Solutions and Vulnerabilities of Security of Web Application

Intisar Milad Mohamed ALSSULL[1], Ibtisam Abdalsalam Mohamed SIDOUN[2], Laila Yousef FANNAS[3]

[1] Lecturer, Dept. of Internet Systems, Information Technology college, Misurata University, Libya

[2] Lecturer, Dept. of Computer Science, Information Technology college, Misurata University, Libya

[3] Assistant lecturer, Dept. of Internet Systems, Information Technology college, Misurata University, Libya

Article information

Key words
Security, Application, Web, Solutions

Received 26 June 2021,
Accepted 22 July 2021,
Available online 31 July 2021

Abstract

In this study, the top 10 web application security vulnerabilities published by OWASP, the sources of the vulnerabilities and the security solutions used to prevent attacks that exploit these vulnerabilities were investigated, the precautions that can be taken against the attacks that can be carried out using these vulnerabilities were evaluated in terms of usage areas, platform-independence, working logic and efficiency have been compared. In line with the information and findings obtained, suggestions on what kind of security solution should be taken and preferred against which types of attacks, and solutions for increasing awareness and web application security were presented.

I. INTRODUCTION

SQL (Structured Query Language) is a structured query language used to perform operations such as web services and applications have shaped information sharing and increased the use of web applications with the widespread use of the Internet [1]. Web applications can be used for financial transactions, information sharing, socialization and communication, as well as for the provision of public services over the internet by states. Web applications that enable many transactions become the target of attacks because they contain various information such as personal information, bank account information and corporate information [2].

Ensuring the security of these environments is directly related to the security of web applications. Web application security refers to all the measures taken to ensure the confidentiality, integrity and accessibility of the data it contains. Abuse of web application vulnerabilities can cause material and moral damages for individuals and institutions. On October 10, 2014, it was revealed that the user account was compromised by exploiting a vulnerability in the application code of PayPal, an online payment method [3].

An example of cyber-attacks on web applications is cyber-attacks on Estonia by Russia. In Estonia, which provides most of its services over the web, life has come to a standstill as a result of attacks. In order to prevent these attacks, security measures are implemented to ensure the security of intrusion detection/prevention system, firewall, application layer firewall, database firewall and application code. In this way, work continues to ensure the safety of these environments with many techniques and technologies [4].

In this study, web application vulnerabilities and the environments in which these vulnerabilities exist were investigated in order to increase web application security. Studies on security solutions used against attacks by exploiting vulnerabilities on these environments have been examined. Security solutions were compared in terms of their working logic, performance and success against attacks, and which solutions should be applied in which types of attacks. In this way, a comprehensive research has been presented on what kind of security solution security experts prefer against which types of attacks.

II. SECURITY STATUS AND WEB APPLICATIONS OVERVIEW

The Internet is the largest network and with the increase in the use of the Internet, the applications used in information systems are also developing in an Internet network-oriented manner. With the spread of the Internet, web applications have started to be used instead of desktop applications. Web applications

occupy an important place in our daily life and offer rich features such as pictures, videos and audio materials with an interactive structure [5].

Many factors such as curiosity, malicious intent, and the hope of making money play a role in making web applications the target of attacks. These factors are triggered by the fact that the functionality of the developed application is taken into account rather than its security, web application development by people who lack security knowledge, and the resulting vulnerabilities. There are various reports that reveal the security status of web applications. According to the "Web Site Security Statistics" report published by "White Hat Security", a company that provides solutions for web security, the vulnerability detection values of the websites in different domains analyzed by us have been charted by us as percentiles. It has also been stated [6].

According to the "Application Vulnerability Trends Report 2014" of CENZIC, an institution that provides application security for the continuous assessment of cloud, mobile and web vulnerabilities, 96% of applications have one or more serious vulnerabilities, is known to exist. In addition, while the number of openings per application was 13 in 2012, this figure increased to 14 [7].

Statistical data show that something is wrong in this regard and those different security solutions should now be developed and implemented. In this study, vulnerability sources and security solutions in web applications were examined and suggestions were made to security managers about the issues that they should pay attention to when choosing the right security solution [8].

III. SECURITY VULNERABILITIES OF WEB APPLICATION

When the literature is examined, OWASP stands out as the most comprehensive organization working on security risks. OWASP is an organization focused on improving software security worldwide. The organization's mission is to promote software security so individuals and organizations around the world can make informed decisions about real software security risks. The organization publishes the most common web application vulnerabilities to raise awareness in web application security [9].

Web applications are insecure; the source code of the application is caused by the vulnerabilities on the communication infrastructure between the server on which the application is running or the server client [10].

While evaluating the affected party, the affected party is considered as the server, since when the server is affected, the client will also be affected directly or indirectly. In cases where only the user is affected, the client is considered as the affected party in cases where both the server and the client are affected separately.

The OWASP organization publishes the most common web application vulnerabilities to raise awareness in web application security. According to the list published by OWASP, the top 10 vulnerabilities are discussed under the following headings as application code vulnerabilities, server-based vulnerabilities and communication infrastructure vulnerabilities. While 8 of the vulnerabilities are application code, the others are discussed under the headings of server and communication infrastructure.

A. Vulnerabilities of Source Code

Defects in the code are under this category. Attacks that can cause serious damage to the system can be organized on these vulnerabilities. The majority of attacks on web applications are caused by not checking the input values received from the user [4]. SQL injection and cross-site scripting attacks are also attacks that exploit failure to check the compatibility of input values [11]. The vulnerabilities and exploit methods that are in the OWASP list of 10 most known vulnerabilities originating from the application code are explained below.

Injection: Injection vulnerabilities occur when untrusted data is sent to the interpreter in part of the command or query. The attacker's malicious data can mislead the interpreter to execute undesired commands or gain access to the data without proper authorization. Injection attacks can be made via SQL, OS and LDAP commands [12].

Broken Authentication and Session Management occurs as a result of not implementing the authentication and session management related functions of the Application functions. Through this vulnerability, attackers can use the vulnerability in the form of passwords, keys, session tokens or to guess other users' information.

Cross-Site Scripting (XSS), Cross-Site Scripting vulnerability occurs when applications are not properly validated when receiving or sending untrusted data through a web browser. This vulnerability allows attackers to run scripts on their victim's browser. As a result of the attack, the user's session information can be stolen, websites can be damaged or users can be directed to malicious sites [13].

Insecure Direct Object References, Direct object reference occurs as a result of the application developer not doing proper access control to the references of internal application objects used in the application. As a result of the lack of access control or other protection of the references, attackers can change these references by gaining unauthorized access to the data [14].

Sensitive Data Exposure does not properly protect credit cards, tax numbers, and identification information in most Web applications. Attackers can also steal or exchange poorly protected data for credit card fraud, identity theft or other crimes. This

openness allows unauthorized access to sensitive information on the application. Extra precautions such as encryption should be taken when storing or sending sensitive data to avoid exposure [13].

Missing Function Level Access Control In most Web applications, functions are validated at the access rights level before the user interface becomes available. However, applications must perform these access controls on the server when accessing each function. If the request is not authenticated, it is possible for attackers to access functions with fake requests without proper authentication [13].

Components such as Using Known Vulnerable Components, Software modules, frameworks, and libraries generally run with full authority. If the vulnerabilities of these components can be exploited, attackers can facilitate data loss or server hijacking on the system through the vulnerabilities. The use of components with known vulnerabilities in applications can reduce the security level of the application and allow possible attack areas and effects [14].

Invalidated Redirects and Forwards Web applications frequently redirect users to other pages and websites and use untrusted data to decide destination pages. In this type of attack, attackers can redirect users to fake or malicious sites or forward them to access unauthorized pages [14].

B. Server Inherited Vulnerabilities

The vulnerabilities that arise due to not properly configuring the security on the server where the application is running are evaluated under this category [15]. The server-based vulnerabilities and exploit methods, which are on the OWASP list of 10 most known vulnerabilities, are discussed below.

Security Misconfiguration, Security needs for applications, frameworks, application server, web server, database server and environments must be found; secure configuration must be defined and applied in order to ensure the security of the application. Because the default security settings are insecure, the security configuration must be defined, implemented and maintained, plus the software must be up-to-date. As a result of not applying the security controls properly, the vulnerability on the server can be exploited and access to the data on the application can be achieved. An example of this is the situation where no password is set in the database connection and the remote connection feature is turned on.

C. Vulnerabilities of Communication Infrastructure

The vulnerabilities on the communication infrastructure between the server and the client are covered under this category. These are vulnerabilities that arise from vulnerabilities in the transmission layer, network layer, or other layers of the OSI model. The vulnerability and exploitation method, which is in the OWASP list of 10 most known vulnerabilities and originating from the communication infrastructure, is given below.

Cross-Site Request Forgery (CSRF) these attacks force the logged in user's browser to send fake HTTP requests containing the user's session cookie or other automatically added credential to create clarity on the web application. This allows attackers to see incoming requests from the vulnerable web application as legitimate requests from the user. In this way, information such as the e-mail address of the users on the system can be changed [16].

IV. METHODS OF PROTECTION FROM ATTACKS OF WEB APPLICATIONS

There are many academic and commercial security solutions developed to prevent attacks on web applications. When security solutions are examined, PHAN, SWAP, Diglossia, PSIAQOP security solutions are suggested to ensure the security of the application code. In addition, Intrusion Detection Systems approaches are used as signature-based security systems and anomaly-based security systems [17]. In this section, Intrusion Detection Systems and methods used to secure application code are explained.

As a result of the research, the criteria that provide original and effective solutions were selected among the features of security solutions. These criteria are; signature-based, performance-enabled, programming language-specific, successful in a zero-day attack. These criteria and security solutions are classified in Table 1.

TABLE I. COMPARISON OF SECURITY SOLUTIONS

Trust Method	***Signature Based***	***Performance Enabled***	***Programming Language Specific***	***Succeed in Zero-Day Attack***
Signature Based Systems	✔	✔	×	×
Anomaly Based Systems	×	×	×	✔
PHAN	✔	×	✔	✔
SWAP	✔	✔	×	✔
Diglossia	✔	✔	×	×
PSIAQOP	✔	×	×	×

A. Systems of Intrusion Detection

Intrusion detection systems are a security system that is used to detect whether there is an attack on a system and provides protection against both internal and external attacks. This system; There are two types as terminal and network-based intrusion detection systems, and terminal-based intrusion detection systems analyze data on one or more terminal systems. This information is usually; system calls, application logs, file system changes.

Network-based intrusion detection systems, on the other hand, analyze the traffic passing over it at a location that will pass over all traffic in the network topology [18]. Intrusion detection systems are discussed under two headings as anomaly-based and signature-based. These methods are described below.

Anomaly-based intrusion detection system, anomaly-based intrusion detection system works on a predetermined behavior reference. In this approach, two profiles are used, namely the determined behavior and the instantaneous behavior of the system. In the behavior determination process, taken from various sources; Various information such as processor usage, TCP connection number, monitoring events, keystroke records, system calls, network packets, user session times, number of emails, access to file systems and secure working conditions are used. Behavior determination can be done statically or dynamically. After the behavior determination process, the known behavior and the instantaneous behavior of the system are compared to determine whether there is an attack [19]. The main problems in anomaly based systems are; the necessity of determining the normal operation of the system beforehand, the change of the conditions determined as the normal operation of the system over time, the perception of attacks as normal behavior while learning the normal behavior, and the difficulty of machine learning techniques. In addition, if the normal behavior of the system changes very often over time, malicious people can spread the attack over a wide period of time and make it unnoticed on the system. Anomaly-based approaches provide more successful protection against zero-day attacks because they continuously monitor the events on the system and do not work depending on a certain condition. However, false warnings are common with this approach [20].

Signature-based intrusion detection system is an approach that works on signatures created from known attacks, using the knowledge obtained previously from signature-based systems. In signature-based systems, although there is an attack, it cannot be protected against unique attacks that could not be detected beforehand and that are not found in the signature database. However, effective protection is provided against previously known attacks found in the signature database. The signature-based approach is more acceptable in the real world because it generates fewer false alerts than other approaches [21].

The purpose of intrusion detection systems is to detect unauthorized access, infiltration of information systems, unwanted and malicious network traffic, and minimize the damage of malicious software such as viruses, Trojan horses and worms. Since intrusion detection systems are designed to protect information systems against attacks that may come over the network, they can also provide protection against attacks that can be made over the system in attacks against web applications. Intrusion detection systems can detect attacks that exploit known vulnerability components and exploit security misconfiguration vulnerabilities [22].

B. Methods Used to Securing Application Code

There are also methods to ensure the security of the application code in order to increase web application security. These methods have limitations such as being specific to the attack type and programming language. There are many methods in the literature to ensure the security of the application code.

Diglossia is a tool that can precisely and effectively detect whether there is an injection attack in SQL and NoSQL queries in web applications. Diglossia can protect against SQL and NoSQL injection attacks, which are attacks using the injection vulnerability [23].

SWAP (Secure Web Application Proxy) is a server-side security solution. SWAP can protect against attacks that use cross-site scripting vulnerability by running a reverse proxy on the system and thus controlling the content of all HTML responses [12].

PHAN (PHP Hybrid Analyzer) is a tool that runs in the PHP programming language, using a hybrid approach to security of web applications by combining the strengths of static and dynamic methods. Phan can protect against attacks using injection vulnerabilities [4].

PSIAQOP (Preventing SQL Injection Attack based on Query Optimization Process) is a unique approach that provides protection against all known SQL injection attacks by optimizing the query [26]. Query optimization depends on heuristic rules. PSIAQOP can protect against SQL injection attacks using the injection vulnerability [16].

V. CONCLUSIONS AND RECOMMENDATIONS

In this article, the top 10 web application vulnerabilities published by OWASP, vulnerability sources and security solutions used against attacks that exploit these vulnerability sources are discussed. According to the results obtained as a result of the research:

1. Checking and closing the vulnerabilities in web application security, which are summarized in this study and consist of application code, server and communication infrastructure,
2. The diversity of security solutions and the selection of these security solutions;
   a. The software language in which the application is developed,
   b. The number of servers the application is running on,
   c. The software and hardware features of the server on which the application is running,
   d. The web server program on which the application runs.
3. Attention to criteria such as
4. The majority of vulnerabilities are caused by application code flaws,
5. Information security awareness level of people who use web applications as well as web application developers is of great importance,

6. Making use of the application code security methods presented in this study in order to increase web application security,
7. Timely detection of attacks is very important in order to ensure application security and the detection methods presented in Chapter IV must be used,
8. It has been seen that the current gaps should be followed and eliminated.

With this study, it is aimed to contribute to increasing web application security by helping security experts to choose a security solution against which kinds of vulnerabilities in web application security.

REFERENCES

[1] Kuypers, Marshall A., Thomas Maillart, and Elisabeth Paté-Cornell. "An empirical analysis of cyber security incidents at a large organization." *Department of Management Science and Engineering, Stanford University, School of Information, UC Berkeley* 30 (2016).

[2] Alzahrani, Abdulrahman, Ali Alqazzaz, Ye Zhu, Huirong Fu, and Nabil Almashfi. "Web application security tools analysis." In *2017 ieee 3rd international conference on big data security on cloud (bigdatasecurity), ieee international conference on high performance and smart computing (hpsc), and ieee international conference on intelligent data and security (ids)*, pp. 237-242. IEEE, 2017.

[3] Huang, Hsiu-Chuan, Zhi-Kai Zhang, Hao-Wen Cheng, and Shiuhpyng Winston Shieh. "Web application security: threats, countermeasures, and pitfalls." *Computer* 50, no. 6 (2017): 81-85.

[4] ur Rehman, Habib, Mohammed Nazir, and Khurram Mustafa. "Security of web application: state of the art." In *International Conference on Information, Communication and Computing Technology*, pp. 168-180. Springer, Singapore, 2017.

[5] Kumar, Sandeep, Renuka Mahajan, Naresh Kumar, and Sunil Kumar Khatri. "A study on web application security and detecting security vulnerabilities." In *2017 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, pp. 451-455. IEEE, 2017.

[6] Shuaibu, Bala Musa, Norita Md Norwawi, Mohd Hasan Selamat, and Abdulkareem Al-Alwani. "Systematic review of web application security development model." *Artificial Intelligence Review* 43, no. 2 (2015): 259-276.

[7] Lazzez, Amor, and Thabet Slimani. "Forensics investigation of web application security attacks." *International Journal of Computer Network and Information Security* 7, no. 3 (2015): 10-17.

[8] Hasan, Ashikali M., Divyakant T. Meva, Anil K. Roy, and Jignesh Doshi. "Perusal of web application security approach." In *2017 International Conference on Intelligent Communication and Computational Techniques (ICCT)*, pp. 90-95. IEEE, 2017.

[9] Joshi, Chanchala, and Umesh Kumar Singh. "Performance evaluation of web application security scanners for more effective defense." *International Journal of Scientific and Research Publications (IJSRP)* 6, no. 6 (2016): 660-667.

[10] Sönmez, Ferda Özdemir. "Security qualitative metrics for open web application security project compliance." *Procedia Computer Science* 151 (2019): 998-1003.

[11] Peltier, T. R. (2016). Information Security Policies, Procedures, and Standards: guidelines for effective information security management. CRC Press.

[12] Burkhead, R. L. (2014). A phenomenological study of information security incidents experienced by information security professionals providing corporate information security incident management (Doctoral dissertation, Capella University).

[13] Iskandar, Akbar, Muhammad Resa Fahlepi Tuasamu, Suryadi Syamsu, M. Mansyur, Tri Listyorini, Sulfikar Sallu, S. Supriyono, Kundharu Saddhono, Darmawan Napitupulu, and Robbi Rahim. "Web based testing application security system using semantic comparison method." In *IOP Conference Series: Materials Science and Engineering*, vol. 420, no. 1, p. 012122. IOP Publishing, 2018.

[14] Agrawal, Alka, Mamdouh Alenezi, Rajeev Kumar, and Raees Ahmad Khan. "A unified fuzzy-based symmetrical multi-criteria decision-making method for evaluating sustainable-security of web applications." *Symmetry* 12, no. 3 (2020): 448.

[15] Li, Jinfeng. "Vulnerabilities mapping based on OWASP-SANS: a survey for static application security testing (SAST)." *Annals of Emerging Technologies in Computing (AETiC), Print ISSN* (2020): 2516-0281.

[16] Rexha, Blerim, Arbnor Halili, Korab Rrmoku, and Dren Imeraj. "Impact of secure programming on web application vulnerabilities." In *2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS)*, pp. 61-66. IEEE, 2015.

[17] Touseef, Pariwish, Khubaib Amjad Alam, Abid Jamil, Hamza Tauseef, Sahar Ajmal, Rimsha Asif, Bisma Rehman, and Sumaira Mustafa. "Analysis of automated web application security vulnerabilities testing." In *Proceedings of the 3rd International Conference on Future Networks and Distributed Systems*, pp. 1-8. 2019.

[18] Herdiana, Dody. "Website Security Analysis using Open Web Application Security Project 10." *J-Tin's-Jurnal Teknik Informatika* 1, no. 2 (2017).

[19] Babincev, Ivan M., and Dejan V. Vuletić. "Web application security analysis using the Kali Linux operating system." *Vojnotehnički glasnik* 64, no. 2 (2016): 513-531.

[20] Agrawal, Alka, Abhishek Kumar Pandey, Abdullah Baz, Hosam Alhakami, Wajdi Alhakami, Rajeev Kumar, and Raees Ahmad Khan. "Evaluating the security impact of healthcare Web applications through fuzzy based hybrid approach of multi-criteria decision-making analysis." *IEEE Access* 8 (2020): 135770-135783.

[21] Perera, Ashan Chulanga, Krishnadeva Kesavan, Sripa Vimukthi Bannakkotuwa, Chethana Liyanapathirana, and Lakmal Rupasinghe. "E-commerce (WEB) Application security: Defense against Reconnaissance." In *2016 IEEE International Conference on Computer and Information Technology (CIT)*, pp. 732-742. IEEE, 2016.

[22] Pooj, Karishma, and Sonali Patil. "Understanding File Upload Security for Web Applications." *International Journal of Engineering Trends and Technology* 42, no. 7 (2016): 342-347.

[23] Hakim, Hela, Asma Sellami, and Hanene Ben Abdallah. "Evaluating security in web application design using functional and structural size measurements." In *2016 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA)*, pp. 182-190. IEEE, 2016.