Navigational Data Exfiltration: A Security Policy

December 5, 2016

Adrian Bjugård
bjugard@student.chalmers.se

TDA602 - Language-Based Security
EDA263 - Computer Security
EDA491 - Network Security

Kim Kling
kkim@student.chalmers.se

TDA602 - Language-Based Security
EDA263 - Computer Security
EDA491 - Network Security
DAT076 - Web applications

Keywords content restrictions, web security, security policy, http, data exfiltration, browser navigation

1 Introduction

The Internet was originally intended as nothing more than a research network, and thus the protocols used were designed around this premise. Since then the Internet has blossomed into, by far, the largest distributed network of information in existence.

When the Hypertext Transfer Protocol (HTTP) protocol emerged to become the Web, security was once again not part of the specification. Many attempts over the years have tried to add security on top of the Web, for example, the use of Transport Layer Security (TLS) diminishes the ability of malicious entities to hijack communication channels, and Content Security Policy (CSP) limits the ways that a Cross-Site Scripting (XSS) attack can exfiltrate information from an infected browsing session.

CSP, however, does not completely erase the damages of an XSS attack. If malicious content has been added to a web application (for example via some unsanitised user input), CSP might stop the attack from *silently* exfiltrating data to the attackers domain by way of a CSP whitelist. The attacker may still be able to trick a user of the web application to click a modified link, and exfiltrate the desired data that way. This is a symptom of the problem we intend to solve.

2 Context

The proposed work is within the area of web security and uses content restriction as the mechanism to provide an increased level of security. Previous work within this field is mainly the article "Reining in the web with content security policy" by Stamm et. al [1]. This has become a widely used standard in modern browsers and is in active development [2]. In the future work section of their article on the subject, Stamm et. al proposes that navigation could be limited using a similar standard, or even in the same standard as that of CSP.

3 Goals and Challenges

When we envision the concept of "perfect security" for the Web, the final product is a system where a web client can fully verify the authenticity of the resulting view, and know empirically that it is in fact the intended result of the code provided by the server. The dynamic nature of the Internet, however, makes this a difficult task to achieve. For example, web applications often include advertising that may be changed by a third party without the knowledge of the developer, causing the client output to be slightly different than what the developer expected, making attempts to verify the output difficult.

This vision can be split up into separate parts. Ensuring messages are not modified in transit between the server and the client (TLS), ensuring that client communication is kept to a pre-determined set of domains (CSP), ensuring that navigational directives do not lead to unwanted domains, and verifying that the content of a dynamic web application is as intended, the last two of which are as yet unsolved.

Our goal is to contribute to the field of web security by proposing a standard for how a web application may define to which locations a web browser may navigate.

This work faces a number of technical challenges such as ensuring that our work is properly understood and thus simple to implement. Another challenge will be to ensure that our solution, when correctly implemented, does not cause unintentional side effects with existing systems. Furthermore, we intend to build a flexible solution, allowing for multiple different modes of operation providing different levels of security.

4 Approach

Our approach consists of two major phases~~;~~ designing the standard and creating reference implementations.

In the first phase we will define how the problem should be solved, by creating a standard. This standard will have to be designed to enable smooth adoption and with a minimal impact on ~~the~~ performance~~, as well as size of~~ a web application. It should, however, not compromise our goal of increasing the security for the user and web application. We will have to decide the syntax for our scheme and take in to account how similar schemes, such as CSP, defines their language.

The second phase consist~~s~~ of implementing the solutions in both a web server and a browser. Again, in the implementation one of the primary goals is to leave a minimal impact on the performance. Among the decisions we need to make are the choice of web browser for our client side implementation and architectural choices regarding plugin versus native implementation.

References

[1] S. Stamm, B. Sterne, and G. Markham, "Reining in the web with content security policy," in *Proceedings of the 19th international conference on World wide web*, ACM, 2010, pp. 921–930.

[2] *Can i use content security policy*, `http://caniuse.com/contentsecuritypolicy`, [Online; accessed 2016-12-05].