National College of Ireland

Configuration Manual

MSc Internship
MSc in Cyber Security

Lyubka Dencheva
Student ID: x20195907

School of Computing
National College of Ireland

Supervisor: Mark Monaghan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Lyubka Dencheva
Student ID:	x20195907
Programme:	MSc in Cyber Security **Year:** 2022
Module:	Internship
Supervisor:	Mark Monaghan
Submission Due Date:	15/08/2022
Project Title:	Comparative analysis of Static application security testing (SAST) and Dynamic application security testing (DAST) by using open-source web application penetration testing tools

Word Count: 1391 **Page Count: 11**

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.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Lyubka Dencheva

Date: 15th August, 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

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

Configuration Manual

Lyubka Dencheva
Student ID: x20195907

1 Introduction

The configuration manual describes the methodology and execution framework presented in the master's thesis, providing information on the installation, configuration, and execution of the necessary tools for the purposes of a research paper.
System configuration begins with the installation and configuration of a Kali Linux Virtual Machine to provide high functionality and flexibility to perform the intended tests. After its successful launch, the installation and configuration of the used Static Application Security Testing (SAST) and Dynamic Application Security Testing (DAST) tools is performed, followed by the actual execution of the security vulnerability scan and the generation of its results. Subsequently, they are used by the selected Benchmark to prepare scorecards with summarized, systematized, and accurate results. Based on them, the comparative analysis between open-source SAST and DAST tools is performed, which is the focus of the master's thesis.

2 System configuration

1) Host machine

Operation System (OS): Microsoft Windows 10 Enterprise
System type: x64 – based PC
Processor: Intel(R) Core(TM) i5-4210M CPU @2.6GHz, 2601Mhz, 2 Core(s), 4 Logical Processor(s)
Memory: 12GB

2) Virtual machine

Operation System (OS): Kali Linux version 5.9
System type: 64-bit operating system
Processor: 4 processor cores
Memory: 7451 MB
Video memory: 16 MB

2.1 Installation and configuration of the Virtual Machine Kali Linux

In order to install and use a Virtual Machine, a platform is needed to run it. For this purpose, the open source software Oracle VM VirtualBox Manager version 6.1 is used, downloaded from its official site[1], which is installed on the Windows OS host machine and the Operation System Kali Linux is configured on it. Regarding this configuration, Jon Watson's research

[1] Oracle VM VirtualBox: https://www.virtualbox.org/

paper[1] explains in great detail, step by step, how the entire process is carried out. The end result is to start the virtual machine as shown in Figure 1.

Figure 1: Virtual Machine Kali Linux on Oracle VM VirtualBox Manager

3 Environment Setup

For the purposes of the master's thesis, in addition to a Kali Linux virtual machine, Static Application Security Testing (SAST) and Dynamic Application Security Testing (DAST) tools are needed, the productivity and efficiency of which are being investigated. Based on them and a specially selected Benchmark, a qualitative, detailed and accurate comparative analysis is performed between the these two types of security testing. The overall process to achieve it through specialized, installations and configuration is presented in the following subsections.

3.1 Installation and configuration of Benchmark

The implementation process begins with downloading the open-source OWASP Benchmark zip files from the GitHub platform[2] and placing it on the Virtual Machine. Next, unzipping the file with the unzip command and accessing the VMs directory, via the following command:
cd root/Desktop/BenchmarkJava-master/VMs

Then, it is necessary to build and run the docker, which is done by successively running the files BuildDockerImage.sh and runDockerImage.sh with the bash command (see Figure 2).

[2] OWASP Benchmark Java Project: https://github.com/OWASP-Benchmark/BenchmarkJava

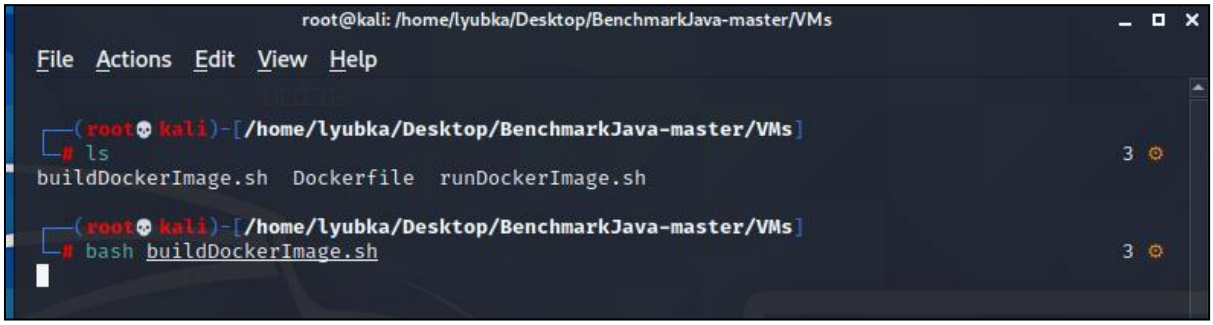

Figure 2: Running Docker scripts

After the above commands are executed, OWASP Benchmark is started and can be accessed via the web browser at the localhost URL on port 8443 as follows (see Figure 3): https://localhost:8443/benchmark

At the above URL, a benchmark has been positioned and loaded and it is now ready to start scanning for security vulnerabilities, as shown in Figure 3:

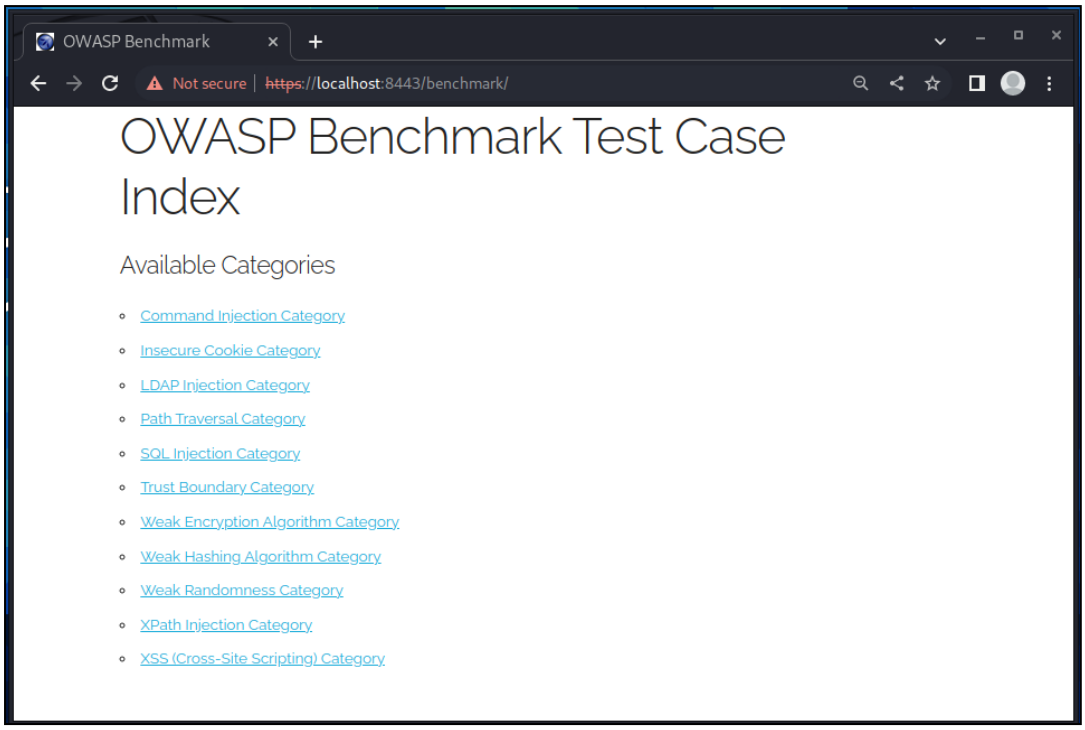

Figure 3: OWASP Benchmark Java Project

3.2 Installation and configuration of SAST tools

3.2.1 LGTM.COM

LGTM.COM is an online tool that is a code analysis platform to identify vulnerabilities and prevent them from reaching production. It requires no installation and is accessed through the web browser on its official website[3]. In this tool, there is an extremely large number of already loaded open-source projects, among which is OWASP Benchmark (see Figure 4).

[3] LGTM.COM: https://lgtm.com/

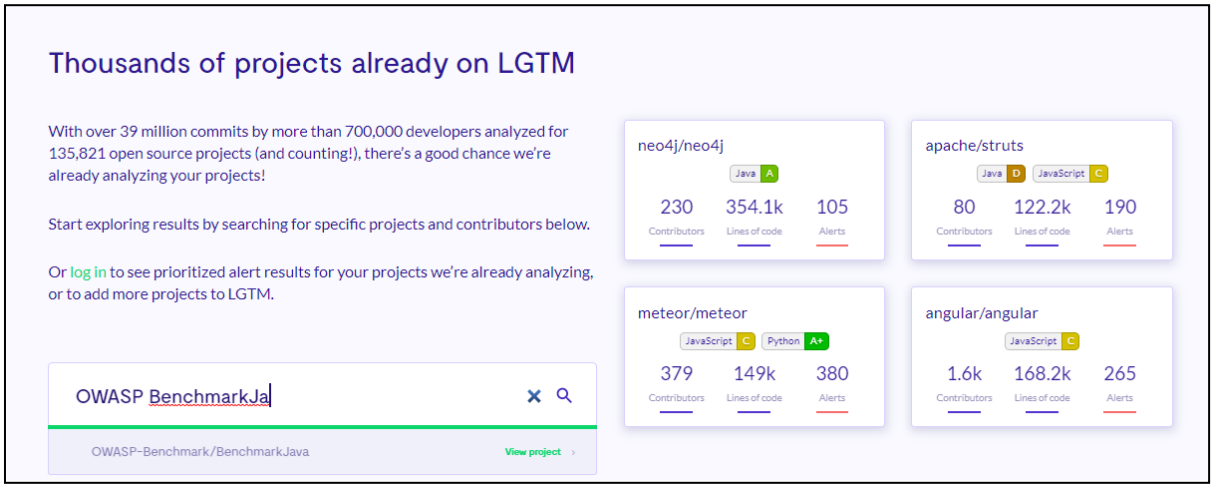

Figure 4: LGTM.COM

3.2.2 Visual Code Grepper (VCG)

The open-source Visual Code Grepper (VCG) scanning tool is installed on the Windows host machine for faster and more convenient scanning. The installation files of the tool are downloaded from the GitHub platform[4] and the installer is launched to begin the installation. Once completed, VCG is launched by clicking on the icon that is generated after successful installation.

3.3 Installation and configuration DAST tools

3.3.1 Arachni

The files provided on the official site[5] are used to install the Arachni open-source tool. After downloading, they are positioned on the Linux machine, with the aim of higher speed and flexibility. Then, from the Linux terminal, set the tool directory and specifically the bin directory as follows:

cd root/Desktop/Arachni/bin

After that, the command to start arachni_web is set (see Figure 5), after which the tool is ready to perform security vulnerability scans. It is accessed through the web browser as the URL of the localhost with port 9292 (see Figure 6):

http://localhost:9292

Then, it is necessary to enter the basic data for access which are:

Email – admin@admin.admin
Password – administrator

[4] Visual Code Grepper (VCG): https://github.com/nccgroup/VCG
[5] Arachni: https://www.arachni-scanner.com/download/

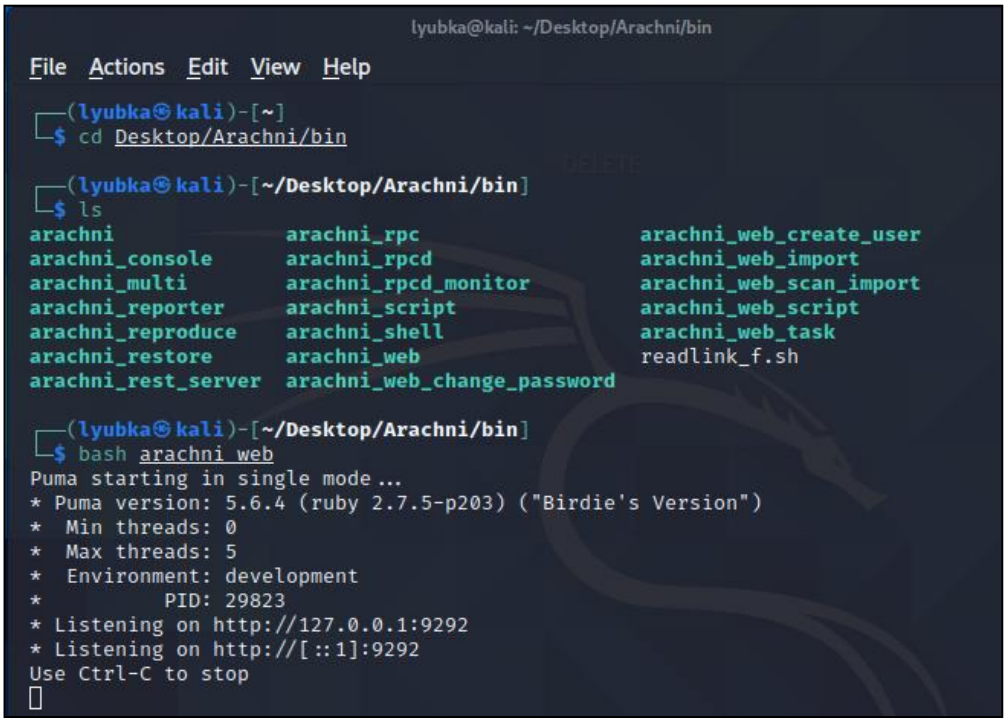

Figure 5: Starting of arachni_web

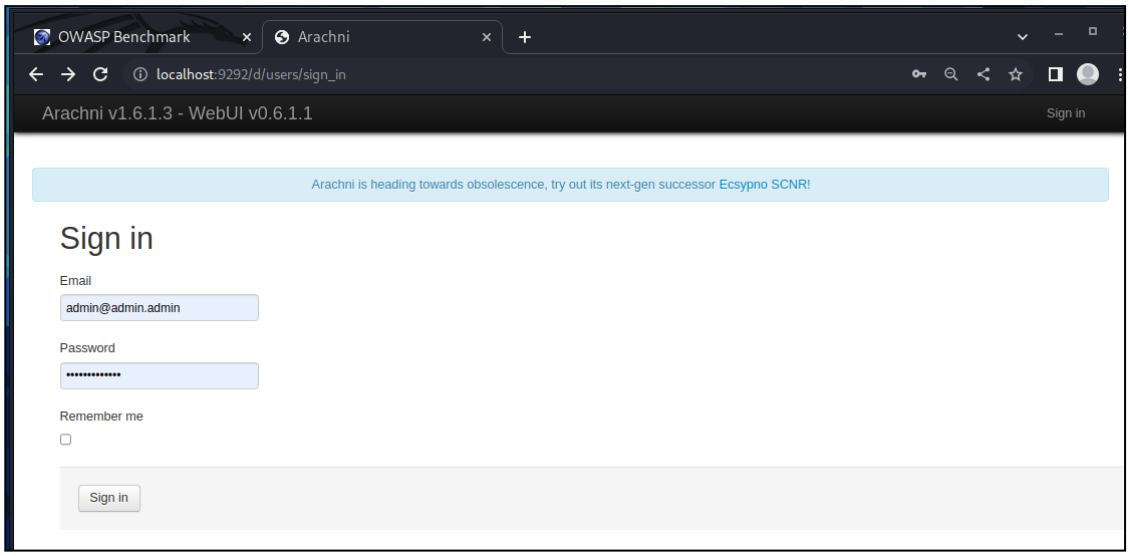

Figure 6: Aranchi

After the successful entry into the system, the admin panel of the Arachni is visualized, from where the configuration and management of the tool is carried out.

3.3.2 OWASP ZAP

The installation files for the open-source tool OWASP ZAP can be downloaded from the official website[6]. Then, they are loaded through the Linux terminal by running the zap.sh file as shown in Figure 7. This launches OWASP ZAP and the tool is ready to perform a security vulnerability scan.

[6] OWASP ZAP: https://www.zaproxy.org/docs/desktop/releases/2.9.0/

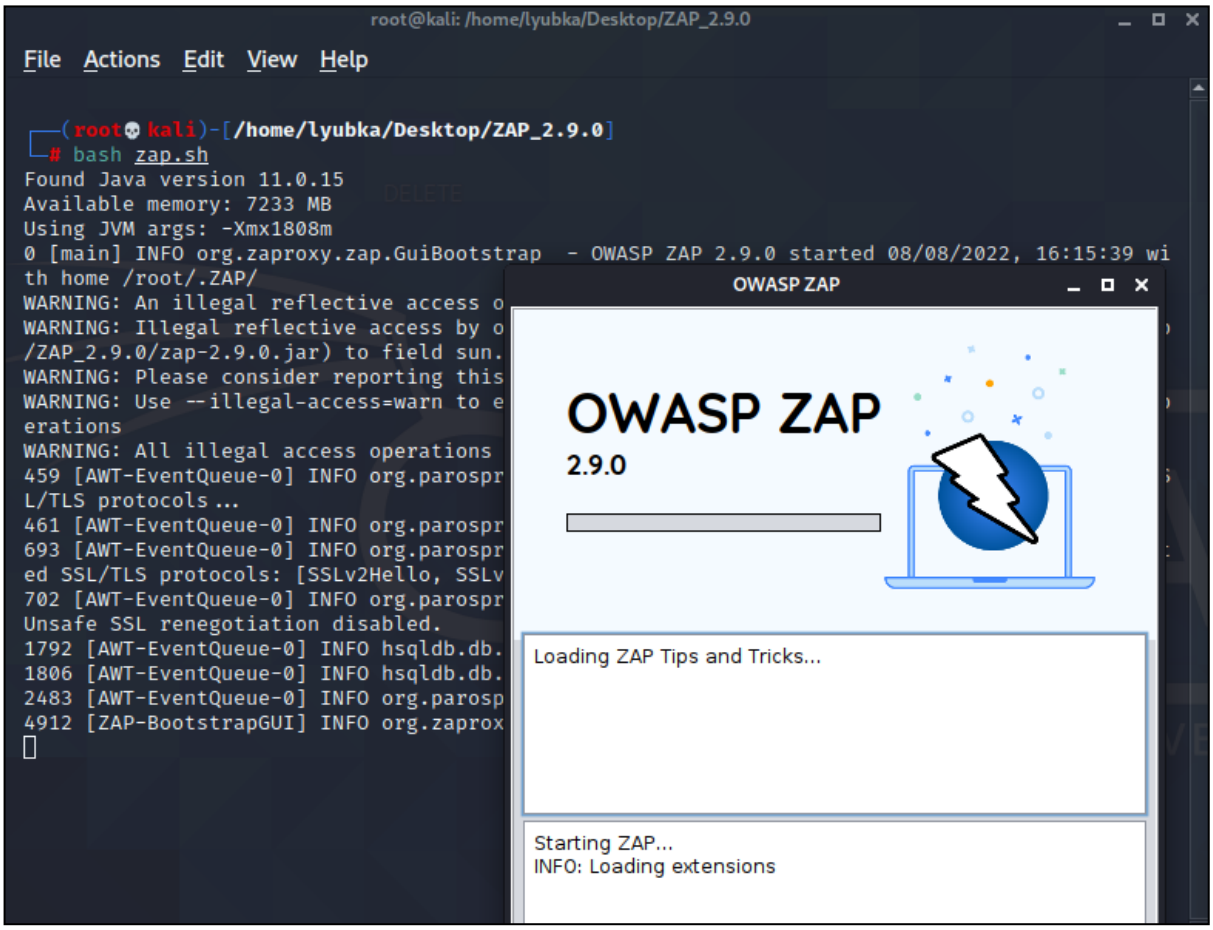

Figure 7: OWASP ZAP

4 Implementation and execution of used tools

4.1 SAST tools

4.1.1 LGTM.COM

As already established, the LGTM.COM tool contains an already analyzed OWASP Benchmark Project. Therefore, to obtain the results of the performed security vulnerability scan, it is only necessary to download the files from the scan, as shown in Figure 8:

Figure 8: LTGM.COM – export results

4.1.2 Visual Code Grepper (VCG)

There is an important feature when performing a security vulnerability scan with Visual Code Grepper (VCG). As soon as the tool is launched, it is necessary to select the programming language in which the scan will be performed. The test files and/or directories are then loaded, and the vulnerability scan is started, as shown in Figure 9:

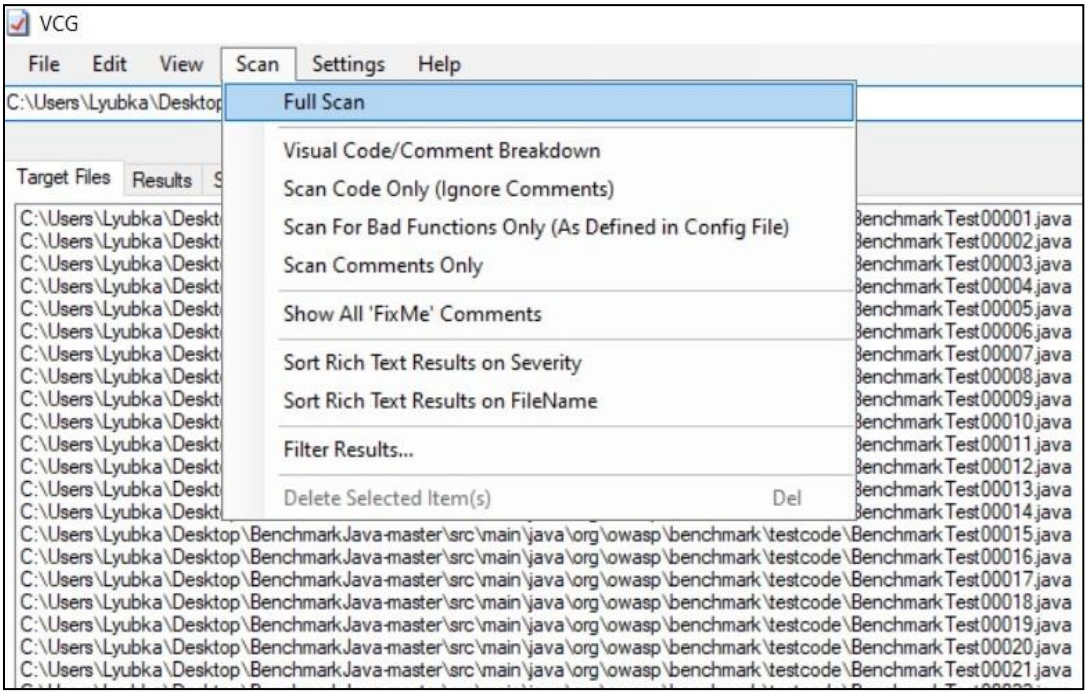

Figure 9: VCG Scan

After the scan is complete, the results can be exported in various formats, but for the current research work, xml format is used (see Figure 10).

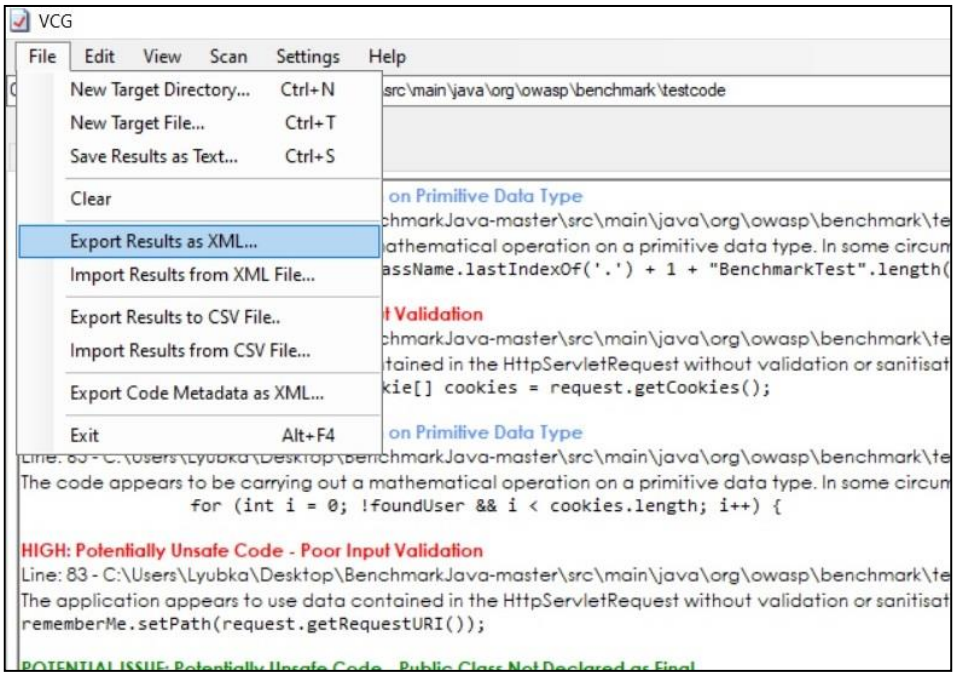

Figure 10: VCG – export results

4.2 DAST tools

4.2.1 Arachni

The start of the scan is set from the Arachni admin panel, which is accessible through the web browser, as already discussed. In the target field, the URL address of the OWASP Benchmark is placed and the scan is started, as shown in Figure 11:

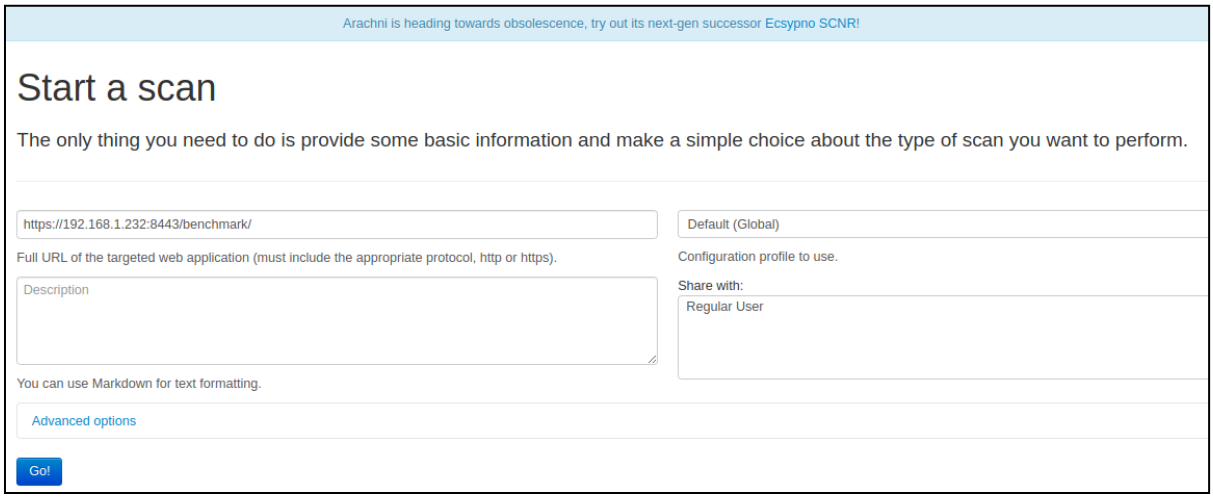

Figure 11: Arachni Scan

After completing the scan for security vulnerabilities, the tool presents an option to export the obtained results. They are used to perform further analysis (see Figure 12).

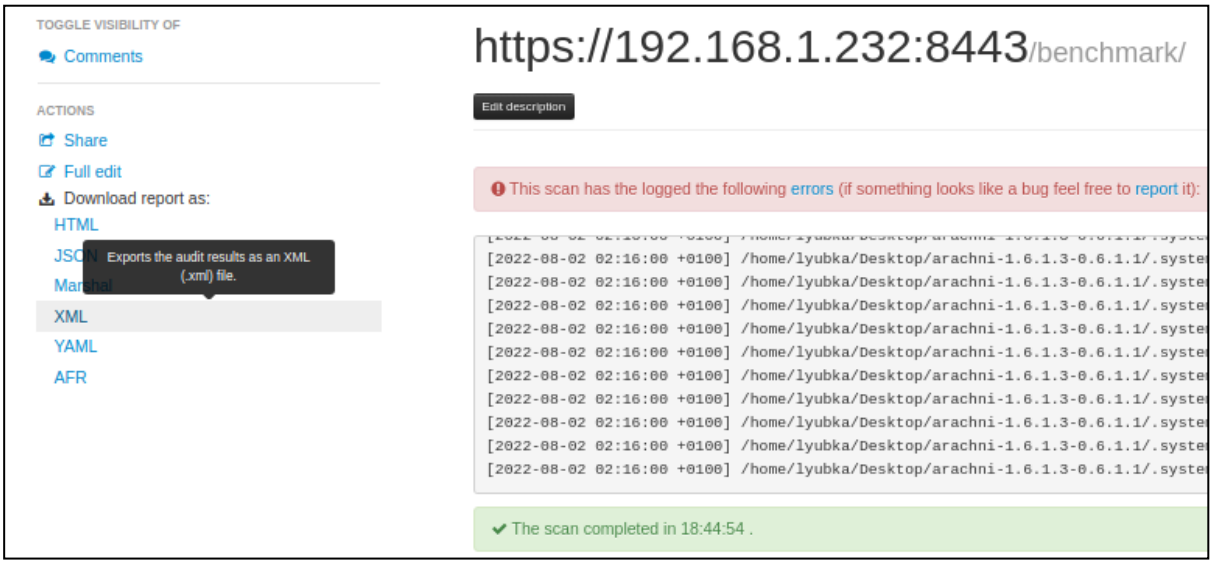

Figure 12: Arachni – export results

4.2.1 OWASP ZAP

Before starting the scan in OWASP ZAP, it is good to configure the scan setting by increasing the number of possible results (see Figure 13). Then, in the field for URL attack address, the address of OWASP Benchmark is set and the scan for security vulnerabilities is

started. This tool, as well as the ones discussed above, also provides an opportunity to export the results (see Figure 14) that will be needed for the research work.

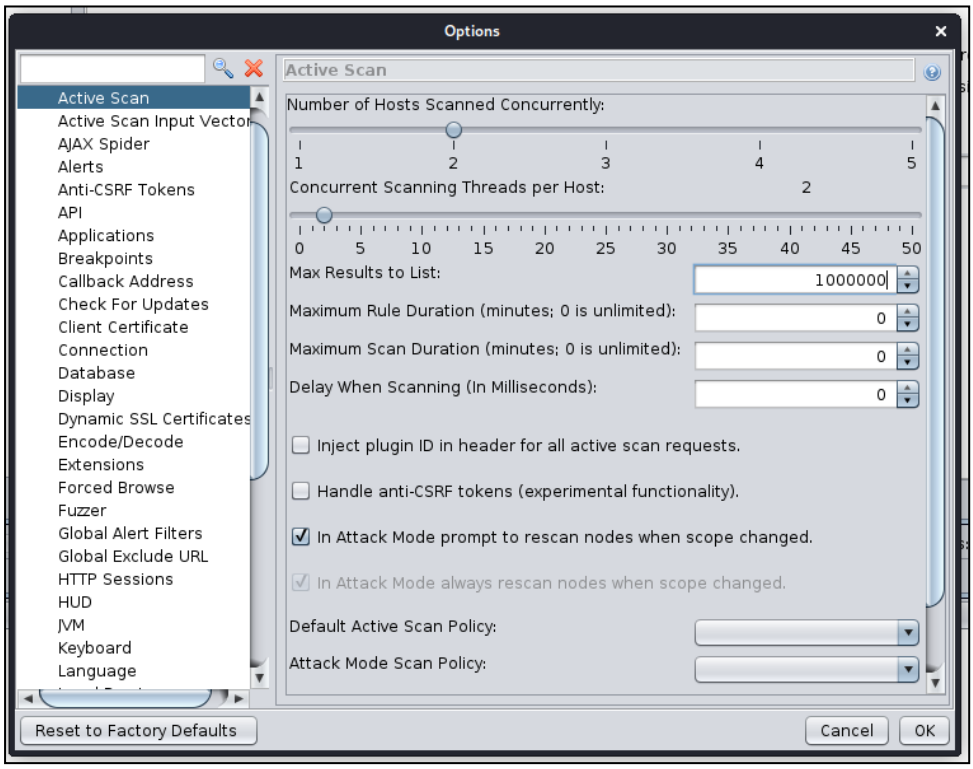

Figure 13: OWASP ZAP – configuration

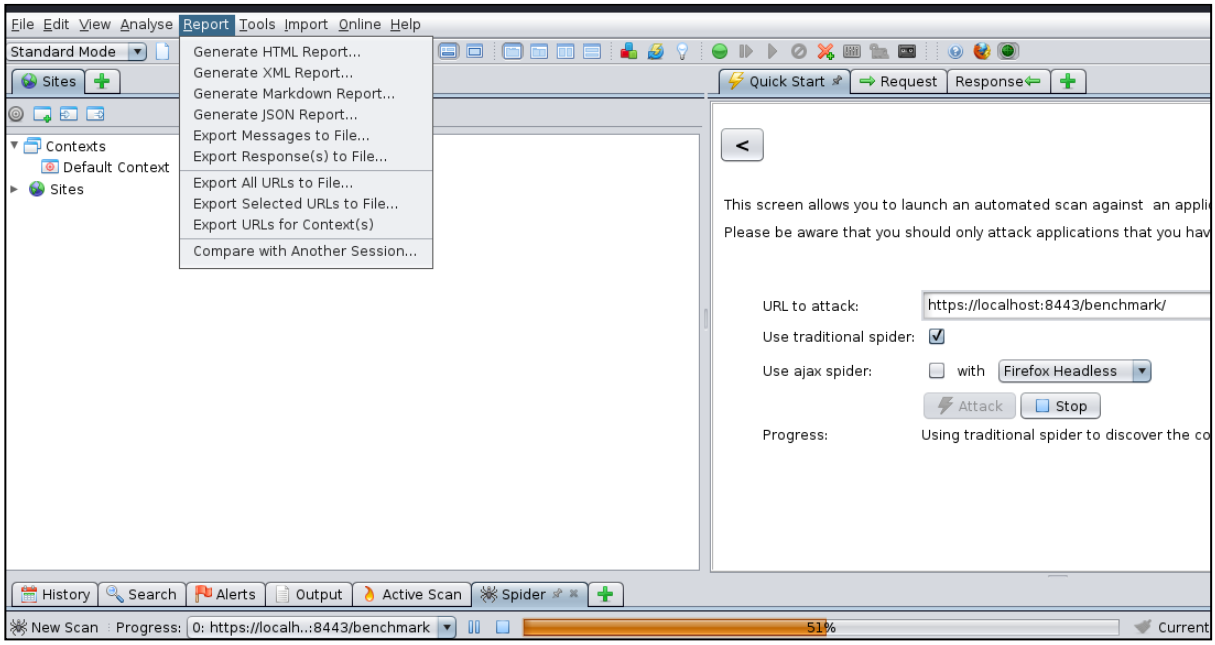

Figure 14: OWASP ZAP – export results

5. Generate a scorecard for analysis

After completing the scan for security vulnerabilities with the above-mentioned and discussed SAST and DAST tools, the results are exported as xml and serif formats, as they are accepted by selected Benchmark for further analysis. These files are placed in the OWASP Benchmark directory, in folder results, as shown in Figure 15:

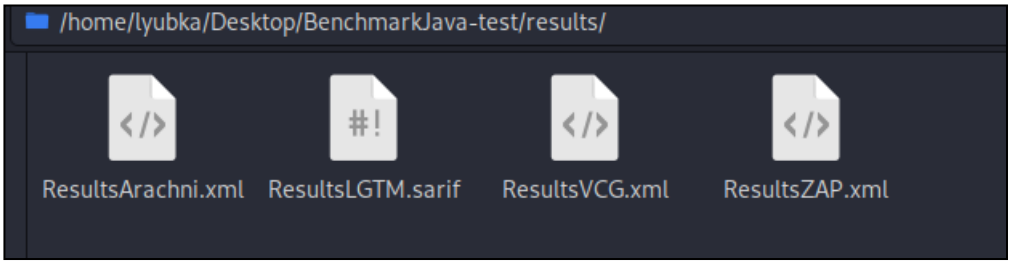

Figure 15: OWASP Benchmark - Results folder

Then the createScorecard.sh script is run from the Linux terminal (see Figure 16).

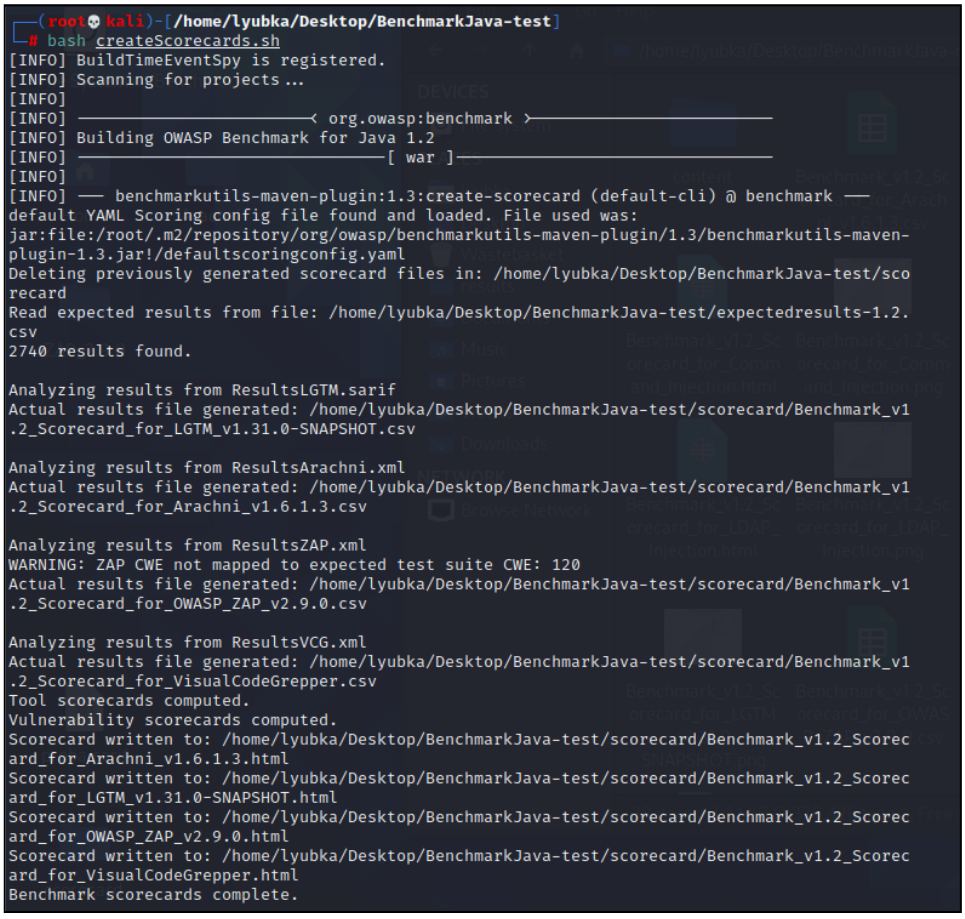

Figure 16: Script createScorecard.sh

This script generates scorecards for each of the Static Application Security Testing (SAST) and Dynamic Application Security Testing (DAST) tools and placed generated files in the OWASP Benchmark directory in the scorecard folder, as shown in Figure 17:

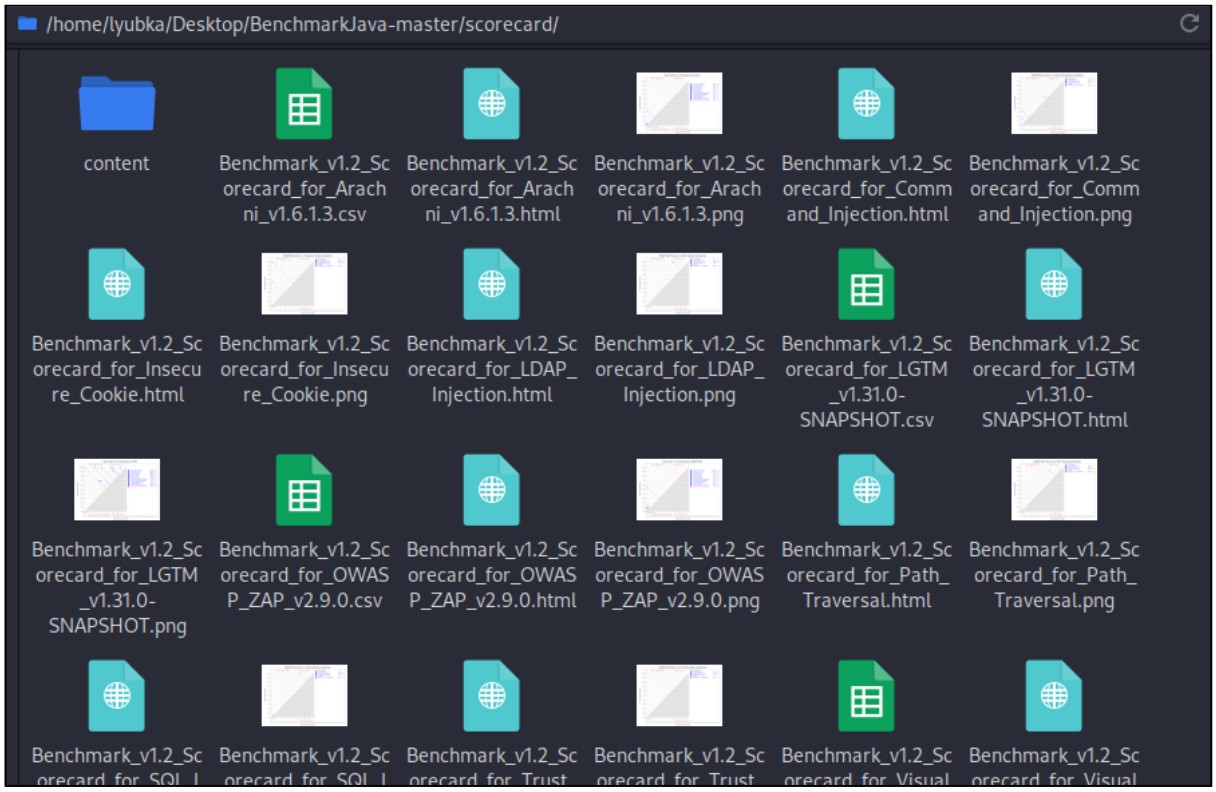

Figure 17: scorecard folder

The generated scorecards are extremely efficient and useful for performing a comparative analysis between the investigated types of security testing tools, SAST and DAST, which are the subject of this research work.

References

[1] Watson, J. (2008). VirtualBox: Bits and Bytes Masquerading as Machines. *Belltown Media*, *2008*(166), 1.