An Empirical Study on the Security Measurements of Websites of Jordanian Public Universities

Mokhled S. Al-Tarawneh *mokhledl@mutah.edu.jo*
Faculty of Engineering /Computer Engineering Department
Mutah University
Mutah, 61710 Jordan

Abstract

Most of the Jordanian universities' inquiries systems, i.e. educational, financial, administrative, and research systems are accessible through their campus networks. As such, they are vulnerable to security breaches that may compromise confidential information and expose the universities to losses and other risks. At Jordanian universities, security is critical to the physical network, computer operating systems, and application programs and each area has its own set of security issues and risks. This paper presents a comparative study on the security systems at the Jordanian universities from the viewpoint of prevention and intrusion detection. Robustness testing techniques are used to assess the security and robustness of the universities' online services. In this paper, the analysis concentrates on the distribution of vulnerability categories and identifies the mistakes that lead to a severe type of vulnerability. The distribution of vulnerabilities can be used to avoid security flaws and mistakes.

Keywords: Information System, Information Security, Information-security Measurement, Security Threats, Vulnerability Measurement and Penetration Test.

1. INTRODUCTION

Now a day, the Internet is used to provide a growing number of services for Jordanian universities like online registration, online libraries systems, online fees payment, and online portals, all these services are transferring confidential information over public networks, and the internet. This development has heightened user sensitivity to security violations. Most users use browsers to access these online services which making web based application a critical part of the internet communication infrastructure. A web based application developed their specific services over hyper text transfer protocol to the data base accesses, such implementation can introduce security holes, if they aren't implemented with care. Such typical security holes of web applications are well-known to hackers; there are even tools on the internet, which automatically look for and use certain security holes. Security for web applications became therefore an urgent problem (Peine & Mandel, 2006). Once the online services using web applications platform so the infrastructure platform must be investigated for discovering and recognizing the malignant patterns for intrusion detection. There is a need for evaluating the security of application systems by attack simulation of indoor and outdoor malicious. A fake attacking process involves an active analysis of the application systems for important and critical vulnerabilities that could results from poor system design and configuration or operational weaknesses. To get vulnerabilities classification, This analysis is carried out from the attack vector and its behavior (Hansman & Hunt, 2004). Web site and whole information domains analysis need a good information security metrics as an important factor in making sound decisions about various aspects of security, ranging from the design of security architectures and controls to the effectiveness and efficiency of security operations. Security metrics strive to offer a quantitative and objective basis for security assurance (Berinato, 2005). The main uses fall into several broad classes such as strategic support, security quality assurance and tactical oversight from point of view of monitoring and reporting of security status. The security requirements of web based applications include confidentiality, authentication, availability, authorization, integrity and non-repudiation, so

how to efficiently evaluate the security of web application services is a challenging research topic. There can be no information security without information security triad or given six basic security concepts to be covered by security testing (Feruza & Kim, 2007). The educational web service security requirements must satisfy addressing in the context of comprehensive application system architecture. This should include institutional portals, central directory service and user trust relationships (Gleason, 2002). According to the characteristics of educational web service application and the classification threats, security decisions must always be made with an understanding of these threats. The top threats facing web services can be classified into the following categories (Singhal, Winograd, & Scarfone, 2007):

1. Message alteration, where the attacker can insert, remove or modify information within a message to deceive the receiver.
2. Loss of confidentiality; where the information within a message is disclosed to an unauthorized individual.
3. Falsified messages, where fictitious messages that an attacker intends the receiver to believe are sent from a valid sender.
4. Principal spoofing: where the attacker constructs and sends a message with credentials such that it appears to be from a different, authorized principal.
5. Forged claims, where the attacker constructs a message with false credentials that appear valid to the receiver.
6. Replay of message, where the attacker resends a previously sent message.
7. Replay of message parts, where the attacker includes portions of one or more previously sent messages in a new message.
8. Denial of service, where the attacker causes the system to expend resources disproportionately such that valid requests cannot be met.
9.

Research into security policies, security infrastructure, threats and vulnerabilities of computer information systems continues to grow because it's evolving nature and significant economic impact on organizations. Information security policy means protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, perusal, inspection, recording or destruction (Johnson & Merkow, 2010). As information is an extremely valuable and important corporate asset that requires protection against risks that would threaten its confidentiality, integrity and/or availability (Feruza & Kim, 2007). Suitable information security controls must therefore be selected and implemented. Therefore, the logic of well planned policy is depending on the needs of organizational policy definition that guides personal and technology decision which serve the organizational needs. By definition, security policy refers to clear, comprehensive, and well-defined plans, rules, and practices that regulate access to an organization's information systems. Information system security relates to the adequacy of management controls to prevent, avoid, detect and recover from whole range of threats that could cause damage or disruption to computer systems. Good security policy protects not only information systems, but also individual employees and the organization as a whole. It also serves as a prominent statement to the outside world about the organization's commitment to security. Security measurements have proven to be much more successful when the target of assessment is audited and penetrated carefully to avoid security holes. Vulnerability analysis and penetration test attempt to exploit any one of the vulnerabilities to gain unauthorized access (Arkin, Stender, & McGraw, 2005). The valuable educational information such as student identification number, student mark records, as well as academic and employee identification numbers, and their confidential financial records are subject of attack. Attacker are well aware of pervious valuable information accessible through web applications and their attempts to get at it are judged by several important factors (Stolfo, Bellovin, & Evans, 2011). Gartner reported that 75% of hacks happen at the application layer, not at the network or system layer, (Arkin, Stender, & McGraw, 2005; , "www.gartner.com"). To check application layer there is a need for information gathering where the tester used an automated scanning, web crawlers, and social engineering, to gain information about the target application. Gathered information is used to generate possible attack in which testers use the identified information, together with domain knowledge about possible vulnerabilities to generate attacks (Dahl, 2005). Penetration test and

vulnerabilities analysis based on fake attacks used to find logs information about the attack and to generate a report that details the discovered vulnerabilities and corresponding remedy, a result information can be used by application developers to eliminate the vulnerabilities and improve the security of their produced application software (Petukhov & Kozlov, 2008). The searching and eliminating vulnerabilities in the web application is one of the common ways of securing mechanism, the other ways are included in safe development (Cook & Rai, 2005; Meier, Mackman, & Wastell, 2005), implementing intrusion detection and/or protection systems (Halfond & Orso, 2005; Su & Wassermann, 2006), and web application firewalls (Ristic, 2006). Penetration test provides detailed information on actual, exploitable security threats if it is encompassed into an organization's security doctrine and processes. This will help the organization to identify quickly and accurately real and potential vulnerabilities for the purpose of isolate and prioritize vulnerabilities, and assist the organization fine-tune, test configuration changes or patches to proactively eliminate identified risks (Bacudio, Yuan, Chu, & Jones, 2011). In this paper I will tackle the issue of Information systems safety in Jordanian educational organizations, considering public universities as a case study and propose a circulated model for enhancing the security of information system in educational organizations. Through the special case study on Mutah University, the experimental results provide valuable reference to check out security vulnerabilities of Web service and help to optimize the system's security design. In this paper, Section 2 presents an elaborate discussion of the experimental platforms as well analysis of their results; while Section 3 presents the remedies for detected vulnerabilities; and Section 4 concludes the paper and points out the next work.

2. CURRENT SITUATION ANALYSIS

In this paper, penetration test and web application scanners were used to discover the importance of each security parameter and to evaluate the whole web site response. This test is designed to evaluate an information system's defense and discover weaknesses in the whole domain website design and its resources from ethical hacking point of view and how the site reacts to an attack, whether or not a whole system's defenses can be breached, and what information can be acquired from the whole system. A comprehensive security health check performed using Netsparker Professional Edition version 2.3.0.9 ("www.mavitunasecurity.com/netsparker") to evaluate the current security status of Jordanian public universities from point of view of vulnerability types and severity groups. Test was done on the universities as given Table1; it covered all security scanning tests on the whole domain scope with 25 concurrent connections to the target back end database following the crawling and attacking principles. All universities sites were checked using Netsparker as whole domains; much iteration of crawling tests was performed to find the vulnerability holes in given sites. Results is used to take the next steps to reinforce these universities to defense their sites from injections such as SQL, Boolean SQL, Blind SQL, Command, Blind command, Expression language and HTTP header, as well Open Redirection, Local, Remote file inclusion and Remote code evaluation Table2.

University	DNS Name	IP address
Jordan	ju.edu.jo	87.236.232.79
Yarmouk	yu.edu.jo	87.236.233.10
Mutah	mutah.edu.jo	87.236.232.229
Jordan Scince	just.edu.jo	87.236.232.175
Al al-Bayt	aabu.ed.jo	87.236.233.39
Hashemite	hu.edu.jo	87.236.232.216
Balqa Applied	bau.edu.jo	87.233.233.117
Al Hussein	ahu.edu.jo	87.236.233.71
Tafila Technical	ttu.edu.jo	87.236.233.188
Germany Jordan	gju.edu.jo	87.236.233.228

TABLE 1: Jordanian Universities Domains and IP Addresses.

Severity Type	Vulnerability Hole	University									
		Jordan	Yarmouk	Mutah	Jordan Science	Al al-Bayt	Hashemite	Balqa	Al Hussein	Tafila	Germany Jordan
Critical	SQL Injection						x	x			
	Boolean Based SQL Injection	x									
Important	Database User Has Admin Privileges	x									
	Local File Inclusion	x						x			
	Cross-Site Scripting			x	x	x					x
	Pasword Transmitted Over HTTP			x			x	x	x		x
Medium	MAC is not Enabled in ViewState	x			x				x		
	Password Transmitted Over Query String			x							
	Invalid SSL Certificate Detected				x						
	ColdFusion Source Code Disclosure			x							
	Open Redirection				x						
Low	Internal Server Error	x	x	x	x	x	x	x	x	x	x
	Cookie Not Marked As Http Only	x	x	x	x					x	
	ASP.NET version Disclosure	x	x		x			x	x		x
	Social Security Number Disclosure	x		x		x			x		
	ASP.NET Stack Trace Disclosure					x	x				x
	ViewState is not Encrypted	x			x			x	x		x
	MS Office Information Disclosure	x		x		x	x			x	
	Internal IP Address Leakage	x			x	x		x	x		
	Redirected Response BODY Has Two responses	x									
	PHP Version Disclosure		x	x						x	
	Auto Complete Enabled			x	x		x	x	x		x

	Apache Version Disclosure			x						x	
	Backup File Found			x				x			
	Tomcat Version Disclosure					x					
	Tomcat Exception Report Disclosure					x					
	Database Error Message							x			
Informatio	NTLM Authorization Required	x			x						
	Microsoft SQL server Identified	x									
	Forbidden Resource	x	x	x	x	x	x	x	x	x	x
	ASP.NET Identified	x			x						
	E-mail Address Disclosure	x	x	x	x	x	x	x	x	x	x
	IIS Version Disclosure	x	x		x		x	x	x		x
	Internal Path Leakage (Windows)	x		x	x		x	x	x	x	x
	Apache Version Is Out Of Date			x						x	
	PHP Version Is Out Of Date			x						x	
	Tomcat Version Is Out Of Date					x					
	Redirected Response BODY IS Too Large					x					x
	Sitemap Identified					x				x	
	Internal Path Leakage (*nix)					x					
	Directory Listing (IIS)						x				
	File Upload Functionality Identified	x									

Table 2: Jordanian Public Universities Vulnerabilities and Severities Evaluation.

Test was done for each site to get the severity of prioritize vulnerabilities as critical, important, medium, low and informational vulnerability types as defined by payment card industry (PCI) standard, where the critical level is given to flaws that could be easily exploited by a remote unauthenticated attacker and lead to system compromise without requiring user interaction. Important rating is given to flaws that can easily compromise the confidentiality, integrity, or availability of resources. Medium rating is given to flaws that may be more difficult to exploit but could still lead to some compromise of the confidentiality, integrity, or availability of resources, under certain circumstances. Low rating is given to all other issues that have a security impact. Figures 1, 2, 3, 4, 5 shows all individual prioritize vulnerabilities parameters. Figure 1 shows that Hashemite University web application site is the most critical vulnerable problems; Figures 2, 5 shows that Al al-Bayt University is the highest in important and information severity problems, while Figures 3, 4 indicate that Mutah University whole domains website is the most in medium and low severity problems.

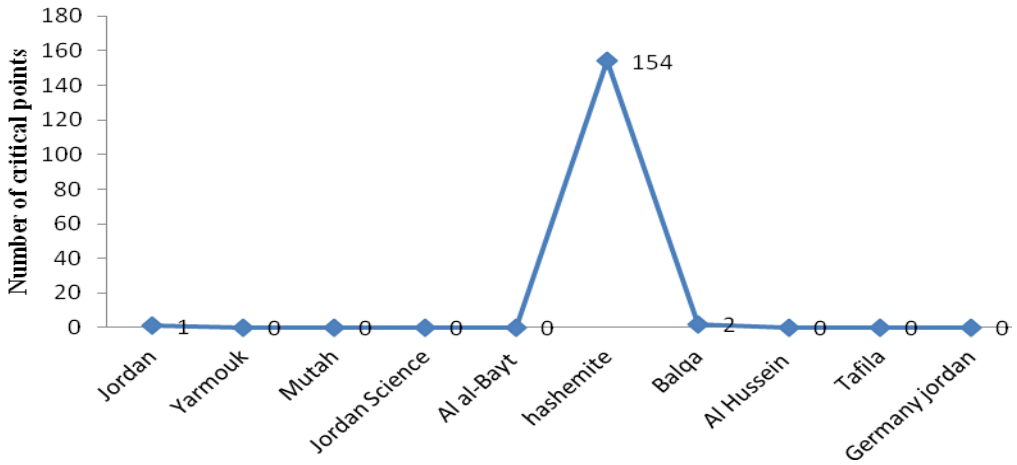

FIGURE 1: Critical Severity Problems Among University Sites.

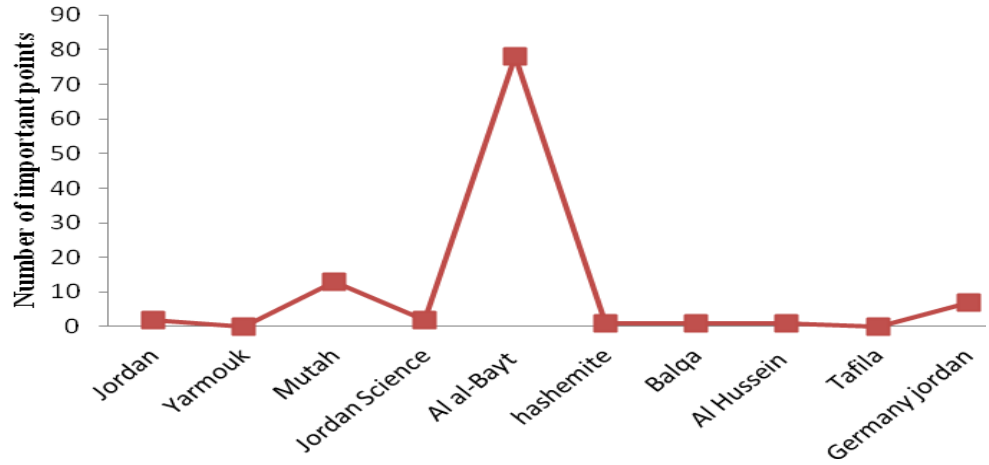

FIGURE 2: Important Severity Problem Among University Sites.

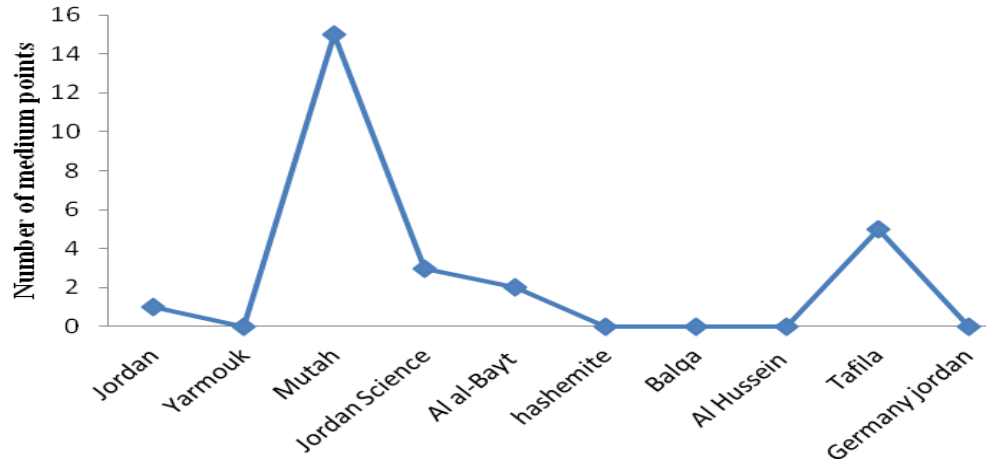

FIGURE 3: Medium Severity Problem Among University Sites.

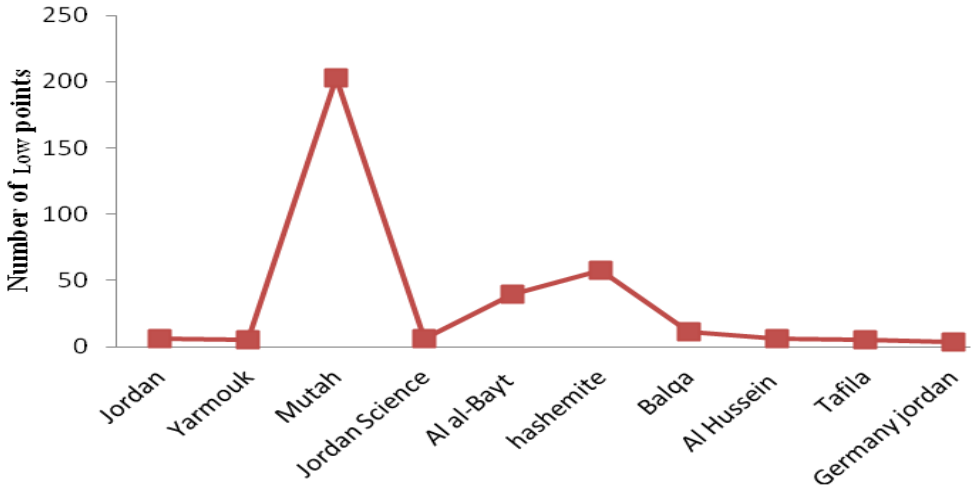

FIGURE 4: Low Severity Problem Among University Sites.

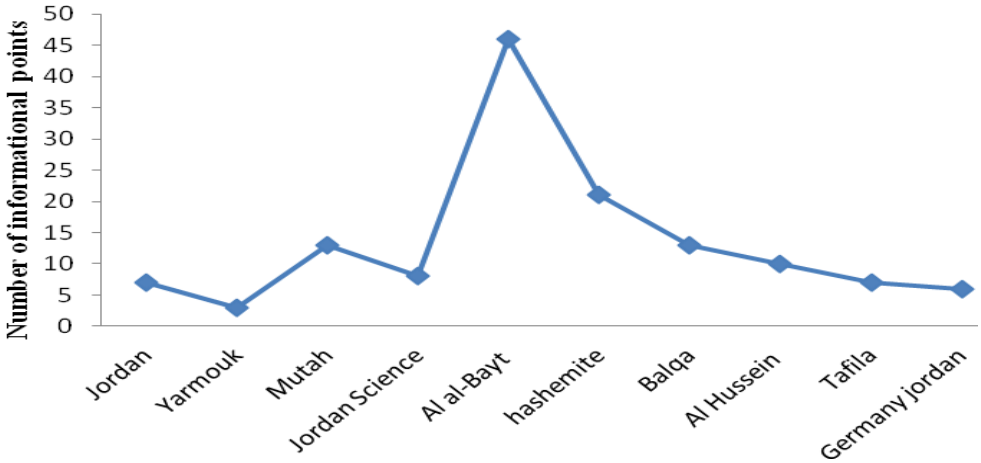

FIGURE 5: Information Severity Problem Among University Sites.

A conclusion of previous figures shown indicates that the Hashemite University with the highest critical vulnerability holes, Balqa applied and Jordan universities were investigated with minimum range of holes from 1 to 2, while the other universities were free from critical holes. Important investigated holes were found in Al al-Bayt, Mutah, Germany-Jordan and Jordan University, respectively. Medium holes were found to be high in Mutah whole domain site as well as low vulnerability holes. Table 2 gives all details of severity vulnerability types, here a concentration will be on the major holes and their remedies. Critical risk severities were found in.

3. DETECTED VULNERABILITIES, ANALYSIS AND REMEDIATION

According to the given results in Table 2, with the investigation of critical and important vulnerabilities there found the following attacks:

SQL and Boolean Based injection: It allows an attacker to alter backend SQL statements by manipulating the user input. An SQL injection occurs when web applications accept user input that is directly placed into a SQL statement and doesn't properly filter out dangerous characters. This is an extremely common vulnerability and its successful exploitation can have critical implications.

SQL injection impact: It depends on the backend database, database connection settings and the operating system; an attacker can compromise the integrity of database and expose sensitive information by mount one or more of the following type of attacks successfully:

- Reading, Updating and Deleting arbitrary data and tables from the database.
- Executing commands on the underlying operating system.
-

SQL injection remedy: A very robust method for mitigating the SQL injection threat is to use parameterized queries. Wherever possible do not create dynamic SQL queries or SQL queries with string concatenation.

Database User Has Admin Privileges: It is related to SQL Injection vulnerability in the application which identified that the target web site is connecting to the backend database by using a user that has administrative privileges.

Database User Has Admin Privileges impact: An attacker can gain extra privileges via SQL Injection attacks. The attacker might carry out the following type of attacks:

- Gain full access permission to the database server.
- Gain a reverse shell to the database server and execute commands on the underlying operating system to gain administrator access to the target system.

Database User Has Admin Privileges remedy: A database user must be created with the least possible permissions for hosted application and connect to the database with that user.

Local File Inclusion: It is occurred when a file from the target system is injected into the attacked server page.

Local File Inclusion impact: Depends on exploitation and the read permission of the web server user, an attacker might carry out one or more of the following attacks:

- Gather usernames and harvest useful information from the log files
- Remotely execute commands via combining this vulnerability with some of other attack vectors such as file upload vulnerability or log injection.
-

Local File Inclusion remedy: A file path must not be permitted to be appended directly, it must be hard coded via and index variable. An application programming interface must be limited to allow inclusion only from a directly directories bellow it to avoid traversal attack.

Cross-site Scripting: A dynamic script in the context of the application is allowed and executed by attacker. This allows several different attack opportunities, mostly hijacking the current session of the user or changing the look of the page by changing the HTML on the fly to steal the user's credentials. It targets the users of the application and administrator instead of the server to gain full control over the application and administration panel.

Cross-site Scripting impact: Using cross site scripting, the following attacks can be leveraged:

- Hi-jacking users' active session and intercept data.
- Mounting phishing attacks.
-

Cross-site Scripting remedy: To avoid this kind of attack, output should be encoded according to the output location and context.

Password Transmitted Over HTTP: The criticality of this vulnerability that password data is sent over HTTP.

Password Transmitted Over HTTP impact: An attacker can intercept network traffic to steal user’s credentials.

Password Transmitted Over HTTP remedy: All sensitive data should be transferred and served over HTTPS and rather than HTTP.

The other detected vulnerabilities categorized into medium, low and informational levels as shown in table2, an attacker might carry out one or more of the following attacks: application tampering, performing man in the middle attack and encryption traffic between website and the visitors observation, obtaining server side source code of the web application, which can contain sensitive data such as database connection strings, usernames and passwords along with the technical and business logic of the application. Redirect Users to other malicious web sites which can be used for phishing and similar attacks. Internal server error or server responded with an HTTP status 500. Providing an additional layer of protection against Cross-site Scripting attacks because of cookie was not marked as HTTPOnly. Auto complete was enabled in one or more of the form fields.

Previous attacks remedies: Secure socket layer certificate activation to provide secure communication between your website and its visitors. Ensure that the server has all the current security patches applied and remove all temporary and backup files from the web server. Ensure that you only accept URLs which are located on accepted domains. Analyze and review the application code in order to handle unexpected errors. Mark the cookie as HTTPOnly. Add the attribute auto complete="off" to the form tag or to individual "input" fields. Avoid disclosed information in website platforms.

4. CONCLUSION AND FUTURE WORK

This paper has described a full detailed analysis of real and potential vulnerabilities in public Jordanian universities. It's providing the information required to effectively and efficiently isolate and prioritize vulnerabilities for the purpose of assisting the universities to fine-tune and test configuration changes or patches to proactively eliminate identified risks. Based on given analysis a unified security policy for educational institutes will be proposed in the future work.

5. REFERENCES

[1] Arkin, B., Stender, S., & McGraw, G. (2005). Software penetration testing. Security & Privacy, IEEE, 3, 84-87.

[2] Bacudio, A., Yuan, X., Chu, B., & Jones, M. (2011). An Overview of Penetration Testing. Journal of Network Security & Its Applications (IJNSA), 3, 19-38.

[3] Berinato, S. (2005). A Few Good Information Security Metrics. CSO Magazine.

[4] Cook, W., & Rai, S. (2005). Safe Query Objects: Statically Typed Objects as Remotely Executable. Paper presented at the 27th International Conference on Software Engineering.

[5] Dahl, O. M. (2005). Using Coloured Petri Nets in Penetration Testing. Unpublished Master's thesis, Gjøvik University College, Norway.

[6] Feruza, S., & Kim, T.-h. (2007). IT Security Review: Privacy, Protection, Access Control, Assurance and System Security. International Journal of Multimedia and Ubiquitous Engineering, 2, 17-31.

[7] Gleason, B. (2002). Web Services in Higher Education - Hype, Reality, Opportunities. Educause Quarterly, 25, 11-13.

[8] Halfond, W., & Orso, A. (2005). AMNESIA: Analysis and Monitoring for Neutralizing SQL-Injection Attacks. Paper presented at the International Conference on Automated Software Engineering.

[9] Hansman, S., & Hunt, R. (2004). A taxonomy of network and computer attacks. Elsevier, Computers & Security, 24, 31-43.

[10] Johnson, R., & Merkow, M. (2010). Security Policies and Implementation Issues: Jones and Bartlett Learning.

[11] Meier, J., Mackman, A., & Wastell, B. (2005). Threat Modeling Web Applications: Microsoft Patterns & practices, Microsoft Corporation.

[12] Peine, H., & Mandel, S. (2006). Security Test Tools for Web Applications (No. IESE Report-Nr. 048.06/D).

[13] Petukhov, A., & Kozlov, D. (2008). Detecting Security Vulnerabilities in Web Applications Using Dynamic Analysis with Penetration Testing. Paper presented at the Application Security Conference.

[14] Ristic, I. (2006). Web application firewalls primer. SECURE, 1, 6-10.

[15] Singhal, A., Winograd, T., & Scarfone, K. (2007). Guide to Secure Web Services (No. MD 20899-8930). Gaithersburg: National Institute of Standards and Technology.

[16] Stolfo, S., Bellovin, S., & Evans, D. (2011). Measuring Security. Security & Privacy, IEEE, 9, 60 - 65.

[17] Su, Z., & Wassermann, G. (2006). The essence of command injection attacks in web applications. In ACM SIGPLAN Notices (Vol. 41, pp. 372-382).

[18] www.gartner.com. Retrieved 26-12, 2012

[19] www.mavitunasecurity.com/netsparker.