Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)

journal homepage: http://journal.lembagakita.org/index.php/jtik

Security Audit for Vulnerability Detection and Mitigation of UPT Integrated Laboratory (ILab) ITERA Website Based on OWASP Zed Attack Proxy (ZAP)

Ilham Firman Ashari [1*], Muhammad Affandi [2], Hendri Tri Putra [3], Muhammad Telaga Nur [4]

[1,2,3,4] *Informatics Engineering Study Program, Faculty of Production Technology and Industry, Institut Teknologi Sumatera.*

a r t i c l e i n f o

Article history:
Received 3 June 2022
Received in revised form
5 October 2022
Accepted 5 November 2022
Available *online* January 2023

DOI:
https://doi.org/10.35870/jtik.v7i1.657

Keywords:
Security; Audit; OWASP ZAP; Vulnerabilities.

Kata Kunci:
Keamanan; Pemeriksaan; OWASP ZAP; Kerentanan.

a b s t r a c t

Information technology now has many positive and negative effects on comfort. One of the negative effects of this technology is high level security attacks that can exploit various vulnerabilities and loopholes. Vulnerability testing (security audits) is therefore necessary to identify and overcome the vulnerabilities of the risks raised. The author is UPT Terpada Laboratory (Ilab) ITERA (https:http://ilab.itera.ac.id), a website maintained by his UPT Institute at the University of Technology Sumatra. This website contains all information about the labs of the University of Technology of Sumatra. Security audits are performed using the OWASP ZAP tool. A security check was performed on web ilab.itera.ac.id and the high priority alert results are: 1 vulnerability, medium priority warning: Three vulnerabilities, low priority warnings: Seven vulnerabilities and information: 3 vulnerabilities.

a b s t r a k

Teknologi informasi kini semakin memberikan banyak kemudahan dampak positif dan negatif di dalamnya. Dimana salah satu dampak negatif dari teknologi adalah tingginya tingkat serangan keamanan yang dapat menggunakan berbagai kerentanan dan celah. Sehingga diperlukan pengujian kerentanan (security audit) guna mengidentifikasi dan mengatasi resiko kerentanan yang dialami. Penulis melakukan security auditing pada Website UPT Terpada Laboratorium (Ilab) ITERA (https://ilab.itera.ac.id) yang merupakan website yang dikelola oleh UPT Laboratorium Institut Teknologi Sumatera. Website ini memuat segala informasi yang berkaitan dengan laboratorium di Institut Teknologi Sumatera. Audit Keamanan dilakukan dengan menggunakan alat OWASP ZAP. Security Auditing dilakukan pada web ilab.itera.ac.id, dan di hasilkan High Priority Alert: 1 vulnerability, Medium Priority Alert: 3 vulnerabilities, Low Priority Alert: 7 vulnerabilities, dan Information : 3 vulnerabilities.

*Corresponding author. Email: firman.ashari@if.itera.ac.id [1].

1. Introduction

Evaluation of the security of an information system at an agency in the education sector is very important because the threat of cyber attacks in 2021 according to the National Cyber and Password Agency for the Education sector is the biggest target compared to others [1]. Web hacking in agencies should be of special concern, because on the Education web there are important personal data [2]. Therefore, testing security vulnerabilities is an important thing, the rapid development of technology is also accompanied by high security attacks and loopholes.

Along with the development of information technology that continues to advance and develop, in harmony most institutions currently also support this development through activities and activities of the academic community based on information technology [3]. ITERA through UPT TIK (Integrated Service Unit of Information and Communication Technology) created a website to facilitate the implementation of practicum, namely ILab ITERA. ILab ITERA makes it easier for practitioners, practicum assistants, laboratory assistants, and lecturers to monitor the implementation of the practicum so that it is more optimal. The important assets on this website are data on the value of students undergoing practicum, laboratory data, asprak honorarium data. Therefore, the aspect of information security is something that needs to be considered.

Weak information asset security controls are a problem that must be prevented and addressed by responsible parties. Many security problems or disturbances are scattered on the internet, these disturbances can be in the form of malware attacks, exploits, database injection and so on [4]. The internet traffic control agency concluded that in 2016 approximately 90% of internet crimes were committed by attacking web applications with the most popular attack being database injection [5]. The total attacks against this database reached 47.06% of the total popular attacks. Security audits are often used to determine compliance with regulations, which determine how an organization should handle information [6].

Security audits, vulnerability assessments, and penetration testing are the three main types of security diagnostics. Each of the three takes a different approach and may be best suited for a particular purpose. Security audit measures the performance of information systems against a list of criteria, reputation damage, non-compliance, privacy violation, financial damage [7][8]. Vulnerability assessment, on the other hand, involves a comprehensive study of the entire information system, looking for potential security weaknesses. Penetration testing is a covert operation, in which a number of attacks are attempted to ascertain whether a system can withstand the same type of attack from malicious hackers (black hat or cracker) [9]. There are several factors that cause a lack of security on the website, including errors in writing program code and misconfiguration. This can be exploited by attackers, in this case the attacks that are often carried out are SQL Injection, Authentication and XSS. As released by webappsec.org shows that SQL Injection (47.06%) and XSS (35.33%) are the most frequent attacks [10][11].

The solution to monitor and improve web security from hacker intrusions or attacks can be done by means of a self-test, namely testing that is carried out on a web server legally with hacker-like activities. Self-test can be done with several penetration testing methods, one of which is the Open Web Application Security Project (OWASP) [12]. OWASP (Open Web Application Security Project) ZAP is an open source web application security scanner, used by those new to application security as well as the most active professional penetration testers and has been awarded the status of excellence [13]. Features in OWASP ZAP include Intercepting Proxy, Active and Passive Scanners, spider scan, report Generation, Brute Force (using OWASP dirbuster code), Fuzzing (using fuzzdb & OWASP JBrosfuzz), Extensibility, Auto tagging, Port scanner, Parameter analysis , Smart card support, Session comparison, invoke external apps, Api + headless mode, Dynamic SSL Certificates, Anti CSRF token handling [14]. With the many features contained in OWASP ZAP, it can provide convenience in scanning a web. Using OWASP Zap helps speed up the penetration testing process of the UPT Lab Lab and can sort out attacks on the web. In addition, OWASP ZAP is very easy to use, making it easier for beginners to scan the web [15].

2. Method

In the experiment, the author uses the OWASP ZAP tools. The steps involved in conducting Penetration Testing. The stages of testing the Penetration Testing method are presented in the following figure [16].

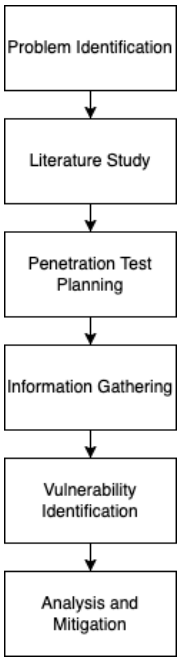

Figure 2. Penetration Testing Stages

The description of the steps from Figure 2 is as follows:

1) Problem Identification
   At this stage the scope of the problem is determined. Based on information from the laboratory staff at UPT Lab ITERA, the Ilab website was successfully penetrated by hackers. This indicates that there are still vulnerabilities on the Ilab Itera website.
2) Literature Study
   At this stage, further information is collected from the existing problems. Information is obtained from various written sources by studying the data.
3) Penetration Test Planning
   At this stage, preparations are made before conducting a penetration test on Ilab ITERA. The preparations made are as follows:
   a. Request permission from the ITERA Integrated Lab UPT to perform a penetration test on the duplicated ITERA Ilab for penetration testing purposes.
   b. Installing OWASP Zap.
4) Collecting Application Information
   At this stage, all Ilab ITERA information is collected through the ZAP proxy server. The information obtained in the form of URLs, class names, function names, POST or GET parameters, http responses and others will be stored in the OWASP ZAP session.
5) Identifying Application Vulnerabilities
   At this stage, a vulnerability scan for the SIPADUKO application is carried out with the help of OWASP ZAP active scan. All information stored in the OWASP ZAP session will be scanned for vulnerabilities. The results of this scan will also be stored in the OWASP ZAP session.

In the experiments carried out, first, using the existing web browser on the device, this time using mozilla firefox. The first thing to do is to generate a certificate from the OWASP to access this tool. After that, the certificates are added to the web browser by importing and then setting on the network, on the network using a more specific proxy, namely Manual proxy by using HTTP Proxy localhost and then accompanied by port 8080. After setting it correctly, testing is done by accessing the website will be observed [17].

Observation

From the experiments that have been carried out, observations are obtained as shown in Figure 3:

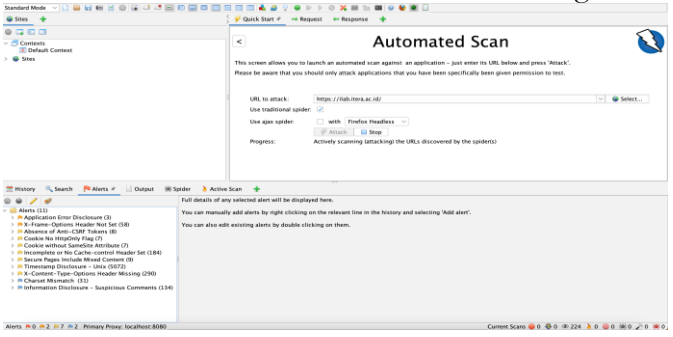

Figure 3. Fullscreen OWASP ZAP

From Figure 3 is an image of the display using OWASP tools, in the scanning process using a web browser, namely Mozilla Firefox by setting the settings on the Local Host with port 8080.

Figure 4. SQL Injection

From the results that have been obtained, there is a security risk for SQL injection with the result of risk: High, which is a vulnerability that is vulnerable and will be repaired immediately. Otherwise, it can cause considerable problems with a high degree of vulnerability.

Figure 5. SQL Injection Description

The description in Figure 5 above is the result of what was found by the SQL injection loophole. SQL injection is a technique that can attack the database layer; therefore, it is very dangerous for the Ilab ITERA web.

Figure 6. Medium Priority Alert 1

Figure 7. Application Error Disclosure

Figure 6 and Figure 7 show some lists of existing scanning results using OWASP which are Medium, the medium level of vulnerability is not more dangerous than the High level but at this level at any time it can cause problems on this website, and it is recommended to be repaired immediately. From the results of scanning at the medium level, there is an error in the css section.

Figure 8. Application Error Disclosure Description

From the results obtained in Figure 8, it is found that it is not too problematic because it only contains files that are placed outside, from the existing root folder, causing OWASP to consider this a medium level or gap.

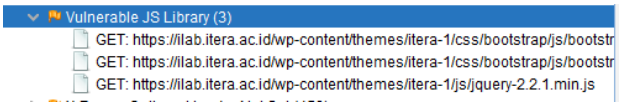

Figure 9. Medium Priority Alert 2

Figure 10. Vulnerable JS Library

Figure 9 and Figure 10 show the scanning results from using OWASP Medium. The vulnerability level of Medium is no more dangerous than the High level, but this level can at any time cause problems on the website and is recommended to be fixed immediately. The results of scanning at the medium level found an error in the javascript section.

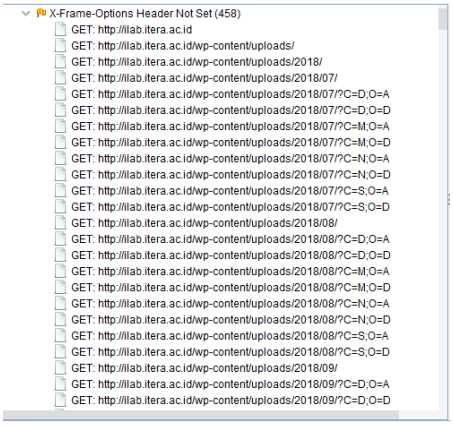

Figure 11. Medium Priority Alert 3

Figure 11 shows a list of scan results using Medium OWASP in the X-Frame option Header section.

Figure 12. X-Frame option Header

Figure 12 is an X-Frame options that are not contained in the HTTP response to prevent/counter Clickjacking which is an action by an attacker by tricking the user into clicking a button/link, so data theft can be done when clicking the link will control

the user's computer. It is old even though it is still medium.

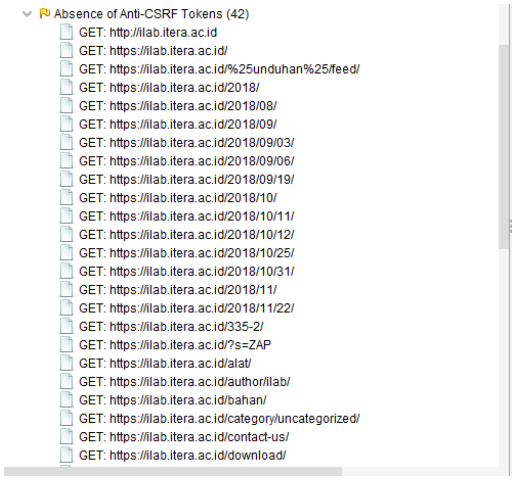

Figure 13. Low Priority Alert 1

Figure 14. Absence of Anti-CSRF Tokens

Figure 13 and Figure 14 show a list of low OWASP scanning results. This low level is not too dangerous for the website, it doesn't mean it's not important not to fix it. Based on the description of the results above, the website can be affected by Brute Force Attacks because there is no Anti_CSRF that has been found in the submission from HTML.

Figure 15. Low Priority Alert 2

Figure 16. Cookie no HttpOnly Flag

In Figure 16, a cookie has been set using the HttpOnly Flag, which is a cookie that can be accessed by JavaScript. If a malicious script is run on this page, then the cookie will be accessible and transmitted to other parts.

Figure 17. Low Priority Alert ke 3

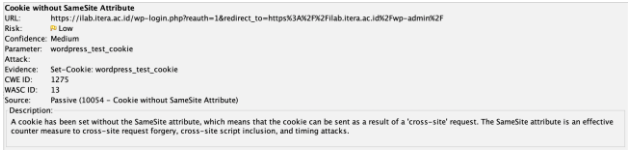

Figure 18. Cookie Without Same Site Attribut

Figure 17 and Figure 18 show a list of scan results using OWASP which is low in the Cookie section. Cookies are set without using the SameSite attribute, which means that cookies can be sent via cross-site requests, Samesite is very effective at preventing cross-site occurrences.

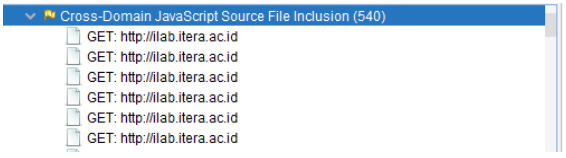

Figure 19. Low Priority Alert ke 4

Figure 20. Cross Domain Javascript

Figure 19 and Figure 20 show the results of scanning using OWASP which are low in cross-domain javascript security gaps. This page contains one or more files from third parties. Too many will cause vulnerabilities on this website.

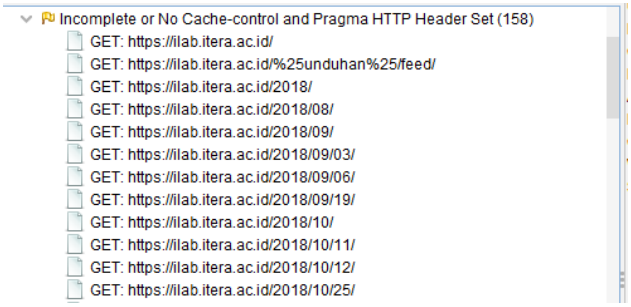

Figure 21. Low Priority Alert ke 5

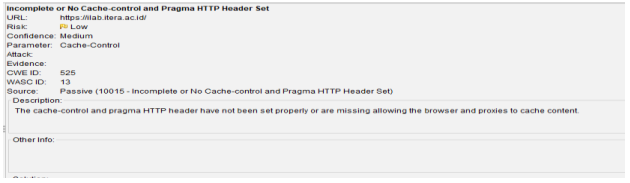

Figure 22. Incomplete or No Cache-control and Pragma HTTP Header Set

Figure 21 and Figure 22 show the results of scanning using OWASP which are low on the Incomplate or No Cache-control and Pragma HTTP Header Set security vulnerabilities. The description above explains that the unwritten pragma will cause the attacker to take cookies from the website.

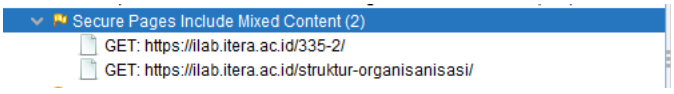

Figure 23. Low Priority Alert ke 6

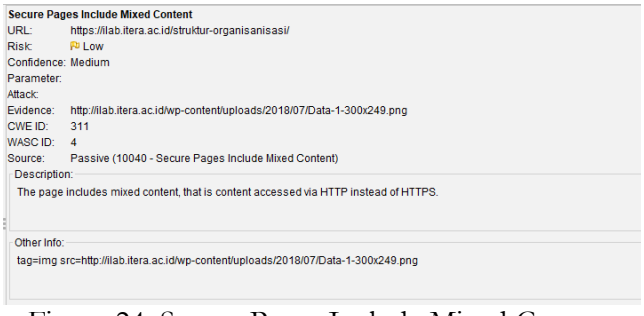

Figure 24. Secure Pages Include Mixed Content

Figure 23 dan Figure 24 menunjukkan hasil scanning menggunakan OWASP bersifat low pada Secure Pages Include Mixed Content. Deskripsi menjelaskan terdapat miss konten via HTTP pada HTTPS.

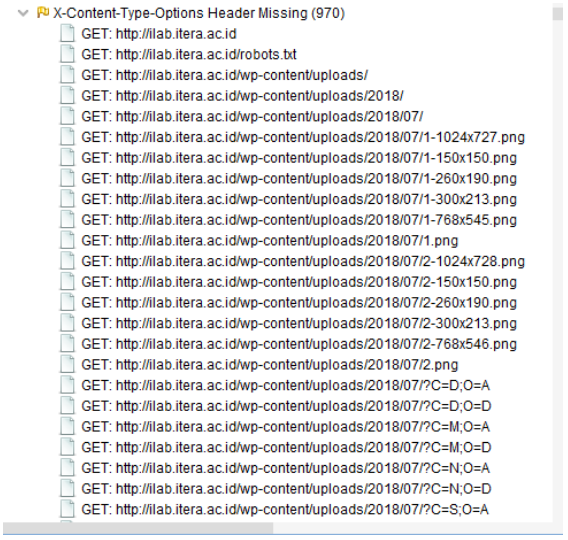

Figure 25. Low Priority Alert 7

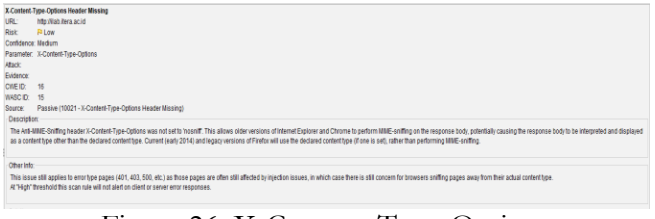

Figure 26. X-Content-Type-Options

Figure 25 and Figure 26 show the results of scanning using OWASP are low in the X-Content-Type-Options section. The description explains that the browser cannot be played by some robots, for example the Google robot to play data from this website. So, the picture above explains that the website must be set to no sniff so that there is no data game in it.

Figure 27. Informational 1

Figure 28. Informational Description 1

Figure 27 and Figure 28 show the results of scanning using OWASP which are Informational. The scan results that have been obtained are risk: Informational, which is not harmful to the website. The description above found a mismatch or mismatch between the HTTP Header and the Charset that was defined by the HTML or XML.

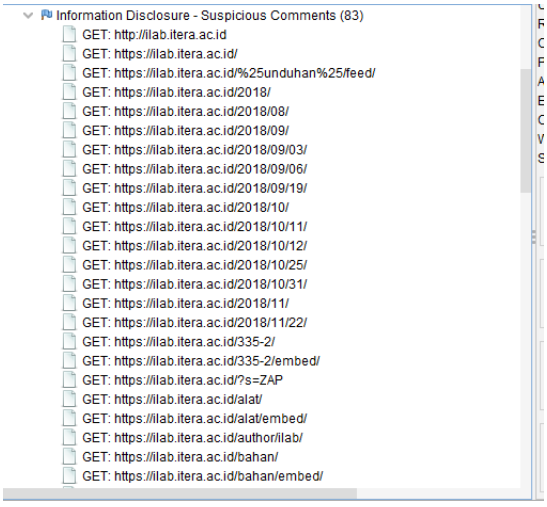

Figure 29. Informational 2

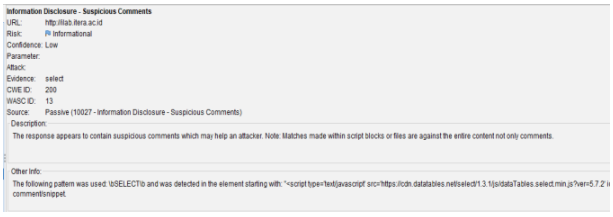

Figure 30. Informational 2 Description

Figure 29 and Figure 30 show the results of scanning using OWASP that are Informational in the Information Disclosure section which has sensitive information on suspicious comments. The description of the image above contains several comments or hints that make it easier for attackers or attackers to access the web.

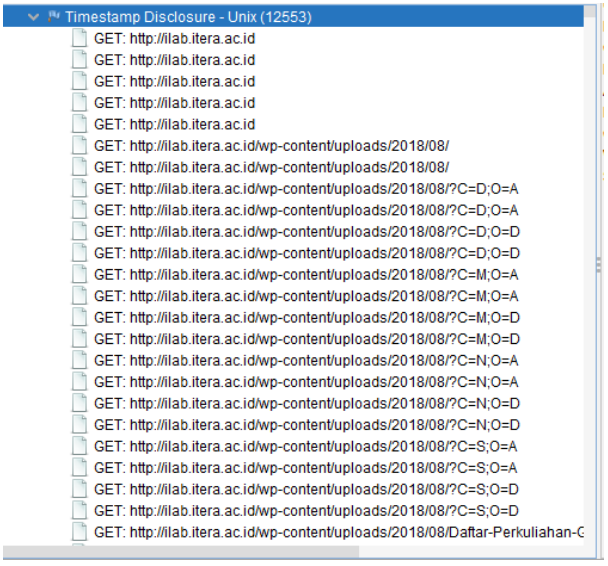

Figure 31. Informational 3

Figure 32. Informational 3 Description

Figure 31 and Figure 32 show a list of scanning results using OWASP which is Informational in the Timestamp Disclosure-Unix section. The description above explains the differences/errors in time between the web browser and the web server.

Analysis and Mitigation

At this stage, an analysis of the factors that cause security gaps is carried out and provides recommendations for mitigation steps from attacks that may be carried out.

3. Result and Discussion

Analysis of the Causes of Security Vulnerabilities in Websites

From the experiments and tests conducted on the Ilab ITERA website to find website security vulnerabilities, the causes of website security vulnerabilities can be analyzed and found there are 14 security vulnerabilities, namely 1 High Priority Alert, 3 Medium Priority Alert, 7 Low Priority Alert and 3 Information Alert.

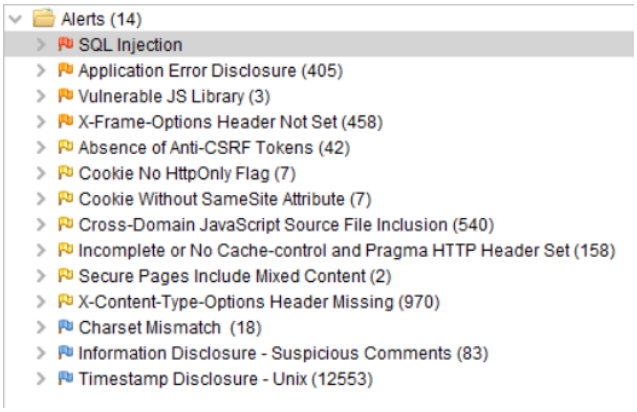

Figure 33. Information on Security Vulnerabilities on the Ilab ITERA Website

High Priority Alert

High Priority Alert is a very fatal level of vulnerability to external attacks, through experiments and analysis on the Ilab ITERA website the High Priority Alert security vulnerability was found in the SQL Injection section. SQL injection is a practice of using SQL by entering untrusted data into the interpreter as a command or query for the purpose of gaining access to the database without having to have a username and password.

Medium Priority Alert

There are 3 points of security holes in the medium priority alert as follows:

1) Application Error Disclosure
   Found a security vulnerability in the medium category with the type of Application Error Disclosure as many as 405 loopholes.
2) Vulnerable JS Library
   The vulnerability in Vulnerable JS Library Information is a type of website that uses a script with the aim of attacking the database without permission.
3) X – Frame Option Header Not Set
   The Frame Option Not Set loophole is a type of attack on a website that displays a button option. But the button is not found in the website settings. so that when the button is clicked by the user it can perform other functions created by the attacker. There are 458 X-Frame Option No Set loopholes on the test website.

Low Alert Priority

In Low Alert Priority there are 7 points of security holes, as follows:

1) Absence of Anti CSRF Tokens
   Absence of Anti-CSRF tokens. This token serves as a link between the browser and the server. There are 42 gaps by this type of attack
2) Cookies No HttpOnly Flag
   This website uses cookies to store sensitive information, but cookies are not marked with the HttpOnly flag. There are 7 loopholes that can be attacked.
3) Secure Pages Include Mixed Content
   In the Ilab ITERA crack detection experiment, there were 2 gaps through secure pages including mixed content.

Information Alert

1) Charset Mismatch
   This check identifies responses in which the HTTP Content Type header declares a character set that differs from the character set specified by the HTML or XML body. When there is a charset mismatch between the HTTP headers and the content the web browser can be forced into unwanted content-sniffing mode to determine the correct character set of content.
2) Information Disclosure – Suspicious Comment
   This page contains one or more comments that could reveal sensitive information to an attacker. This vulnerability allows attackers to view information that should not be accessible or data that is not theirs. There are many ways to find this gap. And there are also many types of data that may be exposed illegally.
3) Timestamp Disclosure – UNIX
   A timestamp is a sequence of distinct characters or information that has been encoded to aid in the identification of when an event will occur. Usually centered around using the date and time of day. On a computer, timestamps indicate the time the event was first recorded by the computer. It doesn't always focus on the time of the event. The Unix timestamp refers to the number of seconds that have elapsed since January 1, 1970, without the inclusion of a leap second. This will convert the timestamp using milliseconds and microseconds.

Suggestions for Overcoming Security Vulnerabilities

There are ways to overcome the High Priority Alert, Medium Priority Alert, Low Priority Alert, and

Information Alert security holes on the Ilab ITERA website:

1) High Priority Alert
   The solution to this vulnerability is to use values that are sent via URLs or fields in the form. Do not use parameters from URLs or forms without doing validation.
2) Medium Priority Alert
   a. Application Error Disclosure
      The risks associated with FPD can yield a variety of outcomes. For example, if a webroot is leaked, an attacker could misuse the knowledge and use it in combination with a file inclusion vulnerability (see PHP File Inclusion) to steal configuration files regarding web applications or other operating systems.
   b. Vulnerability JS Library
      Vulnerable JS Library is vulnerable to attack through JS libraries, the occurrence of security vulnerabilities is due to using libraries that have been inserted scripts that can enter the database of a website, so the solution to prevent this attack is to accurately identify which library vulnerabilities are valid and safe. for the website that we use, the second way is to do regular security updates and finally upgrade bootstrap to the latest version.
   c. X – Frame Option Header Not Set
      The X-Frame-Options HTTP response header can be used to indicate whether the browser should be allowed to render pages in <frame>, <iframe>, <embed> or <object>. Sites can use this to avoid clickjacking attacks, by ensuring that their content is not embedded into other sites
3) Low Alert Priority
   a. Absence of Anti CSRF Tokens
      The suggestion for this loophole is to Verify the referral header as the only protection. This can be easily manipulated. Submission of two cookies when the cookie used is a session cookie. This exposes the session cookie to JavaScript. Always mark session cookies "HTTP Only" so they cannot be accessed with JavaScript.
   b. Cookie No HttpOnly Flag
      The suggestion for this loophole is to make sure that the HttpOnly flag is installed in settings and views.
   c. Secure Pages Include Mixed Content
      Mixed Content is a secure web page in which scripts, styles, images, and other content are linked. Mixed Content is created through the HTTP protocol and can pose unexpected security risks for website visitors and the website itself. In the Ilab ITERA crack detection experiment, there were 2 gaps through secure pages including mixed content. to be able to prevent attacks through this loophole is by way of, pages available via SSL/TLS must consist of full content sent via SSL/TLS, this page must not contain any content sent via unencrypted http, this includes content from sites third party [18].
4) Information Alert
   a. Charset Mismatch
      Change the text encoding to UTF-8 for all text content in HTTP headers and meta tags in HTML or encoding in XML.
   b. Information Disclosure – Suspicious Comment
      These comments should be investigated and, if necessary, removed from the page
   c. Timestamp Disclosure – UNIX
      The Unix Timestamp converter comes in different formats and has been built into many software and websites that allow you to easily change the format from Unix to date and back.

4. Conclusion and Recommendation

From the experiments that have been carried out, it can be concluded that the Security Auditing conducted on the ilab.itera.ac.id web resulted in the results of High Priority Alert: 1, Medium Priority Alert: 3, Low Priority Alert: 7, and Information: 3. high priority alert with sql injection security vulnerability. Medium alert with application error disclosure security vulnerability, vulnerable js library, x-frame-options header not set and low priority low priority alert with security vulnerability absence of anti CSRF tokens, cookie no HttpOnly Flag, cookie without same attribute, cross domain javascript

source file inclusion, incomplete or no cache-control and pragma HTTP header set, secure pages include mixed content, x-content -type-options header mixing. From these results it can be concluded that the ilab.itera.ac.id website has an insecure vulnerability, so it is necessary to repair the found loopholes and further strengthen the protection so that it remains protected from people who make harm to the website. The suggestions from this Security Audit are as follows:

1) Routinely check and repair the web security system.
2) Upgrade the security system that has a high risk of being hacked so that the website is not easily attacked by intruders (hackers).
3) The gap is quite safe, protected even more strongly.

5. References

[1] Krisdiyawan, R.D. and Kuswantoro, R.H., 2017. Audit keamanan sistem informasi pada rs mata dr. Yap yogyakarta menggunakan framework cobit 5. *Jurnal Ilmiah Manajemen Informasi dan Komunikasi*, *1*(1), pp.8-15.

[2] Aritonang, I.J., Udayanti, E.D. and Iksan, N., 2018. Audit Keamanan Sistem Informasi Menggunakan Framework Cobit 5 (APO13). *ITEJ (Information Technology Engineering Journals)*, *3*(2), pp.6-10. DOI: 10.24235/itej.v3i2.27.

[3] Ashari, I.F., Aryani, A.J. and Ardhi, A.M., 2022. Design and Build Inventory Management Information System Using The Scrum Method. *JSiI (Jurnal Sistem Informasi)*, *9*(1), pp.27-35. DOI: https://doi.org/10.30656/jsii.v9i1.4050.

[4] Ashari, I.F., 2021. The Evaluation of Image Messages in MP3 Audio Steganography Using Modified Low-Bit Encoding. *Evaluation*, *14*(2).

[5] Kusumoningtyas, A.A., 1997. Dilema Hak Perlindungan Data Pribadi Dan Pengawasan Siber: Tantangan Di Masa Depan. *Law Review*, *66*, pp.177-205. DOI: 10.54629/jli.v17i2.706.

[6] Purba, A.D., Purnawan, I.K.A. and Pratama, I.P.A.E., 2018. Audit Keamanan TI Menggunakan Standar ISO/IEC 27002 Dengan COBIT 5. *Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi)*, pp.148-158. DOI: 10.24843/jim.2018.v06.i03.p01.

[7] Wicaksono, B., 2020. *Pengujian Celah Keamanan Aplikasi Berbasis Web Menggunakan Teknik Penetration Testing Dan Dast (Dynamic Application Security Testing)* (Doctoral dissertation, Institut Sains dan Teknologi AKPRIND Yogyakarta).

[8] Ghozali, B., Kusrini, K. and Sudarmawan, S., 2019. Mendeteksi Kerentanan Keamanan Aplikasi Website Menggunakan Metode Owasp (Open Web Application Security Project) Untuk Penilaian Risk Rating. *Creative Information Technology Journal*, *4*(4), pp.264-275. DOI: 10.24076/citec.2017v4i4.119.

[9] Sunaringtyas, S.U. and Prayoga, D.S., 2021. Implementasi Penetration Testing Execution Standard Untuk Uji Penetrasi Pada Layanan Single Sign-On. *Edu Komputika Journal*, *8*(1), pp.48-56. DOI: https://doi.org/10.15294/edukomputika.v8i1.47179.

[10] Elanda, A. and Buana, R.L., 2020. Analisis Keamanan Sistem Informasi Berbasis Website Dengan Metode Open Web Application Security Project (OWASP) Versi 4: Systematic Review. *CESS (Journal of Computer Engineering, System and Science)*, *5*(2), pp.185-191. DOI: 10.24114/cess.v5i2.17149.

[11] Riadi, I., Umar, R. and Lestari, T., 2020. Analisis Kerentanan Serangan Cross Site Scripting (XSS) pada Aplikasi Smart Payment Menggunakan Framework OWASP. *JISKA (Jurnal Informatika Sunan Kalijaga)*, *5*(3), pp.146-152. DOI: 10.14421/jiska.2020.53-02.

[12] Ashari, I.F. and Alfarizi, M., 2022. Vulnerability Analysis And Proven On The neonime. co Website using OWASP Zap 4 and XSpear. *JTKSI (Jurnal Teknologi Komputer dan Sistem Informasi)*, *5*(2), pp.75-81. DOI: 10.56327/jtksi.v5i2.1130.

[13] Hariyadi, D. and Nastiti, F.E., 2021. Analisis Keamanan Sistem Informasi Menggunakan Sudomy dan OWASP ZAP di Universitas Duta Bangsa Surakarta. *Jurnal Komtika (Komputasi dan Informatika)*, *5*(1), pp.35-42. DOI: 10.31603/komtika.v5i1.5134.

[14] Yudiana, Y., Elanda, A. and Buana, R.L., Analisis Kualitas Keamanan Sistem Informasi E-Office Berbasis Website Pada STMIKRosma Dengan Menggunakan OWASP Top 10. *CESS (Journal of Computer Engineering, System and Science)*, *6*(2), pp.37-43. DOI: 10.24114/cess.v6i2.24777.

[15] Mburano, B. and Si, W., 2018, December. Evaluation of web vulnerability scanners based on owasp benchmark. In *2018 26th International Conference on Systems Engineering (ICSEng)* (pp. 1-6). IEEE. DOI: 10.1109/ICSENG.2018.8638176.

[16] Zen, B.P., Gultom, R.A. and Reksoprodjo, A.H., 2020. Analisis Security Assessment Menggunakan Metode Penetration Testing dalam Menjaga Kapabilitas Keamanan Teknologi Informasi Pertahanan Negara. *Teknologi Penginderaan*, *2*(1).

[17] Ashari, I.F., 2020. Implementation of cyber-physical-social system based on service oriented architecture in smart tourism. *Journal of Applied Informatics and Computing*, *4*(1), pp.66-73. DOI: https://doi.org/10.30871/jaic.v4i1.2077.

[18] Ashari, I.F. and Munir, R., 2018, September. Graph Steganography Based On Multimedia Cover To Improve Security and Capacity. In *2018 International Conference on Applied Information Technology and Innovation (ICAITI)* (pp. 194-201). IEEE. DOI: 10.1109/ICAITI.2018.8686741.