hfiref0x/UACME: Defeating Windows User Account Control

github.com/hfiref0x/UACME

hfiref0x

hfiref0x/**UACME**

Defeating Windows User Account Control

2 Contributors | 1 Issue | 4k Stars | 1k Forks

build passing

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

folder.

Note: Since 3.5.0 version all "fixed" methods are considered obsolete and removed altogether with all supporting code/units. If you still need them - use v3.2.x branch

Keys (click to expand/collapse)

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
    - Code status: removed starting from v3.5.0 🚜
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
    - Code status: removed starting from v3.5.0 🚜
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
    - Code status: removed starting from v3.5.0 🚜

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
    - Code status: removed starting from v3.5.0 🚜
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
    - Code status: removed starting from v3.5.0 🚜
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
    - Code status: removed starting from v3.5.0 🚜
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
    - Code status: removed starting from v3.5.0 🚜

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
    - Code status: removed starting from v3.5.0 🚜
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
    - Code status: removed starting from v3.5.0 🚜
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
    - Code status: removed starting from v3.5.0 🚜

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
    - Code status: removed starting from v3.5.0 🚜
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
    - Code status: removed starting from v3.5.0 🚜
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
    - Code status: removed starting from v3.5.0 🚜
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
    - Code status: removed starting from v3.5.0 🚜

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
    - Code status: removed starting from v3.5.0 🚜
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
    - Code status: removed starting from v3.5.0 🚜
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
    - Code status: removed starting from v3.5.0 🚜

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
    - Code status: removed starting from v3.5.0 🚜
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
    - Code status: removed starting from v3.5.0 🚜
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
    - Code status: removed starting from v3.5.0 🚜

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: MitigationPolicy->ProcessImageLoadPolicy->PreferSystem32Images
    - Code status: removed starting from v3.5.0 🚜
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: -
    - Code status: added in v2.5.0
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: -
    - Code status: added in v2.5.1
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
    - Code status: removed starting from v3.5.0 🚜

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
    - Code status: removed starting from v3.5.0 🚜
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
    - Code status: removed starting from v3.5.0 🚜
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
    - Code status: removed starting from v3.5.0 🚜

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)
    - Code status: removed starting from v3.5.0 🚜
29. Author: Enigma0x3
    - Type: Shell API
    - Method: Registry key manipulation
    - Target(s): \system32\sdclt.exe
    - Component(s): Attacker defined
    - Implementation: ucmAppPathMethod
    - Works from: Windows 10 TH1 (10240)
    - Fixed in: Windows 10 RS3 (16215)
        How: Shell API update
    - Code status: removed starting from v3.5.0 🚜
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: -
    - Code status: added in v2.7.0
31. Author: Enigma0x3
    - Type: Shell API
    - Method: Registry key manipulation
    - Target(s): \system32\sdclt.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
    - Code status: removed starting from v3.5.0 🚜

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: -
    - Code status: added in v2.7.1
33. Author: winscripting.blog
    - Type: Shell API
    - Method: Registry key manipulation
    - Target(s): \system32\fodhelper.exe
    - Component(s): Attacker defined
    - Implementation: ucmShellRegModMethod
    - Works from: Windows 10 TH1 (10240)
    - Fixed in: unfixed 🙈
        How: -
    - Code status: added in v2.7.2
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: -
    - Code status: added in v2.7.2

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: Windows 10 RS5 (17686)
        How: ntoskrnl.exe->SeTokenCanImpersonate additional access token check added
    - Code status: removed starting from v3.5.0 🚜
36. Author: Thomas Vanhoutte aka SandboxEscaper
    - Type: Race condition
    - Method: NTFS reparse point & Dll Hijack
    - Target(s): wusa.exe, pkgmgr.exe
    - Component(s): Attacker defined
    - Implementation: ucmJunctionMethod
    - Works from: Windows 7 (7600)
    - Fixed in: unfixed 🙈
        How: -
    - Code status: added in v2.7.4
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: ucmSXSDccwMethod
    - Works from: Windows 7 (7600)
    - Fixed in: unfixed 🙈
        How: -
    - Code status: added in v2.7.5
38. Author: Clement Rouault
    - Type: Whitelisted component
    - Method: APPINFO command line spoofing
    - Target(s): \system32\mmc.exe
    - Component(s): Attacker defined
    - Implementation: ucmHakrilMethod
    - Works from: Windows 7 (7600)
    - Fixed in: unfixed 🙈
        How: -
    - Code status: added in v2.7.6

39. Author: Stefan Kanthak
    - Type: Dll Hijack
    - Method: .NET Code Profiler
    - Target(s): \system32\mmc.exe
    - Component(s): Attacker defined
    - Implementation: ucmCorProfilerMethod
    - Works from: Windows 7 (7600)
    - Fixed in: unfixed 🙈
        How: -
    - Code status: added in v2.7.7
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: ucmCOMHandlersMethod
    - Works from: Windows 7 (7600)
    - Fixed in: Windows 10 19H1 (18362)
        How: Side effect of Windows changes
    - Code status: removed starting from v3.5.0 🚜
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: -
    - Code status: added in v2.7.9
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
    - Code status: removed starting from v3.5.0 🚜

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: -
    - Code status: added in v2.8.3
44. Author: bytecode77
    - Type: Shell API
    - Method: Environment variables expansion
    - Target(s): Multiple auto-elevated processes
    - Component(s): Various per target
    - Implementation: ucmVolatileEnvMethod
    - Works from: Windows 7 (7600)
    - Fixed in: Windows 10 RS3 (16299)
        How: Current user system directory variables ignored during process creation
    - Code status: removed starting from v3.5.0 🚜
45. Author: bytecode77
    - Type: Shell API
    - Method: Registry key manipulation
    - Target(s): \system32\slui.exe
    - Component(s): Attacker defined
    - Implementation: ucmSluiHijackMethod
    - Works from: Windows 8.1 (9600)
    - Fixed in: Windows 10 20H1 (19041)
        How: Side effect of Windows changes
    - Code status: removed starting from v3.5.0 🚜
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
    - Code status: removed starting from v3.5.0 🚜

47. Author: clavoillotte & 3gstudent
    - Type: COM Handler Hijack
    - Method: Registry key manipulation
    - Target(s): \system32\mmc.exe
    - Component(s): Attacker defined
    - Implementation: ucmCOMHandlersMethod2
    - Works from: Windows 7 (7600)
    - Fixed in: Windows 10 19H1 (18362)
        How: Side effect of Windows changes
    - Code status: removed starting from v3.5.0 🚜
48. Author: deroko
    - Type: Elevated COM interface
    - Method: ISPPLUAObject
    - Target(s): Attacker defined
    - Component(s): Attacker defined
    - Implementation: ucmSPPLUAObjectMethod
    - Works from: Windows 7 (7600)
    - Fixed in: Windows 10 RS5 (17763)
        How: ISPPLUAObject interface method changed
    - Code status: removed starting from v3.5.0 🚜
49. Author: RinN
    - Type: Elevated COM interface
    - Method: ICreateNewLink
    - Target(s): \system32\TpmInit.exe
    - Component(s): WbemComn.dll
    - Implementation: ucmCreateNewLinkMethod
    - Works from: Windows 7 (7600)
    - Fixed in: Windows 10 RS1 (14393)
        How: Side effect of consent.exe COMAutoApprovalList introduction
    - Code status: removed starting from v3.5.0 🚜
50. Author: Anonymous
    - Type: Elevated COM interface
    - Method: IDateTimeStateWrite, ISPPLUAObject
    - Target(s): w32time service
    - Component(s): w32time.dll
    - Implementation: ucmDateTimeStateWriterMethod
    - Works from: Windows 7 (7600)
    - Fixed in: Windows 10 RS5 (17763)
        How: Side effect of ISPPLUAObject interface change
    - Code status: removed starting from v3.5.0 🚜

51. Author: bytecode77 derivative
    - Type: Elevated COM interface
    - Method: IAccessibilityCplAdmin
    - Target(s): \system32\rstrui.exe
    - Component(s): Attacker defined
    - Implementation: ucmAcCplAdminMethod
    - Works from: Windows 7 (7600)
    - Fixed in: Windows 10 RS4 (17134)
        How: Shell API update
    - Code status: removed starting from v3.5.0 🚜
52. Author: David Wells
    - Type: Whitelisted component
    - Method: AipNormalizePath parsing abuse
    - Target(s): Attacker defined
    - Component(s): Attacker defined
    - Implementation: ucmDirectoryMockMethod
    - Works from: Windows 7 (7600)
    - Fixed in: unfixed 🙈
        How: -
    - Code status: added in v3.0.4
53. Author: Emeric Nasi
    - Type: Shell API
    - Method: Registry key manipulation
    - Target(s): \system32\sdclt.exe
    - Component(s): Attacker defined
    - Implementation: ucmShellRegModMethod
    - Works from: Windows 10 (14393)
    - Fixed in: unfixed 🙈
        How: -
    - Code status: added in v3.1.3
54. Author: egre55
    - Type: Dll Hijack
    - Method: Dll path search abuse
    - Target(s): \syswow64\SystemPropertiesAdvanced.exe and other SystemProperties*.exe
    - Component(s): \AppData\Local\Microsoft\WindowsApps\srrstr.dll
    - Implementation: ucmEgre55Method
    - Works from: Windows 10 (14393)
    - Fixed in: Windows 10 19H1 (18362)
        How: SysDm.cpl!_CreateSystemRestorePage has been updated for secured load library call
    - Code status: removed starting from v3.5.0 🚜

55. Author: James Forshaw
    - Type: GUI Hack
    - Method: UIPI bypass with token modification
    - Target(s): \system32\osk.exe, \system32\msconfig.exe
    - Component(s): Attacker defined
    - Implementation: ucmTokenModUIAccessMethod
    - Works from: Windows 7 (7600)
    - Fixed in: unfixed 🙈
        How: -
    - Code status: added in v3.1.5
56. Author: Hashim Jawad
    - Type: Shell API
    - Method: Registry key manipulation
    - Target(s): \system32\WSReset.exe
    - Component(s): Attacker defined
    - Implementation: ucmShellRegModMethod2
    - Works from: Windows 10 (17134)
    - Fixed in: Windows 11 (22000)
        How: Windows components redesign
    - Code status: removed starting from v3.5.7 🚜
57. Author: Leo Davidson derivative by Win32/Gapz
    - Type: Dll Hijack
    - Method: IFileOperation
    - Target(s): \system32\sysprep\sysprep.exe
    - Component(s): unattend.dll
    - Implementation: ucmStandardAutoElevation
    - Works from: Windows 7 (7600)
    - Fixed in: Windows 8.1 (9600)
        How: sysprep.exe hardened LoadFrom manifest elements
    - Code status: removed starting from v3.5.0 🚜
58. Author: RinN
    - Type: Elevated COM interface
    - Method: IEditionUpgradeManager
    - Target(s): \system32\clipup.exe
    - Component(s): Attacker defined
    - Implementation: ucmEditionUpgradeManagerMethod
    - Works from: Windows 10 (14393)
    - Fixed in: unfixed 🙈
        How: -
    - Code status: added in v3.2.0

59. Author: James Forshaw
    - Type: AppInfo ALPC
    - Method: RAiLaunchAdminProcess and DebugObject
    - Target(s): Attacker defined
    - Component(s): Attacker defined
    - Implementation: ucmDebugObjectMethod
    - Works from: Windows 7 (7600)
    - Fixed in: unfixed 🙈
        How: -
    - Code status: added in v3.2.3
60. Author: Enigma0x3 derivative by WinNT/Glupteba
    - Type: Shell API
    - Method: Registry key manipulation
    - Target(s): \system32\CompMgmtLauncher.exe
    - Component(s): Attacker defined
    - Implementation: ucmGluptebaMethod
    - Works from: Windows 7 (7600)
    - Fixed in: Windows 10 RS2 (15063)
        How: CompMgmtLauncher.exe autoelevation removed
    - Code status: removed starting from v3.5.0 🚜
61. Author: Enigma0x3/bytecode77 derivative by Nassim Asrir
    - Type: Shell API
    - Method: Registry key manipulation
    - Target(s): \system32\slui.exe, \system32\changepk.exe
    - Component(s): Attacker defined
    - Implementation: ucmShellRegModMethod
    - Works from: Windows 10 (14393)
    - Fixed in: unfixed 🙈
        How: -
    - Code status: added in v3.2.5
62. Author: winscripting.blog
    - Type: Shell API
    - Method: Registry key manipulation
    - Target(s): \system32\computerdefaults.exe
    - Component(s): Attacker defined
    - Implementation: ucmShellRegModMethod
    - Works from: Windows 10 RS4 (17134)
    - Fixed in: unfixed 🙈
        How: -
    - Code status: added in v3.2.6

63. Author: Arush Agarampur
    - Type: Dll Hijack
    - Method: ISecurityEditor
    - Target(s): Native Image Cache elements
    - Component(s): Attacker defined
    - Implementation: ucmNICPoisonMethod
    - Works from: Windows 7 (7600)
    - Fixed in: unfixed 🙈
        How: -
    - Code status: added in v3.2.7
64. Author: Arush Agarampur
    - Type: Elevated COM interface
    - Method: IIEAxiAdminInstaller, IIEAxiInstaller2, IFileOperation
    - Target(s): IE add-on install cache
    - Component(s): Attacker defined
    - Implementation: ucmIeAddOnInstallMethod
    - Works from: Windows 7 (7600)
    - Fixed in: unfixed 🙈
        How: -
    - Code status: added in v3.5.1
65. Author: Arush Agarampur
    - Type: Elevated COM interface
    - Method: IWscAdmin
    - Target(s): Shell Protocol Hijack
    - Component(s): Attacker defined
    - Implementation: ucmWscActionProtocolMethod
    - Works from: Windows 7 (7600)
    - Fixed in: unfixed 🙈
        How: -
    - Code status: added in v3.5.2
66. Author: Arush Agarampur
    - Type: Elevated COM interface
    - Method: IFwCplLua, Shell Protocol Hijack
    - Target(s): Shell protocol registry entry and environment variables
    - Component(s): Attacker defined
    - Implementation: ucmFwCplLuaMethod2
    - Works from: Windows 7 (7600)
    - Fixed in: unfixed 🙈
        How: -
    - Code status: added in v3.5.3

67. Author: Arush Agarampur
    - Type: Shell API
    - Method: Shell Protocol Hijack
    - Target(s): \system32\fodhelper.exe
    - Component(s): Attacker defined
    - Implementation: ucmMsSettingsProtocolMethod
    - Works from: Windows 10 TH1 (10240)
    - Fixed in: unfixed 🙈
        How: -
    - Code status: added in v3.5.4
68. Author: Arush Agarampur
    - Type: Shell API
    - Method: Shell Protocol Hijack
    - Target(s): \system32\wsreset.exe
    - Component(s): Attacker defined
    - Implementation: ucmMsStoreProtocolMethod
    - Works from: Windows 10 RS5 (17763)
    - Fixed in: unfixed 🙈
        How: -
    - Code status: added in v3.5.5
69. Author: Arush Agarampur
    - Type: Shell API
    - Method: Environment variables expansion, Dll Hijack
    - Target(s): \system32\taskhostw.exe
    - Component(s): pcadm.dll
    - Implementation: ucmPcaMethod
    - Works from: Windows 7 (7600)
    - AlwaysNotify compatible
    - Fixed in: unfixed 🙈
        How: -
    - Code status: added in v3.5.6
70. Author: V3ded
    - Type: Shell API
    - Method: Registry key manipulation
    - Target(s): \system32\fodhelper.exe, \system32\computerdefaults.exe
    - Component(s): Attacker defined
    - Implementation: ucmShellRegModMethod3
    - Works from: Windows 10 (10240)
    - Fixed in: unfixed 🙈
        How: -
    - Code status: added in v3.5.7

71. Author: Arush Agarampur
    - Type: Dll Hijack
    - Method: ISecurityEditor
    - Target(s): Native Image Cache elements
    - Component(s): Attacker defined
    - Implementation: ucmNICPoisonMethod2
    - Works from: Windows 7 RTM (7600)
    - Fixed in: unfixed 🙈
        How: -
    - Code status: added in v3.5.8
72. Author: Emeric Nasi
    - Type: Dll Hijack
    - Method: Dll path search abuse
    - Target(s): \syswow64\msdt.exe, \system32\sdiagnhost.exe
    - Component(s): BluetoothDiagnosticUtil.dll
    - Implementation: ucmMsdtMethod
    - Works from: Windows 10 (10240)
    - Fixed in: unfixed 🙈
        How: -
    - Code status: added in v3.5.9
73. Author: orange_8361 and antonioCoco
    - Type: Shell API
    - Method: .NET deserialization
    - Target(s): \system32\mmc.exe EventVwr.msc
    - Component(s): Attacker defined
    - Implementation: ucmDotNetSerialMethod
    - Works from: Windows 7 RTM (7600)
    - Fixed in: unfixed 🙈
        How: -
    - Code status: added in v3.6.0

Note:

- Method (30) (63) and later implemented only in x64 version;
- Method (30) require x64 because it abuses WOW64 subsystem feature;
- Method (55) is not really reliable (as any GUI hacks) and included just for fun.

Run examples:

- akagi32.exe 23
- akagi64.exe 61
- akagi32 23 c:\windows\system32\calc.exe
- akagi64 61 c:\windows\system32\charmap.exe

Warning

- This tool shows ONLY popular UAC bypass method used by malware, and re-implement some of them in a different way improving original concepts. There are different, not yet known to the general public, methods. Be aware of this;
- 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 - 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 exists and working - here is the explanation - an official Microsoft WHITEFLAG (including totally incompetent statements as bonus) https://devblogs.microsoft.com/oldnewthing/20160816-00/?p=94105

Windows 10 support and testing policy

- UACMe tested only with LSTB/LTSC variants (1607/1809) and Last RTM-1 versions, e.g. if current version is 2004 it will be tested on 2004 (19041) and previous version 1909 (18363);
- Insider builds are not supported as methods may be fixed there.

Protection

Account without administrative privileges.

Malware usage

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 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 2015 and later versions.

Compiled Binaries

- They are not provided since 2.8.9 and will never be provided in future. The reasons (and why you too should not provide them to the general public):
    - If you look at this project in a nutshell it is a HackTool, despite initial goal to be a demonstrator. Of course several AV's detects it as HackTool (MS WD for example), however most of VirusTotal patients detects it as generic "malware". Which is of course incorrect, however unfortunately some lazy malware writers blindly copy-paste code to their crapware (or even simple use this tool directly) thus some AV created signatures based on project code parts;
    - By giving compiled binaries to everyone you make life of script-kiddies much easier because having need to compile from source works as perfect barrier for exceptionally dumb script-kiddies and "button-clickers";
    - Having compiled binaries in the repository will ultimately lead to flagging this repository pages as malicious (due to above reasons) by various content filters (SmartScreen, Google Safe Browsing etc).
- This decision is a final and won't be changed.

Instructions

- Select Platform ToolSet first for project in solution you want to build (Project->Properties->General):
    - v140 for Visual Studio 2015;
    - v141 for Visual Studio 2017;
    - v142 for Visual Studio 2019.
- For v140 and above set Target Platform Version (Project->Properties->General):
    - If v140 then select 8.1 (Note that Windows 8.1 SDK must be installed);
    - If v141/v142 then select 10.
- The following SDK are required for building the binaries:
    - Windows 8.1 or Windows 10 SDK (tested with 19041 version)
    - NET Framework SDK (tested with 4.8 version)
- To build working binary:
    - Compile payload units
    - Compile Naka module
    - Encrypt all payload units using Naka module
    - Generate secret blobs for these units using Naka module
    - Move compiled units and secret blobs to the Akagi\Bin directory
    - Rebuild Akagi

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, https://www.kernelmode.info/forum/viewtopicf985.html?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:
    1. https://tyranidslair.blogspot.ru/2017/05/reading-your-way-around-uac-part-1.html
    2. https://tyranidslair.blogspot.ru/2017/05/reading-your-way-around-uac-part-2.html
    3. https://tyranidslair.blogspot.ru/2017/05/reading-your-way-around-uac-part-3.html
- Research on CMSTP.exe, https://msitpros.com/?p=3960
- UAC bypass via elevated .NET applications, https://offsec.provadys.com/UAC-bypass-dotnet.html
- UAC Bypass by Mocking Trusted Directories, https://medium.com/tenable-techblog/uac-bypass-by-mocking-trusted-directories-24a96675f6e
- Yet another sdclt UAC bypass, http://blog.sevagas.com/?Yet-another-sdclt-UAC-bypass
- UAC Bypass via SystemPropertiesAdvanced.exe and DLL Hijacking, https://egre55.github.io/system-properties-uac-bypass/
- Accessing Access Tokens for UIAccess, https://tyranidslair.blogspot.com/2019/02/accessing-access-tokens-for-uiaccess.html
- Fileless UAC Bypass in Windows Store Binary, https://www.activecyber.us/1/post/2019/03/windows-uac-bypass.html
- Calling Local Windows RPC Servers from .NET, https://googleprojectzero.blogspot.com/2019/12/calling-local-windows-rpc-servers-from.html

- Microsoft Windows 10 UAC bypass local privilege escalation exploit, https://packetstormsecurity.com/files/155927/Microsoft-Windows-10-Local-Privilege-Escalation.html
- UACMe 3.5, WD and the ways of mitigation, https://swapcontext.blogspot.com/2020/10/uacme-35-wd-and-ways-of-mitigation.html
- UAC bypasses from COMAutoApprovalList, https://swapcontext.blogspot.com/2020/11/uac-bypasses-from-comautoapprovallist.html
- Utilizing Programmatic Identifiers (ProgIDs) for UAC Bypasses, https://v3ded.github.io/redteam/utilizing-programmatic-identifiers-progids-for-uac-bypasses
- MSDT DLL Hijack UAC bypass, https://blog.sevagas.com/?MSDT-DLL-Hijack-UAC-bypass
- UAC bypass through .Net Deserialization vulnerability in eventvwr.exe, https://twitter.com/orange_8361/status/1518970259868626944

Authors