Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 10 (20): 2447-2453, 2010
ISSN 1812-5654

Evaluating the Vulnerability and the Security of Public Organizations Websites in Jordan

[1,2]Amer N. Abu Ali, [1]Alaa K. Alnaimat and [2]Haifa Y. Abu-Addose
[1]Department of Computer Information System, Philadelphia University, Jorden
[2]Department of Computer Information System, Philadelphia University, USA

Abstract: In this study, we evaluate governmental agencies from a security perspective. In particular, we focus on assessing the vulnerability of those websites and their security weaknesses or holes. We use several selected metrics available for this purpose. As a result we show those websites strengths and weaknesses from the vulnerability viewpoint as well as classify those weaknesses or problem for their severity levels, hence classify those websites from their security levels from the most secure website to the least secure one. Finally, we provide recommendations on how some of those weaknesses can be reduced or eliminated.

Key words: Cryptographic, confidentiality, integrity, availability, vulnerability

INTRODUCTION

Nowadays websites and applications are important and integral to our daily life activities such as: education, employment, government, commerce, health care and others. As a result, there are increasing concerns about the reliability and security of the developed websites and applications, in order to ensure that services will be provided to customers with the maximum possible security, to guarantee the integrity of the system and the privacy of online users.

A common security infrastructure is established by International Standard Organization (ISO) for IT security evaluation to help in the assessment of security concerns in IT products. The so called 'Protection Profile' provides the minimum set of security requirements for security protection. For example Windows 2000 takes the following security measurements into account [Common Criteria for Information Technology Security Evaluation July 2009 http://commoncriteria.org/cc/cc.html]:

- Identification and authentication
- Accounting and auditing
- Access control
- System security management
- User data protection
- Security function protection
- Cryptographic support
- Resource utilization
- Session locking
- Configurable access banners
- Interior data replication
- Session initialization
- Trusted path

Standardize security criteria can help us develop accepted secure applications.

In computer systems security requirements must be taken into accounts which prevent the damage or loss of valuable information especially in mission-critical systems.

The general requirements are (Nguyen and Johsonn, 2006):

- **Confidentiality:** Deny or prevent any unauthorized access to information, using access control, passwords, encryption, or provide higher level of privacy and conducting a policy of ethics
- **Integrity:** prevention of altering or corrupting transaction data and assuring that any modification of created information is authorized in order to keep data accurate and complete. This is done by configuration and auditing management
- **Availability:** Availability is ensuring that authorized users have access to information and associated assets when required. This done by many techniques such as data backup plan and disaster recovery plan which include business continuity plan or business resumption plan

However, there are two major types of security solution in computer environment (Drum, 2006): instruction detection system (IDS) and instruction prevention system (IPS). Which provide secure techniques for protecting the system from both external and interior attacks. For websites, not all conventional web security techniques comprehensively applicable, because of the dynamic nature of websites security requirement. In market there are multiple types

Corresponding Author: Amer N. Abu Ali, Department of Computer Information System, Yarmouk University, Jorden

of IDS include: network-based, application-based and host-based IDS. Each one of those focuses on one of the computer system components (Eschelbeck and Krieger, 2003).

Vulnerability assessment attracts many organizations in order to direct their effort toward overcoming all the weaknesses of vulnerability problems. Examples of entities who work on this subject include: National Cyber Security Division (NCSD), sponsored Software Assurance Metrics and Tool Evaluation (SAMATE).

STATEMENT OF PROBLEM

In Jordan, one of the main barriers that limit our abilities to build e-commerce websites or businesses is the high standard required for security to ensure that customers can buy or sell online securely and safely. Large legal problems may decrease the customers' level of confidence on such business and deprive them the opportunity to take advantage of this technology.

In this research, we try to study how people assess the vulnerability of websites and their networks or channels. The study tries to gather all required characteristics or metrics to build a security framework. This general framework is illustrated in Fig. 1, can be used by any website evaluator, auditor, or quality assurance member to verify that such website has the basic or minimum reasonable requirements that enable the website to be involved on online transactions.

The process will first specify the major vulnerability weaknesses in any website. Once those weaknesses are specified, we will try to find online free tools that can assess or measure those elements or metrics. We will select several websites to be our sample study for the research purpose, E-government sectors in Jordan.. In order to assess the vulnerability in a system, we consider the following factors Port scan: -by checking TCP ports.

- Vulnerability test: For more than 3,600 weaknesses
- Mail proxy checker: To specify spam sent from the server
- Weak username and password
- Website spidering: Testing the website for cross-site-scripting (XSS) and HTML code

There is another problem facing the security industry, which is the way vulnerabilities are named or grouped. The same vulnerability can have multiple names which is confusing to the security participants; who work in the practical field. To solve this problem Common Vulnerability and Exposure (CVE) is established, in order to contain a set of standard for naming convention of security (InfoSec Reading Room, 2001) Vulnerability Assessment. SANS Institute, http://www.sans.org/reading_room/whitepapers/tools/ pocket-nessus_407.

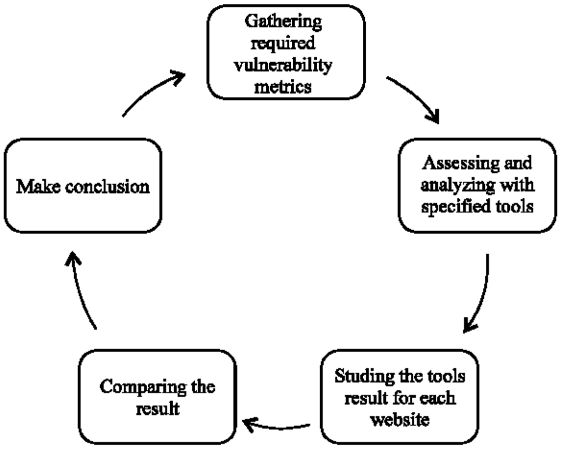
Gathering required vulnerability metrics
Assessing and analyzing with specified tools
Studing the tools result for each website
Comparing the result
Make conclusion

Fig. 1: The general proposed security framework

THE EVALUATION OF BANKS WEBSITES SECURITY

Many researches are interested in evaluating the security of bank websites, because it is considered as the most important requirement for having a successful bank business. For instance website authentication is assessed in order to provide the users all the secure guides when they log in their e-bank accounts. The study examines 67 bank customers by a simulation built for this purposes. They observe the user behavior toward some security vulnerability indicators which are: HTTP presence and absence from the URLs, the site authentication-image and the web browser security agent. But in spite of it were obvious indications for those vulnerabilities, the banks clients behave insecurely. The research goal is summarized in protection of the banks websites from the vulnerabilities exploited by man-in-the-middle and give the banks customers with security guide line that must follows when navigate the bank website. While, Falk *et al.* (2007) addressed the necessity for a secure framework in banks websites, not all bank clients perform their money transactions via websites. Because 79% of bank websites have at least one vulnerability, 68% of bank websites have two flaws and 10% of websites have all the flaws. They analyze 214 of financial websites for the vulnerabilities. Then they evaluate the websites against the following five flaws: break in the chain of trust, presenting secure login options on insecure pages, contact information/security advice on insecure pages, inadequate policies for user ids and passwords, e-mailing security sensitive information insecurely. Then they result in: 30% of the website break the chain of trust, 47% present a login page on an insecure page, 55% put valuable information at insecure pages and 31% provide e-mail addresses as user names, however just 24% of the sites were free of those design flaws. An automated tool is used for detection of these five flaws.

Both Jahangir and Begum (2008) provide a conceptual framework of compromising both of usability and security through the banks customer's attitudes, toward very critical secure system used by banks.

DIFFERENT VULNERABILITY FRAMEWORKS

Sun *et al.* (2006) developed a systematic framework that measures the trust quantitatively with mathematical properties of trust, in addition to the dynamic properties of trust. They built trust model that its quantification depends on three axioms for the basic rules in third party trust, axiom1: state rule for concatenation trust propagation, axiom2: describe the rule for multiple propagation, axiom3: addresses correlations among recommendation.

While, Eusgeld *et al.* (2008) proposed a framework for vulnerability analysis of critical interconnected network infrastructure, correlated with object-oriented modeling that's for enhancing the vulnerability screened scenarios, applied on Swiss high-voltage grid, which is considered as critical complex interconnected system.

METHODS AND TOOLS FOR VULNERABILITY ASSESSMENT

Vulnerability in any software system attracts the attackers to exploit the vulnerable system and so, a secure system is needed, for that Halkidis and Chatzigeorgious (2006) built two systems for estimating the resistance of particular security patterns against STRIDE attacks, the first one was concerned about the security patterns and the second one without, in order to evaluate these secure patterns in the terms of STRIDE attacks. They aimed to introduce a new security patterns and upon those patterns, to use an effective security tools, for the evaluation of computer system security. For their mission they used AppScan vulnerability scanning tool then conducted other vulnerability assessment tools, after that they selected the STRIDE attacks as a vulnerability measures. The first system is a conventional e-commerce application, without security patterns and multiple attacks are injected. On the other hand; they used the second system which includes security patterns and the attacks sources are omitted. The platform for both systems is J2EE, for application server JBoss 4.0.3 is selected and MySQL 5.0 for a database. The first system attacked by various types of vulnerabilities which is: eleven cross site scripting attacks, three SQL injection attacks, HTTP response splitting attacks, three eavesdropping and sex of Servlet member variable race conditions. In addition it is not use SSL. In contrast to the first system, the second one uses SSL and have multiple security patterns are:"one instance of the secure proxy pattern, login tunnel variant, one instance of the secure pipe pattern, seventeen instances of the Secure Logger pattern, Secure Log Store Strategy, a twenty one instances of the Intercepting Validation pattern and nine instances of the Container Managed Security pattern". The assessment of both systems conducted through two approaches, automatically by AppScan and manually by contest newsgroup.

BUILDING NEW VULNERABILITY DETECTION ALGORITHM

Again, Xie and Aiken (2006) developed a static algorithm for tools to detect the vulnerability on PHP scripting language used for building server-side web applications that have been widely used. Static analysis for scripting language can reliably detect a critical vulnerability on the web application. They added some feature such as: including the program code, variable that change during the execution, operations with semantics, wide use of hash tables and regular expressions. Tree-tier analysis was used for capturing information in a decreasing level of details at the intra-block analysis, intra-procedural analysis and inter-procedural analysis. Then they illustrated the using of the static algorithm to find the SQL injection vulnerability and how to do so with cross site scripting vulnerability (XSS). Then they verified the implemented tools by applying it upon six web applications open source PHP files, as a consequence they found 105 new vulnerabilities. Finally they analyzed 2 case studies about vulnerabilities in PHP-fusion which is content management system built on PHP and My-SQL which contain 16,000 line of PHP code.

Since, the SQL injection vulnerability has 10% of the overall vulnerabilities from 2002 to 2007. Thomas *et al.* (2009) developed an algorithm named as PSR and a corresponding tools, which aimed to remove SQL injection vulnerability, this vulnerability allow unauthorized access to the database by the hacker, then stealing the valuable information in that database. The PSR algorithm analyze the source code searching for SQL injection vulnerability by separating the SQL structure from the SQL statement, in addition creating assistant vector used for including any new string and the algorithm generate a new string object. After implementing the algorithm they conducted an empirical study to assess this algorithm, the consequent results where: PSR algorithm delete 94% of the SQL injection vulnerabilities of the case studies. The 6% of the SQL injection vulnerabilities where not delete

THE MAIN VULNERABILITY TOOLS

The main selected tools were: the Wikto, the Acunetix Web Vulnerability Scanner,cgi and the NStalker free edition. These tools starts with the typing the URL of the studied website and then analyzing it page by page the website and even its transcript by a predefined algorithm which was designed by highly specialized companies to measure those issues.

Wikto measure multiple vulnerability metrics, it is not a web application scanner, but it finds the directions and files on websites looking for sample scripts that can be abused or finds known vulnerabilities in web server implementation itself, that will be mentioned in details in the experiment chapter but the Acunetix Web Vulnerability Scanner had been developed since 1997, in order to detect and analyzed vulnerabilities. The free vision of this scanner crawls web site, automatically analyzes the web applications and finds only Cross site scripting vulnerability with high severity level. It measure only one type of the vulnerability which is the cross site scripting (XSS). XSS is defined as is a type of computer security vulnerability typically found in web applications which allow code injection by malicious web users into the web pages viewed by other users [http://en.wikipedia.org/wiki/Cross-site_scripting,]; A hacker may use XSS to send a malicious script to a user. The end user's browsers don't know exactly which script that should be trusted and there are no ways to know and they will execute the script. Because they think that the script came from a secure and trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by your browser and used with that site.

WEBSITE UNDER STUDY

Selecting the websites was a major challenging subject, so it was based on a highly personal data. In addition to a highly classified governmental data and information, which may lead to breach in a country security or even failure in the development of an electronic easy accessible governmental system for better and faster data retrieval by the citizen, which also may reflect on the governmental plans to develop an electronic commerce system and when the trust are lost by the citizens, the whole project will fail and will not reach to the zero ground level. The selected websites (Six) are shown in Table 1.

Table 1: List of websites under study (WUS)

Governmental/Website	Website URL
Civil Service Bureau	http://www.csb.gov.jo/
Ministry of Education	http://www.moe.gov.jo/
Ministry of higher education	http://www.mohe.gov.jo/
Ministry of Information Communications and Technology	http://www.moict.gov.jo/MoICT
Foreign Ministry	http://www.mfa.gov.jo/wps/portal/FMArabicSite
Ministry of Interior	http://www.moi.gov.jo/

EXPERIMENTS AND RESULTS

The scope of the analysis phase is to remotely audit and analyze the websites under the study. This provides a hacker's eye view of the websites to discover its security vulnerabilities and weaknesses to possible hacker penetration or attack. This chapter is divided into two sections; the first section is evaluation of each Websites under Study (WUS) with corresponding tools, the second is analyzing security levels for each WUS.

EVALUATION OF EACH WEBSITE UNDER THE STUDY WITH CORRESPONDING TOOLS

In order to achieve our objectives, we consider four vulnerability scanner tools: wikto, CGI, acunetix web vulnerability scanner and N-Stalker free 2009. Those in turn applied to all websites under the study.

ACUNETIX WEB VULNERABILITY SCANNER RESULTS ANALYSIS

Here, the description of the analysis obtained from acunetix web vulnerability scanner will be mentioned.

Acunetix web vulnerability scanner had been developed since 1997, in order to detect and analyzed vulnerabilities. The free vision of this scanner crawls web site, automatically analyzes the web applications and finds only Cross site scripting vulnerability with high severity level.

By applying acunetix to our 6 Websites Under Study (WUS), we gain results only for four out thirteen websites which are: ministry of higher education,. The results have shown in Table 3, with their respective scan time that has been taken for each test.

- Ministry of higher education

Table 2: Acunetix result for ministry of higher education

No.	Affected Item	Vulnerability type	Severity	No. of Items	Scan Time
1	Errorpages.aspx	XSS	High	26	9 h,54 min

Total No. of gross site scripting: 26

Table 3: The summarized Results of the WUS

Summary of Acunetix Scanner of website under the study

Website name	Scan time
Ministry on information and communication technology	4 h, 43 min
Ministry of education	4 h,58 min
Ministry of forgien	15 h, 24 min
Civil service bureau	2 h,2 min
Ministry in internal	29 h,50 min

In Table 2, ministry of higher education (MOH) website, the results that had been got was in 26 cross site scripting vulnerabilities which is considered as high severity, this scan took 9 h and 54 min.

CGI SCANNER RESULTS ANALYSIS

Our mission here was to retrieve all the results, then organize it in readily way, after that try to understand each vulnerability occurs and finally determine the severity level for each vulnerability.

The above pie chart in Fig 2, illustrate the severity level percentage that was obtained and analyzed previously for BLOM Website:

- Ministry of Information Communications and Technology

Table 4 describes the results gain from CGI for Ministry of Information Communications and Technology website.

In Table 4, the first row shows that remote attackers can determine the physical path of the server by typing an invalid URL path, the name of such URL contain a standard DOS device name, [http://www.juniper.net/security/auto/vulnerabilities/vuln1608.html].

Table 4: CGI result for ministry of information communication and technology

Ministry of information communication and technology

Vulnerability description	Severity	No. of times
Frontpage98 Hole (_vti_inf.htm1)	Moderate	1

Total No. of vulnerability: 1

N-STALKER FREE 2009 RESULTS ANALYSIS

N-Stalker Web Application Security Scanner 2009 is a sophisticated web security assessment solution developed by N-Stalker. But the freeware edition that we were used has two main limitations: (1) crawling just 100 web pages for each website, (2) checking websites for XSS and web server security.

N-Stalker applied on the WUS. Table 5 summarizes the needed result that we were extracted from this tool.

Table 5: N-stalker results for all WUS

Website under of study (WUS)	Total No. of high severity risk	Total No. of moderate severity risk	Total No. of low severity risk	Scan time
Civil service bureau	0	4	0	17 h, 4 min
Ministry of education	0	5	0	19h, 27 min
Ministry of higher education	0	36	0	7 h
Ministry of information communications and technology	0	1	0	8 h, 50 min
Foreign ministry	0	0	0	94h, 53 min
Ministry of internal	0	12	0	23h, 16 min

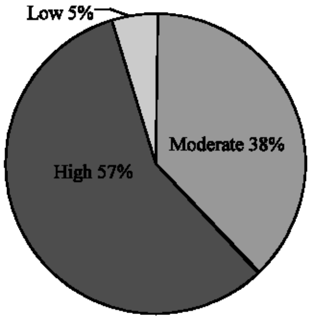

Fig. 2: The Severity level for BLOM-CGI

Table 5 illustrates the three main severity levels for our experiment and total number of each level for each websites. In addition, the scanning time was taken for testing WUS. For instance, we can read the first row as following: Civil Service Bureau neither has high sever risks nor low-level risks and it has 4 risks that considered moderate risks.

ANALYZING THE SECURITY LEVEL FOR EACH WEBSITE UNDER STUDY (WUS)

In order to achieve our goals and to determine which is the most secure website and the least secure one.

From Table 6, we can count the same severity level for the same website from all the tools result. The high severity denote with the red color, the moderate severity with blue and the low severity with green, respectively for each WUS.

The total number of the high severity level for all WUS from every tool was 540 and for moderate was 278. We will ignore the low severity level from our analysis and our concern is about high and moderate risk level. Table 7 provides the accumulative total for both high and

Table 6: Comparison between WUS's severity levels

WUS	Severity Level	Wikto	Acunetix	CGI	N-Stalker	Total
Civil service bureau	High	0	0	0	0	0
	Moderate	0	0	2	4	6
	Low	0	0	0	0	0
Ministry of education	High	0	0	0	0	0
	Moderate	0	0	0	5	5
	Low	0	0	0	0	0
Ministry of higher education	High	0	26	0	0	26
	Moderate	0	0	0	36	36
	Low	0	0	0	0	0
Ministry on information and communication technology	High	0	0	0	0	0
	Moderate	0	0	1	1	2
	Low	0	0	0	0	0
Foreign ministry	High	0	0	0	0	0
	Moderate	0	0	0	0	0
	Low	0	0	0	0	0
Ministry in internal	High	0	0	0	0	0
	Moderate	0	0	0	12	12
	Low	0	0	0	0	0

Table 7: The accumulative High and Moderate severity

Website Under Study (WUS)	Moderate severity percentage	High severity percentage
Civil service bureau	6	0
Ministry of education	5	0
Ministry of higher education	36	26
Ministry of Information communications and technology	2	0
Foreign ministry	0	0
Ministry of internal	12	0
Total	61	26

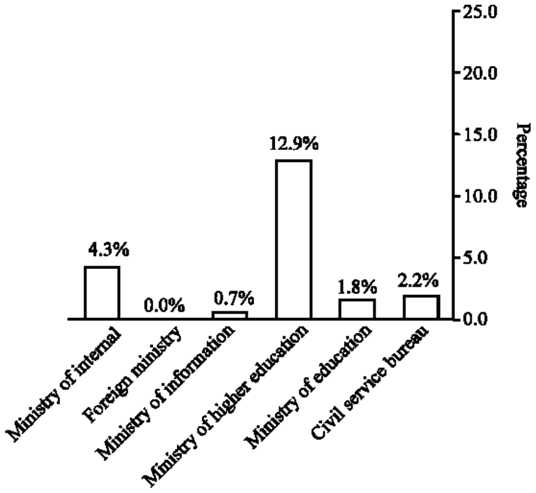

Fig. 3: Comparative between all WUS's moderate level

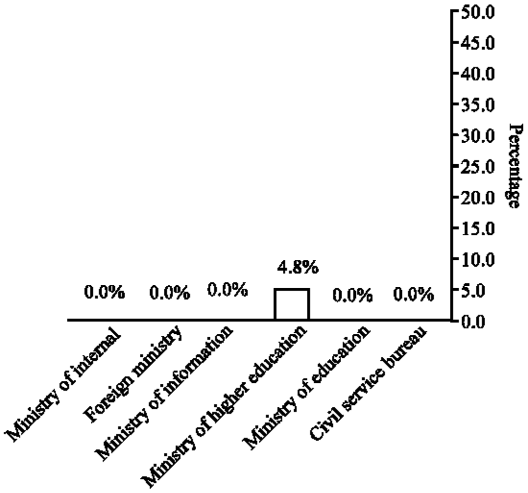

Fig. 4: Comparative between all WUS's High level

moderate level assigned for every WUS. From this table we calculate the percentage of each security level for each website (e.g., for civil service bureau website we found 6 moderate level risks obtained from CGI (2) and N-stalker(4), out of 278 as a total from all tools, the equation is (6/278)% = 2.2%) 55.

Table 7 shows accumulative severity level for each WUS, as shown in this table the high severity percentage shown in Fig. 3, Moderate severity percentage shown in Fig. 4 and comparative between WUS's High and Moderate Severity Levels in Fig. 5

From Fig. 3, we notice that the most secure website that not include any type of the moderate risk level is foreign ministry: (0%) percent.

From Fig. 4 we notice that the most secure websites that not include any type of the high risk level are: foreign ministry, civil service bureau, ministry of education, ministry of higher education, ministry of interior.

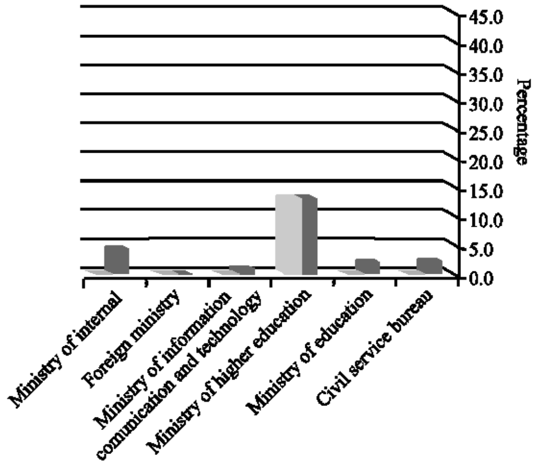

Fig. 5: Comparative between WUS's high and moderate severity levels

By a quick to look to Fig. 5, we can decide that the most vulnerable website due to both high and moderate levels is ministry of interior (MOI), then civil service bureau (CSB), after that ministry of education (MOE), then the ministry of information communication and technology (MOICT) and finally the most secure website with vulnerability free from both high and moderate level is Foreign Ministry (FM).

CONCLUSION

Website security is important and necessary. This is vital for e-business websites. Websites e-readyness depends largely on their security metrics.

Looking at the selected Jordanian websites, we are found out that further security assessments are required before evaluating the security stand of those websites. Websites owner need to pay attention to some high risky vulnerability that may endanger the reliability and integrity of their websites. Those websites should be frequently audited as risks are continuously evolving and progressing, this research was done 2009 in Jordan.

REFERENCES

Drum, R., 2006. IDS AND IPS placement for network protection. CISSP, White Paper. http://www.infosecwriters.com/text_resources/pdf/IDS_Placement_RDrum.pdf.

Eschelbeck, G. and M. Krieger, 2003. Eliminating Noise from Intrusion Detection Systems. Elsevier, New York, pp: 26.

Eusgeld, I., W. Kroger, G. Sansavini, M. Schlapfer and E. Zio, 2008. The role of network the oryand object-oriented modeling within a framework for the vulnerability analysis of critical infrastructures. Reliability Eng. Syst. Safety, 94: 956-956.

Halkidis, S.T. and E. Chatzigeorgiou, 2006. A Practical Evaluation of Security Patterns. Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece.

InfoSec Reading Room, 2001. Vulnerability Assessment. SANS Institute, Cary, NC., USA.

Jahangir, N. and N. Begum, 2008. The role of perceived usefulness, perceived ease of use, security and privacy and customer attitude to engender customer adaptation in the context of electronic banking. Afr. J. Business Manage., 2: 32-40.

Nguyen, H. and B. Johnson, 2006. Web Application Testing on the Web. 2nd Edn., John Wiley and Sons, New York.

Sun, Y.L., H. Zhu, W. Yu and K.J.R. Liu, 2006. A trust evaluation framework in distributed networks: vulnerability analysis and defense against attacks. Proceedings of 25th IEEE International Conference on Computer Communications, April 23-29, Barcelona, Spain, IEEE Press, pp: 1-1.

Thomas, S., L. Williams and T. Xie, 2009. On automated prepared statement generation to remove SQL injection vulnerabilities. Inform. Software Technol., 51: 589-598.

Xie, Y. and A. Aiken, 2006. Static detection of security vulnerabilities in scripting languages. Peoceeding of the 15th USENIX Security Symposium, Sept. 20, Stanford University, USENIX Association, pp: 179-192.