Implementation of a Security Strengthening System to Mitigate Vulnerabilities in Academic Web Applications

Ing. Jimmy Fernando Ramírez Márquez
Investigador Independiente, jramirez@gmail.com

Ing.Joffre Stalin Monar Monar
Escuela Superior Politécnica de Chimborazo, omartinez@espoch.edu.ec

Ing.Oswaldo Geovanny Martínez Guashima
Escuela Superior Politécnica de Chimborazo, omartinez@espoch.edu.ec

Ing.Luis Fabian Brito Mancero
Investigador Independiente, Lbritom0977@hotmail.com

Ing. Marcela Yolanda Brito Mancero
Escuela Superior Politécnica de Chimborazo, mybrito@espoch.edu.ec

Ing. Galuth Irene Garcia Camacho
Universidad Estatal de Bolivar, ggarcia@ueb.edu.ec

Abstract

This research work proposes the implementation of a system to strengthen the security of academic web applications based on a study carried out at a public university in Ecuador. Within a comparative analysis, OWASP is selected as the base methodology for the development of tests, OWASP ZAP as the penetration test tool, and within the vulnerability scanning, the ten (10) categories of the OWASP 2021 methodology are taken into account. We will focus on two (2) test scenarios; the first one, in which we will examine vulnerabilities in web applications in production and, the second one, which we will apply the methodology we have established in a controlled environment; according to data obtained from the test scenarios, it was contrasted that the application of this security strengthening system reduces vulnerabilities in academic web applications by 98.83%, and with the statistical test of chi square with a confidence level of 95% it is demonstrated that this methodology does reduce the number of vulnerabilities, resulting in the use of more secure academic web applications.Keywords- Web application, security, vulnerability.

Keywords: *Security, Vulnerability. Web Application.*

I. INTRODUCTION

Digital statistics for 2022 [1] show that 75.6% of people in Ecuador use the Internet, while 62.5% of people worldwide have access to it. A 33.5% increase in Internet use is evident if we compare this figure with that of 2021.

This has also led to an increase in the number of electronic devices that consumers use to

access the web, which has led to an increase in the number of routine tasks being moved to web applications.

Web apps, according to [2], are programs that let users connect to a web server across a network using a particular browser.

According to [3], a vulnerability is a flaw in a computer asset that can be used by a threat to compromise its availability, integrity, and confidentiality.

Most web applications are created with the intention of providing a specific function, often leaving security on the back burner. This can lead to risks of cyber attacks that exploit vulnerabilities in web applications at some point in their lifetime.

Web application security is an increasingly important issue today, due to the large amount of sensitive data being handled online. Web applications are vulnerable to a wide range of security threats, which can expose users' personal and financial information to malicious attackers.

Thus, security in academic web applications that are the subject of this study is particularly important because these applications contain personal and financial information of students and faculty, as well as sensitive data related to academic performance, attendance, and medical records. A successful attack on an academic web application could result in the disclosure of confidential information, manipulation of academic records, or disruption of academic operations.

Therefore, security in academic web applications is essential to protect the personal and financial information of students and faculty, prevent tampering with academic records, maintain academic integrity, and reduce legal and financial risks. Educational institutions must implement effective security measures in their academic web applications to ensure the protection of confidential data and the well-being of their users.

In Esmeraldas, Ecuador, is the Universidad Técnica Luis Vargas Torres (UTELVT). Almost 12,000 students are enrolled at this public institucion of higher education, which offers a variety of professional paths. They use the academic web applications for enrollment processes and grade reviews, among other things.

The main objective of this research is to propose a security hardening system in this University and to reduce the vulnerabilities found in academic web applications. To verify the security level of this methodology, a vulnerability analysis was performed in two (2) scenarios; before and after the implementation of the hardening system.

This article is organized as follows: the second section determines the tool to be used for penetration testing and the definition of the methodology to mitigate vulnerabilities in academic web applications; the third section describes the results obtained from the analysis of vulnerabilities detected before and after the methodology; and finally the fourth section details some conclusions of the research work.

II. MATERIALS AND METHODS

To achieve the purpose of this research, the following procedure was applied:

A. Research Design

Since instruments in scales or numerical ranges were used, this study has a quantitative approach. This methodology allows the collection of sufficient data for the analysis of the study variables.

Since the vulnerabilities to which the web applications of a Public University of Ecuador are exposed were evaluated in a determined academic time, the research design is quasi-experimental, transversal and descriptive.

The descriptive scope facilitates the specification of the vulnerabilities of online academic applications. However, using the deductive approach in this study, it was possible to determine which vulnerabilities needed to be corrected first to strengthen online academic applications [4].

B. Penetration testing tool selection

A penetration test, sometimes known as a pent test, is a process that employs a number of techniques and methodologies to mimic an attack on a system and allow for the evaluation of the security of computer systems, networks, and applications, according to [5].

There are a number of methodologies for penetration testing used in the field of systems auditing; however, a study [6] found that OWASP is the best methodology for this research. Another factor taken into consideration for decision-making were the stages that each of these methodologies go through for the development of tests and are described in [7].

Due to the approach required for the application of the methodology, it is demonstrated that OWASP ZAP is the best penetration tool [8] that performs web application vulnerability detection with the best efficiency in the study.

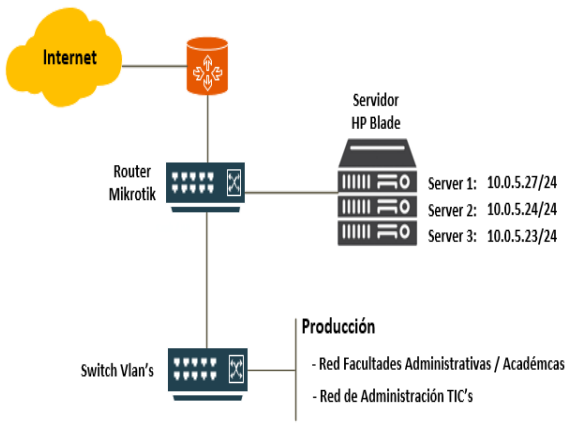
TABLE I: SELECTION OF THE TOOL FOR TESTING

Tool	CHARACTERÍSTICS					
	User-friendly interface	Ease of installation	Legible final report	Detection speed	System requirements	Total
Acunetix	5	3	5	5	3	21
Nmap	5	2	3	3	3	16
Owasp zap	5	5	5	5	3	**23**
Kali Linux	5	3	5	3	3	19

C. Test Scenarios

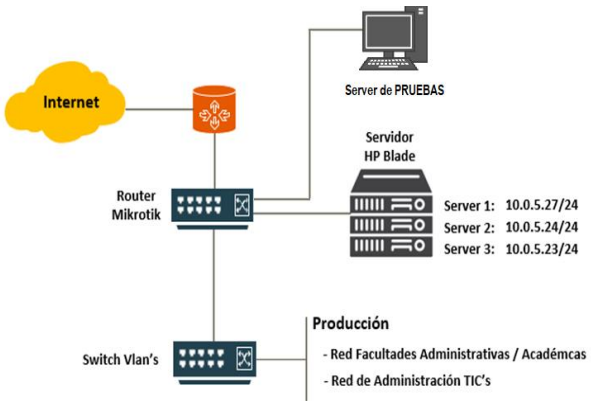
Two scenarios were proposed to validate the proposed method: In the first, a security test was performed on the academic web applications of an Ecuadorian public university, and several vulnerabilities were found.

Fig. 1. Scenario 1.



In the second case, protections or countermeasures were suggested for these academic web applications, and corrective measures were subsequently implemented. The security test was then repeated on the web applications under study.

Fig. 2. Scenario 2.



The risk level to be taken into account for each vulnerability to be analyzed is given in Table II according to [9].

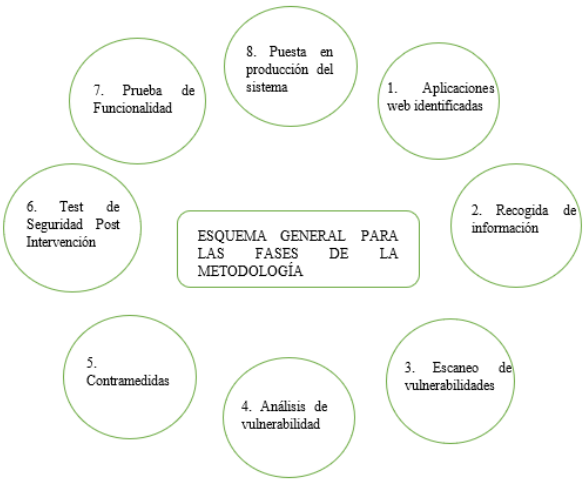
TABLE II: RISK LEVEL

Risk	Impact	Description
High	Between: 6 y <=10	Vulnerability that, if exploited, would compromise the information security, causing a negative impact. negative impact, must be solved immediately.
Medium	Between: 4 y <6	Vulnerability that if exploited would have a slight impact on systems operations. impact on system operations.
Under	Between: 1 y <4	Vulnerability that if exploited would not cause major drawbacks
Informative	Between: 0 y <1	Warnings alerting about possible configuration failures, with very low impact on the operation of the applications.

D. Proposed Techniques

The following general outline shows the methodology for security and integrity in academic web applications.

Fig. 3. General scheme with the 8 phases of the methodology.



GENERAL OUTLINE OF THE METHODOLOGY PHASES

1. Web applications identified

It was determined which group of academic web applications are currently in production and which will be subjected to penetration testing to look for vulnerabilities.

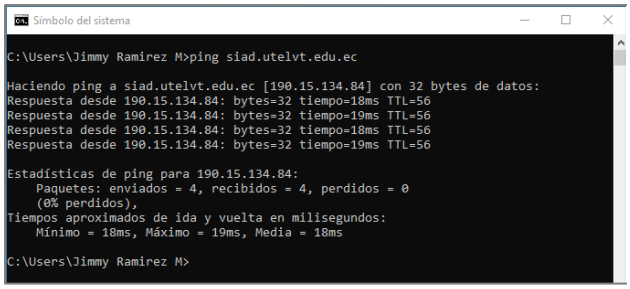
TABLE III: UTELVT ACADEMIC WEB APPLICATIONS

Academia Group	Web Address	Status
Student Enrollment	https://estudiante.utelvt.edu.ec/	in service
Academic Management	https://administrativo.utelvt.edu.ec/	in service
Teachers	https://docente.utelvt.edu.ec/	in service
SIAD	https://siad.utelvt.edu.ec/	in service

2. Information gathering.

Using the Windows ping command, the IPs of the academic web pages were identified.

Fig. 4. Executing the ping command

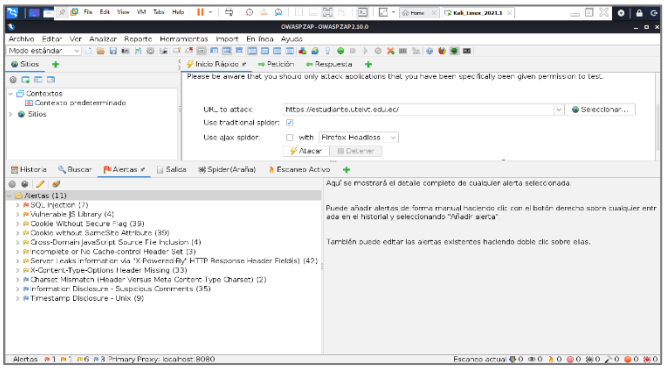


3. Vulnerability scanning

The OWASP ZAP tool was used to scan for vulnerabilities.

Figure 5 shows the results of the scan performed on one of the web applications. There are seven (7) critical flaws. They include four (4) library vulnerabilities and the remaining SQL injection vulnerabilities.

Fig. 5. Vulnerability Scan of Web Application 1.



4. Vulnerability Analysis

An analysis of the vulnerabilities found in the scanning of the academic web applications in production was performed.

After scanning the academic web applications of the University in production, thirteen (13) types of vulnerabilities were found with a total of 342 incidents, which are shown in Table IV.

TABLE IV: VULNERABILITIES DETECTED

Vulnerability	Grade	Quantity	OWASP Category	Solution
Cookie without SameSite attribute	Under	49	A05	The SameSite attribute must be set to 'lax' or preferably 'strict' for all cookies.
The server discloses information via an HTTP "X-Powered-By" response header field(s)	Under	57	A05	Verify that the web server, application server, etc. is configured to suppress "X-Powered-By" headers.

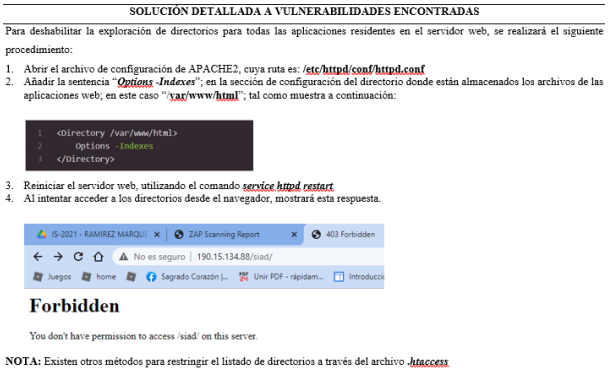
Vulnerable JS Library	Medium	8	A06	Bootstrap should be upgraded to its latest version
The X-Content-Type-Options header is missing	Under	63	A04	Verify that the web application/server sets the Content-Type header correctly and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.
Inclusion of cross-domain JavaScript source files	Under	4	A05	JavaScript source files should be loaded only from trusted sources, and end users of the application should not be able to control the sources.
Directory browsing	Medium	2	A04	Directory scanning should be disabled. Verify that none of the listed files pose a risk if necessary.
Time Stamp Disclosure – Unix	Under	25	A04	Manually confirm that timestamp data is not sensitive, and that it cannot be aggregated into exploitable disclosure patterns.
Disclosure of information	Informative	79	A04	Remove any comment that returns information that could help the attacker
Secure pages include mixed content	Under	3	A02	A page that is available via SSL/TLS must not have any content that is transmitted over unencrypted HTTP.
Cookie without Secure flag	Under	44	A05	Cookies that have session tokens or contain sensitive data must always be transmitted over an encrypted channel. Verify that the secure flag is set for these cookies.
Incompatibility of characters	Informative	2	A04	Force UTF-8 for all text content, both in HTPP header and meta tags in HTML or encoding declarations in XML
Incomplete header set or no cache control	Under	5	A04	Verify that the HTTP cache control header is set to no-cache, no-store, must-revalidate.
Missing Content-Type header	Informative	1	A04	each page must set the content type value specific to and appropriate for the content being delivered.
Total vulnerabilities		342		

5. Implementation of the countermeasure

For this step a test environment was created with a copy of the web server in production.

Figure 6 below shows the solution to a vulnerability taken from the methodology designed to combat the thirteen (13) vulnerabilities detected.

Fig. 6. Solution to a vulnerability found.



SOLUCIÓN DETALLADA A VULNERABILIDADES ENCONTRADAS

Para deshabilitar la exploración de directorios para todas las aplicaciones residentes en el servidor web, se realizará el siguiente procedimiento:

1. Abrir el archivo de configuración de APACHE2, cuya ruta es: **/etc/httpd/conf/httpd.conf**
2. Añadir la sentencia "***Options -Indexes***"; en la sección de configuración del directorio donde están almacenados los archivos de las aplicaciones web; en este caso "**/var/www/html**"; tal como muestra a continuación:

```
<Directory /var/www/html>
    Options -Indexes
</Directory>
```

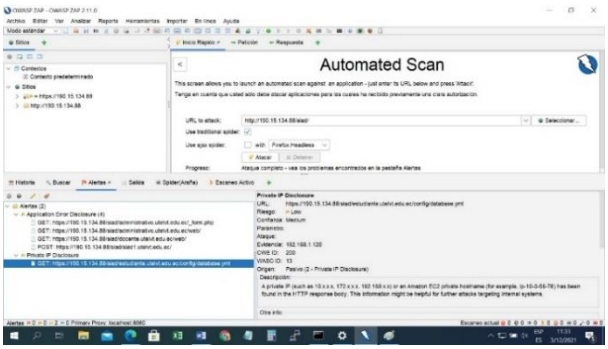
3. Reiniciar el servidor web, utilizando el comando ***service httpd restart***
4. Al intentar acceder a los directorios desde el navegador, mostrará esta respuesta.

NOTA: Existen otros métodos para restringir el listado de directorios a través del archivo ***.htaccess***

6. Post Intervention Security Test

Penetration testing was performed after implementing the safeguards designed based on the vulnerabilities found in the academic web applications with the purpose of testing the strengthening of the security of the web applications under study, all this in a testing environment as shown in Fig. 7.

Fig. 7. Vulnerability scanning Scenario 2



From the scan it is observed that there are threats that still prevail, but they are informative and are motivated by the changes that were made to the web applications in the test environment, these vulnerabilities will disappear when they are implemented in the production environment.

7. Functionality Tests

Once the correction of the vulnerabilities is verified, a verification of the applications was made, by means of the navigation in the browser, examining the functionality and operability of all its components.

8. Implementation of the strengthened system

Once all the previous phases have been completed, we first reconfigured the real web server, taking into account the proposed countermeasures, and then we copied the intervened applications.

III. RESULTS

A. Vulnerability determination results

The following vulnerabilities were discovered through the scanning of academic web applications, both before and after applying the methodology. The vulnerabilities were weighted after performing the corresponding tests and scanning with the OWASP ZAP tool with the intention of analyzing them and prioritizing the safeguards to be applied to minimize the security risks of the University's academic web applications.

This is a list of the vulnerabilities that were found during the scanning of the academic applications both before and after using the methodology.

TABLE V: VULNERABILITIES DETECTED BEFORE AND AFTER APPLYNG THE METHODOLOGY

WEB APPLICATIONS	Vulnerabilities			
	Before	After	Corrected	% Corrected
estudiante.utelvt.edu.ec/	205	1	204	99,51 %
administrativo.utelvt.edu.ec/	72	1	71	98,61 %
docente.utelvt.edu.ec/	60	1	59	98,33 %
siad.utelvt.edu.ec/	5	1	4	80,00 %
Total	342	4	338	98,83 %

The table shows that 98.83% of the occurrences of vulnerabilities were corrected, demonstrating the effectiveness of the proposed methodology.

Fig. 8. Vulnerabilities detected and corrected in the two scenarios.

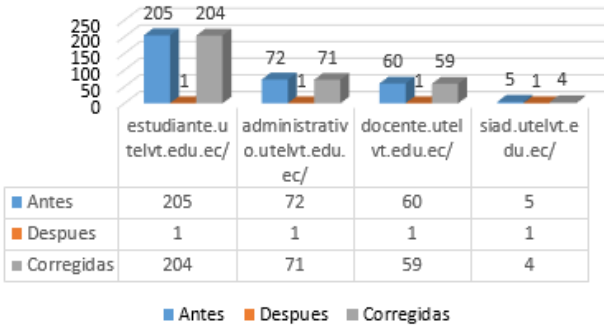


Fig. 8 shows a significant reduction in the number of vulnerabilities detected and corrected.

B. Hypothesis Testing

What is relevant in this research is the analysis of the findings that show the beneficial effects of implementing the system to improve security and integrity in order to reduce vulnerabilities in academic web applications. These findings include:

• Ha= The implementation of a security methodology is significantly related to the number of mitigated vulnerabilities in the academic web application.

• Ho= The implementation of a security methodology is not significantly related to the number of mitigated vulnerabilities in the academic web application.

The analysis of the variables defined to prove the hypothesis is detailed in Table VI.

TABLE VI: FREQUENCY OF OBSERVED VALUES

Observed Values			
Vulnerabilities	Before	After	Total
Quantity detected	342	4	346
Quantity eliminated	0	338	338
Total	342	342	684

As shown in Table VII, the Chi-square value is 629.59 with 1 degree of freedom and a significance level of 0.05, the critical value is therefore 3.84; so X^2 calc is greater than X^2 Critical, so the H0 is rejected and Ha is accepted, so the implementation of a security methodology is significantly related to the number of vulnerabilities mitigated in the academic web application and therefore improves the level of security in these web applications.

TABLE VII: RESULTS OF THE CHI-SQUARE TEST

	Value	gl	Significance level
Pearson´s Chi-square	629,59	1	0,05

IV. DISCUSIÓN

Web application security is a topic of great importance, as more and more people and companies depend on them to carry out critical activities. There is a lot of research that has been done on this topic, some of which have found certain weaknesses in web application security, while others have proposed solutions to improve it.

In the work of [10], using their applied type research, they evaluated the computer security of a web page hosted free of charge by the Firavitoba technical institution for the

detection and remediation of information risks and vulnerabilities. This led to the tabulation of possible solutions, and they were able to implement corrective measures to prevent possible failures and loss of information based on the vulnerabilities found in the analyzed system, but no security policies were established to minimize risks.

On the other hand, using the OWASP methodology, [11] performed a descriptive, cross-sectional and quantitative analysis on the e-commerce system siembraviva.com, and [8] did the same for the website of the distance and virtual education department of the technical university of Ambato, but some processes need to be updated.

In a similar vein, the study by [12] elaborated policies for security risk management in software development at the Judicial Council and posed two scenarios, before and after applying the policies, but used another methodology.

Although these researches are important to improve the security level in web applications, some of them do not focus on academic applications and others do not work with the proposed methodology.

The present research proposes a security and integrity strengthening system for academic web applications using the OWASP methodology [9] as a basis, which will serve to mitigate the vulnerabilities that occur in this type of applications, and can be applied and implemented in any Higher Educational Institution to improve information protection.

V. CONCLUSIONS

The main contribution of this research work is to propose a methodology for the security and integrity of academic web applications, which focuses on a security test with 8 phases to be implemented in order to reduce vulnerabilities in web applications.

This work proves that the implementation of a security strengthening system is significantly related to the number of vulnerabilities found in academic web applications and therefore the improvement in the level of security.

The present study can be complemented by adding other phases to the security hardening system derived from the update of the implemented OWASP methodology, or from the appearance of new vulnerabilities.

VI. CONFLICT OF INTEREST

The authors declare that they have no conflict of interest in the development of this research, the results obtained are authentic and are the product of the analysis of the data obtained in the two test scenarios.

VII. CONTRIBUTIONS OF THE AUTHORS

Jimmy Ramirez carried out the research by setting up the necessary infrastructure for the test scenarios; Joffre Monar processed the data and supported in the development of the methodology; Diego Bastidas wrote the article, all authors after several revisions approved the final version for submission to the journal. the methodology; Diego Bastidas wrote the article, all authors after several revisions approved the final version for submission to the journal.

Reference

[1] AMD Agencia Digital, “Cifras Estadísticas Digitales en Ecuador 2022,” AMD Agencia Digital, 2022. https://agenciadigitalamd.com/marketing-digital/estadisticas-digitales-ecuador/ (accessed Feb. 15, 2023).

[2] S. Lujan, Programación de aplicaciones web: historia, principios básicos y clientes web, Editorial. Alicante, 2002.

[3] S. M. Quiroz-Zambrano and D. G. Macías-Valencia, "Seguridad en informática: consideraciones," Dominio de las Ciencias, ISSN-e 2477-8818, Vol. 3, No. Extra 3, 2017, págs. 676-688, vol. 3, no. 3, pp. 676–688, 2017, doi: 10.23857/dom.cien.pocaip.2017.3.5.agos. 676-688.

[4] G. González, "Método deductivo," 2020. https://www.lifeder.com/metodo-deductivo/ (accessed Oct. 25, 2022).

[5] J. L. Ramos Ramos, "Revista de Información, Tecnología y Sociedad," Revista de Información, Tecnología y Sociedad, vol. 8, p. 31, 2013, Accessed: Feb. 21, 2023. [Online]. Available: http://www.revistasbolivianas.ciencia.bo/scielo.php?script=sci_arttext&pid=&lng=es&nrm=iso&tlng=

[6] F. López Provencio, "Metodologias para el desarrollo de software seguro," Universidad Politécnica de Cataluña, 2015. Accessed: Feb. 21, 2023. [Online]. Available: http://upcommons.upc.edu/handle/2099.1/24902

[7] V. K. Álvarez Intriago, "PROPUESTA DE UNA METODOLOGÍA DE PRUEBAS DE PENETRACIÓN ORIENTADA A RIESGOS," Universidad Espíritu Santo, Guayaquil, 2018. Accessed: Oct. 25, 2022. [Online]. Available: http://repositorio.uees.edu.ec/handle/123456789/2525

[8] J. A. Sánchez Freire, "Análisis de vulnerabilidades y diseño de procesos correctivos de la página web de la Dirección de Educación a Distancia y Virtual de la Universidad Técnica de Ambato," Universidad Técnica de Ambato, Ambato, 2017. Accessed: Oct. 25, 2022. [Online]. Available: https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/25531

[9] OWASP Top 10 team, "OWASP Top 10:2021," 2021. https://owasp.org/Top10/ (accessed Oct. 25, 2022).

[10] G. Rincón and F. Albarracín, "Análisis y evaluación de la seguridad informática para la página web publicada en hosting gratuito de la Institución Técnica de Firavitoba, para la detección y remediación de vulnerabilidades y riesgos en la información.," Proyecto aplicado, Universidad Nacional Abierta y a Distancia UNAD, BOYACA, 2018. Accessed: Oct. 25, 2022. [Online]. Available: https://repository.unad.edu.co/handle/10596/17281

[11] J. Días and M. Marulanda, "Aplicación de la metodología de pruebas OWASP para el mejoramiento de la seguridad en el sistema e-commerce siembraviva.com," Proyecto aplicado, Universidad Nacional Abierta y a Distancia UNAD, Manizales, 2018. Accessed: Oct. 25, 2022. [Online]. Available: https://repository.unad.edu.co/handle/10596/20479

[12] R. G. Montalvo Armijos, "Generación de políticas para la gestión de riesgos de seguridad en el desarrollo de software.," Escuela Superior Politécnica de Chimborazo, Riobamba, 2017. Accessed: Oct. 25, 2022. [Online]. Available: http://dspace.espoch.edu.ec/handle/123456789/7224