author: ExpLife
    - Type: Elevated COM interface
    - Method: IARPUninstallStringLauncher
    - Target(s): Attacker defined
    - Component(s): Attacker defined
    - Implementation: ucmUninstallLauncherMethod
    - Works from: Windows 7 (7600)
    - Fixed in: Windows 10 RS3 (16199)
      - How: UninstallStringLauncher interface removed from COMAutoApprovalList
28. Author: Exploit/Sandworm
    - Type: Whitelisted component

- Method: InfDefaultInstall
- Target(s): Attacker defined
- Component(s): Attacker defined
- Implementation: ucmSandwormMethod
- Works from: Windows 7 (7600)
- Fixed in: Windows 8.1 (9600)
  - How: InfDefaultInstall.exe removed from g_lpAutoApproveEXEList (MS14-060)

29. Author: Enigma0x3
    - Type: Shell API
    - Method: Registry key manipulation
    - Target(s): \system32\sdctl.exe
    - Component(s): Attacker defined
    - Implementation: ucmAppPathMethod
    - Works from: Windows 10 TH1 (10240)
    - Fixed in: Windows 10 RS3 (16215)
      - How: Shell API update
30. Author: Leo Davidson derivative, lhc645
    - Type: Dll Hijack
    - Method: WOW64 logger
    - Target(s): \syswow64\{any elevated exe, e.g wusa.exe}
    - Component(s): wow64log.dll
    - Implementation: ucmWow64LoggerMethod
    - Works from: Windows 7 (7600)
    - Fixed in: unfixed 🙈
      - How: -

31. Author: Enigma0x3
    - Type: Shell API
    - Method: Registry key manipulation
    - Target(s): \system32\sdctl.exe
    - Component(s): Attacker defined
    - Implementation: ucmSdcltIsolatedCommandMethod
    - Works from: Windows 10 TH1 (10240)
    - Fixed in: Windows 10 RS4 (17025)
        - How: Shell API / Windows components update
32. Author: xi-tauw
    - Type: Dll Hijack
    - Method: UIPI bypass with uiAccess application
    - Target(s): \Program Files\Windows Media Player\osk.exe, \system32\EventVwr.exe, \system32\mmc.exe
    - Component(s): duser.dll, osksupport.dll
    - Implementation: ucmUiAccessMethod
    - Works from: Windows 7 (7600)
    - Fixed in: unfixed 🙈
        - How: -
33. Author: winscripting.blog
    - Type: Shell API
    - Method: Registry key manipulation
    - Target(s): \system32\fodhelper.exe, \system32\computerdefaults.exe
    - Component(s): Attacker defined
    - Implementation: ucmMsSettingsDelegateExecuteMethod
    - Works from: Windows 10 TH1 (10240)

- Fixed in: unfixed 🙊
  - How: -

34. Author: James Forshaw
    - Type: Shell API
    - Method: Environment variables expansion
    - Target(s): \system32\svchost.exe via \system32\schtasks.exe
    - Component(s): Attacker defined
    - Implementation: ucmDiskCleanupEnvironmentVariable
    - Works from: Windows 8.1 (9600)
    - AlwaysNotify compatible
    - Fixed in: unfixed 🙊
      - How: -
35. Author: CIA & James Forshaw
    - Type: Impersonation
    - Method: Token Manipulations
    - Target(s): Autoelevated applications
    - Component(s): Attacker defined
    - Implementation: ucmTokenModification
    - Works from: Windows 7 (7600)
    - AlwaysNotify compatible, see note
    - Fixed in: unfixed 🙊
      - How: -
36. Author: Thomas Vanhoutte
    - Type: Race condition
    - Method: NTFS reparse point & Dll Hijack

- Target(s): wusa.exe
- Component(s): dcomcnfg.exe, mmc.exe, ole32.dll, MsCoree.dll
- Implementation: ucmJunctionMethod
- Works from: Windows 7 (7600)
- Fixed in: unfixed 🙈
  - How: -

37. Author: Ernesto Fernandez, Thomas Vanhoutte
    - Type: Dll Hijack
    - Method: SxS DotLocal, NTFS reparse point
    - Target(s): \system32\dccw.exe
    - Component(s): GdiPlus.dll
    - Implementation: ucmSXSMethodDccw
    - Works from: Windows 7 (7600)
    - Fixed in: unfixed 🙈
      - How: -
38. Author: Clement Rouault
    - Type: Whitelisted component
    - Method: APPINFO command line spoofing
    - Target(s): \system32\mmc.exe
    - Component(s): Attacker defined
    - Implementation: ucmMethodHakril
    - Works from: Windows 7 (7600)
    - Fixed in: unfixed 🙈
      - How: -
39. Author: Stefan Kanthak

- Type: Dll Hijack
- Method: .NET Code Profiler
- Target(s): \system32\mmc.exe
- Component(s): Attacker defined
- Implementation: ucmMethodCorProfiler
- Works from: Windows 7 (7600)
- Fixed in: unfixed 🙈
  - How: -

40. Author: Ruben Boonen
    - Type: COM Handler hijack
    - Method: Registry key manipulation
    - Target(s): \system32\mmc.exe, \System32\recdisc.exe
    - Component(s): Attacker defined
    - Implementation: ucmMethodCOMHandlers
    - Works from: Windows 7 (7600)
    - Fixed in: unfixed 🙈
      - How: -
41. Author: Oddvar Moe
    - Type: Elevated COM interface
    - Method: ICMLuaUtil
    - Target(s): Attacker defined
    - Component(s): Attacker defined
    - Implementation: ucmCMLuaUtilShellExecMethod
    - Works from: Windows 7 (7600)

- Fixed in: unfixed 🙈
  - How: -

42. Author: BreakingMalware and Enigma0x3
    - Type: Elevated COM interface
    - Method: IFwCplLua
    - Target(s): Attacker defined
    - Component(s): Attacker defined
    - Implementation: ucmFwCplLuaMethod
    - Works from: Windows 7 (7600)
    - Fixed in: Windows 10 RS4 (17134)
      - How: Shell API update
43. Author: Oddvar Moe derivative
    - Type: Elevated COM interface
    - Method: IColorDataProxy, ICMLuaUtil
    - Target(s): Attacker defined
    - Component(s): Attacker defined
    - Implementation: ucmDccwCOMMethod
    - Works from: Windows 7 (7600)
    - Fixed in: unfixed 🙈
      - How: -
44. Author: bytecode77
    - Type: Shell API
    - Method: Environment variables expansion
    - Target(s): Multiple auto-elevated processes
    - Component(s): Various per target

    - Implementation: ucmMethodVolatileEnv
    - Works from: Windows 7 (7600)
    - Fixed in: Windows 10 RS3 (16299)
        - How: Current user system directory variables ignored during process creation
45. Author: bytecode77
    - Type: Shell API
    - Method: Registry key manipulation
    - Target(s): \system32\slui.exe
    - Component(s): Attacker defined
    - Implementation: ucmMethodSluiHijack
    - Works from: Windows 8.1 (9600)
    - Fixed in: unfixed 🙊
        - How: -
46. Author: Anonymous
    - Type: Race Condition
    - Method: Registry key manipulation
    - Target(s): \system32\BitlockerWizardElev.exe
    - Component(s): Attacker defined
    - Implementation: ucmBitlockerRCMethod
    - Works from: Windows 7 (7600)
    - Fixed in: Windows 10 RS4 (>16299)
        - How: Shell API update

Note:

- Method (6) unavailable in wow64 environment starting from Windows 8;

- Method (11) implemented in x86-32 version;
- Method (13) (19) (38) implemented only in x64 version;
- Method (14) require process injection, wow64 unsupported, use x64 version of this tool;
- Method (26) is still working, however it main advantage was UAC bypass on AlwaysNotify level. Since 15031 it is gone;
- Method (30) require x64 because it abuses WOW64 subsystem feature;
- Method (35) AlwaysNotify compatible as there always will be running autoelevated apps or user will have to launch them anyway;
- Method (38) require internet connection as it executes remote script located at github.com/hfiref0x/Beacon/blob/master/uac/exec.html.

Run examples:

- akagi32.exe 1
- akagi64.exe 3
- akagi32 1 c:\windows\system32\calc.exe
- akagi64 3 c:\windows\system32\charmap.exe

Warning

- This tool shows ONLY popular UAC bypass method used by malware, and reimplement some of them in a different way improving original concepts. There are exists different, not yet known to general public methods, be aware of this;
- Using (5) method will permanently turn off UAC (after reboot), make sure to do this in test environment or don't forget to re-enable UAC after tool usage;
- Using (5), (9) methods will permanently compromise security of target keys (UAC Settings key for (5) and IFEO for (9)), if you do tests on your real machine - restore keys security manually after you complete this tool usage;

- This tool is not intended for AV tests and not tested to work in aggressive AV environment, if you still plan to use it with installed bloatware AV soft - you use it at your own risk;
- Some AV may flag this tool as HackTool, MSE/WinDefender constantly marks it as malware, nope;
- If you run this program on real computer remember to remove all program leftovers after usage, for more info about files it drops to system folders see source code;
- Most of methods created for x64, with no x86-32 support in mind. I don't see any sense in supporting 32 bit versions of Windows or wow64, however with small tweaks most of them will run under wow64 as well.

If you wondering why this still exist and work here is the explanation, an official Microsoft WHITEFLAG (including totally incompetent statements as bonus) https://blogs.msdn.microsoft.com/oldnewthing/20160816-00/?p=94105

Protection

- Account without administrative privileges.

Malware usage

- It is currently known that UACMe used by Adware/Multiplug (9), by Win32/Dyre (3), by Win32/Empercrypt (10 & 13), by IcedID downloader (35 & 41). We do not take any responsibility for this tool usage in the malicious purposes. It is free, open-source and provided AS-IS for everyone.

Other usage

- Currently used as "signature" by "THOR APT" scanner (handmade pattern matching fraudware from Germany). We do not take any responsibility for this tool usage in the fraudware;

- The scamware project called "uacguard" has references to UACMe from their platform. We do not take any responsibility for this tool usage in the scamware. The repository https://github.com/hfiref0x/UACME and it contents are the only genuine source for UACMe code. We have nothing to do with external links to this project, mentions anywhere as well as modifications (forks);
- In July 2016 so-called "security company" Cymmetria released report about script-kiddie malware bundle called "Patchwork" and false flagged it as APT. They stated it was using "UACME method", which in fact is just slightly and unprofessionally modified injector dll from UACMe v1.9 and was using Carberp/Pitou hybrid method in malware self-implemented way. We do not take any responsibility for UACMe usage in the dubious advertising campaigns from third party "security companies".

Build

- UACMe comes with full source code, written in C;
- In order to build from source you need Microsoft Visual Studio 2013/2015 U2 and later versions.

References

- Windows 7 UAC whitelist, http://www.pretentiousname.com/misc/win7_uac_whitelist2.html
- Malicious Application Compatibility Shims, https://www.blackhat.com/docs/eu-15/materials/eu-15-Pierce-Defending-Against-Malicious-Application-Compatibility-Shims-wp.pdf
- Junfeng Zhang from WinSxS dev team blog, https://blogs.msdn.microsoft.com/junfeng/
- Beyond good ol' Run key, series of articles, http://www.hexacorn.com/blog
- KernelMode.Info UACMe thread, http://www.kernelmode.info/forum/viewtopic.php?f=11&t=3643
- Command Injection/Elevation - Environment Variables Revisited, https://breakingmalware.com/vulnerabilities/command-injection-and-elevation-environment-variables-revisited

- "Fileless" UAC Bypass Using eventvwr.exe and Registry Hijacking, https://enigma0x3.net/2016/08/15/fileless-uac-bypass-using-eventvwr-exe-and-registry-hijacking/
- Bypassing UAC on Windows 10 using Disk Cleanup, https://enigma0x3.net/2016/07/22/bypassing-uac-on-windows-10-using-disk-cleanup/
- Using IARPUninstallStringLauncher COM interface to bypass UAC, http://www.freebuf.com/articles/system/116611.html
- Bypassing UAC using App Paths, https://enigma0x3.net/2017/03/14/bypassing-uac-using-app-paths/
- "Fileless" UAC Bypass using sdclt.exe, https://enigma0x3.net/2017/03/17/fileless-uac-bypass-using-sdclt-exe/
- UAC Bypass or story about three escalations, https://habrahabr.ru/company/pm/blog/328008/
- Exploiting Environment Variables in Scheduled Tasks for UAC Bypass, https://tyranidslair.blogspot.ru/2017/05/exploiting-environment-variables-in.html
- First entry: Welcome and fileless UAC bypass, https://winscripting.blog/2017/05/12/first-entry-welcome-and-uac-bypass/
- Reading Your Way Around UAC in 3 parts:
    i. https://tyranidslair.blogspot.ru/2017/05/reading-your-way-around-uac-part-1.html
    ii. https://tyranidslair.blogspot.ru/2017/05/reading-your-way-around-uac-part-2.html
    iii. https://tyranidslair.blogspot.ru/2017/05/reading-your-way-around-uac-part-3.html
- Research on CMSTP.exe, https://msitpros.com/?p=3960

Authors

3rd party components usage

MinHook - The Minimalistic x86/x64 API Hooking Library for Windows, https://github.com/TsudaKageyu/minhook