Vulnerability Assessment of Some Key Nigeria Government Websites

Article *in* International Journal of Digital Information and Wireless Communications · November 2017

DOI: 10.17781/P002309

CITATIONS
6

READS
2,118

5 authors, including:

Ismaila Idris
Federal University of Technology Minna
42 PUBLICATIONS **326** CITATIONS
SEE PROFILE

Muhammad Umar Majigi
Federal University of Technology Minna
5 PUBLICATIONS **7** CITATIONS
SEE PROFILE

Morufu Olalere
Federal University of Technology Minna
23 PUBLICATIONS **84** CITATIONS
SEE PROFILE

Shafi'i Muhammad Abdulhamid
Federal University of Technology Minna
105 PUBLICATIONS **1,389** CITATIONS
SEE PROFILE

International Journal of Digital Information and Wireless Communications (IJDIWC) 7(2): 143-152
The Society of Digital Information and Wireless Communications, 2017 ISSN: 2225-658X (Online); ISSN 2412-6551 (Print)

Vulnerability Assessment of Some Key Nigeria Government Websites

[1]Ismaila Idris, [2]Mohammad Umar Majigi, [3]Shafii Abdulhamid, [4]Morufu Olalere, [5]Saidu Isah Rambo

[1,2,3,4]Department of Cyber Security Science, School of Information and Communication Technology, Federal University of Technology, P.M.B 65, Minna, Nigeria
[5]Nigeria Defense Academy, P.M.B 2901 Mando, Afaka, Kaduna, Nigeria

[1]ismi.idris@futminna.edu.ng, [2]muhammad.pg611700@st.futminna.edu.ng,
[3]shafii.abdulhamid@futminna.edu.ng, [4]lerejide@futminna.edu.ng, [5]sadurambo@yahoo.co.uk

ABSTRACT

Ministries, Department and Agencies (MDA's) websites are useful constituents for information dissemination and citizen centric services. Various vulnerabilities exist in this websites. In this paper, vulnerabilities found in MDA's website are categorized and analyzed based on Open Web Application Security Project (OWASP) Top 10 to understand impact of these vulnerabilities on web security of MDA's websites. In this study we have analyzed security pertaining to 10 MDA's websites. We found vulnerabilities in all websites with different degree of security risk. To achieve the results we have cross tabulated vulnerabilities found in these websites with their security risk level. As a result the research work found that vulnerability A4-insecure direct object reference with 49% is the main contributor of web security risk in MDA's websites. Apart from this it is clearly evident that majority of the vulnerabilities found in MDA's websites belongs to informational risk group with 45.82% but still few high impacting vulnerabilities exists and needs to be handle without delay. Thus, the paper contributed towards the understanding of web security risk in MDA's websites.

KEYWORDS

Web Vulnerability Scanner, OWASP, Security Risk, Vulnerability Assessment, MDA's.

1 INTRODUCTION

As information technologies continue to expand, especially for the web applications, Internet have become an integral part of any modern corporate information system. This is irrespective of the organization's line of business or industry. It is not just commercial organizations that develop and create their own web wealth; agencies of Government are similarly actively involved in the development of web-based services that are aimed at local and national levels [1].

E-commerce has grown significantly due to exponential increase in online transactions in the past few years. For instance US online retail sales grew 12.6% in 2010 to reach $176.2 billion. 2010 to 2015 is expected 10% compound annual growth rate (CAGR), in 2015 US e-commerce was expected to reach $278.9 billion [2].

Hacktivist group Anonymous declared a cyberwar campaign against the government of Nigeria in a fight against corruption, poverty and theft. In an online post, the amorphous online activist collectively called on its followers to "take out" the websites of Nigeria's Finance, Foreign Affairs and Justice Ministry as well as the Federal Capital Territory Administration. All four websites came under attack [3].

The data handle by web applications, such as shopping activity and credit card numbers information are normally of great value to the service providers and the users. For sustainability, web applications security should protect the user's data from modification, unauthorized access, disclosure, use, disruption, inspection, perusal and destruction or recording. However, it fails to satisfy above requirements often. Vulnerabilities in web applications is the root cause of most security risks on the Web [4], [5].

The Open Web Application Security Project (OWASP) is a worldwide free, open community focused on enlightening the security of software (application) and also Non Profit Charitable Organization with the mission to make software

security visible to persons and organizations to brand informed choices about their software security risks. [6] TOP 10 OWASP is a security project funded by OWASP. The list of current top 10 web application security risks project was published worldwide. The list explains the threat agents, attack vectors, security weakness, technical impacts and business impacts with a relevant example and how to prevent it. However, every three years, OWASP releases a report on the ten most critical web application security risks. The most recent version of the top 10 list was published last in June 2013. OWASP prioritized the top 10 according to their exploitability, prevalence common, detectability, and impact severe [7]. The OWASP Top Ten provides a powerful awareness document for web application security and it also represents a broad consensus about what the most critical web application security flaws are [8]. Furthermore, Management of security risk provides means to cope the rising threats to organizational infrastructures. Management of vulnerability is vital for mitigation of critical security risks [9].

Table 1. 2013 - OWASP TOP 10 Vulnerability [8]

S/NO	OWASP TOP 10 - 2013
1	AI - Injection
2	A2 - Broken Authentication and Session Management
3	A3 - Cross Site Scripting (XSS)
4	A4 – Insecure Direct Object Reference
5	A5 – Security Misconfiguration
6	A6 – Sensitive Data Exposure
7	A7 – Missing Function Level Access Control
8	A8 – Cross Site Request Forgery (CSRF)
9	A9 – Using Known Vulnerable Component
10	A10 – Unvalidated Redirects and Forwards

With the increase in E-governance initiatives Government websites are the main source of information for the people and key component in E-Government projects, various citizen centric services provided by Government are through these websites [10].

In Nigeria, it is obligatory to conduct security audit time - to - time from the enlisted Ministries, Departments and agencies (MDA's) of Government websites/web application. It is expected that the security audit be done before hosting and after addition of a new module. Aside these, each MDA's need to have a security policy to address numerous security problems related to a web application/website

1.1 Statement of the Problem

With the development in web technologies the popularity of web-based applications has grown tremendously. Today, in security critical environments such as ministries, departments and agencies, web based applications are used. Similarly, as web application user's vulnerabilities are increasing in our organizations, this will without doubt expose more web application users to malicious attacks. Government websites are useful constituents for information distribution and citizen centric services. A number of vulnerabilities exist in MDA's websites

2 LITERATURE REVIEW

2.1 Introduction

This section explains the various works carried out by researchers in the area of vulnerability analyses using security web vulnerability scanners. Different approaches used by researchers to successfully check for vulnerabilities will be reviewed.

2.1 Review of Related Works

This study aims to analyze vulnerability assessment of some key Ministries, Departments and Agencies (MDA's) websites in Nigeria. This section reviews eight (8) literatures of related vulnerability assessment based works.

In 2012, Chander *et al.* [11] proposed use of a web vulnerability scanner called Acunetix, the scanner is used for checking cross site scripting, SQL injection and other type of web vulnerabilities. It does this security scanning by checking the strength of the password on authentication pages and audit web applications automatically. A comprehensive report showing the details of scan is generated and the report

shows the areas where those vulnerabilities occurs. The result shows that most of the security alerts are informational type and the alerts are not categorized based on high level priority, the research concludes that websites with few vulnerabilities takes less time during the scanning. Though, the research was limited to only two websites, for a proper testing and analysis an e-government portal should be used as those websites tend to have more web vulnerabilities.

Marco *et al.* [12] presented four types of vulnerability scanners that are commercial source. The two versions of HP Web Inspect are IBM Rational AppScan and Acunetix Web Vulnerability Scanner, these scanners are used in the detection of the web vulnerabilities in over 300 web services that are available publicly. The result shows that it is hard to select vulnerability scanner for service on the web, as each scanner detects different web service vulnerabilities and false positive are usually high, which reduces the vulnerabilities detection precision and it also shows that many vulnerabilities are undetectable based on the fact that cases of detection are very low. The paper is limited to the fact that it doesn't perform vulnerabilities on web service that has authentication as its function and the vulnerabilities where performed using selected set of web representations.

Rafique *et al.* [13] Review various detection methods used for web vulnerabilities. The goal of the review is to determine which tools available for detection, the approaches used as well as the associated problems with those approaches are known. The authors extracted over 600 papers, 56 of those papers were used for analysis. The result shows that the approaches used for securing` web applications are different, which implies that there is not a standard method for security of web applications. It also shows that OWASP Top 10 has gained more popularity, which was due to the viewpoint of major stakeholder of web security. This paper is limited in its review as it lacking in the area of current state of the art support in this study.

Karumba *et al.* Presented "a hybridization algorithm for the detecting web application vulnerabilities" [14]. This proposed hybrid algorithm is capable of reporting more vulnerabilities and the reports of discovered vulnerability are presented in a proficient manner. This proposed system did not scan all the existing vulnerabilities, thereby limiting its optimization as it does not have the capacity of detecting more vulnerabilities. The structure of the reporting and analysis needs to be improved as well as having a better sophisticated crawling mechanism.

In 2015, Kagorora *et al.* [15] presented five web application scanners namely Acunetix, Arachni, Iron WASP, Vega and ZAP. The efficiency of vulnerability detections hidden behind AJAX/JSON was performed on the five scanners used. The result shows that the execution of AJAX code is one of the major challenges been faced by the tool. It also shows that in order to increase stored vulnerabilities detection ability, a scanner should crawl the application again after injecting attack codes and verify if its own codes did not cover all main features, they do not provide a complete assessment and comparison of WAVSs. Furthermore, need for further research on crawling modern web technologies, automated detection of complex vulnerabilities such as stored XSS, stored SQL injection and custom session management ability.

According to Ertaul *et al.* [16] the study implemented OWASP Top 10 vulnerabilities in a test application and tested two WAVSs against the application. Detection results indicate that the biggest challenge for these two WAVSs is to exploit stored and multistep vulnerabilities.

Chen, S. presented "a Price and Feature Comparison of Web Application Scanners" [17]. Detection results show that many scanners performed fairly well in detecting reflected XSS, and first-order SQL injections and a number of scanners found Path Traversal/ Local File Inclusion. Scanners, as a group, performed very poorly in detecting old, backup and unreferenced files, and un-validated redirect. Chen also summarized the research based on documentations, scanners audit features, scan barrier and input vector support, and authentication features.

3 METHODOLOGY

This paper is aimed at carrying out OWASP top 10 vulnerability assessment in some MDA's websites using Acunetix vulnerability scanner. In this section all materials, methods, steps and processes undertaking to achieve the paper's aim and objectives are listed and explained. Including the software tools used, how to identify vulnerabilities, techniques used for identification by the tool for the purpose of categorizations and analysis.

Web application security scanners are used in order to successfully carry out vulnerabilities analysis and categorization. The study collect security reports related to some key 10 websites. In this section of the research we have analyzed the OWASP TOP 10 Vulnerability found in MDA's Websites in Nigeria.

3.1 OWASP Top 10: Application Security

A1 – Injection: Under the right circumstances, third parties can inject a small quantity of code to trick an application into performing unintended actions. The most common (and most well-known) injection attack is an SQL Injection, where an attacker inserts an SQL statement into an application to, for example, dump the database contents onto the attacker. OWASP recommends you check incoming requests to determine their trustworthiness, and keep untrusted data separated from the systems that run your application.

A2 – Broken Authentication and Session Management: You know the user credentials of people accessing your systems, but do you know who is actually behind the keyboard? Hackers can hijack user identities and hide behind a genuine user ID to gain easy access to your data and programs. OWASP recommend Implementation of strong authentication and session management controls and ensure your users are who they say they are.

A3 – Cross-Site Scripting (XSS): An XSS vulnerability elevates the trust a user has given to a specific site to also include a second, potentially malicious, site. When the user permits certain actions to run on a site they believe to be secure, this can allow a malicious actor to modify the intended web page in interesting ways (resulting in the dissemination of sensitive data or the spread of malware). XSS vulnerabilities are common, but should be fairly simple to remediate. Separate untrusted, user-inputted data from active content within your webpage (for example, hyperlinks and the like).

A4 – Insecure Direct Object References: Most websites store user records based on a key value that the user controls (such as a username or email address). When a user inputs their key, the system retrieves the corresponding information and presents it to the user. An Insecure Direct Object Reference occurs when an authorization system fails to prevent one user from gaining access to another user's information. OWASP recommends you secure your authorization channels by implementing access control checks for each user-accessible object (such as files, webpages, and other information).

A5 – Security Misconfiguration: Security Misconfiguration is a general reference to application security systems that are incomplete or poorly managed. Security misconfiguration can occur at any level and in any part of an application, and thus is both highly common and easily detectable. There are myriad ways in which you may be vulnerable to software misconfiguration, so be sure to read up on OWASP's vulnerability report.

A6 – Sensitive Data Exposure: Unintended data display is a serious problem to anyone operating a web application that contains user data. Although OWASP points out that the full perils of insecure data extend well beyond the scope of the OWASP Top 10, they do recommend a handful of minimum steps, including to encrypt all sensitive data at rest and in transit and discard sensitive data as soon as you can.

A7 – Missing Function Level Access Control: A missing or improperly configured user access control system can grant users the ability to perform functions above their level (this vulnerability has some overlap with A4 – Insecure Direct Object Reference). Although it

can be difficult to avoid security vulnerabilities in function level access control systems, OWASP advocates several methods to secure your applications (including the establishment of "deny-by-default" rules to only allow function access to users you know and trust).

A8 – Cross-Site Request Forgery (CSRF): A CSRF attack involves sending a request to a vulnerable web application using a trusted user's credentials. Although an untrusted third party generates the request, the attacker uses the victim's browser to piggy-back on the victim's credibility. A CRSF attack exploits a trusted user's authentication to trick a system into performing a malicious action. To reduce risk of forgery, OWASP suggests that you include a unique, hidden token in every web request.

A9 – Using [Open Source Software] Components with Known Vulnerabilities: Open source development practices drive innovation and reduce development costs. But despite the benefits of open source software, the 2016 Future of Open Source Survey found that significant challenges remain in security and management practices. It is critical that organizations gain visibility into and control of open source software in their applications and Docker containers.

A10 – Un-validated Redirects and Forwards: When a web application accepts unverified input that affects URL redirects, third parties can redirect users to malicious websites. In addition, hackers can alter automatic forwarding routines to gain access to sensitive information. The Open Web Application Security Project suggests you avoid using redirects and forwards whenever possible, but if abstinence is unrealistic, don't let users affect the destination.

All the above web application security vulnerabilities are interlinked, one weakness can lead to the other. It is therefore, essential one understand concept of the application security landscape and how to mitigate risk. The site of OWASP is full of suitable information. The open source community supports Black Duck and of recent propelled black duck research, the security advisory board to uphold application security research world-wide.

The research framework that is involved in this paper is briefly discussed below.

The introduction and background of the study give a brief introduction of web application vulnerability and the various ways sensitive data could be exposed to threats that could be possibly caused exploits to the network.
Some notable challenges encountered in web application vulnerabilities are explained in details. Several related web application vulnerabilities works carried out by various authors are listed and reviewed in the course of this research.
All the materials and methods used in the analysis of the chosen web vulnerability scanner tools are stated including research instruments.

The user will input the URL (uniform resource locator) of the web application to be scanned and click on the scan button.

The scanning process involves crawling and parsing and the discovery of the vulnerabilities. During this process vulnerabilities are displayed until the scanning is completed, the displayed shows the discovered vulnerabilities and their location. The scanning process includes, crawling and fuzzing. After the scanning process is completed, the results are saved, the process continuous for several MDA's URL, the results are then categories into OWASP Top 10 vulnerabilities as shown in table 2.0.

Input: The URL of the web application to be tested. This is provided by the user who initiates the web scanning process.

Processing: This involves crawling all the web pages, fuzzing and identification of any weakness and firing inputs to check for any vulnerability.

Output: The results of processing are display for categorization and analysis.

Below Figure 1 illustrate system flowchart.

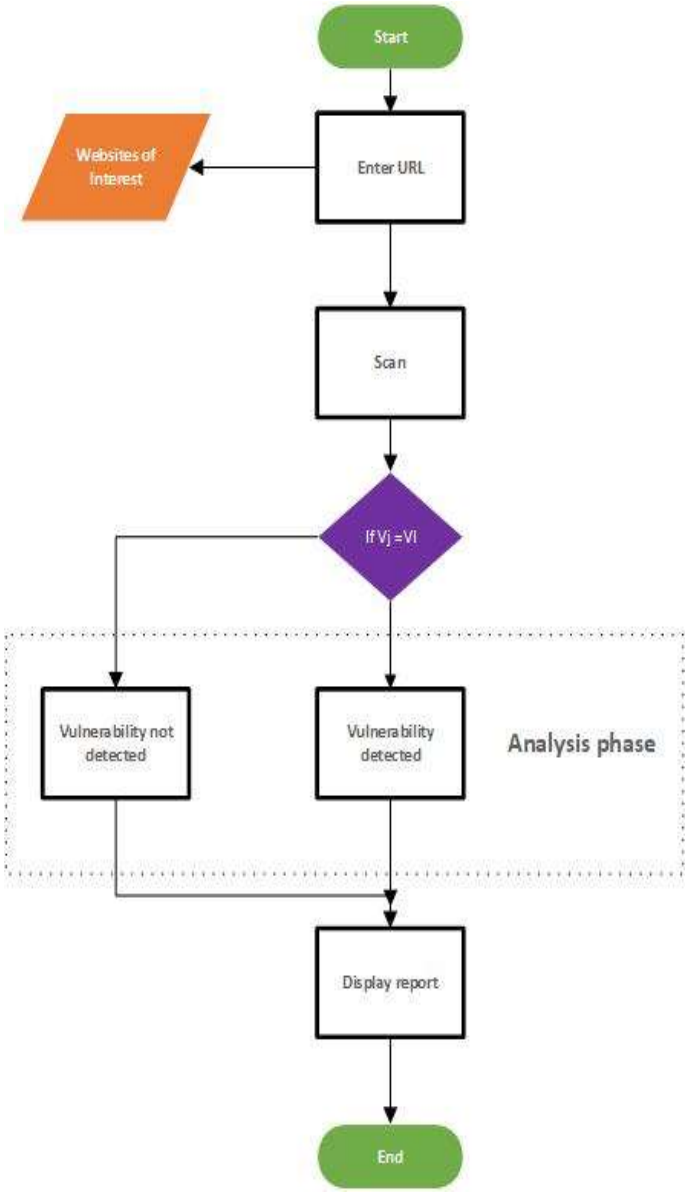

Figure 1. Flowchart of system framework.

Where:

V_j Represents Top 10 OWASP Vulnerability
V_l Represents threshold vulnerability index

3.2 Research Instruments

Tools and software used in this research work in order to successfully carry out the analysis all these are listed and explained below:

Hp laptop running windows 8.1, 64bit operating system was used to install and run the Acunetix web vulnerability scanner.

3.3 Working of Acunetix Web Vulnerability Scanner (WVS) Overview

Acunetix Web Vulnerability Scanner (WVS) is an automated web application security testing tool that audits your web applications by checking for vulnerabilities like SQL Injections, Cross site scripting and other exploitable hacking vulnerabilities [18]. Acunetix WVS scans any website or web application that is accessible via a web browser. It also offers a strong and unique solution for analyzing off-the-shelf and custom web applications including those relying on client scripts such as JavaScript, AJAX and Web 2.0 web applications. It is suitable for any small, medium sized and large organizations with intranets, extranets, and websites aimed at exchanging and/or delivering information with/to customers, vendors, employees and other stakeholders. Acunetix WVS works in the following manner:

1. The Crawler analyzes the entire website by following all the links on the site and in the robots.txt file and sitemap.xml (if available). WVS will then map out the website structure and display detailed information about every file. It also analyses hidden application files, such as web.config.

2. After the crawling process, it launches a series of vulnerability attacks on each page found, in essence emulating a hacker. Also, WVS analyses each page for places where it can input data, and subsequently attempts all the different input combinations. This is the Automated Scan Stage.

3. During the scan process, a port scan is also launched against the web server hosting the website. If open ports are found, Acunetix WVS will perform a range of network security checks against the network service running on that port.

4. As vulnerabilities are found, Acunetix WVS reports these in the ‘Alerts’ node. Each alert contains information about the vulnerability such as POST variable name, affected item, http response of the server

5. If open ports are found, they will be reported in the 'Knowledge Base' node. The list of open ports contains information such as the banner returned from the port and if a security test failed.

6. After a scan has been completed, it can be saved to file for later analysis and for comparison to previous scans. Using the Acunetix reporter a professional report can be created summarizing the scan [18].

4 RESULTS AND DISCUSSION

This section involves the discussion of results generated by use of Acunetix web application vulnerability scanner, this results are categories and analyse based on OWASP Top 10. Table 2 shows the categorized vulnerability detection result based on OWASP top 10 vulnerabilities.

4.1 Discussion of Results

To Scan Acunetix checks if the target in question is reachable and running a web server and serving requests over the HTTP protocol. Acunetix fingerprints the web server to identify a popular technologies that the web server might be using. This allows the scanner to identify the type of web server (e.g. Apache HTTP Server, Nginx, IIS…), the server-side language being used (e.g. PHP, ASP.NET, Java/J2EE, Python, NodeJS…) as well as the operating system the web server is running on. This information allows the scanner to automatically tune itself to the target to be scanned. The index file is requested from the web server. This is determined by the start URL (e.g. http://www.futminna.edu.ng/ will load index.html). Once a response is received, DeepScan is launched, executing any JavaScript present on the web page. The Crawler, hand-in-hand with DeepScan will follow links, map input fields and parameters. This contributes to building a list of directories and files within the site. As Acunetix discovers vulnerabilities, alerts are reported in real-time. Each alert produces detailed information about the vulnerability, recommendations on how to fix it, as well as several links through which the user can learn more about the reported vulnerability. Its performance is not 100% accurate but it has a higher capacity to detect more vulnerabilities.

	MDAs											
OWASP Top 10 Vulnerabilities	1	2	3	4	5	6	7	8	9	10	Total	%
A1-Injection	3	0	0	0	4	0	0	0	0	0	7	2.7
A2-BASM	1	1	0	2	0	0	0	0	0	1	5	1.9
A3-XSS	1	1	0	23	1	0	0	1	0	1	28	11
A4-Insecure Direct Object Ref.	7	65	17	8	2	0	0	19	2	8	128	49
A5-Security Misconfiguration	6	1	3	2	4	1	1	1	2	6	27	10
A6-Sensitive Data Exposure	3	2	6	1	4	0	2	0	0	1	19	7.3
A7-MFLAC	0	1	1	1	1	1	1	1	1	1	9	3.5
A8-CSRF	1	14	4	3	1	1	1	2	1	8	36	14
A9-UCKV	1	0	0	0	0	0	0	0	0	0	1	0.4
A10-URF	0	0	0	0	0	0	0	0	0	0	0	0
Total	23	85	31	40	17	3	5	24	6	26	260	

Table 2. Categorized Vulnerability Detection Result Based on OWASP Top 10 Vulnerabilities

4.2 Vulnerability Analysis

We have reviewed security report of 10 MDA's websites. We have found vulnerabilities in all websites. From the OWASP TOP 10 vulnerabilities mentioned in Table 1, we have observed nine vulnerabilities viz. A1, A2, A3, A4, A5, A6, A7, A8 and A9 in MDA's websites. The % of web resources affected by these vulnerabilities is depicted in figure 2. The web resource includes files, directories web pages that reside on webserver and together work as functioning website.

As shown in below Figure 2, vulnerability A4 is the major contributor in MDA's website with 49% vulnerability. The remaining vulnerabilities remain less than 14.1%.

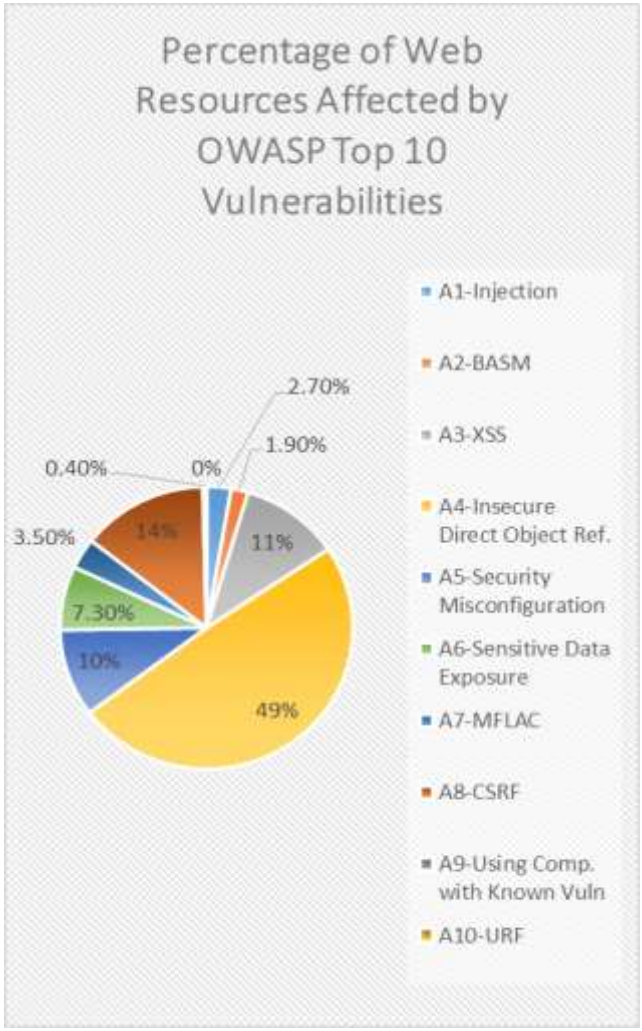

Figure 2. Percentage of web Resources affected by OWASP Top 10 Vulnerabilities

To get a better picture of the overall trends, figure 3. shows the graph of web resources affected by OWASP Top 10. With Insecure direct object reference at maximum point 49% and Un- validated redirects and forwards with No vulnerability. In summary, the research analyses shows that A4 Insecure direct object reference was the most widespread critical vulnerability, follow by Cross – site request forgery (CSRF) to the least widespread vulnerability Using components with known vulnerabilities.

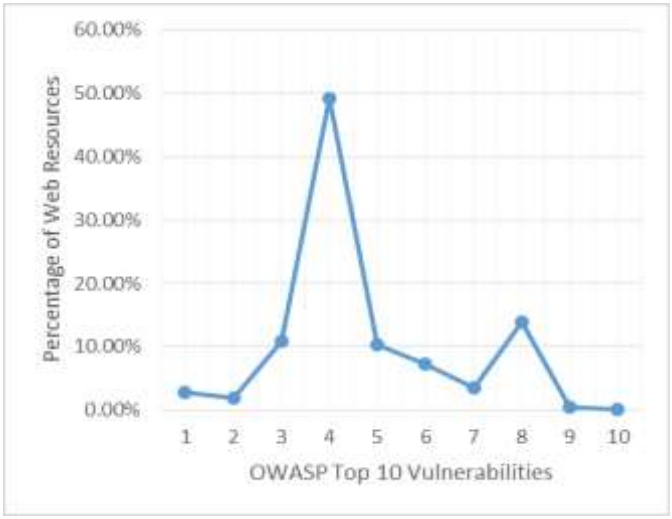

Figure 3. Graph of Web Resources Affected by OWASP Top 10

1. Injection
2. Broken Authentication and Session Management
3. Cross – Site Scripting (XSS)
4. Insecure Direct Object Reference
5. Security Misconfiguration
6. Sensitive Data Exposure
7. Missing Function Level Access Control
8. Cross – Site Request Forgery (CSRF)
9. Using Component with Known Vulnerability
10. Un – validated Redirect and Forward

The research work also analyzed vulnerabilities according to the risk level. In security risk level wise analysis, the research observed vulnerability in 251 web resources according to the risk level viz. high, medium, low and informational.

In this section, OWASP TOP 10 vulnerabilities was analyzed by cross tabulating it with risk level. Table 3. Illustrates cross tabulation of risk levels with OWASP TOP 10 vulnerabilities found in MDA's websites.

Table 3. OWASP TOP 10 vulnerabilities vs Risk Levels

OWASP Top 10 Vulnerabilities	High	Medium	Low	Informational	Total
A1- Injection	7	-	-	-	7
A2 - BASM	1	1	2	-	4
A3 - XSS	26	2	-	-	28
A4 – Insecure Direct Object Ref.	-	-	23	105	128
A5 - Security Misconfiguration	6	9	6	1	22
A6-Sensitive Data Exposure	4	-	4	7	15
A7 - MFLAC	-	-	8	1	9
A8 - CSRF	-	9	27	1	37
A9 - UCKV	-	1	-	-	1
A10 - URF	-	-	-	-	-
Total	44	22	70	115	**251**
% Total	**17.53**	**8.77**	**27.88**	**45.82**	

The below figure 4, shows the Vulnerability Risk Level Wise % of Web Resources.

Figure 4. Vulnerability Risk Level Wise % of Web Resources

As seen from the figure 4, vulnerabilities having Informational risk level is the most predominant among the MDA's websites with 45.82%. Low risk level 27.88%, Medium risk level 8.77% and High risk level 17.53%. Web resources with high risk vulnerability needs immediate attention by concern authorities to halt web security risk.

From the above table 3, we can easily deduce that A1, A2, A3, A5 and A6 are contributing to the critical high risk to the MDA's websites. A3 is the major contributor of 17.53% in high risk followed by A1, A5, A6 and A2 respectively. A4 is the largest contributor of informational type risk level with more than 93% prevalence in it risk category and second to low type. Similarly, A8 is the main contributor to medium and low risk level which tally with A5 (Security Misconfiguration). From the above, it is clear comparatively that A4, A8 and A3 are generated due to Insecure Direct Object Reference, Cross-Site Request Forgery and Cross-Site Scripting respectively. Vulnerabilities in web resource mainly affects confidentiality, integrity and availability of information resource. Regular website security auditing is required to restrain the constantly emerging threats and keep MDA's websites secure and safe.

The graph below depicts vulnerability risk levels % of web resource

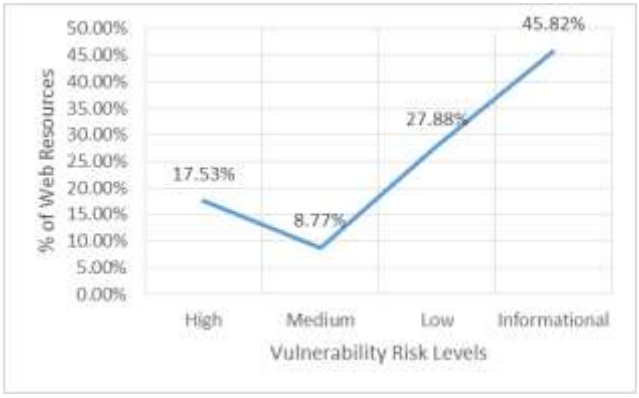

Figure 5. Percentage Vulnerability risk levels

5 CONCLUSION

Having surveyed 10 MDA's, these findings are extremely troubling. In the race to produce user-friendly interfaces and centered apps, MDA's are leaving their precious data wide open to cyber criminals. One look at the news headlines shows cyber-attacks are all too common. With nearly half of web apps containing both informational and low security vulnerability risk level such as Insecure Direct Object Reference and Cross-Site Request Forgery (CSRF), it's just like leaving your wallet or unlocked phone lying around in a public place. It's more a question of how long it takes, rather than if at all, before you are compromised.

It is clear from above that MDA's websites required web security due to its vital role in

information dissemination and citizen centric services. In this paper, we can easily conclude that there are vulnerabilities in MDA's websites among them A4-Insecure Direct Object Reference is the single largest contributor of web security risk in MDA's websites. In this study, A4 found to be affecting 49% web resource in various MDA's websites. Majority of the vulnerabilities belongs to the informational risk group viz. high, low, informational with 17.53%, 27.88%, 45.82% respectively. Medium risk vulnerabilities are very few with 8.77% authorities must deal with these vulnerabilities urgently to prevent any risk to the MDA's websites.

6 RECOMMENDATION

Acunetix is the market leader in web application security technology, founded to combat the alarming rise in web attacks. Its products and technologies are the result of a decade of work by a team of highly experienced security developers. Aside recommendations on possible ways to fix identified vulnerabilities also defense mechanism for all vulnerabilities should be included in the application. I recommends National Information and Technology Development Agency (NITDA) to design a dynamic comprehensive security policies for Ministries, Departments and Agencies (MDA's) of Government in Nigeria.

REFERENCES

1. Positive Technology.: Web application vulnerability statistics, https://www.ptsecurity.com/
2. Mulpuru, S., Vikram, S., Patti, F. E., Andy, H., Douglas, R.: US Online Retail Forecast, https://www.forrester.com/report/US+Online+Retail+Forecast+2011+To+2016/-/E-RES60672
3. Morgan, W.: Anonymous Nigeria Hacks Government Websites, Declares Cyberwar against Corruption, Poverty, Theft, http://www.ibtimes.com/anonymous-nigeria-hacks-government-websites-declares-cyberwar-against-corruption-2257401
4. The Open Web Application Security Project (OWASP) Foundation: Top Ten Web Application SecurityRisks,http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project
5. Nahari, H., Krutz, R. L.: Web Commerce Security: Design and Development, (OWASP 2011). https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project
6. OWASP Home Page, https://www.owasp.org/index.php/Main_Page.
7. IBM Developer Work Library, http://www.ibm.com/developerworks/library/se-owasptop10/
8. OWASP Category Project, https://www.owasp.org/index.php/Category:OWASP_Project
9. Singh, U. K., Chanchala, J., Neha G.: Information security assessment by quantifying risk level of network vulnerabilities. *International Journal of Computer Applications*, vol. 156.2, pp. 37-44, 2016
10. Mittal, P., Kaur, A.: E-Governance - A challenge for India, *International Journal of Advanced Research in Computer Engineering & Technology* (IJARCET 2013), vol. 2, no. 3, 2013.
11. Chander, S., Kush, A.: Vulnerabilities in web pages and web sites. *International Journal of Advanced Research in IT and Engineering, vol. 1*(2), pp. 47-57, 2012
12. Marco, V., Nuno, A., Henrique, M.: Using web security scanners to detect vulnerabilities in web services. *International Conference on Dependable Systems & Networks*. IEEE, 2009.
13. Rafique, S., Humayun, M., Gul, Z., Abbas, A., Javed, H.: Systematic Review of Web Application Security Vulnerabilities Detection Methods. *Journal of Computer and Communications, vol. 3*, pp. 28-40, 2015.
14. Karumba, M. C., Ruhiu, S., Moturi, C. A.: A Hybrid Algorithm for Detecting Web Based Applications Vulnerabilities. *American Journal of Computing Research Repository, vol. 4*(1), pp. 15-20, 2016. doi: 10.12691/ajcrr-4-1-3
15. Kagorora, F., Li, J., Hanyurwimfura, D., Camara, L.: Effectiveness of Web Application Security Scanners at Detecting Vulnerabilities behind AJAX/JSON. *International Journal of Innovative Research in Science, Engineering and Technology, vol. 4*(6), pp. 4179-4188. (IJIRSET 2015)
16. Ertaul, L., Martirosyan, Y.: Implementation of a Web Application for Evaluation of Web Application Security Scanners, in Proc. 2012 International Conference on Security and Management, pp. 82-89, 2012.
17. Chen, S.: Price and Feature Comparison of Web Application Scanners, http://sectoolmarket.com
18. w4rri0r - Hacking Is Not A Crime - It's an art of Awareness,(2017) http://www.w4rri0r.com/index.php/component/content/article/2-penetration-testing-tools/297-acunetix-web-vulnerability-scanner