



Agent-based Vulnerability Assessment of Government of Kenya Web Applications

Wekesa Bernard Bongo

P58/76392/2012

Master of Science in Computer Science

Supervisor: E. A. Miriti



Introduction

- This study involved theory review, evaluation and development of an application system using multi-agents that implemented the proposed solution to demonstrate its application in real-world setting.
- It involved an agent based system using the JADE platform
- The system allows testing for XSS and SQL injection attack vulnerabilities in websites.



Problem Statement

- Recently, the Government webserver was hacked and accessed by unauthorized person who defaced its websites, making web services unavailable.
- In an article (Fayo 2012) says the hacker known as *direxer* took down 103 government of Kenya websites overnight citing unfixed programming errors in code.
- Government webserver lacks the system to sanitize user provided data and thus enabling attackers to inject attack code written in different static or dynamic contents or other browser supported technology
- cross-site scripting (XSS) and SQL injection consists in the exploitation of input validation flaws, with the purpose of injecting arbitrary script code which is later executed at the web browser of the victim
- Several prevention technique attempts have been made by many scholars but the use of multi-agents have not been fully exploited.



System Objectives

The main objective of the project is to study, identify, design and implement an agent based security solution suitable for web testing web application vulnerabilities. It will entail researching on the following:

- Formulate an agent-based software application for testing web application vulnerability
- Provide a tool for testing web applications in the development environment before uploading in the production environment.



Abstract

- The growth of the internet in recent times has led to the spread of information crimes in renewed and changing ways. Today almost all organizations including the government of Kenya have improved their performance through allowing more information exchange within and without their organization using web support.
- Databases are central to the modern websites as they provide necessary data and store critical information such as user credentials etc. these websites have been continuously targeted by highly motivated malicious users to acquire their intentions.



Abstract

- Structured Query Language (SQL) injection and Cross Site Scripting Attack (XSS) is perhaps one of the most common application layer attack techniques used by hackers to deface websites, manipulate and/or delete the database contents through inputting unwanted command strings and using session cookies.
- SQL injection and XSS attacks are ranked as the two top most vulnerability attacks by the Open Web Application Security Project (OWASP) top 10, 2013 vulnerability list and has resulted in massive attacks on a number of websites including the government of Kenya ones recently.
- Agent orientation is emerging as a dominant research area and also prevails as a new paradigm constructing solutions to problems. Agents provide developers and designers with a way of structuring applications around autonomous and communicative elements.



Abstract

- In this study, we present a system that uses multi-agents to detect both SQL injection and XSS attacks vulnerabilities on web applications. The system has been developed in Java programming language and using Prometheus methodology as an Agent Oriented Software (AOS). It will specifically target websites in development environment for testing the vulnerabilities before being hosted in the production environment. We have also incorporated the testing of already hosted websites for the two vulnerabilities. Tests against a set of SQL injection and XSS attacks show the effectiveness of the proposed system to be used by web developers and owners of websites.



Limitations/Assumptions

- This study assumed that every ministry and government agency had a website in place which is updated and that the website is visited frequently by various users.
- The study was limited to the use of multi-agents framework towards solving XSS and SQL injection attacks. It involved development of an agent based system that implements the detection of XSS and SQL injection vulnerabilities because they are the most powerful and easiest attack methods on the web application.

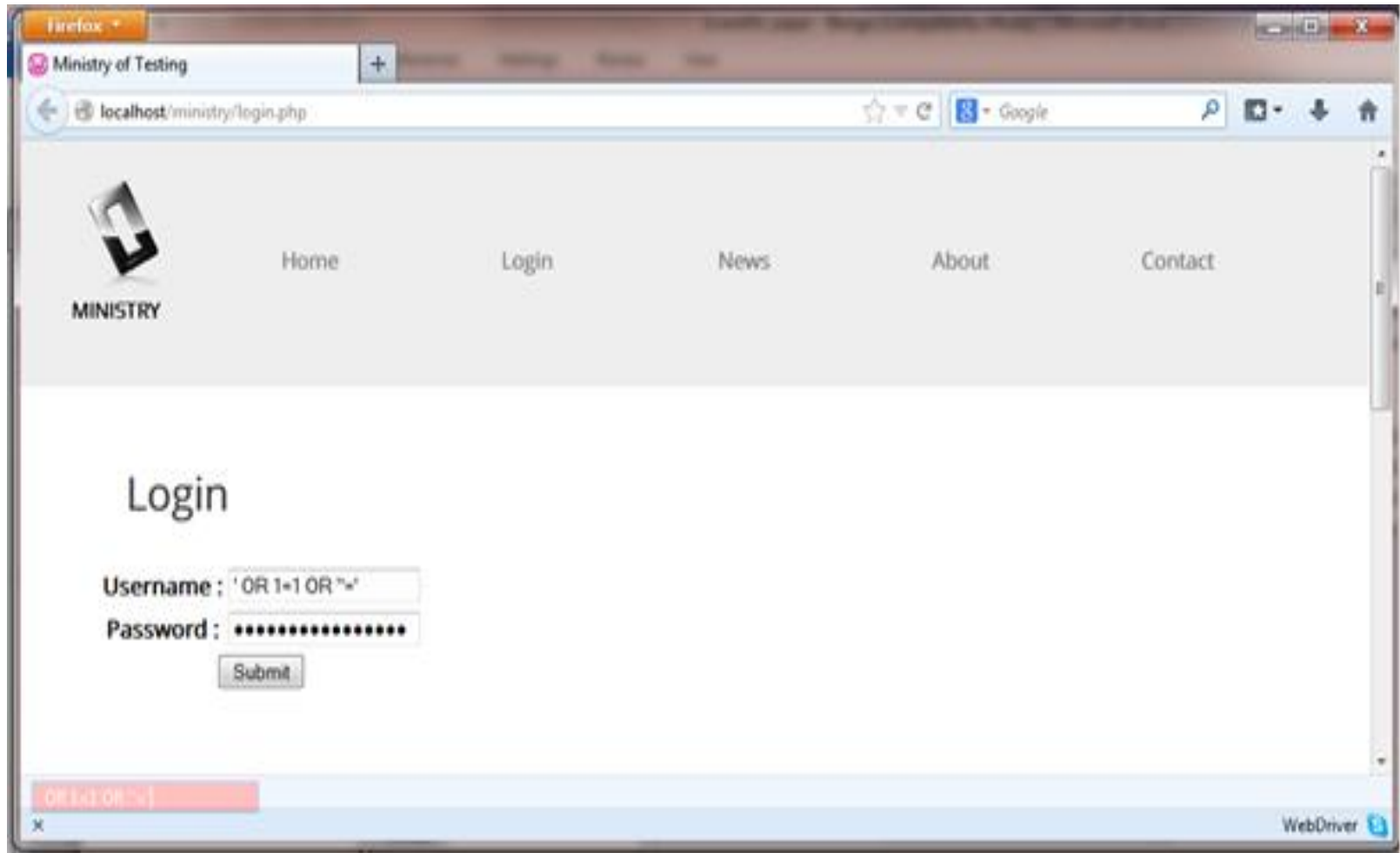
Sample Results



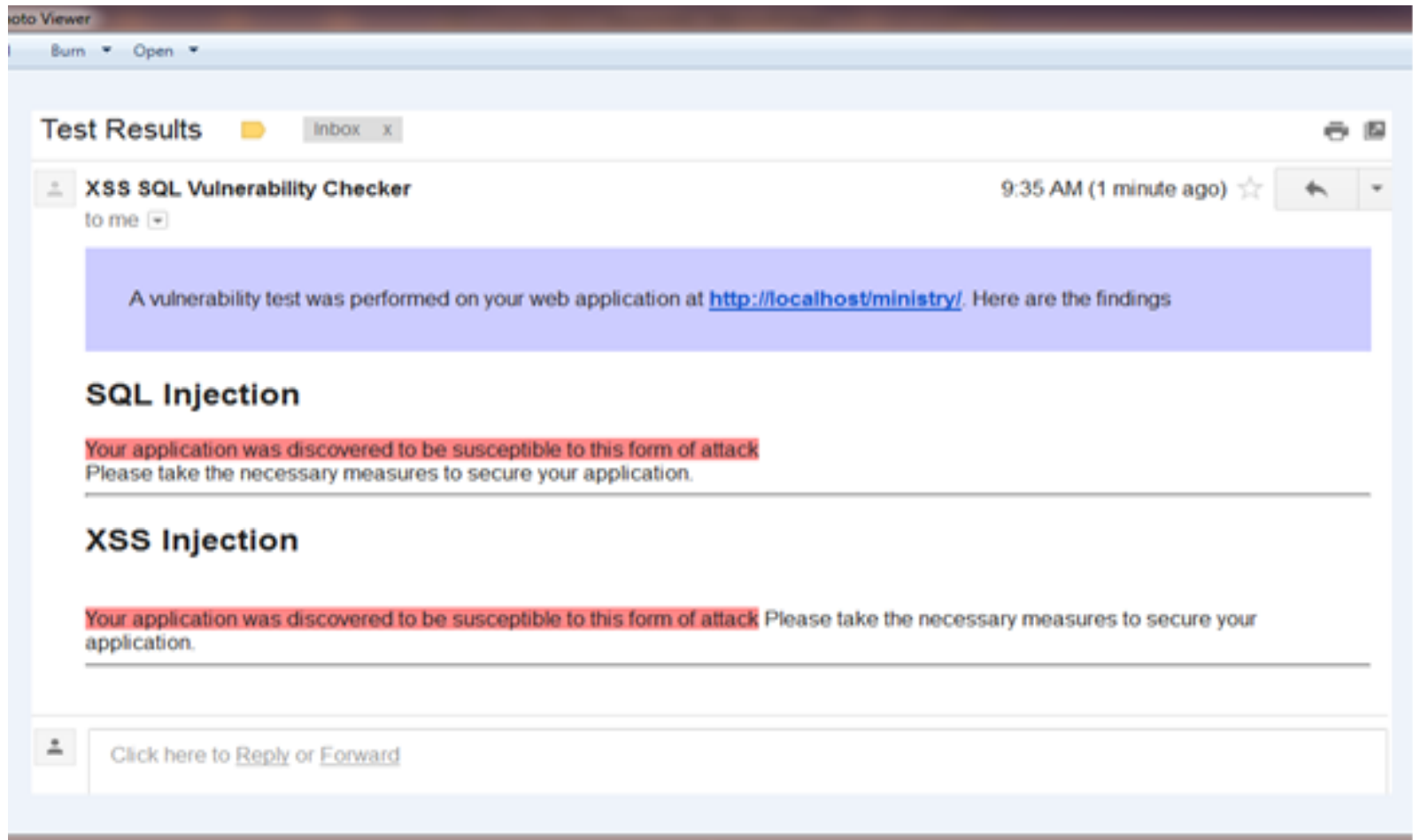
Sample Results



Sample Results



Test Case/Evidence



The screenshot shows an email client window titled "Photo Viewer" with a menu bar containing "Burn" and "Open". The email is titled "Test Results" and is in the "inbox" folder. The sender is "XSS SQL Vulnerability Checker" and the recipient is "to me". The email was received at "9:35 AM (1 minute ago)".

The main body of the email contains a purple highlighted text block: "A vulnerability test was performed on your web application at <http://localhost/ministry/>. Here are the findings".

Below this, there are two sections of findings:

- SQL Injection**
Your application was discovered to be susceptible to this form of attack
Please take the necessary measures to secure your application.
- XSS Injection**
Your application was discovered to be susceptible to this form of attack
Please take the necessary measures to secure your application.

At the bottom of the email, there is a footer with a person icon and the text "Click here to [Reply](#) or [Forward](#)".



Conclusions

- This research project was set out to develop a system using multi-agents to detect SQL injection and XSS attack vulnerabilities that hackers use to deface websites. It was motivated by the recent defacement of government of Kenya websites thus causing panic and denial of services to citizens. Cross-site scripting and SQL injection are the two most serious attacks that are used to compromise websites and reveal sensitive information by unauthorized parties. This research project has proposed a web security system that checks these two vulnerabilities.



Conclusions

- This system is applicable to almost all web-based and computer supported cooperative systems over the web. It will protect the interests of web administrators as well as clients. Tests using various XSS and SQL injection attacks on the proposed system attest to the quality of the design. This together with other solutions available at the clients' side would provide confidence to site users.



Future Work/Recommendations

It has been proved that agents can be used to do the work for us by specifying to them the terms of reference, otherwise known as ontologies. It is recommended that in future more vulnerabilities will be solved by multi-agents and therefore further research on extending and refining the use of agents should be pursued to verify their usefulness. Since my research has concentrated on the two most serious ones according to OWASP top 10, 2013, there is need to conduct further research to other types of vulnerabilities using agents as well.

References

1. B.Henderson-Sellers P. G. Brian Henderson-Sellers Agent-Oriented Methodologies, April, 2005.
2. Elhakeem, Y.F.G.M. & Barry, B.I. a., 2013. Developing a security model to protect websites from cross-site scripting attacks using ZEND framework application. 2013 International Conference on Computing, Electrical and Electronic Engineering (Iccee), pp.624–629. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6634012>.
3. Fayo, G 2012, 'Hacker knocks out 103 Kenyan govt sites', IT Web security, 18 Jan 2012, viewed on 29 March 2014, http://www.itweb.co.za/index.php?option=com_content&view=article&id=50696
4. https://www.owasp.org/index.php/Top_10_2013-Top_10, [Accessed on 6/3/2014]
5. J. Guillaumier, "Cross Site Scripting - XSS - The underestimated exploit" [Online] <http://www.acunetix.com/websitesecurity/xss.htm>.
6. Sedaghat, S. & Sydney, W., A Dynamic Web Agent for Verifying the Security and Integrity of a Web Site ' s Contents. , pp.330–337.
7. Web Application Security and the OWASP Top 10, Available at: www.sapien.com/assets/ImageDownloader/813/Web_Application_Security.pdf. [Accessed on 5th march 2014]