Preventing Downloads in Sandboxed Iframes

This Document is Public

Authors: yaoxia@chromium.org

Last updated: October 21, 2019

One-page overview

Summary

Content providers should be able to restrict whether downloads can be initiated for content in iframes. Thus, we plan to prevent downloads initiated from sandboxed iframes, and this restriction could be lifted via an 'allow-downloads' keyword, if present in the sandbox attribute list.

Platforms

All except for iOS.

Team

chrome-ads-core@google.com

Bug

crbug/539938

Code affected

Sandbox, Download

Design

Background

How are downloads triggered in Chrome?

Downloads can be triggered in a wide variety of manners:

- Navigations to non-web-renderable content.
- Click on <a download> links.
- Users dragging-and-dropping links or images to the desktop.
- Context menus triggering downloads: “Save as…”, “Save link as…”
- Alt-Click on links.

NOTE: Those cases are not mutually exclusive. E.g. Click on <a download> link may turn into a navigation; Click on link can have both Alt modifier and download attribute, in which case the Alt modifier will suppress the download attribute; and etc.

Downloads we are targeting to prevent

In sandboxed iframe, we plan to disallow “Navigations to non-web-renderable content” and “Click on <a download> links” as those two types can be triggered without strong user intention, even though they could be triggered with user gesture. On the other hand, we are not restricting drag-and-drop, context menu, or alt-click triggered download.

We will block a download when:

1. If it’s <a download> click triggering a direct download without navigation involved:
    - The link lives in a sandboxed iframe and the tokens don’t contain the “allow-downloads” keyword.
2. If it’s a navigation becoming a download:
    - At least one of the security context among [the security context of the document initiating the navigation and the security context of the frame where navigation is happening] is sandboxed, and the tokens don’t contain the “allow-downloads” keyword.

Please see the pull request that defines the behavior using more whatwg/html terms.

Any UI change?

No. We just let the download fail silently. Developers will receive a console error.

Implementation

- *<a download>*

  No-op in *HTMLAnchorElement::HandleClick* when the frame has the download bit sandboxed and it's about to turn into a direct download instead of a navigation.

- *Navigations to non-web-renderable content*

  Add a new enum value *kSandbox* to *NavigationDownloadPolicy* which can be possibly set in *RenderFrameImpl::BeginNavigationInternal, RenderFrameImpl::OpenURL* or *RenderFrameProxy::Navigate* when the frame has the download bit sandboxed.

  pre-network-service
  NavigationDownloadPolicy will be propagated to resource requests and be translated to *ResourceInterceptPolicy::kAllowPluginOnly* in the case of *kSandbox*.

  At the time that *MimeSniffingResourceHandler::MaybeStartInterception()* decides that the resource load for the frame will be intercepted as a download, it will check the resource intercept policy associated with the request. If downloads are to be prevented, the main resource load will be aborted and the download will not initiate.

  post-network-service
  NavigationDownloadPolicy will be propaged to *NavigationRequest::OnResponseStarted()* and will set |is_download_| to false in the case of *kSandbox*. This bit will be controlling whether the download is going to happen.

Metrics

Success metrics

When the feature launches, the following use counters should drop to zero:

- DownloadInSandbox

Regression metrics

Standard heartbeat metrics, including stability metrics.

Experiments

N/A

Rollout plan

Waterfall.

Core principle considerations

Speed

There are no speed considerations. The extra computations it brings are checking some booleans at most once per click or per navigation, which is negligible.

Security

This is a security win, since downloads are a vector to vulnerabilities in lots of cases. And this doesn't introduce new security vulnerabilities, as we simply block the code path to download in some conditions.

Predictability

Compatibility
According to HTML & JavaScript usage, about 0.0028% page loads have downloads in sandbox so the compatibility risk is small.

With a brief scan on UKM query (need UKM access), none of the use cases are in particular malicious. Those legitimate use cases can be categorized as follows:

- Some sites start download by a click triggering a top frame navigation or a popup, and the new page is triggering an automatic download.
- In some video converting/downloading sites, a click will start converting the video first and download will start automatically afterwards.
- Some downloads are triggered by dynamically appending an invisible iframe with src being a download, or with script in the iframe triggering a download.

Given that the percentage of breakage is small, we should still implement the prevention as it allows the content providers to control what the embedded content can do. Besides, the "allow-downloads" attribute is always a simple solution to any breakage.

Interoperability
No official signals from other browsers. The spec discussion, pull request for spec change, and tentative WPT is going to alert the change.

Privacy considerations

None.

Testing plan

Tentative web platform tests are added.

Followup work

Clean up the feature once after the code has reached the stable channel.