Starred by 3 users

Owner:	drubery@chromium.org
CC:	yelizaveta@google.com jacastro@chromium.org nparker@chromium.org drubery@chromium.org vakh@chromium.org
Status:	Fixed *(Closed)*
Components:	Services>Safebrowsing>VRP
Modified:	Jul 21, 2022
Backlog-Rank:	----
Editors:	----
EstimatedDays:	----
NextAction:	----
OS:	Linux
Pri:	1
Type:	Bug-Security

Hotlist-Merge-Review
SafeBrowsing-Triaged
Arch-x86_64
Hotlist-Merge-Approved
Security_Severity-High
allpublic
reward-inprocess
reward-15000
Via-Wizard-Security
CVE_description-submitted
external_security_report
M-98
Target-98
FoundIn-98
Security_Impact-Extended
merge-merged-4758
merge-merged-98
merge-merged-4844
merge-merged-99
merge-merged-4896
merge-merged-100

Release-1-M99
CVE-2022-0973

Issue 1297498: UAF in ThreatDetailsCacheCollector::OpenEntry

Reported by ssl.b...@gmail.com on Tue, Feb 15, 2022, 6:49 AM EST

Code

UserAgent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.82 Safari/537.36

Steps to reproduce the problem:
VULNERABILITY DETAILS

There are several raw pointer in `ThreatDetailsCacheCollector` object such as `resources_`, `results_` [1] which are owned by `ThreatDetails`. UAF will occur in `ThreatDetailsCacheCollector::OpenEntry()`[2] if `ThreatDetailsCacheCollector` outlive `ThreatDetails`.

VERSION

Chrome Version: 99.0.4838
Operating System: Test on Ubuntu 18.04, should exist in all platform

REPRODUCTION CASE

To reproduce this case, we use a local build of chromium with `API_KEY` which has requested the permission of `SafeBrowsingAPI`, and set the chromium of enhanced protection mode.

We try to trigger this UAF in two ways, both of them can be triggered **without** a compromised renderer.

The first way is `SafeBrowsingBlockingPage::OnInterstitialClosing()`[3],
this will eventually invoke `OpenEntry()`. Visit a page that is blocked by safe browsing enhanced mode will reach this function.

The steps are as follow:
1. python3 -m http.server 8887
2. ./chrome http://127.0.0.1:8887/poc1.html
3. Click the "trigger" button.
4. Repeatly press F5 to reload the malware page.

You may try several times to trigger the bug (about 15 times for me).

The second way is `AdSamplerTrigger::DidFinishLoad()`[4]. Visit a page with `google_ads_iframe` will reach this function.

The steps are as follow:
1. git apply < patch.diff
2. python3 -m http.server 8887
3. ./chrome http://127.0.0.1:8887/poc2.html
4. Repeatly click the "trigger" button.

In the release build, this only has a rare chance to trigger the bug, because there are several delayed tasks that make it hard

to win the race. Besides we can only reach one `OpenEntry()` per 1000 iframes (only 10 times per day). But this doesn't block the way to escape the sandbox, the `patch.diff` patch has some limits to make it easier to win the race.

The `demo.mp4` show the second PoC.

[1]
https://source.chromium.org/chromium/chromium/src/+/main:components/safe_browsing/content/browser/threat_details_cache.h;drc=b99e35c74dcf3c62dbac78fd06696bebb2e6b9cd;l=62

[2]
https://source.chromium.org/chromium/chromium/src/+/main:components/safe_browsing/content/browser/threat_details_cache.cc;drc=b99e35c74dcf3c62dbac78fd06696bebb2e6b9cd;l=68

[3]
https://source.chromium.org/chromium/chromium/src/+/main:components/safe_browsing/content/browser/safe_browsing_blocking_page.cc;l=119

[4]
https://source.chromium.org/chromium/chromium/src/+/main:components/safe_browsing/content/browser/triggers/ad_sampler_trigger.cc;l=108

FOR CRASHES, PLEASE INCLUDE THE FOLLOWING ADDITIONAL INFORMATION

Type of crash: browser

CREDIT INFORMATION

avaue and Buff3tts at S.S.L.

What is the expected behavior?
browser process crash

What went wrong?
May lead to sandbox escape without a compromised renderer

Did this work before? N/A

Chrome version: 99.0.4838 Channel: n/a
OS Version: 18.04

poc1.html
390 bytes View Download

poc2.html
531 bytes View Download

patch.diff
2.6 KB View Download

asan1.log
44.3 KB View Download

asan2.log
21.4 KB View Download

Comment 1 by sheriffbot on Tue, Feb 15, 2022, 6:52 AM EST Project Member

Comment 1 by sheriffbot on Tue, Feb 15, 2022, 6:52 AM EST Project Member

Labels: external_security_report

Comment 2 by yelizaveta@google.com on Tue, Feb 15, 2022, 10:49 AM EST Project Member

~~Issue 1297499~~ has been merged into this issue.

Comment 3 by yelizaveta@google.com on Tue, Feb 15, 2022, 10:53 AM EST Project Member

Labels: Needs-Feedback

I don't see a demo.mp4, could you re-upload that please?

I attempted to reproduce this and did not beat the race condition for either case after numerous attempts. Do you have any other POCs I could try?

Comment 4 by ssl.b...@gmail.com on Tue, Feb 15, 2022, 8:54 PM EST

Sorry for the delay, to reproduce the bug, you will need an ASAN build with API_KEY and enable enhanced safe browsing protection in the setting.

[Deleted] **demo.mp4**

Comment 5 by sheriffbot on Tue, Feb 15, 2022, 9:00 PM EST Project Member

Cc: yelizaveta@google.com
Labels: -Needs-Feedback

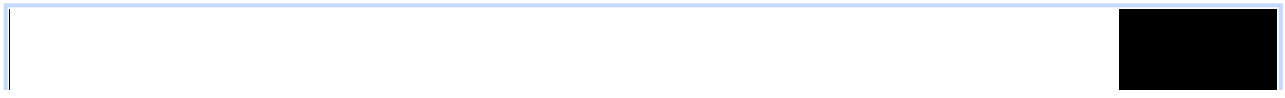
Thank you for providing more feedback. Adding the requester to the cc list.

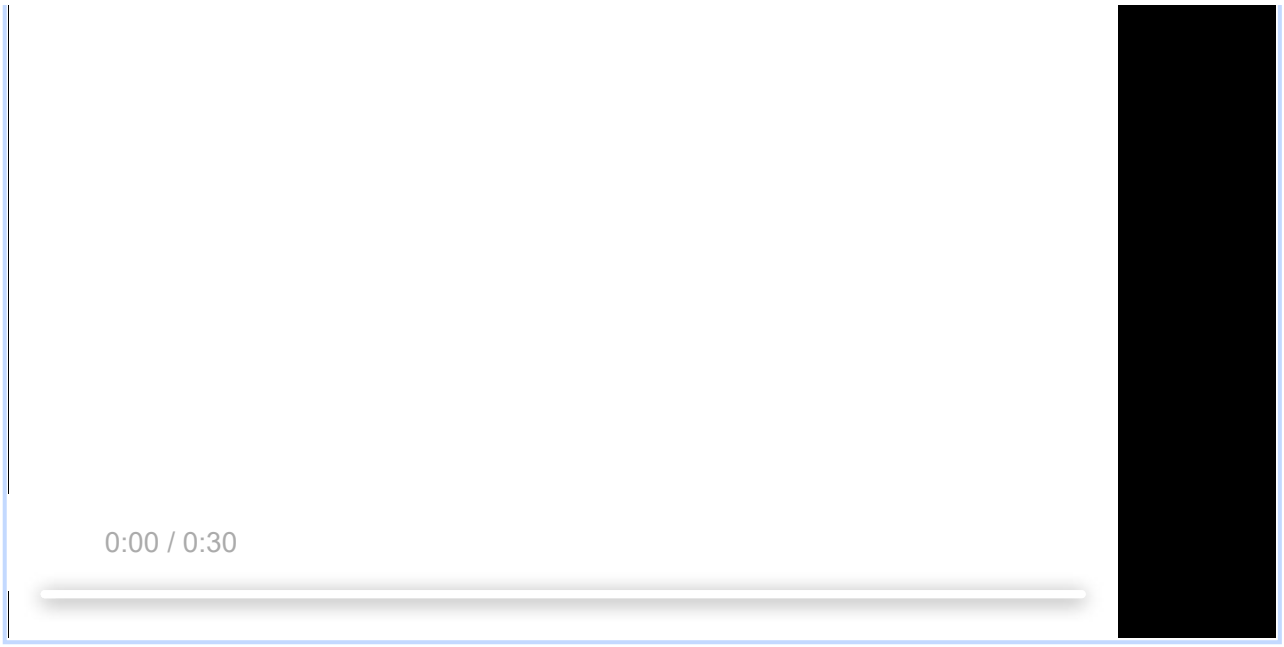
For more details visit https://www.chromium.org/issue-tracking/autotriage - Your friendly Sheriffbot

Comment 6 by ssl.b...@gmail.com on Tue, Feb 15, 2022, 9:26 PM EST

Here is the demo.mp4 for the second poc.

demo.mp4
6.9 MB View Download

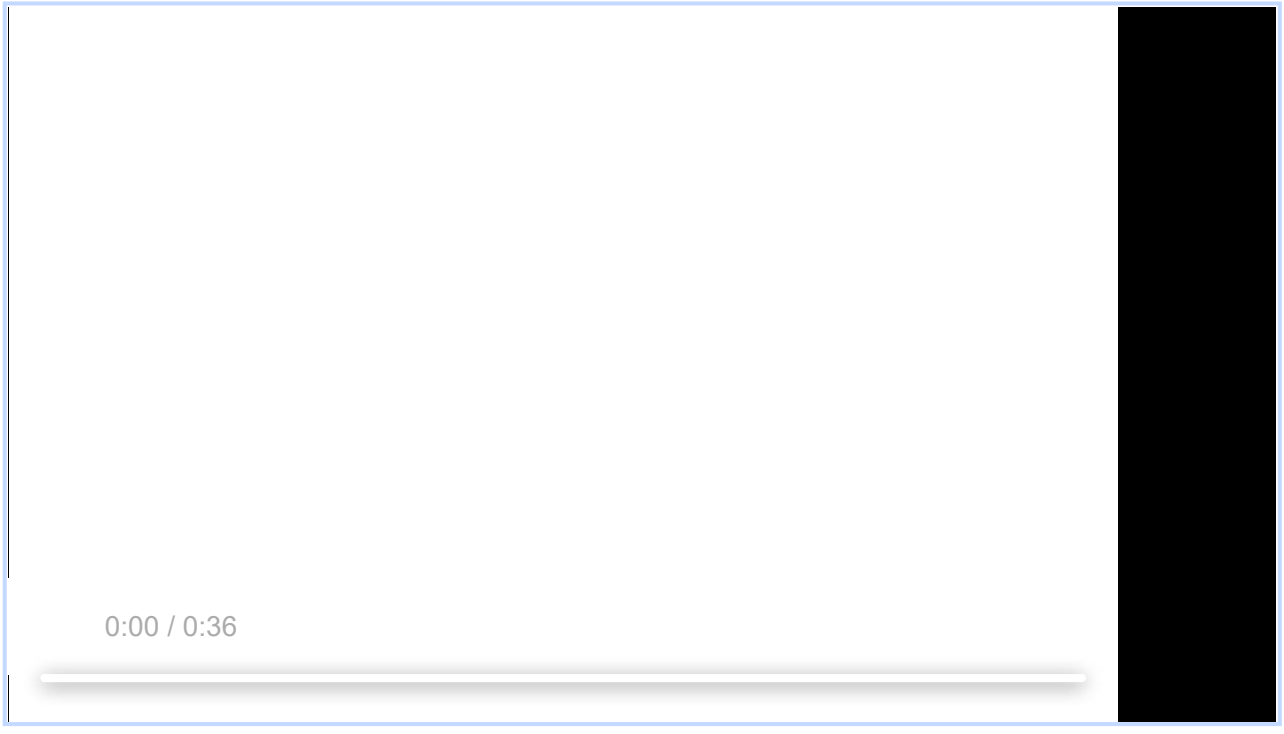




Comment 7 by ssl.b...@gmail.com on Tue, Feb 15, 2022, 9:56 PM EST

Upload the demo video for the first poc. Both poc need enable "enhanced protection mode" in settings.

demo-1.mp4
6.2 MB View Download



Comment 8 by yelizaveta@google.com on Wed, Feb 16, 2022, 1:37 PM EST Project Member

Owner: drubery@chromium.org
Cc: nparker@chromium.org
Labels: Security_Severity-High FoundIn-98 Pri-1
Components: Services>Safebrowsing>VRP

Thanks for the video! I did manage to reproduce this locally, and the extra demos helped.

I'm labeling this as high severity because I think a compromised renderer could use Mojo APIs to trigger this, which would

I'm labeling this as high severity because I think a compromised renderer could use Mojo APIs to trigger this, which would make exploitation much easier than the provided reproduction steps.

If that's an incorrect assumption we can lower the severity.

Not super sure of the root cause, OpenEntry is called in StartCacheCollection[0] which is passed in as a callback to StartHistoryCollection[1] and run in AllDone[2]. I suspect that the race condition has something to do with how AllDone() is called when urls_it_ is at urls_.end(), and rapidly clicking on the trigger button in the PoC eventually adds a url before the size of urls_ and iterator are properly updated.

[0]https://source.chromium.org/chromium/chromium/src/+/main:components/safe_browsing/content/browser/threat_details_cache.cc;l=57;bpv=1;bpt=1
[1]https://source.chromium.org/chromium/chromium/src/+/main:components/safe_browsing/content/browser/threat_details_history.cc;l=36;drc=c26af9bfc075bbed0c70d2371b9021d3a3479d92
[2]https://source.chromium.org/chromium/chromium/src/+/main:components/safe_browsing/content/browser/threat_details_history.cc;l=107;drc=c26af9bfc075bbed0c70d2371b9021d3a3479d92

Comment 9 by sheriffbot on Wed, Feb 16, 2022, 1:43 PM EST Project Member

Labels: Security_Impact-Extended

Comment 10 by sheriffbot on Wed, Feb 16, 2022, 2:17 PM EST Project Member

Status: Assigned (was: Unconfirmed)

Comment 11 by ssl.b...@gmail.com on Wed, Feb 16, 2022, 9:47 PM EST

Since the trigger to this bug doesn't need a compromised renderer and it a browser process UAF, I don't know the severity should be `critical` or `high` because the need of user interaction.

According to the ASAN log, the UAF occur in `ThreatDetailsCacheCollector::OpenEntry` and `ThreatDetailsCacheCollector::AllDone`(not `ThreatDetailsRedirectsCollector::AllDone`)

For `ThreatDetailsCacheCollector::OpenEntry`, UAF is a read access to string value `resources_it_->first`.

```
void ThreatDetailsCacheCollector::OpenEntry() {
  DCHECK_CURRENTLY_ON(BrowserThread::UI);
  DVLOG(1) << "OpenEntry";

  if (resources_it_ == resources_->end()) {
    AllDone(true);                          // <================== call to AllDone
    return;
  }

  if (!url_loader_factory_) {
    DVLOG(1) << "Missing URLLoaderFactory";
    AllDone(false);
    return;
  }

  // ...


  auto resource_request = std::make_unique<network::ResourceRequest>();
  resource_request->url = GURL(resources_it_->first); // <================== UAF(resources_it_->first)
```

```
  resource_request->url = GURL(resources_it_->first);  // <================== UAF(resources_it_->first)
  // Only from cache, and don't use cookies.
  resource_request->load_flags =
      net::LOAD_ONLY_FROM_CACHE | net::LOAD_SKIP_CACHE_VALIDATION;
  resource_request->credentials_mode = network::mojom::CredentialsMode::kOmit;
  current_load_ = network::SimpleURLLoader::Create(std::move(resource_request),
                                                   traffic_annotation);
  current_load_->DownloadToStringOfUnboundedSizeUntilCrashAndDie(
      url_loader_factory_.get(),
      base::BindOnce(&ThreatDetailsCacheCollector::OnURLLoaderComplete,
                     base::Unretained(this)));
}
```

`resources_it_` is an iterator for `resources_` and `resources_` is a raw pointer to `ThreatDetails::resources_`. `ThreatDetailsCacheCollector` may outlive `ThreatDetails` but the raw pointer in `ThreatDetailsCacheCollector` is not cleared in time.

```
void ThreatDetails::OnRedirectionCollectionReady() {
  // ...

  cache_collector_->StartCacheCollection(
      url_loader_factory_, &resources_, &cache_result_, // <================= pass as raw pointer
      base::BindOnce(&ThreatDetails::OnCacheCollectionReady, GetWeakPtr()));
}
```

```
void ThreatDetailsCacheCollector::StartCacheCollection(
    scoped_refptr<network::SharedURLLoaderFactory> url_loader_factory,
    ResourceMap* resources,
    bool* result,
    base::OnceClosure callback) {
  // Start the data collection from the HTTP cache. We use a URLFetcher
  // and set the right flags so we only hit the cache.
  DVLOG(1) << "Getting cache data for all urls...";
  url_loader_factory_ = url_loader_factory;
  resources_ = resources;
  resources_it_ = resources_->begin();
  result_ = result;

  // ...
}
```

For `ThreatDetailsCacheCollector::AllDone`, `result_` is also a raw pointer to `ThreatDetails::cache_result_` so a write access lead to UAF.

```
void ThreatDetailsCacheCollector::AllDone(bool success) {

  DVLOG(1) << "AllDone";
  DCHECK_CURRENTLY_ON(BrowserThread::UI);
  *result_ = success;                    // <---------------------- UAF(write to *result_)
```

```
  *result_ = success;                    // <==================== UAF(write to *result_)
  content::GetUIThreadTaskRunner({})->PostTask(FROM_HERE, std::move(callback_));
}
```

Comment 12 by sheriffbot on Thu, Feb 17, 2022, 12:47 PM EST Project Member

Labels: M-98 Target-98

Setting milestone and target because of high severity.

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

Comment 13 by yelizaveta@google.com on Thu, Feb 17, 2022, 12:50 PM EST Project Member

Hello,

While your PoC does work with an uncompromised renderer, the race condition is so tight that it requires a lot of user interaction to trigger. Without the patch a user would see the malicious content warning in each clickthrough, so the probability of an attacker successfully convincing a user to click through enough times to beat the race condition is not super high.

If there's a way to beat the race condition in an uncompromised renderer that requires less user interaction then the severity could be increased.

Comment 14 by flowerhorne@chromium.org on Fri, Feb 18, 2022, 10:48 AM EST Project Member

Labels: SafeBrowsing-Triaged

Comment 15 by Git Watcher on Wed, Feb 23, 2022, 12:07 AM EST Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/43bd823074abd33f430bbb94448107910680d85a

commit 43bd823074abd33f430bbb94448107910680d85a
Author: Daniel Rubery <drubery@chromium.org>
Date: Wed Feb 23 05:06:54 2022

Use WeakPtr and unique_ptr for ownership in ThreatDetails

Currently we use scoped_refptr for the ThreatDetailsCacheCollector,
which holds pointers into resources owned by ThreatDetails. When the
ThreatDetails is destroyed, it's intended that the
ThreatDetailsCacheCollector is destroyed with it, but that does not
occur if there is a pending task with a reference to the
ThreatDetailsCacheCollector.

By having the ThreatDeails hold a unique_ptr, we can ensure that
destruction happens as planned.

~~Bug: 1297498~~
Change-Id: I5f14a33d56a86c271b249534ee7410f4045f4f32
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/3482677

Reviewed-by: Xinghui Lu <xinghuilu@chromium.org>
Commit-Queue: Daniel Rubery <drubery@chromium.org>
Cr-Commit-Position: refs/heads/main@{#974062}

Cr-Commit-Position: refs/heads/main@{#974062}

[modify]
https://crrev.com/43bd823074abd33f430bbb94448107910680d85a/chrome/browser/safe_browsing/threat_details_unittest.cc
[modify]
https://crrev.com/43bd823074abd33f430bbb94448107910680d85a/components/safe_browsing/content/browser/threat_details_history.h
[modify]
https://crrev.com/43bd823074abd33f430bbb94448107910680d85a/components/safe_browsing/content/browser/threat_details_cache.cc
[modify]
https://crrev.com/43bd823074abd33f430bbb94448107910680d85a/components/safe_browsing/content/browser/threat_details_history.cc
[modify]
https://crrev.com/43bd823074abd33f430bbb94448107910680d85a/components/safe_browsing/content/browser/threat_details.h
[modify]
https://crrev.com/43bd823074abd33f430bbb94448107910680d85a/components/safe_browsing/content/browser/threat_details_cache.h
[modify]
https://crrev.com/43bd823074abd33f430bbb94448107910680d85a/components/safe_browsing/content/browser/threat_details.cc

Comment 16 by drubery@chromium.org on Thu, Feb 24, 2022, 4:37 PM EST Project Member

Status: Fixed (was: Assigned)

Comment 17 by sheriffbot on Sun, Feb 27, 2022, 12:41 PM EST Project Member

Labels: reward-topanel

Comment 18 by sheriffbot on Sun, Feb 27, 2022, 1:40 PM EST Project Member

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

Comment 19 by sheriffbot on Sun, Feb 27, 2022, 2:00 PM EST Project Member

Labels: Merge-Request-100 Merge-Request-98 Merge-Request-99

Requesting merge to stable M98 because latest trunk commit (974062) appears to be after stable branch point (950365).

Requesting merge to beta M99 because latest trunk commit (974062) appears to be after beta branch point (961656).

Requesting merge to dev M100 because latest trunk commit (974062) appears to be after dev branch point (972766).

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

Comment 20 by sheriffbot on Sun, Feb 27, 2022, 2:01 PM EST Project Member

Labels: -Merge-Request-100 Merge-Approved-100 Hotlist-Merge-Approved

Merge approved: your change passed merge requirements and is auto-approved for M100. Please go ahead and merge the CL to branch 4896 (refs/branch-heads/4896) manually. Please contact milestone owner if you have questions.

Merge instructions:
https://chromium.googlesource.com/chromium/src.git/+/refs/heads/main/docs/process/merge_request.md

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

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

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

Comment 21 by sheriffbot on Sun, Feb 27, 2022, 2:01 PM EST Project Member

Labels: -Merge-Request-99 Hotlist-Merge-Review Merge-Review-99

Merge review required: M99 has already been cut for stable release.

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), ceb (ChromeOS), pbommana (Desktop)

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

Comment 22 by sheriffbot on Sun, Feb 27, 2022, 2:01 PM EST Project Member

Labels: -Merge-Request-98 Merge-Review-98

Merge review required: M98 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), matthewjoseph (ChromeOS), srinivassista (Desktop)

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

Comment 23 by srinivassista@google.com on Tue, Mar 1, 2022, 12:36 PM EST Project Member

This bug has been approved for merge to M100, I am cutting RC build later today for release this week so, please help complete your merge(to branch: please refer to go/chrome-branches) before 3pm PST today March 1st 2022

Comment 24 by sheriffbot on Thu, Mar 3, 2022, 12:19 PM EST Project Member

This issue has been approved for a merge. Please merge the fix to any appropriate branches as soon as possible!

If all merges have been completed, please remove any remaining Merge-Approved labels from this issue.

Thanks for your time! To disable nags, add the Disable-Nags label.

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

Comment 25 by amyressler@google.com on Thu, Mar 3, 2022, 5:23 PM EST Project Member

Labels: -reward-topanel reward-unpaid reward-15000

*** 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 26 by amyressler@chromium.org on Thu, Mar 3, 2022, 5:44 PM EST Project Member

Congratulations! The VRP Panel has decided to award you $15,000 for this report. Despite this being a bit unreliable to trigger due to the tight race condition and less reliant for targeted exploitation, it was a clever discovery and a high-quality write-up and we appreciate your efforts in finding and reporting this issue. A member of our finance team will reach out soon to arrange payment. Thanks again for reporting this issue to us and great work!

Comment 27 by amyressler@google.com on Fri, Mar 4, 2022, 6:28 PM EST Project Member

Labels: -reward-unpaid reward-inprocess

Comment 28 by drubery@chromium.org on Fri, Mar 4, 2022, 6:35 PM EST Project Member

Regarding the merge review:

Please answer the following questions so that we can safely process your merge request:
1. This is a high-severity security fix.
2. The change from #15, https://chromium-review.googlesource.com/c/chromium/src/+/3482677
3. Yes
4. Not a new feature, security fix
5. N/A
6. No manual verification required

Comment 29 by Git Watcher on Fri, Mar 4, 2022, 7:36 PM EST Project Member

Labels: -merge-approved-100 merge-merged-4896 merge-merged-100

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/99c2812f32825223c4f3f362296f1e1b85acaf8e

commit 99c2812f32825223c4f3f362296f1e1b85acaf8e
Author: Daniel Rubery <drubery@chromium.org>
Date: Sat Mar 05 00:35:12 2022

[M100] Use WeakPtr and unique_ptr for ownership in ThreatDetails

Currently we use scoped_refptr for the ThreatDetailsCacheCollector, which holds pointers into resources owned by ThreatDetails. When the ThreatDetails is destroyed, it's intended that the ThreatDetailsCacheCollector is destroyed with it, but that does not occur if there is a pending task with a reference to the ThreatDetailsCacheCollector.

By having the ThreatDeails hold a unique_ptr, we can ensure that destruction happens as planned.

(cherry picked from commit 43bd823074abd33f430bbb94448107910680d85a)

~~Bug: 1297498~~
Change-Id: I5f14a33d56a86c271b249534ee7410f4045f4f32
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/3482677
Reviewed-by: Xinghui Lu <xinghuilu@chromium.org>
Commit-Queue: Daniel Rubery <drubery@chromium.org>
Cr-Original-Commit-Position: refs/heads/main@{#974062}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/3502824
Auto-Submit: Daniel Rubery <drubery@chromium.org>
Commit-Queue: Rubber Stamper <rubber-stamper@appspot.gserviceaccount.com>
Bot-Commit: Rubber Stamper <rubber-stamper@appspot.gserviceaccount.com>
Cr-Commit-Position: refs/branch-heads/4896@{#289}
Cr-Branched-From: 1f63ff4bc27570761b35ffbc7f938f6586f7bee8-refs/heads/main@{#972766}

[modify]
https://crrev.com/99c2812f32825223c4f3f362296f1e1b85acaf8e/chrome/browser/safe_browsing/threat_details_unittest.cc
[modify]
https://crrev.com/99c2812f32825223c4f3f362296f1e1b85acaf8e/components/safe_browsing/content/browser/threat_details_history.h
[modify]
https://crrev.com/99c2812f32825223c4f3f362296f1e1b85acaf8e/components/safe_browsing/content/browser/threat_details_cache.cc
[modify]
https://crrev.com/99c2812f32825223c4f3f362296f1e1b85acaf8e/components/safe_browsing/content/browser/threat_details_history.cc
[modify]
https://crrev.com/99c2812f32825223c4f3f362296f1e1b85acaf8e/components/safe_browsing/content/browser/threat_details.h
[modify]
https://crrev.com/99c2812f32825223c4f3f362296f1e1b85acaf8e/components/safe_browsing/content/browser/threat_details_cache.h
[modify]
https://crrev.com/99c2812f32825223c4f3f362296f1e1b85acaf8e/components/safe_browsing/content/browser/threat_details.cc

Comment 30 by amyressler@chromium.org on Mon, Mar 7, 2022, 12:49 PM EST Project Member

Labels: -Merge-Review-98 -Merge-Review-99 Merge-Approved-98 Merge-Approved-99

M99 merge approved; please merge to branch 4844 before noon PST Thursday, 10 March so this fix can be in the next stable security refresh

M98 merge approved, please merge to branch 4758 so this fix can be included in Extended stable support ... thank you!

M98 merge approved, please merge to branch 4758 so this fix can be included in Extended stable support -- thank you!

Comment 31 by Git Watcher on Mon, Mar 7, 2022, 11:31 PM EST Project Member

Labels: -merge-approved-99 merge-merged-4844 merge-merged-99

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/906da494a320cd54e56004fdb496648deb4e3ce6

commit 906da494a320cd54e56004fdb496648deb4e3ce6
Author: Daniel Rubery <drubery@chromium.org>
Date: Tue Mar 08 04:30:28 2022

[M99] Use WeakPtr and unique_ptr for ownership in ThreatDetails

Currently we use scoped_refptr for the ThreatDetailsCacheCollector,
which holds pointers into resources owned by ThreatDetails. When the
ThreatDetails is destroyed, it's intended that the
ThreatDetailsCacheCollector is destroyed with it, but that does not
occur if there is a pending task with a reference to the
ThreatDetailsCacheCollector.

By having the ThreatDeails hold a unique_ptr, we can ensure that
destruction happens as planned.

(cherry picked from commit 43bd823074abd33f430bbb94448107910680d85a)

~~Bug: 1297498~~
Change-Id: I5f14a33d56a86c271b249534ee7410f4045f4f32
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/3482677
Reviewed-by: Xinghui Lu <xinghuilu@chromium.org>
Commit-Queue: Daniel Rubery <drubery@chromium.org>
Cr-Original-Commit-Position: refs/heads/main@{#974062}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/3508760
Auto-Submit: Daniel Rubery <drubery@chromium.org>
Commit-Queue: Rubber Stamper <rubber-stamper@appspot.gserviceaccount.com>
Bot-Commit: Rubber Stamper <rubber-stamper@appspot.gserviceaccount.com>
Cr-Commit-Position: refs/branch-heads/4844@{#1006}
Cr-Branched-From: 007241ce2e6c8e5a7b306cc36c730cd07cd38825-refs/heads/main@{#961656}

[modify]
https://crrev.com/906da494a320cd54e56004fdb496648deb4e3ce6/chrome/browser/safe_browsing/threat_details_unittest.cc
[modify]
https://crrev.com/906da494a320cd54e56004fdb496648deb4e3ce6/components/safe_browsing/content/browser/threat_details_history.h
[modify]
https://crrev.com/906da494a320cd54e56004fdb496648deb4e3ce6/components/safe_browsing/content/browser/threat_details_cache.cc
[modify]
https://crrev.com/906da494a320cd54e56004fdb496648deb4e3ce6/components/safe_browsing/content/browser/threat_details_history.cc

[modify]
https://crrev.com/906da494a320cd54e56004fdb496648deb4e3ce6/components/safe_browsing/content/browser/threat_details.h

ails.h
[modify]
https://crrev.com/906da494a320cd54e56004fdb496648deb4e3ce6/components/safe_browsing/content/browser/threat_details_cache.h
[modify]
https://crrev.com/906da494a320cd54e56004fdb496648deb4e3ce6/components/safe_browsing/content/browser/threat_details.cc

Comment 32 by Git Watcher on Tue, Mar 8, 2022, 1:05 AM EST Project Member

Labels: -merge-approved-98 merge-merged-4758 merge-merged-98

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/d987062f10146d85be4e4ca6658a3e747f84c253

commit d987062f10146d85be4e4ca6658a3e747f84c253
Author: Daniel Rubery <drubery@chromium.org>
Date: Tue Mar 08 06:03:58 2022

[M98] Use WeakPtr and unique_ptr for ownership in ThreatDetails

Currently we use scoped_refptr for the ThreatDetailsCacheCollector,
which holds pointers into resources owned by ThreatDetails. When the
ThreatDetails is destroyed, it's intended that the
ThreatDetailsCacheCollector is destroyed with it, but that does not
occur if there is a pending task with a reference to the
ThreatDetailsCacheCollector.

By having the ThreatDeails hold a unique_ptr, we can ensure that
destruction happens as planned.

(cherry picked from commit 43bd823074abd33f430bbb94448107910680d85a)

~~Bug: 1297498~~
Change-Id: I5f14a33d56a86c271b249534ee7410f4045f4f32
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/3482677
Reviewed-by: Xinghui Lu <xinghuilu@chromium.org>
Commit-Queue: Daniel Rubery <drubery@chromium.org>
Cr-Original-Commit-Position: refs/heads/main@{#974062}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/3508406
Auto-Submit: Daniel Rubery <drubery@chromium.org>
Commit-Queue: Xinghui Lu <xinghuilu@chromium.org>
Cr-Commit-Position: refs/branch-heads/4758@{#1236}
Cr-Branched-From: 4a2cf4baf90326df19c3ee70ff987960d59a386e-refs/heads/main@{#950365}

[modify]
https://crrev.com/d987062f10146d85be4e4ca6658a3e747f84c253/chrome/browser/safe_browsing/threat_details_unittest.cc
[modify]
https://crrev.com/d987062f10146d85be4e4ca6658a3e747f84c253/components/safe_browsing/content/browser/threat_details_history.h
[modify]

https://crrev.com/d987062f10146d85be4e4ca6658a3e747f84c253/components/safe_browsing/content/browser/threat_details_cache.cc
[modify]

[modify]
https://crrev.com/d987062f10146d85be4e4ca6658a3e747f84c253/components/safe_browsing/content/browser/threat_details_history.cc
[modify]
https://crrev.com/d987062f10146d85be4e4ca6658a3e747f84c253/components/safe_browsing/content/browser/threat_details.h
[modify]
https://crrev.com/d987062f10146d85be4e4ca6658a3e747f84c253/components/safe_browsing/content/browser/threat_details_cache.h
[modify]
https://crrev.com/d987062f10146d85be4e4ca6658a3e747f84c253/components/safe_browsing/content/browser/threat_details.cc

Comment 33 by amyressler@chromium.org on Fri, Mar 11, 2022, 3:26 PM EST Project Member

Labels: Release-1-M99

Comment 34 by amyressler@google.com on Mon, Mar 14, 2022, 6:13 PM EDT Project Member

Labels: CVE-2022-0973 CVE_description-missing

Comment 35 by drubery@chromium.org on Mon, May 16, 2022, 5:26 PM EDT Project Member

Cc: jacastro@chromium.org

Comment 36 by sheriffbot on Fri, Jun 3, 2022, 1:31 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 37 by amyressler@google.com on Thu, Jul 21, 2022, 5:06 PM EDT Project Member

Labels: CVE_description-submitted -CVE_description-missing

Comment 38 by amyressler@chromium.org on Thu, Jul 21, 2022, 6:14 PM EDT Project Member

Labels: -CVE_description-missing --CVE_description-missing