Are You Coding Safely? A Guideline for Web Developers

Chun-Chan Cheng
Department of Computer Science and Engineering
Texas A&M University
Email: aznchat@tamu.edu

Chia-Cheng Tso
Department of Computer Science and Engineering
Texas A&M University
Email: jjjjj222@tamu.edu

I. Introduction

Since the invention of the World Wide Web (WWW) by Tim Berners-Lee, it has been drastically gaining popularity in the world in the last two decades. The WWW provides a vast source of information of almost every types, ranging from stock databases to your every day weather report. Yet, in order to access the WWW, a browser is required. So came the different browsers from different cooperates and different web applications associated with different browsers we are using today. In fact, due to the proliferation of hand-holding devices in the past five years, the ubiquitous accessibility of the web browser has been a norm for today's applications. Now basically every service supports an on demand access through all kinds of devices.

Web applications have many advantages over traditional ones which require installation. First of all, it works on every platform as long as there's a web browser. This takes makes developing web application a lot easier for engineers, since once a job done, it works everywhere. No more customization for different platforms. No more separated team members. Secondly, application can be patched instantly, which people usually ignore. No more asking for user to upgrade Last of all, web applications are fast deployable, fast prototype, light-weighted and is easy to get feed backs from the market.

Due to these advantages, tons of frameworks and languages, from server side php to client side html, emerged for web developing. With the help of new language, new framework building an website or web application becomes very easy. Anybody can build a web application by just learning the basic of these languages. On the other hand, frameworks help you fasten the production of your web application by including third party libraries and various helpers that assist you through the development process. With the help of frameworks and new languages, no more tedious work to read through all the manual script to just understand one of the languages to write a web application.

Despite the glorious future for web development, there are still many pitfall. While engineers are busy on bringing out all the functionalities, they may not have time to work on security. Which may lead to devastating consequences. Our proposal here is to provide some simple, clear but also useful guidelines to help engineer code with "good habits"

Note that our goal here is not to cover every vulnerabilities, which is intuitively impossible. Our goal here is to cover as many problems as possible with minimal efforts. One might think this is not good enough. However, the concept of security is that if the value of the data in your website is less than the effort that one needs to break in, or in other words, the attacker would need to put a time that is not worth to steal your data, then, in this case, the protection should be sufficient.

According to the "Pareto principle" , or known as the 80-20 law, most of the easy vulnerabilities should be able to eliminated, if one avoids most of vulnerability by eliminating bad coding style. It is not likely to notice every single problem without the help of scanner or static analyzer, however, by just following the guidelines of the different frameworks and languages that we propose, we believe that security breaches in the web applications may be minimized.

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.
[2] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.