☆ Starred by 3 users

Owner:	rhalavati@chromium.org
CC:	adetaylor@chromium.org mek@chromium.org
Status:	Fixed *(Closed)*
Components:	Blink>Storage Internals>Core Privacy>Incognito
Modified:	Dec 17, 2020
Backlog-Rank:	----
Editors:	----
EstimatedDays:	----
NextAction:	2020-09-15
OS:	Linux, Windows, Chrome, Mac, Fuchsia
Pri:	1
Type:	Bug-Security

Hotlist-Merge-Review
Security_Impact-Stable
Arch-x86_64
Deadline-Exceeded
Security_Severity-High
allpublic
reward-inprocess
Unreproducible
reward-15000
Via-Wizard-Security
Test-Predator-Auto-Components
Target-84
Target-85
M-85
merge-merged-4183
merge-merged-85
merge-merged-4240
merge-merged-86
Release-2-M85

Blocking:	Issue 1105910

Issue 1100136: heap-buffer-overflow in storage::ObfuscatedFileUtilMemoryDelegate(browser process)

Reported by cdsrc...@gmail.com on Sun, Jun 28, 2020, 12:04 PM EDT

Code

UserAgent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.97 Safari/537.36

Steps to reproduce the problem:
1.downloa or build latest chrome with asan version(Chromium 86.0.4185.0)
2../chrome --incognito poc.html

If necessary, I will fill in the detailed analysis report later.

What is the expected behavior?

What went wrong?

```
==11238==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x62d0013ce3f9 at pc 0x561c64f5bb84 bp 0x7fa421a3fdf0 sp 0x7fa421a3fde8
READ of size 1 at 0x62d0013ce3f9 thread T7 (Chrome_IOThread)
    #0 0x561c64f5bb83 in construct<unsigned char, char &> buildtools/third_party/libc++/trunk/include/memory:1865:35
    #1 0x561c64f5bb83 in __construct<unsigned char, char &> buildtools/third_party/libc++/trunk/include/memory:1757:18
    #2 0x561c64f5bb83 in construct<unsigned char, char &> buildtools/third_party/libc++/trunk/include/memory:1584:14
    #3 0x561c64f5bb83 in __construct_range_forward<char *, unsigned char *> buildtools/third_party/libc++/trunk/include/memory:1677:17
    #4 0x561c64f5bb83 in __construct_at_end<char *> buildtools/third_party/libc++/trunk/include/vector:1076:5
    #5 0x561c64f5bb83 in std::__1::enable_if<(__is_forward_iterator<char*>::value) && (is_constructible<unsigned char, std::__1::iterator_traits<char*>::reference>::value),
std::__1::__wrap_iter<unsigned char*> >::type std::__1::vector<unsigned char, std::__1::allocator<unsigned char> >::insert<char*>(std::__1::__wrap_iter<unsigned char
const*>, char*, char*) buildtools/third_party/libc++/trunk/include/vector:1996:17
    #6 0x561c71cc7952 in storage::ObfuscatedFileUtilMemoryDelegate::WriteFile(base::FilePath const&, long, net::IOBuffer*, int)
storage/browser/file_system/obfuscated_file_util_memory_delegate.cc:468:29
    #7 0x561c71cf9851 in storage::MemoryFileStreamWriter::Write(net::IOBuffer*, int, base::OnceCallback<void (int)>)
storage/browser/file_system/memory_file_stream_writer.cc:44:35
    #8 0x561c71cf51d0 in storage::SandboxFileStreamWriter::WriteInternal(net::IOBuffer*, int) storage/browser/file_system/sandbox_file_stream_writer.cc:114:21
    #9 0x561c71cf55dd in storage::SandboxFileStreamWriter::DidInitializeForWrite(net::IOBuffer*, int, int) storage/browser/file_system/sandbox_file_stream_writer.cc:216:22
    #10 0x561c71cf712e in Run base/callback.h:99:12
    #11 0x561c71cf712e in storage::SandboxFileStreamWriter::DidGetUsageAndQuota(base::OnceCallback<void (int)>, blink::mojom::QuotaStatusCode, long, long)
storage/browser/file_system/sandbox_file_stream_writer.cc:201:23
    #12 0x561c71cf909d in Invoke<void (storage::SandboxFileStreamWriter::*)(base::OnceCallback<void (int)>, blink::mojom::QuotaStatusCode, long, long),
base::WeakPtr<storage::SandboxFileStreamWriter>, base::OnceCallback<void (int)>, blink::mojom::QuotaStatusCode, long, long> base/bind_internal.h:498:12
    #13 0x561c71cf909d in MakeItSo<void (storage::SandboxFileStreamWriter::*)(base::OnceCallback<void (int)>, blink::mojom::QuotaStatusCode, long, long),
base::WeakPtr<storage::SandboxFileStreamWriter>, base::OnceCallback<void (int)>, blink::mojom::QuotaStatusCode, long, long> base/bind_internal.h:657:5
    #14 0x561c71cf909d in RunImpl<void (storage::SandboxFileStreamWriter::*)(base::OnceCallback<void (int)>, blink::mojom::QuotaStatusCode, long, long),
std::__1::tuple<base::WeakPtr<storage::SandboxFileStreamWriter>, base::OnceCallback<void (int)> >, 0, 1> base/bind_internal.h:710:12
    #15 0x561c71cf909d in base::internal::Invoker<base::internal::BindState<void (storage::SandboxFileStreamWriter::*)(base::OnceCallback<void (int)>,
blink::mojom::QuotaStatusCode, long, long), base::WeakPtr<storage::SandboxFileStreamWriter>, base::OnceCallback<void (int)> >, void (blink::mojom::QuotaStatusCode,
long, long)>::RunOnce(base::internal::BindStateBase*, blink::mojom::QuotaStatusCode, long, long) base/bind_internal.h:679:12
    #16 0x561c71d1bd03 in Run base/callback.h:99:12
    #17 0x561c71d1bd03 in storage::(anonymous namespace)::DidGetUsageAndQuotaStripBreakdown(base::OnceCallback<void (blink::mojom::QuotaStatusCode, long,
long)>, blink::mojom::QuotaStatusCode, long, long, mojo::StructPtr<blink::mojom::UsageBreakdown>) storage/browser/quota/quota_manager.cc:207:23
    #18 0x561c71d387ec in Invoke<void (*)(base::OnceCallback<void (blink::mojom::QuotaStatusCode, long, long)>, blink::mojom::QuotaStatusCode, long, long,
mojo::StructPtr<blink::mojom::UsageBreakdown>), base::OnceCallback<void (blink::mojom::QuotaStatusCode, long, long)>, blink::mojom::QuotaStatusCode, long, long,
mojo::StructPtr<blink::mojom::UsageBreakdown> > base/bind_internal.h:393:12
    #19 0x561c71d387ec in MakeItSo<void (*)(base::OnceCallback<void (blink::mojom::QuotaStatusCode, long, long)>, blink::mojom::QuotaStatusCode, long, long,
mojo::StructPtr<blink::mojom::UsageBreakdown>), base::OnceCallback<void (blink::mojom::QuotaStatusCode, long, long)>, blink::mojom::QuotaStatusCode, long, long,
mojo::StructPtr<blink::mojom::UsageBreakdown> > base/bind_internal.h:637:12
    #20 0x561c71d387ec in RunImpl<void (*)(base::OnceCallback<void (blink::mojom::QuotaStatusCode, long, long)>, blink::mojom::QuotaStatusCode, long, long,
mojo::StructPtr<blink::mojom::UsageBreakdown>), std::__1::tuple<base::OnceCallback<void (blink::mojom::QuotaStatusCode, long, long)> >, 0> base/bind_internal.h:710:12
    #21 0x561c71d387ec in base::internal::Invoker<base::internal::BindState<void (*)(base::OnceCallback<void (blink::mojom::QuotaStatusCode, long, long)>,
blink::mojom::QuotaStatusCode, long, long, mojo::StructPtr<blink::mojom::UsageBreakdown>), base::OnceCallback<void (blink::mojom::QuotaStatusCode, long, long)> >,
void (blink::mojom::QuotaStatusCode, long, long, mojo::StructPtr<blink::mojom::UsageBreakdown>)>::RunOnce(base::internal::BindStateBase*,
blink::mojom::QuotaStatusCode, long, long, mojo::StructPtr<blink::mojom::UsageBreakdown>&&) base/bind_internal.h:679:12
    #22 0x561c71d2e25a in Run base/callback.h:99:12
    #23 0x561c71d2e25a in storage::QuotaManager::UsageAndQuotaInfoGatherer::Completed() storage/browser/quota/quota_manager.cc:296:26
    #24 0x561c71d568b4 in storage::QuotaTask::CallCompleted() storage/browser/quota/quota_task.cc:41:5
    #25 0x561c6b1adaf3 in Run base/callback.h:99:12
    #26 0x561c6b1adaf3 in base::(anonymous namespace)::BarrierInfo::Run() base/barrier_closure.cc:34:30
    #27 0x561c71d2efe9 in Run base/callback.h:99:12
    #28 0x561c71d2efe9 in storage::QuotaManager::UsageAndQuotaInfoGatherer::OnGotHostUsage(base::OnceCallback<void ()>, long,
mojo::StructPtr<blink::mojom::UsageBreakdown>) storage/browser/quota/quota_manager.cc:345:32
    #29 0x561c71d2fe9e in void base::internal::FunctorTraits<void (storage::QuotaManager::UsageAndQuotaInfoGatherer::*)(base::OnceCallback<void ()>, long,
mojo::StructPtr<blink::mojom::UsageBreakdown>), void>::Invoke<void (storage::QuotaManager::UsageAndQuotaInfoGatherer::*)(base::OnceCallback<void ()>, long,
mojo::StructPtr<blink::mojom::UsageBreakdown>), base::WeakPtr<storage::QuotaManager::UsageAndQuotaInfoGatherer>, base::RepeatingCallback<void ()>, long,
mojo::StructPtr<blink::mojom::UsageBreakdown> >(void (storage::QuotaManager::UsageAndQuotaInfoGatherer::*)(base::OnceCallback<void ()>, long,
mojo::StructPtr<blink::mojom::UsageBreakdown>), base::WeakPtr<storage::QuotaManager::UsageAndQuotaInfoGatherer>&&, base::RepeatingCallback<void ()>&&, long&&,
mojo::StructPtr<blink::mojom::UsageBreakdown>&&) base/bind_internal.h:498:12
    #30 0x561c71d2fc34 in MakeItSo<void (storage::QuotaManager::UsageAndQuotaInfoGatherer::*)(base::OnceCallback<void ()>, long,
mojo::StructPtr<blink::mojom::UsageBreakdown>), base::WeakPtr<storage::QuotaManager::UsageAndQuotaInfoGatherer>, base::RepeatingCallback<void ()>, long,
mojo::StructPtr<blink::mojom::UsageBreakdown> > base/bind_internal.h:657:5
    #31 0x561c71d2fc34 in RunImpl<void (storage::QuotaManager::UsageAndQuotaInfoGatherer::*)(base::OnceCallback<void ()>, long,
mojo::StructPtr<blink::mojom::UsageBreakdown>), std::__1::tuple<base::WeakPtr<storage::QuotaManager::UsageAndQuotaInfoGatherer>, base::RepeatingCallback<void
()> >, 0, 1> base/bind_internal.h:710:12
    #32 0x561c71d2fc34 in base::internal::Invoker<base::internal::BindState<void (storage::QuotaManager::UsageAndQuotaInfoGatherer::*)(base::OnceCallback<void ()>,
long, mojo::StructPtr<blink::mojom::UsageBreakdown>), base::WeakPtr<storage::QuotaManager::UsageAndQuotaInfoGatherer>, base::RepeatingCallback<void ()> >, void
(long, mojo::StructPtr<blink::mojom::UsageBreakdown>)>::RunOnce(base::internal::BindStateBase*, long, mojo::StructPtr<blink::mojom::UsageBreakdown>&&)
base/bind_internal.h:679:12
    #33 0x561c71d6091e in Run base/callback.h:99:12
    #34 0x561c71d6091e in storage::UsageTracker::FinallySendHostUsageWithBreakdown(storage::UsageTracker::AccumulateInfo*, std::__1::basic_string<char,
std::__1::char_traits<char>, std::__1::allocator<char> > const&) storage/browser/quota/usage_tracker.cc:332:25
    #35 0x561c6b1adaf3 in Run base/callback.h:99:12
    #36 0x561c6b1adaf3 in base::(anonymous namespace)::BarrierInfo::Run() base/barrier_closure.cc:34:30
    #37 0x561c71d60d78 in Run base/callback.h:99:12
    #38 0x561c71d60d78 in storage::UsageTracker::AccumulateClientHostUsage(base::OnceCallback<void ()>, storage::UsageTracker::AccumulateInfo*,
std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::allocator<char> > const&, storage::QuotaClientType, long) storage/browser/quota/usage_tracker.cc:315:23
    #39 0x561c71d662e1 in Invoke<void (storage::UsageTracker::*)(base::OnceCallback<void ()>, storage::UsageTracker::AccumulateInfo *, const
std::__1::basic_string<char> &, storage::QuotaClientType, long), base::WeakPtr<storage::UsageTracker>, base::RepeatingCallback<void ()>,
storage::UsageTracker::AccumulateInfo *, std::__1::basic_string<char>, storage::QuotaClientType, long> base/bind_internal.h:498:12
    #40 0x561c71d662e1 in MakeItSo<void (storage::UsageTracker::*)(base::OnceCallback<void ()>, storage::UsageTracker::AccumulateInfo *, const
std::__1::basic_string<char> &, storage::QuotaClientType, long), base::WeakPtr<storage::UsageTracker>, base::RepeatingCallback<void ()>,
storage::UsageTracker::AccumulateInfo *, std::__1::basic_string<char>, storage::QuotaClientType, long> base/bind_internal.h:657:5
    #41 0x561c71d662e1 in RunImpl<void (storage::UsageTracker::*)(base::OnceCallback<void ()>, storage::UsageTracker::AccumulateInfo *, const
std::__1::basic_string<char> &, storage::QuotaClientType, long), std::__1::tuple<base::WeakPtr<storage::UsageTracker>, base::RepeatingCallback<void ()>,
storage::UsageTracker::AccumulateInfo *, std::__1::basic_string<char>, storage::QuotaClientType>, 0, 1, 2, 3, 4> base/bind_internal.h:710:12
    #42 0x561c71d662e1 in base::internal::Invoker<base::internal::BindState<void (storage::UsageTracker::*)(base::OnceCallback<void ()>,
storage::UsageTracker::AccumulateInfo*, std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::allocator<char> > const&, storage::QuotaClientType, long),
base::WeakPtr<storage::UsageTracker>, base::RepeatingCallback<void ()>, storage::UsageTracker::AccumulateInfo*, std::__1::basic_string<char,
std::__1::char_traits<char>, std::__1::allocator<char> >, storage::QuotaClientType>, void (long)>::RunOnce(base::internal::BindStateBase*, long) base/bind_internal.h:679:12
    #43 0x561c71d6866e in Run base/callback.h:99:12
    #44 0x561c71d6866e in storage::ClientUsageTracker::GetHostUsage(std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::allocator<char> > const&,
base::OnceCallback<void (long)>) storage/browser/quota/client_usage_tracker.cc:143:25
    #45 0x561c71d5fd97 in storage::UsageTracker::GetHostUsageWithBreakdown(std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::allocator<char> > const&,
base::OnceCallback<void (long, mojo::StructPtr<blink::mojom::UsageBreakdown>)>) storage/browser/quota/usage_tracker.cc:158:23
    #46 0x561c71d2d992 in GetHostUsageWithBreakdown storage/browser/quota/quota_manager.cc:1173:26
    #47 0x561c71d2d992 in storage::QuotaManager::UsageAndQuotaInfoGatherer::Run() storage/browser/quota/quota_manager.cc:247:16
    #48 0x561c71d1cef0 in storage::QuotaManager::GetUsageAndQuota(url::Origin const&, blink::mojom::StorageType, base::OnceCallback<void
(blink::mojom::QuotaStatusCode, long, long)>) storage/browser/quota/quota_manager.cc:991:11
    #49 0x561c71cf62c1 in storage::SandboxFileStreamWriter::DidCreateSnapshotFile(base::OnceCallback<void (int)>, base::File::Error, base::File::Info const&,
base::FilePath const&, scoped_refptr<storage::ShareableFileReference>) storage/browser/file_system/sandbox_file_stream_writer.cc:180:41
    #50 0x561c71cf8ae8 in void base::internal::FunctorTraits<void (storage::SandboxFileStreamWriter::*)(base::OnceCallback<void (int)>, base::File::Error, base::File::Info
const&, base::FilePath const&, scoped_refptr<storage::ShareableFileReference>), void>::Invoke<void (storage::SandboxFileStreamWriter::*)(base::OnceCallback<void (int)>,
base::File::Error, base::File::Info const&, base::FilePath const&, scoped_refptr<storage::ShareableFileReference>), base::WeakPtr<storage::SandboxFileStreamWriter>,
base::OnceCallback<void (int)>, base::File::Error, base::File::Info const&, base::FilePath const&, scoped_refptr<storage::ShareableFileReference> >(void
(storage::SandboxFileStreamWriter::*)(base::OnceCallback<void (int)>, base::File::Error, base::File::Info const&, base::FilePath const&,
scoped_refptr<storage::ShareableFileReference>), base::WeakPtr<storage::SandboxFileStreamWriter>&&, base::OnceCallback<void (int)>&&, base::File::Error&&,
base::File::Info const&, base::FilePath const&, scoped_refptr<storage::ShareableFileReference>&&) base/bind_internal.h:498:12
    #51 0x561c71cf8870 in MakeItSo<void (storage::SandboxFileStreamWriter::*)(base::OnceCallback<void (int)>, base::File::Error, const base::File::Info &, const
base::FilePath &, scoped_refptr<storage::ShareableFileReference>), base::WeakPtr<storage::SandboxFileStreamWriter>, base::OnceCallback<void (int)>, base::File::Error,
const base::File::Info &, const base::FilePath &, scoped_refptr<storage::ShareableFileReference> > base/bind_internal.h:657:5
    #52 0x561c71cf8870 in RunImpl<void (storage::SandboxFileStreamWriter::*)(base::OnceCallback<void (int)>, base::File::Error, const base::File::Info &, const
base::FilePath &, scoped_refptr<storage::ShareableFileReference>), std::__1::tuple<base::WeakPtr<storage::SandboxFileStreamWriter>, base::OnceCallback<void (int)> >,
0, 1> base/bind_internal.h:710:12
    #53 0x561c71cf8870 in base::internal::Invoker<base::internal::BindState<void (storage::SandboxFileStreamWriter::*)(base::OnceCallback<void (int)>, base::File::Error,
base::File::Info const&, base::FilePath const&, scoped_refptr<storage::ShareableFileReference>), base::WeakPtr<storage::SandboxFileStreamWriter>,
base::OnceCallback<void (int)> >, void (base::File::Error, base::File::Info const&, base::FilePath const&,
scoped_refptr<storage::ShareableFileReference>)>::RunOnce(base::internal::BindStateBase*, base::File::Error, base::File::Info const&, base::FilePath const&,
scoped_refptr<storage::ShareableFileReference>&&) base/bind_internal.h:679:12
    #54 0x561c71c7c7cd in Run base/callback.h:99:12
    #55 0x561c71c7c7cd in storage::FileSystemOperationRunner::DidCreateSnapshot(unsigned long, base::OnceCallback<void (base::File::Error, base::File::Info const&,
base::FilePath const&, scoped_refptr<storage::ShareableFileReference>)>, base::File::Error, base::File::Info const&, base::FilePath const&,
scoped_refptr<storage::ShareableFileReference>) storage/browser/file_system/file_system_operation_runner.cc:688:23
    #56 0x561c71c818f2 in void base::internal::FunctorTraits<void (storage::FileSystemOperationRunner::*)(unsigned long, base::OnceCallback<void (base::File::Error,
base::File::Info const&, base::FilePath const&, scoped_refptr<storage::ShareableFileReference>)>, base::File::Error, base::File::Info const&, base::FilePath const&,
scoped_refptr<storage::ShareableFileReference>), void>::Invoke<void (storage::FileSystemOperationRunner::*)(unsigned long, base::OnceCallback<void (base::File::Error,
base::File::Info const&, base::FilePath const&, scoped_refptr<storage::ShareableFileReference>)>, base::File::Error, base::File::Info const&, base::FilePath const&,
scoped_refptr<storage::ShareableFileReference>), base::WeakPtr<storage::FileSystemOperationRunner>, unsigned long, base::OnceCallback<void (base::File::Error,
base::File::Info const&, base::FilePath const&, scoped_refptr<storage::ShareableFileReference>)>, base::File::Error, base::File::Info const&, base::FilePath const&,
scoped_refptr<storage::ShareableFileReference> >(void (storage::FileSystemOperationRunner::*)(unsigned long, base::OnceCallback<void (base::File::Error, base::File::Info
const&, base::FilePath const&, scoped_refptr<storage::ShareableFileReference>)>, base::File::Error, base::File::Info const&, base::FilePath const&,
scoped_refptr<storage::ShareableFileReference>), base::WeakPtr<storage::FileSystemOperationRunner>&&, unsigned long&&, base::OnceCallback<void (base::File::Error,
base::File::Info const&, base::FilePath const&, scoped_refptr<storage::ShareableFileReference>)>&&, base::File::Error&&, base::File::Info const&, base::FilePath const&,
scoped_refptr<storage::ShareableFileReference>&&) base/bind_internal.h:498:12
    #57 0x561c71c8166b in MakeItSo<void (storage::FileSystemOperationRunner::*)(unsigned long, base::OnceCallback<void (base::File::Error, const base::File::Info &, const
base::FilePath &, scoped_refptr<storage::ShareableFileReference>)>, base::File::Error, const base::File::Info &, const base::FilePath &,
scoped_refptr<storage::ShareableFileReference>), base::WeakPtr<storage::FileSystemOperationRunner>, unsigned long, base::OnceCallback<void (base::File::Error, const
base::File::Info &, const base::FilePath &, scoped_refptr<storage::ShareableFileReference>)>, base::File::Error, const base::File::Info &, const base::FilePath &,
scoped_refptr<storage::ShareableFileReference> > base/bind_internal.h:657:5
    #58 0x561c71c8166b in RunImpl<void (storage::FileSystemOperationRunner::*)(unsigned long, base::OnceCallback<void (base::File::Error, const base::File::Info &, const
base::FilePath &, scoped_refptr<storage::ShareableFileReference>)>, base::File::Error, const base::File::Info &, const base::FilePath &,
scoped_refptr<storage::ShareableFileReference>), std::__1::tuple<base::WeakPtr<storage::FileSystemOperationRunner>, unsigned long, base::OnceCallback<void
(base::File::Error, const base::File::Info &, const base::FilePath &, scoped_refptr<storage::ShareableFileReference>)> >, 0, 1, 2> base/bind_internal.h:710:12
    #59 0x561c71c8166b in base::internal::Invoker<base::internal::BindState<void (storage::FileSystemOperationRunner::*)(unsigned long, base::OnceCallback<void
(base::File::Error, base::File::Info const&, base::FilePath const&, scoped_refptr<storage::ShareableFileReference>)>, base::File::Error, base::File::Info const&, base::FilePath
const&, scoped_refptr<storage::ShareableFileReference>), base::WeakPtr<storage::FileSystemOperationRunner>, unsigned long, base::OnceCallback<void
(base::File::Error, base::File::Info const&, base::FilePath const&, scoped_refptr<storage::ShareableFileReference>)> >, void (base::File::Error, base::File::Info const&,
base::FilePath const&, scoped_refptr<storage::ShareableFileReference>)>::RunOnce(base::internal::BindStateBase*, base::File::Error, base::File::Info const&, base::FilePath
const&, scoped_refptr<storage::ShareableFileReference>&&) base/bind_internal.h:679:12
    #60 0x561c71ca2976 in Run base/callback.h:99:12
    #61 0x561c71ca2976 in storage::(anonymous namespace)::GetFileInfoHelper::ReplySnapshotFile(base::OnceCallback<void (base::File::Error, base::File::Info const&,
base::FilePath const&, scoped_refptr<storage::ShareableFileReference>)>) storage/browser/file_system/async_file_util_adapter.cc:89:25
    #62 0x561c71ca4ed1 in Invoke<void (storage::(anonymous namespace)::GetFileInfoHelper::*)(base::OnceCallback<void (base::File::Error, const base::File::Info &, const
base::FilePath &, scoped_refptr<storage::ShareableFileReference>)>), storage::(anonymous namespace)::GetFileInfoHelper *, base::OnceCallback<void (base::File::Error,
const base::File::Info &, const base::FilePath &, scoped_refptr<storage::ShareableFileReference>)> > base/bind_internal.h:498:12
    #63 0x561c71ca4ed1 in MakeItSo<void (storage::(anonymous namespace)::GetFileInfoHelper::*)(base::OnceCallback<void (base::File::Error, const base::File::Info &,
const base::FilePath &, scoped_refptr<storage::ShareableFileReference>)>), storage::(anonymous namespace)::GetFileInfoHelper *, base::OnceCallback<void
(base::File::Error, const base::File::Info &, const base::FilePath &, scoped_refptr<storage::ShareableFileReference>)> > base/bind_internal.h:637:12
    #64 0x561c71ca4ed1 in RunImpl<void (storage::(anonymous namespace)::GetFileInfoHelper::*)(base::OnceCallback<void (base::File::Error, const base::File::Info &, const
base::FilePath &, scoped_refptr<storage::ShareableFileReference>)>), std::__1::tuple<base::internal::OwnedWrapper<storage::(anonymous namespace)::GetFileInfoHelper,
std::__1::default_delete<storage::(anonymous namespace)::GetFileInfoHelper> >, base::OnceCallback<void (base::File::Error, const base::File::Info &, const base::FilePath
&, scoped_refptr<storage::ShareableFileReference>)> >, 0, 1> base/bind_internal.h:710:12
    #65 0x561c71ca4ed1 in base::internal::Invoker<base::internal::BindState<void (storage::(anonymous namespace)::GetFileInfoHelper::*)(base::OnceCallback<void
(base::File::Error, base::File::Info const&, base::FilePath const&, scoped_refptr<storage::ShareableFileReference>)>), base::internal::OwnedWrapper<storage::(anonymous
namespace)::GetFileInfoHelper, std::__1::default_delete<storage::(anonymous namespace)::GetFileInfoHelper> >, base::OnceCallback<void (base::File::Error,
base::File::Info const&, base::FilePath const&, scoped_refptr<storage::ShareableFileReference>)> >, void ()>::RunOnce(base::internal::BindStateBase*)
base/bind_internal.h:679:12
    #66 0x561c6b365c0b in Run base/callback.h:99:12
    #67 0x561c6b365c0b in base::(anonymous namespace)::PostTaskAndReplyRelay::RunReply(base::(anonymous namespace)::PostTaskAndReplyRelay)
base/threading/post_task_and_reply_impl.cc:114:29
    #68 0x561c6b365e80 in Invoke<void (*)(base::(anonymous namespace)::PostTaskAndReplyRelay), base::(anonymous namespace)::PostTaskAndReplyRelay>
base/bind_internal.h:393:12
    #69 0x561c6b365e80 in MakeItSo<void (*)(base::(anonymous namespace)::PostTaskAndReplyRelay), base::(anonymous namespace)::PostTaskAndReplyRelay>
base/bind_internal.h:637:12
    #70 0x561c6b365e80 in RunImpl<void (*)(base::(anonymous namespace)::PostTaskAndReplyRelay), std::__1::tuple<base::(anonymous
namespace)::PostTaskAndReplyRelay>, 0> base/bind_internal.h:710:12
    #71 0x561c6b365e80 in base::internal::Invoker<base::internal::BindState<void (*)(base::(anonymous namespace)::PostTaskAndReplyRelay), base::(anonymous
namespace)::PostTaskAndReplyRelay>, void ()>::RunOnce(base::internal::BindStateBase*) base/bind_internal.h:679:12
    #72 0x561c6b2dbeb3 in Run base/callback.h:99:12
    #73 0x561c6b2dbeb3 in base::TaskAnnotator::RunTask(char const*, base::PendingTask*) base/task/common/task_annotator.cc:142:33
    #74 0x561c6b316ab9 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWorkImpl(base::sequence_manager::LazyNow*)
base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:333:23
    #75 0x561c6b3163c8 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWork()
base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:253:36
    #76 0x561c6b418016 in base::MessagePumpLibevent::Run(base::MessagePump::Delegate*) base/message_loop/message_pump_libevent.cc:209:55
    #77 0x561c6b317d39 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::Run(bool, base::TimeDelta)
base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:452:12
    #78 0x561c6b28b4e6 in base::RunLoop::Run() base/run_loop.cc:124:14
    #79 0x561c63dab07e in content::BrowserProcessSubThread::IOThreadRun(base::RunLoop*) content/browser/browser_process_sub_thread.cc:144:11
    #80 0x561c6b36f107 in base::Thread::ThreadMain() base/threading/thread.cc:382:3
    #81 0x561c6b3ef41d in base::(anonymous namespace)::ThreadFunc(void*) base/threading/platform_thread_posix.cc:81:13
    #82 0x7fa43c7446da in start_thread (/lib/x86_64-linux-gnu/libpthread.so.0+0x76da)

0x62d0013ce3f9 is located 7 bytes to the left of 32768-byte region [0x62d0013ce400,0x62d0013d6400)
allocated by thread T7 (Chrome_IOThread) here:
    #0 0x561c604fb6bd in operator new[](unsigned long) /b/s/w/ir/cache/builder/src/third_party/llvm/compiler-rt/lib/asan/asan_new_delete.cpp:102:3
    #1 0x561c6bf6433a in IOBuffer net/base/io_buffer.cc:29:11
    #2 0x561c6bf6433a in net::IOBufferWithSize::IOBufferWithSize(int) net/base/io_buffer.cc:47:7
    #3 0x561c71c8b97b in MakeRefCounted<net::IOBufferWithSize, const int &> base/memory/scoped_refptr.h:99:16
    #4 0x561c71c8b97b in storage::FileWriterDelegate::FileWriterDelegate(std::__1::unique_ptr<storage::FileStreamWriter,
std::__1::default_delete<storage::FileStreamWriter> >, storage::FlushPolicy) storage/browser/file_system/file_writer_delegate.cc:38:18
    #5 0x561c71c78375 in storage::FileSystemOperationRunner::Write(storage::FileSystemURL const&, std::__1::unique_ptr<storage::BlobDataHandle,
std::__1::default_delete<storage::BlobDataHandle> >, long, base::RepeatingCallback<void (base::File::Error, long, bool)> const&)
storage/browser/file_system/file_system_operation_runner.cc:270:59
    #6 0x561c641d7871 in content::FileSystemManagerImpl::Write(GURL const&, std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::allocator<char> > const&,
long, mojo::PendingReceiver<blink::mojom::FileSystemCancellableOperation>, mojo::PendingRemote<blink::mojom::FileSystemOperationListener>)
content/browser/file_system/file_system_manager_impl.cc:500:43
    #7 0x561c62972b27 in blink::mojom::FileSystemManagerStubDispatch::Accept(blink::mojom::FileSystemManager*, mojo::Message*)
gen/third_party/blink/public/mojom/filesystem/file_system.mojom.cc:4970:13
    #8 0x561c6b87a8a1 in mojo::InterfaceEndpointClient::HandleValidatedMessage(mojo::Message*) mojo/public/cpp/bindings/lib/interface_endpoint_client.cc:554:54
    #9 0x561c6b887358 in mojo::MessageDispatcher::Accept(mojo::Message*) mojo/public/cpp/bindings/lib/message_dispatcher.cc:46:24
    #10 0x561c6b892ece in mojo::internal::MultiplexRouter::ProcessIncomingMessage(mojo::internal::MultiplexRouter::MessageWrapper*,
mojo::internal::MultiplexRouter::ClientCallBehavior, base::SequencedTaskRunner*) mojo/public/cpp/bindings/lib/multiplex_router.cc:953:42
    #11 0x561c6b891733 in mojo::internal::MultiplexRouter::Accept(mojo::Message*) mojo/public/cpp/bindings/lib/multiplex_router.cc:620:38
    #12 0x561c6b887442 in mojo::MessageDispatcher::Accept(mojo::Message*) mojo/public/cpp/bindings/lib/message_dispatcher.cc:41:19
    #13 0x561c6b873c98 in mojo::Connector::DispatchMessage(mojo::Message) mojo/public/cpp/bindings/lib/connector.cc:509:49
    #14 0x561c6b875607 in mojo::Connector::ReadAllAvailableMessages() mojo/public/cpp/bindings/lib/connector.cc:567:14
    #15 0x561c6b8da332 in Run base/callback.h:133:12
    #16 0x561c6b8da332 in mojo::SimpleWatcher::OnHandleReady(int, unsigned int, mojo::HandleSignalsState const&) mojo/public/cpp/system/simple_watcher.cc:292:14
    #17 0x561c6b2dbeb3 in Run base/callback.h:99:12
    #18 0x561c6b2dbeb3 in base::TaskAnnotator::RunTask(char const*, base::PendingTask*) base/task/common/task_annotator.cc:142:33
    #19 0x561c6b316ab9 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWorkImpl(base::sequence_manager::LazyNow*)
base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:333:23
    #20 0x561c6b3163c8 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWork()
base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:253:36
    #21 0x561c6b418016 in base::MessagePumpLibevent::Run(base::MessagePump::Delegate*) base/message_loop/message_pump_libevent.cc:209:55
    #22 0x561c6b317d39 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::Run(bool, base::TimeDelta)
base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:452:12
    #23 0x561c6b28b4e6 in base::RunLoop::Run() base/run_loop.cc:124:14
    #24 0x561c63dab07e in content::BrowserProcessSubThread::IOThreadRun(base::RunLoop*) content/browser/browser_process_sub_thread.cc:144:11
    #25 0x561c6b36f107 in base::Thread::ThreadMain() base/threading/thread.cc:382:3
    #26 0x561c6b3ef41d in base::(anonymous namespace)::ThreadFunc(void*) base/threading/platform_thread_posix.cc:81:13
    #27 0x7fa43c7446da in start_thread (/lib/x86_64-linux-gnu/libpthread.so.0+0x76da)

Thread T7 (Chrome_IOThread) created by T0 (chrome) here:
    #0 0x561c604bd1aa in __interceptor_pthread_create /b/s/w/ir/cache/builder/src/third_party/llvm/compiler-rt/lib/asan/asan_interceptors.cpp:214:3
    #1 0x561c6b3ee63a in base::(anonymous namespace)::CreateThread(unsigned long, bool, base::PlatformThread::Delegate*, base::PlatformThreadHandle*,
base::ThreadPriority) base/threading/platform_thread_posix.cc:120:13
    #2 0x561c6b36dfe6 in base::Thread::StartWithOptions(base::Thread::Options const&) base/threading/thread.cc:186:15
    #3 0x561c64ae2dab in content::BrowserTaskExecutor::CreateIOThread() content/browser/scheduler/browser_task_executor.cc:334:19
    #4 0x561c6a177c2c in content::ContentMainRunnerImpl::RunServiceManager(content::MainFunctionParams&, bool) content/app/content_main_runner_impl.cc:962:9
    #5 0x561c6a1774aa in content::ContentMainRunnerImpl::Run(bool) content/app/content_main_runner_impl.cc:880:12
    #6 0x561c6a31bf46 in service_manager::Main(service_manager::MainParams const&) services/service_manager/embedder/main.cc:453:29
    #7 0x561c6a1723d6 in content::ContentMain(content::ContentMainParams const&) content/app/content_main.cc:19:10
    #8 0x561c604fe7c4 in ChromeMain chrome/app/chrome_main.cc:118:12
    #9 0x7fa43505ab96 in __libc_start_main /build/glibc-OTsEL5/glibc-2.27/csu/../csu/libc-start.c:310

SUMMARY: AddressSanitizer: heap-buffer-overflow buildtools/third_party/libc++/trunk/include/memory:1865:35 in construct<unsigned char, char &>
Shadow bytes around the buggy address:
  0x0c5a80271c20: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c5a80271c30: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c5a80271c40: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c5a80271c50: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c5a80271c60: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
=>0x0c5a80271c70: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa[fa]
  0x0c5a80271c80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c5a80271c90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c5a80271ca0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c5a80271cb0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c5a80271cc0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Shadow byte legend (one shadow byte represents 8 application bytes):
  Addressable:           00
  Partially addressable: 01 02 03 04 05 06 07
  Heap left redzone:       fa
  Freed heap region:       fd
  Stack left redzone:      f1
  Stack mid redzone:       f2
  Stack right redzone:     f3
  Stack after return:      f5
  Stack use after scope:   f8
  Global redzone:          f9
  Global init order:       f6
  Poisoned by user:        f7
  Container overflow:      fc
  Array cookie:            ac
  Intra object redzone:    bb
  ASan internal:           fe
  Left alloca redzone:     ca
  Right alloca redzone:    cb
  Shadow gap:              cc
==11238==ABORTING
```

Did this work before? N/A

Chrome version: Chromium 86.0.4185.0   Channel: n/a
OS Version: 18.04
Flash Version:

poc.html
456 bytes View Download

Comment 1 by ClusterFuzz on Mon, Jun 29, 2020, 5:17 PM EDT   Project Member

ClusterFuzz is analyzing your testcase. Developers can follow the progress at https://clusterfuzz.com/testcase?key=5645076925054976.

Comment 2 by ClusterFuzz on Mon, Jun 29, 2020, 6:15 PM EDT   Project Member

ClusterFuzz is analyzing your testcase. Developers can follow the progress at https://clusterfuzz.com/testcase?key=5660411770241024.

Comment 3 by bdea@chromium.org on Mon, Jun 29, 2020, 6:46 PM EDT   Project Member

Owner: rhalavati@chromium.org
Labels: Security_Impact-Stable Security_Severity-Medium
Components: Blink>Storage Privacy>Incognito

@rhalavati could you take a look at this?

Comment 4 by bdea@chromium.org on Mon, Jun 29, 2020, 7:06 PM EDT   Project Member

Status: Assigned (was: Unconfirmed)

Comment 5 by sheriffbot on Tue, Jun 30, 2020, 2:10 PM EDT   Project Member

Labels: Target-84 M-84

Setting milestone and target because of Security_Impact=Stable and medium severity.

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

Comment 6 by sheriffbot on Tue, Jun 30, 2020, 2:46 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 7 by sheriffbot on Sun, Jul 12, 2020, 1:32 PM EDT   Project Member

rhalavati: Uh oh! This issue still open and hasn't been updated in the last 14 days. This is a serious vulnerability, and we want to ensure that there's progress. Could you please leave an update with the current status and any potential blockers?

If you're not the right owner for this issue, could you please remove yourself as soon as possible or help us find the right one?

If the issue is fixed or you can't reproduce it, please close the bug. If you've started working on a fix, please set the status to Started.

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 8 by adetaylor@google.com on Mon, Jul 13, 2020, 1:18 PM EDT   Project Member

Labels: -Security_Severity-Medium Security_Severity-High

This appears to be a browser process heap buffer overflow triggered directly from web content, which is a Critical bug. It's mitigated by the fact that it (presumably) only works in Incognito mode, so I'm going to call this High, but it's borderline High/Critical. We absolutely need to get this fix into the first M84 refresh.

rhalavati@ please take a look urgently.

Comment 9 by sheriffbot on Mon, Jul 13, 2020, 1:32 PM EDT Project Member

rhalavati: Uh oh! This issue still open and hasn't been updated in the last 15 days. This is a serious vulnerability, and we want to ensure that there's progress. Could you please leave an update with the current status and any potential blockers?

If you're not the right owner for this issue, could you please remove yourself as soon as possible or help us find the right one?

If the issue is fixed or you can't reproduce it, please close the bug. If you've started working on a fix, please set the status to Started.

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 10 by ClusterFuzz on Mon, Jul 13, 2020, 3:49 PM EDT Project Member

Labels: Unreproducible

ClusterFuzz testcase 5660411770241024 appears to be flaky, updating reproducibility label.

Comment 11 by ClusterFuzz on Mon, Jul 13, 2020, 3:49 PM EDT Project Member

Detailed Report: https://clusterfuzz.com/testcase?key=5660411770241024

Fuzzer:
Job Type: linux_asan_chrome_mp
Platform Id: linux

Crash Type: CHECK failure
Crash Address:
Crash State:
  r. Sending zygote magic failed in zygote_linux.cc
  content::Zygote::ProcessRequests
  content::ZygoteMain

Sanitizer: address (ASAN)

Crash Revision: https://clusterfuzz.com/revisions?job=linux_asan_chrome_mp&revision=783658

Reproducer Testcase: https://clusterfuzz.com/download?testcase_id=5660411770241024

The reproduce tool requires a ClusterFuzz source checkout. To prepare one, run:

```
git clone https://github.com/google/clusterfuzz && cd clusterfuzz && git checkout tags/reproduce-tool-stable
```

To reproduce this issue, run:

```
./reproduce.sh -t https://clusterfuzz.com/testcase-detail/5660411770241024 -b /path/to/build
```

Please use the GN arguments provided in this report when building the binary. If you have any feedback on reproducing test cases, let us know at https://forms.gle/Yh3qCYFveHj6E5jz5 so we can improve.

************************* UNREPRODUCIBLE *************************
Note: This crash might not be reproducible with the provided testcase. That said, for the past 14 days, we've been seeing this crash frequently.

It may be possible to reproduce by trying the following options:
- Run testcase multiple times for a longer duration.
- Run fuzzing without testcase argument to hit the same crash signature.

If it still does not reproduce, try a speculative fix based on the crash stacktrace and verify if it works by looking at the crash statistics in the report. We will auto-close the bug if the crash is not seen for 14 days.
****

Comment 12 by ClusterFuzz on Mon, Jul 13, 2020, 3:56 PM EDT Project Member

Labels: Test-Predator-Auto-Components
Components: Internals>Core

Automatically applying components based on crash stacktrace and information from OWNERS files.

If this is incorrect, please apply the Test-Predator-Wrong-Components label.

Comment 13 by rhalavati@chromium.org on Tue, Jul 14, 2020, 4:17 AM EDT Project Member

Status: Started (was: Assigned)

Sorry for the delay, starting this.

Comment 14 by rhalavati@chromium.org on Tue, Jul 14, 2020, 7:18 AM EDT Project Member

Labels: Needs-Feedback

I build chrome on Linux with the following goma args:

```
is_asan = true
is_debug = false
enable_full_stack_frames_for_profiling = true
```

And ran out/asan/chrome --incognito poc.html

And the error did not reproduced. Have I missed anything?

Comment 15 by cdsrc...@gmail.com on Tue, Jul 14, 2020, 1:37 PM EDT

Sorry, I missed 1 step.You must use webserver.

```
python3.6m -m http.server 8605
google-chrome --incognito http://127.0.0.1:8605/poc.html
```

If it doesn't repro,try poc1.html and original.html .

However, there are many kinds of crashes in these two pocs, sometimes null pointer or UAF. But stracktrace is similar(tested with Version 86.0.4196.0 (Developer Build) (64-bit)).

poc2.zip
8.2 KB Download

Comment 16 by rhalavati@chromium.org on Wed, Jul 15, 2020, 6:34 AM EDT Project Member

Thank you. I could reproduce the issue. Will update after investigating.

Comment 17 by rhalavati@chromium.org on Thu, Jul 16, 2020, 9:33 AM EDT Project Member

Blocking: 1105910

Comment 18 by rhalavati@chromium.org on Thu, Jul 16, 2020, 9:34 AM EDT Project Member

Cc: mek@chromium.org
Labels: -Needs-Feedback

Comment 19 by rhalavati@chromium.org on Tue, Jul 21, 2020, 9:44 AM EDT Project Member

Cc: adetaylor@chromium.org

I spent a few days on this, on my only guess till now is this:
The script generates numerous small files and keeps appending to them. Through these operations, and many STL functions for resizing the arrays and getting/releasing memory, heap gets fragmented enough to fail in some cases and do unexpected behavior. Since we don't get any error when allocations fail, we get an issue later.

If there would be a safe memory allocation function, or a function that can check an address is already allocated or not it might make debug easier.

P.S.,
Just an observation, in crrev.com/c/2308721 I have commented out the part that actually allocates memory for the file contents. This reduces the frequency of errors by a huge factor, but still sometimes there are entries in the directory tree (lines 454-456) which are null, and that cannot exist.

Comment 20 by adetaylor@chromium.org on Tue, Jul 21, 2020, 11:18 AM EDT Project Member

rhalavati@ are you using an ASAN build? Did you manage to reproduce it with ASAN? In the event that you've got memory problems (e.g. a use-after-free) it will give you three different call stacks; for the original allocation, the free, and the use.

I am doubtful that it's an underlying error in the allocator, but I guess it's possible...

Comment 21 by rhalavati@chromium.org on Wed, Jul 22, 2020, 7:23 AM EDT Project Member

Thank you for pointing that out. I was using the ASAN build, but had not paid attention that the different stack traces are related to different threads.

It seems that this issue is due to a thread race. I've enclosed the debug report w.r.t. http://crrev.com/c/2308721/2.
The scenario seems to be that thread T6 is trying to write to a file, and to do so is expanding file's buffer and changing the buffer address. At the same time thread T27 is trying to remove the file and it is freeing the memory from the previous buffer address.

Am I reading this correctly? Should this module be able to handle multiple threads at the same time?

P.S., related lines in report:

```
==137596==ERROR: AddressSanitizer: attempting free on address which was not malloc()-ed: 0x61d000271468 in thread T27 (ThreadPoolForeg)
   #15 0x55c5e2316f1e in erase ./../../buildtools/third_party/libc++/trunk/include/map:1304:25
   #16 0x55c5e2316f1e in storage::ObfuscatedFileUtilMemoryDelegate::DeleteFile(base::FilePath const&)
./../../storage/browser/file_system/obfuscated_file_util_memory_delegate.cc:231:33

0x61d000271468 is located 1000 bytes inside of 2000-byte region [0x61d000271080,0x61d000271850)
allocated by thread T6 (Chrome_IOThread) here:
   #6 0x55c5e231aedd in resize ./../../buildtools/third_party/libc++/trunk/include/vector:2022:15
   #7 0x55c5e231aedd in storage::ObfuscatedFileUtilMemoryDelegate::WriteFile(base::FilePath const&, long, net::IOBuffer*, int)
./../../storage/browser/file_system/obfuscated_file_util_memory_delegate.cc:468:29
```

asan_debug_report.txt
33.5 KB View Download

Comment 22 by adetaylor@chromium.org on Wed, Jul 22, 2020, 11:44 AM EDT Project Member

OK! I agree with your interpretation that both the IO thread and thread pool workers seem to be fiddling with the same data structures and that this is bad. I'm not sure that this can be the whole picture, since the original report didn't involve multiple threads, though I can imagine there could be ways that this data race could corrupt data in such a way that we'd get the single-threaded ASAN problems reported at the outset.

Comment 23 by rhalavati@chromium.org on Fri, Jul 31, 2020, 8:26 AM EDT Project Member

I could not reproduce the original issue, but experienced different errors, all centered on the vector resize and vector insert operations (and none on actual data write).
Therefore I think although the original report doesn't have any multi-thread signature, that one can also be due to some wrongly handled the data structure because of thread races.
I think now that we know this code can be triggered by multiple threads, and we know that it is not thread-safe, it's reasonable to fix that issue, and look for this anomaly again after that is fixed.
What I wonder here is why there isn't any thread race handling in the not-in-memory filesystem. It's hard to assume that everything is handled by the OS.

Comment 24 by mek@chromium.org on Fri, Jul 31, 2020, 12:10 PM EDT Project Member

Is the race only between WriteFile/ReadFile (and perhaps sometimes GetFileInfo) being called on the IO thread, and all other methods called on a ThreadPool seqeuence? I think that makes sense. WriteFile/ReadFile are not part of the ObfuscatedFileUtilDelegate interface; they were just added to implement FileStreamReader and FileStreamWriter for the in-memory file system. And while normally all ObfuscatedFileUtilDelegate methods get called on a threadpool sequence, FileStreamReader/FileStreamWriter on the other hand always get called on the IO thread.

One way to fix this would be to pass the correct task runner to use when creating a FSWriter/Reader for the in-memory file system (file_system_context_->default_file_task_runner() is passed to the native file file stream reader already), and then post to that task runner to call the delegate... Or just make the delegate thread safe I guess.

Comment 25 by rhalavati@chromium.org on Mon, Aug 3, 2020, 10:54 AM EDT Project Member

Thank you Mek, I will be on it.

Comment 26 by rhalavati@chromium.org on Fri, Aug 7, 2020, 7:38 AM EDT Project Member

Status update:
crrev.com/c/2308721 and crrev.com/c/2339343 in review to update FileStreamWriter and FileStreamReader to use the same thread as the other operations.

On a more general scope, starting another CL to update the memory pointers to ObfuscatedFileUtilMemoryDelegate.

Comment 27 by sheriffbot on Wed, Aug 26, 2020, 1:38 PM EDT Project Member

Labels: -M-84 Target-85 M-85

Comment 28 by bugdroid on Fri, Aug 28, 2020, 4:47 AM EDT Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src.git/+/88d45f783b7ee184994690303e7c01ace3105c45

commit 88d45f783b7ee184994690303e7c01ace3105c45
Author: Ramin Halavati <rhalavati@chromium.org>
Date: Fri Aug 28 08:46:51 2020

Run ObfuscatedFileUtilMemoryDelegate entirely on TaskRunner.

MemoryFileStreamWriter called some ObfuscatedFileUtilMemoryDelegate
functions through IO thread while other functions in OFUMD are called

on a threadpool sequence. This could result in races in updating
directory structure.

To fix the issue, MemoryFileStreamWriter and MemoryFileStreamReader are
updated to call all OFUMD on the default task runner of the file system
context.

~~Bug: 1100136~~
Change-Id: I59146ca690eee810c52f807bd1fb4ef2b1f2c929
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2308721
Commit-Queue: Ramin Halavati <rhalavati@chromium.org>
Reviewed-by: Marijn Kruisselbrink <mek@chromium.org>
Cr-Commit-Position: refs/heads/master@{#802584}

[modify] https://crrev.com/88d45f783b7ee184994690303e7c01ace3105c45/storage/browser/file_system/file_stream_reader.h
[modify] https://crrev.com/88d45f783b7ee184994690303e7c01ace3105c45/storage/browser/file_system/file_stream_test_utils.cc
[modify] https://crrev.com/88d45f783b7ee184994690303e7c01ace3105c45/storage/browser/file_system/file_stream_test_utils.h
[modify] https://crrev.com/88d45f783b7ee184994690303e7c01ace3105c45/storage/browser/file_system/file_stream_writer.h
[modify] https://crrev.com/88d45f783b7ee184994690303e7c01ace3105c45/storage/browser/file_system/file_system_file_stream_reader.cc
[modify] https://crrev.com/88d45f783b7ee184994690303e7c01ace3105c45/storage/browser/file_system/memory_file_stream_reader.cc
[modify] https://crrev.com/88d45f783b7ee184994690303e7c01ace3105c45/storage/browser/file_system/memory_file_stream_reader.h
[modify] https://crrev.com/88d45f783b7ee184994690303e7c01ace3105c45/storage/browser/file_system/memory_file_stream_reader_unittest.cc
[modify] https://crrev.com/88d45f783b7ee184994690303e7c01ace3105c45/storage/browser/file_system/memory_file_stream_writer.cc
[modify] https://crrev.com/88d45f783b7ee184994690303e7c01ace3105c45/storage/browser/file_system/memory_file_stream_writer.h
[modify] https://crrev.com/88d45f783b7ee184994690303e7c01ace3105c45/storage/browser/file_system/memory_file_stream_writer_unittest.cc
[modify] https://crrev.com/88d45f783b7ee184994690303e7c01ace3105c45/storage/browser/file_system/obfuscated_file_util_memory_delegate.cc
[modify] https://crrev.com/88d45f783b7ee184994690303e7c01ace3105c45/storage/browser/file_system/obfuscated_file_util_memory_delegate.h
[modify] https://crrev.com/88d45f783b7ee184994690303e7c01ace3105c45/storage/browser/file_system/sandbox_file_stream_writer.cc

Comment 29 by rhalavati@chromium.org on Fri, Aug 28, 2020, 4:54 AM EDT Project Member
Status: Fixed (was: Started)

Abandoning crrev.com/c/2371625, crrev.com/c/2339343, and the effort to make all delegates in ObfuscatedFileUtil scoped_ref.

Comment 30 by bugdroid on Fri, Aug 28, 2020, 12:54 PM EDT Project Member
The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src.git/+/ce1b93429a4946d330d02d920821dd4e7f7da034

commit ce1b93429a4946d330d02d920821dd4e7f7da034
Author: Marijn Kruisselbrink <mek@chromium.org>
Date: Fri Aug 28 16:52:35 2020

Revert "Run ObfuscatedFileUtilMemoryDelegate entirely on TaskRunner."

This reverts commit 88d45f783b7ee184994690303e7c01ace3105c45.

Reason for revert: Unfortunately this seems to be causing frequent flaky crashes in ECKIncognitoEncryptedMediaTest.FileIO

https://ci.chromium.org/p/chromium/builders/ci/Linux%20Ozone%20Tester%20%28X11%29/16596

```
BrowserTestBase received signal: Segmentation fault. Backtrace:
#0 0x5556b3a3edb9 base::debug::CollectStackTrace()
#1 0x5556b39b4743 base::debug::StackTrace::StackTrace()
#2 0x5556b3f82016 content::(anonymous namespace)::DumpStackTraceSignalHandler()
#3 0x7f4df599a4c0 (/lib/x86_64-linux-gnu/libc-2.23.so+0x354bf)
#4 0x7f4df5ab3095 (/lib/x86_64-linux-gnu/libc-2.23.so+0x14e094)
#5 0x5556b130ff73
_ZNSt3__16vectorIhNS_9allocatorIhEEE6insertIPKcEENS_9enable_ifIXaasr21__is_forward_iteratorIT_EE5valuesr16is_constructibleIhNS_15iterator_traitsIS8_E9referenc
eEEE5valueENS_11__wrap_iterIPhEEE4typeENSC_IPKhEES8_S8_
#6 0x5556b5ed32e7 storage::ObfuscatedFileUtilMemoryDelegate::WriteFile()
#7 0x5556b5ee0cdf base::internal::Invoker<>::RunOnce()
...
```

Original change's description:
> Run ObfuscatedFileUtilMemoryDelegate entirely on TaskRunner.
>
> MemoryFileStreamWriter called some ObfuscatedFileUtilMemoryDelegate
> functions through IO thread while other functions in OFUMD are called
> on a threadpool sequence. This could result in races in updating
> directory structure.
>
> To fix the issue, MemoryFileStreamWriter and MemoryFileStreamReader are
> updated to call all OFUMD on the default task runner of the file system
> context.
>
> ~~Bug: 1100136~~
> Change-Id: I59146ca690eee810c52f807bd1fb4ef2b1f2c929
> Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2308721
> Commit-Queue: Ramin Halavati <rhalavati@chromium.org>
> Reviewed-by: Marijn Kruisselbrink <mek@chromium.org>
> Cr-Commit-Position: refs/heads/master@{#802584}

TBR=mek@chromium.org,rhalavati@chromium.org

Change-Id: Ib856bac5a978b8da33e74f8646ddd5be2a285865
No-Presubmit: true
No-Tree-Checks: true
No-Try: true
~~Bug: 1100136~~
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2382051
Reviewed-by: Marijn Kruisselbrink <mek@chromium.org>
Commit-Queue: Marijn Kruisselbrink <mek@chromium.org>
Cr-Commit-Position: refs/heads/master@{#802687}

[modify] https://crrev.com/ce1b93429a4946d330d02d920821dd4e7f7da034/storage/browser/file_system/file_stream_reader.h
[modify] https://crrev.com/ce1b93429a4946d330d02d920821dd4e7f7da034/storage/browser/file_system/file_stream_test_utils.cc
[modify] https://crrev.com/ce1b93429a4946d330d02d920821dd4e7f7da034/storage/browser/file_system/file_stream_test_utils.h
[modify] https://crrev.com/ce1b93429a4946d330d02d920821dd4e7f7da034/storage/browser/file_system/file_stream_writer.h
[modify] https://crrev.com/ce1b93429a4946d330d02d920821dd4e7f7da034/storage/browser/file_system/file_system_file_stream_reader.cc
[modify] https://crrev.com/ce1b93429a4946d330d02d920821dd4e7f7da034/storage/browser/file_system/memory_file_stream_reader.cc
[modify] https://crrev.com/ce1b93429a4946d330d02d920821dd4e7f7da034/storage/browser/file_system/memory_file_stream_reader.h
[modify] https://crrev.com/ce1b93429a4946d330d02d920821dd4e7f7da034/storage/browser/file_system/memory_file_stream_reader_unittest.cc

[modify] https://crrev.com/ce1b93429a4946d330d02d920821dd4e7f7da034/storage/browser/file_system/memory_file_stream_writer.cc
[modify] https://crrev.com/ce1b93429a4946d330d02d920821dd4e7f7da034/storage/browser/file_system/memory_file_stream_writer.h
[modify] https://crrev.com/ce1b93429a4946d330d02d920821dd4e7f7da034/storage/browser/file_system/memory_file_stream_writer_unittest.cc
[modify] https://crrev.com/ce1b93429a4946d330d02d920821dd4e7f7da034/storage/browser/file_system/obfuscated_file_util_memory_delegate.cc
[modify] https://crrev.com/ce1b93429a4946d330d02d920821dd4e7f7da034/storage/browser/file_system/obfuscated_file_util_memory_delegate.h
[modify] https://crrev.com/ce1b93429a4946d330d02d920821dd4e7f7da034/storage/browser/file_system/sandbox_file_stream_writer.cc

Comment 31 by sheriffbot on Fri, Aug 28, 2020, 3:10 PM EDT Project Member

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

Comment 32 by sheriffbot on Fri, Aug 28, 2020, 3:30 PM EDT Project Member

Labels: Merge-Request-85

Requesting merge to stable M85 because latest trunk commit (802584) appears to be after stable branch point (782793).

Requesting merge to beta M85 because latest trunk commit (802584) appears to be after beta branch point (782793).

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

Comment 33 by sheriffbot on Fri, Aug 28, 2020, 3:31 PM EDT Project Member

Labels: -Merge-Request-85 Merge-Review-85 Hotlist-Merge-Review

This bug requires manual review: Request affecting a post-stable build
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 master/ToT?
4. Why are these changes required in this milestone after branch?
5. Is this a new feature?
6. If it is a new feature, is it behind a flag using finch?

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

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

Comment 34 by mek@chromium.org on Fri, Aug 28, 2020, 3:47 PM EDT Project Member

Status: Assigned (was: Fixed)

Unfortunately had to revert the fix since it broke some plugin private filesystem tests (which share the same file system backend). Not sure what is going on there though...

Comment 35 by sheriffbot on Sat, Aug 29, 2020, 3:13 PM EDT Project Member

Labels: Deadline-Exceeded

We commit ourselves to a 60 day deadline for fixing for high severity vulnerabilities, and have exceeded it here. If you're unable to look into this soon, could you please find another owner or remove yourself so that this gets back into the security triage queue?

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

Comment 36 by rhalavati@chromium.org on Tue, Sep 1, 2020, 3:24 AM EDT Project Member

Status: Started (was: Assigned)

Comment 37 by srinivassista@google.com on Tue, Sep 1, 2020, 2:32 PM EDT Project Member

per comment #34, looks like this is not ready for merge for M85 yet, pls confirm and remove the merge-request-85 label.

Comment 38 by adetaylor@google.com on Tue, Sep 1, 2020, 3:31 PM EDT Project Member

Labels: -Merge-Review-85

Comment 39 by rhalavati@chromium.org on Wed, Sep 2, 2020, 7:27 AM EDT Project Member

Update:
Still trying to reproduce the issue on Linux Ozone Tester (X11).

Tried to reproduce it locally and failed with the following gn args:

```
is_component_build = false
is_debug = false
ozone_platform = "headless"
use_bundled_weston = true
use_goma = true
use_ozone = true
```

and test config:

```
out/ozone/browser_tests --test-launcher-bot-mode --cfi-diag=0 --ozone-platform=x11 --enable-features=UseOzonePlatform
```

Now trying to run the trybot on gerrit.

Comment 40 by mek@chromium.org on Wed, Sep 2, 2020, 1:11 PM EDT Project Member

I don't think the test failures were related to ozone, that just happened to be the first failing build I came across. Per https://analysis.chromium.org/p/chromium/flake-portal/flakes/occurrences?key=ag9zfmZpbmRpdC1mb3ItbWVyRwsSBUZsYWtlIjxjaHJvbWl1bUBicm93c2VyX3Rlc3RzQEVDS0luY29nbml0b0VuY3J5cHRlZE1lZGlhVGVzdC5GaWxlSU8M it seems it was flakily failing on a number of different linux and windows configurations.

Unfortunately I haven't been able to reproduce it locally yet either...

Comment 41 by mek@chromium.org on Wed, Sep 2, 2020, 1:17 PM EDT Project Member

I wonder if the problem is the lifetime of the net::IOBuffer that was passed to MemoryFileStreamWriter::Write. Other FileStreamWriter implementations pass that as a scoped_refptr<net::IOBuffer> to any deferred tasks, making sure to keep it alive. But currently MemoryFileStreamWriter just keeps it as a raw pointer.

(of course it shouldn't have been a raw pointer in the FileStreamWriter interface to begin with, but legacy code and all that... it's how we used to write code...)

Comment 42 by mek@chromium.org on Wed, Sep 2, 2020, 1:21 PM EDT Project Member

(and if memory_file_stream_reader/writer.cc had had a #include "net/base/io_buffer.h", this would have been obvious, as the code wouldn't have even compiled because base::Bind doesn't let you bind ref counted objects via raw pointers...)

Comment 43 by rhalavati@chromium.org on Thu, Sep 3, 2020, 9:59 AM EDT Project Member

Thanks mek@, great hints.
Started crrev.com/c/2390641

Comment 44 by bugdroid on Wed, Sep 9, 2020, 1:11 AM EDT Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src.git/+/0e61c69ebd476e5b688f341f8d0bf69fe814c515

commit 0e61c69ebd476e5b688f341f8d0bf69fe814c515
Author: Ramin Halavati <rhalavati@chromium.org>
Date: Wed Sep 09 05:10:19 2020

Reland Run ObfuscatedFileUtilMemoryDelegate entirely on TaskRunner.

MemoryFileStreamWriter called some ObfuscatedFileUtilMemoryDelegate functions through IO thread while other functions in OFUMD are called on a threadpool sequence. This could result in races in updating directory structure.

To fix the issue, MemoryFileStreamWriter and MemoryFileStreamReader are updated to call all OFUMD on the default task runner of the file system context.

This CL was landed in crrev.com/c/2308721 and reverted due to flakiness. The flaky crashes are believed to be because the buffer passed to MemoryFileStreamReader::Read and MemoryFileStreamWrite::Write are not thread safe.

Patchset1 is a copy of the previous CL and the issue is fixed in the next patchsets.

~~Bug: 1100136~~
Change-Id: I619b82c2f4d23a020e9ce7e5e6c16980907b501b
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2398701
Reviewed-by: Marijn Kruisselbrink <mek@chromium.org>
Commit-Queue: Ramin Halavati <rhalavati@chromium.org>
Cr-Commit-Position: refs/heads/master@{#805198}

[modify] https://crrev.com/0e61c69ebd476e5b688f341f8d0bf69fe814c515/storage/browser/file_system/file_stream_reader.h
[modify] https://crrev.com/0e61c69ebd476e5b688f341f8d0bf69fe814c515/storage/browser/file_system/file_stream_test_utils.cc
[modify] https://crrev.com/0e61c69ebd476e5b688f341f8d0bf69fe814c515/storage/browser/file_system/file_stream_test_utils.h
[modify] https://crrev.com/0e61c69ebd476e5b688f341f8d0bf69fe814c515/storage/browser/file_system/file_stream_writer.h
[modify] https://crrev.com/0e61c69ebd476e5b688f341f8d0bf69fe814c515/storage/browser/file_system/file_system_file_stream_reader.cc
[modify] https://crrev.com/0e61c69ebd476e5b688f341f8d0bf69fe814c515/storage/browser/file_system/memory_file_stream_reader.cc
[modify] https://crrev.com/0e61c69ebd476e5b688f341f8d0bf69fe814c515/storage/browser/file_system/memory_file_stream_reader.h
[modify] https://crrev.com/0e61c69ebd476e5b688f341f8d0bf69fe814c515/storage/browser/file_system/memory_file_stream_reader_unittest.cc
[modify] https://crrev.com/0e61c69ebd476e5b688f341f8d0bf69fe814c515/storage/browser/file_system/memory_file_stream_writer.cc
[modify] https://crrev.com/0e61c69ebd476e5b688f341f8d0bf69fe814c515/storage/browser/file_system/memory_file_stream_writer.h
[modify] https://crrev.com/0e61c69ebd476e5b688f341f8d0bf69fe814c515/storage/browser/file_system/memory_file_stream_writer_unittest.cc
[modify] https://crrev.com/0e61c69ebd476e5b688f341f8d0bf69fe814c515/storage/browser/file_system/obfuscated_file_util_memory_delegate.cc
[modify] https://crrev.com/0e61c69ebd476e5b688f341f8d0bf69fe814c515/storage/browser/file_system/obfuscated_file_util_memory_delegate.h
[modify] https://crrev.com/0e61c69ebd476e5b688f341f8d0bf69fe814c515/storage/browser/file_system/sandbox_file_stream_writer.cc

Comment 45 by rhalavati@chromium.org on Thu, Sep 10, 2020, 2:03 AM EDT Project Member
Status: Fixed (was: Started)

There seem to be no more flakiness.
Marking as fixed again.

Comment 46 by adetaylor@chromium.org on Thu, Sep 10, 2020, 11:49 AM EDT Project Member
Labels: Merge-Request-85 Merge-Request-86

mek@ normally we'd merge a high severity security bug back to stable (i.e. M85) and per #c8 this is nudging Critical severity. Sheriffbot would shortly add merge requests but I'll short-circuit the process.

It's obviously a fairly complex fix. How confident are you? Can we merge to M86 now? (Unfortunately we just missed a beta). And then aim to merge to M85 in about a week, in order to get it into the final M85 stable security refresh which will be in about ~10 days?

Comment 47 by sheriffbot on Thu, Sep 10, 2020, 11:50 AM EDT Project Member
Labels: -Merge-Request-86 Merge-Review-86

This bug requires manual review: Reverts referenced in bugdroid comments after merge request.
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 master/ToT?
4. Why are these changes required in this milestone after branch?
5. Is this a new feature?
6. If it is a new feature, is it behind a flag using finch?

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

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

Comment 48 by mek@chromium.org on Thu, Sep 10, 2020, 12:03 PM EDT Project Member
I'm reasonably confident about this fix. So merging to M86 now and M85 in a week sounds reasonable to me.

Comment 49 by adetaylor@chromium.org on Thu, Sep 10, 2020, 12:24 PM EDT Project Member
Labels: -Merge-Review-86 Merge-Approved-86

OK, thanks. In that case, approving merge to M86, branch 4240.

I usually do a trawl for candidate fixes shortly before we make a new stable refresh, and I expect to approve this bug for M85 at that time, probably in about a week. Do let us know if any hiccups occur before then.

Comment 50 by bugdroid on Fri, Sep 11, 2020, 1:50 AM EDT Project Member
Labels: -merge-approved-86 merge-merged-4240 merge-merged-86

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src.git/+/f076f21125c727093158ce0076606ea7ed5b9803

commit f076f21125c727093158ce0076606ea7ed5b9803
Author: Ramin Halavati <rhalavati@chromium.org>
Date: Fri Sep 11 05:50:08 2020

Reland Run ObfuscatedFileUtilMemoryDelegate entirely on TaskRunner.

MemoryFileStreamWriter called some ObfuscatedFileUtilMemoryDelegate
functions through IO thread while other functions in OFUMD are called
on a threadpool sequence. This could result in races in updating
directory structure.

To fix the issue, MemoryFileStreamWriter and MemoryFileStreamReader are
updated to call all OFUMD on the default task runner of the file system
context.

This CL was landed in crrev.com/c/2308721 and reverted due to flakiness.
The flaky crashes are believed to be because the buffer passed to
MemoryFileStreamReader::Read and MemoryFileStreamWrite::Write are not
thread safe.

Patchset1 is a copy of the previous CL and the issue is fixed in the
next patchsets.

TBR:mek@chromium.org

(cherry picked from commit 0e61c69ebd476e5b688f341f8d0bf69fe814c515)

~~Bug: 1100136~~
Change-Id: I619b82c2f4d23a020e9ce7e5e6c16980907b501b
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2398701
Reviewed-by: Marijn Kruisselbrink <mek@chromium.org>
Commit-Queue: Ramin Halavati <rhalavati@chromium.org>
Cr-Original-Commit-Position: refs/heads/master@{#805198}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2404845
Reviewed-by: Ramin Halavati <rhalavati@chromium.org>
Cr-Commit-Position: refs/branch-heads/4240@{#604}
Cr-Branched-From: f297677702651916bbf65e59c0d4bbd4ce57d1ee-refs/heads/master@{#800218}

[modify] https://crrev.com/f076f21125c727093158ce0076606ea7ed5b9803/storage/browser/file_system/file_stream_reader.h
[modify] https://crrev.com/f076f21125c727093158ce0076606ea7ed5b9803/storage/browser/file_system/file_stream_test_utils.cc
[modify] https://crrev.com/f076f21125c727093158ce0076606ea7ed5b9803/storage/browser/file_system/file_stream_test_utils.h
[modify] https://crrev.com/f076f21125c727093158ce0076606ea7ed5b9803/storage/browser/file_system/file_stream_writer.h
[modify] https://crrev.com/f076f21125c727093158ce0076606ea7ed5b9803/storage/browser/file_system/file_system_file_stream_reader.cc
[modify] https://crrev.com/f076f21125c727093158ce0076606ea7ed5b9803/storage/browser/file_system/memory_file_stream_reader.cc
[modify] https://crrev.com/f076f21125c727093158ce0076606ea7ed5b9803/storage/browser/file_system/memory_file_stream_reader.h
[modify] https://crrev.com/f076f21125c727093158ce0076606ea7ed5b9803/storage/browser/file_system/memory_file_stream_reader_unittest.cc
[modify] https://crrev.com/f076f21125c727093158ce0076606ea7ed5b9803/storage/browser/file_system/memory_file_stream_writer.cc
[modify] https://crrev.com/f076f21125c727093158ce0076606ea7ed5b9803/storage/browser/file_system/memory_file_stream_writer.h
[modify] https://crrev.com/f076f21125c727093158ce0076606ea7ed5b9803/storage/browser/file_system/memory_file_stream_writer_unittest.cc
[modify] https://crrev.com/f076f21125c727093158ce0076606ea7ed5b9803/storage/browser/file_system/obfuscated_file_util_memory_delegate.cc
[modify] https://crrev.com/f076f21125c727093158ce0076606ea7ed5b9803/storage/browser/file_system/obfuscated_file_util_memory_delegate.h
[modify] https://crrev.com/f076f21125c727093158ce0076606ea7ed5b9803/storage/browser/file_system/sandbox_file_stream_writer.cc

Comment 51 by rhalavati@chromium.org on Fri, Sep 11, 2020, 1:53 AM EDT Project Member

NextAction: 2020-09-15

Check for hiccups.

Comment 52 by adetaylor@google.com on Mon, Sep 14, 2020, 2:31 PM EDT Project Member

Labels: reward-topanel

Comment 53 by adetaylor@google.com on Tue, Sep 15, 2020, 1:07 PM EDT Project Member

Labels: -Merge-Request-85 Merge-Approved-85

Approving merge to M85, branch 4183. Please merge, assuming things are looking good in Canary and no hiccups have appeared :)

Comment 54 by rhalavati@chromium.org on Wed, Sep 16, 2020, 1:38 AM EDT Project Member

I don't see any issues, merging in M-85.

Comment 55 by bugdroid on Wed, Sep 16, 2020, 6:22 AM EDT Project Member

Labels: -merge-approved-85 merge-merged-85 merge-merged-4183

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src.git/+/b28bcfdb914e090683bdfdfeedb57941a78000bc

commit b28bcfdb914e090683bdfdfeedb57941a78000bc
Author: Ramin Halavati <rhalavati@chromium.org>
Date: Wed Sep 16 10:21:07 2020

Reland Run ObfuscatedFileUtilMemoryDelegate entirely on TaskRunner.

MemoryFileStreamWriter called some ObfuscatedFileUtilMemoryDelegate
functions through IO thread while other functions in OFUMD are called
on a threadpool sequence. This could result in races in updating
directory structure.

To fix the issue, MemoryFileStreamWriter and MemoryFileStreamReader are
updated to call all OFUMD on the default task runner of the file system
context.

This CL was landed in crrev.com/c/2308721 and reverted due to flakiness.
The flaky crashes are believed to be because the buffer passed to
MemoryFileStreamReader::Read and MemoryFileStreamWrite::Write are not
thread safe.

Patchset1 is a copy of the previous CL and the issue is fixed in the
next patchsets.

TBR: mek@chromium.org

(cherry picked from commit 0e61c69ebd476e5b688f341f8d0bf69fe814c515)

~~Bug: 1100136~~
Change-Id: I619b82c2f4d23a020e9ce7e5e6c16980907b501b
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2398701
Reviewed-by: Marijn Kruisselbrink <mek@chromium.org>
Commit-Queue: Ramin Halavati <rhalavati@chromium.org>
Cr-Original-Commit-Position: refs/heads/master@{#805198}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2412335
Reviewed-by: Ramin Halavati <rhalavati@chromium.org>
Cr-Commit-Position: refs/branch-heads/4183@{#1840}
Cr-Branched-From: 740e9e8a40505392ba5c8e022a8024b3d018ca65-refs/heads/master@{#782793}

[modify] https://crrev.com/b28bcfdb914e090683bdfdfeedb57941a78000bc/storage/browser/file_system/file_stream_reader.h
[modify] https://crrev.com/b28bcfdb914e090683bdfdfeedb57941a78000bc/storage/browser/file_system/file_stream_test_utils.cc
[modify] https://crrev.com/b28bcfdb914e090683bdfdfeedb57941a78000bc/storage/browser/file_system/file_stream_test_utils.h
[modify] https://crrev.com/b28bcfdb914e090683bdfdfeedb57941a78000bc/storage/browser/file_system/file_stream_writer.h
[modify] https://crrev.com/b28bcfdb914e090683bdfdfeedb57941a78000bc/storage/browser/file_system/file_system_file_stream_reader.cc
[modify] https://crrev.com/b28bcfdb914e090683bdfdfeedb57941a78000bc/storage/browser/file_system/memory_file_stream_reader.cc
[modify] https://crrev.com/b28bcfdb914e090683bdfdfeedb57941a78000bc/storage/browser/file_system/memory_file_stream_reader.h
[modify] https://crrev.com/b28bcfdb914e090683bdfdfeedb57941a78000bc/storage/browser/file_system/memory_file_stream_reader_unittest.cc
[modify] https://crrev.com/b28bcfdb914e090683bdfdfeedb57941a78000bc/storage/browser/file_system/memory_file_stream_writer.cc
[modify] https://crrev.com/b28bcfdb914e090683bdfdfeedb57941a78000bc/storage/browser/file_system/memory_file_stream_writer.h
[modify] https://crrev.com/b28bcfdb914e090683bdfdfeedb57941a78000bc/storage/browser/file_system/memory_file_stream_writer_unittest.cc
[modify] https://crrev.com/b28bcfdb914e090683bdfdfeedb57941a78000bc/storage/browser/file_system/obfuscated_file_util_memory_delegate.cc
[modify] https://crrev.com/b28bcfdb914e090683bdfdfeedb57941a78000bc/storage/browser/file_system/obfuscated_file_util_memory_delegate.h
[modify] https://crrev.com/b28bcfdb914e090683bdfdfeedb57941a78000bc/storage/browser/file_system/sandbox_file_stream_writer.cc

Comment 56 by adetaylor@google.com on Wed, Sep 16, 2020, 7:15 PM EDT 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 57 by adetaylor@google.com on Wed, Sep 16, 2020, 7:17 PM EDT Project Member
Congratulations, the VRP panel has decided to award $15,000 for this report.

Comment 58 by adetaylor@google.com on Mon, Sep 21, 2020, 1:18 PM EDT Project Member
Labels: Release-2-M85

Comment 59 by adetaylor@google.com on Mon, Sep 21, 2020, 1:36 PM EDT Project Member
Labels: OS-Chrome OS-Fuchsia OS-Mac OS-Windows

Assuming this affects all the usual platforms.

Comment 60 by adetaylor@google.com on Thu, Sep 24, 2020, 1:36 PM EDT Project Member
Labels: -reward-unpaid reward-inprocess

Comment 61 by sheriffbot on Thu, Dec 17, 2020, 1:53 PM EST 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