An Analysis of XSS, CSRF and SQL Injection In Colombian Software And Web Site Development

Danny Alvarez E.
Student of Master in Systems Engineering
National University of Colombia - Medellín
daalvareze@unal.edu.co

Daniel Correa B.
Master in Systems Engineering
National University of Colombia - Medellín
dcorreab@unal.edu.co

Fernando Arango I.
Ph.D in Systems Engineering
National University of Colombia - Medellín
farango@unal.edu.co

ABSTRACT

Software development and web applications have become fundamental in our lives. Millions of users access these applications to communicate, obtain information and perform transactions. However, these users are exposed to many risks; commonly due to the developer's lack of experience in security protocols. Although there are many researches about web security and hacking protection, there are plenty of vulnerable websites. This article focuses in analyzing 3 main hacking techniques: XSS, CSRF, and SQL Injection over a representative group of Colombian websites. Our goal is to obtain information about how Colombian companies and organizations give (or not) relevance to security; and how the final user could be affected.

CCS Concepts

•**Software and its engineering** → **Software reliability;**

Keywords

CSRF; hacking; SQL Injection; software development; web security; websites; XSS

1. INTRODUCTION

Software development and web application usage have become very important in the modern world. However, due to the popularity of web technologies, they have become common cyber-criminals targets [9].

Many authors have addressed web application security topics, as: defense mechanisms [11], studies about security issues and vulnerabilities [5, 8, 14], security tools [7], and others. For example, in the field of data validation vulnerabilities in web applications, many techniques have been proposed to detect and prevent data input vulnerabilities, including static code analysis, prevention in the design and implementation phases, client validations and others [10]. However, websites continue being targeted and the number of vulnerable web applications is growing every day [9].

Unlike Countries like U.S., where there are universities and organizations engaged in collecting data regarding vulnerabilities such as the National Vulnerability Database (NVD), in Colombia there are very few studies about web security and there is no clear statistics about how these systems are being protected.

Because of this, we decided to test how websites from different Colombian economic sectors deal with three of the main and most common hacking techniques: Cross Site Scripting (XSS) [17], Cross Site Request Forgery (CSRF) and SQL injection [15]; a JavaScript, HTML, and Structured Query Language related attack respectively. To this purpose, we selected a group of Colombian representative websites, verified the technologies they are built on, and tested how vulnerable they are to these threats; this tests were made on march of 2015.

In the first section of this paper, we describe how this attack techniques work and what is recommended to prevent them. In the second section we present how we search for vulnerable websites, and how we did the vulnerability testing for the Colombian scenario. In the third section we present the results of our search and in the fourth section we present the conclusion and future work.

2. HACKING TECHNIQUES

Three of the main techniques used by cyber-criminals to break into websites are XSS, CSRF, and SQL injections. These attack techniques allow to access private data, steal information from the client's browser, impersonate identity and execute malicious code. These actions could put in risk not only the software integrity but also the final user information integrity.

2.1 XSS

Cross Site Scripting attacks are known as one of the main problems that web developers face in the web security field [3, 6]. XSS attacks consist in executing malicious scripts in the victim's browser using a prepared link or exploiting the website security so that the malicious code is delivered by the site itself. Exploiting this vulnerability allows to abuse the browser and steal data from it, including capturing the typed keys on the keyboard, showing non desired content and even stealing cookie's data (which can be used to supplant the client's session) and many other actions [15].

EATIS '16 April 27–29, 2016, Cartagena de Indias, Bolivar, Colombia.

DOI: 10.1145/1235

2.2 CSRF

Cross Site Request Forgery is also a mayor security threat [15] consisting in sending a malicious request to vulnerable website, usually from an authenticated client of the server trusts; the malicious request could include performing data deletion, doing transactions or changing passwords. This attack is successful due the fact that developers tend to trust that a client will never send a request he/she is not entail to or one that the GUI is not designed to dispatch. Unlike XSS attacks, that exploits the trust of the client in the website, this attack exploits the trust of the website in the client.

2.3 SQL injection

Websites use forms and URLs data input to craft the SQL sentences needed to retrieve or writing data from a database. SQL injections consist in manipulating this inputs to change the semantics of the SQL sentences. This way, an attacker could send malicious request to gain some control of the SQL sentences delivered to the database querying the database in a way different to what the developer intended. A common goal for this kind of attack is private data theft [4] or malicious manipulation of the data base's stored records.

3. AN OVERVIEW OF THE COLOMBIAN ENVIRONMENT

The task of analyzing Colombian websites was divided in stages. First, we established a mechanism to select relevant websites. Second, we collected data about the technologies used to develop those websites: programming language, additional tools or frameworks used, and what type of servers are they hosted in; this data was used to analyze and classify our results. Third, we executed an analysis of the three main hacking techniques over the relevant sites; this process was performed automatically by using a software tool.

3.1 Looking for websites

To get a good sample of the different websites in Colombia we decided to considerate the main Colombian economic sectors such as health, mining, agriculture, and so, as reported in [12]. Later we looked for the bigger companies in those fields in [13], also we checked the most visited websites in Alexa Colombian ranking [2]. Finally we also considered some websites developed by Colombian software companies. Using this method we identified 130 websites which are the base for this study.

3.2 Websites classification

We collected the following data for each relevant site:

- Name
- Economic sector
- Programming language
- Content Management System (CMS)
- Web Framework

Distinguishing this technological groups is important because they offer different security features developers can take advantage from. Note that, even when all websites are built in a programming language, not all of them are built using a Web Framework or a CMS; thus, not all of them can benefit from the same security features. Also, given that novice developers tend to naively trust their users and leave security as low development priorities, it is relevant to distinguish websites built from scratch from those that use a CMS or a Web Framework and analyze the vulnerability incidence on those groups. This being said, we can expect independent results for each group.

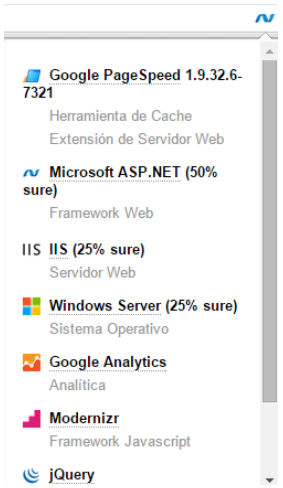

Figure 1: Wappalyzer analysis over Semana website.

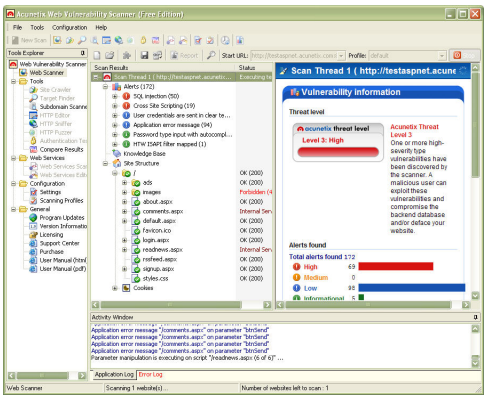

Figure 2: Acunetix reporting several web vulnerabilities.

In order to gather this information, we decided to use Wapppalyzer tool [16] available as a Google Chrome plugin. This tool uncovers the technologies used in a website, in Figure 1 we present a portion of the unveiled technologies in Semana website; note that the uncovering process is subject to a percentage of trust. The study of that percentage and means to improve it are out of the scope of this paper.

We replicated the same process over the whole 130 relevant websites and collected the previous information.

3.3 Vulnerability analysis

Testing the three vulnerabilities over the 130 relevant websites requires a lot of time and effort. Instead of a manual vulnerability analysis, we selected and used an automatic web vulnerability scanner: Acunetix Trial Edition [1]. This tool collects information about different vulnerabilities just by providing it with the website URL. Figure 2 shows an example of how Acunetix tool finds and collects vulnerabilities in a specific website.

By using this tool we were able to collect an important

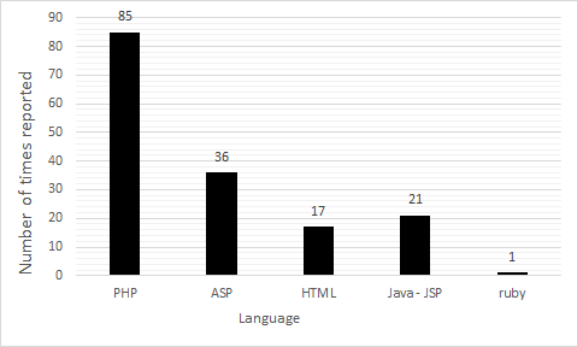

Figure 3: Programming languages reported by Wappalyzer.

amount of data. However, due to the big amount of links or sections that some sites contained, in some cases Acunetix could not complete the analysis in a reasonable amount of time, for this reason we decided to limit the execution time to two hours for every website. So, if the execution reached two hours, the test was manually stopped and the partial results were collected for further analysis.

4. RESULTS

First we identified the programming languages and the main tools used to build the websites. Figure 3 shows the number of websites that were developed in each programming language (based on the Wappalyzer analysis). This figure shows that PHP was the most popular programming language. It is important to mention that some websites could contain different section developed with different programming languages, but only the main URL address was analyzed.

Figure 4 and Figure 5 show the Framework and CMS based websites and additional tools used in their development. We found that CMSs frameworks are widely used; being Joomla the most popular. We found many Framework based websites, being Microsoft ASP the most used Framework. Note that we are distinguishing between websites built from scratch, using a Framework or a CMS and, also note, that websites built using a particular language are not forced to use a CMS or Framework. This explains why the most used Language and CMS don't correspond to the most used Framework's language; CMS, Framework or plain-language based websites are different subsets over the 130 websites that constitutes our testing sample.

Figure 6 shows that XSS, CSRF and SQL injections are present in many websites, being CSRF the most persistent security failure. Besides, XSS and SQL injections are present in almost a quarter of the total analyzed websites.

Finally, Figure 7 shows all vulnerabilities found grouped by economic sectors. It shows that no matter the sector there are many security issues, although, the education sector is the most unsecure. Similarly, Figure 8 shows that Colombian websites are equally vulnerable no matter if they are built from scratch, using a Web Framework or a CMS.

5. CONCLUSIONS

Colombian reality as we could survey in march of 2015 is far from the ideal, we found dozens of vulnerable web-

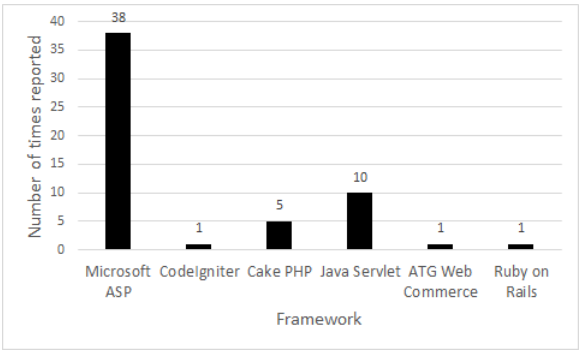

Figure 4: Web Frameworks reported by Wappalyzer.

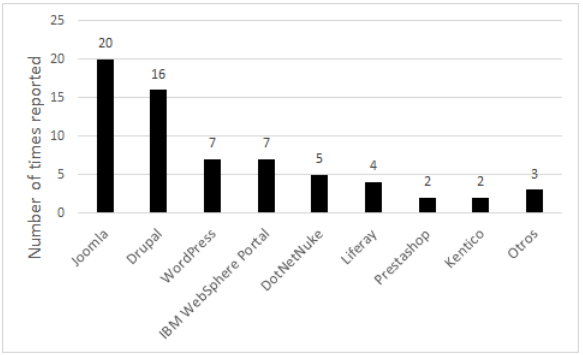

Figure 5: CMSs reported by Wappalyzer.

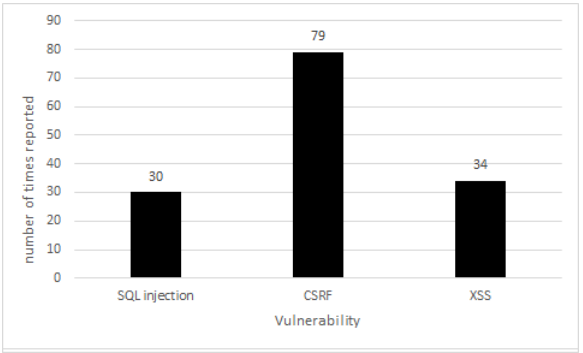

Figure 6: Website vulnerabilities reported by Acunetix.

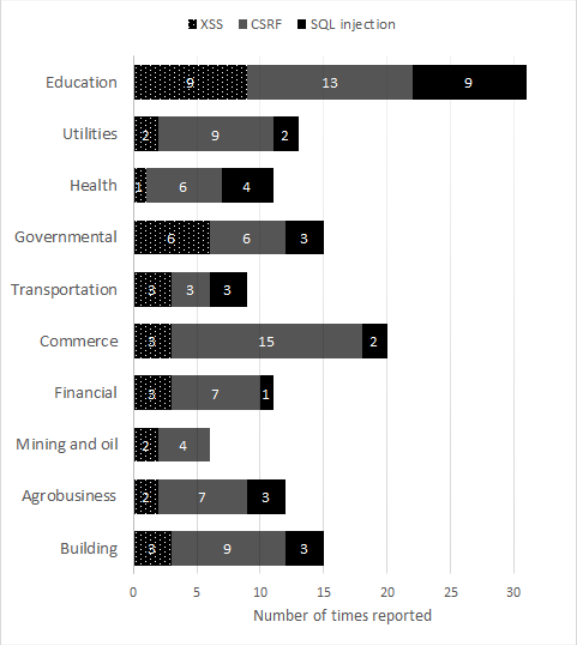

Figure 7: Vulnerabilities grouped by economic sectors.

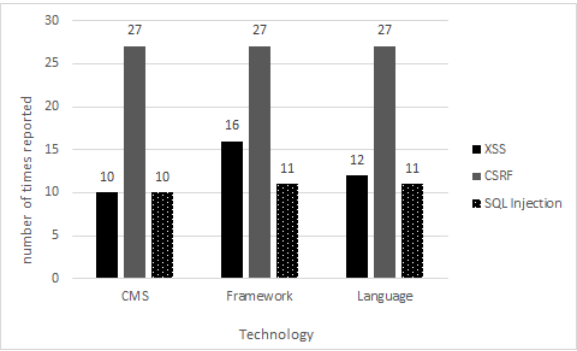

Figure 8: Vulnerabilities grouped by development approach.

sites, some of them being vulnerable to multiple basic attacks; almost an eighty percent of the analyzed websites presented at least one basic vulnerability. Today basic techniques for attacking websites remain being important security threats. Even when these techniques are well known and studied, nowadays there are many Colombian websites that still present these vulnerabilities, and it seems to be that Colombian web developers continue ignoring the protection mechanisms against these basic techniques.

Seems natural to guess that the Education sector should be the less affected, but in fact we found it has more security issues than any other. This is alarming since most websites in this sector belong to Colombian Universities in which there are programs related to computer science and software engineering.

Give proper education to software engineers and developers closer to the needs of modern industry is still an important problem to solve; the three techniques we studied are rather common and easy to patch, but it seems that developers don't know about them or just give less importance or underestimate their threat. Moreover, our results from Figure 8 raises more questions about the developers' awareness about the different security features available in CMS and Web Frameworks, as websites built from scratch seems to have the same security level; more testing is needed to answer this questions.

In the near future companies, universities and government should focus on web security and properly train developers in this field, in order to mitigate the impact of these threats and their negative impacts over the entire society. Also, we want to point out the highly importance of the collaboration between universities and modern industry as a core-way to bridge the breach between what is taught and the actual skills a professional must have to build quality applications.

As future work we will promote the inclusion of courses about web security, and we will also promote an advisory center to help Colombian Universities and developers in general to train software engineers in a better way. We will impulse our goals through a website dedicated to collect data and detect security threats.

6. REFERENCES

[1] Acunetix. Acunetix web vulnerability scanner, 2015.

[2] Alexa. Alexa - Top Sites in Colombia, 2014.

[3] D. Bates, A. Barth, and C. Jackson. Regular expressions considered harmful in client-side xss filters. In *Proceedings of the 19th international conference on World wide web - WWW '10*, page 91, New York, New York, USA, apr 2010. ACM Press.

[4] G. T. Buehrer, B. W. Weide, and P. A. G. Sivilotti. Using parse tree validation to prevent sql injection attacks. In *Proceedings of the 5th international workshop on Software engineering and middleware - SEM '05*, page 106, New York, New York, USA, sep 2005. ACM Press.

[5] A. Chaudhuri and J. S. Foster. Symbolic security analysis of ruby-on-rails web applications. In *Proceedings of the 17th ACM conference on Computer and communications security - CCS '10*, page 585. ACM Press, October 2010.

[6] V. A. Díaz. Owasp top 10 2013: actualización de los riesgos más extendidos asociados a las aplicaciones

web. *Revista SIC: ciberseguridad, segu-ridad de la información y privacidad*, pages 92–96, 2010.

[7] E. Fong and V. Okun. Web Application Scanners: Definitions and Functions. In *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, pages 280b–280b. IEEE, jan 2007.

[8] P. Roberts-Morpeth and J. Ellman. Some security issues for web based frameworks. *Communica-tion Systems Networks and Digital Signal Processing (CSNDSP)*, pages 726–731, 2010.

[9] T. Scholte, D. Balzarotti, and Kirda. Have things changed now? an empirical study on input validation vulnerabilities in web applications. *Computers & Security*, 31(3):344–356, May 2012.

[10] T. Scholte, W. Robertson, D. Balzarotti, and E. Kirda. An empirical analysis of input validation mechanisms in web applications and languages. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing - SAC '12*, page 1419, New York, New York, USA, mar 2012. ACM Press.

[11] R. Sekar. An efficient black-box technique for defeating web application attacks. *NDSS*, 2009.

[12] Semana. Balance de la economía colombiana en 2013, Economía - Edición Impresa Semana.com. http://www.semana.com/economia/articulo/balance-de-la-economia-colombiana-en-2013/369104-3. [Online; accessed 23-April-2014].

[13] Semana. Las 100 empresas mas grandes de colombia. http://www.slideshare.net/jorgeburgos100/las-100-empresas-mas-grandes-de-colombia-12916766. [Online; accessed 23-April-2014].

[14] S. Subashini and V. Kavitha. A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications*, 34(1):1–11, Jan. 2011.

[15] A. Uskov. Hands-on teaching of software and web applications security. In *2013 3rd Interdisciplinary Engineering Design Education Conference*, pages 71–78. IEEE Journal, mar 2013.

[16] Wappalyzer. Wappalyzer - identifies software on the web, 2015.

[17] J. Williams, J. Manico, and N. Mattatall. XSS (Cross Site Scripting) Prevention Cheat Sheet, 2013.