Position paper on Web Packaging for the ESCAPE workshop

Date: 2019-06-01

Intro

As Technical Steering Committee of the AMP Project we are excited about Web Packaging as technology to form the basis of the web platform's support of privacy-preserving preloading. We feel that introducing such an approach for loading to the web platform enables the web's continued aim of providing excellent user experiences and access to content. In a landscape of proprietary content distribution formats locked to their owner's platforms, Web Packaging enables the ability to provide web experiences consistent with users' experience expectations while ensuring full control of publishers over their content through tamper-proof digital signatures.

At the same time, we recognize that Web Packaging does allow for decentralized delivery of content, which, while allowing for exciting new use cases, also comes with inherent new risks. Thus we welcome a deep discussion of this new technology and its impact on the shape of the web.

Ensuring publisher control of their content

Context

User's expectations today are formed through the entirety of the apps they use on their phones. These expectations are independent of underlying technologies and apply to the web just like they do to the native apps installed on the device. For the web to stay relevant, it has to work towards matching the experiences provided by walled-garden native or web apps that tightly control UX and monetization of content presented in them, while not sacrificing what makes the web the greatest computing platform of all time.

From the earliest days, the AMP Project aimed to enable the user experience benefits of content being embedded into content consumption apps while affording control of the publisher over their monetization (ads, paywalls, direct payments), analytics, design, branding, and navigation. The workshop invitation mentions multiple content embedding formats, but we want to note that AMP materially differs from them by not mandating or even putting a preference on monetization, measurement, or similar technology from any particular vendor. Similarly, AMP allows for implementing full website experiences within the embedded context, so that there are no limitations on publishers in guiding users into onward journeys outside of the embedded context. This, again, stands in deep contrast to other technologies mentioned in the workshop proposal.

Web Packaging

We are excited about Web Packaging technology because it allows AMP to maintain its model for privacy-preserving prefetching, and thus instant-loading, while ensuring that content can be rendered under the publisher's own authority and cannot be modified by the distributing cache, as guaranteed by cryptographic signatures.

Web Packaging puts publishers in control by ensuring that no intermediate party who participates in content distribution can in any way change publishers' content.

Combined with AMP's freedom in monetization, analytics, design, branding, navigation, etc. Web Packaging ensures that publishers can participate in embedded content experiences without having to worry about losing control of their content.

Additionally, we'd like to highlight that Web Packaging is an "initial delivery" method. Subsequent requests, such as those that are made after the package is loaded are directed to the respective origin and are not controlled by the party that did the initial delivery.

Stronger Origin Control with Signed Exchanges

Currently, deploying TLS while minimizing round trip time (RTT) requires both a CDN and an extension of trust (i.e. TLS private key or signing access) to every POP in the CDN. Even with good key management practices, every POP necessarily represents attack surface for malicious content alteration. In common origin-pull CDN configurations, such risk is unnecessary. With signed exchanges (SXG), Web Packaging can decouple integrity control from distribution to reduce this risk. Such decoupling can simultaneously support the high performance of wide distribution while retaining a minimal attack surface against integrity. Systems that already use offline signing with mirrors (e.g. RPM/Apt) demonstrate the value and viability of separating the concerns of integrity from distribution.

Content management and packaging approaches like Jekyll, Hugo, Hexo, Middleman, and webpack (among many others) would particularly benefit from the offline signing model supported by SXG because they already pre-generate their assets in a trusted build process. This build process provides a natural integration point for offline signing. Integrating SXG into a content build process -- rather than relying only on TLS -- would reduce the attack surface against document integrity from the entire web stack (including CDN POPs) to merely the asset build process.

Combining SXG with TLS maximizes both integrity control (via SXG) and privacy (via TLS). Such a model offers strictly better security -- in all respects -- than TLS alone whenever it is viable.

Privacy-preserving preloading with Web Packaging

AMP implements privacy-preserving preloading, a technique that allows content to be loaded before a user expresses their explicit interest to view the content and without the content publisher learning about the user's potential interest. As soon as the user expresses their explicit interest in the content (e.g. by navigating to it) the publisher learns about the user in the same way the publisher would have learned about the user without preloading.

AMP's current implementation of privacy-preserving preloading is done on the application layer, relying on a carefully designed interplay between web components and AMP Caches. Web Packaging greatly simplifies this model to rely entirely on web platform capabilities. This simplifies AMP, but it also makes the same privacy-preserving preloading available to content not written in AMP.

Vendor lock-in

AMP was designed with open distribution of content in mind. While the publisher gets to control monetization, analytics, design, branding, and navigation, AMP does not provide for a method to control the platform that content can be shown in–just like the web doesn't provide for a mechanism to control who can link to a web page.

This was done to avoid publishers specializing their content for any particular company, such as Google, and it was successful in allowing a multitude of platforms to launch AMP integrations without each needing to contact every publisher to update their documents to be compatible with their platform.

The current state of the Web Packaging spec is written in similar spirit. The publisher gets cryptographic-signature guaranteed integrity of their content, but they don't get control who may deliver it on their behalf. There are proposals to change this which in turn address security issues that may arise through unintended distribution. It may well be inevitable to make such compromises, but the AMP Project would prefer a future of open content distribution over a system of per-platform opt-in.

What if we don't act

We feel that Web Packaging provides for a good trade-off in making the web competitive with native- or webview-based implementations of similar patterns that don't need to work within the web platform, while keeping the publisher in full control of their content.

Failing to make the web competitive would likely result in these use cases moving away from the web, as has already happened with content distributions formats mentioned in the workshop description other than AMP. We believe this would deteriorate publisher control and increase platform lock-in.

The AMP Technical Steering Committee

Chris Papazian, Pinterest
David Strauss, Pantheon
Dima Voytenko, Google
Malte Ubl, Google
Paul Armstrong, Twitter
Rudy Galfi, Google
Saulo Santos, Microsoft