Journal of Information Systems Engineering and Management
2025, 10(10s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Assessing Security Risks in Information Systems through the Utilization of the Open Web Application Security Project (OWASP) Framework

Gusti Ayu Prathita Ananta[1], Benfano Soewito[2]
Computer Science Department, BINUS Graduate Program – Master of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia
gusti.ananta@binus.ac.id, bsoewito@binus.edu

ARTICLE INFO

Received: 15 Nov 2024

Revised: 26 Dec 2024

Accepted: 16 Jan 2025

ABSTRACT

The proliferation of digital technology has resulted in the growing utilization of applications to facilitate the execution of organizational business activities. This has a direct influence on the growing likelihood of cybercrime. To mitigate cybercrime, it is imperative to ascertain the present state of security of the IT infrastructure. Multiple paradigms exist for assessing the present security of IT infrastructure. Nevertheless, these frameworks still exhibit deficiencies, as none of them offer a comprehensive elucidation of the execution of security evaluation. Consequently, the author of this research paper presents a more comprehensive framework for evaluating the existing condition of IT infrastructure. The implementation of security evaluation consists of two distinct phases: penetration testing and risk measurement. The initial stage involves the utilization of the Web Security Testing Guide, a penetration testing framework developed by OWASP to enhance the security of web-based applications. The second phase employs the OWASP Risk Rating Methodology, a systematic approach for assessing the anticipated level of risk. The research yielded the discovery of 7 vulnerabilities, with 4 vulnerabilities classified as having a medium risk level and 3 vulnerabilities classified as having a low risk level. Security evaluation is conducted on vulnerabilities categorized as having a medium risk level due to their potential to cause losses and harm to IT infrastructure. The assessment is conducted by offering suggestions for enhancement. These recommendations are anticipated to serve as a guide for enhancing security on IT infrastructure in the future.

Keywords: IT Infrastructure Security; penetration testing; OWASP; Web Security Testing Guide, Risk Rating Methodology

INTRODUCTION

Companies employ a diverse range of software to execute their business activities. The proliferation of applications is paralleled by a corresponding rise in the potential for cybercrime[1]. Enhancing security in apps is of utmost importance [2]. It is important to understand the current status of security for infrastructure, apps, and other software. Therefore, it is imperative to assess the risk associated with the IT infrastructure utilized, including applications, internet networks, servers, and other related components. Implementing risk measurement poses inherent obstacles.

As in the research by Meland, Nesheim, Bernsmed, and Sindre [3] utilized the OWASP Risk Rating technique to determine the expected probability of threats. According to them, issues occur when doing an assessment on a fresh system that lacks a record of past events. The assessment is conducted by administering tests to determine the entities involved in the system (threat actors), favorable circumstances that contribute to the likelihood of attacks (opportunity), tools or resources required to execute attacks (means), and the drive behind carrying out attacks (motive). According to the findings of the security testing, it has been determined that the system is vulnerable to cyber attacks and there is a risk of losing valuable assets.

According to Sanjaya, Sasmita, and Arsa's research [4], penetration testing is the best way to secure the system. The Information System Security Assessment Framework (ISSAF) is the framework that is employed. SQL Injection and Cross Site Scripting (XSS) attacks were discovered in the results. Then, ISO 31000 is used for risk management. As a result, there are two high-level vulnerabilities and four medium-level vulnerabilities.

Kurniawan & Trianton [5] in their work suggested performing a security assessment as a means of safeguarding Android-based applications. The OWASP Mobile Top Ten 2016 framework is the one in use. Furthermore, the Common Vulnerability Scoring System (CVSS) 3.1 is also used for vulnerability evaluation.

The Information System Security Assessment Framework (ISSAF) and the OWASP Web Security Testing Guide are two frameworks that can be used to assess the security level of apps, according to the study mentioned above. Two stages make up the assessment of the IT infrastructure's security level in this study. Implementing penetration testing is the first step, and implementing risk measurement is the second.

Penetration testing, sometimes referred to as ethical hacking [6], is the initial stage in which the security of IT infrastructure is tested by taking advantage of the system and looking for weaknesses. The OWASP Web Security Testing Guide (WSTG) framework is used in penetration testing. One of the penetration testing frameworks created by the OWASP Foundation with an emphasis on web application security is called WSTG [7]. Penetration testing discoveries and vulnerabilities are compiled into a list.

The next stage involves the risk measurement. The OWASP Risk Rating Methodology is used to assess the risk associated with the list of vulnerabilities. The OWASP Risk Rating Methodology is utilized to assess the projected amount of risk by considering four elements [8]. These elements consist of threat agent factors and system vulnerability factors, which are used to assess the possibility of an event occurring. Additionally, technical and business aspects are considered to identify the potential impact of the event. Once the assessment is conducted using these parameters, the measurement process proceeds by mapping the risks against the assessment results in order to ascertain the level of risk. There are three distinct levels of risk, specifically high, medium, and low. The mapping findings are subsequently provided with recommendations and suggestions for further enhancements.

MATERIAL AND METHOD

Materials

In this study, the term "related research" pertains to previous research that addresses similar topics or utilizes similar research methodologies. Anak Agung Bagus Arya Wiradharma and Gusti Made Arya Sasmita conducted a security review on web-based applications. They performed penetration testing using the OWASP Testing Guide framework and risk assessment using the ISO 31000 framework [9]. For his investigation, the individual utilized tools suggested in the OWASP Testing Guide, and also incorporated tools from OSINT (Open Source Intelligence), such as Maltego. These tools are believed to enhance the outcomes of penetration testing. Penetration testing falls under the Information Gathering category, which encompasses 10 different types of tests. Out of a total of 10 tests, 6 tests detected vulnerabilities. Subsequently, a thorough risk assessment is conducted, revealing that there are 4 vulnerabilities with a moderate risk level and 2 vulnerabilities with a low risk level.

The following study was undertaken by Nuur Ezaini Akmar Ismail, Noraida Haji Ali, Masita Abdul Jalil, Farizah Yunus, and Ahmad Dahari Jarno [10]. The analysis indicated a surge in the adoption of cloud computing services amidst the COVID-19 outbreak. This rise undoubtedly leads to heightened security vulnerabilities. This promotes the evaluation of vulnerabilities and the testing of infiltration, also known as vulnerability assessment and penetration testing (VAPT). The research yields a novel paradigm that is both appropriate and feasible for addressing the situation at hand. The development of the new framework is a direct outcome of studying and gaining knowledge from established frameworks like OWASP and NIST. The new framework is specifically built for deployment in a cloud computing environment, encompassing three distinct services: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). The framework comprises five testing components: web-based applications, servers, mobile-based applications, APIs, and networks.

Additional relevant research has been conducted by Tika Astriani, Avon Budiyono, and Adityas Widjajarto [11]. According to his research, he asserted that the growth of virtual technology is presently on the rise, since it has the potential to enhance services in the commercial domain. Vulnerability level testing is conducted to safeguard virtual devices against cyber threats. The Docker container is utilized as a virtual device. The test was performed in accordance with the NIST 800-115 standard, utilizing the open-source tools OpenVAS and Docker Scan. The utilization of the OpenVAS tool reveals a total of 5 discoveries with a moderate risk level, 2 findings with a low risk level, and 24 findings categorized as information risk level. Using the Docker Scan tool, the results indicated a moderate risk level with 6 findings and a low risk level with 2 findings.

Method

The first step in providing security suggestions and assessing IT infrastructure security is determining the infrastructure's present security level. Penetration testing was done in this study to determine the present state of security on IT infrastructure [12]. As a result, penetration testing is done using the steps shown in Fig. 1.

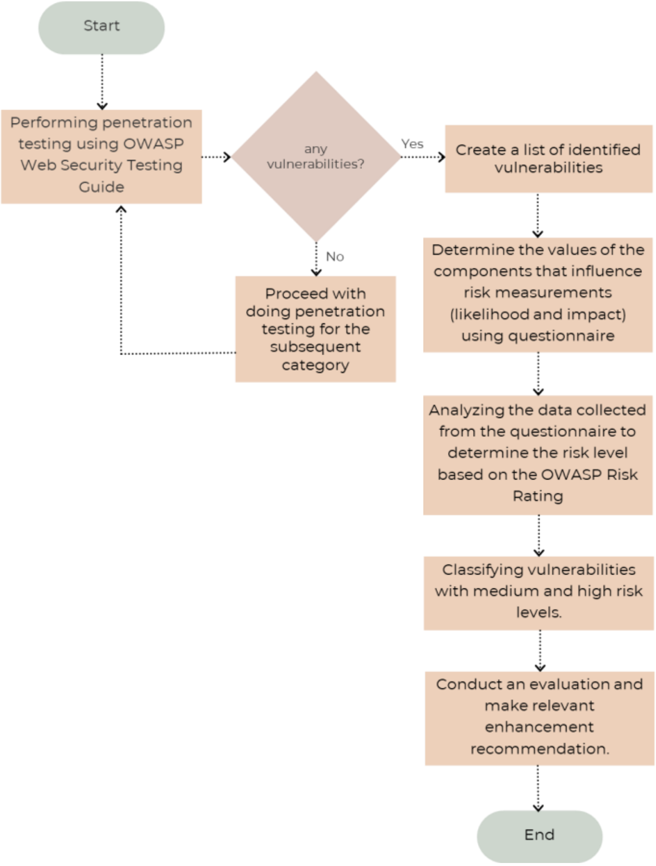

Start
Performing penetration testing using OWASP Web Security Testing Guide
any vulnerabilities?
Yes
No
Create a list of identified vulnerabilities
Proceed with doing penetration testing for the subsequent category
Determine the values of the components that influence risk measurements (likelihood and impact) using questionnaire
Analyzing the data collected from the questionnaire to determine the risk level based on the OWASP Risk Rating
Classifying vulnerabilities with medium and high risk levels.
Conduct an evaluation and make relevant enhancement recommendation.
End

Fig. 1 Risk Measurement Implementation Flowchart

The initial step is doing penetration testing using the OWASP Web Security Testing Guide (WSTG) as a basis. The document utilized is OWASP WSTG version 4.2. The initial section of the Web Application Security Testing Guide (WSTG) is titled "Information Gathering" and encompasses the range of testing tasks. Information Gathering is a testing category that involves collecting data and information about the system. The purpose is to gain a comprehensive understanding of the system's workflow and its primary components. Within the Information Gathering category, there exist a total of 7 action modules. Table I illustrates the approach of conducting penetration testing on two modules, while the remaining modules are executed using the identical process.

Table I

Execution Of Modules In Penetration Testing

ID	**Module**	**Objective**	**Tools**
WSTG-INFO-01	*Conduct Search Engine Discovery for Information Leakage*	Identify the specific types of confidential data that could potentially be discovered inside the application's architecture and setup.	Dork
WSTG-INFO-02	*Fingerprint Web Server*	Determine the specific identify of the web server.	ZAP

If no vulnerabilities are found in the findings of penetration testing, the testing can proceed to the next category. Nevertheless, in the event of a vulnerability, a comprehensive list of vulnerabilities is compiled.

The next stage is risk measurement. The risk of the vulnerability list is measured using the OWASP Risk Rating Methodology. The factors that influence risk measurement are the likelihood and impact. The risk value of each vulnerability is computed based on the likelihood and impact values listed. Risk measurement is determined using 1 as shown below:

Risk = likelihood * impact (1)

The evaluation of likelihood is influenced by two components, specifically the threat agent factor and the vulnerability factor. Meanwhile, the evaluation of the impact is determined by technical factor and business factor.

A questionnaire was completed to assess the values of the determining factors for likelihood and impact. The questionnaire was filled out by 4 respondents with qualifications having an educational background in the computer field and having knowledge related to system security. The respondents were assigned the task of answering the questions presented in the questionnaire. The questionnaire consists of multiple-choice questions, with each choice assigned a certain weight. By selecting from these answer possibilities, one can ascertain the weight assigned to each determining factor. Table 2 displays the questionnaire form.

Table II. Questionnaire Form

ID	**Vulnerability**	**Likelihood**			**Impact**		
		A1	***A2***	***Avg.***	***B1***	***B2***	***Avg.***
WSTG-INFO-01	1. ...... 2.	...	....		...	....	

Description:

- A1: Threat Agent as determining factor number 1 of Likelihood
- A2: Vulnerability as determining factor number 2 of Likelihood
- B1: Technical as determining factor number 1 of Impact
- B2: Technical as determining factor number 2 of Impact
- Avg: Average

Table II displays the wthe determining factors for the likelihood and impact of the vulnerabilities. The weight for each determining criteria is established in the column. Upon completion of the questionnaire, the data from the questionnaire was subsequently processed. Data processing involves determining the final value of likelihood and impact by calculating the average weight. The final value is then used to determine the level of the likelihood and impact. The determination of the level can be observed in Table III.

Table III. Range of Values Used to Determine the Levels of Likelihood and Impact

0 to < 3	Low
3 to < 6	Medium
6 to 9	High

Once the likelihood and impact levels have been determined, proceed to compute the risk level of each vulnerability using (1). The computation results are subsequently organized in the risk mapping matrix, which is displayed in Table IV.

Table IV. Risk Mapping Matrix

Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH

The mapping process yields vulnerabilities that have been identified based on their corresponding risk levels. Next, the categorization of vulnerabilities with moderate and significant risk is conducted. The classification results are subsequently compiled into a prioritized list of vulnerabilities, for which recommendations for improvement and suitable follow-up will be provided.

RESULT AND DISCUSSION

Penetration Testing

The initial stage of the deployment involved conducting penetration testing to evaluate the security of the existing IT infrastructure and quantify potential threats. Penetration testing falls under the Information Gathering area. Within the Information Gathering category, a total of 7 modules have been completed. Here are the outcomes of the penetration testing installation.

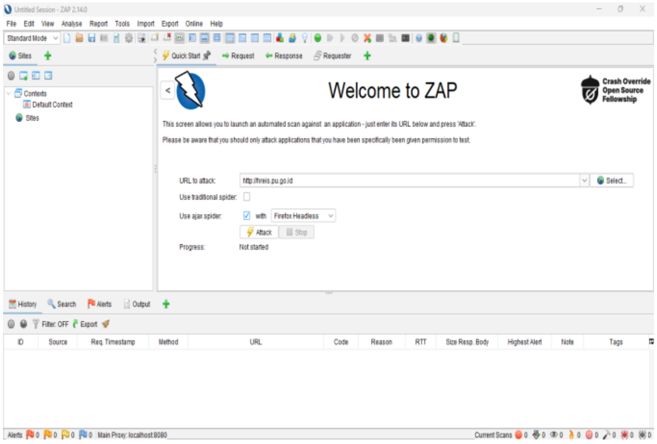
1) Conduct Search Engine Discovery for Information Leakage

During the initial testing phase, surveillance of a website was conducted by utilizing a search engine to identify any potential vulnerabilities related to data leaking. The employed tools include Google Dork and Google Hacking Database.

Testing using Google Dork is a technique of injecting keywords into a search engine by utilizing search operators. Site: and filetype: are the search operators that are employed, with site: being a search operator for locating a particular website URL. On the other hand, filetype: is a search operator that helps locate files on a website of a specific type. The Google Hacking Database is a combination of search operators produced by security researchers and kept in a publicly available database.

2) Fingerprint Web Server

The purpose of this second module is to do tests in order to determine the identification of the webserver, including its kind and version. The testing methodology employed involves banner grabbing, which involves issuing an HTTP request and analyzing the response header. Zed Attack Proxy (ZAP) was used. Fig. 2 shows the ZAP display.



Welcome to ZAP

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

URL to attack: http://hreis.pu.go.id

Use traditional spider: □

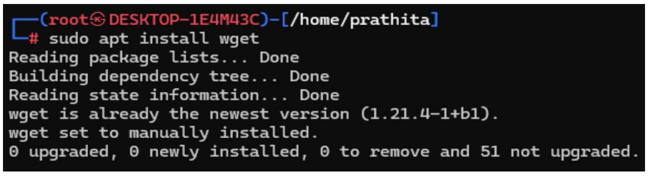
Use ajax spider: ☑ with Firefox Headless

Progress: Not started

Fig. 2 The ZAP Interface to Initiate an Attack

3) Review Web Server Metafiles for Information Leakage

For this third test, a thorough examination of the metafiles on the website was conducted. The review was conducted by locating three files, specifically sitemap.xml, security.txt, and human.txt, and executing them using the wget program. Wget program runs on Ubuntu. The sitemap.xml file is where the application developer includes information about the web page, such as videos, images, and other associated assets. Security.txt is a file that encompasses the security policy implemented on a website, as well as the contact information for the firm. And the file human.txt contains data about to the hierarchical arrangement and groups engaged in the process of creating a website. Fig. 3 shows the wget installation process.

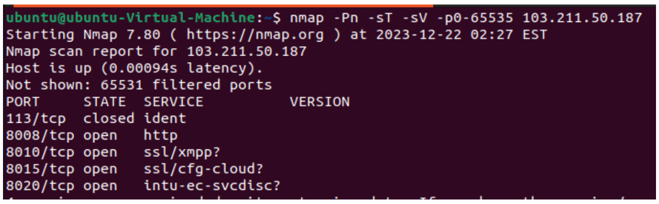


```
┌──(root㉿DESKTOP-1E4M43C)-[/home/prathita]
└─# sudo apt install wget
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
wget is already the newest version (1.21.4-1+b1).
wget set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 51 not upgraded.
```

Fig. 3 Installation of Wget

4) Enumerates Application on Web Server

The fourth test module focuses on precisely identifying running programs and files stored on the web server. Identification is achieved by the detection of suspicious or unfamiliar files, the discovery of unusual port types, and the retrieval of Domain Name Server (DNS) information. The employed tools include Nmap and Netcraft. Fig. 4 displays the results of the port scanning using Nmap.



```
ubuntu@ubuntu-Virtual-Machine:~$ nmap -Pn -sT -sV -p0-65535 103.211.50.187
Starting Nmap 7.80 ( https://nmap.org ) at 2023-12-22 02:27 EST
Nmap scan report for 103.211.50.187
Host is up (0.00094s latency).
Not shown: 65531 filtered ports
PORT      STATE  SERVICE           VERSION
113/tcp  closed ident
8008/tcp open   http
8010/tcp open   ssl/xmpp?
8015/tcp open   ssl/cfg-cloud?
8020/tcp open   intu-ec-svcdisc?
```

Fig. 4 Port Scanning results using Nmap

5) Review Webpage Content for Information Leakage

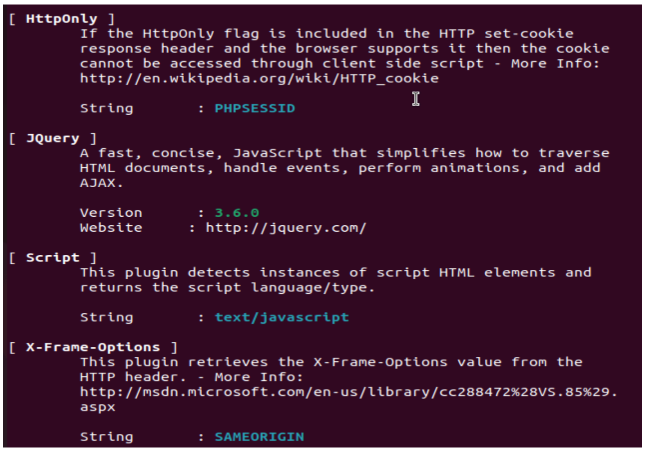
Testing is conducted in the fifth module to evaluate the content displayed on the web page. This test involves reviewing comments and metadata present in HTML source codes to identify any remarks or notes left by programmers that may reveal sensitive information such as id, username, password, and database details. Furthermore, the detection of JavaScript files (inside script tags or in .js files) is also performed.The used tools are the Developer Tools within a web browser.

6) Map Execution Path Through Application

The sixth test involves conducting a comprehensive analysis of the application's structure and components to gain a thorough understanding of its primary process. The utilized tools include the AJAX Spider within the ZAP vulnerability scanner application. Following the examination, other vulnerabilities were discovered.

7) Fingerprint Web Application Framework

The seventh testing module focuses on identifying the distinctive attributes of the framework employed by the web application. The distinctive attributes in question are associated with the technology employed in constructing the application. The utilized tools include WhatWeb and Wapplyzer. Fig. 5 shows one of the scanning results obtained using WhatWeb.



```
[ HttpOnly ]
        If the HttpOnly flag is included in the HTTP set-cookie
        response header and the browser supports it then the cookie
        cannot be accessed through client side script - More Info:
        http://en.wikipedia.org/wiki/HTTP_cookie

        String       : PHPSESSID

[ JQuery ]
        A fast, concise, JavaScript that simplifies how to traverse
        HTML documents, handle events, perform animations, and add
        AJAX.

        Version      : 3.6.0
        Website     : http://jquery.com/

[ Script ]
        This plugin detects instances of script HTML elements and
        returns the script language/type.

        String       : text/javascript

[ X-Frame-Options ]
        This plugin retrieves the X-Frame-Options value from the
        HTTP header. - More Info:
        http://msdn.microsoft.com/en-us/library/cc288472%28VS.85%29.
        aspx

        String       : SAMEORIGIN
```

Fig. 5 Scanning results using WhatWeb

A list of vulnerabilities was compiled based on the findings of the penetration testing. Table V displays a list of vulnerabilities.

Table V. List of Vulnerabilities

ID	Module	Tools	Result
WSTG-INFO-01	Conduct Search Engine Discovery Reconnaissance for Information Leakage	Google Dork, Google Hacking Database	There exist files and web pages that can be accessed by users who do not have the necessary permissions, even though these materials should only be accessible to people with the appropriate access rights.
WSTG-INFO-02	Fingerprint Web Server	ZAP	- Identified the web server's type and version information. - Discovered a cookie that contains session ID data.
WSTG-INFO-03	Review Webserver Metafile for Information Leakage	wget	Discovered data within the 'Meta' tags, including directory specifics for the primary image on the website's main page.

ID	Module	Tools	Result
WSTG-INFO-04	Enumerates Application on Webserver	Nmap, Netcraft	- Discovered 4 port services that are currently open - Discovered DNS data, including the IP address and domain name.
WSTG-INFO-05	Review Webpage Content for Information Leakage	Developer Tools	- Discovered the HTML type and version data - Discovered identification data, key and token data
WSTG-INFO-07	Map Execution Path Through Application	AJAX Spider	- The Content-Security-Policy header does not have any configurations. - The X-Frame-Option header does not have any configurations. - The URL rewriting includes session ID information
WSTG-INFO-08	Fingerprint Web Application Framework	WhatWeb, Wapplyzer	- Details regarding plugins, including programming language, operating system, web server, JavaScript library, and other relevant information - Details regarding alternative technologies employed for data visualization, including Tableau, CSS, and Google Maps

Risk Measurement

Following the completion of penetration testing, the findings are acquired in the format of a comprehensive inventory of vulnerabilities. Subsequently, an assessment of risk is conducted for each vulnerability, specifically by determining the likelihood and impact values. The likelihood and impact values were obtained by completing a questionnaire.

After collecting and processing the data from the questionnaire, the likelihood and impact values are obtained as shown in Table VI below. This value is the final value, where previously the average likelihood and impact values of 4 respondents were calculated. While the level is determined by referencing Table III.

Table VI. Final Result of Likelihood dan Impact

ID	Module	Likelihood		Impact	
		Score	Level	Score	Level
WSTG-INFO-01	Conduct Search Engine Discovery Reconnaissance for Information Leakage	5.0	Medium	2.66	Low
WSTG-INFO-02	Fingerprint Web Server	5.46	Medium	3.34	Medium
WSTG-INFO-03	Review Webserver Metafile for Information Leakage	4.92	Medium	2.91	Low
WSTG-INFO-04	Enumerates Application on Webserver	4.86	Medium	3.81	Medium

ID	Module	Likelihood		Impact	
		Score	Level	Score	Level
WSTG-INFO-05	Review Webpage Content for Information Leakage	4.88	Medium	3.22	Medium
WSTG-INFO-07	Map Execution Path Through Application	5.28	Medium	3.25	Medium
WSTG-INFO-08	Fingerprint Web Application Framework	5.25	Medium	2.56	Low

Once the likelihood and impact values have been acquired, the process of risk mapping is conducted by utilizing Table 4 to ascertain the risk level associated with each vulnerability. The outcome is a level of risk that is seen in Table VII.

Table VII. Risk Level Assigned to Each Vulnerability Following the Completion of Risk Mapping

ID	Module	Risk Level
WSTG-INFO-01	Conduct Search Engine Discovery Reconnaissance for Information Leakage	Low
WSTG-INFO-02	Fingerprint Web Server	Medium
WSTG-INFO-03	Review Webserver Metafile for Information Leakage	Low
WSTG-INFO-04	Enumerates Application on Webserver	Medium
WSTG-INFO-05	Review Webpage Content for Information Leakage	Medium
WSTG-INFO-07	Map Execution Path Through Application	Medium
WSTG-INFO-08	Fingerprint Web Application Framework	Low

Evaluation and Recommendation

In this research, the evaluation stage is carried out by providing recommendations for improvement and follow-up related to the findings of the test results. To provide recommendations for improvement and follow-up, a classification is carried out on modules with vulnerabilities that have risks at high and medium levels. This is because risks with high and medium levels have a greater potential for system damage which certainly causes losses to the company.

The Table VII displays 3 modules with vulnerabilities categorized as low risk, and 4 modules categorized as medium risk. Based on this, the implementation of evaluation and recommendations is carried out on modules with a medium risk level. The modules are Fingerprint Webserver, Enumerates Application on Webserver, Review Webpage Content for Information Leakage, and Map Execution Path Through Application. The evaluation and recommendation displayed in Table VIII.

Table VIII. Evaluation and Recommendation

ID	Module	Recommendation
WSTG-INFO-02	Fingerprint Web Server	Modifying the web server configuration to conceal web server details that are commonly exposed in the HTTP Header Ensure that the web server consistently update to the most recent version.
WSTG-INFO-04	Enumerates Application on Webserver	Disable open ports when they are not necessary
WSTG-INFO-05	Review Webpage Content for Information Leakage	Eliminate or erase intricate HTML data Concealing source code that contains confidential information using web browser extensions
WSTG-INFO-07	Map the Target Application and Understand the Principal Workflows	Configure Content-Security-Option and X-Frame-Option headers on all web pages. Storing session ID information in cookies Update the library to the latest version

Table VIII displays the results of the evaluation and provides recommendations for improvement and follow-up. The following is a detailed explanation of the implementation of recommendations related to how the steps and procedures for these recommendations will be implemented.

1) Fingerprint Web Server

This test module provides recommendations in the form of setting session cookies to hide cookie information that can be seen in the HTTP header response. This is done in several ways:

To overcome session cookies without the HttpOnly flag, it is necessary to add the HttpOnly flag to the HTTP header response [13]. To add the flag, the PHP script that contains the session cookie settings is given the value true in the HttpOnly flag parameter. This way, if someone tries to access the cookie, access will be denied and the cookie will return a document with an empty string.

In addition to adding the HttpOnly flag, it is also necessary to add a secure flag. The secure flag is added by setting the secure flag parameter to a true value. This is to prevent cookie access through requests without https.

The next recommendation is to regularly update the web server to ensure that it is always up to date. This is done by first backing up and updating the repository on Ubuntu by running the 'apt-get update' command. Then install the web server by running the command 'apt-get install'. This will update the web server to the latest version.

2) Enumerates Application on Webserver

This test module recommends closing open ports if they are not needed. To close a port, the first step is to find the open port. On Ubuntu, use the LISTEN command to get service information from the port. Once you have found the open port and service name, run the "systemctl stop (service name)" command to stop the service. Then, to make sure the service is no longer active, run the 'systemctl disable (service name)' command.

3) Map Execution Path Through Application

In this test module, recommendations are given to make settings by configuring the Content-Security-Option and X-Frame-Option headers to avoid clickjacking and injection vulnerabilities. The settings are made by setting the Content-Security-Option and X-Frame-Option headers to DENY mode to prevent the use of iframe tags [14]. The iframe tag is a web page element that allows web content to include (embed) photos, videos, and other files. This is a loophole for other websites to embed malicious content into the website.

CONCLUSION

Penetration testing is a suitable method for evaluating the existing level of security in an IT infrastructure. The reason for conducting penetration testing is to detect different types of vulnerabilities present in the IT infrastructure, which can potentially lead to financial losses for the firm. After analyzing the literature, it has been determined that the OWASP Web Security Testing Guide is the most appropriate penetration testing framework that fulfills the research requirements. By utilizing this methodology, one can identify vulnerabilities that inadvertently serve as entry points for attackers to carry out cybercriminal activities and assess the effectiveness of measures taken to address these security loopholes.

Aside from doing penetration testing, risk assessment is also performed. Risk assessment is conducted to ascertain the magnitude or level of risk associated with a vulnerability. The objective is to safeguard the company against financial losses, thereby enabling the identification of appropriate risk mitigation measures based on the risk level. The OWASP Risk Rating Methodology is the framework employed in this research. The risk levels are categorized into three distinct levels: high, medium, and low.

The findings of penetration testing consist of 7 test modules conducted in line with the OWASP Risk Rating and measured using the OWASP Risk Rating Methodology. Out of these modules, 4 have vulnerabilities at a moderate level, while 3 have vulnerabilities at a low level. This demonstrates that the security measures implemented on IT infrastructure are still inadequate and require enhancement.

Security enhancements are implemented by conducting evaluations and offering recommendations in the form of improvement proposals, followed by monitoring the effects of identified vulnerabilities. Assessment and suggestions are conducted on risks categorized as high and medium, as these discoveries possess the capacity to result in system harm and illegal data use. The assessments and recommendations have been implemented through the literature review approach, specifically by analyzing prior studies and employing recommendations from the WSTG, as well as expert judgment.

This research can be a guide for improving security in IT infrastructure, where the results of the study can be a reference for conducting IT infrastructure security evaluations in other applications. For future research can assess the effectiveness of the recommendations proposed in this study.

Acknowledgment

Gusti Ayu Prathita Ananta, a student in the Computer Science Department at BINUS Graduate Program – Master of Computer Science, Bina Nusantara University, wrote this journal article based on a research report titled "Assessing Security Risks In Information Systems Through The Utilization Of The Open Web Application Security Project (OWASP) Framework". Benfano Soewito, a lecturer and deputy dean of the Master of Computer Science program at Bina Nusantara University, is overseeing this effort. The funding for this project has been provided by Bina Nusantara University. The opinions expressed here are those of the authors and may not necessarily align with the perspectives of the funding agency.

REFERENCES

[1] A. Fadlil, I. Riadi, and M. A. Mu'Min, "Mitigation from SQL Injection Attacks on Web Server using Open Web Application Security Project Framework," *International Journal of Engineering*, vol. 37, no. 4, pp. 635–645, Apr. 2024, doi: 10.5829/ije.2024.37.04a.06.

[2] O. Ben Fredj, O. Cheikhrouhou, M. Krichen, H. Hamam, and A. Derhab, "An OWASP Top Ten Driven Survey on Web Application Protection Methods," in *Risks anf Security of Internet and Systems: 15th International Conference, CRiSIS 2020, Paris, Fance, November 4-6, 2020, Revised Selected Papers 15*, Springer Science and Business Media Deutschland GmbH, 2021, pp. 235–252. doi: 10.1007/978-3-030-68887-5_14.

[3] P. H. Meland, D. A. Nesheim, K. Bernsmed, and G. Sindre, "Assessing cyber threats for storyless systems," *Journal of Information Security and Applications*, vol. 103050, p. 64, Feb. 2022, doi: 10.1016/j.jisa.2021.103050.

[4] I. G. A. S. Sanjaya, G. M. A. Sasmita, and D. M. Sri Arsa, "Information technology risk management using ISO 31000 based on issaf framework penetration testing (Case study: Election commission of x city)," *International Journal of Computer Network and Information Security*, vol. 12, no. 4, pp. 30–40, Aug. 2020, doi: 10.5815/ijcnis.2020.04.03.

[5] C. Kurniawan and N. Trianto, "Security Assessment pada Aplikasi Mobile Android XYZ dengan Mengacu pada Kerentanan OWASP Mobile Top Ten 2016," *Info Kripto*, vol. 15, no. 1, pp. 11–18, 2021, doi: https://doi.org/10.56706/ik.v15i1.2.

[6] B. A. Chandrakant and J. P. Prakash, "VULNERABILITY ASSESSMENT AND PENETRATION TESTING AS CYBER DEFENCE," *International Journal of Engineering Applied Sciences and Technology*, vol. 4, pp. 72–76, 2019, [Online]. Available: http://www.ijeast.com

[7] E. Saad and R. Mitchell, *OWASP Web Security Testing Guide* , V4.2. OWASP Foundation , 2020. Accessed: May 27, 2024. [Online]. Available: https://owasp.org/www-project-web-security-testing-guide/

[8] J. Williams, "OWASP Risk Rating Methodology." Accessed: May 27, 2024. [Online]. Available: https://owasp.org/www-community/OWASP_Risk_Rating_Methodology

[9] A. A. B. A. Wiradarm and G. M. A. Sasmit, "IT Risk Management Based on ISO 31000 and OWASP Framework using OSINT at the Information Gathering Stage (Case Study: X Company)," *International Journal of Computer Network and Information Security*, vol. 11, no. 12, pp. 17–29, Dec. 2019, doi: 10.5815/ijcnis.2019.12.03.

[10] N. E. A. Ismail, N. H. Ali, M. A. Jalil, F. Yunus, and A. D. Jarno, "A Proposed Framework of Vulnerability Assessment and Penetration Testing (VAPT) in Cloud Computing Environments from Penetration Tester Perspective," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 39, no. 1, pp. 1–14, Sep. 2024, doi: 10.37934/araset.39.1.114.

[11] T. Astriani, "Analisa Kerentanan Pada Vulnerable Docker Menggunakan Scanner Openvas Dan Docker Scan Dengan Acuan Standar NIST 800-115," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 8, no. 4, pp. 2041–2051, 2021, [Online]. Available: http://jurnal.mdp.ac.id

[12] L. Erdődi, Å. Å. Sommervoll, and F. M. Zennaro, "Simulating SQL injection vulnerability exploitation using Q-learning reinforcement learning agents," *Journal of Information Security and Applications*, vol. 61, p. 102903, Sep. 2021, doi: 10.1016/j.jisa.2021.102903.

[13] S. Hidayatulloh and D. Saptadiaji, "Penetration Testing pada Website Universitas ARS Menggunakan Open Web Application Security Project (OWASP)," *Jurnal Algoritma*, vol. 18, no. 1, pp. 77–86, 2021, doi: https://doi.org/10.33364/algoritma/v.18-1.827.

[14] M. Dewi, A. Budiono, and U. Y. K. S. Hediyanto, "Vulnerability Assessment pada Website Rekruitasi Asisten (IRIS) Fakultas Rekayasa Industri menggunakan Nikto dan Nessus," *eProceedings of Engineering*, vol. 10, no. 2, p. 1632, 2023.