Closed Bug 1936982 (CVE-2025-1010) Opened 1 year ago Closed 1 year ago

heap-use-after-free at remove in mozilla::dom::AbstractRange::UnregisterClosestCommonInclusiveAncestor

Categories

Product: Core
Component: DOM: Selection

Type: defect

Priority: *Not set* Severity: S2

Tracking

Status: RESOLVED FIXED

Milestone: 136 Branch

Tracking Flags:

	Tracking	Status
firefox-esr115	135+	fixed
firefox-esr128	135+	fixed
firefox133	---	wontfix
firefox134	---	wontfix
firefox135	+	fixed
firefox136	+	fixed

People (Reporter: attekett, Assigned: jjaschke)

Details (Keywords: csectype-uaf, reporter-external, sec-high, Whiteboard: [adv-main135+][adv-ESR115.20+][adv-ESR128.7+])

Attachments

fuzzingfunction-heap-use-after-free-remove-AbstractRangeUnregisterClosestCommonInclusiveAncestor-AbstractRangeUnregisterSelection-StyledRangesUnregisterSelection.html Details
1 year ago **Atte Kettunen**
1.66 KB, text/html

raw-heap-use-after-free-remove-AbstractRangeUnregisterClosestCommonInclusiveAncestor-AbstractRangeUnregisterSelection-StyledRangesUnregisterSelection.html Details
1 year ago **Atte Kettunen**
1.00 KB, text/plain

full-asan-trace.txt Details
1 year ago **Atte Kettunen**
18.53 KB, text/plain

Bug 1936982 - Clean up `AbstractRange::UnregisterClosestCommonInclusiveAncestor()`. r=smaug! Details | Review
1 year ago **Jan Jaeschke [:jjaschke]**
48 bytes, text/x-phabricator-request

RyanVM : **approval-mozilla-beta+**
RyanVM : **approval-mozilla-esr115+**
RyanVM : **approval-mozilla-esr128+**
tjr : **sec-approval+**

advisory.txt Details
1 year ago **Tyson Smith [:tsmith] (PTO)**
176 bytes, text/plain

Bottom ↓ Tags ▾ Timeline ▾

Atte Kettunen Reporter
Description • 1 year ago

*Attached file **fuzzingfunction-heap-use-after-free-remove-AbstractRangeUnregisterClosestCommonInclusiveAncestor-AbstractRangeUnregisterSelection-StyledRangesUnregisterSelection.html** — Details*

Tested on:
OS: Ubuntu 22.04
Firefox:

```
fuzzfetch fetched asan build:
```

> Identified task: https://firefox-ci-tc.services.mozilla.com/api/index/v1/task/gecko.v2.mozilla-central.latest.firefox.linux64-asan-opt
> Task ID: ddn0wwpGSZyr_vbMtiUYhA
> Rank: 1734038673
> Changeset: a551f98b192c80a6b84cf2e9d9f441251d522f67
> Build ID: 20241212212433

```
fuzzfetch fetched asan-fuzzing build:
```

> Identified task: https://firefox-ci-tc.services.mozilla.com/api/index/v1/task/gecko.v2.mozilla-central.latest.firefox.linux64-fuzzing-asan-opt
> Task ID: DM5PaO_YTkKG7lOT9fkAjg
> Rank: 1734038673
> Changeset: a551f98b192c80a6b84cf2e9d9f441251d522f67
> Build ID: 20241212212433

Reproducing test case uses FuzzingFunctions garbage collection features for reliable reproduction. Test case also has `setTimeout(()=>{location.reload()},1000)`, so that is will eventually crash on regular asan build, but on my machine it can take up to 20 retries.

The original reproducing test case was simpler, but very unreliable, I'll add it as an attachment also, for reference on which parts are actually needed to reproduce.

ASAN-output snippet:

> ==43416==ERROR: AddressSanitizer: heap-use-after-free on address 0x5030008c5e10 at pc 0x706f93827533 bp 0x7ffcaa565260 sp 0x7ffcaa565258
> WRITE of size 8 at 0x5030008c5e10 thread T0 (file:// Content)
> #0 0x706f93827532 in remove /builds/worker/workspace/obj-build/dist/include/mozilla/LinkedList.h:244:18
> #1 0x706f93827532 in mozilla::dom::AbstractRange::UnregisterClosestCommonInclusiveAncestor(nsINode*, bool) /builds/worker/checkouts/gecko/dom/base/AbstractRange.cpp:483:3
> #2 0x706f9382b5dc in mozilla::dom::AbstractRange::UnregisterSelection(mozilla::dom::Selection const&) /builds/worker/checkouts/gecko/dom/base/AbstractRange.cpp:424:5
> #3 0x706f93c632b7 in mozilla::dom::Selection::StyledRanges::UnregisterSelection() /builds/worker/checkouts/gecko/dom/base/Selection.cpp:2208:24
> #4 0x706f93c6cd9e in mozilla::dom::Selection::Clear(nsPresContext*) /builds/worker/checkouts/gecko/dom/base/Selection.cpp:1447:17
> #5 0x706f93c63fb8 in mozilla::dom::Selection::RemoveAllRangesInternal(mozilla::ErrorResult&) /builds/worker/checkouts/gecko/dom/base/Selection.cpp:2420:3
> ...
> 0x5030008c5e10 is located 0 bytes inside of 24-byte region [0x5030008c5e10,0x5030008c5e28)
> freed by thread T0 (file:// Content) here:
> #0 0x56e6ab8a44b6 in free /builds/worker/fetches/llvm-project/compiler-rt/lib/asan/asan_malloc_linux.cpp:52:3
> #1 0x706f93e36ce9 in operator delete /builds/worker/workspace/obj-build/dist/include/mozilla/cxxalloc.h:51:10
> #2 0x706f93e36ce9 in operator() /builds/worker/workspace/obj-build/dist/include/mozilla/UniquePtr.h:460:5
> #3 0x706f93e36ce9 in reset /builds/worker/workspace/obj-build/dist/include/mozilla/UniquePtr.h:302:7
> #4 0x706f93e36ce9 in ~UniquePtr /builds/worker/workspace/obj-build/dist/include/mozilla/UniquePtr.h:250:18
> ...
> previously allocated by thread T0 (file:// Content) here:
> #0 0x56e6ab8a474f in malloc /builds/worker/fetches/llvm-project/compiler-rt/lib/asan/asan_malloc_linux.cpp:68:3
> #1 0x56e6ab8ebc95 in moz_xmalloc /builds/worker/checkouts/gecko/memory/mozalloc/mozalloc.cpp:52:15
> #2 0x706f9382b079 in operator new /builds/worker/workspace/obj-

> build/dist/include/mozilla/cxxalloc.h:33:10
> #3 0x706f9382b079 in MakeUnique<mozilla::LinkedList<mozilla::dom::AbstractRange> >
> /builds/worker/workspace/obj-build/dist/include/mozilla/UniquePtr.h:606:23
> #4 0x706f9382b079 in
> mozilla::dom::AbstractRange::RegisterClosestCommonInclusiveAncestor(nsINode*)
> /builds/worker/checkouts/gecko/dom/base/AbstractRange.cpp:450:14
> ...
> SUMMARY: AddressSanitizer: heap-use-after-free /builds/worker/workspace/obj-build/dist/include/mozilla/LinkedList.h:244:18 in remove

On Debug build we hit:

> [30037] Hit MOZ_CRASH(mozilla::LinkedList<mozilla::dom::AbstractRange>::~LinkedList() [T = mozilla::dom::AbstractRange] has a buggy user: it should have removed all this list's elements before the list's destruction) at /builds/worker/workspace/obj-build/dist/include/mozilla/LinkedList.h:469

Atte Kettunen Reporter
Comment 1 • 1 year ago

Attached file ***raw-heap-use-after-free-remove-AbstractRangeUnregisterClosestCommonInclusiveAncestor-AbstractRangeUnregisterSelection-StyledRangesUnregisterSelection.html*** *— Details*

Minimized test case, without FuzzingFunctions and other tricks that improve crash rate.

Andrew McCreight [:mccr8]
Updated • 1 year ago

Group: firefox-core-security → dom-core-security
Component: General → DOM: Selection
Product: Firefox → Core

Andrew McCreight [:mccr8]
Comment 2 • 1 year ago

This reproduced immediately for me on MacOS, though I think my stacks were slightly different.

```
WRITE of size 8 at 0x6030000eb300 thread T0
  #0 0x0001311380a0 in mozilla::dom::AbstractRange::UnregisterSelection(mozilla::dom
  #1 0x0001315a21a4 in mozilla::dom::Selection::Clear(nsPresContext*).
  #2 0x000131598824 in mozilla::dom::Selection::RemoveAllRangesInternal(mozilla::Err
  #3 0x0001388e5408 in nsFrameSelection::RemoveHighlightSelection(nsAtom*)
  #4 0x0001314b72a4 in mozilla::dom::HighlightRegistry::Set(nsTSubstring<char16_t> c
  #5 0x000131a692ac in mozilla::dom::HighlightRegistry_Binding::set(JSContext*, JS::
[...]
freed by thread T0 here:
    #0 0x000105cb4f98 in free
    #1 0x000131788244 in nsINode::nsSlots::~nsSlots()
    #2 0x00013142c410 in mozilla::dom::FragmentOrElement::nsDOMSlots::~nsDOMSlots()
    #3 0x00013178e8d8 in nsINode::LastRelease()
    #4 0x0001314266a4 in nsIContent::Release()
    #5 0x000131137e04 in mozilla::dom::AbstractRange::UnregisterSelection(mozilla::d
    #6 0x0001315a21a4 in mozilla::dom::Selection::Clear(nsPresContext*)
    #7 0x000131598824 in mozilla::dom::Selection::RemoveAllRangesInternal(mozilla::E
    #8 0x0001388e5408 in nsFrameSelection::RemoveHighlightSelection(nsAtom*)
    #9 0x0001314b72a4 in mozilla::dom::HighlightRegistry::Set(nsTSubstring<char16_t>
    #10 0x000131a692ac in mozilla::dom::HighlightRegistry_Binding::set(JSContext*, J
```

Jan, could you take a look? Thanks.

Severity: -- → S2
Flags: needinfo?(jjaschke)
Keywords: csectype-uaf, sec-high

Andrew McCreight [:mccr8]
Comment 3 • 1 year ago

(My stacks might just be different due to an opt build vs a noopt build or something as there's at least some overlap.)

Atte Kettunen Reporter
Comment 4 • 1 year ago

*Attached file **full-asan-trace.txt** — Details*

Adding full opt-asan build asan trace to attachments for reference.

Jan Jaeschke [:jjaschke] Assignee
Comment 5 • 1 year ago

Here's a Pernosco session, which hits the MOZ_CRASH mentioned in ~~comment 0~~.

Here is my interpretation of the stack trace and the pernosco session:

Setting a new highlight replaces the existing one. This triggers the destruction of the `Selection` which represented the old highlight, which ends up calling `AbstractRange::UnregisterSelection()`, which then calls `AbstractRange::UnregisterClosestCommonInclusiveAncestor()`.

In this method, `aNode->GetExistingClosestCommonInclusiveAncestorRanges()` is called, and returns a linked list containing two ranges.
Then, `AbstractRange::mRegisteredClosestCommonInclusiveAncestor` is set to nullptr. In this case this seems to be the last reference to that node (and also `aNode`!), so it is being destroyed. However, this also attempts to destroy `nsINode::nsSlots::mClosestCommonInclusiveAncestorRanges`, which is the same object we retrieved further up in the method, and contains two ranges. Destroying a `LinkedList` crashes for non-empty lists.

So, this alone doesn't look good. If `aNode == mRegisteredClosestCommonAncestor` and `mRegisteredClosestCommonAncestor` is the last reference, we would definitely uaf later in this method. Also, as seen in the nondebug stacktraces, the linked list in `nsSlots` isn't cleaned up properly.

What's unclear to me is why there are two ranges in the first place -- looking at the minimized test case, this should not happen. And the mentioned uafs are just a symptom of something going wrong in a different place. I'll continue to investigate.

Flags: ~~needinfo?(jjaschke)~~

Jan Jaeschke [:jjaschke] Assignee
Comment 6 • 1 year ago

No, I was wrong in interpreting the `LinkedList`. There's only one element in it.

This simplifies the situation. We're calling `AbstractRange::UnregisterClosestCommonInclusiveAncestor()` with a `aNode` which is held alive by the `AbstractRange` instance.

Jan Jaeschke [:jjaschke] Assignee
Comment 7 • 1 year ago

Attached file ***Bug 1936982 - Clean up `AbstractRange::UnregisterClosestCommonInclusiveAncestor()`. r=smaug!*** — *Details*

Phabricator Automation
Updated • 1 year ago

Assignee: nobody → jjaschke
Status: NEW → ASSIGNED

Jan Jaeschke [:jjaschke] Assignee
Comment 8 • 1 year ago

Attached patch moves (potentially) destructing the node to the end of the function. With this applied, I can't reproduce the crash anymore.

The regressor seems to be ~~Bug 1828469~~, which changed `mRegisteredClosestCommonAncestor` to be a strong reference, which was necessary for allowing StaticRanges in Selection (and to prevent the exact situation that happened here).

The patch also simplifies the code a bit, by removing the `aNode` parameter to `UnregisterClosestCommonInclusiveAncestor()` -- it *must* be `mRegisteredClosestCommonAncestor` in all cases (there's an assert for that).

Andrew McCreight [:mccr8]
Updated • 1 year ago

status-firefox133: --- → wontfix
status-firefox134: --- → fix-optional
status-firefox135: --- → affected
status-firefox-esr115: --- → affected
status-firefox-esr128: --- → affected

Jan Jaeschke [:jjaschke] Assignee
Comment 9 • 1 year ago

Comment on attachment 9443646 [details]
~~Bug 1936982~~ - Clean up `AbstractRange::UnregisterClosestCommonInclusiveAncestor()`. r=smaug!

Security Approval Request

- **How easily could an exploit be constructed based on the patch?**: Create a Highlight using CSS Highlight API using Static Ranges, then remove the DOM nodes the static range points to.
- **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?**: beta, release, ESR
- **If not all supported branches, which bug introduced the flaw?**: ~~Bug 1828469~~
- **Do you have backports for the affected branches?**: Yes
- **If not, how different, hard to create, and risky will they be?**: Fix should be trivial.
- **How likely is this patch to cause regressions; how much testing does it need?**: Not expecting any regressions.
- **Is the patch ready to land after security approval is given?**: Yes
- **Is Android affected?**: Yes

Attachment #9443646 - Flags: sec-approval?

Frederik Braun [:freddy]
Updated • 1 year ago

Flags: sec-bounty?

Andrew McCreight [:mccr8]
Updated • 1 year ago

Keywords: reporter-external

Ryan VanderMeulen [:RyanVM]
Updated • 1 year ago

status-firefox134: fix-optional → wontfix
tracking-firefox135: --- → +
tracking-firefox-esr115: --- → 135+
tracking-firefox-esr128: --- → 135+

Tom Ritter [:tjr]
Comment 10 • 1 year ago

Comment on attachment 9443646 [details]
~~Bug 1936982~~ - Clean up `AbstractRange::UnregisterClosestCommonInclusiveAncestor()`. r=smaug!

Approved to land and uplift

Attachment #9443646 - Flags: sec-approval? → sec-approval+

Pulsebot
Comment 11 • 1 year ago

Pushed by rvandermeulen@mozilla.com:
https://hg.mozilla.org/integration/autoland/rev/f8dfe669e0f8
Clean up `AbstractRange::UnregisterClosestCommonInclusiveAncestor()`. r=smaug

Sebastian Hengst [:aryx] (needinfo me if it's about an intermittent or backout)
Comment 12 • 1 year ago

https://hg.mozilla.org/mozilla-central/rev/f8dfe669e0f8

Group: dom-core-security → core-security-release
Status: ASSIGNED → RESOLVED
Closed: 1 year ago
status-firefox136: --- → fixed
Resolution: --- → FIXED
Target Milestone: --- → 136 Branch

Ryan VanderMeulen [:RyanVM]
Comment 13 • 1 year ago

Please nominate this for Beta, ESR128, and ESR115 approval. It grafts cleanly to all branches.

tracking-firefox136: --- → +
Flags: needinfo?(jjaschke)

Jan Jaeschke [:jjaschke] Assignee
Comment 14 • 1 year ago

Comment on attachment 9443646 [details]
~~Bug 1936982~~ - Clean up `AbstractRange::UnregisterClosestCommonInclusiveAncestor()`. r=smaug!

Beta/Release Uplift Approval Request

- **User impact if declined/Reason for urgency**: It's a use after free in Custom Highlight API when using static ranges.
- **Is this code covered by automated tests?**: No
- **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)**: Not risky. This patch just extends the lifetime of an object until it's safe to dispose of it.
- **String changes made/needed**:

- **Is Android affected?**: Yes

ESR Uplift Approval Request

- **If this is not a sec:{high,crit} bug, please state case for ESR consideration**:
- **User impact if declined**: It's a use after free in Custom Highlight API when using static ranges.
- **Fix Landed on Version**: 136
- **Risk to taking this patch**: Low
- **Why is the change risky/not risky? (and alternatives if risky)**: Not risky. This patch just extends the lifetime of an object until it's safe to dispose of it.

Flags: ~~needinfo?(jjaschke)~~
Attachment #9443646 - Flags: approval-mozilla-esr128?
Attachment #9443646 - Flags: approval-mozilla-esr115?
Attachment #9443646 - Flags: approval-mozilla-beta?

Ryan VanderMeulen [:RyanVM]
Comment 15 • 1 year ago

Comment on attachment 9443646 [details]
~~Bug 1936982~~ - Clean up `AbstractRange::UnregisterClosestCommonInclusiveAncestor()`. r=smaug!

Approved for 135.0b3.

Attachment #9443646 - Flags: approval-mozilla-beta? → approval-mozilla-beta+

Pulsebot
Comment 16 • 1 year ago

uplift

https://hg.mozilla.org/releases/mozilla-beta/rev/9ea70ec80ef2

Ryan VanderMeulen [:RyanVM]
Updated • 1 year ago

status-firefox135: affected → fixed

Andrei Vaida [:avaida]
Updated • 1 year ago

QA Whiteboard: [post-critsmash-triage]

Ryan VanderMeulen [:RyanVM]
Comment 17 • 1 year ago

Comment on attachment 9443646 [details]

~~Bug 1936982~~ - Clean up `AbstractRange::UnregisterClosestCommonInclusiveAncestor()`. r=smaug!

Approved for 128.7esr and 115.20esr.

Attachment #9443646 - Flags: ~~approval-mozilla-esr128?~~
Attachment #9443646 - Flags: approval-mozilla-esr128+
Attachment #9443646 - Flags: ~~approval-mozilla-esr115?~~
Attachment #9443646 - Flags: approval-mozilla-esr115+

Pulsebot
Comment 18 • 1 year ago

uplift

`https://hg.mozilla.org/releases/mozilla-esr128/rev/f13889889a47`

Pulsebot
Comment 19 • 1 year ago

uplift

`https://hg.mozilla.org/releases/mozilla-esr115/rev/c10395abe543`

Ryan VanderMeulen [:RyanVM]
Updated • 1 year ago

status-firefox-esr115: affected → fixed
status-firefox-esr128: affected → fixed

Daniel Veditz [:dveditz]
Updated • 1 year ago

Flags: sec-bounty? → sec-bounty+

Tyson Smith [:tsmith] (PTO)
Updated • 1 year ago

Whiteboard: [adv-main135+]

Tyson Smith [:tsmith] (PTO)
Updated • 1 year ago

Whiteboard: [adv-main135+] → [adv-main135+][adv-ESR115.20+][adv-ESR128.7+]

Tyson Smith [:tsmith] (PTO)
Comment 20 • 1 year ago

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

Frederik Braun [:freddy]
Updated • 1 year ago

Alias: CVE-2025-1010

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 ↑