Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 3, Issue 8, 2025

ENHANCING AND EVALUATING THE SECURITY VULNERABILITIES OF E-GOVERNMENT PLATFORM

Muhammad Janas khan[1], Shirjan Mujahid[2]

Department of Computer Science, Bakhtar University Afghanistan
Email: muhammadjanaskhan@gmail.com, mujahid.shirjan@gmail.com

DOI https://doi.org/10.5281/zenodo.17285912

Keywords
Web Security, Web Usability, Web Accessibility, Vulnerabilities. governmental websites, Vulnerability assessment and penetration testing.

Article History
Received on 08 June 2025
Accepted on 08 July 2025
Published on 16 Aug 2025

Corresponding Author: *
Muhammad Janas khan

Abstract
Government websites are vital for connecting with citizens and providing transparent services. However, Afghanistan's government sites are frequently hacked because they are built and deployed insecurely. They are also difficult to use, which frustrates citizens.
The said research tested the security of these websites using hacking tools (like ZAP and Arachni) and evaluated their ease of use with surveys and scanning software (Site Analyzer, AChecker). For comparison, we also analyzed a website from Azerbaijan's government.
The results were clear, Afghan websites have major security holes that attackers can easily exploit, and users are very unhappy with how they function. The automated tests showed they have numerous usability and accessibility problems, performing much worse than the Azerbaijani site. To fix this, we propose a three-part solution focusing on better development, deployment, and policies.

INTRODUCTION

In 2011, the Afghan government adopted a national e-governance strategy to enhance transparency and improve public service delivery [1]. Today, over 250 government websites provide digital services, streamlining bureaucracy and reducing red tape. These websites support various interaction models—G2C, G2B, G2G, C2G, and B2G [2] but face serious challenges in security, usability, and accessibility. No central authority exists to address these issues. Most sites are vulnerable to cyberattacks, lack user-friendly design [3], and are inaccessible to users with disabilities or low literacy levels [4].

This study analyzes Afghan government websites from three perspectives:

Security, using open-source VAPT tools (Arachni and ZAP)

- **Usability**, via user questionnaires and Site Analyzer
- **Accessibility**, based on WCAG 2.0 compliance

The paper concludes with recommendations to improve these aspects across government web platforms.

1. Literature review

Some of the studies talk about automated tools' effectiveness in detecting vulnerabilities in a reasonable time frame. In other hand these automated tools have some limitations to be improved[5], Jai Narayan Goel [6] discussed the effectiveness of vulnerability assessment and penetration testing as a technique for cyberattacks and how to use VAPT tools to speed up detecting vulnerabilities before the attackers do[6]. Therefore, there are many kinds of research and studies available focusing on analyzing governmental websites' status using different automated tools from which we are trying to mention the work done by some of these researches in this paper.

1.1 Security Evaluation of Saudi Arabia's Websites Using Open-Source Tools

In 2015 a group of university students scanned different websites such as commercial, governmental, academic, and financial for different vulnerabilities using different open-source automated vulnerability scanners.
Furthermore, these students compared commercial websites with governmental websites based on the number of vulnerabilities detected during the vulnerability assessment process. In the end, the result of this paper shows that clickjacking was one of the vulnerabilities that exist in most governmental websites of Saudi Arabia [7].

1.2 Vulnerabilities of Government Websites in a Developing Country – The Case of Burkina Faso

In 2017, many researchers from different universities of Burkina Fase researched governmental websites for well-known and straightforward vulnerabilities using a framework known as s2e-gov, a security testing framework for websites. This research's motivation was from April 2017; a large number of government websites were attacked and returned an error page only while browsing. Researchers found many reasons behind this hack. Most of these websites were made using known Content Management Systems (CMS) with known vulnerabilities. The result of this research shows that more websites were built using Joomla 1.5 CMS, which had 14 known vulnerabilities at that time. Furthermore, Attackers take advantage of these known vulnerabilities and can attack many websites at once. Therefore, this research is based on an E-government security testing framework, s2e-gov, which helped them uncover many known vulnerabilities in governmental websites [8].

.3 Evaluating Usability analysis of Jordan's universities

Suleiman H. Mustafa university students in Jordan researched in 2016 to analyze some university websites' usability [9]. Two automated tools were used in this research to evaluate the website's usability from a source code perspective. Moreover, for evaluating the external attributes of these websites, a questionnaire is used. Automated tools used in this research are HTML toolbox, which evaluates Load time, HTML check errors, browser compatibility problems. Simultaneously, the web page analyzes to check for the source code and analyze different aspects such as the average number of HTML files, the average size of an HTML page, the average size of images, the average number of images, and the average number and size of CSS files.

.4 Developing Countries E-government Site usability analysis based on users perspective

A group of students from various universities in England conducted a 2019 study to assess the usability of e-government web portals in developing countries. As these services are vital for efficient citizen engagement, the study aimed to evaluate user experience. Choudrie, Jyoti et al. [10] designed a questionnaire with 10 items, tested on 50 e-government portals by 50 randomly selected participants. Each participant reviewed five portals, focusing on aspects like navigation, search functionality, text, colors, layout, FAQs, contact details, and help features. The study covered countries from nine regions and revealed several usability challenges that hinder effective portal usage.

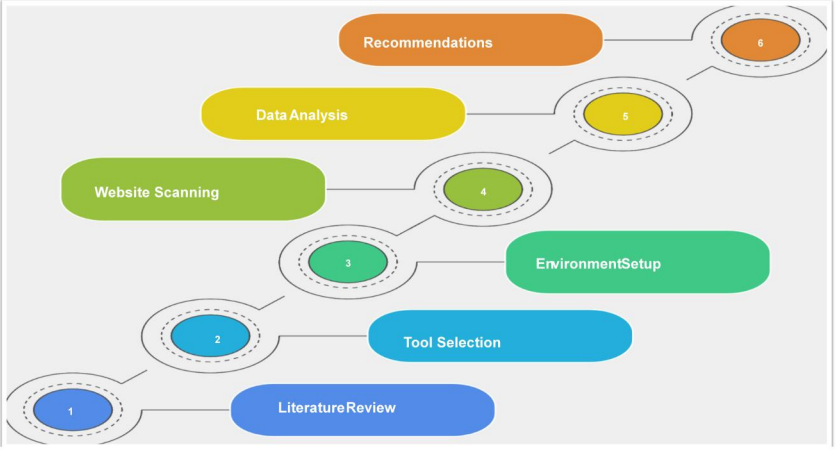
2. Research Methodology

The methodology used for this research is mixed (qualitative and quantitative), the is quantitative because all the results collected are in numbers. This paper explains two different aspects of a website: usability and security. Therefore, this chapter is divided into two different parts. Two flow charts are designed to show the step-by-step process for analyzing website security and usability. furthermore, the process of website assessment is explained briefly. Automated VAPT tools are selected for scanning websites for known vulnerabilities. Furthermore, each of these tools is explained to show the reader how they can detect and exploit different known vulnerabilities. The second part explains two different techniques for analyzing website usability in this research questionnaire and automated usability analysis tools. Accessibility is the subset of website usability; therefore, two different tools are selected and explained shortly. Site analyzer to scan websites for usability issues and AChecker to scan the website for accessibility issues. Finally, one website from a developed country Azerbaijan is selected to compare the result with Afghanistan governmental website.

.1 Security analysis process flow

Analyzing website security consist of six different steps. Literature review to collect background knowledge about the process, selection of automated VAPT tools to scan the selected website, simulation of the website to be scanned by the automated VAPT tools, the process of scanning the website using automated VAPT tools for known vulnerabilities, generating report and collecting results, analyzing the result based on the current state of website and giving recommendations for mitigating the existing security issues

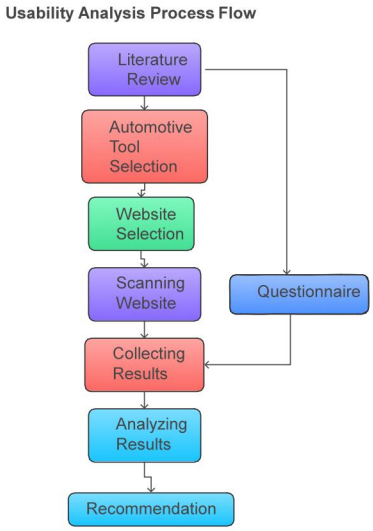
Research Methodology Process Flow



2.2 Usability analysis process flow

The usability of websites is evaluated using two different approaches in this research, questionnaire and automated tools. A literature review for conducting background knowledge about website usability is done. Meanwhile, two different processes, questionnaires, and automated tools start to analyze governmental websites' status [11].

Moreover, automated tools are selected to scan the selected websites, different websites from the different governmental organizations are selected and then scanned using usability and accessibility analysis tools. Furthermore, the results from the questionnaire and automated tools are collected and analyzed. Finally, some recommendations will be provided describe the mitigation process for the governmental website's usability and accessibility [12].



2.3 Assessing Websites

The central part of this research is about assessing websites for security and usability issues. As per the definition of [12], assessment means to define or discover the quality of something. Therefore, governmental websites are assessed for usability and security issues in this research. For assessing the targeted website against known vulnerabilities, two open-source, automated VAPT tools are selected (Arachni and ZAP).

On the other hand, for assessing the websites usability issues, two different approaches are used in this paper automated tools for defining usability issues in the source code, called internal attributes, and collecting user experience on using these governmental websites to discover some external usability attributes.

2.4 Web Security Scanning

Different techniques are available for vulnerability assessment and penetration testing such as static analysis, Fuzz testing, manual testing, and automated testing

The automated testing approach is used in this paper to find and detect different types of known

vulnerabilities with less amount of time. Furthermore, to collect accurate results from scanning the website, two different automated VAPT tools will be selected [13].

For automated testing, a VAPT tool is needed to scan the website for known vulnerabilities, exploit detected vulnerabilities, collect the result, and suggests some countermeasures for removing and solving those known vulnerabilities.

2.5 VPAT Tool selection

For this paper we are using the two best free, open-source web security scanners OWASP ZAP and Arachni.

Automated penetration testing tools, both free and commercial, offer diverse capabilities, ranging from scanning simple web pages to complex enterprise applications with multiple workflows [14]. Benchmarks like WIVET [15], WAVSEP [16], and OWASP Benchmark assess these tools using test cases with known vulnerabilities to evaluate detection accuracy and false positives. Metrics include test type (black-box, white-box, grey-box), crawling method, scan speed, and detection rates [14].

According to M. Shah [15], OWASP ZAP outperformed other open-source tools in vulnerability detection. Since no single scanner covers all vulnerabilities, using multiple tools yields more reliable results. This paper utilizes OWASP ZAP and Arachni, detailed in the following sections.

3.5.1 Arachni

Arachni is an open-source, cross-platform web security scanner built with Ruby, designed for penetration testers and administrators to detect known vulnerabilities. It supports major operating systems and offers portable packages for easy deployment.

Its embedded browser enables effective scanning of modern web apps using client-side code like JavaScript. Arachni uses passive scanning to detect files, directories, and HTTP configurations, and active scanning to identify vulnerabilities such as SQL injection, XSS, CSRF, and file inclusion.

Reports are generated in multiple formats including HTML, JSON, XML, TEXT, and Arachni's own AFR format [17].

OWASP ZAP

Zed Attack Proxy (ZAP) is a free, open-source VAPT tool developed by OWASP and maintained by international volunteers. It supports passive and active scanning of web applications, passive scans are safe and non-intrusive, while active scans simulate real attacks to exploit known vulnerabilities, requiring full permission due to potential risks.

ZAP offers two crawling methods: the fast traditional spider and the slower AJAX spider, which handles JavaScript-generated links. It also supports manual and automated scanning. In manual mode, testers explore pages and submit forms manually while the passive scanner runs in the background. Automated scans use spiders to explore the site and then perform active scans to detect vulnerabilities [18].

Reports can be generated in formats like HTML and XML, detailing alerts, severity levels, and other key findings.

Questionnaire Design

User experience (UX), a key aspect of Human-Computer Interaction (HCI), is commonly assessed through methods like questionnaires, interviews, observations, and focus groups. According to J. Avila et al. [19], questionnaires are most effective for collecting UX data from large groups.

These questionnaires include:

Pre-questionnaire gathers demographics (age, education, internet and browser usage).

Usability questionnaire: evaluates content, navigation, cross-browser/platform compatibility,

and page design. Post-questionnaire: collects final impressions and feedback. Questions may be open-ended for detailed responses or closed-ended for structured data [20].

Characteristic	Webpage Design	Navigation	Compatibility
Readability	Correct layout	Self-explanatory	Up-to-date
Finding Content	Design	Redirection	Cross-browser
User-Friendliness	Layout	Menus	Cross-device
Search	Search facility	Difference	Cross-browser/device

3.5.4 Usability and accessibility analysis based on automated tools

Website usability issues such as HTML errors, page size, download time, and SEO performance can be efficiently analyzed using automated tools . These tools scan source code to assess compliance with accessibility guidelines and evaluate technical aspects like image size, broken links, alt text, browser compatibility, and cross-device responsiveness [21].

Compared to manual methods, automated tools offer faster and more scalable usability assessments [22].

3.5.5 Site Analyzer

Site analyzer, page analysis feature was used to evaluate five usability attributes: SEO, content, design, performance, and accessibility, with results exportable as a PDF [23]

3.5.6 Achecker

It is a free online tool for evaluating HTML accessibility via file upload or URL. It integrates with W3C and CSS validators [24], and outputs reports in formats like PDF, HTML, and CSV. Results are categorized into known, likely, and potential problems, some requiring human review [25–26]

5 Setup up environment

As described by ZAP the tester needs to have full permission from the web application author for an active scan because it may cause some problems to the web application. Therefore, with the MCIT Content Management System (CMS) team's coordination, a simulated version of MCIT's current website http://mcit.gov.af is deployed under a subdomain https://cms.mcit.gov.af. However, the data, pages, and forms are less than the real website.

. Result and Discussion

This research presents the results collected from different methodologies to evaluate website security, usability and accessibility. Furthermore, it represents the results collected from scanning the target simulated version of the MCIT website

with Arachni and OWASP ZAP. Moreover, these results are analyzed for evaluating the security of the selected website.

At the end of the security evaluation process, vulnerability scanners used in this research are compared using the vulnerability detection rate and the severity of those vulnerabilities.

The second part of this research explains the evaluation procedures for analyzing websites' usability. Questionnaire results collected from fifty participants present the user comfort level with governmental websites. Moreover, these results describe websites' usability based on four usability attributes: content, design, navigation, and cross-browser/ cross-device compatibility. On the other hand, Site Analyzer and AChecker represents the website usability issues that occur in source-code.

Finally, for mitigating these web security and usability problems, an action plan is presented, containing some recommendations for the websites' developers, administrators and Afghan government.

4.1 VAPT Tool Results

Various techniques exist for analyzing web application security, all aiming to detect known vulnerabilities, but differing in execution [27]. This research uses automated testing, combining active and passive scans to assess a selected website. Permission for active scanning was obtained from the Afghanistan National Data Center (ANDC).

Automated penetration testing is faster, more effective, and reliable [28]. The process involves entering the website URL, crawling for vulnerabilities, and then attacking them using known exploits. After scanning, a detailed report is generated, listing vulnerabilities, severity levels, and countermeasures.

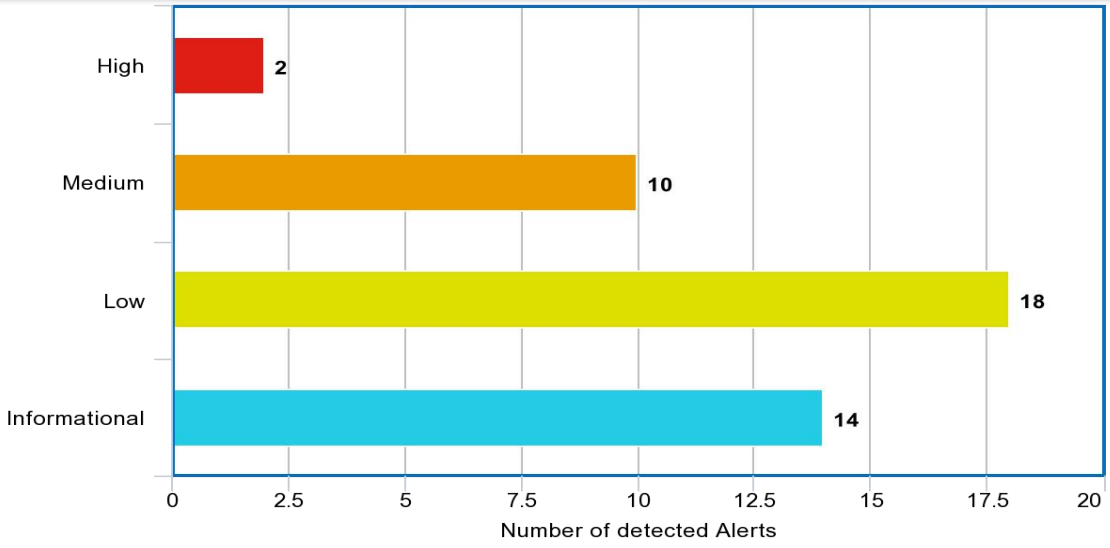
The website's security is evaluated based on the number and types of vulnerabilities detected by both scanners [29].

ZAP Result

ZAP supports both automated scans and manual exploration. While automated scans using traditional and AJAX spiders are a good starting point, they offer limited control over page exploration and attack types. Manual exploration provides deeper access to website functionalities.

In this study, manual exploration was used alongside both spiders. Forms were filled with varied inputs, and all pages and links were manually explored. After exploration, ZAP performed an active scan using built-in exploits, which lasted over four hours, sending more than 60,000 requests and detecting 575 alerts across various categories.

A bar chart below illustrates the number of alerts by severity level.



The above bar chart shows only the number of detected alerts with their level of severity. In contrast, each alert level has different alert categories detected in different sources of the target website. Therefore, each category of alert with their level of impact is presented in the following table. Furthermore, for each alert's reference, the CWE(Common Weakness Enumeration) number is given so that the reader can find more information about each of these alerts.

Vulnerability	Instances	Impact	Reference
PII Disclosure	9	high	359
Anti-CSRF Tokens Check	509	high	352
SubResource Integrity Attribute Missing	1863	Medium	16
CSP Scanner: WildCard Directive	9	Medium	16
CSP Scanner: Script-src-unsafe-inline	9	Medium	16
CSP Scanner: Style-src-unsafe-inline	9	Medium	16
Cross-Domain misconfiguration	47	Medium	264
Proxy Disclosure	49	Medium	200
Reverse Tabnabbing	434	Medium	1022
Source Code Disclosure -Perl	2	Medium	540
Vulnerable JS library	6	Medium	829
X-Frame-Options Header Not Set	9	Medium	16
Absence of Anti-CSRF Tokens	763	Low	352
Strict-Transport-Security Header Not Set	885	Low	16
Server Leaks Version Information via"Server" HTTP Response Header Field	964	Low	200
In Page Banner Information Leak	57	Low	200
Feature Policy Header Not Set	664	Low	16

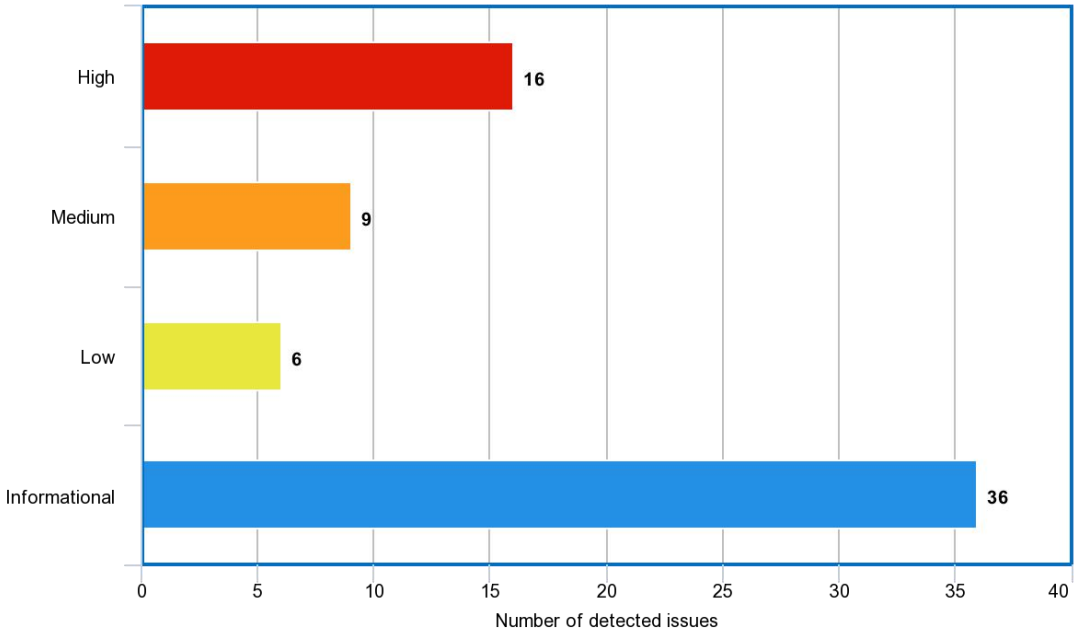
X-Content-Type-Options Header Missing	29	Low	16
Content Security Policy (CSP) Header Not Set	511	Low	16
Incomplete or No Cache-control and Pragma HTTP Header Set	575	Low	525
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	452	Low	200
Secure Pages Include Mixed Content	13	Low	311
Cookies without secure Flag	3	Low	614
Cookie No HttpOnly Flag	1	Low	16
Cookie Without SameSite Attribute	3	Low	16
Cross-Domain JavaScript Source File Inclusion	1486	Low	829
CSP Scanner: Notices	9	Low	16
Big Redirect Detected (Potential Sensitive Information Leak)	1	Low	201
X-Backend-Server Header Information Leak	1	Low	200
Dangerous JS Functions	30	Low	749

The table above represents different alert categories in high, medium, and low severity. However, ZAP detected a total of 104 Informational alerts from fourteen different categories as well. As far as Informational alerts do not directly affect the system security. Therefore, those alerts are not mentioned in the above table.

4.1.2 Arachni Result

Arachni offers a browser-based GUI and supports three scan types: SQL injection, XSS, and Global. This study used the Global scan to cover all active and passive vulnerabilities. The scan was performed on Ubuntu 21, as recommended, by entering the target website's URL and scanning all pages.

The process took over eight hours and resulted in an HTML report. Arachni found issues in 28% of the explored URLs (56 out of 200). The report identified 71 issues, 4 untrusted (requiring manual verification) and 67 trusted categorized by severity: High, Medium, Low, and Informational. A health map also showed safe vs. unsafe URLs.



Each severity level has several issues detected in different categories. Therefore, each category of alert with their level of impact is presented in the following table. Furthermore, for each issue reference, the CWE number is given so that the reader can find more information about each of t

hese issues.

Vulnerability	Instances	Impact	Reference
CSRF	15	High	352
Blind-SQL injection	1	High	89
Missing 'Strict-Transport-Security' header	1	Medium	200
Common Directory	8	Medium	538
Common Sensitive file	5	Low	.
Missing 'X-Frame-Options' header	1	Low	693

The above table represents different categories of vulnerabilities detected by Arachni. Furthermore, it shows how many times a specific vulnerability occurred on the website. However, Arachni detected a total of 36 Informational alerts from two different categories as well. As far as Informational alerts do not directly affect the system security, those alerts are not mentioned in the above table.

ZAP Vs Arachni

Scanning websites with different VAPT tools will generate different reports with different numbers and types of vulnerabilities. Each of these tools uses different criteria for crawling web-pages and exploiting vulnerabilities. Therefore, the result of OWASP ZAP and Archni generated in this research is also different.

Arachni and ZAP detected the same number of high severity vulnerabilities. In contrast, the

number of vulnerabilities detected with medium severity was different in both scanners. ZAP detected ten different types of vulnerabilities, but Arachni only detected two different medium level vulnerabilities. Meanwhile, the number of low severity vulnerabilities detected by ZAP was also more than Arachni. Moreover, Arachni took almost twice as ZAP time to complete the scan.

4.3 Usability Evaluation procedure

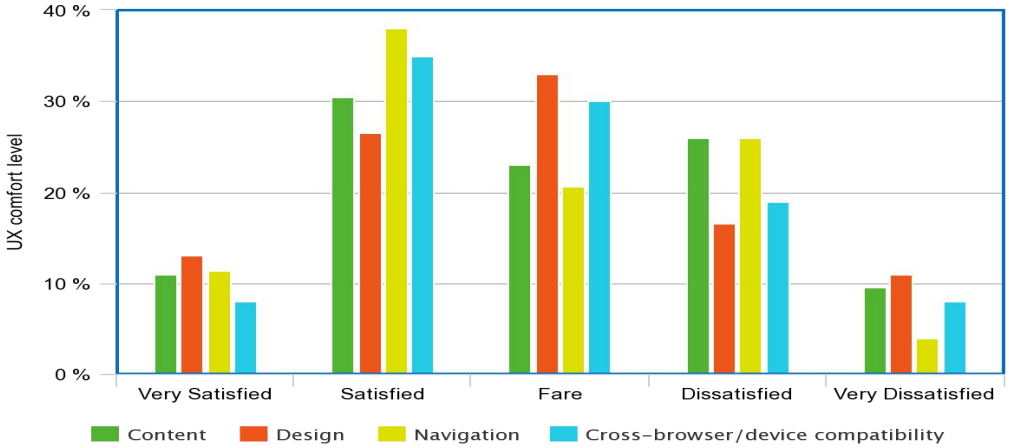
The two different methodologies are used in this research to assess and analyze governmental websites' usability and accessibility. A questionnaire is used for analyzing the external usability attributes of websites based on users' experience. On the other hand, automated tools are used to evaluate the website source code to analyze internal attributes such as content, design, navigation, cross-browser, and cross-device compatibility. Furthermore, questionnaire results are collected from 50 participants from Afghanistan. On the other hand, both the Site analyzer and AChecker found different website usability issues and accessibility issues. After completing the scan process, these automated tools generated a report that shows the number of usability and accessibility issues found on the website and some countermeasures for solving those issues.

Survey Result

The questionnaire for user feedback regarding governmental websites is created with the google form survey. Furthermore, the link of that questionnaire is shared with fifty participants from Afghanistan. The result of user demographics shows that most of these participants were already familiar with using governmental websites. Furthermore, from 50 participants of the survey, 76% were between 25 to 27 years old. Moreover, all of these participants hold a master's or bachelor's degrees, which means that they have a good familiarity with using the internet. Furthermore, the result shows that more than 90% of these participants are using Google chrome for browsing these websites.

The second part of this questionnaire is the central part, which asks questions about four different usability attributes of a website. The user experience comfort level in this part is analyzed with five levels of user satisfaction in each question asked about a parameter in website usability such as Very Satisfied (5), Satisfied (4), Neither Satisfied nor dissatisfied (3), Dissatisfied (2) and Very Dissatisfied (1).

Bar charts representing UX comfort level in four different usability categories selected in this research are shown below.



The above bar chart shows the UX(user Experience) comfort levels with different parts of the website such as Content, Design, Navigation and Cross-browser and device compatibility.
Finally the results of post-questionnaire shows that more than 40% of users are facing errors while using governmental websites. On the other hand less than 40% of users are satisfied with the overall layout of websites from a usability perspective.

4.4 Site Analyzer Result

The Site Analyzer scans the home page of a website. It evaluates the home page against five different attributes that affect a website's usability, such as SEO, performance, design, content, and accessibility. Furthermore, the website is evaluated under specific cases for each attribute to analyze the website's usability based on those rules. As an example for checking the website performance, Site Analyzer evaluates the web-page size, compression time, and execution time [17]. As far as Site analyzer is a commercial automated tool for analyzing websites for usability, it gives 14 days of trial usage. Therefore in this paper, the websites are scanned and analyzed with the trial version. The results are presented in the below table.

Website	SEO	Performance	Design	Content	Accessibility	Avg.
https://mcit.gov.af	33	44	56	60	66	51.8
https://pashtanybank.com.af	90	60	62	62	58	66.4
https://dabs.gov.af	36	61	49	25	52	44.6
https://aop.gov.af	7	100	55	0	29	38.2
https://bma.com.af	57	40	64	51	67	55.8
https://moi.gov.af	50	73	71	53	58	61
https://mudh.gov.af	43	61	50	25	23	40.4
https://mopw.gov.af	62	73	50	40	48	54.6
http://mof.gov.af	42	62	50	46	36	47
https://asan.gov.af	30	28	56	59	44	43.4
Average	**45**	**60.1**	**56.3**	**42.1**	**48.1**	.
https://asan.gov.az/en	**55.7**	**55.2**	**71.6**	**73.2**	**79.8**	**67.1**

The trial version of Site Analyzer scanned these websites and generated these results shown in the above table. Every attribute of a website's usability has 100 points. More- over, the average of all those five attributes is calculated and written in the last column. The GMIC website got 100 points in performance among all these websites but got significantly fewer points in SEO and Content. To better understand these numbers, the UNDP website is scanned and analyzed by this tool to compare the results with Afghanistan governmental websites. However, the results show that none of the Afghanistan government websites got usability scores as the UNDP website. Each attribute in the UNDP website got more than a 50% score.

.5 AChecker (Web Accessibility Checker)

This tool checks the website home page for accessibility issues against the known accessibility standards such as WCAG, Section 508, and

Stanca Act. The research uses WCAG 2.0 with Level AA, which has different guidelines for checking the accessibility of a website. Furthermore, after scanning and analyzing the websites, results are mentioned in the below table with detected known and likely problems in the specified website. Moreover, to better understand these numbers, a website from the Azerbaijan government is also scanned and analyzed with this tool to compare the results with Afghanistan governmental websites.

Website	Known Problems	Likely Problems
https://mcit.gov.af	76	20
https://pashtanybank.com.af	46	10
https://dabs.gov.af	30	0
https://aop.gov.af	11	10
https://bma.com.af	86	1
https://moi.gov.af	72	0
https://mudh.gov.af	6	0
https://mopw.gov.af	4	1
http://mof.gov.af	77	4
https://asan.gov.af	54	0
https://asan.gov.az/en	**1**	**2**

The numbers shown in the above table are the number of accessibility mistakes present in the website. Moreover, the results show that communication and information technology and the Ministry of Interior websites have the most accessibility issues. In comparison, the ministry of public health website has the least number of accessibility problems to be fixed.

To better understand the result of selected governmental websites, the result of the Azerbaijan government website is also presented in the table.

The result shows that Afghanistan governmental websites have more errors compared to Azerbaijan government website. Except for the ministry of public helath website, which has the same number of known problems as the Azerbaijan website but has less likely problems than the Azerbaijan website.

The number of known and likely problems found in these websites is mentioned here. These results are discussed further in the usability results discussion section.

. Result Discussion

The website's usability, accessibility, and security are evaluated and discussed. At the end of this part, the reader will know about governmental websites' current situation from a security, usability, and accessibility perspective.

This paper discusses two main parts of a website assessment, usability, and security. therefore this part also divided into two different parts, each discussing the results generated from different techniques. The website security is discussed based on the results of two VAPT tools ZAP and Arachni. However, the usability is discussed based on users' perspective, and automated tools result such as AChecker and Site analyzer.

.1 Security Result Discussion

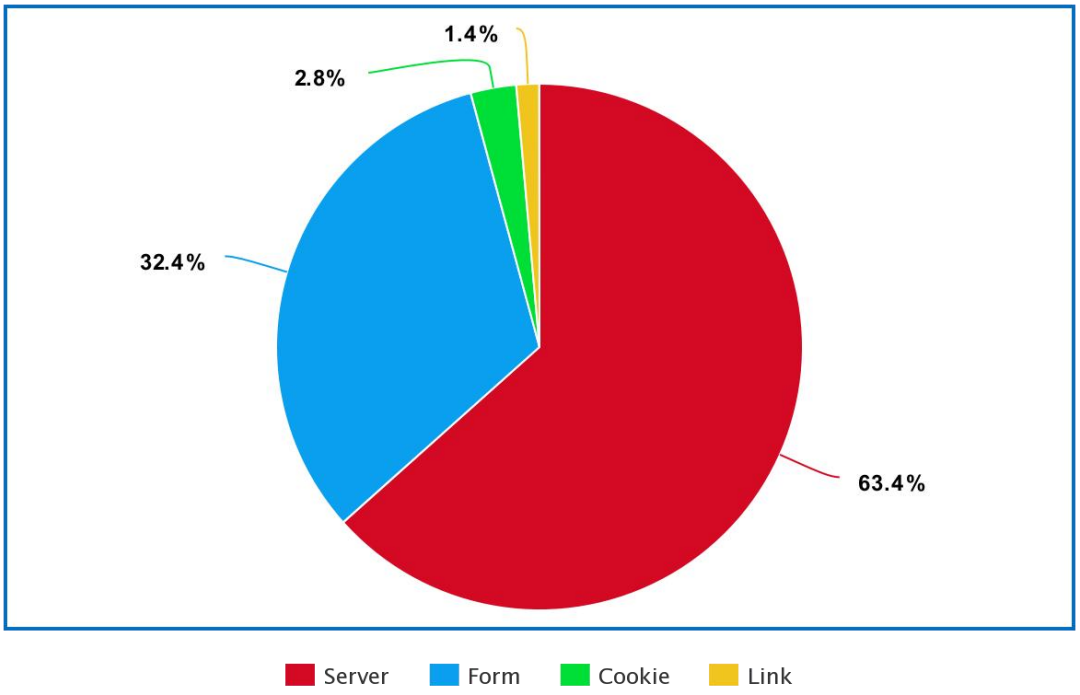
Based on the results from Arachni and ZAP, both tools detected two high-severity vulnerabilities. One was Anti-CSRF token check, indicating guessable form fields. However, since the website lacks authentication and sensitive form submissions, this was marked as a false positive [18].

Arachni identified a blind SQL injection via differential analysis, while ZAP found nine instances of PII disclosure, unauthorized access to personal data [19]. These correspond to OWASP categories A1-Injection and A3-Sensitive Data Exposure, requiring urgent fixes [20].

ZAP detected more medium-severity vulnerabilities (10 types) than Arachni (2 types), mostly due to web server misconfigurations such as missing headers and proxy disclosures. Inclusion of insecure third-party libraries was also common. While medium-level issues may not directly compromise the system, they can aid attackers and must be addressed.

Low-severity vulnerabilities provide indirect security risks. Arachni found two types, including sensitive file exposure. ZAP detected 18 types, with seven linked to CWE configuration issues, indicating improper server setup [21].

Informational alerts, though not direct threats, reveal details about server setup, technologies, and configurations. ZAP detected 14 types, while Arachni found four. Overall, most vulnerabilities stem from server misconfigurations and poorly designed forms.



The pie chart above shows that more than 50% of detected vulnerabilities are related to poorly configured web servers. Many websites share the same web-server in ANDC resulted by reverse IP lookup [22]. Therefore, if an attacker compromises one website, the other website hosted on the same web server will be affected.

Furthermore, more than 50% of these websites were built with CMS frameworks, and 110 of these websites are using the MCIT template created with drupal [30]. Therefore, the results

collected from the simulated MCIT website can give better information about the governmental web- sites created with the same MCIT template.

The above results for this paper are from the simulated version of the MCIT website, which has fewer web pages than the live system. Therefore, running VAPT tools against the live website may return more security issues than the simulated version. Moreover, using the CMS framework for building websites has some advantages and disadvantages, such as websites built with these frameworks are fast and have good community support. On the other hand, these CMS frameworks have some disadvantages due to a security point of view. These frameworks are open-source and used all over the world. Therefore, it is easy for an attacker to find known problems in these frameworks and then immediately use it on a website. Therefore, these frameworks release different patches time by time so that the administrator can update their website as soon as the new patch is released and defend themselves against the known bugs. The current version of drupal

used by the MCIT template is 8.6.10, while the drupal 11.12.2. is released in June 2025[23].

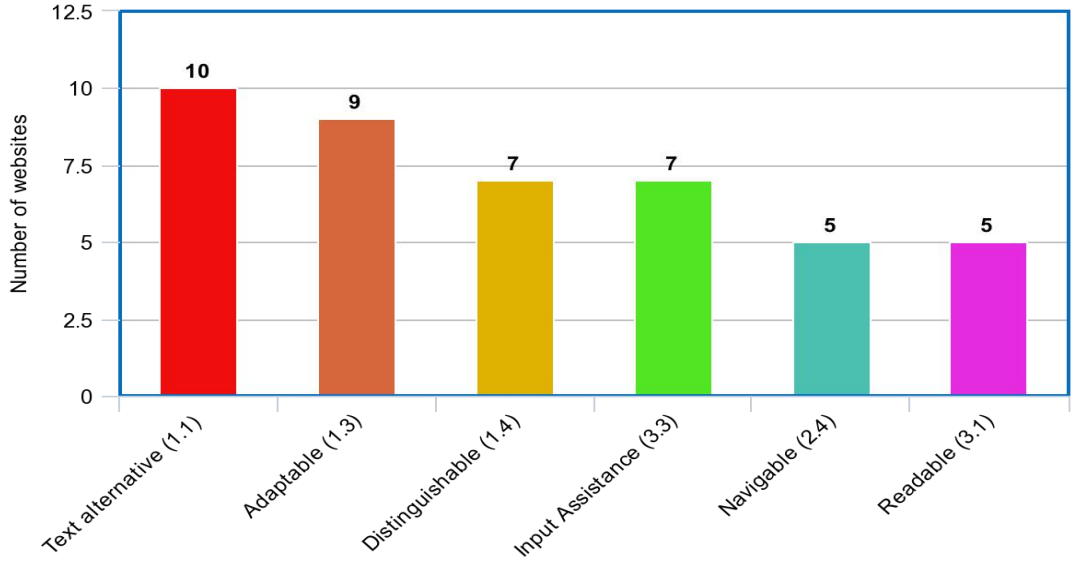
5.2 Usability Result Discussion

The results from automated tools and questionnaires indicate that Afghanistan's government websites lack proper usability and accessibility. Although positive responses slightly outnumber negative ones for usability attributes, content (40.5%), design (36.9%), navigation (49.3%), and compatibility (43%), none exceed 50%, and each has over 26% negative feedback. Navigation received the most positive responses, while content had the fewest negative ones.

The survey targeted educated users (bachelor's/master's degree holders), suggesting results may differ for less experienced users. Automated tools like Site Analyzer assessed websites based on usability attributes and deducted scores for issues. Findings show severe usability problems across all tested sites, including missing meta descriptions (8/10 sites), short page titles (5 sites), and poor text/code ratios (<15%).

Design issues were minimal, though two sites lacked HTML5 doctype. Performance analysis revealed three sites exceeded the 50 KB page size, two exceeded the 0.5s execution time, and two had file compression issues. The Da Afghanistan Bank website scored highest, while AOP scored lowest. Compared to an Azerbaijan government site (69.7% average), six Afghan sites scored below 50%, indicating lower usability.

Accessibility analysis via AChecker showed most sites failed to meet WCAG guidelines. The following chart presents the number of websites and WCAG rules not followed



The above bar chart shows that Text alternative is the only guidelines not followed by any of the selected websites. Moreover, Adaptability is the other guideline only followed by one website. Moreover, distinguishable and input assistance are only followed by three different websites. In the end, navigability and readability are those guidelines that are followed by half of the selected websites. Moreover, three other guidelines are not followed by one or two websites, such as compatibility, keyboard accessibility, and enough time for reading a text or other media. To better understand the result of AChecker, these selected websites are compared with the Azerbaijan website, which shows that Afghan government website are not accessible compared to http://asan.gov.az website.

6.
7.
8. **Recommendation**

This part of the paper gives recommendations for mitigating current security, usability, and accessibility issues in governmental websites. furthermore, it is divided into two different parts, first recommendations for website security and in the other part recommendations for mitigating website usability and accessibility issues.

.1 Recommendation for Website Security

Security is not a one step process, but it has to be applied as a continuous process in the whole SDLC processes. Therefore we divided these recommendations into three parts such as development phase, deployment phase and policy level.

Development phase

Security requirements should be integrated from the initial requirement gathering through to final development. Key recommendations include:
Define misuse cases and security needs early.
Use prepared statements to prevent SQL injection.
Validate and sanitize user inputs to avoid XSS.
Implement CSRF tokens in all forms.

- Encrypt sensitive data during storage and transmission.
- Customize error pages and hide source code.
- Remove or rename default files, folders, and admin panel access.
- Restrict browser access to sensitive files and folders.
- Use trusted libraries and keep them updated.
- Avoid passing sensitive data in URIs.
- Perform unit tests on critical security functionalities.

8.1.2 Deployment phase

The main cause of security issues in Afghan government websites is the lack of a dedicated team for proper configuration and vulnerability testing before deployment. A security checklist must be completed during deployment. Key recommendations include:

- Conduct security testing before going live.
- Use SSL with strong encryption.
- Configure servers securely and install only trusted third-party software.
- Enforce HTTPS and redirect HTTP requests.
- Keep all third-party tools, libraries, and CMS updated.
- Restrict direct access to sensitive files and folders.
- Set secure cookie attributes.
- Avoid unauthorized software installations.
- Allow only necessary HTTP methods.
- Continuously monitor and patch systems.
- Re-test security after any code changes.

8.1.3 Policy

In order to mitigate the security issues in governmental website some policies need to be developed and applied in each governmental organization.

There is no specific governmental authority that verifies and confirms the security level before deploying and launching the website into the live environment. Therefore, a central authority needed to test and verify website security before launching them into the live environment.

Continuously monitor the website to discover any new vulnerabilities.

Provide security training for website developers and administrators.

. Conclusion

The Afghan government has adopted e-government to modernize bureaucracy and improve service delivery. While websites offer public services and information in national languages, they suffer from significant security, usability, and accessibility issues.

This research analyzed these aspects using automated VAPT tools (ZAP and Arachni). Scanning a simulated MCIT website revealed multiple vulnerabilities, including high-severity ones. ZAP detected more issues than Arachni, making it the more effective tool. Although only one site was tested, many government websites share similar CMS templates and configurations, suggesting widespread vulnerabilities that require urgent attention.

Usability was assessed through user feedback and automated tools. Questionnaire results showed user dissatisfaction, and Site Analyzer scans of ten websites revealed consistent usability flaws. Accessibility analysis via AChecker confirmed that most sites fail to meet WCAG 2.0 guidelines.

In conclusion, Afghan government websites face critical security, usability, and accessibility challenges. This research aims to guide future improvements in their development and maintenance.

0. Future work

This research lays a strong foundation for understanding website usability and accessibility in the Afghan context. While automated tools and questionnaires provided valuable insights, future work can build on these methods to enhance accuracy and inclusivity.

Additionally, expanding tool capabilities to analyze more comprehensive portions of website source code will further strengthen evaluations.

The current study benefited from the input of well-educated participants with prior experience in web usage. To better reflect the broader Afghan population, where literacy rates vary, future studies could incorporate feedback from a more diverse user base, including individuals with limited digital exposure. Testing website functionalities with randomly selected users unfamiliar with the site can offer a more authentic measure of usability.

Moreover, accessibility assessments will be enriched by involving disabled users and citizens from various educational backgrounds. Their experiences and feedback are crucial for developing truly inclusive digital platforms. By embracing these opportunities, future research can contribute to more equitable and user-friendly web environments for all Afghans.

References

[1] Z. Dzhusupova, M. Shareef, A. Ojo, and T. Janowski, "*Methodology for E-Government Readiness Assessment: Models, Instruments and Implementation*", in *Proc. Int. Conf. on Society and Information Technologies (ICSIT2010)*, Orlando, FL, USA, Apr. 6–9, 2010, pp. 6. [Online]. Available: https://collections.unu.edu/view/UNU:3035

[2] Sheridan, D. Reynolds, and K. Glover, "*Improving Access to Government through Better Use of the Web*, W3C eGovernment Interest Group Note", World Wide Web Consortium (W3C), Sep. 2009. [Online]. Available: https://www.w3.org/TR/egov-improving/

[3] USF Health, "What is website usability & why is it important?" University of South Florida, 2021. [Online]. Available: https://health.usf.edu

[4] S. Abou-Zahra, S. L. Henry, and J. Brewer, "*Introduction to Web Accessibility*, Web Accessibility Initiative (WAI)", World Wide Web Consortium (W3C), Dec. 4, 2017. [Online]. Available: https://www.w3.org/WAI/videos/standards-and-benefits/

[5] J. Bau, E. Bursztein, D. Gupta, and J. Mitchell, "State of the art: Automated black-box web application vulnerability testing", IEEE, pp. 332–345, 2010.

[6] J. N. Goel and B. Mehtre, "Vulnerability assessment & penetration testing as a cyber defence technology", Elsevier, vol. 57, pp. 710–715, 2011.

[7] M. S. Al-Sanea and A. A. Al-Daraiseh, "Security evaluation of Saudi Arabia's websites using open source tools", IEEE, pp. 1–5, 2012.

[8] T. F. Bissyandé et al., "Vulnerabilities of government websites in a developing country: the case of Burkina Faso", Springer, pp. 123–135, 2013.

[9] C. M. Barnum, "Usability Testing Essentials: Ready, Set ...Test!', Morgan Kaufmann, 2010.

[10] M. Matera, F. Rizzo, and G. T. Carughi, "Web usability: principles and evaluation methods", in Web Engineering, Springer Verlag, pp. 143–180, 2014.

[11] M. P. Shah, '"Comparative analysis of the automated penetration testing tools", National College of Ireland, Dublin, 2015.

[12] E. Tatli and B. Urgun, 'WIVET: benchmarking coverage qualities of web crawlers', OUP, vol. 60, no. 4, pp. 555–572, 2016.

[13] S. Chen, 'WAVSEP 2017/2018 - evaluating DAST against PT/SDL challenges', 2018.

[14] Y. Matsumoto et al., *Ruby Programming Language Documentation*, Ruby Core Team, Ruby-

lang.org, 2025. [Online]. Available: https://www.ruby-lang.org/en/documentation/

[15] S. Bennetts, *OWASP ZAP: Introduction and Getting Started*, OWASP Foundation, Apr. 15, 2021. [Online]. Available: https://owasp.org/www-chapter-belgium/assets/2021/2021-04-15/ZAP-Intro-and-latest.pdf

[16] S. Kaur, K. Kaur, and P. Kaur, 'Analysis of website usability evaluation methods', IEEE, pp. 1043–1046, 2017.

[17] J. Bargas-Avila and K. Hornbæk, 'Foci and blind spots in user experience research', ACM, vol. 19, no. 6, pp. 24–27, 2016.

[18] N. Mathers, N. Fox, and A. Hunn, "Surveys and questionnaires: the NIHR RDS for the East Midlands/Yorkshire & the Humber", pp. 20–33, 2013.

[19] S. Adepoju and I. Shehu, "Usability evaluation of academic websites using automated tools", IEEE, pp. 186–191, 2018.

[20] R. Kumar and N. Hasteer, "Evaluating usability of a web application: A comparative analysis of open-source tools", IEEE, pp. 350–354, 2019.

[21] Site Analyzer, *Website Analysis and SEO Optimizer*, Site Analyzer, 2025. [Online]. Available: https://www.site-analyzer.com

[22] S. L. Henry, J. Brewer, and WAI Contributors, *Web Accessibility Evaluation Tools List*, Web Accessibility Initiative (WAI), World Wide Web Consortium (W3C), Aug. 2023. [Online]. Available: https://www.w3.org/WAI/test-evaluate/tools/list/.

[23] G. Oskoboiny et al., *About the W3C Markup Validation Service*, World Wide Web Consortium (W3C), 2023. [Online]. Available: https://validator.w3.org/about.html

[24] W3C, *About the W3C CSS Validation Service*, World Wide Web Consortium, 2023. [Online]. Available: https://jigsaw.w3.org/css-validator/about.html

[25] Inclusive Design Research Centre, *AChecker: Web Accessibility Checker*, OCAD University, Toronto, Canada, 2023. [Online]. Available: https://achecker.ca.

[26] TutorialsPoint, *Penetration Testing – Manual & Automated*, TutorialsPoint, 2025. Available: https://www.tutorialspoint.com/penetration_testing/penetration_testing_manual_automated.htm

[27] MITRE Corporation, *CWE-352: Cross-Site Request Forgery (CSRF)*, Common Weakness Enumeration, Version 4.17, Apr. 2025. Available: https://cwe.mitre.org/data/definitions/352

[28] CWE, 'CWE-359: Exposure of private personal information to an unauthorized actor', v4.2, 2022.

[29] CWE, 'CWE-16: Configuration', v4.2, 2022.

[30] Bounteous, 'Drupal 9 is coming: Upgrading has never been this easy', 2019.