International Journal of Engineering & Technology, 7 (2.30) (2018) 6-9

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Ramification of hustling the web application development

Daljit Kaur [1] *, Dr. Parminder Kaur [2]

[1] Assistant Professor, Lyallpur Khlasa College, Jalandhar
[2] Assistant Professor, GNDU, Amritsar
**Corresponding author E-mail: jeetudaljit@gmail.com*

Abstract

With the growth of web and Internet, every era of human life has been affected. People want to make their or their organization's presence globally visible through this medium. Web applications and/or mobile apps are used for the purpose of making their recognition as well as to attract the clients worldwide. With the demand of putting the business or services online faster than anyone else, web applications are developed in hustle and under pressure by developers and most of the times they ignore the few essential activities for securing them from severe attacks, which may be a greater loss for the business. This work is an effort to understand the complex distributed environment of web applications and show the impact of husting the web development process.

Keywords: *Flaws in Development; Security; Vulnerabilities; Web Applications; Web Development.*

1. Introduction

In today's age of Internet, web applications are not only important and popular software systems but also have become an essential part of the business and its strategy. The Web Application puts business online and allows the visitors to access its most critical resources such as web server, application server, and the database server. Global commerce as its exists today would be impossible without complex distributed systems as Internet [1].
Developers of web applications spend a great deal of time on the appearance, features and functionality like any other software system. It has been seen that they give petty preference to security while developing web applications. Generally, they do not care about security as they believe that it is the task of client to ask for and moreover it can be implemented on the server using external perimeters. It is not only developers, actually project managers who lack the understanding of the security on the part of developers or lack of time dedicated to security on the part of the project managers. So, the applications are often pierced with security flaws, which are used by attackers to gain access to the assets of the web applications. Many studies show that half of the web applications have a high risk level and other shows that 80 % of the web applications have at least one critical security threats [2-4]. Securing the web applications is always a great challenge as they are at high risk [5]. Further, it has also been realized in the past that application layer security problems can be tackled during development as web application themselves are the culprit. But, still web applications are developed with vulnerabilities in them in the race of getting online faster. This research work is an effort to know the ramification of hustling the development process and ignoring the few easy but very crucial activities during development that may save web applications from severe attacks.
The paper is organized as: Section II gives the brief overview of web applications and the environment in which they are developed and run. Section III presents the work done in the same area. Section IV prepares the web applications test set for research work and gathers the result of vulnerabilities present in them. Further in section V, web applications are modified in order to fix the vulnerabilities found in them and again tested for verifying the elimination of vulnerabilities. Last section concludes the work done and proposes for the further directions.

2. Web applications and their environment

According to the Web Application Security Consortium (WASC), web application is a software application, executed by a web server, which responds to dynamic web page requests over Hyper Text Transfer Protocol (http://wwwwebappsec.org/projects/glossary/). Generally, a web application consists of scripts that reside on the web server and interact with the databases or other sources of dynamic content. Such web applications allow clients and users to access and manipulate information in a platform independent manner using the infrastructure of the web. Usually, there is web client (web browser), web server and a database server in three tier model as shown in figure 1.

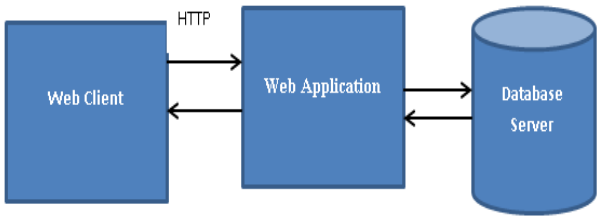

Fig. 1: Three Tiered Web Architecture.

The users of the web applications send request via HTTP protocol to the web server hosting the application using web browser like Opera, Internet Explorer, Chrome and which on turn responds with the requested page and information.
There are number of technologies used at client and server side to construct the web applications. Some broad categories of web application technologies are communication protocols, formats, server-side and client-side scripting languages, browser plug-ins,

and web server API (Application Programming Interface). The technologies used at the server side include PHP, Active Server Pages (ASP), ASP.NET, Perl and Java Server Pages (JSP),etc. While at the client side technologies like scripting languages JavaScript, VBScript, Flash can be used. In the complex environment of the web, a typical three tier model is very effective and useful to differentiate the logic of the application from its appearance i.e. front end and database i.e. back end. It makes understanding of the flow of logic of vulnerabilities in web applications easy in addition to its flexible and scalable features. The first component of this model is executed by the end users of the web applications. The second major component is the web application which contains the business logic, runs on the web server. The third component provides the access to the information stored in the database using some database management system (DBMS) like MySQL, Oracle and SQL server as per the instructions generated from the second component. All the components can have their own flaws and vulnerabilities, attackers just need one to exploit and breach the security of the system.

2.1. Issues in web applications' development

Web applications and their development are different in terms of the traditional software and their development. The development of the web applications usually involves heterogeneous stakeholders than in the development of the traditional software. Also, the web applications are executed in the distributed environment of the web which involves more technologies at client and server side along with network through which they are accessed. The prospective of the security in web applications is even more complex as they are developed under stressful environment on tight schedules for developers who have very little or almost no security knowledge. The classical scenario for the development of the web applications is like: Decision makers seek a business opportunity to provide a service via the web, so they gather a team that generates design reflecting the requirements and precedes it to the web developers and programmers. The programmers and developers are in head to head battle to develop the applications within limited resources, budget and schedule. After the completion of the development process of the web applications, testing is performed for the assurance of the quality which mainly focuses on the performance and functionality, rather than security. The nearly accomplished project might undergo final user acceptance testing to ensure that it meets the laid business requirements. Once the application runs through those tests, it is announced complete and passes to the production at the earliest possible to realize the profit. In this whole procedure of development, security is accidentally missing. There is no surprise that application layer has become the soft target of mostly attackers, as Gartner group research (http://www.gartner.com/technology/research.jsp) reported that 75% of hacks happen on web sites targeting the application layer than network, server, database, and web server layers.

3. Literature review

Li. X, et.al. In [6], have surveyed the area of web application security and presented the unique aspects of web application development. The web platform and its development is composed of large number of components and technologies including client and server side development technologies. They have identified three main security properties of web applications that should be preserve: input validity, state integrity and logic correctness. The existing vulnerabilities and attack vectors that violate these security properties also have been described. Besides, for securing web applications, based on the design philosophy existing work is categorized into three parts: Security by construction, Security by verification and security by protection.

Lomte,P. et al. in [7] have discussed different types of web application attacks including DoS, XSS, SQL injection and Request encoding. They have surveyed these attacks from the year 2012 to 2014 and also proposed countermeasures of each attack.

Chavan, S. B., et al. in [8] have described the classification of the attacks and vulnerabilities that can affect website, its data or/and its users. These vulnerabilities are classified with respect to the phase of the development life cycle in which they arise. Vulnerabilities like Broken Access Control, authentication, Improper error handling, XSS, XSRF(Cross Site Request Forgery), Information Leakage, content spoofing, buffer overflow, injection related and many others have been presented. Also the countermeasures of the vulnerabilities and their weaknesses is discussed in tabular form.

Symantec report has revealed that for the year 2016, 76% of the total scanned websites are vulnerable and it also shows that in the year 2016, more than 1.1 billion identities were stolen in data breaches, almost double the number stolen in 2015, when just over 563 million identities were stolen [9]. This is the despite the fact that the number of data breaches actually fell between 2015 and 2016—dropping from 1,211 to 1,209.

In the top ten list released by OWASP for the year 2017 in [10], Injection flaws are still leading in the web applications as in its previous years reports.

Lots of research has been done in the domain of identifying vulnerabilities in web applications, securing web applications from attacks and vulnerabilities but the gap has been realized in visualizing and demonstrating the effect of hustling the development process and vulnerabilities in web applications.

4. Original state of web applications

To know the exact status of the web applications, we have prepared a data set of 25 web applications from many fresh developers and professional freelancer developers. These web applications are then hosted on local server with XAMPP, and client machine was set up using Kali Linux operating system which provides wide variety of tools for testing the web applications for vulnerabilities and performing attacks on them. We have used OWASP Zed Attack Proxy (ZAP) security tool to find the vulnerabilities from the set of web applications on an individual basis. The results are then compartmentalized based on the type of the vulnerabilities detected in them. Broadly, we have considered two categories as Development Vulnerability (DV) and Configuration vulnerability (CV) depending upon the phase in which a particular vulnerability arises. Each vulnerability may have one or more instances, and the number of instances detected for development vulnerabilities and configuration vulnerabilities are added to get the value for DVI (Development Vulnerable Instances) and CVI (Configuration Vulnerable Instances) for respective web application.

Table1: Vulnerabilities in Original Set of Web Applications

App-name	DVI	CVI	Total VI	Vul. Categ
Apps World	23	111	134	5
Bookshop	65	81	146	6
Career	91	5	96	6
Caterer	19	144	163	5
CntrlStudio	2	156	158	4
Guidance	28	104	132	7
e-course	1	52	53	4
e-comm	49	83	132	7
enrlSys	51	24	75	7
Socialnw	478	1069	1547	7
Giftshop	148	99	247	9
Jewllery	14	74	88	7
Library	168	327	495	7
Musiclive	0	4	4	2
Taxionline	9	81	90	5
Payroll	13	16	29	5
onlineTutorial	28	86	114	7
Bakery	44	72	116	7
Shoestore	17	153	170	8
Sportskart	165	392	557	7
webFileMgr	15	136	151	7
Institute	23	80	103	6
Foodsite	741	1261	2002	8

Footwear	80	146	226	7
matrimonial	25	85	110	7

The true state of the original web applications as they are received has been presented in table 1. Also, the different number of configuration and development vulnerable instances detected in web applications are plotted in graphs shown in figure 2 and 3 respectively.

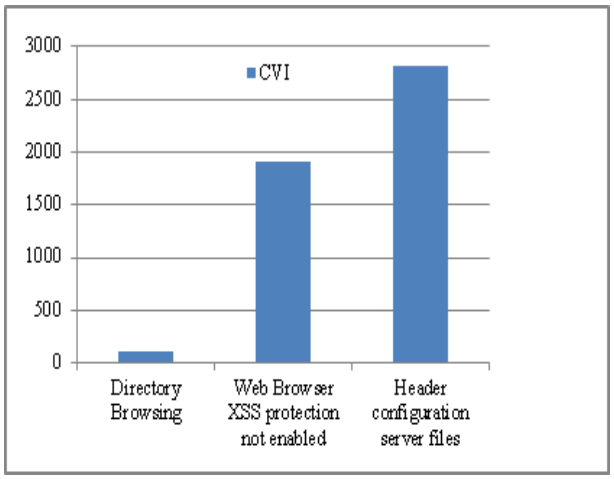

Fig. 2: Configuration Vulnerabilities.

Figure 2 depicts that the vulnerable instances detected in web applications belong to mainly three categories which includes misconfiguration of the header files in server, web browser and directory browsing. While on the other hand, there is more number of development vulnerabilities categories detected which consists of mainly Application Error Disclosure, Cookie no-Http set, SQL injection and Cross-Site Scripting(XSS) as presented in figure3. In average, every web application has more than six vulnerable categories and approximately 92 development vulnerable instances.

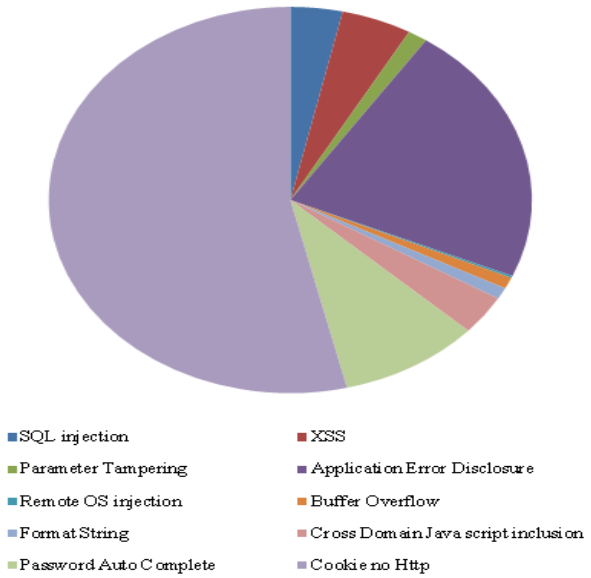

Fig. 3: Development Vulnerabilities Distribution.

5. Modified state of web applications

Now, the web applications detected with flaws are selected one by one for locating the vulnerabilities and then fixing them in order to improve their security status. Also, the time spent on each web application for performing all these activities has been noted down and every web application is again tested with ZAP under the same environment. The results obtained are organized in table 2.

Table 2: Vulnerabilities in Modified Set of Web Applications

App-name	DVI	CVI	Total_ VI	Vul. Categ	Time spent in hrs
Apps World	0	0	0	0	6.25
Bookshop	0	0	0	0	3.75
Career	1	0	1	1	6.75
Caterer	0	0	0	0	6.95
CntrlStudio	0	0	0	0	3.15
Guidance	0	0	0	0	4.25
e-course	0	0	0	0	3.05
e-comm	0	0	0	0	4.75
enrlSys	45	0	45	3	5.75
Socialnw	8	5	13	3	0.5
Giftshop	0	0	0	0	12.75
Jewllery	9	0	9	5	6.75
Library	0	0	0	0	3.75
Musiclive	0	0	0	0	2.75
Taxionline	0	0	0	0	3.25
Payroll	0	0	0	0	4.75
onlineTutorial	0	0	0	0	5.75
Bakery	0	0	0	0	6.25
Shoestore	4	0	4	4	10.25
Sportskart	67	0	67	1	4.75
webFileMgr	0	0	0	0	5.25
Institute	0	0	0	0	4.75
Foodsite	0	0	0	0	9.75
Footwear	0	0	0	0	5.75
matrimonial	0	0	0	0	4.75

The values obtained in the previous version of the web applications and modified version indicates the large drop in the vulnerable instances. This comparison is plotted in figure 4, which clearly indicates that in the hustle development of web applications, they are inculcated with large number of flaws which may risk the organization's confidential data and its reputation throughout the world.

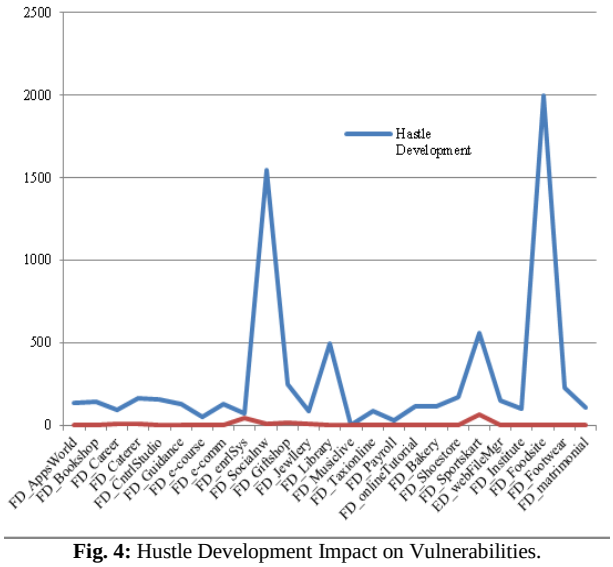

Fig. 4: Hustle Development Impact on Vulnerabilities.

When we calculated the percentage of the improvement in the eliminating the security flaws for each web application and the time spent on doing so, it has been found that in average less than six hours can make a web application 97% free from common and known flaws originated at development and configuration level. The graph depicts this truth in figure 5. Moreover, this time includes the total time spent on locating, editing and again testing the web applications. If these activities will be taken care at the initial levels of the development, even less time is required to make them better. Only very petty web applications have shown exception by little improvement as their flaws are needed to be tackled at design or requirement level of the development process.

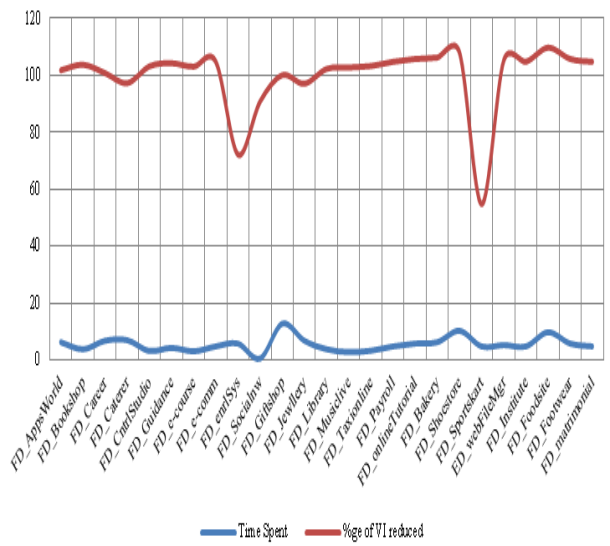

Fig. 5: Percentage of Improvement vs Time Spent.

6. Conclusion

As it is said that Precaution is better than cure; similarly developing web applications with security in mind from the very first thought can be survival for them from many serious attacks. It can be seen that hustling the development of the web applications results in flaws which sometimes become severe and difficult to eliminate after the development and risks the organization and its customers' sensitive data as well as reputation. Alternatively, if development team and the clients are provided awareness of the consequences of insecure development of web applications and the impact of flaws, it may help to contribute to the secure surfing over the web.

Acknowledgement

We are extremely thankful to Dr. Hardeep Singh, Professor, Department of Computer Science, Guru Nanak Dev University, for being constant source of support, knowledge, encouragement and suggestions that lead us to conduct this work.

References

[1] World Development Indicators 2000. World Development Indicators, Washington, DC. Retrieved from: https://openknowledge.worldbank.org/handle/10986/13828 on 13 January 2017.

[2] Web application security statistics, 2008. Web Application Security Consortium. Retrieved From: http://projects.webappsec.org/w/page/13246989/webapplicationsecuritystatistics

[3] Website Security Statistics Report ninth Edition, spring 2010. *WhiteHat Security*. Retrieved from https://www.whitehatsec.com/

[4] Lomte,P., Rajesh, M. And Bhura,S.(2013). Survey of Different Web Application Attacks & Its Preventive Measures, *IOSR Journal of Computer Engineering (IOSR-JCE)*, 14(5). 46-51.

[5] Abusaimeh, H. and Shkoukani, M. (2012). Survey of Web Application and *Internet Security Threats. International Journal of Computer Science and Network Security*. 12 (12), 67-76.

[6] Li, X., & Xue, Y. (2011). A survey on web application security. *Nashville, TN USA.* Retrieved from http://www.isis.vanderbilt.edu/node/4478 on February 1, 2014.

[7] Lomte,P., Rajesh, M. and Bhura,S.(2013). Survey of Different Web Application Attacks & Its Preventive Measures, *IOSR Journal of Computer Engineering (IOSR-JCE)*, 14(5). 46-51.

[8] Chavan, S. B., & Meshram, B. B. (2013). Classification of web application vulnerabilities. *International Journal of Engineering Science and Innovative Technology (IJESIT), 2(13)*, pp. 226-234.

[9] Symantec. (2017). Internet Security Threat Report, Symantec,(22), retrieved from: https://www.symantec.com/content/dam/symantec/docs/reports/istr-22-2017-en.pdf

[10] The Ten Most critical Web Application Security Risks, *Open Web Application Security Project* Top 10. Retrieved from http://www.owasp.org/