APM 3329 - OWASP ZAP Research

The research outcome is to identify and compare commercial and open source tools used for scanning security vulnerabilities.

# Penetration Testing:

Penetration test is an authorized simulated attack on a computer system, performed to evaluate the security of the system.

**Web Application Vulnerability Scanners** are automated tools that scan web applications, normally from the outside, to look for Security Vulnerabilities such as Cross-site scripting, SQL Injection, Command Injection, Path Traversal, insecure server configuration etc.

# Security Vulnerability Testing Tools:

These are some of the security vulnerability testing tools that are available.

| Commercial Tools | Open Source Tools |
| --- | --- |
| * [Burp Suite](https://portswigger.net/burp) * [Netsparker](https://www.netsparker.com/web-vulnerability-scanner/) * [GamaScan](http://www.gamasec.com/gamascan.aspx) * [AppScan](https://www.ibm.com/security/application-security/appscan) * [Acunetix](https://www.acunetix.com/) | * [OWASP-ZAP](https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project) * [Wapiti](http://wapiti.sourceforge.net/) * [Vega](https://subgraph.com/vega/) * [Skipfish](https://tools.kali.org/web-applications/skipfish) * [SQLMap](http://sqlmap.org/) * [W3af](http://w3af.org/) * [Metasploit](https://www.metasploit.com/) * [Arachni](http://www.arachni-scanner.com/) |

# Evaluation Criteria:

* General Features - GUI, Config, Stability, Performance
* Authentication, Control and Connection Features - Proxy, SSL, Cert, Custom Cookie, Logout/Session Timeout Detection
* Coverage Features - Manual Crawl, HTML Crawler, REST crawler
* Input Vector Support - GET, POST, Cookie, XML, JSON
* Audit Features -  SQLi, CSRF, RXSS

# Comparison between the Tools:

Specialized tools are readily available for discovering vulnerabilities and security gaps in applications. In this comparison, we will compare the list of tools with the evaluation criteria mentioned above.

These tools alert testers of weaknesses that are readily exploitable by cyber attackers (e.g. a SQL Injection flaw or cross-site scripting issue)

## General Features:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Features | ZAP | Burp Suite | Net Sparker | Arachni | AppScan | Wapiti | Vega | Skipfish | SQLMap | W3af | Acunetix |
| GUI |  |  |  |  |  |  |  |  |  |  |  |
| Performance | Fast | Very Fast | Fast | Fast | Fast | Fast | Very Fast | Very Fast | Slow | Fast | Fast |
| Report Generation |  |  |  |  |  |  |  |  |  |  |  |
| Save/Load Scan Session |  |  |  |  |  |  |  |  |  |  |  |
| Usage | Very Simple | Very Simple | Very Simple | Very Simple | Simple | Complex | Very Simple | Simple | Simple | Complex | Very Simple |

## Authentication, Control and Connection Features:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Features | ZAP | Burp Suite | Net Sparker | Arachni | AppScan | Wapiti | Vega | Skipfish | SQLMap | W3af | Acunetix |
| Custom Cookie |  |  |  |  |  |  |  |  |  |  |  |
| Custom Header |  |  |  |  |  |  |  |  |  |  |  |
| Basic Authentication Support |  |  |  |  |  |  |  |  |  |  |  |
| Digest Authentication Support |  |  |  |  |  |  |  |  |  |  |  |
| NTLM Authentication Support |  |  |  |  |  |  |  |  |  |  |  |
| Kerberos Authentication Support |  |  |  |  |  |  |  |  |  |  |  |
| Html Form Authentication Support |  |  |  |  |  |  |  |  |  |  |  |
| Proxy |  |  |  |  |  |  |  |  |  |  |  |
| SSL |  |  |  |  |  |  |  |  |  |  |  |
| Cert |  |  |  |  |  |  |  |  |  |  |  |
| Logout/ Session timeout detection |  |  |  |  |  |  |  |  |  |  |  |

# Coverage Features:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Features | ZAP | Burp Suite | Net Sparker | Arachni | AppScan | Wapiti | Vega | Skipfish | SQLMap | W3af | Acunetix |
| Manual Crawl |  |  |  |  |  |  |  |  |  |  |  |
| URL File Parsing support |  |  |  |  |  |  |  |  |  |  |  |
| HTML Crawler |  |  |  |  |  |  |  |  |  |  |  |
| Ajax Crawler |  |  |  |  |  |  |  |  |  |  |  |
| Flash Crawler |  |  |  |  |  |  |  |  |  |  |  |
| Applet Crawler |  |  |  |  |  |  |  |  |  |  |  |
| Silverlight Crawler |  |  |  |  |  |  |  |  |  |  |  |
| WSDL Crawler |  |  |  |  |  |  |  |  |  |  |  |
| REST Crawler |  |  |  |  |  |  |  |  |  |  |  |
| Field Autofill |  |  |  |  |  |  |  |  |  |  |  |
| Smart Autofill |  |  |  |  |  |  |  |  |  |  |  |
| Anti CSRF Support |  |  |  |  |  |  |  |  |  |  |  |
| View state Support |  |  |  |  |  |  |  |  |  |  |  |
| CAPTCHA Bypass |  |  |  |  |  |  |  |  |  |  |  |
| WAF Bypass |  |  |  |  |  |  |  |  |  |  |  |

# Input Vector Support:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Features | ZAP | Burp Suite | Net Sparker | Arachni | AppScan | Wapiti | Vega | Skipfish | SQLMap | W3af | Acunetix |
| GET |  |  |  |  |  |  |  |  |  |  |  |
| POST |  |  |  |  |  |  |  |  |  |  |  |
| Cookie |  |  |  |  |  |  |  |  |  |  |  |
| Header |  |  |  |  |  |  |  |  |  |  |  |
| Secret Parameters |  |  |  |  |  |  |  |  |  |  |  |
| XML attributes |  |  |  |  |  |  |  |  |  |  |  |
| XML Tags |  |  |  |  |  |  |  |  |  |  |  |
| JSON |  |  |  |  |  |  |  |  |  |  |  |
| Web socket |  |  |  |  |  |  |  |  |  |  |  |
| Custom |  |  |  |  |  |  |  |  |  |  |  |

# Audit Features

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Features | ZAP | Burp Suite | Net Sparker | Arachni | AppScan | Wapiti | Vega | Skipfish | SQLMap | W3af | Acunetix |
| SQLi |  |  |  |  |  |  |  |  |  |  |  |
| Server-Side Java Script Injection |  |  |  |  |  |  |  |  |  |  |  |
| RXSS |  |  |  |  |  |  |  |  |  |  |  |
| PXSS |  |  |  |  |  |  |  |  |  |  |  |
| DXSS |  |  |  |  |  |  |  |  |  |  |  |
| JSONh |  |  |  |  |  |  |  |  |  |  |  |
| LFI/RFI |  |  |  |  |  |  | * / |  |  |  |  |
| Command Injection |  |  |  |  |  |  |  |  |  |  |  |
| Code Injection |  |  |  |  |  |  |  |  |  |  |  |
| MXi(Email Injection) |  |  |  |  |  |  |  |  |  |  |  |
| Server-Side Injection |  |  |  |  |  |  |  |  |  |  |  |
| XMLi |  |  |  |  |  |  |  |  |  |  |  |
| Weak Session Identifier |  |  |  |  |  |  |  |  |  |  |  |
| CSRF |  |  |  |  |  |  |  |  |  |  |  |
| ADoS |  |  |  |  |  |  |  |  |  |  |  |

# Pros and Cons of Security Tools

We have outlined some of the important pros and cons of security testing tools here.

## **OWASP ZAP**

**PROs**

* Since OWASP ZAP is written in Java it is platform independent.
* One of the unique features of ZAP is that its sensitivity and scan aggressiveness can be manually configured. There are three sensitivity settings (high, medium and low).
* ZAP allows a user to save sessions and persist sessions

**CONs:**

* ZAP does best on older applications with older vulnerabilities.
* if the scanner found ten occurrences of cross-site scripting, ZAP will create a unique row for each of these ten. To make things worse, these ten rows will not be grouped but distributed throughout the report.

## **Burp Suite**

**PROs**

* Penetration testing of web applications
* Web vulnerability scanning
* Customized scan and attack applications

**CONs**

* Easy to use, but difficult to master, but only because of the extensive functionality and customization options.
* Some polish to the GUI and reports would be nice.

## **Metasploit**

**PROs**

* Workspaces: Metasploit allows for the creation of "workspaces," which allow for shared and collaborative penetration testing.
* Information management: Metasploit stores and displays detailed information about devices and networks that would otherwise be difficult to manage.
* Community driven: Many developers from all over the world contribute to Metasploit. This helps to keep it functioning well and up-to-date.

**CONs**

* It supports payload written only in Ruby language.

## **AppScan**

**PROs**

* A high-quality scanner with acceptable results on most sites.

**CONs**

* JavaScript crawling was not as effective as would be desire

## **Netsparker**

**PROs**

* Provides detailed information about the vulnerabilities in your websites and proof of exploits. This improves your ability to identify, prioritize and fix the vulnerabilities.
* Netsparker can be set to automatically scan hundreds of sites. You no longer have to scan each site individually and spend your time writing reports.
* Includes AJAX and JavaScript support, improving your ability to detect client-side vulnerabilities.

**CONs**

* Netsparker provides a detailed list of vulnerabilities, and may provide an attacker with all the information they need to exploit identified vulnerabilities.

## **Arachni**

**PROs**

* Results in different output formats
* Easy and simple to use
* Lots of plugins

**CONs**

* Doesn’t scan a directory (scanner continues on the base URL)
* No scanning profiles

## **Wapiti**

**PROs**

* Generates vulnerability reports in various formats
* Fast and easy way to activate/deactivate attack modules
* Can give you colors in the terminal to highlight vulnerabilities

**CONs**

* Finds less vulnerabilities
* Return Memory Errors for some scans
* Very small community

## **W3AF**

**PROs**

* Plug-in approach – use what you want, write your own tests
* Many plugins provided
* Made for command line execution (but GUI available)

**CONs**

* Problems with BlindSQL-and Evalplugin (Too many retries)
* XML report badly structured

## **Vega**

**PROs**

* Vega has a well-designed graphical user-interface.
* Multi-Platform - Vega is written in Java and runs on Linux, OS X, and Windows.
* Extensible - Vega detection modules are written in JavaScript. It is easy to create new attack modules using the rich API exposed by Vega.
* The tool also allows you to set preference such as maximum and minimum requests per second, the number of path descendants and number of nodes, etc.

**CONs**

* Reports cannot be exported

## **Skipfish**

**PROs**

* Easy to install and run
* The software claims to handle 2K requests per second, without displaying CPU footprints
* Cutting-edge security logic: high quality, low false positive, differential security checks, capable of spotting a range of subtle flaws, including blind injection vectors.

**CONs**

* It is no longer maintained. Last version, 2.10 beta, released in December 2012, can be still downloaded from Google Code Archive.

## **SQLMap**

**PROs**

* Huge amount of options
* Contains lots of evasion filters

**CONs**

* Huge amount of options
* Default option are in ‘safe mode’

## **Acunetix**

**PROs**

* Acunetix was a close third behind AppScan after being trained to find every link.

**CONs**

* Acunetix missed 53% of the vulnerabilities even after being trained to know all the pages. Acunetix missed 31% of the vulnerabilities after training and 37% without training. This is a significant cause for concern as they should be aware of the links vulnerabilities on their own site and be able to crawl and attack them. These test sites are relatively small; in any site that cannot be completely crawled manually, testers should be wary of relying exclusively on Acunetix given the weakness of its crawler.

# Glossary

* **Crawler**: A crawler is a program that visits Web sites and reads their pages and other information to create entries for a [search engine](http://searchmicroservices.techtarget.com/definition/search-engine) index. The major search engines on the Web all have such a program, which is also known as a "spider" or a "bot." Crawlers are typically programmed to visit sites that have been submitted by their owners as new or updated. Entire sites or specific pages can be selectively visited and indexed. Crawlers apparently gained the name because they crawl through a site a page at a time, following the links to other pages on the site until all pages have been read.
* **CSRF:** Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated. CSRF attacks specifically target state-changing requests, not theft of data, since the attacker has no way to see the response to the forged request.
* **NTLM Authentication:** In a Windows network, NT LAN Manager (NTLM) is a suite of Microsoft security protocols that provides authentication.
* **ADoS**: Application denial of service is an attack technique with the intent of preventing a web site from serving normal user activity.
* **XSS**: Cross-site scripting (XSS) is a type of injection security attack in which an attacker injects data, such as a malicious script, into content from otherwise trusted websites.
* **Digest Authentication**: Digest authentication is a method of authentication in which a request from a potential user is received by a network server and then sent to a domain controller. The domain controller sends a special key, called digest session key, to the server that received the original request.

# References:

* [WASC Threat Classification v2.0](http://projects.webappsec.org/w/page/13246978/Threat%20Classification)
* [Oracle Blog - Five most popular penetration testing tools](https://blogs.oracle.com/ravello/five-most-popular-penetration-testing-tools)
* [Powerful Security Testing Tools](http://www.softwaretestinghelp.com/penetration-testing-tools/)
* [OWASP - Vulnerability Scanning Tools](https://www.owasp.org/index.php/Category:Vulnerability_Scanning_Tools)
* [Web Application Security Scanner Evaluation criteria](http://projects.webappsec.org/w/page/13246986/Web%20Application%20Security%20Scanner%20Evaluation%20Criteria)
* [Penetration testing tools that will do the job](https://www.networkworld.com/article/2944811/network-security/8-penetration-testing-tools-that-will-do-the-job.html)
* [Top5 Vulnerability scanners](https://academy.ehacking.net/blog/56540/top-best-web-vulnerability-scanners)
* [Feature comparison of web application scanners](http://www.sectoolmarket.com/price-and-feature-comparison-of-web-application-scanners-unified-list.html)
* [OWASP Comparison](https://www.owasp.org/images/2/2c/OWASP-NL_Chapter_Meeting_2014-04_-24_dont_be_a_tools_fool.pdf)