Journal of Theoretical and Applied Information Technology
20th March 2014. Vol. 61 No.2
© 2005 - 2014 JATIT & LLS. All rights reserved·

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ASSESSMENT OF OPEN SOURCE WEB APPLICATION SECURITY SCANNERS

[1] FAKHRELDEEN ABBAS SAEED, [2]ELTYEB E. ABED ELGABAR

[1,2]Asstt Prof., Department of Information Technology, College of Computer Science and Information Technology, KAU, Khulais, Saudi Arabia

E-mail: [1]fakhry00@gmail.com, [2]tayesamani@hotmail.com

ABSTRACT

The web application security has currently become a very significant area of scholarship, the best way to deal with it is to use web application security scanner to discover the architectural weaknesses and vulnerabilities in the web application. A standard has been constructed by OWASP which lists common risks. The goal of this paper is to use OWASP Top 10 to compare and contrast the Open Source Web Application Security Scanners, and then determine the best of them. The study shows that W3AF 1.2, arachniv0.4.0.3 and Skipfish 2.07 are the most suitable ones because they have 0.863826, 0.79922, and 0.781676 averages respectively. So the web developer or administrator can use them together, choose one, or modify it by adding the missing feature and make his/her own application.

Keywords: *Open Source, Web Application Security Scanner, OWASP, Evaluation*

1. INTRODUCTION

Web applications are complex entities that have a lot of flaws. [1] Web application security scanners are automated tools that check out web applications for security vulnerabilities, without access to the application's source code. [2] Our goal in this paper is to show the differences between Open Source Web Application Security Scanners and show the strengths and limitations of them; to guide a developer of web application how to choose his/her scanner. In this paper, we explain how to assess Open Source Web Application Security Scanner depending on the OWASP Top 10-2013 application security risks. [3]

This paper is structured as follows: Section 2 provides a brief introduction about the web application security, web application security tools and scanner. Section 3 describes our approach for evaluate the Open Source Web Application Security Scanner. Section 4 presents the evaluation results with discusses. Section 5 conclusion of the paper.

2. WEB APPLICATION SECURITY TOOLS

"Information security means protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction". [4] The branch of information security that deals with all aspect of web security like application, services and sites is called web application security, and also the web application security is application security principles applied to internet and web systems .When Web 2.0 is introduced, the information shared start grow fast through social networking and change the way of doing business and delivering service, this is lead the hackers to attach the websites, so the industry increased attention to web application security[5].

There are a number of technical solutions to consider when designing, building and testing secure web applications [5]. These solutions include:

- White Box testing tools such as static source code analyzers [6], [7], [8], [9], [10], [11] allowing the recognition of vulnerabilities before web application deployment.
- Detection and possible sanitization at runtime of malicious requests before they reach the server. The corresponding tools can run on the server [12], [13], or between the client and the server acting as a proxy [14].
- Black Box testing tools such as web application security scanners, vulnerability scanners and penetration testing software [15]. These tools consist in crawling the target application to identify reachable pages and possible input vectors, and generate specially

crafted inputs to determine the presence of vulnerabilities.

A large number of vulnerability scanners have been developed, including commercial tools and open source tools. In this paper, we focus on Open Source Web Application Security Scanners.

2.1. Web Application Security Scanners

A web application security scanner communicates with a web application to identify potential security vulnerabilities in the web application and architectural weaknesses. It is one of Black Box testing tools; perform scanning without having to access to the source code and therefore detect vulnerabilities by actually performing attacks.

Although some researchers have shown the limitations of Web Application Security Scanners in detecting some vulnerabilities [2, 16, 17, 18], Scanners became widely adopted due to the usability, automation, and independence from the web application technology used.

3. THE CRITERIA OF ASSESSMENT

Generally, the steps of evaluation of a system are selecting the evaluation criteria, suitable environment, and correct tools. In this study we used the following Steps to compare and assess the Open Source Web application Security Scanners [19]:

1. Putting the assessment and comparison criteria.
2. Listing available platforms.
3. Demonstrate the result of assessment depend on the criteria.

3.1. Security Standards

The Open Web Application Security Project (OWASP) is an open community consecrated to provide organizations by information that help to develop, purchase, and maintain secure applications. It produces free and open Application security standards and tools. We have used the OWASP Top 10-2013 as assessment criteria; the following list mentions main categories of web application security risks. [3]

1. **Injection:** Injection flaws occur when untrusted data is sent to an interpreter as part of a command or query by an attacker to trick the interpreter into executing unintended commands or accessing data without proper authorization.
2. **Broken Authentication and Session Management:** the developer of web site often implement functions related to authentication and session management incorrectly ,that allowing attackers to exploit other implementation flaws to assume other users' identities ,or to compromise passwords, keys, or session tokens.
3. **Cross-Site Scripting (XSS):** the attackers can execute scripts in the victim's browser to hijack his sessions, deface web sites, or redirect the user to malicious sites by XSS.
4. **Insecure Direct Object References:** the attacker can access unauthorized data, when a reference to an internal implementation object is exposed without an access control check or other protection.
5. **Security Misconfiguration:** When the configuration of the application, frameworks, application server, web server, database server, and platform is unsecure or poorly configured security controls may be allow hackers to attach the system [3].
6. **Sensitive Data Exposure:** attackers can steal or modify improperly protect sensitive data, such as credit cards, tax IDs, and authentication credentials.
7. **Missing Function Level Access Control:** Function level access must be verified before making its functionality visible in the UI. However, applications need to perform the same access control checks on the server when each function is accessed. If requests are not verified, attackers will be able to forge requests in order to access functionality without proper authorization.
8. **Cross-Site Request Forgery (CSRF)**: A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to force the victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim.
9. **Using Known Vulnerable Components**: Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications using

components with known vulnerabilities may undermine application defenses and enable a range of possible attacks and impacts.

10. **Unvalidated Redirects and Forwards**: Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.

The next table illustrates the assessment criteria and sub feature of it. We aren't going to use "Using Known Vulnerable Components" in our evaluation.

Table 1: Criteria and Sub Feature of Assessment

No	The criteria	Feature
1	Injection	SQLi, BSQLi, SSJSi, CMDExec, CRLFi, LDAPi, XPAPHi, MXi, SSI, CODEi, XMLi, Eli, BUFFERo, INTEGERo
2	Broken Authentication and Session Management	AUTHb, PRIVe, SESSION, FIXATION, Custom Cookie, Custom Header, BASIC, DIGEST, NTLM, NTLMv2, KERBEROS, FORM, CERT, Logout Detection, Exclude Logout, Anti CSRF Support, CAPTCHA Bypass, COOKIE, Dir & File Enumeration
3	Cross-Site Scripting	RXSS, PXSS, DXSS
4	Insecure Direct Object References:	LFI
5	Security Misconfiguration	WebServer Hardening
6	Sensitive Data Exposure	PADDING, SSL
7	Missing Function Level Access Control	BACKUPf, Exclude URL, Exclude Param
8	Cross-Site Request Forgery	CSRF
9	Using Known Vulnerable Components	
10	Unvalidated Redirects and Forwards	REDIRECT

3.2. Open Source Web Application Security Scanner

Table 2 shows a list of Open Source Web Application Security Scanners with some information such as their version, license, technology and last update. [20]

Table 2: List of Open Source Web Application Security Scanners

	Open source web scanner	Version	License /Technology	Last Update
1	IronWASP v0.9.1.0	0.9.1.0 (GA)	GPL3/.Net 2.0	2/7/2012
2	Zed Attack Proxy (ZAP) v1.4.0.1	1.4.0.1 (GA)	ASF2/Java 1.6.x	9/4/2012
3	***sqlmap*** v1.0-Jul-5-2012 (Github)	1.0 (GA)	GPL2/ Python 2.6.x	5/7/2012
4	***W3AF*** 1.2-rev509 (SVN)	1.2 (Beta)	GPL2/ Python 2.6.x	1/7/2012
5	arachniv0.4.0.3	0.4.0.3 (GA)	GPL2/Ruby 1.9.x	12/3/2012
6	Skipfish 2.07b	2.07 (Beta)	ASF2/C	24/5/2012
7	***Watobo*** v0.9.8-rev724	0.9.8 (Beta)	GPL2/Ruby 1.8.x	18-04-201
8	***VEGA*** 1.0 beta (Subgraph)	1.0 (Beta)	EPL1/Java 1.6.x	29/6/2011
9	Andiparos v1.0.6	1.0.6 (GA)	GPL2/Java 1.5.x	19/10/2010
10	ProxyStrikev2.2	2.2 (GA)	GPL2/Python 2.6.x	25/4/2009
11	Wapiti v2.2.1	2.2.1 (GA)	GPL2/Python 2.6.x	19/12/2009
12	Paros Proxy v3.2.13	3.2.13 (Final)	Clarified Artistic License/Java 1.4.x	8/8/2006
13	Grendel Scan v1.0	1.0 (Final)	GPL3/Java 1.5.x	26/8/2008
14	PowerFuzzer v1.0	1.0 (Beta)	GPL/Python 2.5.x	1/1/2009
15	Oedipus v1.8.1	1.8.1 (Beta)	GPL2/Ruby 1.8.x	8/4/2006
16	UWSS (Uber Web Security Scanner) v0.0.2	0.0.2 (Alpha)	GPL3/Python 2.6.x	22/7/2009
17	Grabber v0.1	0.1 (Beta)	BSD/Python 2.4.x	1/1/2008
18	WebScarabv20100820	20110329 (GA)	GPL/Java 1.5.x	29/3/2011
19	Mini MySqlat0r v0.5	0.5 (GA)	GPL/Java 1.6.x	6/11/2009
20	***WSTool*** v0.14001	0.14001 (Alpha)	GPL /PHP	1/2/2007
21	crawlfish v0.92	0.92 (Beta) /Build 2	GPL2/.Net 1.1	28/8/2007
22	Gamja v1.6	1.6 (Beta)	GPL/Perl 5.x	14/11/2006
23	iScan v0.1	0.1 (Beta)	GPL2/Java 1.6.x	17/12/2009
24	DSSS (Damn Simple SQLi Scanner) v0.1h	0.1h (Beta)	GPL2/Python 2.6.x	28/7/2011
25	Secubat v0.5	0.5 (Alpha)	LGPL/ .Net 2.0	27/1/2010
26	SQID (SQL Injection Digger) v0.3	0.3 (Pre-Alpha)	GPL2/Ruby 1.8.x	14/1/2008
27	SQLiX v1.0	1.0 (End-of-Life)	FOSS/Perl 5.x	1/7/2006
28	Xcobra v0.2	0.2 (Beta)	GPL3/Python 2.6.x	16/9/2010
29	XSSploit v0.5	0.5 (GA)	GPL2/Python 2.5.x	14/5/2009
30	***XSSS*** v0.40	0.40 (Beta)	GPL2/Perl 5.x	28/7/2005
31	***XSSer*** v1.5-1	1.5 (Beta)	GPL3/Python 2.5.x	24/2/2011
32	***aidSQL*** 02062011	02062011 (Beta)	GPL2/PHP	02-02-201

4. ASSESSMENT THE OPEN SOURCE WEB APPLICATION SECURITY SCANNERS:

We focus on the first three application security risks (Injection, Broken Authentication and Session Management, and) and remove the platform don't support these risks, Figure 2 shows the comparison of all platform with the first risk (Injection), we remove crawlfish v0.92, XSSploit v0.5, XSSS v0.40 and XSSer v1.5-1 for unsupported injection risk.

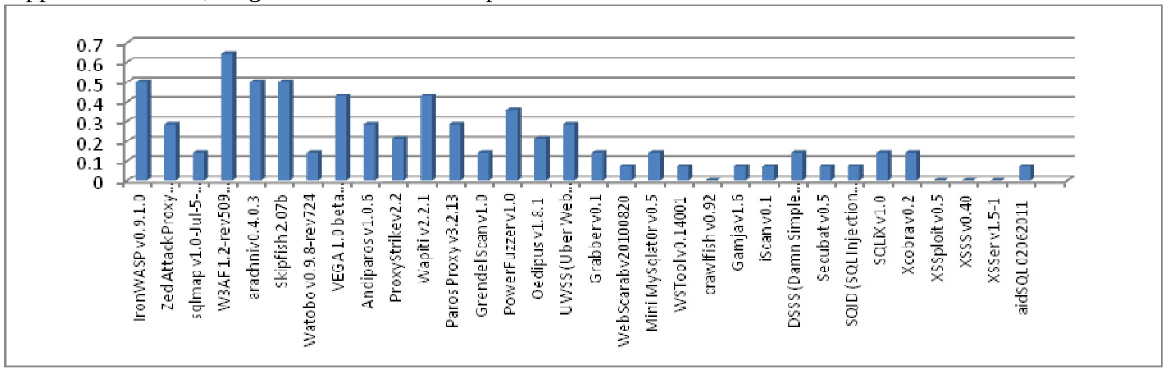

Figure 1: Injection Comparison

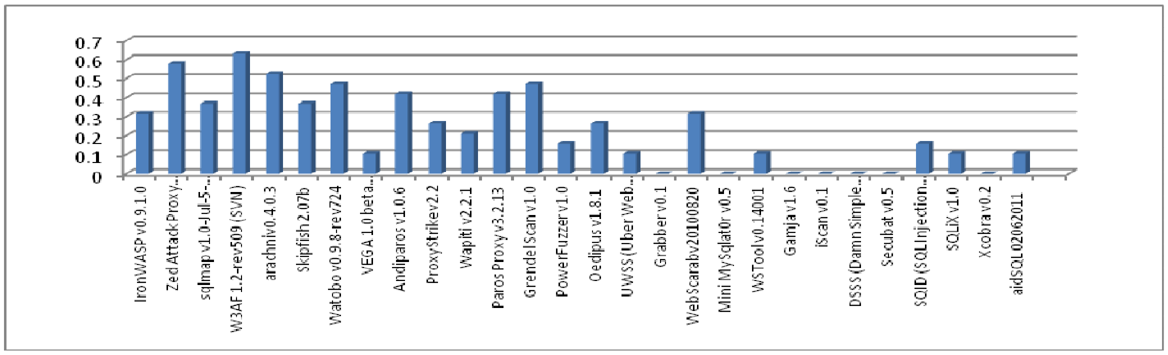

Figure 2: Broken Authentication and Session Management Comparison

Like we did in Figure 1 above, we are going to remove all the platform that unsupported second risk (Broken Authentication and Session Management) shows in Figure 2, the platform are Grabber v0.1, Mini MySqlat0r v0.5, Gamja v1.6, iScan v0.1, DSSS v0.1h, Secubat v0.5, Xcobra v0.2.Also removed sqlmap v1.0, SQID v0.3, SQLiX v1.0 and aidSQL 02062011 from Figure 3 because unsupported third risk (XSS), finally we have only 17 Platform.

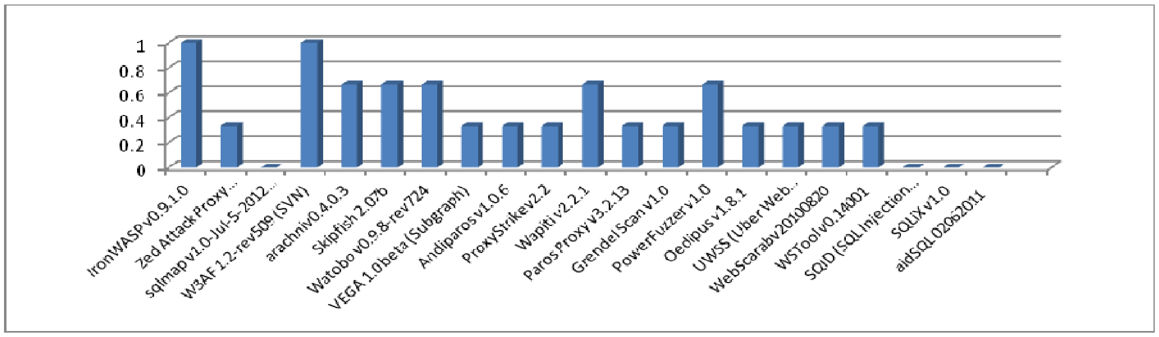

Figure 3: XSS comparison

Figure 4 below illustrates comparing the 17 Platform with all application security risks.

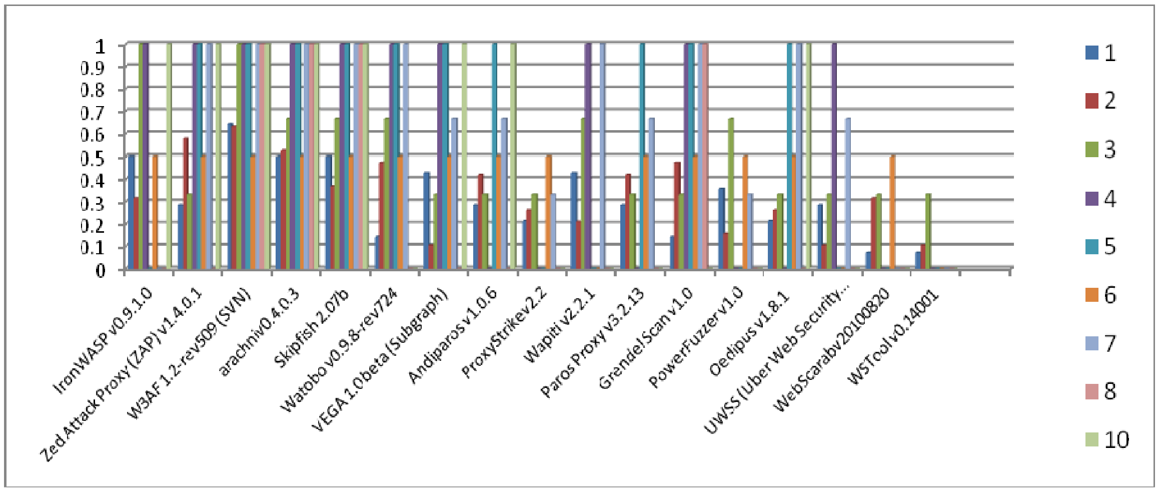

Figure 4: Comparison of The Rest Platform With All Risk

The last column in Table 3 shows the number of application security risks supported by the platforms. We choose the platform that supports eight or nine application security risks and put them in Table 4 to determine which is better to use as a web application security scanner.

Table 3: Average Of The Application Security Risks

	1	2	3	4	5	6	7	8	10	Risk_sup
IronWASP v0.9.1.0	0.5	0.315789	1	1	0	0.5	0	0	1	6
Zed Attack Proxy v1.4.0.1	0.285714	0.578947	0.333333	1	1	0.5	1	0	1	8
W3AF 1.2-rev509	0.642857	0.631579	1	1	1	0.5	1	1	1	9
arachniv0.4.0.3	0.5	0.526316	0.666667	1	1	0.5	1	1	1	9
Skipfish 2.07b	0.5	0.368421	0.666667	1	1	0.5	1	1	1	9
Watobo v0.9.8-rev724	0.142857	0.473684	0.666667	1	1	0.5	1	0	0	7
VEGA 1.0 beta	0.428571	0.105263	0.333333	1	1	0.5	0.666667	0	1	8
Andiparos v1.0.6	0.285714	0.421053	0.333333	0	1	0.5	0.666667	0	1	7
ProxyStrikev2.2	0.214286	0.263158	0.333333	0	0	0.5	0.333333	0	0	5
Wapiti v2.2.1	0.428571	0.210526	0.666667	1	0	0	1	0	0	5
Paros Proxy v3.2.13	0.285714	0.421053	0.333333	0	1	0.5	0.666667	0	0	6
Grendel Scan v1.0	0.142857	0.473684	0.333333	1	1	0.5	1	1	0	8
PowerFuzzer v1.0	0.357143	0.157895	0.666667	0	0	0.5	0.333333	0	0	5
Oedipus v1.8.1	0.214286	0.263158	0.333333	0	1	0.5	1	0	1	7
UWSS v0.0.2	0.285714	0.105263	0.333333	1	0	0	0.666667	0	0	5
WebScarabv20100820	0.071429	0.315789	0.333333	0	0	0.5	0	0	0	4
WSTool v0.14001	0.071429	0.105263	0.333333	0	0	0	0	0	0	3

Table 4 Shows The Averages Of The Application Have 8 Or 9 Features Support.

	1	2	3	4	5	6	7	8	10		average
Zed Attack Proxy v1.4.0.1	0.285714	0.578947	0.333333	1	1	0.5	1	0	1	8	0.633111
W3AF 1.2-rev509	0.642857	0.631579	1	1	1	0.5	1	1	1	9	0.863826
arachniv0.4.0.3	0.5	0.526316	0.666667	1	1	0.5	1	1	1	9	0.79922
Skipfish 2.07b	0.5	0.368421	0.666667	1	1	0.5	1	1	1	9	0.781676
VEGA 1.0 beta	0.428571	0.105263	0.333333	1	1	0.5	0.666667	0	1	8	0.559315
Grendel Scan v1.0	0.142857	0.473684	0.333333	1	1	0.5	1	1	0	8	0.605542

Depending on the averages that show at Table 4 we found that W3AF 1.2-rev509, arachniv0.4.0.3 and Skipfish 2.07b are better than the rest of Applications in the table.

As the result of this study the web developer can use one of these three platforms; we recommend W3AF 1.2 because its average is higher than the others.

5. CONCLUSION

Although there are many Open Source Web Application Security Scanners and they have some similar functions, we should choose the best of them. In this paper we have compared and assessed a list of Open Source Web Application Security Scanners with a focus on OWASP Top 10-2013 application security risks. One of the significant results of this research is that W3AF 1.2, arachniv0.4.0.3 and Skipfish 2.07 are the best of our sample. We showed the difference between Open Source Scanners concentrated on Injection, Cross-Site Scripting and Broken Authentication and Session Management.

REFRENCES:

[1] Jan-Min Chen and Chia-Lun Wu, “An Automated Vulnerability Scanner for Injection Attack Based on Injection Point”, *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, 2010, pp..

[2] J. Bau, E. Bursztein, D. Gupta, and J. Mitchell, “State of the Art: Automated Black-Box Web Application Vulnerability Testing”, *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, 2010, pp.332-345.

[3] OWASP Top 10 2013, https://www.owasp.org/index.php/Top_10_2013

[4]http://en.wikipedia.org/wiki/Information_security

[5]http://en.wikipedia.org/wiki/Web_application_security.

[6]Y.-W Huang, F. Yu, C. Huang, C.-H.Tsai, D.-T.Lee, and S.-Y Kuo, “Securing Web Application code by static analysis and runtime protection”, *Proc. 13th Int. Conf. on World Wide Web (WWW’04), NY, USA. ACM*, pp. 40-52.

[7] V.B. Livshits and M. S. Lam, “Finding security errors in Java program with static analysis”, *Proc. 14th Usenix Security Symposium, Baltimore, MD, USA*, 2005.

[8] N. Jovanovic, C. Kruegel, and E. Kirda, “Static analysis for detecting taint-style vulnerabilities in web applications”, *Journal of Computer Security*, 18 (2010), pp. 861-907.

[9]Y. Xie, A. Aiken, “Static detection of vulnerabilities in scripting languages”, *Proc. 15th USENIX Security Symposium*, 2006, pp. 179-192

[10] G. Wassermann and Z. Su, “Sound and precise analysis of web applications for injection vulnerabilities”, *SIGPLAN Notices, vol 42, n06*, 2007, pp.32-41.

[11]M. S. Lam, M. Martin, B. Livshits, and J. Whaley, “Securing Web Applications with static and dynamic information flow tracking”, *Proc . of the 2008 ACM SIGPLAN Symposium on Partial evaluation and semantics based program manipulation (PEPM’08), New York, NY, USA : ACM*, 2008, pp. 3-12.

[12] T. Pietraszek, C.V. Berghe, “Defending against injection attacks through context sensitive string evaluation”, *Recent Advances in Intrusion Detection (RAID-2005), Seattle, WA, USA*, 2005.

[13] C. Kruegel, G. Vigna, “Anomaly Detection of Web-based Attacks”, *Proc. of the 10th ACM Conference on Computer and Communication Security (CCS’03*, October 2003*)*, pp. 251-261.

[14] E. Kirda, C. Kruegel, G. Vigna, and N. Jovanovic, “Noxes: A client-side solution for mitigating cross-side scripting attacks”, *21st*

ACM Symposium on Applied Computing (SAC2006), Dijon, France, 2006.

[15] K.Stefan, E. Kirda, C. Kruegel and N. Jovanovic,“SecuBat: a web vulnerability scanner”, *Proc. of the 15th int. conf. on World Wide Web (WWW ’06), Edinburgh, Scotland*, 2006.

[16] Zoran Djuric, ”A Black-box Testing Tool for Detecting SQL Injection Vulnerabilities”, *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, 2010, pp.216-221.

[17] A. Doup´e, M. Cova, and G. Vigna, “Why Johnny Can’t Pentest: An Analysis of Black-box Web Vulnerability Scanners”, July 2010

[18] J. Fonseca, M. Vieira, and H. Madeira, "Testing and Comparing Web Vulnerability Scanning Tools for SQL Injection and XSS Attacks", *prdc, 13th Pacific Rim International Symposium on Dependable Computing (PRDC 2007)*, 2007, pp.365-372

[19] Fakhreldeen Abbas Saeed, “Comparing and Evaluating Open Source E-learning Platforms”, *International Journal of Soft Computing and Engineering (IJSCE)*, ISSN: 2231-2307, Volume-3, Issue-3, July 2013,pp.244-249.

[20]http://www.sectoolmarket.com/, 27/08/2012.