Performance Evaluation of Automated Web vulnerability scanners for cross platforms -Red Teaming

MSc Research Project

MSs cyber security

Rohan Anand Gowda

Student ID: X21178003

School of Computing

National College of Ireland

Supervisor: Mr. Nail Heffernan

National College of Ireland

Project Submission Sheet – 2022/2023

Student Name:	Rohan Anand Gowda		
Student ID:	X21178003		
Programme:	MSCCYB1	**Year:**	1
Module:	MSc. Research project		
Lecturer:			
Submission Due Date:	18/09/2023		
Project Title:	Performance Evaluation of Automated Web vulnerability scanners for cross platforms -Red Teaming		
Word Count:	1547 including references		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

<u>ALL</u> internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature: Rohan Anand Gowda

Date: 18/09/2023

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Rohan Anand Gowda
Student ID: x21178003

1. Introduction

We require software and hardware system configurations in order to put into practice and illustrate the framework described in the research thesis. The chapters that follow will cover this arrangement in detail. Choosing a base machine with sufficient capabilities and installing a VMware player on it are the first steps in system configuration. Once the system is configured we proceed with the installation of the tools namely HCL Appscan, Netsparker, Burpsuite and Nikto.

2. System Configuration

System 1:
Base machine: Windows 11
Processor: Quad core processor
Memory: 16 GB
System type: 64 bit operating system
HDD: 200 GB of free space

System 2:
Virtual Machine: Kali Linux 2023.2
Processor: Two processors
Memory: 4 GB
System type: 64 bit
HDD: 10 GB of free space

3. Tools installation

This section deals with the installation and requirements of our experiment.

3.1. OWASP Benchmark Project:

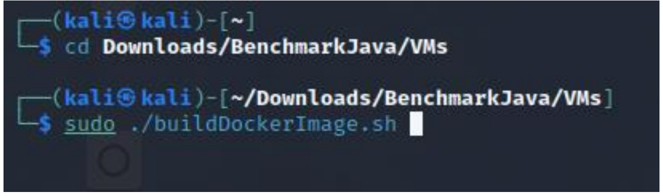
The application is installed from github: https://github.com/OWASP-Benchmark/BenchmarkJava

For windows it can be installed in the form of a zip fule and for linux it can be installed by using the command git clone

Step 1: Download the git file, unzip it, and then use the command line to go to the project folder.

• Move to /root/Downloads/Benchmark/VMs.

Step 2: Execute "BuildDockerImage.sh" located in the VMs folder. The project's docker will now be generated.
Step 3: Execute the 'runDockerImage.sh' file after the Docker image has been successfully built.
By doing so, the project and application will be launched.
Step 4: To access the Benchmark web application, open a web browser and type the following URL.
Benchmarking URL: https://127.0.0.1:8443/benchmark
Note: Running benchmark on windows requires maven to be installed to compile the entire application. On linux docker is used as it runs all the dependencies required by the application making it run smoothly.

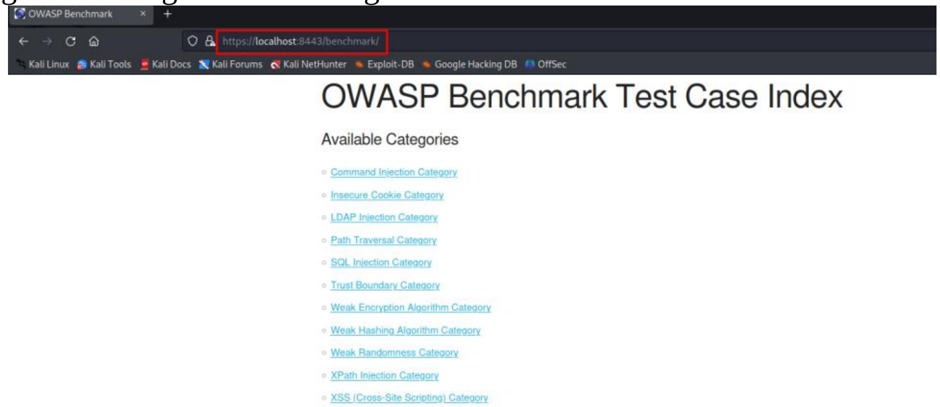


```
┌──(kali㉿kali)-[~]
└─$ cd Downloads/BenchmarkJava/VMs

┌──(kali㉿kali)-[~/Downloads/BenchmarkJava/VMs]
└─$ sudo ./buildDockerImage.sh
```

Fig 3,1: building the benchmark

On accessing the url we get the following interface:



OWASP Benchmark Test Case Index

Available Categories

- Command Injection Category
- Insecure Cookie Category
- LDAP Injection Category
- Path Traversal Category
- SQL Injection Category
- Trust Boundary Category
- Weak Encryption Algorithm Category
- Weak Hashing Algorithm Category
- Weak Randomness Category
- XPath Injection Category
- XSS (Cross-Site Scripting) Category

Figure 3.2 OWASP benchmark.

3.2.OWASP Juiceshop:

This is a real time application hosted on a server and hence we can navigate to the page b y accessing the url: https://juice-shop.herokuapp.com/#/

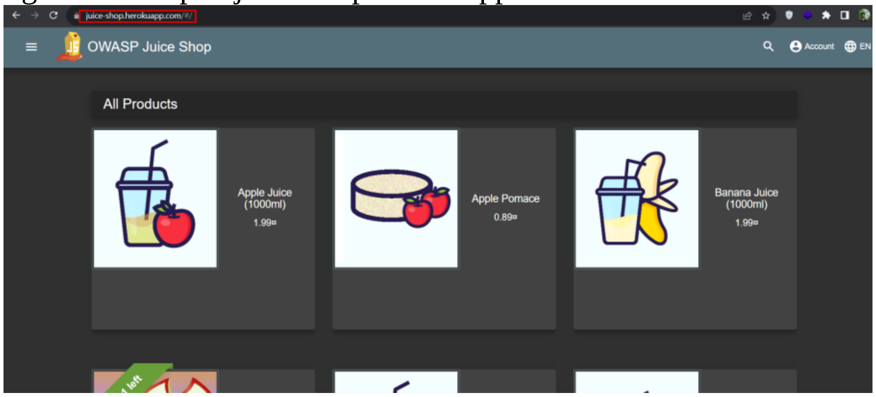


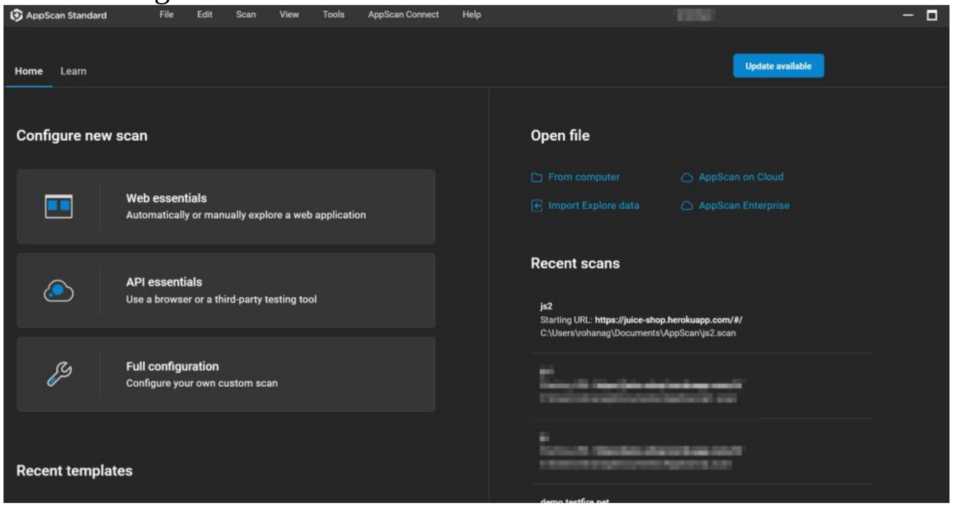
Figure 3.3 OWASP juice shop

3.3.HCL Appscan

This is a windows tool and can be installed by using the installer package on a gui interface. The procedure is as follows:

- Before installing this version, uninstall any subsequent versions of AppScan® Standard that may be on your computer.
- The package is isnstalled from https://www.hcltech.com/brochures/software/hcl-appscan-standard
- Any open Microsoft® Office programs should be closed.
- Open up AppScan setup.
- As soon as it launches, the InstallShield Wizard verifies that your workstation satisfies the minimal installation requirements. The welcome screen for the AppScan installation wizard then appears.

To finish installing AppScan, adhere to the wizard's instructions. Once done and opened we see the following interface:



Configure new scan

Web essentials
Automatically or manually explore a web application

API essentials
Use a browser or a third-party testing tool

Full configuration
Configure your own custom scan

Recent templates

Open file

Recent scans

js2
Starting URL: https://juice-shop.herokuapp.com/#/
C:\Users\rohanag\Documents\AppScan\js2.scan

[redacted]

[redacted]

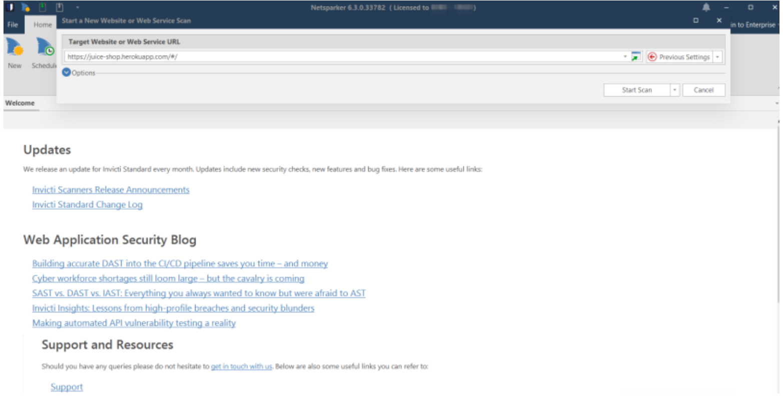
demo.testfire.net

Figure 3.4 HCL APPSCAN interface

3.4.Netsparker

The procedure is similar to that of appscan where the tool is downloaded from https://www.invicti.com/support/installing-invicti-standard/.

Once installed we ge the following interface:



Netsparker 6.3.0.33782 (Licensed to [redacted])

Start a New Website or Web Service Scan

Target Website or Web Service URL
https://juice-shop.herokuapp.com/#/

Updates

We release an update for Invicti Standard every month. Updates include new security checks, new features and bug fixes. Here are some useful links:

Invicti Scanners Release Announcements
Invicti Standard Change Log

Web Application Security Blog

Building accurate DAST into the CI/CD pipeline saves you time – and money
Cyber workforce shortages still loom large – but the cavalry is coming
SAST vs. DAST vs. IAST: Everything you always wanted to know but were afraid to AST
Invicti Insights: Lessons from high-profile breaches and security blunders
Making automated API vulnerability testing a reality

Support and Resources

Should you have any queries please do not hesitate to get in touch with us. Below are also some useful links you can refer to:

Support

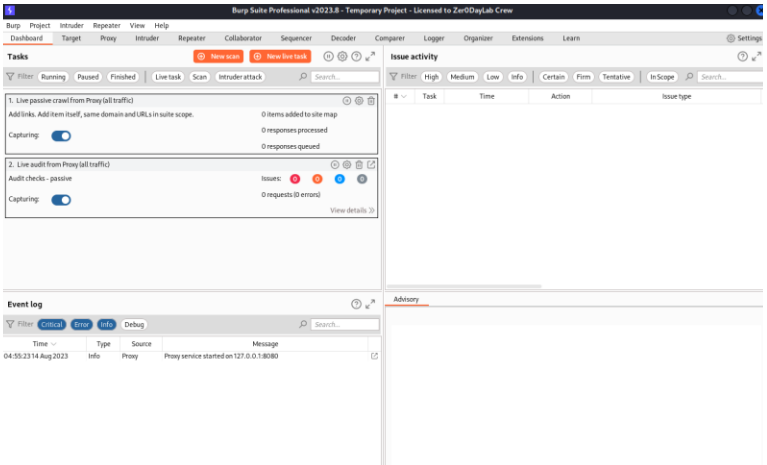
Fig 3.5 Netsparker interface

3.5.Burpsuite

Burpsuite packages are installed from https://portswigger.net/burp/documentation/desktop/getting-started/download-and-install.

- Step1: Use the aforementioned link to select your program.
- Step 2: Open Burp Suite after launching the installer. To avoid this for the time being, simply click Next and then Start Burp when prompted to choose a project file and configuration. Note: Enter your license key when prompted if you're using Burp Suite Professional. You can subscribe or ask for a trial if you don't already have one.
- Step 3: Investigate Burp Suite. If you're brand-new to Burp Suite, continue reading for an interactive, guided tour of the essential functions.

To run the tool we use the command: java -jar run burploader.jar. once opened we see he following:


Burp Suite Professional v2023.8 - Temporary Project - Licensed to Zer0DayLab Crew

Tasks

1. Live passive crawl from Proxy (all traffic)
Add links. Add item itself, same domain and URLs in suite scope.
Capturing:
0 items added to site map
0 responses processed
0 responses queued

2. Live audit from Proxy (all traffic)
Audit checks - passive
Capturing:
Issues: 0 0 0 0
0 requests (0 errors)

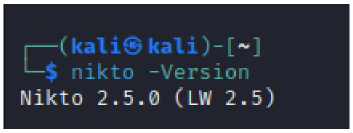
Event log

Time	Type	Source	Message
04:55:23 14 Aug 2023	Info	Proxy	Proxy service started on 127.0.0.1:8080

Figure 3.6 Bursuite Professonal

3.6. Nikto

Nikto come sinbuilt with Kali linux. If not the we can use the command sudo ap install nikto2. We can see the version below:



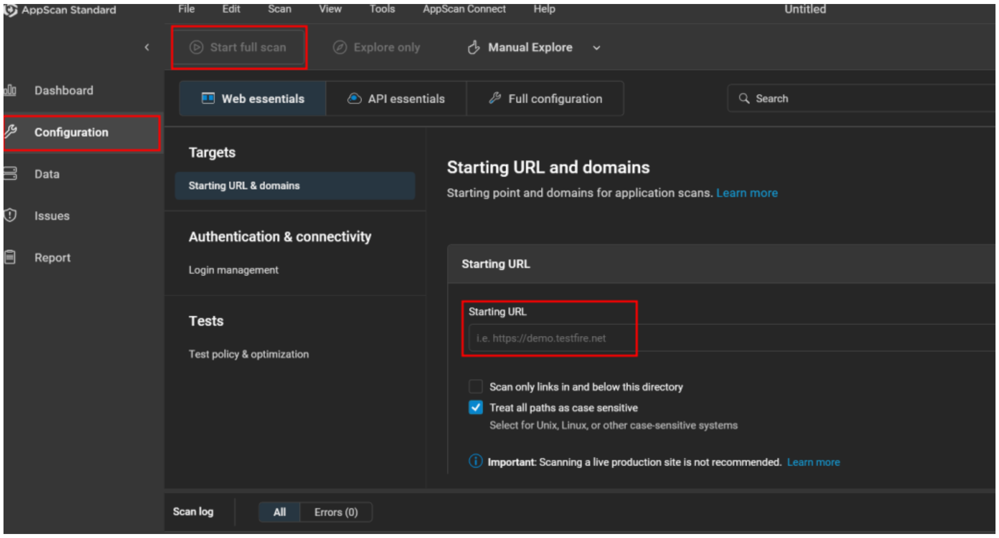
```
┌──(kali㉿kali)-[~]
└─$ nikto -Version
Nikto 2.5.0 (LW 2.5)
```

Figure 3.7 Nikto on cli

4. Tools configuration

4.1.HCL apscan

Open the application and navigate to file -> new -> web application scan:. Once created enter the target url in the scan field and click on start scan in the configuration tab.



Starting URL and domains

Starting point and domains for application scans. Learn more

Starting URL: i.e. https://demo.testfire.net

☐ Scan only links in and below this directory

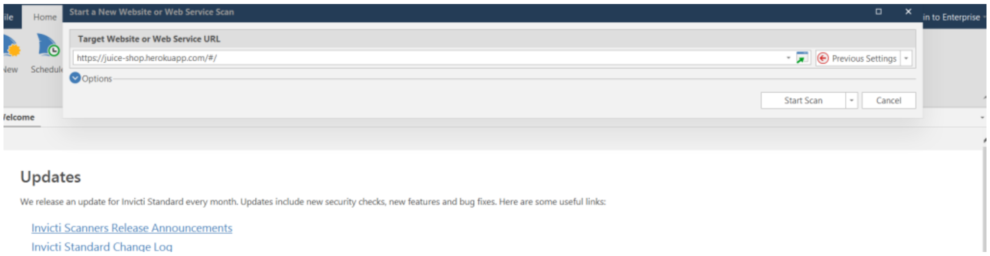
☑ Treat all paths as case sensitive
Select for Unix, Linux, or other case-sensitive systems

Important: Scanning a live production site is not recommended. Learn more

Fig 4.1 HCL appscan configuration

4.2.Netsparker

The process is simple where the targer url is entered in the san filed as shown below:



Target Website or Web Service URL: https://juice-shop.herokuapp.com/#/

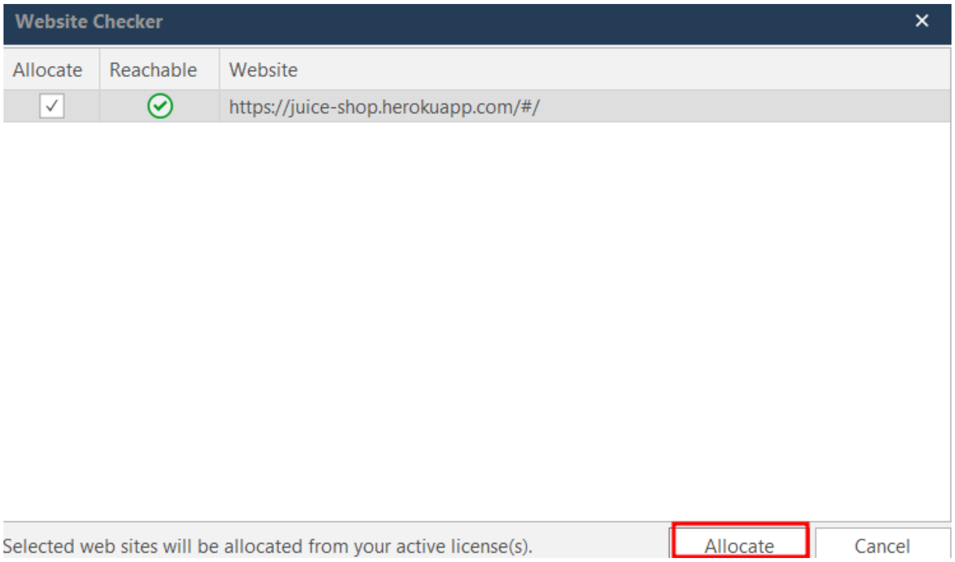
Updates

We release an update for Invicti Standard every month. Updates include new security checks, new features and bug fixes. Here are some useful links:

Invicti Scanners Release Announcements

Invicti Standard Change Log

Figure 4.2 Netsparker configuration tab



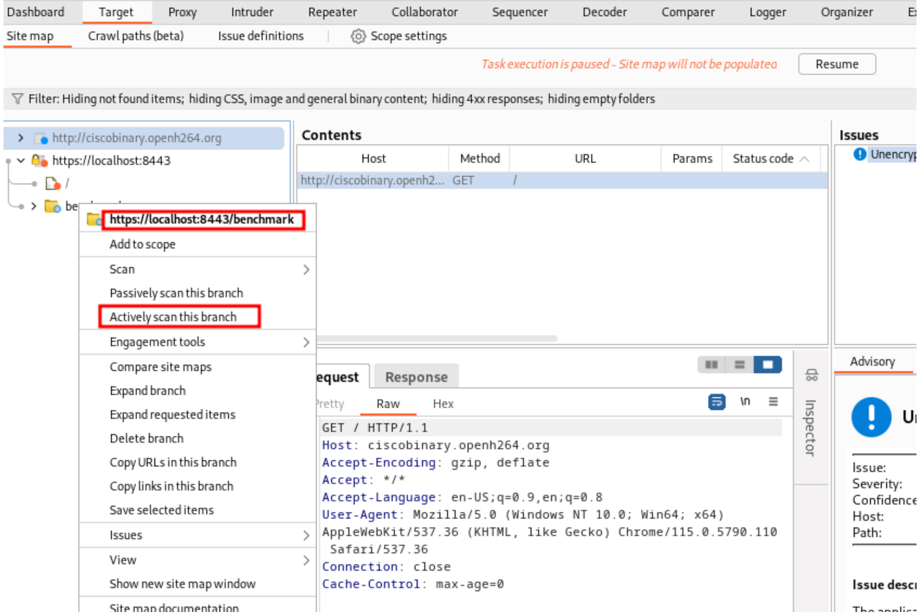
Allocate	Reachable	Website
☑	✓	https://juice-shop.herokuapp.com/#/

Selected web sites will be allocated from your active license(s).

Fig 4.3 Allocating scan

4.3.Burpsuite professional

Crawl the entire Benchmark first before scanning. Right-click Benchmark in the Site Map and choose Scan->Open scan launcher to do a crawl. Then select Crawl and press OK. Then, save the project in case the scan crashes. Choose the /Benchmark URL and say "Actively scan this branch" after that. Prior to all of this, you might want to open Burp using the following command: java -Xmx2G -jar burpsuite_pro.jar.



```
GET / HTTP/1.1
Host: ciscobinary.openh264.org
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0.5790.110
 Safari/537.36
Connection: close
Cache-Control: max-age=0
```

Fig 4.4Burpsuite scan

4.4.Nikto

- Step 1: To start the host scan, enter the following command into the terminal: Benchmarks -format nikto -host https://127.0.0.1:8443 xml
- Step 2: Use the following command to copy the created xml report to the Benchmarks results folder.
- Step 3: Copy the results to the root directory under "./format.xml". The tool can be configured in the following ways for configured scans

4.5.Owasp benchmark score generation

To start the host scan, enter the following command into the terminal:

- Benchmarks nikto -host https://127.0.0.1:8443 -format xml
- Use the following command to copy the created xml report to the Benchmarks results folder .Copy the results to the root directory under "./format.xml"
- For configured, also scanGather all scan outputs in.xml format and copy them to the benchmark results folder per Step 1.
- ./Downloads/Benchmark/results, the results folder
- Run the 'createScorecard.sh' file located in the Benchmark folder in step 2.
- •Create Scorecard.sh
- All of the reports in the "reports" folder will generate benchmark scores and graphical representations, which will be stored in the "scorecard" folder.

5. References

Albahar, M., Alansari, D. and Jurcut, A. (2022) ‘An Empirical Comparison of Pen-Testing Tools for Detecting Web App Vulnerabilities’, *Electronics*, 11(19), p. 2991. Available at: https://doi.org/10.3390/electronics11192991.

Antunes, N. and Vieira, M. (2015) ‘On the Metrics for Benchmarking Vulnerability Detection Tools’, in *2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. IEEE, pp. 505–516. Available at: https://doi.org/10.1109/DSN.2015.30.

Casola, V. *et al.* (2018) ‘Towards Automated Penetration Testing for Cloud Applications’, in *2018 IEEE 27th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. IEEE, pp. 24–29. Available at: https://doi.org/10.1109/WETICE.2018.00012.

Chad Kime (2023) *The 8 Best Vulnerability Scanner Tools for 2023*. Available at: https://www.esecurityplanet.com/networks/vulnerability-scanning-tools/#invicti (Accessed: 16 April 2023).

Chanchala Joshi and Umesh Kumar Singh (no date) ‘Performance Evaluation of Web Application Security Scanners for More Effective Defense’, *International Journal of Scientific and Research* [Preprint], (2016). Available at: https://d1wqtxts1xzle7.cloudfront.net/47370836/ijsrp-p5490-libre.pdf?1468998289=&response-content-disposition=inline%3B+filename%3DPerformance_Evaluation_of_Web_Applicatio.pdf&Expires=1691166593&Signature=OtVU~bjZpaPhoLCaLnJ3UFbAAdjJa2qcAHw3RSQdRQFn6gWwhx2ffxnQOVeXwYNe1H2WnUJBlcFzalTLs2qD2Hn3Odx7ooNeKGo9cqbUZqRjMeEtC-LyE2rlcHa~0zzmhQYjOBmIgtolfTyIERI15thTuTtffTt5zn5r~OfzongRrm25pFYKOxlicwfJZAWRVmw4Genm5mxBz~MZwwM5VcyfpLIE2v7Wnsnhu2K-QI2Fsnoledv3t7ZM0S4-T32kfMG8Vjc~84qVMZcHRYUdsGFwBJ5J5cALPa81Os9zVidzuzKqOMxf1MCXlOWuuWFaaVTL-QR8BuTGJzxm6SAjVg__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA (Accessed: 4 August 2023).

Chuck Brooks (2023) *Cybersecurity Trends & Statistics For 2023; What You Need To Know*. Available at: https://www.forbes.com/sites/forbesdigitalcovers/2018/07/19/the-inside-story-of-papa-johns-toxic-culture/? (Accessed: 16 April 2023).

Díaz, G. and Bermejo, J.R. (2013) ‘Static analysis of source code security: Assessment of tools against SAMATE tests’, *Information and Software Technology*, 55(8), pp. 1462–1476. Available at: https://doi.org/10.1016/j.infsof.2013.02.005.

Fortra (2022) ‘2022 Penetration Testing Report’, pp. 4–4. Available at: https://static.fortra.com/core-security/pdfs/guides/cs-2022-pen-testing-report.pdf (Accessed: 16 April 2023).

Francesc Mateo Tudela *et al.* (2020) ‘On Combining Static, Dynamic and Interactive Analysis Security Testing Tools to Improve OWASP Top Ten Security Vulnerability Detection in Web Applications’, *MDPI* [Preprint]. Available at: https://www.mdpi.com/2076-3417/10/24/9119 (Accessed: 9 August 2023).

HCL Technologies (2023) ‘HCL Appscan’. Available at: https://www.hcltech.com/brochures/software/hcl-appscan-standard#:~:text=HCLTech%20AppScan%20Standard%20is%20a,web%20applications%20and%20web%20services. (Accessed: 11 August 2023).

InterviewBit (2023) *VMware vs VirtualBox: What's The Difference?* Available at: https://www.interviewbit.com/blog/vmware-vs-virtualbox/ (Accessed: 11 August 2023).

Invicti (2023a) *Accunetix, 2023*. Available at: https://www.acunetix.com/ (Accessed: 16 April 2023).

Invicti (2023b) *False Positive rate detection* **-** *Netsparker*. Available at: https://www.invicti.com/blog/news/comparison-web-vulnerability-scanners-netsparker-2013-2014/#:~:text=False%20Positives%20and%20Web%20Security%20Scans%20Time%20Consumption&text=Funnily%20enough%20Netsparker%2C%20the%20only,false%20positive%20SQL%20Injection%20vulnerabilities. (Accessed: 14 August 2023).

Invicti (2023c) 'Netsparker'. Available at: https://www.invicti.com/web-vulnerability-scanner/ (Accessed: 10 August 2023).

Josephine S Akosa (2017) 'Predictive Accuracy: A Misleading Performance Measure for Highly Imbalanced Data '. Available at: https://support.sas.com/resources/papers/proceedings17/0942-2017.pdf (Accessed: 12 August 2023).

Mansour Alsaleh, Noura Alomar and Monirah Alshreef (2017) 'Performance-Based Comparative Assessment of Open Source Web Vulnerability Scanners'. Available at: https://www.hindawi.com/journals/scn/2017/6158107/ (Accessed: 16 April 2023).

Mburano, B. and Si, W. (2018) 'Evaluation of Web Vulnerability Scanners Based on OWASP Benchmark', in *2018 26th International Conference on Systems Engineering (ICSEng)*. IEEE, pp. 1–6. Available at: https://doi.org/10.1109/ICSENG.2018.8638176.

Michael Massoth, Saed Alavi and Niklas Bessler (2018) 'A Comparative Evaluation of Automated Vulnerability Scans versus Manual Penetration Tests on False-negative Errors'. Available at: http://personales.upv.es/thinkmind/dl/conferences/cyber/cyber_2018/cyber_2018_1_10_80034.pdf (Accessed: 16 April 2023).

Mohd. Ehmer Khan and Farmeena Khan (2012) 'A Comparative Study of White Box, Black Box and Grey Box Testing Techniques', *(IJACSA) International Journal of Advanced Computer Science and Applications*, 3. Available at: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ab0ed151c010e03e2d34284ae5f89756372e7690#page=22 (Accessed: 16 April 2023).

Nagpure, S. and Kurkure, S. (2017) 'Vulnerability Assessment and Penetration Testing of Web Application', in *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*. IEEE, pp. 1–6. Available at: https://doi.org/10.1109/ICCUBEA.2017.8463920.

'Nikto 2' (2023). Available at: https://cirt.net/Nikto2 (Accessed: 11 August 2023).

OWASP (2023a) *Owasp interpretation guide*. Available at: https://owasp.org/www-project-benchmark/ (Accessed: 12 August 2023).

OWASP (2023b) *OWASP Web Security Testing Guide*. Available at: https://owasp.org/www-project-web-security-testing-guide/ (Accessed: 16 April 2023).

Portswigger (2023) *Burpsuite*. Available at: https://portswigger.net/burp (Accessed: 16 April 2023).

Stefinko, Y., Piskozub, A. and Banakh, R. (2016) 'Manual and automated penetration testing. Benefits and drawbacks. Modern tendency', in *2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET)*. IEEE, pp. 488–491. Available at: https://doi.org/10.1109/TCSET.2016.7452095.

Xiaohong Yuan *et al.* (2011) 'An Overview of Penetration Testing'. Available at: https://www.researchgate.net/publication/274174058_An_Overview_of_Penetration_Testing (Accessed: 16 April 2023).