Web Application Security Vulnerabilities Detection Approaches: a Systematic Mapping Study

Sajjad Rafique[1], Mamoona Humayun[2], Bushra Hamid[3], Ansar Abbas[4]
Muhammad Akhtar[5], Kamil Iqbal[6]

Department of Computer Science, University Institute of Information Technology

PMAS-Arid Agriculture University
Rawalpindi, Pakistan

sajjad394@gmail.com mamoona@uaar.edu.pk bushrakiani@uaar.edu.pk

abbas.ansar514@gmail.com Muhammadakhtar58@gmail.com kamiliqbal2009@gmail.com

***Abstract*-Number of security vulnerabilities in web application has grown with the tremendous growth of web application in last two decades. As the domain of Web Applications is maturing, large number of empirical studies has been reported in web applications to address the solution of vulnerable web application. However, before advancing towards finding new approaches of web applications security vulnerability detection, there is a need to analyze and synthesize existing evidence based studies in web applications area. To do this, we have planned to conduct a systematic mapping study to view and report the state-of-the-art of empirical work in existing research of web applications. In this paper, we aimed at providing a description of mapping study for synthesizing the reported empirical research in the area of web applications security vulnerabilities detection approaches. The proposed solutions are mapped against: (1) the software development stages for which the solution has been proposed and (2) the web application vulnerabilities mapping according to OWASP Top 10 security vulnerabilities. To do this, existing literature has been surveyed using a systematic mapping study by phrasing two research questions. In the mapping study, a total of 41 studies dating from 1994 to 2014 were evaluated and mapped against the aforementioned categories.**

The outcome of this mapping study is current state-of-the-art of empirical research in web application area, strength and weaknesses of existing empirical work, best practices and possible directions for future research.

***Keywords**: Systematic mapping study, web application, security, vulnerability, State-of-the-art*

1. INTRODUCTION

Web application (WB) technologies provide a promising mechanism of integrating multiple functional components over the internet and thus enable individuals and organizations to interact each other using application program interface along huge geographical distances. Billions of individuals all over the world use WB technologies to get information, perform financial transactions, and have fun and communicate and to socialize themselves [1, 2, 3]. WB grew tremendously in the last few decades and it has brought great benefits to the people, however, these benefits are associated with some challenges and one of the most important challenges is that of security. Security in WB refers to the threat which occurs due to flaws in software design, coding, testing and implementation. WB services are more prone to cyber attacks due to their public access. Attacker or hackers sometimes breach this security by changing the original mapping of software, which in some instances causes a great loss [9, 11].

Vulnerability refers to a weakness in system's security requirement, design, coding or operation that could accidently occur or intentionally violated and result in security failure. In last few years, number of reported WB security vulnerabilities has increased. Some commonly reported WB vulnerabilities include SQL injection, cross site scripting, command line injection, cross site request forgery and malicious file execution [3, 4]. Research in the area of WB has also grown tremendously; researchers have provided various automated tools and techniques to overcome these vulnerabilities. However, before moving towards finding further new tools and techniques, there is a need to synthesize existing work to find out actual state-of-the-art of this field, to identify the strength and quality of existing proposed solutions and scope of future work. This call for taking on a systematic mapping study before doing further research in the field.

We have presented the results of mapping study to identify available solutions on web application security vulnerabilities for software system development and have categorized these solutions against: (1) software development stages in software development cycle, and, (2) well established policy principles for web application security presented in [9]. Specifically, our mapping study addressed the following research questions (1) which solutions of web application security have been proposed for software system development? (2) Can we categorize these solutions using OWASP Top 10?

The rest of the paper comprises of following sections; section II describes background and motivation, section III

978-1-4799-8676-7/15/$31.00 copyright 2015 IEEE
SNPD 2015, June 1-3 2015, Takamatsu, Japan

outlines our research methodology, section IV concludes the paper along with directions for future research.

2. BACKGROUND

The main objective of undertaking this mapping study is to identify and analyze all empirical studies that provide some approaches of detecting security vulnerabilities in WBs, aggregating these empirical studies and summarizing them for future use. Empirical studies are those studies which are based on some evidence i.e. experiment, case study, survey etc. The reason of selecting only empirical studies are that, evidence based studies is more reliable than those studies which are based upon researcher's personal opinion. Further, we have classified empirically evaluated vulnerabilities using OWASP top 10 web application security vulnerabilities identified in 2010 [17]. The reason of using OWASP top 10 for classification are many; firstly, OWASP is well known in developing security standards and processes for WBs. Secondly, The focus of OWASP is specifically on improving WB security and services through various projects, e.g. enterprise security application programming interface whose purpose is incorporating security into existing and new WBs. Finally, researchers have broad consensus over OWASP top 10 regarding the main critical security vulnerabilities of WBs [3, 4, 7, 8, 11, 12, 13, 14, 17].

Although all layers of WB are prone to security risk; but our focus in this study is mainly on the application layer. The reason of only targeting application layer level security is that, application layer level security vulnerabilities are intrinsic in WBs code, regardless of the technology of implementation or the security of web server and backend database [10]. Some common reason for the increase of vulnerabilities in application layers include: code written in easy programming language and by inexperienced users, and use of third party components [4]. According to different authenticated sources like MITRE and OWASP, 75 percent of security attacks occur at application layer [11, 12, 13].

3. THE SYSTEMATIC MAPPING PROCEDURE

For our mapping study, we followed the guidelines provided in [7, 8]. Accordingly, our mapping study was conducted in three stages. In Stage 1, we defined the scope, the search strategy and the selection criteria. In second stage primary studies were selected by applying the search strategy and the selection criteria. Lastly, in Stage 3, the selected studies are classified into different categories.

Stage 1: Defining Scope, search strategy and selection criteria

We define the scope of study as follows. The population of the study is selected as the set of articles addressing web application security. As intervention, we selected Solutions proposed in the literature for detecting security vulnerabilities of web applications development cycle (e.g., requirements engineering, design, testing, etc.). The outcome of our study is a mapping of selected solutions to web application vulnerabilities found in [9]. Our search string for conducting the research was:

("Web application" OR www OR "web service" OR "web-based application" OR "internet application" OR "World Wide Web" OR net-centric OR "web hypermedia") AND (security OR secure OR safety OR protection)

Our search resources include Science Direct, Springer Link, IEEE and ACM digital library.

If we found any reference in the primary study that is beyond our search database than we have manually added the conference proceedings/journal in which the paper was published. Further conference/workshop/Journal that specifically address web application security issue are also added in our search

To select relevant studies, we used the following inclusion and exclusion criteria.

Inclusion criteria: Only those research articles are included in our Mapping study which is based on some empirical evidence related to WB security VDM (vulnerability detection method) approaches. If same results are reported n multiple studies, then only the latest version is considered.

Exclusion criteria: Editorials, discussions, comments, workshop brief, and panels is excluded. The studies with less focus on WB security VDM or with absence of empirical evidence is also excluded.

Stage 2. Selecting primary studies

We applied the string "empirical studies on web application security" in IEEE digital library as IEEE is considered as one of the well known library, the purpose was to ensure that whether there exist enough empirical studies to conduct Mapping study and to collect some primary studies that may be used in the future for the validation of search string. We exported the abstract and citation of these studies in Endnote software [18]. We studied the abstracts of these papers and selected most relevant 10 empirical studies as the primary studies so that we may use them in the future for validation of our refined string. In first iteration, the search string was used at each resource. All references along with their abstracts were downloaded in Endnote [11] reference library. We downloaded 3,570 references. In the second iteration, abstract of all reference were read and relevant studies which explicitly addressed the web application vulnerabilities with contribution towards software system development were selected and placed in another library of selected papers. In this iteration, 140 studies were selected. We selected 90 papers from IEEE, 20 papers from ACM, 35 papers from Science Direct and 5 papers from Springerlink. In the third iteration, full texts of these 140 studies were downloaded. We read all the articles one by one and applied the inclusion and exclusion criteria and finally selected 41 studies in our third iterative phase.

We placed our 15 doubtful studies in the pending folder. In the fourth iteration, we discussed these doubtful studies and decided to accept 8 studies and to reject 7 studies. The breakdown of the results from each of the source is presented in Table 1, whereas Table 2 shows the distribution of our four iterative phases and the number of studies which were retained in each phase. In Table 1, we summarize the most relevant publication channels.

TABLE 1. NUMBER OF STUDIES AT EACH SOURCE

Resources	*No. of studies*	*No. of selected studies*	*%age*
IEEE	3420	18	0.52%
ACM	70	9	12%
Science Direct	68	14	20%
Springer link	12	0	0%
Total	3570	41	1.14%

TABLE 2. NO. OF STUDIES AT EACH ITERATION

1st Iteration	*2nd Iteration*	*3rd Iteration*	*4th Iteration*
3570	140	36	41

The IEEE Digital Library had yielded the most number of papers (3570), followed by ACM (70), Science Direct (68), and Springerlink (12). It is noteworthy that the most relevant studies were found in Science Direct (20%) and the least were found in Springerlink (0%). ACM had 12% and IEEE Digital Library had 0.52% relevant studies, respectively. As part of our inclusion criteria, we included studies from the year 2002 to 2014. For the year 2002 we did not find any relevant study. However, from the years 2003 to 2010 the number of relevant studies increased steadily with a sharp increase in the year 2010 (frequency=23). The only exception to the trend is the year 2006 where the total number was reduced to only 0. In 2010 the number was again increased to 9 studies showing a positive trend. This trend of number relevant studies per year is given in Table 3.

TABLE 3. PERCENTAGE OF STUDIES AT EACH YEAR

Years	*Relevant Studies*	*Selected Studies*	*%age*
2002	3	0	0%
2003	4	1	2.4%
2004	2	1	2.4%
2005	10	4	9.7%
2006	2	0	0%
2007	6	2	4.8%
2008	5	3	7.3%
2009	21	5	12.1%
2010	31	9	21.9%
2011	13	3	7.3%
2012	22	4	9.7%
2013	17	4	9.7%
2014	13	5	12.1%
Total	149	41	

Stage 3. Classifying selected Studies

In the next stage, we divided our studies into three categories. In the first category, the approach used in primary study was selected. The relevant studies is used for those novel techniques that have not been implemented and are validated through experiments in development environment. The selected studies is used to evaluate the techniques that have been implemented in practice. This research type explores how well the technique has been implemented. In the solution proposal either a novel solution is proposed or an existing solution is extended significantly. The *philosophical papers* propose either a conceptual framework to structure concepts into a new taxonomy. On the other hand *opinion papers* express personal opinion of the authors about a technique and the *experience papers* explain the experience of the authors of how a technique has been Implemented in practice.

TABLE 4. RESEARCH TYPE AND SOFTWARE DEVELOPMENT PHASES

Phases	*Response %age*	*Responses*
Requirement	7%	3
Design	17%	7
Implementation	66%	27
Testing	51%	21
Maintenance	0%	0
Mean: 4.439		St. Deviation: 1.537
Satisfaction Rate: 53.448	Variance: 2.363	Std. Error: 0.202

Table 4 shows the distribution of research type of the selected studies. The results of this classification are summarized in Figure 1. We also classified the studies on the basis of different stages of software development. Specifically, we grouped the software development stages into: *requirements, design, implementation, testing,* and *maintenance*. The breakdown of the classification of the selected studies is given in Table 4. The majority of selected primary studies addressed the Implementation phase of the software development (*f*=27), followed by the Testing phase (*f=21*), while some of the studies were classified under the Requirement and Design phase respectively (*f*=3&7). We did not find any study related to software Maintenance phase.

Our next categorization was based on the OWASP Top 10 [9]. Table 5 shows the security vulnerabilities from OWASP Top 10. According to OWASP Top 10 Injection vulnerability and Cross site scripting are most common with

high percentage of 68% and 49% respectively. Beside this, the vulnerabilities falling in other category also yielding 54% security threats The individuals should be aware of the nature stored data, its location and its access control policy. Similarly Cross site request forgery (CSRF) is 12% damaging vulnerability in web applications. Broken Authentication and Session management is 10% security flaw. Insecure Cryptographic Storage is detected with 5% security gap. While some of the vulnerabilities like Insecure Direct Object References, Security Misconfiguration, and Failure to Restrict URL Access are shown with 2% security threats. From the list of OWASP Top 10 the vulnerabilities such as Invalidated Redirects and Forwards and Insufficient Transport Layer Protection are not addressed in any study under our observations.

As reflected in the data shown in the table 5, we found many single vulnerability that effects the web application in number of times.

TABLE 5. DETECTION OF SECURITY VULNERABILITIES FROM OWASP TOP 10

Vulnerability Name	***Resp. %age***	***Responses***
Injection Vulnerability	68%	28
Cross Site Scripting (XSS)	49%	20
Broken Authentication and Session Management	10%	4
Insecure Direct Object References	2%	1
Cross Site Request Forgery (CSRF)	12%	5
Security Misconfiguration	2%	1
Failure to Restrict URL Access	2%	1
Invalidated Redirects and Forwards	0%	0
Insecure Cryptographic Storage	5%	2
Insufficient Transport Layer Protection	0%	0
Others	54%	22
Mean: 9.31	Std. Deviation: 6.33	Satisfaction Rate:35.47
Variance: 40.09	Std. Error: 0.691	

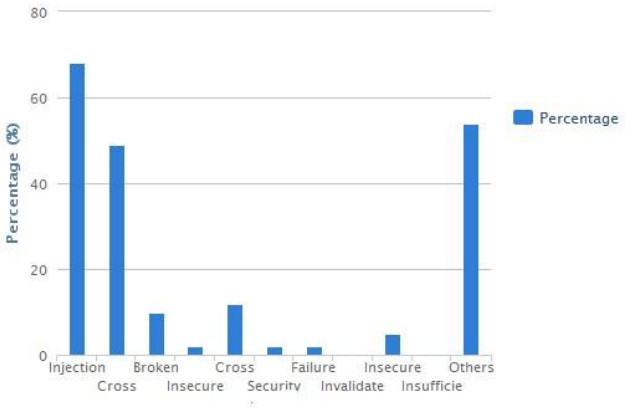

Fig. 3. Security Vulnerabilities from OWASP Top 10

4. RESEARCH QUESTIONS

RQ 1: Researchers have employed which methods to detect security vulnerabilities of web application?

The main objective of answering this question is to identify the current state-of-the-art with regard to WB security vulnerability detection approaches. To answer this question in detail we have posed following sub question

1.1. In which phases of WB Vulnerability detection method (VDM) is applied?

The process of developing WB consists of many phases, i.e., requirement, design, coding and testing etc.

The VDM is applied frequently in implementation and testing phases with percentage of 66% and 51% respectively. It is less number of time addressed in design and requirement phases which is 17% and 7% respectively. Therefore, we can extract the conclusion that most of the studies comprises on implementation phase. Similarly, there is also most of the concerns of studies on testing phase, which could more alarming situation in detection of security flaws in web applications. It is most surprising that least of the studies address the vulnerabilities in initial phases such requirements.

1.2. Which security vulnerability is frequently addressed empirically?

The mapping of studies selected against OWASP Top 10 given in table 5 and illustrated in Fig. 3, shows the most occurring vulnerabilities in web applications. The results highlight that Injection and Cross Site Scripting vulnerabilities are occurred most number of times. Similarly except OWASP Top 10 vulnerabilities, some others yielding 54% are creating security flaws. The coverage of rest of the vulnerabilities was not very encouraging.

1.3. What are the characteristics of WBs in which VDM is applied?

The answer of this question is best explained by the following table which shows data characteristics with percentage and responses. The best response is from academia 51% with

frequency of (f=21) and industrial is 37% with frequency (f=15). The response is mixed 12% with frequency (f=5) and response from Government is 5% with frequency (f=2). The response from others is very appreciable 24% with frequency (f=10).

TABLE 6. DATA CHARACTERISTICS

Data Field	*Resp. %age*	*Responses*
Academia	51%	21
Mixed	12%	5
Industrial	37%	15
Government	5%	2
Others	24%	10
Mean: 3.268	Std. Deviation: 1.672	Satisfaction Rate:38.208
Variance: 2.796	Std. Error: 0.23	

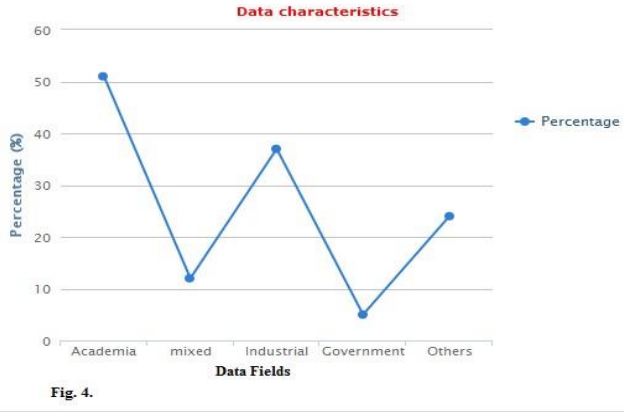

Fig. 4.

RQ 2: Can we categorize the vulnerability under discussion using OWASP top 10?

For RQ2 we have checked that whether the vulnerability under discussion fall in the list of OWASP top 10 or not. The vulnerabilities categorization is shown in table 5.

TABLE 7. EMPIRICAL VALIDATION OF VDMS

Validation Method	*Resp. %age*	*Responses*
Case study	24%	10
Experiment	61%	25
Survey	5%	2
Experience Report	12%	5
Observational Study	7%	3
Action Research	2%	1
Others	7%	3
Mean: 3.122	Std. Deviation: 1.716	Satisfaction Rate:26.871
Variance: 2.946	Std. Error: 0.245	

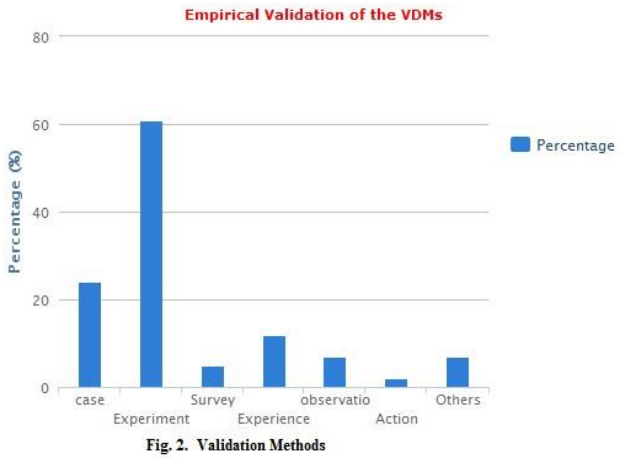

Fig. 2. Validation Methods

5. DATA EXTRACTION STRATEGY

In order to extract data from selected studies we have designed a data extraction form. Each selected paper was assigned a unique ID. First section of data extraction form consist of general question like title of publication, author's name, year of publication, reference type, and name of publisher. Then the data that is specifically related to our research questions were extracted from each study. The extracted data for research questions is as:

For RQ1 we extracted following information: phase(s) of WB in which VDM is applied (requirement, design, coding, testing, implementation), name of vulnerability that is discussed in that study, data characteristics (academia/ industrial/mixed/government).

During the process of data extraction, we recorded major terms extracted from primary studies so that we could use them in future for quality analysis and theme generation.

a. Data synthesis strategy

We used both qualitative and quantitative synthesis methods in order to analyze the extracted data obtained from selected literature. We classified our data based on security vulnerabilities of web application studied, type of empirical study, study setting, and classification according to OWASP top 10. Following data was extracted from each study,

2. Mapping of study according to software development context
3. Which are the most investigated security vulnerabilities of web application
4. Distribution of study setting according to data characteristics
5. Distribution of empirical research
6. No of studies addressing OWASP Top 10 vulnerabilities
7. Number of studies at each data source
8. Number of studies at each year.

B.Threats to validity

Below are some threats to validity of our study

Research questions: The research questions we posed in this SLR might not cover the whole web application security domain, which means that one may not be able to find the answers to the questions that concern them. Although we have tried to select the most asked and open

issues but still we cant claim that we have selected optimum questions.

We used bar graph and pie charts to depict these quantitative information. We also performed a thorough qualitative analysis of the data in order to identify certain research patterns, existing gaps and directions for future research.

Publication bias: although we have included well known libraries in our publication sources, but still there is a possibility that some relevant study is not chosen due to our access to relevant sources and restriction of English language. However, we tried to overcome this threat by following references of primary studies.

6. CONCLUSION AND FUTURE WORK

WB is a mature discipline and has long history of development and research which has its own series of conferences, journals and workshops and abundant literature has been published in this domain. WB is a commonly used and widespread interaction medium. At the same time, it faces the challenges of security vulnerabilities that endanger user's data and cause great damage. Researchers have identified various WB security vulnerabilities and their detection approaches. However, there is a lack of study which represents current state-of-the-art of empirically supported work in this area. To bridge this gap, this paper presents a systematic mapping study in order to present the current status of the field, possible gaps and directions for future research. This study will help researchers and practitioners in the area of WB to find out more mature practices and techniques, and to know the problems that need more empirical evaluation.

REFERENCE:

[1] Kaur, N., & Kaur, P. (2014). Input Validation Vulnerabilities in Web Applications. Journal of Software Engineering, 8(3), 116-126.

[2] Austin, A., Holmgreen, C., & Williams, L. (2013). A comparison of the efficiency and effectiveness of vulnerability discovery techniques. Information and Software Technology, 55(7), 1279-1288.

[3] Zhang, D., Liu, D., Csallner, C., Kung, D., & Lei, Y. (2014). A distributed framework for demand-driven software vulnerability detection. Journal of Systems and Software, 87, 60-73.

[4] Huang, C. C., Lin, F. Y., Lin, F. Y. S., & Sun, Y. S. (2013). A novel approach to evaluate software vulnerability prioritization. Journal of Systems and Software,86(11), 2822-2840.

[5] Lee, I., Jeong, S., Yeo, S., & Moon, J. (2012). A novel method for SQL injection attack detection based on removing SQL query attribute values. Mathematical and Computer Modelling, 55(1), 58-68.

[6] Corona, I., Giacinto, G., & Roli, F. (2013). Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues. Information Sciences, 239, 201-225.

[7] Balasundaram, I., & Ramaraj, E. (2012). An Efficient Technique for Detection and Prevention of SQL Injection Attack using ASCII Based String Matching.Procedia Engineering, 30, 183-190.

[8] Davanzo, G., Medvet, E., & Bartoli, A. (2011). Anomaly detection techniques for a web defacement monitoring service. Expert Systems with Applications,38(10), 12521-12530.

[9] Shar, L. K., & Tan, H. B. K. (2012). Automated removal of cross site scripting vulnerabilities in web applications. Information and Software Technology, 54(5), 467-478.

[10] Goseva-Popstojanova, K., Anastasovski, G., Dimitrijevikj, A., Pantev, R., & Miller, B. (2014). Characterization and classification of malicious Web traffic.Computers & Security, 42, 92-115.

[11] Avancini, A., & Ceccato, M. (2013). Comparison and integration of genetic algorithms and dynamic symbolic execution for security testing of cross-site scripting vulnerabilities. Information and Software Technology, 55(12), 2209-2222.

[12] Jang, Y. S., & Choi, J. Y. (2014). Detecting SQL injection attacks using query result size. Computers & Security, 44, 104-118.

[13] Shahriar, H., Weldemariam, K., Zulkernine, M., & Lutellier, T. (2014). Effective detection of vulnerable and malicious browser extensions. Computers & Security, 47, 66-84.

[14] Scholte, T., Balzarotti, D., & Kirda, E. (2012). Have things changed now? An empirical study on input validation vulnerabilities in web applications. Computers & Security, 31(3), 344-356.

[15] Woo, S. W., Joh, H., Alhazmi, O. H., & Malaiya, Y. K. (2011). Modeling vulnerability discovery process in apache and iis http servers. Computers & Security, 30(1), 50-62.

[16] Shar, L. K., & Tan, H. B. K. (2013). Predicting SQL injection and cross site scripting vulnerabilities through mining input sanitization patterns. Information and Software Technology, 55(10), 1767-1780.

[17] Wang, S., Gong, Y., Chen, G., Sun, Q., & Yang, F. (2013). Service vulnerability scanning based on service-oriented architecture in Web service environments.Journal of Systems Architecture, 59(9), 731-739.

[18] Awoleye, O. M., Ojuloge, B., & Ilori, M. O. (2014). Web application vulnerability assessment and policy direction towards a secure smart government. Government Information Quarterly, 31, S118-S125.