» Issue39421

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/83602

classification

Title: Use-after-free in heappushpop() of heapq module	
Type: crash	**Stage**: resolved
Components: Extension Modules	**Versions**: Python 3.9, Python 3.8, Python 3.7

process

Status: closed	**Resolution**: fixed
Dependencies:	**Superseder**:
Assigned To:	**Nosy List**: alex, corona10, dk0n9, methane, miss-islington, ned.deily, pablogsal, vstinner
Priority: normal	**Keywords**: patch, security_issue

Created on **2020-01-22 15:11** by **dk0n9**, last changed **2022-04-11 14:59** by **admin**. This issue is now **closed**.

Pull Requests

URL	Status	Linked	Edit
PR 18118	merged	pablogsal, 2020-01-22 15:54	
PR 18144	closed	miss-islington, 2020-01-23 14:07	
PR 18145	merged	miss-islington, 2020-01-23 14:07	
PR 18146	merged	miss-islington, 2020-01-23 14:07	
PR 18149	merged	miss-islington, 2020-01-23 15:05	

Messages (10)

msg360474 - (view) **Author: Dk0n9 (dk0n9)** **Date: 2020-01-22 15:13**

The variable `heap` in heappushpop does not add a reference count

```c
    cmp = PyObject_RichCompareBool(PyList_GET_ITEM(heap, 0), item, Py_LT);
    if (cmp < 0)
        return NULL;
    if (cmp == 0) {
```

```
        Py_INCREF(item);
        return item;
    }
```

POC :
```python
import heapq

class h(int):
    def __lt__(self, o):
        list1.clear()
        return NotImplemented

list1 = []

heapq.heappush(list1, h(0))
heapq.heappushpop(list1, 1)
```

Crash detail with asan:

```
==62141==ERROR: AddressSanitizer: heap-use-after-free on address 0x6060000fd778
at pc 0x00000049cdce bp 0x7ffe9690f650 sp 0x7ffe9690f640
READ of size 8 at 0x6060000fd778 thread T0
    #0 0x49cdcd in long_richcompare Objects/longobject.c:3047
    #1 0x4f9495 in do_richcompare Objects/object.c:802
    #2 0x4f9495 in PyObject_RichCompare Objects/object.c:846
    #3 0x4f9495 in PyObject_RichCompareBool Objects/object.c:868
    #4 0x7ff74c523594 in _heapq_heappushpop_impl /home/******/Python-
3.9.0a2/Modules/_heapqmodule.c:267
    #5 0x7ff74c523594 in _heapq_heappushpop /home/******/Python-
3.9.0a2/Modules/clinic/_heapqmodule.c.h:109
    #6 0x854c30 in cfunction_vectorcall_FASTCALL Objects/methodobject.c:366
    #7 0x443885 in _PyObject_VectorcallTstate Include/cpython/abstract.h:111
    #8 0x443885 in _PyObject_Vectorcall Include/cpython/abstract.h:120
    #9 0x443885 in call_function Python/ceval.c:4850
    #10 0x443885 in _PyEval_EvalFrameDefault Python/ceval.c:3306
    #11 0x5e1d76 in _PyEval_EvalFrame Include/internal/pycore_ceval.h:43
    #12 0x5e1d76 in _PyEval_EvalCode Python/ceval.c:4142
    #13 0x5e2207 in _PyEval_EvalCodeWithName Python/ceval.c:4174
    #14 0x5e2207 in PyEval_EvalCodeEx Python/ceval.c:4190
    #15 0x5e2207 in PyEval_EvalCode Python/ceval.c:717
    #16 0x6862fc in run_eval_code_obj Python/pythonrun.c:1125
    #17 0x6862fc in run_mod Python/pythonrun.c:1147
    #18 0x6862fc in PyRun_FileExFlags Python/pythonrun.c:1063
    #19 0x6867b2 in PyRun_SimpleFileExFlags Python/pythonrun.c:428
    #20 0x446495 in pymain_run_file Modules/main.c:369
    #21 0x446495 in pymain_run_python Modules/main.c:553
    #22 0x446495 in Py_RunMain Modules/main.c:632
    #23 0x446f86 in pymain_main Modules/main.c:662
    #24 0x446f86 in Py_BytesMain Modules/main.c:686
    #25 0x7ff74f34882f in __libc_start_main (/lib/x86_64-linux-
gnu/libc.so.6+0x2082f)
```

```
    #26 0x444478 in _start (/home/***/Python-3.9.0a2/python+0x444478)

0x6060000fd778 is located 24 bytes inside of 56-byte region
[0x6060000fd760,0x6060000fd798)
freed by thread T0 here:
    #0 0x7ff7500b72ca in __interceptor_free (/usr/lib/x86_64-linux-
gnu/libasan.so.2+0x982ca)
    #1 0x52a5d9 in subtype_dealloc Objects/typeobject.c:1291
    #2 0x4222a6 in _Py_DECREF Include/object.h:478
    #3 0x4222a6 in frame_dealloc Objects/frameobject.c:636
    #4 0x422088 in _Py_DECREF Include/object.h:478
    #5 0x422088 in function_code_fastcall Objects/call.c:335
    #6 0x53aac6 in _PyObject_VectorcallTstate Include/cpython/abstract.h:111
    #7 0x53aac6 in vectorcall_unbound Objects/typeobject.c:1459
    #8 0x53aac6 in slot_tp_richcompare Objects/typeobject.c:6703
    #9 0x4f921d in do_richcompare Objects/object.c:796
    #10 0x4f921d in PyObject_RichCompare Objects/object.c:846
    #11 0x4f921d in PyObject_RichCompareBool Objects/object.c:868
    #12 0x7ff74c523594 in _heapq_heappushpop_impl /home/******/Python-
3.9.0a2/Modules/_heapqmodule.c:267
    #13 0x7ff74c523594 in _heapq_heappushpop /home/******/Python-
3.9.0a2/Modules/clinic/_heapqmodule.c.h:109
    #14 0x854c30 in cfunction_vectorcall_FASTCALL Objects/methodobject.c:366
    #15 0x443885 in _PyObject_VectorcallTstate Include/cpython/abstract.h:111
    #16 0x443885 in _PyObject_Vectorcall Include/cpython/abstract.h:120
    #17 0x443885 in call_function Python/ceval.c:4850
    #18 0x443885 in _PyEval_EvalFrameDefault Python/ceval.c:3306
    #19 0x5e1d76 in _PyEval_EvalFrame Include/internal/pycore_ceval.h:43
    #20 0x5e1d76 in _PyEval_EvalCode Python/ceval.c:4142
    #21 0x5e2207 in _PyEval_EvalCodeWithName Python/ceval.c:4174
    #22 0x5e2207 in PyEval_EvalCodeEx Python/ceval.c:4190
    #23 0x5e2207 in PyEval_EvalCode Python/ceval.c:717
    #24 0x6862fc in run_eval_code_obj Python/pythonrun.c:1125
    #25 0x6862fc in run_mod Python/pythonrun.c:1147
    #26 0x6862fc in PyRun_FileExFlags Python/pythonrun.c:1063
    #27 0x6867b2 in PyRun_SimpleFileExFlags Python/pythonrun.c:428
    #28 0x446495 in pymain_run_file Modules/main.c:369
    #29 0x446495 in pymain_run_python Modules/main.c:553
    #30 0x446495 in Py_RunMain Modules/main.c:632
    #31 0x446f86 in pymain_main Modules/main.c:662
    #32 0x446f86 in Py_BytesMain Modules/main.c:686
    #33 0x7ff74f34882f in __libc_start_main (/lib/x86_64-linux-
gnu/libc.so.6+0x2082f)

previously allocated by thread T0 here:
    #0 0x7ff7500b7602 in malloc (/usr/lib/x86_64-linux-gnu/libasan.so.2+0x98602)
    #1 0x6dbfd5 in _PyObject_GC_Alloc Modules/gcmodule.c:2146
    #2 0x6dbfd5 in _PyObject_GC_Malloc Modules/gcmodule.c:2173
    #3 0x527d20 in PyType_GenericAlloc Objects/typeobject.c:1014
    #4 0x4b890b in long_subtype_new Objects/longobject.c:5132
    #5 0x4b890b in long_new_impl Objects/longobject.c:5075
    #6 0x4b890b in long_new Objects/clinic/longobject.c.h:36
    #7 0x52f606 in type_call Objects/typeobject.c:973
    #8 0x462b46 in _PyObject_MakeTpCall Objects/call.c:189
    #9 0x436b70 in _PyObject_VectorcallTstate Include/cpython/abstract.h:109
```

```
    #10 0x436b70 in _PyObject_Vectorcall Include/cpython/abstract.h:120
    #11 0x436b70 in call_function Python/ceval.c:4850
    #12 0x436b70 in _PyEval_EvalFrameDefault Python/ceval.c:3337
    #13 0x5e1d76 in _PyEval_EvalFrame Include/internal/pycore_ceval.h:43
    #14 0x5e1d76 in _PyEval_EvalCode Python/ceval.c:4142
    #15 0x5e2207 in _PyEval_EvalCodeWithName Python/ceval.c:4174
    #16 0x5e2207 in PyEval_EvalCodeEx Python/ceval.c:4190
    #17 0x5e2207 in PyEval_EvalCode Python/ceval.c:717
    #18 0x6862fc in run_eval_code_obj Python/pythonrun.c:1125
    #19 0x6862fc in run_mod Python/pythonrun.c:1147
    #20 0x6862fc in PyRun_FileExFlags Python/pythonrun.c:1063
    #21 0x6867b2 in PyRun_SimpleFileExFlags Python/pythonrun.c:428
    #22 0x446495 in pymain_run_file Modules/main.c:369
    #23 0x446495 in pymain_run_python Modules/main.c:553
    #24 0x446495 in Py_RunMain Modules/main.c:632
    #25 0x446f86 in pymain_main Modules/main.c:662
    #26 0x446f86 in Py_BytesMain Modules/main.c:686
    #27 0x7ff74f34882f in __libc_start_main (/lib/x86_64-linux-
gnu/libc.so.6+0x2082f)

SUMMARY: AddressSanitizer: heap-use-after-free Objects/longobject.c:3047
long_richcompare
Shadow bytes around the buggy address:
  0x0c0c80017a90: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c0c80017aa0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c0c80017ab0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c0c80017ac0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c0c80017ad0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
=>0x0c0c80017ae0: fd fd fd fd fd fd fd fa fa fa fa fa fd fd fd[fd]
  0x0c0c80017af0: fd fd fd fa fa fa fa fa fd fd fd fd fd fd fd fa
  0x0c0c80017b00: fa fa fa fa fd fd fd fd fd fd fd fd fa fa fa fa
  0x0c0c80017b10: 00 00 00 00 00 00 00 00 fa fa fa fa fd fd fd fd
  0x0c0c80017b20: fd fd fd fa fa fa fa fa fd fd fd fd fd fd fd fa
  0x0c0c80017b30: fa fa fa fa fd fd fd fd fd fd fd fa fa fa fa fa
Shadow byte legend (one shadow byte represents 8 application bytes):
  Addressable:           00
  Partially addressable: 01 02 03 04 05 06 07
  Heap left redzone:       fa
  Heap right redzone:      fb
  Freed heap region:       fd
  Stack left redzone:      f1
  Stack mid redzone:       f2
  Stack right redzone:     f3
  Stack partial redzone:   f4
  Stack after return:      f5
  Stack use after scope:   f8
  Global redzone:          f9
  Global init order:       f6
  Poisoned by user:        f7
  Container overflow:      fc
  Array cookie:            ac
  Intra object redzone:    bb
  ASan internal:           fe
==62141==ABORTING
```

msg360475 - (view) **Author: Dong-hee Na (corona10) *** **Date: 2020-01-22 15:31**

Reproducible.

It looks similar to ~~bpo-38588~~.
We will apply the same solution as we did at ~~bpo-38588~~?
or do we plan to apply the solution which is suggested on msg359023?

msg360477 - (view) **Author: Pablo Galindo Salgado (pablogsal) *** **Date: 2020-01-22 15:46**

To be honest, given how many ways this bug happens I think its time to consider msg359023.

msg360478 - (view) **Author: Dong-hee Na (corona10) *** **Date: 2020-01-22 15:49**

> To be honest, given how many ways this bug happens I think its time to consider msg359023.

+1 to me also

msg360479 - (view) **Author: Pablo Galindo Salgado (pablogsal) *** **Date: 2020-01-22 15:55**

AS this discussion will take a while and likely will have deeper consequences, in the meantime I created ~~PR18118~~ to specifically fix this.

msg360484 - (view) **Author: Dong-hee Na (corona10) *** **Date: 2020-01-22 16:13**

@pablogsal

I agree with hotfix is needed and also for discussion.
I left a comment for ~~PR 18118~~. Please take a look :)

msg360557 - (view) **Author: Pablo Galindo Salgado (pablogsal) *** **Date: 2020-01-23 14:07**

New changeset 79f89e6e5a659846d1068e8b1bd8e491ccdef861 by Pablo Galindo in branch 'master':
~~bpo-39421~~: Fix posible crash in heapq with custom comparison operators (~~GH-18118~~)
https://github.com/python/cpython/commit/79f89e6e5a659846d1068e8b1bd8e491ccdef861

msg360558 - (view) **Author: miss-islington (miss-islington)** **Date: 2020-01-23 14:25**

New changeset 958064f8d2b84062b0582bbae911df8ccfc11fd6 by Miss Islington (bot) in branch '3.7':
~~bpo-39421~~: Fix posible crash in heapq with custom comparison operators (~~GH-18118~~)
https://github.com/python/cpython/commit/958064f8d2b84062b0582bbae911df8ccfc11fd6

msg360561 - (view) **Author: Ned Deily (ned.deily) *** **Date: 2020-01-23 14:49**

New changeset c563f409ea30bcb0623d785428c9257917371b76 by Ned Deily (Miss Islington (bot)) in branch '3.6':
~~bpo-39421~~: Fix posible crash in heapq with custom comparison operators (~~GH-18118~~) (~~GH-18146~~)
https://github.com/python/cpython/commit/c563f409ea30bcb0623d785428c9257917371b76

msg360564 - (view) **Author: miss-islington (miss-islington)** **Date: 2020-01-23 15:22**

```
New changeset 993811ffe75c2573f97fb3fd1414b34609b8c8db by Miss Islington (bot) in
branch '3.8':
bpo-39421: Fix posible crash in heapq with custom comparison operators (GH-18118)
https://github.com/python/cpython/commit/993811ffe75c2573f97fb3fd1414b34609b8c8db
```

History

Date	User	Action	Args
2022-04-11 14:59:25	admin	set	github: 83602
2020-01-23 15:23:27	pablogsal	set	status: open -> closed resolution: fixed stage: patch review -> resolved
2020-01-23 15:22:29	miss-islington	set	messages: + **msg360564**
2020-01-23 15:05:37	miss-islington	set	pull_requests: + **pull_request17535**
2020-01-23 14:49:23	ned.deily	set	nosy: + **ned.deily** messages: + **msg360561**
2020-01-23 14:25:34	miss-islington	set	nosy: + **miss-islington** messages: + **msg360558**
2020-01-23 14:07:43	miss-islington	set	pull_requests: + **pull_request17532**
2020-01-23 14:07:37	miss-islington	set	pull_requests: + **pull_request17531**
2020-01-23 14:07:29	miss-islington	set	stage: needs patch -> patch review pull_requests: + **pull_request17530**
2020-01-23 14:07:16	pablogsal	set	messages: + **msg360557**
2020-01-23 13:54:44	alex	set	keywords: + **security_issue** nosy: + **alex**
2020-01-22 16:13:21	corona10	set	messages: + **msg360484**
2020-01-22 15:55:25	pablogsal	set	messages: + **msg360479** stage: patch review -> needs patch
2020-01-22 15:54:31	pablogsal	set	keywords: + **patch** stage: needs patch -> patch review pull_requests: + **pull_request17505**
2020-01-22 15:49:45	corona10	set	messages: + **msg360478**
2020-01-22 15:46:55	pablogsal	set	messages: + **msg360477**
2020-01-22 15:32:01	corona10	set	nosy: + **vstinner**
2020-01-22 15:31:53	corona10	set	stage: needs patch versions: - Python 3.6
2020-01-22 15:31:24	corona10	set	nosy: + **pablogsal**, **corona10**, **methane** messages: + **msg360475**
2020-01-22 15:13:43	dk0n9	set	messages: + **msg360474**
2020-01-22 15:11:46	dk0n9	create	