Starred by 8 users

Owner:	marja@chromium.org
CC:	adetaylor@chromium.org mvsta...@chromium.org jkummerow@chromium.org mythria@chromium.org pbomm...@chromium.org verwa...@chromium.org ishell@chromium.org vahl@chromium.org leszeks@chromium.org ecmziegler@google.com
Status:	Verified *(Closed)*
Components:	Blink>JavaScript>Runtime Blink>JavaScript>Compiler
Modified:	Sep 2, 2021
Backlog-Rank:	----
Editors:	----
EstimatedDays:	----
NextAction:	----
OS:	Linux, Android, Windows, Chrome, Fuchsia, Lacros
Pri:	1
Type:	Bug-Security

Hotlist-Merge-Review
Security_Impact-Stable
Security_Severity-High
allpublic
reward-inprocess
ClusterFuzz-Verified
reward-20000
CVE_description-submitted
M-90
Target-90
LTS-Security-86
LTS-Security-NotApplicable-86
external_security_report
merge-merged-90
merge-merged-9.0
merge-merged-9.1
merge-merged-91
Release-3-M90

Issue 1203122: Security: Type confusion bug in LoadSuperIC

Reported by laura...@gmail.com on Mon, Apr 26, 2021, 11:22 PM EDT

Code

VULNERABILITY DETAILS

Type confusion in the IC system of v8. The incorrect code path is:
  [0] AccessorAssembler::LoadSuperIC
  [1] AccessorAssembler::HandleLoadICHandlerCase

```
void AccessorAssembler::LoadSuperIC(const LoadICParameters* p) {
  ExitPoint direct_exit(this);

  TVARIABLE(MaybeObject, var_handler);
  Label if_handler(this, &var_handler), no_feedback(this),
      non_inlined(this, Label::kDeferred), try_polymorphic(this),
      miss(this, Label::kDeferred);

  GotoIf(IsUndefined(p->vector()), &no_feedback); <------- [0]

  // The lookup start object cannot be a SMI, since it's the home object's
  // prototype, and it's not possible to set SMIs as prototypes.
  TNode<Map> lookup_start_object_map =
      LoadReceiverMap(p->lookup_start_object());
  GotoIf(IsDeprecatedMap(lookup_start_object_map), &miss);

  TNode<MaybeObject> feedback = <------- [1]
      TryMonomorphicCase(p->slot(), CAST(p->vector()), lookup_start_object_map,
                         &if_handler, &var_handler, &try_polymorphic);

  BIND(&if_handler); <------- [2]
  {
    LazyLoadICParameters lazy_p(p);
    HandleLoadICHandlerCase(&lazy_p, CAST(var_handler.value()), &miss,
                            &direct_exit);
  }

  BIND(&no_feedback); <------- [3]
  { LoadSuperIC_NoFeedback(p); }

  BIND(&try_polymorphic); <------- [4]
  TNode<HeapObject> strong_feedback = GetHeapObjectIfStrong(feedback, &miss);
  {
    Comment("LoadSuperIC_try_polymorphic");
    GotoIfNot(IsWeakFixedArrayMap(LoadMap(strong_feedback)), &non_inlined);
    HandlePolymorphicCase(lookup_start_object_map, CAST(strong_feedback),
```

```
                    &if_handler, &var_handler, &miss);
  }

  BIND(&non_inlined); <------- [5]
  {
    // LoadIC_Noninlined can be used here, since it handles the
    // lookup_start_object != receiver case gracefully.
    LoadIC_Noninlined(p, lookup_start_object_map, strong_feedback, &var_handler,
                      &if_handler, &miss, &direct_exit);
  }

  BIND(&miss); <------- [6]
  direct_exit.ReturnCallRuntime(Runtime::kLoadWithReceiverIC_Miss, p->context(),
                                p->receiver(), p->lookup_start_object(),
                                p->name(), p->slot(), p->vector());
}
```

The LoadSuperIC is called for a number of times. At the beginning, because there's
no feedback installed. The method will end up with calling LoadSuperIC_NoFeedBack.
This is [0] -> [3]. This happens for the first few calls.

Then at some point the feedback is installed, the code path becomes [1]-[4]-[5].
The comment at [5] is very interesting, saying that LoadIC_Noninlined handles the
lookup_start_object != receiver case. This is VERY important! You can't handle
this case carefully enough.

At first, the code will reach [6], calling Runtime_LoadWithReceiverIC_Miss. Again, you
can notice that this function accepts both p->receiver() and p->lookup_start_object().
Good boy!

Then, the code will reach [2], meaning that we found a handler!
```
void AccessorAssembler::HandleLoadICHandlerCase(
    const LazyLoadICParameters* p, TNode<Object> handler, Label* miss,
    ExitPoint* exit_point, ICMode ic_mode, OnNonExistent on_nonexistent,
    ElementSupport support_elements, LoadAccessMode access_mode) {
  Comment("have_handler");

  TVARIABLE(Object, var_holder, p->lookup_start_object());
  TVARIABLE(Object, var_smi_handler, handler);

  Label if_smi_handler(this, {&var_holder, &var_smi_handler});
  Label try_proto_handler(this, Label::kDeferred),
      call_handler(this, Label::kDeferred);

  Branch(TaggedIsSmi(handler), &if_smi_handler, &try_proto_handler);

  BIND(&try_proto_handler);
  {
    GotoIf(IsCodeMap(LoadMap(CAST(handler))), &call_handler);
    HandleLoadICProtoHandler(p, CAST(handler), &var_holder, &var_smi_handler,
                             &if_smi_handler, miss, exit_point, ic_mode,
                             access_mode);
  }

  // |handler| is a Smi, encoding what to do. See SmiHandler methods
  // for the encoding format.
  BIND(&if_smi_handler);
  {
    HandleLoadICSmiHandlerCase(
        p, var_holder.value(), CAST(var_smi_handler.value()), handler, miss,
        exit_point, ic_mode, on_nonexistent, support_elements, access_mode);
  }

  BIND(&call_handler); <------- [6]
  {
    exit_point->ReturnCallStub(LoadWithVectorDescriptor{}, CAST(handler),
                               p->context(), p->receiver(), p->name(),
                               p->slot(), p->vector());
  }
}
```
The crash point is at [6]. What we found is a call hander.
So the method just calls the handler, but pass the p->receiver()
instead of p->lookup_start_object(). We are from LoadSuperIC, but the
HandleLoadICHandlerCase here forgets this fact! In other words, it forgets
the VERY important point: lookup_start_object != receiver.

Finally we get a type confusion bug!

This bug is very interesing. And I have provide the full exploit.
The basic idea is as follows:
It uses the bug for three times.
[0] Confuse a Object as StringWrapper to leak address of fixed array (elements of the object).
[2] Confuse a Array as StringWrapper to achieve a "mostly workable" arbitray read.
    Say it "mostly workable" becase the length field of String is not stored as SMI.
[3] Confuse a Array as Function to fake a object. Then achieve arbitray read and write.

VERSION
D8 version: V8 9.0.257.23
Chrome Version: 90.0.4430.93 stable
Operating System: 5.11.0-7612-generic Ubuntu 20.04

REPRODUCTION CASE
```
function main() {
    class C {
        m() {
            super.prototype
        }
    }
    function f() {}
```

```
  C.prototype.__proto__ = f

  let c = new C()
  c.x0 = 1
  c.x1 = 1
  c.x2 = 1
  c.x3 = 1
  c.x4 = 0x42424242 / 2

  f.prototype
  c.m()
}
for (let i = 0; i < 0x100; ++i) {
  main()
}
```
In Release version d8, it will crash immediately
```
Received signal 11 SEGV_ACCERR 297c42424241

==== C stack trace ===============================

 [0x558a5f55ad87]
 [0x7ff15c6093c0]
 [0x558a5f3eef6f]
[end of stack trace]
[1]    1332956 segmentation fault (core dumped)
```
In Debug version d8, it will complain:
```
# Fatal error in ../../src/compiler/code-assembler.cc, line 1726
# Type cast failed in Parameter 0 at ../../src/builtins/builtins-handler-gen.cc:315
  Expected JSFunction but found 0x12eb0816f4c9: [JS_OBJECT_TYPE]
```
Obviously, it's warning a type confusion bug (JS_OBECT as JSFunction).

CREDIT INFORMATION
Reporter credit: laural

poc.js
314 bytes View Download

exp.js
3.8 KB View Download

Comment 1 by sheriffbot on Mon, Apr 26, 2021, 11:24 PM EDT Project Member
Labels: external_security_report

Comment 2 by ajgo@google.com on Tue, Apr 27, 2021, 4:27 PM EDT Project Member
Status: Assigned (was: Unconfirmed)
Owner: marja@chromium.org
Cc: ishell@chromium.org jkummerow@chromium.org mvsta...@chromium.org mythria@chromium.org verwa...@chromium.org
Labels: Security_Impact-Stable Security_Severity-High OS-Android OS-Chrome OS-Fuchsia OS-Linux OS-Windows OS-Lacros Pri-1
Components: Blink>JavaScript>Compiler Blink>JavaScript>Runtime

I can confirm that this gives an access violation, sadly my symbolization is broken:-

```
C:\src\chromium\src [(b6e0b41...)]> .\out\Asan\d8.exe C:\src\pocs\1203122\poc.js
=================================================================
==53324==ERROR: AddressSanitizer: access-violation on unknown address 0x126042424241 (pc 0x1260000924cf bp 0x00c47a9feb80 sp 0x00c47a9feb58 T0)
==53324==The signal is caused by a READ memory access.
==53324==*** WARNING: Failed to initialize DbgHelp!              ***
==53324==*** Most likely this means that the app is already      ***
==53324==*** using DbgHelp, possibly with incompatible flags.    ***
==53324==*** Due to technical reasons, symbolization might crash ***
==53324==*** or produce wrong results.                           ***
    #0 0x1260000924ce  (<unknown module>)
    #1 0x12600017c619  (<unknown module>)
    #2 0xc47a9febb7  (<unknown module>)
    #3 0x22  (<unknown module>)
    #4 0x12251d22010f  (<unknown module>)
    #5 0x1b  (<unknown module>)
    #6 0xc47a9febb7  (<unknown module>)
    #7 0x126000089900  (<unknown module>)
    #8 0x45  (<unknown module>)
    #9 0x126008212b04  (<unknown module>)
```

Adding some v8 people, assigning to marja based on a code search, feel free to assign to a more appropriate owner.

Comment 3 by ClusterFuzz on Tue, Apr 27, 2021, 4:31 PM EDT Project Member
ClusterFuzz is analyzing your testcase. Developers can follow the progress at https://clusterfuzz.com/testcase?key=5687733440544768.

Comment 4 by marja@chromium.org on Wed, Apr 28, 2021, 8:01 AM EDT Project Member
Status: Started (was: Assigned)

Yep, this is mine, looking...

Comment 5 by marja@chromium.org on Wed, Apr 28, 2021, 8:19 AM EDT Project Member
Looks like a receiver vs lookup_start_object confusion. Working on a fix. Thanks for the bug report!

Comment 6 by marja@chromium.org on Wed, Apr 28, 2021, 8:56 AM EDT Project Member
Fix: https://chromium-review.googlesource.com/c/v8/v8/+/2856538

The bug is exactly where OP says it is:

> So the method just calls the handler, but pass the p->receiver()
> instead of p->lookup_start_object(). We are from LoadSuperIC, but the
> HandleLoadICHandlerCase here forgets this fact! In other words, it forgets
> the VERY important point: lookup_start_object != receiver.

Very cool!

Comment 7 by laura...@gmail.com on Wed, Apr 28, 2021, 9:26 PM EDT

Thanks, my pleasure, :D

Comment 8 by marja@chromium.org on Thu, Apr 29, 2021, 5:54 AM EDT Project Member

The fix described in OP is not enough; there's another related bug. It's about when we create the LoadIC_FunctionPrototype handler. We should only create it if the *holder* is a function, not when the *receiver* is a function.

Repro:

```
function main() {
  class A {}
  A.prototype.prototype = 'lol';
  class C extends A {
    m() {
      return super.prototype;
    }
  }
  function f() {}

  let c = new C()
  c.x0 = 1
  c.x1 = 1
  c.x2 = 1
  c.x3 = 1
  c.x4 = 0x42424242 / 2

  // Create handler; receiver is a function
  C.prototype.m.call(f);
  // Use handler; receiver not a function
  C.prototype.m.call('lol');

}
for (let i = 0; i < 0x100; ++i) {
  main();
}
```

-> I'll fix both at once :)

Comment 9 by sheriffbot on Thu, Apr 29, 2021, 12:47 PM EDT Project Member

Labels: M-90 Target-90

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 10 by laura...@gmail.com on Fri, Apr 30, 2021, 12:42 AM EDT

> It's about when we create the LoadIC_FunctionPrototype handler.
For the "create" process, do you mean here:

```
Handle<Object> LoadIC::ComputeHandler(LookupIterator* lookup) {
  Handle<Object> receiver = lookup->GetReceiver();
  ReadOnlyRoots roots(isolate());

  // `in` cannot be called on strings, and will always return true for string
  // wrapper length and function prototypes. The latter two cases are given
  // LoadHandler::LoadNativeDataProperty below.
  if (!IsAnyHas() && !lookup->IsElement()) {
    if (receiver->IsString() && *lookup->name() == roots.length_string()) {
      TRACE_HANDLER_STATS(isolate(), LoadIC_StringLength);
      return BUILTIN_CODE(isolate(), LoadIC_StringLength);
    }

    if (receiver->IsStringWrapper() && <------- [0]
        *lookup->name() == roots.length_string()) {
      TRACE_HANDLER_STATS(isolate(), LoadIC_StringWrapperLength);
      return BUILTIN_CODE(isolate(), LoadIC_StringWrapperLength);
    }

    // Use specialized code for getting prototype of functions.
    if (receiver->IsJSFunction() && <------- [1]
        *lookup->name() == roots.prototype_string() &&
        !JSFunction::cast(*receiver).PrototypeRequiresRuntimeLookup()) {
      TRACE_HANDLER_STATS(isolate(), LoadIC_FunctionPrototypeStub);
      return BUILTIN_CODE(isolate(), LoadIC_FunctionPrototype);
    }
  }
```

This is where I come up with the exploits.
  [0] LoadIC_StringWrapperLength can be used to leak information.
  [1] LoadIC_FunctionPrototype can be used to fake object.
They are just enough to complete the whole exploit. No more and No less ： D
Were it somewhere else, maybe the create of LoadIC_StringWrapperLength also deserves having a look at.

Comment 11 by Git Watcher on Fri, Apr 30, 2021, 4:12 AM EDT Project Member

The following revision refers to this bug:
  https://chromium.googlesource.com/v8/v8/+/387c803020c331ea4203c85b3bb6d9d714457375

commit 387c803020c331ea4203c85b3bb6d9d714457375
Author: Marja Hölttä <marja@chromium.org>
Date: Thu Apr 29 12:00:22 2021

[super IC] Fix a receiver vs lookup start object confusion bug

~~Bug: chromium:1203123~~
Change-Id: I80a22bbc1e700cca33e26d6a1cf294a5e9a334eb
Reviewed-on: https://chromium-review.googlesource.com/c/v8/v8/+/2856538
Reviewed-by: Igor Sheludko <ishell@chromium.org>
Commit-Queue: Marja Hölttä <marja@chromium.org>
Cr-Commit-Position: refs/heads/master@{#74290}

[modify] https://crrev.com/387c803020c331ea4203c85b3bb6d9d714457375/src/ic/accessor-assembler.cc
[modify] https://crrev.com/387c803020c331ea4203c85b3bb6d9d714457375/src/ic/ic.cc

Comment 12 by marja@chromium.org on Fri, Apr 30, 2021, 8:23 AM EDT Project Member

Status: Fixed (was: Started)

Comment 13 by ClusterFuzz on Fri, Apr 30, 2021, 12:30 PM EDT Project Member

Status: Verified (was: Fixed)
Labels: ClusterFuzz-Verified

ClusterFuzz testcase 5687733440544768 is verified as fixed in https://clusterfuzz.com/revisions?job=linux_asan_d8&range=74289:74290

If this is incorrect, please add the ClusterFuzz-Wrong label and re-open the issue.

Comment 14 by sheriffbot on Fri, Apr 30, 2021, 12:42 PM EDT Project Member

Labels: reward-topanel

Comment 15 by sheriffbot on Fri, Apr 30, 2021, 2:02 PM EDT Project Member

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

Comment 16 by sheriffbot on Fri, Apr 30, 2021, 2:22 PM EDT Project Member

Labels: Merge-Request-90 Merge-Request-91

This is sufficiently serious that it should be merged to stable. But I can't see a Chromium repo commit here, so you will need to investigate what - if anything - needs to be merged to M90. 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.

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 M91. 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 17 by sheriffbot on Fri, Apr 30, 2021, 2:25 PM EDT Project Member

Labels: -Merge-Request-91 Hotlist-Merge-Review Merge-Review-91

This bug requires manual review: M91's targeted beta branch promotion date has already passed, so this requires manual review
Before a merge request will be considered, the following information is required to be added to this bug:

1. Does your merge fit within the Merge Decision Guidelines?
- Chrome: https://chromium.googlesource.com/chromium/src.git/+/master/docs/process/merge_request.md#when-to-request-a-merge
- Chrome OS: https://goto.google.com/cros-release-branch-merge-guidelines
2. Links to the CLs you are requesting to merge.
3. Has the change landed and been verified on ToT?
4. Does this change need to be merged into other active release branches (M-1, M+1)?
5. Why are these changes required in this milestone after branch?
6. Is this a new feature?
7. If it is a new feature, is it behind a flag using finch?

Chrome OS Only:
8. Was the change reviewed and approved by the Eng Prod Representative? See Eng Prod ownership by component: http://go/cros-engprodcomponents

Please contact the milestone owner if you have questions.
Owners: benmason@(Android), bindusuvarna@(iOS), kbleicher@(ChromeOS), pbommana@(Desktop)

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

Comment 18 by pbommana@google.com on Sun, May 2, 2021, 11:33 PM EDT Project Member

Cc: adetaylor@chromium.org pbomm...@chromium.org

marja@ please reply to questions posted in comment#17. Thank you.

+Adetaylor@(Security TPM)

Comment 19 by marja@chromium.org on Mon, May 3, 2021, 5:21 AM EDT Project Member

1 Should be merged, because it's security fix

2 https://chromium-review.googlesource.com/c/v8/v8/+/2856538

3 Yes

4 Should be merged to all branches; afaics M90 is the first milestone which has this bug.

5 Security fix

6 No; related to the Super IC feature launched in M90.

Comment 20 by adetaylor@google.com on Mon, May 3, 2021, 11:09 AM EDT Project Member

Labels: -Merge-Review-91 Merge-Approved-91

Approving merge to M91.

Comment 21 by Git Watcher on Tue, May 4, 2021, 4:24 AM EDT Project Member

Labels: merge-merged-9.1

The following revision refers to this bug:
https://chromium.googlesource.com/v8/v8/+/d975d62c00cab65041b2e5a3db1ef0e79ffdc4cf

commit d975d62c00cab65041b2e5a3db1ef0e79ffdc4cf
Author: Marja Hölttä <marja@chromium.org>
Date: Thu Apr 29 12:00:22 2021

Merged: [super IC] Fix a receiver vs lookup start object confusion bug

(cherry picked from commit 387c803020c331ea4203c85b3bb6d9d714457375)

No-Try: true
No-Presubmit: true
No-Tree-Checks: true
Tbr: ishell@chromium.org
~~Bug: chromium:1203122~~
Change-Id: I34f2dfae082cab24ef92dd627bc64e78b889b9ca
Reviewed-on: https://chromium-review.googlesource.com/c/v8/v8/+/2871448
Reviewed-by: Marja Hölttä <marja@chromium.org>

Commit-Queue: Marja Hölttä <marja@chromium.org>
Cr-Commit-Position: refs/branch-heads/9.1@{#39}
Cr-Branched-From: 0e4ac64a8cf298b14034a22f9fe7b085d2cb238d-refs/heads/9.1.269@{#1}
Cr-Branched-From: f565e72d5ba88daae35a59d0f978643e2343e912-refs/heads/master@{#73847}

[modify] https://crrev.com/d975d62c00cab65041b2e5a3db1ef0e79ffdc4cf/src/ic/accessor-assembler.cc
[modify] https://crrev.com/d975d62c00cab65041b2e5a3db1ef0e79ffdc4cf/src/ic/ic.cc

Comment 22 by pbommana@google.com on Tue, May 4, 2021, 7:08 AM EDT Project Member

[Bulk Edit] Your change has been approved for M91. Please go ahead and merge the CL to branch 4472 (refs/branch-heads/4472) manually asap so that it would be part of tomorrow's Beta release.

Comment 23 by marja@chromium.org on Tue, May 4, 2021, 7:20 AM EDT Project Member

Labels: -Merge-Approved-91

merged inside v8, trying to set the right labels to indicate that...

Comment 24 by adetaylor@google.com on Tue, May 4, 2021, 12:48 PM EDT Project Member

Labels: -Merge-Request-90 Merge-Approved-90

Approving merge to M90, assuming no problems have shown up in Canary. Please merge by EOD PST Thursday for inclusion in next week's security refresh.

Comment 25 by gov...@chromium.org on Tue, May 4, 2021, 2:11 PM EDT Project Member

Please merge your change to M90 branch 4430 ASAP so we can pick it up for next M90 respin. Thank you.

Comment 26 by Git Watcher on Fri, May 7, 2021, 11:33 AM EDT Project Member

Labels: merge-merged-9.0

The following revision refers to this bug:
https://chromium.googlesource.com/v8/v8/+/e04c49f5fa4204903c3c6d3a938e017a6ef6ff92

commit e04c49f5fa4204903c3c6d3a938e017a6ef6ff92
Author: Marja Hölttä <marja@chromium.org>
Date: Thu Apr 29 12:00:22 2021

Merged: [super IC] Fix a receiver vs lookup start object confusion bug

(cherry picked from commit 387c803020c331ea4203c85b3bb6d9d714457375)

Tbr: ishell@chromium.org
Bug: chromium:1203123
No-Try: true
No-Presubmit: true
No-Tree-Checks: true
Change-Id: Ic3c130c4b5b892dd242ef002b3d61ce8c8718e8f
Reviewed-on: https://chromium-review.googlesource.com/c/v8/v8/+/2880537
Reviewed-by: Marja Hölttä <marja@chromium.org>
Commit-Queue: Marja Hölttä <marja@chromium.org>
Cr-Commit-Position: refs/branch-heads/9.0@{#57}
Cr-Branched-From: bd0108b4c88e0d6f2350cb79b5f363fbd02f3eb7-refs/heads/9.0.257@{#1}
Cr-Branched-From: 349bcc6a075411f1a7ce2d866c3dfeefc2efa39d-refs/heads/master@{#73001}

[modify] https://crrev.com/e04c49f5fa4204903c3c6d3a938e017a6ef6ff92/src/ic/accessor-assembler.cc
[modify] https://crrev.com/e04c49f5fa4204903c3c6d3a938e017a6ef6ff92/src/ic/ic.cc

Comment 27 by gov...@chromium.org on Fri, May 7, 2021, 12:25 PM EDT Project Member

Labels: -Merge-Approved-90 merge-merged-90 merge-merged-91

Already merged to M90 at #26 so removing "Merge-Approved-90" label. Thank you.

Comment 28 by amyressler@chromium.org on Fri, May 7, 2021, 5:08 PM EDT Project Member

Labels: Release-3-M90

Comment 29 by vsavu@google.com on Mon, May 10, 2021, 9:32 AM EDT Project Member

Labels: LTS-Security-86 LTS-Merge-Request-86

Comment 30 by amyressler@google.com on Mon, May 10, 2021, 9:54 AM EDT Project Member

Labels: CVE-2021-30517 CVE_description-missing

Comment 31 by vsavu@google.com on Wed, May 12, 2021, 8:55 AM EDT Project Member

https://bugs.chromium.org/p/v8/issues/detail?id=9237 has not been merged into the 8.6 branch (in M86), is this fix required for the LTS branch?

The change does not seem easy to merge back without pulling in the entire chain.

Comment 32 by marja@chromium.org on Wed, May 12, 2021, 10:56 AM EDT Project Member

The fix is not required for any branch below 9.0, since the SuperIC feature was shipped only in that version. Versions which don't have the SuperIC feature are not vulnerable.

Comment 33 by vsavu@google.com on Wed, May 12, 2021, 11:21 AM EDT Project Member

Labels: -LTS-Merge-Request-86 LTS-Security-NotApplicable-86

Comment 34 by amyressler@google.com on Wed, May 12, 2021, 7:12 PM EDT Project Member

Labels: -reward-topanel reward-unpaid reward-20000

*** 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 35 by amyressler@chromium.org on Wed, May 12, 2021, 7:33 PM EDT Project Member

Congratulations, laural! The VRP Panel has decided to award you $20,000 for this report. A member of our finance team will be in touch in the coming days to arrange payment. Very excellent work and we appreciate this submission and helping keep Chrome users safe!

Comment 36 by laura...@gmail.com on Sun, May 16, 2021, 11:52 PM EDT
Thanks, :D

Comment 37 by amyressler@google.com on Mon, May 17, 2021, 2:11 PM EDT Project Member
Labels: -reward-unpaid reward-inprocess

Comment 38 by amyressler@google.com on Fri, Jun 4, 2021, 7:23 PM EDT Project Member
Labels: -CVE_description-missing CVE_description-submitted

Comment 39 by sheriffbot on Fri, Aug 13, 2021, 1:30 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 40 by Git Watcher on Thu, Sep 2, 2021, 6:14 AM EDT Project Member
The following revision refers to this bug:
https://chromium.googlesource.com/v8/v8/+/76adfd1c86e2c6e1e54ad5c23695e9a0089852ea

commit 76adfd1c86e2c6e1e54ad5c23695e9a0089852ea
Author: Marja Hölttä <marja@chromium.org>
Date: Thu Sep 02 06:51:49 2021

[super ic] Add tests for an already fixed security bug

~~Bug: chromium:1203122~~
Change-Id: Ief88320b620dbf2f347bf6f6fc1ebd459e60af3d
Reviewed-on: https://chromium-review.googlesource.com/c/v8/v8/+/3138194
Reviewed-by: Igor Sheludko <ishell@chromium.org>
Commit-Queue: Marja Hölttä <marja@chromium.org>
Cr-Commit-Position: refs/heads/main@{#76639}

[add] https://crrev.com/76adfd1c86e2c6e1e54ad5c23695e9a0089852ea/test/mjsunit/regress/regress-crbug-1203122-1.js
[add] https://crrev.com/76adfd1c86e2c6e1e54ad5c23695e9a0089852ea/test/mjsunit/regress/regress-crbug-1203122-2.js