☆ Starred by 2 users

Owner:	mythria@chromium.org **Last visit > 30 days ago**
CC:	mvsta...@chromium.org tebbi@chromium.org neis@chromium.org rmcilroy@chromium.org ishell@chromium.org mstarzinger@chromium.org vahl@chromium.org ecmziegler@google.com
Status:	Fixed *(Closed)*
Components:	Blink>JavaScript
Modified:	Apr 28, 2020
Backlog-Rank:	----
Editors:	----
EstimatedDays:	----
NextAction:	----
OS:	Linux, Android, Windows, Chrome, Mac, Fuchsia
Pri:	1
Type:	Bug-Security

reward-2000
Security_Impact-Stable
Security_Severity-Medium
allpublic
reward-inprocess
CVE_description-submitted
Release-0-M81
CVE-2020-6430

Issue 1031479: Security: Debug check failed: has_feedback_vector()

Reported by b3nd3...@gmail.com on Fri, Dec 6, 2019, 6:09 AM EST

Code

Target : ASAN-D8-DBG Latest
Crash Type: Debug check
Crash State: Debug check failed: has_feedback_vector().

```
#
# Fatal error in ../../src/objects/js-objects-inl.h, line 460
# Debug check failed: has_feedback_vector().
#
#
#
#FailureMessage Object: 0x7ffefaea1c50
```

POC:

```
--------------------------
function main() {
function v0(v1,v2,v3,v4) {
    const v6 = [1337,1337,1337];
    const v8 = [-3458580188,-3458580188,-3458580188,v6];
    const v9 = [];
    function v10(v11,v12,v13,v14) {
        const v16 = ["c19rXGEC2E"];
        try {
            v16.e = v9;
            const v17 = v8.__proto__;
            const v19 = {set:v10};
            const v21 = Object.defineProperty(v17,"e",v19);
        } catch(v22) {
            for (const v24 in "c19rXGEC2E") {
            }
        }
    }
    const v25 = v10();
}
const v26 = v0();
for (let v30 = 0; v30 < 9; v30++) {
    const v33 = new ArrayBuffer(1073741824);
}
const v34 = v0();
}
main();

--------------------------
```

* flags to reproduce -   "--interrupt-budget=1024"

*** This sample was found through context aware fuzzing .

Comment 1 by ClusterFuzz on Fri, Dec 6, 2019, 11:10 AM EST Project Member

ClusterFuzz is analyzing your testcase. Developers can follow the progress at https://clusterfuzz.com/testcase?key=5329484961939456.

Comment 2 by ClusterFuzz on Fri, Dec 6, 2019, 11:34 AM EST Project Member

Testcase 5329484961939456 failed to reproduce the crash. Please inspect the program output at https://clusterfuzz.com/testcase?key=5329484961939456.

Comment 3 by metzman@chromium.org on Fri, Dec 6, 2019, 1:55 PM EST Project Member

Status: Assigned (was: Unconfirmed)
Owner: ishell@chromium.org
Cc: mstarzinger@chromium.org
Labels: Security_Needs_Attention-Severity Security_Severity-Low OS-Android OS-Chrome OS-Fuchsia OS-Linux OS-Mac OS-Windows

Thanks for this report!
I was able to repro locally.
ishell@ could you PTAL?

Comment 4 by metzman@chromium.org on Fri, Dec 6, 2019, 1:56 PM EST Project Member

Components: Blink>JavaScript

Comment 5 by sheriffbot@chromium.org on Sat, Dec 7, 2019, 10:31 AM EST Project Member

Labels: Pri-2

Setting Pri-2 to match security severity Low. If this is incorrect, please reset the priority. Sheriffbot won't make this change again.

For more details visit https://www.chromium.org/issue-tracking/autotriage - Your friendly Sheriffbot

Comment 6 by ishell@chromium.org on Mon, Dec 9, 2019, 5:07 AM EST Project Member

Owner: mythria@chromium.org
Cc: ishell@chromium.org neis@chromium.org

Seems to be related to lazy feedback allocation. Mythri, PTAL.

Comment 7 by mythria@chromium.org on Wed, Dec 11, 2019, 5:24 AM EST Project Member

Status: Started (was: Assigned)

I started looking into this. I am not yet sure why we don't find feedback vector when trying to OSR. This code causes a StackOverflow (because of infinite recursion) and throws an error. The code catches this error and continues execution. The catch block has a for loop which actually triggers an OSR. This still doesn't explain why there is no feedback vector.

Comment 8 by mythria@chromium.org on Wed, Dec 11, 2019, 12:28 PM EST Project Member

This is happening because we are marking one closure for OSRing and optimizing a different closure. Typically all closures should share the same feedback vector since they share the same feedback cell. Though, if the bytecode gets flushed we also reset the feedback cells and hence the closures created before the flushing of bytecode and after flushing don't share the same feedback cell. In this particular example, we mark the closure created before bytecode flushing for optimization but actually optimize the one that is created after flushing which doesn't have feedback vector.

Here's the slightly simplified code that causes this problem:

```
var i = 0;
function main() {
function v0() {
  function v10(a) {
      i++;
      var cur_i = i;
      try {
          // This triggers the use of old closure that was installed in the
          // earlier invocation of v10 and causes an infinite recursion. At
          // some point we throw from here.
          [].e = 1;

          // Throw when the new closure is on the stack to trigger a
          // OSR on the new closure that doesn't have a feedback vector.
          if (cur_i == 2) throw 1;
      } catch(v22) {
          // This loop triggers OSR.
          for (const v24 in "c19rXGEC2E") {
          }
      }
  }
  const v25 = v10(1);
  // We install v10's closure here. The bytecode for v10 gets flushed when we
  // allocate large ArrayBuffers.
  const v21 = Object.defineProperty([].__proto__,"e",{set:v10});
}
const v26 = v0();
// Causes multiple GCs which flushes the bytecode for both v0 and v10. This
// resets the ClosureFeedbackCellArray on v0. Hence the v10 closures created
// by v0 after this point doesn't share the same feedback cell.
for (let v30 = 0; v30 < 9; v30++) {
   const v33 = new ArrayBuffer(1073741824);
}
const v34 = v0();
}
main();
```

There are multiple options here:
1. The quick and easy fix for this is to check if we have feedback vector and abort optimization if there is no feedback vector. Though I think that is not the right fix.

2. I think the real fix should be that we should only OSR for the closures that are marked for optimization and not others. The reason this happens currently is because the osr nesting level[1] that triggers OSR is on the bytecode which is shared across the closures. I think we should really move osr nesting level to feedback vector to avoid such issues. Though that is not entirely trivial especially since it may have performance implications. The JumpLoop bytecode handler uses this information. So, the bytecode handler has to do few extra loads to get this information from the feedback vector.

3. The other option is to not reset the ClosureFeedbackCellArray on bytecode flush. Currently, we just reset the raw feedback cell to Undefined. We could instead reset it to the ClosureFeedbackCellArray. That still resets the feedback vector so may be memory regression may not be too high. This would fix this particular case. I am not entirely sure if it would be still possible to have JSClosures with different feedback cells which share the same SFI.

I think I will do the quick fix (option 1) first to mitigate this problem. Though I think long term solution should be either 2 or 3. Any other ideas?

[1] https://source.chromium.org/chromium/chromium/src/+/master:out/chromeos-Debug/gen/v8/torque-generated/field-offsets-tq.h;l=215?q=kOsrNestingLevelOffset&ss=chromium%2Fchromium%2Fsrc&originalUrl=https:%2F%2Fcs.chromium.org%2F

Comment 9 by mythria@chromium.org on Wed, Dec 11, 2019, 12:29 PM EST Project Member

Forgot to mention, we need these flags to reproduce the problem on the code in comment#8: --interrupt-budget=10 --stack-size=50 --budget_for_feedback_vector_allocation=10

Comment 10 by mythria@chromium.org on Wed, Dec 11, 2019, 12:30 PM EST Project Member

Cc: rmcilroy@chromium.org mvsta...@chromium.org

Comment 11 by neis@chromium.org on Thu, Dec 12, 2019, 5:06 AM EST Project Member

Cc: tebbi@chromium.org

Comment 12 by bugdroid on Thu, Dec 12, 2019, 10:42 AM EST Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/v8/v8.git/+/83fd3e84ac43c6dcad47df3075215b31c1aada49

commit 83fd3e84ac43c6dcad47df3075215b31c1aada49
Author: Mythri A <mythria@chromium.org>
Date: Thu Dec 12 15:42:16 2019

Check if a function has feedback vector before OSRing.

With bytecode flushing and the current OSR triggering mechanism which
stores OSR nesting level on bytecode array it is possible to trigger
OSR on a closure that doesn't have feedback vector.

~~Bug: chromium:1031479~~
Change-Id: I4c62486f6b0eb6d6f9c96f98c1c1b275f3e6d6d5
Reviewed-on: https://chromium-review.googlesource.com/c/v8/v8/+/1962850
Commit-Queue: Mythri Alle <mythria@chromium.org>
Reviewed-by: Michael Stanton <mvstanton@chromium.org>
Reviewed-by: Ross McIlroy <rmcilroy@chromium.org>
Cr-Commit-Position: refs/heads/master@{#65431}

[modify] https://crrev.com/83fd3e84ac43c6dcad47df3075215b31c1aada49/src/runtime/runtime-compiler.cc
[add] https://crrev.com/83fd3e84ac43c6dcad47df3075215b31c1aada49/test/mjsunit/regress/regress-crbug-1031479.js

Comment 13 by mythria@chromium.org on Tue, Dec 17, 2019, 10:50 AM EST Project Member

I had an offline chat with Ross and the summary of the discussion is we will implement option 3 (reset the feedback cell to closure feedback cell array). I think it will be really rare (even if possible) that there would be a recursion involving closures from two different native contexts. It will be still nice to move osr triggering mechanism (option 2) to feedback vector as well. I will create a tracking bug for that. We might at some point add feedback for JumpLoops as well and then moving OSR feedback level to feedback vector would be easier.

I will work on a cl that fixes bytecode flushing this week.

Comment 14 by mvsta...@chromium.org on Tue, Dec 17, 2019, 12:23 PM EST Project Member

Sounds good, thanks Mythri!

Comment 15 by bugdroid on Mon, Jan 20, 2020, 11:13 AM EST Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/v8/v8.git/+/92df7d10f634855bb0422c51e49091161497f645

commit 92df7d10f634855bb0422c51e49091161497f645
Author: Mythri A <mythria@chromium.org>
Date: Mon Jan 20 16:12:42 2020

Only flush feedback vector on bytecode flush

When bytecode is flushed we also want to flush the feedback vectors to
save memory. There was a bug in this code and we flushed
ClosureFeedbackCellArray too. Flushing ClosureFeedbackCellArrays causes
the closures created by this function before and after the bytecode
flush to have different feedback cells and hence different feedback
vectors. This cl fixes it so we only flush feedback vectors on a
bytecode flush.

Also this cl pretenures ClosureFeedbackCellArrays. Only FeedbackCells
and FeedbackVectors can contain ClosureFeedbackCellArrays which are
pretenured, so it is better to pretenure ClosureFeedbackCellArrays as
well.

~~Bug: chromium:1031479~~
Change-Id: I7831441a95420b9e5711f4143461f1eb7fa1616a
Reviewed-on: https://chromium-review.googlesource.com/c/v8/v8/+/1980582
Commit-Queue: Mythri Alle <mythria@chromium.org>
Reviewed-by: Ross McIlroy <rmcilroy@chromium.org>
Reviewed-by: Ulan Degenbaev <ulan@chromium.org>
Reviewed-by: Michael Stanton <mvstanton@chromium.org>
Cr-Commit-Position: refs/heads/master@{#65866}

[modify] https://crrev.com/92df7d10f634855bb0422c51e49091161497f645/src/heap/factory.cc
[modify] https://crrev.com/92df7d10f634855bb0422c51e49091161497f645/src/heap/mark-compact.cc
[modify] https://crrev.com/92df7d10f634855bb0422c51e49091161497f645/src/objects/feedback-cell-inl.h
[modify] https://crrev.com/92df7d10f634855bb0422c51e49091161497f645/src/objects/feedback-cell.h
[modify] https://crrev.com/92df7d10f634855bb0422c51e49091161497f645/src/objects/js-objects-inl.h
[modify] https://crrev.com/92df7d10f634855bb0422c51e49091161497f645/src/objects/js-objects.h
[modify] https://crrev.com/92df7d10f634855bb0422c51e49091161497f645/src/runtime/runtime-compiler.cc

Comment 16 by mythria@chromium.org on Tue, Jan 21, 2020, 9:54 AM EST Project Member

Status: Fixed (was: Started)

I think this is all we wanted to do here. Moving OSR triggering mechanism to feedback vector is out of scope and not immediately needed. Closing this for now. Feel free to reopen if needed.

Comment 17 by sheriffbot@chromium.org on Tue, Jan 21, 2020, 10:42 AM EST Project Member

Labels: -Restrict-View-SecurityTeam Restrict-View-SecurityNotify

Comment 18 by natashapabrai@google.com on Tue, Jan 21, 2020, 11:56 AM EST Project Member

Labels: reward-topanel

Comment 19 by natashapabrai@google.com on Wed, Jan 29, 2020, 7:07 PM EST Project Member

Labels: -reward-topanel reward-0

mythria@chromium.org - can you provide the Panel with more information re: the exploitability of this bug

Comment 20 by natashapabrai@google.com on Wed, Jan 29, 2020, 7:07 PM EST Project Member

Labels: -reward-0 reward-topanel

Comment 21 by mythria@chromium.org on Fri, Jan 31, 2020, 9:36 AM EST Project Member

This bug is related to type confusion. We have a closureFeedbackCellArray (which is kind of array of pointers) but we interpret it as a FeedbackVector which has a different layout and different size. This means we could potentially do OOB reads. We would need a recursion with a two different closures of the same function which doesn't happen often but possible to construct as shown in this test case.

Comment 22 by adetaylor@google.com on Wed, Feb 5, 2020, 6:39 PM EST Project Member

Labels: -Security_Severity-Low -Security_Needs_Attention-Severity Security_Severity-Medium Security_Impact-Stable

Assuming this impacts stable. Bumping to medium per #c21.

Comment 23 by natashapabrai@google.com on Wed, Feb 5, 2020, 6:59 PM EST Project Member

Labels: -reward-topanel reward-unpaid reward-2000

*** Boilerplate reminders! ***
Please do NOT publicly disclose details until a fix has been released to all our users. Early public disclosure may cancel the provisional reward. Also, please be considerate about disclosure when the bug affects a core library that may be used by other products. Please do NOT share this information with third parties who are not directly involved in fixing the bug. Doing so may cancel the provisional reward. Please be honest if you have already disclosed anything publicly or to third parties. Lastly, we understand that some of you are not interested in money. We offer the option to donate your reward to an eligible charity. If you prefer this option, let us know and we will also match your donation - subject to our discretion. Any rewards that are unclaimed after 12 months will be donated to a charity of our choosing.

Please contact security-vrp@chromium.org with any questions.
****

Comment 24 by pabrai@chromium.org on Wed, Feb 5, 2020, 7:06 PM EST Project Member

Congrats the Panel decided to award $2,000 for this report!

Comment 25 by natashapabrai@google.com on Wed, Feb 5, 2020, 7:13 PM EST Project Member

Labels: -reward-unpaid reward-inprocess

Comment 26 by sheriffbot@chromium.org on Thu, Feb 6, 2020, 11:58 AM EST Project Member

Labels: -Pri-2 Pri-1

Setting Pri-1 to match security severity Medium. If this is incorrect, please reset the priority. Sheriffbot won't make this change again.

For more details visit https://www.chromium.org/issue-tracking/autotriage - Your friendly Sheriffbot

Comment 27 by adetaylor@google.com on Fri, Mar 13, 2020, 1:44 PM EDT Project Member

Labels: Release-0-M81

Comment 28 by adetaylor@chromium.org on Fri, Mar 13, 2020, 2:30 PM EDT Project Member

Labels: CVE-2020-6430 CVE_description-missing

Comment 29 by adetaylor@chromium.org on Tue, Apr 14, 2020, 3:14 PM EDT Project Member

Labels: -CVE_description-missing CVE_description-submitted

Comment 30 by sheriffbot on Tue, Apr 28, 2020, 2:54 PM EDT Project Member

Labels: -Restrict-View-SecurityNotify allpublic

This bug has been closed for more than 14 weeks. Removing security view restrictions.

For more details visit https://www.chromium.org/issue-tracking/autotriage - Your friendly Sheriffbot