A Mapping Study of Security Vulnerability Detection Approaches for Web Applications

Karishma Rahman
Gianforte School of Computing
Montana State University
Bozeman, MT, USA
karishma.rahman@student.montana.edu

Clemente Izurieta
Gianforte School of Computing
Montana State University
Idaho National Laboratories
Bozeman, MT, USA
clemente.izurieta@montana.edu

***Abstract*—For the last few decades, the number of security vulnerabilities has been increasing with the development of web applications. The domain of Web Applications is evolving. As a result, many empirical studies have been carried out to address different security vulnerabilities. However, an analysis of existing studies is needed before developing new security vulnerability testing techniques. We perform a systematic mapping study documenting state-of-the-art empirical research in web application security vulnerability detection. The aim is to describe a roadmap for synthesizing the documented empirical research. Existing research and literature have been reviewed using a systematic mapping study. Our study reports on work dating from 2001 to 2021. The initial search retrieved 150 papers from the IEEE Xplore and ACM Digital Libraries, of which 76 were added to the study. A classification scheme is derived based on the primary studies. The study demonstrates that vulnerability detection in web applications is an ongoing field of research and that the number of publications is increasing. Our study helps illuminate research areas that need more consideration.**

***Index Terms*—web application, security vulnerability, systematic mapping study**

I. Introduction

The Web substantially influences all aspects of our everyday social lives nowadays. Billions worldwide use different web applications to get information, play games, communicate, execute financial transactions, and socialize. Thus, they allow people and organizations to communicate utilizing different applications regardless of the potentially substantial geographical distances. Though this technology has brought numerous advantages to our lives, they also come with various challenges. The most significant challenge is the security of web applications [7]. Security in web applications refers to threats because of the unstructured designs of the software, inadequate testing, and poor coding.

Vulnerabilities are manifestations of weaknesses in a system. They can occur accidentally because of the carelessness of the system designer and can cause failures in the security of the system [1]. Over the past few years, vulnerabilities in applications have been continuously increasing, and most of the common vulnerabilities found include SQL injection, cross-site scripting, broken authentication, command-line injection, and identification and authentication failures [1]. Many researchers nowadays investigate these vulnerabilities and develop different automated techniques and tools to overcome these vulnerabilities [2]. Researchers have suggested numerous methods for detecting security vulnerabilities of web applications for the past decade, and paper numbers in this area are increasing. Systematically identifying, interpreting, and classifying the publications is essential to present a summary of the specific domain's trends. Therefore, a systematic mapping study is needed.

Petersen et al. [3] stated that a systematic mapping study is used to examine, categorize and structure articles of particular research areas in software engineering. The objective of the mapping study is to acquire knowledge of a research area through classification. We follow the recommended five steps, which include defining research questions, searching for suitable papers, stating selection criteria, extracting data, and mapping. The main contributions of this paper include *i)* a classification scheme for categorizing the articles, and *ii)* a systematic mapping study that consists of related research over the past 20 years (2001-2021) by analyzing 76 articles.

II. Background and Related work

Verification techniques in current web development practices are either incomplete or erroneous, which introduces vulnerabilities to web applications. In turn, vulnerabilities allow a malicious user to introduce harmful artifacts (e.g., via script injections, data flow attacks, and input validation attacks) into web content [1]. These harmful artifacts include cross-site scripting, directory traversal, SQL injection, response splitting, and filename inclusion.

Prior studies have attempted to synthesize web application vulnerability detection. Specifically, Alalfi et al. [4] present a survey that uses 24 different modeling techniques in web verification, validation, and testing. The survey classifies, contrasts, and examines the modeling techniques. Marin et al. [5] provide a brief overview of current testing techniques for web applications. They discuss the limitations of these techniques for testing web applications. Garousi et al. [6] developed a method for classifying papers in the testing of web applications. Their paper is the first systematic mapping study in web application testing. A systematic mapping study of functional testing is conducted, which analyses

79 papers. Rafique et al. [7] synthesize empirical studies in web application vulnerability detection approaches. Their findings correspond with the software development steps, and the vulnerabilities correspond to OWASP's Top 10 security vulnerabilities. Li et al. [11] include static, dynamic, and hybrid analyses in their study. Deepa et al. [12] concentrate on detecting and preventing attacks targeting injection and logic vulnerabilities. Chang et al. [13] describe two web-based malware detection methods, i.e., virtual machine-based and signature-based detection. A comprehensive survey by Gupta et al. [14] describes emerging web application weaknesses, avoidance mechanisms, detection, and attack patterns for all critical web threats in OWASP 2013. A survey by Seng et al. [15] describes web application security scanners and their qualities. Finally, Atashzar et al. [16] survey the web application security features, where features include critical vulnerabilities, hacking tools, and approaches at a high level.

The studies mentioned above have various weaknesses which restrict replication, generalization, and usability. Some studies are conducted without any systematic approach for reviewing the papers. Further, the selection criteria of some studies are not explicitly described, making it impossible to reproduce results. In our mapping study, we mitigate these shortcomings.

III. Method

This study is conducted following the guidelines for systematic mapping suggested by Petersen et al. [3].

A. Goal and Research Questions

The study identifies, examines, and synthesizes the research articles published in the last twenty years in web application vulnerability detection. This mapping study addresses the following research questions:

RQ 1– How many papers introduce methods/techniques, tools, models, frameworks, comparison analysis, or processes? The first question identifies the type of contribution made [8].

RQ 2– What are the research methods used in the papers? Petersen et al. propose the following research methods in their systematic mapping guideline- (1) solution proposal, (2) experience papers, (3) evaluation research, (4) validation research, and (5) opinion papers.

RQ 3- What are the testing techniques presented in the papers? The testing techniques include generating test cases, using scanners, injecting faults, etc.

RQ 4– How many approaches are manual versus automated that detect vulnerabilities of web applications?

RQ 5– How many approaches are evaluated on dynamic versus static web applications?

RQ 6– How many papers propose a working detection tool? What are the names, and how many are freely available for use?

RQ 7- What are the common security vulnerabilities in web applications found on those papers?

RQ 8– What is the annual number of publications or the publication rate in this field?

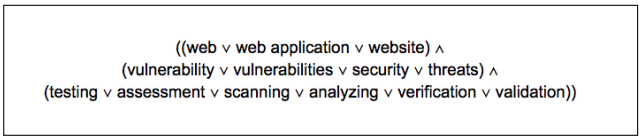

Fig. 1. Formulated search query for the selection of the relevant articles

RQ 9– What are the citation rates of the papers in this area?

B. Paper selection strategy

We mined IEEE Xplore[1] and the ACM Digital Library[2]. Papers published between 2001 and 2021 are included in the pool of papers. Search keywords have been identified using the PICO (Population, Intervention, Comparison, and Outcomes) technique, which is suggested by Kitchenham and Charters [9] to formulate search strings from research questions. The identified keywords are web application, vulnerability, and detecting/testing, which are grouped into sets. We formulate the search string along with their synonyms as shown in Figure 1.

C. Exclusion and inclusion criteria

Articles are selected based on the titles and abstracts, keywords, and reading of the evaluation section as suggested in [8]. Both authors reviewed each article to increase reliability. Full-text reading of the paper is taken into account only when in doubt. The inclusion criteria applied to the collection of titles and abstracts required that *i)* research articles were based on empirical evidence related to vulnerability detection methods of web applications, *ii)* that if multiple studies were reported by the same author with the same result, only the latest study was considered, and *iii)* that studies were published from 2001 to 2021. Summaries, editorials, non-peered reviewed studies, studies in other languages, and books and magazines were excluded.

After applying the selection criteria to 150 papers, the collection size decreased to 76. The list of 76 papers can be found in the online repository [18].

D. Classification Scheme and Data Extraction

A classification scheme is also known as a systematic map [3], and Table I shows how each attribute maps to a research question. The classification scheme is created iteratively while collecting the data. After developing the classification scheme, the papers are then classified using the scheme. The online repository records the publications numbers in each classification [18].

IV. Results of the Mapping

Herein, we address each research question.

RQ 1– Figure 2 shows the distribution of the papers by the type of contributions for the 76 papers in the study. Some papers are classified under more than one type based on their

[1]http://ieeexplore.ieee.org
[2]http://dl.acm.org

TABLE I
CLASSIFICATION SCHEME

Attributes	Research question
Contribution type of the paper	RQ 1
Research type of the paper	RQ 2
Type of testing activity/technique	RQ 3
Manual versus automated approach	RQ 4
Static web application versus dynamic web application	RQ 5
Presented tools in the papers	RQ 6
Vulnerability type addressed	RQ 7
Publication year	RQ 8
Number of citations	RQ 9

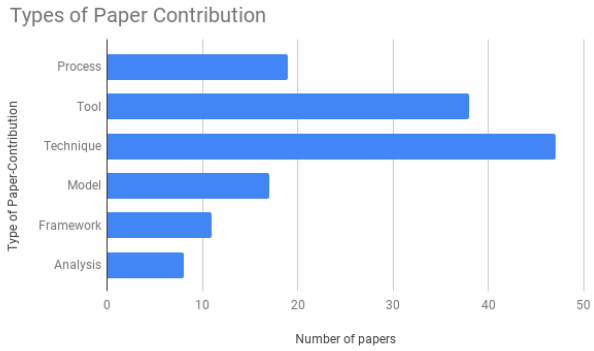

Fig. 2. Types of contribution

contributions. For example, paper number 10 [18] made two contributions: (1) a test method (Metamorphic testing based approach), and (2) a test tool called SLMR.

RQ 2-Figure 3 illustrates papers by research facet. The research in web application vulnerability detection is dominated by solution proposals (29 papers: 38.2%) and validation studies (22 papers: 28.9%).

RQ 3- Figure 4 displays the distribution of various testing techniques used in the papers. The ratio of this category is comparatively spread out among the types of techniques.

RQ 4- Testing automation is a known research concern [4]. Forty-one papers describe full automation, while eleven papers are fully manual. The remaining papers use both.

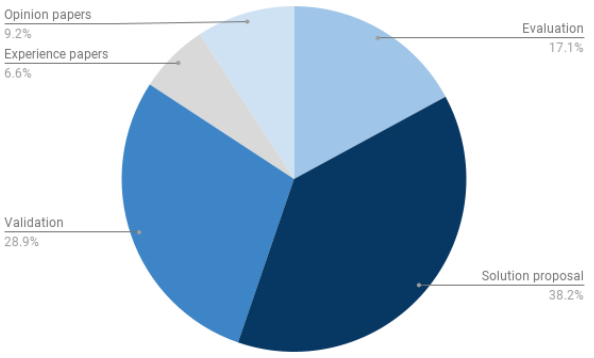

Fig. 3. Types of research paper

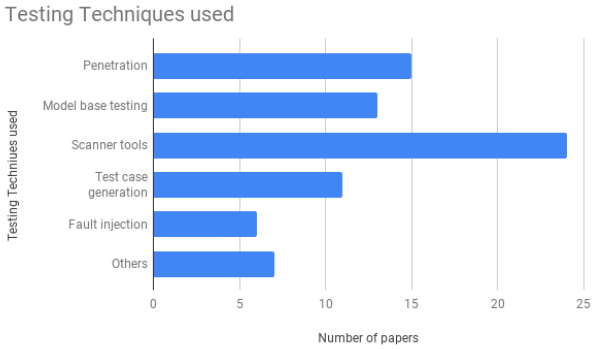

Fig. 4. Types of testing techniques

RQ 5- Sixty-seven papers analyze dynamic web applications, and nine analyze static websites. The testing of dynamic web applications is more widespread.

RQ 6- Eighteen papers describe tools. Some tools include SMLR (paper no. 10), Escrow (paper no. 24), DEKANT (paper no. 60), and MoScan (paper no.74) [18].

RQ 7- We categorized vulnerabilities based on the OWASP Top 10[3] shown in Figure 5. Our results suggest that SQL Injection vulnerability and Cross-site scripting are the most common, with 51 and 45 counts respectively. Besides this, Broken Access Control can be found in 38 papers, and Identification and Authentication Failures can be found in 23 papers. Other vulnerabilities from the OWASP Top 10 can be found in some papers. Others vulnerabilities that are not included in the top 10 list can be found in 29 papers.

RQ 8- Figure 6 shows the publication trend of the studies. We observe that the number of papers is higher in 2010, 2015, and 2016. However, in the years between, paper counts were somewhat lower. A decreasing trend in publications is observed science 2016.

RQ 9- Citation data is extracted from Google Scholar (August 2021). Figure 6 visualizes the counts. We observe that the papers from 2006 to 2016 have more citations than the earlier and later papers. This trend is very common as the papers from 2020-2021 are comparatively newer. This result also helps to reveal the top-cited papers. Paper 46 [18] is the top cited paper with 111 citations.

V. DISCUSSION, CONCLUSION AND FUTURE WORK

Most papers propose new vulnerability detection techniques or improve existing vulnerability detection techniques implemented for web applications. Our study indicates that the most common vulnerabilities in web applications can be found on the OWASP top ten vulnerability list, consistent with expectations. There is an enormous scope for future research in this area, which suggests that the number of papers in this domain will likely increase. Also, although many papers suggest different techniques, only a few tools can be downloaded. This is a scary reality that the research

[3]https://owasp.org/www-project-top-ten/

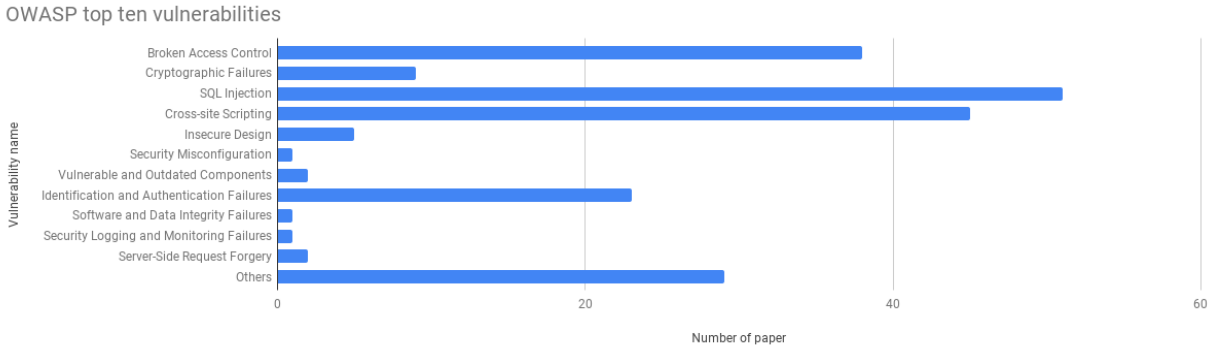

Fig. 5. Detection of security vulnerabilities from OWASP top 10

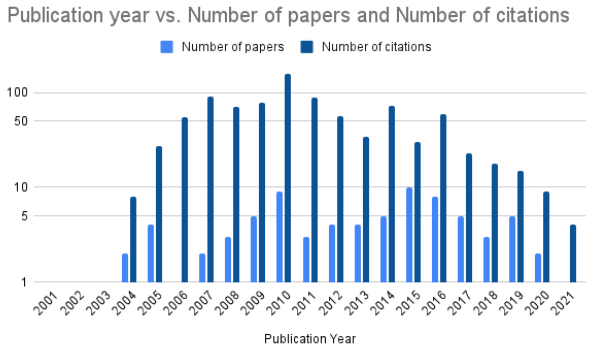

Fig. 6. Publication trend per year & citation count vs. publication year.

community must address if they want to influence industry practitioners.

Some testing techniques depend on an implicit test oracle. An implicit oracle depends on implicit knowledge, differentiating between the right and wrong behavior of the system [10]. Despite there being many proposed security testing approaches, the oracle problem is still not appropriately addressed. This opens up various avenues for future studies.

Potential threats to the validity of this study were studied according to Wohlin et al. [17]. The search string poses an internal validity threat; however, this was mitigated by using a known construction technique. Other threats include selection criteria and author judgments which were mitigated by agreements. Conclusions are directly traceable to the data sets associated with each research question.

In conclusion, the web applications vulnerability detection domain has a long history of development and research. This paper delivers a systematic mapping study that shows current trends. Also, it presents potential gaps and suggestions for prospective studies to help bridge this gap. The study focuses on the last 20 years. It analyzes 76 relevant selections while providing a classification scheme by examining the primary studies.

REFERENCES

[1] N. Kaur and P. Kaur, "Input Validation Vulnerabilities in Web Applications", Journal of Software Eng., vol. 8, no. 3, pp. 116-126, 2014.

[2] A. Austin, C. Holmgreen and L. Williams, "A comparison of the efficiency and effectiveness of vulnerability discovery techniques", Information and Software Technology, vol. 55, no. 7, pp. 1279-1288, 2013.

[3] K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, Systematic mapping studies in software engineering. 12th International Conference on Evaluation and Assessment in Software Engineering (EASE), 2008, pp. 71–80.

[4] M.H. Alalfi, J.R. Cordy, T.R. Dean Modelling methods for web application verification and testing: state of the art Software Testing, Verification and Reliability, 19 (2009), pp. 265-296

[5] B. Marin, T. Vos, G. Giachetti, A. Baars, P. Tonella Towards testing future web applications Proc. 5th International Conference on Research Challenges in Information Science (RCIS), IEEE (2011), pp. 1-12

[6] V. Garousi, A. Mesbah, A. Betin-Can, S. Mirshokraie, A systematic mapping study of web application testing, Information and Software Technology, Vol. 55, Issue 8, 2013, PP: 1374-1396, ISSN 0950-5849

[7] S. Rafique, M. Humayun, B. Hamid, A. Abbas, M. Akhtar and K. Iqbal, "Web application security vulnerabilities detection approaches: A systematic mapping study," 2015 IEEE/ACIS 16th Intl Conf. on SE, AI, Net and Parallel/Distributed Computing (SNPD), 2015, pp. 1-6.

[8] K. Petersen, S. Vakkalanka, L. Kuzniarz, Guidelines for conducting systematic mapping studies in software engineering: An update, Information and Software Technology, Vol. 64, 2015, PP: 1-18, ISSN 0950-5849

[9] B. Kitchenham, S. Charters, Guidelines for Performing Systematic Literature Reviews in Software Engineering, Tech. rep., Technical report, EBSE Technical Report EBSE-2007-01, 2007

[10] P. X. Mai, F. Pastore, A. Goknil and L. Briand, "Metamorphic Security Testing for Web Systems," 2020 IEEE 13th Intl. Conf. on Software Testing, Validation and Verification (ICST), 2020, pp. 186-197.

[11] X. Li and Y. Xue. 2014. A survey on server-side approaches to securing web applications. ACM Comp. Surv. 46, 4, Article 54 (March 2014).

[12] G. Deepa and P. S. Thilagam. 2016. Securing web applications from injection and logic vulnerabilities: Approaches and challenges. Information and Software Technology. 74 (June 2016), 160-180.

[13] J. Chang, K. K. Venkatasubramanian, A. G. West, and I. Lee. 2013. Analyzing and defending against web-based malware. ACM Comput. Surv. 45, 4, Article 49 (August 2013), 35 pages.

[14] S. Gupta and B. B. Gupta. Detection, Avoidance, and Attack Pattern Mechanisms in Modern Web Application Vulnerabilities: Present and Future Challenges. Intl. Jnl of Cloud Apps and Comp. 7, 3(2017), 1-43.

[15] L. K. Seng, N. Ithnin and S. Z. M. Said. 2018. The approaches to quantify web application security scanner quality, a review. International Journal of Advanced Computer Research. 8, 38(2018).

[16] H. Atashzar, A. Torkaman, M. Bahrololum and M. H. Tadayon. 2011. A Survey on Web Application Vulnerabilities and Countermeasures. In Proceedings of 6th Intl. Conf. on Computer Sciences and Convergence Information Technology (ICCIT), 647-652.

[17] C. Wohlin, P. Runeson, M. Hst, M. C. Ohlsson, B. Regnell, and A. Wessln, Experimentation in Software Engineering. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[18] K. Rahman, "Paper_selection," Excel spredsheet, [Revised June. 2022]. Available: https://docs.google.com/spreadsheets/d/1H_C7m0sVKU4gnJNzGKvTPr0jDikaHuDfkkDzIQ7qXCg/edit?usp=sharing