New issue

[PATCH] Stack overflow in Storable retrieve_hook #15831

Closed

Labels distro-All hasPatch type-library

p5pRT opened on Jan 24, 2017

Migrated from rt.perl.org#130635 (status was 'resolved')

Searchable as RT130635$

p5pRT on Jan 24, 2017 Author

From @lightsey

Created by @lightsey

This is a bug report for perl from john@nixnuts.net,
generated with the help of perlbug 1.40 running under perl 5.25.9.

AFL detected a stack overflow in Storable's retrieve_hook() function.

The problem essentially is that a hook's classname length is read into a signed integer, compared to the size of a stack buffer, then used to read the classname. The size comparison treats the length as signed, while the read treats the length as unsigned.

▶ Perl Info

p5pRT on Jan 24, 2017 Author

From @lightsey

▶ 0001-Fix-stack-buffer-overflow-in-deserialization-of-hook.patch

p5pRT on Jan 25, 2017 Author •••

From @jkeenan

On Tue, 24 Jan 2017 19:22:28 GMT, john@nixnuts.net wrote:

> This is a bug report for perl from john@nixnuts.net,
> generated with the help of perlbug 1.40 running under perl 5.25.9.
>
> ---
> AFL detected a stack overflow in Storable's retrieve_hook() function.
>
> The problem essentially is that a hook's classname length is read into
> a signed integer, compared to the size of a stack buffer, then used to
> read the classname. The size comparison treats the length as signed,
> while the read treats the length as unsigned.

Available for smoke-testing in this branch:

smoke-me/jkeenan/130635-storable

I corrected one spelling error in a test description and incremented the VERSION number.

--
James E Keenan (jkeenan@cpan.org)

p5pRT on Jan 25, 2017 Author •••

The RT System itself - Status changed from 'new' to 'open'

p5pRT on Jan 25, 2017 Author •••

From @jkeenan

On Wed, 25 Jan 2017 02:05:45 GMT, jkeenan wrote:

> On Tue, 24 Jan 2017 19:22:28 GMT, john@nixnuts.net wrote:
>
> > This is a bug report for perl from john@nixnuts.net,
> > generated with the help of perlbug 1.40 running under perl 5.25.9.

> ---
> AFL detected a stack overflow in Storable's retrieve_hook() function.
>
> The problem essentially is that a hook's classname length is read
> into
> a signed integer, compared to the size of a stack buffer, then used
> to
> read the classname. The size comparison treats the length as signed,
> while the read treats the length as unsigned.
>
> Available for smoke-testing in this branch:
>
> smoke-me/jkeenan/130635-storable
>
> I corrected one spelling error in a test description and incremented
> the VERSION number.

This revision failed on FreeBSD-11. See: http://perl5.test-smoke.org/report/53470

When I ran the test file individually, it hung at 'ok 24', then printed these error messages:

#####
swap_pager: out of swap space
swap_pager_getswapspace(16): failed
swap_pager_getswapspace(16): failed
swap_pager_getswapspace(16): failed
#####

kernel: pid 18627 (perl), uid 1001, was killed: out of swap space

So something is clearly amiss with this patch.

Thank you very much.

--
James E Keenan (jkeenan@cpan.org)

p5pRT on Jan 25, 2017 Author •••

From @jkeenan

On Wed, 25 Jan 2017 04:04:15 GMT, jkeenan wrote:

> On Wed, 25 Jan 2017 02:05:45 GMT, jkeenan wrote:
>
> > On Tue, 24 Jan 2017 19:22:28 GMT, john@nixnuts.net wrote:

> > This is a bug report for perl from john@nixnuts.net,
> > generated with the help of perlbug 1.40 running under perl 5.25.9.
> >
> > ----------
> > AFL detected a stack overflow in Storable's retrieve_hook()
> > function.
> >
> > The problem essentially is that a hook's classname length is read
> > into
> > a signed integer, compared to the size of a stack buffer, then used
> > to
> > read the classname. The size comparison treats the length as
> > signed,
> > while the read treats the length as unsigned.
>
> Available for smoke-testing in this branch:
>
> smoke-me/jkeenan/130635-storable
>
> I corrected one spelling error in a test description and incremented
> the VERSION number.

This revision failed on FreeBSD-11. See: http://perl5.test-smoke.org/report/53470

When I ran the test file individually, it hung at 'ok 24', then
printed these error messages:

#####
swap_pager: out of swap space
swap_pager_getswapspace(16): failed
swap_pager_getswapspace(16): failed
swap_pager_getswapspace(16): failed
#####

kernel: pid 18627 (perl), uid 1001, was killed: out of swap space

So something is clearly amiss with this patch.

Thank you very much.

Similar results on FreeBSD-10.3:
http://perl5.test-smoke.org/report/53481
--
James E Keenan (jkeenan@cpan.org)

p5pRT on Jan 25, 2017 Author •••

From @jkeenan

On Wed, 25 Jan 2017 15:21:36 GMT, jkeenan wrote:

> On Wed, 25 Jan 2017 04:04:15 GMT, jkeenan wrote:
>
> > On Wed, 25 Jan 2017 02:05:45 GMT, jkeenan wrote:
> >
> > > On Tue, 24 Jan 2017 19:22:28 GMT, john@nixnuts.net wrote:
> > >
> > > > This is a bug report for perl from john@nixnuts.net,
> > > > generated with the help of perlbug 1.40 running under perl 5.25.9.
> > > >
> > > > ----------
> > > > AFL detected a stack overflow in Storable's retrieve_hook()
> > > > function.
> > > >
> > > > The problem essentially is that a hook's classname length is read
> > > > into
> > > > a signed integer, compared to the size of a stack buffer, then used
> > > > to
> > > > read the classname. The size comparison treats the length as
> > > > signed,
> > > > while the read treats the length as unsigned.
> > >
> > > Available for smoke-testing in this branch:
> > >
> > > smoke-me/jkeenan/130635-storable
> > >
> > > I corrected one spelling error in a test description and incremented
> > > the VERSION number.
> >
> > This revision failed on FreeBSD-11. See: http://perl5.test-
> > smoke.org/report/53470
> >
> > When I ran the test file individually, it hung at 'ok 24', then
> > printed these error messages:
> >
> > #####
> > swap_pager: out of swap space
> > swap_pager_getswapspace(16): failed
> > swap_pager_getswapspace(16): failed
> > swap_pager_getswapspace(16): failed
> > #####
> >
> > kernel: pid 18627 (perl), uid 1001, was killed: out of swap space
> >
> > So something is clearly amiss with this patch.
> >
> > Thank you very much.

> Similar results on FreeBSD-10.3:
> http://perl5.test-smoke.org/report/53481

On (at least) FreeBSD-10.3, it appears that the test failure occurs when I configure with -DDEBUGGING:

#####
[perl] $ gitcurr
130635-storable
[perl] $ git show | head -1
commit `706cfb1`
[perl] $ ./perl -Ilib -V | grep config_args
  config_args='-des -Dusedevel -Duseithreads -Doptimize=-O2 -pipe -fstack-protector -fno-strict-aliasing -DDEBUGGING'
#####
cd t;./perl harness -v ../dist/Storable/t/store.t; cd -
...
ok 24 - RT 130098: no segfault in Storable::fd_retrieve()

Failed test 'No stack smashing error when retrieving hook'
at t/store.t line 109.
Looks like you failed 1 test of 25.
not ok 25 - No stack smashing error when retrieving hook
Dubious, test returned 1 (wstat 256, 0x100)
Failed 1/25 subtests
...
#####

In the same branch, at the same commit, configuring exactly the same except without -DDEBUGGING, the test file completes successfully.

--
James E Keenan (jkeenan@cpan.org)

p5pRT on Jan 25, 2017 Author •••

From @lightsey

On Wed, 2017-01-25 at 07:21 -0800, James E Keenan via RT wrote:

> On Wed, 25 Jan 2017 04:04:15 GMT, jkeenan wrote:

> > This revision failed on FreeBSD-11.Â Â See:Â Â http://perl5.test-smoke.org/report/53470
> > Â
> > When I ran the test file individually, it hung at 'ok 24', then
> > printed these error messages:
> > Â
> > #####
> > swap_pager: out of swap space
> > swap_pager_getswapspace(16): failed
> > swap_pager_getswapspace(16): failed
> > swap_pager_getswapspace(16): failed
> > #####
> > Â
> > kernel: pid 18627 (perl), uid 1001, was killed: out of swap space
> > Â
> > So something is clearly amiss with this patch.
> > Â
> > Thank you very much.
>
> Similar results on FreeBSD-10.3:
> http://perl5.test-smoke.org/report/53481

Makes sense.. When the bug is fixed, it ends up allocating an insanely large amount of memory. On Linux, malloc doesn't even try to allocate the memory. It sounds like BSD does try to allocate it.

Maybe it's possible to limit the maximum allowed classname length to what Perl actually supports. I'll try and adjust the patch to do this.

p5pRT on Jan 25, 2017 Author •••

From @jkeenan

On Wed, 25 Jan 2017 15:42:21 GMT, jkeenan wrote:

> On Wed, 25 Jan 2017 15:21:36 GMT, jkeenan wrote:
>
> > On Wed, 25 Jan 2017 04:04:15 GMT, jkeenan wrote:
> >
> > > On Wed, 25 Jan 2017 02:05:45 GMT, jkeenan wrote:
> > >
> > > > On Tue, 24 Jan 2017 19:22:28 GMT, john@nixnuts.net wrote:
> > > >
> > > > > This is a bug report for perl from john@nixnuts.net,
> > > > > generated with the help of perlbug 1.40 running under perl
> > > > > 5.25.9.

> > > > > > ---
> > > > > > AFL detected a stack overflow in Storable's retrieve_hook()
> > > > > > function.
> > > > > >
> > > > > > The problem essentially is that a hook's classname length is
> > > > > > read
> > > > > > into
> > > > > > a signed integer, compared to the size of a stack buffer, then
> > > > > > used
> > > > > > to
> > > > > > read the classname. The size comparison treats the length as
> > > > > > signed,
> > > > > > while the read treats the length as unsigned.
> > > > >
> > > > > Available for smoke-testing in this branch:
> > > > >
> > > > > smoke-me/jkeenan/130635-storable
> > > > >
> > > > > I corrected one spelling error in a test description and
> > > > > incremented
> > > > > the VERSION number.
> > > >
> > > > This revision failed on FreeBSD-11. See: http://perl5.test-
> > > > smoke.org/report/53470
> > > >
> > > > When I ran the test file individually, it hung at 'ok 24', then
> > > > printed these error messages:
> > > >
> > > > #####
> > > > swap_pager: out of swap space
> > > > swap_pager_getswapspace(16): failed
> > > > swap_pager_getswapspace(16): failed
> > > > swap_pager_getswapspace(16): failed
> > > > #####
> > > >
> > > > kernel: pid 18627 (perl), uid 1001, was killed: out of swap space
> > > >
> > > > So something is clearly amiss with this patch.
> > > >
> > > > Thank you very much.
> > >
> > > Similar results on FreeBSD-10.3:
> > > http://perl5.test-smoke.org/report/53481
>
> On (at least) FreeBSD-10.3, it appears that the test failure occurs
> when I configure with -DDEBUGGING:

> #####
> [perl] $ gitcurr
> 130635-storable
> [perl] $ git show | head -1
> commit `706cfb1`
> [perl] $ ./perl -Ilib -V | grep config_args
> config_args='-des -Dusedevel -Duseithreads -Doptimize=-O2 -pipe
> -fstack-protector -fno-strict-aliasing -DDEBUGGING'
> #####
> cd t;./perl harness -v ../dist/Storable/t/store.t; cd -
> ...
> ok 24 - RT 130098: no segfault in Storable::fd_retrieve()
>
> # Failed test 'No stack smashing error when retrieving hook'
> # at t/store.t line 109.
> # Looks like you failed 1 test of 25.
> not ok 25 - No stack smashing error when retrieving hook
> Dubious, test returned 1 (wstat 256, 0x100)
> Failed 1/25 subtests
> ...
> #####
>
> In the same branch, at the same commit, configuring exactly the same
> except without -DDEBUGGING, the test file completes successfully.

On FreeBSD-11, I get the same difference in results between with/without -DDEBUGGING.

On Linux, with -DDEBUGGING, the test file eventually completes ... but clearly hangs for some time after 'ok 21' and takes 30s to run.

So the patch does not play well with -DDEBUGGING regardless of OS.

Thank you very much.
--
James E Keenan (jkeenan@cpan.org)

p5pRT on Jan 25, 2017 Author •••

From @lightsey

On Wed, 2017-01-25 at 08:12 -0800, James E Keenan via RT wrote:

> On FreeBSD-11, I get the same difference in results between with/without
> -DDEBUGGING.
>
> On Linux, with -DDEBUGGING, the test file eventually completes ... but clearly
> hangs for some time after 'ok 21' and takes 30s to run.

> So the patch does not play well with -DDEBUGGING regardless of OS.
>
> Thank you very much.

I'm attaching an updated patch that croaks before these oversized New() calls
rather than trying to handle the failures they generate.

p5pRT on Jan 25, 2017 Author •••

From @lightsey

▶ 0001-Fix-stack-buffer-overflow-in-deserialization-of-hook.patch

p5pRT on Jan 25, 2017 Author •••

From @jkeenan

On Wed, 25 Jan 2017 19:44:02 GMT, john@nixnuts.net wrote:

> On Wed, 2017-01-25 at 08:12 -0800, James E Keenan via RT wrote:
>
> > On FreeBSD-11, I get the same difference in results between
> > with/without
> > -DDEBUGGING.
> >
> > On Linux, with -DDEBUGGING, the test file eventually completes ...
> > but clearly
> > hangs for some time after 'ok 21' and takes 30s to run.
> >
> > So the patch does not play well with -DDEBUGGING regardless of OS.
> >
> > Thank you very much.
>
> I'm attaching an updated patch that croaks before these oversized
> New() calls
> rather than trying to handle the failures they generate.

On Linux, with a -DDEBUGGING build, the runtime of the test came down to a reasonable length.

However on both FreeBSD 10 and 11, the results under -DDEBUGGING were still unsatisfactory. Running the test manually on 10.3, I had to Ctrl-C the test after 2 minutes. Running smoke tests on 11, it appears that the test completed successfully once but then got the same swap errors previously reported, leading to the curious result of "PASS-so-far":
http://perl5.test-smoke.org/report/53487
--
James E Keenan (jkeenan@cpan.org)

p5pRT on Jan 26, 2017 Author •••

From @lightsey

On Wed, 2017-01-25 at 14:31 -0800, James E Keenan via RT wrote:

> On Wed, 25 Jan 2017 19:44:02 GMT, john@nixnuts.net wrote:
>
> > On Wed, 2017-01-25 at 08:12 -0800, James E Keenan via RT wrote:
> >
> > > On FreeBSD-11, I get the same difference in results between
> > > with/without
> > > -DDEBUGGING.
> > >
> > > On Linux, with -DDEBUGGING, the test file eventually completes ...
> > > but clearly
> > > hangs for some time after 'ok 21' and takes 30s to run.
> > >
> > > So the patch does not play well with -DDEBUGGING regardless of OS.
> > >
> > > Thank you very much.
> > > Â
> > > I'm attaching an updated patch that croaks before these oversized
> > > New() calls
> > > rather than trying to handle the failures they generate.
>
> On Linux, with a -DDEBUGGING build, the runtime of the test came down to a reasonable length.
>
> However on both FreeBSD 10 and 11, the results under -DDEBUGGING were still unsatisfactory.Â Â Running the test manually on 10.3, I had to Ctrl-C the test after 2 minutes.Â Â Running smoke tests on 11, it appears that the test completed successfully once but then got the same swap errors previously reported, leading to the curious result of "PASS-so-far":
> http://perl5.test-smoke.org/report/53487

I brought up a FreeBSD 10.3 AMD64 VM to test and saw a similar issue when I ran the new unit test with the older version of Storable.Â The long pause on BSD seemed to be caused by coredumps being enabled (default on FreeBSD, not default on Linux.)

I didn't see any crashes on the FreeBSD VM when testing blead with the new version of the patch. The configure settings I used were:

./Configure -des -DDEBUGGING -Dusedevel

Is it possible that some of the failures you're seeing were caused by using the wrong version of Storable?

p5pRT on Jan 26, 2017 Author •••

From @jkeenan

On Thu, 26 Jan 2017 21:02:12 GMT, john@nixnuts.net wrote:

> On Wed, 2017-01-25 at 14:31 -0800, James E Keenan via RT wrote:
>
> > On Wed, 25 Jan 2017 19:44:02 GMT, john@nixnuts.net wrote:
> >
> > > On Wed, 2017-01-25 at 08:12 -0800, James E Keenan via RT wrote:
> > >
> > > > On FreeBSD-11, I get the same difference in results between
> > > > with/without
> > > > -DDEBUGGING.
> > > >
> > > > On Linux, with -DDEBUGGING, the test file eventually completes
> > > > ...
> > > > but clearly
> > > > hangs for some time after 'ok 21' and takes 30s to run.
> > > >
> > > > So the patch does not play well with -DDEBUGGING regardless of
> > > > OS.
> > > >
> > > > Thank you very much.
> > >
> > > I'm attaching an updated patch that croaks before these oversized
> > > New() calls
> > > rather than trying to handle the failures they generate.
> >
> > On Linux, with a -DDEBUGGING build, the runtime of the test came down
> > to a
> > reasonable length.

> > However on both FreeBSD 10 and 11, the results under -DDEBUGGING were still
> > unsatisfactory.Â Â Running the test manually on 10.3, I had to Ctrl-C the test
> > after 2 minutes.Â Â Running smoke tests on 11, it appears that the test
> > completed successfully once but then got the same swap errors previously
> > reported, leading to the curious result of "PASS-so-far":
> > http://perl5.test-smoke.org/report/53487
>
> I brought up a FreeBSD 10.3 AMD64 VM to test and saw a similar issue when I ran
> the new unit test with the older version of Storable.Â The long pause on BSD
> seemed to be caused by coredumps being enabled (default on FreeBSD, not default
> on Linux.)
>
> I didn't see any crashes on the FreeBSD VM when testing blead with the new
> version of the patch. The configure settings I used were:
>
> ./Configure -des -DDEBUGGING -Dusedevel
>
> Is it possible that some of the failures you're seeing were caused by using the
> wrong version of Storable?

I used the most recent version of this branch in the perl 5 repository:

smoke-me/jkeenan/130635-storable

... which I believe incorporates your revised patch.

As previously reported, I configure with:

"-des -Dusedevel -Duseithreads -Doptimize='-O2 -pipe -fstack-protector -fno-strict-aliasing' -DDEBUGGING"

... because that gets us very close to the way that the FreeBSD port of perl is configured.

Thank you very much.

--
James E Keenan (jkeenan@cpan.org)

p5pRT on Jan 29, 2017 Author •••

From @lightsey

On Thu, 2017-01-26 at 13:48 -0800, James E Keenan via RT wrote:

> As previously reported, I configure with:
>
> "-des -Dusedevel -Duseithreads -Doptimize='-O2 -pipe -fstack-protector -fno-strict-aliasing' -DDEBUGGING"
>
> ... because that gets us very close to the way that the FreeBSD port of perl is configured.

Excellent, thanks.

The problem turned out to be that the AFL generated payload was hitting two other memory allocation errors before it even entered retrieve_hook(). That combination of flags on FreeBSD seems to crash whenever Storable tries to allocate too much memory.

I adjusted the test data to use more realistic sizes when it enters retrieve_hash() and retrieve_flag_hash() so that it's only focusing on the stack overflow in retrieve_hook().

I also cleaned up the test output formatting a bit.

An updated patch is attached.

2 remaining items

Load more

p5pRT on Jan 30, 2017 Author •••

From @lightsey

On Sun, 2017-01-29 at 21:12 -0800, Tony Cook via RT wrote:

> On Sat, 28 Jan 2017 16:41:50 -0800, john@nixnuts.net wrote:
>
> > An updated patch is attached.
>
> Won't the allocated buffer leak if the load_module() fails, no STORABLE_thaw is defined, or if STORABLE_thaw dies?

Yes, Storable has several codepaths where it leaks memory or crashes while allocating memory. I'd be happy to try and fix these problems more systematically, but IMHO they are separate issues from the stack overflow.

p5pRT on Jan 30, 2017 Author •••

From @jkeenan

On Sun, 29 Jan 2017 00:41:50 GMT, john@nixnuts.net wrote:

> On Thu, 2017-01-26 at 13:48 -0800, James E Keenan via RT wrote:
>
> > As previously reported, I configure with:
> >
> > "-des -Dusedevel -Duseithreads -Doptimize='-O2 -pipe -fstack-protector -fno-strict-aliasing' -DDEBUGGING"
> >
> > ... because that gets us very close to the way that the FreeBSD port of perl is configured.
>
> Excellent, thanks.
>
> The problem turned out to be that the AFL generated payload was hitting two other memory allocation errors before it even entered retrieve_hook(). That combination of flags on FreeBSD seems to crash whenever Storable tries to allocate too much memory.
>
> I adjusted the test data to use more realistic sizes when it enters retrieve_hash() and retrieve_flag_hash() so that it's only focusing on the stack overflow in retrieve_hook().
>
> I also cleaned up the test output formatting a bit.
>
> An updated patch is attached.

The third patch you submitted differs from the second patch submitted only in some test input data in t/store.t. You haven't changed any source code. Are you sure you're not simply fudging the test?

Thank you very much.

--
James E Keenan (jkeenan@cpan.org)

p5pRT on Jan 30, 2017 Author •••

From @lightsey

On Mon, 2017-01-30 at 09:58 -0800, James E Keenan via RT wrote:

> On Sun, 29 Jan 2017 00:41:50 GMT, john@nixnuts.net wrote:
>
> > On Thu, 2017-01-26 at 13:48 -0800, James E Keenan via RT wrote:
> >
> > > As previously reported, I configure with:
> > >
> > > "-des -Dusedevel -Duseithreads -Doptimize='-O2 -pipe -fstack-protector -fno-
> > > strict-aliasing' -DDEBUGGING"
> > >
> > > ... because that gets us very close to the way that the FreeBSD port of perl
> > > is configured.
> >
> > Â
> > Excellent, thanks.
> > Â
> > The problem turned out to be that the AFL generated payload was hitting two
> > other memory allocation errors before it even entered retrieve_hook(). That
> > combination of flags on FreeBSD seems to crash whenever Storable tries to
> > allocate too much memory.
> > Â
> > I adjusted the test data to use more realistic sizes when it enters
> > retrieve_hash() and retrieve_flag_hash() so that it's only focusing on the stack
> > overflow in retrieve_hook().
> > Â
> > I also cleaned up the test output formatting a bit.
> > Â
> > An updated patch is attached.
>
> The third patch you submitted differs from the second patch submitted only in some test input data in t/store.t.Â Â You haven't changed any source code.Â Â Are you sure you're not simply fudging the test?

Yes, that's accurate. The original payload generated by AFL hit three different errors:

- memory allocation failure in retrieve_hash
- memory allocation failure in retrieve_flag_hash
- stack overflow in retrieve_hook

The retrieve_hash() crash, for instance, is essentially this code:

RLEN(len);
hv = newHV();
hv_ksplit(hv, len + 1);

For some reason, FreeBSD with the flags you provided can't handle this when len is too large to allocate.

My patch only fixes the stack overflow, so I changed the test payload to only generate the stack overflow error.

p5pRT on Jan 30, 2017 Author •••

From @tonycoz

On Mon, 30 Jan 2017 09:33:44 -0800, john@nixnuts.net wrote:

> On Sun, 2017-01-29 at 21:12 -0800, Tony Cook via RT wrote:
>
> > On Sat, 28 Jan 2017 16:41:50 -0800, john@nixnuts.net wrote:
> >
> > > An updated patch is attached.
> >
> > Won't the allocated buffer leak if the load_module() fails, no STORABLE_thaw
> > is defined, or if STORABLE_thaw dies?
>
> Yes, Storable has several codepaths where it leaks memory or crashes while
> allocating memory. I'd be happy to try and fix these problems more
> systematically, but IMHO they are separate issues from the stack
> overflow.

There's no need to add more leaks though.

You can use SAVEFREEPV() to free the allocated memory on scope exit.

Tony

p5pRT on Jan 30, 2017 Author •••

From @lightsey

On Mon, 2017-01-30 at 14:17 -0800, Tony Cook via RT wrote:

> On Mon, 30 Jan 2017 09:33:44 -0800, john@nixnuts.net wrote:
>
> > On Sun, 2017-01-29 at 21:12 -0800, Tony Cook via RT wrote:
> >
> > > On Sat, 28 Jan 2017 16:41:50 -0800, john@nixnuts.net wrote:
> > >
> > > > An updated patch is attached.
> > >
> > > Won't the allocated buffer leak if the load_module() fails, no
> > > STORABLE_thaw
> > > is defined, or if STORABLE_thaw dies?
> >
> > Â
> > Yes, Storable has several codepaths where it leaks memory or crashes
> > while
> > allocatingÂ Â memory. I'd be happy to try and fix these problems more
> > systematically, but IMHO they are separate issues from the stack
> > overflow.
>
> There's no need to add more leaks though.
>
> You can use SAVEFREEPV() to free the allocated memory on scope exit.

It's not clear to me what memory you believe my patch is leaking that wasn't leaking in the original code.

The two assertions I added are before the matching calls to New().

Which variable have I created a new memory leak on?

p5pRT on Feb 6, 2017 Author •••

From @tonycoz

On Mon, 30 Jan 2017 14:53:44 -0800, john@nixnuts.net wrote:

> It's not clear to me what memory you believe my patch is leaking that wasn't leaking in the original code.
>
> The two assertions I added are before the matching calls to New().
>
> Which variable have I created a new memory leak on?

You're right.

Thanks, I've applied your patch as `3e998dd`.

I've fixed that specific leak in `da1ec2b`.

Tony

p5pRT on Feb 6, 2017 Author

@**tonycoz** - Status changed from 'open' to 'pending release'

p5pRT on May 30, 2017 Author

From @khwilliamson

Thank you for filing this report. You have helped make Perl better.

With the release today of Perl 5.26.0, this and 210 other issues have been resolved.

Perl 5.26.0 may be downloaded via:
https://metacpan.org/release/XSAWYERX/perl-5.26.0

If you find that the problem persists, feel free to reopen this ticket.

p5pRT on May 30, 2017 Author

@**khwilliamson** - Status changed from 'pending release' to 'resolved'

p5pRT closed this as completed on May 30, 2017

p5pRT added **Severity Low** **affects-5.25** **distro-All** **hasPatch** **type-library** on Oct 19, 2019

Metadata

Assignees

No one assigned

Labels

distro-All hasPatch type-library

Type

No type

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

No branches or pull requests

Participants