Vulnerabilities that threaten web applications in Afghanistan

Sayed Mansoor Rahimy
Dept of Computer Science, Salam University
Kabul, Afghanistan
sayedmansoorrahimy@gmail.com

Dr. Sayed Hassan Adelyar
Dept of Software Engineering, Kabul University
Kabul, Afghanistan
sadelyar@gmail.com

Said Rahim Manandoy
Dept Computer Science, Salam University
Kabul, Afghanistan

Corresponding author: saidrahim.nwaddm@gmail.com

Abstract

Familiarizing web developers with different types of vulnerabilities lead to the creation of secure web applications. In the last few decades there has been considerable interest in web hacking which leads to different types of web attacks that can cause financial damages, privacy loss, data loss and life-threatening situations. The aim of this study is to discover the most common web vulnerabilities that exist in Afghanistan's web applications. We conducted this study by using Netsparker, Skipfish, and Acunetix web vulnerability scanners with the standard web vulnerability assessment (WVA) method. The result shows that almost all the web applications in Afghanistan are vulnerable to different types of cyber-attacks. A total of 997 instances of various types of vulnerabilities detected on 109 web applications from three different domains. This study presents 25 common vulnerabilities, which is more than prior studies. The results of this study familiarize web developers to the most common vulnerabilities that can exist in a typical web application. Therefore, this study will encourage them to take these vulnerabilities into considerations during software development life cycle.

Keywords: Cyber-attacks, vulnerability assessment, web vulnerability scanners, web applications.

1. Introduction

Web applications become one of the most dominant technologies for delivering dynamic services over the Internet. They are used for delivering various types of services such as e-government, financial, social, Learning and Management Systems (LMSs). For this reason, a huge amount of sensitive data is exchanged through web applications. Web applications are

cross-platform, responsive, interactive, remotely accessible, fast deployable, and user friendly. Web platform is composed of different parts. Web server provides the web application services. Web client access the web application via HTTP protocol in a web browser. A wide range of technologies are available for web application development. At the server side technologies such as PHP, ASP, or CGI can be used. At the client side technologies such as HTML, CSS, Java Script, and Flash can be used (Tatnall, 2009).

Web applications are facing a huge number of security challenges and threats due to a wide range of available technologies for their development and their complex infrastructure for hosting and deployment. As a result, developing and deploying a secure web application is rigid. Therefore, web security is a vital component to be considered by all organizations over the world.

Here in Afghanistan, most of the web designers and developers focus on product functionality and Quality of Experience (QoE) rather than security requirements and best practices. Lack of security awareness and experts in the organizations in Afghanistan lead to a wide range of security breaches in their web applications. Lack of due diligence in most of the web users from cyber-attacks cause them to loss their sensitive information and even threatening their life. Consequently, we assume that a high number of web vulnerabilities exist in most of the web applications in Afghanistan.

To the best of our knowledge there are no studies in the literature regarding vulnerability assessment and security of web applications in Afghanistan. Therefore, the aim of the present work is to discover the most common web vulnerabilities in mostly used Afghani web applications. The results of this study provide web designers and developers a productive method for web vulnerability assessment at minimum cost, time and effort. Moreover, it will help them to be aware of some common web vulnerabilities. The following are the three most important research questions, which are answered in this paper:

(i). What are the most common web vulnerabilities in Afghani web applications?

(ii). Which security vulnerability assessment method is suitable for identifying these vulnerabilities?

(iii). What are the threats that are associated with the most common detected vulnerabilities?

The remaining parts of the paper are structured as follows. Section II provides related works and additional information relevant for the vulnerabilities of web application. Section III presents the methodology and tools used for vulnerability assessment of web application. Section IV presents the result of vulnerability assessment. Section V presents a detailed discussion and implications. Finally, Section VI concludes this paper by summarizing the research work, giving the contributions achieved and

showing directions for future work.

2. Literature Review

In this era of digitalization, most of the businesses are relying on web applications. Service providers use web applications to communicate with their subscribers. Therefore, web applications become interesting targets for most of the attackers on the Internet. In the following sub sections, we briefly review some of the existing literatures regarding web application security vulnerability assessment and web vulnerability scanners.

2.1 Web Application Vulnerability Assessment

Vulnerability assessment is a proactive approach through which we can identify and scan for the existing vulnerabilities in the system before they could be exposed by the attackers with malicious intents (Devi et al., 2020; Ansari, 2015). There are different methods for identifying vulnerabilities. Static analysis, attack graph analysis, and vulnerability scanners are the most well-known methods for vulnerability assessment (Ansari, 2015). Static analysis analyze the structure of the program and the code of the program to detect the flaws. Some techniques used in static analysis are lexical analysis, type reference, constraint analysis and many more (Ansari, 2015). Attack graph analysis represents all the paths followed by an attackers to achieve their desired goals. This method is used for identifying vulnerabilities inside a network. For instance, some techniques which are used for generating the attack graphs are: Clustered adjacency matrix, hierarchical aggregation, minimization analysis, ranking graphs, and game theoretic (Ansari, 2015). Vulnerability scanners are the software tools used to identify vulnerabilities in a network system and/or in a software application (Ansari, 2015). There are different vulnerability scanners used for network vulnerability scanning, operating system vulnerability scanning and web vulnerability scanning.

Farah and et al. performed a black-box testing in (2016) to identify XSS and CSRF vulnerabilities in 500 Bangladeshi web applications. This study showed that 30% of the Bangladeshi web applications are vulnerable to XSS and CSRF attacks. From 500 web applications, 335 of them were found vulnerable to either XSS or CSRF, or to both attacks. Their results has shown that, about 65% of the 335 of the web applications were vulnerable to XSS attacks and 75% of them were vulnerable to CSRF attacks.

Moniruzzaman and et al. conducted a black box and white box testing research in (2019) on identifying common vulnerabilities in Bangladeshi websites. The aim of this study was to represent a framework for identifying maximum vulnerabilities at minimum cost and effort. They considered six different attack vectors which are: SQLi, XSS, BAS, CSRF, Unusual

Ports, and TLS. Black box testing conducted with the help of Kali Linux penetration testing tools and white box testing conducted with the help of static code analysis techniques. In this study they found that 36% of the websites in Bangladesh are secure and 64% of the them are running with various vulnerabilities.

In (2018), Ali and Murah analyzed security of 16 Libyan governmental websites. This study proposed a safety classification matrix for the 16 websites using four safety categories: safe, somewhat unsafe, unsafe, and highly unsafe. To classify a website in one of these safety level, they first assessed the website for vulnerability using Netsparker and Acunetix. Secondly, they determined whether a sensitive information is encrypted or not during transactions. Finally, they evaluated SSL encryption using Qualys SSL Labs tool. They analyzed one website as highly unsafe, six websites as unsafe, eight websites as somewhat unsafe, and one website as safe.

Nirmal K and et al. stated in (2018) that it is very critical to hardened web applications due to their existence in various businesses. This paper focuses on performing vulnerability assessment and penetration testing during various phases of Software Development Life Cycle (SDLC). Security consideration and best practices should be embedded in each phase of the web application development life cycle.

In (2020), Sri Devi and Kumar executed a vulnerability analysis on 100 websites. This study used Nikto and OWASP Zed Attack Proxy (ZAP) vulnerability scanners and provides a comparison of these scanners. The study shows that both vulnerability scanner identifies various vulnerabilities. There are some vulnerabilities that detected by the Nikto but not by the ZAP and vice versa. Nikto provides some additional information such as server, SSL information and ciphers.

2.2 Web Application Vulnerability Scanners

Web vulnerability scanners are used to identify various flaws and weaknesses such as misconfigurations, outdated files, and common vulnerabilities that can be found in a web application. Various web vulnerability scanners are available in the market. Some of the most well-known web vulnerability scanners are: Netsparker, Acunetix, Nessus, Nikto, Skipfish, Dirbuster, Burp Suite, Vega, OpenVAS, ZAP Proxy, Sqlmap, W3af, Xsser, and many more.

Bairwa and et al. have conducted a comparative study on five vulnerability scanners in (2014). Their observation has shown that different scanners identify different types of vulnerability but a single tool is not capable of detecting all types of vulnerabilities. They identified the capability of each vulnerability scanner by running each one of them against a number of web applications. They highlighted that Nessus is the only scanner that has detected most of the vulnerabilities followed by Acunetix and Burp Suite.

In (2017), Rajan and Erturk conducted a case study on Acunetix WVS (Web Vulnerability Scanner). In this study they focused on how important it is to scan web application for its vulnerabilities with the help of WVSs. This study has shown that WVSs help to speed up the web applications vulnerability scanning process.

In (2015), Patil and Gosavi presents a vulnerability scanning system architecture based on HTTP methods. The ingredients of the system are: URL Crawling, Domain Reputation, CMS Scan, URL Scan, Search Engine, Remote Site and 3rd Party Databases.

In (2017), Huang and et al. introduced a new vulnerability scanner, VulScan. VulScan automatically generate test data and can discover injection and cross-site scripting (XSS) vulnerabilities by using penetration testing and evasion techniques. This study also proposes three main categories of countermeasures for mitigating SQL injections and XSS attacks. The countermeasures are secure implementation, defense mechanism deployment and Penetration Testing.

3. Methodology

An empirical research study was conducted based on a standard vulnerability assessment method described in the literature (Qiangain et al., 2014; Antunes et al., 2009; Felderer et al., 2016; Singh, 2020). In brief, this section reviews the method with some minor modifications. The vulnerability assessment was conducted on 109 web applications from 3 top level governmental (.gov.af), educational (.edu.af), and commercial (.com.af) domains. The vulnerability assessment was carried out in four steps as shown in *Figure 1*.

Step 01 – Reconnaissance

It was carried out to identify the targets and gather as much information as possible from targeted web applications. The black box testing techniques are used in this step. Tools that were used for this step are the Whois, Dig, Nslookup, theHarvester, Robtex, and Netcraft.

Step 02 – Enumeration and Scanning

The purpose of this step is to enumerate and scan for information such as the web server, underlying operating system, virtual host environment, load balancers, and proxies. The tools that were used in this step are the nmap, recon-ng, Knockpy, and NetCat.

Step 03 – Vulnerability Scanning

This step was carried out in order to find vulnerabilities in the target web applications. Using a single vulnerability scanner can lead to false positive results. Therefore, three different web vulnerability scanners were used. The Netsparker and Acunetix are commercial and have a nice graphical user interface. Skipfish is a free command line tool available in Kali Linux.

Step 04 – Vulnerability Analysis and Reporting

This step provides an in depth analysis and statistics of the detected web vulnerabilities. Furthermore, a detailed descriptions of the vulnerabilities are presented and reported to all governmental, educational, and commercial organizations.

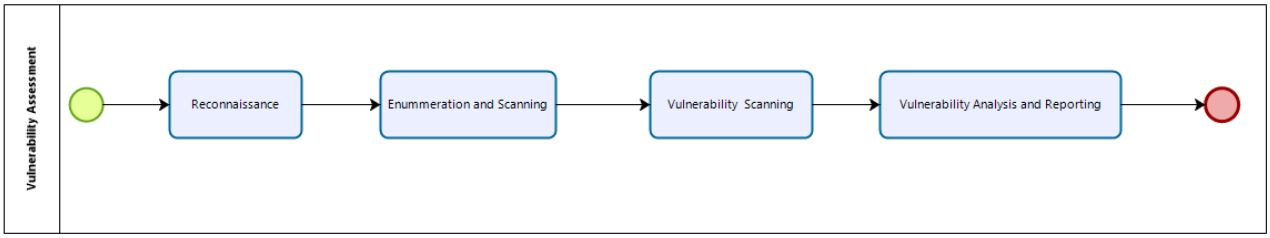

Fig. 1- Vulnerabiltiy Assessment Method

4. Result

In this study, a total of 109 web applications from three different domains were selected in reconnaissance phase. From 109 web applications, 62 of them are governmental, 39 of them are educational and 8 of them are commercial. *Figure 2* shows the percentage of these three different types of web applications for various domains. Significantly, 997 instances of different types of vulnerabilities detected by Netsparker, Skipfish, and Acunetix WVSs in the vulnerability scanning phase. These WVSs classified vulnerabilities according to CVE and CVSS into High, Medium, Low and Informational levels. From 997 instances of vulnerabilities, 86 instances are High level, 167 instances are Medium level, 311 instances Low level and 433 instances are informational level. *Figure 3* presents the dominance of these levels in percentages using pie chart.

In this study, we found 55 different forms of vulnerabilities. Among them, some of them have high frequency. *Table 4* presents some common of them that exist across multiple web applications.

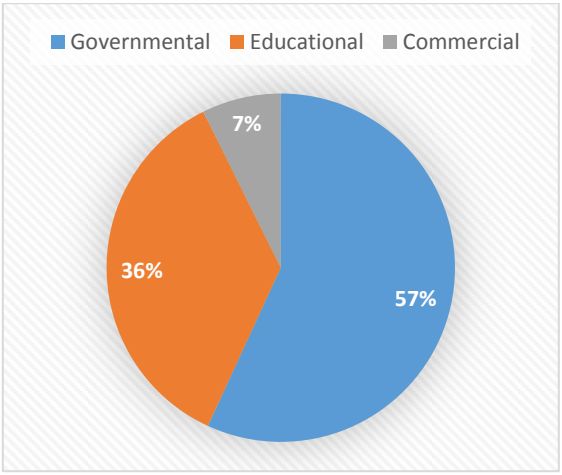

Fig. 2 – Percentages of Web Applications

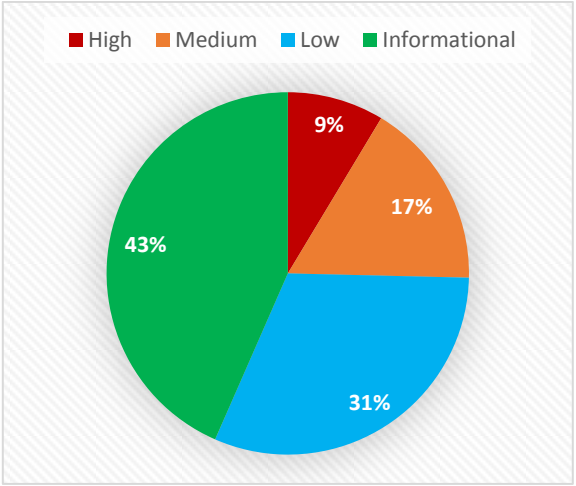

Fig. 3 – Percentages of Vulnerabilities

Table 1 - Common Web Vulnerabilities

Vulnerability	**Level of Severity**	**Instances**	**Number of Websites**
Cross site scripting	High	73	5
SQL injection	High	3	1
Microsoft IIS tilde directory enumeration	High	6	2
Long password denial of service	High	1	1
Elasticsearch service accessible	High	1	1
Vulnerable JavaScript libraries	Medium	48	27
TLS 1.0 enabled	Medium	24	24
User credentials are sent in clear text	Medium	12	12
Slow HTTP Denial of Service Attack	Medium	11	11
Source code disclosures	Medium	8	8
TLS/SSL Sweet32 attack	Medium	7	7
TLS/SSL Weak Cipher Suites	Medium	7	7
Clickjacking: X-Frame-Options header missing	Low	60	60
Unencrypted connection	Low	45	45
HSTS not implemented	Low	32	32
Cookies with missing, inconsistent or contradictory properties	Low	31	31
Login page password-guessing attack	Low	26	26
Cookies without HttpOnly flag set	Low	24	24
Cookies without Secure flag set	Low	23	23
Insecure Referrer Policy	Informational	76	76
Content Security Policy (CSP) not implemented	Informational	74	71
No HTTP Redirection	Informational	38	38
Outdated JavaScript libraries	Informational	38	23
Subresource Integrity (SRI) not implemented	Informational	30	27

5. Discussions

The aim of this study was to discover the most common web vulnerabilities in mostly commonly visited web applications in Afghanistan. The results highly supported our hypothesis. It was assumed that a high numbers of web vulnerabilities exist in most of the web applications in Afghanistan. While not all of the vulnerabilities were high level, the overall results presents a large numbers of vulnerabilities across multiple web applications in various domains.

A similar study is conducted by Ali and Murah in (2018). They discovered a total of 522 instances of different types of vulnerabilities in 16 Libyan governmental websites by using Netsparker and Acunetix WVSs. We discovered a total of 997 instances of different types of vulnerabilities on 109 web applications using Netsparker, Skipfish,

and Acunetix WVSs. Thus, the number of vulnerabilities in Libyan governmental websites are more than the Afghani websites. However, Ali and Murah presented 9 common web vulnerabilities in 16 Libyan governmental websites and we presented 24 almost different common web vulnerabilities in 109 web applications.

With the existence of these vulnerabilities in web applications, a huge set of threats can be launched. Here, we use STRIDE threat modeling to categorize these vulnerabilities (Shostack, 2014). For example, the existence of SQL Injection vulnerability in a web applications could cause to a massive financial loss (tampering). *Table 5* shows vulnerabilities and their related threats. Furthermore, additional study is required to find measures and mitigation techniques to defend against these threats.

As mentioned in the Introduction, that most of the web designers and developers focus on product functionality rather than security considerations. The results of this study will familiarize and encourage them to take some security considerations during software development life cycle.

6. Conclusion

Vulnerability assessment completed against 109 web applications and totally 997 different instances of web vulnerabilities discovered. Additionally, a detailed description of these vulnerabilities were reported and presented to all related organizations. Furthermore, a statistical analysis of these vulnerabilities have been presented using pie charts and tables. The result of this study will help web designers and developers to build secure web applications. Moreover, the results of the identified vulnerabilities will be shared with their corresponding web applications owners to be addressed soon. Although, our results are limited to Afghani web applications. However, it remains to be further clarified whether our results could be applied to other web applications in other countries. Future work will mainly cover the presentation of countermeasures that will help to mitigate these web vulnerabilities.

Table 2 – Associated threats with common detected vulnerabilities

Vulnerability	**Threats**
Cross site scripting	Tampering
SQL injection	Tampering
Microsoft IIS tilde directory enumeration	Information Disclosure
Long password denial of service	DoS
Elasticsearch service accessible	Information Disclosure
Vulnerable JavaScript libraries	One or more
TLS 1.0 enabled	Information Disclosure

User credentials are sent in clear text	Spoofing
Slow HTTP Denial of Service Attack	DoS
Source code disclosures	Information Disclosure
TLS/SSL Sweet32 attack	Information Disclosure
TLS/SSL Weak Cipher Suites	One or more
Clickjacking: X-Frame-Options header missing	Spoofing, Information Disclosure and Elevation of Privileges
Unencrypted connection	Information Disclosure
HSTS not implemented	Information Disclosure
Cookies with missing, inconsistent or contradictory properties	Elevation of Privileges
Login page password-guessing attack	Spoofing
Cookies without HttpOnly flag set	Tampering
Cookies without Secure flag set	Information Disclosure
Insecure Referrer Policy	Information Disclosure
Content Security Policy (CSP) not implemented	Tampering
No HTTP Redirection	Information Disclosure
Outdated JavaScript libraries	One or more
Sub Resource Integrity (SRI) not implemented	Spoofing

5. Reference

Ali, A. A. and Murah, M. Z. (2018) Security Assessment of Libyan Government Websites. IEEE Cyber Resilience Conference (CRC)

Alzahrani, A., Alqazzes, A., Fu, H., Almashfi, N., and Zhu, Y. () Web Application Security Tools Analysis. IEEE 3rd International Conference On Big Data Security on Cloud.

Ansari, J. A. (2015) Web Penetration Testing with Kali Linux. Packet Publishing Ltd, Livery Place. Pp. 54 – 257

Antunes, N., and Vieira, M. (2009) Comparing the Effectiveness of Penetration Testing and Static Code Analysis on the Detection of SQL Injection Vulnerabilities in Web Services. 15th IEEE Pacific Rim International Symposium on Dependable Computing.

Bairwa, S., Mewara, B., and Gajrani, J. (2014) Vulnerability Scanners: A Proactive Approach to Access Web Application Security. International Journal on Computational Sciences & Applications (IJCSA) 4(1):

Devi, R. S. and Kumar, M. M. (2020) Testing for Security Weakness of Web Applications using Ethical Hacking. Proceedings of the Fourth International Conference on Trends in Electronics and Informatics.

Farah, T., Shojol, M. Hassan, Md. M. and Alam, D. (2016) Assessment of vulnerabilities of web applications of Bangladesh: A case study of XSS & CSRF. Sixth International Conference on Digital Information and Communication Technology and its Applications (DICTAP).

Felderer, M., Buchler, and M., Johns (2016) Security Testing: A Survey . The University of Sheffield.

Haung, HC., Zhang, ZK., Cheng, HW., and Shieh, S., W. (2017) Web Application Security: Threats, Countermeasures, and Pitfalls. IEEE Computer Society

Jain, T., and Jain, N. (2019) Framework for Web Application Vulnerability Discovery and Mitigation by Customizing Rules through ModSecurity. 6th International Conference on Signal Processing and Integrated Networks (SPIN).

K, N. Janet, B., and Kumar, R. (2018) Web Application Vulnerabilities – The Hacker's Treasure. Proceedings of the International Conference on Inventive Research in Computing Applications (ICIRCA)

Moniruzzaman, Md., Chowdhury, F., and Ferdous, Md. S. (2019) Measuring Vulnerabilities of Bangladishi Websites. International Conference on Electrical, Computer and Communication Engineering (ECCE)

Patil, H., P., and Gosavi, B. (2015) Web Vulnerability Scanner by Using HTTP Method. , International Journal of Computer Science and Mobile Computing, 4(9): 255-260

Qianqian, W., and Xiangjun, L. (2014) Research and Design on Web Application Vulnerability Scanning Service. IEEE 5th International Conference on Software Engineering and Service Science

Rajan, A., and Erturk, E. (2017) Web Vulnerability Scanners: A Case Study. arXivLabs Cryptography and Security.

Shostack, A. (2014) Threat Modeling Designing for Security. John Wiley & Sons, Inc., Indianapolis Indiana Pp. 61-85.

Singh, H., and Sharma, H. (2020) Hands-On Web Penetration Testing with Metasploit. Packet Publishing, Liver Place, Pp. 2-20

Tatnall, A. (2009) Web Technologies: Concepts, Methodologies, Tools, and Applications (Volume 1). In: Corazza, L., ICT and Interculture Opportunities Offered by the Web. Pp 1-11. Victoria University, Australia.

Wang, B. Liu, L., Li, F., Zhang, J., Chen, T., and Zou, Z. (2019) Research On Web Application Security Vulnerability Scanning Technology. IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC).