Supporting Web Archiving via Web Packaging

Sawood Alam[1], Michele C Weigle[1], Michael L Nelson[1], Martin Klein[2], and Herbert Van de Sompel[3]
[1]Department of Computer Science, Old Dominion University, Norfolk, Virginia, USA
{salam,mweigle,mln}@cs.odu.edu
[2]Los Alamos National Laboratory, New Mexico, USA
mklein@lanl.gov
[3]Data Archiving and Networked Services, Netherlands
herbert.van.de.sompel@dans.knaw.nl

ABSTRACT

We describe challenges related to web archiving, replaying archived web resources, and verifying their authenticity. We show that *Web Packaging* has significant potential to help address these challenges and identify areas in which changes are needed in order to fully realize that potential.

1 INTRODUCTION

Web archiving is the practice of preserving representations of web resources to enable replaying them in the future as accurately as they were at the time of capture. A web archive can be seen as a datetime indexed caching proxy server that preserves every transaction (including responses that traditional proxy servers would be asked not to cache) indefinitely, allowing future replay. Depending on the capabilities of available tools, computing and storage resources, curatorial workforce, intended use cases, and objectives, a web archive may choose to preserve complete *HTTP Request* and *Response* messages, *DNS* resolutions, *TLS* exchanges, and other provenance metadata for each observation. Two standards are widely used in web archiving: *WARC files* [3, 15] for preservation of observations and the *Memento protocol* [24] for datetime content negotiation at replay. Most web archives preserve these resources in the well-established *ISO* standard *WARC* file format, which is a container file for an arbitrary number of records and metadata. *WARC* is somewhat like *tarball*, but for web archiving, in which individual *HTTP* transactions and other record types are prefixed with *HTTP*-like *WARC headers* for additional metadata and record length (for framing).

The *Memento* protocol specifies time-based content negotiation so that a user-agent can request a representation of an original *resource URI* (or *URI-R*) at or close to a given time through intermediation of a *TimeGate resource* (or *URI-G*). This past version can be from a version-aware origin server itself (e.g., *Git* and *Wiki*) or an observation recorded by a third party web archive (e.g., archive.org, archive.is, and perma.cc). A *memento* is a timestamped archived version of a resource representation that can be retrieved from a *memento URI* (or *URI-M*). A *composite memento* is an archived web page along with all its page requisites observed in a small temporal window close to the primary web page that are necessary for its proper rendering and meaningful interactions.

Web Packaging is an emerging standard [30] that enables content aggregators and distributors to deliver related groups of resources from various origins to user-agents in the form of a package on behalf of publishers. It replaces a prior work called *Packaging on the Web* [23]. Currently, its specification is split in three different modular layers, namely *Signing* [29], *Bundling* [27], and *Loading* [28]. The *Signed HTTP Exchanges* specification allows an origin to digitally sign one or more *HTTP Exchanges* (an *HTTP Exchange* is a pair of an *HTTP Request* and corresponding *HTTP Response*) so that they can be distributed on behalf of the origin by intermediaries while maintaining the authenticity of the content. The *Bundled HTTP Exchanges* specification describes how a group of one or more signed or unsigned *HTTP Exchanges* from one or more origins can be bundled together for distribution. The *Loading Signed Exchanges* specification describes a set of algorithms to check whether a signature on an exchange is valid. All three layers of *Web Packaging* have the potential to play a significant role in web archiving. They can help crawl resources more effectively, replay more accurately, and facilitate fixity and non-repudiation on archived resources. However, realizing that potential requires some changes to the specification as it stands. These changes include support for the *Memento* protocol and long-lived trust of signed exchanges. We urge the *Web Packaging* community to consider how it can help to archive the web.

2 WEB PACKAGING IN WEB ARCHIVING

Despite a wealth of activities, internationally, related to web archiving that started more than two decades ago, mainstream web systems and protocols have put insufficient emphasis on the need to be able to preserve web resources and access those preserved resources in the future. The focus of technical advancements is on speed, efficiency, user experience, and security, but lacks in consideration of archivability and access to archived resources as a significant aspect. This is disconcerting from a societal perspective, because without an archivable and archived web, revisiting the history of our era will be all but impossible [9]. When the *Web Packaging* specification was announced, the web archiving community took notice [16] with the hope that it might help to mainstream web archiving. Some others from web publishing and personal archiving backgrounds were also interested in utilizing *Web Packaging* as a means to preserve content in an immutable manner with built-in long-term trust [14]. As a result an archival use case [25] was added, but with the requirement of the content being unsigned [26] to avoid expired signatures. The remainder of this section describes the benefits *Web Packaging* can bring to web archiving by facilitating effective crawling, accurate replay, and non-repudiation.

2.1 Effective Crawling With Bundled HTTP Exchanges

With the proliferation of *JavaScript* on the web it has become increasingly difficult to crawl web resources of a domain effectively

and completely using traditional crawlers like *Heritrix* [21]. Resources that do not appear in the plain *HTML* or *CSS* and are fetched only after client-side rendering, and possibly after a user interaction, are often not preserved [11]. Large-scale crawlers maintain a frontier queue using data structures like priority queue and a set of recently seen *URIs*. This means some page requisites may be captured long after their parent pages and by then their state might have changed. While there exist headless browser-based crawlers (e.g., *Brozzler*[1] and *Squidwarc*[2]), they are an order of magnitude or two slower than static crawlers. ***Bundled HTTP Exchanges* can be helpful in this case by serving a complete set of temporally coherent resources and saving the crawler from parsing a great deal of markup, assuming the server knows about all the requisites and bundles them effectively.**

2.2 Coherent Replay With Bundled HTTP Exchanges

Web archives serve *mementos* on behalf of a different origin while the pages were designed with the original domain in mind. This poses many difficulties in replaying a *composite memento* correctly such as live-leakage [10], temporal violations [1], origin violations [7], cookie violations [4], and broken links; all of which may yield a rendition of a page that never existed on the live web (e.g., a weather page saying sunny, but showing a rainy satellite image) [2] and some may pose security risks [20]. Archival replay systems often perform extensive *URL* rewriting to ensure that the subsequent page requisite requests are routed to the archives and not the live site or an invalid location. *URLs* that are generated by *JavaScript* are difficult to identify and rewrite, resulting in broken *composite mementos*. Proxy or browser extension-based solutions exist to mitigate this, but they do not work out of the box and require users to configure their browsers. Some replay systems use client-side rewriting (e.g., *Wombat* [18]) or *Service Worker*-based rerouting (e.g., *Reconstructive* [5, 17]) to ensure that requests maintain the desired origin boundary. The *UK Web Archive* limits its replay to certain whitelisted sites that adhere to and advertise certain usage policies, and otherwise returns an *HTTP 451* status code [8]. It whitelists some domains like `twitter.com`, but fails to recognize its *CDNs*, resulting in broken pages.

We envision a future in which web archives would have preserved signed or unsigned *Bundled HTTP Exchanges* related to a requested *composite memento* or could effectively identify all the resources needed (from one or more origins) for it to bundle them all in a single unsigned package with appropriate origin boundaries. This means the user-agent would not need to resolve for any resources on the live web to render the *composite memento*, thus avoiding many of the issues listed above. This also means that archival replay systems would not need aggressive *URL* rewriting and could serve originally preserved bytes on behalf of respective origins (except a few places where some rewriting might be inevitable). We believe that effective use of *Bundled HTTP Exchanges* can eventually solve many archival replay problems, resulting in temporally coherent and accurate *composite mementos*.

[1] https://github.com/internetarchive/brozzler
[2] https://github.com/N0taN3rd/Squidwarc

It is worth noting that *mementos* served from a web archive are "a representation of a resource at a *URI* **as observed at a given time in the past**" instead of "a representation of a resource at a *URI*", hence there might be many timestamped versions of the same resource in *Bundles* and *HTTP cache*. Currently, *Loading Signed Exchanges* favor the most recent version from a *stashed exchange* or *HTTP cache*, but in an archival context every version is equally as important. *Memento* compliant web archives resolve to a specific version of a resource when the *TimeGate* associated with that resources receives a request with an `Accept-Datetime` header. The returned *memento* has a *Memento-Datetime* header to express the time when it was archived, as well as links pertaining to datetime negotiation in the `Link` header. However, these additional headers and content negotiations are provisioned by an archival replay server and are not part of the original request and response (unless a web server is itself *Memento* compliant). In case of *Signed Exchanges*, altering messages on the server side is not possible, hence any time-based content negotiation needs to be done on the client-side after the signature validation. Alternatively, a *Memento-Datetime* header can be returned with the *Bundle* that can be used to namespace a cache and a special header to indicate resource resolution policy that tells the user-agent to not resolve a request if it is not present in the namespaced cache. Such namespaced caches have an added advantage of creating a security boundary to limit some downgrade attacks if their access is tied to the origin of the bundle distributor. **This means, to leverage *Web Packaging* in web archiving to its full potential, *Loading Signed Exchanges* and *Fetch* algorithms need to be extended to support time-based content negotiation (i.e., built-in *TimeGate* support within the *Bundle*) for versioned resources to ensure resolution of the correct and temporally coherent version of resources.**

2.3 Fixity and Non-repudiation With Signed Exchanges

In order to use web archives in a legal environment it is essential to be able to prove that a *memento* in question was not forged or altered (i.e., maintain *fixity*) beyond what is necessary for proper replay and the content was indeed produced by the said origin (i.e., establish *non-repudiation*). Due to the lack of technical means of proving *fixity* and *non-repudiation* of *mementos*, currently archive personnel has to certify *fixity* when necessary (e.g., the case of Joy-Ann Reid claiming that copies of her blog in the Internet Archive has been hacked [12, 22]).

Examining fixity of *mementos* is difficult and often impossible in *JavaScript*-rich *composite mementos* due to inconsistencies in successive replays. There are archival fixity proposals using web archives themselves [6] or a *Blockchain* [13], but they require ahead of time content digest advertisement and additional overhead of resources. Moreover, these approaches can only track the fixity of a resource as advertised by a web archive, which can be different from what the origin of the resources has originally returned (i.e., lack of *non-repudiation*). In the event of an *HTTPS* communication it sounds plausible that if original encrypted bits along with the complete *TLS* handshake log were preserved, an archive should be able to establish *non-repudiation*, but it can not, because *HTTPS* traffic is encrypted using a shared key not an asymmetric one.

Consequently, while the archive itself can rest assured the response indeed came from the said origin, it has the ability to fake the *TLS* handshake log, hence cannot prove the origin to anyone else.

This is where *Signed Exchanges* can have an impact, but unfortunately, the trust of such signatures is short-lived, which is not suitable for archival time scale. We see great technical potential in *Web Packaging* of being helpful for web archiving if we can build a long-lasting history-aware temporal signature validation model. By this we mean, rather than a digital signature being either "*valid*" or "*invalid*", introduce another state "*temporally valid*", that indicates that the signature would have been valid at a given time in the past. The *Memento* framework already provides a standard means to express that a resource representation is historical, not live, using the `Memento-Datetime` header. Using this, a user-agent would know that it needs to validate the signature in a temporal context and acknowledge the state visually (e.g., web browsers showing the state of a certificate in the address bar) or by some other means as suitable. Fortunately, due to the rapid adoption of the tamper-proof and publicly auditable *Certificate Transparency* [19] by many certificate authorities, it seems possible to build a temporally-aware digital signature trust model. It is worth noting that once a *private key* is compromised, corresponding historical signatures will become "*invalid*" too, because one can create back-dated fake records and sign them with the stolen key.

We feel that with the current short-lived validity of *Signed Exchanges*, the *Web Packaging* favors aggregators that are interested in the recent and live web (e.g., search engines, social media, and *CDNs*), but hurts many culturally and historically important systems that require a trust system (both online and offline) that lasts long after the origins of the resources are gone (e.g., web archives, digital libraries, book readers, data sharing systems, and publications). We do not think that this is intentional, rather a consequence of how our existing digital signature system on the web works. **Hence, we believe that a temporal certificate validation extension would make *Web Packaging* more inclusive and welcoming for entities at the lower end of the power graph.**

3 CONCLUSIONS

We believe that the web archiving community is generally receptive of *Web Packaging*, but would welcome changes that further increase its potential for web archiving and web archive access. Web archiving is becoming increasingly challenging due to the rapid evolution of web technologies, but *Web Packaging* can ameliorate some of those challenges with changes along the lines we described. The *Web Packaging* community has a unique opportunity to devise a technology that supports web archiving and provides a much needed capability to verify the integrity of archived web resources. We hope it will decide to embrace this opportunity and we express our willingness to collaborate to make this happen.

4 ACKNOWLEDGEMENTS

This work is supported in part by the Andrew W. Mellon Foundation (AMF) grant 11600663.

REFERENCES

[1] Scott Ainsworth, Michael L. Nelson, and Herbert Van de Sompel. 2014. A Framework for Evaluation of Composite Memento Temporal Coherence. arXiv:1402.0928 (2014). http://arxiv.org/abs/1402.0928

[2] Scott G. Ainsworth, Michael L. Nelson, and Herbert Van de Sompel. 2015. Only One Out of Five Archived Web Pages Existed as Presented. In *Proceedings of the 26th ACM Conference on Hypertext & Social Media, HT 2015*. 257–266. https://doi.org/10.1145/2700171.2791044

[3] Sawood Alam. 2018. Web ARChive (WARC) File Format. https://www.slideshare.net/ibnesayeed/web-archive-warc-file-format.

[4] Sawood Alam. 2019. Cookie Violations Cause Archived Twitter Pages to Simultaneously Replay in Multiple Languages. https://ws-dl.blogspot.com/2019/03/2019-03-18-cookie-violations-cause.html.

[5] Sawood Alam, Mat Kelly, Michele Weigle, and Michael L. Nelson. 2017. Client-side Reconstruction of Composite Mementos Using ServiceWorker. In *Proceedings of the 17th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '17)*. 237–240. https://doi.org/10.1109/JCDL.2017.7991579

[6] Mohamed Aturban, Sawood Alam, Michael L. Nelson, and Michele C. Weigle. 2019. Archive Assisted Archival Fixity Verification Framework. In *Proceedings of the 19th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '19)*. 162–171. https://doi.org/10.1109/JCDL.2019.00032

[7] John Berlin. 2017. CNN.com has been unarchivable since November 1st, 2016. https://ws-dl.blogspot.com/2017/01/2017-01-20-cnncom-has-been-unarchivable.html.

[8] Tim Bray. 2016. An HTTP Status Code to Report Legal Obstacles, Internet RFC 7725. https://tools.ietf.org/html/rfc7725.

[9] Niels Brügger and Ian Milligan. 2018. *The SAGE Handbook of Web History*. SAGE Publications Limited.

[10] Justin F. Brunelle. 2012. Zombies in the Archives. https://ws-dl.blogspot.com/2012/10/2012-10-10-zombies-in-archives.html.

[11] Justin F. Brunelle, Michele C. Weigle, and Michael L. Nelson. 2015. Archiving Deferred Representations Using a Two-Tiered Crawling Approach. In *Proceedings of iPRES 2015*.

[12] Chris Butler. 2018. Addressing Recent Claims of "Manipulated" Blog Posts in the Wayback Machine. http://blog.archive.org/2018/04/24/addressing-recent-claims-of-manipulated-blog-posts-in-the-wayback-machine/.

[13] John Collomosse, Tu Bui, Alan Brown, John Sheridan, Alexander L. Green, Mark Bell, Jamie Fawcett, Jez Higgins, and Olivier Thereaux. 2018. ARCHANGEL: Trusted Archives of Digital Public Documents. In *Proceedings of the ACM Symposium on Document Engineering 2018, DocEng 2018*. 31:1–31:4. https://doi.org/10.1145/3209280.3229120

[14] Craig Francis. 2018. Archived / immutable content. https://github.com/WICG/webpackage/issues/105.

[15] ISO 28500:2017. 2017. WARC file format. https://iso.org/standard/68004.html.

[16] Andy Jackson. 2018. Web Packaging a Compliment or Replacement of WARC. https://twitter.com/anjacks0n/status/950861384266416134.

[17] Mat Kelly, Sawood Alam, Michael L. Nelson, and Michele C. Weigle. 2016. InterPlanetary Wayback: Peer-To-Peer Permanence of Web Archives. In *Proceedings of the 20th International Conference on Theory and Practice of Digital Libraries*. 411–416. https://doi.org/10.1007/978-3-319-43997-6_35

[18] Ilya Kreymer and John Berlin. 2014. Wombat JS-Rewriting Library. https://github.com/webrecorder/pywb/blob/master/pywb/static/wombat.js.

[19] Ben Laurie, Adam Langley, and Emilia Kasper. 2013. Certificate Transparency, Internet RFC 6962. https://tools.ietf.org/html/rfc6962.

[20] Ada Lerner, Tadayoshi Kohno, and Franziska Roesner. 2017. Rewriting History: Changing the Archived Web from the Present. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017*. 1741–1755. https://doi.org/10.1145/3133956.3134042

[21] Gordon Mohr, Michael Stack, Igor Rnitovic, Dan Avery, and Michele Kimpton. 2004. Introduction to Heritrix. In *Proceedings of the 4th International Web Archiving Workshop*.

[22] Michael L. Nelson. 2018. Why we need multiple web archives: the case of blog.reidreport.com. https://ws-dl.blogspot.com/2018/04/2018-04-24-why-we-need-multiple-web.html.

[23] Jeni Tennison. 2018. Packaging on the Web. https://w3ctag.github.io/packaging-on-the-web/.

[24] Herbert Van de Sompel, Michael L. Nelson, and Robert Sanderson. 2013. HTTP Framework for Time-Based Access to Resource States – Memento, Internet RFC 7089. https://tools.ietf.org/html/rfc7089.

[25] Jeffrey Yasskin. 2018. Add an archival use case. https://github.com/WICG/webpackage/pull/137.

[26] Jeffrey Yasskin. 2018. Use Cases and Requirements for Web Packages. https://wicg.github.io/webpackage/draft-yasskin-webpackage-use-cases.html.

[27] Jeffrey Yasskin. 2019. Bundled HTTP Exchanges. https://wicg.github.io/webpackage/draft-yasskin-wpack-bundled-exchanges.html.

[28] Jeffrey Yasskin. 2019. Loading Signed Exchanges. https://wicg.github.io/webpackage/loading.html.

[29] Jeffrey Yasskin. 2019. Signed HTTP Exchanges. https://wicg.github.io/webpackage/draft-yasskin-http-origin-signed-responses.html.

[30] Jeffrey Yasskin. 2019. Web Packaging. https://github.com/WICG/webpackage.