Jurnal Mandiri IT
Vol. 12 No. 2, October (2023), pp. 58-65
ISSN 2301-8984 (Print)

Penetration testing on mail server website using the OWASP method

Ali Zainal Abidin[1], Hendra Saputra[2], Faldi[3], Muhammad Taufiq Sumadi[4]
[1,2,3,4] Computer Engineering, Muhammadiyah University of East Kalimantan, Samarinda, Indonesia

ARTICLE INFO

Article history:

Received Aug 19, 2023
Revised Aug 21, 2023
Accepted Aug 24, 2023

Keywords:

Acunetix
OWASP Zap
Penetration Testing
Self-testing
System Security

ABSTRACT

Technological advancements have positively impacted various fields, including the Internet. Awareness of system security has become a crucial concern for application developers. Protecting networks from disruptions or hacker attacks can be achieved through self-testing methods, such as Penetration Testing (Pentest). This research conducts a penetration test on the mail server domain, mail.umtk.sch.id, using the tools OWASP Zap and Acunetix. The results of this testing reveal the detection of 9 vulnerabilities and based on the OWASP Top 10 2017 vulnerability categories, five categories were identified: Broken Authentication, Sensitive Data Exposure, Broken Access Control, Security Misconfiguration, and Using Components with Known Vulnerabilities.

Corresponding Author:

Hendra Saputra,
Computer Engineering, Faculty of Science and Technology,
Muhammadiyah University of East Kalimantan,
IR Juanda No 7, Samarinda, East Kalimantan, 75124, Indonesia.
Email: hs048@umkt.ac.id

1. INTRODUCTION

Technological advancements have positively impacted various domains, including the Internet (Galperin & Fernanda Viecens, 2017; Raja & Nagasubramani, 2018). Websites have become a primary alternative for corporations to communicate, self-promote, and interact with audiences (Taneja & Toombs, 2014; Thoma et al., 2018). Furthermore, easy website access has enabled many people to connect anywhere and anytime (Infante & Mardikaningsih, 2022). A digital report from We Are Social (Hootsuite) highlights that Indonesia's internet users reached an impressive 212.0 million in 2023, equivalent to 77% of the total population (We Are Social, 2022).

However, along with these benefits, serious challenges arise concerning data security and information stored within the Internet network (Aulianisa & Indirwan, 2020). Cyber threats such as exploitation, malware, and database injections can compromise data integrity, confidentiality, and availability (Cazorla et al., 2016). This security concern has become increasingly urgent in response to the growing awareness of risks and threats within the digital environment, particularly for application developers (Belapurkar et al., 2009). Consequently, safeguarding valuable information and data from these attacks has become an imperative and crucial matter.

Implementing HTTPS and SSL to secure communication between clients and servers, reduce vulnerabilities, and prevent attacks such as SQL injection (Wibowo et al., 2020). Meanwhile, in the study "Attack Patterns for Black-Box Security Testing of Multi-Party Web Applications," an approach is proposed to test the security of multi-party web applications based on familiar attack patterns (Sudhodanan et al., 2017). Additionally, the "Analysis of Web Security Using Open Web Application Security Project 10" research provides an overview of the security testing framework OWASP Top 10 for web applications (Agreindra Helmiawan et al., 2020). The study "OWASP ZAP

vs Snort for SQLi Vulnerability Scanning" compares vulnerability scanning tools to protect web applications from SQLi threats (Kalaani, 2023).

In response to these challenges, several steps can be taken. One of them is through independent testing, such as self-tests, which can aid in identifying vulnerabilities within the web server (Karimov et al., 2022). The approach of penetration testing, as employed in the Open Web Application Security Project (OWASP) Top 10, has proven effective in identifying and mitigating vulnerabilities within application systems (Li, 2020; Mateo Tudela et al., 2020).

By adopting this approach, the research focuses on implementing penetration testing using the OWASP Top 10 framework on a local Mail Server, which is a duplicate mail server.umkt.sch.id. The main objective is to provide concrete recommendations to reduce vulnerability levels within this system while also contributing to developing safer and more reliable technology. This approach is grounded in a series of previous studies investigating the performance and effectiveness of the OWASP framework. Some of these studies include "A Comparative Study of Web Application Security Parameters" (Shahid et al., 2022), "Evaluation of Web Vulnerability Scanners Based on OWASP Benchmark" (Mburano & Si, 2019), as well as the study "A Security Analysis of Email-Based Password Reset Procedures" (Innocenti et al., 2021). The information obtained from these studies is the foundation for selecting the most appropriate approach to securing the local mail server.

2. RESEARCH METHOD

The research methodology will be carried out through several stages, beginning with the literature review phase, and culminating in the analysis and reporting phase. The chosen stages align with the process outlined in Figure 1.

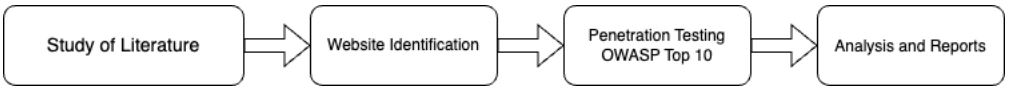


Figure 1. The research processes.

1. Study of Literature

In this phase, a literature survey is conducted to outline supportive theories that serve as the foundation for the research. This literature review draws from various sources such as books, research articles, and online resources.

2. Website Identification

The website to be tested is a local server with the domain mail.umkt.sch.id and an IP address of 192.168.20.17. This website is created using VirtualBox, operating on the Windows 10 operating system. The vulnerability level of the website server will be assessed using several tools designed for penetration testing.

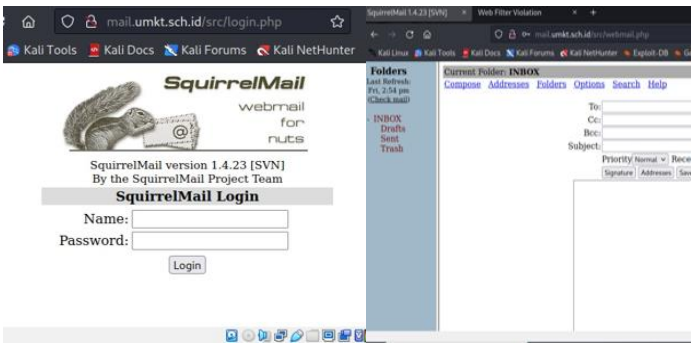


Figure 2. Web Mail server Login and Dashboard pages

Figure 3 displays the SquirrelMail-based webmail server's login page and dashboard interface with the domain mail.umkt.sch.id. The server operates on the Linux Debian 8.11.0 operating system.

3. Penetration Testing

The testing will be conducted using the OWASP Top 10 methodology. The testing process based on the OWASP Top 10 methodology can be referred to in Table 1:

Tabel 1. Pengujian Penetrasi OWASP Top 10

OWASP TOP 10	TOOLS
A1-Injection	Acunetix
A2-Broken Authentication	Acunetix
A3-Sensitive Data Exposure	Acunetix
A4-XML External Entities (XXE)	Acunetix
A5-Broken Access Control	Acunetix
A6-Security Misconfiguration	Acunetix
A7-Cross-Site Scripting (XSS)	Acunetix
A8-Insecure Deserialization	Acunetix
A9-Using Components with Known Vulnerabilities	Acunetix
A10-Insufficient Logging & Monitoring	Acunetix

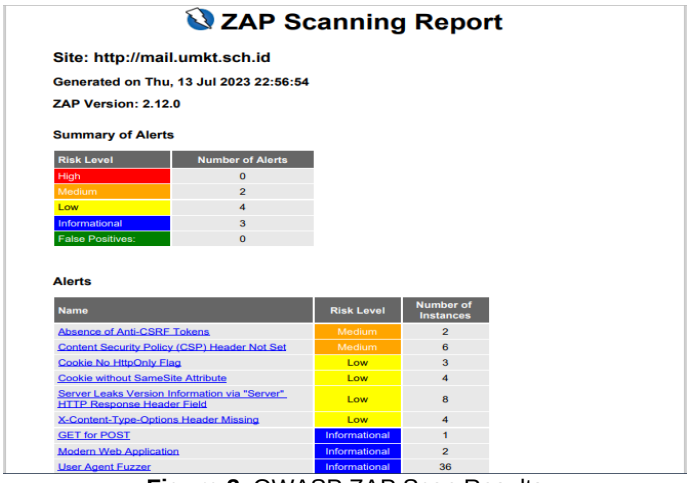
4. Analysis and Report

In this phase, analysis and report of the penetration testing using the OWASP Top 10 methodology will be conducted. The analysis and report will be presented as suggestions or recommendations for improving the mail server website.

3. RESULTS AND DISCUSSIONS

1. Vulnerability Identification

Identification in this research involves utilizing OWASP ZAP tools to determine the vulnerability level present in the web server with the domain server mail.umkt.sch.id. The following presents the scan results obtained from OWASP ZAP:


ZAP Scanning Report

Site: http://mail.umkt.sch.id

Generated on Thu, 13 Jul 2023 22:56:54

ZAP Version: 2.12.0

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	2
Low	4
Informational	3
False Positives:	0

Alerts

Name	Risk Level	Number of Instances
Absence of Anti-CSRF Tokens	Medium	2
Content Security Policy (CSP) Header Not Set	Medium	6
Cookie No HttpOnly Flag	Low	3
Cookie without SameSite Attribute	Low	4
Server Leaks Version Information via "Server" HTTP Response Header Field	Low	8
X-Content-Type-Options Header Missing	Low	4
GET for POST	Informational	1
Modern Web Application	Informational	2
User Agent Fuzzer	Informational	36

Figure 3. OWASP ZAP Scan Results

In Figure 3, the results of the scanning process on the domain mail.umkt.sch.id using OWASP ZAP tools reveal the presence of 9 vulnerabilities. For more detailed information, please refer to Table 2.

Table 2. Vunerability Results

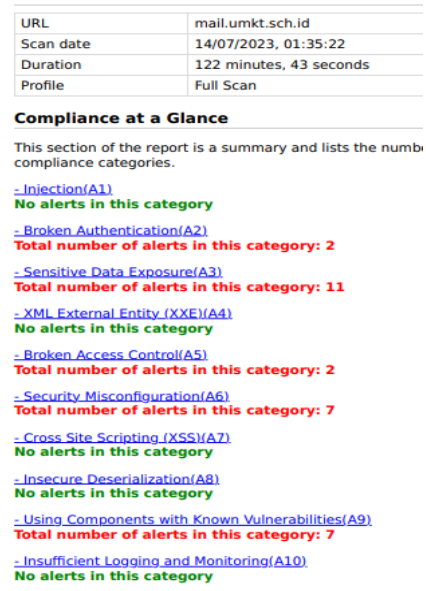
NO	Name	Vulnerablitiry Level	Number of Vulnerabilities
1	Absence of Anti-CSRF Tokens	Medium	2
2	Content Security Policy (CSP) Header Not Set	Medium	6
3	Cookie No HttpOnly Flag	Low	3
4	Cookie without SameSite Attribute	Low	4
5	Server Leaks Version Information via "Server" HTTP Response Header Field	Low	8

6	X-Content-Type-Options Header Missing	Low	4
7	GET for POST	Low	1
8	Modern Web Application	Informational	2
9	User Agent Fuzzer	Informational	36

In Table 4, This web application has several security vulnerabilities with varying levels of risk and different numbers of vulnerabilities. Some vulnerabilities, such as "Absence of Anti-CSRF Tokens" and "Content Security Policy Header Not Set," have a moderate level of risk and significant vulnerabilities. In contrast, others have low risk or are merely informational. Remedial actions are necessary to address these vulnerabilities and maintain the web application's security.

2. OWASP Top 10 Penetration Testing

The subsequent step involves penetration testing on the mail server domain mail.umkt.sch.id, following the guidelines of OWASP Top 10, utilizing the Acunetix tools for penetration testing. Acunetix Web Vulnerability Scanner is an application used for auditing the security of web applications. It is designed to simulate the actions of a hacker to discover vulnerabilities such as SQL Injection and Cross Site Scripting attacks. Acunetix detects and reports various vulnerabilities in diverse architectures like WordPress, PHP, ASP.NET, Java Frameworks, Ruby on Rails, and more. The security scan results generated by the Acunetix Web Vulnerability Scanner can be used as a report presented to website or application developers (Afrih Juhad et al., 2016). The depiction of several threats aligned with the OWASP Top 10 criteria, identified using the Acunetix tool, can be observed in Figure 5.



URL	mail.umkt.sch.id
Scan date	14/07/2023, 01:35:22
Duration	122 minutes, 43 seconds
Profile	Full Scan

Compliance at a Glance

This section of the report is a summary and lists the numb
compliance categories.

- Injection(A1)
No alerts in this category

- Broken Authentication(A2)
Total number of alerts in this category: 2

- Sensitive Data Exposure(A3)
Total number of alerts in this category: 11

- XML External Entity (XXE)(A4)
No alerts in this category

- Broken Access Control(A5)
Total number of alerts in this category: 2

- Security Misconfiguration(A6)
Total number of alerts in this category: 7

- Cross Site Scripting (XSS)(A7)
No alerts in this category

- Insecure Deserialization(A8)
No alerts in this category

- Using Components with Known Vulnerabilities(A9)
Total number of alerts in this category: 7

- Insufficient Logging and Monitoring(A10)
No alerts in this category

Figure 5. Acunetix Results Based on OWASP Top 10 2017.

Based on the penetration testing results on the domain website server mail.umkt.sch.id using Acunetix tools, the website server exhibits 11 vulnerabilities across five vulnerability categories by the OWASP Top 10 2017 criteria. The testing outcomes are presented in tabular form, as shown in Table 3.

Tabel 3. OWASP Top 10 2017 Test Results

OWASP TOP 10	Tools	Vulnerability	The total number of vulnerabilities
A1-Injection	Acunetix	Non-vulnerable	0
A2-Broken Authentication	Acunetix	Vulnerable	2
A3-Sensitive Data Exposure	Acunetix	Vulnerable	9
A4-XML External Entities (XXE)	Acunetix	Non-vulnerable	-

A5-Broken Access Control	Acunetix	Vulnerable	2
A6-Security Misconfiguration	Acunetix	Vulnerable	6
A7-Cross-Site Scripting (XSS)	Acunetix	Non-vulnerable	0
A8-Insecure Deserialization	Acunetix	Non-vulnerable	0
A9-Using Components with Known Vulnerabilities	Acunetix	Vulnerable	6
A10-Insufficient Logging & Monitoring	Acunetix	Non-vulnerable	0

Based on table 3 shows that 10 penetration testing results following the “OWASP Top 10 2017” criteria on the domain website server mail.umkt.sch.id using Acunetix tools. Vulnerabilities were identified in A3-Sensitive Data Exposure, A5-Broken Access Control, A6-Security Misconfiguration, and A9-Using Components with Known Vulnerabilities.

3. Analysis and Report of Penetration Test results

In this phase, analysis and report are conducted for the penetration testing in the form of suggestions or recommendations for improving the website to minimize vulnerabilities identified in the previous testing. Fundamentally, threats to websites that occurred in 2017 have been documented by OWASP (Open Web Application Security Project) and are documented in the OWASP Top 10 Security – 2017. The OWASP Top 10 Security outlines several threats and the level of risk associated with the impact of attacks classified by OWASP. The threat levels are assigned values calculated using a specialized calculator called CVSS (Common Vulnerability Scoring System) provided by NIST (National Institute of Standards and Technology), with a score range from 0.0 to 10.0. Vulnerability assessments are used as a reference for determining the severity of vulnerabilities experienced. Table 4 presents the analysis of testing results, the suggestions/recommendations provided, and the level of threat encountered.

Table 4. Recommendations on the results of security holes found.

Clickjacking: X-Frame-Options header missing		Solution
Affected item	Web Server	
CVSS2	Base Score: 4.3 (medium) Access Vector: Network_accessible Access Complexity: Medium Integrity Impact: Partial	Added "X-Frame-Options" header to server response.
CWE	CWE-693	
Cookie(s) without HttpOnly flag set (verified)		Solution
Affected item	Web Server	
CVSS2	Base Score: 0.0 (None) Access Vector: Network_accessible Access Complexity: Low	Sets the HttpOnly flag on cookies sent by the server
CWE	CWE-16	
Cookie(s) without Secure flag set (verified)		Solution
Affected item	Web Server	
CVSS2	Base Score: 0.0 (none) Access Vector: Network_accessible Access Complexity: Low	Sets the Secure flag on the cookies sent by the server.
CWE	CWE-16	
Documentation file (verified)		Solution
Affected item	/plugins/demo/README	
CVSS2	Base Score: 5.0 (medium) Access Vector: Network_accessible Access Complexity: Low Confidentiality Impact: Partial	Removing/limiting public access to found documentation files

CWE	CWE-538	
Login page password-guessing attack		Solution
Affected item	/src/redirect.php	
CVSS2	Base Score: 5.0 (Medium) Access Vector: Network_accessible Access Complexity: Low Confidentiality Impact: Partial	Implements a login attempt limiting mechanism, Uses the Captcha verification method to ensure that a human, not an automated bot, is attempting to login.
CVSS3	Base Score: 5.3 (medium) Attack Vector: Network Attack Complexity: Low Scope: Unchanged Availability Impact: Low	
CWE	CWE-307	
Possible sensitive files		Solution
Affected item	/plugins/test/test.php	
CVSS2	Base Score: 5.0 (medium) Access Vector: Network_accessible Access Complexity: Low Confidentiality Impact: Partial	Implement additional protection such as a firewall, IDS (Intrusion Detection System), or IPS (Intrusion Prevention System) to protect sensitive files from attack or unauthorized access.
CVSS3	Base Score: 7.5 (High) Attack Vector: Network Attack Complexity: Low Scope: Unchanged Confidentiality Impact: High	
CWE	CWE-200	
Unencrypted connection (verified)		Solution
Affected item	WebServer	
CVSS2	Base Score: 5.8 (medium) Access Vector: Network_accessible Access Complexity: Medium Confidentiality Impact: Partial	Uses secure encryption protocols such as HTTPS
CVSS3	Integrity Impact: Partial Base Score: 9.1 (critical) Attack Vector: Network Attack Complexity: Low Scope: Unchanged Confidentiality Impact: High Integrity Impact: High	
CWE	CWE-310	
Content Security Policy (CSP) not implemented		Solution
Affected item	WebServer	
CVSS2	Base Score: 0.0 (None) Access Vector: Network_accessible Access Complexity: Low	Implement and configure a CSP on a web server
CWE	CWE-16	
Content type is not specified (verified)		Solution
Affected item	/plugins/demo/README	
CVSS2	Base Score: 0.0 (None) Access Vector: Network_accessible Access Complexity: Low	Use the "Content-Type" header in the server response with the appropriate value for the type of content sent.
CWE	CWE-16	
Error page web server version disclosure		Solution
Affected item	WebServer	Remove version information from
CVSS2	Base Score: 5.0 (medium)	error messages. Use a Web

	Access Vector: Network_accessible Access Complexity: Low Confidentiality Impact: Partial Base Score: 0.0 (none)	Application Firewall (WAF) that can detect and prevent attacks related to disclosing web server versions.
CVSS3	Attack Vector: Network Attack Complexity: Low Scope: Unchanged	
CWE	CWE-200	
Password type input with auto-complete enabled		Solution
Affected item	WebServer	
CVSS2	Base Score: 0.0 (none) Access Vector: Network_accessible Access Complexity: Low	
CVSS3	Base Score: 7.5 (High) Attack Vector: Network Attack Complexity: Low Scope: Unchanged Confidentiality Impact: High	Manually disable the "auto-complete" feature on their browser
CWE	CWE-200	
Parameter	login_form	

From the table of 4 security vulnerabilities identified in this web application, there is a diverse range of risks. Some vulnerabilities, such as Clickjacking, Web Server Version Disclosure, and Login Page Password-Guessing Attacks, carry moderate to high risks. The recommended solutions encompass implementing security headers, setting cookie flags, and providing additional protection against the exposure of sensitive files.

Meanwhile, several vulnerabilities with lower impact or are merely informative, such as Cookie(s) without Secure Flag Set or Content-Type Not Specified. Despite their relatively lower risk, preventive measures are still necessary to uphold the integrity and security of the web application. It is essential to adhere to the recommended solution steps to mitigate these vulnerabilities. These steps involve adding headers, configuring flags, using secure encryption, and applying protective mechanisms like firewalls or intrusion detection systems. By implementing these solutions, it is possible to enhance the web application's security and safeguard it against various types of attacks.

4. CONCLUSION

This study identified various security vulnerabilities in the analyzed web application using tools such as OWASP ZAP and Acunetix. Critical vulnerabilities like Clickjacking, Server Version Information Disclosure, and Password Guessing Attacks on the Login Page pose high risks and require serious attention. The recommended solutions include implementing strong encryption, configuring cookie flags, and safeguarding against sensitive file exposure. These recommendations can assist developers in enhancing the security of their web applications. Despite the existence of lower-risk vulnerabilities, preventive measures remain crucial. This research provides insights into common vulnerabilities in web applications and offers practical recommendations for improving their security. However, the study has limitations in terms of the scope of the analyzed application. Therefore, future research could expand to a broader range of applications, explore new vulnerabilities, and test the effectiveness of solutions in real-world scenarios. Overall, this study guides the landscape of web application security and contributes to efforts to enhance the security of web applications in the ever-evolving digital era.

REFERENCES

Afrih Juhad, H., Isnanto, R. R., & Widianto, E. D. (2016). Analisis Keamanan pada Aplikasi Her-registrasi Online Mahasiswa Universitas Diponegoro. *Jurnal Teknologi Dan Sistem Komputer*, *4*(3). https://doi.org/10.14710/jtsiskom.4.3.2016.479-484

Agreindra Helmiawan, M., Firmansyah, E., Fadil, I., Sofivan, Y., Mahardika, F., & Guntara, A. (2020). Analysis of Web Security Using Open Web Application Security Project 10. *2020 8th International Conference on Cyber and IT Service Management, CITSM 2020*. https://doi.org/10.1109/CITSM50537.2020.9268856

Aulianisa, S. S., & Indirwan, I. (2020). Critical Review of the Urgency of Strengthening the Implementation of Cyber Security and Resilience in Indonesia. *Lex Scientia Law Review*, *4*(1), 31–45.

Belapurkar, A., Chakrabarti, A., Ponnapalli, H., Varadarajan, N., Padmanabhuni, S., & Sundarrajan, S. (2009). *Distributed systems security: issues, processes and solutions*. John Wiley & Sons.

Cazorla, L., Alcaraz, C., & Lopez, J. (2016). Cyber stealth attacks in critical information infrastructures. *IEEE Systems Journal*, *12*(2), 1778–1792.

Galperin, H., & Fernanda Viecens, M. (2017). Connected for development? Theory and evidence about the impact of internet technologies on poverty alleviation. *Development Policy Review*, *35*(3), 315–336.

Infante, A., & Mardikaningsih, R. (2022). The Potential of social media as a Means of Online Business Promotion. *Journal of Social Science Studies (JOS3)*, *2*(2), 45–49.

Innocenti, T., Mirheidari, S. A., Kharraz, A., Crispo, B., & Kirda, E. (2021). You've Got (a Reset) Mail: A Security Analysis of Email-Based Password Reset Procedures. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12756 LNCS*. https://doi.org/10.1007/978-3-030-80825-9_1

Kalaani, C. (2023). *OWASP ZAP vs Snort for SQLi Vulnerability Scanning*.

Karimov, M. M., Arzieva, J. T., & Rakhimberdiev, K. (2022). Development of approaches and schemes for proactive information protection in computer networks. *2022 International Conference on Information Science and Communications Technologies (ICISCT)*, 1–5.

Li, J. (2020). Vulnerabilities mapping based on OWASP-SANS: a survey for static application security testing (SAST). *ArXiv Preprint ArXiv:2004.03216*.

Mateo Tudela, F., Bermejo Higuera, J.-R., Bermejo Higuera, J., Sicilia Montalvo, J.-A., & Argyros, M. I. (2020). On Combining Static, Dynamic and Interactive Analysis Security Testing Tools to Improve OWASP Top Ten Security Vulnerability Detection in Web Applications. *Applied Sciences*, *10*(24), 9119.

Mburano, B., & Si, W. (2019). Evaluation of web vulnerability scanners based on OWASP benchmark. *26th International Conference on Systems Engineering, ICSEng 2018 - Proceedings*. https://doi.org/10.1109/ICSENG.2018.8638176

Raja, R., & Nagasubramani, P. C. (2018). Impact of modern technology in education. *Journal of Applied and Advanced Research*. https://doi.org/10.21839/jaar.2018.v3is1.165

Shahid, J., Hameed, M. K., Javed, I. T., Qureshi, K. N., Ali, M., & Crespi, N. (2022). A Comparative Study of Web Application Security Parameters: Current Trends and Future Directions. *Applied Sciences*, *12*(8), 4077.

Sudhodanan, A., Armando, A., Carbone, R., & Compagna, L. (2017). *Attack Patterns for Black-Box Security Testing of Multi-Party Web Applications*. https://doi.org/10.14722/ndss.2016.23286

Taneja, S., & Toombs, L. (2014). Putting a face on small businesses: Visibility, viability, and sustainability the impact of social media on small business marketing. *Academy of Marketing Studies Journal*, *18*(1), 249.

Thoma, B., Murray, H., Huang, S. Y. M., Milne, W. K., Martin, L. J., Bond, C. M., Mohindra, R., Chin, A., Yeh, C. H., Sanderson, W. B., & Chan, T. M. (2018). The impact of social media promotion with infographics and podcasts on research dissemination and readership. *Canadian Journal of Emergency Medicine*, *20*(2). https://doi.org/10.1017/cem.2017.394

We Are Social. (2022). *DIGITAL 2022: ANOTHER YEAR OF BUMPER GROWTH*. https://wearesocial.com/uk/blog/2022/01/digital-2022-another-year-of-bumper-growth-2/

Wibowo, F., Nuha, H. H., & Wibowo, S. (2020). Network Security Analysis Using HTTPS with SSL on General Election Quick Count Website. *2020 IEEE International Conference on Communication, Networks and Satellite, Comnetsat 2020 - Proceedings*. https://doi.org/10.1109/Comnetsat50391.2020.9328940