Web Server Security Analysis Against Cross Site Scripting (XSS) Attacks using Penetration Testing

Rohmatul Mungfaridah
Department of Information System
Universitas Ahmad Dahlan
Yogyakarta of Indonesia

Imam Riadi
Department of Information System
Universitas Ahmad Dahlan
Yogyakarta of Indonesia

ABSTRACT

A web application is a program that can be accessed online via an intranet or the internet. This web app is a digital donation service available on mobile and on the website. Web apps that have not undergone security testing are vulnerable to hacker attacks. Web application performance will decrease due to vulnerabilities caused by hackers. The problem with implementing web apps security is that they have never tested the security of web apps, have not implemented a good standard of security analysis, especially in terms of dealing with Cross Site Scripting (XSS) attacks, and indeed needs to be tested because to avoid the risks that will occur. Penetration testing is carried out to secure web apps which are used as recommendations for follow-up repair solutions in securing web apps. Penetration testing is a popular technique, by actively evaluating defenses and web servers through the preparation and execution of all feasible attacks to find and exploit existing vulnerabilities. In this study, security testing was carried out using penetration testing with the zap and acunetix tools . This penetration testing consists of seven stages, namely: pre-engagement, information gathering, threat modeling, vulnerability analysis, exploitation, post exploitation, and reporting . The test results with Acunetix found a medium level Cross site Scripting (XSS) vulnerability, while the ZAP tool testing that has been carried out has identified 11 vulnerabilities, 2 medium level vulnerabilities, 7 low level vulnerabilities, and 2 informational vulnerabilities. The results of the recommendations are in accordance with the results of the analysis, so web apps need to use input validation for acceptable input that is truly in accordance with the specifications.

Keywords

Web Apps Security, OWASP-ZAP, Acunetix, Penetration Testing, Cross-Site Scripting (XSS)

1. INTRODUCTION

The rapid development of information technology, especially in the midst of the Covid 19 pandemic, where almost all activities are carried out online, and indirectly support government programs, activities are carried out online, therefore a system security is needed, to test the security of the system. information from hacking or system service interruption[1] . Because there is a collection of important data in the form of company information that needs to be protected by taking a comprehensive and structured approach to the risks that the organization may face [2] . In addition, the risk of lack of security on the system has the potential for hackers to break into the system. This will affect the damage and switching features of the system made [3] .

Hackers with high skills can perform cross site scripting attacks [4] . The appearance of XSS attacks on the data certainly shows that XSS attacks occur and develop consistently. Therefore, it is important to formulate and develop more effective methods to deal with this type of attack [5] .This must be anticipated to avoid obstacles or attacks such as Cross Site Scripting (XSS) [6] . There are many ways to evaluate an organization's information security attitude, Penetration testing is the best technique to confirm that the system is secure, penetration testing in most cases can assess security analysis, the possibility of finding and exploiting new vulnerabilities [7] And In contrast to cybercriminals, organizations those tested have authorized penetration testers to perform this test [8] , therefore data and information security is an important aspect in maintaining website resilience [9] . Damage or data loss can threaten this application at any time due to increasing human resources [10] . Especially for developers, the lack of awareness and understanding of system security problems is always the case [11] . Security needs to be done on web apps to avoid various attacks and interference by hackers [12] . Utilizing software that is made explicitly to identify system vulnerabilities, such as the Acunetix Vulnerability Scanner, is one technique to assess the security of a website [13] This security can be achieved in several ways, one of which is by using the Open Web Application framework **.** Security Project Zed Attack Proxy (OWASP ZAP) Top 10 [14] .According to [15] in his research that the risk assessment methodology developed by OWASP is a direct method for calculating and assessing the hazards associated with the application. OWASP lists 10 hazards for online applications, including cross-site scripting, companies are encouraged to use caution in the use of known vulnerability components, and inadequate vulnerability recording and monitoring [16] .

The importance of this research is to conduct vulnerability testing to the security of web apps so that later they can secure user data on web apps from hacker attacks that can harm users in the event of hacking, which is likely to be misused by irresponsible parties. This test provides solutions and recommendations for vulnerabilities from the test results so that the web server is not easily attacked by hackers.

Based on the description above, to determine the security of the web server from Cross Site Scripting (XSS) attacks using OWASP-ZAP, then will conduct research on Web Server Security Analysis Against Cross Site Scripting (XSS) Attacks using Penetration Testing. under attack by hackers.

2. STUDY LITERATURE

2.1 Information System

Technically, a system is a collection of five related parts that work together to collect, process, store, and distribute data to support organizational control and decision making. Information systems can assist managers and employees with

decision making, coordination, and control, as well as in problem solving, visualizing difficult concepts, and developing new products [17] . This process involves analysis of weaknesses, vulnerabilities and technical weaknesses. A vulnerability is a weakness in the design, implementation, operation, or management of a system that can be exploited that compromises system security objectives [18] . Security testing is a method used to assess the security of a computer system or network by validating and verifying the effectiveness of application security controls. A computer network is a group of interconnected computers that can transfer resources within a system, communicate with each other, and access data [19] .

2.2 Cybercrime

Cybercrime is a type of crime that targets information technology [20] . There are several different characteristics of this form of attack, namely: in the event of a denial of service attack the possibility of not being able to open all online services, the correct website address being moved to the wrong website address, slow network system, storage in log files, error messages messages such as 500 errors, "internal server error," and "problem processing your request. [21] .Indications of web attacks Indications of web attacks make it easier to identify hidden vulnerabilities, and can make it easier to quickly exploit exploits and then fix them by developing patches [22] .

2.3 Penetration Testing

Penetration testing, commonly referred to as pentesting or PT, is a standard procedure for actively evaluating the defenses of computer networks and web servers by setting up and executing all potential attacks to identify and exploit known weaknesses [23] . Following are the penetration testing stages in Figure 1

Figure 1. Penetration Testing Stages

Proper penetration testing always ends with detailed suggestions for dealing with and resolving issues found during testing. This technique helps secure computers, networks, and web servers as a whole against threats, test results, and vulnerabilities found. this can then be reduced before hackers exploit the web server [24] .

2.4 Cross Site Scripting (XSS)

The cross-site scripting (XSS) vulnerability is one of the most common vulnerabilities on the Internet and is a direct response to the increased user interaction in today's web applications [25] . Cross-Site Scripting (XSS) is a code injection attack, this attack is a very serious attack on web applications. This attack occurs when hackers enter JavaScript code through client-side input. This attack aims to get cookies and tokens and even issue other attacks. Cross-Site Scripting (XSS) occurs due to security vulnerabilities in HTML, fash , JavaScript, AJAX, [22] .

2.5 OWASP ZAP

The Open Web Application Security Project (OWASP) is a group committed to developing companies that intend to: create, acquire, and follow Trusted software. OWASP visitors will find everything openly free. All programs, papers, forums and anyone can access the OWASP branch for free [26] . OWASP ZAP has test strings for various technologies which are useful for first identifying the technology the target is using, in order to optimize scanning and reduce the chances of being detected or causing service degradation [27] . Zed Attack Proxy (ZAP) is a simple integrated penetration testing tool used to detect web application vulnerabilities. Zed Attack Proxy (ZAP) is designed for developers and functional testers who are new to penetration testing. Zed Attack Proxy (ZAP) provides an automated scanner, and various tools that can be used to find vulnerabilities manually [28] .

2.6 Acunetix

Acunetix is an automated tool for assessing the security of web applications. It audits your web applications looking for exploitable vulnerabilities like SQL Injection, cross-site scripting and other flaws. Acunetix usually scans any website or online application that is accessible via a web browser and uses the HTTP/HTTPS protocol. Acunetix offers a unique and reliable solution for the analysis of pre-built and custom web applications, including those using JavaScript, AJAX, and Web 2.0 web applications. Acunetix has an advanced crawler that can find almost any file. This is very important because unchecked items can not be found [29] .

2.7 Web Application

A website is a group of pages in a website domain on the internet that are created with a specific purpose, are linked to each other, and can be visited from different locations using the website URL. The web server acts as a service provider for the browser, enabling it to display data or pages requested by their internet service customers. When processing web pages containing documents, videos, photos, or other types of files, the web server generally plays an important role in controlling and coordinating between the browser and the server [30] .A web application is a program that can be accessed online via an intranet or the internet. A web application is a piece of computer software that supports web-based programs written in languages such as CSS, HTML, Ruby, JavaScript, PHP, Python, and other programming languages. Slightly different are web applications. Web applications are dynamic, such as desktop software (Word, Photoshop, Skype)

3. METHODOLOGY

3.1 Research subject

The research subject that will be explained is "Analysis of Web Server Security Against Cross Site Scripting (XSS) Attacks using Penetration Testing. This study discusses the security of web apps using Penetration Testing which focuses on Cross Site Scripting (XSS) attacks. In this study, it is expected to find out the resilience of web apps in the event of

a Cross-Site-Scripting (XSS) attack to improve the security of web apps from Cross-Site-Scripting (XSS) attacks, so that they can implement good quality standardization of Penetration Testing security.

3.2 Research Flow

In this research, the research flow is carried out in several stages of web apps security research in collecting the data and materials needed, which are shown in Figure 2.

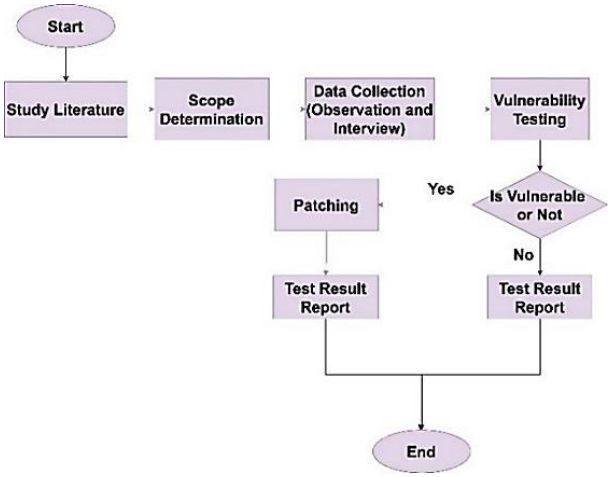


Figure 2. Research flow testing web apps

1. Study of literature
   The method of collecting data through research, reading, and collecting texts as references is known as literature study. Books, journals, and articles related to the topic in question are used as reference materials, regarding Web Server Security Analysis Against Cross Site Scripting (XSS) Attacks using Penetration Testing.

2. Determination of Scope
   In this determination of scope detecting vulnerability to attack web apps using OWASP-ZAP and acunetix . This research is limited to the detection of Cross Site Scripting (XSS) attacks in this study using the Open Web Application Security Project (OWASP) rules to detect Cross-Site Scripting (XSS) attacks. The tools used as research tools are ZAP and acunetix.

3. Observation Method
   The Observation Method is the initial stage that is carried out when conducting research, at this stage is conducting security observations of wab apps.

4. Interview Method
   The interview is a method carried out by asking questions to the Director of the company as the application manager, the results of an interview with one of the managers of web apps, it turns out that until now the company has not carried out penetration testing security tests and has not implemented good security analysis standards, especially in terms of handling Cross attacks. Site Scripting (XSS), and Testing should be done to reduce potential hazards. This danger stems from the loss of internal company data. Controlling the security of information systems is one method to prevent hackers from accessing company data because the risks that may arise must be managed to prevent damage.

5. Vulnerability testing
   Finding web apps vulnerabilities is the goal of vulnerability testing, this test uses the OWASP ZAP tool acunetix, using penetration testing.

6. Test result report
   At this stage, displaying the results of vulnerability testing presents any vulnerabilities that exist in web apps.

7. Patching (Patching)
   At this stage, if from the test results there are vulnerabilities, then patching the web apps, if no vulnerabilities are found in these web apps, the stages are only to display and present the test results.

3.3 Research methods

At this stage of research, conducting a simulation to implement OWASP in detecting cross site scripting (XSS) attacks using penetration testing, this research was conducted to determine the vulnerability of web apps. The stages carried out in this study are as shown in Figure 3.

Figure 3 . Web apps Security Testing Method

The stages of testing using penetration testing in this test are:

1. Pre-engagement
   At this stage is to interact with the client to conduct testing so that testing can run smoothly. Pre-engagement has a purpose, namely to present and explain the tools and techniques for which testing takes place and at this stage the test permit is the most important thing.

2. Information Gathering
   At this stage is collecting information, which is used to conduct testing.

3. Threat Modeling

At this stage, the threat modeling approach will occur for the proper implementation of penetration testing.

4. Vulnerability Analysis
   At this stage perform a vulnerability analysis that aims to find or identify vulnerabilities so that they can be exploited.

5. Exploitation
   At this stage run the exploitation of vulnerabilities that have been found. This stage is done by compiling a hacker attack scenario that hacks the vulnerabilities that have been found. According to the threat model, the attack scenario describes the attack strategy and the threat category based on the damage done. Gaining access to assets while causing the least disruption of service performance is the goal of the exploit stage.

6. Post Exploitation
   At this stage it collects information about the system being attacked, such as searching for important files, and so on.

7. Reporting
   This reporting stage is the last stage of testing the penetration test and writing the reports that have been obtained after doing the penetration test.

3.4 Assault Scenario

This scenario is made to explain how the stages of cross site scripting attacks. This hack is carried out, namely when the hacker accesses the web apps then the hacker creates a script and then sends it to the server, then when the user visits the web apps, the server sends the script, and the user receives a response from the command sent to the server, at this time the hacking of the web apps occurs. .

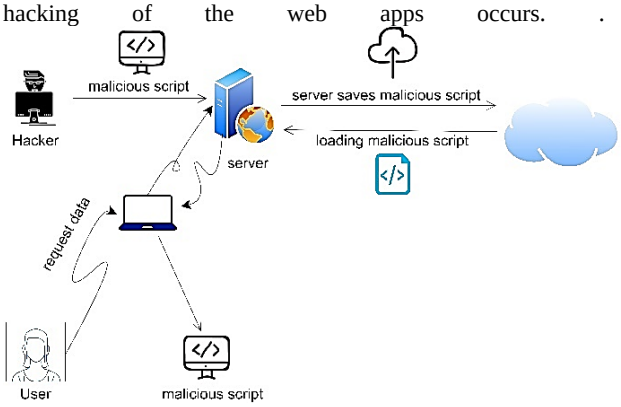


Figure 4. Web apps attack scenario

Figure 4 shows the stages of an XSS attack on Web Apps through one of the application vulnerabilities, hackers inject malicious scripts into the server to manage user sessions, steal data, and execute malicious code. The purpose of this research scenario is to facilitate the process of vulnerability analysis on web apps against cross site scripting attacks using ZAP and Acunetix tools .

4. RESULTS AND DISCUSSION

4.1 Vulnerability Testing

At the testing stage of web apps using penetration testing, there are several stages of testing, namely:

4.1.1 Pre-engagement

In the pre-engagement stage, there are several activities carried out based on the agreement between the pentester and the company (the owner of the application). The following are the activities carried out in the pre-engagement and the results can be seen in Table 1.

Table 1. Pre-engagement Results

Activity	Status	Results
Scope Identification	Worked	- Target Penetration Testing
Determining the Purpose of the Penetration Test	Worked	- Perform vulnerability testing on web apps . - Prepare recommendations and develop application patching results from testing resistance to Cross-Site-Scripting (XSS) attacks that can be used to improve the security of web apps .
Organizational Readiness Analysis	Worked	- Readiness for security on web apps in the company is still low so it is necessary to do security testing and analyze vulnerabilities.
Develop REO (Roles Of Engagement)	Worked	- Timeline - Test location - Test permit
Establishing Communication Lines	Worked	- Chat privately with one of the directors at the company.

4.1.2 Information Gathering

At the information gathering stage, there is some information obtained, namely from this stage, namely doing the information gathering using the OWASP ZAP tool, this information is collected by accessing the menu through a browser that has been installed on the OWASP ZAP proxy. This information is collected for each user data in the application and stores the results in OWASP ZAP, the results can be seen in Figure 5.

Figure 5. Results of ZAP Information Collection

4.1.3 Threat Modeling

At this stage of Cross Site Scripting Threat Modeling there is some information that is obtained, namely when a web app cannot filter the input sent by hackers, so that hackers can inject malicious code in the form of JavaScript, the purpose of this attack is to get cookies and session to gain access rights in the web apps. How Cross Site Scripting (XSS) attacks work can be seen in Figure 6.

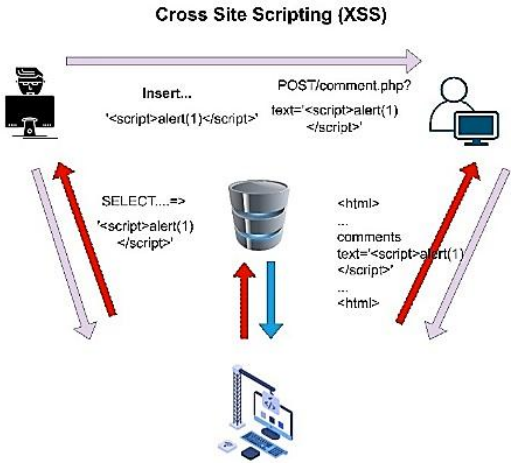


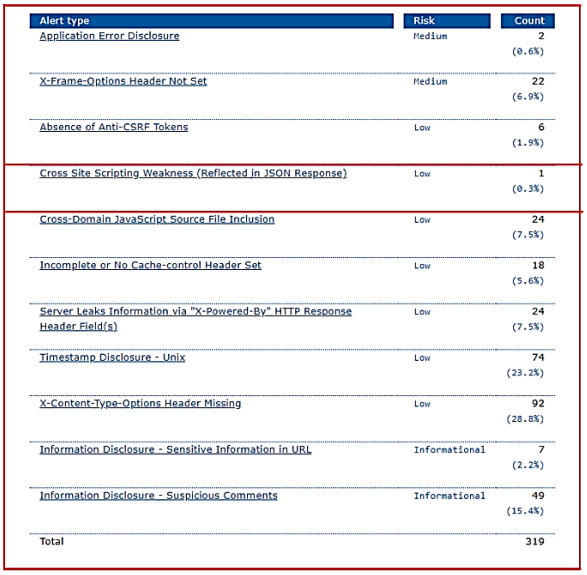
Figure 6. Cross Site Scripting (XSS) Attack Modeling

Figure 6 shows the threat modeling of cross site scripting attacks that might occur on web apps.

4.1.4 Vulnerability Analysis

4.1.4.1 Vulnerability Analysis using OWASP-ZAP

At the stage of vulnerability analysis carried out in this study using tools, namely the ZAP tool using Attack Mode, this attack mode is a new mode in scope that is scanned actively as soon as it is found. There are 2 methods or ways to use the ZAP tool , namely: manual-active and auto-activated. After the success of the automatic-active scanning of the web apps, several security vulnerabilities have been found in the web apps, with various levels such as low, high , medium and informational. This ongoing auto-activated vulnerability scanning process takes approximately 10 minutes, after which the vulnerability scan findings are obtained automatically. There are 11 vulnerabilities that have been obtained, namely 2 medium-level vulnerabilities, 7 low-level vulnerabilities, 2 cases of vulnerabilities are only informational. The number of warnings of each type of warning, along with the level of danger of that type. Each count is shown as a percentage of the total number of alerts contained in this report, rounded to one decimal point, in parentheses. For more details on vulnerabilities and the percentage of warnings on web apps, see Figure 7.



Alert type	Risk	Count
Application Error Disclosure	Medium	2 (0.6%)
X-Frame-Options Header Not Set	Medium	22 (6.9%)
Absence of Anti-CSRF Tokens	Low	6 (1.9%)
Cross Site Scripting Weakness (Reflected in JSON Response)	Low	1 (0.3%)
Cross-Domain JavaScript Source File Inclusion	Low	24 (7.5%)
Incomplete or No Cache-control Header Set	Low	18 (5.6%)
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Low	24 (7.5%)
Timestamp Disclosure - Unix	Low	74 (23.2%)
X-Content-Type-Options Header Missing	Low	92 (28.8%)
Information Disclosure - Sensitive Information in URL	Informational	7 (2.2%)
Information Disclosure - Suspicious Comments	Informational	49 (15.4%)
Total		319

Figure 7. Warnings by OWASP ZAP Vulnerability Type

4.1.4.2 Vulnerability Analysis using Acunetix

At the stage of vulnerability analysis on web apps, which was carried out in research using acunetix tools, this tool is a tool used for testing web apps security that can check for cross site scripting vulnerabilities, in this phase using acunetix crack version 12 and this application is website-based. In the scanning process on web apps, a yellow alert is obtained, indicating that the status is medium. When the alert is opened, it generates the information in Figure 8.

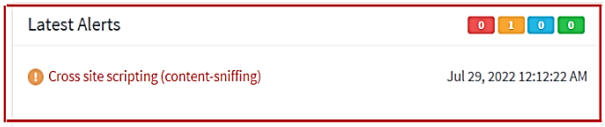


Figure 8. Scanning results with Acunetix 12

The results of scanning using acunetix version 12 have found a cross site scripting vulnerability whose status is medium.

4.1.5 Exploitation

At this exploitation stage, penetration testing will be carried out on web apps, the penetration testing carried out is cross site scripting testing, which takes advantage of security gaps by inputting javascript scripts in the form provided by web apps at each target URL on the web apps. At this stage of exploitation, the first step is to collect the target URL. After collecting the test, enter the cross site scripting (XSS) script on the existing input form. The scripts that are entered in the input form are the <script>alert(document.cookie)</script> script, and <imgsrc =x onError =alert(document.cookie)> .

4.1.6 Post Exploitation

In this post- exploitation stage, penetration testing will be

carried out on web apps . Penetration testing is carried out by testing cross site scripting , which takes advantage of security holes by inputting script javascript on form input provided by web apps after the URL collection stage and the input form then insert the script and see the response after the script is inserted, for details, see below:

1. Testing on the Register Page
   Testing xss on the register page, after inputting the xss script in the full name and password section when clicking the register button, the cross site scripting script was saved and became the full name and password of the user, Figure 9.


```
{
    "id": "05f0cdb0-7afa-448c-afe7-5426b5991a81",
    "full_name": "<script>alert(2)</script>",
    "email": "ujixsasasas@gmail.com",
    "phone": "09876543212",
    "username": "testxssasa",
    "role": "user",
    "contacts": [],
    "social_links": [],
    "updated_at": "2022-08-15T09:46:11.331Z",
    "created_at": "2022-08-15T09:46:11.331Z"
}
```


Figure 9. Response Result of Register Test

Figure 9 is the result of the command that has been carried out by the user so that the response from the command can be seen that the exploitation process of this register page was successful.

2. Testing on Login Page
   Testing cross site scripting (XSS) on the login page, after entering the xss script in the password section when clicking the login button, the cross site scripting (XSS) script was saved and became the password of the user, as shown in Figure 10


```
{
    "access_token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IndybDM2NWw2dn
c4NWwyIiwidXNlciI6eyJpZCI6IjQxZjEyMjE4LTVmMzItNDI5YS05ODA1LT
NhZGNmZmRjZTNlNSIsInJvbGUiOiJ1c2VyIn0sImlhdCI6MTY2MDYzNjcxOS
wiZXhwIjoxNjYwNzIzNDE5fQ.6D3ayCo2w6-
1ucfdTck9wzx6eKC8Aq9OntDBvMc-IqQ",
    "access_token_expires_at": "2022-08-17T08:03:39.682Z",
    "refresh_token": "eMJJ.C57-z3fwfc7wcP6Uubp-
uVD3afmgDN_zWTz",
    "refresh_token_expires_at": "2022-08-23T07:58:39.682Z",
    "scope": "default",
    "user": {
        "id": "41f12218-5f32-429a-9805-3adcffdce3e5",
        "email": "rida123@gmail.com",
        "username": "ridarelawan",
        "full_name": "ridarelawan",
        "phone": null,
        "profile_photo": null,
        "role": "user",
        "scope": "default"
    }
}
```


Figure 10. Response Results of Login Page Tests

Figure 10 is the result of the command that has been carried out by the user so that the response from the command can be seen that the exploitation process of this login page is successful.

3. Testing on Search Pages
   Testing cross site scripting (XSS) on the search page (search), after inputting the XSS script in the search form section when then clicking enter, the cross site scripting (XSS) script was saved and the page will display the results of the command from the user, as shown in the image 11.


```
{
    "pageProps": {
        "q":
"%3Cscript%3Ealert(document.cookie)%3C/script%3E"
    },
    "__N_SSP": true
}
```


Figure 11. Response Results of Testing Search Pages

Figure 11 is the result of the command that has been carried out by the user so that the response from the command can be seen that the exploitation process of this search page was successful.

4. Testing on Edit Profile Page
   Testing cross site scripting (XSS) on the edit profile page, after entering the XSS script in the name form section when then clicking enter, the cross site scripting (XSS) script was saved and the page will display the results of the command from the user, can be seen in the picture 12.


```
{
    "id": "8432c402-b34a-4789-abcc-cdfa3c9daaad",
    "full_name": "<script>alert(document.cookie)</script>",
    "email": "ghsghs@gmail.com",
    "username": "ridacantikbanget",
    "phone": "081234567890",
    "description": null,
    "role": "user",
    "scope": "default",
    "profile_photo": null,
    "verified_at": null,
    "email_verification_token": null,
    "email_verification_token_expired_at": null,
    "pixel_id": null,
    "ga_id": null,
    "created_at": "2022-06-22 13:18:35",
    "updated_at": "2022-08-15 21:22:01",
    "deleted_at": null,
    "contacts": [],
    "social_links": []
}
```


Figure 12. Test response results Edit Profile page

Figure 12 is the result of the command that has been carried out by the user so that the response from the command can be seen that in the exploitation process this profile edit was successful.

5. Testing on the Donation Page on the Operational Support input form
   Testing cross site scripting (XSS) on the Donation page on the Operational Support input form, after inputting the cross-site scripting (xss) script in the operational support form section when then clicking continue payment, then go directly to the next page. The possibility of inputting the script is successful but because it is not continued in the next step, namely the donation payment step, the response cannot be seen.

4.1.7 Reporting

At this reporting stage, from all the testing phases that have been carried out, it can conclude the test results using penetration testing on web apps, for details, see Table 2.

Table 2. Penetration Testing Results

No	Attack Type	Tools	Status
1	Cross Site Scripting (XSS)	OWASP-ZAP	Succeed
2	Cross Site Scripting (XSS)	Acunetix	Succeed

From the types of attacks that are known at the post exploitation stage, cross site scripting attacks have been successfully exploited. Based on the results of web apps security testing, recommendations for improvement from the findings of cross site scripting vulnerabilities in web apps can be seen in Table 3.

Table 3. Repair Solutions and Recommendations

Security gaps	Solution
Cross Site Scripting (XSS)	For any data to be output to another web page, especially data received from external input , use the appropriate encoding for all non-alphanumeric characters. See the XSS Prevention Cheat Sheet for more details on the types of ciphers and escaping required.For each generated web page, use and specify a character encoding such as ISO-8859-1 or UTF-8.set the session cookie to HttpOnly.this attribute can prevent a user's session cookie from being accessed by malicious client-side scripts that use document.cookie .'accept known good' input validation strategy , i.e., use a whitelist of acceptable inputs that strictly conform to the specification.

5. CONCLUSION

Based on the findings of this study, it is possible to draw the conclusion that the analysis of website application vulnerabilities using the penetration testing method is able to ascertain the degree of information system vulnerabilities with a specific risk of Cross Site Scripting attacks, which have the potential to result in significant data leaks. The penetration testing method is thought to be utilized as a standard for evaluating web-based application security on a website application, commencing from the pre-engagement stage through reporting, through the security testing phases that have been conducted. To detect various security vulnerabilities, it is advised to apply a variety of tools and techniques in future development. This study can serve as a guide for everyone testing an internet application's security, both

6. REFERENCES

[1] S. Utoro, BA Nugroho, M. Meinawati, and SR Widianto, “Analysis of E-Learning Website Security at SMKN 1 Cibatu Using the Penetration Testing Execution Standard Method,” Multinetics , vol. 6, no. 2, pp. 169–178, 2020, doi:10.32722/multinetics.v6i2.3432.

[2] I. Riadi, A. Yudhana, and Y. W, “Security Analysis of Open Journal System Website Using Vulnerability Assessment Method,” J. Teknol. inf. and Computer Science. , vol. 7, no. 4, p. 853, 2020, doi:10.25126/jtiik.2020701928.

[3] F. Fachri, A. Fadlil, and I. Riadi, "Analysis of Webserver Security using Penetration Test," J. Inform. , vol. 8, no. 2, pp. 183–190, 2021, doi: 10.31294/ji.v8i2.10854.

[4] Y. W, I. Riadi, and A. Yudhana, "Analysis of Vulnerability Detection in Web Server Open Journal System Using OWASP Scanner," Journal of Information Technology Engineering (JURTI) , vol. 2, no. 1. p. 1, 2018, doi:10.30872/jurti.v2i1.1319.

[5] B. Darmajaya, “Method for Detection and Mitigation Cross Site Scripting Attack on Multi-Websites,” pp. 26–32, 2021, [Online]. Available: http://www.victim.site/search.php?keyword=.

[6] R. Umar, I. Riadi, and GM Zamroni, “Mobile forensic tools evaluation for digital crime investigation,” Int. J. Adv. science. eng. inf. Technol. , vol. 8, no. 3, pp. 949–955, 2018, doi:10.18517/ijaseit.8.3.3591.

[7] SRM Zeebaree, K. Jacksi, and RR Zebari, “Impact analysis of SYN flood DDoS attack on HAProxy and NLB cluster-based web servers,” Indonesia. J. Electr. eng. Comput. science. , vol. 19, no. 1, pp. 505–512, 2020, doi:10.11591/ijeecs.v19.i1.pp505-512.

[8] AR Kelrey and A. Muzaki, “The Effect of Ethical Hacking on Corporate Data Security,” Cyber Secur. and Digit Forensics. , vol. 2, no. 2, pp. 77–81, 2019, doi:10.14421/csecurity.2019.2.2.1625.

[9] RR Prayogo, "Security analysis using bwapp web application against XSS (Cross Site Scripting) and SQL Injection attacks," Dr. Diss. Univ. muhammdiyahjember,2016, doi: 10.20710/dojo.11.4_383.

[10] AP Dewanto, "Penetration Testing on the uii.ac.id Domain Using OWASP 10," Https://Dspace.Uii.Ac.Id/ , 2018, [Online]. Available: https://dspace.uii.ac.id/bitstream/handle/123456789/11281/13523025-Adetya Putra D-laporan thesis.pdf?sequence=1&isAllowed=y.

[11] I. Syarifudin, Pentesting and Analysis of Early Childhood Education Web Security . 2018.

[12] I. Riadi, R. Umar, and T. Lestari, "Cross Site Scripting (XSS) Attack Vulnerability Analysis on Smart Payment Applications Using the OWASP Framework," JISKA (Jurnal Inform. Sunan Kalijaga) , vol. 5, no. 3, pp. 146–152, 2020, doi:10.14421/jiska.2020.53-02.

[13] A. Zirwan, “Website Security Testing and Analysis Using Acunetix Vulnerability Scanner,” J. Inf. and Technol. , vol. 4, no. 1, pp. 70–75, 2022, doi:10.37034/jidt.v4i1.190.

[14] Sunardi, I. Riadi, and PA Raharja, “Vulnerability analysis of E-voting application using open web application security project (OWASP) framework,” Int. J. Adv. Comput. science. app. vol. 10, no. 11, pp. 135–143,2019, doi:10.14569/IJACSA.2019.0101118.

[15] B. Ghozali, K. Kusrini, and S. Sudarmawan, “Detecting Website Application Security Vulnerabilities Using Owasp (Open Web Application Security Project) Method

for Risk Rating Assessment," Creat. inf. Technol. J. , vol. 4, no. 4, p. 264, 2019, doi:10.24076/citec.2017v4i4.119.

[16] M. Yunus, "Web-Based Application Vulnerability Analysis Using a Combination of Security Tools Project Based on Owasp Framework Version 4," J. Ilm. information. computer. , vol. 24, no. 1, pp. 37–48, 2019, doi:10.35760/ik.2019.v24i1.1988.

[17] NF Ningsih and I. Riadi, "Risk Assessment Analysis on Library Information System using OCTAVE Allegro Framework," Int. J. Comput. app. , vol. 183, no. 28, pp. 6–13, 2021, doi:10.5120/ijca2021921620.

[18] E. Saad and R. Mitchell, "web security testing guide version 4.2," vol. 4.2, 2020, pp. 1–465.

[19] HRS Nadenggan and I. Riadi, "Analysis of Local Area Network Performance using Quality of Service," Int. J. Comput. app. , vol. 183, no. 46, pp. 43–51, 2022, doi:10.5120/ijca2022921866.

[20] DA Mu'minin and I. Riadi, "Cybercrime Data Search Fraud Case on Mobile based MiChat Service," Int. J. Comput. app. , vol. 183, no. 47, pp. 43–49, 2022, doi:10.5120/ijca2022921880.

[21] E. Council, Computer Forensics: Investigating Network Intrusions and Cyber Crime . 2009.

[22] BB Gupta and P. Chaudhary, Cross-Site Scripting Attacks: Classification, Attack, and Countermeasures (Security, Privacy, and Trust in Mobile Communications) . 2020.

[23] MC Ghanem and TM Chen, "Reinforcement learning for efficient network penetration testing," Inf. , vol. 11, no. 1, pp. 1–23, 2020, doi:10.3390/info11010006.

[24] D. Kennedy, "The Basics of Hacking and Penetration Testing," Netw. Secur. , vol. 2011, no. 12, p. 4, 2011, doi:10.1016/s1353-4858(11)70127-1.

[25] A. Hoffman, Web Application Security Exploitation and Cointerm Measures for Modern Web Application . 2020.

[26] A. Aliefyan, "Penetration Testing To Find Out Web Application Security Vulnerabilities Using OWASP 10 Standard on Enterprise Web Domains" ResearchGate , no. July, 2020.

[27] G. Nájera-Gutiérrez, Kali Linux Web Penetratino Testing Cookbook . 2016.

[28] J. Curiel, "Creating OWASP ZAP Extensions and Add-ons," 2013.

[29] Acunetix, "v12 Product Manual," 2017, [Online]. Available: https://www.acunetix.com/resources/wvsmanual.pdf.