Closed Bug 1929623 (CVE-2025-0240) Opened 1 year ago Closed 1 year ago

MOZ_CRASH(*** Compartment mismatch 7f95a68d0b30 vs. 7f95a68d0030 at argument 1) at s/src/vm/JSContext-inl.h:56

Categories

Product: Core
Component: JavaScript Engine

Type: defect

Priority: P1 Severity: S3

Tracking

Status: RESOLVED FIXED

Milestone: 135 Branch

Tracking Flags:

	Tracking	Status
firefox-esr115	---	unaffected
firefox-esr128	134+	fixed
firefox132	---	wontfix
firefox133	---	wontfix
firefox134	+	fixed
firefox135	+	fixed

People (Reporter: sm-bugs, Assigned: jonco)

References (Blocks 2 open bugs, Regression)

Details (4 keywords, Whiteboard: [adv-main134+][adv-ESR128.6+])

Attachments

Bug 1929623 - The result of evaluating a JSON module should be |undefined| r?jandem Details | Review
1 year ago **Jon Coppeard (:jonco)**
48 bytes, text/x-phabricator-request
pascalc : **approval-mozilla-beta+**
pascalc : **approval-mozilla-esr128+**

advisory.txt Details
1 year ago **Frederik Braun [:freddy]**
217 bytes, text/plain

Bottom ↓ Tags Timeline

Nils Bars Reporter
Description • 1 year ago

Steps to reproduce:

Version: `ff95c781c335e83ba8f5be401e479201fe28a3f5`
Args `js --fuzzing-safe <test-case>`
Input:

```
a = "".startsWith("")
function b() { return newGlobal(b) }
b.newCompartment = a
with (b()) this.moduleEvaluate(parseModule("45172", "", "json"))
```

Actual results:

```
MOZ_CRASH(*** Compartment mismatch 7f95a68d0b30 vs. 7f95a68d0030 at argument 1) at s

#0 0x56530ecb0ade in MOZ_Crash(char const*, int, char const*) reproducebuild/dist/in
#1 0x56530ecb0ade in js::ContextChecks::fail(JS::Compartment*, JS::Compartment*, int
#2 0x56530ecb0ade in js::ContextChecks::check(JS::Compartment*, int) js/src/vm/JSCon
#3 0x56530ecb0ade in js::ContextChecks::check(JSObject*, int) js/src/vm/JSContext-in
#4 0x56530ee2a144 in void JSContext::checkImpl<JS::Handle<JSObject*>, JS::Handle<JS:
#5 0x56530f16b63f in void JSContext::check<JS::Handle<JSObject*>, JS::Handle<JS::Val
#6 0x56530f16b63f in js::ResolvePromiseInternal(JSContext*, JS::Handle<JSObject*>, J
#7 0x56530f0feabe in SyntheticModuleEvaluate(JSContext*, JS::Handle<js::ModuleObject
#8 0x56530f0feabe in JS::ModuleEvaluate(JSContext*, JS::Handle<JSObject*>, JS::Mutab
#9 0x56530ebea7d6 in ModuleEvaluate(JSContext*, unsigned int, JS::Value*) js/src/she
#10 0x56530ecb811e in CallJSNative(JSContext*, bool (*)(JSContext*, unsigned int, JS
#11 0x56530ecb737f in js::InternalCallOrConstruct(JSContext*, JS::CallArgs const&, j
#12 0x56530eccf574 in js::CallFromStack(JSContext*, JS::CallArgs const&, js::CallRea
#13 0x56530eccf574 in js::Interpret(JSContext*, js::RunState&) js/src/vm/Interpreter
#14 0x56530ecb61b0 in js::RunScript(JSContext*, js::RunState&) js/src/vm/Interpreter
#15 0x56530ecbb561 in js::ExecuteKernel(JSContext*, JS::Handle<JSScript*>, JS::Handl
#16 0x56530ecbbd6c in js::Execute(JSContext*, JS::Handle<JSScript*>, JS::Handle<JSOb
#17 0x56530ef0ac69 in ExecuteScript(JSContext*, JS::Handle<JSObject*>, JS::Handle<JS
#18 0x56530ef0aee7 in JS_ExecuteScript(JSContext*, JS::Handle<JSScript*>) js/src/vm/
#19 0x56530ec12cee in RunFile(JSContext*, char const*, _IO_FILE*, CompileUtf8, bool,
#20 0x56530ec11d95 in Process(JSContext*, char const*, bool, FileKind) js/src/shell/
#21 0x56530ebcb0a9 in ProcessArgs(JSContext*, js::cli::OptionParser*) js/src/shell/j
#22 0x56530ebcb0a9 in Shell(JSContext*, js::cli::OptionParser*) js/src/shell/js.cpp:
#23 0x56530ebc1f0d in main js/src/shell/js.cpp:12690:12
#24 0x7f882f3c23b7 in __libc_start_call_main csu/../sysdeps/nptl/libc_start_call_mai
```


Nils Bars Reporter
Updated • 1 year ago

Blocks: 1903968
Group: firefox-core-security → core-security
Component: Untriaged → JavaScript Engine
Product: Firefox → Core
Version: Firefox 132 → Trunk

Andrew McCreight [:mccr8]
Updated • 1 year ago

Group: core-security → javascript-core-security

Daniel Veditz [:dveditz]
Updated • 1 year ago

Keywords: reporter-external

Jan de Mooij [:jandem]
Comment 1 • 1 year ago

Reduced:

```
var m = parseModule("{}", "", "json");
newGlobal({newCompartment: true}).moduleEvaluate(m);
```

The JS shell `ModuleEvaluate` function calls `JS::ModuleEvaluate` which calls `SyntheticModuleEvaluate`. This function does this:

```
  if (!AsyncFunctionReturned(cx, resultPromise, result)) {
    return false;
  }
```

I think the problem is that `result` is the original `args.rval()` from the shell function and hasn't been set yet?

Flags: needinfo?(jcoppeard)

Jan de Mooij [:jandem]
Updated • 1 year ago

Keywords: regression
Regressed by: ~~1877791~~

BugBot [:suhaib / :marco/ :calixte]
Comment 2 • 1 year ago

Set release status flags based on info from the regressing ~~bug 1877791~~

status-firefox132: --- → affected
status-firefox133: --- → affected
status-firefox134: --- → affected
status-firefox-esr128: --- → affected

Ryan VanderMeulen [:RyanVM]
Updated • 1 year ago

status-firefox132: affected → wontfix
status-firefox-esr115: --- → unaffected

Jon Coppeard (:jonco) Assignee
Comment 3 • 1 year ago

parseModule and moduleEvaluate are shell-only functions and not available in the browser. However the underlying functions do get called and I'm not sure what consumes the evaluation result in the browser so there's a possibility that something could go awry.

Jon Coppeard (:jonco) Assignee
Updated • 1 year ago

Assignee: nobody → jcoppeard
Flags: ~~needinfo?(jcoppeard)~~

Jon Coppeard (:jonco) Assignee
Comment 4 • 1 year ago

The result of evaluation a JSON module should be |undefined| as per https://tc39.es/proposal-json-modules/#sec-smr-Evaluate step 10 and the |steps| set by https://tc39.es/proposal-json-modules/#sec-smr-Evaluate .

Jon Coppeard (:jonco) Assignee
Comment 5 • 1 year ago

Attached file ***Bug 1929623 - The result of evaluating a JSON module should be |undefined| r?jandem*** — *Details*

Jon Coppeard (:jonco) Assignee
Comment 6 • 1 year ago

Comment on attachment 9436848 [details]
~~Bug 1929623~~ - The result of evaluating a JSON module should be |undefined| r?jandem

Security Approval Request

- **How easily could an exploit be constructed based on the patch?**: Not possible as it uses shell-only functions.
- **Do comments in the patch, the check-in comment, or tests included in the patch paint a bulls-eye on the security problem?**: No
- **Which branches (beta, release, and/or ESR) are affected by this flaw, and do the release status flags reflect this affected/unaffected state correctly?**: Everything back to 125
- **If not all supported branches, which bug introduced the flaw?**: ~~Bug 1877791~~

- **Do you have backports for the affected branches?**: No
- **If not, how different, hard to create, and risky will they be?**: Should be trivial.
- **How likely is this patch to cause regressions; how much testing does it need?**: Very unlikely. It just sets a return value to undefined in one place.
- **Is the patch ready to land after security approval is given?**: Yes
- **Is Android affected?**: Yes

Steven DeTar [:sdetar]
Updated • 1 year ago

Blocks: sm-security
Severity: -- → S3
Priority: -- → P1

Andrew McCreight [:mccr8]
Comment 7 • 1 year ago

So, is this is a security problem or not? ~~Comment 3~~ says you aren't sure, but ~~comment 6~~ says there's no way to construct an exploit. What should the security rating for this be if any? Can it be unhidden? Thanks.

Flags: needinfo?(jcoppeard)

Daniel Veditz [:dveditz]
Comment 8 • 1 year ago

User script can certainly import an empty JSON module, but that will be loaded into the existing compartment won't it? How could user script trigger the creation of a new compartment with it?

Jon Coppeard (:jonco) Assignee
Comment 9 • 1 year ago

(In reply to Andrew McCreight [:mccr8] from ~~comment #7~~)
Oh I guess I was thinking based on the test code rather than based on the patch which is what the question asks. The answer is it would be very difficult to create an exploit based on the patch as it's not clear what the problem is.

I'd like to keep this a sec issue out of caution but I think it can be sec-moderate.

Flags: ~~needinfo?(jcoppeard)~~

Nils Bars Reporter
Updated • 1 year ago

Flags: sec-bounty?

Andrew McCreight [:mccr8]
Updated • 1 year ago

Keywords: csectype-uaf, sec-moderate

Ryan VanderMeulen [:RyanVM]
Updated • 1 year ago

status-firefox133: affected → wontfix

BugBot [:suhaib / :marco/ :calixte]
Comment 10 • 1 year ago

Set release status flags based on info from the regressing ~~bug 1877791~~

status-firefox135: --- → affected

Pulsebot
Comment 11 • 1 year ago

Pushed by jcoppeard@mozilla.com:
https://hg.mozilla.org/integration/autoland/rev/72c0f9e48dfb
The result of evaluating a JSON module should be |undefined| r=jandem

Pulsebot
Comment 12 • 1 year ago

Backout by amarc@mozilla.com:
https://hg.mozilla.org/integration/autoland/rev/c90592d2c066
Backed out changeset 72c0f9e48dfb for causing linting failures @ testcase.js CLOSED TREE

Sebastian Hengst [:aryx] (needinfo me if it's about an intermittent or backout)
Comment 13 • 1 year ago

Backed out for causing linting failures @ testcase.js

- Backout link
- Push with failures
- Failure Log
- Failure line:

```
TEST-UNEXPECTED-ERROR | /builds/worker/checkouts/gecko/js/src/testcase.js:1:1 | 'a'
```


Flags: needinfo?(jcoppeard)

Pulsebot
Comment 14 • 1 year ago

Pushed by jcoppeard@mozilla.com:
https://hg.mozilla.org/integration/autoland/rev/964fb7c2ab4e
The result of evaluating a JSON module should be |undefined| r=jandem

Jon Coppeard (:jonco) Assignee
Updated • 1 year ago

Flags: ~~needinfo?(jcoppeard)~~

Sebastian Hengst [:aryx] (needinfo me if it's about an intermittent or backout)
Comment 15 • 1 year ago

https://hg.mozilla.org/mozilla-central/rev/964fb7c2ab4e

Group: javascript-core-security → core-security-release
Status: NEW → RESOLVED
Closed: 1 year ago
status-firefox135: affected → fixed
Resolution: --- → FIXED
Target Milestone: --- → 135 Branch

BugBot [:suhaib / :marco/ :calixte]
Comment 16 • 1 year ago

The patch landed in nightly and beta is affected.
:jonco, is this bug important enough to require an uplift?

- If yes, please nominate the patch for beta approval.
- If no, please set `status-firefox134` to `wontfix`.

For more information, please visit BugBot documentation.

Flags: needinfo?(jcoppeard)

Ryan VanderMeulen [:RyanVM]
Updated • 1 year ago

tracking-firefox134: --- → +
tracking-firefox135: --- → +
tracking-firefox-esr128: --- → 134+

Daniel Veditz [:dveditz]
Updated • 1 year ago

Flags: sec-bounty? → sec-bounty+

Daniel Veditz [:dveditz]
Comment 17 • 1 year ago • Edited

We think it's unlikely to be exploitable by web content (~~comment 9~~) but we're still glad to have it fixed so we don't have to worry about it.

Jon Coppeard (:jonco) Assignee
Comment 18 • 1 year ago

Comment on attachment 9436848 [details]
~~Bug 1929623~~ - The result of evaluating a JSON module should be |undefined| r?jandem

Beta/Release Uplift Approval Request

- **User impact if declined/Reason for urgency**: Possible security vulnerability.
- **Is this code covered by automated tests?**: Yes
- **Has the fix been verified in Nightly?**: Yes
- **Needs manual test from QE?**: No
- **If yes, steps to reproduce**:
- **List of other uplifts needed**: None
- **Risk to taking this patch**: Low
- **Why is the change risky/not risky? (and alternatives if risky)**: This is a simple change that has been on central without for 6 days with no problems.
- **String changes made/needed**:
- **Is Android affected?**: Yes

Flags: ~~needinfo?(jcoppeard)~~
Attachment #9436848 - Flags: approval-mozilla-beta?

Jon Coppeard (:jonco) Assignee
Comment 19 • 1 year ago

Comment on attachment 9436848 [details]
~~Bug 1929623~~ - The result of evaluating a JSON module should be |undefined| r?jandem

ESR Uplift Approval Request

- **If this is not a sec:{high,crit} bug, please state case for ESR consideration**: I don't think this is exploitable but I'd like to err on the side of caution. The fix is simple.
- **User impact if declined**: Possible security vulnerability.
- **Fix Landed on Version**: 135
- **Risk to taking this patch**: Low

- **Why is the change risky/not risky? (and alternatives if risky)**: This is a simple change that has been on central without for 6 days with no problems.

Attachment #9436848 - Flags: approval-mozilla-esr128?

Pascal Chevrel:pascalc
Updated • 1 year ago

Attachment #9436848 - Flags: approval-mozilla-beta? → approval-mozilla-beta+

Pulsebot
Comment 20 • 1 year ago

uplift

https://hg.mozilla.org/releases/mozilla-beta/rev/f8ed3214e774

Pascal Chevrel:pascalc
Updated • 1 year ago

Attachment #9436848 - Flags: approval-mozilla-esr128? → approval-mozilla-esr128+

Pulsebot
Comment 21 • 1 year ago

uplift

https://hg.mozilla.org/releases/mozilla-esr128/rev/2697c92966e3

Pascal Chevrel:pascalc
Updated • 1 year ago

status-firefox134: affected → fixed
status-firefox-esr128: affected → fixed

Pascal Chevrel:pascalc
Comment 22 • 1 year ago

Jon, the uplift in esr128 is causing failures, could you have a look? https://treeherder.mozilla.org/jobs?repo=mozilla-esr128&selectedTaskRun=Ue-a8NncTJKqOVQAFAJrrg.0

Flags: needinfo?(jcoppeard)

Serban Stanca [:SerbanS]
Updated • 1 year ago

Regressions: ~~1936853~~

Andrei Vaida [:avaida]
Updated • 1 year ago

QA Whiteboard: [post-critsmash-triage]
Flags: qe-verify-

Frederik Braun [:freddy]
Updated • 1 year ago

Whiteboard: [adv-main134+][adv-ESR128.6+]

Frederik Braun [:freddy]
Comment 23 • 1 year ago

Attached file ***advisory.txt*** — *Details*

Jon Coppeard (:jonco) Assignee
Updated • 1 year ago

Flags: ~~needinfo?(jcoppeard)~~

Frederik Braun [:freddy]
Updated • 1 year ago

Alias: CVE-2025-0240

Daniel Veditz [:dveditz]
Updated • 10 months ago

Group: ~~core-security-release~~

You need to log in before you can comment on or make changes to this bug.

Top ↑