A

Major Project

on

**Automated reconnaissance tool for pentesting engagements**

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

**Sundeep Varma(187R1A0558)**

Under the Guidance of

**J.Narasimharao**

(Associate Professor)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CMR TECHNICAL CAMPUS**

**UGC AUTONOMOUS**

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE,NewDelhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

**2018-22**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

This is to certify that the project entitled “**AUTOMATED RECONNAISSANCE TOOL FOR PENTESTING ENGAGEMENTS**” being submitted by **SUNDEEP VARMA (187R1A0558)** in partial fulfillmentof the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him under our guidance and supervision during the year 2021-22.

The results embodied in this thesis has not been submitted to any other University or Institute for the award of any degree or diploma.

**J.Narasimharao Dr. A. Raji Reddy**

**Associate Professor DIRECTOR**

**INTERNAL GUIDE**

**Dr. K. Srujan Raju**  **EXTERNAL EXAMINER**

**HoD**

**Submitted for viva voice Examination held on**



**ACKNOWLEDGEMENT**

**Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.**

**We take this opportunity to express my profound gratitude and deep regard to my guide J. Narasimharao Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of  life on which we are about to embark.**

**I also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC)  Dr. M. Vara Prasad  Rao,  Mr. J. Narasimha Rao, Mr. A. Uday Kiran**

**Dr. T. S. Mastan Rao, Dr. Suwarna Gothane , Mr. A. Kiran Kumar, Mrs. G. Latha for their cordial support, valuable information and guidance, which  helped us in completing this task through various stages.**

**We are also thankful to Dr. K. Srujan Raju, Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.**

**We are obliged to Dr. A. Raji Reddy, Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. Ch. Gopal Reddy, Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.**

**The guidance and support received from all the members of CMR Technical Campus who contributed to the completion of the project. We are grateful for their constant support and help.**

**Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.**

**SUNDEEP VARMA(187R1A0558)**

**ABSTRACT**

Reconnaissance denotes the work of information gathering before any real attacks are planned. The idea is to collect as much interesting information as possible about the target. To achieve this, many different publicly available sources of information are used. The extracted information will often already allow a detailed insight into the affected systems.This encompasses technical as well as nontechnical information. Technical information may be IP-ranges, insight into the (internal) network infrastructure, used hardware and even passwords. There are various tools available in the community to do the reconnaissance.The tool I propose chains all the tools commonly used during the reconnaissance process like sublister,nmap,search sploit, aquatone,eyewitness, HTTPX, dirbuster, gobuster etc. Chaining all these tools system offers reconnaissance for both bug bounty and CTF styled processes. In the end this project provides a clean report of all the findings.

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NUMBER** | **FIGURE NAME** | **PAGE NUMBER** |
| Figure 3.1 | Project Architecture of the antivirus | 07 |
| Figure 3.2 | ML Architecture of the antivirus | 07 |
| Figure 3.3 | Use Case Diagram of the antivirus | 08 |
| Figure 3.4 | Sequence Diagram of the antivirus | 09 |
| Figure 3.5 | Activity Diagram of the antivirus | 10 |
| Figure 3.6 | Data Flow Diagram of the antivirus | 11 |

CMRTC **ii**

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLENUMBER** | **TABLE NAME** | **PAGE NUMBER** |
| Figure 6.2.3.1 | Test Cases of the antivirus | 21 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

CMRTC  **iii**

**LIST OF SCREENSHOTS**

|  |  |  |
| --- | --- | --- |
| **SCREENSHOT NUMBER** | **SCREENSHOT NAME** | **PAGE NUMBER** |
| SCREENSHOT 4.1 | Safe Results | 18 |
| SCREENSHOT 4.2 | Malicious Results | 18 |

**iv**

CMRTC

|  |  |
| --- | --- |
| **TABLE OF CONTENT** |  |
| **ABSTRACT** | i |
| **LIST OF FIGURES** | ii |
| **LIST OF TABLES** | iii |
| **TABLE OF SCREENSHOTS** | iv |
| **1 INTRODUCTION** | 1 |
| 1.1 PROJECT SCOPE | 01 |
| 1.2 PROJECT PURPOSE | 01 |
| 1.3 PROJECT FEATURES | 01 |
| **2 SYSTEM ANALYSIS** |  |
| 2.1 PROBLEM DEFINITION | 02 |
| 2.2 EXISTING SYSTEM | 03 |
| 2.2.1 LIMITATIONS OF EXISTING SYSTEM | 03 |
| 2.3 PROPOSED SYSTEM | 03 |
| 2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM | 04 |
| 2.4 FEASIBILITY STUDY | 04 |
| 2.4.1 ECONOMIC FEASIBILITY | 04 |
| 2.4.2 TECHNICAL FEASIBILITY | 05 |
| 2.4.3 BEHAVIORAL FEASIBILITY | 05 |
| 2.5.1 HARDWARE REQUIREMENTS | 05 |
| 2.5.2 SOFTWARE REQUIREMENTS: | 06 |
| **3 ARCHITECTURE** |  |
| 3.1 ARCHITECTURE | 07 |
| 3.2 DESCRIPTION | 08 |

|  |  |
| --- | --- |
| 3.2.1 ML Model | 08 |
| 3.3 USE CASE DIAGRAM | 08 |
| 3.4 SEQUENCE DIAGRAM | 09 |
| 3.5 ACTIVITY DIAGRAM | 10 |
| 3.6 DATA FLOW DIAGRAM | 11 |
| **4 IMPLEMENTATION** |  |
| 4 IMPLEMENTATION | 12 |
| **5 RESULTS** |  |
| 5 RESULTS | 19 |
| **6 TESTING** |  |
| 6.1 INTRODUCTION TO TESTING | 20 |
| 6.2 TYPES OF TESTING | 20 |
| 6.2.1 UNITTESTING | 20 |
| 6.2.2 INTEGRATION TESTING | 22 |
| 6.2.3 FUNCTIONAL TESTING | 22 |
| 6.2.4 TEST CASES | 22 |
| **7 CONCLUSION & FUTURE SCOPE** |  |
| 7.1 PROJECT CONCLUSION | 23 |
| 7.2 FUTURE SCOPE | 23 |
| **8 BIBLIOGRAPHY** |  |
| 8.1 REFERENCES | 24 |
| 8.2 WEBSITES | 24 |

**1. INTRODUCTION**

An Antivirus withMachine Learning

**INTRODUCTION**

**1.1 PROJECT SCOPE**

This project is titled as “Automated reconnaissance tool for pentesting engagements”. This script helps people playing CTFs to perform the reconnaissance on the target machine. It finds the active ports and enumerates the possible exploits for the services running on the port. The script also offers the reconnaissance for bug bounties. It takes the target domain and performs subdomain enumeration with 3 different tools along with filtering all the active hosts.It presents a pdf report for either of the cases.

**1.2 PROJECT PURPOSE**

This has been developed for the people who play ctfs and for the people who do bug bounties.As the recon in either of the engagements need new tools to run manually and filter the results, this project aims to automate the entire process and generate a pdf report of all the findings ultimately making the recon fast and automated.

**1.3 PROJECT FEATURES**

This project offers different services based on the enumeration type. In the ctf segment the tool performs NMAP scan and collects active ports.It automatically enumerates the services running on the port and perform online search for the possible exploits on the service. It also performs different web attacks as XSS,SQLI,File upload etc. For the bug bounty engagement the tool enumerated all the subdomains using sublister, asstefinder and subfinder. It filters all live subdomains using httprobe. It also presents a pdf of all the findings.

CMRTC 1

**2. SYSTEM ANALYSIS**

An Antivirus with Machine Learning

**SYSTEM ANALYSIS**

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified.

Once analysis is completed the analyst has a firm understanding of what is

to be done.

**2.1 PROBLEM DEFINITION**

A detailed study of the process must be made by various techniques like Image processing, feature recognition etc. The data collected by these sources must be scrutinized to arrive at a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

CMRTC 2

An Antivirus with Machine Learning

**2.2 EXISTING SYSTEM**

There are not many tools online which could automate the reconnaissance, Even if there are any tools they just automated only one segment ie either directory brute forcing, or searching the exploit for services.There are no tools which would integrate the recon of bug bounty or of CTFs.

**2.2.1 LIMITATIONS OF EXISTING SYSTEM**

* Not many tools for Entire recon automation.
* Few are Paid.
* They have complex syntax.
* NO proper report generator.
* No Tool integrating bug bounty and CTF recon.

**2.3** **PROPOSED SYSTEM**

The system/Program proposed does the entire recon automated and is heavily customizable, Not only it offers the automation in recon for CTFs but also for bug bounty programs. It generates a proper report of the entire findings. It uses the best of common tools to rely on results.

CMRTC 3

An Antivirus with Machine Learning

**2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM**

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features

* Proper report.
* Heavily customizable.
* Offers recon on Bug Bounty and CTFs
* Pre installs all the libraries and tools required..

**2.4** **FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis are

Economic Feasibility

Technical Feasibility

Social Feasibility

**2.4.1 ECONOMIC FEASIBILITY**

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on a project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

The costs conduct a full system investigation.

The cost of the hardware and software.

The benefits in the form of reduced costs or fewer costly errors.

CMRTC 4

An Antivirus with Machine Learning

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it gives an indication that the system is economically possible for development.

**2.4.2 TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system

**2.4.3 BEHAVIORAL FEASIBILITY**

This includes the following questions:

Is there sufficient support for the users?

Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed

And installed. All behavioral aspects are considered carefully and conclude that the

project is behaviorally feasible.

**2.5.1 HARDWARE REQUIREMENTS:**

Hardware interfaces specifies the logical characteristics of each interface

between the software product and the hardware components of the system. The

the following are some hardware requirements.

● **System** : AMD A4 or higher

**Hard Disk** : 5 GB or Higher

● **Ram** : 4-GB.or Higher

● **Input Devices**  : Keyboard,Mouse.

CMRTC 5

An Antivirus with Machine Learning

**2.5.2 SOFTWARE REQUIREMENTS:**

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

**Operating system** : Kali /Parrot Linux

**Coding Language**  : GO, BASH, RUST, DOCKER, Python

● **Tool** : Vscode,Intellij

CMRTC 6

**3. ARCHITECTURE**

An Antivirus with Machine Learning

**ARCHITECTURE**

The images below show the project architecture fig 3.1 is the overall architecture of a project which includes the different modules involved . The specific functionality of each modules is given below

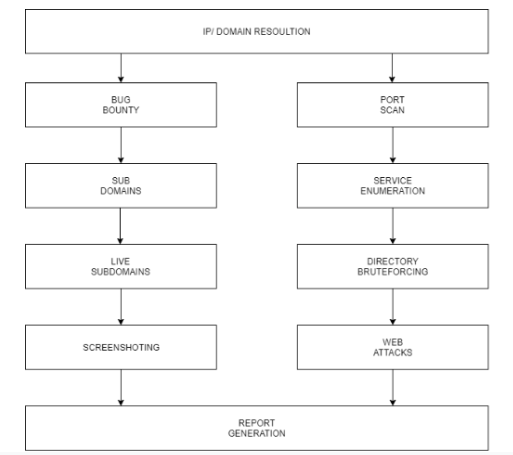


Fig 3.1: Project Architecture of Antivirus

**3.2 DESCRIPTION**

IP Module :ip modules the ip is taken and analyzed for missing octals or domain lookup is done.

Port Scan : Nmap is made Scan on the target for checking the open ports.

Common exploits are made search on the obtained running services

Subdomains : Subdomains are analyzed and obtained with various tools

Live Subdomains Active and live subdomains are filtered by HTTPX

Screenshotting : Screenshotting is done using eyewitness or aquatone

Directory brute forcing : Directory brute forcing is done using gibuster or dirbuster with common seclist wordlist

Web Attacks :Web Attacks like sql injection ,xss , rce, injection attacks are performed using various tools

CMRTC 7

An Antivirus with Machine Learning

**3.3 USE CASEDIAGRAM**

As the scripts offer different modules, users can access all the modules to enumerate either ctfs or bug bounties and a pdf of result!

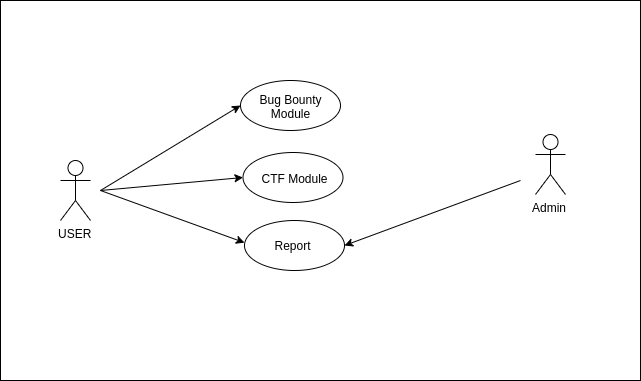


Fig 3.3: Use Case Diagram of the antivirus

CMRTC 8

An Antivirus with Machine Learning

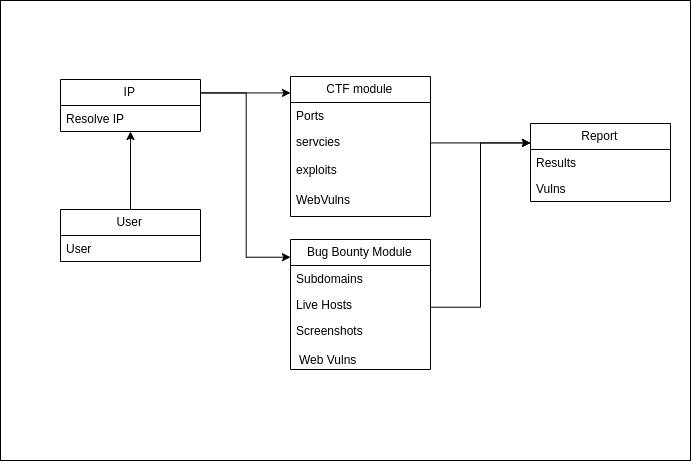
**3.4 SEQUENCE DIAGRAM**

User enters the target ip as command line argument, tool takes the ip and checks if it's valid which is passed to ctf module where port scanning,service enumeration, exploit finder, webattacts are performed. The results are written to a pdf which is accessed by the user. Same with the bug bounty module. Domain is entered by user which is checked and subdomain enumeration is done which is further written into a pdf report, This report is presented to the user.

Fig 3.4: Sequence Diagram of the antivirus

**3.4 CLASS DIAGRAM**

Different functions/classes are implemented to achieve the different functionalities. When the user enter the IP ip module check if its th valid IP, when a domain is entered for bug bounty it check the validity of the domain and then pass it ti CTF/Bug bounty modules/functions where different attacks/enumeration is performed and results are passed ro report module where with obtained results a pdf is generated



CMRTC 9

An Antivirus with Machine Learning

**3.5 Activity Diagram**

We use Activity Diagrams to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram.

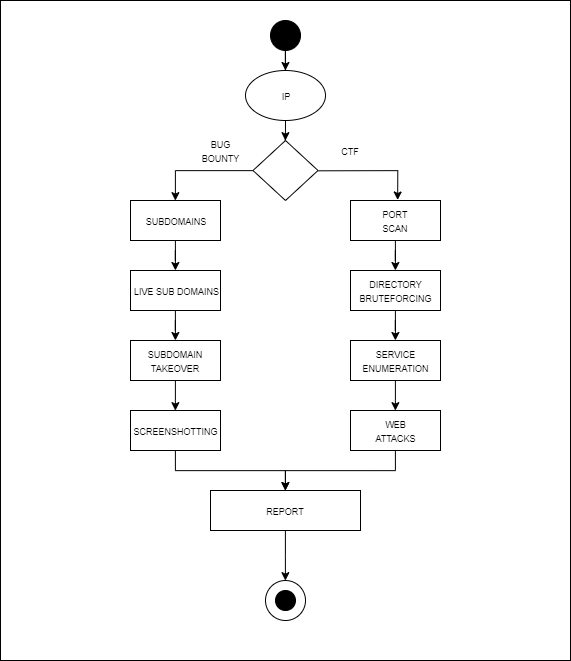


Fig 3.5: Activity Diagram of the antivirus

CMRTC 10

An Antivirus with Machine Learning

CMRTC 11

**4.IMPLEMENTATION**

An Antivirus with Machine Learning

CMRTC 12

An Antivirus with Machine Learning

CMRTC 13

An Antivirus with Machine Learning

**Python code:**

import argparse

from pickle import GLOBAL

from time import sleep

from fpdf import FPDF

import socket

import sys

import re

import os

import sublist3r

import subprocess as sp

from sqlalchemy import true

from datetime import datetime

from datetime import date

today = date.today()

now = datetime.now()

current\_time = now.strftime("%H:%M:%S")

daten = today.strftime("%B %d, %Y")

stg1=False

mainip=''

homedir=''

bbdomain=''

aldmn1=''

unqdmn1=''

lvdmn1=''

ports=''

services=''

exploits=''

versions=''

webatt=''

title='Scan report'

class PDF(FPDF):

def header(self):

# Arial bold 15

self.set\_font('Arial', 'B', 15)

# Calculate width of title and position

w = self.get\_string\_width(title) + 6

self.set\_x((210 - w) / 2)

# Colors of frame, background and text

self.set\_draw\_color(0, 20, 180)

self.set\_fill\_color(230, 230, 0)

self.set\_text\_color(220, 50, 50)

# Thickness of frame (1 mm)

self.set\_line\_width(1)

# Title

self.cell(w, 9, title, 1, 1, 'C', 1)

# Line break

self.ln(10)

def footer(self):

# Position at 1.5 cm from bottom

self.set\_y(-15)

# Arial italic 8

self.set\_font('Arial', 'I', 8)

# Text color in gray

self.set\_text\_color(128)

# Page number

self.cell(0, 10, 'Page ' + str(self.page\_no()), 0, 0, 'C')

def add\_title(self, label):

# Arial 12

self.set\_font('Arial', '', 12)

# Background color

self.set\_fill\_color(200, 220, 255)

# Title

self.cell(0, 6, label, 0, 1, 'L', 1)

# Line break

self.ln(4)

def add\_body(self, name):

# Read text file

with open(name, 'rb') as fh:

txt = fh.read().decode('UTF-8')

# Times 12

self.set\_font('Times', '', 12)

self.multi\_cell(0, 5, txt)

# Line break

self.ln()

# Mention in italics

self.cell(0, 5, '(end of excerpt)')

def print\_chapter(self,title, name):

self.add\_page()

self.add\_title(title)

self.add\_body(name)

def getdomain(hostname):

try:return socket.gethostbyname(hostname)

except: return False

def isValidDomain(ip):

regex = "^((?!-)[A-Za-z0-9-]" + "{1,63}(?<!-)\\.)" + "+[A-Za-z]{2,6}"

p = re.compile(regex)

if (ip == None):

return False

if(re.search(p, ip)):

return True

else:

return False

def validIPAddress(IP):

s=IP

def isIPv4(s):

try: return str(int(s)) == s and 0 <= int(s) <= 255

except: return False

def isIPv6(s):

if len(s) > 4:

return False

try : return int(s, 16) >= 0 and s[0] != '-'

except:

return False

try:

if IP.count(".") == 3 and all(isIPv4(i) for i in IP.split(".")):

return 1

except:

return 2

try:

if IP.count(":") == 7 and all(isIPv6(i) for i in IP.split(":")):

return 1

except:

return 2

return 2

def handledirs(target):

os.system('mkdir '+ target)

global homedir

homedir='/'+target

return True

def portscan(ip):

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Nmap Scan Started\*\*\*\*\*\*\*\*\*\*\*\*\*")

os.system('nmap -sC -sV -A '+ip +' -Pn- -oG nmapo3.txt > /dev/null 2>&1')

print('\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Nmap Scan Completed\*\*\*\*\*\*\*\*\*\*\*\*\*')

filename='nmapo3.txt'

f = open(filename)

lines=f.readlines()

portlist=lines[2]

val=portlist.split("\t")

nl=val[1:-1]

listtostr=''.join([str(elem) for elem in nl])

ns=listtostr[6:].split(",")

filenm=ip+'.txt'

file1 = open( filenm, 'w')

for k in ns:

file1.write(k)

file1.write("\n")

file1.close()

global ports

oports =sp.check\_output('awk -F "/" \'{print $1}\' '+filenm , shell=True)

rversions=sp.check\_output('awk -F "/" \'{print $7}\' '+filenm , shell=True)

rservices= sp.check\_output('awk -F "/" \'{print $5}\' '+filenm , shell=True)

prt=oports.decode("utf-8")

prt=list(prt.split("\n"))

ports=prt

filename=ip+'\_ports.txt'

os.system('touch '+filename)

file1 = open(filename, 'w')

file1.write("ports")

for k in prt:

file1.write(k)

file1.write("\n")

file1.close()

global versions

global services

ver=rversions.decode("utf-8")

ver=list(ver.split("\n"))

versions=ver

ser=rservices.decode("utf-8")

ser=list(ser.split("\n"))

filenm=ip+'\_services.txt'

file1 = open( filenm, 'w')

for k in ver:

v=k.replace('(','').replace(')','')

file1.write(v)

file1.write("\n")

file1.close()

rservices= sp.check\_output('awk -F " " \'{print $1" " $2}\' '+filenm , shell=True)

ser=rservices.decode("utf-8")

ser=list(ser.split("\n"))

print('\n\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Ports open\*\*\*\*\*\*\*\*\*\*\*\*\*')

print("Ports")

for k in prt:

print(k)

print('\n\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*services and versions\*\*\*\*\*\*\*\*\*\*\*\*\*')

services=ser

print("services-versions")

for k in ser:

if len(k)>2:

print(k)

def service\_enum():

pass

def search\_vuln():

explist=[]

global exploits

for k in services:

if len(k)>2:

print('\n seraching exploits for :' + k)

print('searchsploit '+k)

serres= sp.check\_output('searchsploit ' +k, shell=True)

ser=serres.decode("utf-8")

explist.append(ser)

print('\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Possible exploits\*\*\*\*\*\*\*\*\*\*\*\*\*')

exploits=explist

for k in explist:

print(k)

def web\_attacks(ip):

global webatt

print('\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"Trying web attacks"\*\*\*\*\*\*\*\*\*\*\*\*\*')

os.system('uniscan -d -u '+ip)

'''filenm=ip+'\_attacks.txt'

file1 = open( filenm, 'w')

for v in webattacks:

file1.write(v)

file1.write("\n")

file1.close()'''

def ctf(ip):

global stg1

global mainip

if validIPAddress(ip)==1:

stg1=true

mainip=ip

elif validIPAddress(ip)==2:

if isValidDomain(ip):

if getdomain(ip):

stg1=true

mainip=getdomain(ip)

else: print('check the domain')

else: print('check the ip')

if stg1:

handledirs(mainip)

portscan(mainip)

search\_vuln()

web\_attacks(ip)

def getsublister(bbdomain):

print('\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Started Sublister\*\*\*\*\*\*\*\*\*\*\*\*\*')

filenm='sublister.txt'

subdomains = sublist3r.main(bbdomain, 40, filenm , ports= None, silent=False, verbose= False, enable\_bruteforce= False, engines=None)

def getassetfinder(bbdomain):

print('\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Started assetfinder\*\*\*\*\*\*\*\*\*\*\*\*\*')

filenm='assetfinder.txt'

os.system('assetfinder --subs-only '+bbdomain+' >>'+filenm)

def getsubfinder(bbdomain):

print('\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Started subfinder\*\*\*\*\*\*\*\*\*\*\*\*\*')

os.system('subfinder -d '+bbdomain+' -o subfinder.txt')

def rem\_noise():

global aldmn1

global unqdmn1

print('\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Removing noise\*\*\*\*\*\*\*\*\*\*\*\*\*')

os.system('cat assetfinder.txt sublister.txt subfinder.txt >> all\_domains.txt')

aldmn = sp.getoutput("wc -l all\_domains.txt | grep -o '[0-9]\+'")

aldmn1=aldmn

print('subdomains aquired : '+aldmn)

print('\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Removing duplicate subdomains\*\*\*\*\*\*\*\*\*\*\*\*\*')

os.system('cat all\_domains.txt| sort | uniq >> uniq\_domains.txt')

print('unique domains')

unqdmn = sp.getoutput("wc -l uniq\_domains.txt | grep -o '[0-9]\+'")

unqdmn1=unqdmn

print('subdomains aquired : '+unqdmn)

def getlive\_d():

global lvdmn1

print('\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*getting live domains\*\*\*\*\*\*\*\*\*\*\*\*\*')

os.system('cat uniq\_domains.txt | httprobe >> live\_domains.txt')

lvdmn = sp.getoutput("wc -l live\_domains.txt | grep -o '[0-9]\+'")

lvdmn1=lvdmn

print('No of live domains : '+lvdmn)

def bugbounty(domaint):

global bbdomain

bbdomain=domaint

global current\_time

global daten

handledirs(bbdomain)

if isValidDomain(bbdomain):

getsublister(bbdomain)

getassetfinder(bbdomain)

getsubfinder(bbdomain)

rem\_noise()

getlive\_d()

pdf = PDF()

pdf.add\_page()

pdf.set\_title('Scan results')

pdf.set\_author('Agastya')

pdf.add\_title('Scan Date : '+daten+' Time : '+current\_time)

pdf.add\_title('General Info')

pdf.cell(0, 5, 'Total Domains enumrated : '+aldmn1)

pdf.ln()

pdf.cell(0, 5, 'Non Duplicate Domains enumrated : '+unqdmn1)

pdf.ln()

pdf.cell(0, 5, 'Live Domains enumrated : '+lvdmn1)

pdf.ln()

pdf.add\_title('Domain : '+bbdomain)

pdf.add\_title('Live Domains are listed below')

pdf.add\_body('live\_domains.txt')

pdf.output(bbdomain+'\_results.pdf', 'F')

else:

print('check the domain')

parser = argparse.ArgumentParser()

group=parser.add\_mutually\_exclusive\_group()

parser.add\_argument("-t", help='enter the ip target of ctf challenge')

parser.add\_argument("-d", help='enter the domain')

group.add\_argument("-c", "--ctf", action='store\_true', help="Run only ctf module")

group.add\_argument("-b", "--bugbounty" ,action='store\_true', help="Run only bug bounty module")

args=parser.parse\_args()

if args.ctf:

ip=args.t

ctf(ip)

elif args.bugbounty:

domaint=args.d

bugbounty(domaint)

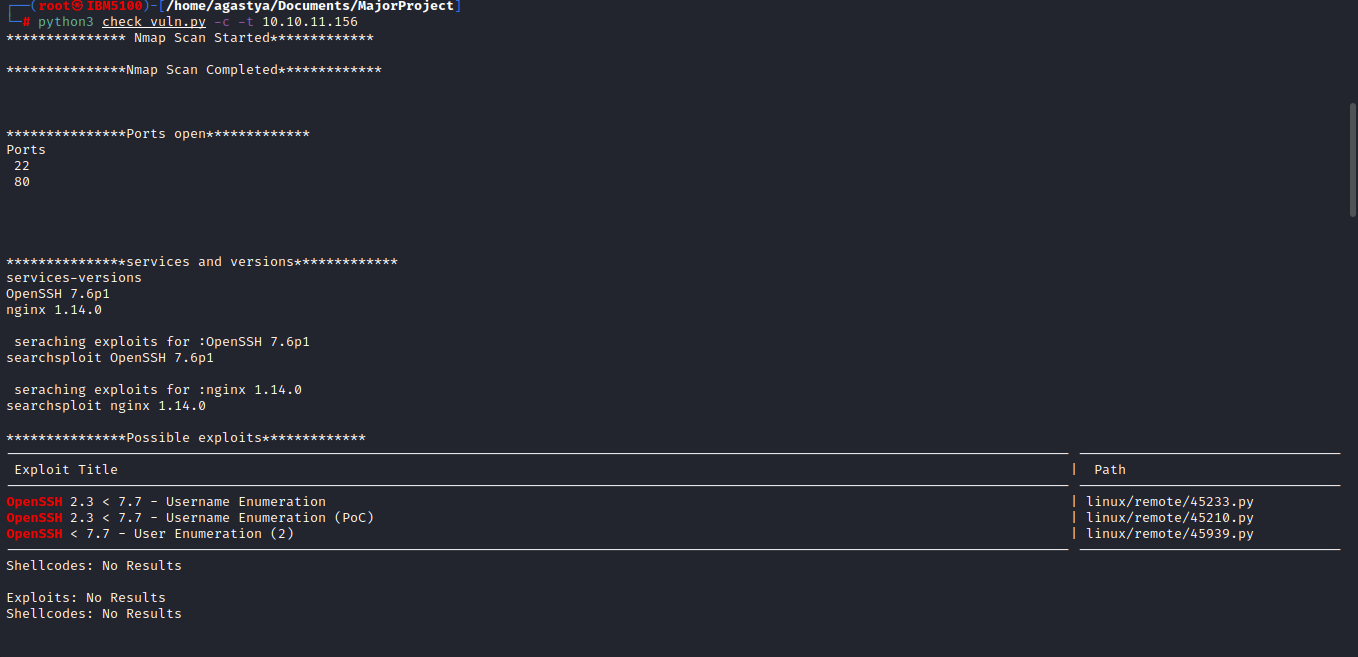
17

**5.RESULTS**

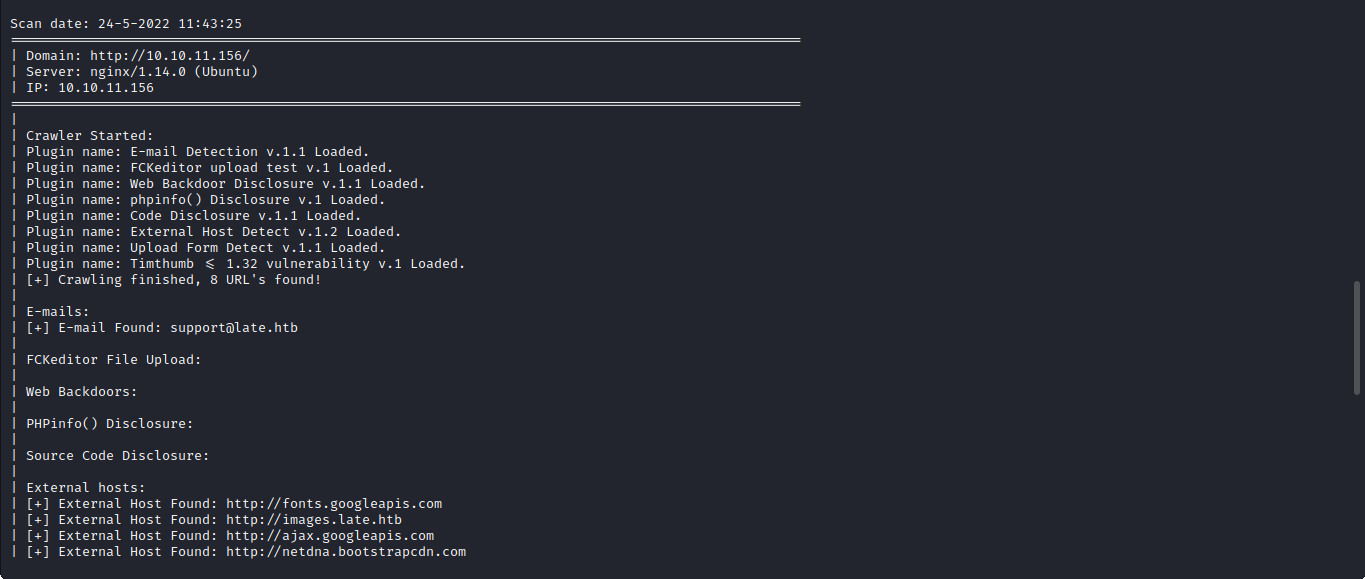
An Antivirus with Machine Learning

**5.RESULTS**

The output of the CTF module



Service enumeration



Web attacks



Subdomain enumeration

CMRTC 18

**6.TESTING**

An Antivirus with Machine Learning

**6. TESTING**

**6.1 INTRODUCTION TO TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement**.**

**6.2 TYPES OFTESTING**

**6.2.1 UNITTESTING**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results**.**

**6.2.2 INTEGRATION TESTING**

Integration tests are designed to test integrated software components to determine if they actually 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 satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of component

CMRTC 19

An Antivirus with Machine Learning

**6.2.3 FUNCTIONALTESTING**

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

Functional testing is centered on the following items:

**Valid Input** : identified classes of valid input must be accepted.

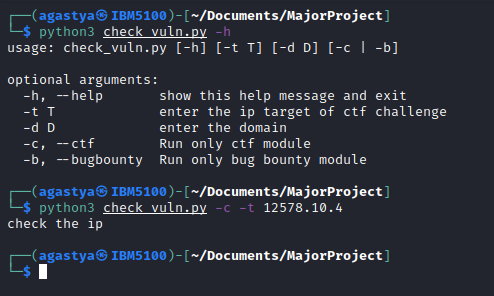
**Invalid Input** : identified classes of invalid input must be rejected.

**Functions** : identified functions must be exercised.

**Output** : identified classes of application outputs must be exercised.

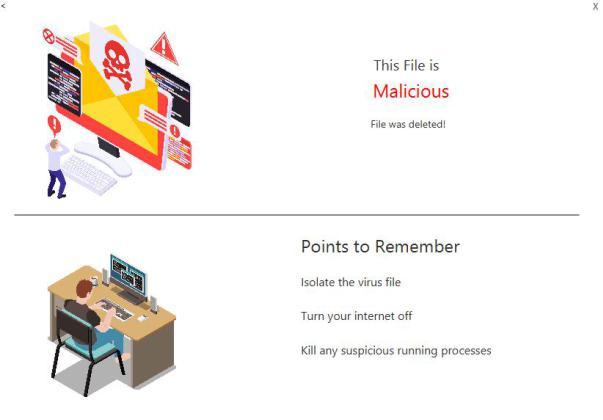
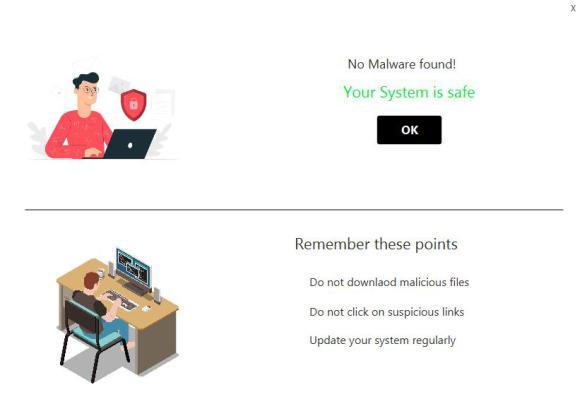
**Systems/Procedures**:interfacing systems or procedures must beinvoked. Orginization

and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes.



Testing inputs

CMRTC 20



CMRTC 21

**CONCLUSION**

An Antivirus with Machine Learning

**CONCLUSION & FUTURESCOPE**

**7.1 PROJECT CONCLUSION**

As reconnaissance is an important test and needs manual effort to run tools and enumerate. Using this tool during any penesting engagements or CTFs and bug bounties in general this will automate the entire process and generates a pdf report using this report pentesters can focus on what's important and save enough time

**7.2 FUTURE SCOPE**

In the future updates more modules can be added to enumerate on given service with more web attacks. More subdomain enumeration techniques and filtering could be added

CMRTC 22

**8. BIBILOGRAPHY**

An Antivirus with Machine Learning

**8.1 REFERENCES**

<https://www.sciencedirect.com/science/article/pii/S1084804519303868#sec1>

<https://arxiv.org/ftp/arxiv/papers/1205/1205.3062.pdf>

<https://www.n-able.com/blog/using-machine-learning-create-smart-antivirus>

[https://media.kaspersky.com/en/enterprise-security/Kaspersky-Lab- Whitepaper-](https://media.kaspersky.com/en/enterprise-security/Kaspersky-Lab-Whitepaper-Machine-Learning.pdf)Machine-Learning.pdf

<https://link.springer.com/article/10.1007/s13748-020-00220-4>

**8.2 WEBSITES**

<https://github.com/starhopp3r/ML-Antivirus>

<https://github.com/N00NY/Antivirus-Python-/blob/master/SystemFileScanner.py>

<https://github.com/N00NY/Antivirus-Python->

CMRTC 23