FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 9 No. 12, December (Special Issue), 2025, pp 79 – 85
DOI: https://doi.org/10.33003/fjs-2025-0912-3163

AN ASSESSMENT OF SECURITY VULNERABILITIES IN LIBREHEALTH HER

Yusuf, A. S.

Department of Computer Science, Federal University Dustin-Ma, Katsina State, Nigeria.

*Corresponding authors' email: mailyusufabubakarsadiq@gmail.com

ABSTRACT

Vulnerability assessment is an aspect of cyber security that involves investigating a system to discover its security weaknesses and also to provide a measure to protect the system against possible exploit due to its weakness. In this digital age, it is essential for businesses to protect their systems against cyber attack. Open source software is a free to use system provided and maintained by an individual or group of individuals, such system is relied on by some individuals and businesses to carry out their daily operation, LibreHealth EHR is an open source Electronic Health Record with a moderate level of popularity and engagement within the GitHub community. Open source applications such as LibreHealth EHR could also be susceptible to network attack and Such event could be catastrophic to its user. This study explores an instance of LibreHealth EHR to assess its security vulnerabilities using Zed Attack Proxy and Burp Suite security tools. 48 issues were discovered with these tools. Librehealth EHR was tested against the vulnerabilities identified by the security tools to verify their certainty, the outcome of the test revealed that out of the 48 issues 36 of them were false positives while the remaining 12 were true positives issues. Thereafter a remediation action was taken against the few true positive issues. Hence improving the application security performance.

Keywords: Web Application Security, Open Source Software, Penetration Testing Tools.

INTRODUCTION

Vulnerability assessment is an aspect of cyber security that aims to identify security weaknesses in a system. This assessment is commonly performed using automated security scanning tools such as zed attack proxy, burp suite, wapiti, arachni, acunetix e.t.c (Al Anhar & Suryanto, 2021, Aljebry, Alqahtani & Sulaiman 2022, Onyango & Wang, 2024), alternatively the assessment could also be performed manually (Singh, Meherhomji, & Chandavarkar, 2020, Rane & Qureshi, 2024). The number of new cyber threats keeps doubling and the cost of a system exploit can be prohibitively catastrophic (Laksmiati, 2023, Al Anhar & Suryanto, 2021, Dong, Liu, & Wu, 2022). Cyber security has become an important facet for most businesses, in this digital age, it is essential for businesses to protect their system against cyber attack such as data theft, denial of service, phishing, malware, password cracking e.t.c. to ensure order in business activity (Laksmiati, 2023).

A lot of business and individual users rely on open source software to meet their business needs, an open source software is an alternative to commercial software, it is a free to use application provided and maintained by an individual or group of individuals (Ahmad, 2021). This can be a very cost effective solution to some businesses, however Such application, when in operation can also be susceptible to network attacks, and the event of an attack can be catastrophic to the end user.

LibreHealth EHR is an open source Electronic Health Record with a moderate level of popularity and engagement within the GitHub community. LibreHealth EHR is starred by 225 users, watched by 255 users, and forked by 39 users.

The proposed study aims to explore an instance of LibreHealth EHR to assess its security vulnerabilities and provide a security countermeasure. Hence improving the application security performance. The rest of the paper contains the related work, methods, result and the conclusion section.

Related Work

Wordpress being one of the popular target for attack by hackers, Laksmiati, (2023) Conducted a vulnerability assessment on a wordpress based website using NMap and WPScan network scanner, result shows that wordpress based site are associated with a lot of security weakness, they also emphasis the importance of conducting regular vulnerability assessments on WordPress-based websites to reduce the risk of security attack. Dong, Liu, & Wu, (2022) also considered network attacks for a business application deployed on cloud as an example and also provide the security countermeasures for the application**.** Priyawati, Rokhmah, & Utomo, (2022) performs a gray box penetration testing on an internet management website using the OWASP method and the OWASP ZAP tool. Among the 12 vulnerabilities detected are 5 medium level 4 low level and 2 informational level and 1 high level vulnerability. The vulnerabilities are related to Broken Access Control, Injection, Security Misconfiguration, and Software and Data Integrity Failures. Wibowo & Sulaksono (2021), also used OWASP Security Shepherd to assess XSS vulnerability on an application. Karayat, Jadhav, Kondaka, & Nambiar, (2022) also perform similar investigation by developing an automated tool using python with an integration of skipfish.

There is a performance variance among the varieties security scanners available. For instance Al Anhar, & Suryanto, (2021) conducted an evaluation studies to analyze the performance of OWASP ZAP, Wapiti, Arachni, and Burp Suite Professional using a NodeJs application as a benchmark, the result of the study shows that Burp Suite Professional outweigh other scanner with having the best True Positive (TP) and Recall values, while Arachni for perfect Precision valued for both benchmark targets. Similarly Aljebry, Alqahtani & Sulaiman (2022) analyzed the performance of ZAP and Acunetix, the investigation results show that zap is better. Onyango & Wang (2024) investigated the effectiveness of various testing tools within the kali linux environment by simulating various kinds of attacks with these tools, the analysis of the investigation shows that burp Suite and OWASP ZAP demonstrated robust performance in

scanning vulnerabilities while Metasploit and BeEF demonstrated greater exploitation techniques.

While some of these scanning tools are commercial and some are non-commercial, Burp Suite seems to be better among the commercial and OWASP ZAP among the non-commercial. As investigated by Albahar, Alansari & Jurcut (2022), in a score based comparative study on commercial and non-commercial web scanning tools. The results showed that Burp Suite Professional scored the highest out of the commercial tools, while OWASP ZAP scored the highest out of the non-commercial tools. Similarly, Jarupunphol, Seatun & Buathong, (2023). also compared Burp Suite and OWASP ZAP using a university web application as a sample, the result showed that Burp Suite detected more vulnerabilities and alerts than OWASP ZAP, with a higher proportion of high-risk vulnerabilities. However, OWASP ZAP had a higher proportion of medium-confidence vulnerabilities.

As it is evidenced from the above that Burp Suite and OWASP ZAP are more popular and effective, in this study, both OWASP ZAP and Burp Suite are used to assess LibreHealth EHR security vulnerabilities.

Research Gap

Several studies have been conducted on assessing vulnerabilities of web applications, both commercial and non commercial, using various security scanning tools, additionally It has also been evidenced from some comparative studies that their exist performance variance among these tools, and recent studies has shown that Burp suite among the commercial tools is more effective and ZAP among the non-commercial tools is more effective. In this study, both OWASP ZAP and Burp Suite are used to assess LibreHealth EHR security vulnerabilities.

MATERIALS AND METHODS

The study employs a qualitative methodology of research by reviewing existing related studies, a penetration testing was conducted on LibreHealth EHR using OWASP ZAP and Burp Suite to discover security vulnerabilities in the application, application code was also reviewed and tested as a countermeasure to the system vulnerabilities.

Table 1: Summary of Related Studies

Authors	Methods	Result
Laksmiati, D. (2023)	Used Nmap and WPScan to conduct vulnerability assessment on a WordPress based website.	Several vulnerabilities were identified relating to open ports,configurations and sensitive information leak.
Dong, G., Liu, F., & Wu, G. (2022)	analyzed the network attacks intercepted by a cloud based website.	Analysis result shows that website suffered from Code execution SQL injection Webshell and Local include
Priyawati, D.,Rokhmah, S., & Utomo, I. C. (2022)	performed a gray box penetration testing technique using the OWASP method and the OWASP ZAP tool.	The test results show the target application website has 12 vulnerabilities related to A01-Broken Access Control, A03-Injection, A05-Security Misconfiguration, and A08-Software and Data Integrity failure.
Wibowo, R. M., & SulaksonA. (2021)	Also used OWASP Security Shepherd to perform simulation experiment on XSS vulnerabilities on web application	N/A
Karayat,R.,Jadhav,M.,Kondaka, L.S.,&Nambiar, A.(2022).	Employed skipfish and an independent automated tool developed with python for detection of vulnerability on a python goat.	Vulnerabilities detected include SQLI, XSS & a Spider(Content Discovery Tool) os command injection
Al Anhar, A., & Suryanto, Y. (2021).	Performed evaluation study on web security Scanners tools such as OWASP ZAP, Wapiti, Arachni, and Burp Suite Professional with NodeJS-based benchmark targets.	The result of the study shows that Burp Suite Professional outweighs other scanners with having the best True Positive (TP) and Recall values, while Arachni for perfect Precision value for both benchmark targets.
Aljebry, A. F., Alqahtani, Y. M., &Sulaiman, N. (2022)	Analyzed the performance of ZAP and Acunetix	Investigation result shows that zap is more effective.
Onyango, C., & Wang, X. (2024)	analyzed the effectiveness of the various pen testing tools, ZAP Burp Suite, Metasploit and BeEF.	The analysis of the investigation shows that burp Suite and OWASP ZAP demonstrated robust performance in scanning vulnerabilities while Metasploit and BeEF demonstrated greater exploitation techniques.
Albahar, M., Alansari, D., & Jurcut, A. (2022).	Conducted a score based comparative study on commercial and non-commercial web scanning tools.	The results showed that Burp Suite Professional scored the highest out of the commercial tools, while OWASP ZAP scored the highest out of the non-commercial tools.
Jarupunphol, P., Seatun, S., & Buathong, W. (2023).	Compared Burp Suite and OWASP ZAP using a university web application as a sample.	the result showed that Burp Suite detected more vulnerabilities and alerts than OWASP ZAP, with a higher proportion of high-risk vulnerabilities. However, OWASP ZAP had a higher proportion of medium-confidence vulnerabilities.

ZAP (Zed Attack Proxy)

ZAP is a non-commercial integrated penetration testing tool provided by OWASP (Open Web Application Security) for discovering vulnerabilities in web applications (Altulaihan, Alismail, & Frikha, 2023). ZAP was firstly used to assess the security weakness in LibreHealth EHR. Figure 1 presents a chart of the number of security vulnerabilities and their categories discovered with the ZAP tool.

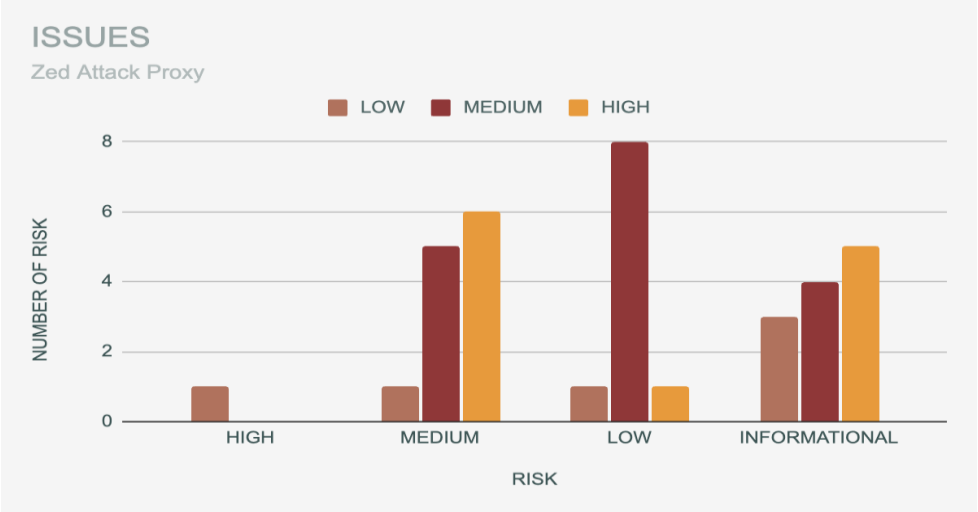

Figure 1: ZAP tool identified vulnerabilities

A total of 35 issues were identified with ZAP tools, these issues include 1 high issue (path traversal), 12 medium issues (absence of anti-csrf tokens, application error disclosure, buffer overflow, content security policy (wildcard directive), content security policy (script- src unsafe-eval), content security policy (script- src unsafe-inline), content security policy (style- src unsafe-eval), content security policy (header not set), directory browsing, hidden file found, missing anti-clickjacking header, vulnerable JS Library), 10 low issues (big redirect detected(potential sensitive information leak), cookie no httponly flag, cookie without samesite attribute, cross-domain javascript source file inclusion, information disclosure- debug error messages, private ip disclosure, server leaks information via x-powered-by http response header field(s), server leaks version information via server http response header field, timestamp disclosure - unix, x-content-type-options header missing), 12 informational issues (authenticated request identified, csp: x-content-security-policy, csp: x-webkit-csp, content-type header missing, information disclosure-sensitive information in url, information disclosure suspicious comments, loosely scoped cookie, modern web application, obsolete content security policy, session management response identified, user agent fuzzer, user controllable HTML element attribute (potential XSS)). Among these issues are 3 % high issue with low confidence, 3% moderate issue with low confidence, 14% moderate issue with moderate confidence, 17% moderate issue with high confidence, 3% low issues with low confidence, 23% low issues with moderate confidence, 3% low issues with high confidence, 3% Low issues with low confidence, 9% Informational issues with low confidence, 11% Informational issues with medium confidence and 14% Informational issues with high confidence.

Vulnerabilities Verification

Libra Health EHR was tested to determine the certainty of the identified issues from ZAP tools. Figure 2 presents the outcome of the verification process.

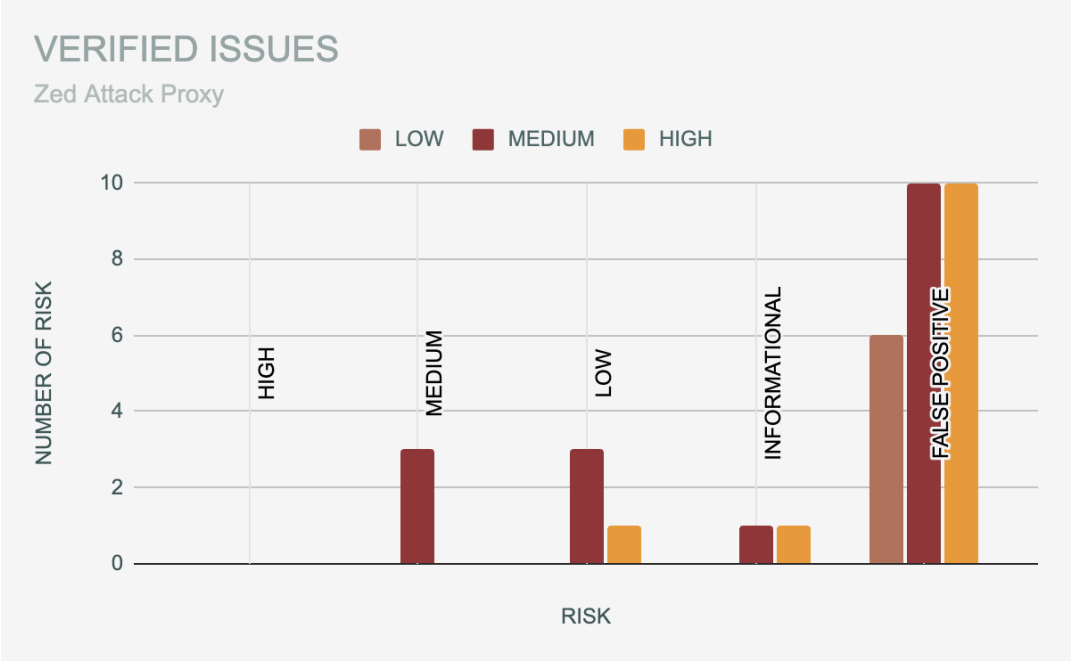

Figure 2: Verified vulnerabilities chart

The result of the verification reveals that, out of the 35 issues identified, 26 of them are false positives, 9 are certain issues. The 9 certain issues includes 4 medium issues 3 with medium confidence (Directory Browsing, Missing Anti-clickjacking Header, Vulnerable JS Library,), 4 low issues, 3 with medium confidence, 1 with high confidence (Cookie No HttpOnly Flag, Cookie without SameSite Attribute, Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s), Server Leaks Version Information via "Server" HTTP Response Header Field), 2 informational issues, 1 with medium confidence, the other with high confidence (Authentication Request Identified, Modern Web Application).

Burp Suite

Burp Suite is a commercial web application security scanning tool for intercepting traffic generated by a web browser during an interaction with an application (Bouafia, Benbrahim & Amine, 2023). After the security assessment with ZAP, and a remediation action based on the issue identified. Burp Suite was further used to test LibreHealth EHR. Figure 3 presents a chart of the number of security vulnerabilities and their categories discovered with the Burp Suite tool.

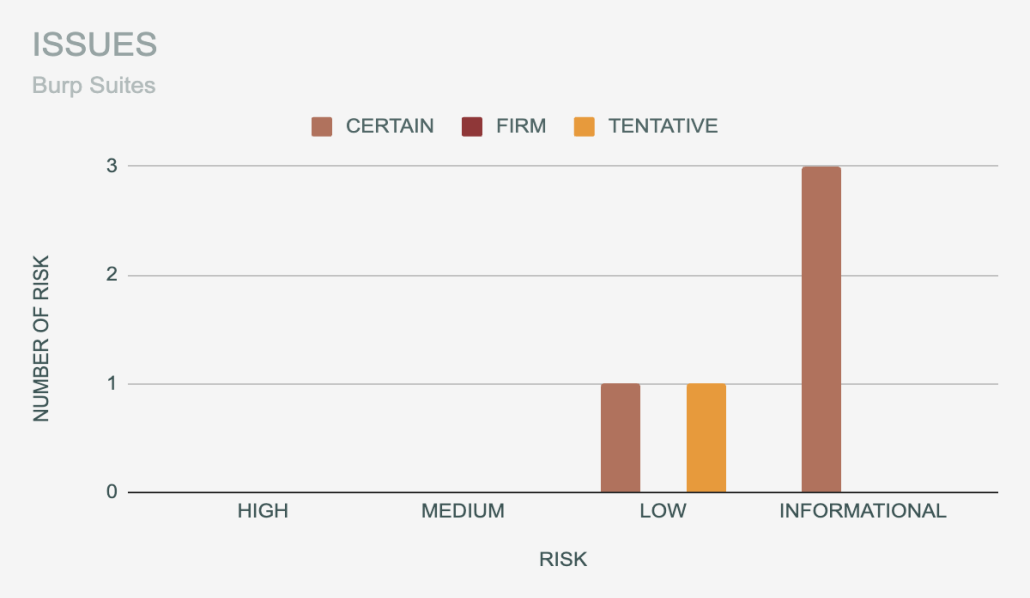

Figure 3: Burp Suite tool identified vulnerabilities chart

The result of the further assessment with burp suite indicated a significant improvement in Libre health security performance. A total of 7 issues were identified, which were mainly low issues and informational issues, these issues include 1 low issues with certain confidence(Unencrypted communications), 1 low issues with tentative confidence(Vulnerable JavaScript dependency), 3 informational issues with certain confidence (HTTP TRACE method is enabled, Email addresses disclosed, HTML does not specify charset).

Vulnerabilities Verification

Libra Health EHR was also further tested to determine the certainty of the identified issues from Burp Suite. Figure 4 present the outcome of the verification process.

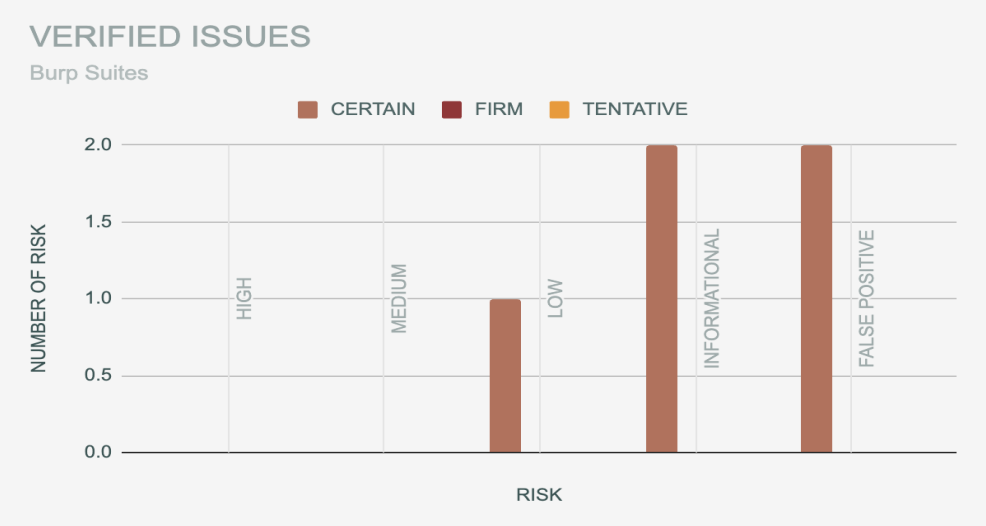

Figure 4: Verified vulnerabilities

Figure 4 is a result of verifying vulnerabilities identified with burp suite scanners. The result reveals that, out of the 5 issues identified, 2 of them are false positives, 3 are certain issues. The 3 certain issues includes 1 low issue with certain confidence, (Vulnerable JS Library), 2 medium issues with certain confidence (HTTP TRACE method is enabled, HTML does not specify charset)

Preventative Measure

A remediation action was taken to tackle the security issues ascertained with LibreHealth EHR, the following table presents the verified issues and their preventative measure.

Table 2: Preventative Measures

Issue 1	**Directory Browsing Medium**
Description	It is possible to view the directory listing in the application. Directory listing may reveal hidden scripts, include files, backup source files, etc. which can be accessed to read sensitive information.
Counter Measure	Configure web server to disable directory listing
Security Mechanism Improvement	Including the following code in the .htaccess file Options -Indexes
Issue 2	**Missing Anti-clickjacking Header Medium**
Description	Click-Jacking attack is a form of attack whereby an innocent web user is deceived to click on an unintended link. This attack usually occurs when a response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.
Counter Measure	Implementing X-Frame-Options HTTP headers to SAMEORIGIN.
Security Mechanism Improvement	Denying X-Frame option in httpd.conf web server configuration file using the following lines of code Header always append X-Frame-Options "sameorigin"
Issue 3	**Vulnerable JS Library Medium**
Description	**Jquery** version **3.1.1** was found among the application files which has the following vulnerabilities i. CVE-2019-11358: jQuery before 3.4.0, as used in Drupal, Backdrop CMS, and other products, mishandles jQuery.extend(true, {}, ...) because of Object.prototype pollution ii. CVE-2020-11023: passing HTML containing <option> elements from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. iii. CVE-2020-11022: Regex in its jQuery.htmlPrefilter sometimes may introduce XSS
Counter Measure	Upgrading jquery to the latest version.
Security Mechanism Improvement	Using the latest jquery version
Issue 4	**Cookie No HttpOnly Flag Low**
Description	A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.
Counter Measure	Ensure that the HttpOnly flag is set for all cookies.
Security Mechanism Improvement	HttpOnly parameter of the cookie is set to true
Issue 5	**Cookie without SameSite Attribute Flag Low**
Description	A cookie has been set without the SameSite attribute, which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective countermeasure to cross-site request forgery, cross-site script inclusion, and timing attacks.
Counter Measure	Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.
Security Mechanism Improvement	SameSite parameter of the cookie is set to strict
Issue 6	**Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) Flag Low**
Description	The web/application server is leaking information via one or more "X-Powered-By" HTTP response headers. Access to such information may facilitate attackers identifying other frameworks/components your web application is reliant upon and the vulnerabilities such components may be subject to.
Counter Measure	Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.
Security Mechanism Improvement	Disabled the X-Powerd header by from the httpconf file using the below lines of code Header always unset X-Powered-By

Issue 7	Server Leaks Version Information via "Server" HTTP Response Header Field Flag **Low**
Description	The web/application server is leaking version information via the "Server" HTTP response header. Access to such information may facilitate attackers identifying other vulnerabilities your web/application server is subject to.
Counter Measure	Ensure that the web server is configured to suppress the "Server" header or provide generic details.
Security Mechanism Improvement	Remove server header completely by configuring the httpconf file using the below lines of code ServerTokens Prod ServerSignature Off
Issue 8	**Authentication Request Identified Informational**
Description	A request was identified as an authentication request.
Counter Measure	This is an informational alert rather than a vulnerability and so there is nothing to fix.
Issue 9	**Modern Web Application Informational**
Description	The Application under test appears to be a modern web application.
Counter Measure	This is an informational alert and so no changes are required.
Issue 10	**HTTP TRACE method is enabled Informational**
Description	The HTTP TRACE method is designed for diagnostic purposes. If enabled, the web server will respond to requests that use the TRACE method by echoing in its response the exact request that was received. This behavior is often harmless, but occasionally leads to the disclosure of sensitive information such as internal authentication headers appended by reverse proxies. This functionality could historically be used to bypass the HttpOnly cookie flag on cookies, but this is no longer possible in modern web browsers.
Counter Measure	The TRACE method should be disabled on production web servers.
Security Mechanism	Setting TraceEnable Off in httpd.conf configuration file
Issue 11	**HTML does not specify charset Informational**
Description	If a response states that it contains HTML content but does not specify a character set, then the browser may analyze the HTML and attempt to determine which character set it appears to be using. Even if the majority of the HTML actually employs a standard character set such as UTF-8, the presence of non-standard characters anywhere in the response may cause the browser to interpret the content using a different character set. This can have unexpected results, and can lead to cross-site scripting vulnerabilities in which non-standard encodings like UTF-7 can be used to bypass the application's defensive filters. In most cases, the absence of a charset directive does not constitute a security flaw, particularly if the response contains static content.
Counter Measure	The response is a static content page, also it is just an informational alert and so no changes are required.

CONCLUSION

The study conducted a vulnerability assessment on librehealth EHR using security tools (Zed Attack Proxy, Burp Suite), LibrehealtEhr was tested against the issues identified by the security tools to verify their certainty, the outcome of the test revealed that out of the 48 issues identified by theses security tools 36 of them were false positives while the remaining 12 were true positives issues. Thereafter a remediation action was taken against the few valid issues, to improve the application security performance.

REFERENCES

Laksmiati, D. (2023). Vulnerability assessment with network-based scanner method for improving website security. *Journal of Computer Networks, Architecture and High Performance Computing*, *5*(1), 38-45.

Al Anhar, A., & Suryanto, Y. (2021, June). Evaluation of web application vulnerability scanner for modern web application. In *2021 International Conference on Artificial Intelligence and Computer Science Technology (ICAICST)* (pp. 200-204). IEEE.

Priyawati, D., Rokhmah, S., & Utomo, I. C. (2022). Website vulnerability testing and analysis of website application using OWASP. *International Journal of Computer and Information System (IJCIS)*, *3*(3), 142-147.

Dong, G., Liu, F., & Wu, G. (2022). A Website's Network Attack Analysis and Security Countermeasures. *Procedia Computer Science*, *208*, 577-582.

Ahmad, R. (2021). A critical review of open source software development: freedom or benefit libertarian view versus corporate view. *IT Professional*, *23*(1), 16-26.

Aljebry, A. F., Alqahtani, Y. M., & Sulaiman, N. (2022). Analyzing Security Testing Tools for Web Applications. In *International Conference on Innovative Computing and Communications: Proceedings of ICICC 2021, Volume 1* (pp. 411-419). Springer Singapore.

Karayat, R., Jadhav, M., Kondaka, L. S., & Nambiar, A. (2022, March). Web application penetration testing & patch development using Kali Linux. In *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)* (Vol. 1, pp. 1392-1397). IEEE.

Wibowo, R. M., & Sulaksono, A. (2021). Web vulnerability through Cross Site Scripting (XSS) detection with OWASP

security shepherd. *Indonesian Journal of Information Systems*, *3*(2), 149-159.

Onyango, C., & Wang, X. (2024). Enhancing Web Application Security Through Penetration Testing.

Jarupunphol, P., Seatun, S., & Buathong, W. (2023). Measuring Vulnerability Assessment Tools' Performance on the University Web Application. *Pertanika Journal of Science & Technology*, *31*(6).

Riepponen, M. (2024). Selection of open-source web vulnerability scanner as testing tool in continuous software development.

Albahar, M., Alansari, D., & Jurcut, A. (2022). An empirical comparison of pen-testing tools for detecting web app vulnerabilities. *Electronics*, *11*(19), 2991.

Altulaihan, E. A., Alismail, A., & Frikha, M. (2023). A survey on web application penetration testing. *Electronics*, *12*(5), 1229.

Bouafia, R., Benbrahim, H., & Amine, A. (2023, October). Automatic Protection of Web Applications Against SQL Injections: An Approach Based On Acunetix, Burp Suite and SQLMAP. In *2023 9th International Conference on Optimization and Applications (ICOA)* (pp. 1-6). IEEE.

Singh, N., Meherhomji, V., & Chandavarkar, B. R. (2020, July). Automated versus manual approach of web application penetration testing. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp. 1-6). IEEE.

Rane, N., & Qureshi, A. (2024, April). Comparative Analysis of Automated Scanning and Manual Penetration Testing for Enhanced Cybersecurity. In *2024 12th International Symposium on Digital Forensics and Security (ISDFS)* (pp. 1-6). IEEE.