International Journal of Computer and Information System (IJCIS)
Peer Reviewed – International Journal
Vol : Vol. 03, Issue 03, September 2022
e-ISSN : 2745-9659
https://ijcis.net/index.php/ijcis/index

Website Vulnerability Testing and Analysis of Internet Management Information System Using OWASP

1st Diah Priyawati, 2nd Siti Rokhmah, 3rd Ihsan Cahyo Utomo
[1,3]Universitas Muhammadiyah Surakarta
[2]Institut Teknologi Bisnis AAS Indonesia Surakarta
[1,3]Jl. Ahmad Yani, Pabelan, Kartasura, Sukoharjo, Central Java, Indonesia 57162
[2]Jl. Slamet Riyadi No. 361 Windan, Makamhaji, Kartasura, Sukoharjo, Indonesia
[1]diah.priyawati@ums.ac.id, [2]alfathiey@gmail.com, [3]icu886@ums.ac.id

***Abstract*—** *Many businesses, organizations, and social institutions use websites to support their main tasks. The various benefits of the website must be supported by the security aspects of the website in order to avoid hacking. Cyber attacks or hackers can do dangerous things like get more valuable data. So it is necessary to test a good website to find out the level of vulnerability of application features in it. A suitable test for websites where the website is distributed over a network is the grey box penetration test. This study performs a gray box penetration testing technique using the OWASP method and the OWASP ZAP tool. The test steps are collecting test target information, performing automatic scanning with the help of OWASP ZAP, exploiting the scan results, reporting, and providing recommendations. The test results show the target application website has 12 vulnerabilities with 8.3% at the high level vulnerability or 1 alert, 41.7% at the medium level or 5 alerts, 33.3% at the low level or 4 alerts, and 16.7 at the informational level or 2 alerts. These vulnerabilities are related to matters related to A01-Broken Access Control, A03-Injection, A05-Security Misconfiguration, and A08-Software and Data Integrity Failures.*

Keywords : OWASP, Vulnerability testing, Website testing.

I. INTRODUCTION

Web programming consists of two words, namely programming and web. Programming is the process or algorithm of how the program works. The Web is a computer network of a collection of Internet sites offering text, graphics, sound and animated data sources via a hypertext transfer protocol [1]. The development of website technology is quite fast. Many businesses, organizations, and social institutions use websites to support their main tasks. The website can connect between customers, partners or employees in business activities. The website can also be a liaison between the community and government officials. The various benefits of the website must be supported by the security aspects of the website in order to avoid hacking.

According to data from the Ministry of Communications and Information Technology [2], cyber attacks against company websites in Indonesia increased by 31% from 2020 to 2021, from 206 attacks to 270 attacks per year. Writing wrong program code is one part that can be exposed to cyber attacks such as SQL injection, authentication, and XSS (cross site scripting). In 2021, data from webappsec.org shows cyber attacks against XSS reach 43% and SQL injection by 6% [3]. Cyber attacks or hackers can do dangerous things like get more valuable data. So it is necessary to test a good website to find out the level of vulnerability of application features in it. so the website is strong against attempted cyber attacks.

Software testing is an important process of software development, namely the verification process to determine software quality. The software testing stage has a portion of 30% of the resources (time and budget) of software development [4]. So the selection of testing techniques must be correct in order to achieve the level of reliability, maintainability, security, attack resistance and efficiency. Software testing techniques that can be used to find out system errors or system penetrations are divided into three, namely white box, black box, and grey box [5], [6].

White box testing is a detailed system testing method that corrects the structure of the code so that the test team requires in-depth knowledge of the code in the system. Black box testing is a way of system testing that does not require in-depth knowledge of system code. Black box testing only examines the fundamental aspects of system performance. Black box testing is a way of system testing that does not require in-depth knowledge of system code. Black box testing only examines the fundamental aspects of system performance. Whereas the grey box testing is a combination of white box and black box testing. Grey box testing is a system testing method that does not require in-depth knowledge of system code but has basic system performance capabilities. A suitable test for websites where the website is distributed over a network is the grey box penetration test [7], [8]. The superior method for testing website security vulnerabilities is the OWASP method and applies the OWASP top vulnerabilities list [9].

This study aims to perform a grey box penetration testing technique using the OWASP method to determine the vulnerability of the features in it and analyze the test results so that it can provide information for developers to improve security. The object of the attack test in this study is one of website application of Kragan, Karanganyar, for village internet management.

II. RESEARCH METHODS

2.1. Website Vulnerability Testing

Testing website security systems with penetration techniques is testing by simulating the occurrence of cyber attacks on the website being tested. System testing can be

done manually or using tools to support the attack [10]. Manual testing is also very useful because some types of application vulnerabilities can only be discovered through the observations of testers. while testing uses tools that can be used for testing such as google chrome, brutus, webscarab, wfuzz, dirb, OWASP ZAP, Mozilla firefox. Penetration testing is carried out regularly in order to minimize cyber attacks that penetrate the security of website applications. The tester will carry out attacks on the website application and try to penetrate its security. If the tester manages to penetrate the security of the website, the test results become information to improve the website application system and continue to improve its security.

2.2. Security Level Assessment with OWASP

One type of risk level assessment of website application vulnerabilities is OWASP (Open Web Application Security Project) [11]. OWASP was developed with the aim of providing a data source so that application developers can learn about website security systems and improve their security. OWASP 10 is a list of 10 website security standards so that developers can make sure the website is safe or not, by doing a checklist based on these standards [12], including.

1. Injection: is a cyber attack by sending untrusted data. The security measure is to reject data that looks suspicious,
2. Broken authentication: is an attack on the login system to be able to enter a user account and control the system. The form of security is to use 2-factor authentication,
3. Sensitive data exposure: is an attack by stealing sensitive data. The security step is to encrypt sensitive data and ensure the website does not store sensitive data that is not needed,
4. XML external entities: is an attack on XML input and recognizes its weaknesses. The security step is to build a website application with the type of data that is not too complex,
5. Broken access control: or a control system whose functionality is not good enough to allow cyber attacks to enter the system,
6. Security misconfiguration: is caused by the system still using the default configuration regardless of the need.
7. Cross site scripting (XSS): is to allow website users to add custom code to the URL path so that this feature becomes a loophole for attacks,
8. Insecure deserialization: it can be one side of cyber attacks by utilizing untrusted data so that it can damage DDOS. The security measure is to prohibit deserialization of untrusted data,
9. Using components with known vulnerabilities: is an additional feature that is sometimes used by developers to make web applications more efficient. However, this becomes part of the attack if the feature libraries are not updated,
10. Insufficient logging and monitoring: is an action to monitor websites so that they can be quickly identified in the event of an attack. The average developer does not realize if the website is attacked and only realizes the attack after 200 days.

2.3. Research Flow

The OWASP risk assessment methodology is a simple approach to calculating and assessing the risks associated with an application and deciding what to do about those risks. Knowing the risks that will occur will be beneficial to the developer, save time and reduce the occurrence of more serious risks [13]. The stages of this research consist of 5 stages, namely planning, scanning, gaining, maintaining access, and analysis as shown in Figure 1.

The first stage, planning, is to determine the scope and objectives of the test including the system to be handled and the testing method to be used. This stage ends by collecting information for testing such as network and domain information, existing servers, how to work and the flow of the computer system used.

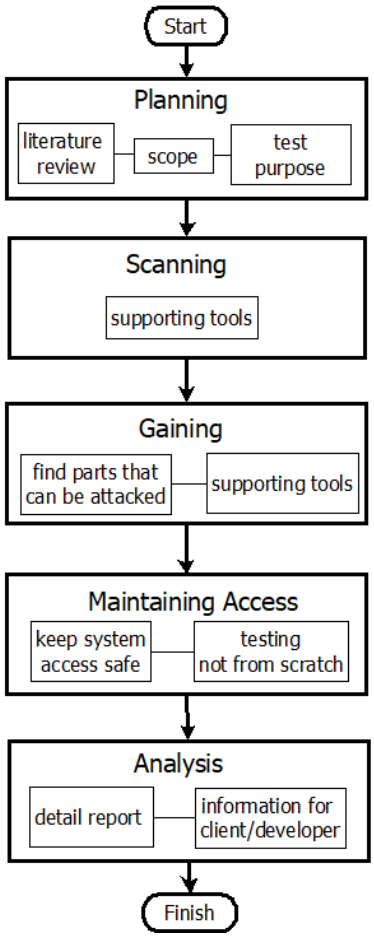

Figure 1. Research Flow

The second stage, scanning, is to move the target using various supporting tools to find security holes that can be done to enter the system. Zed Attack Proxy (ZAP) from OWASP is one that is recommended because it is easy, free of charge, and capable of performing automatic scans of the target system to determine which parts of the system are vulnerable to cyber attacks. open ports and these vulnerabilities can be used for gaining access or exploitation stages.

The third stage, gaining access, is an attempt to break into the system based on the scanning results in the previous stage. Security vulnerabilities can be in the form of XSS, SQL injection, and backdoors. Testers can try to exploit security holes and take root or administrator accounts, steal data, or intercept traffic.

The fourth stage, maintaining access, is the process of testing whether the access rights of previous attacks can be kept open or closed. The purpose of this step is to ensure that the attacker's permissions can be closed so that the system can be safe again.

The fifth stage, analysis, is the result of penetration testing which is compiled into a complete report consisting of exploitable security vulnerabilities, data that can be retrieved, and the time taken to break into the system.

III. RESULT AND ANALYSIS

The stages of testing in this study are shown in Figure 2. Testing begins with collecting target information, selecting, proving system vulnerabilities, test result information and recommendations.

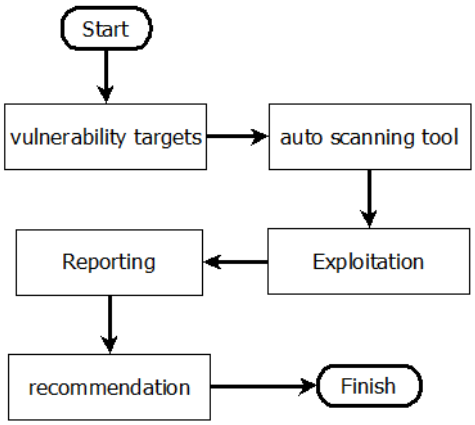

Figure 2. Application Testing Flow

3.1 Vulnerability Targets

This stage is preparation (planning) before penetration testing on the website application. User data is needed for the internet management website application of Kragan. Testing support equipment using computer assisted audit technique (CAAT) and OWASP ZAP tools as shown in Table 1.

Table 1. Equipment Specifications

Tool Name	Spesification
Laptop	OS: Windows 10 64 bit Processor : Intel Core i7-8565U quad-core 1,8GHz RAM : 24.0 GB DDR4 VGA : NVidia GeForce MX250 VRAM 2GB GDDR5 SSD : 500GB
OWASP ZAP	Version 2.11.1
Internet Connection	Up to 100Mbps
Web Browser	Google Chrome Version 105.0.5

3.2 Vulnerability Scanning

This stage is to scan the vulnerabilities of the internet management website application of Kragan using OWASP ZAP with an automated scan menu. Based on the information that has been collected in the previous stage, scanning process produces 1 high alerts, 5 medium alerts, 4 low alerts, and 2 informational alerts. Scan results can be seen in figure 3.

Figure 3. Scan result from target website

3.3 Exploitation

This stage aims to verify the results of the vulnerability scanning. Exploitation is done manually because some types of application vulnerabilities can only be identified through the observations of testers. The exploitation results obtained are as follows:

1. SQL Injection Vulnerability: SQL Injection is a vulnerability caused by input parameters that are not validated and executed by DBMS queries. The attacker will send a payload in the form of an SQL script on the input parameters. A successful attack will cause the SQL script sent to be executed by the DBMS query,
2. Absence of Anti-CSRF Tokens Absence Vulnerability: Absence of Anti-CSRF Tokens is a vulnerability caused by not using anti-CSRF tokens on the form. This vulnerability can be used by attackers to perform cross-site request forgery attacks on applications. Verification is done by checking the response from the application manually.
3. Content Security Policy (CSP) Header Not Set Vulnerability: is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page-covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files. Verification is done by checking the response from the application manually.
4. Directory Browsing Vulnerability: It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files, backup source files, etc. which can be accessed to read sensitive information. OWASP ZAP provides a solution to disable directory browsing. If this

is required, make sure the listed files does not induce risks. Verification is done by checking the response from the application manually.

5. Missing Anti-clickjacking Header Vulnerability: The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks. Verification is done by checking the response from the application manually. Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by the website.
6. Vulnerable JS Library: Identified Jquery library, version 3.4.1 which is vulnerable to attack. Verification is done by checking the response from the application manually by upgrading to the latest version of jquery.
7. Cookie Without SameSite Attribute Vulnerability: is caused by not including the same site attribute for cookies in the http response. This vulnerability is on the web server side that can be exploited by attackers to carry out cross-site request forgery attacks, cross-site script inclusion, and timing attacks. Verification is done by checking the http response from the web server. The results of this vulnerability verification found http response does not include the same site attribute as Figure 4.

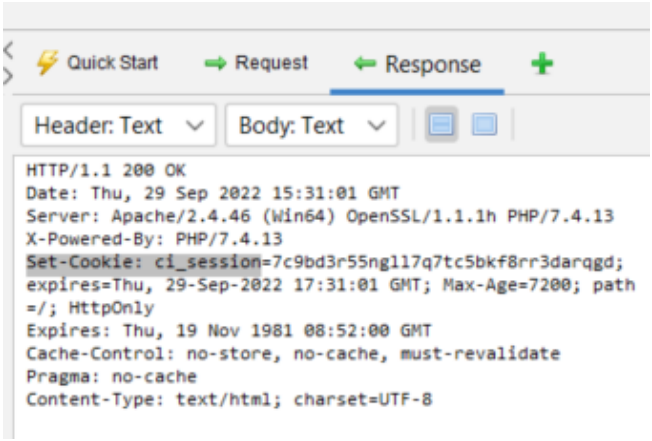

Figure 4. HTTP response web server application

8. Server leaks information via "X-Powered-By" HTTP Response Header Field(s): Access to such information may facilitate attackers identifying other frameworks/components of website application is reliant upon and the vulnerabilities such components may be subject to. Verification is done by checking the response from the application manually.
9. Timestamp Disclosure – Unix Vulnerability: is caused by displaying unix timestamp information in the browser. This vulnerability can be exploited by attackers as a means of gathering information to carry out attacks. Verification is done manually by checking the timestamp that appears to contain important information or not.
10. X-Content-Type-Options Header Missing Vulnerability: This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Verification is done manually by ensuring that the website application sets the Content-Type header appropriately and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.
11. Information Disclosure – Suspicious Comments Vulnerability: is caused by commenting out coding that is considered suspicious or contains sensitive data. This vulnerability can be exploited by attackers to gather information and carry out attacks. Verification is done by checking the comments out of the application manually.
12. Loosely Scoped Cookie Vulnerability: Cookies can be scoped by domain or path. This check is only concerned with domain scope. The domain scope applied to a cookie determines which domains can access it. Verification is done manually by changing scope cookies to a fully qualified domain name.

3.4 Reporting and Recommendation

This stage aims to report the part of the system that is a security vulnerability of the website. The test results using OWASP Zap found several vulnerabilities based on OWASP Top 10 2021 which can be seen in Table 2.

Table 2. Vulnerabilities Based on OWASP Top 10 2021

Vulnerability	Recommendation
A01:2021-Broken Access Control	Implementing access control system work and reused in all applications so as to minimize CORS (Cross Origin Resource Sharing) users, so that users cannot create, read, update, or delete records freely. The access control model should limit this by using ownership for each record, and disable the web server listing directory
A03:2021-Injection	prevented by validating and cleaning the data entered by the user. The validation referred to here is to reject data that looks suspicious. Cleaning data means deleting all suspicious data.
A05:2021-Security Misconfiguration	System management can review and correct appropriate configurations for all security notes, updates and patches as part of the patch management process
A08:2021-Software and Data Integrity Failures	Use digital signatures or similar mechanisms to verify that software or data comes from an expected source and is not manipulated

The results of system testing vulnerabilities using OWASP Zap tools found four vulnerabilities that must be fixed, namely A01-Broken Access Control, A03-Injection, A05-Security Misconfiguration, and A08-Software and Data Integrity Failures.

VI. CONCLUSION

Website vulnerability testing on the internet management website application of Kragan, Karanganyar, has been successfully conducted with the grey box penetration testing method using computer-assisted audit techniques and the OWASP ZAP tool. Based on the results of the testing and analysis above, it shows that the internet management website application of Kragan, Karanganyar, has 12 vulnerabilities with 8.3% at the high level vulnerability or 1 alert, 41,7% at the medium level or 5 alerts, 33.3% at the low level or 4 alert, and 16.7 at the informational level or 2 alerts. The test results can be used by IT officers as a reference to improve the security of the website application, especially regarding matters related to A01-Broken Access Control, A03-Injection, A05-Security Misconfiguration, and A08-Software and Data Integrity Failures.

REFERENCES

[1] R. R. Rerung, *Pemrograman Web Dasar*. Deepublish, 2018.

[2] B. H. K. Kominfo, "Kementerian Komunikasi dan Informatika." https://www.kominfo.go.id/content/detail/43363/siaran-pers-no-306hmkominfo072022-tentang-tantangan-keamanan-siber-makin-besar-indonesia-dorong-tata-kelola-data-lintas-negara/0/siaran_pers (accessed Sep. 02, 2022).

[3] I. M. Edy Listartha, I. M. A. Premana Mitha, M. W. Aditya Arta, and I. K. W. Yuda Arimika, "Analisis Kerentanan Website SMA Negeri 2 Amlapura Menggunakan Metode OWASP (Open Web Application Security Project)," *Simkom*, vol. 7, no. 1, pp. 23–27, 2022, doi: 10.51717/simkom.v7i1.63.

[4] S. Roohullah Jan, S. Tauhid Ullah Shah, Z. Ullah Johar, Y. Shah, and F. Khan, "An Innovative Approach to Investigate Various Software Testing Techniques and Strategies," *Int. J. Sci. Res. Sci. Eng. Technol.*, vol. 2, no. 2, pp. 682–689, 2016.

[5] M. Ehmer and F. Khan, "A Comparative Study of White Box, Black Box and Grey Box Testing Techniques," *Int. J. Adv. Comput. Sci. Appl.*, vol. 3, no. 6, pp. 12–15, 2012, doi: 10.14569/ijacsa.2012.030603.

[6] A. Bansal, "A Comparative Study of So ware Testing Techniques A Comparative Study of Software Testing Techniques," *IJCSMC J.*, vol. 3, no. 6, pp. 579–584, 2014, [Online]. Available: www.ijcsmc.com%0AInternational.

[7] I. R. Dhaifullah, H. Muttanifudin H, A. A. Salsabila, and M. A. Yakin, "Survei Teknik Pengujian Software," *JACIS J. Autom. Comput. Inf. Syst.*, vol. 2, no. 1, pp. 31–38, 2022.

[8] W. D. W. I. Nugroho, "Pengujian Greybox pada Sistem Pemantauan Kualitas Udara dalam Ruangan Berbasis Arduino Mega WAAHID DWI NUGROHO, Unan Yusmaniar O, S.T., M.Sc., Ph.D.," 2021.

[9] B. T. K. Dewi and M. A. Setiawan, "Kajian Literatur: Metode dan Tools Pengujian Celah Keamanan Aplikasi Berbasis Web," *Automata*, 2022, [Online]. Available: https://journal.uii.ac.id/AUTOMATA/article/view/21883.

[10] L. Dukes, X. Yuan, and F. Akowuah, "A case study on web application security testing with tools and manual testing," *Conf. Proc. - IEEE SOUTHEASTCON*, 2013, doi: 10.1109/SECON.2013.6567420.

[11] E. B. Setiawan and A. Setiyadi, "Web vulnerability analysis and implementation," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 407, no. 1, pp. 0–8, 2018, doi: 10.1088/1757-899X/407/1/012081.

[12] R. Revo, G. Made, A. Sasmita, I. P. Agus, and E. Pratama, "Testing for Information Gathering Using OWASP Testing Guide v4 (Case Study : Udayana University SIMAK-NG Application)," *J. Ilm. Teknol. dan Komput.*, vol. 1, no. 1, 2020.

[13] B. Ghozali, K. Kusrini, and S. Sudarmawan, "Mendeteksi Kerentanan Keamanan Aplikasi Website Menggunakan Metode Owasp (Open Web Application Security Project) Untuk Penilaian Risk Rating," *Creat. Inf. Technol. J.*, vol. 4, no. 4, p. 264, 2019, doi: 10.24076/citec.2017v4i4.119.