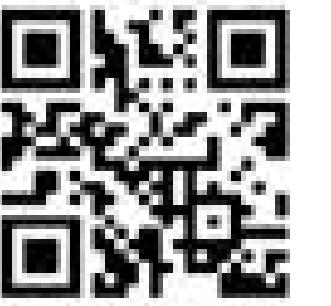


Security Test Driven Development Lifecycle

Analysis of The Effectiveness of Software Security Testing in the Web Application Development Process

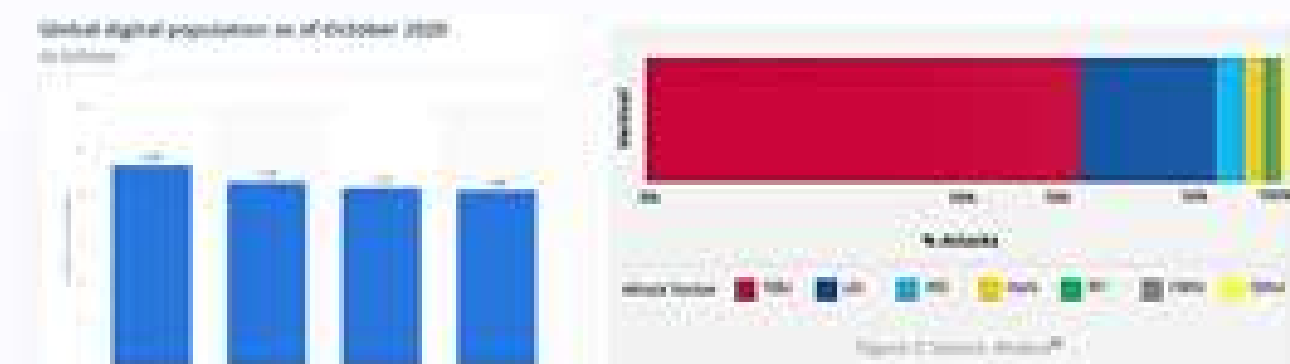
Project submitted as part of the requirements
for the award of BSc (Hons) Computing
(Web Development)

By: Mark Huntly (P1809693)
E: markhuntly@gmail.com



1.0 - Introduction

Over half the global population are internet users and the number still grows. Exploitation of vulnerabilities within web applications is prominent. Attack Vectors are still predominantly code injection while personal data and Credentials are the highest level of theft.



Web applications are inherently vulnerable due to access through a public interface and connections to back-end data.

The proposed hypothesis is that implementing a test driven development methodology to incorporate security into the software development lifecycle will create more security focused development processes. This will efficiently mitigate security concerns within web applications and allow a more secure deployment.

2.0 - Aim and Objectives

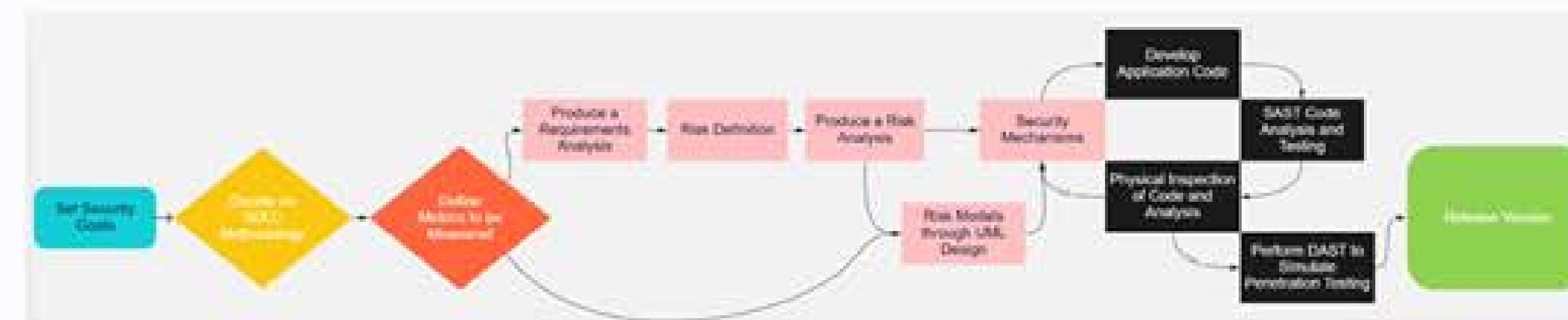
Aim

To analyse the effectiveness of tools and techniques for implementing security testing within web application development.

Objectives

- Rigorously investigate common security vulnerabilities associated with web applications and identify appropriate security testing tools and techniques to be used during development.
- Evaluate the effectiveness of integrating security in the web development process by developing a web application using appropriate testing strategies.
- Critically evaluate tools, procedures and processes used in

3.0 - Methodology



Using the Agile methodology, an application is developed with security in mind as above, creating a test driven environment. Tests are conducted on the application using the following steps:

- 1 - Save each file in turn
- 2 - Analyse the code through Static Analysis Security Tests (SAST)
- 3 - Analyse the application surface through Dynamic Analysis Security Tests (DAST)
- 4 - Commit iteration to GitHub



4.0 - Implementation and Testing



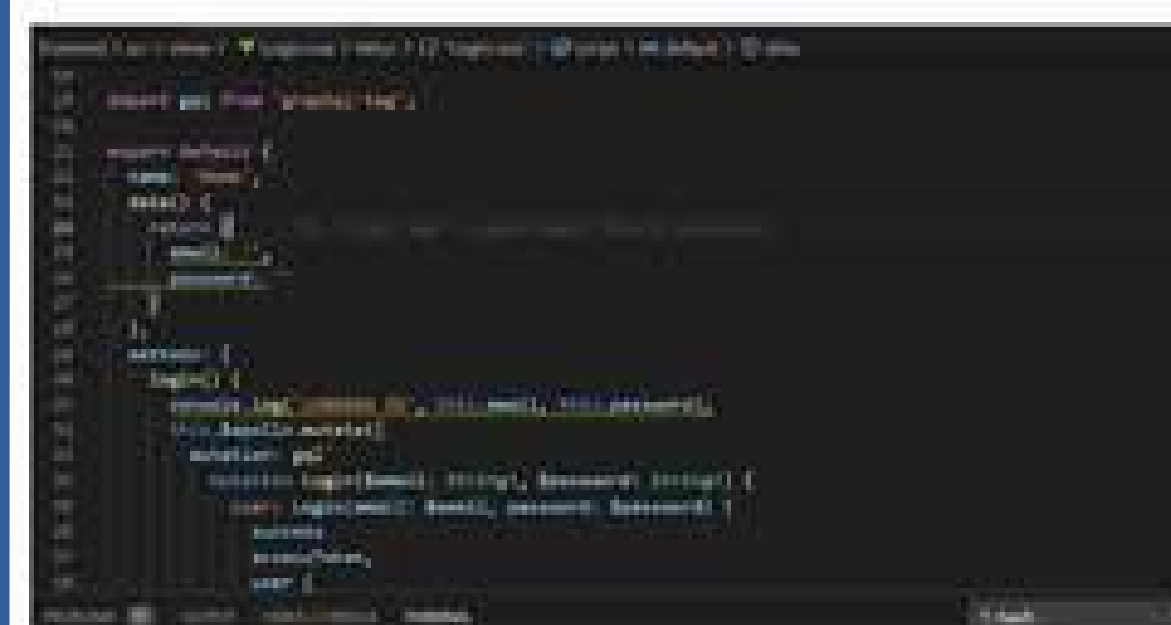
Resources for implementing the application

An inventory management application for a college media department was developed with vulnerable code built in for testing purposes.

A series of tests were implemented to simulate a test driven development environment



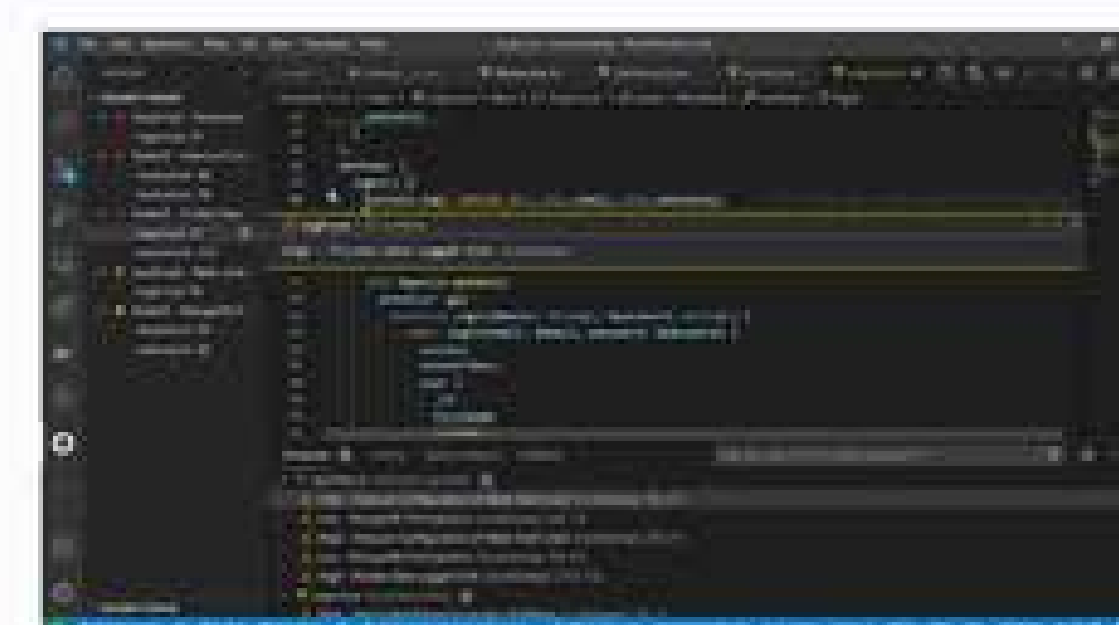
Resources for implementing the testing



Application code with vulnerabilities written in

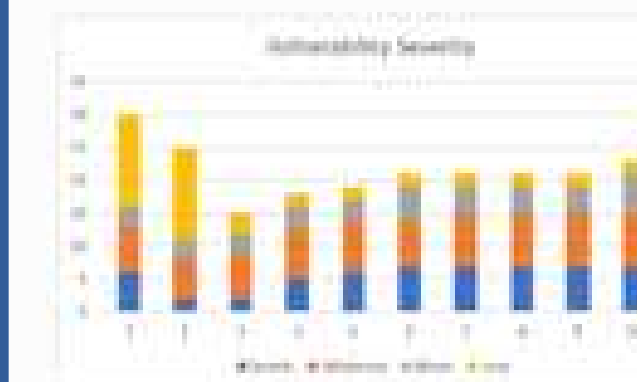
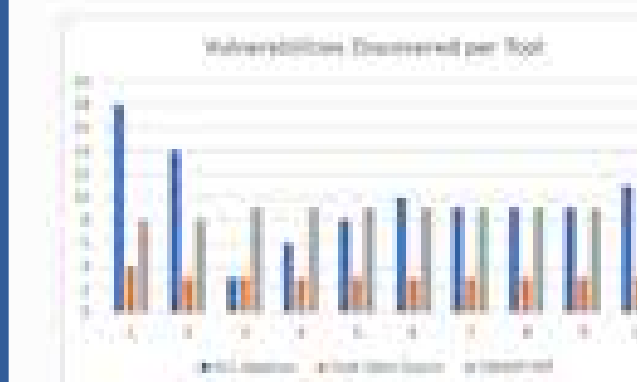
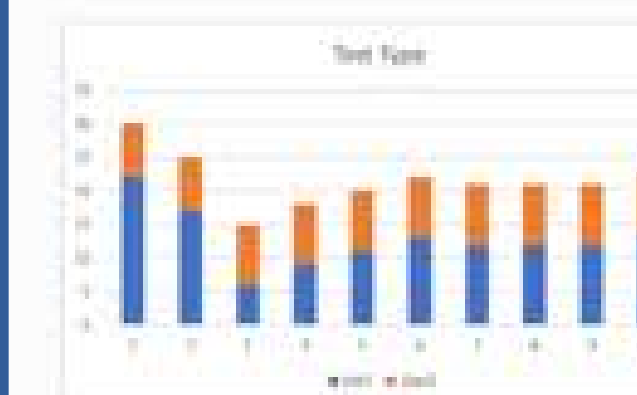


Preset test rules



Static analysis security test in operation using HCL AppScan CodeSweep

5.0 - Evaluation



- A consistently level set of
- Over 59% of vulnerabilities discovered overall through SAST
- HCL discovered most vulnerabilities at almost 45% overall
- Over 59% of vulnerabilities discovered overall through SAST
- Just over 26% of vulnerabilities overall were severe
- The most vulnerabilities found overall were moderate level at 35%
- Of the OWASP Top Ten Code Injection was the most at risk at over 29% detected
- Sensitive Data Exposure stays consistent across all 10 iterations

6.0 - Conclusion

- The testing supports the results from the literature review, showing code injection to be the biggest threat still within web applications.
- A comprehensive toolkit should be used combining different SAST and DAST tools for covering different aspects.
- By using TDD, it will aim to mitigate the threat of vulnerabilities through writing secure code to satisfy testing over iterations.
- A more secure application will be produced on deployment.

Future Study:

- Investigate better and more specific testing
- Development and research of function applicable tests