MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Project Report

on

ANTIVIRUS WITH PYTHON USING GUI

A project report submitted in partial fulfilment of the requirement for the digree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted By:

Rajkumar Dohare

0901CS191091

Ravi Rathour

0901cs191092

Faculty Mentor:

Rajni Ranjan Singh Makwana

Assistant Professor , Computer Science and Engineering

Submitted to

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

GWALIOR - 474005 (MP) est. 1957

MAY-JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR
(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Project Report

on

ANTIVIRUS WITH PYTHON USING GUI

A project report submitted in partial fulfilment of the requirement for the digree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted By:

Rajkumar Dohare
0901CS191091
Ravi Rathour
0901cs191092

Faculty Mentor:

Rajni Ranjan Singh Makwana
Assistant Professor , Computer Science and Engineering

Submitted to:

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
GWALIOR - 474005 (MP) est. 1957
MAY-JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that **Rajkumar Dohare** (0901cs191091) has submitted the project report titled "ANTIVIRUS WITH PYTHON USING GUI" under the mentorship of Prof. Mir Shahnawaz sir, in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

9/5/22

Prof. Rajni Ranjan Singh Makwana
Faculty Mentor
Assistant Professor
Computer Science and Engineering

09/05/2022

Dr. Manish Dixit
Professor and Head,
Computer Science and Engineering

Dr. Manish Dixit
Professor & HOD
Department of CSE
M.I.T.S. Gwalior

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that **Ravi Rathour** (0901cs191092) has submitted the project report titled "ANTIVIRUS WITH PYTHON USING GUI" under the mentorship of Prof. Mir Shahnawaz sir, in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

9/5/22

Prof. Rajni Ranjan Singh Makwana
Faculty Mentor
Assistant Professor
Computer Science and Engineering

09/05/2022

Dr. Manish Dixit
Professor and Head,
Computer Science and Engineering

Dr. Manish Dixit
Professor & HOD
Department of CSE
M.I.T.S. Gwalior

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

DECLARATION

We hereby declare that the work being presented in this project report, for the partial fulfilmentof requirement for the award of the degree of Bachelor of Technology in Computer Scienceand Engineering at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Rajni Ranjan Singh Makwana**,Assistant Professor, Computer Science and Engineering.

We declare that we have not submitted the matter embodied in this report for the award of anydegree or diploma anywhere else.

Rajkumar Dohare
0901CS191091
3rd Year
Computer Science and Engineering

Ravi Rathour
0901CS191092
3rd Year
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary/interdisciplinary project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Department of Computer Science and Engineering, for allowing** me to explore this project. I humbly thank **Dr. Manish Dixit,** Professor and Head, Department of Computer Science and Engineering, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance of **Rajni Ranjan Singh Makwana, Assistant Professor, Computer Science and Engineering** , for his continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department

Rajkumar Dohare
0901CS191091
3rd Year
Computer Science and Engineering

Ravi

Ravi Rathour
0901CS191092
3rd Year
Computer Science and Engineering

Abstract

Antivirus software, or antivirus software (abbreviated to **AV software**), also known as **anti-malware**, is a computer program used to prevent, detect, and remove malware.Antivirus software was originally developed to detect and remove computer viruses, hence the name.In this project we try to develop a antivirus with the help of graphical user interface , to scan a system files and network for malicious activities and quarantine these to protect computers from malicious viruses.In particular, antivirus software can protect users from malicious browser helper objects (BHOs), browser hijackers, ransomware, keyloggers, backdoors, rootkits, trojan horses, worms, malicious LSPs, dialers, fraud tools, adware, and spyware.[1] Some products also include protection from other computer threats, such as infected and malicious URLs, spam, scam and phishing attacks, online identity (privacy), online banking attacks, social engineering techniques, advanced persistent threat (APT), and botnet DDoS attacks.

सार

एंटीवायरस सॉफ़्टवेयर, या एंटीवायरस सॉफ़्टवेयर (संक्षिप्त रूप में AV सॉफ़्टवेयर), जिसे एंटी-मैलवेयर के रूप में भी जाना जाता है, एक कंप्यूटर प्रोग्राम है जिसका उपयोग मैलवेयर को रोकने, पता लगाने और हटाने के लिए किया जाता है। एंटीवायरस सॉफ्टवेयर मूल रूप से कंप्यूटर वायरस का पता लगाने और हटाने के लिए विकसित किया गया था, इसलिए नाम। इस परियोजना में हम ग्राफिकल यूजर इंटरफेस की मदद से एक एंटीवायरस विकसित करने का प्रयास करते हैं, सिस्टम फाइलों और नेटवर्क को दुर्भावनापूर्ण गतिविधियों के लिए स्कैन करते हैं और कंप्यूटर को इससे बचाने के लिए संगरोध करते हैं। दुर्भावनापूर्ण वायरस। विशेष रूप से, एंटीवायरस सॉफ़्टवेयर उपयोगकर्ताओं को दुर्भावनापूर्ण ब्राउज़र सहायक ऑब्जेक्ट (बीएचओ), ब्राउज़र अपहर्ताओं, रैंसमवेयर, कीलॉगर्स, बैकडोर, रूटकिट्स, ट्रोजन हॉर्स, वर्म्स, दुर्भावनापूर्ण एलएसपी, डायलर, धोखाधड़ी उपकरण, एडवेयर और स्पाइवेयर से बचा सकता है। कुछ उत्पादों में अन्य कंप्यूटर खतरों से सुरक्षा भी शामिल है, जैसे संक्रमित और दुर्भावनापूर्ण URL, स्पैम, स्कैम और फ़िशिंग हमले, ऑनलाइन पहचान (गोपनीयता), ऑनलाइन बैंकिंग हमले, सोशल इंजीनियरिंग तकनीक, उन्नत लगातार खतरा (APT), और बॉटनेट DDoS हमले।

TABLE OF CONTENTS

TITLE	PAGE NO.
Abstract	
सार	
List of figures	
Chapter 1 : Project Overview	**1**
1.1 Introduction	1
1.2 Overview	1
1.3 Objective	1
1.4 Methodology	1
Chapter 2: Literature Review	**2**
2.1Litetature Survey	2
2.2 Activities	3
2.2.1 Prevent	3
2.2.2 Scanning	3
2.2.3 Detecting	3
2.2.4 Deleting Virus	4
2.3 About us	4
2.4 Project Requirements	5
Chapter 3 : Preliminary Design	**6**
3.1 Software Development Life Cycle Model	6
3.2 Tools and Technologies	7
3.2.1 python	7
3.2.2 Jupyter notebook	7
3.2.3 Visual Studio Code	7
3.2.4 Anaconda	7

3.2.5 Spyder 7

3.3 Libraries Used 8

3.3.1 tkinter 8

3.3.2 threading 8

3.3.3 Os 8

3.3.4 sys 8

3.3.5 urlib 8

3.3.6 glob 8

3.3.7 time 9

3.3.8 hashlib 9

3.3.9 socket 9

3.3.10 subprocess 9

3.4 Figures 10

Chapter 4 : Final Analysis 14

4.1 Result 14

4.2 Application 15

4.3 Limitations 15

4.4 Problem From Antivirus 15

Chapter 5 : Conclusion 16

5.1 Conclusion 16

References 17

LIST OF FIGURES

Figure Number	Figure caption	Page No .
3.1	Life cycle diagram	6
3.2	Program Code module 1	10
3.3	Program Code module 2	11
3.4	Program Code module 3	11
3.5	Output 1	12
3.6	Output 2	12
3.7	Output 3	13

CHAPTER 1 : PROJECT OVERVIEW

1.1 Introduction

This is our project report on "Anti-virus with python using gui". We made this project by taking the thread of virus very seriously and we want to provide security against them. In computer systems, the security of data is always a major concern because there are some unidentified people(known as hackers) who always try to steal or harm the personal data or information of the users using viruses, worms, trojans, etc. So, to protect computer systems from these viruses or any other harmful activity, a software is developed and that software is known as Antivirus software.

1.2 Overview

This report discusses the result of the work done in development of "Anti-virus" on Python IDE Platform. It is a part of the ASSET Security project going in Computer Science Department, Madhav Institute of Technology at the development of an application framework for providing a common platform for facilitating the use of file security services developed by the team and integration of various tools developed during the execution of the project.

1.3 Objective

When we discussed technology , we can't take a chance with securities . it can harm very dangerously . in this contrast we will make a antivirus project . antivirus scan a system for malicious file , it will protect computer from malicious activities such and assure a user to safe with technical credentials.

1.4 Methodology

To implement the above goals, the following methodology needs to be followed:

1. Specifying the Application and various components of the Architecture.
2. Specifying the bindings between the tasks and the resources either manually or by the design Tools.
3. Specifying the port interconnections between the resources.
4. Analysis: Extracting the data required for application.

CHAPTER 2 : LITERATURE REVIEW

2.1 LITERATURE SURVEY :

Software that is created specifically to help detect, prevent and remove malware (malicious software).

Antivirus is a kind of software used to **prevent, scan, detect** and **delete viruses** from a **computer**. Once installed, most antivirus software runs automatically in the background to provide real-time **protection** against **virus attacks.**

Comprehensive virus protection programs help protect your files and hardware from **malware** such as **worms, Trojan horses** and **spyware**, and may also offer additional protection such as **customizable firewalls** and **website blocking**.

As the Internet of Things (IoT) grows, so does the risk of cybercrime for mobile phones and other internet-connected devices, not just your personal computer. According to Symantec's Internet Security Threat Report 2018, malware for mobile devices including spyware, ransomware and viruses increased 54% in 2017; and data breaches and identity theft are also on the rise.

Antivirus programs and computer protection software are designed to evaluate data such as web pages, files, software and applications to help find and eradicate malware as quickly as possible.

Most provide real-time protection, which can protect your devices from incoming threats; scan your entire computer regularly for known threats and provide automatic updates; and identify, block and delete malicious codes and software.

Because so many activities are now conducted online and new threats emerge continuously, it's more important than ever to install a protective antivirus program. Fortunately, there are a number of excellent products on the market today to choose from.

2.2 Activities:

2.2.1 Prevent:

Antivirus products work by detecting, quarantining and/or deleting malicious code, to prevent malware from causing damage to your device. Modern antivirus products update themselves automatically, to provide protection against the latest viruses and other types of malware.

2.2.2 Scanning:

Virus scans search through your system to locate and remove any malicious threats on your device. You'll find most antivirus software guards against malware. This can include threats like viruses and worms, as well as, spyware, Trojans, ransomware, and adware.

To expand this virus scan definition: A good antivirus product has tools for proactive and reactive protection:

- Proactive protection should help you to spot and block any potential infections before they get into your device.
- Reactive protection should be ready to take on any infections that already exist or slip past the proactive defenses.

2.2.3 Detecting:

Malware detection trends with outcomes: a basic report with a summary or a trend of malicious software detection, also showing the system and the outcome (cleaned or left alone) is a good starting point.

Detect-only events from anti-virus tools: all anti-malware tools log the cases where malicious software was detected but not cleaned (for various reasons); such logged "leave-alones" have helped many organization to avoid massive damage.

2.2.4 Deleting virus:

An antivirus software works by scanning incoming files or code that's being passed through your network traffic. Companies who build this software compile an extensive database of already known viruses and malware and teach the software how to detect, flag, and remove them.

When files, programs, and applications are flowing in and out of your computer, the antivirus compares them to its database to find matches. Matches that are similar or identical to the database are isolated, scanned, and removed.

While you can adjust settings so that your antivirus runs automatic scans of your computer for malicious files, you can also opt-in to manual scans, which let you sit and see in real-time which malicious files were found and neutralized.

Some antivirus software will ask for your permission before "cleaning" a file to remove malicious code. If you prefer a hands-off approach, you can adjust the settings so the software automatically removes malicious files.

2.3 About Us :

Antivirus software keeps an eye on all the files that enter your system. All those files are put under a scan to check for any peculiarity or maliciousness. Viruses can easily be transmitted to your network via infected files, and these, in turn, can potentially harm your data and files. You may even suffer the complete loss of your precious data at the hands of such viruses. .

2.4 Project Requirements :

Windows Based Requirements:
Computers running Microsoft Windows must meet the following minimum hardware and software
requirements.
Microsoft Windows: 7/8/10/11
4 GB RAM minimum, 8 GB RAM recommended
1GB of available disk space minimum
1280 * 800 minimum screen resolution
Software Requirement: Python 3.10.4
Hardware Requirement: Laptop/Computer
Internet Connectivity

CHAPTER 3 : PRELIMINARY DESIGN

3.1 Software Development Life Cycle Model

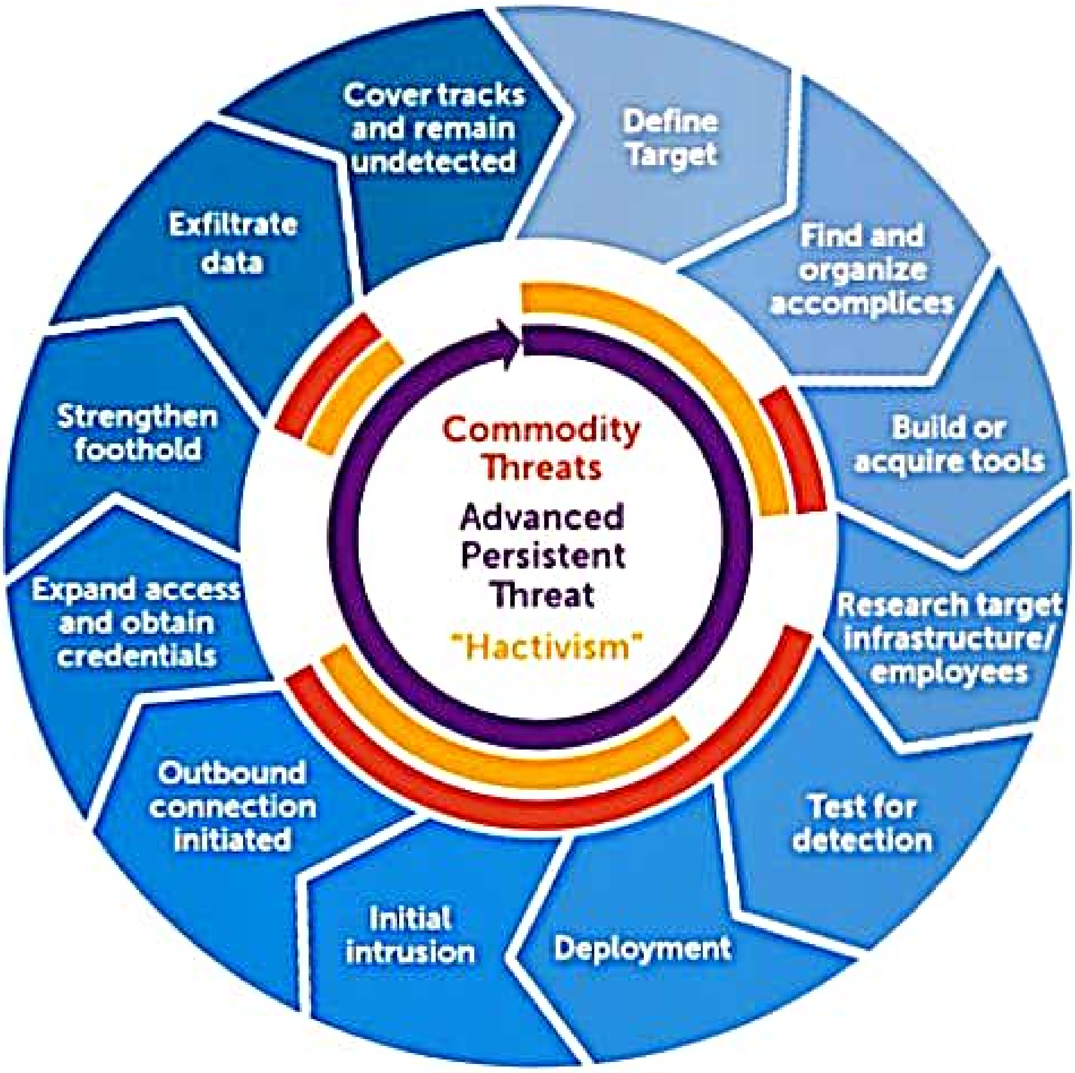

Define Target
Find and organize accomplices
Build or acquire tools
Research target infrastructure/ employees
Test for detection
Deployment
Initial intrusion
Outbound connection initiated
Expand access and obtain credentials
Strengthen foothold
Exfiltrate data
Cover tracks and remain undetected
Commodity Threats
Advanced Persistent Threat
"Hactivism"

fig 3.1

3.2 Tools & Technologies

3.2.1 Python

Python is a high-level, interpreted, interactive, and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactical constructions than other languages.

3.2.2 Jupyter Notebook

Jupyter Notebook is a web-based interactive computational environment for creating notebook documents. A Jupyter Notebook document is a browser-based REPL containing an ordered list of input/output cells which can contain code, text, mathematics, plots and rich media. Underneath the interface, a notebook is a JSON document, following a versioned schema, usually ending with the ".ipynb" extension.

3.2.3 Visual Studio Code

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS.Features include support for debugging, syntax highlighting, intelligent code completion, snippets,code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences,and install extensions that add additional functionality.

3.2.4 Anaconda

Anaconda is a distribution of the Python and R programming languages for scientific computing that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. We used anaconda to get access to Anaconda Navigator and Anaconda Prompt.

3.2.5 Spyder

Spyder, the Scientific Python Development Environment, is a free integrated development environment (IDE) that is included with Anaconda. It includes editing, interactive testing, debugging, and introspection features.

3.3 Libraries Used :

3.3.1 Tkinter

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Import the Tkinter module.

3.3.2 Threading

The threading module **exposes all the methods of the thread module and provides some additional methods** – threading. activeCount() – Returns the number of thread objects that are active. threading. currentThread() – Returns the number of thread objects in the caller's thread control.

3.3.3 Os

The OS module in Python **provides functions for creating and removing a directory (folder), fetching its contents, changing and identifying the current directory, etc.** You first need to import the os module to interact with the underlying operating system.

3.3.4 Sys

The sys module in Python **provides various functions and variables that are used to manipulate different parts of the Python runtime environment.** It allows operating on the interpreter as it provides access to the variables and functions that interact strongly with the interpreter.

3.3.5 Urlib

Urllib package is the **URL handling module for pythonn.** It is used to fetch URLs (Uniform Resource Locators). It uses the urlopen function and is able to fetch URLs using a variety of different protocols.

3.3.6 Glob

glob (short for global) is **used to return all file paths that match a specific pattern.** We can use glob to search for a specific file pattern, or perhaps more usefully, search for files where the filename matches a certain pattern by using wildcard characters.

3.3.7 Time

Python's time module **provides a function for getting local time from the number of seconds elapsed since the epoch called localtime()** . Notice that tm_isdst=0 . Since DST matters with local time, tm_isdst will change between 0 and 1 depending on whether or not DST is applicable for the given time.

3.3.8 Hashlib

For generating python secure hash message, we need to use hashlib module. **Python hashlib hashing function takes variable length of bytes and converts it into a fixed length sequence**. This is a one way function. That means, you hash a message, you get a fixed length sequence.

3.3.9 Socket

The socket module from the Python Standard Library **provides the equivalent of BSD socket interface**. The socket module provides various objects, constants, functions and related exceptions for building full-fledged network applications including client and server programs.

3.3.10 Subprocess

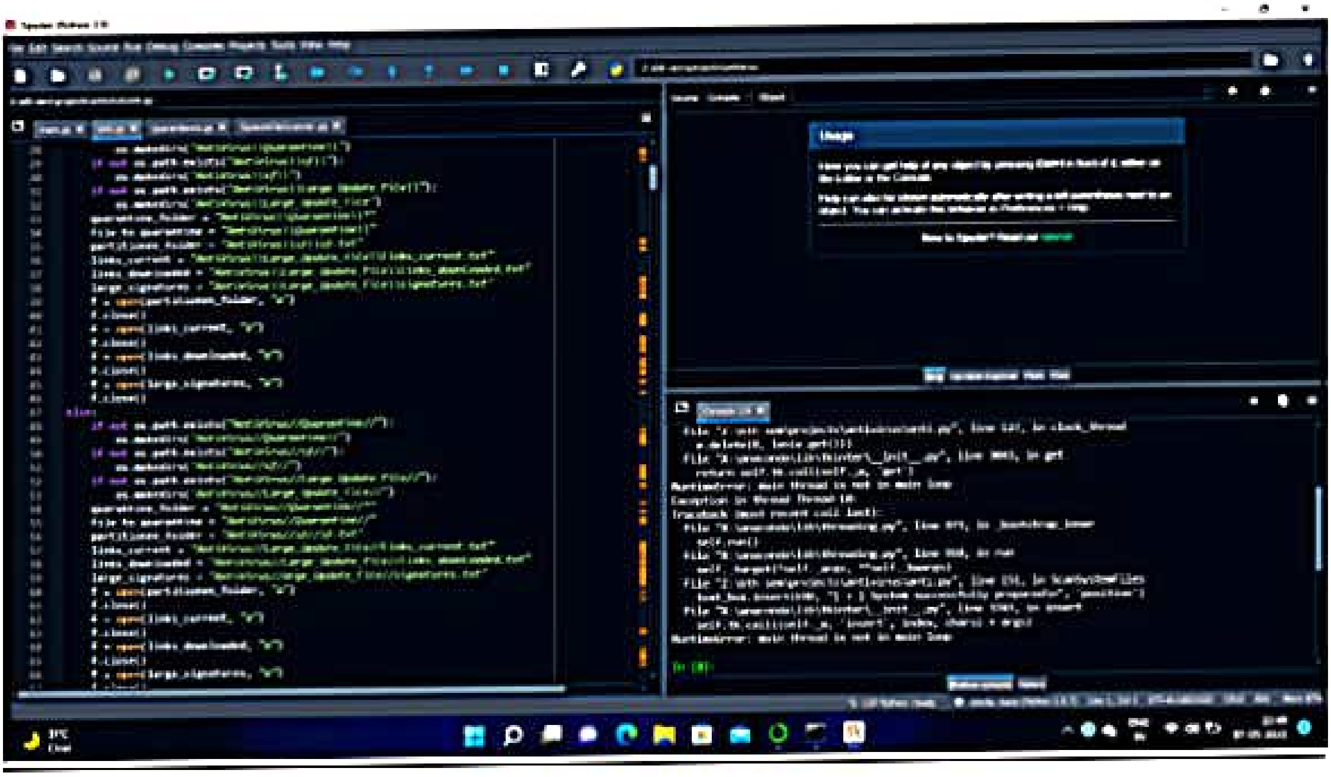
Subprocess in Python **is a module used to run new codes and applications by creating new processes**. It lets you start new applications right from the Python program you are currently writing.

3.3.11 Base64

In Python the base64 module is **used to encode and decode data**. First, the strings are converted into byte-like objects and then encoded using the base64 module. The below example shows the implementation of encoding strings isn't base64 characters.

3.4 Figures :

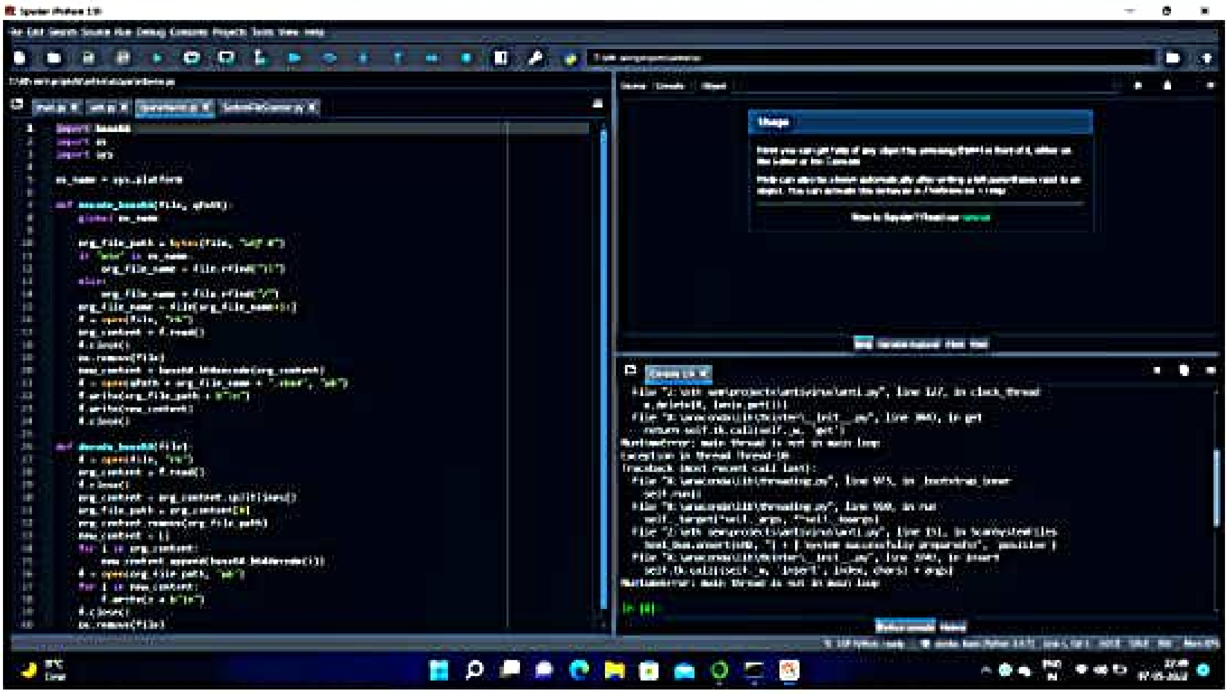
Program Code 1 :



```
        os.makedirs([illegible])
    if not os.path.exists([illegible]):
        os.makedirs([illegible])
    if not os.path.exists([illegible]):
        os.makedirs([illegible])
    quarantine_folder = [illegible]
    file_to_quarantine = [illegible]
    partitiones_folder = [illegible]
    links_current = [illegible]
    links_downloaded = [illegible]
    large_signatures = [illegible]
    f = open(partitiones_folder, "w")
    f.close()
    f = open(links_current, "w")
    f.close()
    f = open(links_downloaded, "w")
    f.close()
    f = open(large_signatures, "w")
    f.close()
else:
    if not os.path.exists([illegible]):
        os.makedirs([illegible])
    if not os.path.exists([illegible]):
        os.makedirs([illegible])
    if not os.path.exists([illegible]):
        os.makedirs([illegible])
    quarantine_folder = [illegible]
    file_to_quarantine = [illegible]
    partitiones_folder = [illegible]
    links_current = [illegible]
    links_downloaded = [illegible]
    large_signatures = [illegible]
    f = open(partitiones_folder, "w")
    f.close()
    f = open(links_current, "w")
    f.close()
    f = open(links_downloaded, "w")
    f.close()
    f = open(large_signatures, "w")
```

Fig 3.2 (Main Module)

Program Code : 2

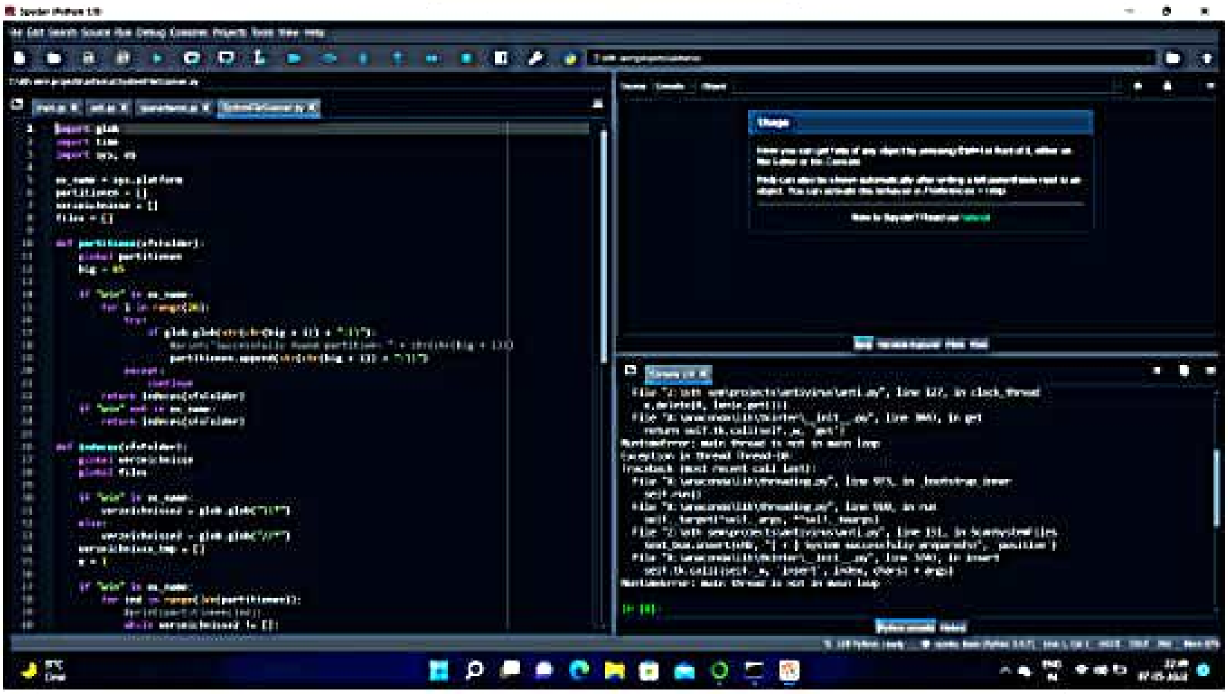


```
import base64
import os
import sys

os_name = sys.platform

def encode_base64(file, qPath):
    global os_name

    org_file_path = bytes(file, "utf-8")
    if "win" in os_name:
        org_file_name = file.rfind("\\")
    else:
        org_file_name = file.rfind("/")
    org_file_name = file[org_file_name+1:]
    f = open(file, "rb")
    org_content = f.read()
    f.close()
    os.remove(file)
    new_content = base64.b64encode(org_content)
    f = open(qPath + org_file_name + "[illegible]", "wb")
    f.write(org_file_path + b"\n")
    f.write(new_content)
    f.close()

def decode_base64(file):
    f = open(file, "rb")
    org_content = f.read()
    f.close()
    org_content = org_content.splitlines()
    org_file_path = org_content[0]
    org_content.remove(org_file_path)
    new_content = []
    for i in org_content:
        new_content.append(base64.b64decode(i))
    f = open(org_file_path, "wb")
    for i in new_content:
        f.write(i + b"\n")
    f.close()
    os.remove(file)
```

```
  File "[illegible]\anti.py", line 127, in clock_thread
    x.delete(0, len(x.get()))
  File "[illegible]\tkinter\__init__.py", line [illegible], in get
    return self.tk.call(self._w, 'get')
RuntimeError: main thread is not in main loop
Exception in thread Thread-10:
Traceback (most recent call last):
  File "[illegible]\threading.py", line 973, in _bootstrap_inner
    self.run()
  File "[illegible]\threading.py", line 910, in run
    self._target(*self._args, **self._kwargs)
  File "[illegible]\anti.py", line 151, in ScanSystemFiles
    text_box.insert(END, "[ + ] System successfully prepared\n", 'position')
  File "[illegible]\tkinter\__init__.py", line 3740, in insert
    self.tk.call((self._w, 'insert', index, chars) + args)
RuntimeError: main thread is not in main loop

In [8]:
```

Fig 3.3(Quearantene Module)

Program code module : 3

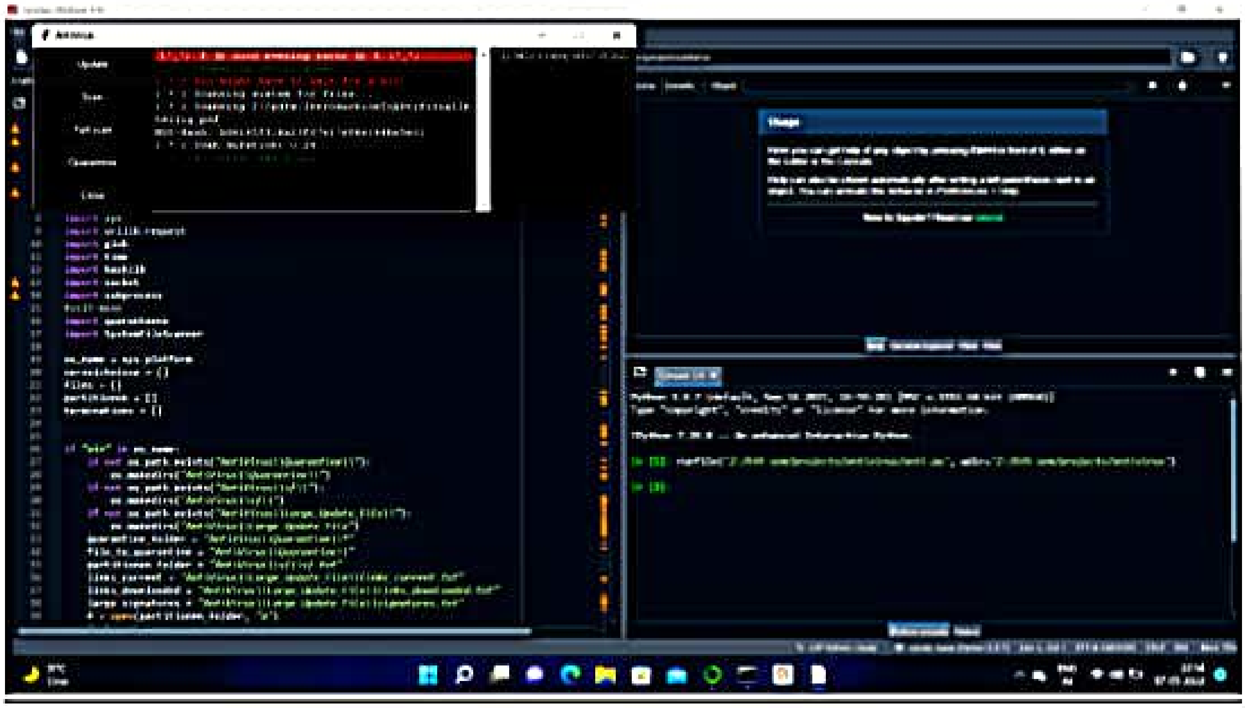


```
import glob
import time
import sys, os

os_name = sys.platform
partitionsx = []
[illegible] = []
files = []

def partitions(sfsfolder):
    global partitionsx
    big = 65

    if "win" in os_name:
        for i in range(26):
            try:
                if glob.glob(str(chr(big + i)) + ":\\"):
                    #print("Successfully found partition: " + str(chr(big + i)))
                    partitionsx.append(str(chr(big + i)) + ":\\")
            except:
                continue
        return indexes(sfsfolder)
    if "win" not in os_name:
        return indexes(sfsfolder)

def indexes(sfsfolder):
    global [illegible]
    global files

    if "win" in os_name:
        [illegible] = glob.glob("\\*")
    else:
        [illegible] = glob.glob("/*")
    [illegible]_tmp = []
    x = 1

    if "win" in os_name:
        for ind in range(len(partitionsx)):
            #print(partitionsx[ind])
            while [illegible] != []:
```

```
  File "[illegible]\anti.py", line 127, in clock_thread
    x.delete(0, len(x.get()))
  File "[illegible]\tkinter\__init__.py", line [illegible], in get
    return self.tk.call(self._w, 'get')
RuntimeError: main thread is not in main loop
Exception in thread Thread-10:
Traceback (most recent call last):
  File "[illegible]\threading.py", line 973, in _bootstrap_inner
    self.run()
  File "[illegible]\threading.py", line 910, in run
    self._target(*self._args, **self._kwargs)
  File "[illegible]\anti.py", line 151, in ScanSystemFiles
    text_box.insert(END, "[ + ] System successfully prepared\n", 'position')
  File "[illegible]\tkinter\__init__.py", line 3740, in insert
    self.tk.call((self._w, 'insert', index, chars) + args)
RuntimeError: main thread is not in main loop

In [8]:
```

Fig 3.4(SystemFileScanner module)

Output : 1



```
import sys
import urllib.request
import glob
import time
import hashlib
import socket
import subprocess
[illegible]
import quarantaene
import SystemFileScanner

os_name = sys.platform
verzeichnisse = []
files = []
partitionen = []
terminations = []

if "win" in os_name:
    [illegible]
```

Fig 3.5(system scan output)

Output :2

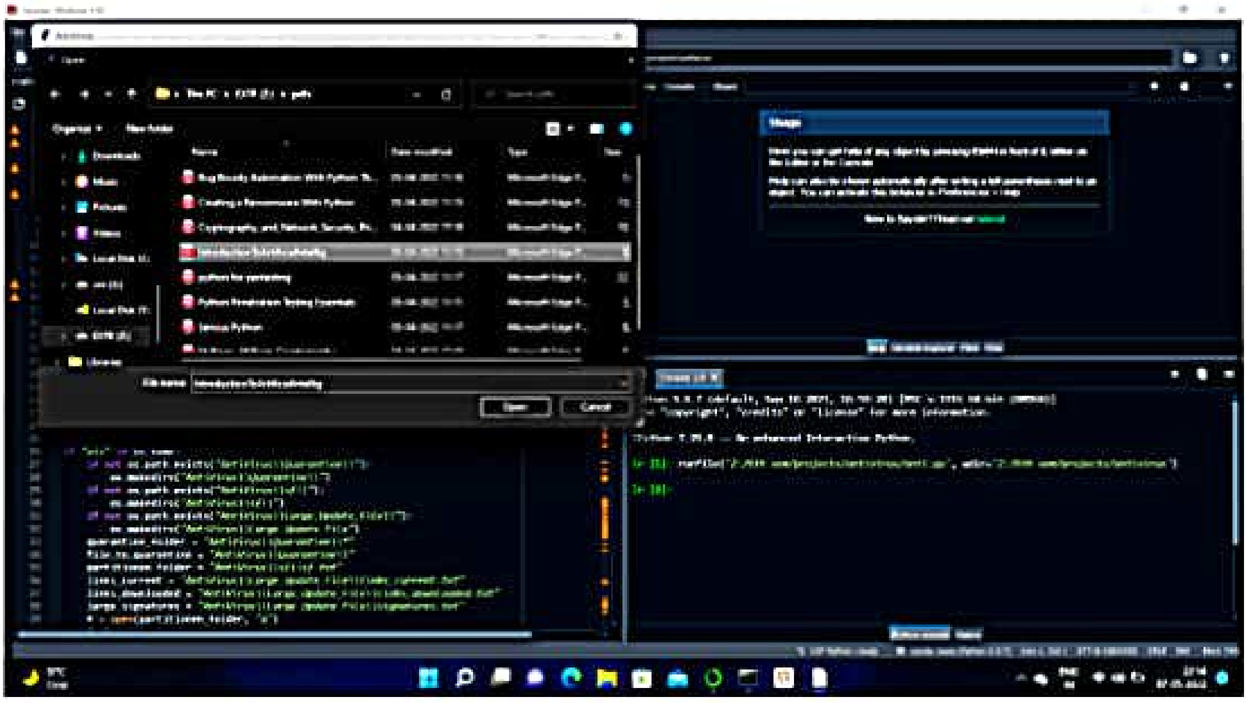


Fig 3.7 (output)

Output : 3

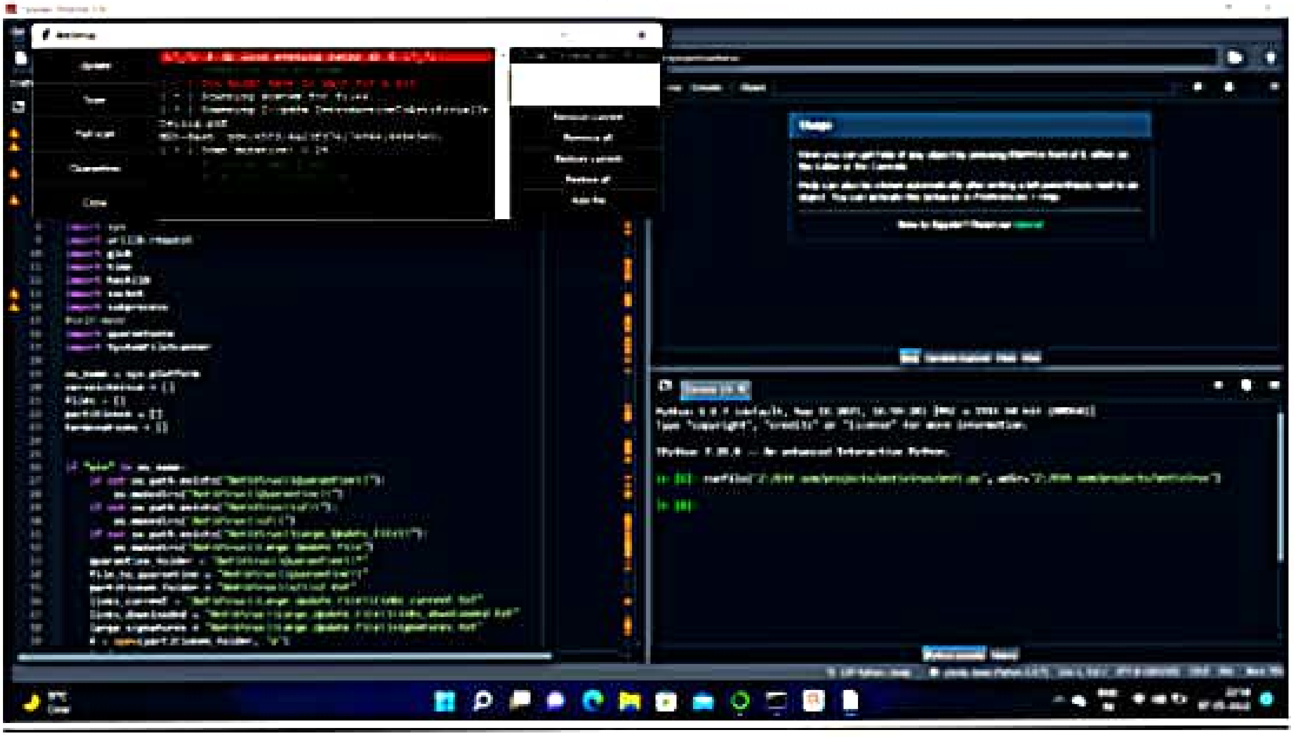


Fig 3.6 (system scan output)

CHAPTER 4 : FINAL ANALYSIS

This software perform a good plateform for virus protection From malicious activities and threats for a computer laptop or mobile plateform this is very useful as usual because of as of there are many increased development of emerging technologies as well as technologies experts , who compromised another systems for gain unauthorized access and other malicious activities like harvest username or other administrator level authentication , cookies , spying , disclose personel credentials as well as personel pics or videos . these activities make very increasing varity of cybercrimes .
Antivirus is a good plateform for prevention from crime against users.

Antivirus software utilizes several methodologies in scanning, detecting, and protecting computers and systems from viruses. As understanding increases about the vectors malicious code uses to attack and how antivirus software protects computer systems from the viruses, people will be able to more effectively help in creating an environment that is secure and virus free. This paper examines the techniques of signature detection, heuristics, and general decryption that antivirus software uses to detect and clean viruses.

4.1 Result :

Antivirus protect our system from scanning whole system , also a single file scan for malicious thread , and quarantine or delete file in the level of malicious .

4.2 Application :

Antivirus **is a kind of software used to prevent, scan, detect and delete viruses from a computer.** Once installed, most antivirus software runs automatically in the background to provide real-time protection against virus attacks. Every computer /laptop /mobule user who use new computer for working , education or any purpose will must use a application or firewall to protect from harmless activities .

4.3 Limitations :

- Antivirus is a great technology from preventing some malicious activities but have some limitations :
- It works In system files so , malware can harm with some extra advanced new urls and malicious softwares or network traffic .
- There are several other technologies like Firewall, IDS , IPS for preventing malicious activities.

4.4 Problem From Antivirus :

Antivirus or these types of software having administrator privileges . so if anyone can break these software so they have direct access to everything in a computer .

CHAPTER 5 : CONCLUSION

5.1 Conclusion

In conclusion, all antivirus software has its strengths and weaknesses. Each customer willneed to decide which software is best for them in meeting their needs. Even if we acquire the best antiviral software today, it may be of little value in the future because of the rapidchanges in cybersecurity and hackers' ability to engage new technology. We must continuously reassess our needs for protection in a changing world.

References :

Book 1 - Mastering Python for Networking and Security Leverage the scripts and libraries of Python version 3.7 and beyond to overcome networking and security issues, 2nd Edition by Jose Manuel Ortega (2021)

Book 2 - Introduction to Modern Cryptography, 3rd Edition by Jonathan Katz, Yehuda Lindell (2020)