Preventing Drive-By-Downloads in Ad Frames

Authors: yaoxia@chromium.org
Last updated: Feb 7, 2019

One-page overview

Summary

We plan to prevent downloads initiated from ad frames that lack a user gesture to prevent unwanted drive-by-downloads.

Platforms

All except for iOS.

Team

chrome-ads-core@google.com

Bug

crbug/929911

Code affected

Download

Design

Background

What is an ad frame?

An iframe marked as ad by the Chromium ad detection infrastructure AdTagging.

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.

Only <a download> clicks (without Alt modifier) and navigations could be automatic; other types of download will definitely need user initiated work.

Downloads we are targeting to prevent

The only kinds of downloads that can occur without a user gesture are navigations and simulated clicks on <a download> links. Therefore, our intervention will block such downloads if they occur without a user gesture.

We intend to block a download if all of the following conditions are met:

1. If it’s <a download> click triggering a direct download without navigation involved:
    - The link lives in an ad frame
    - The frame does not have a transient user gesture at the moment of the click.
2. If it’s a navigation becoming a download:
    - At least one of frames among [the frame initiating the navigation and the frame where navigation is happening] is an ad frame. (Note that the 2 frames can also be the same frame.)
    - The navigation doesn’t have a user gesture.

Any User Facing 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 is an ad frame, and the user activation bit is not set, 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 *kAdFrameNoGesture* to *NavigationDownloadPolicy* which can be possibly set in *RenderFrameImpl::BeginNavigationInternal, RenderFrameImpl::OpenURL, RenderFrameProxy::Navigate* or *RenderFrameProxyHost::OnOpenURL* when the frame is an ad frame and the transient user activation bit is not set.

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

  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 *kAdFrameNoGesture*. 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:

- DownloadInAdFrameWithoutUserGesture

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

Based on the calculation from the UMA dashboard, downloads in ad frame without user gesture (DownloadInAdFrameWithoutUserGesture) account for 0.00001% page loads so the compatibility risk is trivial.

With a quick scan on the top URLs given by the UKM query, the behavior (DownloadInAdFrameWithoutUserGesture) is unreproducible. It's not unexpected since the usage is quite low and it might just be a small sets of ads doing automatic downloads. It's very unlikely we are going to break a majority of legitimate cases, so it would be low risk to add the intervention, although we would be more confident in adding the intervention if we can identify a single illegitimate use case.

For interoperability, there's no plan to standardize this behavior so chrome will move away from other browsers.

Privacy considerations

None.

Testing plan

Browser tests should cover all the download paths affected. Web platform tests are not used as getting AdTagging to work there might be difficult. No special manual testing is needed.

Followup work

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