Ruby trunk - Bug #13438

Fix heap overflow due to configure.in not being updated for HEAP_* -> HEAP_PAGE_* variable renaming

04/14/2017 11:14 PM - jeremyevans0 (Jeremy Evans)

Status:	Closed		
Priority:	Normal		
Assignee:	jeremyevans0 (Jeremy Evans)		
Target version:			
ruby -v:	ruby 2.5.0dev (2017-04-15 trunk 58358) [x86_64-openbsd]	**Backport:**	2.2: DONTNEED, 2.3: DONTNEED, 2.4: REQUIRED, 2.5: REQUIRED

Description

An OpenBSD user reported that ruby 2.4.1 fails on OpenBSD when malloc canaries are enabled. I verified this is true, not just on ruby 2.4.1, but also on trunk:

```
MALLOC_OPTIONS=C ruby25 -v
ruby 2.5.0dev (2017-04-15 trunk 58358) [x86_64-openbsd]
ruby25(13588) in free(): chunk canary corrupted 0x8dbf2bb0000 0x3fd8@0x3fd8
Abort trap (core dumped)
```

MALLOC_OPTIONS=C here turns on malloc canaries. From the OpenBSD malloc.conf(5) man page:

```
     C       "Canaries".  Add canaries at the end of allocations in order to
             detect heap overflows.  The canary's content is checked when
             free(3) is called.  If it has been corrupted, the process is
             aborted.
```

So what we have here is a (probably small) heap overflow. Here's the backtrace:

```
(gdb) bt
#0  0x000008dcc9a1014a in thrkill () at {standard input}:5
#1  0x000008dcc99ead29 in *_libc_abort () at /usr/src/lib/libc/stdlib/abort.c:52
#2  0x000008dcc99e1346 in wrterror (d=0x7f7ffffceda0, msg=0x8dcc9b6fea0 "chunk canary corrupted %p
 %#tx@%#zx") at /usr/src/lib/libc/stdlib/malloc.c:306
#3  0x000008dcc99e1422 in validate_canary (d=Variable "d" is not available.
) at /usr/src/lib/libc/stdlib/malloc.c:1047
#4  0x000008dcc99e28ee in ofree (argpool=0x8dcd75392f0, p=0x8dbf2bb0000, clear=0) at /usr/src/lib/
libc/stdlib/malloc.c:1334
#5  0x000008dcc99e2bdd in free (ptr=0x8dbf2bb0000) at /usr/src/lib/libc/stdlib/malloc.c:1414
#6  0x000008dcb4ffdb52 in aligned_free (ptr=0x8dbf2bb0000) at gc.c:7678
#7  0x000008dcb4fef9df in heap_page_free (objspace=0x8dbf1f35400, page=0x8dc825ed800) at gc.c:1446
#8  0x000008dcb4fef752 in rb_objspace_free (objspace=0x8dbf1f35400) at gc.c:1341
#9  0x000008dcb515c168 in ruby_vm_destruct (vm=0x8dc8ca68c00) at vm.c:2191
#10 0x000008dcb4fe12c8 in ruby_cleanup (ex=0) at eval.c:227
#11 0x000008dcb4fe1528 in ruby_run_node (n=0x14) at eval.c:297
#12 0x000008d9eb600624 in main (argc=2, argv=0x7f7ffffcf248) at main.c:36
```

Note that this doesn't tell you where the heap overflow happened, it only shows where the heap overflow is detected.

I checked and ruby 1.8.7p374, 1.9.3p551, 2.0.0p648, 2.1.9, 2.2.7, and 2.3.4 do not suffer from this issue.

I determined via git bisect dea8ea61ea0bf08adb35be6ad47abe3ab955afc4 b58b970db5156766d6e19606d79afc68e4c2df7c the problem was introduced between r53467 and r53471.

With r53467 (git checkout 066b825400349c559aa3c1ca7769516c967c41b9):

```
ruby24 -v
ruby 2.4.0dev (2016-01-08 trunk 53467) [x86_64-openbsd]
# no error
```

With r53471 (git checkout fca0cf6e6b8ab8882f3403e5909c8eb91c5c351e):

```
ruby24 -v
ruby 2.4.0dev (2016-01-09 trunk 53471) [x86_64-openbsd]
ruby24(85533) in free(): chunk canary corrupted 0xf7bcfc1c000 0x3fd8@0x3fd8
Abort trap (core dumped)
```

There isn't much between r53467 and r53471 that could indicate a potential overflow introduction. All the diff does is rename variables. However, I noticed one part of the diff that is interesting:

```
-#ifndef HEAP_ALIGN_LOG
+#ifndef HEAP_PAGE_ALIGN_LOG
 /* default tiny heap size: 16KB */
-#define HEAP_ALIGN_LOG 14
+#define HEAP_PAGE_ALIGN_LOG 14
 #endif
```

If HEAP_ALIGN_LOG is already defined, it had an effect before, but no longer has an effect now. I searched for HEAP_ALIGN_LOG and sure enough it is still used by configure.in. Updating configure.in to change it to HEAP_PAGE_ALIGN_LOG fixes things.

Attached is a patch to fix this. This should be applied and backported to the 2.4 branch.

Associated revisions

Revision 0b899a25 - 11/29/2018 06:16 AM - shyouhei (Shyouhei Urabe)

Remove HEAP_ALIGN_LOG setting in configure.ac for OpenBSD/MirOS

The ruby setting was renamed to HEAP_PAGE_ALIGN_LOG, but the
configure.in (now configure.ac) file was not updated, so the
setting had no effect. The configure setting is unnecessary
after OpenBSD 5.2 and MirOS has been discontinued (with the last
release being over 10 years ago), so it is better to just remove
the related configure setting.

Fix [Bug #13438]
From: Jeremy Evans code@jeremyevans.net

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@66086 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 66086 - 11/29/2018 06:16 AM - shyouhei (Shyouhei Urabe)

Remove HEAP_ALIGN_LOG setting in configure.ac for OpenBSD/MirOS

The ruby setting was renamed to HEAP_PAGE_ALIGN_LOG, but the
configure.in (now configure.ac) file was not updated, so the
setting had no effect. The configure setting is unnecessary
after OpenBSD 5.2 and MirOS has been discontinued (with the last
release being over 10 years ago), so it is better to just remove
the related configure setting.

Fix [Bug #13438]
From: Jeremy Evans code@jeremyevans.net

Revision 66086 - 11/29/2018 06:16 AM - shyouhei (Shyouhei Urabe)

Remove HEAP_ALIGN_LOG setting in configure.ac for OpenBSD/MirOS

The ruby setting was renamed to HEAP_PAGE_ALIGN_LOG, but the
configure.in (now configure.ac) file was not updated, so the
setting had no effect. The configure setting is unnecessary
after OpenBSD 5.2 and MirOS has been discontinued (with the last
release being over 10 years ago), so it is better to just remove
the related configure setting.

Fix [Bug #13438]
From: Jeremy Evans code@jeremyevans.net

History

#1 - 04/15/2017 02:10 AM - ko1 (Koichi Sasada)

- Assignee set to ko1 (Koichi Sasada)

Thank you for your survey. Great help.

Just for confirmation.
Your patch doesn't solve memory problem, but for configuration, right?

Thanks,
Koichi

#2 - 04/15/2017 05:41 AM - jeremyevans0 (Jeremy Evans)

ko1 (Koichi Sasada) wrote:

> Just for confirmation.
> Your patch doesn't solve memory problem, but for configuration, right?

This is not just a configuration issue, this patch actually solves the heap overflow issue by making sure the correct #define is in config.h. Without this patch, on OpenBSD (and maybe other operating systems) HEAP_PAGE_ALIGN_LOG is set to 14 instead of 12, making HEAP_PAGE_SIZE and HEAP_PAGE_BITMAP_SIZE larger. I'm not actually sure where the overflow occurs, and I think the overflow may only be a single byte (0x3fd8@0x3fd8 I believe means that malloc allocated 16344 bytes and there was a write at offset 16344).

Thanks,
Jeremy

#3 - 04/16/2017 03:43 AM - jeremyevans0 (Jeremy Evans)

jeremyevans0 (Jeremy Evans) wrote:

> ko1 (Koichi Sasada) wrote:
>
> > Just for confirmation.
> > Your patch doesn't solve memory problem, but for configuration, right?
>
> This is not just a configuration issue, this patch actually solves the heap overflow issue by making sure the correct #define is in config.h. Without this patch, on OpenBSD (and maybe other operating systems) HEAP_PAGE_ALIGN_LOG is set to 14 instead of 12, making HEAP_PAGE_SIZE and HEAP_PAGE_BITMAP_SIZE larger. I'm not actually sure where the overflow occurs, and I think the overflow may only be a single byte (0x3fd8@0x3fd8 I believe means that malloc allocated 16344 bytes and there was a write at offset 16344).

Koichi,

After giving this some more thought, it's possible this doesn't fix the underlying memory issue. There are two possibilities:

1) The heap overflow only happens when the operating system uses <16kb pages and ruby is set to use 16k heap pages.

2) The heap overflow always happens when ruby uses 16kb heap pages.

If 1) is true, then this should be the only fix necessary. If 2) is true, then there is a separate memory issue that should be fixed. I suspect the problem is more likely 2), but this is outside my area of expertise.

In either case, I think this patch should be applied.

Thanks,
Jeremy

#4 - 04/17/2017 08:36 PM - jeremyevans0 (Jeremy Evans)

- File 0001-Fix-heap-overflow-by-allocating-more-memory-per-heap.patch added

jeremyevans0 (Jeremy Evans) wrote:

> 1) The heap overflow only happens when the operating system uses <16kb pages and ruby is set to use 16k heap pages.
>
> 2) The heap overflow always happens when ruby uses 16kb heap pages.
>
> If 1) is true, then this should be the only fix necessary. If 2) is true, then there is a separate memory issue that should be fixed. I suspect the problem is more likely 2), but this is outside my area of expertise.

I did some testing with different versions of HEAP_PAGE_ALIGN_LOG. Here's the results of my testing, with the first entry being HEAP_PAGE_ALIGN_LOG, the second being the page size, and the third being the result:

```
6  64B   Couldn't build: SIGFPE
7  128B  No error
8  256B  No error
9  512B  No error
```

```
10 1KB   No error
11 2KB   No error
12 4KB   No error
13 8KB   chunk canary corrupted 0x1fd8@0x1fd8
14 16KB  chunk canary corrupted 0x3fd8@0x3fd8
15 32KB  chunk canary corrupted 0x7fd8@0x7fd8
16 64KB  chunk canary corrupted 0xffd8@0xffd8
17 128KB chunk canary corrupted 0x1ffd8@0x1ffd8
18 256KB chunk canary corrupted 0x3ffd8@0x3ffd8
19 512KB Couldn't build: Failed to allocate memory
```

I first thought that when using >4KB pages, there is a heap overflow, but the heap overflow doesn't happen when using <=4KB pages. However, I think there may always be a heap overflow, even when using <=4KB pages. It turns out the OpenBSD malloc canary support is only turned on when allocating >=4KB. This leads me to believe the issue is that there is always a heap overflow, no matter the HEAP_PAGE_ALIGN_LOG value.

I tried increasing the size passed to aligned_malloc to see if I could determine the size of the overflow. It turns out that it overflows not by a single byte, but by 40 bytes. Coincidentally, that is also the value of REQUIRED_SIZE_BY_MALLOC. Maybe REQUIRED_SIZE_BY_MALLOC just needs to be added when calling aligned_malloc? I tried that and it appears to fix things.

The attached patch should fix the heap overflow for all page sizes that compile (tested on HEAP_PAGE_ALIGN 7..18).

#5 - 04/18/2017 12:33 AM - jeremyevans0 (Jeremy Evans)

jeremyevans0 (Jeremy Evans) wrote:

> I tried increasing the size passed to aligned_malloc to see if I could determine the size of the overflow. It turns out that it overflows not by a single byte, but by 40 bytes. Coincidentally, that is also the value of REQUIRED_SIZE_BY_MALLOC. Maybe REQUIRED_SIZE_BY_MALLOC just needs to be added when calling aligned_malloc? I tried that and it appears to fix things.
>
> The attached patch should fix the heap overflow for all page sizes that compile (tested on HEAP_PAGE_ALIGN 7..18).

After some more analysis and research, I don't think this patch to gc.c is necessary. I think this is a problem on OpenBSD when calling posix_memalign with allocations over 4KB that are slightly less than the aligned size when using malloc canaries, and that there isn't actually a heap overflow. So the patch to gc.c can be ignored, but the patch to configure.in should still be applied so that HEAP_PAGE_ALIGN_LOG is set correctly.

Thanks,
Jeremy

#6 - 04/18/2017 12:42 AM - ko1 (Koichi Sasada)

> I think this is a problem on OpenBSD when calling posix_memalign with allocations over 4KB that are slightly less than the aligned size when using malloc canaries

OMG. Thank you for your analysis.
So you mean we should reduce HEAP_PAGE_ALIGN_LOG value on OpenBSD?

#7 - 04/18/2017 02:09 AM - jeremyevans0 (Jeremy Evans)

ko1 (Koichi Sasada) wrote:

>> I think this is a problem on OpenBSD when calling posix_memalign with allocations over 4KB that are slightly less than the aligned size when using malloc canaries
>
> OMG. Thank you for your analysis.
> So you mean we should reduce HEAP_PAGE_ALIGN_LOG value on OpenBSD?

I don't think we need to make any changes to gc.c if configure.in is fixed so that HEAP_PAGE_ALIGN_LOG is set correctly.

Is there a reason that heap page allocation sizes are reduced by REQUIRED_SIZE_BY_MALLOC (40 bytes)? It seems like it would be better if HEAP_PAGE_ALIGN and HEAP_PAGE_SIZE were the same. The only reason I can think of is to work around a performance issue in a malloc implementation that stores metadata in the 40 bytes after the allocation. On OpenBSD, having them the same size can be better for security, because if you enable guard pages any overflow for heap pages would result in a segmentation fault.

Thanks,
Jeremy

#8 - 04/18/2017 04:23 PM - jeremyevans0 (Jeremy Evans)

jeremyevans0 (Jeremy Evans) wrote:

> After some more analysis and research, I don't think this patch to gc.c is necessary. I think this is a problem on OpenBSD when calling posix_memalign with allocations over 4KB that are slightly less than the aligned size when using malloc canaries, and that there isn't actually a heap overflow. So the patch to gc.c can be ignored, but the patch to configure.in should still be applied so that HEAP_PAGE_ALIGN_LOG is set correctly.

I've confirmed this was a problem with OpenBSD's posix_memalign when using malloc canaries. It has been fixed in OpenBSD-current. So this can be closed once the configure.in patch is committed. I think it should be backported to 2.4, to ensure that ruby GC respects the operating system page size (as it did in ruby 2.3 and below).

Thanks,
Jeremy

#9 - 04/18/2017 11:21 PM - ko1 (Koichi Sasada)

On 2017/04/19 1:23, merch-redmine@jeremyevans.net wrote:

> I think it should be backported to 2.4, to ensure that ruby GC respects the operating system page size (as it did in ruby 2.3 and below).

Note that gc's PAGE is not relative to OS/machie page.

--
// SASADA Koichi at atdot dot net

#10 - 04/18/2017 11:50 PM - jeremyevans0 (Jeremy Evans)

- Backport changed from 2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: REQUIRED to 2.2: DONTNEED, 2.3: DONTNEED, 2.4: DONTNEED

- File 0001-Remove-overriding-of-HEAP_PAGE_ALIGN_LOG.patch added

ko1 (Koichi Sasada) wrote:

> On 2017/04/19 1:23, merch-redmine@jeremyevans.net wrote:
>
> > I think it should be backported to 2.4, to ensure that ruby GC respects the operating system page size (as it did in ruby 2.3 and below).
>
> Note that gc's PAGE is not relative to OS/machie page.

Ah. I see. Reading more of configure.in, HEAP_ALIGN_LOG is only set to 12 or 13 support OpenBSD <5.2 and MirOS. At this point, we may want to drop OpenBSD <5.2 support, or maybe the entire branch unless there are MirOS #10semel users. The MirOS #10semel release was over 9 years ago, and the recommended development snapshots work with the standard 16kb pages according to the comment in configure.in. Attached is a patch that remove the HEAP_ALIGN_LOG setting in configure.in and HEAP_PAGE_ALIGN_LOG overriding in gc.c.

Thanks,
Jeremy

#11 - 04/28/2017 01:45 PM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Assigned

#12 - 05/19/2017 02:42 AM - ko1 (Koichi Sasada)

- Assignee changed from ko1 (Koichi Sasada) to jeremyevans0 (Jeremy Evans)

Sorry for late response.

> we may want to drop OpenBSD <5.2 support

On https://bugs.ruby-lang.org/projects/ruby-trunk/wiki/SupportedPlatforms

You can decide it. I don't have any idea we should or we shouldn't. Could you decide and commit it?

Thanks,
Koichi

#13 - 05/23/2017 04:57 PM - jeremyevans0 (Jeremy Evans)

ko1 (Koichi Sasada) wrote:

> > we may want to drop OpenBSD <5.2 support

> On https://bugs.ruby-lang.org/projects/ruby-trunk/wiki/SupportedPlatforms
>
> You can decide it. I don't have any idea we should or we shouldn't. Could you decide and commit it?

I think we should drop OpenBSD <5.2 support. I doubt anyone running such an old version of OpenBSD would want to run ruby 2.5+. The OpenBSD project only supports OpenBSD 6.1 and 6.0 currently.

I don't currently have commit rights, so I can't currently commit the patch myself. Hopefully another developer can commit it.

Thanks,
Jeremy

#14 - 09/25/2017 12:11 PM - shyouhei (Shyouhei Urabe)

We looked at this issue several times in recent developer meeting and it seems okay to merge.

#15 - 06/06/2018 07:31 PM - jeremyevans0 (Jeremy Evans)

- File 0001-Remove-HEAP_ALIGN_LOG-setting-in-configure.ac-for-Op.patch added

shyouhei (Shyouhei Urabe) wrote:

> We looked at this issue several times in recent developer meeting and it seems okay to merge.

This hasn't been merged yet, and the most recent patch was before the configure.in to configure.ac renaming. Attached is an updated patch that should apply to current trunk.

#16 - 11/29/2018 04:40 AM - jeremyevans0 (Jeremy Evans)

jeremyevans0 (Jeremy Evans) wrote:

> shyouhei (Shyouhei Urabe) wrote:
>
> > We looked at this issue several times in recent developer meeting and it seems okay to merge.
>
> This hasn't been merged yet, and the most recent patch was before the configure.in to configure.ac renaming. Attached is an updated patch that should apply to current trunk.

This was approved back in September 2017, but it didn't make the 2.5 release. I would like it to make the 2.6 release. The latest patch still applies (with offset). Could a committer please merge it?

#17 - 11/29/2018 06:16 AM - shyouhei (Shyouhei Urabe)

- Status changed from Assigned to Closed

Applied in changeset trunk|r66086.

Remove HEAP_ALIGN_LOG setting in configure.ac for OpenBSD/MirOS

The ruby setting was renamed to HEAP_PAGE_ALIGN_LOG, but the
configure.in (now configure.ac) file was not updated, so the
setting had no effect. The configure setting is unnecessary
after OpenBSD 5.2 and MirOS has been discontinued (with the last
release being over 10 years ago), so it is better to just remove
the related configure setting.

Fix [Bug #13438]
From: Jeremy Evans code@jeremyevans.net

#18 - 11/29/2018 06:19 AM - shyouhei (Shyouhei Urabe)

Oops, stunned that nobody noticed this issue for over a yer.
Had just pushed the patch. Sorry for the delay!

#19 - 11/29/2018 06:56 AM - mame (Yusuke Endoh)

- Backport changed from 2.2: DONTNEED, 2.3: DONTNEED, 2.4: DONTNEED to 2.2: DONTNEED, 2.3: DONTNEED, 2.4: REQUIRED, 2.5: REQUIRED

Files

0001-Fix-heap-overflow-if-system-uses-16kb-pages.patch	1.12 KB	04/14/2017	jeremyevans0 (Jeremy Evans)
0001-Fix-heap-overflow-by-allocating-more-memory-per-heap.patch	929 Bytes	04/17/2017	jeremyevans0 (Jeremy Evans)
0001-Remove-overriding-of-HEAP_PAGE_ALIGN_LOG.patch	2.28 KB	04/18/2017	jeremyevans0 (Jeremy Evans)
0001-Remove-HEAP_ALIGN_LOG-setting-in-configure.ac-for-Op.patch	2.28 KB	06/06/2018	jeremyevans0 (Jeremy Evans)