SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Analysis of Web Vulnerability Using Open-Source Scanners on Different Types of Small Entrepreneur Web Applications in Malaysia

Alya Geogiana Buja[1], Nurul Natasha Mohamad Amirul Asri Low[1], Anwar Farhan Zolkeplay[1,*], Nurul Alieyah Azam[1], Fuad Mat Isa[2]

[1] College of Computing, Informatics and Mathematics, Universiti Teknologi MARA (UiTM) Melaka Branch, Jasin Campus, 77300 Merlimau, Melaka, Malaysia
[2] Operation Team, Tabsquare, Jalan Dua, Chan Sow Lin, 55200 Kuala Lumpur, Wilayah Persekutuan Kuala Lumpur, Malaysia

ARTICLE INFO

Article history:
Received 22 June 2023
Received in revised form 7 September 2023
Accepted 12 November 2023
Available online 19 February 2024

Keywords:
Cyber security; Open-Source scanners; Small entrepreneur; Web vulnerability

ABSTRACT

Most of Malaysia's small entrepreneurs have switched to online platforms as an alternative to their physical businesses. Social media sites such as TikTok, Instagram, Twitter, and Facebook provide free advertising tools and convenient access to a broader global target. However, security issues on these websites remain questionable as users are exposed to web attacks due to the vulnerabilities on the websites. Considering the cost and lack of awareness of the importance of cybersecurity, some organizations find it not profitable to invest in securing their websites. Therefore, this paper aims to test Malaysian small entrepreneurs' web applications using open-source scanners and analyse the results of web vulnerabilities detected. To do so, two types of open-source scanners, OWASP ZAP and IRONWASP, were installed to scan five websites found through advertisements on social media sites. The web vulnerability identification was based on the top 5 OWASP web vulnerability reports, and the results showed that five types of web vulnerabilities were detected. The analysis of the results showed that the top 5 web vulnerabilities in Malaysian small entrepreneurs' websites are the 'Missing Session Timeout' vulnerability with 81.84 percent, the 'Sensitive Information Passed as Clear Text in GET URL' vulnerability with 14.74 percent, and the 'Session ID Cookies not Marked Secure' vulnerability with 2.47 percent. This paper provides security analysis on Small Medium Enterprise (SME) websites for future enhancement and consideration during development and implementation to avoid possible attacks. Therefore, developers are advised to handle these vulnerabilities by carefully managing the session timeout, and users are recommended to log out from the websites immediately after they are done.

1. Introduction

In curbing the spread of the COVID-19 pandemic, the Malaysian government imposed a Movement Control Order (MCO), which subsequently brought an unprecedented slump in numerous sectors, including economic activities [1]. Regarding this situation, the most affected party in the

* *Corresponding author.*
E-mail address: anwarfarhan@uitm.edu.my

https://doi.org/10.37934/araset.40.1.174188

economic sector was the small and medium enterprises (also known as SMEs) [2]. For instance, during the MCO, farmers in Cameron Highlands, Pahang, faced difficulties marketing their products due to logistics and transportation limitations; thus, they had to discard all their products due to storage limitations. However, a solution is devised to counter such issues by using e-commerce platforms [3]. Eventually, using online means for SME operators can reduce the costs of advertisements and marketing targeting customer data and build a broader customer base [4].

Meanwhile, the benefits of online means for consumers include lower prices and broader product varieties compared to traditional store prices and providing convenience and safety, especially during COVID-19 [4]. A study by [3] shows that 62% of respondents in Malaysia preferred online purchases during the pandemic to reduce time spent outside and practice social distancing. This proves that customers' preference for online businesses is mainly due to their time and cost-saving benefits. Moreover, during the COVID-19 pandemic, countless entrepreneurs switched the nature of their businesses from traditional to online businesses to minimize physical contact.

Nonetheless, [5] found that 68% of small entrepreneurs need a systematic approach to ensuring web application security [5]. Even though web applications are mainly created for business purposes, some were not tested for vulnerabilities, thus causing numerous security threats for their users [6]. In addition, according to [7], 84% of SMEs in Malaysia experienced cyber-attacks in 2019. The reason is that most SMEs have a limited amount of labour that can manage the security of their businesses' web applications; plus, some SME operators assume that their online business operations will not be the target for cyber-attackers due to their small profits. The reality is that they are still exposed to various vulnerabilities, such as SQL injection and Cross-site Scripting (XSS), which allow attackers to access and exploit the user's information by manipulating their web applications.

Additionally, web application users are exposed to cyber-attacks, and attackers can exploit their details in the web application. This problem may lead to a more severe situation when attackers gain access to a tiny website, providing them more access to many other web applications in a single attack. Eventually, business operators may face losses due to insufficient knowledge, information about web vulnerabilities, and suitable testing tools to test web vulnerabilities.

Therefore, analysing the web vulnerability of SMEs' web applications is essential to detect the types of web vulnerability. Next, this analysis is vital to ensure customers can safely purchase online and secure their personal information in web applications. Besides, this analysis is also essential to discover suitable open-source tools to detect web vulnerabilities in web applications. Subsequently, the information and data in online businesses can be secured, thus ensuring a safe business environment for entrepreneurs and consumers.

1.1 Problem Statement

During the COVID-19 outbreak, the number of online businesses in Malaysia increased mainly due to various restrictions during the MCO. In addition, during the MCO, many non-essential physical business operations were closed due to the government's rules, affecting their incomes. Alternatively, consumers and entrepreneurs shifted from physical business transactions to online platforms, especially on social media sites like Instagram and Facebook. Moreover, the increasing preference for online shopping is mainly due to health safety concerns and convenience, as consumers and entrepreneurs can conduct business transactions from home.

With this, every online entrepreneur aims to provide the best services to gain customers' trust, primarily by increasing online sales and discounts. Nonetheless, there are several concerns related to online shopping, such as goods quality and security issues, as cyber criminals who would trap any customer from getting the deals [8]. Besides, unpatched software holes help hackers gain

unauthorized access to someone's computer and break the network. Personal details entered in online shopping are very valuable, and a hacker can use the details to attack the site used by the customer and commit identity theft to pretend to be a customer and then manipulate the financial account [8]. If a site asks for more personal information than name, address, and credit card number, such as social security number or bank account number, the person may be a victim of fraud [9]. An online retailer's most common security threat is credit card fraud, which occurs when an attacker gets to steal a customer's personal and payment information and then sells it on black markets [10].

According to [7] 2018, 80% of SMEs decided not to invest in cybersecurity due to cost, and some needed to be made aware of cybersecurity concerns. In addition, more than half of SME organizations in Malaysia have experienced a security breach. Still, only 47% of them have taken the initiative to investigate or data breach assessment to improve their web applications [6]. The impact that SMEs probably face in a significant cyber incident is in terms of their relationship with customers with a 60% probability, affected company profits by 59%, and affected reputation in the market by 58% [11]. The standard breached data files are customers' records that contain payment information and personal details, research and development (R&D) data, Intellectual Property (IP) data, and financial performance data, and these data are beneficial to attackers [11]. To date, a web vulnerability scanner that can find all the OWASP Top 5 vulnerabilities using a single tool still does not exist.

Solving these problems can help SMEs enhance their web applications' security and protect their customers' information and data. This will also minimize the concerns about online purchases by providing more knowledge on the types of web vulnerabilities and suitable tools to scan web vulnerabilities. Subsequently, this may also enhance SMEs' income and national income.

Based on studies conducted on the open-source web vulnerability scanner, past researchers have attempted to discover the difference between paid and open-source tools based on detection accuracy and Top 5 web vulnerabilities. Ultimately, these researchers found open-source tools more accurate and cost-effective, suitable for small entrepreneurs' web applications with a small budget [12]. Furthermore, [13] conducted a study to compare the commercial and open-source scanners' performance and found that both scanners have equal efficiency in detecting some vulnerabilities.

In the present study, the analysis will apply open-source tools to test small entrepreneurs' web applications and detect vulnerabilities based on OWASP's Top 5 web vulnerabilities. The list of small entrepreneurs' web applications will be collected on social media and test the web application using selected open-source tools, analyse web vulnerabilities, and find suitable tools to scan.

1.2 Research Objectives

i. To identify web vulnerabilities in Malaysian small entrepreneurs' websites using open-source scanners, namely OWASP ZAP and IronWASP.
ii. To analyse the result of web vulnerabilities detected in Malaysian small entrepreneurs' websites based on OWASP ZAP and IronWASP.

1.3 Literature Review

1.3.1 Physical business to online business

A study by [14] found that small entrepreneurs switched to online businesses due to countless benefits and opportunities in online platforms, such as affordability and accessibility, making reaching out and communicating with their customers convenient. Furthermore, an online business also saves cost and time, allowing small entrepreneurs with a small budget to do business at home instead of buying or renting a place to run their business. Besides, during the COVID-19 pandemic,

many small physical entrepreneurs faced a massive drop in sales that affected their income; thus, they either had to restart their businesses or switch from physical to online businesses [14].

1.3.2 Web vulnerability

Five common types of web vulnerabilities that attackers illegally use to perform cyber-attacks are SQL injection, Cross-Site Scripting (XSS), command injection, file inclusion (LFI/RFI), and Cross-Site Request Forgery (CSRF). However, many other vulnerabilities can also be exploited for web application exploitation [15]. In 2019, Symantec came out with the Internet Threat Report 2019, reporting that web application attack was growing by 56% in 2018, and the average of attacks detected per month is 30 to 40 million [16]. A web vulnerability is an unintended flaw or weakness in a web application that an attacker can use to perform a cyber-attack. Three aspects of web vulnerability are application weakness, the attacker's unauthorized access to application defects, and the attacker's capability to exploit the flaw [17].

1.3.2.1 Cross-Site (XSS) scripting

Cross-Site Scripting (XSS) is a code injection vulnerability that allows malicious users to send malicious scripts to web browsers. This happens when a web application uses user information without proper validation in response pages; the browser will execute the malicious script from the application when a user enters the infected web page. A research paper by [18] used static analysis to scan XSS vulnerabilities by combining static taint that can detect XSS vulnerability and string analysis. Based on their policy, practical checking algorithms were provided. The first phase is the analysis of output statements, followed by analysing control structure and construction context-free grammar (CFG) [18]. The research on XSS used a static analysis approach to identify the vulnerability in the coding phase or software development life cycle (SDLC). Static analysis is used to extract any valid or invalid input condition, and it reviews the source code automatically; the analysis finds the fundamental cause of security problems and many errors in early development.

1.3.2.2 Missing session timeout

A missing session timeout is an inactive timeout not configured correctly and can help attackers gain unauthorized access to web applications. According to [19], this web vulnerability has affected educational institutions the most, with 45.5% compromising their admin access. This is followed by e-commerce, which is affected by 4.5%, and medical institutions, online portals, and government websites with 13.6%, 9.1%, and 27.3%, respectively [19]. There are two ways to detect a missing session timeout: first, check on the web deployment descriptor file "web.xml" and second, the session object in the program. The steps to test this vulnerability manually are to log into the web application and minimize the browser, stop all interaction, and then open the browser and interact with the application to check whether the session is still valid [20].

1.3.2.3 Sensitive information passed as clear text in get URL

A query string can be recorded in the browser's history, passed through Referrers to other websites, saved in weblogs, or recorded in other sources. If there is any sensitive information, such as session identifiers in the query string, the attackers can launch attacks using this information [21]. This vulnerability can be detected using black box testing because this method can be used when the

source code is unavailable [21]. The study on consumer IoT medical devices by [22] found that sensitive information passed as clear text in GET URL vulnerability in Withing Blood Pressure Monitor, and the researchers found a stock photo of a person using the device after they tried using GET request. In addition, sensitive information passed as clear text in GET URL vulnerability was found in six different Industrial Control Systems (ICS). This type of vulnerability can cause an unauthorized user to sniff sensitive data in the web application because all the data and information in cleartext are easily read by an attacker [23].

1.3.2.4 Session ID cookies not marked secure

A cookie is a bit of data in a web application that will be sent to the web browser, and the browser may store and send the data back with an HTTP request to the same web application; the cookies in the server-side web application are used to identify the user, the state, and preferences. In addition, cookies may contain sensitive information, such as user profiles, user privileges, cached data from the back-end store, browsing history, page flow state, and CSRF prevention tokens [24]. If session ID cookies for web applications are not marked as secure, the browser may send them over an unencrypted HTTP request, and the attackers can access and view cookies in clear text. To detect the vulnerability, the researcher logs into a webpage and finds cookie leakage, then performs a test; if it is successful, the webpage is insecure. Session ID cookies not marked secure can help an attacker discover the cookie value, and the attacker can make Cross-site scripting (XSS) attacks by inserting HTML and JavaScript into the page and exploiting the victim's session. Additionally, an attacker can use traffic sniffing to read the values of the cookies when a web application does not use HTTPS [24].

1.3.2.5 SQL injections

SQL Injection is a type of security exploit that executes SQL queries without properly validating user inputs and altering or accessing data. As a result, malicious users put some information, and an SQL explanation will be constructed using the built data [25]. The SQL Injection attack in December 2011 affected almo160,00sites using SQL Server framework, Microsoft's Internet Information Services (IIS), and ASP.NET [25]. An attacker exploits expression parts like the WHERE clause to control the data requested and update the database. To detect SQL injection, the research started with data set extraction, test pre-processing that involves R Scripting and regular expression pattern, feature hashing, filter-based feature choice for top relevant vectors, split between training and testing data, and train prediction model [26]. In detecting SQL injection, this present research used a static analysis tool, SAFELI. It is an automated tool to test SQL injection vulnerabilities on web applications, and SAFELI can discover source code more delicate than black-box vulnerability scanners; the components in SAFELI are MSIIL Instrumentor, a symbolic execution engine, and a library of attack patterns: constraint solver and test case generator.

1.3.3 Open-source web vulnerability scanner

A web vulnerability scanner is a tool capable of automatically scanning potential vulnerabilities in web applications. For example, the most known web vulnerabilities are SQL injection and Cross-Site Scripting (XSS) [27]. A web vulnerability scanner is widely used to find web application vulnerabilities, which many types of tools available can execute; however, all web vulnerabilities cannot be detected by only a single web vulnerability scanner as every scanner acts differently. Open-source and commercial scanners can record the false-positives rate of vulnerabilities, but open-

source tools have higher rates [13]. Nevertheless, several open-source scanners are effectively functional, like some commercial scanners; with Acunetix offering the best performance, placed first on top with scores of 81%, followed by Appscan in second place with a 65% score, while third and fourth place best performance are taken by open-source scanners, Skipfish and ZAP with scores of 43% and 40% respectively [13]. Skipfish and ZAP are efficient open-source scanners that detect vulnerabilities such as command execution, XSS, and SQL injection [13].

1.3.3.1 OWASP ZAP

OWASP ZAP is a standard open-source tool that performs better than other scanning tools with a user-friendly interface and is used for penetration testing; hence, this tool is usable by anyone with different abilities in security software [13]. In addition, OWASP ZAP is capable of critical scanning vulnerabilities such as SQL injection, Cross-Site Scripting, remote OS command, Path Traversal, External Redirect, and Remote File Inclusion [28]. Furthermore, OWASP ZAP can also detect command execution vulnerabilities due to high results for true and false positives, resulting in the researcher concluding that OWASP ZAP performs better than other vulnerability scanners [12]. Moreover, based on other research, they discovered that OWASP ZAP defeats the commercial tools by being the highest in detecting RFI vulnerability; therefore, it is recommended for web security managers and web developers [29].

1.3.3.2 IronWASP

Iron Web Application Advanced Security Testing Platform, also known as IronWASP, was created by Lavakumar Kuppan; it was scripted for Python and Ruby 31 to give full access to the IronWASP framework. However, IronWASP does not support authentication; hence, this tool cannot detect vulnerabilities accessible after authentication [12]. In addition, IronWASP is an advanced open-source tool with many external libraries, such as JSON, .NET, IronPython, and IronRuby [13]. Although IronWASP offers plugins compatible with Ruby and Python, this scanner tool can detect all cleartext credential vulnerabilities, session token in URL vulnerability, password auto-enabled, and missing anti-Cross-Site Request Forgery token vulnerability [12]. Several advantages of using IronWASP include recording the login sequence, an effective scan engine enabling most common vulnerability detection, easy customization for new vulnerability testing, and generating vulnerability reports.

2. Methodology

2.1 Information Gathering Phase

The first phase of this research is the information-gathering phase, which involves collecting information regarding building the research. The data and information gathered were obtained from resources such as books, journal articles, websites, etc., and were utilized to identify the paper's research objectives and requirements. The gathered information was mainly on the types of vulnerabilities based on OWASP Top 5 web vulnerabilities and open-source web vulnerabilities scanner and study previous and related research work.

2.2 Identifying the List of Websites & Tools / Scanners Used

Social media platforms such as Instagram, Twitter, and Facebook were used to list the small entrepreneurs' web applications for the present study's analysis. Since many people in this era are

constantly using social media in their daily lives, countless SME operators in Malaysia are using social media as an alternative for promoting their businesses by attaching web application links in their advertisements. In addition, advertising on social media is considered more effective in promoting any business than traditional advertising. The advertisement on social media will appear in the story and feed based on the social media algorithm, which commonly contains descriptions of products, photos, and a link to the website. Users can freely view the website by clicking on the link, purchasing an order, and making product payments. Therefore, a vulnerability scanner ensures the security of the user's privacy. However, attackers could exploit the website link once vulnerabilities are detected, and the user's data and information could be taken advantage of.

2.2.1 List of websites

Many SMEs' web applications can be obtained from social media advertisements where online payment options are provided via the web application, which generally contain information and data about the seller and customer. Advertisements on social media mainly consist of web applications from different types of businesses, such as games, merchandise, stationaries, food, clothes, beauty products, and many more. Therefore, in the present study's analysis, only five web applications were tested for web vulnerabilities due to the time consumption of the scanning process. In this analysis, the names of the websites were replaced with WA, WB, WC, WD, and WE to not harm the websites.

First, WA is a Contemporary Artisan Matcha brand that focuses on premium-quality Matcha, showcased with its aesthetic touch of modern minimalist concept, which describes the state-of-the-art lifestyle while preserving the vital traditional elements of Matcha nature. Second, WB is a global e-sports brand that strives to bring together the gaming community to rethink how they have been merchandising in the past couple of years. They make every garment to order directly to and for our customers: no stock, no sales, and no unsold clothing ending up in landfills. Third, WC is a cosmetic brand introduces the latest skincare and makeup trends through aggressive, innovative products. In addition, the brand focuses on zero weight, zero dimension, and zero skill. Fourth, WD is a women's clothing store that offers a wide range of apparel to women with a unique sense of style. Five, WE are an online store selling musical instruments, professional audio systems, and visual and lighting equipment.

2.2.2 List of tools or scanners

The open-source scanner is the most suitable tool for this analysis as it only incurs a minimal budget for security. Hence, in assisting SMEs in Malaysia with web application security, open-source is the best solution to scan web vulnerabilities in web applications. An open-source scanner is cost-effective and performs well in detecting web vulnerabilities like a commercial scanner. Hence, the two scanners chosen for the current analysis were OWASP ZAP and IronWASP. These two scanners have an excellent performance in detecting web vulnerabilities among open-source scanners; moreover, these scanners could detect many OWASP web vulnerabilities. Mostly, the researchers used these two scanners due to their performance and capabilities to scan web vulnerabilities in web applications.

2.3 Experiments

After scanning the web applications, the following process involved in the present analysis was to retrieve a report from the scanners. In many formats, such as HTML, txt, and XML, the report was

then used to analyse web vulnerabilities and suitable open-source tools to scan—the OWASP ZAP scanner reports in HTML format. The report's information on vulnerability severity is classified into High, Medium, and Low. The figure also shows the type of vulnerabilities detected with the count of vulnerabilities occurring in the form of a bar chart—meanwhile, the IronWASP scanner reports in HTML format. In the report, the findings are separated based on the type of vulnerabilities, and their severity level is classified based on High, Medium, and Low. Meanwhile, the additional information is classified as Info and Test Leads.

2.4 Result and Discussion

The next step was identifying the web vulnerabilities in Malaysian SMEs' web applications. This process identified the web vulnerabilities based on the Top 5 OWASP vulnerabilities, as shown in Table 1. The report sorted the web vulnerabilities based on the Top 5 OWASP web vulnerabilities to identify the types of web vulnerabilities in the SMEs' web applications.

Table 1
OWASP Top 5 Web Vulnerabilities

No.	OWASP TOP 5 WEB VULNERABILITIES
1	Cross-Site (XSS) Scripting
2	Missing Session Timeout
3	Sensitive Information Passed as Clear Text in Get URL
4	Session ID Cookies Not Marked Secure
5	SQL Injections

Following this, the next step was analysing the result of web vulnerabilities in the SMEs' web applications. This step analysed the result based on the total web vulnerabilities in all five websites tested to see the common web vulnerabilities threatening SMEs' web applications. Next, the total vulnerabilities in each website tested were analysed to identify and determine the most threatened websites.

3. Analysis

3.1 Vulnerability Analysis Common Vulnerability

3.1.1 Cross-Site scripting (XSS)

Figure 3 shows that the *Cross-Site Scripting (XSS)* vulnerability (V1) is detected in Malaysian SMEs' websites. The website with the highest number of Cross-site Scripting (XSS) vulnerabilities was WE, with 43 vulnerabilities; WC was in second place, with 23 vulnerabilities, followed by WA, with 14 vulnerabilities. Meanwhile, WB and WD, each with web vulnerabilities of 14 and 5, respectively, were found to have the lowest value of web vulnerabilities. Thus, to sum up, the total number of Cross-Site Scripting (XSS) vulnerabilities detected in all five websites was 95.

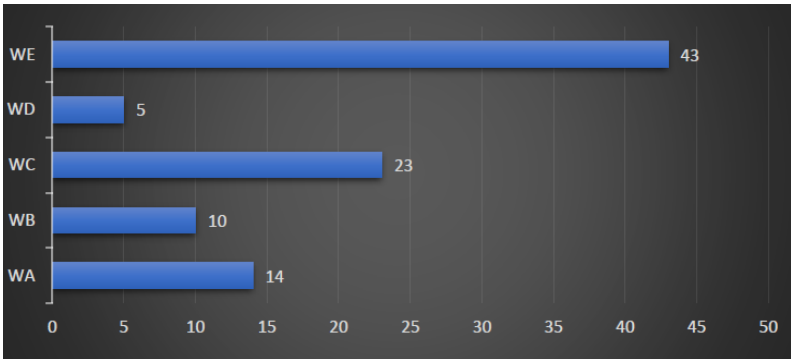

Fig. 3. Cross-Site Scripting (XSS) Vulnerability Result

3.2.2 Missing session timeout

As shown in Figure 4, the *Missing Session Timeout* vulnerability (V2) was also detected in all five websites tested, with 10,156 vulnerabilities. The highest value of vulnerabilities was seen in WC, with 3,438 vulnerabilities, followed by WD, with a value of 3,332. Meanwhile, WE and WA were ranked third and fourth with 1,358 and 1,154 vulnerabilities, respectively. Lastly, WB recorded the lowest value of web vulnerabilities, with 874 vulnerabilities detected.

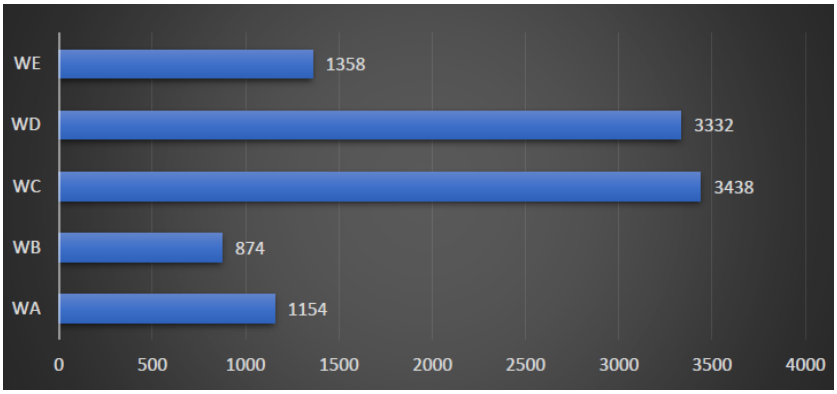

Fig. 4. Missing Session Timeout Vulnerability Result

3.2.3 Sensitive information passed as clear text in GET URL

As shown in Figure 5, the *Sensitive Information Passed as Clear Text in GET URL* vulnerability reported 1,837 vulnerabilities detected from all five websites tested. The highest vulnerability value among all five websites was discovered in WB, with 549 vulnerabilities, followed by WA, with 494 vulnerabilities. The third rank was held by WE, with 399 vulnerabilities, and WC, with 276 vulnerabilities, ranked fourth. Finally, the lowest value of vulnerabilities was recorded in WD, with 119 vulnerabilities.

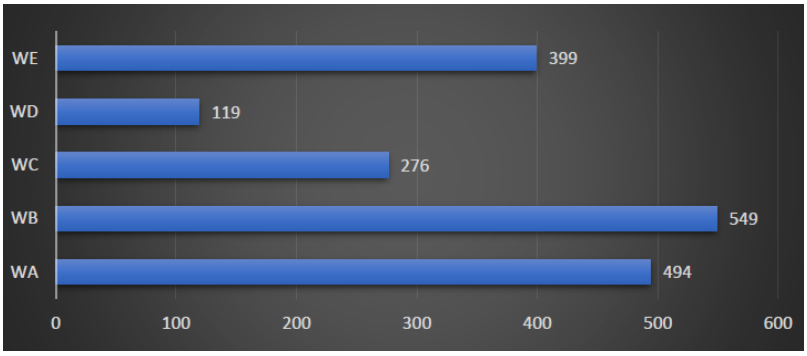

Fig. 5. Sensitive Information Passed as Clear Text in GET URL Vulnerability Result

3.2.4 Session ID cookies not marked secure

Figure 6 depicts that all five tested websites were proven to contain the *Session ID Cookies not Marked* Secure (V4) vulnerability, with 308 vulnerabilities. WC recorded the highest vulnerabilities detected, with a total of 174 vulnerabilities. Next in order was WA with 98 vulnerabilities, followed by WB with 26 vulnerabilities. Lastly, WE and WD reported the lowest vulnerabilities among the five websites, with the values of 9 and 1, respectively.

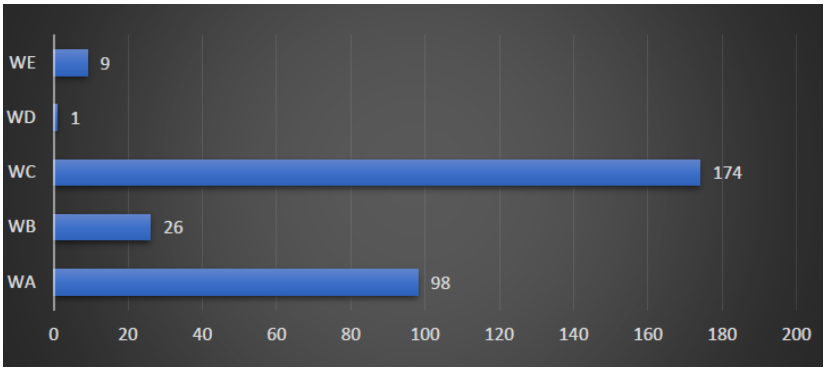

Fig. 6. Session ID Cookies not Marked Secure Vulnerability Result

3.2.5 SQL injections

According to Figure 7, the *SQL injection* vulnerability (V5) was only detected in three out of five tested websites: WA, WC, and WE, with vulnerabilities 69. WA and WC shared the similar highest number of SQL injection vulnerabilities of 34 vulnerabilities, whereas WE reported the lowest value of vulnerabilities, with only one vulnerability detected.

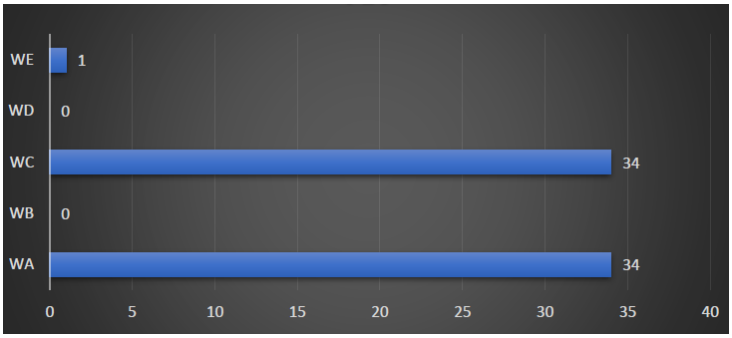

Fig. 7. SQL Injections Vulnerability Result

3.3 Total Web Vulnerabilities in the Malaysian SMEs' Websites

Figure 8 shows that five of 20 web vulnerabilities were detected in the Malaysian SMEs' websites. Among the five web vulnerabilities, the top three are the *Missing Session Timeout vulnerability* by, 81.48 percent, followed by the *Sensitive Information Passed as Clear Text in GET URL* vulnerability by 14.74 percent, and the *Session ID Cookies not Marked Secure* vulnerability by 2.47 percent.

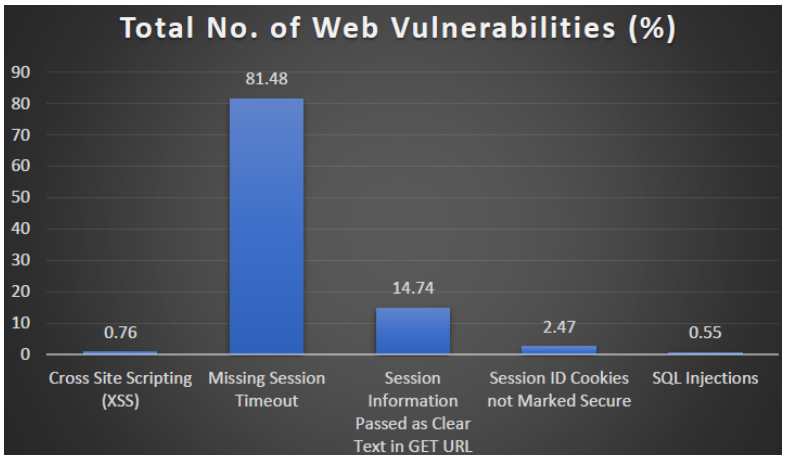

Fig. 8. Total No. of Web Vulnerabilities (in percentage)

Cybersecurity Malaysia reported roughly 446 identity cases in Malaysia in 2018, which had risen about 20 percent compared to 2017, with many *Missing Session Timeout* vulnerability cases [29]. In relation to the present study, according to Figure 8, the *Missing Session Timeout* was discovered to have the highest number of web vulnerabilities detected in Malaysian SMEs' websites, hence proving that this type of vulnerability is commonly occurring.

Next in order is the *Sensitive Information Passed as Clear Text in GET URL* vulnerability, also known as *Sensitive Information Exposure*, the second highest vulnerability detected in Malaysian SMEs' websites by 14.74 percent. The severity of damages to both the victims and the websites makes this type of web vulnerability a typical cyber threat to websites, businesses, customers, and visitors. Additionally, this vulnerability would occur on websites without HTTPS and SSL, specifically on websites that store information using weak cryptographic algorithms, or the database may be compromised by SQL Injection or other attacks [31]. Next, the *Session ID Cookies Not Marked Secure* vulnerability ranked as the third highest vulnerability detected in the Malaysian SMEs' websites at 2.47 percent, followed by the *Cross-site Scripting (XSS)* vulnerability by 0.76 percent, and the SQL *injection* vulnerability, by 0.55 percent.

In 2018, HackerOne reported that Cross-site Scripting (XSS) vulnerability was a vulnerability commonly found in all types of websites, and 40 percent of all applications tested recorded by Veracode detected the existence of Cross-site Scripting (XSS) vulnerability [32]. However, according to a study by [28], the *Cross-site Scripting (XSS)* vulnerability is undetected in the government websites in Malaysia, which also shows that this type of vulnerability in the Malaysian SMEs' websites was low.

3.4 Most Threatened Malaysian SMEs' Website by Web Vulnerability

As shown in Figure 9, web vulnerabilities were detected in each of the five tested websites based on the Top 5 OWASP web vulnerabilities. In addition, the total vulnerabilities from each website were retrieved to identify the websites with the most and the least amount of web vulnerabilities.

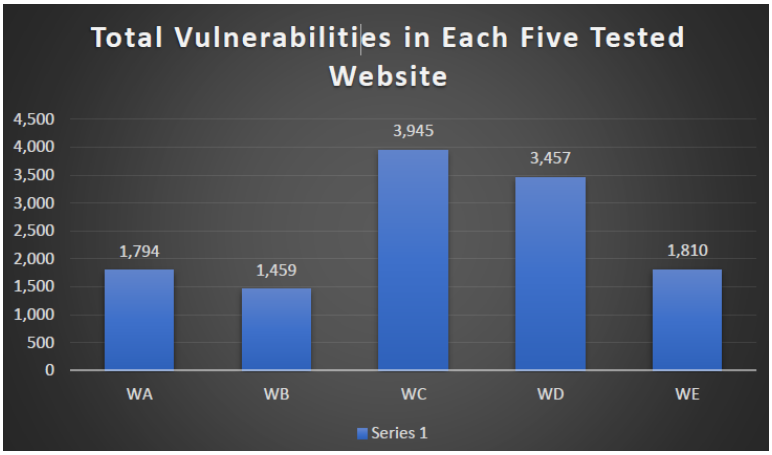

Fig. 9. Total Vulnerabilities in Each Five Tested Websites

According to Figure 9, the most threatened website was WC, with 3,945 web vulnerabilities detected, implying that WC is the easiest target compared to the other four websites. Meanwhile, WD was ranked second with a total amount of vulnerabilities value of 3,457, threatened by four types of web vulnerabilities, in which the highest type of web vulnerability in this website was the *Missing Session Timeout* vulnerability. WE were next in the third rank, with 1,810 web vulnerabilities and five types of vulnerabilities threatening it. WA follows this in the fifth rank with 1,794 vulnerabilities and five types of vulnerabilities detected. Finally, WB was the least threatened website, with 1,459 vulnerabilities and four web vulnerabilities.

Based on Acunetix research, web vulnerabilities are detected in almost 84 percent of websites worldwide, which can happen anytime. Even though Google commonly blocks websites threatened by web vulnerabilities to protect users' safety, at least 10,000 suspicious websites have been quarantined daily. Hence, according to the figure above, WC, WD, and WE were shown to have higher chances of getting attacked by the attackers.

4. Discussion

Five Malaysian SMEs' websites were tested in this analysis—these websites were found in social media advertisements such as Instagram and Facebook. Social media platforms are becoming the primary preference among SME operators as they can advertise products or businesses for free and conveniently. In doing so, entrepreneurs commonly attach their website's URL in their advertisements, where customers can click to purchase or ask any question regarding the advertised product or service.

Table 2 shows the total web vulnerabilities detected in the SMEs' websites in Malaysia, which were chosen and underwent the scanning test using two open-source scanners. Each website's total web vulnerabilities were evaluated based on the highest number of vulnerabilities detected using the OWASP ZAP and IronWASP scanners. As shown in Table 2, it can be seen that the OWASP ZAP scanner detected the highest web vulnerabilities as compared to the IronWASP scanner.

Table 2
Total Vulnerabilities in Each of the Five Tested Websites

Vulnerabilities / Website	V1	V2	V3	V4	V5	TOTAL
WA	14	1,154	494	98	34	1,794
WB	10	874	549	26		1,459
WC	23	3,438	276	174	34	3,945
WD	5	3,332	119	1		3,457
WE	43	1,358	399	9	1	1,810
TOTAL	95	10,156	1,837	308	69	**12,465**

	OWASP ZAP
	IronWASP
	Both

Out of the five vulnerabilities for each of the five SMEs' websites, five types of web vulnerabilities were detected using both scanners, and each of the five vulnerabilities was higher than the others. The types of web vulnerabilities detected were Cross-Site Scripting (V1), Missing Session Timeout (V2), Sensitive Information Passed as Clear Text in GET URL (V3), Session ID Cookies not Marked Secure (V4), and lastly, SQL Injections (V5).

The highest amount of web vulnerabilities detected was the *Missing Session Timeout vulnerability* (V2), with a total number of 10,156 vulnerabilities, followed by the *Sensitive Information Passed as Clear Text in GET URL* vulnerability (V3), with a total number of 1,837 vulnerabilities; which was also detected from all five websites. Meanwhile, the *Session ID Cookies not Marked Secure* vulnerability (V4) was ranked third, with a total of 308 vulnerabilities that were also found in all five websites. In contrast, both the *Cross-Site Scripting (XSS)* vulnerability (V1) and *SQL Injections* (V5) vulnerability were ranked in the fourth and fifth places with a total number of 95 and 69 vulnerabilities, respectively. However, unlike the other five vulnerabilities, *SQL Injections* could only be detected in three websites: WA, WC, and WE.

As shown in Table 2 above, WC was considered the most threatened website, with 3,945 web vulnerabilities. The second most threatened website was WD, with a total of 3,457 web vulnerabilities, followed by WE, with a total of 1,810 web vulnerabilities. Meanwhile, WA had a total 1,794, and WB recorded the lowest value of 1,459 web vulnerabilities detected.

5. Conclusion

In conclusion, web vulnerabilities detected in Malaysian SMEs' websites must be promptly handled to protect users and business operators. At the same time, SMEs or customers should be aware of web vulnerabilities and protect their confidential information, such as credit card details, addresses, and passwords. Suppose an attacker attacks a small website and sensitive information is stolen. In that case, the attacker can take advantage of the information by making profits and using it to attack other prominent websites. This paper provides security analysis, which benefits small entrepreneurs and customers by raising awareness of web security among small entrepreneurs, enabling them to protect their businesses from cyber-attacks. Customers can safely make online purchases with protected personal information and purchase details.

For future research, it is recommended that researchers carry out related studies on other types of websites containing private information, such as banking and educational institute websites, to improve the websites' web security. This is because these websites tend to be targeted by attackers, as confidential information such as credit card information, passwords, identification card numbers,

or addresses contained in the websites are considered highly valuable and profitable and should not fall into the wrong hands.

Acknowledgment
Our sincere appreciation goes to Kementerian Pengajian Tinggi Malaysia (KPT), Fundamental Research Grant Scheme (FRGS/1/2021/ICT07/UITM/02/1), RMC, and Universiti Teknologi MARA (UiTM) for the support given to this research endeavour.

References

[1] Kidam, Kamarizan, Siti Aishah Rashid, Jafri Mohd Rohani, Hafizah Mahmud, Hamidah Kamarden, Fateha Abdul Razak, Nurul Nasuha Mohd Nor, and Nur Kamilah Abdul Jalil. 2022. "Development of Instrument to Measure the Impact of COVID-19 and Movement Control Order to Safety and Health Competent Person and Training Provider." *Journal of Advanced Research in Technology and Innovation Management* 2 (1): 22–28. https://akademiabaru.com/submit/index.php/jartim/article/view/4449/3309.

[2] Hanafi, Wan Noordiana Wan, Wan Mohammad Taufik Wan Abdullah, Siti Norhidayah Toolib, Salina Daud, and Nurul Nadiah Ahmad. "Effects of COVID-19 Pandemic on SMEs' Business Activities: A Descriptive Analysis." *Global Business & Management Research* 13 (2021).

[3] Mustafa, Firuza Begham. "The impact of COVID-19 on agriculture in Malaysia: Insights from mixed methods." In *COVID-19, Business, and Economy in Malaysia*, pp. 24-35. Routledge, 2021. https://doi.org/10.4324/9781003182740-2

[4] Ahmad, Syed Zamberi, Norita Ahmad, and Abdul Rahim Abu Bakar. "Reflections of entrepreneurs of small and medium-sized enterprises concerning the adoption of social media and its impact on performance outcomes: Evidence from the UAE." *Telematics and Informatics* 35, no. 1 (2018): 6-17. https://doi.org/10.1016/j.tele.2017.09.006

[5] Belás, Jaroslav, Mária Mišanková, Jaroslav Schönfeld, and Beáta Gavurová. "Credit risk management: financial safety and sustainability aspects." *Journal of Security and Sustainability Issues* (2017). https://doi.org/10.9770/jssi.2017.7.1(7)

[6] C. Nabe, "Impact of COVID-19 on Cybersecurity," Deloitte Switzerland, 2020. https://www2.deloitte.com/ch/en/pages/risk/articles/impact-covid-cybersecurity.html

[7] Mutalib, Megat Muazzam Abdul, Zuraini Zainol, and Mohd Hazali Mohamed Halip. "Mitigating Malware Threats at Small Medium Enterprise (SME) Organisation: A Review and Framework." In *2021 6th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, vol. 6, pp. 1-6. IEEE, 2021. https://doi.org/10.1109/ICRAIE52900.2021.9703991

[8] Lagazio, Monica, Nazneen Sherif, and Mike Cushman. "A multi-level approach to understanding the impact of cyber crime on the financial sector." *Computers & Security* 45 (2014): 58-74. https://doi.org/10.1016/j.cose.2014.05.006

[9] Okenyi, Peter O., and Thomas J. Owens. "On the anatomy of human hacking." *Information Systems Security* 16, no. 6 (2007): 302-314. https://doi.org/10.1080/10658980701747237

[10] A. Menzheres, "Recent E-commerce Security Issues and Best Practices (2018)," Eteam.io, 2018. https://www.eteam.io/blog/e-commerce-security-issues

[11] OECD, "Coronavirus (COVID-19): SME policy responses," OECD, Jul. 15, 2020. https://www.oecd.org/coronavirus/policy-responses/coronavirus-covid-19-sme-policy-responses-04440101/

[12] McQuade, Kinnaird. "Open source web vulnerability scanners: the cost effective choice?." In *Proceedings of the Conference for Information Systems Applied Research*, vol. 2167, p. 1508. 2014.

[13] Amankwah, Richard, Jinfu Chen, Patrick Kwaku Kudjo, and Dave Towey. "An empirical comparison of commercial and open-source web vulnerability scanners." *Software: Practice and Experience* 50, no. 9 (2020): 1842-1857. https://doi.org/10.1002/spe.2870

[14] Prassl, Jeremias. *Humans as a service: The promise and perils of work in the gig economy*. Oxford University Press, 2018. https://doi.org/10.1093/oso/9780198797012.001.0001

[15] Kirstens, "Cross-Site Scripting (XSS) | OWASP," Owasp.org, 2020. https://owasp.org/www-community/attacks/xss/

[16] Symantec, "2019 Internet Security Threat Report," Feb. 2019. https://docs.broadcom.com/doc/istr-24-executive-summary-en#:~:text=The%20report%20analyzes%20data%20

[17] I. Chua, "Real Life Examples of Web Vulnerabilities (OWASP Top 10)," Horangi Cybersecurity, 2022. https://www.horangi.com/blog/real-life-examples-of-web-vulnerabilities

[18] Wassermann, Gary, and Zhendong Su. "Static detection of cross-site scripting vulnerabilities." In *Proceedings of the 30th international conference on Software engineering*, pp. 171-180. 2008. https://doi.org/10.1145/1368088.1368112
[19] Hassan, Md Maruf, Shamima Sultana Nipa, Marjan Akter, Rafita Haque, Fabiha Nawar Deepa, Mostafijur Rahman, Md Asif Siddiqui, and Md Hasan Sharif. "Broken authentication and session management vulnerability: a case study of web application." *Int. J. Simul. Syst. Sci. Technol* 19, no. 2 (2018): 1-11. https://doi.org/10.5013/IJSSST.a.19.02.06
[20] Hassan, Md Maruf, Shamima Sultana Nipa, Marjan Akter, Rafita Haque, Fabiha Nawar Deepa, Mostafijur Rahman, Md Asif Siddiqui, and Md Hasan Sharif. "Broken authentication and session management vulnerability: a case study of web application." *Int. J. Simul. Syst. Sci. Technol* 19, no. 2 (2018): 1-11. https://doi.org/10.5013/IJSSST.a.19.02.06
[21] Fry, Ann. "A forensic web log analysis tool: Techniques and implementation." PhD diss., Concordia University, 2011.
[22] Wood, Daniel, Noah Apthorpe, and Nick Feamster. "Cleartext data transmissions in consumer iot medical devices." In *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*, pp. 7-12. 2017. https://doi.org/10.1145/3139937.3139939
[23] Makrakis, Georgios Michail, Constantinos Kolias, Georgios Kambourakis, Craig Rieger, and Jacob Benjamin. "Vulnerabilities and attacks against industrial control systems and critical infrastructures." *arXiv preprint arXiv:2109.03945* (2021).
[24] Palmer, Chris. "Secure Session Management with cookies for Web applications." *iSEC Partners, Inc* (2008).
[25] Gupta, B. B., Shashank Gupta, S. Gangwar, M. Kumar, and P. K. Meena. "Cross-site scripting (XSS) abuse and defense: exploitation on several testing bed environments and its defense." *Journal of Information Privacy and Security* 11, no. 2 (2015): 118-136. https://doi.org/10.1080/15536548.2015.1044865
[26] Khalid, Muhammad Noman, Humera Farooq, Muhammad Iqbal, Muhammad Talha Alam, and Kamran Rasheed. "Predicting web vulnerabilities in web applications based on machine learning." In *Intelligent Technologies and Applications: First International Conference, INTAP 2018, Bahawalpur, Pakistan, October 23-25, 2018, Revised Selected Papers 1*, pp. 473-484. Springer Singapore, 2019. https://doi.org/10.1007/978-981-13-6052-7_41
[27] Rodríguez, Germán E., Jenny G. Torres, Pamela Flores, and Diego E. Benavides. "Cross-site scripting (XSS) attacks and mitigation: A survey." *Computer Networks* 166 (2020): 106960. https://doi.org/10.1016/j.comnet.2019.106960
[28] Ravindran, Urshila, and Raghu Vamsi Potukuchi. "A Review on Web Application Vulnerability Assessment and Penetration Testing." *Review of Computer Engineering Studies* 9, no. 1 (2022). https://doi.org/10.18280/rces.090101
[29] Idrissi, S. E., Naoual Berbiche, Fatima Guerouate, and M. Shibi. "Performance evaluation of web application security scanners for prevention and protection against vulnerabilities." *International Journal of Applied Engineering Research* 12, no. 21 (2017): 11068-11076.
[30] Heikal Ismail, Muhammad, Tinia Idaty Mohd Ghazi, Muhammad Hazwan Hamzah, Latifah Abd Manaf, Ramli Mohd Tahir, Ahadi Mohd Nasir, and Ammar Ehsan Omar. "Impact of movement control order (Mco) due to coronavirus disease (covid-19) on food waste generation: A case study in klang valley, malaysia." *Sustainability* 12, no. 21 (2020): 8848. https://doi.org/10.3390/su12218848
[31] Alzahrani, Abdulrahman, Ali Alqazzaz, Ye Zhu, Huirong Fu, and Nabil Almashfi. "Web application security tools analysis." In *2017 ieee 3rd international conference on big data security on cloud (bigdatasecurity), ieee international conference on high performance and smart computing (hpsc), and ieee international conference on intelligent data and security (ids)*, pp. 237-242. IEEE, 2017. https://doi.org/10.1109/BigDataSecurity.2017.47
[32] HackerOne, "Top Ten Vulnerabilities | HackerOne," www.hackerone.com. https://www.hackerone.com/top-ten-vulnerabilities.