Prospectus

Automating DOM-based Cross Site Scripting Protections on Chromium and Chromium-based browsers

(Technical Report)

Investigating the barriers to pipelining excess and recyclable foods for Charlottesville's food insecure

(STS Research Paper)

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Dale Wilson

Fall, 2019

Department of Computer Science

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

Signed: ______________________________

Approved: ______________________________ Date 12/11/2019

Sean Ferguson, Department of Engineering and Society

Approved: Yuan Tian Date 12/05/2019

Yuan Tian, Department of Computer Science

Introduction

Security thrives as a field built to prevent, mitigate, and predict the occurence of danger. In an age dominated by digital infrastructures that manage how we socialize, travel, and eat, a vast array of nascent technologies grow to secure the modern digital landscape. Non-profit efforts have evolved to leverage these digital systems for their unparalleled efficiencies, exposing stakeholders to a new age of technical vulnerabilities regarding privacy (Council of Nonprofits, 2019). A collective push for digitizing commercial and nonprofit organizations stresses the need for comprehensive security measures across an ever-increasing number of distributed web applications.

The technical aspect of my prospectus will focus on preventing DOM-based Cross Site Scripting (XSS for short) attacks on the open source Chromium project, the project upon which Google's Chrome and Microsoft's Edge browsers are built. XSS attacks occur when malicious code is injected into benign websites, which will then distribute the code to unwitting users and attack their internet browsers. The research produced by the prospectus should yield a low-overhead JavaScript library that works in tandem with existing XSS security measures already implemented within browsers while maximizing compatibility with inline and dynamically loaded scripts. It should also produce a contribution to the chromium project that updates the runtime script runner strategy for the chromium browser.

My STS research topic investigates the viability of a secured web platform to connect excess foods to food insecure individuals in the greater Charlottesville area. I analyze two case studies that detail the roadblocks experienced by food-waste reduction advocates abroad and at an American university. The two investigations cover socio-political aspects of the food-waste issue and provide perspective on the differences in perceived American disdain for food recycling and international disdain for food waste. This relates to the overall team blueprint as it illuminates possible cultural and social roadblocks for proper adoption of a web platform that aims to reduce food waste.

Technical Topic

XSS attacks have been one of the most prevalent threats to the modern web over the past decade. Web development has progressed in a direction where heavy javascript is executed on the browser, exposing the browser to more Document Object Model-XSS (DOM-XSS) attacks that are undetectable by servers distributing web applications. Detection of these DOM-XSS attacks leverages taint tracking to identify whether data from attack-controlled sources can reach sensitive sink functions (Melicher W., Das A., Sharif M., Bauer L., Jia L. 2018)). These sensitive sink functions directly modify the DOM on the browser. This attack surface can be abused in a variety of ways but the vectors of greatest interest are those by which an attacker can execute code via URL-based sources. Methods used by CMU's DOM-XSS research projects have found 83% more vulnerabilities than that of previous studies, indicating that DOM-XSS attacks can abuse an increasing attack surface (Melicher W., Das A., Sharif M., Bauer L., Jia L. 2018).

The UVA research project aims to contribute DOMinatriXSS, a browser native defense against DOM-XSS (Tian Y. 2015). There are two components to the project: DOMinatriXSS, an externally loaded JavaScript library maximizing the defense's adoptability, and DOMinatriXSSStatic, the Chromium project contribution to enable the 'disable-dynamic' CSP directive which maximizes security. I plan to work on DOMinatriXSS first, as the implementation operates independently from the browser implementation. I can later update the library's behavior to handle the 'disable-dynamic' CSP directive necessary for DOMinatriXSSStatic. I hope to find an already developed model for testing the effectiveness of DOMinatriXSS, else I will develop a model using the same guidelines described by Tian's approach using the Alloy modeling language (Tian Y. 2015).

I segmented the development process of the DOMinatriXSS JavaScript library into three phases, (1) the inline event handler conversion of scripts to functions, (2) modifying/injecting a meta tag to enforce generated script nonce, and (3) adding generated script nonce to imported, dynamic JavaScript libraries via document.createElement. These three phases parallel the three core components of the library. I've confirmed the effectiveness of the inline event handler conversion and meta tag enforcement by creating a version of DOMinatriXSS and a test HTML file that enforces a csp with a script nonce but does not rewrite the inline event handlers. This has allowed me to confirm buttons or elements modified with this version of DOMinatriXSS fail to respond to their handled events (onEvent, onClick, onHover…). To confirm the effectiveness of updating the document.createElement() function to include the enforced script nonce, I created a script that will alert the webpage (using the onLoad event handler), create a script using document.createElement(), assign its src to the external script itself, and append itself as a script to the DOM. The expected behavior of the test version of DOMinatriXSS is an infinite looping of alerts while the creation of script elements with an incorrect nonce would result in termination from the loop. In its current form, the JavaScript library can compatably be included as a dependency in web applications to rewrite and protect a browser's DOM at runtime by, more effectively than before, restricting the path from user controlled inputs (URL's, form input, etc.)

to sensitive functions called by the DOM that directly execute these inputs as code. By automating the application of script nonces to scripts added by developers, the attack surface for malicious JavaScript in user-controlled sinks is limited. Next steps for further work on the JavaScript library aspect of this project should include open sourcing the defense for use as a package in popular package managers used by web application developers (NPM, PIP, etc).

The development process for the DOMinatriXSStatic implementation will be broken into three phases, (1) updating the function that creates document fragments, (2) updating the script runner object that runs all script elements on a webpage, and (3) updating DOMinatriXSS to remove disable-dynamic after injecting the meta tag enforcing CSP 1.1 with the generated script nonce. The function responsible for creating document fragments is located at "third_party/blink/renderer/core/dom/document_fragment.cc" on line 31, and the "ParseHTML" function based on the passed "parser_content_policy" on line 72 will check for whether "disable-dynamic" is specified. The script runner object will be updated to prevent scripts of nesting level higher than 0 from executing if 'disable-dynamic' is specified, the function that checks for script nesting level is located at "third_party/blink/renderer/core/script/html_parser_script_runner.cc:" on line 624. Phase (3) will require the finished implementation of DOMinatriXSS which is why I plan on developing the DOMinatriXSS JavaScript library previous to implementing the browser aspect of the defense., I hope to finish the development of DOMinatriXSStatic by the 7-8th week of the 2020 Spring Semester.

For testing the implementations of DOMinatriXSS and DOMinatriXSS, I hope to use a DOM-XSS commercial fuzzer such as DOMinator to evaluate their respective performance.

STS Topic

Food waste is a problem prevalent throughout American communities. Attempts to remedy wanton waste of food through changes to policy or marketing have great room for increased efficiency and effectiveness. A far too common adversarial attitude among Americans towards the issue of food waste has contributed to an incomplete adoption of proper food waste management. Efficient solutions such as food recycling or reuse fail to attract proper adoption due to this attitude (Dang K., 2014). An analysis and comparison between attitudes toward French and American food waste management systems illuminates ways in which each can be made more efficient (Mourad, 2016). These beg to ask the question: If there were a web platform to match food insecure organizations with excess food from around the community, would the platform attract the necessary users by circumventing the current attitude towards food recycling? Furthermore, what socio-political problems have groups encountered when attempting to redistribute excess food?

The issue of food waste should concern every American, especially after realizing Americans waste 40% of human consumed food, or 420 pounds of food per person yearly (Chen R., Chen R. 2018). The sheer amount of wasted food deals a great blow to metrics describing general community and environmental health as well as economic efficiency. Despite the fact that 1 out of 8 Americans struggle with food insecurity, the general attitude towards food waste is counterproductive to efforts to rectify the situation (Chen R., Chen R. 2018). The ease with which some Americans can purchase food has resulted in social barriers surrounding food recycling and redistribution. When using the lens of Actor-Network theory to understand how American food infrastructure contributes to these social barriers as an actor, it becomes easier to identify how the lacking care for wasted food came about. Kelsey Dang recognizes animosity after interviewing a food waste activist seeking to redistribute extra food from a Stanford campus event who was told, "You're not welcome at these events unless you act normal" (Dang K., 2014). This general animosity indicates the existence of friction between those with extra food to potentially redistribute and those who may feel ashamed to ask for assistance when managing food insecurity.

Efforts to combat food waste have arisen within the services industry, as web applications aiming to reduce local and domestic food waste enter the market. These apps struggle with an anxiety with regards to interacting with strangers which comes into play with any crowdsourcing application (Weymes M., Davis R. 2018). The applications themselves as well as the infrastructure upon which they operate must also be considered actors if a proper understanding of how they can be improved is to be reached. Research indicates that the aforementioned anxiety could be remedied if a middleman could operate within the realm of a food redistribution system. This brings to light interesting possibilities, could a non-profit leverage currently established transportation infrastructure like Uber to serve as a middleman between donors and the food insecure? Perhaps an anonymizing, recognizable, and reliable guarantor for food distribution transactions could serve to alleviate the users' worries of such a food distribution system. The introduction of an actor to combat the current attitude towards food waste could serve as a solution to the current state of ICT within the space of food redistribution, as seen in how communal recycling bins are engineered as actors to place onus on consumers to recognize and sort their own waste (Lorton J., 2015).

While nonprofits have historically stood at the forefront of food recovery and redistribution, private companies have launched platforms and applications to contribute to the effort. These private companies are often incentivized in the form of tax reductions and tax deductions. Social attitudes toward food recovery could contribute to the extent to which governments encourage these platforms, as seen in how France's incentives are some of the most generous around the world while the US lacks the same energy (Mourad M., 2016). Actor-Network theory allows tax benefits to be seen as actors within the systems by which food redistribution operate, which establishes their importance with respect to the successful adoption of food redistribution networks.

Conclusion

While the technical topic differs from the STS topic of my capstone project, they both address the issue of increasing user trust and comfort in the use of distributed web applications. The technical aspect of my project focuses on securing the digital experience of users on platforms like Google Chrome from attacks undetectable by servers. The successful integration of the technical research in open-source projects like Chromium could directly impact millions of users by limiting the current vectors for DOM-XSS attacks. For my STS topic, I investigate how domestic food redistribution efforts can be made more effective through the adoption of a web application based platform. I will need to continue to leverage community resources such as advice from members of the UVA Food Assist group. Interaction with the Charlottesville community at large will better flesh out an understanding of how a web application can increase the efficiency of local food redistribution networks and food waste reduction efforts.

References

Chen, C., & Chen, R. (2018). Using Two Government Food Waste Recognition Programs to Understand Current Reducing Food Loss and Waste Activities in the U.S. *Sustainability*, *10*(8), 2760. doi: 10.3390/su10082760

Council of Nonprofits. (2019). Washington, D.C. Retrieved from: https://www.councilofnonprofits.org/tools-resources/cybersecurity-nonprofits. 8 Nov. 2019.

Dang, K. (2014). Recover, Redistribute, and Reduce: Food Waste in the Stanford Community. *Intersect: The Stanford Journal of Science, Technology, and Society*.

Lorton, J. (2015). Technology as Scripting, Constructed and Relational: Three Narratives about Food Waste Recycling in Edinburgh. *Irish Journal of Applied Social Sciences*.

Mourad, M. (2016). Recycling, recovering and preventing “food waste”: competing solutions for food systems sustainability in the United States and France. *Journal of Cleaner Production*, *126*, 461–477. doi: 10.1016/j.jclepro.2016.03.084

Tian, Y. (2015). DOMinatriXSS: Automated DOM-Based Cross-Site Scripting Protection. *Network and Distributed Systems Security (NDSS) Symposium 2015*.

Weymes, M. and Davies, A. R. (2018). Disruptive technologies? Scaling relational geographies of ICT-mediated surplus food redistribution, SHARECITY working paper 3, Trinity College Dublin.

William, M., Das, A., Sharif, M., Bauer, L., & Jia, L. (2018). Riding out DOMsday: Toward Detecting and Preventing DOM Cross-Site Scripting. *Network and Distributed Systems Security (NDSS) Symposium 2018*.