Reporting and tracking regressions across the ecosystem

Gustavo Padovan (Collabora)
Ricardo Cañuelo (Collabora)
Thorsten Leemhuis
Regressions are hard!

Getting regressions reported, tracked and fixed is a complex process.
Goals for the BoF

Hear you beyond the frustration.

- What can we do to improve?
- What are some of the key bottlenecks (beyond maintainers time)?
- How can regzbot and CI systems help more?

Before we dive into the discussion, short update of **regzbot** and other open topics that we missed on Monday’s talk
Thorsten's kernel regression tracking efforts in a nutshell
Doing regression tracking for ~2 years now
Doing it with the help of regzbot, a "regression tracking bot" I wrote
A regression report from Jiri in ZHc2fm+9daF6cgCE@krava illustrates how it works.
From: Jiri Olsa <olsajiri@gmail.com>
To: Lorenzo Stoakes <lstoakes@gmail.com>
Cc: linux-mm@kvack.org, linux-kernel@vger.kernel.org, linux-fsdevel@vger.kernel.org, [...]
Subject: Re: [PATCH v8 1/4] fs/proc/kcore: avoid bounce buffer for ktext data
Date: Wed, 31 May 2023 13:58:54 +0200  [thread overview]
Message-ID: <Zhc2fm+9daF6cgCEkrava> (raw)
In-Reply-To: <fd39b0bfa7edc76d360def7d034baee71d90158.1679566220.git.lstoakes@gmail.com>

On Thu, Mar 23, 2023 at 10:15:16AM +0000, Lorenzo Stoakes wrote:
> Commit df04abf181a ("fs/proc/kcore.c: Add bounce buffer for ktext data")
> introduced the use of a bounce buffer to retrieve kernel text data for
> /proc/kcore in order to avoid failures arising from hardened user copies
> [...]

hi,
sorry for late feedback, but looks like this one breaks reading
/proc/kcore with objdump for me:

# cat /proc/kallsyms | grep ksys_read
fffffff8150ebc0 T ksys_read
# objdump -d --start-address=0xffffffff8150ebc0 --stop-address=0xffffffff8150ebd0 /proc/kcore

/proc/kcore: file format elf64-x86-64

objdump: Reading section load1 failed because: Bad address
On 31.05.23 13:58, Jiri Olsa wrote:
> On Thu, Mar 23, 2023 at 10:15:16AM +0000, Lorenzo Stoakes wrote:
>> Commit df04abfd181a ("fs/proc/kcore.c: Add bounce buffer for ktext data")
>> introduced the use of a bounce buffer to retrieve kernel text data for
>> /proc/kcore in order to avoid failures arising from hardened user copies
>> enabled by CONFIG_HARDENED_USERCOPY in check_kernel_text_object().
>> ...
>> Signed-off-by: Lorenzo Stoakes <lstoakes@gmail.com>
> Reviewed-by: David Hildenbrand <david@redhat.com>
>
> sorry for late feedback, but looks like this one breaks reading
> /proc/kcore with objdump for me:
> [...]

Thanks for the report. To be sure the issue doesn't fall through the cracks unnoticed, I'm adding it to regzbot, the Linux kernel regression tracking bot:

#regzbot ^introduced 2e1c017677
#regzbot title mm / fs/proc/kcore: reading /proc/kcore with objdump broke
#regzbot ignore activity
side note: Jiri could have added the report to the tracking himself by including:

#regzbot introduced 2e1c017077
2023-03-23 10:15  [PATCH v8 1/4] fs/proc/kcore: avoid bounce buffer for ktext data Lorenzo Stoakes
2023-05-31 11:58  Jiri Olsa
2023-07-21 13:48  Baoquan He [this message]
2023-07-21 14:13  Jiri Olsa
2023-07-24 06:23  David Hildenbrand
2023-07-24 08:08  Baoquan He
2023-07-24 08:18  Jiri Olsa
2023-07-24 14:33  Baoquan He
2023-07-31 19:21  Lorenzo Stoakes
2023-07-31 19:24  David Hildenbrand
2023-07-31 19:40  Lorenzo Stoakes
2023-07-31 20:34  Jiri Olsa
2023-07-31 21:12  Lorenzo Stoakes
2023-07-31 21:50  Jiri Olsa
2023-07-31 21:58  Lorenzo Stoakes
2023-07-24 09:38  Linux regression tracking (Thorsten Leemhuis)
From: Lorenzo Stoakes <lstoakes@gmail.com>
To: linux-mm@kvack.org, linux-kernel@vger.kernel.org,
Andrew Morton <akpm@linux-foundation.org>
Cc: Baoquan He <bhe@redhat.com>, Uladzislau Rezki <urezki@gmail.com>,
linux-fsdevel@vger.kernel.org, Jiri [..]
Subject: [PATCH] fs/proc/kcore: reinstate bounce buffer for KCORE_TEXT regions
Date: Mon, 31 Jul 2023 22:50:21 +0100  [thread overview]
Message-ID: <20230731215021.709111-1-lstoakes@gmail.com> (raw)

Some architectures do not populate the entire range categorised by
KCORE_TEXT, so we must ensure that the kernel address we read from is
valid.

Unfortunately there is no solution currently available to do so with a
purely iterator solution so reinstate the bounce buffer in this instance so
we can use copy_from_kernel_nofault() in order to avoid page faults when
regions are unmapped.

This change partly reverts commit 2e1c0176771e ("fs/proc/kcore: avoid
bounce buffer for ktext data"), reinstituting the bounce buffer, but adapts
the code to continue to use an iterator.

Fixes: 2e1c0176771e ("fs/proc/kcore: avoid bounce buffer for ktext data")
Reported-by: Jiri Olsa <olsajiri@gmail.com>
Closes: https://lore.kernel.org/all/2Hc2fm+9daF6cgCE@krave
Cc: stable@vger.kernel.org
Signed-off-by: Lorenzo Stoakes <lstoakes@gmail.com>

---
fs/proc/kcore.c | 1 +
Thread overview: 11+ messages / expand[flat|nested] mbox.gz Atom feed top

2023-07-31 21:50 Lorenzo Stoakes [this message]
2023-08-01  8:27 ` Will Deacon
2023-08-01  9:05 ` David Hildenbrand
2023-08-01 16:33 ` Lorenzo Stoakes
2023-08-01 16:34 ` David Hildenbrand
2023-08-01 16:39 ` Lorenzo Stoakes
2023-08-01 18:14 ` David Hildenbrand
2023-08-01 15:57 ` Baoquan He
2023-08-01 16:01 ` Baoquan He
2023-08-01 16:22 ` Lorenzo Stoakes
fs/proc/kcore: reinstate bounce buffer for KCORE_TEXT regions

Some architectures do not populate the entire range categorised by KCORE_TEXT, so we must ensure that the kernel address we read from is valid.

Unfortunately there is no solution currently available to do so with a purely iterator solution so reinstate the bounce buffer in this instance so we can use copy_from_kernel_nofault() in order to avoid page faults when regions are unmapped.

This change partly reverts commit 2e1c0170771e ("fs/proc/kcore: avoid bounce buffer for ktext data"), reinstating the bounce buffer, but adapts the code to continue to use an iterator.

[lstoakes@gmail.com: correct comment to be strictly correct about reasoning]

Link: https://lkml.kernel.org/r/525a3f14-74fa-4c22-9fca-9dab4de8a6c3@lucifer.local
Link: https://lkml.kernel.org/r/20230731215021.70911-1-lstoakes@gmail.com
Fixes: 2e1c0170771e ("fs/proc/kcore: avoid bounce buffer for ktext data")
Signed-off-by: Lorenzo Stoakes <lstoakes@gmail.com>
Reported-by: Jiri Olsa <olsajiri@gmail.com>
Closes: https://lore.kernel.org/all/ZHc2fm+9daF6cgCE@krava
Tested-by: Jiri Olsa <jolsa@kernel.org>
Tested-by: Will Deacon <will@kernel.org>
fs/proc/kcore: reinstate bounce buffer for KCORE_TEXT regions

Some architectures do not populate the entire range categorised by KCORE_TEXT, so we must ensure that the kernel address we read from is valid.

Unfortunately there is no solution currently available to do so with a purely iterator solution so reinstate the bounce buffer in this instance so we can use copy_from_kernel_nofault() in order to avoid page faults when regions are unmapped.

This change partly reverts commit 2e1c0170771e ("fs/proc/kcore: avoid bounce buffer for ktext data"), reinstating the bounce buffer, but adapts the code to continue to use an iterator.

[lstoakes@gmail.com: correct comment to be strictly correct about reasoning]
Link: https://lkml.kernel.org/r/525a3f14-74fa-4c22-9fca-9dab4de8a6c3@lucifer.local
Link: https://lkml.kernel.org/r/202307312115021.70911-1-lstoakes@gmail.com
Fixes: 2e1c0170771e ("fs/proc/kcore: avoid bounce buffer for ktext data")
Signed-off-by: Lorenzo Stoakes <lstoakes@gmail.com>
Reported-by: Jiri Olsa <olsajiri@gmail.com>
Closes: https://lore.kernel.org/all/ZHc2fm+9daF6cgCE@krava
Tested-by: Jiri Olsa <jolsa@kernel.org>
Tested-by: Will Deacon <will@kernel.org>
works somewhat similar with bugzilla.kernel.org links
[and arbitrary links as well, up to a point]
I keep an eye on tracked regressions
I keep an eye on tracked regressions and will show up with questions if things stall ;-}
Linux kernel regression status


current cycle (v6.6.. aka v6.6-post), culprit identified

- 5cfd69775eb5 "mainline build failure due to 5cfd69775eb5 ("bcaches: Array bounds fixes") by Sudip Mukherjee (Codethink)
  Earliest & latest activity: 6 days ago & 17 hours ago. Noteworthy: [fix incoming].
- 7966f319c66d "mainline build failure due to 7966f319c66d ("drm/amd/display: Introduce DML2") by Sudip Mukherjee (Codethink)
  Earliest & latest activity: 4 days ago & 53 hours ago.
- b6e3d1ba4fcf new iwltfmi firmware statistics API triggers stack trace by Niklāvs Kolesnikovs and Niklāvs Kolesnikovs
  Earliest & latest activity: 4 days ago & 3 days ago.
- 9c66dc94b62a "mainline build failure due to 9c66dc94b62a ("bpf: Introduce css_task open-coded iterator kfuncs") by Sudip Mukherjee (Codethink)
  Earliest & latest activity: 4 days ago & 4 days ago. Noteworthy: [fix incoming].

current cycle (v6.6.. aka v6.6-post), unknown culprit

none known by regbot

previous cycle (v6.5..v6.6), culprit identified, with activity in the past three months

- 6fadda69f62 "stable offsets directory operation support triggers offset_ctx->xa memory leak by vladbu and vladbu
  Earliest & latest activity: 26 days ago & 3 days ago. Noteworthy: [1].
- 088a40980efb "Regression apparently caused by commit 088a40980efb2c449b72f0f2c7ebd82f71d082e "ASoC: amd: acp: add pm ops support for acp pci driver" by Marian Posteva
  Earliest & latest activity: 14 weeks ago & 3 days ago.
- 9d1bd9346241 "Logitech G915 Wireless Keyboard key event only detects few key codes by LinuxCat and LinuxCat
  Earliest & latest activity: 5 days ago & 4 days ago. Noteworthy: [1].
- 276e14e6c3 "HID: input: stylus of Xp-Pen Artist 24 Pro display tablet changed behavior by jiri Kosina, Unknown, and Unknown
  Earliest & latest activity: 5 days ago & 62 hours ago.
- c7d80059b086 "sysfs: cannot create duplicate filename.../system76_acpi:kbd_backlight/color by Johannes Penßel and Johannes Penßel
  Earliest & latest activity: 12 days ago & 51 days ago.
- 024218477809 "quota: boot on Intel Merrifield after merge commit 15000e7e072e6 by Andy Shevchenko
  Earliest & latest activity: 20 days ago & 14 days ago. Noteworthy: [patch].
Linux kernel regression status

[next] [mainline] [stable/longterm] • [new] • [all] • [resolved] [inconclusive]

current cycle (v6.6.. aka v6.6-post), culprit identified

- 5cfd6977eb5

  ▼ mainline build failure due to 5cfd6977eb5 ("bcachefs: Array bounds fixes") by Sudip Mukherjee (Codethink)
  Earliest & latest activity: 6 days ago & 17 hours ago. Noteworthy: [fix incoming].
  Fix incoming: 6dfa10ab22a6 ("bcachefs: Fix build errors with gcc 10")
  17 hours ago

  Latest five known activities:
  - 6dfa10ab22a6, the fix specified through '#regzbot fix: earlier landed in next
    17 hours ago, by Kent Overstreet
  - 0591f9e25f0b, the fix specified through '#regzbot fix: earlier landed in next
    4 days ago, by Kent Overstreet
  - ad77f25f730e, the fix specified through '#regzbot fix: earlier landed in next
    5 days ago, by Kent Overstreet
  - Re: mainline build failure due to 5cfd6977eb5 ("bcachefs: Array bounds fixes")
    5 days ago, by Linux regression tracking (Thorsten Leemhuis)
  - Re: mainline build failure due to 5cfd6977eb5 ("bcachefs: Array bounds fixes")
    5 days ago, by Sudip Mukherjee

Regzbot command history:
  - note: 'fix' commit '6dfa10ab22a6' now in 'next'
    17 hours ago, by Kent Overstreet
  - note: 'fix' commit '0591f9e25f0b' now in 'next'
    4 days ago, by Kent Overstreet
  - note: 'fix' commit 'ad77f25f730e' now in 'next'
    5 days ago, by Kent Overstreet
  - fix: bcachefs: Fix build errors with gcc 10
    5 days ago, by Linux regression tracking (Thorsten Leemhuis)
  - fix: bcachefs: Fix build errors with gcc 10
    5 days ago, by Linux regression tracking (Thorsten Leemhuis)
  - introduced: 5cfd6977eb5460ef78bb5034a37eb0dc52ab65d
    6 days ago, by Sudip Mukherjee (Codethink)
Hi Linus. Just three 6.6 regression remain on my list after a few I tracked were resolved last week. One of the remaining ones is new: module loading trouble on some laptops. Not nice, but likely nothing many users will encounter. The quota compilation oddity problem from Andy is also still around (unless it was fixed without me noticing); and a memleak, too. See below for details.

FWIW, there was some news wrt to the two 6.5 regressions I mentioned in last weeks report[1]:

* There was another report about a blank screen during boot on a Lenovo
On Sun, 29 Oct 2023 at 03:52, Regzbot (on behalf of Thorsten Leemhuis) writes:

> One of the remaining ones is new:
> module loading trouble on some laptops. Not nice, but likely nothing
> many users will encounter. The quota compilation oddity problem from
> Andy is also still around (unless it was fixed without me noticing); and
> a memleak, too.
Developer can interact with regzbot via regzbot commands in a reply to the report, for example when a Link:/Closes: tag was forgotten
Developer can interact with regzbot but do not have to or care about regzbot at all.
They don't have to care about my regression tracking work either...
...unless of course it looks like a regression is not handled appropriately
in the ideal case adding the report to the tracking is thus the only extra-work required
Regzbot itself is pretty basic and sometimes a bit rough, but does what it's designed for
Important features on the road map (priority):

- support for issues submitted to github and gitlab projects (WIP)

- separate actionable vs non-actionable reports in the UI (actionable: a sane report with a bisection result)

- make it more obvious in the UI when a fix is out for review

- make it dead simple to add regressions to the tracking where a mail contains both report and a fix
Important features on the road map (later):

- various UI fine tuning (command line interface, website)
- mark some regressions as "priority"
- tagging to identify subsystems or sources of reports (like CIs)
- subsystem specific webpages & reports
- export data better to make it more useful for subsystem and stable maintainers
Ideas for more features:

- check pull requests to Linus and yell if they are known to cause regressions?
That's the state of things right now!
...which brings us to...
What do you want regzbot or me to do?
some people want to add regressions to regzbot that CI systems found
I see that it would be useful and definitely want that to happen in the long term.
I see that it would be useful and definitely want that to happen in the long term.

At the same time I need to be able to stay on top of tracked regressions.
this hence definitely needs separation between actionable vs non-actionable reports
this hence definitely needs separation between actionable vs non-actionable reports

and maybe some CI reports should only become "actionable" after some human performed a sanity check
Other open issues
Integration of regression status in CI systems:
- Modelling of regression life cycle
- Close the loop between regression detection and developers:
  - CI systems provide data to users (unidirectional)
  - Status changes generated by users could be communicated back to CI systems
Regzbot provides these features
- Integrating them into the source CI data can lead to more useful and up-to-date reports