I. J. Education and Management Engineering, **2022, 2, 21-29**
Published Online April 2022 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijeme.2022.02.03

Security Analysis of Government & Financial Websites of Bangladesh

Md. Asaduzzaman Masum
Department of Computer Science and Engineering, Stamford University Bangladesh
E-mail: masumsubcse@gmail.com

Md. Rishad Istiak Sachcha
Department of Computer Science and Engineering, Stamford University Bangladesh
E-mail: sachchathegreat16@gmail.com

Abu Nayem
Department of Computer Science and Engineering, Stamford University Bangladesh
E-mail: jacknayeem@gmail.com

Received: 17 November 2021; Revised: 09 February 2022; Accepted: 12 March 2022; Published: 08 April 2022

Abstract: The vision 2021 of Bangladesh had to transform into a digital country, where the digital platform was a significant part of it. To make a digital platform, the Bangladesh government announced plans to build web applications in government, non-government, financial, educational and other sectors. By increasing the number of websites, the security risk is growing because of vulnerable coding practices. If those security risks are not fixed, attackers could exploit these vulnerabilities and perform various malpractices like data breaches, injected spam content, spreading viruses, malicious redirects, Denial-of-service, or even website defacements. This paper focuses on vulnerability assessment on Bangladeshi government and financial websites to show the security posture of these sites. This study scanned and analyzed four types of risk alerts High, Medium, Low and Informational using Acunetix and ZAP tools. In addition, the selected top five vulnerabilities are CJ, MC, CSRF, ID and XSS in terms of single vulnerability-type detected for targeted websites. The report has described representing the security condition of Bangladesh official websites. Also, it provided mitigation techniques for these vulnerabilities to avoid security risk, which is less discussed in this country.

Index Terms: Website vulnerabilities, Firewalls, Top five vulnerabilities, Mitigation Techniques, Implementing Secure Websites.

1. Introduction

The use of web applications is constantly increasing in Bangladesh and cyber-crime as well. Recently, we can see a ton of activity of cybercriminals. They are targeting the government, financial, eCommerce, educational and other reputed organizations. According to the security magazine, a single cyber-attack is happening every 39 seconds on average over the world [18]. One of the biggest cyber-attacks happened in Bangladesh called Bangladesh Bank cyber heist. The attempted value was all about US$1 billion from the Federal Reserve Bank of New York account owned by the central bank of Bangladesh [19,20], where five of the thirty-five fraudulent instructions were successful, and Bangladesh lost US$101 million.

According to the Kaspersky Security Bulletin 2020, Bangladesh took 8th position in online infection and 7th local infection compared with other countries. The statistic has shown that 13.75% and 54.74% of users faced a high risk of online infection and local infection [21]. That's mean Bangladesh is not safe yet.

During COVID-19, cybercrime has been increased by 600% [8]. Usually, the attackers find a way to breach the security holes of applications. So, web-based programming languages release their newer version to build an effective security protocol based on their vulnerability. Old versions of plugins or libraries might have flawed and risky than newer ones. Hackers could effortlessly attack those websites. Even if we use up-to-date libraries or plugins, we are not safe yet, because of insecure coding practice. Recent security attacks prove that most Bangladeshi websites are leaving security holes. Therefore, it is necessary to make enough secure web applications to prevent different types of attacks.

Vulnerability assessment is an effective technique to take preventive measures of precision and security. Several research works have considered different kinds of vulnerabilities on Bangladeshi web applications. A few research teams have worked on educational, eCommerce, news, financial site to show the analytical result.

Only they worked on specific vulnerability techniques such as SQLi, XSS or others [1,2,3,4,5,6,7]. Also, they did not provide any mitigation technique. At this point, we have worked in different. The central objective of this research is to discover security holes of the Bangladesh official applications, detect the most significant types of vulnerabilities and their mollification techniques.

In this study, we have used a penetration testing methodology [3]. Using this technique, we have selected our target websites and gathered different kinds of information such as the name of programming language, CMS, web server, firewall. Then, we have find out the subdomains of each target and scan them with our selected vulnerability scanners. After that, we have saved the results and analyzed them thoroughly to discover the top five vulnerabilities of the considered applications. At last, we have shown the mitigation techniques that we implemented in our system. Those techniques anyone can follow to secure their applications.

This paper is organized as follows; the related work is discussed in Section 2. Section 3 is about methodology. Section 4 illustrates the analytical result. Finally, section 5 concludes this paper with future research directions.

2. Related Work

Most of the research has evaluated website vulnerabilities and shared their prevention techniques. Among those, a few studies discussed Bangladeshi website vulnerabilities. This section has discussed those studies.

Delwar Alam et al. explained the various types of SQL injection vulnerabilities on government and non-government websites of Bangladesh [1,2]. Nine hundred applications, which were country code top-level domain (ccTLD) for Bangladesh, were tested by SQL injections. Among Six hundred vulnerable web applications, 510 and 90 applications were vulnerable to GET and POST base. Another analysis was on 359 web applications, where they have got 86% vulnerable applications. Among those vulnerabilities, Low, Medium and High vulnerabilities were 19%, 18% and 63% consistently. The research objective was statistical evaluation.

T. Farah et al. researched SQL injection testing on web applications of Bangladesh [3,4]. In [3], the SQLi testing was on 108 financial and educational web applications, where 60% of websites were inadequate security, 30% were MOD security vulnerability and 10% others. In another paper [4], they assessed XSS and CSRF on 500 websites. The research showed that 75% and 65% of applications were vulnerable to CSRF and XSS, respectively. They described those vulnerabilities and their possible solutions. The paper only considered SQL injection, XSS and CSRF vectors for experimentation purposes.

Moniruzzaman et al. studied the assessment of Bangladeshi web applications, including government, news and media, banks, defence, stock exchange, advertisements, torrents and educational websites [5]. The main objective of this research was to identify maximum vulnerabilities in the selected applications with minimum overhead. This research reported several web applications that suffer from security vulnerabilities seriously and given importance to mitigate.

Rahman et al. conducted an empirical analysis to evaluate the web application vulnerabilities for Bangladeshi e-commerce sites [6]. The paper concluded that CSRF were the most frequent vulnerability for Bangladeshi eCommerce sites and the XSS was the top position for high-level risk.

Totul et al. analyzed and compared web performance issues for Bangladeshi eCommerce sites, where three tools were applied for performance evaluation [7]. They considered Load Time, First Byte, Start Render, First Contentful Paint, Speed Index, Largest Contentful Paint, Cumulative Layout Shift, Total Blocking Time and Time to Interactive parameters for web performance evaluation and comparison purposes. The study concluded that 0.03 and 17.78 seconds were the minima and maximum values for ‘total blocking time’ and ‘load time’ parameters, respectively.

3. Methodology

Web application vulnerabilities cause dangerous consequences for any website or application. Specifically, if the government and financial websites are vulnerable, the cost might be enormously hazardous. This research methodology would help us to find out the different vulnerabilities. We used several tools and techniques to collect data to analyze the security holes. And, that analytical information might help to improve consciousness during web application development and eliminate existing vulnerabilities in present websites. This section has discussed the methods and tools used in this experiment. Three lead steps, Experimental Information Gathering, Website and tools Selection, Environment Setup and Vulnerability Assessment, are used in our methodology. Fig 1 demonstrates the overview of our proposed model.

At first, we selected our target websites and gathered different kinds of information. Secondly, we discovered the subdomains of each target and scanned them with our picked vulnerability scanners. Thirdly, we saved the results and analyzed them thoroughly to detect the top five vulnerabilities. Finally, we provide the mitigation techniques. The technique has been implemented in our system that anyone can follow to secure their website.

3.1. Experimental Information Gathering

In this step, we have collected information about applied firewalls and subdomains in our targeted websites, where the Wappalyzer extension [9], wafw00f [10] and sublit3r tools [11] are used. All these are open-source tools. After that, we assessed the vulnerability using two vulnerability scanners Acunetix and ZAP are more popular and effective [12,13]. The ZAP is open-source, Acunetix is payable, limited features are free. These tools find the vulnerability for each targeted website.

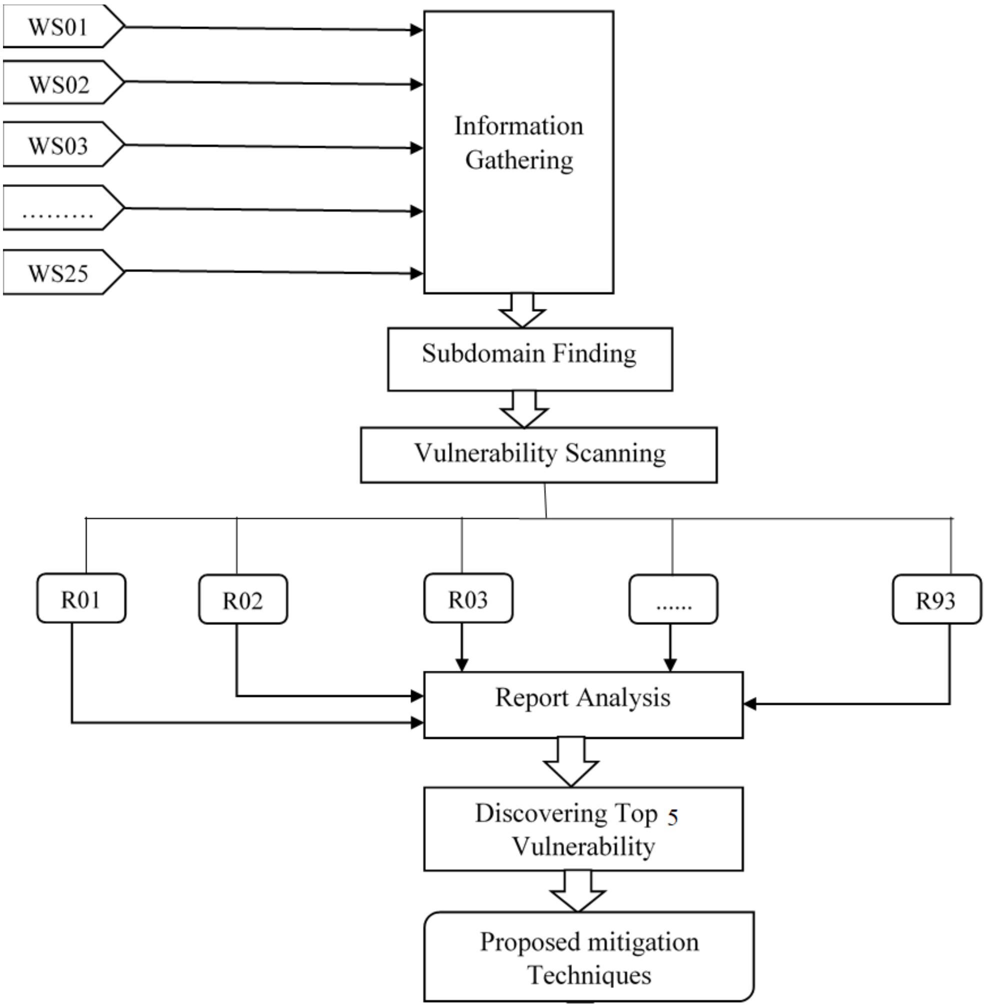

Fig. 1. Overview of the proposed method

3.2. Website and tools Selection

The government activity and finance sectors are significant sectors of any country. If hackers successfully attack their websites, the cost will be huge. So, we selected government and financial organizations' websites to perform vulnerability assessments. Due to confidentiality reasons, we have not disclosed the URL and have used pseudonyms.

We selected the Wappalyzer and Wafw00f as technology detection and firewall detection tools. Sublist3r was used to find out the subdomains of each target. ZAP and Acunetix (trial version) are for scanning vulnerabilities of each website. A short description is given below about these tools.

- Wappalyzer: This is a technology detection tool that finds out the technologies used on the website. It detects the used CMS, framework, platform, JavaScript libraries and many more. This tool could uncover more than a thousand technologies in dozens of categories such as programming languages, analytics, marketing tools, payment processors, CRM, CDN and so on.
- Wafw00f: This is a firewall fingerprinting tool to detect applied firewalls in an application. For this purpose, it sends a simple HTTP request and analyses the response to identify the firewall solutions. If the first attempt is failed, it sends several malicious HTTP requests to detect which firewall it is. If it is not successful yet, it analyses the responses previously returned to guess which security solution is responding to our attacks.

- Sublist3r: This is an open-source tool to find out the subdomains of a website. It enumerates subdomains using OSINT, search engines such as Google, Yahoo, Bing, Baidu, and Ask. Sublist3r also enumerates subdomains using Netcraft, Virustotal, ThreatCrowd, DNSdumpster, and ReverseDNS.
- Acunetix: This is a web application vulnerability scanner, limited features are free. Otherwise, it is premium. It could report on a wide range of web vulnerabilities like SQLi and XSS, directory traversal, CSRF etc. This tool is often used in government, financial and military organizations to perform their security test.
- Zed Attack Proxy (ZAP): This is an open-source vulnerability scanner built by OWASP. It takes URLs as an input and performs vulnerability scanning. It has a database of the fingerprints of the vulnerabilities to match with the database and detect vulnerabilities like SQLi, XSS, CSRF, directory traversal, information disclosure, etc.

3.3. Environment Setup

In this experiment, we used our personal computer, where the operating system is the Kali Linux 2020.3 with 5Mbps internet. The hardware setup was Intel core i3 CPU and 8GB memory, 1TB hard disk. However, the recommended setup is CPU 2.00 GHz Intel Core i5 or higher, RAM 8 GB or higher, GPU isn’t mandatory and more than ten Mbps internet connection.

3.4. Vulnerability Assessment

It is the process of prioritizing, classifying, identifying, and quantifying vulnerabilities in a system. An application vulnerability assessment is a technique to evaluate security risk, which is used to mitigate the possibility of web attacks. There are a lot of testing tools to assess the security risk, such as Acunetix, ZAP, Nikto, etc. Where Acunetix and ZAP are used to perform this task as described below.

- Technology Detection: This is the first step of this research where Wappalyzer, wafw00f, sablist3r are used to gather information about the used technologies of the targets. Wappalyzer is a browser add-on. It detects the technologies used by any web application like a programming language, CDN, web Framework, server etc. It can also help to recognize the vulnerable version.
- Subdomain Finding: After getting the information of technologies, we have selected a tool called sublist3r to find out the subdomains of our targets. The command sublist3r –d target.com is used to find the subdomains.
- Firewall Detection: The wafw00f is a tool used to fingerprint the firewall, where wafw00f target.com is used to send malformed packets to the server. Analyzing the responses from the corresponding requests, it detects the firewall.
- Website Scanning: After getting all the information regarding the target. The Acunetix and ZAP are a tool used for vulnerability assessment. Acunetix categorized the results into four categories, high, medium, low and informational alert based on the vulnerability impact. And ZAP detects the vulnerabilities by matching the fingerprints relay on the database and gives results according to the risk categories. This tool also categorized the results into four categories, high, medium, low and informational alert.

4. Result Analysis

In this section, we provided a thorough analytical report. All data we collected from the Acunetix and ZAP scanners. We classified the risk levels into four categories, High, Medium, Low and Info. Table 1 and Table 2 demonstrated the analyzed result of vulnerabilities and the site-wise vulnerabilities respectively. In the end, we discussed the top five security risks of our targeted websites and their mitigation techniques.

4.1. Alerts Distribution

The alerts of the risk levels have been classified into four categories.

- High Risk: This is the most dangerous category. An attacker can fully compromise a system’s confidentiality, integrity, and availability by exploiting this category. It allows an attacker’s lateral movement and even privilege escalation to another system.
- Medium Risk: It is less dangerous than high-level risks. An attacker can partially compromise a system’s confidentiality, integrity and availability by exploiting this category. Specialized access or user interaction is needed for most of the vulnerabilities of this category to be successful.
- Low Risk: For this kind of security risk, an attacker has limited scope to exploit a system. It is hard for an attacker to exploit. It needs an attacker specialized access, user interaction or circumstances to compromise a system.
- Information: These are now not exploitable. Attackers can get information about a system from it. This information can help an attacker to exploit a system in a later period.

4.2. Detected Risk Alerts

Table 1 represents the risk alerts reports, which we get from two scanning tools Acunetix and ZAP. We divided all of the risk alerts into four types (i) High, (ii) Medium, (iii) Low and (iv) Informational (info).

Table 1. Vulnerabilities detected by Acunetix and ZAP

Site	High		Medium		Low		Info	
	Acunetix	ZAP	Acunetix	ZAP	Acunetix	ZAP	Acunetix	ZAP
WS$_1$	3	0	30	6	26	45	26	19
WS$_2$	0	0	2	3	25	14	9	7
WS$_3$	1	0	6	1	7	21	0	12
WS$_4$	0	0	28	5	7	15	3	7
WS$_5$	1	0	4	3	5	13	1	6
WS$_6$	11	0	30	2	13	42	141	24
WS$_7$	1	0	18	7	18	25	2	11
WS$_8$	0	1	9	3	1	8	58	6
WS$_9$	2	0	16	3	9	16	37	9
WS$_{10}$	0	0	0	1	3	7	0	3
WS$_{11}$	0	0	2	1	6	16	2	6
WS$_{12}$	2	1	60	4	17	20	25	13
WS$_{13}$	0	0	1	1	3	20	3	8
WS$_{14}$	4	1	0	2	37	14	1	4
WS$_{15}$	1	0	0	1	2	3	0	2
WS$_{16}$	4	0	0	1	37	7	1	2
WS$_{17}$	6	0	8	6	47	27	31	10
WS$_{18}$	14	0	161	17	67	63	56	15
WS$_{19}$	2	0	29	4	18	15	11	7
WS$_{20}$	16	0	1	1	19	4	5	2
WS$_{21}$	4	0	2	1	37	9	1	5
WS$_{22}$	28	0	1	2	3	7	4	2
WS$_{23}$	0	0	7	3	4	8	0	2
WS$_{24}$	30	0	1	1	38	7	2	3
WS$_{25}$	3	1	50	8	51	27	52	15

However, Acunetix performed better than ZAP for all types of vulnerability. In the total detection by Acunetix, 133 vulnerabilities were for high risk, 466 for medium, 500 for low and 471 for Info. The lowest and highest number of vulnerabilities detected by this tool, where the High risk is 0 and 30, the Medium risk is 0 and 161, The Low risk is 1 and 67, and The Info risk is 0 and 141. On the other hand, ZAP could find 4, 87, 453, 200 total vulnerabilities are for High, Medium, Low, and Informative, respectively. This tool detected for the High risk are 0 and 1, the Medium risk is 1 and 17, the Low risk is 3 and 63, and The Info risk is 2 and 24, which were the lowest and highest number of vulnerabilities. Moreover, WS15 and WS18 websites detected the least and most vulnerabilities applications compared with Acunetix and ZAP.

4.3. Website's Vulnerability Breakdown

The breakdown of detected vulnerabilities by Acunetix and ZAP have been shown in Table 2, where the first column and table heading presents the websites (WS) and vulnerability names respectively. The considered vulnerabilities for this experimentation are (i) SQL Injection (SQLi), (ii) Cross-Site Scripting (XSS), (iii) File upload (FU), (iv) Session Fixation (SF), (v) Weak Password (WP), (vi) HTML Form Redirection (HFR), (vii) File Inclusion (FI), (viii) Denial of Service (DOS), (ix) Cross-Site Request Forgery (CSRF), (x) Same Site Scripting (SSS), (xi) Information Disclosure (ID), (xii) Misconfiguration (MC), (xiii) Vulnerable Library (VL), (xiv) Breach Attack (BA), (xv) HTTP Parameter Pollution (HPP), (xvi) Clickjacking (CJ), (xvii) Directory Listing (DL), (xviii) Poor Encryption (PE). The number '1' represents the existent of vulnerabilities, and the '-' narrates the type of vulnerabilities that do not exist.

Table 2. Vulnerability Breakdown

Site/ vuln	SQLi	XSS	FU	SF	WP	HFR	FI	DOS	CSRF	SSS	ID	MC	VL	BA	HPP	CJ	DL	PE
WS01	-	1	1	-	-	1	-	1	1	-	1	1	1	-	1	1	1	1
WS02	-	-	-	-	-	-	-	-	-	-	1	1	-	1	-	1	-	1
WS03	-	-	-	1	-	-	-	1	1	-	-	1	-	-	-	-	1	-
WS04	-	-	-	-	-	-	1	-	1	1	1	1	-	1	-	-	-	-
WS05	-	-	-	-	-	-	-	1	-	1	-	1	-	-	-	1	-	-
WS06	1	1	-	-	-	-	-	1	1	1	1	1	-	-	-	-	-	1
WS07	-	-	-	-	-	-	1	1	1	1	-	1	-	-	-	-	-	-
WS08	-	-	-	-	-	-	-	-	1	-	1	-	-	-	-	1	-	-
WS09	-	-	-	-	-	-	-	-	1	-	1	-	-	-	-	-	-	1
WS10	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	1	-	-
WS11	-	-	-	-	-	-	1	-	1	-	1	-	-	1	-	1	-	-
WS12	1	1	1	-	-	-	1	1	1	-	-	1	-	-	-	1	-	-
WS13	-	-	-	-	-	-	1	-	-	-	-	1	-	-	-	1	-	-
WS14	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-
WS15	-	-	-	-	-	-	-	-	-	-	1	1	-	-	-	1	-	-
WS16	-	1	-	-	-	-	-	-	-	-	1	-	-	-	-	1	-	-
WS17	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-
WS18	1	1	1	-	1	-	-	1	1	-	1	1	1	-	-	1	1	1
WS19	-	-	-	-	-	-	-	1	1	-	1	1	-	-	-	1	-	1
WS20	-	1	-	-	-	-	-	-	1	-	-	1	-	-	-	1	-	-
WS21	-	1	-	-	-	-	-	-	-	-	1	1	-	-	-	1	-	-
WS22	-	-	-	-	-	-	-	-	1	-	1	1	-	-	-	1	-	-
WS23	-	-	-	-	-	-	-	-	-	-	1	1	-	-	-	1	-	-
WS24	-	1	-	-	-	-	-	-	1	-	-	1	-	-	-	1	-	-
WS25	1	-		-	1	-	-	1	1	-	-	1	-	-	-	1	-	1
Total	4	11	3	1	2	1	5	9	15	4	14	19	2	3	1	20	3	7

In Table 2, the scanner detected the maximum clickjacking (CJ) type for 20 sites. Additionally, misconfiguration vulnerability was in 19 sites. On the other hand, CSRF, ID, XSS, DOS and PE types were detected for 15, 14, 11, 9 and 7 sites respectively. However, the tool could detect only 1 type of vulnerability for SF, HFR and HPP. According to analyzing results, the top 5 detected vulnerabilities are (i) CJ (20), (ii) MC (19), (iii) CSRF (15), (iv) ID (14), (v) XSS (11) in terms of single vulnerability-type detected for selected websites.

4.4. Top 5 Security Risks and Mitigation Techniques

According to the demonstrated results, the top 5 security risks for scanned web apps are (I) CJ, (ii) MC, (iii) CSRF, (iv) ID, and (v) XSS. We will discuss below the mitigation technique for these top five.

Table 3. Detected top 5 vulnerabilities and mitigation techniques.

Sl.	Vulnerability Type	Description	Mitigation
1.	ClickJacking	In this attack, a hacker uses a trick to click on actionable content on a hidden website by clicking on other content in a malicious website [14].	• Set X-Frame-Options: deny instead of same-origin or allow in the httpd.conf file. This might be the best practice. • Another way is to set Content-Security-Policy: frame-ancestors 'none; in the httpd.conf or .htaccess file [14].
2.	Misconfiguration	Attackers will often attempt to exploit unpatched flaws or access default accounts, unused pages, unprotected files and directories and so on to gain unauthorized access or knowledge of the system [14].	• Disable all the unnecessary features & components. • Configure all the permissions properly like, S3 bucket permission on a cloud. • Don't use default credentials like username: admin, password: admin. • Update all the technologies regularly [14].
3.	Cross-Site Request Forgery	In this attack, an attacker forces the browser to make an unintended request on behalf of the victim [23].	• Use csrf token in each form which creates token randomly. • Pass the token as a hidden input type [14].
4.	Information Disclosure	It occurs when a website unintentionally reveals sensitive information to its users. Depending on the context, websites may leak different pieces of information to a potential attacker [14].	• Implement proper access control & authorization. • Avoid storing sensitive data. • Don't hard code any sensitive data, even in the comment form. • Use custom error page [15].
5.	Cross-Site Scripting	XSS allows attackers to execute malicious scripts in the victim's browser, where attackers could hijack user sessions, deface websites, or redirect the user to malicious websites [16].	• Encode the user-supplied data. HTML encoding is recommended to mitigate this vulnerability [10]. • Setup Content-Type and X-Content-Type-Options headers in the httpd.conf or .htaccess file. • Set HTTP only flag [17].

In addition to the above mitigation techniques, there are some other headers like Strict-Transport-Security, Content-Security-Policy, Referrer-Policy, Permission-Policy, which should be configured. Avoid using dangerous function like eval () & passthru(). Also, disable allow_url_include and allow_url_fopen in the php.ini file.

Most of the applications of our research use PHP or are somehow related to PHP. So, we have shown the mitigation techniques for the PHP language. For other languages, the mitigation techniques are almost the same. Some functionalities are different for other languages and will need to do the similar thing but can be a little bit different way. In addition, some mitigation technique which called SQLIA detectors. Those detectors could detect vulnerable source of SQLi [22].

5. Conclusion

Several websites accomplish most of the activities. They perform for several reasons, such as sensitive information stored in a server. This paper demonstrates the security holes in the official site of Bangladesh. The system first scanned and analyzed four types of risk alerts (i) high, (ii) medium, (iii) low and (iv) informational using Acunetix and ZAP tools for each website. Our scanned report shows that Acunetix performed better than ZAP. Among the alert results, many web apps show 0, which indicates either the scanner could not scan efficiently or the site does not contain any alert. Besides the risk-alerts record analysis, the research demonstrated the vulnerability breakdown, where we selected the top five rigorous vulnerabilities by analyzing the obtained security-related records. We also showed mitigation techniques for these vulnerabilities as if the Bangladesh government could focus on those issues and be more conscious of further development. In this initial stage, we have scanned a few websites. And we have planned to continue our exploration in other categories like educational, e-commerce websites as future work.

References

[1] Alam D, Bhuiyan T, Kabir MA, Farah T. SQLi vulnerabilty in education sector websites of Bangladesh. In2015 Second International Conference on Information Security and Cyber Forensics (InfoSec) 2015 Nov 15 (pp. 152-157). IEEE.

[2] Alam D, Kabir MA, Bhuiyan T, Farah T. A case study of sql injection vulnerabilities assessment of. bd domain web applications. In2015 Fourth International Conference on Cyber Security, Cyber Warfare, and Digital Forensic (CyberSec) 2015 Oct 29 (pp. 73-77). IEEE.

[3] Farah T, Alam D, Kabir MA, Bhuiyan T. SQLi penetration testing of financial Web applications: Investigation of Bangladesh region. In2015 World Congress on Internet Security (WorldCIS) 2015 Oct 19 (pp. 146-151). IEEE.

[4] Farah T, Shojol M, Hassan M, Alam D. Assessment of vulnerabilities of web applications of Bangladesh: A case study of XSS & CSRF. In2016 sixth international conference on digital information and communication technology and its applications (DICTAP) 2016 Jul 21 (pp. 74-78). IEEE.

[5] Moniruzzaman M, Chowdhury F, Ferdous MS. Measuring vulnerabilities of bangladeshi websites. In2019 International Conference on Electrical, Computer and Communication Engineering (ECCE) 2019 Feb 7 (pp. 1-7). IEEE.

[6] Rahman MA, Amjad M, Ahmed B, Siddik MS. Analyzing web application vulnerabilities: an empirical study on e-commerce sector in Bangladesh. InProceedings of the international conference on computing advancements 2020 Jan 10 (pp. 1-6).

[7] Hossain M, Hassan R, Amjad M, Rahman M. Web Performance Analysis: An Empirical Analysis of E-Commerce Sites in Bangladesh. International Journal of Information Engineering & Electronic Business. 2021 Aug 1;13(4).

[8] “U.N. Official Warns Cybercrime Up 600% During COVID-19 Pandemic”. Available: https://www.newsy.com/stories/u-n-warns-cybercrime-up-600-during-covid-19-pandemic/ [Accessed: June 2020]

[9] Lizonczyk P. CERN Web Application Detection. Refactoring and release as open-source software. 2015 Aug 28.

[10] Ted Holland, “Understanding IPS and IDS: Using IPS and IDS together for Defense in Depth”. Available https://www.sans.org/reading-room/whitepapers/detection/understanding-ips-ids-ips-ids-defense-in-depth-1381 [Accessed: February 2004].

[11] Li Y, Cheng J, Huang C, Chen Z, Niu W. NEDetector: Automatically extracting cybersecurity neologisms from hacker forums. Journal of Information Security and Applications. 2021 May 1;58:102784.

[12] Kals S, Kirda E, Kruegel C, Jovanovic N. Secubat: a web vulnerability scanner. InProceedings of the 15th international conference on World Wide Web 2006 May 23 (pp. 247-256).

[13] Bennetts S. Owasp zed attack proxy. AppSec USA. 2013.

[14] OWASP, “Top 10 Web Application Security Risks”, Available: https://owasp.org/www-project-top-ten/, [Accessed: April. 10, 2021].

[15] Netsparker, “Netsparker official Website”, Available: https://www.netsparker.com. [Accessed: Feb. 19, 2021].

[16] Portswigger, “Web Security Academy”, Available: https:// portswigger.net/web-security/. [Accessed: Feb 2, 2021].

[17] Baloch R. Ethical hacking and penetration testing guide. CRC Press; 2017 Sep 29.

[18] “Hackers Attack Every 39 Seconds” Available: https://www.securitymagazine.com/articles/87787-hackers-attack-every-39-seconds. [Accessed: Jan 2021]

[19] “The great Bangladesh cyber heist shows the truth is stranger than fiction” Available: https://www.dhakatribune.com/uncategorized/2016/03/12/the-great-bangladesh-cyber-heist-shows-truth-is-stranger-than-fiction [Accessed: Jan 2021]

[20] “Congresswoman wants probe of ‘brazen’ $81M theft from New York Fed” Available: https://nypost.com/2016/03/22/congresswoman-wants-probe-of-brazen-81m-theft-from-new-york-fed/ [Accessed: Feb 2, 2021]

[21] “Kaspersky Security Bulletin 2020. Overall statistics for 2015” Available: https://go.kaspersky.com/rs/802-IJN-240/images/KSB_statistics_2020_en.pdf [Accessed: Feb 2, 2021]

[22] Kumar P, Katti CP. A Parallel-SQLIA Detector for Web Security. International Journal of Information Engineering & Electronic Business. 2016 Mar 1;8(2).

[23] Batarfi OA, Alshiky AM, Almarzuki AA, Farraj NA. Csrfdtool: Automated detection and prevention of a reflected cross-site request forgery. International Journal of Information Engineering and Electronic Business. 2014 Oct 1;6(5):10.

Authors’ Profiles

Md. Asaduzzaman Masum was born in Barishal, Bangladesh. He has completed his BSc in CSE at Stamford University Bangladesh in 2021. His research interests are in Security, Data Analysis & Networking.

Md. Rishad Istiak Sachcha was born in was born in Netrokona, Bangladesh.. He has completed his BSc in Computer Science and Engineering from Stamford University Bangladesh. His research interest is on Information Security, Cryptography & Digital Forensics.

Abu Nayem was born in Noakhali, Bangladesh. He has completed his BSc in CSE at Stamford University Bangladesh in 2019. Currently, he is working on NLP and HCI. His research interest is in Networking, Data Science and Machine Learning.

How to cite this paper: Md. Asaduzzaman Masum, Md. Rishad Istiak Sachcha, Abu Nayem, " Security Analysis of Government & Financial Websites of Bangladesh", International Journal of Education and Management Engineering (IJEME), Vol.12, No.2, pp. 21-29, 2022. DOI: 10.5815/ijeme.2022.02.03