Using WASSEC to Evaluate Commercial Web Application Security Scanners

Fakhreldeen Abbas Saeed

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. The goal of this paper is to use The Web Application Security Scanner Evaluation Criteria (WASSEC) to compare and contrast the Commercial Web Application Security Scanners, and show the differences between them. We used six factors to do this compression (Protocol Support, Authentication, Session Management, Crawling, Parsing and Testing). The study shows that Acunetix WVS, Ammonite and Burp Suite Professional are the most suitable ones because they have 0.831325, 0.771084 and 0.73494 averages respectively. As the result of this study and depend on the information about the Commercial Web Application Security Scanner we collected; the Acunetix WVS, Burp Suite Professional and Ammonite are the best respectively. So the web developer or administrator can use them together or choose one.

***Index Terms*— Web Application Security Scanner, WASSEC, Evaluation.**

I. 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 Commercial 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 Commercial Web Application Security Scanner depending on the WASSEC. [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 Commercial Web Application Security Scanner. Section 4 presents the evaluation results with discusses. Section 5 is conclusion of the paper.

II. 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:

1) 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.
2) 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].
3) 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 Commercial Web Application Security Scanners.

A. *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.

III. 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 Commercial 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.

A. *WASSEC*

The Web Application Security Scanner Evaluation Criteria

Manuscript received on March, 2014.

Dr. Fakhreldeen Abbas Saeed, Department of Computer Science, College of Computer Science and Information Technology , Alneelain University , Khartoum, Sudan.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

(WASSEC) is a document with guidelines to help the professionals to evaluate web application scanners on their ability to effectively test web applications and identify vulnerabilities. It’s published by Web Application Security Consortium in 2009 [3].

The criteria we used to evaluate the Scanners are mention below:

1) Transport Support

In order to test web applications, a scanner must support all communication protocols that are commonly used by web applications and intermediary network devices. The underlying communication protocol used by web applications is the Hypertext Transfer Protocol (HTTP).

2) Authentication Schemes:

A web application security scanner should support the authentication schemes such as Basic, Digest, HTTP Negotiate (NTLM and Kerberos) and HTML Form-based.

3) Session Management :

During a web application security scan, it is crucial that scanners will maintain a “living” and valid session with the application at all times.

4) Crawling:

Crawling is the term used to describe the action taken by a program as it browses from page to page on a website. The crawler will visit a starting page and parse the provided links, crawling to those pages. This process will continue until some user-defined criteria are reached or the process is completed and there are no more links to crawl. Crawling is essential to a web application security scan - it ensures that the scanner is aware of all linked pages that exist on the website. Crawlers should be as configurable as possible, allowing the user to define a large number of criteria to ensure a thorough and efficient crawl.

5) Parsing :

In order to thoroughly scan a web application for security problems, a web application scanner must first map out the web application's structure and functionality. The mapping process is done by the web crawler component, which makes use of different types of content parsers to extract information from web content. This information may include URLs, HTML forms, HTML form parameters, HTML comments, and so forth.

6) Testing:

Testing an application for vulnerabilities is the core functionality of a web application security scanner. This section lists the types of vulnerabilities that a web application scanner should be capable of detecting, as well as the testing-related configuration and customization options that a scanner should provide.

7) Command and Control:

The Command and Control capabilities of web application scanners can have a significant influence on usability and therefore are an important aspect to consider when conducting an evaluation. The types of Command and Control features most valued by an end user will vary based on the user's situation - some of the following features will be important to a large enterprise with many users and web applications to scan, but will not necessarily apply to a small company with a single user looking for an effective, low-cost scanning solution.

8) Reporting:

In order for scanning results to be viewed outside of the tool's interface, web application scanners should be able to generate reports of each scan. Because reports are often used by different groups within an organization, scanners should provide the ability to customize the format and information included in their reports.

The next table illustrates the assessment criteria and sub feature of it. We aren’t going to use “Command and Control” and “Reporting” in our evaluation.

Table 1: Criteria and Sub Feature of Assessment

NO	Criteria	Sub_Criteria	Number of Sub_Criteria
1	Protocol Support	GET, POST, C OKIE, HEADER, SECRET, PName, Custom, PROXY, GZIP, EFLATE, SSL	11
2	Authentication	BASIC, DIGEST, NTLM, NTLMv2, KERBEROS, FORM,CERT, CAPTCHA ,ypass	7
3	Session Management	Custom Cookie, Custom, Header, Logout, Detection, Exclude, Logout, Exclude, URL,Exclude, Param	6
4	Crawling	Manual Crawl,Html Crawler,Ajax Crawler,Flash Crawler,Applet Crawler,Silverlight Crawler,WSDL Crawler,REST Crawler,Field Autofill,Smart Autofill,Anti CSRF Support,Viewstate Support	12
5	Parsing	XML, XmlATT, XmlTAG, JSON, .NetENC, AMF, JavaSER, .NetSER, WCF, WCF-Bin, WebSock, DWR, URL File	13
6	Testing	SQLi, BSQLi, SSJSi, RXSS, PXSS, DXSS, JSONh, LFI, RFI, CMDExec, UPLOAD, REDIRECT, CRLFi, LDAPi, XPAPHi, MXi, SSI, FORMATi, CODEi, XMLi, Eli, BUFFERo, INTEGERo, CODEDisc, BACKUPf, PADDING, AUTHb, PRIVe, XXE, SESSION, FIXATION, CSRF, ADoS	33
7	Command and Control	Omitted	
8	Reporting	Omitted	

B. Commercial Web Application Security Scanner

Table 2 shows a list of Commercial 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

	Commercial web scanner	Version	Technology	Last Update
1	Acunetix WVS (Commercial Edition)	8.0 (GA) Build 20120613	Unknown (Win32)	13-06-2012
2	Ammonite	1.2 (GA)	.Net 2.0	28-04-2012
3	Burp Suite Professional	1.4.10 (Beta)	Java 1.6.x	01-07-2012
4	IBM AppScan	8.5.0.1 (GA) Build 42-SR1434	.Net 3.5	26-03-2012
5	JSky (Commercial Edition)	3.5.1 (GA) Build 905	Unknown (Win32)	01-04-2011
6	Nessus	5.0.1 (GA) Build 20120701	Unknown (Win32)	01-07-2012
7	Netsparker (Commercial Edition)	2.1.0 (GA) Build 45	.Net 4.0	09-02-2012
8	NTOSpider (Obsolete Version / Results)	5.4(Obsolete) (GA) Build 098	Java 1.6.x	27-04-2011
9	ParosPro	1.9.12 (GA)	Java 1.6.x	28-03-2011
10	QualysGuard WAS	2012-07-27 (GA) Build Update	Unknown (Linux)	27-07-2012
11	Syhunt Dynamic (Sandcat Pro)	4.5.0.0 (GA)	Unknown (Win32)	20-06-2012
12	WebCruiser Enterprise Edition	2.5.1 (GA)	.Net 2.0	09-05-2012
13	WebInspect	9.20.277.0 (GA) Build SB 4.08.00	.Net 3.5	22-03-2012

IV. EVALUATE OF COMMERCIAL WEB APPLICATION SECURITY SCANNERS:

Table 3 and Figure 1 explain the evaluation of Commercial Web Application Security Scanners with six factors (Protocol Support, Authentication, Session Management, Crawling, Parsing and Testing). The last column in Table 3 shows the average of these six factors of WASSEC criteria, the averages column shows that Acunetix WVS, Ammonite and Burp Suite Professional are the most suitable ones because they have 0.831325, 0.771084 and 0.73494 averages respectively as shown in Figure 2.

Table 3: comparing the Scanners based on WASSEC criteria

NO	Web Scanner	Protocol Support	Authentication	Session Management	Crawling	Parsing	Testing	Average
1	Acunetix WVS	11	7	6	9	7	29	0.831325
2	Ammonite	10	6	5	9	8	26	0.771084
3	Burp Suite Professional	11	7	6	9	5	23	0.73494
4	IBM AppScan	8	6	6	8	4	22	0.650602
5	JSky	8	6	5	5	2	16	0.506024
6	Nessus	9	7	6	6	1	16	0.542169
7	Netsparker	8	5	5	5	1	15	0.46988
8	NTOSpider	7	5	6	6	1	14	0.46988
9	ParosPro	6	3	6	4	0	13	0.385542
10	QualysGuard WAS	6	5	6	3	0	13	0.39759
11	Syhunt Dynamic	6	4	6	1	0	9	0.313253
12	WebCruiser	5	0	1	1	0	8	0.180723
13	WebInspect	7	7	6	1	0	4	0.301205

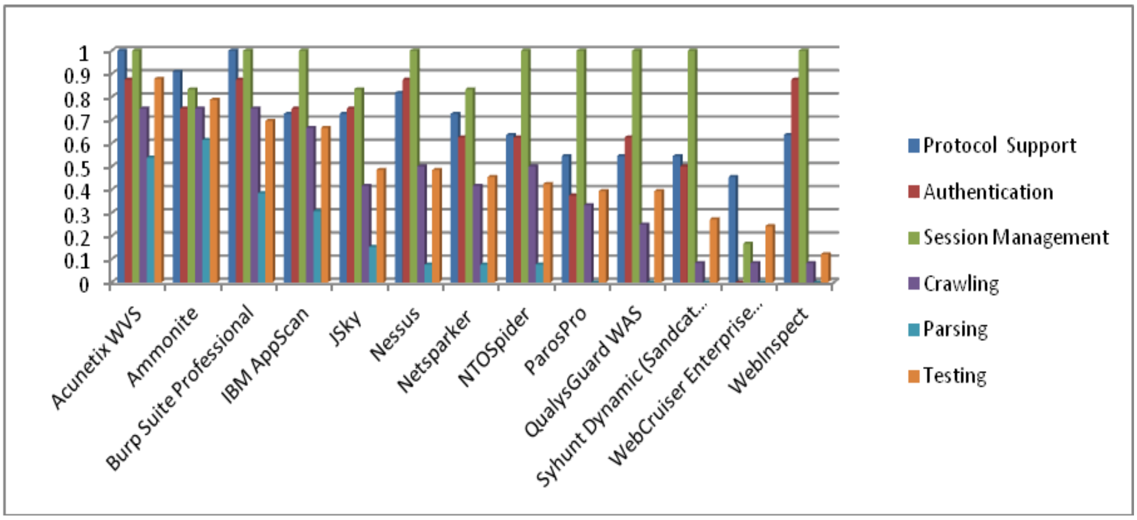

Figure 1: comparing the Scanners based on WASSEC criteria

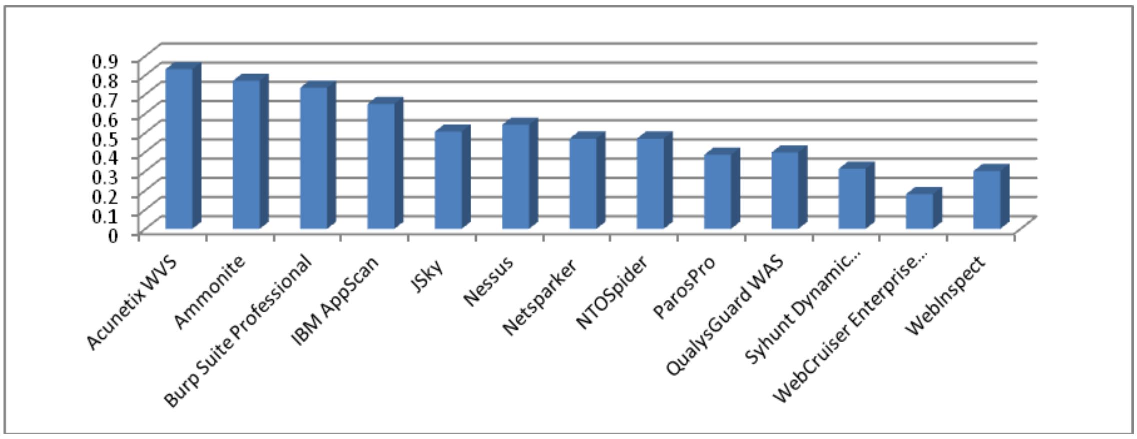

Figure 2: Averages of the evaluation's factors

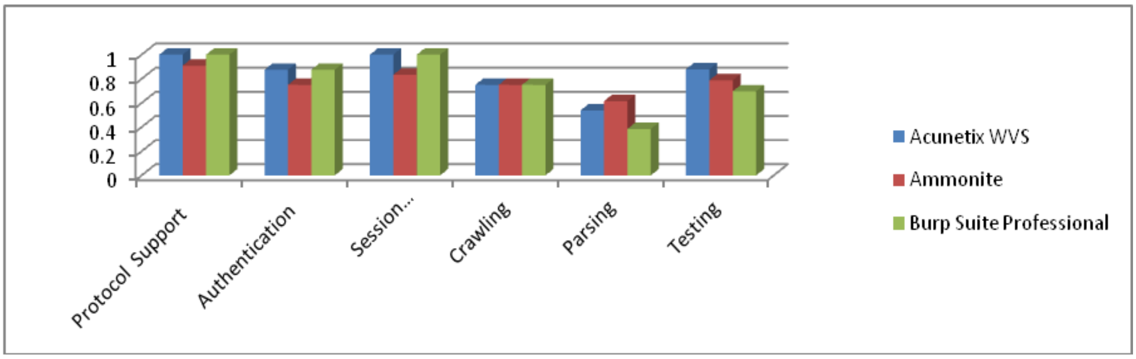

Figure 3: comparing of the best Scanners

Notice that there are variations between the best platforms we choose. These variations illustrate in Figure3. In the Crawling factor the three Scanners are the same. Although Acunetix WVS and Burp Suite Professional are better than Ammonite in Protocol Support, Authentication and Session Management but Ammonite is better than them in parsing factor. So we can say generally Acunetix WVS and Burp Suite Professional are the best; but Acunetix WVS is better in testing. As the result of this study and depend on the information about the Commercial Web Application Security Scanner we collect the Acunetix WVS, Burp Suite Professional and Ammonite are the best respectively.

V.CONCLUSION

Although there are many Commercial 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 Commercial Web Application Security Scanners with a focus on The Web Application Security Scanner Evaluation Criteria (WASSEC). One of the significant results of this research is that Acunetix WVS, Burp Suite Professional and Ammonite are the best of our sample. We showed the difference between Commercial Scanners concentrated on Protocol Support, Authentication, Session Management, Crawling, Parsing and Testing factors.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

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] WASSEC project, Web Application Security Consortium (http://www.webappsec.org).

[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] Price and Feature Comparison of Web Application Scanners (http://www.sectoolmarket.com/) Last updated: 27/08/2012.