## SECUREBYTE - AN APP FOR WEB PENTESTING

#### A PROJECT REPORT

***Submitted by***

**IMMANUEL SANJAY.J [REGISTER NO:211417104087]**

**JEEVA.V [REGISTER NO:211417104095]**

**DINESHKUMAR.A [REGISTER NO:211417104055]**

***in partial fulfillment for the award of the degree of***

### BACHELOR OF ENGINEERING

**IN**

#### COMPUTER SCIENCE AND ENGINEERING

**PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.**

**ANNA UNIVERSITY: CHENNAI 600 025**

**MARCH 2021**

**BONAFIDE CERTIFICATE**

Certified that this project report **“SecureByte - An app for Web Pentesting”** is the bonafide work of **“ IMMANUEL SANJAY .J (2017PECCS403), JEEVA .V (2017PECCS406), DINESHKUMAR .A (2017PECCS385) ”** who carried

out the project work under my supervision.

#### SIGNATURE SIGNATURE

**Dr.S.MURUGAVALLI,M.E.,Ph.D., Dr.S.Balaji,B.Tech.,M.E.,Ph.D.,**

**HEAD OF THE DEPARTMENT SUPERVISOR , ASSOCIATE PROFESSOR,**

DEPARTMENT OF CSE, DEPARTMENT OF CSE,

PANIMALAR ENGINEERING COLLEGE, PANIMALAR ENGINEERING COLLEGE, NAZARATHPETTAI, NAZARATHPETTAI,

POONAMALLEE, POONAMALLEE,

CHENNAI-600 123. CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voce Examination held on

#### INTERNAL EXAMINER EXTERNAL EXAMINER

**ACKNOWLEDGEMENT**

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors

**Tmt.C.VIJAYARAJESWARI**, **Thiru.C.SAKTHIKUMAR,M.E.,** and

**Tmt. SARANYASREE SAKTHIKUMAR B.E.,M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. S.MURUGAVALLI , M.E.,Ph.D.,**

for the support extended throughout the project.

We would like to thank my **Project Guide Dr.S.Balaji,B.Tech., M.E,Ph.D.** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

J. IMMANUEL SANJAY,

V. JEEVA,

A. DINESHKUMAR.

## ABSTRACT

Web pentesting is the process of testing the computer systems,network or web applications to find security vulnerabilities that an attacker could exploit.The main objective of penetration testing to identify the security weaknesses. It involves a methodological series of steps aimed at gathering information about the target system, finding vulnerabilities or faults in them, researching for exploits that will succeed against those faults or vulnerabilities and compromise the web application.

This application combines various types of web pentesting tools which perform various operations on particular targets in a single application.It generates the report of particular operations then we can analyse the generated report.The Personalized reports will be saved as report file in a text format.The report will be sent to an Prior-organization to alert them about their application or websites vulnerabilities and the system weaknesses. And the Organization may get aware about the reports and will immediately fix the loophole of the particular application or website.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT** | iv |
| **1.** | **INTRODUCTION** | 1 |
|  | 1.1 Overview | 2 |
| **2.** | **LITERATURE SURVEY** | 4 |
| **3.** | **SYSTEM ANALYSIS** | 9 |
|  | 3.1 Existing System | 10 |
|  | 3.2 Proposed system | 10 |
|  | 3.3 Requirement Analysis and Specification | 10 |
|  | 3.3.1 Input Requirements | 11 |
|  | 3.3.2 Output Requirements | 11 |
|  | 3.4 Feasibility Study | 11 |
|  | 3.4.1 Technical Feasibility | 11 |
|  | 3.4.2 Economic Feasibility | 11 |
|  | 3.5 Hardware Environment | 12 |
|  | 3.6 Software Environment | 12 |
|  | 3.7 Software Specification | 13 |

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
| **4.** | **SYSTEM DESIGN** | 20 |
|  | 4.1. UML diagrams | 21 |
| **5.** | **SYSTEM ARCHITECTURE** | 29 |
|  | 5.1 Architecture Overview | 30 |
|  | 5.2 System Module | 31 |
|  | 5.2.1 Module Explanation | 31 |
|  | 5.3 Program Design Language | 32 |
| **6.** | **TESTING** | 36 |
|  | 6.1 System Testing | 37 |
|  | 6.2 Unit Testing | 38 |
|  | 6.3 Test Cases & Reports / Performance Analysis | 39 |
| **7.** | **CONCLUSION** | 42 |
|  | 7.1 Conclusion and Future Enhancements | 42 |
|  | **APPENDICES** |  |
|  | A.1 Sample Screens | 43 |
|  | A.2 Sample Code | 50 |
|  | **REFERENCES** | 71 |

### LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
| **4.** | **SYSTEM DESIGN** |  |
|  | 4.1. Use Case Diagram | 25 |
|  | 4.2. Class Diagram | 26 |
|  | 4.3. Sequence Diagram | 27 |
|  | 4.4. Input Design | 28 |
| 5. | **SYSTEM ARCHITECTURE** |  |
|  | 5.1. System Architecture | 30 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
| **3.** | **SYSTEM ANALYSIS** |  |
|  | 3.1. Minimum Hardware Requirements | 12 |
|  | 3.2. Minimum Software Requirements | 12 |
|  |  |  |
| **6.** | **SYSTEM TESTING** |  |
|  | 6.1. Test Case for Pentest tool module | 39 |
|  | 6.2. Test Case for Other tool module | 40 |
|  | 6.3. Test Case for Report module | 40 |

**CHAPTER 1 INTRODUCTION**

**1.1 OVERVIEW**

Pentesting is a specialized security auditing method where a tester simulates an attack on a secured system. Using this application we can perform various operations. The goal of this is not to cause damage,but instead to identify attack surfaces,vulnerabilities,and other security weaknesses from the perspective of an attacker. This application contains port scanner, web server fingerprinting, directory brute force, password cracker, hashing, proxy, wifi security auditing tool,network scanning etc..,Many different tools are available in this application so we won’t go to different platform for various purposes.This helps finding out loopholes related to a particular component, be it web service, coding language, server etc and can be treated then and then.It has various security attack vectors and exploitation of potential vulnerabilities.

Information is Wealth. Each and every bit of information has a cost in this digital world. All that information is stored in the form of Data in Internet. There are two types of data, Public and Private. The public data are resources that are available publicly on the Internet. Ex: data that results from a Google search query. The private data are the resources that are bagged behind a wall of authentication. Ex: Your email data. Emails are protected by a wall of authentication which requires your user name and password to authenticate successfully. But what if someone can read your emails without authentication? Or what if someone can read your emails by acquiring your credentials from you without your knowledge? There comes the need for Web Application Security. Everything is web based now. Most of the Softwares has their own web app version too. But all the Web Applications are prone to Hacking. This is why Web Application Penetration emerges as a need of the hour. Websites need a defence in depth approach to mitigate against the security flaws1 . It is essential to Penetration test every web application before it goes online and gets hacked by a Black Hat cyber warrior out there. Hackers constantly hunt for web app vulnerabilities5 . The best way to mitigate against the hacker attacks is to learn their methodologies2 . Here, we discuss the most mandatory penetration tests that has to be done before the application goes Online and Techniques explaining how to perform those tests.

# CHAPTER 2

# LITERATURE SURVEY

#### Wang, S., Wang, J., Feng, C., & Pan, Z. (2016).

They can analyze the vulnerabilities and types of attacks on WLAN which is specified as IEEE 802.11 standard. The IEEE 802.11 WAN is a wireless network which uses radio waves to transfer the data. So, it is most susceptible to the security issues like WPE/WPA/WPA2 cracking, Denial of Service (DoS), and rogue access points. By siht, a ylisae rekcatta eht csecse eht srefsnart ,atad evitisnes eht s sllawerif eht ssapyb dna ,stekcap eht tpecretni . ,stekcap suoicilamThe penetrating testing ensures the security of the wireless networks. F ot ,oslA . loot gnitidua eht sa desu si SPDIAW ,skcatta sseleriw eht gnitceted ro,NALW eht gnitcetorp rof dna sksir eht etagitim the desu si noisurtni sseleriw. WAIDPS is an open-source wireless Swiss-Knife which works on Linux and is written in Python. This tool is designed to audit the networks and detect wireless intrusion. The results of this research indicated that WAIDPS could efficiently detect the attacks in order to protect WLAN

#### Goel, J. N., & Mehtre, B. (2015).

They can used Vulnerability Assessment and Penetration Testing (VAPT) for cyber defense.eser siht nIar,hc the nefed rebyc eht rof dezylana si TPAV eht fo ecnamrofrepse ygolonhcet to provide the proactive cyber defense. It helps in founding the vulnerabilities in advance for preventing the attacker from compromising a system. eser sihT ar eht debircsed hcPrevalent Vulnerability assessment techniques and VAPT tools. The VAPT process is step by step process that consists of 9 phases in its life cycle. The results of the research shown that VAPT is a useful technique for Cyber defense technology. The proposed method oslaallows an administrator to save the sensitive information and resources. It also helps in achieving the cybersecurity.

#### B L V Vinay Kumar, K Raja Kumar, & V. Santhi (2016).

They investigated different Penetration testing tools using Kali Linux. This research helped to understand how to perform the different penetration tests using private networks, virtualized tools, and systems. t lla tceted ot desu saw gnitset noitartenep ehThe skcatta l gniffins ciffart eki. kcatta elddiM-eht-ni- naM dnaF ces dna sisylana krowten retupmoc eht rou dna pacrettE ,gnitidua ytir desu si euqinhcet tentfirD. The implementation also used the Wireshark for traffic sniffing. The results demonstrated that the proposed technique for penetration testing could be used successfully in real time environment.

#### Fiocca, M. (2009).

He presented an introduction of Penetration testing to address the vulnerability of computer systems. This paper included a literature survey of Penetration testing which is performed by security experts to find the vulnerabilities of the system. This study mainly presented the two main types of penetration testing. These testing are black box testing and white box testing. The study also analyzed various tools of penetration testing specifically vulnerability scanners that included a more explained review of tools such as Nessus.

#### Salas, M., & Martins, E. (2014).

They proposed a technique for security testing which used the two techniques in order to detect the XSS attacks against the web services. The two techniques are Fault Injection and Penetrating Testing. XSS is a cross-site scripting attack on Web services which raises new security challenges. This type of testing technique is used to identify the sender by combining the security tokens and WSS (WS- Security). It also ensures the authorized access to SOAP messages communication. Another injection tool that was used is WSInject which introduces the faults or errors on Web Services for checking the environment behavior. The results indicated that the WSInject tool is better and improves the detection of vulnerability to compete with XSS attacks than soapUI.

#### T. Refaat, T. Abdelhamid and A. Mahmoud Mohamed (2016).

They presented a security solution for WLANs in order to achieve the standard network security requirements having low cost and stability. The proposed security solution works on two levels. These levels are Radio Frequency (RF) and the frame security. This solution is unique from the other as it works in the two WLAN security levels. The proposed solution provides the required frame security by incorporating AES encryption, and conjunction with 802.1x authentication Freeradius server for WLANs. This solution achieves the standard security requirements as AES offers the access control, free RADIUS server offers which is required for authentication, non-repudiation, standard confidentiality and integrity.

**D. Bertoglio and A. Zorzo (2017).**

They demonstrated the overview and open issues on the penetration test. This research also described various methodologies, tools, models, challenges, and application scenarios which are used for the security testing. This research helps in understanding the various aspects and the solutions which are related to the Pentest. In this research, a mapping study was conducted with 1145 papers. Out of which 1090 distinct papers are evaluated. At last, 54 primary studies were selected which were analyzed qualitatively and quantitatively. The author classified the models and tools which were used on Pentest. The results of the research help to define the testing scope and in evaluating the various tools and methodologies which were depending on the context.

# CHAPTER 3 SYSTEM ANALYSIS

### EXISTING SYSTEM

The tools were already available in the kali linux and other places. If we want to use a particular tool we have to use kali linux or other os. It was not easy to use the tools on the computer because of the dependencies. There will be many crashes or errors in the tools.

### PROPOSED SYSTEM

In this application, the various kinds of tools are available on the single application.This application performs the various operation on the particular target.It can generate report of particular operation.Using this report, we can alert the prior-organization about their application or websites vulnerabilities and system weaknesses.And the Organization may get an aware about the reports and will immediately fix the loophole of the particular application or website.

* + 1. Advantages
       1. It saves time complexity.
       2. Gives customer satisfaction.
       3. It is Portable.

4. Stable and more efficient.

### REQUIREMENT ANALYSIS AND SPECIFICATION

The requirement engineering process of feasibility study, requirements elicitation and analysis, requirement specification, requirements validation and requirement management. Requirement elicitation and analysis is an iterative process that can

be represented as a spiral of activities, namely requirements discovery, requirements classification and organization, requirement negotiation and requirements documentation.

### INPUT REQUIREMENT

The input requirement at the base requires data from the user that can be entered through the input box

### OUTPUT REQUIREMENT

The output things necessary for these are a database for storing and manipulation of user data and a knowledge server for manipulation of user data and sending appropriate notification for the respective person.

### FEASIBILITY STUDY

A feasibility study is carried out to select the best system that meets performance requirements. The main aim of the feasibility study activity is to determine that it would be financially and technically feasible to develop the product.

### TECHNICAL FEASIBILITY

This is concerned with specifying the software will successfully satisfy the user requirement. Open source and business-friendly and it is truly cross platform, easily deployed and highly extensible.

### ECONOMIC FEASIBILITY

Economic analysis is the most frequently used technique for evaluating the

effectiveness of a proposed system. The enhancement of the existing system doesn’t incur any kind of increase in the expenses. Programming Language used is python for App development is open source and readily available for all users. Since, the project is runned in the Visual Studio Code and hence it is cost efficient.

### MINIMUM HARDWARE REQUIREMENTS

|  |  |
| --- | --- |
| Processor | Core i3/i5/i7, 2.4 GHz |
| Hard disk | 500 GB |
| RAM | 2GB |
| Monitor | 14/15 inches Color |

* 1. **SOFTWARE REQUIREMENTS**

|  |  |
| --- | --- |
| Development Environment | Visual Studio Code |
| Language | Python 3 |
| Operating System | Windows 7/ 8.1/ 10 |
| Dependencies | Various python modules |

* 1. **SOFTWARE SPECIFICATION**

#### PYTHON

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

## 3.7.1.1 History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

## 3.7.1.2 Python Features

Python's features include −

* Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* Easy-to-read − Python code is more clearly defined and visible to the eyes.
* Easy-to-maintain − Python's source code is fairly easy-to-maintain.
* A broad standard library − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* Interactive Mode − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* Extendable − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* Databases − Python provides interfaces to all major commercial databases.
* GUI Programming − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* Scalable − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below:

* It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* It supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

|  |
| --- |
| **3.7.13 Python Applications** 1) Web Applications We can use Python to develop web applications. It provides libraries to handle internet protocols such as HTML and XML, JSON, Email processing, request, beautifulSoup, Feedparser, etc. One of Python web-framework named Django is used on **Instagram**. Python provides many useful frameworks, and these are given below:   * Django and Pyramid framework(Use for heavy applications) * Flask and Bottle (Micro-framework) * Plone and Django CMS (Advance Content management)  2) Desktop GUI Applications The GUI stands for the Graphical User Interface, which provides a smooth interaction to any application. Python provides a **Tk GUI library** to develop a user interface. Some popular GUI libraries are given below.   * Tkinter or Tk * wxWidgetM * Kivy (used for writing multitouch applications ) * PyQt or Pyside  3) Console-based Application Console-based applications run from the command-line or shell. These applications are computer program which are used commands to execute. This kind of application was more popular in the old generation of computers. Python can develop this kind of application very effectively. It is famous for having REPL, which means **the Read-Eval-Print Loop** that makes it the most suitable language for the command-line applications.  Python provides many free library or module which helps to build the command-line apps. The necessary **IO** libraries are used to read and write. It helps to parse argument and create console help text out-of-the-box. There are also advance libraries that can develop independent console apps. 4) Software Development Python is useful for the software development process. It works as a support language and can be used to build control and management, testing, etc.   * **SCons** is used to build control. * **Buildbot** and **Apache** Gumps are used for automated continuous compilation and testing. * **Round** or **Trac** for bug tracking and project management.  5) Scientific and Numeric This is the era of Artificial intelligence where the machine can perform the task the same as the human. Python language is the most suitable language for Artificial intelligence or machine learning. It consists of many scientific and mathematical libraries, which makes easy to solve complex calculations.  Implementing machine learning algorithms require complex mathematical calculation. Python has many libraries for scientific and numeric such as Numpy, Pandas, Scipy, Scikit-learn, etc. If you have some basic knowledge of Python, you need to import libraries on the top of the code. Few popular frameworks of machine libraries are given below.   * SciPy * Scikit-learn * NumPy * Pandas * Matplotlib  6) Business Applications Business Applications differ from standard applications. E-commerce and ERP are an example of a business application. This kind of application requires extensively, scalability and readability, and Python provides all these features.  Oddo is an example of the all-in-one Python-based application which offers a range of business applications. Python provides a **Tryton** platform which is used to develop the business application. 7) Audio or Video-based Applications Python is flexible to perform multiple tasks and can be used to create multimedia applications. Some multimedia applications which are made by using Python are **TimPlayer, cplay,** etc. The few multimedia libraries are given below.   * Gstreamer * Pyglet * QT Phonon  8) 3D CAD Applications The CAD (Computer-aided design) is used to design engineering related architecture. It is used to develop the 3D representation of a part of a system. Python can create a 3D CAD application by using the following functionalities.   * Fandango (Popular ) * CAMVOX * HeeksCNC * AnyCAD * RCAM  9) Enterprise Applications Python can be used to create applications that can be used within an Enterprise or an Organization. Some real-time applications are OpenERP, Tryton, Picalo, etc. 10) Image Processing Application Python contains many libraries that are used to work with the image. The image can be manipulated according to our requirements. Some libraries of image processing are given below.   * OpenCV * Pillow * SimpleITK |

**3.7.2 TKINTER**

Tk/Tcl has long been an integral part of Python. It provides a robust and platform independent windowing toolkit, that is available to Python programmers using the [tkinter](https://docs.python.org/3/library/tkinter.html#module-tkinter) package, and its extension, the [tkinter.tix](https://docs.python.org/3/library/tkinter.tix.html#module-tkinter.tix) and the [tkinter.ttk](https://docs.python.org/3/library/tkinter.ttk.html#module-tkinter.ttk) modules.The [tkinter](https://docs.python.org/3/library/tkinter.html#module-tkinter) package is a thin object-oriented layer on top of Tcl/Tk. To use [tkinter](https://docs.python.org/3/library/tkinter.html#module-tkinter), you don’t need to write Tcl code, but you will need to consult the Tk documentation, and occasionally the Tcl documentation. [tkinter](https://docs.python.org/3/library/tkinter.html#module-tkinter) is a set of wrappers that implement the Tk widgets as Python classes. In addition, the internal module \_tkinter provides a thread safe mechanism which allows Python and Tcl to interact.[tkinter](https://docs.python.org/3/library/tkinter.html#module-tkinter)’s chief virtues are that it is fast, and that it usually comes bundled with Python. Although its standard documentation is weak, good material is available, which includes: references, tutorials, a book and others. [tkinter](https://docs.python.org/3/library/tkinter.html#module-tkinter) is also famous for having an outdated look and feel, which has been vastly improved in Tk 8.5. Nevertheless, there are many other GUI libraries that you could be interested in.

# CHAPTER 4 SYSTEM DESIGN

### UML DIAGRAMS

UML stands for Unified Modeling Language. It’s a rich language to model software solutions, application structures, system behavior and business processes. There are 14 UML diagram types to help you model these behaviors. Unified Modeling Language™ (UML®) is a standard visual modeling language intended to be used for

* modeling business and similar processes,
* analysis, design, and implementation of software-based systems

UML is a common language for business analysts, software architects and developers used to describe, specify, design, and document existing or new business processes, structure and behavior of artifacts of software systems.

Specification explained that process:

* provides guidance as to the order of a team’s activities,
* specifies what artifacts should be developed,
* directs the tasks of individual developers and the team as a whole, and
* offers criteria for monitoring and measuring a project’s products and activities.

UML is intentionally process independent and could be applied in the context of different processes. Still, it is most suitable for use case driven, iterative and incremental development processes. An example of such process is Rational Unified Process (RUP).UML is not complete, and it is not completely visual. Given some UML diagram, we can't be sure to understand depicted part or behavior of the system from the diagram alone. Some information could be

intentionally omitted from the diagram, some information represented on the diagram could have different interpretations, and some concepts of UML have no graphical notation at all, so there is no way to depict those on diagrams. For example, semantics of multiplicity of actors and multiplicity of use cases on use case diagrams is not defined precisely in the UML specification and could mean either concurrent or successive usage of use cases.

Name of an abstract classifier is shown in italics while final classifier has no specific graphical notation, so there is no way to determine whether classifier is final or not from the diagram.

#### List of UML Diagram Types

So, what are the different UML diagram types? There are two main categories; structure diagrams and behavioral diagrams. Click on the links to learn more about a specific diagram type.

#### Structure Diagrams

Structure diagrams show the things in the modeled system. In a more technical term, they show different objects in a system. Behavioral diagrams show what should happen in a system. They describe how the objects interact with each other to create a functioning system.

#### Class Diagram

Class diagrams are the main building block of any object-oriented solution. It shows the classes in a system, attributes, and operations of each class and the relationship between each class. In most modeling tools, a class has three parts. Name at the top, attributes in the middle and operations or methods at the bottom.

In a large system with many related classes, classes are grouped together to create class diagrams. Different relationships between classes are shown by different types of arrows.

#### Component Diagram

A component diagram displays the structural relationship of components of a software system. These are mostly used when working with complex systems with many components. Components communicate with each other using interfaces. The interfaces are linked using connectors. The image below shows a component diagram.

#### Deployment Diagram

A deployment diagram shows the hardware of your system and the software in that hardware. Deployment diagrams are useful when your software solution is deployed across multiple machines with each having a unique configuration. Below is an example deployment diagram.

#### Package Diagram

As the name suggests, a package diagram shows the dependencies between different packages in a system. Check out this wiki article to learn more about the dependencies and elements found in package diagrams.

#### Composite Structure Diagram

Composite structure diagrams are used to show the internal structure of a class. For a detailed explanation of composite structure diagrams, click here.

#### Use Case Diagram

As the most known diagram type of the behavioral UML diagrams, use case diagrams give a graphic overview of the actors involved in a system, different functions needed by those actors and how these different functions interact.

It’s a great starting point for any project discussion because you can easily identify the main actors involved and the main processes of the system. You can create use case diagrams using our tool and/or get started instantly using our use case templates.

#### Activity Diagram

Activity diagrams represent workflows in a graphical way. They can be used to describe the business workflow or the operational workflow of any component in a system. Sometimes activity diagrams are used as an alternative to State machine diagrams. Check out this wiki article to learn about symbols and usage of activity diagrams.

#### Sequence Diagram

Sequence diagrams in UML show how objects interact with each other and the order those interactions occur. It’s important to note that they show the interactions for a scenario. The processes are represented vertically, and interactions are shown as arrows. This article explains the purpose and the basics of Sequence diagrams. Also, check out this complete Sequence Diagram Tutorial to learn more about sequence diagrams. You can also instantly start drawing using our sequence diagram templates.

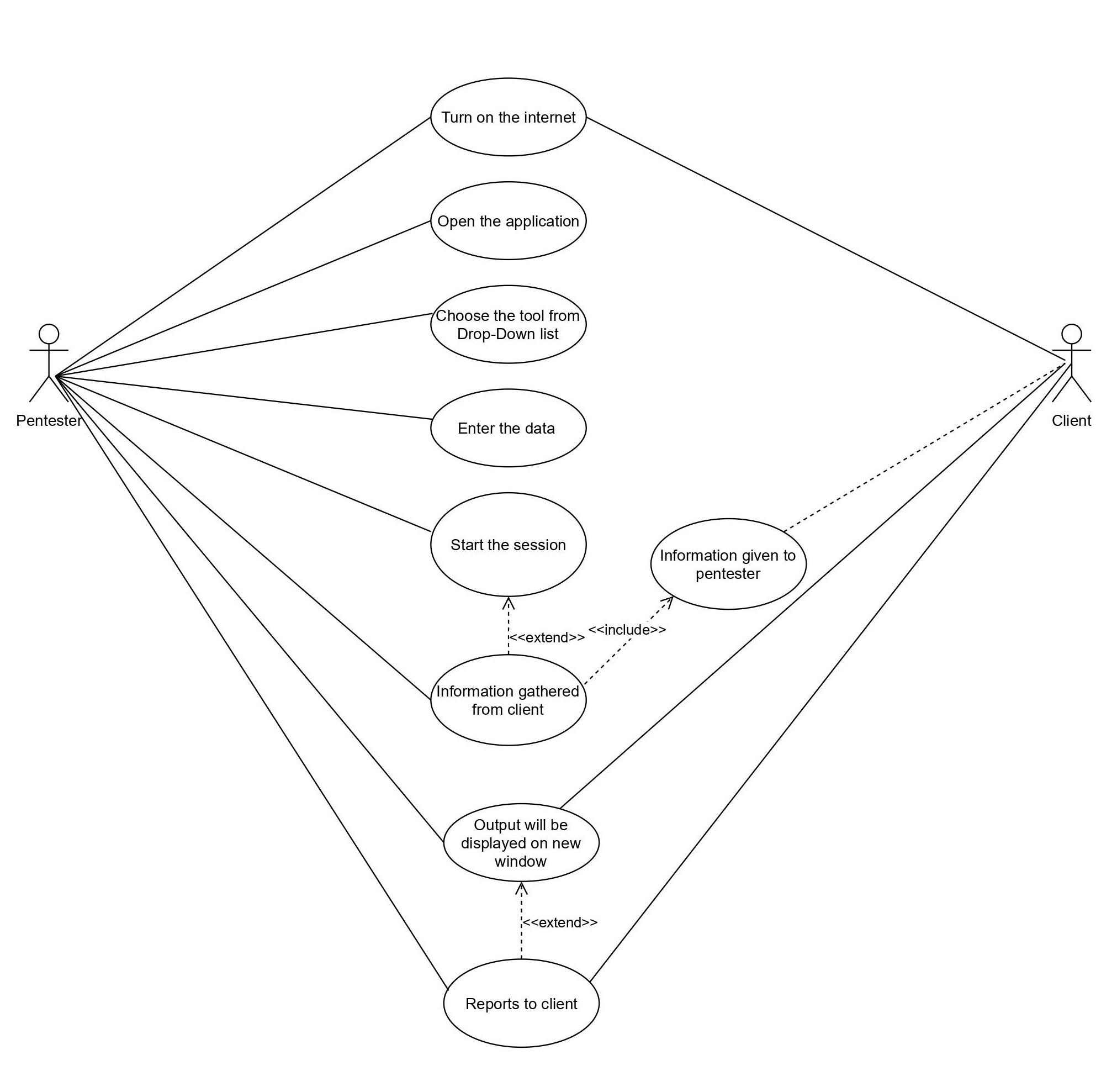


Fig 4.1 Use case diagram

In this use case ,there are two actor such as Pentester and Client.This use case explains about relationship between pentester and client that represents the user’s

interaction with the systems.

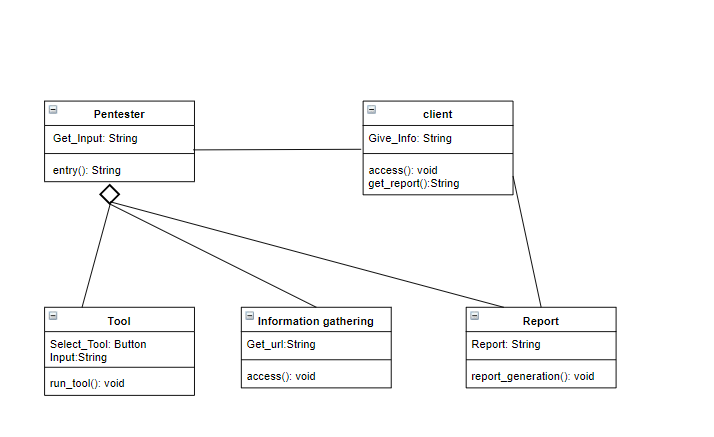


Fig 4.2 Class diagram

This class diagram contains attributes of the main classes as pentester, client and report. The client will send information to the pentester and the pentester will send the information to process the tool. The report will be sent to the pentester from the tool. The client will analyze the report.

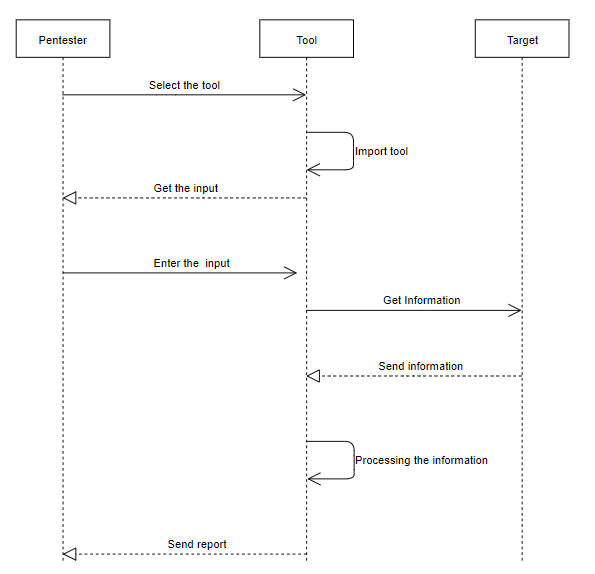
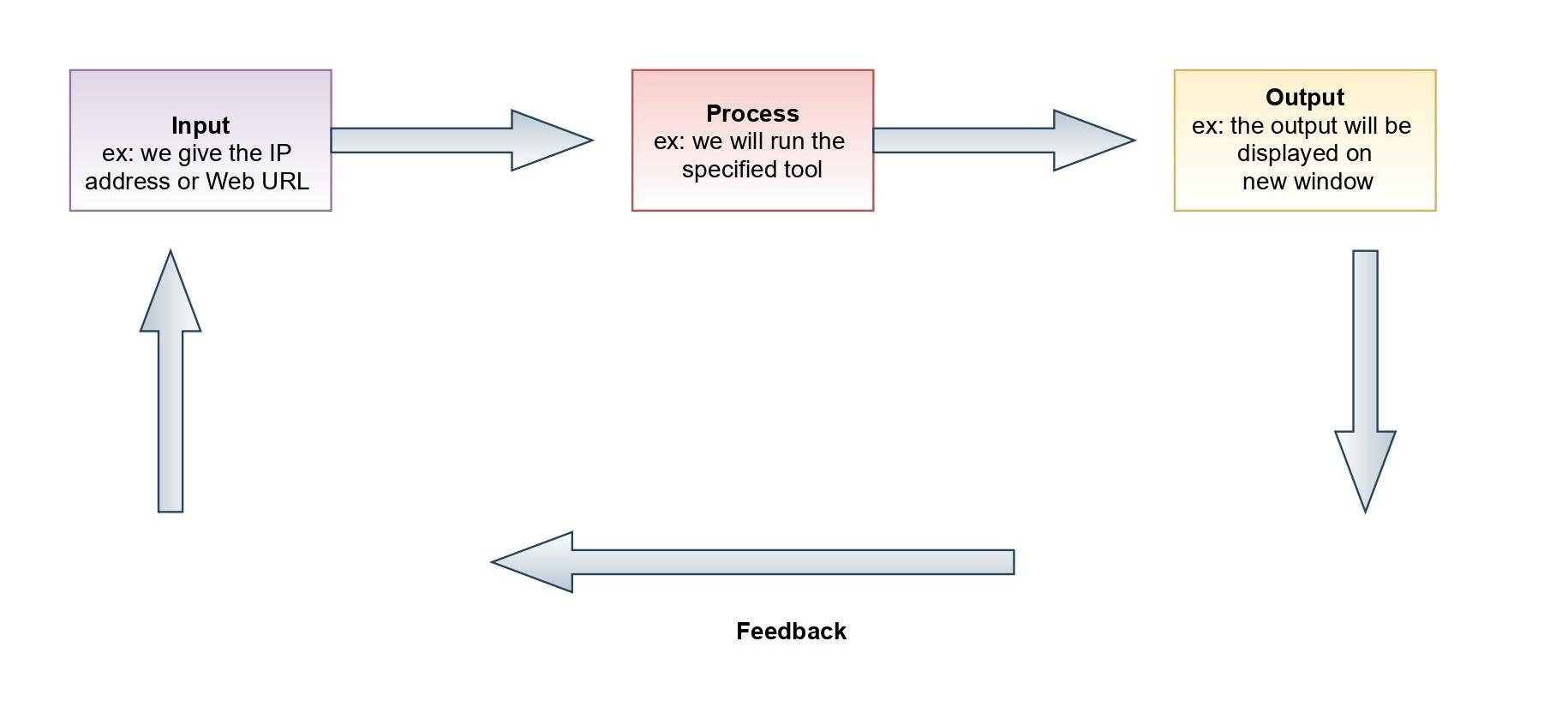


Fig 4.3 Sequence diagram

This sequence diagram explains the sequence of how the application runs the particular target sequentially. Here the pentester passes the command to the tool as a selection process and the tool imports the required input for the pentester. Now the pentester will enter the input in the tool and the tool will pass the input to the target. The tool will process the information and send the report to the pentester.

Fig 4.4 Input Design

This input design diagram shows how we take input from the user and then we process the information by running the tool and then we take the output and generate a report and display to the user.

### CHAPTER 5

### SYSTEM ARCHITECTURE

#### 5.1 ARCHITECTURE OVERVIEW

System architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

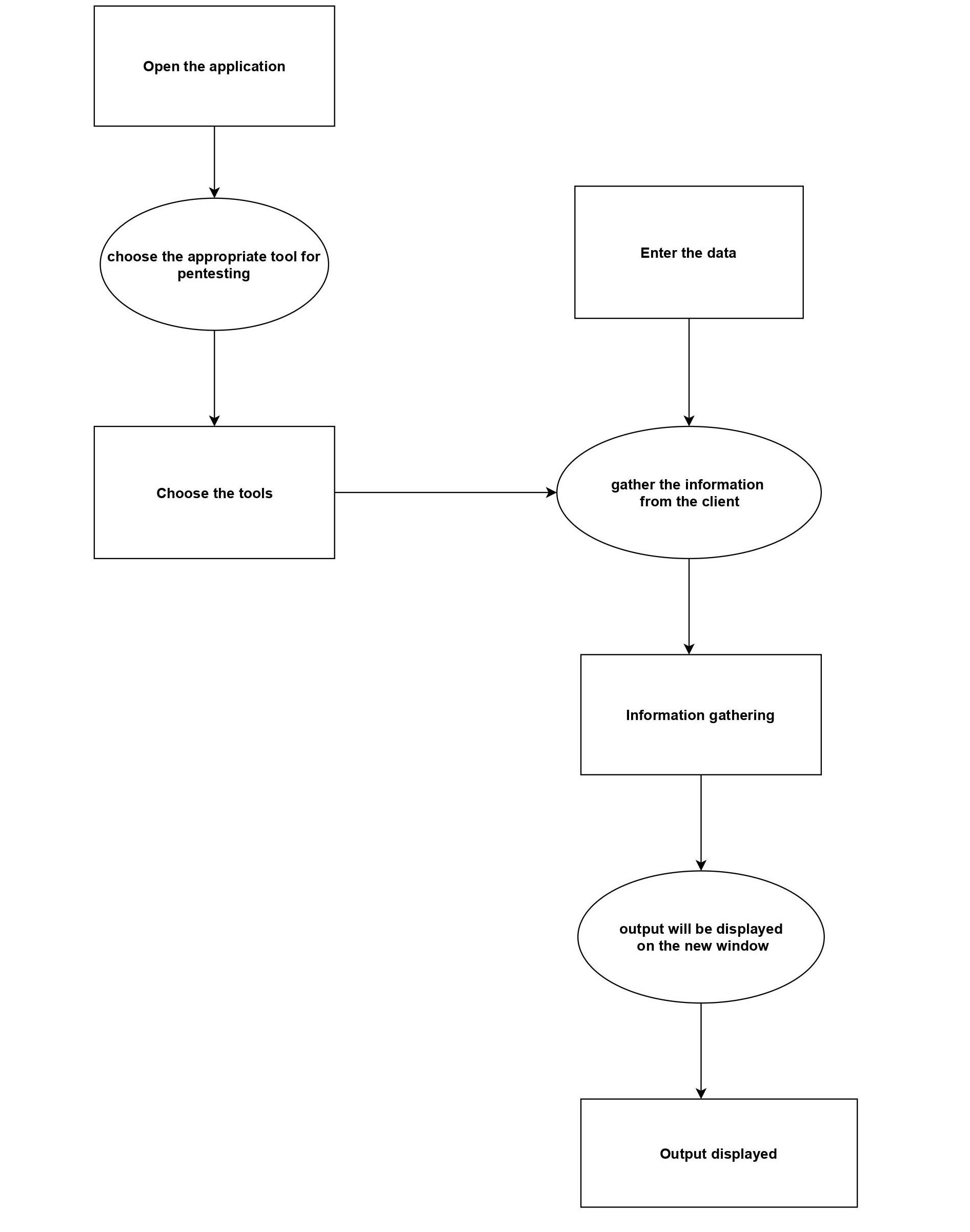


Fig 5.1 System Architecture

#### 5.2 SYSTEM MODULE

The SecureByte contains three modules functions namely:

* Pentest tools
* Other tools
* Report

#### MODULES EXPLANATION

**5.2.1.1 Pentest tools**

Pentest tools allow you to quickly discover and report vulnerabilities in websites,applications and network infrastructures. It provides a set of powerful and tightly integrated pentesting tools which enable you to perform easier, faster and more effective pentest engagements.These Pentest tool perform individual operations such as port scanner, Subdomain scanner, sql scanner, directory scanner, xss scanner.

**5.2.1.2 Other tools**

Other tools are different from the pentest tools and the operation performed is also different. The other tools are hash creator, brute hash, zip file cracker,password brute forcing, system directory scanner.

**5.2.1.3 Report**

The report is an account or statement describing in detail an event, situation, or the like, usually as the result of observation, inquiry, etc.Here the detailed output of the each scanners will be stored as an docx file. This docx file will be retrieved through the prior directory in the file explorer for the future use.

#### PROGRAM LANGUAGE DESIGN

**ALGORITHM:**

1. Make sure the Internet must be turned on for the entire session of the program
2. Open the Application -> SecureByte.exe
3. By clicking on the pentest tools drop-down box different types of scanner that perform various functions and select one of them to perform operations.

* Port scanner
* Subdomain scanner
* Sql Scanner
* Directory Scanner
* XSS scanner

1. On the other side the other tools drop-down box contains some another type of tools

* Brute Hash
* Hash creator
* Password brute forcing
* System directory scanner
* zipfile cracker

1. Output will be displayed on the current screen and also it will be saved as a text file.
2. The text file will be used as a report that can be to the prior-organization .
3. The report section contains the recent outputs that were previously generated.

#### Port Scanner

A port scan is a method for determining which ports on a network are open. As ports on a computer are the place where information is sent and received. Running a port scan on a network or server reveals which ports are open and listening (receiving information), as well as revealing the presence of security devices such as firewalls that are present between the sender and the target. This technique is known as fingerprinting. It is also valuable for testing network security and the strength of the system’s firewall. Due to this functionality, it is also a popular reconnaissance tool for attackers seeking a weak point of access to break into a computer.

#### XSS Scanner

Cross-Site Scripting (XSS) is one of the most well known web application vulnerabilities. ... Test for XSS: For each page discovered in the previous step, the scanner will try to detect if the parameters are vulnerable to Cross-Site Scripting and report them in the results page.

#### Directory Brute Forcing

Brute force directory guessing attacks are very common attacks used against websites and web servers. They are used to find hidden and often forgotten directories on a site to try to compromise.

#### Subdomain Scanner

The Subdomain Scanner is a subdomain discovery tool. It allows you to run a scan for a top-level domain name to discover target organization subdomains configured in its hierarchy.

#### SQL Scanner

A SQL injection scanner is an automated tool used to verify the vulnerability of websites and web apps for potential SQL injection attacks. During a SQL injection attack, the hacker attempts to illegally retrieve stored database information like usernames, passwords, etc. SQL injection scanners test the websites and web apps to check whether they are prone to SQL injection attacks.

#### Hash Creator

The MD5 hashing algorithm is a one-way cryptographic function that accepts a message of any length as input and returns as output a fixed-length digest value to be used for authenticating the original message

This tool is used to create a hash output for the given input string

#### Brute Hash

Brute force is also used to crack the hash and guess a password from a given hash. In this, the hash is generated from random passwords and This tool is used to give the original string for the hashed input string.

#### System directory

The Directory Scanner allows you to scan a certain directory on the file system for files containing XML messages. Once the messages have been read, they can be passed into the core message pipeline where the full collection of message processing filters can act on them. It shows the directories present in a particular system.

**5.3.9 Password brute forcing**

In cryptanalysis and computer security, password brute forcing is the process of recovering passwords from data that has been stored in or transmitted by a computer system in scrambled form. This tool shows how much time it takes to crack a password using brute forcing

**5.3.10 Zip file cracker**

ZIP password cracker is software that you use for recovering passwords of encrypted archive files.This tool is used to crack password protected zipfiles

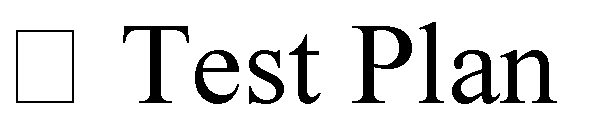
**CHAPTER 6 TESTING**

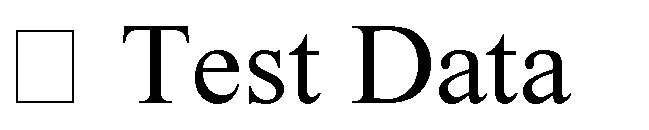
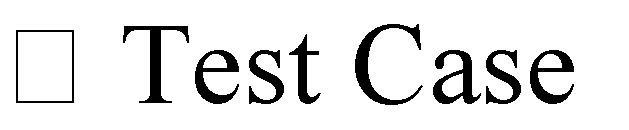
#### SYSTEM TESTING

The testing approach document is designed for Information and Technology Services’ upgrades to SecureByte. The document contains an overview of the testing activities to be performed when an upgrade or enhancement is made, or a module is added to an existing application. The emphasis is on testing critical business processes, while minimizing the time necessary for testing while also mitigating risks. It’s important to note that reducing the amount of testing done in an upgrade increases the potential for problems after go-live. Management will need to determine how much risk is acceptable on an upgrade by upgrade basis.

System testing is simply testing the system as a whole; it gets all the integrated modules of the various components from the integration testing phase and combines all the different parts into a system which is then tested. Testing is then done on the system as all the parts are now integrated into one system the testing phase will now have to be done on the system to check and remove any errors or bugs. In the system testing process the system will be checked not only for errors but also to see if the system does what was intended, the system functionality and if it is what the end user expected.

There are various tests that need to be conducted again in the system testing which include:





If the integration stage was done accurately then most of the test plan and test cases would already have been done and simple testing would only have to be done in order to ensure there are no bugs because this will be the final product. As in the integration stage, the above steps would need to be re-done as now we have

integrated all modules into one system, so we have to check if this runs OK and that no errors are produced because all the modules are in one system.

#### Unit Testing

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

#### TEST CASES

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Cas e Id | Test Cases | Priorit y | Input Test Data | Test Case Descriptio n | Expected Results | Actual Results | Pass/Fai l |
| TU0 1 | Select  the tool | A | Select the tool | Selecting the appropriate tool from the drop down menu | The appropriate tool window has been opened | The appropriate tool window has been opened | Pass |
| TU0 2 | Enter  the input | A | Enter the input | Giving the input for running the tool | Tool has been run | Tool has been run | Pass |
| TU0 3 | Get output | A | - | Getting the appropriate tool output | The output has been displayed on the same window | The output has been displayed on the same window | Pass |

Table 6.1 Test Cases for Pentest tools module

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case Id | Test Cases | Priorit y | Input Test Data | Test Case Description | Expected Results | Actual Results | Pass/Fa il |
| TE0 1 | Select  the tool | A | Select the tool | Selecting the appropriate tool from the drop down menu | The appropriate tool window has been opened | The appropriate tool window has been opened | Pass |
| TE0 2 | Enter  the input | A | Enter the input | Giving the input for running the tool | Tool has been run | Tool has been run | Pass |
| TE03 | Get output | A | - | Getting the appropriate tool output | The output has been displayed on the same window | The output has been displayed on the same window | Pass |

Table 6.2 Test Cases for Other tools module

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case Id | Test Cases | Priority | Input Test Data | Test Case Description | Expected Results | Actual Results | Pass/Fail |
| TC01 | Click the report | A | Click the report. | By clicking the report button the report window has opened | The report has been displayed | The report has been displayed | Pass |

Table 6.3 Test Cases for Report module

## CHAPTER 7

**CONCLUSION AND FUTURE ENHANCEMENT**

#### CONCLUSION

The objectives of this work is to offer a fast, reliable and automated testing tool, which is also easier to use than existing tools. It contains almost all the needed tools for pentesting so it makes the process simple. It is an application so we can add as many tools as we can. It is portable, stable and reliable. Minimal configuration required. Quickly discover the attack surface of a target organization. Speed-up your pentesting engagements.

#### FUTURE ENHANCEMENT

We have created the second version of this application. In that application we have changed the UI and also we have reduce the number of files. So this application is easy to use and its lightweight. The work is under progress.We have added screenshot of SecureByte2.0 in appendices

**APPENDICES**

**A 1. SAMPLE SCREENS**

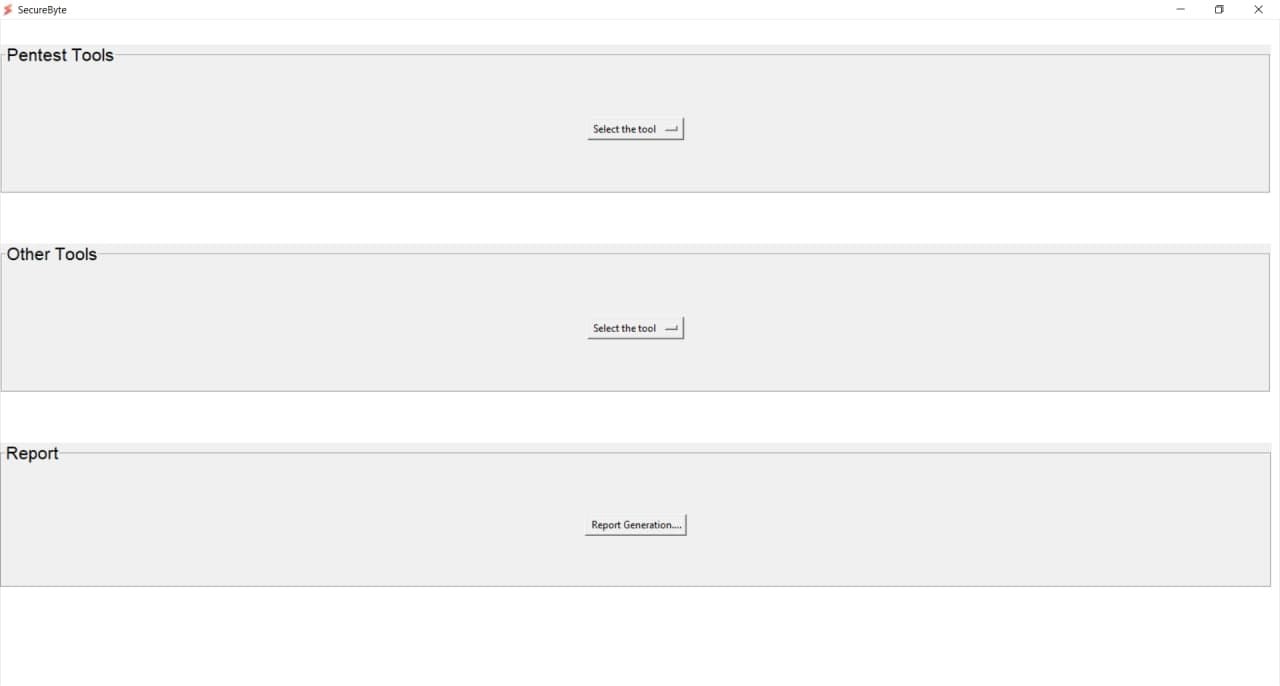


Fig.1 SecureByte Main Window

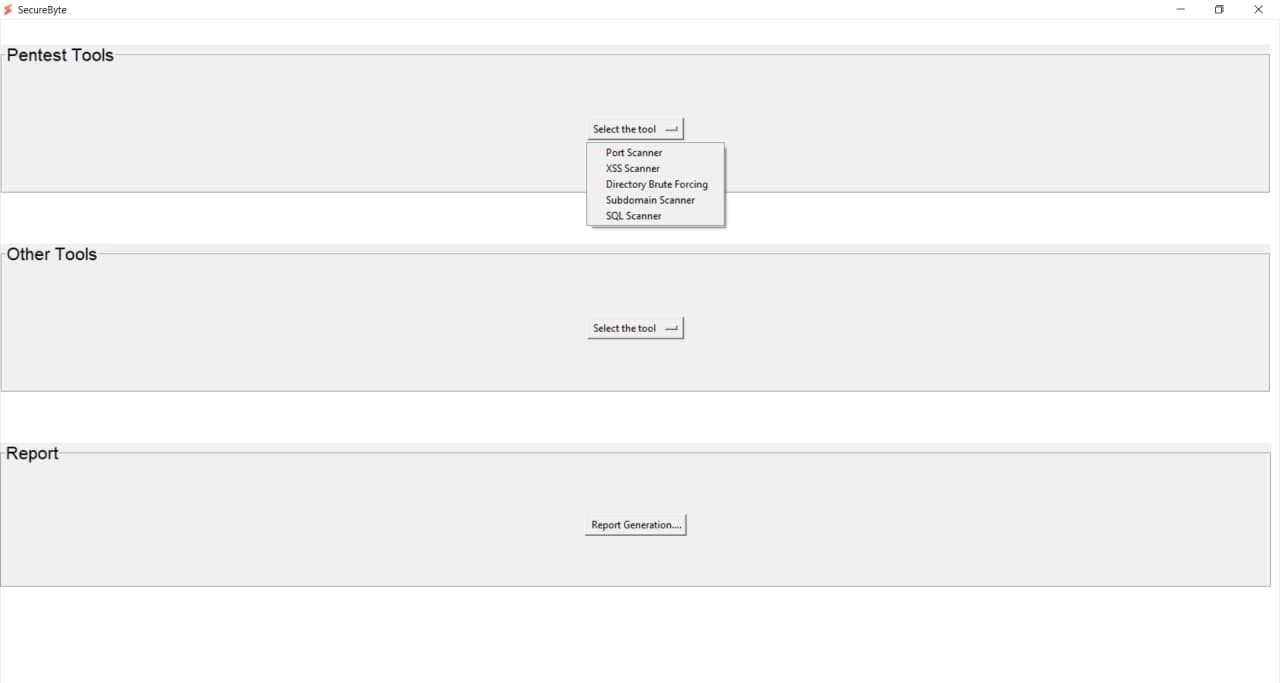


Fig.2 Select The Tool

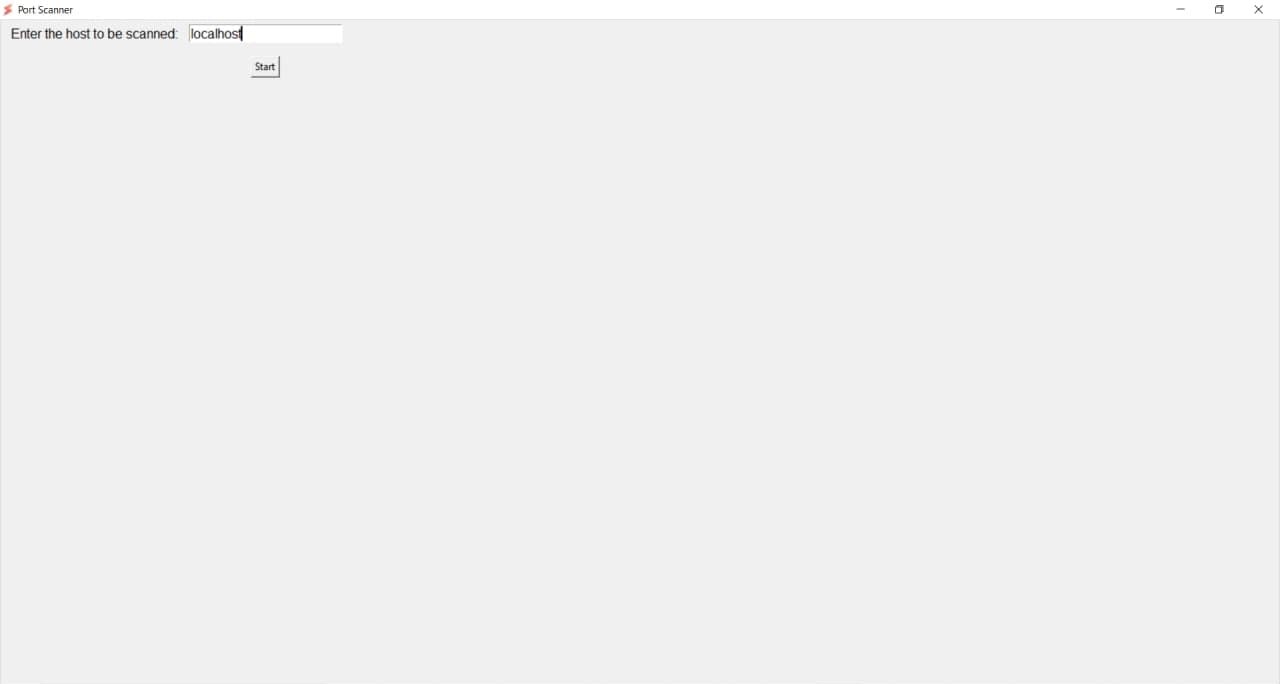


Fig.3 Port Scanner Tool



Fig.4 Port Scanner Output

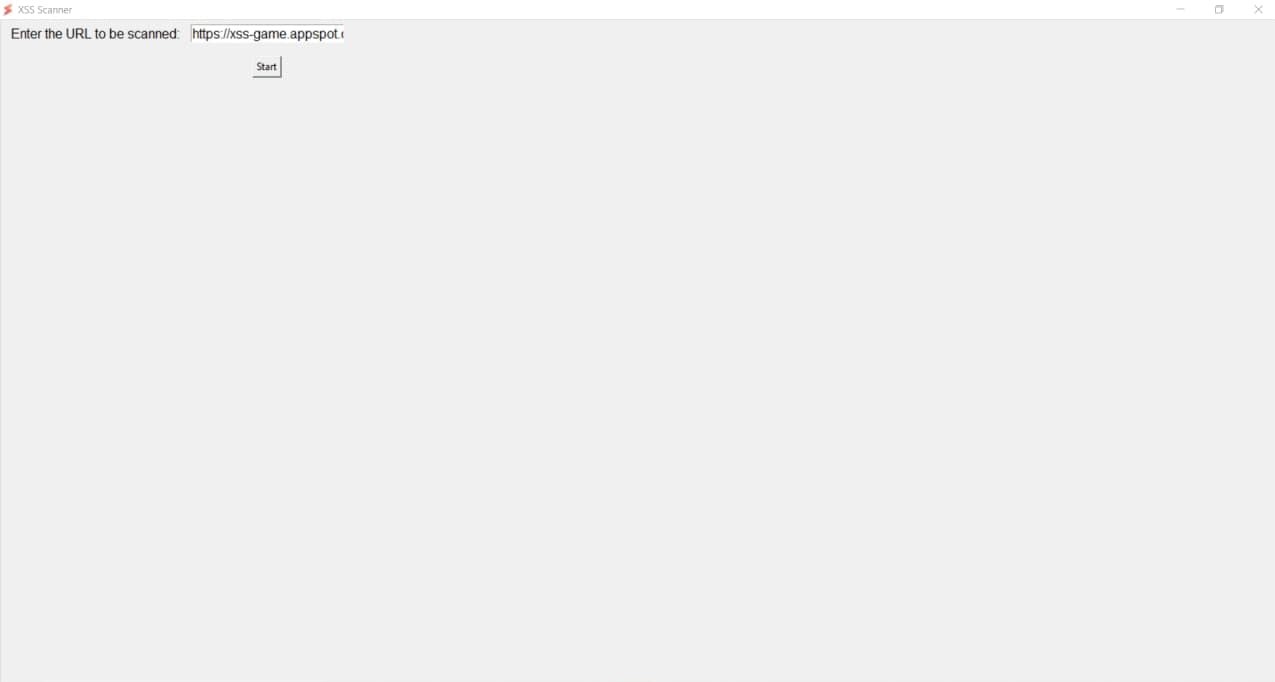


Fig.5 XSS Scanner Tool

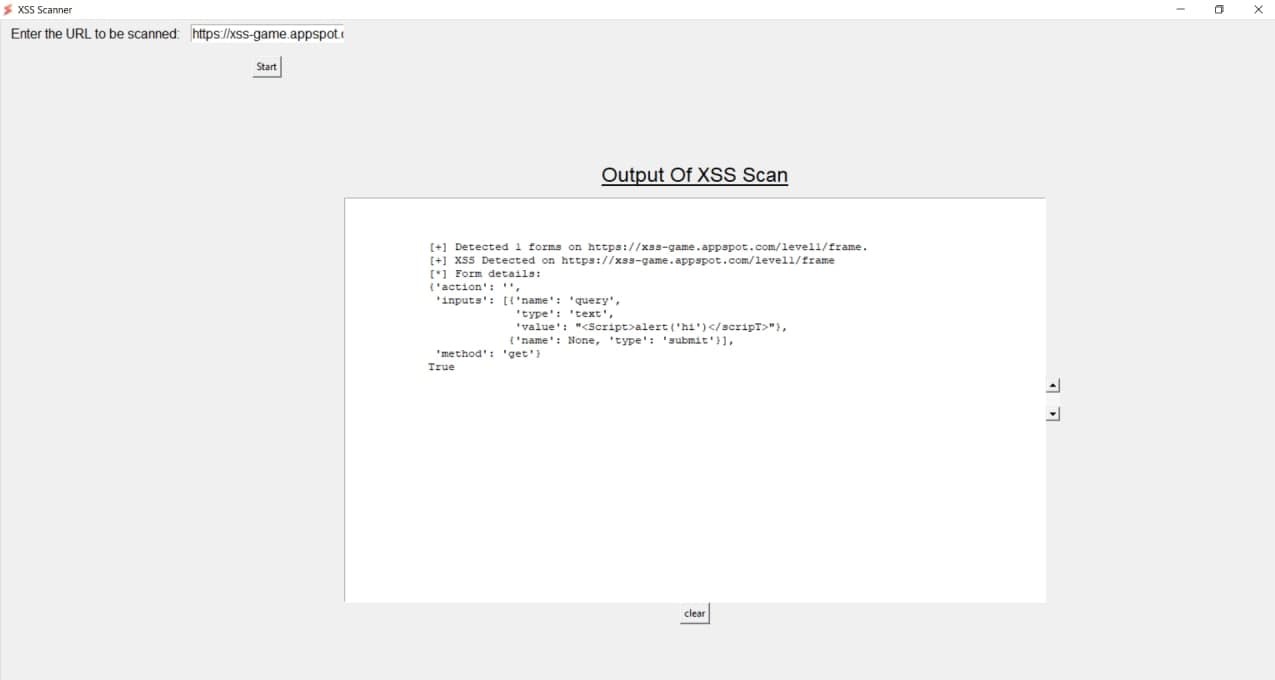


Fig.6 XSS Scanner Output

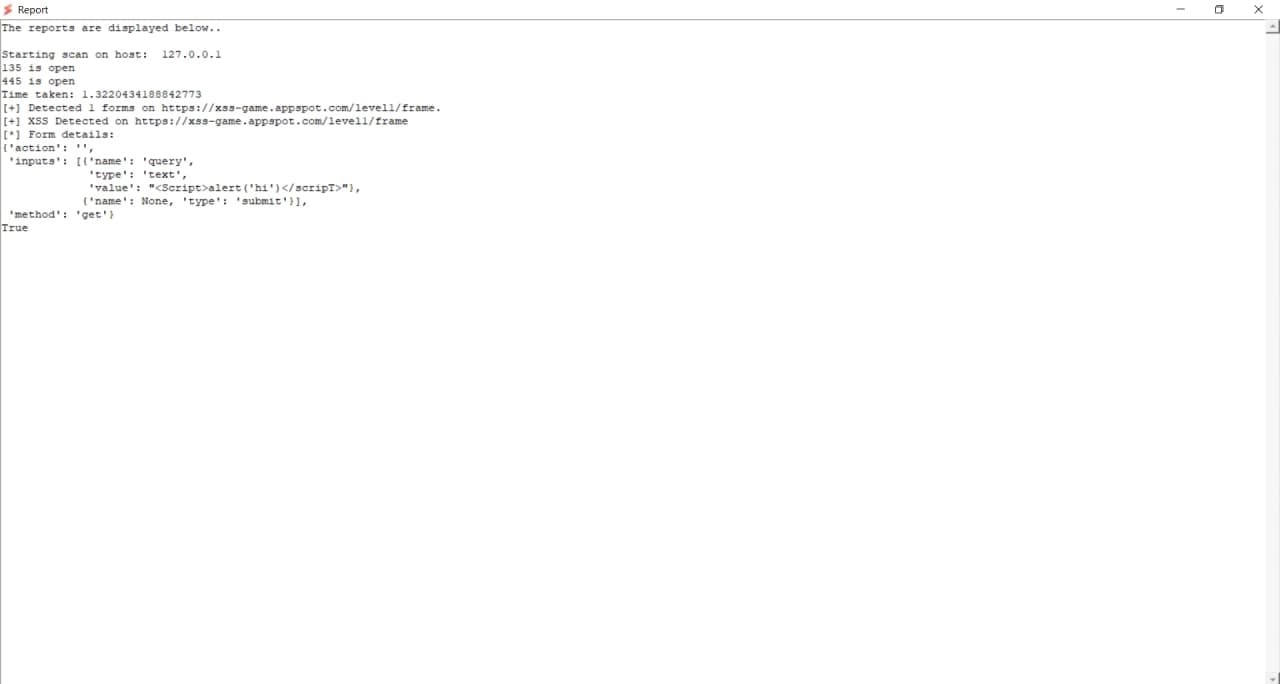


Fig.7 Full Report

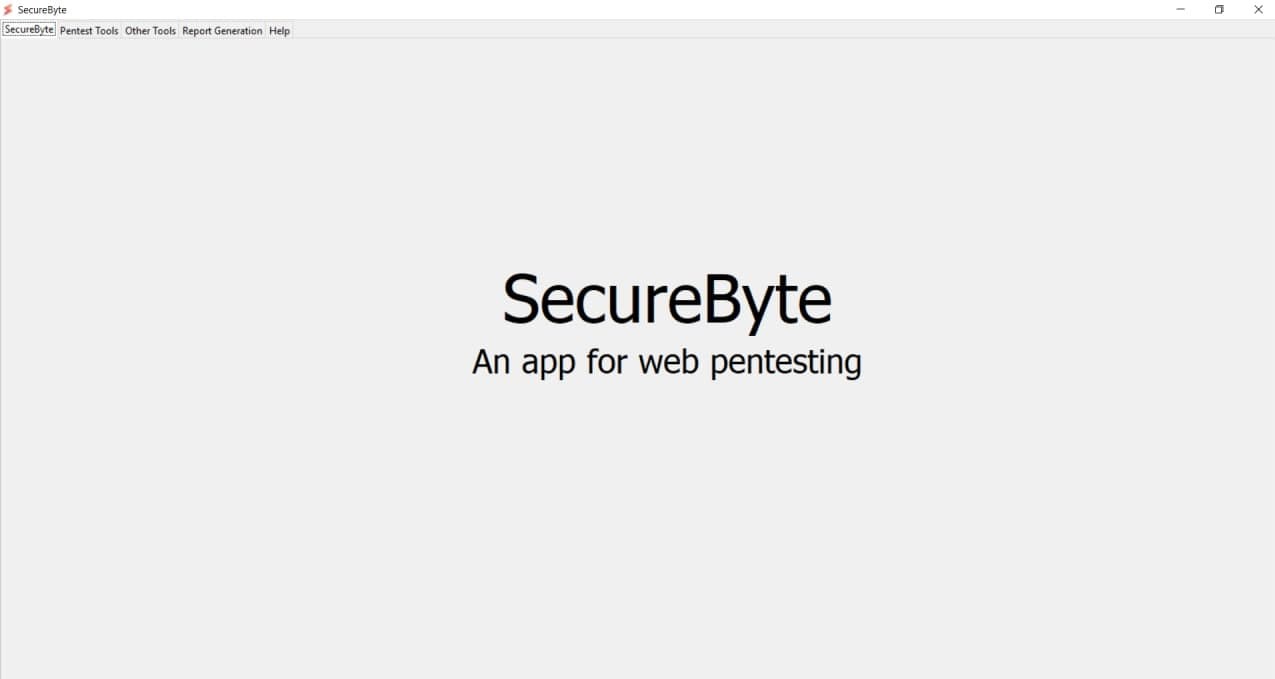


Fig.8 SecureByte 2.0 Welcome Screen

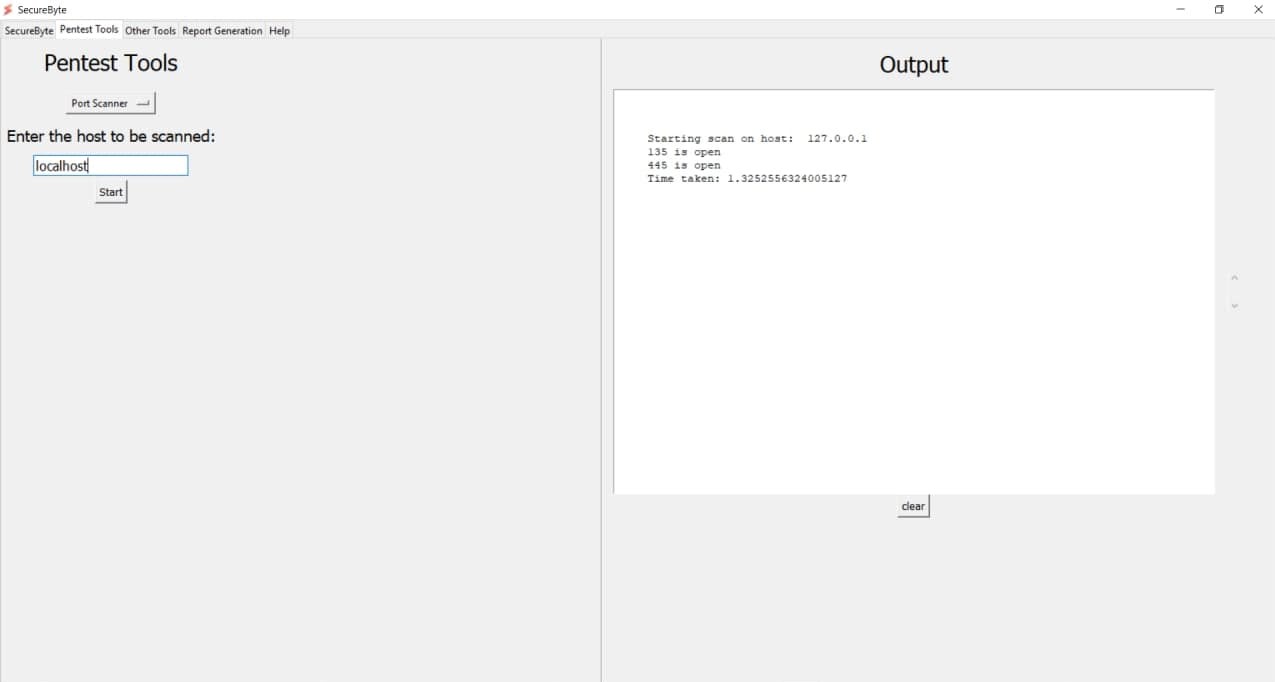


Fig.9 SecureByte 2.0 Port Scanner Input and Output

### A2. SAMPLE CODE

**main.py**

import tkinter as tk

from tkinter import \*

root=Tk()

root.title("SecureByte")

root.geometry("1000x500")

root.configure(bg="white")

root.iconbitmap("logo.ico")

frame1=LabelFrame(root,text="Pentest Tools",font=("Arial",16),padx=680,pady=50)

frame1.grid(row=1,column=1,pady=30)

frame2=LabelFrame(root,text="Other Tools",font=("Arial",16),padx=680,pady=50)

frame2.grid(row=2,column=1,pady=30)

frame3=LabelFrame(root,text="Report",font=("Arial",16),padx=680,pady=50)

frame3.grid(row=3,column=1,pady=30)

options1 = [

"Port Scanner",

"XSS Scanner",

"Directory Brute Forcing",

"Subdomain Scanner",

"SQL Scanner"

]

clicked1 = tk.StringVar()

clicked1.set( "Select the tool" )

def tool1(t):

click1=clicked1.get()

if (click1==options1[0]):

import portscan

elif (click1==options1[1]):

import xssscan

elif (click1==options1[2]):

import directorybruteforcing

elif (click1==options1[3]):

import subdomainscan

elif (click1==options1[4]):

import sqlscan

drop1 = OptionMenu( frame1 , clicked1 , \*options1, command=tool1 )

drop1.grid(row=1,column=0,padx=20,pady=10)

options2 = [

"Hash Creator",

"Brute Hash",

"System Directory Scanner",

"Password Brute Forcing",

"Zip File Cracker"

]

clicked2 = tk.StringVar()

clicked2.set( "Select the tool" )

def tool2(t):

click2=clicked2.get()

if (click2==options2[0]):

import hashcreator

elif (click2==options2[1]):

import brutehash

elif (click2==options2[2]):

import systemdirectory

elif (click2==options2[3]):

import passwordbruteforcing

drop2 = OptionMenu( frame2 , clicked2 , \*options2, command=tool2 )

drop2.grid(row=1,column=0,padx=20,pady=10)

def report():

import report

button=Button(frame3,text=" Report Generation....",command=report)

button.grid(row=1,column=0,padx=20,pady=10)

root.mainloop()

**portscanner.py**

import tkinter as tk

from tkinter import \*

import socket

import time

import threading

from queue import Queue

import sys

root=Tk()

root.title("Port Scanner")

root.geometry("1000x500")

root.iconbitmap("logo.ico")

Label(root,text="Enter the host to be scanned:",font=("Arial",12)).grid(row=1,column=1,padx=10,pady=5)

e1=Entry(root,font=("Arial",12))

e1.grid(row=1,column=2,pady=5)

def portscan():

target = e1.get()

f=open("output.txt",'w')

sys.stdout = f

socket.setdefaulttimeout(0.25)

print\_lock = threading.Lock()

t\_IP = socket.gethostbyname(target)

print ('Starting scan on host: ', t\_IP)

def portscan(port):

s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

try:

con = s.connect((t\_IP, port))

with print\_lock:

print(port, 'is open')

con.close()

except:

pass

def threader():

while True:

worker = q.get()

portscan(worker)

q.task\_done()

q = Queue()

startTime = time.time()

for x in range(100):

t = threading.Thread(target = threader)

t.daemon = True

t.start()

for worker in range(1, 500):

q.put(worker)

q.join()

print('Time taken:', time.time() - startTime)

f.close()

f=open("output.txt",'r').read()

f1 = open("report.txt", "a")

data = f

f1.write(data)

f1.close()

Label(root,text="").grid(row=4,column=3,padx=20,pady=30)

label1=Label(root,text="Output Of Port Scan",font=("Arial 18 underline"))

label1.grid(row=5,column=3,padx=20,pady=10)

#label2=Label(root,text=f,font=("Arial",12))

#label2.grid(row=6,column=3,padx=20,pady=10)

s=Scrollbar(root)

s.grid(row=6,column=4)

t=Text(root,yscrollcommand=s.set,padx=100,pady=50)

t.grid(row=6,column=3)

t.insert(END,data)

s.config(command=t.yview)

def rem():

t.delete("1.0","end")

button1=Button(root,text="clear",command=rem)

button1.grid(row=7,column=3,padx=20)

button=Button(root,text="Start",command=portscan)

button.grid(row=2,column=2,padx=30,pady=10)

root.mainloop()

**xssscanner.py**

import tkinter as tk

from tkinter import \*

import requests

from pprint import pprint

from bs4 import BeautifulSoup as bs

from urllib.parse import urljoin

import sys

root=Tk()

root.title("XSS Scanner")

root.geometry("1000x500")

root.iconbitmap("logo.ico")

Label(root,text="Enter the URL to be scanned:",font=("Arial",12)).grid(row=1,column=1,padx=10,pady=5)

e1=Entry(root,font=("Arial",12))

e1.grid(row=1,column=2,pady=5)

def xss():

url = e1.get()

f=open("output.txt",'w')

sys.stdout=f

def get\_all\_forms(url):

soup = bs(requests.get(url).content, "html.parser")

return soup.find\_all("form")

def get\_form\_details(form):

details = {}

action = form.attrs.get("action").lower()

method = form.attrs.get("method", "get").lower()

inputs = []

for input\_tag in form.find\_all("input"):

input\_type = input\_tag.attrs.get("type", "text")

input\_name = input\_tag.attrs.get("name")

inputs.append({"type": input\_type, "name": input\_name})

details["action"] = action

details["method"] = method

details["inputs"] = inputs

return details

def submit\_form(form\_details, url, value):

target\_url = urljoin(url, form\_details["action"])

inputs = form\_details["inputs"]

data = {}

for input in inputs:

if input["type"] == "text" or input["type"] == "search":

input["value"] = value

input\_name = input.get("name")

input\_value = input.get("value")

if input\_name and input\_value:

data[input\_name] = input\_value

if form\_details["method"] == "post":

return requests.post(target\_url, data=data)

else:

return requests.get(target\_url, params=data)

def scan\_xss(url):

forms = get\_all\_forms(url)

print(f"[+] Detected {len(forms)} forms on {url}.")

js\_script = "<Script>alert('hi')</scripT>"

is\_vulnerable = False

for form in forms:

form\_details = get\_form\_details(form)

content = submit\_form(form\_details, url, js\_script).content.decode()

if js\_script in content:

print(f"[+] XSS Detected on {url}")

print(f"[\*] Form details:")

pprint(form\_details)

is\_vulnerable = True

return is\_vulnerable

print(scan\_xss(url))

f.close()

f=open("output.txt",'r').read()

f1 = open("report.txt", "a")

data = f

f1.write(data)

f1.close()

Label(root,text="").grid(row=4,column=3,padx=20,pady=30)

label1=Label(root,text="Output Of XSS Scan",font=("Arial 18 underline"))

label1.grid(row=5,column=3,padx=20,pady=10)

#label2=Label(root,text=f,font=("Arial",12))

#label2.grid(row=6,column=3,padx=20,pady=10)

s=Scrollbar(root)

s.grid(row=6,column=4)

t=Text(root,yscrollcommand=s.set,padx=100,pady=50)

t.grid(row=6,column=3)

t.insert(END,data)

s.config(command=t.yview)

def rem():

t.delete("1.0","end")

button1=Button(root,text="clear",command=rem)

button1.grid(row=7,column=3,padx=20)

button=Button(root,text="Start",command=xss)

button.grid(row=2,column=2,padx=30,pady=10)

root.mainloop()

**directorybruteforce.py**

import tkinter as tk

from tkinter import \*

import requests

import random

import time

import sys

root=Tk()

root.title("Directory Brute Forcing")

root.geometry("1000x500")

root.iconbitmap("logo.ico")

Label(root,text="Enter the url to scan the website :",font=("Arial",12)).grid(row=1,column=1,padx=10,pady=5)

e1=Entry(root,font=("Arial",12))

e1.grid(row=1,column=2,pady=5)

def directory():

url = e1.get()

wordlist = "common1.txt"

ext = ".php"

f=open("output.txt",'w')

sys.stdout=f

fo = open(wordlist,"r+")

for i in range(30):

word = fo.readline(10).strip()

surl = url+word+ext

response = requests.get(surl)

if (response.status\_code == 200):

print ("[+] found :- ",surl)

else:

print ("[-] Not found :- ",surl)

pass

f.close()

f=open("output.txt",'r').read()

f1 = open("report.txt", "a")

data = f

f1.write(data)

f1.close()

Label(root,text="").grid(row=4,column=3,padx=20,pady=30)

label1=Label(root,text="Output Of Directory Brute Force",font=("Arial 18 underline"))

label1.grid(row=5,column=3,padx=20,pady=10)

s=Scrollbar(root)

s.grid(row=6,column=4)

t=Text(root,yscrollcommand=s.set,padx=100,pady=50)

t.grid(row=6,column=3)

t.insert(END,data)

s.config(command=t.yview)

def rem():

t.delete("1.0","end")

button1=Button(root,text="clear",command=rem)

button1.grid(row=7,column=3,padx=20)

button=Button(root,text="Start",command=directory)

button.grid(row=2,column=2,padx=30,pady=10)

root.mainloop()

**subdomain.py**

import tkinter as tk

from tkinter import \*

import requests

import sys

root=Tk()

root.title("Subdomain Scanner")

root.geometry("1000x500")

root.iconbitmap("logo.ico")

Label(root,text="Enter the domain to scan:",font=("Arial",12)).grid(row=1,column=1,padx=10,pady=5)

e1=Entry(root,font=("Arial",12))

e1.grid(row=1,column=2,pady=5)

def sub():

domain = e1.get()

file = open("subdomains.txt")

content = file.read()

subdomains = content.splitlines()

discovered\_subdomains = []

f=open("output.txt",'w')

sys.stdout=f

for subdomain in subdomains:

url = f"http://{subdomain}.{domain}"

try:

requests.get(url)

except requests.ConnectionError:

pass

else:

print("[+] Discovered subdomain:", url)

discovered\_subdomains.append(url)

f.close()

f=open("output.txt",'r').read()

f1 = open("report.txt", "a")

data = f

f1.write(data)

f1.close()

Label(root,text="").grid(row=4,column=3,padx=20,pady=30)

label1=Label(root,text="Output Of Subdomain Scan",font=("Arial 18 underline"))

label1.grid(row=5,column=3,padx=20,pady=10)

#label2=Label(root,text=f,font=("Arial",12))

#label2.grid(row=6,column=3,padx=20,pady=10)

s=Scrollbar(root)

s.grid(row=6,column=4)

t=Text(root,yscrollcommand=s.set,padx=100,pady=50)

t.grid(row=6,column=3)

t.insert(END,data)

s.config(command=t.yview)

def rem():

t.delete("1.0","end")

button1=Button(root,text="clear",command=rem)

button1.grid(row=7,column=3,padx=20)

button=Button(root,text="Start",command=sub)

button.grid(row=2,column=2,padx=30,pady=10)

root.mainloop()

**sqlscanner.py**

import tkinter as tk

from tkinter import \*

import requests

from pprint import pprint

from bs4 import BeautifulSoup as bs

from urllib.parse import urljoin

import sys

root=Tk()

root.title("SQL Scanner")

root.geometry("1000x500")

root.iconbitmap("logo.ico")

Label(root,text="Enter the URL to be scanned:",font=("Arial",12)).grid(row=1,column=1,padx=10,pady=5)

e1=Entry(root,font=("Arial",12))

e1.grid(row=1,column=2,pady=5)

def sql():

url = e1.get()

s = requests.Session()

s.headers["User-Agent"] = "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.106 Safari/537.36"

f=open("output.txt",'w')

sys.stdout=f

def get\_all\_forms(url):

soup = bs(s.get(url).content, "html.parser")

return soup.find\_all("form")

def get\_form\_details(form):

details = {}

try:

action = form.attrs.get("action").lower()

except:

action = None

method = form.attrs.get("method", "get").lower()

inputs = []

for input\_tag in form.find\_all("input"):

input\_type = input\_tag.attrs.get("type", "text")

input\_name = input\_tag.attrs.get("name")

input\_value = input\_tag.attrs.get("value", "")

inputs.append({"type": input\_type, "name": input\_name, "value": input\_value})

details["action"] = action

details["method"] = method

details["inputs"] = inputs

return details

def is\_vulnerable(response):

errors = {

"you have an error in your sql syntax;",

"warning: mysql",

"unclosed quotation mark after the character string",

"quoted string not properly terminated",

}

for error in errors:

if error in response.content.decode().lower():

return True

return False

def scan\_sql\_injection(url):

for c in "\"'":

new\_url = f"{url}{c}"

print("[!] Trying", new\_url)

res = s.get(new\_url)

if is\_vulnerable(res):

print("[+] SQL Injection vulnerability detected, link:", new\_url)

return

forms = get\_all\_forms(url)

print(f"[+] Detected {len(forms)} forms on {url}.")

for form in forms:

form\_details = get\_form\_details(form)

for c in "\"'":

data = {}

for input\_tag in form\_details["inputs"]:

if input\_tag["type"] == "hidden" or input\_tag["value"]:

try:

data[input\_tag["name"]] = input\_tag["value"] + c

except:

pass

elif input\_tag["type"] != "submit":

data[input\_tag["name"]] = f"test{c}"

url = urljoin(url, form\_details["action"])

if form\_details["method"] == "post":

res = s.post(url, data=data)

elif form\_details["method"] == "get":

res = s.get(url, params=data)

if is\_vulnerable(res):

print("[+] SQL Injection vulnerability detected, link:", url)

print("[+] Form:")

pprint(form\_details)

break

scan\_sql\_injection(url)

f.close()

f=open("output.txt",'r').read()

f1 = open("report.txt", "a")

data = f

f1.write(data)

f1.close()

Label(root,text="").grid(row=4,column=3,padx=20,pady=30)

label1=Label(root,text="Output Of SQL Scan",font=("Arial 18 underline"))

label1.grid(row=5,column=3,padx=20,pady=10)

#label2=Label(root,text=f,font=("Arial",12))

#label2.grid(row=6,column=3,padx=20,pady=10)

s=Scrollbar(root)

s.grid(row=6,column=4)

t=Text(root,yscrollcommand=s.set,padx=100,pady=50)

t.grid(row=6,column=3)

t.insert(END,data)

s.config(command=t.yview)

def rem():

t.delete("1.0","end")

button1=Button(root,text="clear",command=rem)

button1.grid(row=7,column=3,padx=20)

button=Button(root,text="Start",command=sql)

button.grid(row=2,column=2,padx=30,pady=10)

root.mainloop()

**hashcreator.py**

import tkinter as tk

from tkinter import \*

import hashlib, bcrypt

import sys

root=Tk()

root.title("Hash Creator")

root.geometry("1000x500")

root.iconbitmap("logo.ico")

Label(root,text="Enter the password that encoded into hash:",font=("Arial",12)).grid(row=1,column=1,padx=10,pady=5)

e1=Entry(root,font=("Arial",12))

e1.grid(row=1,column=2,pady=5)

def hash():

password = e1.get()

f=open("output.txt",'w')

sys.stdout=f

print("\nSHA1:\n")

for i in range(1):

setpass = bytes(password, 'utf-8')

hash\_object = hashlib.sha1(setpass)

guess\_pw = hash\_object.hexdigest()

print(guess\_pw)

print("\nMD5:\n")

for i in range(1):

setpass = bytes(password, 'utf-8')

hash\_object = hashlib.md5(setpass)

guess\_pw = hash\_object.hexdigest()

print(guess\_pw)

print("\nBCRYPT:\n")

for i in range(1):

hashed = bcrypt.hashpw(setpass, bcrypt.gensalt(10))

print(hashed)

f.close()

f=open("output.txt",'r').read()

f1 = open("report.txt", "a")

data = f

f1.write(data)

f1.close()

Label(root,text="").grid(row=4,column=3,padx=20,pady=30)

label1=Label(root,text="Output Of Hash Creator",font=("Arial 18 underline"))

label1.grid(row=5,column=3,padx=20,pady=10)

s=Scrollbar(root)

s.grid(row=6,column=4)

t=Text(root,yscrollcommand=s.set,padx=100,pady=50)

t.grid(row=6,column=3)

t.insert(END,data)

s.config(command=t.yview)

def rem():

t.delete("1.0","end")

button1=Button(root,text="clear",command=rem)

button1.grid(row=7,column=3,padx=20)

button=Button(root,text="Start",command=hash)

button.grid(row=2,column=2,padx=30,pady=10)

root.mainloop()

**brutehash.py**

import tkinter as tk

from tkinter import \*

from urllib.request import urlopen, hashlib

import sys

root=Tk()

root.title("Brute Hash")

root.geometry("1000x500")

root.iconbitmap("logo.ico")

Label(root,text="Enter the hash that decoded into password:",font=("Arial",12)).grid(row=1,column=1,padx=10,pady=5)

e1=Entry(root,font=("Arial",12))

e1.grid(row=1,column=2,pady=5)

def brute():

sha1hash = e1.get()

.0

f=open("output.txt",'w')

sys.stdout=f

LIST\_OF\_COMMON\_PASSWORDS = str(urlopen('https://raw.githubusercontent.com/danielmiessler/SecLists/master/Passwords/Common-Credentials/10-million-password-list-top-10000.txt').read(), 'utf-8')

for guess in LIST\_OF\_COMMON\_PASSWORDS.split('\n'):

hashedGuess = hashlib.sha1(bytes(guess, 'utf-8')).hexdigest()

if hashedGuess == sha1hash:

print("The password is ", str(guess))

break

elif hashedGuess != sha1hash:

print("Password guess ",str(guess)," does not match, trying next...")

else:

print("Password not in database, we'll get them next time.")

f.close()

f=open("output.txt",'r').read()

f1 = open("report.txt", "a")

data = f

f1.write(data)

f1.close()

Label(root,text="").grid(row=4,column=3,padx=20,pady=30)

label1=Label(root,text="Output Of Brute Hash",font=("Arial 18 underline"))

label1.grid(row=5,column=3,padx=20,pady=10)

s=Scrollbar(root)

s.grid(row=6,column=4)

t=Text(root,yscrollcommand=s.set,padx=100,pady=50)

t.grid(row=6,column=3)

t.insert(END,data)

s.config(command=t.yview)

def rem():

t.delete("1.0","end")

button1=Button(root,text="clear",command=rem)

button1.grid(row=7,column=3,padx=20)

button=Button(root,text="Start",command=brute)

button.grid(row=2,column=2,padx=30,pady=10)

root.mainloop()

**systemdirectory.py**

import tkinter as tk

from tkinter import \*

import os

import sys

root=Tk()

root.title("System Directory Scanner")

root.geometry("1000x500")

root.iconbitmap("logo.ico")

Label(root,text="Enter the path to scan directories in the system:",font=("Arial",12)).grid(row=1,column=1,padx=10,pady=5)

e1=Entry(root,font=("Arial",12))

e1.grid(row=1,column=2,pady=5)

def Sys\_Dir():

directory = e1.get()

f=open("output.txt",'w')

sys.stdout=f

for entry in os.scandir(directory):

if entry.is\_dir():

typ = 'dir'

elif entry.is\_file():

typ = 'file'

elif entry.is\_symlink():

typ = 'link'

else:

typ = 'unknown'

print('{name} {typ}'.format(

name=entry.name,

typ=typ,

))

f.close()

f=open("output.txt",'r').read()

f1 = open("report.txt", "a")

data = f

f1.write(data)

f1.close()

Label(root,text="").grid(row=4,column=3,padx=20,pady=30)

label1=Label(root,text="Output Of System Directory Scan",font=("Arial 18 underline"))

label1.grid(row=5,column=3,padx=20,pady=10)

s=Scrollbar(root)

s.grid(row=6,column=4)

t=Text(root,yscrollcommand=s.set,padx=100,pady=50)

t.grid(row=6,column=3)

t.insert(END,data)

s.config(command=t.yview)

def rem():

t.delete("1.0","end")

button1=Button(root,text="clear",command=rem)

button1.grid(row=7,column=3,padx=20)

button=Button(root,text="Start",command=Sys\_Dir)

button.grid(row=2,column=2,padx=30,pady=10)

root.mainloop()

**passwordbruteforcing.py**

import tkinter as tk

from tkinter import \*

import random

import string

import sys

root=Tk()

root.title("Password Brute Forcing")

root.geometry("1000x500")

root.iconbitmap("logo.ico")

Label(root,text="Enter the Password to crack:",font=("Arial",12)).grid(row=1,column=1,padx=10,pady=5)

chars=string.printable

chars\_list=list(chars)

e1=Entry(root,show="\*",font=("Arial",12))

e1.grid(row=1,column=2,pady=5)

def password():

password=e1.get()

guess\_password=""

f=open("output.txt",'w')

sys.stdout=f

while(guess\_password!=password):

guess\_password=random.choices(chars\_list,k=len(password))

#print("<=========="+str(guess\_password)+"===========>")

if(guess\_password==list(password)):

print("Your Password is : "+ "".join(guess\_password))

break

f.close()

f=open("output.txt",'r').read()

f1 = open("report.txt", "a")

data = f

f1.write(data)

f1.close()

Label(root,text="").grid(row=4,column=3,padx=20,pady=30)

label1=Label(root,text="Output Of Password Brute Forcing",font=("Arial 18 underline"))

label1.grid(row=5,column=3,padx=20,pady=10)

s=Scrollbar(root)

s.grid(row=6,column=4)

t=Text(root,yscrollcommand=s.set,padx=100,pady=50)

t.grid(row=6,column=3)

t.insert(END,data)

s.config(command=t.yview)

def rem():

t.delete("1.0","end")

button1=Button(root,text="clear",command=rem)

button1.grid(row=7,column=3,padx=20)

button=Button(root,text="Start",command=password)

button.grid(row=2,column=2,padx=30,pady=10)

root.mainloop()

**REFERENCES**

1. Mei, J. (2009). An approach for SQL injection vulner-ability detection. In Proceedings of the SixthInternational Conference on InformationTechnology.
2. M. Khari, Sonam, Vaishali And M. Kumar, "Comprehensive Study Of Web Application Attacks And Classification," 2016 3rd International Conference On Computing For Sustainable Global Development (Indiacom), New Delhi, 2016, Pp. 2159-2164.
3. Jose Fonseca, Marco Vieira, And Henrique Madeira, "Evaluation Of Web Security Mechanisms Using Vulnerability & Attack Injection", Dependable And Secure Computing, Ieee Transactions (Volume:11, Issue: 5)
4. https://simplysecure.blog/2017/07/05/FIVE-PHASES-OFPENETRATION-TESTING
5. K. Nirmal, B. Janet And R. Kumar, "Web Application Vulnerabilities - The Hacker's Treasure," 2018 International Conference On Inventive Research In Computing Applications (Icirca), Coimbatore, India, 2018, Pp. 58-62.

Padmaja K,"A Study On Web Application And Protection Against Vulnerability", In International Journal Of Engineering Research And Application, (Ijera),2012, Pp.001-006.

1. "Security Code Review-Identifying Web Vulnerabilities", By Kiran Maraju.
2. M.Khari And N.Kumar, "User Authentication Method Against Sql Injection Attack", International Journal Of Scientific And Engineering Research,2013, Pp. 1649-1653.
3. https://Hackernoon.Com/Timing-Based-Blind-Sql-AttacksBd276dc618dd
4. Guo, F., Yu, Y., & Chiueh, T. (2005). Automated and safe vulnerability assessment. In Proceedings of the 21st Annual Computer Security Applications Conference.
5. Fonseca, J., Vieira, M., & Madeira, H. (2007). Testing and comparing Web vulnerability scanning tools for SQL injection and XSS attacks. In Proceedings of the 13th PacificRim International Symposium on Depend-able Computing (pp. 365-372).
6. Sanfilippo, S. “Hping – Active Network Security Tool,” http://www.hping.org/, accessed on Nov. 23, 2011.
7. Superscan, http://www.mcafee.com/us/downloads/free-tools/superscan.aspx, accessed on Nov. 23, 2011.
8. Xprobe2, http://www.net-security.org/software.php?id=231, accessed on Nov. 23, 2011. International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.6, November 2011 38
9. P0f, http://www.net-security.org/software.php?id=164, accessed on Nov. 23, 2011.
10. Httprint, http://net-square.com/httprint/, accessed on Nov. 23, 2011.
11. Nessus, http://www.tenable.com/products/nessus, accessed on Nov. 23, 2011.
12. Shadow Security Scanner, http://www.safety-lab.com/en/download.htm, accessed on Nov. 23, 2011.
13. Iss Scanner, http://shareme.com/showtop/freeware/iss-scanner.html, accessed on Nov. 23, 2011.
14. GFI LAN guard, http://www.gfi.com/network-security-vulnerability-scanner, accessed on Nov. 23, 2011.
15. Brutus, http://download.cnet.com/Brutus/3000-2344\_4-10455770.html, accessed on Nov. 23, 2011.
16. MetaSploit, http://www.metasploit.com/, accessed on Nov. 23, 2011.
17. Skoudis, E. “Powerful Payloads: The Evolution of Exploit Frameworks,” (2005). http://searchsecurity.techtarget.com/news/1135581/Powerful-payloads-The-evolution-of-exploitframeworks, accessed on Nov. 23, 2011.
18. Andreu, A. (2006). Professional Pen Testing for Web Applications. Wrox publisher, 1st edition.
19. OWASP. “Web Application Penetration Testing,” http://www.owasp.org/index.php/Web\_Application\_Penetration\_Testing, accessed on Nov. 23, 2011.
20. Fiddler, http://www.fiddler2.com/fiddler2, accessed on Nov. 23, 2011.
21. Stuttard, D. and Pinto, M. (2008) The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws,, Wiley. 1st edition.
22. “White Paper on Penetration Testing,” http://www.docstoc.com/docs/70280500/White-Paper-onPenetration-Testing, accessed on Nov. 23, 2011.
23. Neumann, P. (1977) “Computer System Security Evaluation,” Proceedings of AFIPS 1977 Natl. Computer Conf., Vol. 46, pp. 1087-1095.
24. Pfleeger, C. P., Pfleeger, S. L., and Theofanos, M. F. (1989) “A Methodology for Penetration Testing,” Computers & Security, 8(1989) pp. 613-620.
25. Bishop, M. (2007) “About Penetration Testing,” IEEE Security & Privacy, November/December 2007, pp. 84-87. [31]Arkin, B., Stender, S., and McGraw, G. “Software Penetration Testing,” IEEE Security & Privacy, January / February 2005, pp. 32-35.