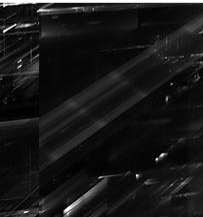


What the DLL is happening?

— A practical approach to identifying SOH.



Presenter



Frank McClain

Senior Detection Engineer

RED CANARY

 @littlemac042

- Detection Engineering training lead within the Red Canary CIRT
- Ran the Security Operations team at a national financial services provider for several years
- Recently got a Mavic Mini drone, and I swear it's been windy ever since...

What is Dynamic Link Library (DLL) search order?

- A means by which DLLs are found and loaded into process memory
- Multiple factors or variables exist, but basically follows this order
 - DLL of the same name already loaded in memory (by another process)
 - DLL is listed in the 'KnownDLLs' registry key
 - DLL of the same name exists in the same directory as the loading process
 - System directory (e.g., '%SystemRoot%\System32')

What is search order hijacking (SOH)?

- Form of DLL hijacking (on Windows)
- Commonly used by adversaries (commodity and advanced)
- Primarily leverages the same directory as the loading process

Why is SOH important?

TECHNIQUE T1038

DLL Search Order Hijacking

Dridex infections are the main reason that we observe high levels of DLL Search Order Hijacking in the environments we monitor. However, we've also observed Emotet and PlugX leveraging the technique.

#8

OVERALL RANK

16%

ORGANIZATIONS AFFECTED

788

CONFIRMED THREATS

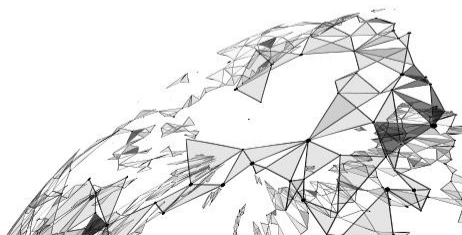
Why is SOH important?

- Common adversary technique
- Persistence mechanism
- Privilege escalation
- Bypass security controls

Covering SOH

SEARCH ORDER HIJACKING

- Identification or detection methods
- Detector concepts (ideas to try)
- Detection examples (SOH in action)
- Mitigation strategies



Identification methods

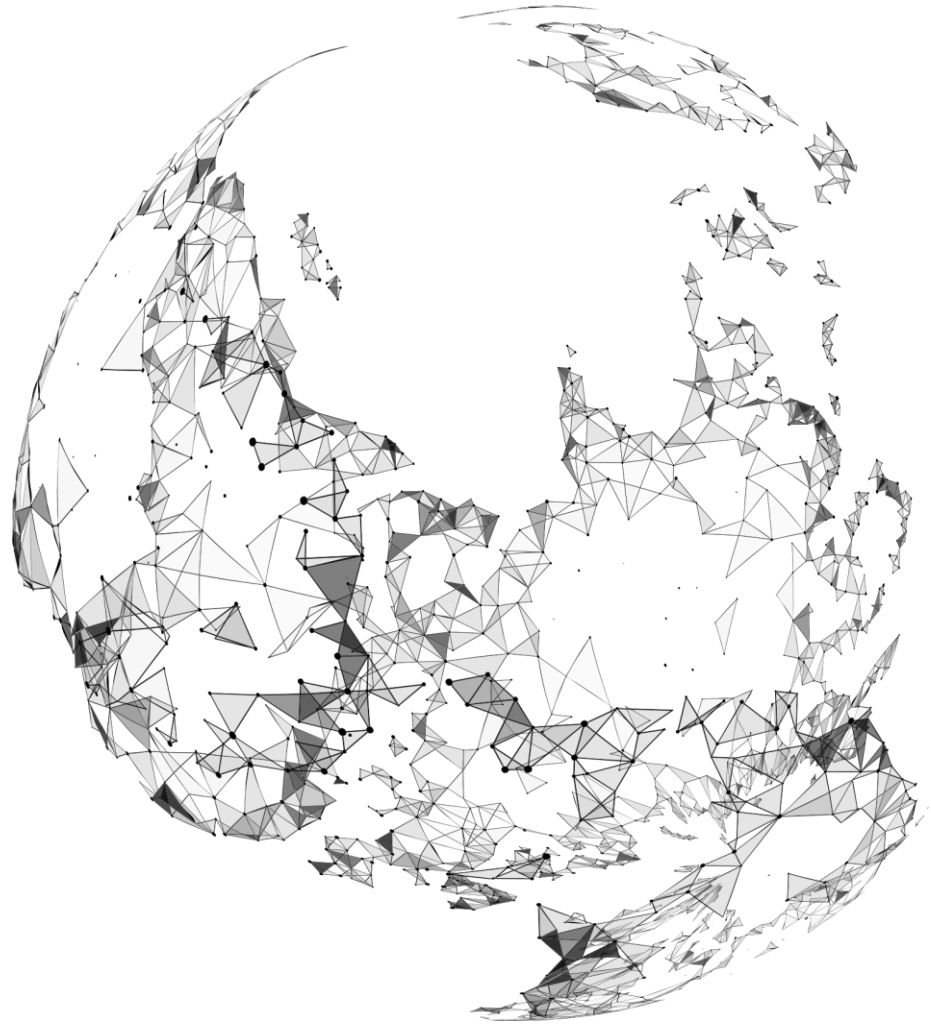
- Creation of a scheduled task in a suspect path (user path, ProgramData, etc.)
- Script processes (wscript.exe, cscript.exe, etc.) spawning an unsigned binary
- Service Host (svchost.exe) spawned by a suspect parent process
- Legitimate system binaries copied to/executing from suspect paths
- Unsigned, unknown DLLs written to/loaded from suspect paths
- And more...

Post-identification validation

- Check to see if binaries are legitimate, signed, or operating system
- Identify if any DLL files are written to the same (suspect) path
- Check DLLs to see if they are legit, or have the same name as legit DLLs
- If the binary was executed from the suspect path, see whether it loaded DLLs from that path

SEARCH ORDER HIJACKING

Detector concepts

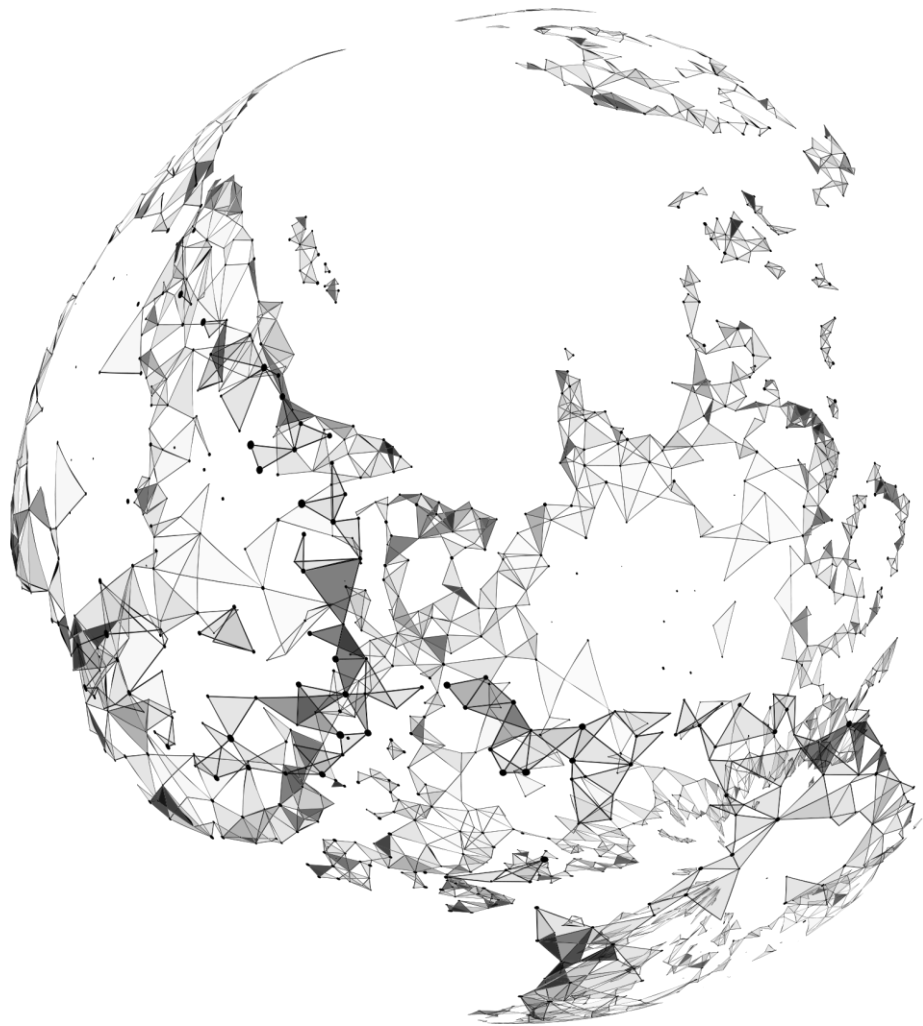


Sample query logic (getting started)

- `Process_Is 'schtasks.exe' AND Command_Line_Contains 'appdata\roaming' AND Command_Line_Contains '*.exe'`
- `Parent_Process_Is 'taskeng.exe' OR Parent_Process_Is 'svchost.exe' AND Binary_Publisher_Is 'Microsoft Windows' AND Process_Path_Is 'appdata\roaming'`
- `Binary_Publisher_Is 'Microsoft Windows' AND Process_Path_Is_Not 'windows\system32' OR Process_Path_Is_Not 'windows\syswow64' OR Process_Path_Is_Not 'windows\winsxs'`

SEARCH ORDER HIJACKING

Detection examples



DETECTION #1

Process spawned by explorer.exe

c:\windows\system32\schtasks.exe 3f9fd6d3b3e96b8f576db72035db38a7

Threat occurred here

Remove

Add annotation

...

Command line:

```
"C:\Windows\System32\schtasks.exe" /Create /F /TN "Gihsqandaf" /TR  
C:\Users\██████████\AppData\Roaming\07p0H9W\BdeUISrv.exe /SC minute /MO 60
```

This command created a scheduled task named **Gihsqandaf** to execute

```
C:\Users\██████████\AppData\Roaming\07p0H9W\BdeUISrv.exe every 60 minutes.
```

SCHEDULED TASK

Setting up to execute a binary within the user profile. This legit binary is part of BitLocker.

DETECTION #1

File last wrote

c:\users\[redacted]\appdata\roaming\o7poh9w\bdeuisrv.exe cc46d3e88a4f2fc4da

Threat occurred here

Remove

Add annotation

FILE COPY

...

This is a legitimate copy of BDE UI Launcher (**BdeUISrv.exe**), written to disk by **cmd.exe** .

Here is where the binary is written to disk under the profile. It is expected to reside in System32.

DETECTION #1

File created
c:\users\[redacted]\appdata\roaming\o7poh9w\wtsapi32.dll

Threat occurred here

Remove

Add annotation

FILE COPY

File written to disk by `cmd.exe`. This is not the correct path for the legitimate `wtsapi32.dll`, which is the **Windows Remote Desktop Session Host Server Service** and should reside in `System32`, `SysWOW64`, or a `WinSXS` subdirectory.

Surrounding context and behavior indicates that this is a malicious DLL with a legitimate name for use in DLL load-order hijacking.

Here is where the malicious DLL is written to disk in the same path under the user profile; when the EXE is launched, this DLL will be loaded into memory.

DETECTION #2

File last wrote

c:\users\ [redacted] \appdata\local\temp\ie.8aa444.tmp 35e4b50f21c4bc2e795c72d30e65976c

Threat occurred here

Remove

Add annotation

MASQUERADING

...

Malicious binary written to disk by `wscript.exe`. It contains metadata from the GNU DiffUtils process `cmp.exe` (used to compare files); however, this binary is **156 KB** in size, and the legitimate `cmp.exe` is **56 KB**. In addition, threat intelligence indicates this is malware from the **Emotet** family.

Malicious binary written to disk, leveraging metadata (or perhaps a compromised binary) to masquerade as GNU DiffUtils.

DETECTION #2

File last wrote

c:\users\ [redacted] \appdata\roaming\oyieq\wfs.exe cd6acf3b997099b6cfb2417d3942f755

Threat occurred here

Remove

Add annotation

FILE COPY

Legitimate Microsoft Windows Fax and Scan binary (**wfs.exe**) written to disk by **cmd.exe** under the user profile. This is typically observed during preparation for a DLL hijack, where a malicious DLL file is written to the same path using a legitimate name, to be loaded into memory by the copied binary.

Legitimate system binary written to disk under the user profile; SOH, here we come!

DETECTION #2

File created
c:\users\██████████\appdata\roaming\oyieq\mfc42u.dll

Threat occurred here

Remove

Add annotation

FILE COPY

Unknown DLL written to disk in the same path as `wfs.exe` , by `cmd.exe` . This was originally written to disk by `explorer.exe` at 201██████████ 17:46:23 UTC and contains metadata to help it masquerade as the legitimate Windows Class Library (`wcl.dll`).

Here is where the malicious DLL is written to disk in the same path under the user profile; when the EXE is launched, this DLL will be loaded into memory.

DETECTION #2

Process spawned by explorer.exe

c:\windows\system32\schtasks.exe 838d346d1d28f00783b7a6c6bd03a0d

Threat occurred here

Remove

Add annotation

SCHEDULED TASK

...

Command line:

```
"C:\Windows\System32\schtasks.exe" /Create /F /TN "Lsbqpnq" /TR  
C:\Users\██████████\AppData\Roaming\oyIEq\WFS.exe /SC minute /MO 60
```

**Setting up to execute a binary within the user profile.
This is the legitimate Windows Fax & Scan binary.**

DETECTION #2

Process spawned

c:\users\[redacted]\appdata\roaming\oyieq\wfs.exe cd6acf3b997099b6cfb2417420426755

Threat occurred here

Remove

Add annotation

EXECUTION

...

Command line:

C:\Users\[redacted]\AppData\Roaming\oyIEq\WFS.exe

Execution of the legitimate Windows Fax & Scan binary, from within the user path (1 hour after the scheduled task creation).

DETECTION #2

Module loaded by wfs.exe

\users\... \appdata\roaming\oyieq\mfc42u.dll 1c33976de1c21e20719185f58b1

Threat occurred here

Remove

Add annotation

SOH!

...

This is the malicious DLL previously written to disk in the same path as the legitimate `wfs.exe`. This action is the DLL being loaded into memory by `wfs.exe`, indicative of a DLL load-order hijack. This binary exists only on this host within the environment.

And now the malicious DLL gets loaded into memory by the legitimate WFS process. Whatever code it contains is now active and being used by wfs.exe.

Can you prevent or mitigate the risk of SOH?

- If you're compiling binaries, make all DLLs explicit, including fully qualified path
- Maintain security hygiene and an active patching program
- Leverage Microsoft provisions
 - SafeDLLSearchMode
 - CWDIllegalInDllSearch
- Good detection methodologies

Reference links

- Red Canary on MITRE ATT&CK®
 - <https://redcanary.com/mitre-attack/>
- Red Canary (2020 Threat Detection Report)
 - <https://redcanary.com/threat-detection-report/techniques/dll-search-order-hijacking/>
- MITRE ATT&CK - Hijack Execution Flow: DLL Search Order Hijacking
 - <https://attack.mitre.org/techniques/T1574/001/>
- Microsoft on Dynamic-Link Library Search Order
 - <https://docs.microsoft.com/en-us/windows/win32/dlls/dynamic-link-library-search-order>

FEEDBACK

Q & A

