The Study on Assessment of Security Web Applications

https://doi.org/10.3991/ijim.v15i23.27357

Mohammad Ali A. Hammoudeh(✉), Ali Alobaid, Ali Alwabli, Faris Alabdulmunim
Department of Information Technology, College of Computer, Qassim University,
Buraydah, Saudi Arabia
maah37@qu.edu.sa

Abstract—Nowadays, technology has an important role of our life, including smart devices, social media due to the importance of security and web for interaction so it has become targeted it by cybercriminal. The growing threat of cybersecurity has prompted the kingdom to pay more attention to its national cybersecurity strategy as the state embarks on a Vision 2030 plan, which aims to diversify the economy and create new jobs. Therefore, Web Applications are always having security threats, which considered as a big problem. Several steps introduced successful analysis of vulnerabilities in web applications. There are now no effective and simple-to-use techniques available for evaluating the security of such apps. This creates a window of opportunity for hackers to launch a successful assault. In this paper, we propose an efficient approach for assessing vulnerability that is based on the Python programming language. which can used to conduct Vulnerability Assessment on web applications. This work will be useful for organizations and programmers to keep their information and applications more secure and viable for usage in sensitive environments.

Keywords—web applications, secure socket layer (SSL), python, analysis, threat modelling, security assessment tools

1 Introduction

This Since the beginning of the web in 1989 and expansion of the internet widely to facilitate life fields such as education, health, transportation, and national security and so many fields but with ease-of-use technology, the security issues come with it. Unauthorized access to web application and manipulate it causes a lot of financial damage and destroys infrastructure, which leads to economic paralysis and threatens public security. Recently security became major factor in our life for every individual, or organization and governments, and security is how to protect your information against any attack or threats. Protection done by detecting threats and try to prevent those threats from affecting the system [1]. There are many types of security and we will focus on information Technology security as in Figure 1 [2].

Weak security can cause many major troubles such as confidential data is compromised, negative public image, financial losses and legal issues, so there is no organization can operate and success with weak security. For many years, Saudi Arabia suffered continuous of numerous cyber-attacks on its infrastructure systems, which cause losses

of billions of riyals. For that, the vision of 2030 came with many goals and ambitions; one of those is to mitigate the risk cyber-security dilemma. Not only Saudi Arabia, but also cyber-attacks also have an impact on the global economy. According to the Forbes report, that the cost of cybercrime will be $10.5 trillion per year in the world by 2025 [3].

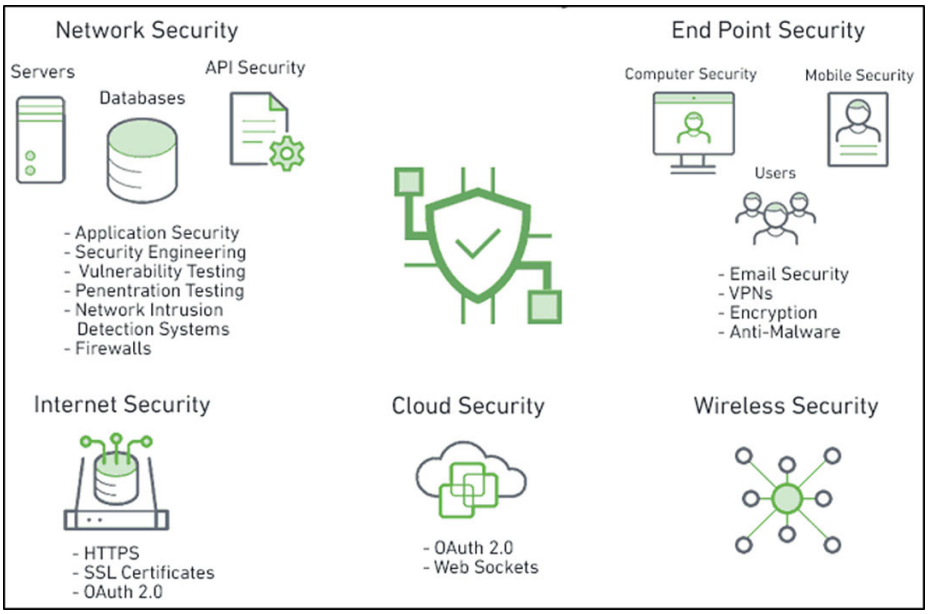

Fig. 1. Types of IT security

The term web application refers to a software that uses web browsers to carry out activities via the web. Web application uses ASP and PHP for store the information and uses HTML languages and JavaScript for displaying the information to the client. As result of importance of the data, the number of attacks has increased and hackers, target web application and that cause several vulnerabilities. Foreign statistics indicated that the cost of cybercrime in the Kingdom amounted to $ 2.6 billion, while the percentage of attacks on Saudi companies reached 69% [4].

Vulnerability assessment is performing a technique to find the security defects and it can be done by manual or automatic search techniques. Vulnerability assessments can target multiple technology layers such as internet layers [5]. The problem started since the emergence of the way of communication though the web application. Vulnerabilities one of the problems that may cause harm to our system for the user and because of the simplicity of the http which using in email so that anyone can communicate with anyone just by having an email. Application layer uses Hypertext Transfer Protocol to disseminate and cooperate hypermedia information systems, while Transport Layer Security encrypts the connection [6]. Even if the connection is encrypted, these tools fail horribly when evaluating HTTPS web apps. It is easy to case hard user and this the pave way for hackers to easily to attack. Therefore, our framework is easily

an efficient method about security assessment using python language on web applications to find a best solution and practices to be protected, which make the economy saved and creates jobs.

This paper is organized as follows: Section 1 presents an introduction to security of application. Section 2 the related work and techniques. Section 3 reviews some discussions about types of vulnerability scanners, feedback and solutions. Section 4 presents the conclusion and future work.

2 Related concepts and techniques

This part presents the related concepts and techniques used in security assessment of web applications. In addition, we define and explain what assessment methods are and how they work.

2.1 History of vulnerability scanning

In the late 90s and early 2000s, the first vulnerability scanners were released. In comparison to the present day, there were not many dangers in those days. In the year 2000, for example, 1,020 vulnerabilities were discovered. In comparison, 2018 saw a whopping 16,555 vulnerabilities discovered.

The scanning and processing in those days was largely manual, with the survey program reporting on the weaknesses found that need to be analyzed to confirm their accuracy and validity by someone in the IT department.

The report would be sent to IT department heads for review and approval. Then once approved system officials will address security vulnerabilities and follow up another scan of security gaps to verify the results. The number of threats has gradually risen over the last decade, with 4,652 new vulnerabilities identified in 2010 and 6,447 new threats revealed in 2016. However, beginning in 2017, there was an uptick in threats that continued until 2019 [7].

2.2 Concept and definitions

The effective examination of web application vulnerabilities has been introduced in various stages throughout the years. Authentication, session management, authorization, cryptographic data validation, denial-of-service attacks, specific risk functionality, and error handling are among them. [8].

An application's security risk assessment finds, assesses, and applies critical security controls. It also focuses on avoiding faults and vulnerabilities in mobile apps. An company may use a risk assessment to examine the whole program from the attacker's point of view.

It provides assistance to managers in making security-related choices. As a result, directing an evaluation is a delicate part of the organization's risk management process to navigate.

Developing an effective security risk assessment methodology involves four stages. [9]:

1. **Identification:** Determine the technological infrastructure's key assets. Following that, diagnose critical data that these assets generate, store, or transfer. Create an individual risk profile for each.
2. **Assessment:** Managing a process for evaluating the particular security threats associated with critical assets. Following assessment, choose the most effective and efficient way to devote time and resources to mitigating risks. The strategy or methodology used to conduct the assessment should examine the connection between vulnerabilities, threats, assets, and mitigating controls.
3. **Mitigation:** Define a risk mitigation strategy and put in place security measures for each one.
4. **Prevention:** Implement tools and procedures to help prevent threats and weaknesses in your firm's resources from occurring.

HTTPS (Hypertext Transfer Protocol over Secure Sockets Layer) was a game changer in the area of online security. Since its debut, an increasing number of businesses have migrated to HTTPS web servers, with search engine behemoths like Google publicly saying that they give a substantial advantage to websites that use HTTPS during searches. Although SSL is supposed to be highly safe, it has been shown to have many hidden weaknesses. Figure 2 shows a comprehensive security assessment.

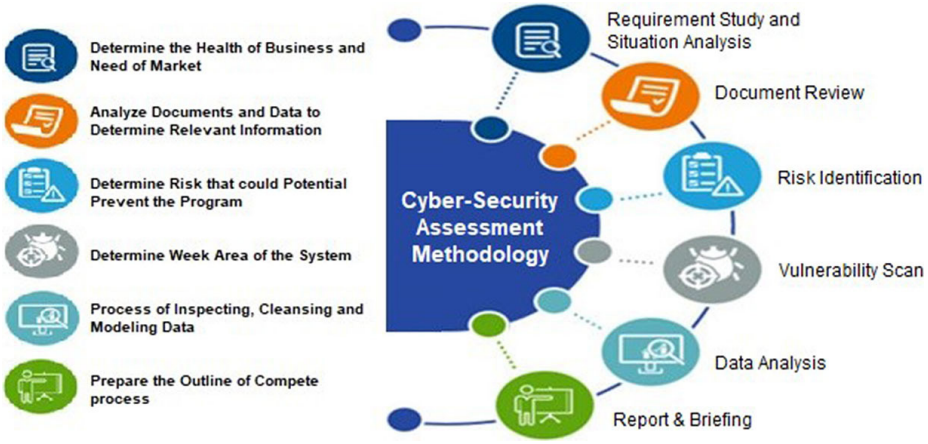

Fig. 2. Security assessment methodology

The architecture structure of web applications determines interactions between applications, intermediate software databases, and systems to confirm that multiple applications can execute together. Clicking on "Transition" after typing in the URL causes the browser to locate and request this particular page from the website's Internet-facing machine. Upon receiving a response from the server, the browser is then sent files from the server. These files are then executed by the browser to display the

requested page for the user. The user may now engage with the website directly. All of this happens in a matter of seconds, of course. Figure 3 shows what might happen if people didn't care about websites. You have the server and the client when dealing with web applications. There are really two applications operating at the same time. [10, 11]:

- The code that runs on the user's computer and reacts to commands sent by the browser.
- The server-side code that responds to HTTP requests.

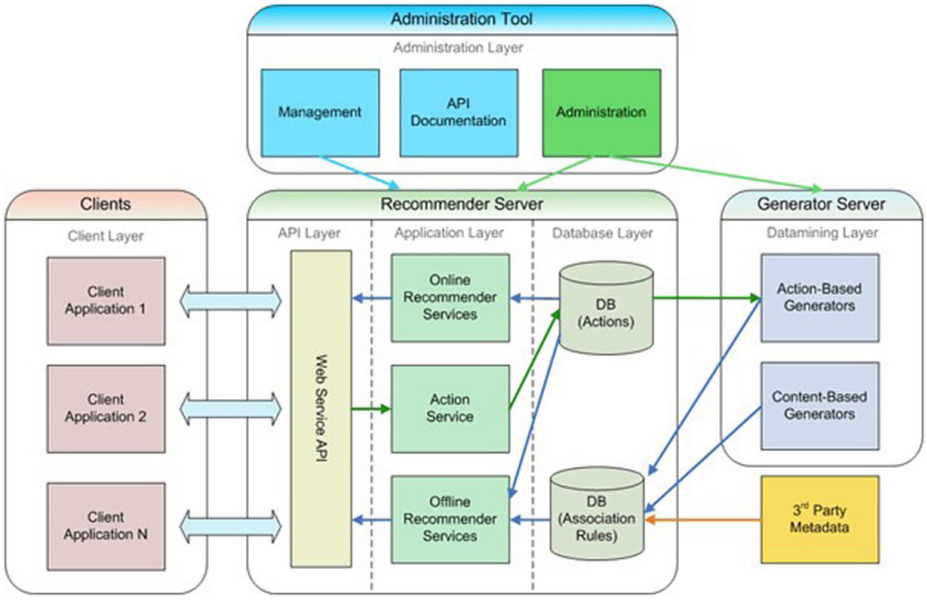

Fig. 3. Web application architecture

2.3 Previous studies & related works

End-user secure online application services, which are increasingly the main issue of each industry, are the focus of sustainable security. A research involving four thousand professionals was conducted by the Global Executive Report on Sustainability. The attitude of constantly forcing them to implement security models for readily accessible online apps was agreed with by 48 percent of respondents. Web application security is often overlooked, according to survey participants, despite the fact that security at the design stage is essential. Most security professionals rely on simple network engineering and flimsy frameworks, which have a negative impact on long-term viability. Indeed, the impact of sustainability on design elements must be assessed. [12]. Table 1 summaries some of Vulnerability Assessment Tools in Previous Studies for Web Applications.

Table 1. Summary of some previous studies

Paper	Threat Name	Problem	Solutions	Advantage	Disadvantage
[13]	Cross site Scripting (XSS)	1-Cookie Stealing 2-Key Logging 3-Phishing	1-Use both black box and white box testing technique. 2-Use of security testing tools to detect XSS, like Burp suite. 3-Use of application level firewall that handles the request between client and server.	1-Help tester in detection. 2-Effective result for prevention of XSS vulnerabilities.	1- It is not sufficient in the prevention of more dangerous XSS payloads. 2-It is not strong enough to handle middleware attacks.
[14]	1-CSRF 2-XST 3-DDoS in Shared Hosting	If the vulnerability test is conducted by only one tool, it may not be possible to investigate all the vulnerabilities.	1- Selecting a set of tools in an optimized and organized way. Proposed testing phases - "Identify", "Analyze", "Test", and "Report". 2-Combination of diversified tools, such as W3AF and Nikto used as a demonstrator.	Help a penetration tester to achieve the rigorous vulnerability assessment tests that can cover the most of the vulnerabilities.	Cannot investigate all the vulnerabilities.
[15]	1-Click-Jacjing 2-SQL injection 3-Cross-site scripting 4-Private ip disclosure	In the financial industry, there are more problems, as well as organizations and customer sensitive and private data.	Observed vulnerability after testing, solved this vulnerability and further a framework is presented which enhances the security of this proposed framework.	It can be used in hotel accommodation web application also; hotel websites are also too vulnerable.	For automated Testing shows less statics numbers than manual testing.
[16]	2-SQL	This may result in the disclosure of administrator credentials, credit card information, social security numbers, and email data, among other things.	Using Escrow broken into four phases : 1-Link Gathering. 2-Analyzing URLs. 3-Identifying Injection Type. 4-Database Exploration.	On a broad scale, the time it takes to detect SQL injection in Escrow is orders of magnitude quicker than the present state of the art.	One of SQLMap's main flaws is its dependence on the Google API, which has placed limitations on the kind of searches it accepts within a certain time frame.

(Continued)

Table 1. Summary of some previous studies *(Continued)*

Paper	Threat Name	Problem	Solutions	Advantage	Disadvantage
[16]	2-SQL	This may result in the disclosure of administrator credentials, credit card information, social security numbers, and email data, among other things.	Using Escrow broken into four phases : 1-Link Gathering. 2-Analyzing URLs. 3-Identifying Injection Type. 4-Database Exploration.	On a broad scale, the time it takes to detect SQL injection in Escrow is orders of magnitude quicker than the present state of the art.	One of SQLMap's main flaws is its dependence on the Google API, which has placed limitations on the kind of searches it accepts within a certain time frame.
[17]	1-SQL 2-XSS 3-CSRF	Data integrity may be compromised, sensitive data can be stolen, and online apps might be unavailable due to security flaws.	1-using automation techniques such as Acunetix, Burp Suit, OWASP Zed Attack Proxy. 2-Using manual testing since some vulnerability is difficult to find using automated tools.	Automation reveals manual penetration test is more effective in terms of accuracy.	While these automations are an important, that a tester should be aware of few holes.
[18]	1-XSS 2-CSRF	Both XSS and CSRF attacks are very common, and both of these attacks are implemented through breaching the users' confidence in the applications and services that they rely upon on a daily basis.	1- Using black box testing approach for XSS and CSRF. 2-Using code pattern for stored XSS.	It takes time to carry out these assaults, particularly when employing a black-box testing approach.	Not all online applications that are XSS susceptible are vulnerable to all three kinds of XSS attacks.

We classify the related work according to Vulnerability consider it as the following:

1-Injection vulnerabilities [19]: Occur when an attacker targets databases or directories linked to your web applications with unfiltered, often dangerous data. Typically, injection attacks are carried out in one of two ways. SQL injection's original application is to target your databases. Second, to compromise routes, LDAP injection is utilized.

The Effects: Injection attacks exploit weaknesses by exploiting input fields that connect with directories and databases. Usernames, passwords, and other places that interact with the target are among them. When a database or directory is being developed, these fields are often left exposed owing to the absence of an input filter.

The Solution: There are many methods in which we may assist in preventing injection attacks. The greatest protection is to add filters to your inputs. With SQL

databases, we may begin by utilizing prepared statements, which assist protect against attackers altering queries. Second, using LDAP injections, we may utilize protocols as escape variables to prevent the directory from being manipulated by characters used in injection attempts.

2-Broken Authentication [20]: Authentication enables applications to identify and verify their users. As a consequence, deactivated authentication may enable attackers to get access to and acquire the same rights as the target user, resulting in significant security holes in the online application. Issues with authentication may allow attackers unfettered access to your data and cause havoc with your online application.

The Effects: Authentication flaws may include poorly hashed and salted passwords, data breaches involving user accounts, incorrectly configured timeouts, brute force attacks, or common password stuffing attacks such as password1 or admin1234.

The Solution: It's possible to protect your web application against authentication flaws with a simple patch. Multi-factor authentication may assist in confirming that the person is who they say they are. Creating strong passwords and updating them on a regular basis may help you avoid common password mistakes. Finally, by correctly setting password security and timeouts in your database, you may avoid authentication problems.

3-Sensitive Data Exposure [21]: Sensitive data is transmitted or kept without encryption or other security measures, making data susceptible to a variety of assaults.

The Effects: Data that isn't password-protected may be accessed in two ways. It is possible to steal data packets while they are being sent between a user and a client using techniques such as a man-in-the-middle attack. In addition, stored data, even if it is more complicated, may be exposed if encryption keys are stored with it, or if the salt/hash is poor.

The Solution: Preventing sensitive data leakage is critical to app security. Due to data vulnerabilities in transit, HTTPS, and PFS, ciphers must be used for incoming data. Deactivating sensitive data caching is another method to safeguard data. Removing obsolete or unwanted data reduces coverage. No data, no risk.

4-Missing Function Level Access Control [20]: When server-side authorisation is incorrect, faulty, or not present, vulnerabilities may arise, which can expose your back-end to malicious attacks and compromise your security.

The Effects: Front-end UIs built with components to allow administrators access to data or other critical app parts are often the target of these assaults. Most users will not be able to view the admin function in this scenario, but those searching for flaws will be able to discover and exploit this vulnerability with malicious queries.

The Solution: Fixing this problem is straightforward. All server-side authentication must be enabled and set in order to prevent unauthorized access from occurring.

5-Security Misconfiguration [21]: Web applications are often misconfigured, exposing attackers to a slew of flaws Non-patched faults, unused pages, obsolete software, and debug mode running may all lead to security misconfigurations.

The Effects: Misconfigurations in your web security may have an impact on every element of your apps. When a configuration error is discovered, a security audit must be performed to look for any attacks or security breaches.

The Solution: Avoiding security configuration flaws is a straightforward process. A deployment protocol may be used to constantly create and distribute updates within a safe environment, or a segmented application design can assist prevent security vulnerabilities from arising. Automatic deployment will help keep your apps up to date and protect you from cyber threats, among other benefits.

6-Cross-Site Scripting (XSS) [22]: Cross-site scripting is a method of attacking a user's web browser by injecting malicious code into seemingly harmless websites. The code will be entered by an attacker via a link, which will entice the user into clicking on the link and running the code in question. Users' camera, location, and other personal data and functionalities are accessible by attackers who utilize JavaScript to exploit XSS vulnerabilities in Java.

The Effects: Unsanitized input is prone to XSS. Aside from stealing cookies, XSS may enable attackers to obtain browsing history and other data.

The Solution: Sanitizing input is the best solution to XSS problems. Python, one of the most generally recognized and extensively used programming languages in 2021, is used to develop a broad range of software, including online applications, video tools, desktop programs, network servers, and machine learning. As a result of this new technology, libraries will have better access to controls and integration. The top programming languages for 2021 are summarized in the IEEE spectrum. The following list of programming languages has been compiled based on the most widely used ones.

3 Discussion

The analysis involves studying the currently available models to lead us to a new model without making the same mistakes, Depending on the analysis of the current models and user requirements. In this section, we analyze the user requirements by collecting data and then we design and develop our model based on an appropriate methodology.

3.1 Types of vulnerability scanners

Vulnerability scanners for web apps that are automated are among the most well-known and highly rated vulnerability scanners available (all of them support HTTP only) as:

1. **Net Nirikshak 1.0:** is a powerful tool for assessing a company's security posture and spotting potential security flaws before they happen. Since it's fully automated and gives the User/Tester an Interactive window, using it is faster and easier while maintaining high levels of Accuracy. It also performs all security auditing procedures in one go, so the tester doesn't have to search for additional tools to round out the job at hand. … read more It comprises five stages: information gathering, scanning, vulnerability detection, exploitation, and report generation. The information gathering stage is the first step [23].

2. **Samurai framework:** is a collection of very strong facilities, each of which targets a distinct set of vulnerabilities. Comes as a Linux distro with the sole purpose of testing browser-based vulnerabilities. It also includes penetration-testing tools like as WebScarab for setting HTTP and W3AF for app-based assaults [8].
3. **Safe3 scanner:** is capable of dealing with HTTP sessions as well as cookies. Safe3 scanners are capable of dealing with virtually any form of authentication. It has a scary web feature that ignores repeated page scans while still detecting JavaScript gaps on the client's end. Safe3 scans also identify the potential of the most recent AJAX-based assaults and even identify script libraries that are at danger. A simple to use visual user interface allows you to create professional-looking management reports in minutes [8].
4. **Websecurify:** If there is a big web farm with a team of engineers or a comparable environment that maintains code, may traverse the code and identify such gaps fast. It is one of the few platform-agnostic tools that also enables mobile coding [24].
5. **SQLmap:** is capable of altering not just SQL-injection errors, but also the database server. Because it is task-oriented, it works fast for fingerprint databases, platform files, and operating system information, and ultimately fetches data from the server. It is compatible with almost all well-known database engines and is capable of performing password guessing attacks [25].

It's clear that the well-known tools for testing web apps for vulnerabilities listed above only support HTTP. There is a pressing need for a new, user-friendly tool for evaluating HTTPS web applications for security flaws. This project aims at development of such a tool, which would form a basis of solving some of the issues faced by HTTPS web applications [26–29].

3.2 Feedback and statistics

In this section, we measured the public awareness of the thread of web application attack to help us identify the extent of people understanding and how to deal with these issues, which includes the most of the responses from beginners and intermediate with almost the same percentage. To collect the data that helps this study, a questionnaire was used to find out how much users know about challenges of security in web application. It has 85 responses collected by google form, which included thirteen specialized questions as in Figure 4.

1. Describe your knowledge in cyber security.
2. Do you trust that your applications are secure?
3. What is the most types of vulnerabilities, which use by attacker?
4. Have you ever been attack through internet?
5. If yes, how did you assess the attack?
6. If yes, please indicate the type of attack you have experienced.
7. What do you think Attack do? Check any that apply.
8. What is the security software or system do you use to protect your application? Check any that apply.
9. How easy would be to protect, prevent, and clean your System?
10. What sort of software products do you use to protect your system against Attackers?
11. What are the factors that may affect your assess of any Security Technique?
12. Who do you think is responsible for assessing you system?
13. Do you agree that your current government is strong enough to protect your privacy from illegal tools?

Fig. 4. The questionnaire sample

We concluded the following after analyzing the results:

1) Most of the responses from beginners and intermediate with almost the same percentage while 24.4% have no idea about cybersecurity but a few of them are experts.
2) More than a half are untrusted with the security of their applications.
3) Injection, sensitive data and using components have the most results as common vulnerabilities than others.
4) More than 25% have been attacked through the internet most of them have been asked for their credit card according to their assessment.
5) Viruses get the highest percentage responses of the experienced attack.
6) Steal my important data, damage my system, observe my actions and activities, control my system by a remote attacker, weaken my protection system last Slow down the Internet connection have been sorted respectively by the response as attack effect.
7) 32% described the treatment cases of the affected system as difficult.
8) The main point in system protection against attacks shows that 20.8% of users do not used protection software on their systems.
9) The following responses in Figure 5 measure the affected factors of the assessment.
10) More than a half agree with that the current government is strong enough to protect their privacy from illegal tools as in Figure 6.

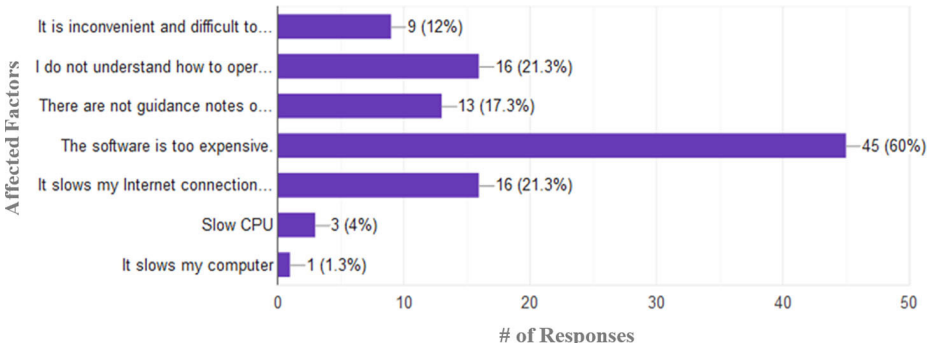

Fig. 5. Assessment of the affected factors

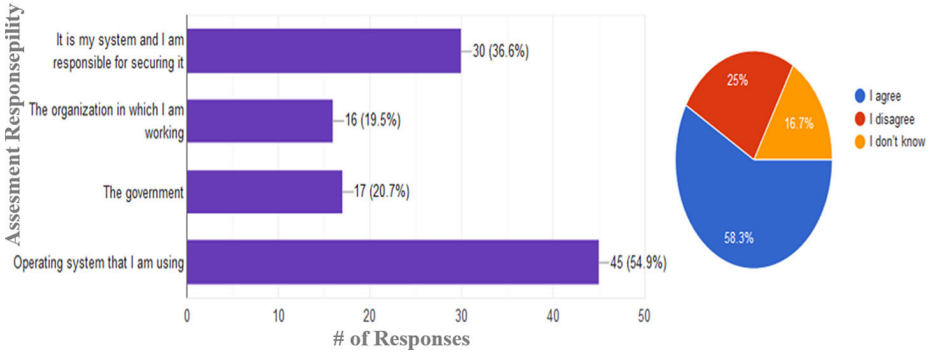

Fig. 6. Responsibility of system assessment

3.3 Solution & recommendation

To create dependable and resilient web applications, we must evaluate the security that allows us to monitor, analyze, and quantify application behavior in the face of a variety of defects and assaults. In this article, we presented a scanning tool (a Python-based automated tool) for analyzing and assessing the vulnerabilities of online applications in real time. It is an online system, which simulates an institution environment. This scanner allows us to determine how attacks and crashes affect web applications, detect attack points, and examine how important it is for web app components to behave during an attack or system error as shown in Figure 7.

The study contains multi pages and databases. HTML and CSS will be used in the presentation layer (Front end) whereas PHP will be used in the backend. For the database, MySQL will be used and few code snippets of JavaScript were used for some client-side validations. We had divided our work in to four pieces. Front End Development, Back End Development, Database Connections and Operations and Hacking Attacks.

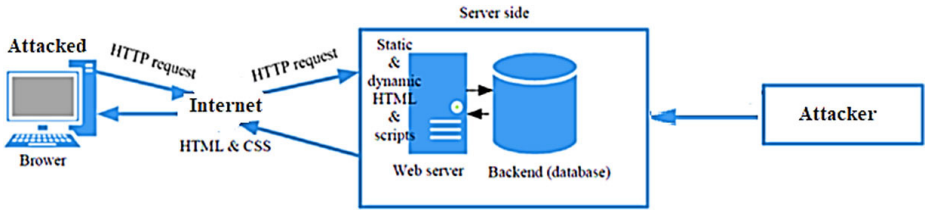

Fig. 7. The proposed security assessment tool

After determining requirements, so the data flow will represent steps as shown in the Figure 8.

The tool contains 4 phases as the following:

1) **Planning:** This phase consisted of all activities that we needed to be performed before the actual vulnerability assessment.
2) **Information Gathering:** we will gathering information about the specific websites, which will be helpful in the finding of vulnerabilities.
3) **Vulnerability Assessment:** Includes Port Scanning, Certificate Verification and Vulnerability Detection.
4) **Report Generation:** This is the last phase of the working of the tool. On completing the vulnerability analysis, a report is generated in the form of a text file containing all the necessary details of the Assessment.

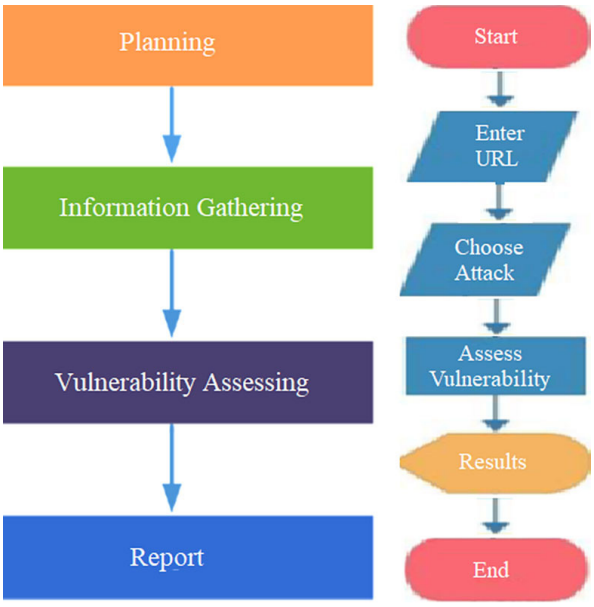

Fig. 8. Security assessment model of web application

4 Conclusion and future work

In this solution model we learned the hacking attacks firstly, then looked for the previous work, searched for assessment tools. This paper discussed the security web application assessment approach for user review classification. The broad implications of this categorization will assist both industry websites and general users in displaying reviews more efficiently and making decisions when they have access to them. Other categories will be dealt with in the future, as well as a deeper analysis of the data in the form of graphs to be implemented.

5 References

[1] Muzaki, R. A., Briliyant, O. C., Hasditama, M. A., and Ritchi, H. (2020). improving security of web-based application using modsecurity and reverse proxy in web application firewall. International Workshop on Big Data and Information Security (IWBIS) IEEE, pp. 85–90. https://doi.org/10.1109/IWBIS50925.2020.9255601

[2] Yu, X., Wu, J., and Yin, X. (2003). Study on conformance testing of hypertext transfer protocol. International Conference on Communication Technology Proceedings, ICCT, April 9 2003, IEEE, pp. 178–181.

[3] Semlambo, A. A., Mkude, C. G., and Lubua, E. W. (2021). factors affecting the security of information systems: a literature review. Engineering and Science, 10: 57–65.

[4] Ministry of Communications and Information in Saudi Arabia (03 April 2021). Available: https://www.mcit.gov.sa/ar/media-center/news/93609

[5] Radu, R., Săndescu, C., Grigorescu, O., and Rughiniş, R. (2020). Analyzing Risk Evaluation Frameworks and Risk Assessment Methods. 19th RoEduNet Conference: Networking in Education and Research (RoEduNet), December 2020, IEEE, pp. 1–6. https://doi.org/10.1109/RoEduNet51892.2020.9324879

[6] Pfleeger, C. P., and Pfleeger, S. L. (2012). Analyzing computer security: A threat/vulnerability/countermeasure approach. Prentice Hall Professional.

[7] Mamilla, S. R. (2021). A Study of Penetration Testing Processes and Tools. Electronic Theses, Projects, and Dissertations, 1220. https://scholarworks.lib.csusb.edu/etd/1220

[8] Ramesh, A. (2016). an automated tool for vulnerability assessment of https web applications (Doctoral dissertation, Institute for Development and Research in Banking Technology).

[9] Abozeid, A., AlHabshy, A. A., & ElDahshan, K. (2021). A Software Security Optimization Architecture (SoSOA) and Its Adaptation for Mobile Applications. International Journal of Interactive Mobile Technologies, 15(11). https://doi.org/10.3991/ijim.v15i11.20133

[10] Walker, J. D., and Chapra, S. C. (2014). A client-side web application for interactive environmental simulation modeling. Environmental Modelling and Software, 55: 49–60. https://doi.org/10.1016/j.envsoft.2014.01.023

[11] How web works - web application architecture for beginners. GeeksforGeeks. Accessed on: 2021, February 4. https://www.geeksforgeeks.org/how-web-works-web-application-architecture-for-beginners/

[12] Agrawal, A., Alenezi, M., Kumar, R., and Khan, R. A. (2019). Measuring the sustainable-security of Web applications through a fuzzy-based integrated approach of AHP and TOPSIS. IEEE Access, 7: 153936–153951. https://doi.org/10.1109/ACCESS.2019.2946776

[13] Shrivastava, A., Choudhary, S., and Kumar, A. (2016). XSS vulnerability assessment and prevention in web application. 2nd International Conference on Next Generation Computing Technologies (NGCT), October 2016, IEEE, pp. 850–853. https://doi.org/10.1109/NGCT.2016.7877529

[14] Alghamdi, M. I. (2020). Survey on Applications of Deep Learning and Machine Learning Techniques for Cyber Security. International Journal of Interactive Mobile Technologies, 14(16). https://doi.org/10.3991/ijim.v14i16.16953

[15] Goutam, A., and Tiwari, V. (2019). Vulnerability Assessment and Penetration Testing to Enhance the Security of Web Application. 4th International Conference on Information Systems and Computer Networks (ISCON), November 2019, IEEE, pp. 601–605. https://doi.org/10.1109/ISCON47742.2019.9036175

[16] Delamore, B., and Ko, R. K. (2014). Escrow: A large-scale web vulnerability assessment tool. 13th International Conference on Trust, Security and Privacy in Computing and Communications, September 2014, IEEE, pp. 983–988. https://doi.org/10.1109/TrustCom.2014.130

[17] Nagpure, S., and Kurkure, S. (2017). Vulnerability assessment and penetration testing of Web application. International Conference on Computing, Communication, Control and Automation (ICCUBEA), August 2017, IEEE, pp. 1–6. https://doi.org/10.1109/ICCUBEA.2017.8463920

[18] Farah, T., Shojol, M., Hassan, M., and Alam, D. (2016). Assessment of vulnerabilities of web applications of Bangladesh: A case study of XSS and CSRF. 6th International conference on digital information and communication technology and its applications (DICTAP), July 2016, IEEE, pp. 74–78. https://doi.org/10.1109/DICTAP.2016.7544004

[19] Stasinopoulos, A., Ntantogian, C., and Xenakis, C. (2015). Commix: Detecting and exploiting command injection flaws. Dept. Digit. Syst., Univ. Piraeus, Piraeus, Greece, White Paper.

[20] Imran, K., Anjum, N., Alghamdi, A., Shaikh, A., Hamdi, M., and Mahfooz, S. (2022). A Secure and Efficient Cluster-Based Authentication Scheme for Internet of Things (IoTs). Computers, Materials & Continua, 70(1), 1033–1052. https://doi.org/10.32604/cmc.2022.018589

[21] Liu, F., Shu, X., Yao, D., and Butt, A. R. (2015). Privacy-preserving scanning of big content for sensitive data exposure with MapReduce. 5th ACM Conference on Data and Application Security and Privacy, March 2015, pp. 195–206. https://doi.org/10.1145/2699026.2699106

[22] Bukhari, S. N., Dar, M. A., and Iqbal, U. (2018). Reducing attack surface corresponding to Type 1 cross-site scripting attacks using secure development life cycle practices. 4th International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), February 2018, IEEE, pp. 1–4. https://doi.org/10.1109/AEEICB.2018.8480945

[23] Shah, S., and Mehtre, B. M. (2014). An automated approach to vulnerability assessment and penetration testing using net-nirikshak 1.0. International Conference on Advanced Communications, Control and Computing Technologies, May 2014, IEEE, pp. 707–712. https://doi.org/10.1109/ICACCCT.2014.7019182

[24] Shaikh, A., Ali, S., Memon, N., and Karampelas, P. (2010). SOA security aspects in web-based architectural design. In From Sociology to Computing in Social Networks (pp. 415–430). Springer, Vienna. https://doi.org/10.1007/978-3-7091-0294-7_22

[25] Imran, K., Anjum, N., Alghamdi, A., Shaikh, A., Hamdi, M., and Mahfooz, S. (2022). A Secure and Efficient Cluster-Based Authentication Scheme for Internet of Things (IoTs). CMC-Computers Materials & Continua, 70(1), 1033–1052. https://doi.org/10.32604/cmc.2022.018589

[26] Quasim, M. T., Khan M. A., Algarni F., Alharthy A., and Alshmrani G. M. M. (2020) , Blockchain Frameworks. In: Khan M., Quasim M., Algarni F., Alharthi A. (eds) Decentralised Internet of Things. Studies in Big Data, vol 71. Springer, DOI: https://doi.org/10.1007/978-3-030-38677-1

[27] Khan M. A.,Quasim, M. T., et al., (2020), Decentralised IoT, Decenetralised IoT: A Blockchain perspective, Springer, Studies in BigData, 2020, DOI: https://doi.org/10.1007/978-3-030-38677-1

[28] Zidoun, Y., Dehbi, R., and Talea, M. (2018). Multi-Criteria Analysis and Advanced Comparative Study between M-learning Development Approaches. International Journal of Interactive Mobile Technologies, 12(3). https://doi.org/10.3991/ijim.v12i3.8083

[29] Quasim, M. T., Khan M. A, et al., (2019). Internet of Things for Smart Healthcare: A Hardware Perspective, 2019 First International Conference of Intelligent Computing and Engineering (ICOICE), Hadhramout, Yemen, 2019, pp. 1–5. DOI: 10.1109/ICOICE48418.2019.9035175 https://doi.org/10.1109/ICOICE48418.2019.9035175

6 Authors

Mohammad Ali A. Hammoudeh, is an Assistant professor at Department of Information Technology, College of Computer, Qassim University, Buraydah 51941, Saudi Arabia.

Ali Alobaid, Department of Information Technology, College of Computer, Qassim University, Buraydah 51941, Saudi Arabia.

Ali Alwabli, Department of Information Technology, College of Computer, Qassim University, Buraydah 51941, Saudi Arabia.

Faris Alabdulmunim, Department of Information Technology, College of Computer, Qassim University, Buraydah 51941, Saudi Arabia.

Article submitted 2021-09-06. Resubmitted 2021-10-18. Final acceptance 2021-10-21. Final version published as submitted by the authors.