Combining Web Application Security Testing Tools

Darragh Madden
School of Enterprise Computing and Digital Transformation, TU Dublin, Ireland
X00180709@myTUDublin.ie

Introduction

2023 has seen numerous high-profile instances of data leaks and breaches across a variety of industries. It is imperative that organisations have a comprehensive security testing strategy in place to mitigate the risk posed by malicious actors. To aid in securing web applications, organisations typically employ one of three approaches: Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), Interactive Application Security Testing (IAST). Research has found that SAST & DAST tools are prone to identifying high numbers of false positives (report vulnerabilities where no issue actually exists). In an effort to reduce the number of false positives reported by tools research has investigated combining tools with increased vulnerability detection being observed.

Research Question: Does combining SAST, DAST & IAST tools result in enhanced vulnerability detection compared to using each tool individually?

Methodology

1. Tool Selection

This research investigated the performance of a combination of Open Source tools: FindSecBugs (SAST), OWASP ZAP (DAST) and Contrast Community Edition (IAST).

2. Benchmark

The OWASP Benchmark is an open source web application built in Java and is deployed in Apache Tomcat. The latest version of the Benchmark (v1.2) contains a set of approx. 2,700 fully exploitable test cases which can be analysed by security testing tools to detect true and false positives. A perfect score would see a tool identify 100% of true positives and 100% of true negatives.

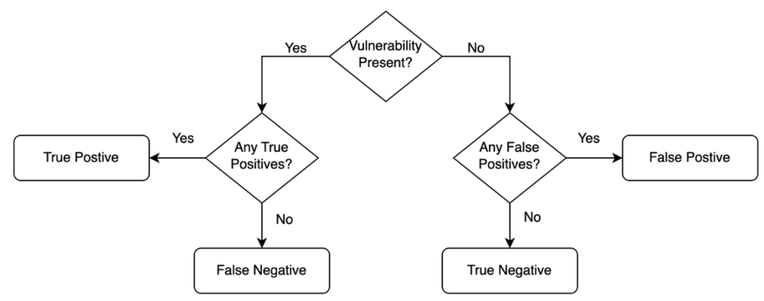
3. Experiment

Each tool and the OWASP benchmark were downloaded and run locally. The benchmark provides shell scripts for running both FindSecBugs and Contrast CE. OWASP ZAP was run in two steps: first using the Spider to identify all relevant HTTP links in the web app and then the Active Scan was run. The benchmark provides a separate shell script to convert the raw results files (XML & LOG) into CSV files containing the test results.

4. Method to Combine Results

Simple 1-out-of-N approach was used to combine the results for every test, for each of the 3 tools, to get a Combined Tools result.

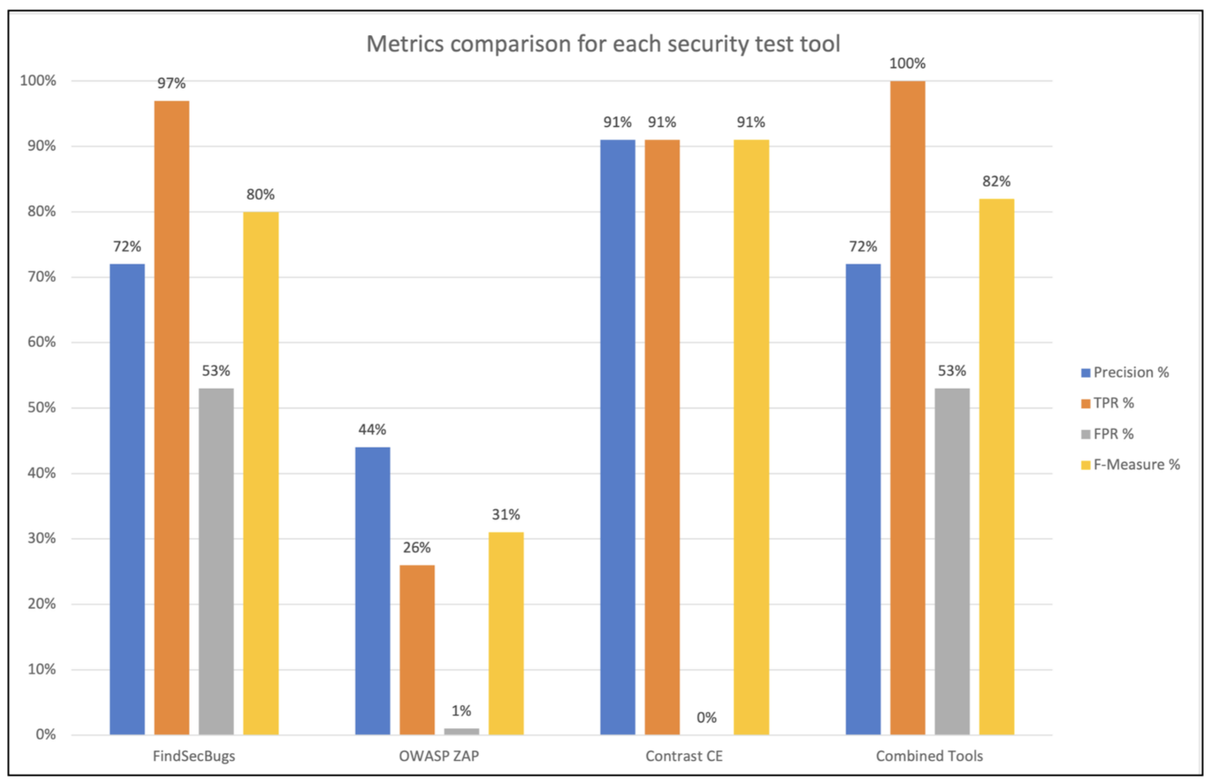
1-Out-of-N Approach

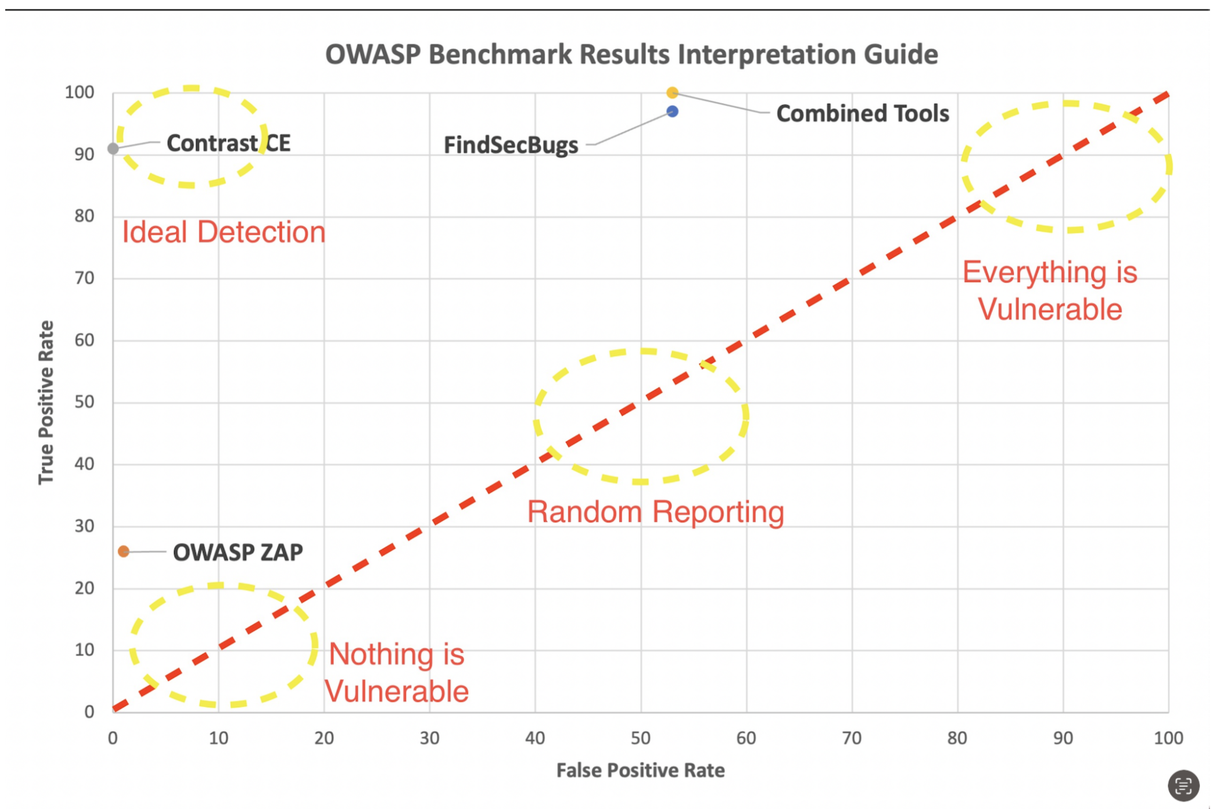


Combined Tool Test Results

	True Positive	False Negative	False Positive	True Negative
Command Injection	126	0	111	14
LDAP Injection	27	0	27	5
Path Traversal	133	0	129	6
Secure Cookie Flag	36	0	0	31
SQL Injection	272	0	210	22
Trust Boundary Violation	83	0	35	8
Weak Cryptography	130	0	0	116
Weak Hashing	129	0	0	107
Weak Randomness	218	0	0	275
XPATH Injection	15	0	19	1
XSS (Cross-site Scripting)	246	0	109	100
Total	1415	0	640	685

Results





Conclusions and Future Work

This study found that a combination approach can lead to enhanced vulnerability detection. A benefit of the approach is that each tool found vulnerabilities that the others did not, which resulted in 100% vulnerability detection.

A vital consideration however are the tools to be included: FindSecBugs reported a large number of false positives which the Combined Tools approach inherited. Surprisingly, OWASP ZAP didn't report many instances of vulnerabilities at all. These findings may bring into question the usefulness of a combined approach. A recommendation to mitigate this, and to maximize efficiency, is to give preference to tools with known low false positive detection.

Future work may research a combination approach utilizing commercial tools or implement a benchmark with modern vulnerability types present such as NISTs JULIET web application. Given the IAST tool was accurate, further research could also investigate combining multiple IAST tools to assess their performance.

QR Code for Recording