» Issue39401



This issue tracker **has been migrated to GitHub**, and is currently **read-only**.
For more information, **see the GitHub FAQs in the Python's Developer Guide.**



This issue has been migrated to GitHub:
https://github.com/python/cpython/issues/83582

classification

Title: [CVE-2020-8315] Unsafe dll loading in getpathp.c on Win7	
Type: security	**Stage**: resolved
Components: Windows	**Versions**: Python 3.8, Python 3.7, Python 3.6

process

Status: closed	**Resolution**: fixed
Dependencies:	**Superseder**:
Assigned To: steve.dower	**Nosy List**: anthonywee, eryksun, jkloth, miss-islington, ned.deily, paul.moore, steve.dower, tim.golden, vstinner, zach.ware
Priority:	**Keywords**: patch

Created on **2020-01-21 01:02** by **anthonywee**, last changed **2022-04-11 14:59** by **admin**. This issue is now **closed**.

Files

File name	Uploaded	Description	Edit
python unsafe dll loading.png	anthonywee, 2020-01-21 01:02		

Pull Requests

URL	Status	Linked	Edit
PR 18231	merged	steve.dower, 2020-01-28 09:57	
PR 18232	merged	steve.dower, 2020-01-28 10:02	
PR 18233	merged	steve.dower, 2020-01-28 10:05	
PR 18234	merged	steve.dower, 2020-01-28 10:06	

Messages (17)

msg360348 - (view) **Author: Anthony Wee (anthonywee) *** **Date: 2020-01-21 01:02**

```
On Win7, running Python in the terminal will attempt to load the "api-ms-win-core-
path-l1-1-0.dll" from various paths outside of the Python directory and the
C:\Windows\System32 directories. This behavior can be verified using Process
Monitor (see attachment).

This is happening due to direct calls to LoadLibraryW() in getpathp.c without any
```

"LOAD_LIBRARY_SEARCH*" flags.

In join():

https://github.com/python/cpython/blob/c02b41b1fb115c87693530ea6a480b2e15460424/PC/getpathp.c#L255

and canonicalize():

https://github.com/python/cpython/blob/c02b41b1fb115c87693530ea6a480b2e15460424/PC/getpathp.c#L291

For both cases, the methods they are trying to load from api-ms-win-core-path-l1-1-0.dll (PathCchCanonicalizeEx and PathCchCombineEx) were introduced in Win8.

I tested on Win7 and Win10 and they differ in how they load these api-ms-win-* dll's and whether they appear in process monitor. In Win7, a CreateFile event appears in procmon, while in Win10 it seems like the OS is automatically loading the module from kernelbase.dll. Also in Win7 the loading of api-ms-win-core-path-l1-1-0.dll will fail while in Win10 it succeeds. However, in Win7 when it fails it results in the standard dll search strategy, which will eventually search outside of the secure directories such as the directories in the PATH env var: **https://docs.microsoft.com/en-us/windows/win32/dlls/dynamic-link-library-search-order**

Each of the problematic methods in cpython have a pattern of attempting to load the dll, then falling back to an older version of the method. Thus in Win7, the dll fails to load and it falls back to the older version of the method. In Win10, the dll load succeeds and we use the new versions of the methods.

I'm working on a fix to pass the LOAD_LIBRARY_SEARCH_DEFAULT_DIRS flag to limit to the dll search path scope.

msg360350 - (view) **Author: Eryk Sun (eryksun) *** **Date: 2020-01-21 04:34**

> On Win7, running Python in the terminal will attempt to load the
> "api-ms-win-core-path-l1-1-0.dll" from various paths outside of the
> Python directory and the C:\Windows\System32 directories.

"api-ms-win-core-path-l1-1-0.dll" is not assigned in the API set schema (in ApiSetSchema.dll) in Windows 7. Since the name is neither in the list of known DLLs nor the list of assigned API sets, the loader searches for it in the normal way. (FYI, the number of API sets increased from 35 in Windows 7 up to 502 in Windows 8.1.)

> I'm working on a fix to pass the LOAD_LIBRARY_SEARCH_DEFAULT_DIRS

I think this could use just LOAD_LIBRARY_SEARCH_SYSTEM32. I see no reason to try to load "api-ms-win-core-path-l1-1-0.dll" from the application directory or user directories.

I'm adding 3.6-3.9 to the list of affected versions. In 3.9 it can use a static import instead (i.e. remove LoadLibraryaExW / GetProcAddress), since only Windows 8.1+ is supported.

msg360529 - (view) **Author: Steve Dower (steve.dower) *** **Date: 2020-01-23 01:56**

Agreed, we can just search System32 for this. Thanks for doing the patch!

For future reference, and for anyone else reading this, we generally prefer unavoidable DLL hijacking bugs to come to the Python Security Response Team first (**security@python.org**).

msg360858 - (view) **Author: Steve Dower (steve.dower) *** **Date: 2020-01-28 10:18**

For clarity, I'm removing 3.9 from the affected versions. This version does not support Windows 7, and only Windows 7 is vulnerable to this DLL hijack.

Also submitting the CVE request.

msg360882 - (view) **Author: Eryk Sun (eryksun) *** **Date: 2020-01-28 16:33**

> For clarity, I'm removing 3.9 from the affected versions. This version
> does not support Windows 7, and only Windows 7 is vulnerable to this
> DLL hijack.

I added 3.9 for the related issue to switch to using a static import, since Windows 7 isn't supported in 3.9. But I guess that should have been made a separate issue, or added to the omnibus issue ~~**bpo-32592**~~.

msg360933 - (view) **Author: Steve Dower (steve.dower) *** **Date: 2020-01-29 02:30**

This is now assigned CVE-2020-8315 (**https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-8315 https://nvd.nist.gov/vuln/detail/CVE-2020-8315**)

Thanks Anthony for the report! I included your name as the reporter, though I don't see it on any of the pages.

msg360935 - (view) **Author: Steve Dower (steve.dower) *** **Date: 2020-01-29 02:46**

New changeset **6a65eba44bfd82ccc8bed4b5c6dd6637549955d5** by Steve Dower in branch 'master':
~~**bpo-39401**~~: Avoid unsafe DLL load on Windows 7 and earlier (~~**GH-18231**~~)
https://github.com/python/cpython/commit/6a65eba44bfd82ccc8bed4b5c6dd6637549955d5

msg360946 - (view) **Author: STINNER Victor (vstinner) *** **Date: 2020-01-29 10:23**

I added **https://python-security.readthedocs.io/vuln/unsafe-dll-load-windows-7.html** to track fixes in all branches.

msg360994 - (view) **Author: Jeremy Kloth (jkloth) *** **Date: 2020-01-29 23:30**

As noted on the PR landing page, this PR has caused failures of 2 buildbots:

https://buildbot.python.org/all/#builders/81/builds/272

https://buildbot.python.org/all/#builders/150/builds/227

(both are Windows 7)

msg360997 - (view) **Author: Steve Dower (steve.dower) *** **Date: 2020-01-30 00:41**

Both of those buildbots should be retired (or repurposed for versions of Python that still support Windows 7) :)

msg360998 - (view) **Author: Eryk Sun (eryksun) *** **Date: 2020-01-30 00:41**

> this PR has caused failures of 2 buildbots

The master branch should no longer get built on Windows 7 machines. The initial build succeeds, but running "_freeze_importlib[_d].exe" fails with STATUS_DLL_NOT_FOUND (0xC0000135, i.e. -1073741515) since "api-ms-win-core-path-l1-1-0.dll" (linked from pathcch.lib) is not a Windows 7 API set.

msg360999 - (view) **Author: Steve Dower (steve.dower) *** **Date: 2020-01-30 00:42**

> I added **https://python-security.readthedocs.io/vuln/unsafe-dll-load-windows-7.html** to track fixes in all branches.

Thanks, Victor!

Python 2.7 and 3.5 are not vulnerable. The issue was added in 3.6 when I added support for installing Python into a long path name on up-to-date OS, which required dynamically loading an OS function. That dynamic load was the problem.

msg361013 - (view) **Author: miss-islington (miss-islington)** **Date: 2020-01-30 06:07**

New changeset **561c59777c8426fde0ef48b57cf02eddaeb2a5b8** by Steve Dower in branch '3.7':
[3.7] ~~**bpo-39401**~~: Avoid unsafe DLL load on Windows 7 and earlier (~~**GH-18231**~~) (~~**GH-18232**~~)
https://github.com/python/cpython/commit/561c59777c8426fde0ef48b57cf02eddaeb2a5b8

msg361014 - (view) **Author: miss-islington (miss-islington)** **Date: 2020-01-30 06:18**

New changeset **ad4a20b87d79a619ffbdea3f26848780899494e5** by Steve Dower in branch '3.8':
[3.8] ~~**bpo-39401**~~: Avoid unsafe DLL load on Windows 7 and earlier (~~**GH-18231**~~) (~~**GH-18234**~~)
https://github.com/python/cpython/commit/ad4a20b87d79a619ffbdea3f26848780899494e5

msg361022 - (view) **Author: STINNER Victor (vstinner) *** **Date: 2020-01-30 08:28**

>> I added **https://python-security.readthedocs.io/vuln/unsafe-dll-load-windows-7.html** to track fixes in all branches.

> Thanks, Victor! Python 2.7 and 3.5 are not vulnerable. The issue was added in 3.6 when I added support for installing Python into a long path name on up-to-date OS, which required dynamically loading an OS function. That dynamic load was the problem.

Oh ok, I updated the page to reflect that. I also added 3.7 & 3.8 commits.

msg361038 - (view) **Author: Anthony Wee (anthonywee) *** **Date: 2020-01-30 11:05**

> Thanks Anthony for the report! I included your name as the reporter, though I don't see it on any of the pages.

No problem! Thanks Steve, Eryk, and Victor for jumping on this!

msg361088 - (view) **Author: Ned Deily (ned.deily) *** **Date: 2020-01-31 02:12**

New changeset **51332c467ed2e07a191f903d554d0c54248e4d88** by Steve Dower in branch '3.6':
[3.6] ~~**bpo-39401**~~: Avoid unsafe DLL load on Windows 7 and earlier (~~**GH-18231**~~) (~~**GH-18233**~~)
https://github.com/python/cpython/commit/51332c467ed2e07a191f903d554d0c54248e4d88

History

Date	User	Action	Args
2022-04-11 14:59:25	admin	set	github: 83582
2020-01-31 02:14:13	ned.deily	set	priority: deferred blocker -> status: open -> closed resolution: fixed stage: patch review -> resolved
2020-01-31 02:12:27	ned.deily	set	messages: + **msg361088**
2020-01-30 11:05:05	anthonywee	set	messages: + **msg361038**
2020-01-30 08:28:35	vstinner	set	messages: + **msg361022**
2020-01-30 06:18:33	miss-islington	set	messages: + **msg361014**
2020-01-30 06:07:21	miss-islington	set	nosy: + **miss-islington** messages: + **msg361013**
2020-01-30 00:42:41	steve.dower	set	messages: + **msg360999**
2020-01-30 00:41:47	eryksun	set	messages: + **msg360998**
2020-01-30 00:41:07	steve.dower	set	messages: + **msg360997**
2020-01-29 23:30:26	jkloth	set	nosy: + **jkloth** messages: + **msg360994**
2020-01-29 10:23:47	vstinner	set	messages: + **msg360946**
2020-01-29 02:46:39	steve.dower	set	messages: + **msg360935**
2020-01-29 02:30:16	steve.dower	set	messages: + **msg360933** title: Unsafe dll loading in getpathp.c on Win7 -> [CVE-2020-8315] Unsafe dll loading in getpathp.c on Win7
2020-01-28 17:41:47	vstinner	set	nosy: + **vstinner**
2020-01-28 16:33:36	eryksun	set	messages: + **msg360882**
2020-01-28 10:18:01	steve.dower	set	messages: + **msg360858** versions: - Python 3.9
2020-01-28 10:06:21	steve.dower	set	pull_requests: + **pull_request17614**

2020-01-28 10:05:38	steve.dower	set	pull_requests: + **pull_request17613**
2020-01-28 10:02:29	steve.dower	set	pull_requests: + **pull_request17612**
2020-01-28 09:57:36	steve.dower	set	keywords: + **patch** stage: needs patch -> patch review pull_requests: + **pull_request17611**
2020-01-28 09:44:52	steve.dower	set	assignee: **steve.dower**
2020-01-23 03:39:19	ned.deily	set	nosy: + **ned.deily**
2020-01-23 03:37:19	ned.deily	set	priority: normal -> deferred blocker
2020-01-23 01:56:01	steve.dower	set	messages: + **msg360529**
2020-01-21 04:34:55	eryksun	set	type: security components: + Windows versions: + Python 3.6, Python 3.7, Python 3.8, Python 3.9 nosy: + **paul.moore**, **eryksun**, **tim.golden**, **zach.ware**, **steve.dower** messages: + **msg360350** stage: needs patch
2020-01-21 01:02:15	anthonywee	create	