

Immunity

Local Privilege Escalation in Origin

2020-07-29

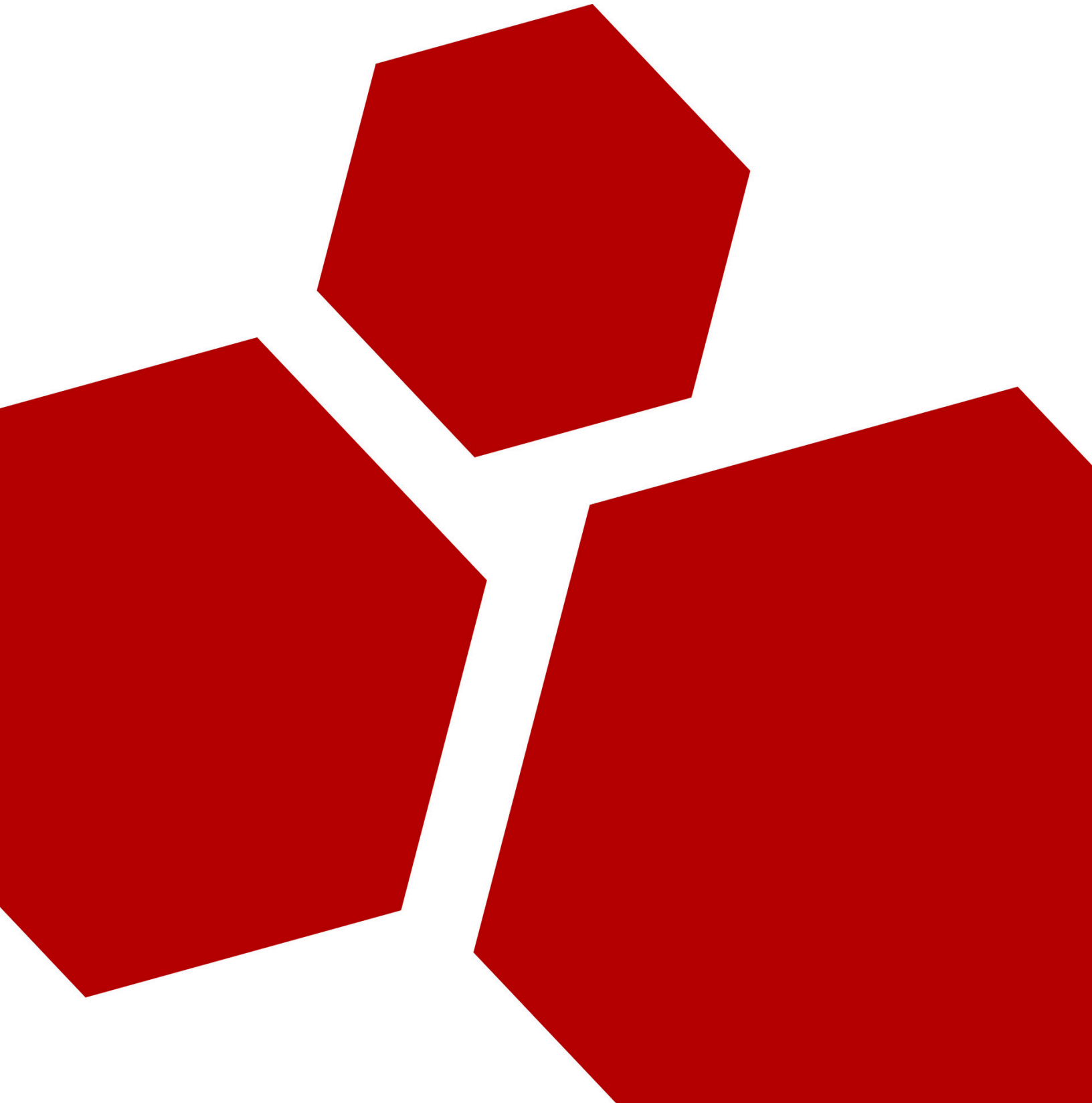


Table of Contents

Advisory Information	2
Vulnerability Information	2
Vulnerability Description	2
Report Timeline	7
Disclaimer.....	7

Advisory Information

Title: Local Privilege Escalation in Origin

Vendors contacted: Electronic Arts Inc.

Release mode: Coordinated Release

Credits: This vulnerability was discovered by Joel Noguera in collaboration with Lautaro Fain.

Vulnerability Information

Class: Uncontrolled Search Path Element [CWE-427]

Affected Version: Origin 10.5.74.41754 (prior versions might be affected)

Remotely Exploitable: No

Locally Exploitable: Yes

Severity: High - 7.8 (CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H)

CVE Identifier: CVE-2020-15524

Vulnerability Description

A Local Privilege Escalation attack scenario is present within the 'OriginClientService.exe' process, this kind of attack allows malicious users to elevate privileges within the target system and therefore perform actions that otherwise they would not be able to execute. In this particular case, attackers will be able to escalate from a Windows Non-Privileged user to NT AUTHORITY/SYSTEM by planting a DLL in a Qt Path on which they have control.

The reason this is happening is because Origin looks for Qt Plugins to load in folders that either do not exist or can be accessed and modified by everyone inside the system, as can be seen in the image below.



Figure 1. Origin looking for the 'plugins' directory under 'E:\Qt\5.8.0\qtbase\'.

At first sight it looks like 'plugin' is the only affected subdirectory, but this is not exclusively true as whenever this path is found, the service keeps digging further in order to load specific Qt Plugin classes¹ looking for their respective folders, as we can see in the table below.

Plugin Class Name	Affected Path
Platforms	E:\Qt\5.8.0\qtbase\plugins\platforms
PlatformThemes	E:\Qt\5.8.0\qtbase\plugins\platformthemes
Styles	E:\Qt\5.8.0\qtbase\plugins\styles

The affected path is hardcoded into a 'Qt5Core.dll' variable called 'qt_prfxpath' as can be seen in the screenshot below.

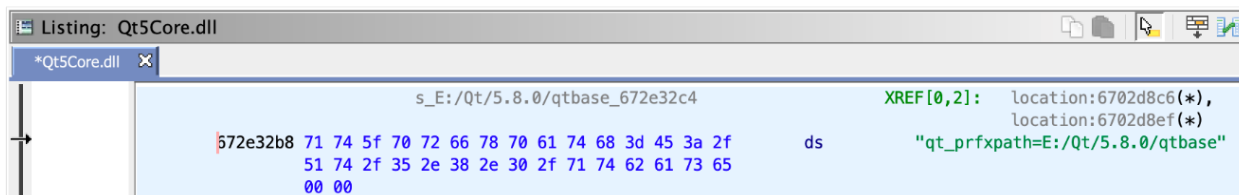
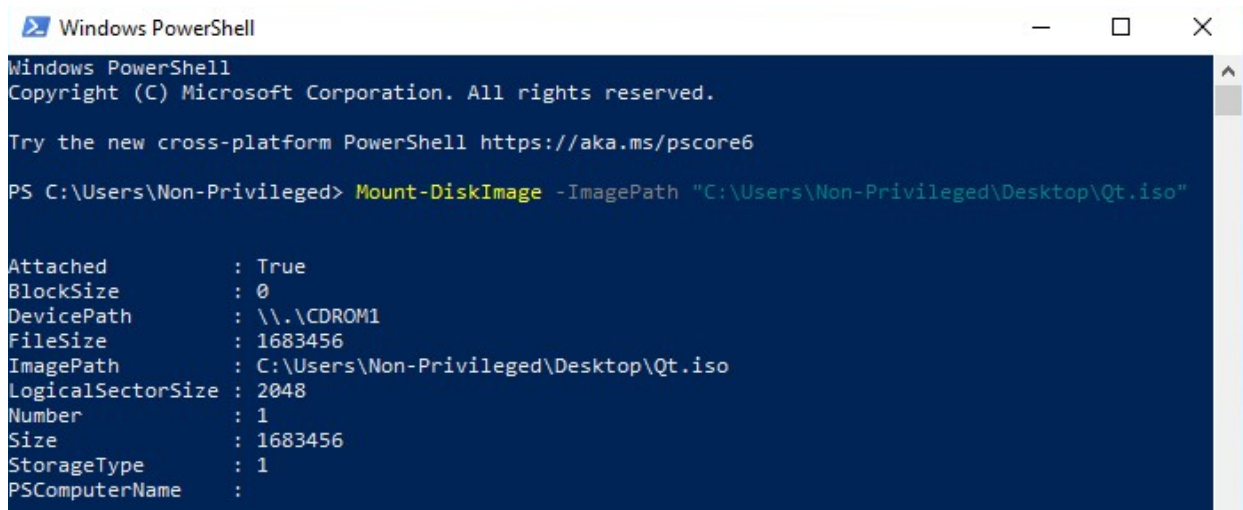


Figure 2. 'Qt5Core.dll' making use of 'qt_prfxpath'.

As this vulnerability comprehends a path that is allocated under an 'E:' drive, an issue that may arise while trying to exploit this vulnerability would be the target machine not having another disk mounted as the mentioned drive. Luckily, Windows treats mounted ISO's pretty much as a new device and these kinds of disk images can be mounted system wide (can be seen and found by all the host users) without any need of special privileges.

In order to mount the ISO file, a non-privileged user can make use of the following command in MS PowerShell (this is what our proof of concept code uses to reproduce the issue).

¹ <https://doc.qt.io/qt-5/plugins-howto.html>



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Non-Privileged> Mount-DiskImage -ImagePath "C:\Users\Non-Privileged\Desktop\Qt.iso"

Attached          : True
BlockSize        : 0
DevicePath       : \\.\CDROM1
FileSize         : 1683456
ImagePath        : C:\Users\Non-Privileged\Desktop\Qt.iso
LogicalSectorSize : 2048
Number           : 1
Size             : 1683456
StorageType      : 1
PSComputerName   :
```

Figure 3. Mounting the ISO file containing the subdirectory structure and the DLL as a non-privileged user.

This very last action should leave the target subdirectory structure deployed in the new system 'drive' and ready to be used by the Origin process when trying to load its contents, which is again, what we expect to achieve when running our proof of concept tool.

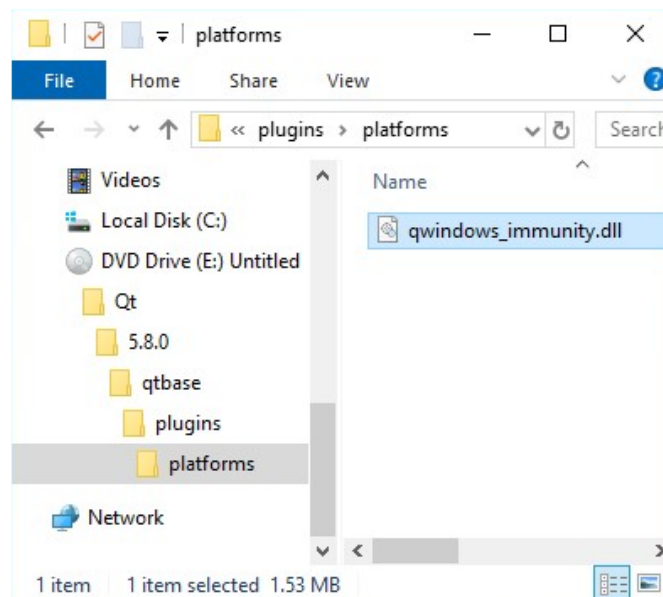


Figure 4. Recreated subdirectory path to the DLL.

After all the requirements are met, Origin should now be restarted or either started in order to force it into loading the new DLL from the affected place.

Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

Time of Day	Process Name	PID	Operation	Path	Result
9:01:50.1909863 AM	OriginClientService.exe	19492	Thread Create		SUCCESS
9:01:50.2024331 AM	OriginClientService.exe	19492	Thread Create		SUCCESS
9:01:50.2398439 AM	OriginClientService.exe	19492	Load Image	E:\Qt5.8.0\qtbase\plugins\platforms\qwindows_immunity.dll	SUCCESS
9:01:50.2400182 AM	OriginClientService.exe	19492	Load Image	C:\Windows\SysWOW64\imm32.dll	SUCCESS
9:01:50.2483434 AM	OriginClientService.exe	19492	Process Create	c:\WINDOWS\SysWOW64\notepad.exe	SUCCESS
9:01:50.2710918 AM	OriginClientService.exe	19492	Load Image	C:\Windows\SysWOW64\kernel.appcore.dll	SUCCESS
9:01:50.6405334 AM	OriginClientService.exe	19492	Thread Create		SUCCESS
9:01:50.6476403 AM	OriginClientService.exe	19492	Thread Create		SUCCESS

Figure 5. 'OriginClientService.exe' loading the 'qwindows_immunity.dll' DLL.

Please note that apart from 'OriginClientService.exe', 'Origin.exe' and 'OriginWebHelperService.exe' also make use of the 'Qt5Core.dll' library that performs the search action so our custom-crafted DLL will be loaded three times in a row almost every time.

And, as a side effect of being loaded this number of times, our DLL will also spawn three different instances of 'notepad.exe' which will run as the user who executed the program. For 'Origin.exe', it will be executed with the same access level that this process has. For 'OriginWebHelperService.exe', the new process will be executed on the context of the LOCAL SERVICE user. And finally, for 'OriginClientService.exe', 'notepad.exe' will run as 'NT AUTHORITY\SYSTEM', and therefore implies that no visual window for this process will be evidenced.

Still, we can see the new process being spawned right from the service through Process Monitor.

Process Name	NT AUTHORITY\SYSTEM	0.03	4,200 K	16,132 K	5372	OriginClientService	Electronic Arts
notepad.exe	NT AUTHORITY\SYSTEM	< 0.01	2,956 K	9,936 K	384	Notepad	Microsoft Corporation

Figure 6. 'notepad.exe' spawned as NT AUTHORITY\SYSTEM from 'OriginClientService.exe'.

If we inspect the new thread closely, we will be able to see more detailed information such as its parent process, its time of spawn and, as mentioned before, the user who has run it.

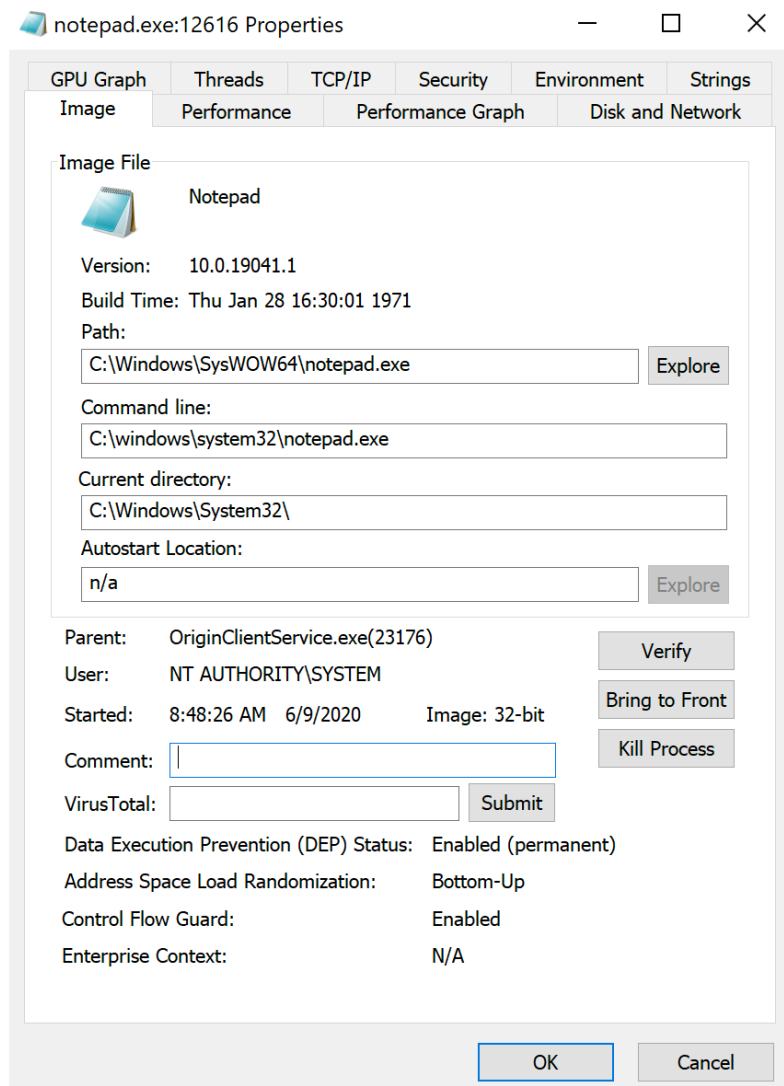


Figure 7. More detailed info on the 'notepad.exe' process spawned as NT AUTHORITY\SYSTEM.

To sum things up, it is worth mentioning that, in order to be successfully loaded by the application the DLL should pass a number of checks involving its headers and also some exported functions required by QT. For ease of producing a proof of concept (as creating a new Qt Plugin would have taken much more time), we patched an existent well-formed DLL so it would execute our desired payload when loaded.

Report Timeline

2020-06-17: Initial contact with the vendor via secure@ea.com.

2020-06-17: EA Security Team confirmed the reception of the email and requested the draft version of the advisory.

2020-06-17: A draft report with technical details and a proof of concept was sent to the vendor.

2020-06-19: Immunity Inc. ask EA Security Team if they were able to reproduce the vulnerability.

2020-06-19: EA Security Team notifies that they haven't receive any email and mentions that this could be to their email protection.

2020-06-19: Immunity Inc. sends a draft report and encrypts the proof of concept to avoid being block by EA email protection.

2020-06-19: EA Security Team confirmed the reception and opens a case with tracking ID PSECR-267.

2020-06-23: EA Security Team notifies that they were unable to reproduce the issue.

2020-06-24: Immunity Inc. sends a detailed guide on how to reproduce the issue.

2020-06-25: EA Security Team informs they were able to reproduce the issue and are working on a fix.

2020-06-25: Immunity Inc. offers to delay the release of the advisory with tracking ID PSECR-241 even with this vulnerability already fixed because this could lead a potential attacker to identify the vulnerability described on this report.

2020-06-25: EA Security Team agrees with the offer.

2020-07-03: Immunity Inc. sent a request to Mitre for the CVE ID.

2020-07-04: Mitre assigns CVE-2020-15524.

2020-07-08: EA Security Team informs the fix could be released next week and they will provide a date as soon as possible.

2020-07-08: Immunity Inc. acknowledge the status update.

2020-07-10: EA Security Team informs they are planning to release the fix on July 22th.

2020-07-14: Immunity Inc. informs that the advisory report release date is scheduled to July 29th.

2020-07-14: EA Security Team requests an updated version of the advisory report.

2020-07-15: Immunity Inc. send an updated version of the advisory report.

2020-07-20: EA Security Team request to change severity scoring.

2020-07-21: Immunity Inc. agreed and send an updated version of the advisory report.

2020-07-22: EA release the fix, a blog post² and advisory³.

2020-07-29: Immunity Inc. publish the advisory.

Disclaimer

² <https://www.ea.com/security/news/july-origin-security-update>

³ <https://www.ea.com/security/news/easec-2020-001-elevation-of-privilege-vulnerability-in-origin-client>

The contents of this advisory are copyright (c) 2020 Immunity Inc., and are licensed under a Creative Commons Attribution-NoDerivatives 4.0 International (CC BY-ND 4.0): <https://creativecommons.org/licenses/by-nd/4.0/>