



FINAL YEAR PROJECT REPORT

WEB APPLICATION VULNERABILITY SCANNER (WAVS)

**In fulfillment of the requirement
For degree of
BS (COMPUTER SCIENCES)**

By

HUZAIFA SHAHID

65203 (BSCS)

HASNAIN IMTIAZ ABASSI

51792 (BSCS)

SUPERVISED

BY

MR.SAGHIR AHMED

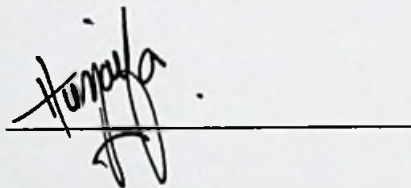
BAHRIA UNIVERSITY (KARACHI CAMPUS)

SPRING-2023

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

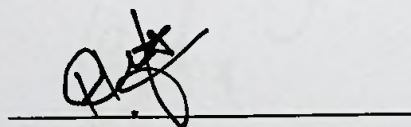
Signature :



Name : **Huzaifa Shahid**

Reg No. : **65203**

Signature :



Name : **Hasnain Imtiaz Abassi**

Reg No. : **51792**

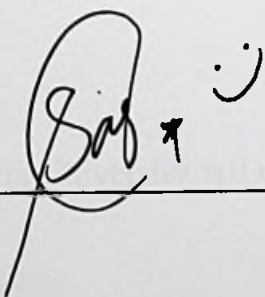
Date : _____

APPROVAL FOR SUBMISSION

We certify that this project report entitled “**Web Application Vulnerability Scanner**” was prepared by **Huzaifa Shahid and Hasnain Abbasi** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science in Computer Science at Bahria University, Karachi.

Approved by,

Signature :

A handwritten signature in black ink, appearing to read 'Sir Saghir', followed by a small smiley face ':)' and a checkmark '✓'. The signature is written over a horizontal line.

Supervisor : Sir Saghir Ahmed

Date : _____

The copyright of this report belongs to Bahria University according to the Intellectual Property Policy of Bahria University BUORIC-P15 amended on April 2019. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2023 Bahria University. All right reserved.

Specially dedicated to

My beloved grandmother, mother and father

Huzaifa Shahid

My beloved grandmother, mother and father

Hasnain Imtiaz Abbasi

(This dedication page is optional)

ACKNOWLEDGEMENTS

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express my gratitude to my project supervisor, Sir **Saghir Ahmed** for his invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, we would also like to express our gratitude to our loving parents and friends who have helped and given us encouragement.

WEB APPLICATION VULNERABILITY SCANNER

ABSTRACT

The internet has provided a vast range of benefits to society, and empowering people in a variety of ways. Due to incredible growth of Internet usage in past decades, everyday a number of new Web applications are also becoming a part of World Wide Web. The distributed and open nature of internet attracts hackers to interrupt the smooth services of web applications. Some of the famous web application vulnerabilities are SQL Injection, Cross Site Scripting (XSS) and Cross Site request Forgery (CSRF). We believe that in order to encounter these vulnerabilities; the web application vulnerabilities scanner should have strong detection and prevention rules to ease the problem. The result of the experiment shows proposed vulnerability scanner tool WAVS which gives less false positive and detect more vulnerabilities in comparison of well-known black box scanners. The system being developed will be a Web application system based on Net framework using PYTHON. It is an "SCANNER" tool which will be used by IT project developers, clients and even IT students For Cybersecurity. The basic feature of this website will be to provide a communication channel between a client, seeking a WEB for any vulnerabilities, and the Attacker, seeking for credential for which they will be paid. The client as well as the Website is required to sign up to the website so they can be considered as the member of the Tool. The Web will have to pass an online test in order to indication of bugs in the online bidding of the projects available on the website. The system will also provide a reporting and a discussion board which will be used for communication between the clients, developers and other interested parties, in an effective and efficient manner.

TABLE OF CONTENTS

DECLARATION	ii
APPROVAL FOR SUBMISSION	iii
ACKNOWLEDGEMENTS	vi
ABSTRACT	vii
TABLE OF CONTENTS	viii
LIST OF TABLES	xiii
LIST OF FIGURES	xiv
LIST OF APPENDICES	xv
CHAPTER	
1	INTRODUCTION 16
1.1	Background 16
1.2	Problem Statement 16
1.3	Aims and Objectives 17
1.4	Research Questions 17
1.5	Scope of Project 17
2	LITERATURE REVIEW 18
2.1	Overview 15
2.2	Web Application Testing 15
2.2.1	Functionality Testing 15
2.2.2	Usability Testing 15

2.2.3	Compatiblity Testing	15
2.2.4	Performance Testing.	16
2.2.5	Security Testing.	16
2.2.6	Stress Testing	16
2.2.7	Accessibility Testing	16
2.2.8	Regression Testing	16
2.3	Vulnerabilities in Web Application	16
2.3.1	Injection Attacks	16
2.3.2	Cross Site Scripting.	17
2.3.3	Cross Site Request Forgery	17
2.3.4	Security Misconfigurations	17
2.3.5	Broken Authentication and Session Management	17
2.3.6	Insecure Direct Object Refrences.	17
2.3.8	Server Side Request Forgery	17
2.3.9	XML External Entity	18
2.3.10	Remote Code Execution	18
2.3.11	File Upload Vulnerability	18
2.3.12	Server Side Template Injection	18
2.4	Software Process Model.	18
2.5	Related Works.	18

3	REQUIREMENT SPECIFICATIONS	23
3.1	Specific Requirements	23
3.1.1	Operating Systems	23
3.1.2	Hardware Requirements.	23

		x
3.1.3	Software Dependencies	24
3.1.4	Network Connectivity	23
3.1.5	Permission and Access	24
3.1.6	Scanning Scope and Configurations	24
3.1.7	Update and Patch Management	24
3.1.8	Resource for Report and Analysis.	25
3.2	System Requirements	25
3.3	Software Requirements	25
4	DESIGN AND METHODOLOGY	26
4.1	Design Overview.	26
4.2	WorkFlow	26
4.2.1	Subdomain Enumeration	26
4.2.2	Ports Scanning	26
4.2.3	Crawling	26
4.2.4	Directoy Brute-Forcing	26
4.2.5	Vulnerability Scanning	26
4.2.6	Cross-Checking	27
4.2.7	Reporting	27
4.3	Methodology	28
4.3.1	Reconnaissance (Vulnerabilty Scanner)	28
4.3.2	Enumeration	28
4.3.3	Vulnerability Scanner	28
4.3.4	Cross Check	29
4.3.5	View Reconnaissance	29
4.3.6	View Vulnerabilty Scan Results	29

		xi
	4.3.7 View Vulnerabilty Report	29
	4.3.8 Generate Scan Report	30
	4.4 Requirements Traceability Matrix	30
5	SYSTEM IMPLEMENTATION	35
	5.1 Determine the Scope and Requirements	35
	5.2 Architecture Design	35
	5.3 Scanning Engine Development	35
	5.4 Vulnerability Database.	35
	5.5 Deployment and Maintenance	35
	5.6 Continuous Improvement	35
	5.7 Scanner Templates	35
	5.7.1 Define Scan Scope	35
	5.7.2 Authentication Settings	35
	5.7.3 Scan Policies and Profiles	35
	5.7.4 Crawling and Spidering Options	37
	5.7.5 Input Injection and Fuzzing	37
	5.7.6 Request Headers and Cookies	37
	5.7.7 Exclusion Rules	37
	5.7.8 Reporting and Output Options	37
	5.7.9 Scheduling and Automation	37
	5.7.10 Integration and APIs	37
6	SYSTEM TESTING AND EVALUATION	39
	6.1 Subdomain Enumeration	39

		xii
6.2	Crawling and Fuzzing	40
6.3	Directory BruteForcing	40
6.4	Vulnerability Scanner.	41
6.5	Reporting	42
7	CONCLUSION	43
7.1	Introduction	43
7.2	Scanner Overview	43
7.3	Key Feature	42
7.3.1	Scanning Engine.	42
7.3.2	Authentication Support	43
7.3.3	Customizable Scanning Policies.	43
7.3.4	Input Injection and Fuzzing	43
7.3.5	Exclusion Rules.	44
7.4	Conclusion	44
	REFERENCES	45
	APPENDICES	46