☆ Starred by 3 users

Owner:	leszeks@chromium.org
CC:	jkummerow@chromium.org mslekova@chromium.org clemensb@chromium.org vahl@chromium.org ecmziegler@google.com
Status:	Fixed *(Closed)*
Components:	Blink>JavaScript
Modified:	Apr 18, 2020
Backlog-Rank:	----
Editors:	----
EstimatedDays:	----
NextAction:	----
OS:	Linux, Android, Windows, Chrome, Mac, Fuchsia
Pri:	2
Type:	Bug-Security

reward-0
Security_Severity-Low
Security_Impact-Stable
allpublic
CVE_description-submitted
Target-79
M-79
Release-0-M81
CVE-2020-6448

Issue 1037872: Security:Potential Use after free in the function PerfJitLogger::LogWriteDebugInfo

Reported by higon...@gmail.com on Thu, Dec 26, 2019, 5:18 AM EST

Code

the function PerfJitLogger::LogWriteDebugInfo https://cs.chromium.org/chromium/src/v8/src/diagnostics/perf-jit.cc?rcl=53366c4eab3aa7def56310d7cbafc23c33179bc6&l=331
uses the same raw point code between Garbage Collect, witch may cause UAF(use after GC move the object)

```
void PerfJitLogger::LogWriteDebugInfo(Code code, SharedFunctionInfo shared) {   ----------------------->1 code is a raw pointer without handlify
  // Compute the entry count and get the name of the script.
  uint32_t entry_count = 0;
  for (SourcePositionTableIterator iterator(code.SourcePositionTable());
       !iterator.done(); iterator.Advance()) {
    entry_count++;
  }
  if (entry_count == 0) return;
  // The WasmToJS wrapper stubs have source position entries.
  if (!shared.HasSourceCode()) return;
  Isolate* isolate = shared.GetIsolate();
  Handle<Script> script(Script::cast(shared.script()), isolate);

  PerfJitCodeDebugInfo debug_info;

  debug_info.event_ = PerfJitCodeLoad::kDebugInfo;
  debug_info.time_stamp_ = GetTimestamp();
  debug_info.address_ = code.InstructionStart();
  debug_info.entry_count_ = entry_count;

  uint32_t size = sizeof(debug_info);
  // Add the sizes of fixed parts of entries.
  size += entry_count * sizeof(PerfJitDebugEntry);
  // Add the size of the name after each entry.

  Handle<Code> code_handle(code, isolate);
  Handle<SharedFunctionInfo> function_handle(shared, isolate);
  for (SourcePositionTableIterator iterator(code.SourcePositionTable());
       !iterator.done(); iterator.Advance()) {
    SourcePositionInfo info(GetSourcePositionInfo(code_handle, function_handle,    ----------------------->2 GetSourcePositionInfo may cause a GC through path
"GetSourcePositionInfo->InliningStack->SourcePositionInfo->GetPositionInfo->InitLineEnds->CalculateLineEnds->NewFixedArray"
                                                    iterator.source_position()));
    size += GetScriptNameLength(info) + 1;
  }

  int padding = ((size + 7) & (~7)) - size;
  debug_info.size_ = size + padding;
  LogWriteBytes(reinterpret_cast<const char*>(&debug_info), sizeof(debug_info));

  Address code_start = code.InstructionStart();        --------------------------->3 raw pointer is used after heap allocation
  for (SourcePositionTableIterator iterator(code.SourcePositionTable());
       !iterator.done(); iterator.Advance()) {
    SourcePositionInfo info(GetSourcePositionInfo(code_handle, function_handle,
                                                  iterator.source_position()));
    PerfJitDebugEntry entry;
    // The entry point of the function will be placed straight after the ELF
    // header when processed by "perf inject". Adjust the position addresses
    // accordingly.
    entry.address_ = code_start + iterator.code_offset() + kElfHeaderSize;
    entry.line_number_ = info.line + 1;
    entry.column_ = info.column + 1;
    LogWriteBytes(reinterpret_cast<const char*>(&entry), sizeof(entry));
    // The extracted name may point into heap-objects, thus disallow GC.
    DisallowHeapAllocation no_gc;
    std::unique_ptr<char[]> name_storage;
    Vector<const char> name_string = GetScriptName(info, &name_storage, no_gc);
    LogWriteBytes(name_string.begin(),
                  static_cast<uint32_t>(name_string.size()) + 1);
  }
  char padding_bytes[8] = {0};
  LogWriteBytes(padding_bytes, padding);
}
```

Acturaly, this issue is similar to https://bugs.chromium.org/p/chromium/issues/detail?id=1033407

Comment 1 by kenrb@chromium.org on Thu, Dec 26, 2019, 10:50 AM EST Project Member

Components: Blink>JavaScript

Comment 2 by adetaylor@google.com on Thu, Dec 26, 2019, 11:50 AM EST Project Member

Status: Untriaged (was: Unconfirmed)
Owner: leszeks@chromium.org
Cc: jkummerow@chromium.org
Labels: Security_Impact-Stable Security_Severity-High OS-Android OS-Chrome OS-Fuchsia OS-Linux OS-Mac OS-Windows

Looks believable to me, and git blame doesn't show major recent changes to this function, so I'm assuming this impacts stable. UaF in renderer => high severity for potential renderer RCE. Assuming it affects all V8 platforms.

Comment 3 by sheriffbot@chromium.org on Fri, Dec 27, 2019, 9:25 AM EST Project Member

Labels: Target-79 M-79

Setting milestone and target because of Security_Impact=Stable and high severity.

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

Comment 4 by sheriffbot@chromium.org on Fri, Dec 27, 2019, 10:05 AM EST Project Member

Labels: Pri-1

Setting Pri-1 to match security severity High. 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 5 by sheriffbot@chromium.org on Fri, Dec 27, 2019, 11:11 AM EST Project Member

Status: Assigned (was: Untriaged)

Comment 6 by leszeks@chromium.org on Tue, Jan 7, 2020, 2:29 AM EST Project Member

Cc: mslekova@chromium.org

Nice catch, should be easy enough to fix.

+mslekova, how come GCMole missed this?

Comment 7 by clemensb@chromium.org on Tue, Jan 7, 2020, 5:49 AM EST Project Member

Labels: -Security_Severity-High Security_Severity-Low Pri-2

PerfJitLogger will only be used if --perf-prof is given on the command line. Hence severity is low.

Comment 8 by mslekova@chromium.org on Tue, Jan 7, 2020, 7:07 AM EST Project Member

@leszeks because dead variable analysis is turned off by default, as noted in https://bugs.chromium.org/p/v8/issues/detail?id=9680#c2.
Unfortunately most of the "false positives" found by the analysis could as well be real failures, that's why these bugs need to be fixed/closed as WAI first:
https://bugs.chromium.org/p/v8/issues/list?q=possibly%20dead%20variables%20gcmole&can=2

Comment 9 by bugdroid on Tue, Jan 7, 2020, 8:45 AM EST Project Member

The following revision refers to this bug:
  https://chromium.googlesource.com/v8/v8.git/+/c2c58856789733d1bbf1ee02f9a129baef44a8d8

commit c2c58856789733d1bbf1ee02f9a129baef44a8d8
Author: Leszek Swirski <leszeks@chromium.org>
Date: Tue Jan 07 13:44:45 2020

[log] Use handles for LogRecordedBuffer

LogWriteDebugInfo can allocate when calculating line ends for source
positions, so make its called, LogRecordedBuffer, take Handles rather
than raw Objects. This also improves its API, as we can change the
maybe-null SharedFunctionInfo argument into a MaybeHandle.

~~Bug: chromium:1037872~~
Change-Id: Ifa3e2d9be7aa7de3b05e5c1e107406004b8963c7
Reviewed-on: https://chromium-review.googlesource.com/c/v8/v8/+/1985995
Commit-Queue: Leszek Swirski <leszeks@chromium.org>
Commit-Queue: Toon Verwaest <verwaest@chromium.org>
Reviewed-by: Toon Verwaest <verwaest@chromium.org>
Reviewed-by: Clemens Backes <clemensb@chromium.org>
Auto-Submit: Leszek Swirski <leszeks@chromium.org>
Cr-Commit-Position: refs/heads/master@{#65603}

[modify] https://crrev.com/c2c58856789733d1bbf1ee02f9a129baef44a8d8/src/diagnostics/perf-jit.cc
[modify] https://crrev.com/c2c58856789733d1bbf1ee02f9a129baef44a8d8/src/diagnostics/perf-jit.h
[modify] https://crrev.com/c2c58856789733d1bbf1ee02f9a129baef44a8d8/src/logging/log.cc
[modify] https://crrev.com/c2c58856789733d1bbf1ee02f9a129baef44a8d8/src/logging/log.h

[modify] https://crrev.com/c2c58856789733d1bbf1ee02f9a129baef44a8d8/src/snapshot/serializer.h
[modify] https://crrev.com/c2c58856789733d1bbf1ee02f9a129baef44a8d8/test/cctest/test-log.cc

Comment 10 by clemensb@chromium.org on Thu, Jan 9, 2020, 7:22 AM EST Project Member

Note: This also fixes a flake we had on our tree for a few days (~~https://crbug.com/v8/10007~~).

Comment 11 by leszeks@chromium.org on Thu, Jan 9, 2020, 7:23 AM EST Project Member

Status: Fixed (was: Assigned)

Nice.

Comment 12 by sheriffbot@chromium.org on Thu, Jan 9, 2020, 10:43 AM EST Project Member

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

Comment 13 by natashapabrai@google.com on Tue, Jan 14, 2020, 11:57 AM EST Project Member

Labels: reward-topanel

Comment 14 by natashapabrai@google.com on Wed, Jan 29, 2020, 7:09 PM EST Project Member

Labels: -reward-topanel reward-0

Unfortunately the Panel declined to reward this report

Comment 15 by adetaylor@google.com on Thu, Jan 30, 2020, 6:44 PM EST Project Member

Cc: clemensb@chromium.org

~~Issue 1035225~~ has been merged into this issue.

Comment 16 by higon...@gmail.com on Mon, Mar 2, 2020, 5:22 AM EST

can you assign a cve to this issue?

Comment 17 by adetaylor@google.com on Fri, Mar 13, 2020, 1:38 PM EDT Project Member

There will be a CVE when it's mentioned in the release notes, which should be soon!

Comment 18 by adetaylor@google.com on Fri, Mar 13, 2020, 1:44 PM EDT Project Member

Labels: Release-0-M81

Comment 19 by adetaylor@chromium.org on Fri, Mar 13, 2020, 2:32 PM EDT Project Member

Labels: CVE-2020-6448 CVE_description-missing

Comment 20 by adetaylor@chromium.org on Tue, Apr 14, 2020, 3:14 PM EDT Project Member

Labels: -CVE_description-missing CVE_description-submitted

Comment 21 by sheriffbot on Sat, Apr 18, 2020, 2:57 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

About Monorail User Guide Release Notes Feedback on Monorail Terms Privacy