I.J. Information Engineering and Electronic Business, **2023, 5, 42-53**
Published Online on October 8, 2023 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2023.05.05

Vulnerabilities Assessment of Financial and Government Websites: A Developing Country Perspective

Md. Asif Khan Rifat
Institute of Information Technology, University of Dhaka, Dhaka 1000, Bangladesh
Email: rifat.asifkhan@gmail.com
ORCID iD: https://orcid.org/0009-0006-7852-915X

Yeasmin Sultana
Institute of Information Technology, University of Dhaka, Dhaka 1000, Bangladesh
Email: yspriya58@gmail.com
ORCID iD: https://orcid.org/0009-0001-7439-7907

B M Mainul Hossain*
Institute of Information Technology, University of Dhaka, Dhaka 1000, Bangladesh
Email: mainul@iit.du.ac.bd
ORCID iD: https://orcid.org/0000-0002-0447-4217
*Corresponding Author

Received: 08 July, 2023; Revised: 02 August, 2023; Accepted: 20 September, 2023; Published: 08 October, 2023

Abstract: The growing number of web applications in a developing country like Bangladesh has led to an increase in cybercrime activities. This study focuses on measuring the vulnerabilities present in financial and government websites of Bangladesh to address the rising security concerns. We reviewed related works on web application vulnerability scanners, comparative studies on web application security parameters, surveys on web application penetration testing methodologies and tools, and security analyses of government and financial websites in Bangladesh. Existing studies in the context of developing countries have provided limited insight into web application vulnerabilities and their solutions. These studies have focused on specific vulnerabilities, lacked comprehensive evaluations of security parameters, and offered a limited comparative analysis of vulnerability scanners. Our study aims to address these gaps by conducting an in-depth analysis using the OWASP ZAP tool to scan and analyze risk alerts, including risk levels such as high, medium, low, and informational. Our investigation unveiled eight key vulnerabilities, including Hash Disclosure, SQL injection (SQLi), Cross-Site Request Forgery (CSRF), missing Content Security Policy (CSP) headers, Cross-Domain JavaScript File Inclusion, absence of X-Content-Type-Options headers, Cache-related concerns, and potential Cross-Site Scripting (XSS), which can lead to revealing hidden information, enabling malicious code, and failing to protect against specific types of attacks. In essence, this study does not only reveal major security weaknesses but also provides guidance on how to mitigate them, thereby playing a vital role in promoting enhanced cybersecurity practices within the nation.

Index Terms: Web Security Vulnerabilities, CSRF, XSS, SQL Injection, CSP Header, OWASP Zap, Mitigation Techniques.

1. Introduction

The number of web applications being used in developing countries such as Bangladesh is growing rapidly, but at the same time, cybercrime is also on the rise. Recently, there has been a surge in the activities of cybercriminals who are specifically targeting government institutions, financial organizations, eCommerce platforms, educational institutions, and other reputable organizations. Cybersecurity statistics indicate that there are approximately 2,200 instances of cyber attacks occurring daily, with a new cyber attack happening roughly every 39 seconds on average [1]. The escalating frequency and severity of cyber incidents involving Bangladesh's online infrastructure underscore the pressing need for comprehensive research aimed at addressing the underlying vulnerabilities in the nation's web applications. The recent

occurrences of extensive data breaches in Bangladesh underscore the alarming vulnerabilities plaguing the digital landscape of the country. One report by TechCrunch, referencing a discovery made by Bitcrack Cyber Security researcher Viktor Markopoulos, brings to light the exposure of sensitive personal information of over 50 million Bangladeshi citizens from a government website [2]. Additionally, a coordinated cyber attack executed by Indian hackers has led to data breaches across 25 public and private institutions, compromising critical data from entities like the Investment Corporation of Bangladesh and the Directorate General of Health Services [3]. The majority of the Bangladeshi government's websites, including the PMO (Prime Minister's Office) and Bangabhaban (President's Office), have been taken down in a significant cyberattack supposedly in support of strikes against quotas in public jobs [4]. The website of Bangladesh's National Defence College (NDC) was hacked in 2017 by hackers from Myanmar. On the NDC website (ndc.gov.bd), hackers have put a statement and some graphic images [5]. These incidents aptly illustrate the grave susceptibility of Bangladeshi websites to cyber threats, emphasizing the urgency for meticulous research endeavors to identify weaknesses and implement effective mitigation strategies. Such research not only safeguards the privacy and security of citizens but also contributes to the overall resilience of the nation's digital infrastructure in the face of escalating cyber threats.

SQLi is one of the most common yet dangerous website security vulnerabilities [6]. This study found that 63% of Bangladesh's educational websites are extremely insecure that any individual is able to edit and retrieve almost any data. In Bangladesh, 64% of the websites are working with common vulnerabilities such as SQL Injection, Cross Site Scripting (XSS), and Transport Layer Security (TLS), and in particular, government websites are in a devastating condition [7]. The survey discovered that 21% of all websites are vulnerable to BAS, 15% of all websites have been found to be sensitive to CSRF, 56% of websites are at risk of SQL Injection, and 53% of websites are vulnerable to XSS.

In order to proactively resolve any security gaps and vulnerabilities, reduce the risk of cyberattacks, and safeguard sensitive data from unauthorized access or exploitation, testing website vulnerability is crucial. Testing a web application manually is a time-consuming process, hence in modern times, testers have switched to automated testing to save time and money. However, the technical capabilities and effectiveness of various automated tools for vulnerability identification differ from one another. As a result, the choice of tools should take into consideration a number of aspects, including the features of the scanner, the accessibility of the documentation, and the scanner's capacity to identify vulnerabilities. As far as we are aware, OWASP ZAP is still the most popular, open-source, and efficient vulnerability scanner that we have used to evaluate website vulnerabilities and generate reports.

The primary research objective of our study is to comprehensively address the vulnerabilities inherent within the web applications of financial and government entities within the context of a developing nation such as Bangladesh. The overarching aim is to effectively tackle the escalating security concerns arising from the surge in cybercrime activities, which have manifested due to the rapid proliferation of web applications in this geographical region.

2. Related Works

Several research studies have been conducted to evaluate the vulnerabilities of web applications, taking into account various factors such as exploitation methods and preventive measures. Within this body of research, we highlight a few studies that specifically focused on assessing the vulnerabilities of websites in Bangladesh.

ALAZMI et al. conducted a Systematic Literature Review (SLR) to identify and analyze the different characteristics and features of web application vulnerability scanners (WVS's). They gathered 320 research articles from various databases such as Google Scholar, ACM Digital Library, and SpringerLink, and selected 90 research papers to evaluate. They found that only fifteen evaluation studies were performed to compare the effectiveness of twelve WVSs in identifying vulnerabilities determined in web applications and that the majority of these evaluations only tested two of the Top Ten vulnerability types identified by the Open Web Application Security Project (OWASP) such as SQL injection (SQLi) and Cross-Site Scripting (XSS) [8].

Shahid et al. conducted a comparative study, analyzing existing research and identifying the key parameters and trends in web application security. In this paper, they used black box penetration testing techniques to evaluate open-source web application assessment tools. They analyze the scanners based on the Top 10 OWASP Web application security vulnerabilities such as 1) Broken Access Control, 2) Cryptographic Failures, 3) Injection, 4) Insecure Design, 5) Security Misconfiguration, 6) Vulnerable and Outdated Components, 7) Identification and Authentication Failures, 8) Software and Data Integrity Failures, 9) Security Logging and Monitoring Failures, and 10) Server-Side Request Forgery. In their opinion, OWASP-ZAP had a better percentage of vulnerability detection in the open-source tools category [9].

Altulaihan et al. conducted a survey, analyzing existing literature to examine the various methodologies and tools used for web application penetration testing. They chose the publications to evaluate from the Saudi Digital Library and Google Scholar databases using the PRISMA methodology. Between January 2018 and December 2022. They chose 13 publications from a total of 22280 papers that discussed penetration testing techniques, cybersecurity concerns, and the web environment. They described how human and automated testing differs from one another. They said that the Acunetix, Netsparker online Vulnerability Scanner, Vega, Wapiti, OWASP ZAP, IronWASP, and W3af were more significant commercial and open-source testing tools than the others. They introduced seven test criteria, including

protection against SQL Injection Attacks, broken authentication, and session hijacking, protection against XSS Attacks, protection against insecure direct object references, protection against missing function-level access control, protection against sensitive data exposure, protection against CSRF Attacks, and protection against unvalidated redirects and forwards [10].

Masum et al. presented a security analysis of government and financial websites in Bangladesh. They used the Wappalyzer tool to identify the technologies being used on the targeted websites, the Wafw00f tool to identify firewalls being used on a website, and the Sublist3r tool to identify subdomains of a website. Using Acunetix and ZAP tools, they identify four types of risk as High, Medium, Low, and Informational for specific websites. According to the authors, Acunetix outperformed ZAP for all types of vulnerability and the top 5 vulnerabilities for certain websites are 1) ClickJacking, 2) Misconfiguration, 3) Cross-Site Request Forgery, 4) Information Disclosure, and 5) Cross-Site Scripting. They only provided mitigation strategies for the PHP language. Different languages have different mitigation techniques and functionalities. But they did not provide any for any other language [11].

Delwar et al. analyzed the SQL injection (SQLi) vulnerabilities present in education sector websites in Bangladesh. They used Manual penetration testing with the black box technique to identify various types of SQLi vulnerabilities such as normal, error-based double query, and blind injection techniques. In their dataset, 309 websites are found vulnerable among 359 educational websites. According to their study, 79 of the 309 websites can have full database access any information including the student's results can be edited and they could extract the table names, column names, user data, admin passwords, and login ids from 120 websites [6].

Moniruzzaman et al. focused on measuring the vulnerabilities present in websites in Bangladesh against six selected attack vectors: SQL Injection, Cross Site Scripting (XSS), Broken Authentication and Session (BAS), CSRF, Port Scan Attack, and Transport Layer Security (TLS). Based on website traffic and analytics, they chose the most well-known government and other websites and gathered the dependents' CVE (Common Vulnerabilities and Exposures) database. They used penetration testing and source code analysis mechanisms to represent the black box and white box testing. They claim that TLS revealed the state of a web server at the transport layer and that XSS and CSRF revealed operational limitations in websites and the status of important ports, while SQL Injection and BAS presented database vulnerability [7].

In addition, a limited number of surveys have been done from the Bangladeshi perspective. These studies highlight the need for more comprehensive work to understand the extent of cyber vulnerabilities and their potential impacts on the country's digital infrastructure.

3. Methodology

Web application vulnerabilities can lead to serious consequences. It is especially risky if the government and financial websites have vulnerabilities, as the potential costs could be extremely harmful. This research methodology aims to identify various vulnerabilities. We utilized different tools and techniques to collect data and analyze security weaknesses. The insights gained from this analysis can help raise awareness during web application development and address existing vulnerabilities in current websites. This section discusses the methods and tools used in our experiment. Our methodology consists of three main steps: Experimental Information Gathering, Website and Tools Selection, and Environment Setup and Vulnerability Assessment. Fig. 1 provides an overview of our proposed model.

To begin, we chose our target websites and gathered various types of information. Next, we identified the subdomains of each target and scanned them using our selected vulnerability scanners. Afterward, we saved and thoroughly analyzed the results to identify the top eight vulnerabilities. Lastly, we present mitigation techniques that can be implemented by anyone to secure their website.

3.1 Websites and Tool Selection

The government activity and finance sectors are significant sectors of any country. If hackers successfully attack those websites, the cost will be huge. For the purpose of this study, 30 Bangladesh-centered financial and government websites are considered. The process of selecting government and financial websites is based on their website traffic and statistics obtained from SimilarWeb and Semrush [12,13]. The list of selected websites along with their corresponding ranking are presented in Table 1. After selecting the target websites, we gathered information on the applied firewalls and subdomains employed in the targeted websites. To accomplish this, we utilized several open-source tools, namely the Wappalyzer extension [14], wafw00f [15], and sublit3r [16]. Subsequently, we evaluated the vulnerability of these websites using an automated vulnerability scanner called Owasp ZAP [17]. ZAP is a widely recognized and efficient open-source tool that identifies vulnerabilities in each of the targeted websites. An overview of the selected tools for this study is given below.

- Wappalyzer: Wappalyzer is a tool that identifies the technologies utilized on a website, including CMS, frameworks, platforms, JavaScript libraries, and more. It has the capability to identify thousands of technologies across various categories such as programming languages, analytics tools, marketing tools, payment processors, CRM systems, CDNs, and more.

- Wafw00f: It is a firewall fingerprinting tool used to detect the presence of firewalls in an application. It accomplishes this by sending a simple HTTP request and analyzing the response to identify the firewall solutions in use. If the initial attempt is unsuccessful, it sends a series of malicious HTTP requests to determine the specific firewall in place. If this approach also fails, Wafw00f analyzes the responses received from previous requests to make an educated guess about the security solution that is responding to our attacks.
- Sublist3r: It is an open-source tool designed to identify the subdomains associated with a website. It utilizes open-source intelligence (OSINT) and search engines such as Google, Yahoo, Bing, Baidu, and Ask to enumerate subdomains. Additionally, Sublist3r leverages resources like Netcraft, Virustotal, ThreatCrowd, DNSdumpster, and ReverseDNS to further expand the subdomain enumeration process.
- ZAP: Zed Attack Proxy (ZAP) is an open-source vulnerability scanner developed by OWASP. It accepts URLs as input and conducts vulnerability scans. ZAP possesses a database containing vulnerability fingerprints that are compared against the target to identify vulnerabilities like SQL injection (SQLi), cross-site scripting (XSS), cross-site request forgery (CSRF), directory traversal, information disclosure, and more.

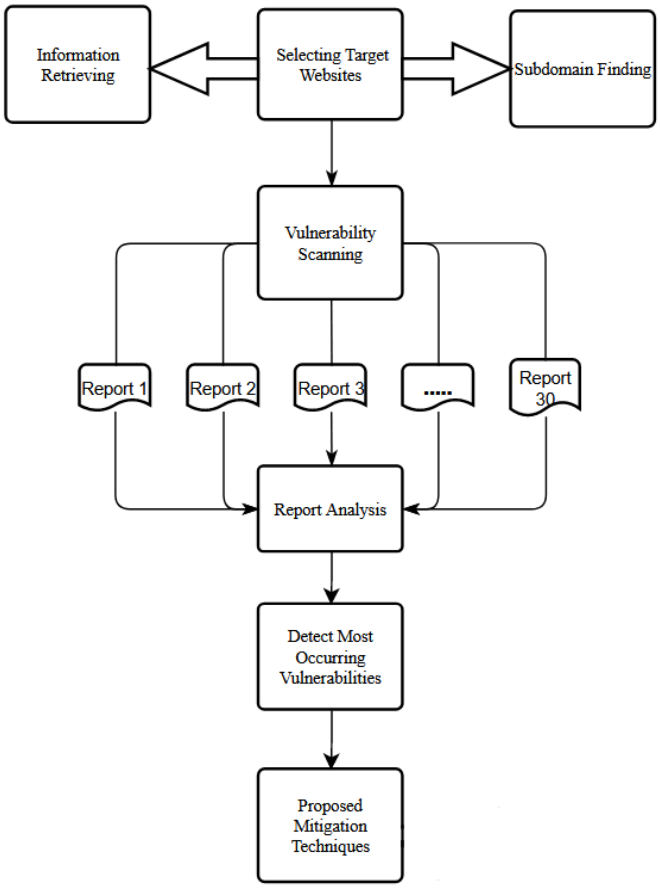

Fig. 1. Diagram of the Research Workflow

3.2 Environment Setup

During the course of this experiment, we utilized a personal computer running on the Kali Linux 2023.1 operating system, equipped with a 10Mbps internet connection. The hardware configuration consisted of an Intel Core i5 8th generation CPU, 16GB of RAM, Nvidia GTX 1060 6GB Graphics, and a 512GB SSD. It is worth mentioning that the recommended setup for conducting similar experiments using the ZAP tool includes a CPU with a minimum clock speed of 2.00 GHz, an Intel Core i5 processor or higher, 8GB of RAM or higher, and a GPU, while not mandatory, can be beneficial. Additionally, a stable internet connection exceeding 10 Mbps is also recommended.

3.3 Vulnerability Assessment

Vulnerability assessment is the process of systematically prioritizing, classifying, identifying, and quantifying vulnerabilities within a system. It serves as a technique to evaluate security risks and mitigate the potential for web attacks. Several testing tools, including Acunetix, ZAP, and Nikto, can be used to assess security risks. In this study, ZAP is employed to perform the vulnerability assessment. The process involved multiple steps:

Table 1. List of Selected Websites for Vulnerabilities Assessment

Category	Serial No.	Websites	URL	Rank in Bangladesh
Financial	1	Bangladesh Bank	https://www.bb.org.bd	104
	2	Sonali Bank	https://www.sonalibank.com.bd	2289
	3	Agrani Bank	https://www.agranibank.org	4058
	4	Eastern Bank	https://www.ebl.com.bd	1164
	5	DBBL	https://www.dutchbanglabank.com	301
	6	IBBL	https://www.islamibankbd.com	187
	7	UCB	https://www.ucb.com.bd	431
	8	Brac Bank	https://www.bracbank.com	354
	9	Bkash	https://www.bkash.com	10
	10	Nagad	https://nagad.com.bd	302
	11	Dhaka Stock Exchange	https://www.dsebd.org	120
	12	Chittagong Stock Exchange	https://www.cse.com.bd	57
	13	LankaBangla Finance	https://lankabd.com	1037
	14	IDLC	https://idlc.com	5180
	15	AmarStock	https://www.amarstock.com	784
	16	EcoSoftBD	https://www.ecosoftbd.com	7016
Governmental	17	Election Commission	http://www.ecs.gov.bd	75
	18	Ministry of Foreign Affairs	http://www.mofa.gov.bd	3385
	19	Ministry of Home Affairs	http://www.mha.gov.bd	684
	20	Ministry of Finance	http://www.mof.gov.bd	132
	21	Law and Justice Division	http://www.lawjusticediv.gov.bd	361
	22	Ministry of Public Admin	https://mopa.gov.bd	1689
	23	ICT Division	https://ictd.gov.bd	191
	24	Public Service Commission	http://www.bpsc.gov.bd	991
	25	Income Tax	https://incometax.gov.bd	681
	26	NID	https://nidw.gov.bd	76
	27	Education Results	http://www.educationboardresults.gov.bd	45
	28	Immigration & Passport	http://www.dip.gov.bd	7568
	29	Dhaka City Corporation	http://www.dncc.gov.bd	5171
	30	Biman Bangladesh Airlines	http://biman.gov.bd	1115

- **Information Retrieving**: The first step involved using tools such as Wappalyzer, wafw00f, and sublist3r to gather information about the technologies utilized by the target websites. Wappalyzer, a browser add-on, detects the technologies employed by web applications, including programming languages, CDNs, web frameworks, and servers. It can also identify vulnerable versions of these technologies.
- **Subdomain Finding**: After gathering information about the technologies, the sublist3r tool was used to discover subdomains associated with the target websites. The command "sublist3r -d target.com" was utilized to identify the subdomains.
- **Firewall Detection**: The wafw00f tool was utilized to fingerprint firewalls. By sending malformed packets to the server and analyzing the responses received, wafw00f determines the presence of firewalls.
- **Website Scanning**: Once all the necessary information about the target websites was obtained, Acunetix and ZAP were employed for vulnerability assessment. Acunetix categorizes the assessment results into four categories: high, medium, low, and informational alerts, based on the impact of the vulnerabilities. ZAP detects vulnerabilities by comparing fingerprints against a database and provides results categorized into the same four risk categories.

3.4 OWASP's Methodology for Risk and Confidence Analysis

The primary concept behind OWASP ZAP's risk and confidence level calculations lies in its ability to assess vulnerabilities and their potential impact. It employs a scoring mechanism that combines various factors to estimate the risk associated with a particular vulnerability.

3.4.1 Risk Rating Approach

The risk score is often derived from a combination of the vulnerability's severity, likelihood of exploitation, and potential impact [18]. Mathematically, the risk level (R) formula is listed in equation (1).

$$Risk = Likelihood \times Impact \quad (1)$$

The risk level assessment in OWASP ZAP is derived from the combination of the following factors:

- Impact: This refers to the potential harm that could result from a successful exploit of the identified vulnerability. Common aspects include data exposure, privilege escalation, and system compromise. The impact is typically assigned a numeric value representing the potential severity of the vulnerability.
- Likelihood: This factor takes into account the likelihood of the vulnerability being exploited. Elements considered include the existence of known exploits, ease of exploitation, and attacker motivation. Likelihood is also assigned a numerical value, reflecting the probability of successful exploitation.
- Vulnerability Type: Different types of vulnerabilities carry varying levels of risk. For instance, a critical remote code execution vulnerability would have a higher inherent risk than a less severe information disclosure vulnerability.
- Exploitability: This factor assesses the ease with which an attacker could exploit the vulnerability. Vulnerabilities that can be easily exploited receive higher risk ratings.

3.4.2 Confidence Level Calculation

The confidence level in OWASP ZAP's findings is determined by various considerations:

- Evidence: The tool evaluates the extent and quality of evidence it has gathered to support the presence of a vulnerability. Strong evidence, such as multiple proof-of-concept exploits, increases confidence.
- False Positives/Negatives: The rate of false positives and false negatives affects the tool's confidence level. A higher rate of accurate findings relative to false alarms increases confidence.
- Repeatability: The tool considers whether the vulnerability can be consistently reproduced upon retesting. High repeatability increases confidence.
- Severity of Vulnerability: The severity of the vulnerability detected also influences confidence. Critical vulnerabilities are subject to more rigorous validation, enhancing confidence.

The proposed methodology effectively aligns with the research objectives by comprehensively addressing vulnerabilities in web applications of financial and government entities within Bangladesh. Recognizing the escalating security concerns stemming from cybercrime proliferation, the methodology advocates for proactive vulnerability resolution. Automated testing, driven by the need for efficiency, is strategically employed while considering the distinct capabilities of vulnerability identification tools. OWASP ZAP is acknowledged for its efficiency in evaluating vulnerabilities. The three-phase methodology encompasses information gathering, subdomain identification, vulnerability assessment on the target websites, and mitigation strategy formulation. The assessment results were categorized based on the severity of the vulnerabilities discovered. By identifying and rectifying vulnerabilities, the study aims to enhancecy bersecurity practices during web application development and bolster the protection of existing

sites, effectively contributing to the resilience of digital infrastructure against burgeoning cyber threats.

4. Result and Analysis

In this section, we present a comprehensive analytical report based on the data collected from the ZAP tool. The risk levels are categorized into four groups: High, Medium, Low, and Information. We provided the analysis results of alert counts by risk and confidence level in Table 2, as well as total vulnerability occurrences in Fig. 3. Furthermore, we discussed the top eight security risks identified in our targeted websites, along with recommended mitigation techniques.

4.1 Distribution of Alerts

The alerts generated by the scanners are classified into four categories:

- High Risk: This category represents the most severe level of risk. Exploiting vulnerabilities in this category can lead to a complete compromise of a system's confidentiality, integrity, and availability. Attackers can gain lateral movement within the system and potentially escalate privileges to other systems.
- Medium Risk: This category poses a lesser degree of risk compared to high-level risks. Exploiting vulnerabilities in this category can result in partial compromise of a system's confidentiality, integrity, and availability. Most vulnerabilities in this category require specialized access or user interaction to be successfully exploited.
- Low Risk: Vulnerabilities in this category offer limited opportunities for attackers to exploit a system. It is challenging for an attacker to successfully exploit these vulnerabilities, as they typically require specialized access, user interaction, or specific circumstances to compromise a system.
- Informational: This category includes vulnerabilities that are currently not exploitable. However, they provide valuable information about the system, which can be utilized by attackers in the future to exploit the system.

By categorizing the alerts into these four groups, we gain a better understanding of the severity and potential impact of the identified vulnerabilities. This information allows us to prioritize the necessary mitigation measures to enhance the security of the targeted websites.

The confidence level, as assessed by the OWASP ZAP tool, indicates the reliability or certainty associated with the identified vulnerabilities. It classifies the confidence level into categories like High, Medium, and Low, reflecting the level of assurance or trust in the accuracy and validity of the identified vulnerabilities.

- High Confidence: It indicates a high level of certainty or confidence in the accuracy and validity of the identified vulnerability.
- Medium Confidence: It suggests a moderate level of confidence in the identified vulnerability, which may require further investigation or verification.
- Low Confidence: It represents a lower level of confidence or uncertainty in the identified vulnerability, which may necessitate additional validation or testing.

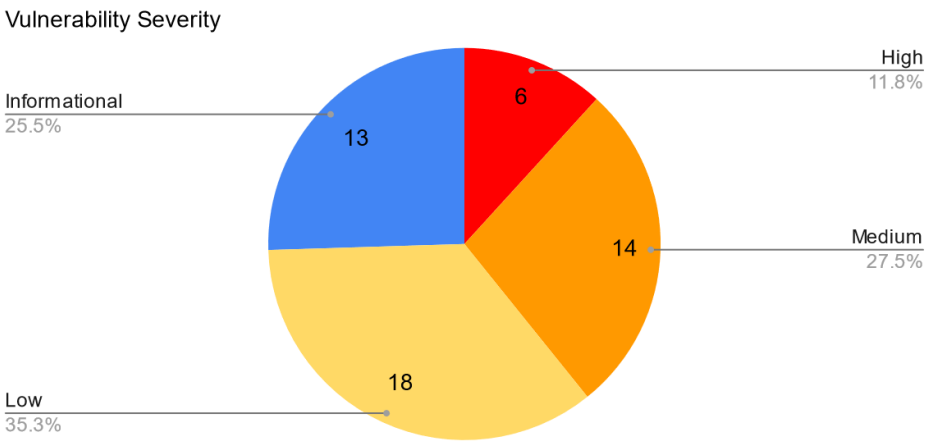

Fig. 2. Alerts based on Severity Level

The OWASP ZAP tool generated a report showing the distribution of alerts based on risk and confidence levels. The key findings reveal that medium-risk alerts have the highest count, comprising 27.5% of the total alerts, followed by low-risk alerts at 35.3%. High-risk alerts make up the smallest portion, accounting for 11.8%. In terms of confidence, medium confidence alerts are the most prevalent, representing 56.9% of the alerts, while high confidence alerts have the

lowest count at 21.6%. Overall, the report suggests a higher occurrence of medium and low-risk alerts, with medium confidence being the most common level.

Table 2. Number of Alerts for Each Level of Risk and Confidence

		Confidence			
		High	Medium	Low	Total
Risk	High	1 (2.0%)	4 (7.8%)	1 (2.0%)	6 (11.8%)
	Medium	4 (7.8%)	8 (15.7%)	2 (3.9%)	14 (27.5%)
	Low	5 (9.8%)	12 (23.5%)	1 (2.0%)	18 (35.3%)
	Informational	1 (2.0%)	5 (9.8%)	7 (13.7%)	13 (25.5%)
	Total	11 (21.6%)	29 (56.9%)	11 (21.6%)	51 (100%)

4.2. Detected Risk Alerts

The breakdown of detected vulnerabilities by ZAP has been shown in Fig. 3.

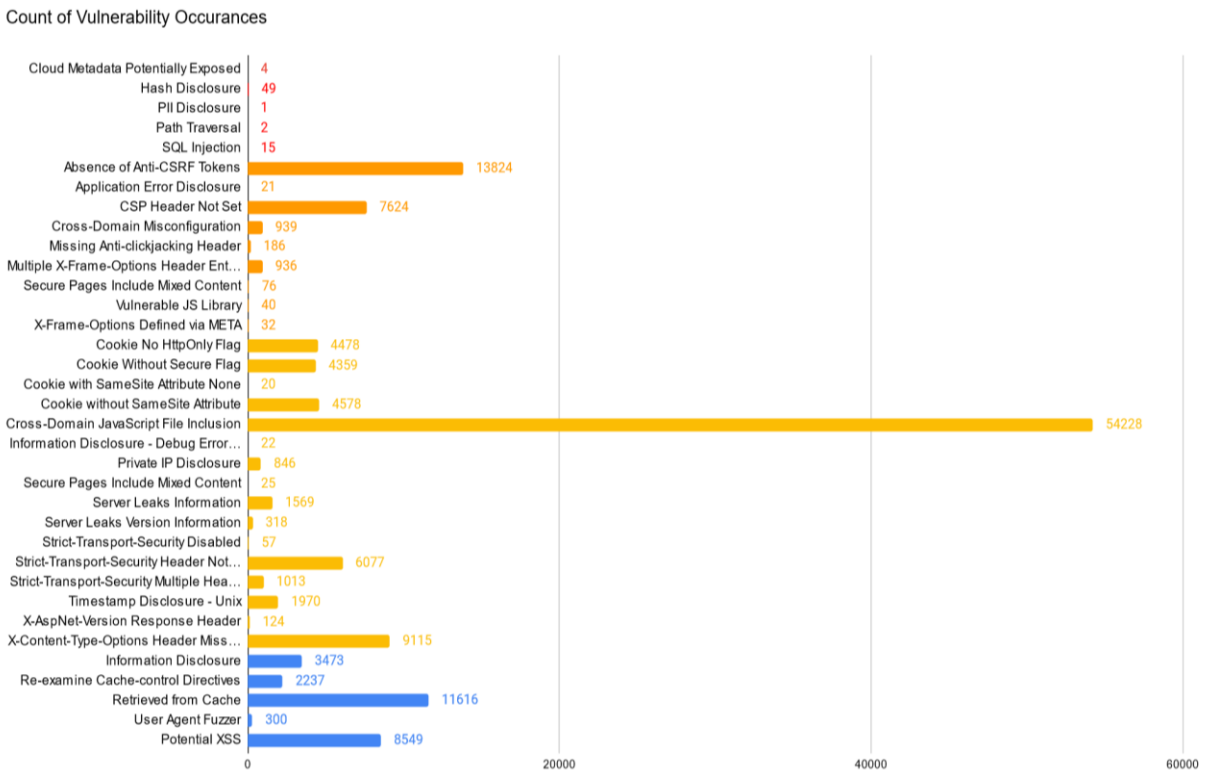

Fig. 3. Total No. of Vulnerabilities Occurrences received from target websites

This horizontal bar chart shows the number of alerts of each alert type, together with the alert type's risk level differentiated by bar color. We obtained a total of 51 vulnerabilities categorized by 4 levels on 30 websites. Due to confidentiality reasons, we have not disclosed which sites are vulnerable to any particular attacks and where those exist. These vulnerabilities are listed below:

- High Risk: Cloud Metadata Potentially Exposed, Hash Disclosure, PII Disclosure, Path Traversal, SQL Injection.
- Medium Risk: Absence of Anti-CSRF Tokens, Application Error Disclosure, CSP Header Not Set, Cross-Domain Misconfiguration, Missing Anti-clickjacking Header, Multiple X-Frame-Options Header Entries, Secure Pages Include Mixed Content, Vulnerable JS Library, X-Frame-Options Defined via META.
- Low Risk: Cookie No HttpOnly Flag, Cookie Without Secure Flag, Cookie with SameSite Attribute None, Cookie without SameSite Attribute, Cross-Domain JavaScript File Inclusion, Information Disclosure - Debug Error Messages, Private IP Disclosure, Secure Pages Include Mixed Content, Server Leaks, Information Server Leaks, Version Information, Strict-Transport-Security Disabled, Strict-Transport-Security Header Not Set, Strict-Transport-Security Multiple Header, Timestamp Disclosure - Unix, X-AspNet-Version Response Header, X-Content-Type-Options Header Missing.
- Informational Risk: Information Disclosure, Re-examine Cache-control Directives, Retrieved from Cache User, Agent Fuzzer, Potential XSS.

5. Discussion

This section highlights the most frequent vulnerabilities and mitigation techniques. Based on the presented findings, we have selected the top 2 vulnerabilities from each category, in total 8 vulnerabilities. Those are (I) Hash Disclosure, (II) SQL Injection, (III) CSRF, (IV) CSP Header Not Set, (V) Cross-Domain JavaScript File Inclusion, (VI) X-Content-Type-Options Header Missing, (VII) Retrieved from Cache, (VIII) User Controllable HTML Element Attribute (Potential XSS). The mitigation techniques provided by the ZAP tool for the top eight alerts are listed in Table 3. The report provides insights into the security status of official websites in Bangladesh and offers mitigation techniques for these vulnerabilities, which have received less attention in the country. The findings of this research contribute to raising awareness among website administrators and stakeholders about the importance of implementing robust security measures to protect sensitive data from unauthorized access and manipulation.

To ensure the reliability and accuracy of the results presented in the aforementioned section, meticulous steps were undertaken throughout the research process. Firstly, the utilization of the OWASP ZAP tool, a well-recognized and widely employed vulnerability scanner, contributes to the credibility of the findings. The rigorous application of the tool facilitated the systematic detection and categorization of vulnerabilities across a representative sample of 30 websites. The tabulated breakdown of alerts and vulnerabilities by risk and confidence levels, as well as the graphical representation in Figure 3, adds transparency and clarity to the presentation of outcomes. Moreover, confidentiality preservation regarding the identification of specific vulnerable sites was upheld, safeguarding sensitive information and reinforcing research ethics. The comprehensive classification of vulnerabilities based on their risk levels, encompassing high, medium, low, and informational, demonstrates a structured and systematic approach to analysis. By adhering to established methodologies, tool reliability, data integrity, and ethical considerations, the research endeavors to establish a reliable foundation for understanding and addressing web application vulnerabilities within the context of the study.

Table 3. Top 8 most occurring vulnerabilities and mitigation techniques.

Serial No.	Vulnerability Name	Description	Mitigation
1	Hash Disclosure	A hash was disclosed by the web server.	● Ensure that hashes that are used to protect credentials or other resources are not leaked by the web server or database. There is typically no requirement for password hashes to be accessible to the web browser. [19]
2	SQL Injection (SQLi)	It occurs when an attacker is able to inject malicious SQL code into a website or application that uses SQL to interact with a database.	● Use of prepared statements (with parameterized queries). ● Use of properly constructed stored procedures. ● Allow-list input validation. ● Escaping all user-supplied input. [20]
3	Cross Site Request Forgery (CSRF)	No Anti-CSRF tokens were found in a HTML submission form.	Use CSRF token in each form which creates token randomly. Pass the token as a hidden input type.

4	CSP Header Not Set	The absence of a Content Security Policy (CSP) header poses a security vulnerability as it fails to detect and mitigate various forms of attacks, such as Cross Site Scripting (XSS) and data injection attacks. These malicious activities can result in data breaches, website tampering, and malware distribution. By implementing CSP, website owners can utilize standardized HTTP headers to specify trusted content sources that browsers are permitted to load on a given page.	• To resolve the issue of an unset Content Security Policy (CSP) header, it is necessary to configure the web server to include the Content-Security-Policy HTTP Header. This header should be assigned appropriate values to effectively control the resources that the browser can load for the page. [21] The syntax: `Content-Security-Policy: <policy-directive>; <policy-directive>`
5	Cross-Domain JavaScript File Inclusion	The page incorporates one or more script files from an external domain.	• Ensure that JavaScript source files are exclusively loaded from trusted origins, and that these sources cannot be manipulated by end users of the application. [22]
6	X-Content-Type-Options Header Missing	The absence of the X-Content-Type-Options header, specifically not set to 'nosniff', allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body. Consequently, the response body may be interpreted and displayed as a different content type than intended.	• To address this, it is important to ensure that the application or web server correctly sets the Content-Type header and includes the X-Content-Type-Options header set to 'nosniff' for all web pages. Whenever possible, encourage end users to utilize modern and standards-compliant web browsers that do not perform MIME-sniffing or can be directed by the web application or server to avoid MIME-sniffing. [23]
7	Retrieved from Cache	The content was fetched from a shared cache, which may lead to the inadvertent exposure of sensitive, personal, or user-specific information if present in the response data.	• Whenever feasible, ensure that HTTP responses do not contain any sensitive, personal, or user-specific data. If such data is present, consider utilizing appropriate 'Cache-Control', 'Pragma', and 'Expires' HTTP response headers to restrict or prevent the storage and retrieval of content from the cache by other users. [24]
8	User Controllable HTML Element Attribute (Potential XSS)	This analysis examines user-supplied input within query string parameters and POST data to identify instances where HTML attribute values could potentially be controlled. This feature acts as a preliminary detection mechanism for cross-site scripting (XSS) vulnerabilities, necessitating further evaluation by a security analyst to determine the potential for exploitation.	• Allow HTML parts injection only where needed. • Validate all input and sanitize output before writing to any HTML attributes. [25]

6. Conclusion

This study has significantly contributed to the advancement of knowledge in the field of cybersecurity by offering a comprehensive assessment of security vulnerabilities present within the financial and government websites of Bangladesh. By utilizing the ZAP tool, the system conducted scans and analyzed risk alerts across four categories: high, medium, low, and informational for each website. During the analysis of the alert results, it is observed that certain web applications showed zero alerts. This could imply either a limitation in the scanning process or the absence of any vulnerabilities in those particular sites. In addition to the examination of risk alerts, this study has presented a breakdown of vulnerabilities, focusing on the top eight most critical vulnerabilities identified through the analysis of security-related records. Mitigation techniques have been provided for these vulnerabilities, with the aim of directing the attention of the Bangladesh government toward addressing these issues and fostering further development.

It is important to note that this study has only encompassed a limited number of websites in the financial and government sectors. The observed absence of alerts in certain web applications underscores the need for further exploration into potential scanning limitations or the absence of vulnerabilities. The comprehensive breakdown of vulnerabilities, coupled with the provision of mitigation strategies, not only contributes to immediate issue resolution but also serve as a foundational resource for the Bangladesh government's cybersecurity efforts. Scientifically, this work provides a robust justification by enhancing the understanding of web application vulnerabilities, thereby informing the development of strategies that enhance online security, safeguard sensitive data, and bolster national cybersecurity

endeavors. The anticipated extensions of this study into other website categories, such as academia and e-commerce, hold the promise of yielding broader insights into the security landscape of Bangladesh. These future undertakings will contribute not only to academic discourse but also to practical solutions that can effectively protect sensitive data and ensure the robustness of the nation's digital platforms.

Acknowledgment

The authors would like to thank the IIT, University of Dhaka for the support to conduct the study.

References

[1] N. James, "Cybersecurity Statistics 2023: Cyber Attacks Per Year & Industry Stats," *www.getastra.com*, Dec. 19, 2022. https://www.getastra.com/blog/security-audit/cyber-security-statistics/ (accessed Jul. 03, 2023).
[2] L. Franceschi-Bicchierai, "Bangladesh government website leaks citizens' personal data," *TechCrunch*, Jul. 07, 2023. https://techcrunch.com/2023/07/07/bangladesh-government-website-leaks-citizens-personal-data/ (accessed Aug. 18, 2023).
[3] "25 Bangladeshi govt, private websites breached by Indian hackers," *Dhaka Tribune*, Aug. 16, 2023. https://www.dhakatribune.com/bangladesh/322710/25-bangladeshi-govt-private-websites-breached-by (accessed Aug. 18, 2023).
[4] "Cyberattack targeting Bangladesh government websites leaves message demanding quota reform," *bdnews24.com*, Apr. 11, 2018. https://bdnews24.com/bangladesh/most-websites-of-bangladesh-government-down (accessed Aug. 18, 2023).
[5] "Myanmar hackers attack Bangladesh govt site," *Dhaka Tribune*, Sep. 09, 2017. https://www.dhakatribune.com/bangladesh/125183/myanmar-hackers-attack-bangladesh-govt-site (accessed Jul. 03, 2023).
[6] D. Alam, T. Bhuiyan, A. Kabir, and T. Farah, "SQLi vulnerabilty in education sector websites of Bangladesh," *2015 Second International Conference on Information Security and Cyber Forensics (InfoSec)*, Nov. 2015, doi: https://doi.org/10.1109/infosec.2015.7435521.
[7] Moniruzzaman, F. Chowdhury, and S. Ferdous, "Measuring Vulnerabilities of Bangladeshi Websites," *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, Feb. 2019, doi: https://doi.org/10.1109/ecace.2019.8679426.
[8] S. Alazmi and D. C. De Leon, "A Systematic Literature Review on the Characteristics and Effectiveness of Web Application Vulnerability Scanners," *IEEE Access*, vol. 10, pp. 33200–33219, 2022, doi: https://doi.org/10.1109/access.2022.3161522.
[9] J. Shahid, M. K. Hameed, I. T. Javed, K. N. Qureshi, M. Ali, and N. Crespi, "A Comparative Study of Web Application Security Parameters: Current Trends and Future Directions," *Applied Sciences*, vol. 12, no. 8, p. 4077, Apr. 2022, doi: https://doi.org/10.3390/app12084077.
[10] E. A. Altulaihan, A. Alismail, and M. Frikha, "A Survey on Web Application Penetration Testing," *Electronics*, vol. 12, no. 5, p. 1229, Mar. 2023, doi: https://doi.org/10.3390/electronics12051229.
[11] Md. A. Masum, Md. R. Istiak Sachcha, and A. Nayem, "Security Analysis of Government & Financial Websites of Bangladesh," *International Journal of Education and Management Engineering*, vol. 12, no. 2, pp. 21–29, Apr. 2022, doi: https://doi.org/10.5815/ijeme.2022.02.03.
[12] "Top Bangladeshi government websites in 2023," *Similarweb*, Jul. 2023. https://www.similarweb.com/website/bangladesh.gov.bd (accessed Jul. 05, 2023).
[13] "Most Visited Investment Websites in Bangladesh 2023," *Semrush*, Jul. 2023. https://www.semrush.com/trending-websites/bd/investment (accessed Jul. 05, 2023).
[14] "Identify technologies on websites," *Wappalyzer*. https://www.wappalyzer.com/ (accessed Jul. 05, 2023).
[15] "wafw00f | Kali Linux Tools," *Kali Linux*. https://www.kali.org/tools/wafw00f/ (accessed Jul. 05, 2023).
[16] "sublist3r | Kali Linux Tools," *Kali Linux*. https://www.kali.org/tools/sublist3r/ (accessed Jul. 05, 2023).
[17] "OWASP ZAP," *Zaproxy*. https://www.zaproxy.org/ (accessed Jul. 05, 2023).
[18] J. Williams, "OWASP Risk Rating Methodology," *owasp.org*. https://owasp.org/www-community/OWASP_Risk_Rating_Methodology (accessed Aug. 18, 2023).
[19] "ZAP – Hash Disclosure," *www.zaproxy.org*. https://www.zaproxy.org/docs/alerts/10097 (accessed Aug. 18, 2023).
[20] "SQL Injection Prevention Cheat Sheet," *Owasp.org*, 2021. https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html (accessed Jul. 05, 2023).
[21] "Content Security Policy (CSP) Header Not Set," *ScanRepeat*. https://scanrepeat.com/web-security-knowledge-base/content-security-policy-csp-header-not-set (accessed Jul. 05, 2023).
[22] "OWASP ZAP – Cross-Domain JavaScript Source File Inclusion," *www.zaproxy.org*. https://www.zaproxy.org/docs/alerts/10017/ (accessed Jul. 05, 2023).
[23] "OWASP ZAP – X-Content-Type-Options Header Missing," *www.zaproxy.org*. https://www.zaproxy.org/docs/alerts/10021/ (accessed Jul. 05, 2023).
[24] "Retrieved from Cache," *ScanRepeat*. https://scanrepeat.com/web-security-knowledge-base/retrieved-from-cache (accessed Jul. 05, 2023).
[25] "User controllable HTML element attribute (potential XSS)," *ScanRepeat*. https://scanrepeat.com/web-security-knowledge-base/user-controllable-html-element-attribute-potential-xss (accessed Jul. 05, 2023).

Authors' Profiles

Md. Asif Khan Rifat is a graduate student at Information Technology (IIT), University of Dhaka. Currently, he is pursuing his Master of Science in Software Engineering. He earned his bachelor of science in Computer Science and Engineering from North South University. His major areas of research interest are machine learning, data mining, web security, and software engineering.

Yeasmin Sultana is studying for her Master's degree in Software Engineering at the University of Dhaka. She completed her bachelor's degree majoring in Computer Science and Engineering from Daffodil International University in the year 2022.

Dr. B. M. Mainul Hossain is a Professor at the Institute of Information Technology (IIT), University of Dhaka, Bangladesh. He received his Ph.D. degree in computer science from the University of Illinois at Chicago, USA. Before that, he earned his Bachelor of Science and Master's degrees from the Department of Computer Science & Engineering, University of Dhaka, Bangladesh. He has the experience of working both in industry and academia. He worked as a Software Engineer at Microsoft Corporation (Redmond, USA) & Accenture Technology Lab (Chicago & California). His core areas of interest are software engineering, security, data mining, and machine learning.

How to cite this paper: Md. Asif Khan Rifat, Yeasmin Sultana, B M Mainul Hossain, "Vulnerabilities Assessment of Financial and Government Websites: A Developing Country Perspective", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.15, No.5, pp. 42-53, 2023. DOI:10.5815/ijieeb.2023.05.05