

An Overview of Black Box Web Vulnerability Scanners

Manju Khari ^{*} Computer Science Department, GGSIPU Delhi, India Neha Singh Computer Science Department, GGSIPU Delhi, India

Abstract— Web application users and web application vulnerabilities are increasing . Today web applications turning out to be tools of everyday use by many users with the growing popularity of the web. With this web application users are more prone to malicious attacks consequently the need of web security testing arises as well. As security testing helps to mitigate vulnerabilities in the web applications which is quite intricate process so requires the use of efficient security testing technique. Frequently occurring security vulnerabilities in web applications result from generic input validation problems. Examples are SQL injection and Cross-Site Scripting (XSS) etc. These vulnerabilities are more often exploited by attackers to access sensitive information form the websites for their personal gain. Black Box scanners offers a good choice to test for vulnerabilities in an automated fashion. Although the majority of web vulnerabilities are easy to understand and to avoid but still many web developers are not security aware. As a result, there exist many web sites on the Internet that are vulnerable. This paper presents the overview of various open source black-box web vulnerability scanners with their comparison table that can be used to test web application and analyze web sites with the aim of finding exploitable SQL injection and XSS vulnerabilities. Also it concludes the strengths and weaknesses of these open scanners with pointing out future scope. These open source web scanner employs a black-box approach i.e point and shoot manner to test any web application regardless of server side language and indicate presence of exploitable SQL injection and XSS vulnerabilities. These tests are incorporated a pragmatic web application.

Keywords—Black Box Scanner, Open Source, Web Security, Web Services, Web Application and Vulnerability.

I. INTRODUCTION

Black-box web vulnerability scanners are a category of tools that can be used to identify security issues in web Applications[1]. These tools are often known as "point-and-click pentesting" tools that automatically assess the security of web applications with little or no human intervention. These tools access a web application in the same way users do, and, therefore they are independent of the particular technology being used to implement the web application at the server side. However, these tools should also be able to access and test the applications[3]. Black-box web application vulnerability scanners are automated tools that explore web applications for security vulnerabilities. In black-box testing, the source code is not examined instead, special input test cases are generated and sent to the application. Then, the results returned by the application are analyzed for unforseen behavior that indicate loopholes or vulnerabilities[2].

Some features of Black-box web vulnerability scanners are:

- Black-box web vulnerability scanners are a modern choice for finding security loopholes in web applications in an automated manner.
- These tools functions in a *point-and-shoot* manner, testing any web application—regardless of the server-side language—for common security vulnerabilities.
- Black-box tools suffer from a number of limitations, particularly when interacting with complex applications that have multiple actions.
- If a vulnerability analysis tool does not take into consideration changes in the web application's state, it might ignore vulnerabilities or completely overlook entire portions of the application[4].

Classical black-box web scanners crawl a web application to enumerate all reachable pages and then inject some input data (URL parameters, form values, cookies) to trigger vulnerabilities. However, this approach ignores a key aspect of modern web applications: The state of the web application changes according to the current request[3]. Web application (black-box) scanners perform security tests on Web applications by (usually) first crawling through the entire Web site that's holding the Web application, and then running specific security test cases wherever possible. All the tests are performed over the HTTP protocol.They are not only effective at finding attack incidents like cross-site scripting and SQL injection , but also at finding configuration management issues (related to Web servers). These tools are usually not aimed at developers, this makes the mitigation process complex[10].



Fig 1: WAS Lifecycle

A. Select web application(s). The web applications selected by the user is used as a scan targets. A web application is defined as a virtual host (IP address or FQDN), starting URL and starting port, as well as optional settings for white/black lists, authentication, and also business information for tracking.

B. Specify WAS scan options. Scan options allow you to configure crawling settings, opt in for sensitive content tests, limit the scan to certain vulnerability tests (instead of all tests), and select a level of password bruteforcing..

C. WAS scan. The sophisticated scanning engine uses several techniques to effectively crawl a web site and identify vulnerabilities, information gathered data and sensitive content data. The test phase of the scan tests for common vulnerabilities such as cross-site scripting ,SQL injection, directory traversal and source disclosure. Running scheduled scans allows users to perform automated scanning and regularly refresh the web application security data in your account [6].

D. Review WAS scan Results and Report generation. The WAS reporting engine displays the problems in the form of different kinds of vulnerabilities and generates summary information across links of web applications. Generating interactive and scorecard reports show the most recent web application scan data in an easy-to-read output.

S.no	Scanners	Vendor	Technology	Platform
1.	Websecurify	GNUCITIZEN	Javascript (General)	Windows, Mac, Linux mainly all major
2.	Skipfish	Michal Zalewski	C (General)	Linux, FreeBSD, Mac OS X, and Windows (Cygwin)
3.	Wapiti	Informática Gesfor	Python (2.6.x)	Unix/Linux, FreeBSD, Mac OS X, and Windows
4.	Parosproxy	MileSCAN	Java 1.4x	Linux, Mac OS X and Windows
5.	Arachni	Tasos Laskos	Ruby (1.9.x)	windows
6.	Openacunetix	John Martinelli	Java1.6	Windows
7.	Grendel-Scan	David Byrne	Java (1.5.x)	Linux, Mac OS X, and Windows
8.	W3af	W3AF developers	Python (2.5.x)	Linux, Mac OS X, and Windows
9.	WebScarab	OWASP	Java (1.5.x)	Linux, Mac OS X, and Windows
10.	Sqlmap	sqlmap developers	Python 2.6	Linux, Mac OS X, and Windows
11.	Zap	OWASP	Java 1.6x	Linux, Mac OS X, and Windows
12.	Andiparos	Compass Security AG	Java1.5x	Linux, Mac OS X, and Windows
13.	watabo	Andreas Schmidt	Ruby1.8x	Linux, Mac OS X, Backtrack and Windows

III. DESCRIPTION OF VARIOUS WEB SCANNERS Table 1: General Features of Web Scanners

Khari et al., International Journal of Advanced I	Research in Computer	• Science and Software	Engineering	4(5),
		May - 2	014, pp. 1527-	1535

Usage scale					Scan Initiation Method			Output		
Scanners	GU I	Configurat ion	Usage	Stabilit y	Perform ance	Spid er	Manu al Crawl ing	File parsi ng	Repo rt	Log
Websecu rify	Yes	Very Simple	Very Simple	Very stable	Fast	Yes	No	no	Yes	No
Skipfish	No	Simple	Compl ex	Stable	Fast	Yes	No	yes	Yes	Yes
Wapiti	No	Complex	Compl ex	Fragile	Fast	Yes	No	No	Yes	Yes
Open Acunetix	Yes	Simple	Simple	Stable	Fast	Yes	Yes	No	No	No
Arachni	Yes	Simple	Compl ex	Stable	Fast	Yes	Yes	No	Yes	No
Parospr oxy	Yes	Very simple	Very Simple	Unstabl e	Slow	Yes	Yes	No	Yes	Yes
Andipar os	Yes	Very simple	Stable	Stable	Fast	Yes	Yes	No	Yes	Yes
Zap	Yes	Very simple	Very Simple	Very stable	Fast	Yes	Yes	No	Yes	Yes
Grendel- Scan	Yes	Simple	Simple	Stable	Slow	Yes	Yes	No	Yes	Yes
W3af	Yes	Complex	Compl ex	Fragile	Slow	Yes	Yes	No	Yes	Yes
Webscar ab	Yes	Very simple	Very Simple	Stable	Fast	Yes	Yes	No	Yes	Yes
Watabo	Yes	Very simple	Very Simple	Unstabl e	Fast	No	Yes	No	No	Yes
Sqlmap	Yes	Complex	Simple	Stable	Slow	No	Yes	Yes	Yes	Yes

Table 2: Usage, Coverage and Output features of Scann	ers
---	-----

lat	Sie 3: Description of the terms used in Table 2					
Title	Possible values					
Configuration & usage scale	 Very Simple - GUI+ Wizard Simple -GUI with simple options, command line with scan configuration file or simple options Complex - GUI with numerous options, Command line with multiple options Very complex - Manual scanning feature dependencies, multiple configuration requirements 					
Stability scale	 Very stable – rarely carshes and never gets stuck Stable – Rarely crashes, gets stuck only in extreme scenarios. Unstable –crashes every once in a while, freezes on a consistent basis Fragile - freezes or crashes on a consistent basis, fails operating the operations in many cases. 					
Performance scale	 <i>Very fast</i> – fast implementation with limited amount of scanning tasks <i>Fast</i>- fast implementation with plenty of scanning tasks <i>Slow</i> – Slow implementation with limited amount of scanning tasks <i>Very slow</i> – slow implementation with plenty of scanning tasks. 					

A. Skipfish

Skipfish is a new open source web application scanner, written in C programming, developed by Google. The goal of Skipfish is similar to the goals of previous web security hole scanners like Nmap and Nessus, it allows web developers to scan their application or site for possible security issues that may be lurking around. Skipfish can be used to determine if code is vulnerable to common attacks such as cross-site scripting (XSS), SQL, and XML injection attacks because it performs high risk flaw, medium risk flaw, and low issue scans.() After Skipfish completes its scan it prepares an interactive site-map for the targeted site by carrying out a recursive crawl and dictionary based probes. Skipfish is said to easily process over two thousand HTTP requests per second if the server being tested can handle the load.

Features

- Available for all major platforms (Windows, Linux, FreeBSD and MacOS)
- Gui based
- High speed
- Ease of use high quality,
- Low false positive,
- Differential security checks,
- Capable of spotting arrange of subtle flaws, including blind injection

B. W3af

W3af (Web Application audit and attack framework) is a framework for auditing and exploitation of web applications. It is an extremely popular, powerful, and flexible framework for finding and exploiting web application vulnerabilities. It is easy to use, extend and features dozens of web assessment and exploitation plugins . In some ways it is like a web focused .W3af is divided in two main parts :the core and the plugins. The core coordinates the process and provides features that plugins use. Plugins find the vulnerabilities and in some cases are able to exploit them . Plugins share information with each other using a knowledge base.

Features

- More than 130 plug-ins.
- Fuzzy and Manual request generator feature.
- Easy to use and extend
- Remote command execution
- Explot the vulnerability that it finds

C. Paros proxy

Paros was written completely in Java by people from ProofSecure.com .Paros Proxy is an intercepting proxy server it can intercept all HTTP and HTTPS dat between server and client ,including cookies and form fields, and they can be modified before data reached destination .It is for evaluating the security of web applications or for hacking and

exploiting .It is good tool to exploit and hack websites. A Java-based web proxy for assessing web application vulnerability. It supports editing/viewing HTTP/HTTPS messages on-the –fly to change items such as cookies and form fields. It includes a web traffic recorder, hash calculator, web spider and a scanner for testing common web application attacks such as SQL injection and cross-site scripting[4].

Features

- Intercepting Proxy
- Active and Passive scanning facility
- Spider crawler
- Ability to extend the features using plugins and extensions
- Graphical user interface
- Filters
- Support client certificate

D. Websecurify

Websecurify is an integrated web security testing environment, which can be used to spot vulnerabilities by using advanced browser automation, detection and fuzzing technologies. It is a powerful web application security testing environment designed from the ground up to offer the best combination of automatic and manual vulnerability testing technologies. The platform is devised to perform manual as well as automated vulnerability tests and it is persistently improved and fine-tuned by a team of world class web application security penetration testers and the feedback from an active open source community. Websecurify uses several key technologies united together to achieve the best possible result when performing automatic and manual tests. This allows Websecurify to get a fine-grained control over the targeted web application and as such detect vulnerabilities that are difficult to find with other tools [8].

Features

- Available for all major platforms (Windows, Mac OS, Linux)
- The basic/default scanner is smarter, faster, lighter and more selective
- The scanning technology is memory efficient and the memory storage is swappable
- Manual scan, ajax scan, google scan and basic scan
- User friendly interface
- Built-in support
- Easily extensible with the help of plugins and add-ons.
- Customizable and exportable reports with any level of detail
- Powerful manual testing tools and helping facilities
- Extensible via many languages including Python, JavaScript, C, C++ and Java

E. Arachni

Arachni is an Open Source, high-performance, feature-full, modular, Ruby framework aimed towards helping penetration testers and administrators assess the security of web applications. <u>Arachni</u> is smart enough that it trains itself with every HTTP response it receives during the auditprocess and is able to perform meta analysis using a number of factors in order to correctly evaluate the accuracy of results and intelligently make out false-positives.

Unlike other scanners, Arachini takes into account the dynamic nature of web applications and can detect changes occurred while travelling through each path of a web application's cyclomatic complexity. Thus the attack/input vectors that are left undetected by non-humans are flawlessly handled by Arachni. Arachni has been designed to automatically identify security issues in the web applications. All it expects is the URL of the target website and after a while it will present you with its findings.

Features

- Cookie-string support.
- Great performance due to its asynchronous HTTP model
- Custom header support is provided.
- It provides SSL support.
- User Agent spoofing.
- Proxy authentication is done.
- Automatic log-out detection and re-login during the audit (when the initial login was performed via the AutoLogin plugin).
- Custom 404 page error detection.
- UI abstraction:
- Command-line Interface.
- Web User Interface.
- Pause/resume options available.
- A stable, high-performance and efficient, framework

F. Grendel-Scan

It is an open source web application security testing tool. It has a feature of identifying common web vulnerabilities and also peneteration tester for manual testing. Its system requirements are windows, linux, java5 and Machintosh.For testing of website it can be run as proxy between browser and website to let the tester allow to view and manipulate http traffic.

Features

- Internal intercepting/testing proxy
- HTTP request fuzzer
- Manual requests
- Automaticfile-not-found profiles
- Upstream proxy support
- Granular scan settings
- URL white-lists & blacklists

G. Wapiti

Wapiti is a kind of tool that performs "black-box" scans and allows to audit the security of the web application. To perform scan it doesn't require the source code but the web pages of the deployed webapp. It looks for scripts html forms or input points in order to inject data. After it gets the list of input points, it acts like a fuzzer to check out if a script is vulnerable. It also helps in detecting database injection ,file handling errors , injection command execution detection and CRLF injection.

Features

- Cookies management
- Fast and easy to use
- Can suspend or resume a scan or an attack.
- Try to extract URL from javascript
- Authentication via several methods like basic, digest, Kerberos etc.
- Main vulnerabilities detected
- File handling errors
- Database injection
- Command execution detection
- Extensibility
- Vulnerability Report generation in various formats.

H. Zaproxy

ZAP is a fork of the open source variant of the Paras Proxy. It is a produced by OWASP. It is easy to perform penetration testing for web application with the help of this tool. Highly experienced security experts needed to use such kind of tool and is ideal for functional testers and developers who are novice penetration tester.ZAP provides a set of tools and automated scanners as well to find security vulnerabilities manually[7].

Features

- It has Cross platform
- Open source
- Simple to install (just requires java 1.7)
- It is completely free
- It also has a help page option.
- Fully internationalized
- It is available in various languages
- It requires human involvement.
- It is under active development by international team of volunteers

I. Watcher

Watcher is a runtime passive-analysis tool for HTTP-based Web applications which aims to assist penetration testers in passively finding Web-application vulnerabilities. Being passive means it won't damage production systems, it's completely safe to use in Cloud computing, shared hosting, and dedicated hosting environments. Watcher detects Web-application security issues as well as operational configuration issues. Watcher provides pen-testers hot-spot detection for vulnerabilities, developers quick sanity checks, and auditors PCI compliance auditing. It looks for issues related to mashups, user-controlled payloads (potential XSS), cookies, comments, HTTP headers, SSL, Flash, Silverlight, referrer leaks, information disclosure, Unicode, and more. Watcher is built in C# as a small framework with 30+ checks already included. It's built so that new checks can be easily created to perform custom audits specific to your organizational policies, or to perform more general-purpose security assessments.

Features

- Detection of privacy, security and PCi issues in HTTP, HTML javascript in a passive manner.
- It can work seamlessly with comlex web application 2.0.
- It will not raise alarms or cause harm to production sites i.e Non intrusive.
- It performs real time analysis and reporting.
- Configurable support with wildcard support.
- Extensible framework for adding new checks.

Some reasons to use Watcher include:

- Safe for the Cloud and hosting environments.
- Safe for production environments.
- Low overhead, no training.

J. Andiparos

Andiparos is a kind of the famous Paros Proxy. It is an open source security assessment tool that offers peneteration testers the ability to spider websites, analyze content, intercept and modify requests. It performs authentication with the support of client certificates on Smart cards. Apart from this it has several small interface enhancements that makes the life.

It is an open source web application security assessment tool that gives penetration testers the ability to spider websites, analyze content, intercept and modify requests, etc. The advantage of Andiparos is mainly the support of Client Certificates on Smartcards. Moreover it has several small interface enhancements, making the task easier for penetration testers.

Features

- BeanShell support
- Smartcard smart
- It has History filter option.
- Passive scanner
- Advanced serach functionality feature
- Mark Request/Response
- MultiTags for request/response
- Better Mac OS X integration and other enhancements.

K. Watobo

WATOBO is developed by Andreas Schmidt, siberas (<u>http://www.siberas.de</u>).Web Application ToolBox aka **Watobo** is a tool that behaves like local proxy, similar to Webscarab, Paros or BurpSuite. It is capable of passive as well as active scanning. It is designed to enable security experts to perform highly efficient web application security audits. As this semi –automated approach the best way to perform an accurate audit to identify most of the vulnerabilities.This tool has no no attacking capabilities and is just provided for security audit purpose only. It actually performs passive and active checks. Passive checks are more like filter functions. They are used to collect useful information, e.g. email or IP addresses. Passive checks instead will produce a high number of requests depending on the check module because they do the automatic part of vulnerability identification, e.g. during a scan.

Features

- It has Session Management capabilitiesWATOBO can act as an transparent proxy.
- It has anti CSRF features
- Runs under Windows,Linux,BackTrack, MacOS
- It can perform vulnerability checks out of the box.
- WATOBO has Inline DE-/encoding feature.
- It has also smart filter functions, so you can easily find and navigate to the most interesting parts of the application.
- WATOBO is basically written in Ruby and enables you to define your own checks.
- Easy to extend and adapt

L. WebScarab

WebScarab is a blueprint for analysing applications that communicate using the HTTP and HTTPS protocols. It is written in Java, and is thus portable to many platforms. WebScarab has several modes of operation, implemented by a number of plugins. In its most common usage, WebScarab operates as an intercepting proxy, allowing the operator to review and modify requests created by the browser before they are sent to the server, and to review and modify responses returned from the server before they are received by the browser. WebScarab is able to intercept both HTTP and HTTPS communication. The operator can also review the conversations (requests and responses) that have passed through WebScarab.

WebScarab is a Java based tool maintained by OWASP (The Open Web Application Security Project) used for intercepting and requests and responses between a browser and HTTP/S server. Using the fuzzer tool, you can find possible vulnerabilities in your web site's code[7].

Features

- It fragments ,extracts scripts and HTML comments from HTML pages as they through the proxy or plugins.
- Proxy
- Manual intercept
- Reveal hidden
- Badwidth simulator
- It has spider crawler that identifies new URLs on the target site, and fetches them on command.
- It has manual request feature.
- Its parameter fuzzer that performs automated substitution of parameter values that are likely to expose incomplete parameter validation, which leads to vulnerabilities like Cross site scripting(XSS) and SQL injection.
- A plugin called as SOAP that parses WSDL, and presents the various functions and the required parameters allowing them to be edited before being sent to the server.
- Also supports Extensions that automates checks for files that were wrongly left in web server's root directory[7].

IV. FREE WEB APPLICATION SCANNERS ADVANTAGES AND LIMITATIONS

A. Strength of open source web scanners

- First of all, there are no costs associated with these tools.
- Free web application scanners have multiple options and in-depth scan coverage capability for any websites
- Free scanner help many security researchers and developers to at least discover the basic vulnerabilities present in their web applications
- It is not easy that every developer can use the commercial tools for testing as they incur a high cost.
- These scanners as the dynamic testing tools are not language dependent means they are independent of technology being implemented at the server side.
- Scanners simulate various malacious users by triggering vulnerabilities in order to perform an attack and to compare the reported results with the expected result.

B. Weaknesses and limitations

- As these tools are free so they are usually not revised with latest security loopholes present in recently updated languages, so attackers might try to exploit these vulnerabilities and perform an attack if they can learn the language used by target website.
- These scanners do not cover the entire features as available in certified versions.
- If u don't have some security expertise then its usually not possible to determine the best performing scanner, due to which many business owners stop running multiple free tolls if they don't get the expected result from the initial one.
- Some attacks like Botnets are difficult to leave some networks that are used by malacious users .
- As these scanners are dynamic tools so they lack the 100% code coverage feature. The penetration tester should look at the code coverage to know if the tool was configured correctly.
- These scanners are not good enough to find logical errors such as weak cryptographic functions, information leakage etc.
- Also technical flaws are hardly detected by the tool if web application doesn't provide certain clues.
- These tools generally have a predefined list of attacks so tool cannot implement all variants of attacks for a vulnerability.
- These free scanners also lack in understanding of behavior of applications with some dynamic content such as Javascript, Flash etc.
- These scanners are also not appropriate for detecting social engineering flaws that are common to attackers.

V. CONCLUSION

In this paper we discussed various open source black box scanners in detail along with their general and coverage features which can help one to decide the better scanner to test his/her particular website for various vulnerabilities with little human intervention. We made a comparison on these tools over various parameters on the basis of which we can conclude that webscarab is the only scanner that meets various requirements and has good performance among all as it has very simple configuration and usage with report generation and scan logging capabilities. Webscarab supports spider as well as manual scanning as deep crawling is necessary to identify all vulnerabilities in an application. Further we discussed the strengths and weaknesses of these scanners. In future these scanners could be chosen to evaluate and aggregate data about all scanners to indicate the vulnerability detection and false positive performance of the finest scanner on each of the several measures and rate on established web applications in order to understand how effective they are, how the tools work and in what situation they fail. Also to identify the tasks that are most challenging for black-box vulnerability scanners.

ACKNOWLEDGEMENT

I take immense pleasure in thanking Ms. Manju Khari Assistant professor at Ambedkar Institute of Advanced Communication Technologies and Research my Guide for having permitted me to carry out this work. I express my deep sense of gratitude for her able guidance and useful suggestions, which helped me in completing the work, in time.

Finally, yet importantly, I would like to express my heartfelt thanks to God, my parents for their blessings, my friends for their help and wishes for the successful completion of this task.

REFERENCES

- [1] T Giuseppe A. Di Lucca and A. R. Fasolino, "*Testing Web-based applications*: The state of the art and future trends", *Elsevier*, Information and Software Technology,2006.
- [2] A. Arora, M. Sinha, "Web Application Testing: A Review on Techniques, Tools and State of Art", International Journal of Scientific & Engineering Research, 2013.
- [3] J. Bau, E. <u>Bursztein</u>, D. Gupta, and J. Mitchell, "State of the Art: Automated Black-Box Web Application Vulnerability Testing", IEEE, 2010.
- [4] B. Shura, "Web application security scanner evaluation criteria", 2009.
- [5] Mark Curphey and Rudolph Araujo, "Web Application Security Assessment Tools", IEEE,2006.
- [6] M. Vieira, N. Antunes, and H. Madeira, "Using Web Security Scanners to Detect Vulnerabilities in Web Services", *IEEE/IFIP Intl* Conf. on Dependable Systems and Networks, June 2009.
- [7] E. Fong and V. Okun, "Web Application Scanners: Definitions and Functions", Hawaii International Conference on System Sciences, January 2007.
- [8] Websecurify scanner. http:// www.websecurify.com/
- [9] Arachni scanner. http://www.arachni-scanner.com/
- [10] A. Doup'e, M. Cova, and G. Vigna, "An Analysis of Black-box Web Vulnerability Scanners", 2012.