2018

Swimming in a Sea of JavaScript or: How I learned to Stop Worrying and Love High-Fidelity Replay

John A. Berlin
Old Dominion University

Michael L. Nelson
Old Dominion University

Michele C. Weigle
Old Dominion University

Original Publication Citation
Berlin, J. A., Nelson, M. L., & Weigle, M. C. (2018, June 6, 2018). *Swimming in a sea of Javascript or: How I learned to stop worrying and love high fidelity replay* [Conference Paper]. Web Archiving and Digital Libraries Workshop, Fort Worth, Texas. http://hdl.handle.net/10919/97988

Swimming In A Sea Of JavaScript Or: How I Learned To Stop Worrying And Love High-Fidelity Replay

John A. Berlin, Michael L. Nelson, and Michele C. Weigle
Old Dominion University
Department of Computer Science
Norfolk, Virginia, USA
{jberlin,mln,mweigle}@cs.odu.edu

1 INTRODUCTION

Preserving and replaying modern web pages in high-fidelity has become an increasingly difficult task due to the increased usage of JavaScript. Reliance on server-side rewriting alone results in live-leakage and or the inability to replay a page due to the preserved JavaScript performing an action not permissible from the archive. The current state-of-the-art high-fidelity archival preservation and replay solutions rely on handcrafted client-side URL rewriting libraries specifically tailored for the archive, namely Webrecoder's and Pywb's wombat.js [12]. Web archives not utilizing client-side rewriting rely on server-side rewriting that misses URLs used in a manner not accounted for by the archive or involve client-side execution of JavaScript by the browser.

We have developed a general framework for the automatic generation of client-side rewriting libraries using the Web Interface Design Language (Web IDL) [10] that is archive and replay system independent. We provide a high-level overview of the auto-generation framework and evaluation performed that tested the auto-generated client-side rewriter's ability to augment the existing server-side rewriting system of the Internet Archive's Wayback Machine [3]. We show that client-side rewriting would both increase the replay fidelity of mementos and enable mementos that were previously unreplayable from the Internet Archive's Wayback Machine to be repayable again.

2 BACKGROUND AND RELATED WORK

Brunelle and Kelly [6] conducted a study of 1,861 URIs which had mementos in the Internet Archive between 2005 to 2012 in order identify the impact of JavaScript on the archivability of web pages. They found that JavaScript was responsible for 52.7% of all missing resources and that by 2012 JavaScript was responsible for 33.2% more missing resources than in 2005. Brunelle and Kelly [4, 5] also conducted a study that looked at the proportion of missing resources for mementos [15] in order to assess their damage, finding that the users' perception of damage to be a more accurate metric for judging archival quality than the proportion of missing resources.

Alam et al. [1] describe an additional solution for mitigating JavaScript replay issues through the usage of a ServiceWorker, which can intercept HTTP requests made by the currently replayed page and rewrite any URI-Rs to URI-Ms, client-side that were missed server-side. Lerner et al. [14] describes attacks, also launched from the live web, targeting web archives that are perpetrated by users of the web archive. The solutions posed by Lerner, namely archival modification of JavaScript at replay time and the separation of replayed content from the archive's presentation components of replay, parallel the existing replay strategies employed by Webrecorder and Perma.cc [7].

3 AUTO-GENERATION

Web IDL was created by the W3C to "describe interfaces intended to be implemented in web browser", "allow the behavior of common script objects in the web platform to be specified more readily", and "provide how interfaces described with Web IDL correspond to constructs within ECMAScript execution environments" [10]. Our framework uses the Web IDL definitions for the JavaScript APIs of the browser included in or link to by the HTML and CSS specification [8, 9] in combination with the description of how Web IDL maps to the JavaScript environment, provided by the Web IDL specification, in order to auto-generate a client-side rewriting library. This allows the generated rewriter to perform the same URL rewriting done server-side in addition to applying targeted overrides to the JavaScript APIs of the browser in order to intercept and rewrite un-rewritten URLs client-side.

We have released the generated client-side rewriter as FireFox[1] and Chrome[2] browser extensions so that others may use it to improve the replay of mementos from the Internet Archive. Note that although the generated client-side rewriter is similar to the *de-facto* implementation for client-side rewriting libraries, wombat.js, it is replay system agnostic.

4 EVALUATION

We retrieved the TimeMaps for the web pages listed in the June 2017 Alexa top 1,000,000 most visited websites and selected the first 700 pages, excluding Google and Facebook pages, that had a mememnto in the Internet Archive between June 1 and June 30. We then pre-crawled the URI-Ms using the Google Chrome browser controlled via the DevTools Protocol[3] removing URI-Ms from the frontier that redirected more than 10 times or took longer than 20 seconds for the browser to navigate to the page, resulting in 577 resolved URI-Ms. We then crawled each composite memento using the controlled browser four times, twice without client-side rewriting and twice with client-side rewriting, recording the number of requests made by the composite memento and the number of requests blocked by the Wayback Machine's content-security policy (CSP).

The crawler visited each composite memento for a maximum of 90 seconds or until network idle was determined. The determination for network-idle was calculated by keeping track of the request and response pairs for a page, and when there was only one in-flight request (no response) for 3 seconds the crawler moved to the next URI-M. Once all crawls had completed, we selected the data generated from one of the two crawls, with or without client-side rewriting, that recorded the most number of requests. We found

[1]https://addons.mozilla.org/en-US/firefox/addon/waybackplusplus/
[2]https://chrome.google.com/webstore/detail/wayback%20%20/kcpoejoblnjdkdfdnjkgcmmmkccjjhka
[3]https://chromedevtools.github.io/devtools-protocol/

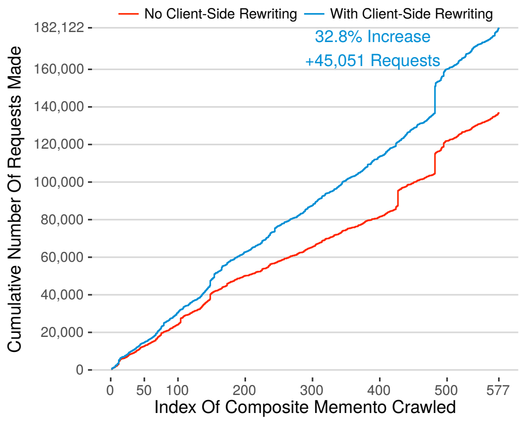

Figure 1: Cumulative number of requests made by 577 composite mementos replayed from the Internet Archive's Wayback Machine with and without client-side rewriting

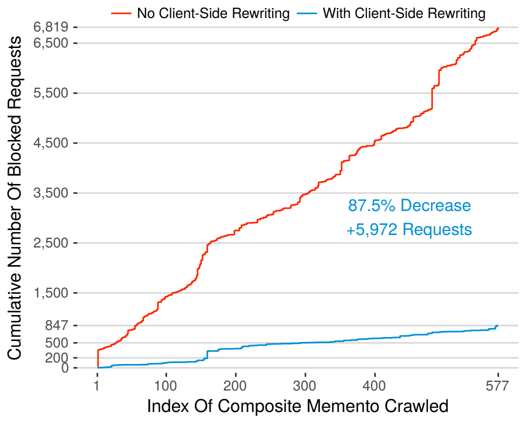

Figure 2: Cumulative number of blocked requests for 577 composite mementos replayed from the Internet Archive's Wayback Machine with and without client-side rewriting

that the composite mementos replayed with client-side rewriting made a total of 45,051 additional requests, a 32.8% increase (Figure 1) via 134,923 rewrites which occurred client-side. By crawling the mementos with client-side rewriting we were able to decrease the number of requests blocked by the CSP of the Wayback Machine by 87.5%, an increase of an additional 5,972 requests (Figure 2).

As a direct result of including the generated the client-side rewriter in the replay of the composite mementos, we were able to make composite mementos which were previously un-replayable, replayable again. The home page of cnn.com became replayable again because the generated client-side rewriter applies an override targeting the document domain issue [2]. Another notable page that became replayable again was the e-commerce site soufeel.com, which used three different ways of lazy loading its images (Figure 3).

5 CONCLUSIONS

One might believe that the usage of client-side rewriting is only limited to the most dynamic of web pages or web applications, but ensuring both high fidelity replay and the secure replay of archived JavaScript necessarily requires an archive to employ client-side rewriting. Client-side rewriting is a general solution to the increasingly difficult problems of mitigating the impact of JavaScript

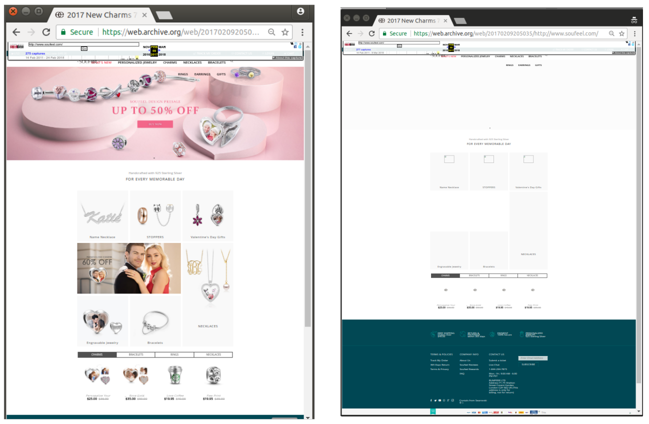

Figure 3: https://web.archive.org/web/20170209205035/http://www.soufeel.com/ increased replay fidelity from the Internet Archive's Wayback Machine with client-side rewriting

on archivability, increasing users' perception of archival quality and ensuring the secure replay of JavaScript [5, 6, 11, 13, 14].

6 ACKNOWLEDGMENTS

This work sponsored in part by NEH HK-50181 and NSF IIS 1526700.

REFERENCES

[1] Sawood Alam, Mat Kelly, Michele C. 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.* 1–4.
[2] John Berlin. 2017. CNN.com has been unarchivable since November 1st, 2016. http://ws-dl.blogspot.com/2017/01/2017-01-20-cnncom-has-been-unarchivable.html. (2017).
[3] John A. Berlin. 2018. *To Relive The Web: A Framework For The Transformation And Archival Replay Of Web Pages.* Master's thesis. Old Dominion University, Department of Computer Science.
[4] Justin Brunelle, Mat Kelly, Hany SalahEldeen, Michele C. Weigle, and Michael L. Nelson. 2015. Not All Mementos Are Created Equal: Measuring the Impact of Missing Resources. *International Journal of Digital Libraries (IJDL)* (2015). https://doi.org/10.1007/s00799-015-0150-6
[5] Justin F. Brunelle, Mat Kelly, Hany SalahEldeen, Michele C. Weigle, and Michael L. Nelson. 2014. Not All Mementos Are Created Equal: Measuring the Impact of Missing Resources. In *Proceedings of ACM/IEEE Digital Libraries (DL).* 321–330.
[6] Justin F. Brunelle, Mat Kelly, Michele C. Weigle, and Michael L. Nelson. 2015. The Impact of JavaScript on Archivability. *International Journal of Digital Libraries (IJDL)* 17 (2015). https://doi.org/10.1007/s00799-015-0140-8
[7] Jack Cushman and Ilya Kreymer. 2017. Thinking like a hacker: Security Considerations for High-Fidelity Web Archives. Presented at International Internet Preservation Consortium (IIPC) Web Archiving Conference (WAC). (June 2017).
[8] W3C Working Group. 2017. *CSS Snapshot 2017.* W3C Editor's Draft. https://www.w3.org/TR/CSS/
[9] WHATWG Working Group. 2017. *HTML Living Standard.* WHATWG Living Standard. The Web Hypertext Application Technology Working Group. https://html.spec.whatwg.org/
[10] WHATWG Working Group. 2017. *WebIDL Level 1.* W3C Recommendation. The Web Hypertext Application Technology Working Group. https://www.w3.org/TR/WebIDL-1/
[11] Mat Kelly, Justin F Brunelle, Michele C Weigle, and Michael L Nelson. 2013. On the change in archivability of websites over time. In *International Conference on Theory and Practice of Digital Libraries (TPDL).* 35–47.
[12] Ilya Kreymer. 2018. wombat.js: Wombat JS-Rewriting Library. As apart of Pywb, https://github.com/webrecorder/pywb/blob/develop/pywb/static/wombat.js. (2018).
[13] Kalev Leetaru. 2017. Are Web Archives Failing The Modern Web: Video, Social Media, Dynamic Pages and The Mobile Web. https://www.forbes.com/sites/kalevleetaru/2017/02/24/are-web-archives-failing-the-modern-web-video-social-media-dynamic-pages-and-the-mobile-web. (2017).
[14] 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.* 1741–1755.
[15] H. Van de Sompel, M. Nelson, and R. Sanderson. 2013. HTTP Framework for Time-Based Access to Resource States – Memento. (12 2013). http://tools.ietf.org/rfc/rfc7089.txt RFC7089.