"StudySandboxx: A Tool for Scraping, Sandboxing, Preserving, and Preparing Interactive Web Sites for Use in Human-computer Interaction and Behavioral Studies"

Journal of open research software

SOFTWARE METAPAPER

ubiquity press

GABI WETHOR

MATTHEW L. HALE

*Author affiliations can be found in the back matter of this article

ABSTRACT

Human-computer interaction and computer-mediated behavioral psychology research studies often rely on capturing user interaction data to characterize online behaviors. IRB considerations, site policies, and/or security and privacy concerns may force researchers to use screenshots or offline copies of pages of interest, instead of live websites, in their study designs. These interaction modalities reduce the fidelity and contextual realism of web content and often affect interface aesthetic quality – due to broken links, missing images, and/or malfunctioning scripts. StudySandboxx is a tool that allows websites to be saved exactly as they appear online. The tool sandboxes websites in a way that removes dangerous scripts that threaten privacy and security. Saved websites are encapsulated into a single portable file that contains all related website resources. Finally, the tool also supports certain types of permutations commonly used in research – such as changing links in a page. The project is housed within a GitHub repository at https://github.com/gewethor/study-sandbox.

CORRESPONDING AUTHOR:
Matthew L. Hale
Assistant Professor, School of Interdisciplinary Informatics, College of Information Science and Technology, University of Nebraska at Omaha, US
mlhale@unomaha.edu

KEYWORDS:
Interactive content; human computer interaction; sandboxing; fidelity; behavioral studies; interaction studies; offline; web

TO CITE THIS ARTICLE:
Wethor G, Hale ML 2022 "StudySandboxx: A Tool for Scraping, Sandboxing, Preserving, and Preparing Interactive Web Sites for Use in Human-computer Interaction and Behavioral Studies". *Journal of Open Research Software*, 10: 6. DOI: https://doi.org/10.5334/jors.274

(1) OVERVIEW

INTRODUCTION

Websites are increasingly rich, make use of client-side responsive rendering, and use sophisticated JavaScript and Cascading Style Sheet (CSS) frameworks to provide dynamic, interactive, and responsive online user experiences. Human-computer interaction (HCI) researchers and behavioral psychologists focusing on human-computer issues often build experiments and conduct studies that involve users interacting with website content. Engagement drivers [19], design aesthetics [16], cognitive overload [3], and workflow efficiency [20] are but a few active subfields in HCI research that rely on user-to-web interaction data. Due to IRB constraints, site policies, security and privacy concerns, or other reasons, studies are often forced to use offline forms of content to study user interaction [5]. This can mean using web pages that have been downloaded and converted for offline use [7] or screenshotting live websites of interest and later showing them as static images for subject assessment [11, 17]. These methods diminish the fidelity, interactivity, and often, the aesthetic quality of the content – which can, in turn, limit the research study design.

User privacy, security, and safety concerns are critically important in experimental design. In web-based studies, leaking participant personal information is a real possibility, since researchers often do not have control of how a participant may interact with a site [1], what data the site collects [2], or the potential network of other sites and services which may have access to user data on a live website [4]. For studies in the cybersecurity research domain, lack of control of webpage content is a large safety concern [7, 12], since content of interest may include malicious links, unexpected popups, malware, and other generally maligned web content.

StudySandboxx is a new content preparation tool that allows websites to be saved exactly as they appear online while ensuring privacy and security constraints are met in research studies. The tool does more than just download a webpage. It downloads every connected resource and collapses the webpage and all of its related resources into a single portable file. The file is then sandboxed (a term that means "to isolate in a contained box") to free it of unwanted, potentially dangerous, malicious, or privacy violating, scripts. StudySandboxx is built with open source technologies and can save, sandbox, and store any website online while preserving the dynamism, interactability, and aesthetic quality of the original.

EXISTING WEBPAGE COLLECTION TECHNIQUES

Multiple works [6, 8, 13] highlight the importance of contextual realism and true-to-form interfaces when exploring research problems related to user engagement and user experience (UX). Without this realism, results may not be meaningful or applicable to the interface(s) of interest [18]. Researchers conducting experiments involving web-based content have the following options available to them for capturing web pages of interest: use the pages as they appear online as hosted by external entities, capture screenshots of the pages, download the pages using a browser, or develop look-alike copies of the pages themselves. Each approach has various benefits and drawbacks. Using real online websites has the most contextual realism, but websites can change at any time, making them difficult to use in studies. Privacy and security concerns also complicate user studies involving IRBs. Screenshots provide the least realism but are stable and free of privacy and security concerns. Downloading pages using a browser and running the sites offline ensure stability of the content, but pages may not render correctly or may run unwanted scripts. Developing look-alike pages deals with privacy and security concerns but is prohibitively expensive in all but trivial cases due to the time and resources required to mimic the look and feel of the real sites.

Sandboxing is a concept that derives its name from the playground. The concept implies a clean separation between what is in the sandbox and what isn't. When applied to web pages, the term is often referred to as "content sandboxing" to indicate that content is rendered in a contained, protected space. Driven by HCI and behavioral psychology, the research community has created a few sandboxing tools. Among these, two are commonly used in research studies. The first, HTTrack [15], downloads an existing website for local use as well as any third-party resources in the site. Built for windows and Linux, HTTrack does not remove external resources and scripts hosted by 3rd parties, e.g. tracking scripts, and does not save webpage as a single file, which complicates hosting the webpages in user studies. The second, commonly used tool, named GrabzIt [10], offers a simple API for creating thumbnails and website screenshots. The tool offers multiple tiers of account plans ranging from free to $54.99 a month. As a screenshot-based approach, GrabzIt does not preserve the fidelity or interactive features of content, reducing realism if used in UX or user engagement research studies. Neither HTTtrack or GrabzIt provide any content customization.

Other researchers, including [7, 9], have used simple approaches tools such as "Save page as" or CURL (a command-line tool for downloading online media) to save pages and their sub resources to for offline usage. The problem with these approaches is that they run the same code the page has online – meaning scripts may connect to third-party resources, log participant data, or run unsafe malicious code that violates safety and privacy requirements of the research space.

Our survey of existing literature and inability to identify extant sandbox tooling that both preserved content fidelity and protected user privacy and security is what ultimately led to the creation of StudySandboxx. To exemplify this motivation further, Figures 1–3 show a network capture of facebook.com using the Chrome Developer Tools

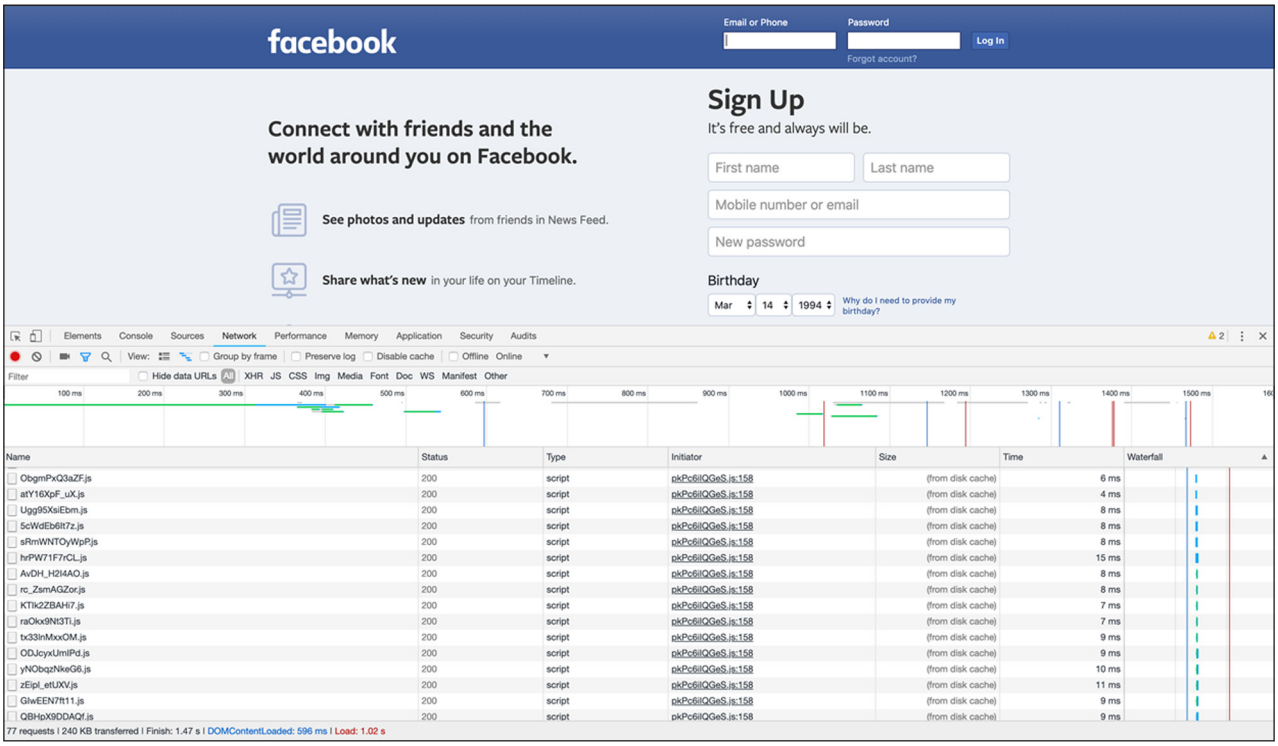


Figure 1 Scripts loaded at run-time by other scripts running on facebook.com, as shown in Chrome Developer Tools.

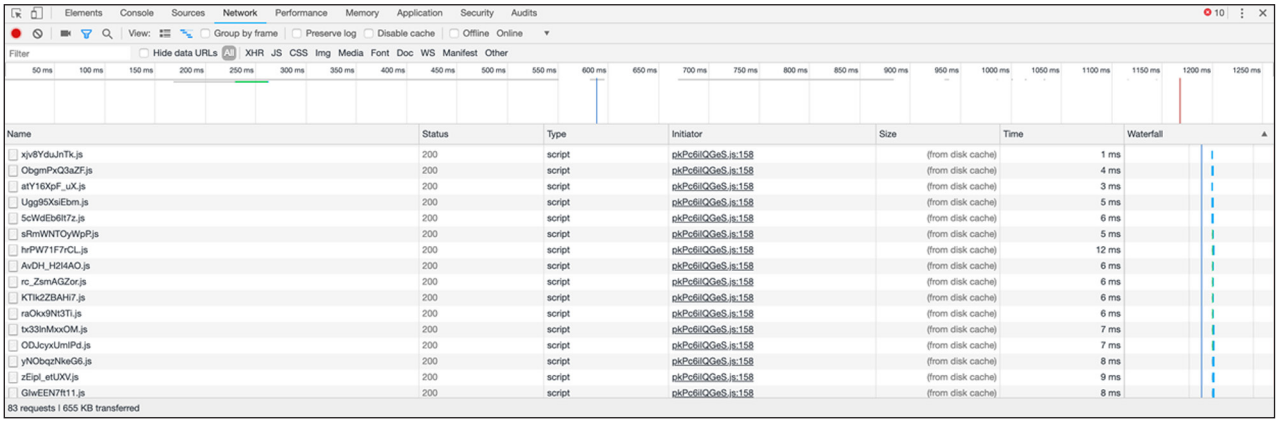


Figure 2 Chrome Developer Tools inspecting the network behavior of "Save as Page" version of facebook.com. Here the site exhibits similar run-time behavior as seen in Figure 1.

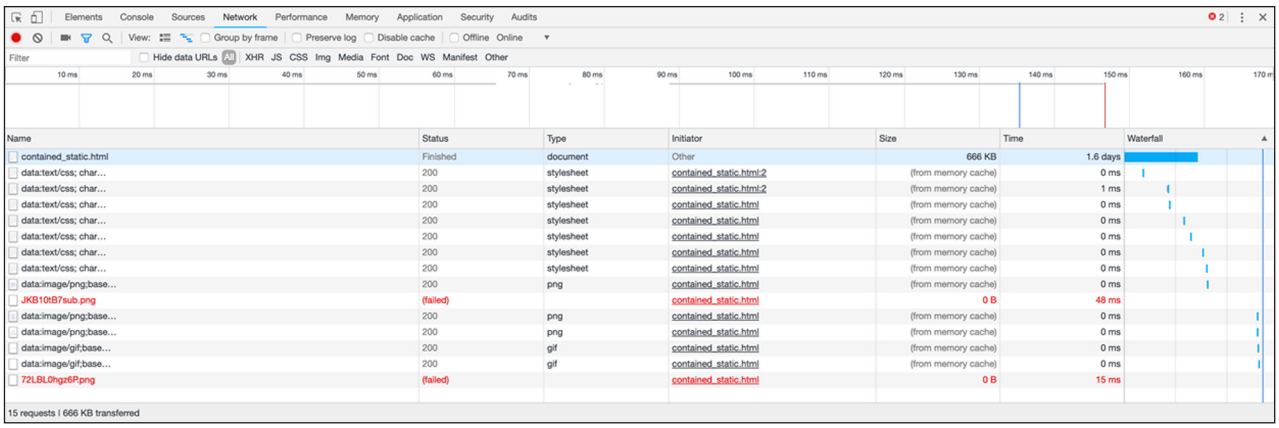


Figure 3 Network behavior of StudySandboxxed version of Facebook.com. Here the site loads only text, CSS, and image files. Overall, the number of requests has been reduced to 15 and no scripts are running in the page. The red failed image resources are advertisements from third parties.

network tab. Figure 1 shows a capture of the live online site as hosted by Facebook. Figure 2 shows the network traffic generated when the page is saved locally and run from a browser. Finally, Figure 2 shows the network traffic generated by the page after it has been saved and sandboxed using StudySandboxx. Both the Facebook origin site and the locally "Save page as" versions of the site show that scripts are running in the background and loading additional scripts dynamically at run time. These may be collecting information about the user, in this case a research participant, or associated research space. By contrast, Figure 3 shows that no scripts or third-party resources are loaded at render time or dynamically during run time afterwards. In this way, StudySandboxx alleviates risk to research participants of accidental data exfiltration.

IMPLEMENTATION AND ARCHITECTURE

In order to create a tool that meets the use cases outlined above, the following technical requirements were specified for StudySandboxx:

1. Encapsulate a webpage into a single file independent of resources hosted by 3rd parties.
2. Minimize any and all changes to the downloaded webpages so that they maintain the same visual effect as the original.
3. Maintain interactive features while ensuring users are not redirected anywhere outside the sandboxed webpage.
4. Prevent information leakage to protect user and researcher privacy.
5. Block external sources from loading potentially malicious resources.
6. Provide ease of use for researchers with limited or no knowledge of web development.
7. Open source

To address these requirements, we created StudySandboxx as a command-line python-based tool. The overall software architecture of StudySandboxx is shown in Figure 4. The section that follows describes the details and design rationale behind the architecture.

StudySandboxx supports several types of content encapsulation (see *Usage* for command-line syntax). The tool allows users to pass a series of arguments that determine if a list of sites should be encapsulated or a single site and if the site should have its links retargeted to another domain (e.g. changing a link such as "facebook.com/login" to "fb.com/login"). The first component in the architecture shown in Figure 4 supports this invocation logic. After the mode has been determined, web content is rendered and related resources (e.g. html, CSS, js, and

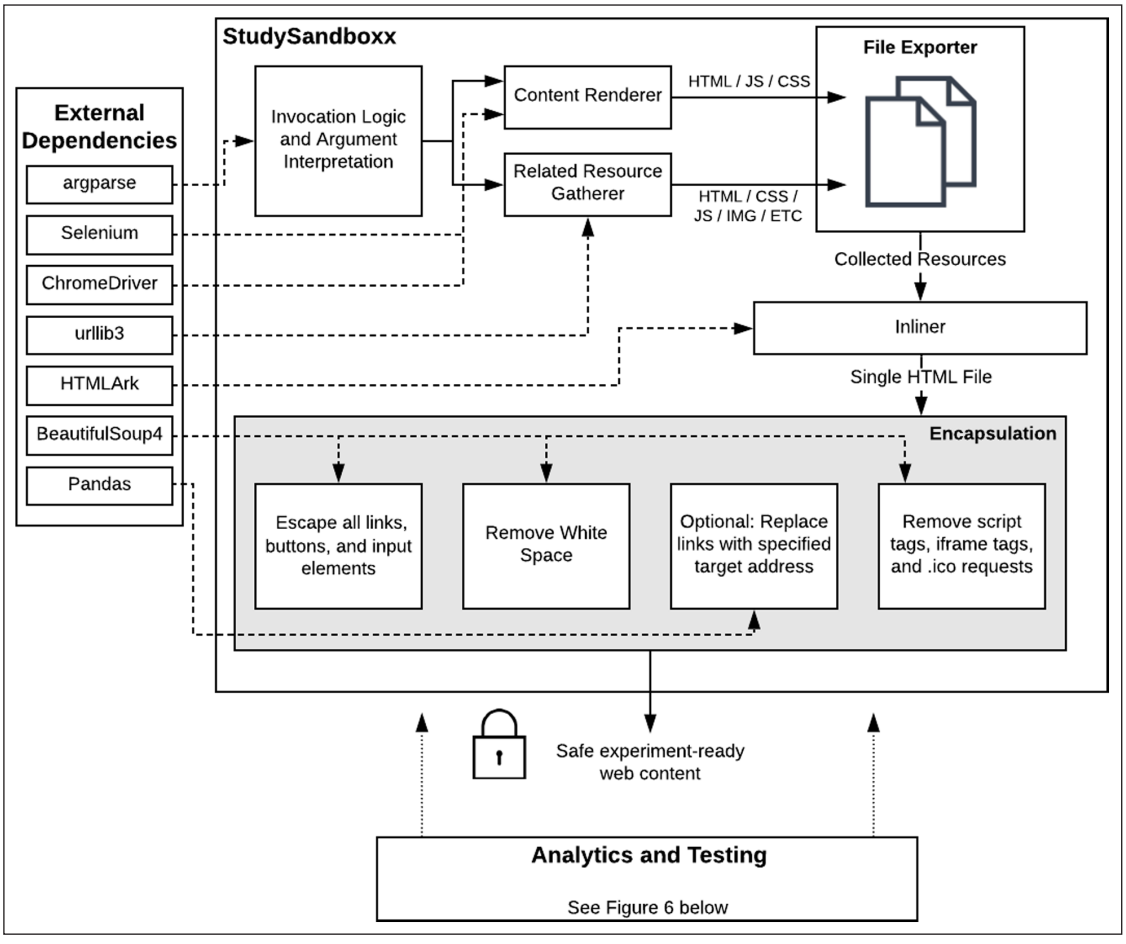


Figure 4 StudySandboxx Software Architecture.

images, from third party sources) are gathered. A feature that separates StudySandboxx from other webpage saving/sandboxing methods is the use of a headless browser (ChromeDriver and Selenium) to actively render the content as it would normally be rendered by a browser. This allows the tool to faithfully render JavaScript heavy client-side web content. Other static-only approaches would not be able to support these sites since the content renders its interface at run-time in the client. Due to the variability in the use of JavaScript within websites, StudySandboxx renders content twice, once using JavaScript and once using static-only resources. Researchers are encouraged to use whichever resource best matches the origin site. Later, in this work, a set of automated tests are described to determine best match using an automated visual analytic method.

Once both the static and JavaScript pages have rendered, they are exported and saved locally before being in-lined and encoded using HTMLArk [14]. Any third-party resources such as images and gifs gathered in earlier steps are base64 encoded into the html file so that the researcher ends up with a single file. This also allows the content to fully render without access to any external resources or locally stored files.

After the in-lining and encoding process, the content undergoes an encapsulation process. This process is another feature that separates StudySandboxx from simply using a HTML in-liner or another containerization method such as HTTtrack or GrabzIt. Using BeautifulSoup4, all interactive elements such as links, buttons, and inputs are escaped so that they will "return false" when users attempt to click on or add personal data into elements. This prevents users from getting redirected to a real site, if the content is presented in an internet-connected machine. All script and iframe tags are removed from the html or sandboxed to prevent run-time loading of potentially malicious scripts into the website. Furthermore, "icon" and "shortcut icon" link attributes are removed to prevent .ico requests from the webpage. The whitespace is also removed from the html file to minimize file size.

StudySandboxx also allows for the optional ability to replace links within the content. While the links will not redirect the user to a new site, hovering over the link in most web browsers will inform the study participant where the link would redirect them. This feature has applications in many research areas such as cybersecurity phishing studies and UI design where experimenters want to tweak web domain as an experimental parameter (e.g. determining how users respond when the see links to certain trustworthy or untrustworthy sites).

The following is a summary of the events that are performed in the data transformation process:

- Retrieves web content and associated resources
- Renders content statically and using JavaScript in a headless browser
- Base64 encodes images, gifs, and other media
- In-lines media and other resources into HTML using HTMLArk
- Escape all links, buttons, and input elements without affecting interactivity
- Removes white space
- Removes and/or sandboxes <script> and <iframe> tags
- Removes HTML attributes allowing .ico file requests
- Optional: replaces links with specified target addresses

Once the file encapsulation has finished, StudySandboxx will output two files per website entered, one statically rendered and the other rendered with JavaScript. The researcher can then determine which of the two HTML files will be most successful in their study. This project provides a testing component (further discussion below) which determines which html file is the closest, visually, to the origin using image processing and pixel comparison.

USAGE

The requirement for usability independent of web development knowledge is fulfilled based on limited installation requirements and the simple command-line execution. To get started using StudySandboxx, simply begin by cloning the GitHub repository at https://github.com/MLHale/study-sandboxx. The repository has further detailed instructions, starting with a list of required dependencies. Command-line arguments accepted by StudySandboxx allow for single website use, multi-website use, and optional change of target link addresses. Specific commands are shown below consistent with the documentation on GitHub.

Single Website Use

To sandbox and encapsulate a single website:

```
python3 contain.py -u [web address of site]
```

Example

`python3 contain.py -u` https://www.facebook.com containerizes the site as shown in Figure 5.

For containerization as well as transformation of content links:

```
python3 contain.py -u [web address of site] -l
[link target address]
```

Example

`python3 contain.py -u facebook.com -l` http://www.anothersite.com

Multiple Website Use

If multiple websites are being containerized, the input must be passed via a csv file without the use of headers. See Table 1 for an example CSV file format.

If the user does not wish the change the target addresses of the content links, the third column should be left blank.

Example

```
python3 contain.py -i [path-to-csv]
```

QUALITY CONTROL

To ensure that StudySandboxx meets its requirements, the project includes an analytics and testing component that compares the output of StudySandboxxed content to its originating site along a number of metrics, described below. The component also compares our sandboxing approach against other commonly used containerization techniques – including saving the website locally using "Save As" with the format "Webpage, HTML Only", and the web capture tool GrabzIt. Figure 6, below, overviews the flow of testing within the analytics and testing component. The flow starts by launching the Selenium web driver. Next, StudySandboxx, Grabzit, and Save As containerization techniques are run. Next the web driver framework is used to gather summary statistics about the operation of the content in its browser-based environment. Next, the content produced by each

Facebook	https://www.facebook.com	
GitHub	https://github.com/	http://www.testingwebsite.com/
Dropbox	https://www.dropbox.com/home	

Table 1 Example CSV for Multiple Webpage Use.

```
Gabis-MacBook-Pro:containerize-experiment-stimuli gabiwethor$ python3 contain.py -u https://www.facebook.com
containerizing https://www.facebook.com
containerizing js rendered html
js rendered html of https://www.facebook.com successfully containerized
static html of https://www.facebook.com successfully containerized
End
```

Figure 5 Single Website Containerization Process.

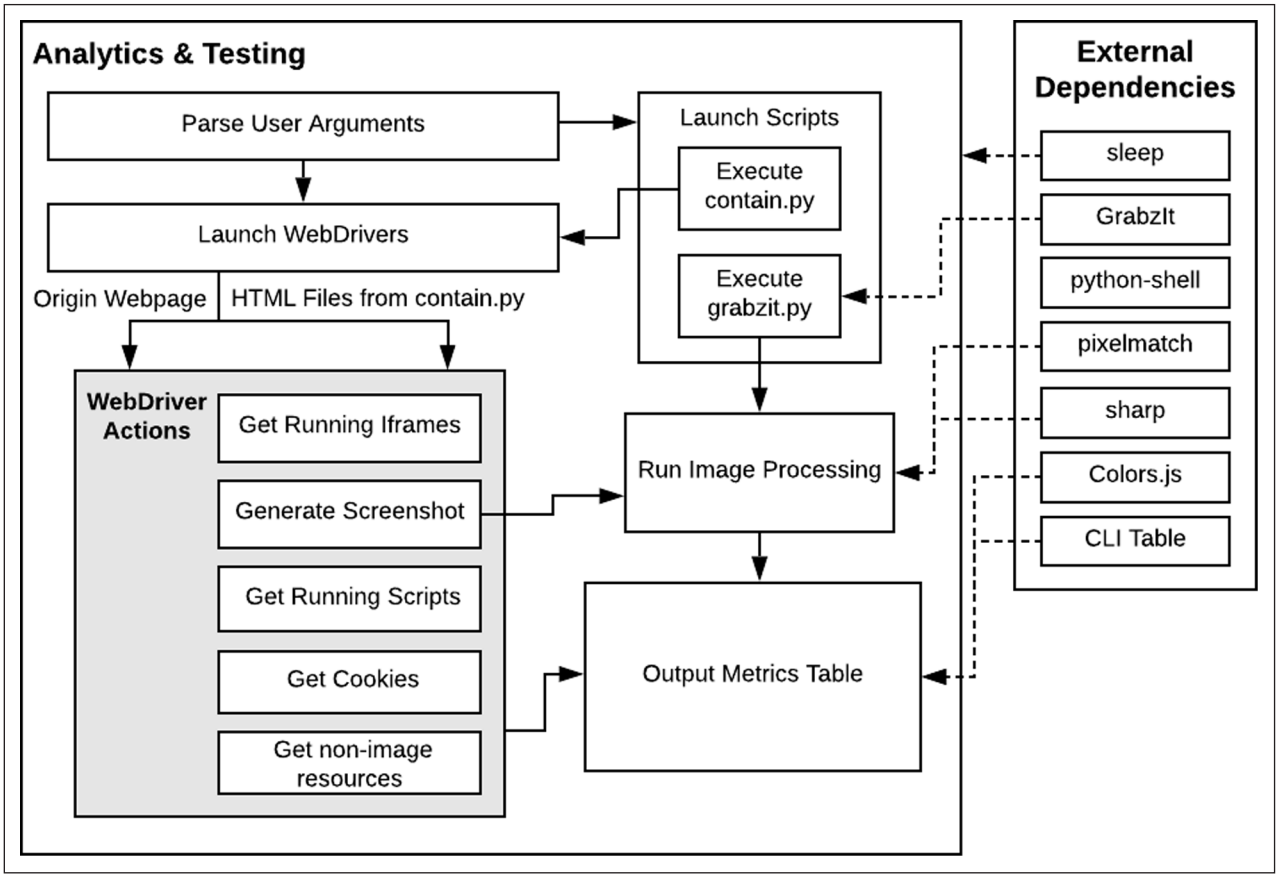


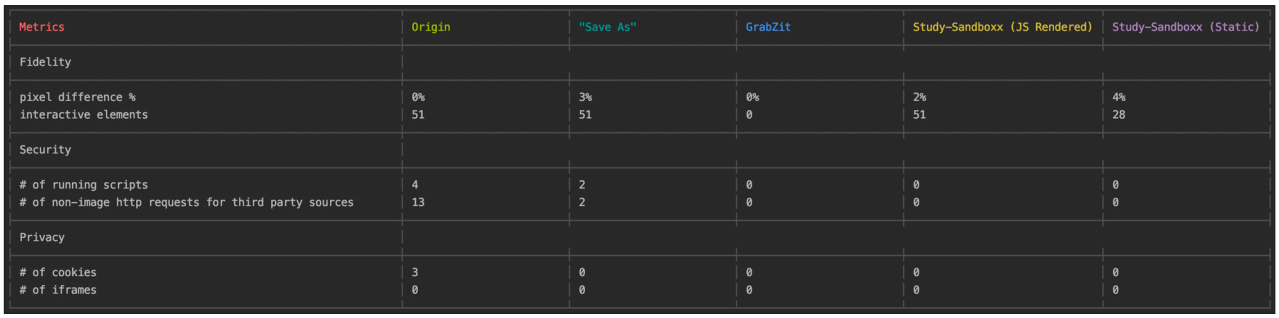
Figure 6 Analytics and Testing Process.

technique is screenshotted and run through a pixel-match comparison library to determine the % difference between origin and technique. Finally, summary statistics and the pixel-match comparison are summarized into an output table and presented to the user as shown in Figure 7 below.

In order to test and compare techniques, the analytics and testing component examines several metrics in its analysis. Three traits were identified from the literature as important to user experimentation: fidelity, security, and privacy. Fidelity, in this context, is the faithfulness of the containerized content to its origin. Perfect fidelity is achieved when there is no perceivable difference between content in an experiment and the origin content it originated from. Security, in this context, is that user and experimentation resources are not compromised or harmed by web content. Privacy concerns relate to keeping user information private by not having those details leak out to unauthorized individuals or entities. In this context, privacy is very important since the experiments may involve IRBs or other protective requirements on user data. The metrics listed below in Table 2 provide objective quantitative touchpoints to examine if our software requirements within the areas of Fidelity, Security, and Privacy are met. They also allow for objective comparison to other techniques.

Most of the metrics are self-explanatory. Two, of them require some further discussion. The first, interactive elements, is the sum of all html elements that have interactive capabilities. Interactive capabilities are those that allow for a user action, such as buttons, links, mouseovers, etc. The second, pixel difference, is the percentage difference between the origin and the content generated by each respective containerization technique. The percentage of pixel difference is calculated by comparing a screenshot of the origin website and a screenshot of the website created by one of the compared techniques. This is done using an image comparison library called pixelmatch. The library creates a diff image of the two screenshots and pixel-by-pixel compares the diff with the origin screenshot.

The total number of different pixels is divided by the total amount of pixels within the screenshot (resolution 1920x1080) then multiplied by 100 to produce a percentage. It should be noted that GrabzIt automatically returns the value 0% for pixel percent difference in the output of the script. This is due to down sampling pixel issues that result from the difference in image size when running pixel comparison between the origin and GrabzIt images.



Metrics	Origin	"Save As"	GrabZit	Study-Sandboxx (JS Rendered)	Study-Sandboxx (Static)
Fidelity					
pixel difference %	0%	3%	0%	2%	4%
interactive elements	51	51	0	51	28
Security					
# of running scripts	4	2	0	0	0
# of non-image http requests for third party sources	13	2	0	0	0
Privacy					
# of cookies	3	0	0	0	0
# of iframes	0	0	0	0	0

Figure 7 Output from the Analytics and Testing Component.

METRIC	DESCRIPTION
Fidelity	
Pixel Difference Percentage	The percent of pixel difference between a screenshot of the origin website and a screenshot of website acquired using each of the content techniques.
Number of Interactive Elements	The total amount of interactive elements (input and link elements) within each webpage.
Security	
Number of Running Scripts	The number of scripts running in the browser.
Number of non-image HTTP Requests for Third-Party Sources	The number of non-image HTTP requests for third-party sources.
Privacy	
Number of Cookies	The number of cookies from the origin website.
Number of iframes	The number of running iframes.

Table 2 Captured Fidelity, Security, and Privacy Metrics for Content Comparison.

(2) AVAILABILITY

OPERATING SYSTEM

Windows, MacOS, Linux

PROGRAMMING LANGUAGE

Python 3.7 (minimum Python 3.0)
JavaScript

ADDITIONAL SYSTEM REQUIREMENTS

Memory: 500MB RAM
Disk space: 1 GB
Processor: 32-bit or 64-bit

DEPENDENCIES

Containerization Script

Selenium 3.141.0
ChromeDriver (2.46)
HTMLArk 1.0.0
argparse 1.4.0
beautifulsoup4 4.7.1
urllib3 (Python 3.4+)
Pandas 0.23.4

Testing Script

sharp 0.21.3
sleep 6.0.0
pixelmatch 4.0.2
request 2.88.0
GrabzIt 3.3.0-1
CLI Table 0.3.1
Colors.js 1.3.3
python-shell 1.0.7

LIST OF CONTRIBUTORS

Gabi Wethor & Matt L. Hale

SOFTWARE LOCATION:

Archive

Name: StudySandboxx (Version 1.0).
Persistent identifier: http://doi.org/10.5281/zenodo.3674294
License: Creative Commons Attribution 4.0 International
Publisher: Matthew L. Hale
Version published: *v1.0.0*
Date published: 18/02/20

Code Repository

Name: StudySandboxx
Identifier: https://github.com/MLHale/study-sandboxx
License: MIT
Date published: 14/03/19

LANGUAGE

English

(3) REUSE POTENTIAL

From communication to news to retail to entertainment, day-to-day needs are increasingly being met online. Large shifts in individual behavior, such as increasing e-commerce, social media usage, and use of curated new sources pose new and varied research questions for behavioral psychologists, and human-computer interaction researchers studying the effects of computer-mediation on social and psychological user behaviors.

In lab studies of behavior, there are limited methods available which accurately emulate real-world web ecosystems in user studies. While some studies allow for participants to interact directly with live websites or in active online communities, for many this is not an option. IRB constraints, site policies, privacy concerns, and other factors can force researchers to use offline content when conducting user research. Furthermore, current offline content acquisition approaches present flaws in interactability, participant safety, and aesthetic quality. This software presents researchers with the ability to rapidly prepare webpage content for user-centric studies while at the same time does not sacrifice content richness.

StudySandboxx is a living project with availability for growth and expansion. Any developers interested in contributing to StudySandboxx can do so by making a pull request on our GitHub page, https://github.com/MLHale/study-sandboxx. Furthermore, if any issues arise while using this software, they may be reported within the issues tab of our repository.

FUNDING STATEMENT

This work was supported in part by a Nebraska Research Initiative (NRI) grant entitled "Identifying, assessing, and mitigating wearable security issues in the internet of things." NRI is an investment by the State of Nebraska to provide a research base to enhance economic growth in business and industry, agriculture, social services and health care. The views expressed in this paper are those of the authors and do not necessarily reflect those of NRI or the State of Nebraska.

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR AFFILIATIONS

Gabi Wethor
Associate Professor, College of Information Science and Technology, University of Nebraska at Omaha, US

Matthew L. Hale orcid.org/0000-0002-8433-2744
Assistant Professor, School of Interdisciplinary Informatics, College of Information Science and Technology, University of Nebraska at Omaha, US

REFERENCES

1. **Ackerman MS, Mainwaring SD.** Privacy issues and human-computer interaction. *Computer*. 2005; 27(5): 19–26.
2. **Adams A, Sasse MA.** Privacy in multimedia communications: Protecting users, not just data. In *People and computers XV—Interaction without frontiers*, 2001; 49–64. 2001. London: Springer. DOI: https://doi.org/10.1007/978-1-4471-0353-0_4
3. **Alomyan H.** Individual differences: Implications for Web-based learning design. *International Education Journal*. 2004; 4(4): 188–196.
4. **Barkhuus L.** The mismeasurement of privacy: using contextual integrity to reconsider privacy in HCI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012, May; 367–376. DOI: https://doi.org/10.1145/2207676.2207727
5. **Beel J, Langer S.** A comparison of offline evaluations, online evaluations, and user studies in the context of research-paper recommender systems. In *International conference on theory and practice of digital libraries*, 2015, September; 153–168. Cham: Springer. DOI: https://doi.org/10.1007/978-3-319-24592-8_12
6. **Carter AS, Hundhausen CD.** How is user interface prototyping really done in practice? a survey of user interface designers. In *2010 IEEE Symposium on Visual Languages and Human-Centric Computing*, 2010, September; 207–211. IEEE. DOI: https://doi.org/10.1109/VLHCC.2010.36
7. **Darwish A, Bataineh E.** Eye tracking analysis of browser security indicators. In *2012 International Conference on Computer Systems and Industrial Informatics*, 2012, December; 1–6. IEEE. DOI: https://doi.org/10.1109/ICCSII.2012.6454330
8. **Dede C.** Immersive interfaces for engagement and learning. *Science*. 2009; 323(5910): 66–69. DOI: https://doi.org/10.1126/science.1167311
9. **Dhamija R, Tygar JD, Hearst M.** Why phishing works. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 2006, April; 581–590. DOI: https://doi.org/10.1145/1124772.1124861
10. **GrabzIt.** 2018. Available: https://grabz.it/.
11. **Kim N, Koo B, Yoon J, Cho K.** Understanding the formation of user›s first impression on an interface design from a Neurophysiological Perspective-EEG Pilot Study. In *Proceedings of HCI Korea*, 2016; 139–145. DOI: https://doi.org/10.17210/hcik.2016.01.139
12. **Miyamoto D, Blanc G, Kadobayashi Y.** November. Eye can tell: On the correlation between eye movement and phishing identification. In *International Conference on Neural Information Processing*, 2009; 3847–3852. Cham: Springer. DOI: https://doi.org/10.1007/978-3-319-26555-1_26
13. **Petrie H, Harrison C.** Measuring users› emotional reactions to websites. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, 2009; 3847–3852. DOI: https://doi.org/10.1145/1520340.1520582
14. **Powell D.** HTMLArk; 2015.
15. **Roche X.** HTTrack Website Copier; 2018.
16. **Schenkman BN, Jönsson FU.** Aesthetics and preferences of web pages. *Behaviour & Information Technology*. 2000; 19(5): 367–377. DOI: https://doi.org/10.1080/014492900750000063
17. **Stojmenovic M, Pilgrim C, Lindgaard G.** Perceived and objective usability and visual appeal in a website domain with a less developed mental model. In *Proceedings of the 26th Australian computer-human interaction conference on design*, 2014, December; 316–323. DOI: https://doi.org/10.1145/2686612.2686660
18. **Tashiro JS, Dunlap D.** The impact of realism on learning engagement in educational games. In *Proceedings of the 2007 conference on Future Play*, 2007, November; 113–120. DOI: https://doi.org/10.1145/1328202.1328223
19. **Webster J, Ahuja JS.** Enhancing the design of web navigation systems: The influence of user disorientation on engagement and performance. *Mis Quarterly*. 2006; 661–678. DOI: https://doi.org/10.2307/25148744
20. **Wiley K, Getto G.** A UX workflow for building awesome applications. *Communication Design Quarterly Review*. 2015; 3(3): 49–52. DOI: https://doi.org/10.1145/2792989.2792996

TO CITE THIS ARTICLE:
Wethor G, Hale ML 2022 "StudySandboxx: A Tool for Scraping, Sandboxing, Preserving, and Preparing Interactive Web Sites for Use in Human-computer Interaction and Behavioral Studies". *Journal of Open Research Software*, 10: 6. DOI: https://doi.org/10.5334/jors.274

Submitted: 26 March 2019 **Accepted:** 06 March 2020 **Published:** 12 July 2022