[1]Ellen F. Mangaoang

[2]Richard N. Monreal

Common Vulnerabilities and Exposures Assessment of Private Higher Educational Institutions Using Web Application Security

Abstract: **-** The advancements in technology has led to advancements in threats and vulnerabilities as well. Academic websites are not spared from this. This study aims to evaluate the common vulnerabilities and exposures of private higher educational institution's website using Open Web Application Security Project. Seven private higher education institutions were evaluated using Open Web Application Security Project Zed Attack Proxy and Open Web Application Security Project top 10. The top vulnerabilities the higher educational institutions were exposed to were broken access control, insecure design and software and data integrity failures. There were two higher educational institutions with high level of risks. Also, a total of thirty-two risks or vulnerabilities were identified in the study. The identification of vulnerabilities or exposures in websites will help the private schools to become aware of possible cyberattacks to their institution's website. This will help the institution to identify and prevent further loss or damage.

Keywords: Cybersecurity, HEI, OWASP ZAP, OWASP Top 10

I. INTRODUCTION

Accessibility from a range of devices with installed web browsers, such as desktops, smartphones, or tablets, is one of the key benefits of web applications. This enables users to access the application whenever and from anyplace they have an internet connection [1].

However, the increased usage of electronic devices has led to the introduction of increasingly sophisticated and cutting-edge Windows, Web, and mobile applications. As a result, it is critical to understand security strategies for protecting our website from hackers and exploits [2].

Even in the process of developing and coding technology, mistakes occur. The remnant of these errors is often referred to as a bug. Although flaws don't necessarily cause harm, many of them can be exploited by malicious actors; and these are referred to as vulnerabilities. Vulnerabilities can be used to make software behave in ways that are not intended, such as gathering knowledge about the security measures that are already in place [3]. [4] defined vulnerability as is a hole or a weakness in the application, which can be a design flaw or an implementation bug, that allows an attacker to cause harm to the stakeholders of an application. Also, it is a system susceptibility or flaw in the design of the hardware or software and can be exploited to gain unauthorized access [5].

Cybersecurity is seen as a contest of intelligence between a hacker trying to uncover gaps and a designer trying to close them. The advantage that the attackers have is that they only need to locate one weakness, whereas the designer needs to identify and address every vulnerability to ensure successful security. Also, Web applications do present several security issues as a result of improper coding. Websites rely on databases to send important data to actors, who are then susceptible to at least one of the numerous hacking approaches, put your sensitive information at considerable danger by launching a web attack [6]. Web application vulnerabilities are the primary cause of the majority of security risks on the internet [7]. The findings of this risk assessment can assist system managers and developers in becoming aware of potential dangers so they can take steps to mitigate and prevent them [8].

[1] College of Information Technology and Computer Science, University of the Cordilleras, Baguio City, Philippines

[2]College of Information Technology and Computer Science, University of the Cordilleras, Baguio City, Philippines

[1] efm6960@students.uc-bcf.edu.ph, [2] rnmonreal@uc-bcf.edu.ph

Academic institutions are not spared from this. The goal of higher education has always been to provide students with the knowledge and skills they need to comprehend socioeconomic issues and apply their learning for resource usage that promotes harmony, prosperity, and inclusive progress. The characteristic has to do with teaching students these values and abilities so they can become capable and original thinkers who are self-assured enough to see issues and look for solutions [9].

Due to the pandemic, universities and colleges resorted to online education. Classes were conducted synchronous and asynchronously. The use of academic websites was utilized to enhance and deliver quality education and information over the internet. With the use of these technologies, security may be at risk. They could be exposed to attacks and other security threats.

A nonprofit organization called OWASP (Open Web Application Security Project) focuses on enhancing software security. It identifies the most serious security risks to web applications and ranks them according to how frequently they occur and how severely they have an effect [10]. It offers details about common vulnerabilities and resources, such as testing tools and programs that are intentionally insecure, in order to help educate developers about potential security issues that could be present in their code [11].

Effective use of these technologies enables firms to pinpoint gaps in their security, fix vulnerabilities, and improve overall security posture [12]. By fixing vulnerabilities, timely updating in accordance with technology is necessary to reduce cyberattacks [13].

The aim of the study is to evaluate the websites of private Higher Educational Institutions (HEIs) in Region 1 using the OWASP Zap and OWASP Top 10. Through this study, these HEIs will be able to determine the different exposures and vulnerabilities they are at risk for attack. The HEIs will be able to take proactive steps to strengthen their defenses and mitigate potential risks.

II. Literature review

2.1 OWASP ZAP

OWASP Zed Attack Proxy (OWASP ZAP) is a free open source tool being maintained by OWASP teams. It is a penetration testing tool which aids online application developers and security experts in identifying and locating vulnerabilities. Insecure deserialization, compromised authentication, exposed sensitive data, security misconfigurations, SQL injection, cross-site scripting (XSS), and components with known vulnerabilities are just a few examples of the problems it can find in online applications [14].

2.2 OWASP Top 10

The OWASP Top 10 is a frequently updated report that highlights the 10 most important vulnerabilities to web application security. A group of international security specialists from several countries put together the report. In order to reduce and/or mitigate security threats, OWASP refers to the Top 10 as an "awareness document" and advises that all businesses include the report into their procedures [15]. Every 3-4 years, OWASP releases a study listing the top 10 security risks. The study has been updated since the top 10 dangers were first released in 2003. Figure 1 presents the updated top 10 vulnerabilities from 2017 to 2021 [16]. Because OWASP's security standards are thorough and specific dependent on the configuration of the website page and server, the OWASP TOP 10 approach is effective as a security standard for carrying out penetration testing on a website [17].

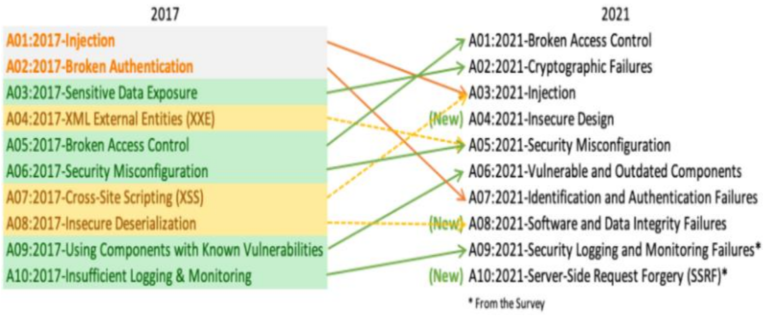

2017
A01:2017-Injection
A02:2017-Broken Authentication
A03:2017-Sensitive Data Exposure
A04:2017-XML External Entities (XXE)
A05:2017-Broken Access Control
A06:2017-Security Misconfiguration
A07:2017-Cross-Site Scripting (XSS)
A08:2017-Insecure Deserialization
A09:2017-Using Components with Known Vulnerabilities
A10:2017-Insufficient Logging & Monitoring
2021
A01:2021-Broken Access Control
A02:2021-Cryptographic Failures
A03:2021-Injection
(New) A04:2021-Insecure Design
A05:2021-Security Misconfiguration
A06:2021-Vulnerable and Outdated Components
A07:2021-Identification and Authentication Failures
(New) A08:2021-Software and Data Integrity Failures
A09:2021-Security Logging and Monitoring Failures*
(New) A10:2021-Server-Side Request Forgery (SSRF)*
* From the Survey

Fig. 1: OWASP Top 10 for 2021

The main reason behind broken access control vulnerability is failing to adhere to secure design principles like enforce appropriate input validation and take action to limit critical information disclosure, setting up secure sessions, and management, control of the readability of the directory [18].

Also, [19] stated that 80 percent of the websites that underwent normal testing still had vulnerabilities to SQL injection attacks, indicating that this significant threat exists for web applications. Attackers can gain access to private data, including databases, through flaws in web programs. This enables the attacker to get data straight from the database.

III. METHODOLOGY

This study utilized the conceptual framework as shown in Figure 2. The steps involved are internet research, vulnerability scanning using OWASP ZAP and report analysis.

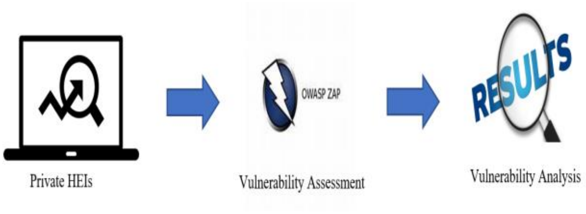
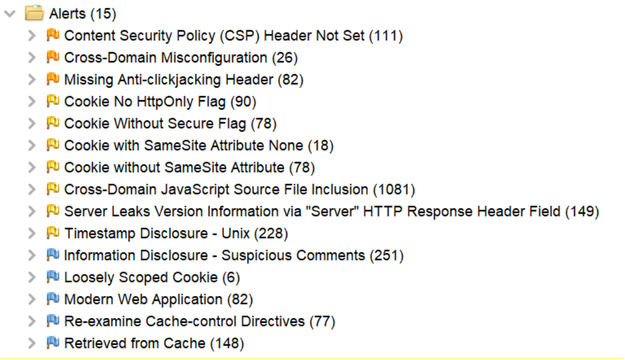


Fig 2: Conceptual Framework

In the internet research phase, a review of researches, blogs and articles was undertaken to have an understanding of the common vulnerabilities and exposures. Also, private HEIs websites within the region was considered for the study.

For vulnerability assessment, the websites was evaluated using the OWASP Zed Attack Proxy (OWASP Zap) 2.12.0 tool. It is a free open source vulnerability tool on Windows 10 operating system. A session was persisted in order to save the analysis session. The website was the starting point of the scan.

After the assessment, alerts or vulnerabilities were automatically generated by the assessment tool as shown in Figure 3.

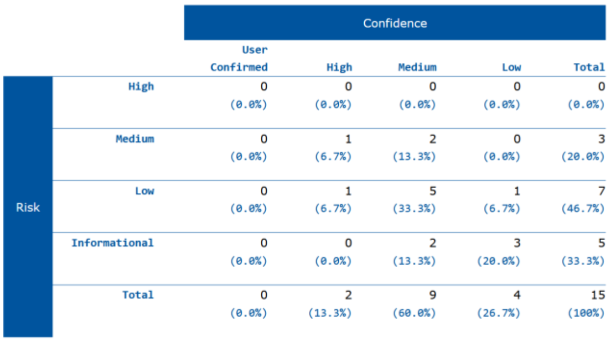


Alerts (15)
- Content Security Policy (CSP) Header Not Set (111)
- Cross-Domain Misconfiguration (26)
- Missing Anti-clickjacking Header (82)
- Cookie No HttpOnly Flag (90)
- Cookie Without Secure Flag (78)
- Cookie with SameSite Attribute None (18)
- Cookie without SameSite Attribute (78)
- Cross-Domain JavaScript Source File Inclusion (1081)
- Server Leaks Version Information via "Server" HTTP Response Header Field (149)
- Timestamp Disclosure - Unix (228)
- Information Disclosure - Suspicious Comments (251)
- Loosely Scoped Cookie (6)
- Modern Web Application (82)
- Re-examine Cache-control Directives (77)
- Retrieved from Cache (148)

Fig. 3: Vulnerability Analysis Result

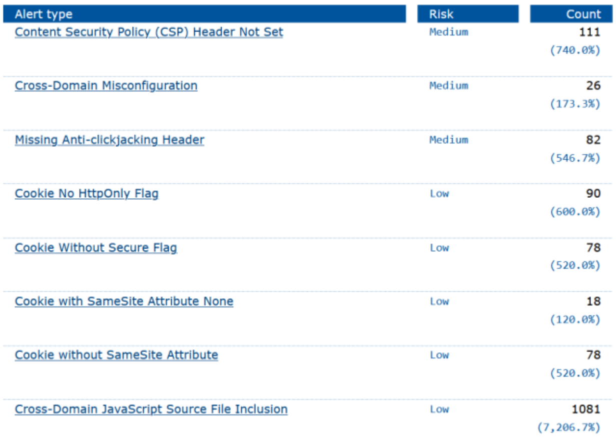
Figure 4 presents the ZAP scanning report which shows the alert counts by risk and confidence with the alert's risk level.

Figure 5 presents the alert counts by alert type wherein the percentages represent the total number of alerts included in the report.



		Confidence				
		User Confirmed	High	Medium	Low	Total
Risk	High	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
	Medium	0 (0.0%)	1 (6.7%)	2 (13.3%)	0 (0.0%)	3 (20.0%)
	Low	0 (0.0%)	1 (6.7%)	5 (33.3%)	1 (6.7%)	7 (46.7%)
	Informational	0 (0.0%)	0 (0.0%)	2 (13.3%)	3 (20.0%)	5 (33.3%)
	Total	0 (0.0%)	2 (13.3%)	9 (60.0%)	4 (26.7%)	15 (100%)

Fig. 4: Alert Counts by Risk and Confidence



Alert type	Risk	Count
Content Security Policy (CSP) Header Not Set	Medium	111 (740.0%)
Cross-Domain Misconfiguration	Medium	26 (173.3%)
Missing Anti-clickjacking Header	Medium	82 (546.7%)
Cookie No HttpOnly Flag	Low	90 (600.0%)
Cookie Without Secure Flag	Low	78 (520.0%)
Cookie with SameSite Attribute None	Low	18 (120.0%)
Cookie without SameSite Attribute	Low	78 (520.0%)
Cross-Domain JavaScript Source File Inclusion	Low	1081 (7,206.7%)

Fig. 5: Alert Counts by Alert Type

IV. RESULTS AND DISCUSSION

The results of the vulnerability assessments performed on the selected higher educational institutions (HEI) is presented. As presented in Figure 6, out of the selected HEIs, HEI5 got the highest number of vulnerabilities of 19, HEI4 got a total to 15 vulnerabilities, HEI2 got 13 vulnerabilities, HEI1 and HEI3 got 12 vulnerabilities each and HEI6 got the lowest number of vulnerabilities which is 11.

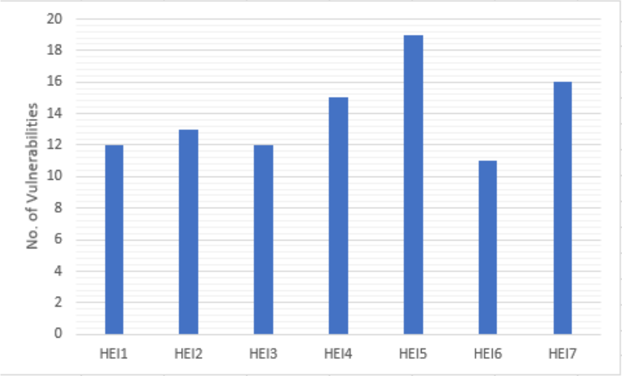


Fig. 6: Vulnerability Summary for the HEIs

Vulnerability Analysis of HEI1.

Out of the 12 vulnerabilities identified, 33.3% were classified as medium, while 41.7% were low. 4 vulnerabilities were also considered as informational. The threat level for the server could be said to be medium since most of the vulnerabilities were medium as shown in Figure 7.

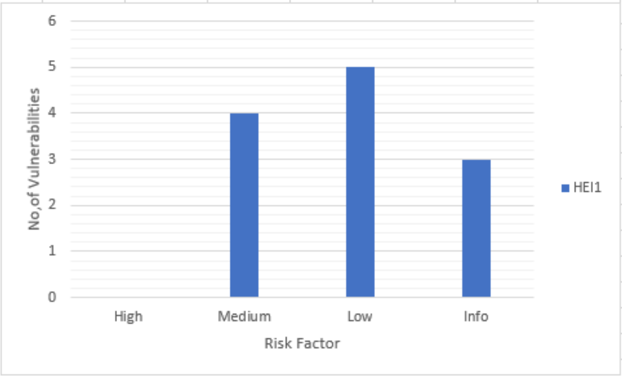


Fig. 7: Vulnerabilities in HEI1

Vulnerability Analysis of HEI2

Figure 8 presents the vulnerability analysis of HEI2. Out of the 13 vulnerabilities, 1 of the severity was posted as high, 4 were posted as medium 4 were low and 4 were informational. The threat level for this HEI was high because the most severe among the identified vulnerabilities was high.

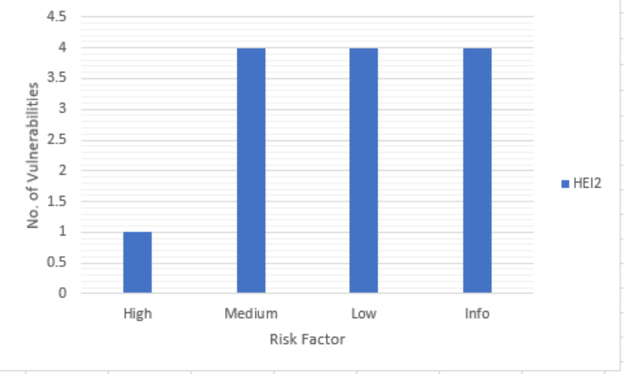


Fig. 8: Vulnerabilities in HEI2

Vulnerability Analysis of HEI3

For HEI3, out of the 12 vulnerabilities identified, medium, low and informational was able to get 4 severities each. The threat level for HEI3 was medium because the highest vulnerability identified was also medium as shown in Figure 9.

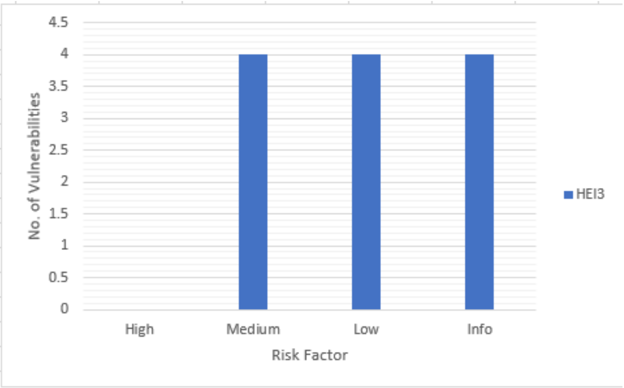


Fig. 9: Vulnerabilities in HEI3

Vulnerability Analysis of HEI4

As presented in Figure 10, HEI 4 was posted with 3 severities as medium, 7 or 20% of severities as low and 5 or 46.7%of severities as informational. The threat level for HEI4 was low.

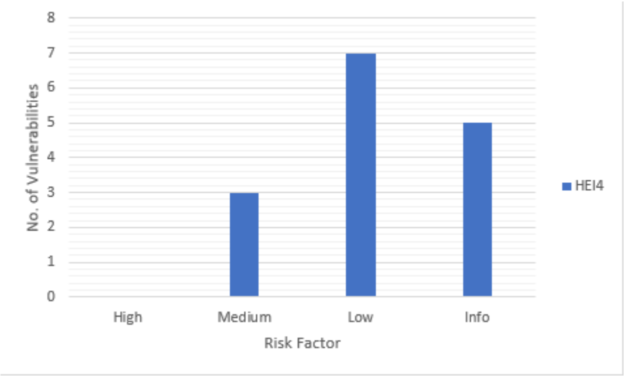


Fig. 10: Vulnerabilities in HEI4

Vulnerability Analysis of HEI5

The vulnerability analysis of HEI5 is presented in Figure 11. Out of 19 vulnerabilities, 1 has a risk factor of high, 3 has a risk factor of medium, 9 had a risk factor of low and 6 were informational. The threat level of HEI5 which got the highest number of vulnerabilities among the selected HEIs was high.

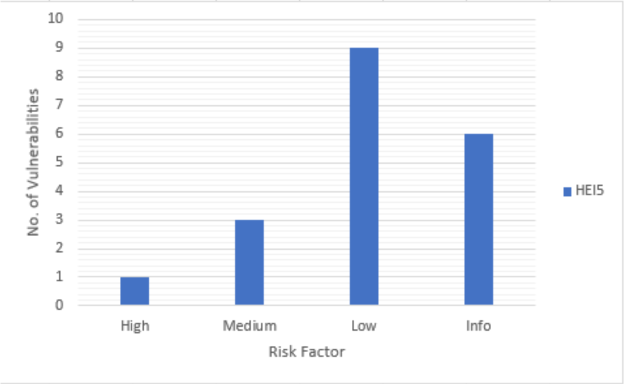


Fig. 11: Vulnerabilities in HEI5

Vulnerability Analysis of HEI6

Figure 12 shows the vulnerability analysis of HEI6. It can be seen that out of the 11 vulnerabilities, 3 has a severity of medium, 7 has a severity of low and 5 were informational. HEI6 got the lowest number of vulnerabilities among the selected HEIs. The threat level of HEI6 is low.

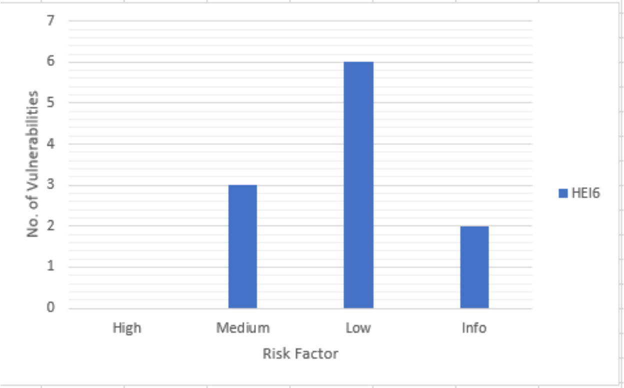


Fig. 12: Vulnerabilities in HEI6

Vulnerability analysis of HEI7

Figure 13 presents the 16 vulnerabilities of HEI7. 37.8% has a severity of medium, 5 had a severity of low and 5 were also informational. The threat level of HEI7 was medium.

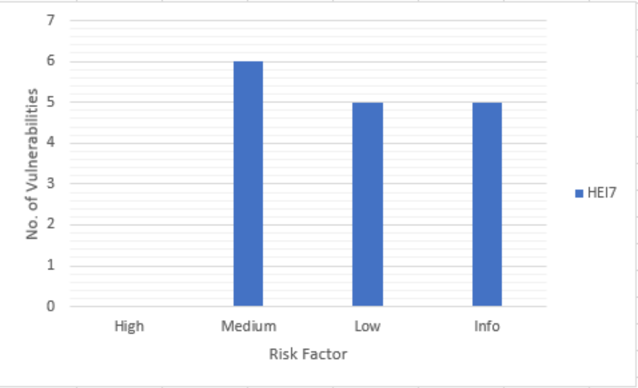
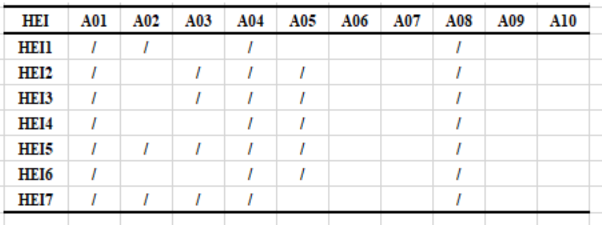


Fig. 13: Vulnerabilities in HEI7

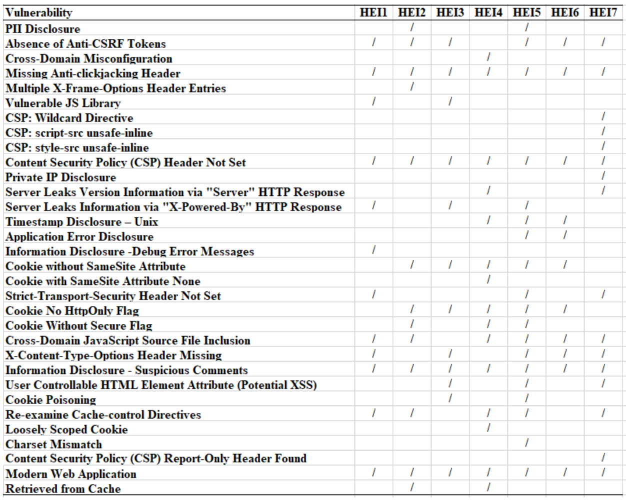
Figure 14 presents the top 10 web application security risks present in the selected HEIs. A01 (Broken Access Control), A04 (Insecure Design) and A08 (Software and Data Integrity Failures) were all present in the selected HEIs. Security misconfiguration was present in HEI2, HEI3, HEI4, HEI5 and HEI6. Injection was found to be vulnerable in HEI2, HEI3, HEI5 and HEI7. HEI1, HEI5 and HEI7 was found to be vulnerable with Cryptographic failures. Out of the selected HEIs, HEI5 got 6 out of 10 from the OWASP top 10 web application security risks. This was the most vulnerable among all the HEIs.



HEI	A01	A02	A03	A04	A05	A06	A07	A08	A09	A10
HEI1	/	/		/				/		
HEI2	/		/	/	/			/		
HEI3	/		/	/	/			/		
HEI4	/			/	/			/		
HEI5	/	/	/	/	/			/		
HEI6	/			/	/			/		
HEI7	/	/	/	/				/		

Fig. 14: OWASP Top 10

Figure 15 shows the vulnerabilities that were identified during the scanning of the selected HEIs. There was a total of 32 vulnerabilities.



Vulnerability	HEI1	HEI2	HEI3	HEI4	HEI5	HEI6	HEI7
PII Disclosure		/			/		
Absence of Anti-CSRF Tokens	/	/	/		/	/	/
Cross-Domain Misconfiguration				/			
Missing Anti-clickjacking Header	/	/	/	/	/	/	/
Multiple X-Frame-Options Header Entries		/					
Vulnerable JS Library	/		/				
CSP: Wildcard Directive							/
CSP: script-src unsafe-inline							/
CSP: style-src unsafe-inline							/
Content Security Policy (CSP) Header Not Set	/	/	/	/	/	/	/
Private IP Disclosure							/
Server Leaks Version Information via "Server" HTTP Response				/			/
Server Leaks Information via "X-Powered-By" HTTP Response	/		/		/		
Timestamp Disclosure – Unix				/	/	/	
Application Error Disclosure					/	/	
Information Disclosure -Debug Error Messages	/						
Cookie without SameSite Attribute		/	/	/	/	/	
Cookie with SameSite Attribute None				/			
Strict-Transport-Security Header Not Set	/				/		/
Cookie No HttpOnly Flag		/	/	/	/	/	
Cookie Without Secure Flag		/		/	/		
Cross-Domain JavaScript Source File Inclusion	/	/		/	/	/	/
X-Content-Type-Options Header Missing	/		/		/	/	/
Information Disclosure - Suspicious Comments	/	/	/	/	/	/	/
User Controllable HTML Element Attribute (Potential XSS)			/		/		/
Cookie Poisoning			/		/		
Re-examine Cache-control Directives	/	/		/	/		/
Loosely Scoped Cookie				/			
Charset Mismatch					/		
Content Security Policy (CSP) Report-Only Header Found							/
Modern Web Application	/	/	/	/	/	/	/
Retrieved from Cache		/		/			

Fig. 15: Vulnerabilities Listing

V. CONCLUSIONS

The study aimed to evaluate the websites of private HEIs in Region 1 to determine the different exposures and vulnerabilities these HEIs may be at risk for attacks. The websites of private universities and colleges in Region 1 were evaluated using OWASP Zap and OWASP Top 10. The websites utilized in the study has vulnerabilities. Broken Access Control, Insecure Design and Software and Data Integrity Failures were the top vulnerabilities the HEIs were exposed to. A total of 32 vulnerabilities were also determined in the study. Necessary actions must be taken by the website administrators to come up with solutions so as not to compromise the operations of the institutions. OWASP Zap is a suitable tool for website vulnerability and exposure evaluation. This tool may be utilized by other agencies, organizations, developers to evaluate the security of their websites.

REFERENCES

[1] Laksmiati, Dewi. (2023). Vulnerability Assessment with Network-Based Scanner Method for Improving Website Security. Journal of Computer Networks, Architecture and High Performance Computing. 5. 38-45. 10.47709/cnahpc.v5i1.1991.

[2] Hassan, Sib. (2022). Analysis of Vulnerabilities in System by Penetration Testing. Pakistan Journal of Scientific

[3] Rapid7. 2023. Vulnerabilities, Exploits, and Threats Explained. Retrieved from https://www.rapid7.com/fundamentals/vulnerabilities-exploits-threats/

[4] OWASP. 2023. Vulnerabilities. Retrieved from https://owasp.org/www-community/vulnerabilities/#:~:text=A%20vulnerability%20is%20a%20hole,that%20rely%20on%20the%20application.

[5] Bihari, Saket. (2023). Cyber Security as an Academic Discipline: Challenges and Opportunities.

[6] Azizi, Neda & Haass, Omid. (2022). Cybersecurity Issues and Challenges. 10.4018/978-1-6684-5284-4.ch002.

[7] Khan, Bilawal & Bangash, Javed & Tariq, Muhammad & Gul, Nida & Zahir, Sana & Kamal, Akhtar. (2023). A Comparative Model to Analyze Various Web Application Penetration Testing Tools for Different Vulnerabilities. 1st International Conference on Computing Technologies, Tools and Applications (ICTAPP-23). At: Institute of Computer Sciences and Information Technology (ICS/IT), The University of Agriculture Peshawar, Pakistan.

[8] Firman Ashari, Ilham & Oktariana, Vina & Sadewo, Galih & Damanhuri, Salman. (2022). Analysis of Cross Site Request Forgery (CSRF) Attacks on West Lampung Regency Websites Using OWASP ZAP Tools. Jurnal Sisfokom (Sistem Informasi dan Komputer). 11. 276-281. 10.32736/sisfokom.v11i2.1393.

[9] Shukla, Manoj. (2023). Quality Enhancement in Higher Education Institutions. International Journal of Multidisciplinary Research and Analysis. 06. 10.47191/ijmra/v6-i2-39.

[10] Edudwar. 2023. OWASP Top 10 Vulnerabilities 2023. Retrieved from https://www.edudwar.com/owasp-top-10-vulnerabilities/

[11] Poston, Howard. (2020). Mapping the OWASP Top Ten to Blockchain. Procedia Computer Science. 177. 613-617. 10.1016/j.procs.2020.10.087.

[12] Singirikonda, Manikanta. (2023). Penetration Testing Tool Guide. Journal of Cybersecurity.

[13] Nidhi, Nidhi & Kadam, Sachin. (2022). A SURVEY ON VULNERABILITY ASSESSMENT OF ACADEMIC WEBSITES SECURITY IN INDIA. VIII. 2021-2022.

[14] HackerOne. (2023). OWASP ZAP Tutorial: Installation and Initial Configuration. Retrieved from https://www.hackerone.com/knowledge-center/owasp-zap-6-key-capabilities-and-quick-tutorial

[15] CloudFare. 2023. What is OWASP? What is the OWASP top 10?. Retrieved from https://www.cloudflare.com/learning/security/threats/owasp-top-10/

[16] Nedeljković, N., Vugdelija, N., & Kojić, N. (2020, October). Use of "OWASP Top 10" in web application security. In Fourth International Scientific Conference on Recent Advances in Information Technology, Tourism, Economics, Management and Agriculture (p. 25). DOI: https://doi.org/10.31410/ITEMA.2020.25

[17] Nurbojatmiko & Lathifah, Ari & Amri, Faaza & Rosidah, Ani. (2022). Security Vulnerability Analysis of the Sharia Crowdfunding Website Using OWASP-ZAP. 1-5. 10.1109/CITSM56380.2022.9935837.

[18] Hassan, M., Ali, M., Bhuiyan, T., Sharif, M., & Biswas, S. (2018, October). Quantitative assessment on broken access control vulnerability in web applications. In International Conference on Cyber Security and Computer Science 2018.

[19] Alanda, Alde & Satria, Deni & Ardhana, M.Isthofa & Dahlan, Andi & Mooduto, Hanriyawan. (2021). Web Application Penetration Testing Using SQL Injection Attack. JOIV : International Journal on Informatics Visualization. 5. 320. 10.30630/joiv.5.3.470.