National College of Ireland

Configuration Manual

A MASVS-aligned evaluation of Android Static Analysis Tools Using the OWApp Benchmark

Jasdev Singh
Student ID: 23423340

School of Computing
National College of Ireland

Supervisor: Dr. Arghir Nicolae Moldovan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Jasdev Singh……

Student ID: 23423340……..……

Programme: MSc in Cybersecurity…… **Year:** 2025 - 26….

Module: Research Practicum Part 2……..……

Lecturer: Dr. Arghir Nicolae Moldovan……..……

Submission Due Date: 11/12/25……..……

Project Title: A MASVS-aligned evaluation of Android Static Analysis Tools Using the OWApp Benchmark
……..……

Word Count: 647…… **Page Count:** 9……..……..……

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: Jasdev Singh……

Date: 10/12/25……

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

Jasdev Singh
Student ID: 23423340

1 Introduction

This config manual documents the hardware, software and tool setup required to replicate the environment used in this research work. The study involved doing analysis of OWApp benchmark which involved evaluation of MASVS aligned vulnerable apps and their evaluation using four Static Application Security Testing (SAST) tools which were part of the benchmark. All environments are conducted within a controlled environment to ensure consistency and isolation from the host system. This manual provides the configuration details, installation steps and tool execution procedures to support replication or future extension of this work.

2 System Configuration

2.1 Host machine Specifications

Following are the specifications of the host machine

- **OS**: Windows 11 Pro
- **Processor**: Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz
- **Memory**: 12GB
- **VM**: VMware Workstation Pro
- **Storage**: 512GB (SSD backed storage)

2.2 Virtual Machine Specification

- **OS:** Kali Linux
- **Release**: 2025.3
- **CPU(s):** 4
- **Model name:** Intel(R) Processor
- **Memory Allocation:** 2GB
- **Disk Allocation:** ~ 50 GB

3 Tools Installed

This section outlines the installation and setup of tools required to scan APKs and process outputs. These steps were executed inside the kali VM.

3.1 Git and Benchmark download

- **Git Clone:**
  https://github.com/Mobile-IoT-Security-Lab/OWApp-Benchmarking-Suite.git

- **APK-file path:**
  OWApp-Benchmarking-Suite/OWApp/apk/ListedAPKs

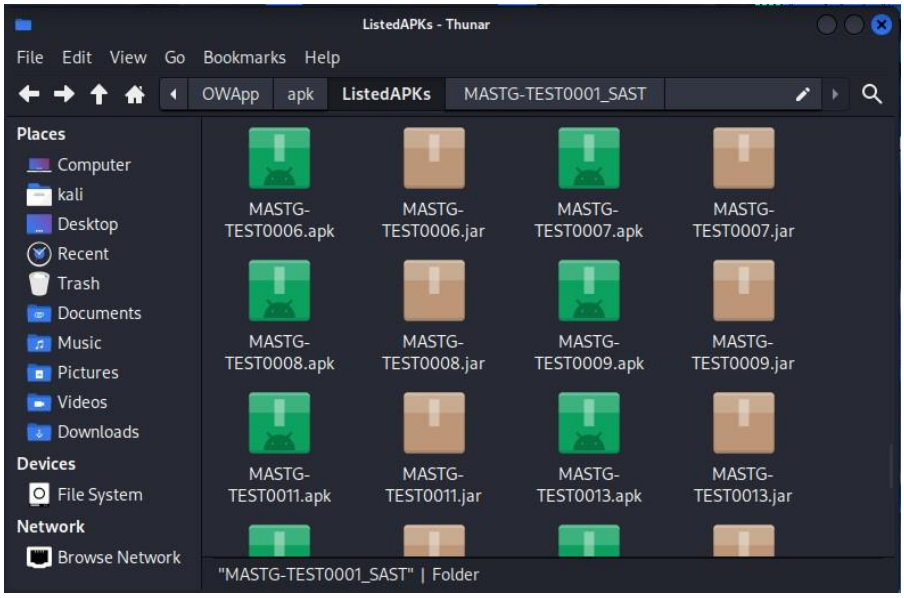

Fig 3.1: OWApp Benchmark APK's

3.2 Installing Go Runtime

Go runtime is required for the execution of APKHunt as it is one of the SAST tools of the benchmark and it requires go runtime to compile.

- **Installation cmd:**
  sudo apt install golang -y
  go version

- **APKHunt compilation cmd:**
  cd ~/APKHunt
  go build apkhunt.go
- **Resulting executable:**
  ~/APKHunt/apkhunt

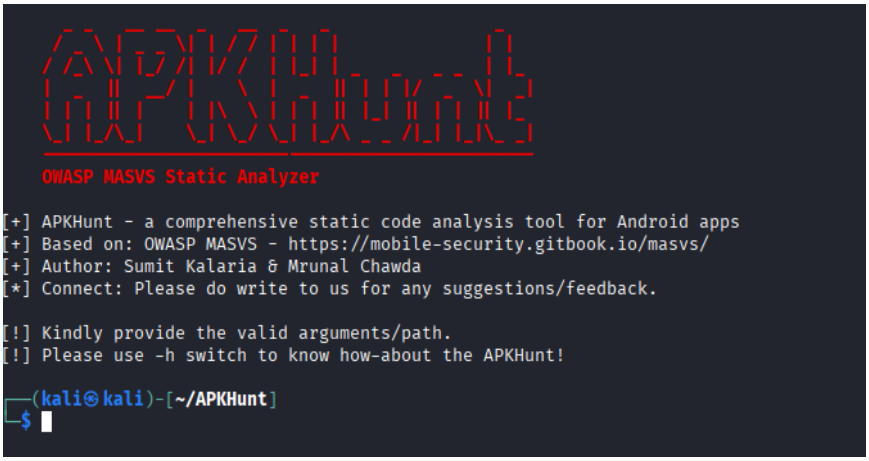

Fig 3.2: APKHunt build output

3.3 Docker Installation (for SEBASTiAN)

sudo apt install docker.io -y
sudo systemctl start docker
docker --version
docker pull talossec/sebastian:latest

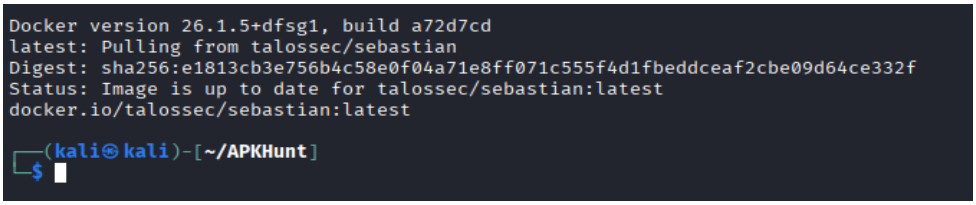

Fig 3.3: Docker installed and SEBASTiAN image pulled

3.4 Python 3 and libraries

- **Python version cmd:**
  python3 –version
- **Libraries cmd:**
  pip3 install pandas regex numpy

3.5 Additional Utilities

- Jadx : for decompilation of apps
- Apktool: for manifest extraction
- Dex2Jar: for converting Dex files to jar files
- grep / coreutils / zip

cmd: sudo apt install jadx apktool default-jdk unzip

4. Tools Configuration

This section explains how each SAST tool was executed and how results were collected.

4.1 OWApp Benchmark Setup

The OWApp Benchmark includes a series of scripts which we need to execute sequentially. Each one of them performs certain tasks including the scanning via SAST tools provided in the benchmark. Following is the link of the **README.md** file for detailed instructions from setting up the benchmark to scanning of the apps. Link: [OWApp-Benchmarking-Suite/README.md at main · Mobile-IoT-Security-Lab/OWApp-Benchmarking-Suite · GitHub]

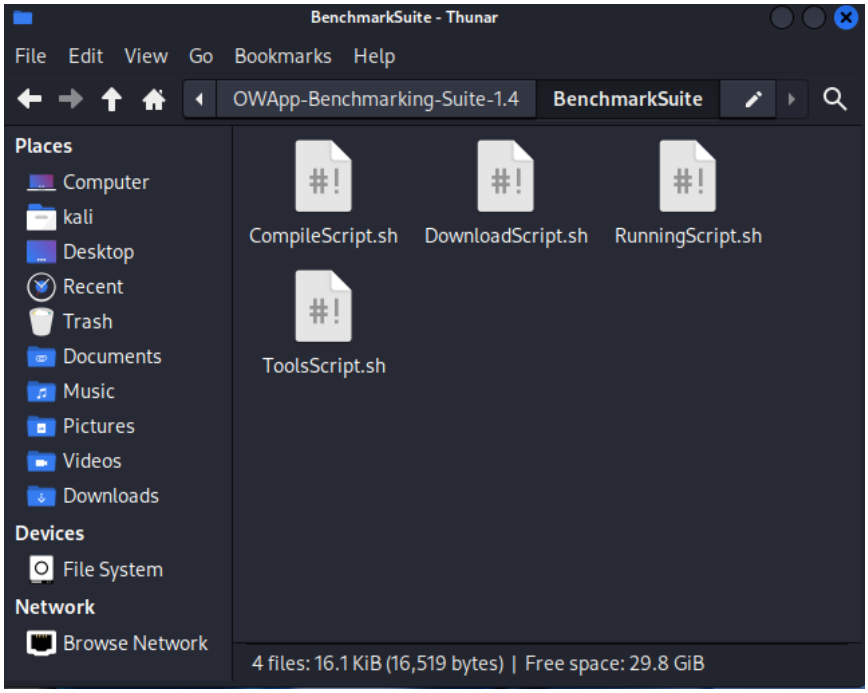

Fig 4.1: Screenshot of the scripts provided by the benchmark

After following each step provided in the README.md file, we will end up with output files after successful scanning from the SAST tools of the benchmark over all the apps in the package. Once we get the output files from the scan we can proceed further for processing and normalizing the output files and have some analysis from the scanned vulnerabilities from each tool.

4.2 Parsing Scripts

Custom python scripts were used to do some analysis, which are provided in the artifacts. Following were few of the metric analysis done in this study.

- Severity unification from all the tools in the benchmark
- Severity matrix for each tool
- App share analysis of each MASVS category in the benchmark

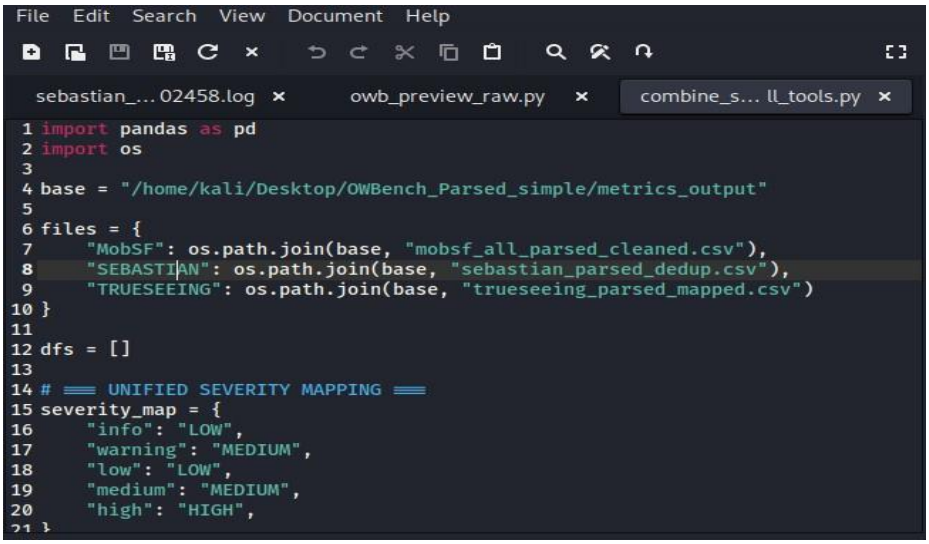

Fig 4.2: Snippet from the python script to unify severity

Fig 4.3: Snippet from the python script to find severity of each SAST tool

Fig 4.3: Snippet from the python script to find each app stats against each MASVS category

5. Summary

This configuration manual provides the essential setup details for reproducing the research environment, including system configuration, tool installation and execution, benchmark preparation and parsing workflows. All SAST scans and analysis steps were performed inside a controlled Kali Linux VM to ensure reproducibility and isolation. The document setup enables future researchers to replicate or extend this work with minimal overhead.