COMP 4110 – Three Amigos, Group 3
Critical summary/contribution
Name: Charles Corro, Keerthana Madhavan, Het Patel

Paper 1:
Title: Online Penetration Testing for Web Application Based on OWASP Framework
Year Published: 2021
Summary:

With many applications no longer needing to be installed into your local computer and being accessible through the web, the security risks for such applications need to be tested ever more strictly for potential vulnerabilities. This paper describes the process and details of creating a functioning web-based penetration testing application to find vulnerabilities within other web applications. The test follows the Open Web Application Security Project, or OWASP for short, Top 10. The OWASP Top 10 are the ten most popular security risks in web applications based on research from this international not-for-profit organization. The web application uses Django as a framework and mainly runs on python. Along with the use of python celery and Redis, this application uses python due to its flexibility and accessibility with many libraries and functions within the python language. Overall, the application will run several tests on a web application, then calculate a grade along with a quick description on the vulnerability of the web application.

Overall, this paper is a very informative paper, since it is very similar to what our team wishes to accomplish by using the OWASP Top 10 on web applications. This paper will help as a guideline and framework that will help our project in accomplishing our goal. Though some of the tests done in the paper are no longer a part of the OWASP Top 10 to this day due to the ever-changing environment web security is in, we will use this paper as a reference and a validation that the process to make our project is possible.

IEEE Citation:
J. Y. Zhong and M. M. Siraj, “Online Penetration Testing for Web Application Based on OWASP Framework.” Universiti Teknologi Malaysia, Johor, Malaysia, 2021.

Paper 2:

Title: Security testing of web applications: A systematic mapping of the literature
Year Published: 2021
Summary:

The purpose of this paper is to collect, analyze, and draw conclusions of where web application testing is headed. The reason for this purpose is to help not only experienced researchers but also new researchers understand a basic grasp on security on web applications and the direction of it in the future. Overall, this paper takes the process of finding 150+ articles on security on web applications, then further filtering and refining the pool to 80 technical articles to base their analysis and findings on. As a result, the findings found included that most research in the field provided methods and approaches to security for web applications, which accounted for 49 out of 80 papers. 12 of the 80 papers provided tools for implementing approaches within the field for easier implementation, and 8 out of 80 spoke of empirical results. Other categories found included: Framework to web security, Processes, and models for secure web applications.

Overall, I think that this paper would bring a good insight into the direction security for web applications are going through and the reasons as to why certain tests are better than others for finding vulnerabilities within web applications. I also think that this paper gives a good basis on the type of information out there on web applications and the testing that is done on them to provide more safety and security.

IEEE Citation:
Murat Aydos, Çiğdem Aldan, Evren Coşkun, Alperen Soydan,
Security testing of web applications: A systematic mapping of the literature,
Journal of King Saud University - Computer and Information Sciences,
2021,

Paper 3:

Title: Analysis of Security Testing Techniques
Year Published: 2021
Summary:

With more and more applications becoming more accessible by being hosted on the web, the risk factor of potential data leaks or data compromises arises. This paper wishes to survey the current topics within the field of security testing techniques. This paper analyzes the papers within multiple academic databases to find different testing techniques, then categorizes and analyses each technique. From this paper, 20 security testing techniques were identified and categorized into 3 levels, with 1 being the highest nodes, and branching off into many leaves in levels 2 and 3. After categorizing this paper proceeds to discuss the techniques and their risk base. The techniques analyzed include black box, white box, gray box, and others such as penetration testing, ethical hacking, etc. This paper proposes a new taxonomy for the classification of security testing techniques and a better way of organizing them using their 3-level method. This paper analyses a total of 292 papers relevant to security testing.

Overall, this paper is very insightful, it includes potential risks and benefits of certain testing techniques and some insight into the use cases. With the new taxonomy for the classification of security testing, the paper will benefit our project is proceeding to help us correctly categorize tests done on a web application and their severity. This paper will also be a good resource to cite diagrams for types of testing and their categories.

IEEE Citation:
O. B. Tauqeer, S. Jan, A. O. Khadidos, A. O. Khadidos, F. Q. Khan et al., "Analysis of security testing techniques," Intelligent Automation & Soft Computing, vol. 29, no.1, pp. 291–306, 2021.

Paper 4:

Title: Systematic review of web applications security development model
Year Published: 2013
Summary:

When it comes to web application security, no one development model is considered the standard model. Most vulnerabilities on the web exist because of the inappropriate software development approaches used by software development teams. The authors in this paper conduct a systematic literature review on secure software development lifecycle papers from prestigious publications such as IEEE, ACM, Science Direct, Scopus, etc. The purpose of this review is to investigate the various security development models used to secure the web application layer, the approaches researchers took to the stage security testing, and the tools used to detect vulnerabilities.
The authors suggest a need to investigate the available development model's security mechanism to determine the best technique for securing the web application layer. In their literature review, 43 papers were selected from 499 reports using various methods such as the PICO paradigm and a set of defined research questions. From the 43 papers, the researchers identified three clusters of studies: security development models, lifecycle where security is emphasized, and security tools and mechanisms for detecting vulnerabilities. They suggested that based on their empirical study, threat modeling was used by most researchers to ensure security in the web application layer. Some of the studies also considered putting security checks within the requirement gathering and system design, while others deemed security checks in all stages of software development. However, the authors recommend that assessing vulnerabilities in code is essential and produces significant results. Most researchers used agile methods for securely developing software such as extreme programming, feature-driven development, scrum, etc. However, more than 67% of the study use a threat modeling approach for tightened security in software development, but I feel like it differs based on business logic.
Finally, the authors recommend practitioners emphasize code vulnerability assessment using various tools to detect loopholes along with other security mechanisms.

IEEE Citation:

Musa Shuaibu, B., Md Norwawi, N., Selamat, M.H. *et al.* Systematic review of web application security development model. *Artif Intell Rev* 43, 259–276 (2015). https://doi.org/10.1007/s10462-012-9375-6

Paper 5:

Title: Research on Software Testing Techniques and Software Automation Testing Tools.
Year Published: 2017
Summary:
With the advancement of technology and software globally, developing reduced or bug-free applications is necessary. Software testing and its various verification techniques and methods are needed before the software goes into production. The researchers argue that Automation Testing has made a significant impact in effectively identifying bugs and errors with less human resources and time. Automation testing contains various test cases that can easily capture different scenarios. In this research paper, the authors explore different types of testing techniques and compare various tools and compare them against automation testing.
The automation testing process includes a test plan, test design, execution of tests, and test evaluation and analysis. There is 2 type of automation tests: testing based on code and testing based on the graphical user interface. From various literature reviews, the researcher concluded that automation testing will increase efficiency and improve the coverage of software code testing and that the Selenium framework is the best for web application testing.
The researchers discuss various testing strategies, including unit testing, integration testing, system testing, and acceptance testing. Some of the methodologies they discuss include white, gray, and black-box testing. The researchers classify testing tools into different categories, including test management, functional, and load testing. Test management tools allow for streamlining the testing procedure, functional testing tools such as selenium allow testers to write tests for major programming languages.
Finally, the researchers concluded that for any software product, testing is essential, and the automation process is efficient and covers the hidden needs in testing. Many testing tools are developed and used by the industry, but the evolving nature of software, security issues, and vulnerabilities require us to adapt to these changes. This statement goes hand in hand with our problem of developing automated testing tools for web application security testing tools.

IEEE Citation:

K. Sneha and G. M. Malle, "Research on software testing techniques and software automation testing tools," 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), 2017, pp. 77-81, doi: 10.1109/ICECDS.2017.8389562.

Paper 6:
Title: A Web Testing Platform Based on Hybrid Automated Testing Framework
Year Published: 2019
Summary:

With Web Applications systems increasing globally, the need for automated testing has also been growing. However, automated testing has many challenges because of the unique nature of different business solutions, test data differentiation, and test case explosion. Automation testing ensures high efficiency, saves costs, and provides high-quality iterations in stress testing, functional testing, performance testing, and regression testing of web systems. Large numbers of testing scenarios of GUI and API-based testing require a new and different approach-based testing framework that supports high concurrency. This research paper proposes a hybrid model using a keyword-driven and data-driven testing framework.
The authors' automated testing framework provides various tools and reusable and automated test modules for test execution and management. Keyword-driven testing is an application-independent framework that needs data tables and keywords to drive the test script code. The data-driven test requires input and output data and significantly needs less code to generate many test cases. The authors want to combine these approaches and produce a hybrid framework consisting of a management layer, engine layer, and third-party test tools.
In web applications, we test requirement analysis, data preparation, test case generation, and result analysis. Test Requirements finds out the similar business model in the whole system. The XML file processes specific formats related to the business logic in the test data preparation phase. In test case generation, the authors use the business logic to generate test cases and scripts with a mapping method. Finally, the result analysis process will check the output with expected output and provide data for the product quality evaluation of the tested software.
Overall, the authors produce a novel framework that analyzes the survivability of large-scale network and complex system testing. They also validate the effectiveness of their framework with a case study. Our paper also considers using business logic and different modules for security testing of web applications using the OWASP framework.

IEEE Citation:

Z. Sun, Y. Zhang and Y. Yan, "A Web Testing Platform Based on Hybrid Automated Testing Framework," 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2019, pp. 689-692, doi: 10.1109/IAEAC47372.2019.8997684.

Paper 7:

Title: Web and Mobile Applications' Testing using Black and White Box approaches
Year Published: 2019
Summary:

Software testing ensures that the coverage of all the requirements and specifications are met by the software. It provides necessary feedback generated by the test cases by detecting errors. Factor covered to choose the right testing strategies: AI, Security focused, fully automated, Heuristics search. General automated testing strategies include search-based software testing for web application testing, Sania, and parse tree method for preventing SQL injections. Search-based automated testing showed a reduction of test efforts by 30%. Sania had shown promising efficiency compared to other vulnerability scanners. The search tree method showed effectiveness to prevent SQL injection attacks. Another testing method uses session data which proved to be effective in finding bugs in web applications. Black-box testing, and white-box testing are some of the common strategies used for testing. Black-box testing is done keeping in mind the business requirements of the clients. Black-box testing used in mobile app testing helps automatize the creation of test cases. For Android, a tool called B Box-Tester is used; it provides detailed information of the software security with high accuracy. A method used in automating the black-box testing is called Auto-black-test. It starts by auto-generating the test cases and then running those test cases on its own without human interference. It generates scenarios using the applications' GUI. On the contrary, to black-box testing, we have a white-box testing strategy which is based on the logical aspect of a program. To automate the security in mobile applications is to start by auto-generating the test cases. For developers to use the automation on large-scale mobile applications, they need to divide the application into smaller parts and test individual parts. This survey was done keeping in mind the four key factors. It helps the developers to decide the necessary testing approaches.

IEEE Citation:
Z. A. Hamza and M. Hammad, "Web and mobile applications' testing using black and white box approaches," *2nd Smart Cities Symposium (SCS 2019)*, 2019, pp. 1-4, doi: 10.1049/cp.2019.0210.

Paper 8:

Title: A Review on Web Applications Testing and its Current Research Directions
Year Published: 2017

Summary:

Web applications are increasing globally, and we are required to produce bug-free applications that require faster and more efficient testing cycles. This paper presents a comparative study on existing techniques and models for web application testing. The cost of fixing bugs is proportional to the time of their discovery that affects customers and businesses, and the versatility of web applications has made software testing harder.
The features of web applications such as heterogeneity, execution environment, high user population, concurrent transactions, state changes, operating systems, maintenance rate, and present architecture pose challenges to web application testing (WAT). The researcher explores different architecture from literature reviews such as Beliefs, the Desires, Intention's architecture (BDI), MDWATP, SDAS, and additional component testing in Web Applications. The authors also discuss different web application testing methodologies, and one that stood out was penetration testing for WAT, which is our focus for this research project. Penetration Testing is an automated test that runs simulated active attacks to expose web application vulnerabilities and susceptibilities. The authors also discuss various sorts of industry-related tools for functional, acceptance, unit, integration, load, performance, and usability testing. Their research shows that we need more Web Application testing tools to be open-sourced because existing tools are expensive and need to be licensed. Especially for penetration testing, most devices such as Acunetic and Fortify require a license and are often hard to use.
The researchers conclude that we need more tools for different types of testing in web applications, especially for non-functional testing requirements. They concluded that the software testing industry needs a proposal metric to indicate the overall health of a web application, such as considering mutation testing.

IEEE Citation:

Lakshmi, D. & Mallika, S.. (2017). A Review on Web Application Testing and its Current Research Directions. International Journal of Electrical and Computer Engineering (IJECE). 7. 2132. 10.11591/ijece.v7i4.pp2132-2141.

Paper 9:

Title: Automated versus Manual Approach of Web Application Penetration Testing

Year Published: 2020

Summary:
Several institutions depend on web applications for interacting with the user related to online banking, social media, and other areas where the user's information becomes very precious. Vulnerabilities can have a bad impact on a company as its customer find their data on their risk. It impacts the organization's value. Penetration testing is one of the testing methods that can help companies to detect vulnerabilities. It is categorized into two types of Manual Penetration testing and Automated Penetration Testing. The manual approach involves the tester using their experience and expertise in writing the scripts to detect the vulnerabilities. While the automated penetration testing approach involves running automated scripts and scanning. These scanners look for the most common and known vulnerabilities. There are certain scenarios that prove that one testing proved to be more effective than the other and vice-versa. Some attacker attempts that are discussed: Clickjacking and Business vulnerability. Attackers perform clickjacking by loading an external website into their iframe. Their own website UI is made invisible and the external website acts as the primary UI. The attackers then get access to their information by disguising the users. Business vulnerability is caused to bad logic rather than poor practice. The impact of this vulnerability is not specific as it depends on flaws that the logic has. If the attackers can find out the loopholes, they may get access to the user's information. X-frame gives the ability to a website by restricting the website from being rendered to iframes. Several automated testing approaches have been proposed to mitigate the business logic. There are advantages and disadvantages associated with manual and automated testing. Some pros of automated testing are: Saves human resources used for scanning applications and it serves faster testing. While automated testing may not be able to catch the edge cases and hence exceptional cases of vulnerabilities are missed out. This con of automated testing is covered by manual testing as with the tester's knowledge and experience, they can locate those edge cases. However manual testing is slower and requires human resources. Yet there are many other scenarios where manual testing is more effective which are not mentioned in this paper. Studying each scenario can be helpful to determine the best testing approach.

IEEE Citation:
N. Singh, V. Meherhomji and B. R. Chandavarkar, "Automated versus Manual Approach of Web Application Penetration Testing," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2020, pp. 1-6, doi: 10.1109/ICCCNT49239.2020.9225385.

Paper 10:

Title: Analysis of Web Security Using Open Web Application Security Project 10

Year Published: 2020

Summary:
With an increased development and evolution in IT, it becomes important to ensure that their websites are free from attacks and abuse. There are some international standards set by the ISO 27001 that defines the requirements which helps in guarding the company's information system. The use of OWASP 10 has guidelines for ensuring the website security. Some steps have been set to conduct research on Penetration Testing. These steps constitute identification of problems, conducting literature review, making test security using OWASP 10 and making a report on Security test. Several tools have been proposed for Penetration Testing: Reconnaissance, Scanning and Exploitation. Reconnaissance is categorized into two types: Active Reconnaissance and Passive Reconnaissance. Active Reconnaissance works by directly interacting with the target and recording IP and activities. Passive Reconnaissance involves using information available on the web. This completed the first part. Afterwards information collected from the Reconnaissance is used to map the IP to open port. This method is called scanning. The final stage of penetration testing is the exploitation where it takes over a computer and gaining the administrative access. This allows them to change any function. It mainly focusses on the we application as most business today have their presence on the web. Exploitation helps in realizing the vulnerabilities. OWASP ZAP study suggests that level of challenge can be categorized into 4 levels: high, medium, low and info. Several threats with varying degrees were found from the results of the exploitation using OWASP ZAP tools. Therefore, it becomes necessary to optimize to prevent the vulnerabilities attack.

IEEE Citation:
M. Agreindra Helmiawan, E. Firmansyah, I. Fadil, Y. Sofivan, F. Mahardika and A. Guntara, "Analysis of Web Security Using Open Web Application Security Project 10," 2020 8th International Conference on Cyber and IT Service Management (CITSM), 2020, pp. 1-5, doi: 10.1109/CITSM50537.2020.9268856.