Presented By Logapriya S

Cyber security project

ABSTRACT

The reconnaissance phase is the longest, tedious, arduous phase in the cyber kill chain. The project is developing a tool for clubbing all individual processes into one, an all-in-one, tool. This makes a long phase easy and the information retrieved is un- changed. A simple Recon Tool that automates your favorite tool sets. The important must-have tools to run this tool are Subfinder, httprobe, aqua tone, subzy, Wayback- urls, gf, Gf-Patterns, sqlmap, replace, go buster.The need for Recon automation is rapidly increasing as ethical hackers are being lazy in performing every little check manually. So as to make the Recon process (Info gathering phase) of penetration test- ing easy, fast and accurate.we introduced the "HACKNOW" tool which is written in bash script. It helps for better recon to find bugs and reduce the time consumption compare to other tools.

Keywords: Recon,Automation tool,infosec,cyber kill chain,Target,Attack

LIST OF FIGURES

17. **poster presentation**35

LIST OF ACRONYMS AND ABBREVIATIONS

infosecInformation Security.

CVECommon Vulnerabilities Exposure. WAFWeb Application Firewall.

IDSIntrusion Detection System

RAMRandom-Access Memory

ISMSInformation security management system OSOperating System

APIApplication Programming Interface

TABLE OF CONTENTS

Page.No

2.
3.
4.
    1.
    2.
    3.

4. **METHODOLOGY8**
    1.

Chapter 1 INTRODUCTION

1. Introduction

The Cyber Kill Chain offers a comprehensive framework as a part of the Intel- ligence Driven Defense model. The Cyber Kill Chain consists of 7 steps: Recon- naissance, weaponization, delivery, exploitation, installation, command and control, and finally, actions on objectives. The main objective is the Reconnaissance phase, where the tedious process of mapping and gaining information about the given tar- get has to be retrieved.Then we conduct in-depth research on the target to identify its vulnerabilities that can be exploited.HACKNOW will deal with all the processes such as active port discovery, traceroute discovery, subdomain discovery, e-mail dis- covery of the domain, maltegio network mapping, operating system fingerprinting, checking for active intrusion detection, and prevention systems, version of software running.

1. Aim of the Project

There are many things that every Penetration Tester and Bug Bounty Hunter does during blackbox testing of web application. These repetitive things cost a lot of time during penetration testing, and the time is usually short. Facing these obstacles, We have created a tool that automates many activities and increases work efficiency the aim of the project is to Shorten the steps and process time of the reconnaissance phase.

1. Project Domain

This project falls under the domain of Cyber security , which is the application of technologies, processes and controls to protect systems, networks, programs, devices and data from cyber attacks.so this project comes under cyber security domain.

1. Scope of the Project

The developed tool eases the burden of the penetration tester and maintains an equilibrium between the recon phase and the remaining stages. Automating this phase would result in more precise results than doing the process manually.

Chapter 2 LITERATURE REVIEW

1. In this paper,Wojciech Mazurczyk, 2021, “Cyber Reconnaissance Techniques”, Association for Computing Machinery:As a general trend, the evolution of smart devices, social media, and IoT-capable applications boosted the amount of information that can be gathered by an attacker and also multiplied the communications paths that can be used to reach the victim. Therefore, the potential attack surface exploitable for reconnaissance techniques is expected to continue to grow,at least shortly.
2. Johannes K Becker, 2021, “Network Reconnaissance for the Internet of Things”, Network reconnaissance is a core networking and security procedure aimed at discovering devices and their properties. For IP-based networks, several network reconnaissance tools are available, such as Nmap.We make our implementation and data available to the research community to allow independent replication of our results and facilitate further development of the tool.
3. Amit Jain, 2021, “Cyber Security Threats And Their Solutions Through Deep Learning”, IEEE. Although in the recent times there have been booming advancements in the field of technology, but it has also brought along various risks includ- ing malware attacks, cloud abuse hacking, insecure API, cyber scams, data breach and so on.Cybersecurity has become an urgent need. Understanding this need, deep learning methods are being used by cybersecurity experts for dealing with various problems foisted by the cyber criminals. This paper reviews various threats in the path of cyber security and it also brings to light various deep learning methods that have been employed to combat with these threats.
4. Ikram ul haq, 2020, “Penetration Frameworks and Development Issues in Se- cure Mobile Application Development”, IEEE This paper presents a comprehensive survey of different penetration tools, evaluated by using criteria such as code anal-

ysis, code review, vulnerability analysis, vulnerability exploit, payload and whether these can be used in vulnerability modeling during the design phase. Our study effectively identifies the issues and gaps which can further help develop a frame- work/tool for

designing a penetration secure mobile application by embedding all the vulnerabilities during the design phase using an android vulnerability repository.

1. Ruhma Sardar, 2021, “Web of Things: Security Challenges and Mechanisms”, IEEE.This paper presents and discussed some of the available mechanisms to overcome security related issues while taking into account the network size and mobility. Authors have used Threat analysis and attack modeling methods to inform the users about defensive measures and to prevent security threats from taking advantage of system flaws Authors have provided the necessary insight into how security can be improved by using certain existing mechanisms and algorithms. .

Chapter 3 PROJECT DESCRIPTION

1. Existing System

Existing Technology of automatic recon for web application is they don’t provide output files after completion of scanning so that we can’t able to see the scan re- sult whenever we want it.And also we need to use different tools for every unique functions of recon.

1. Proposed System

Our Tool HACKNOW is the advanced recon automation tool which provide out- put after the scan completion and able to scan faster because it is written in three different programming language (python,go,bash).

Advantages

1. Improved accuracy .
2. Easy to implement.
3. Helps to reduce cyber crimes.
4. Economically feasible.

1. Feasibility Study

1. **Economic Feasibility**

Cost-benefit analysis of this project, which assesses that it is possible to imple- ment in real time with less cost.This analysis takes into consideration of the cost of both developing and operating the new venture of the project.

1. **Technical Feasibility**

The project process of assessing all the technically possible to manufacture a this product .It also involves the evaluation of the hardware, software, and other technical requirements of the proposed system.

1. **Social Feasibility**

All the levels of cyber security people can able interact with this project.The reach of this tool make a social impact in website security.

1. System Specification

1. **Hardware Specification**
   - Min of 4GB RAM
   - Min of atleast i3 Processor.

1. **Software Specification**
   - Operating System - Kali Linux
   - Programming languages - python3,golang,bash
   - Perquisites tools need to be install: subfinder,httprobe,aquatone,subzy,Waybackurls,gf,Gf Patterns,sqlmap,qsrepalce,gobuster
2. **Standards and Policies**
   - ISO/IEC 27001 - The international standard that describes the requirements for an ISMS (information security management system)
   - ISO 27032 - The international standard offering guidance on cybersecurity management. It provides guidance on addressing a wide range of

cybersecurity risks, including user endpoint security, network security, and critical infrastruc- ture protection.

- ISO 22301:2012 provides a best-practice framework for implementing an optimized BCMS (business continuity management system)

Chapter 4 METHODOLOGY

1. General Architecture

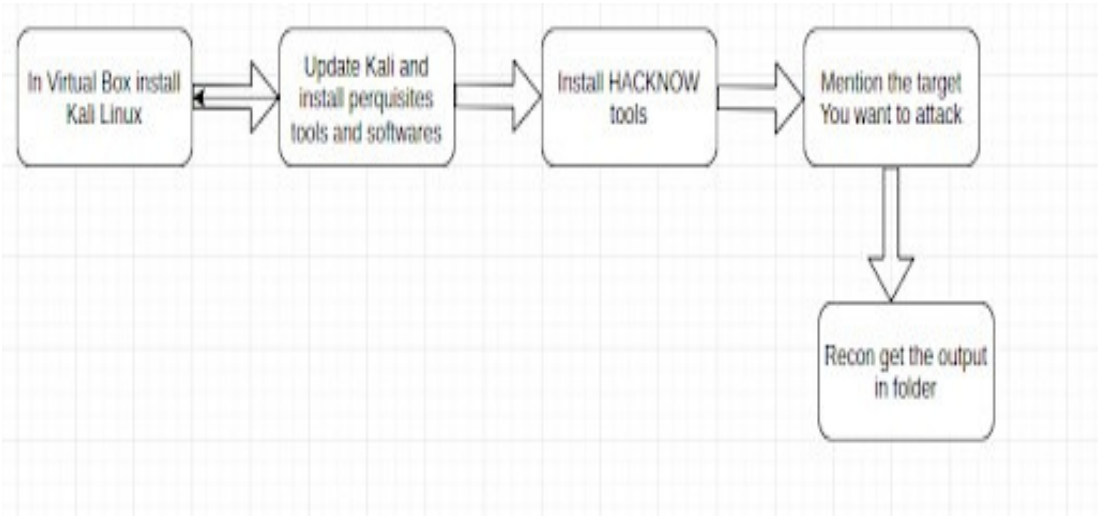

Figure 4.1: **Architecture Diagram**

The architecture of the project is displayed in Figure 4.1.This architectural dia- gram shows the visual representation that maps out the physical implementation for components of a software system. It shows the general structure of the software sys- tem and the associations, limitations, and boundaries between each element of the project.

1. Design Phase

1. Data Flow Diagram

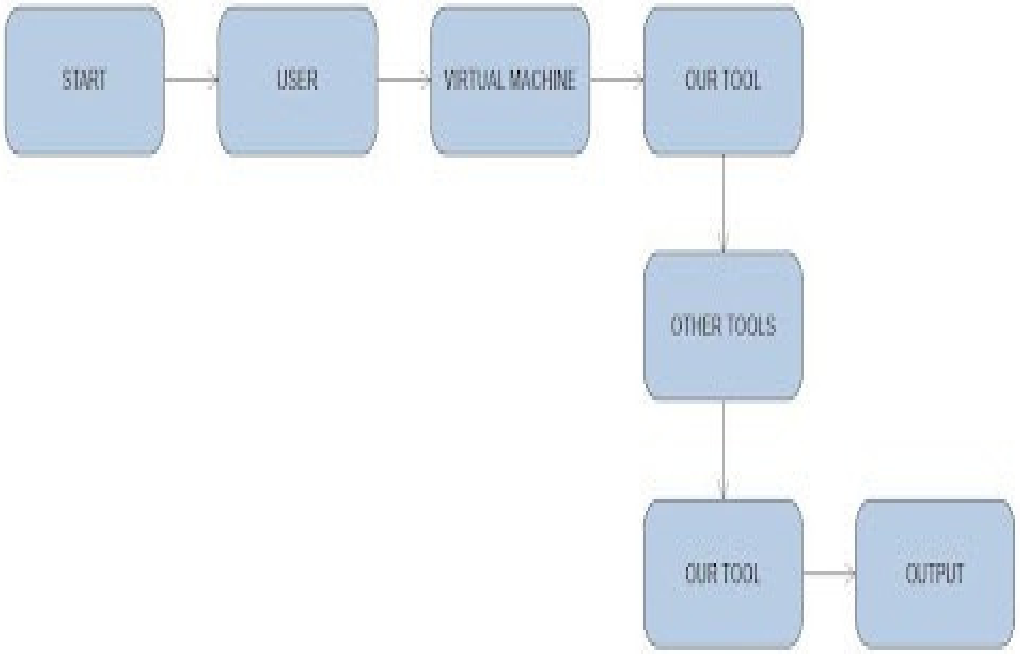

Figure 4.2: **Data Flow Diagram**

The data flow diagram is presented in Figure 4.2. The DFD provides information about the outputs and inputs of each entity and the process of the project. A data-flow diagram has no control flow — there are no decision rules and no loops.

1. **Use Case Diagram**

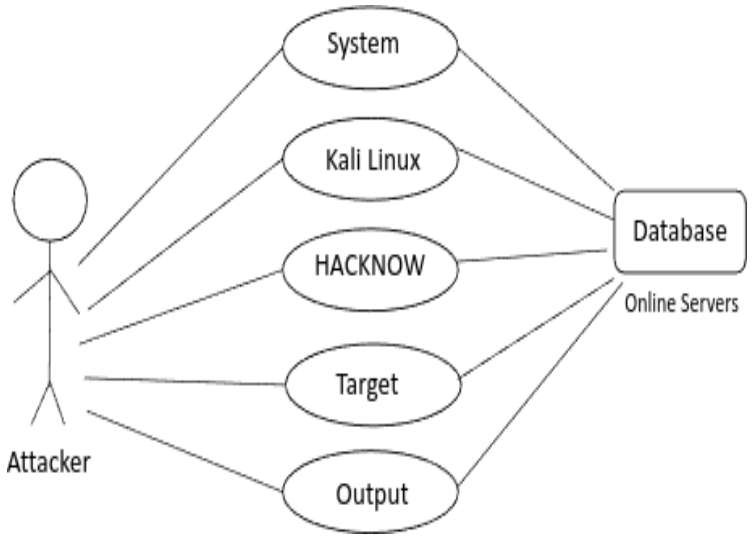

Figure 4.3: **Use Case Diagram**

The Use case diagram is depicted in Figure 4.3. Use case diagram shows how to represent the dynamic behavior of the project system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application.

1. **Class Diagram**

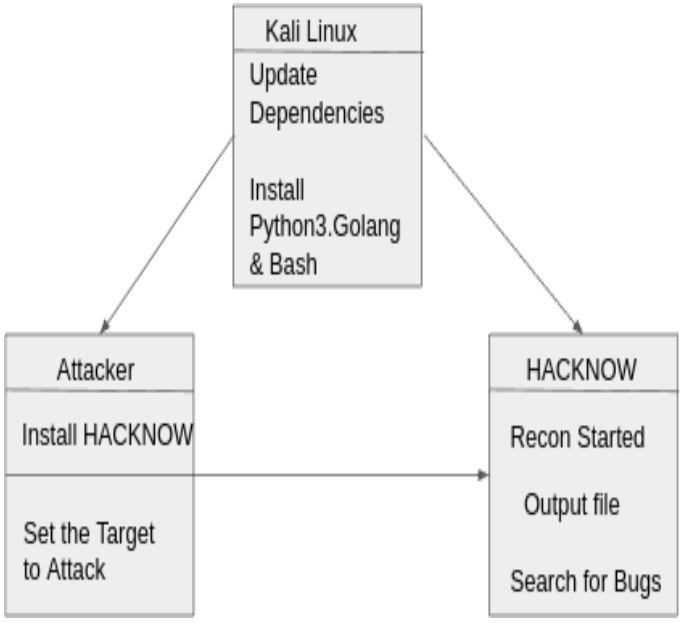

Figure 4.4: **Class Diagram**

The class diagram of this project is presented in the Figure 4.4.Class diagram de- scribes the attributes and operations of a class and also the constraints imposed on the project system. The class diagrams are widely used in the modeling of objec- toriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

1. **Sequence Diagram**

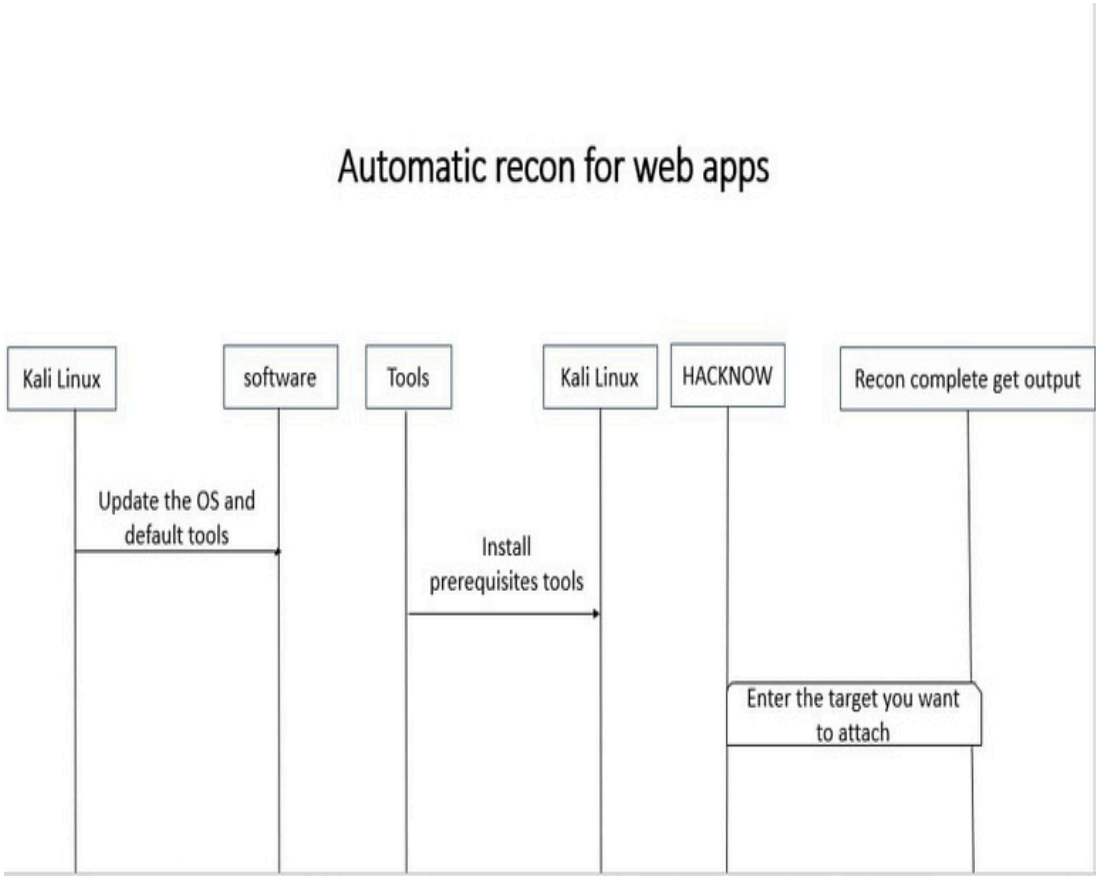

Figure 4.5: **Sequence Diagram**

The sequence diagram of this project is depicted in the Figure 4.5. Sequence diagram shows the type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams is used for software developers and business professionals to understand requirements for a new system or to document an existing process.

1. **Collaboration diagram**

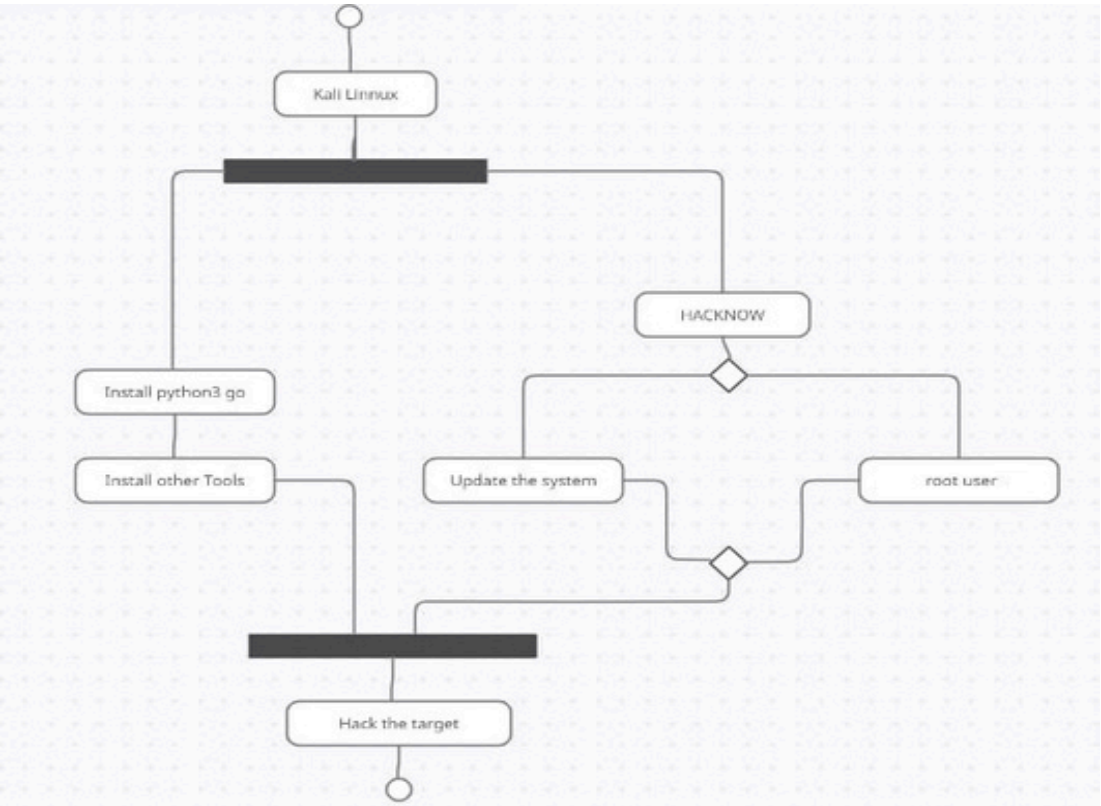

Figure 4.6: **Collaboration Diagram**

Description:

The collaboration diagram of this project is depicted in the Figure 4.6.Collaboration diagrams, also known as communication diagrams, show the relations and the interactions between software objects in the Unified Modeling Language (UML). These diagrams are used to show the dynamic behavior of a

particular use case and define the role of each object.

1. **Activity Diagram**

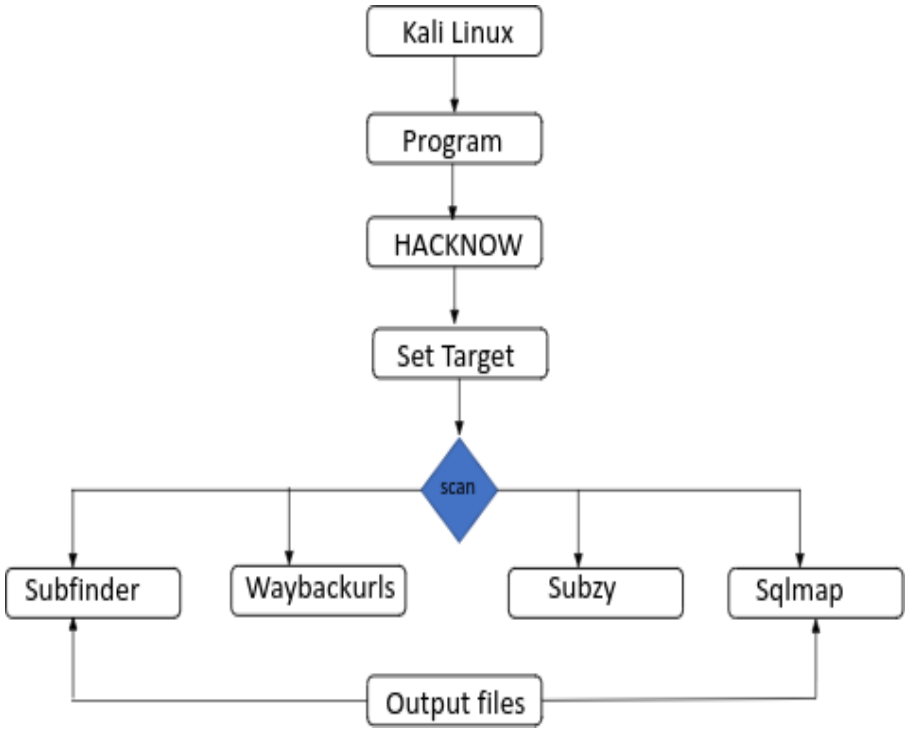

Figure 4.7: **Activity Diagram**

The Activity diagram of this project is depicted in the Figure 4.7 Activity diagram shows a series of actions or flow controls in a system such as a flow chart or data flow diagram. Job drawings are often used to simulate a business process. They can also explain the steps in the application case diagram. The functions performed by the model can be sequential and coherent. In both cases the diagram of the activity will have a beginning (first position) and an end (final state).

1. Algorithm & Pseudo Code

1. Algorithm

Algorithm Followed during project development Step 1: Start

Step 2: Configure HACKNOW in Kali Linux. **Step 3:** Add root permission to HACKNOW. **Step 4:** Declare the target you want to scan **Step 5:** Store the output file as target name.

Step 6: Search for vulnerabilities.

Step 7: Stop

1. **Pseudo Code Pseudocode of our Project**

WebRecon(Target) HACKNOW = Target Output=file.Targetname END FOR

DISPLAY Output END Procedure

1. Module Description

1. **Setting up the Requirements**
   - Configuring the setup to run the HACKNOW tool is the foremost thing to do
   - Hardware and Software requirements are analysed.

Figure 4.8: **Configuring Kali Linux**

1. **Install the perquisite tools**
   - For installing perquisite tools first we need to install python3,golang and bash

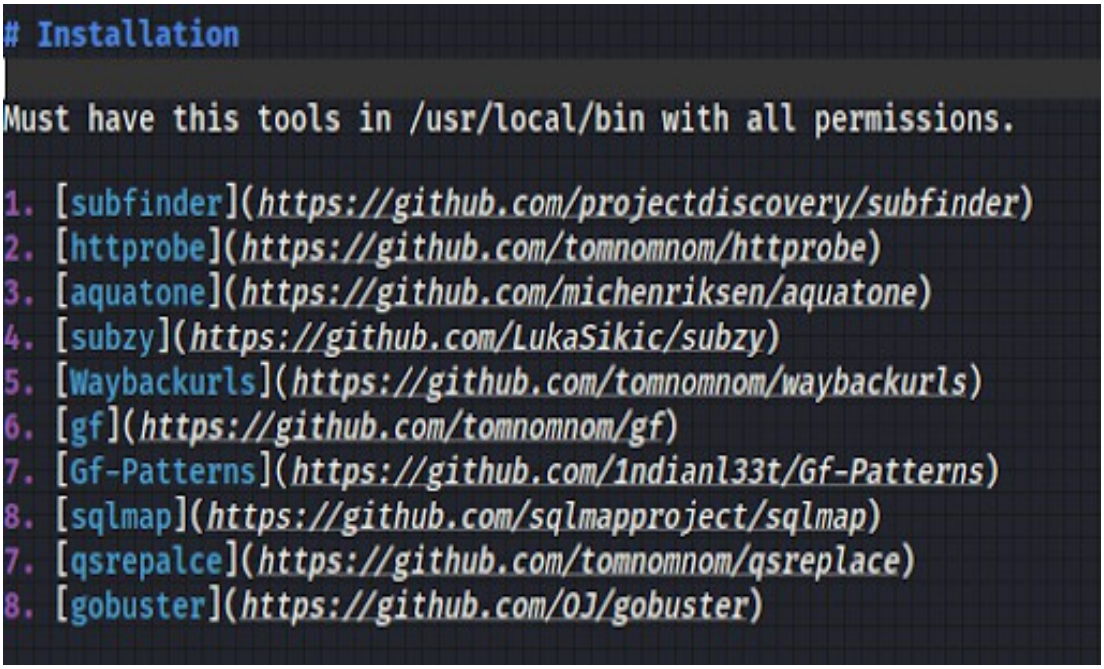

Figure 4.9: **Perquisite tools to install**

1. **Run HACKNOW**
   - Run HACKNOW tool with root permission.

1. Steps to execute/run/implement the project

1. **Step1**

Install Kali Linux OS and Download Install the perquisites softwares and tools

1. **Step2**

Run the tool Hacknow toolwith root permission

1. **Step3**

Output of the target in folder

Chapter 5 IMPLEMENTATION AND TESTING

1. Input and Output

1. **Input Design**

The input for this implementation is to configure HACKNOW in Kali Linux OS than Add root permission to HACKNOW. Atlast declare the target you want to scan

1. **Output Design**

The output design is done where tool store the output folder as target name.Now start to search for bugs.Inside the output folder their will be files which contains more informations about the target subdomains,javascript endpoints,inject-able parameters etc.

1. Testing

The main reason for doing testing is to find mistakes. Testing is a way towards attempt to find the each possible shortcoming in a work item. It gives an approach to check the usefulness of segments, sub-gatherings, congregations as well as a com- pleted item It is the way toward practicing programming with the goal of guarantee- ing that the Project lives up to its necessities and requirements and to make sure that it doesn't flop in a inadmissible way.

1. Types of Testing

1. Unit testing

In unit testing, every block of code is tested independently such that the cases are passed. Every block of code is tested for all boundary conditions which are possible for the project. The blocks are not combined and blocks that require calls to other blocks are combined into a single unit and tested with and without providing inputs according to the requirement of each block.

Input

1

2

3

4

Test result

In this project, unit testing is carried out for each and every module in our project. The results of the testing for each and every module has satisfied all the requirements of the project.

1. **Integration testing**

Integration tests are designed to test integrated software components to determine if they run as one program. Testing is event-driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfied, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifi- cally aimed at exposing the problems that arise from the combination of components.

Input

1

2

3

Test result

All the software modules are integrated logically and tested as a group success- fully and it satisfied all the test cases.

1. **System testing**

System tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Input

1. **Test Result**
2. The system works effectively and it meets all the essential requirements

Chapter 6

RESULTS AND DISCUSSIONS

1. Efficiency of the Proposed System

The Proposed System of the project is to Shorten the steps and process time of the reconnaissance phase.Using this tool we can execute different types of Recon- naissance feature in short period of time Because the tools are developed in python and golang which integrated with bash scripts.

1. Comparison of Existing and Proposed System

In Existing System , For Reconnaissance ethical hackers use different tools for each functionality testing which is mostly written in java which is not much sup- ported by Linux environment and also the execution time increases.To reduce the above factors we build this tool ”HACKNOW” it is used to automate your recon process with more using results.

1. Sample Code

t

Chapter 7 CONCLUSION AND FUTURE ENHANCEMENTS

1. **Conclusion**

The main aim of this project is to develop a tool that envelope many stages/parts of the reconnaissance phase that decides the implementation of the hacking procedure and speeds up the process. It's like creating an API in developing software. The idea is simple and elegant, and the cyber kill chain will be rendered faster and easier, and many hackers don't have to waste so much time gathering information on their target, instead can focus on what to do next. That is the main objective of this project. Thus decreasing the labor of the cyber kill chain by automating the Recon process.

1. Future Enhancements

Future Enhancements of the project we are going take the results from HAC- KNOW tool and use exploitdb.com api to find the vulnerabilities CVE of the target automatically. And also we will add a feature for automatically generating the scan report after finding the vulnerabilities form expliotdb.

Chapter 8 INDUSTRY DETAILS

1. Industry name

PRODIGY INFOTECH

1. **Duration of Internship (From Date - To Date)**

01.08.2024 to 31.08.2024

1. **Duration of Internship in months**

1 Month

1. **Internship offer letter**

PRODIGY INFOTECH

INTERNSHIP OFFER LETTER

Date: **31/07/2024**

CIN: **PIT/AUG24/07375**

Dear **Logapriya S,**

We are pleased to offer you the position of **Cyber Security Intern** at **Prodigy InfoTech.**This is an educational internship. As a valued member of our team, you will have the opportunity to gain hands-on experience in this field.

The internship is scheduled to commence on the **1st August, 2024** and will conclude on the **31st August, 2024**, resulting in a 1-month duration for the program.

By accepting this offer, you acknowledge that you understand participation in this program is not an offer of employment, and successful completion of the program does not entitle you to an employment offer from Prodigy InfoTech.

You also agree that you will follow all of the company's policies that apply to non-employee interns. This letter constitutes the complete understanding between you and the company regarding your internship and supersedes all prior discussions or agreements. This letter may only be modified by a written agreement signed by both of us.

We eagerly anticipate your commencement of the internship program at Prodigy InfoTech and extend our best wishes for a prosperous experience.

Sincerely,

Prodigy InfoTech

PRODIGY INFOTECH
UDYAM-MH-19-0221586

Email
contact@prodigyinfotech.dev

Website
prodigyinfotech.dev

MSME

Figure 8.1: **Internship offer letter**

Chapter 10 SOURCE CODE & POSTER PRESENTATION

1. Source code

```
20
21
22
23 echo -e "\e[1;34mStarting Subdomains Enumerations\e[0m"
24 echo ""
25 subfinder -d $1 | sort -u | tee all.txt
26 echo ""
27 echo -e "\e[1;32m---------------- Completed! Saved as all.txt ------------------\e[0m"
28 echo ""
29 sleep 3
30
31 echo -e "\e[1;34mStarting to filterlive Domains\e[0m"
32 echo ""
33 cat all.txt | httprobe -s -p https:443 | sort -u | grep $1 | sed 's/https
\?:\/\///' | tr -d ":443" | tee alive.txt
34 chmod +X alive.txt
35 echo ""
36 echo -e "\e[1;32m---------------- Completed! Saved as alive.txt ------------------\e[0m"
37 echo ""
38
39 echo -e "\e[1;34mStarting Aquatone for Webscreenshots\e[0m"
40 echo ""
```

```
cat alive.txt |aquatone -out Aquatone -all
echo ""
echo -e "\e[1;32m---------------- Completed! Saved as Aquatone-all.txt
------------------\e[0m"


echo ""
echo -e "\e[1;34mChecking for vlun Subdomains\e[0m"
echo ""
subzy -targets all.txt -hidefails |tee subsdomainvlun.txt
echo ""
echo -e "\e[1;33mWARNING: In txt file only shows vlun domains or else nothingthere that case you can check Manually!\e[0m"
echo ""
echo -e "\e[1;32m---------------- Completed! Saved as subsdomainvlun.txt
------------------\e[0m"
echo ""


echo -e "\e[1;33mNow Starting Wayabackrls and Waybackurls realted Enumerations!
\e[0m"
sleep 2
echo ""
echo -e "\e[1;34m Start running Waybackurls\e[0m"
```

```
60 echo ""
61 waybackurls $1 | sort -u | tee wayback -all/waybackurls.txt | wc -l
62 echo ""
63 echo "How many numbers of urls found shown above!"
64 echo -e "\e[1;32m---------------- Completed! Saved as waybackurls.txt
----------\e[0m"
65 sleep 3

66

67 echo ""
68 echo -e "\e[1;34mEnum for JS from waybackurls! \e[0m"
69 echo ""
70 cat wayback -all/waybackurls.txt | grep js | tee wayback -all/wayback -js.txt |wc
-l
71 echo "Numaber of JS files found shown above"
72 echo -e "\e[1;32m---------------- Completed! Saved as wayback -js.txt
----------\e[0m"

73

74 echo ""
75 echo -e "\e[1;34mEnum for Parameter from waybackurls!\e[0m"
76 cat wayback -all/waybackurls.txt | grep "=" | tee wayback -all/wayback -params.txt
| wc -l
77 echo "Number of params found shown above!"
78 echo -e "\e[1;32m---------------- Completed! Saved as wayback -params.txt
```

```
----------\e[0m"
echo ""

echo -e "\e[1;34mEnum for SSRF from waybackurls!\e[0m"
cat wayback-all/waybackurls.txt |gf ssrf |tee wayback-all/wayback-ssrf.txt |
wc -l
echo "Number of ssrf found shown above!"
echo -e "\e[1;32m---------- Completed! Saved as wayback-ssrf.txt
----------\e[0m"
echo ""

echo ""
echo -e "\e[1;34mEnum for SQLI from waybackurls! \e[0m"
cat wayback-all/waybackurls.txt |gf sqli |tee wayback-all/wayback-sqli.txt |
wc -l
echo "Number of ssrf found shown above!"
echo -e "\e[1;32m---------- Completed! Saved as wayback-sqli.txt
----------\e[0m"
echo ""

echo -e "\e[1;34mEnum for IDORS from waybackurls!\e[0m"
cat wayback-all/waybackurls.txt |gf idor |tee wayback-all/wayback-idor.txt |
```

```
wc -l

echo "Number of ssrf found shown above!"

echo -e "\e[1;32m---------------- Completed! Saved as wayback-idors.txt

------------------\e[0m"

echo ""



echo -e "\e[1;34mEnum for XSS from waybackurls!\e[0m"

cat wayback-all/waybackurls.txt |grep "=" |qsreplace '"><script>confirm(1)</script>' |tee wayback-all/wayback-xss.txt |wc -l

echo "Number of ssrf found shown above!"

echo -e "\e[1;32m---------------- Completed! Saved as wayback-xss.txt

------------------\e[0m"



echo ""

echo -e "\e[1;34mEnum for Redirect from waybackurls!\e[0m"

cat wayback-all/waybackurls.txt |gf redirect |tee wayback-all/wayback-redirect

.txt |wc -l

echo "How many links realeted to rediret shown above!"

echo -e "\e[1;32m---------------- Completed! Saved as wayback-redirect.txt

------------------\e[0m"



echo ""
```

```
113 echo -e "\e[1;34m Testing for sqlinjection urls from wayback - sqli.txt \e[0m"
114 sqlmap -m wayback - all/ wayback - sqli.txt --dbs -- batch |tee sql-test-result.txt
115 echo -e "\e[1;32m---------------- Completed! Saved as sql-test-result.txt.txt
----------\e[0m"
116
117
118
119
120
121
122
123
124
125
```