



MENG INDIVIDUAL PROJECT

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

---

# Webapp exploit detection through semi-automated black-box scanning

---

*Author:*

Abraão Pacheco Dos Santos

Peres Mota

*Supervisor:*

Dr. Sergio Maffei

*Second Marker:*

Dr. ??

January 8, 2018

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Objectives . . . . .	4
1.2	Challenges . . . . .	4
1.3	Contributions . . . . .	4
<b>2</b>	<b>Background</b>	<b>5</b>
<b>3</b>	<b>Evaluation</b>	<b>6</b>
<b>4</b>	<b>Project Plan</b>	<b>7</b>

# Chapter 1

## Introduction

You are therefore advised to keep this part of the interim report short, focusing on the following questions: What is the problem, why is it interesting and what's your main idea for solving it? (DON'T use those three questions as subheadings however! The answers should emerge from what you write.)

Hello [\[Gre93\]](#)

In recent years, use of internet applications has skyrocketed across the world. This is exacerbated by the ubiquity and sheer number of devices that are now connected to the internet. These devices (and their users by extension) place a great deal of implicit trust in the applications and websites they use to power the activities they engage in. This in turn means that the developers behind these web applications have a heavy responsibility on their shoulders - their products must uphold a high standard of security.

Sadly, due to the immaturity of the web development industry and how quickly technologies emerge in the field, web security is an often overlooked aspect of development. The lack of security as a founding basis for development is also due to a gap in developer education, as well as a somewhat high entry barrier to understand and mitigate potential security vulnerabilities. It is not treated as an important focus for new web developers to understand as part of many online and university courses, so many will get by, even work in professional development roles without that knowledge. Common vulnerability mitigation measures are also hidden in the inner workings of many frameworks developers use and become accustomed to. It is very often the case that these will be used without knowledge of what they do and how they work. However, in the case that the developer changes to use another framework without *secure by default* features, or decides to write an application from scratch, they will more often than not be left without a clue how to mitigate vulnerabilities, let alone know that these exist altogether.

This effectively highlights the overarching problem this project aims to tackle - improving security for web applications. It tries to do so through a pragmatic approach; the final goal of this project is to construct a tool which can diagnose vulnerabilities on a target website and goes a step further and attempts to use these to explore potential exploits on the target application. My initial proposal in solving this encompasses writing a browser extension to analyse the website the user is currently visiting, and applying a wide variety of scans and techniques to detect potential security pitfalls the website may have. A browser extension is an appropriate approach to this as it is a lightweight tool running with elevated privileges on the user's browser - this gives it the appropriate 'clearance' level to run a variety of scans on the user's behalf.

A tool of this kind has 2 immediately clear use cases. The first would be to give this tool to a website owner or developer and create it in such a way so that it gives clear and concise suggestions to quantifiably improve the security level of the target website - for example, if an SQL injection has been detected, show the user where on the website this vulnerability can be found, and make effective suggestions as to how to mitigate this (in an SQLi, it may be done by sanitizing user input). The tool could analyse a range of potential vulnerabilities and generate a quantifiable rating for the website in order to give feedback to the developer on where the website needs improving or

immediate work. The slightly alternative use case of this tool provides a more in-depth scan per potential vulnerability, and would be better suited for use by a penetration tester, or an otherwise knowledgeable web security expert. In this use case, the tool would work in a similar fashion, albeit with a different final goal of going 'all the way' by helping the user to detect vulnerabilities and creating exploits using them on the target application. In both potential use cases, the benefits of the tool are twofold - it can be used as an educational stepping stone for developers to further their understanding of security in web applications. It also provides a pragmatic way to improve website security through alternate routes - that of the direct owner improving their website by given suggestions, or the penetration tester showing developers the dangers of ignoring security for their website by exploiting open vulnerabilities.

This project aims to explore the latter of the two approaches mentioned. This is the more encompassing of the two possible use cases, with more information produced than just recommendations on how to improve a website. It also provides appropriate scope to allow for an in-depth analysis per potential vulnerability. Given that, this project aims to explore vulnerabilities in as much depth as possible, with less focus on attempting to detect as many vulnerabilities as possible. This choice is made with the evaluation of the tool in mind - as will be discussed later on. False positives (declaring that a website has a vulnerability when in fact it doesn't) are a major evaluation metric for the project and minimising these through more accurate, in-depth scans will result in a quantifiably better, more usable tool.

User driven approach

This goal poses several challenges, the biggest of which are:

1) Breadth of analysis 2) Depth of analysis 3) Accuracy of the scanner

## **1.1 Objectives**

## **1.2 Challenges**

## **1.3 Contributions**

## Chapter 2

# Background

This should form the bulk of the interim report. You should consider that your objective here is to produce a near final version of the background section, as it will appear in your final report. All of this material should be re-usable, so it is worth getting it right at this stage of the project. The details of what to include can be found in the Project Report guidelines.

## Chapter 3

# Evaluation

Project evaluation is very important, so it's important to think now about how you plan to measure success. For example, what functionality do you need to demonstrate? What experiments do you need to undertake and what outcome(s) would constitute success? What benchmarks should you use? How has your project extended the state of the art? How do you measure qualitative aspects, such as ease of use? These are the sort of questions that your project evaluation should address; this section should outline your plan.

## Chapter 4

# Project Plan

You should explain what needs to be done in order to complete the project and roughly what you expect the timetable to be. Don't forget to include the project write-up (the final report), as this is a major part of the exercise. It's important to identify key milestones and also fall-back positions, in case you run out of time. You should also identify what extensions could be added if time permits. The plan should be complete and should include those parts that you have already addressed (make it clear how far you have progressed at the time of writing). This material will not appear in the final report.

# Bibliography

- [Gre93] George D. Greenwade. The Comprehensive Tex Archive Network (CTAN). *TUGBoat*, 14(3):342–351, 1993.