

# SP PROJECT

## MISSION

CHRIST is a nurturing ground for an individual's holistic development to make effective contribution to the society in a dynamic environment

## VISION

Excellence and Service

## CORE VALUES

Faith in God | Moral Uprightness  
Love of Fellow Beings  
Social Responsibility | Pursuit of Excellence



## Team:-

- ACHINTYA VATSRAJ(1941101)
- KAUSHIK(1941116)
- KEERTHAN GOWDA M(1941117)
- VIJETHA SHRAVYA(1941166)

Under the guidance of Ummesalma M

### MISSION

CHRIST is a nurturing ground for an individual's holistic development to make effective contribution to the society in a dynamic environment

### VISION

Excellence and Service

### CORE VALUES

Faith in God | Moral Uprightness  
Love of Fellow Beings  
Social Responsibility | Pursuit of Excellence

# Our Domain:- Client Side Cyber Security



# Introduction:

Computers are an important part of everyday life to many people across the world. Computers in the hands of consumers who lack the knowledge of protection tools and who have limited administrator skills are vulnerable to virus attacks. A recent trend in attacks has been the attempt to disable security protocols in place at the host machine. This type of attack leaves the host computer completely defenseless and vulnerable to many further exploits through the Internet. To ensure the continuous functioning of the security protocols, a software-based solution is proposed in this thesis. AI and machine learning antivirus leverage sophisticated mathematical algorithms combined with the data from other deployments to understand what the baseline of security is for a given system.

# Objective:-

1. To generate a new, effective framework to analyse the behavioural characteristics of the malware family and the mechanics of its operation.
2. To develop a malware detection Toolkit, consisting of a simple structural prototype built from open source software, Implement scanning and content discovery functionality using Python.
3. To conduct an extensive literature review of the different trials associated with malware detection and to identify the parameters for developing the most efficient tools for a live system
4. Develop an ecosystem of Apps across platforms to provide concrete client side security(such as Desktop GUI(tkinter),Mobile App,Chrome Extension.)

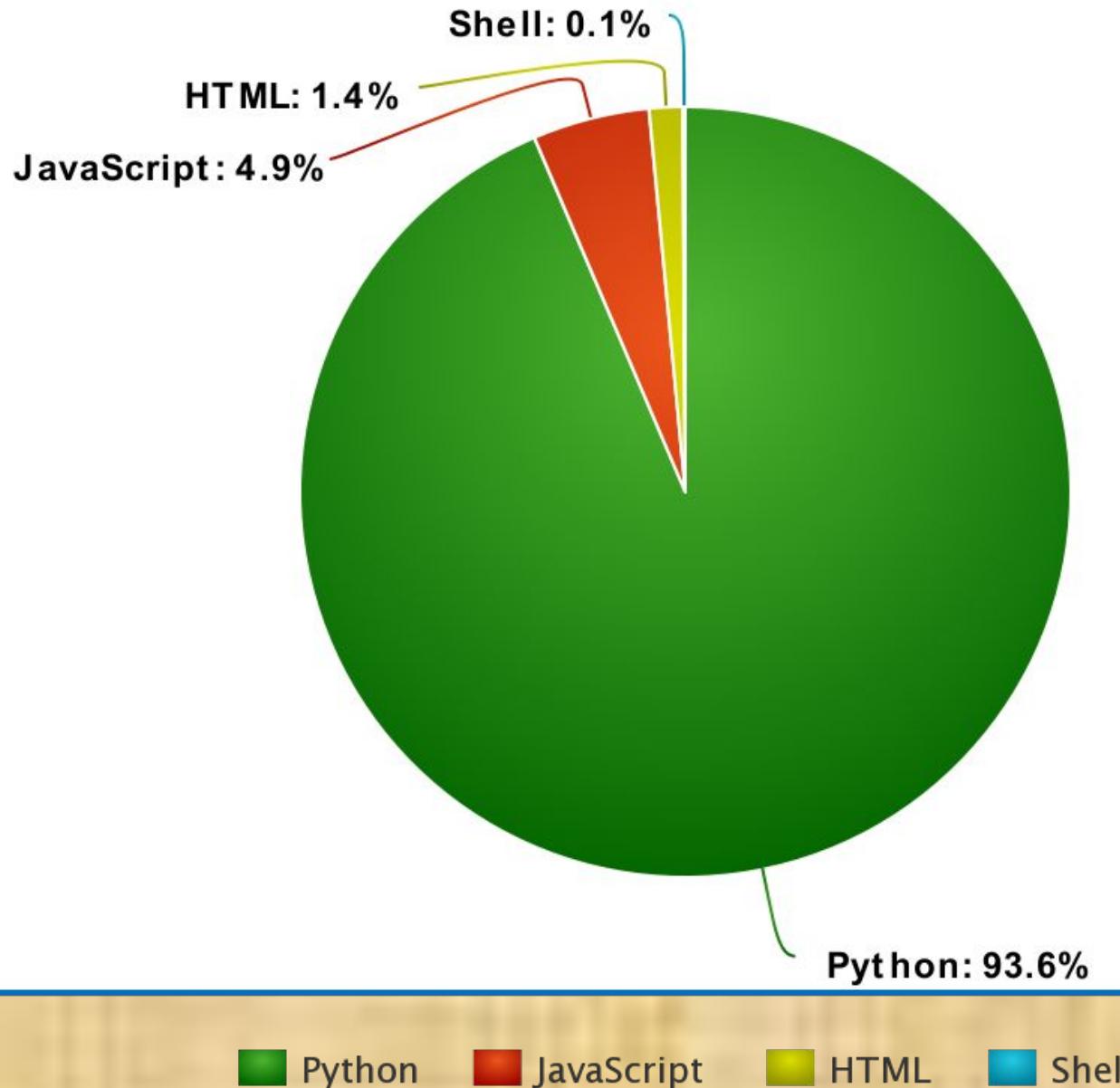
## SYSTEM ANALYSIS:-

The Existing Client Side Security solutions do feature one or the other type of detection and Prevention techniques but they do not for the most part try to mix and match the said methods, and no product on the market has both an extension and mobile application to go with their desktop security solution.

Our Project is using the SQL Database Architecture to store data in relational models.

## Components of the Project:

1. Desktop Application
2. Backend System
3. AI/ML Model
4. Chrome Extension
5. Mobile APP(NOT YET IMPLEMENTED)



## PROFILE OF THE PROBLEM

If you are using a free antivirus program, there is no guarantee that it will provide you the complete protection. Most free antivirus programs out there only offer a basic level of protection. Moreover they are capable of identifying only certain types of threats. In order for acquiring complete level of protection, you have to use a firewall as well. Using an antivirus program means that a lot of resources from the memory and the hard drive is being used. As a result it can drastically slow down overall speed of the computer. Moreover, the process of scanning can also cause lags in the network.

# Gantt chart

Start Date	End Date	Description	Feature	Duration(Rough)
		Initial Documentation and Prepatory		
31-07-21	03-08-21	Phase	Basic Python and related technology installation and Port scanner	4
04-08-21	08-08-21	Phase - 1	Firewall,Raw Socket Sniffer and Basic GUI	5
09-08-21	15-08-21	Phase - 1	Behaviour and Signature Malware Detection	7
16-08-21	04-09-21	Phase - 1	Design and Build Model for Heuristic-Based Detection With DBMS	20
05-09-21	12-09-21	Phase - 1	Process Monitoring and AntiMalware Testing	8
13-09-21	25-09-21	Phase - 1	Sandbox Environment,Web Protection,Threat Score Calculation	13
26-09-21	15-10-21	Phase - 2	Backup Restore,API dev,Cloud Engine Setup,Improve GUI	20
16-10-21	25-10-21	Phase - 3	Further Model Testing and Detailed Documentation	10
26-10-21	31-10-21	Phase - 3	Improve application Logic and develop simple Mobile App with DashBoard(connect with API)	6
01-11-21	10-11-21	Phase - 4	Testing,Demo,Accuracy Data Collection through parallel Container Tests	10
11-11-21	21-11-21	Phase - 4	Publish Product on GitHub and Finish Documentation , in-depth paper on Model as well as Finalise GUI colors	11

# SOFTWARE REQUIREMENTS

## At Developer Side

### Software Used

- HTML
- VISUAL STUDIO CODE
- PYTHON
- SQLITE3 DATABASE
- JAVASCRIPT
- C / Assembly

## At System Users Side

### Software Requirements

- Browser (IE 7.0 or Above, Mozilla Firefox, Google Chrome)
- Browser Must be JavaScript Enabled

# HARDWARE REQUIREMENTS

## At Developer Side

Hardware Used

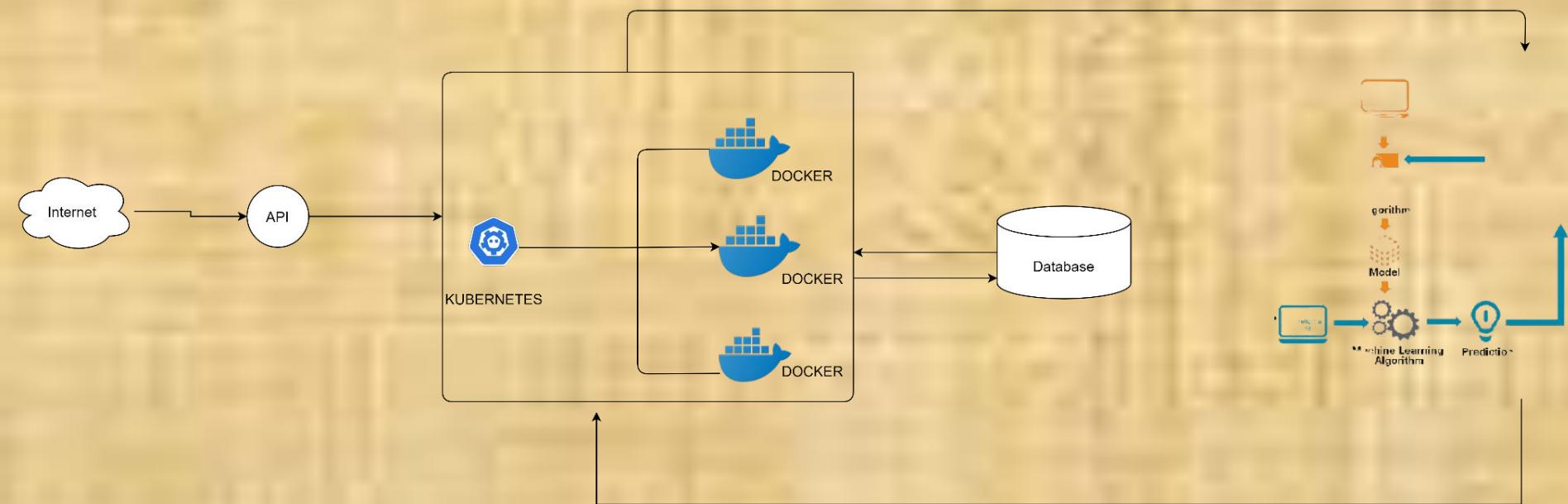
- Intel core i5
- Atleast 16 GB RAM.
- O.S. – Windows 10 or \*Nix

## At System Users Side

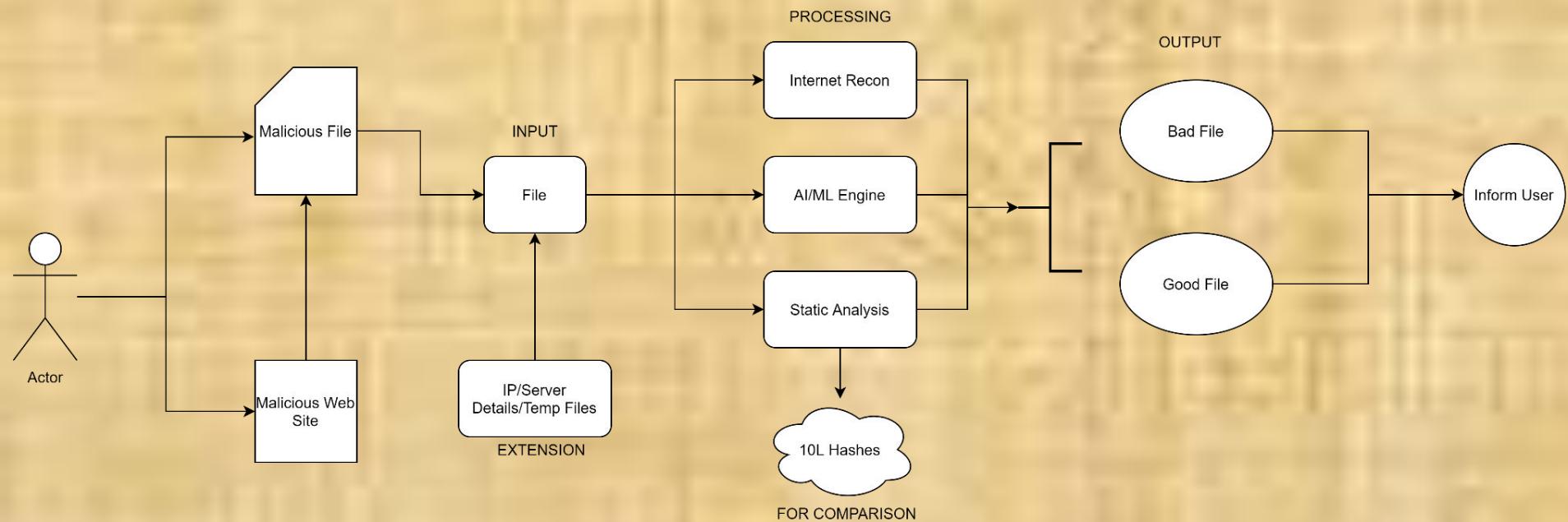
Hardware Requirements

- Intel Core i3
- O.S. – Windows 10 or \*Nix

# Block diagram:-



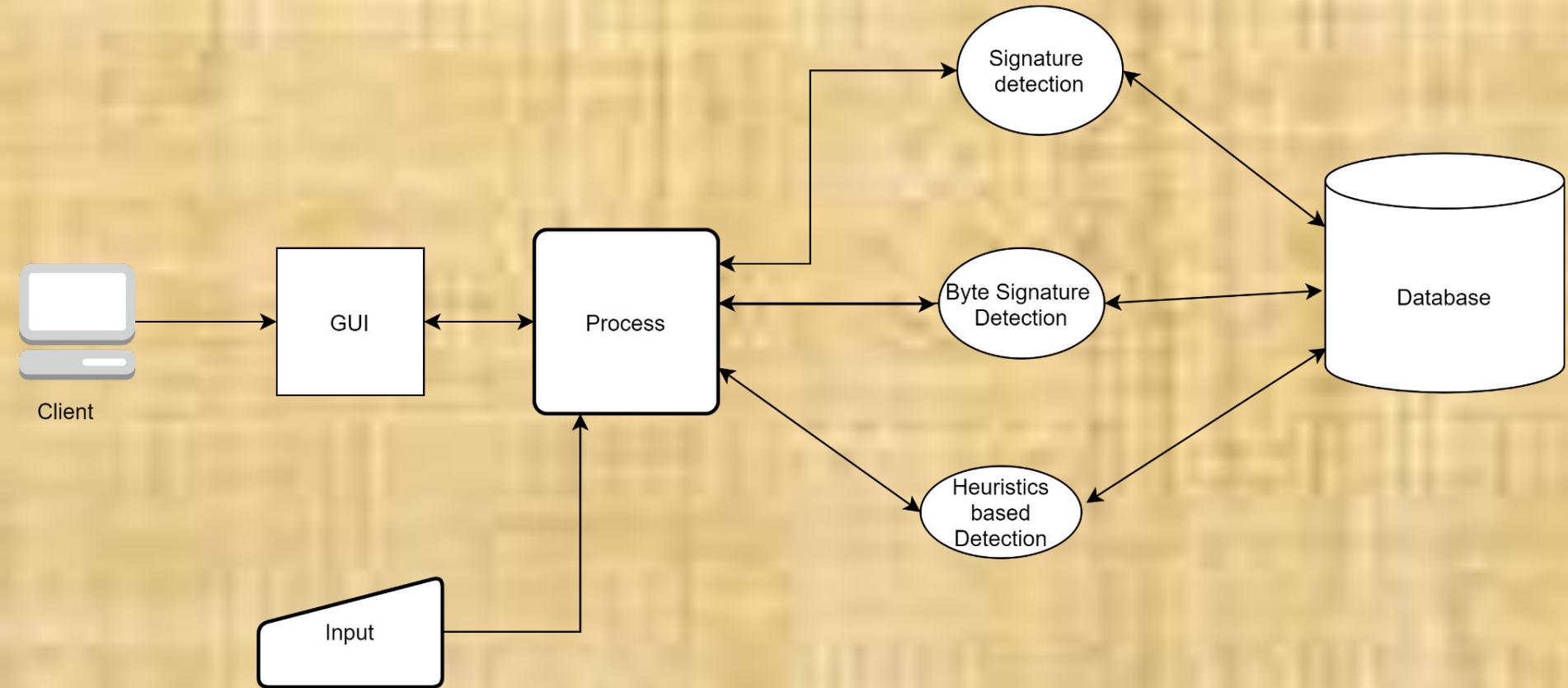
# Data Flow Diagram



# 4 tier architecture

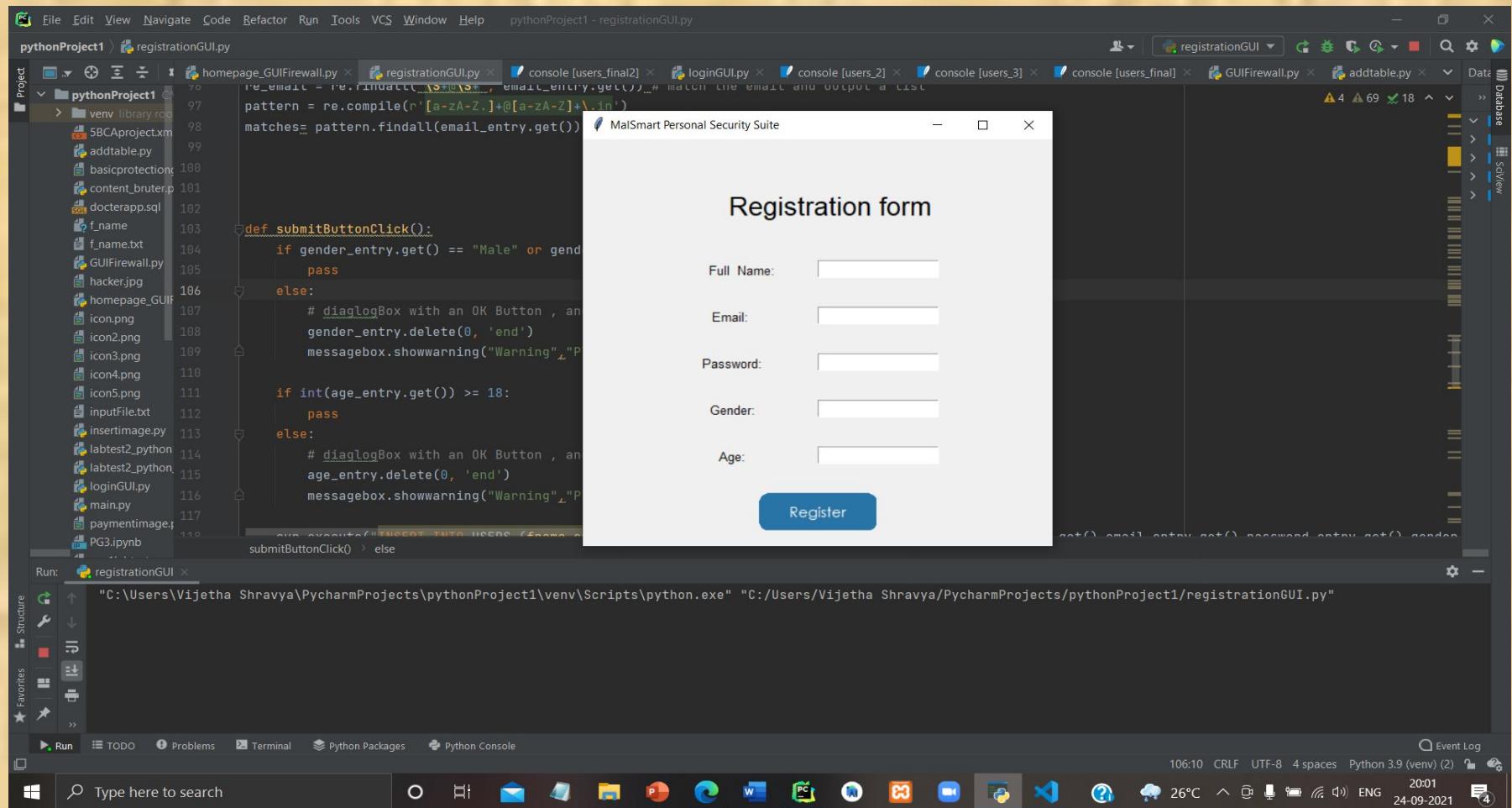


# malware detection cycle

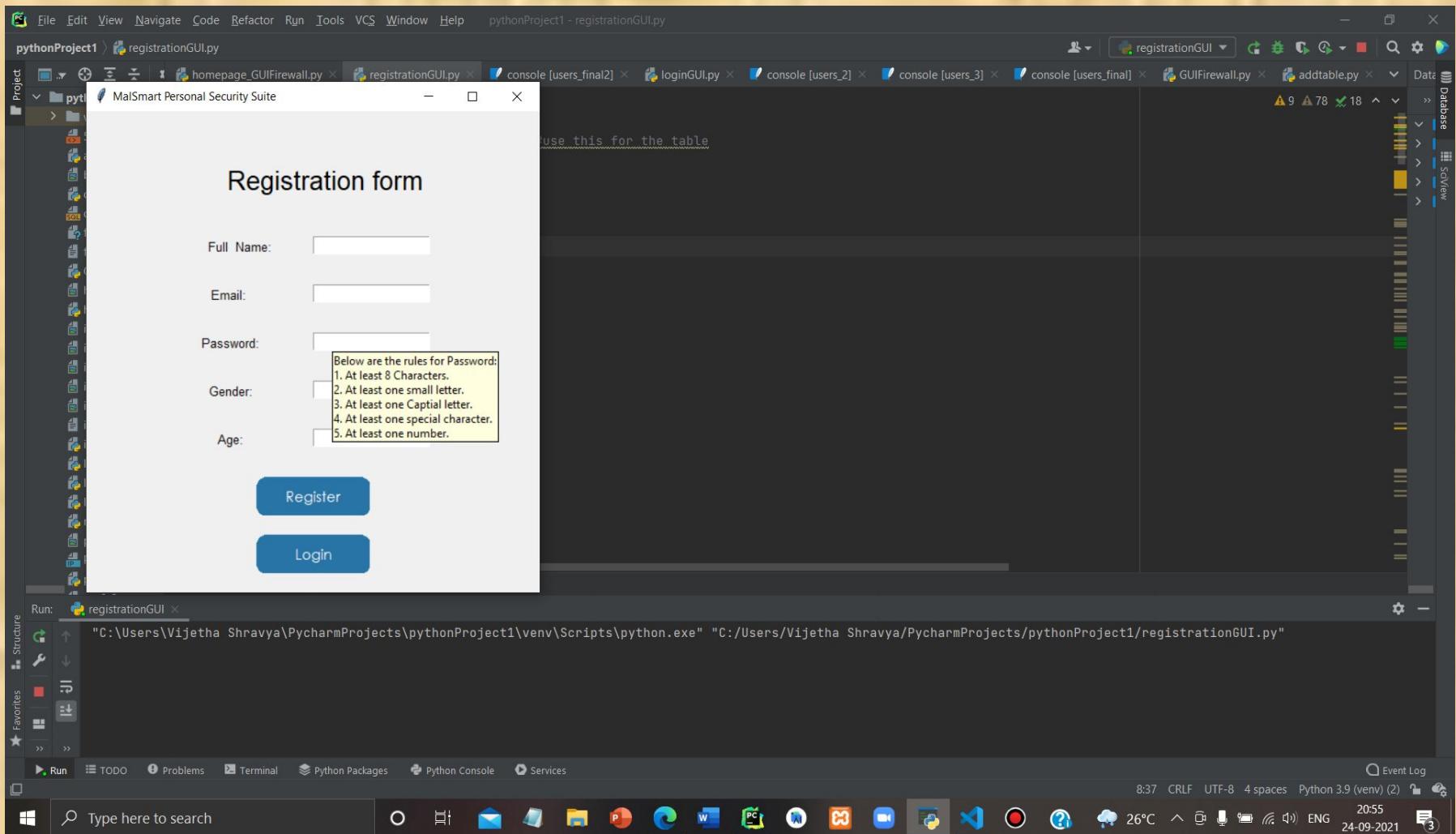


# GUI using python tkinter

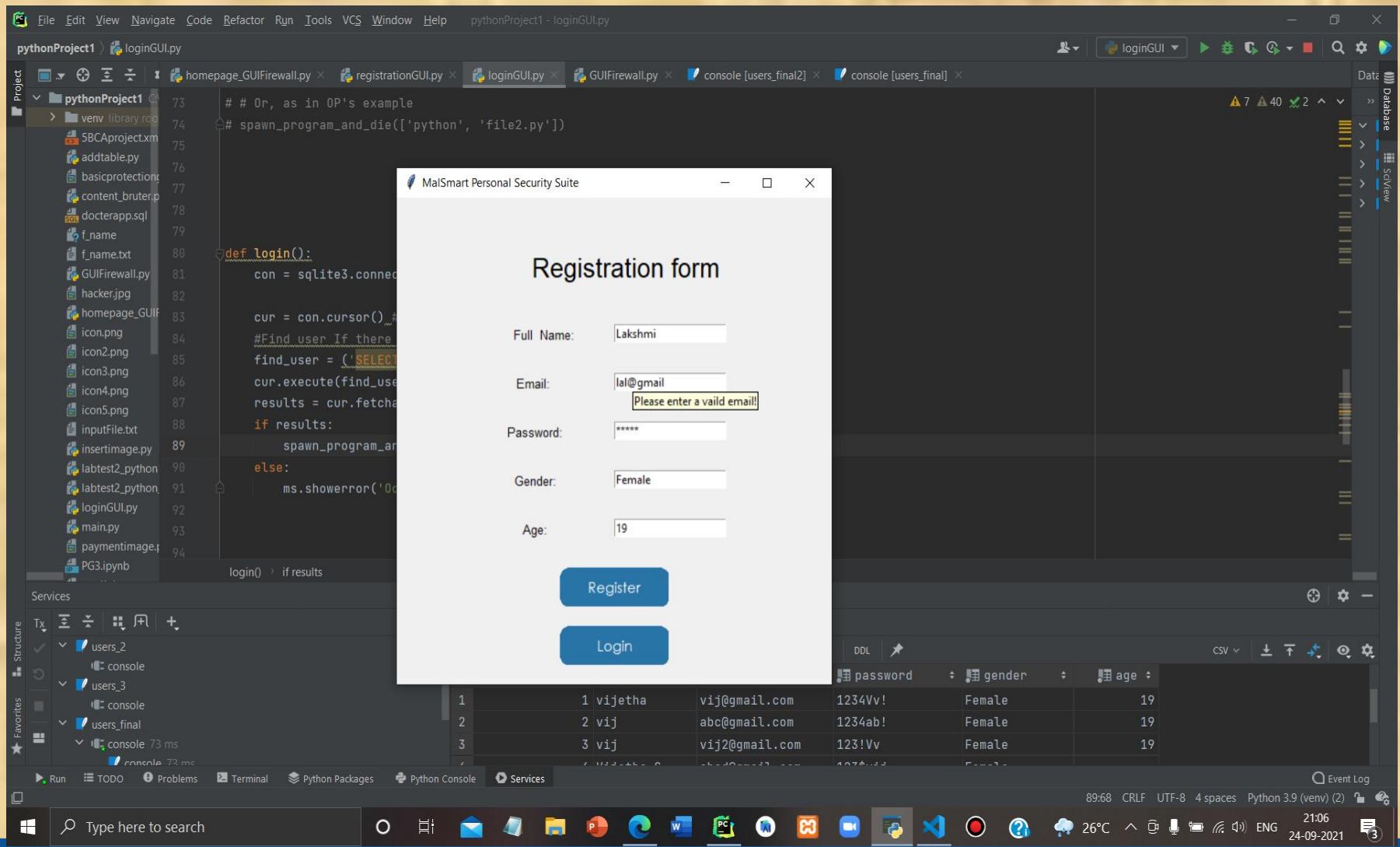
Screenshot 1:



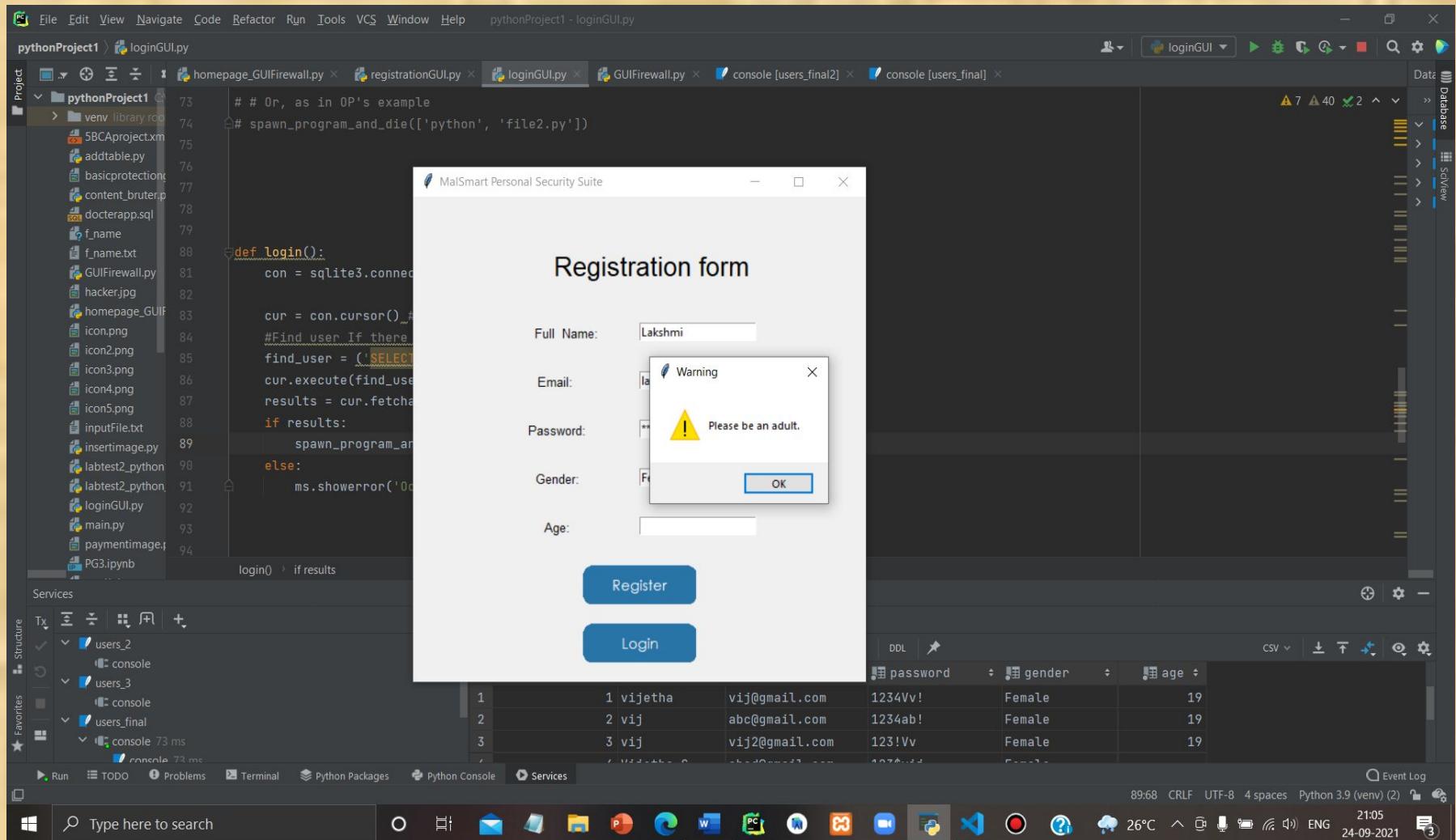
## Screenshot 2:



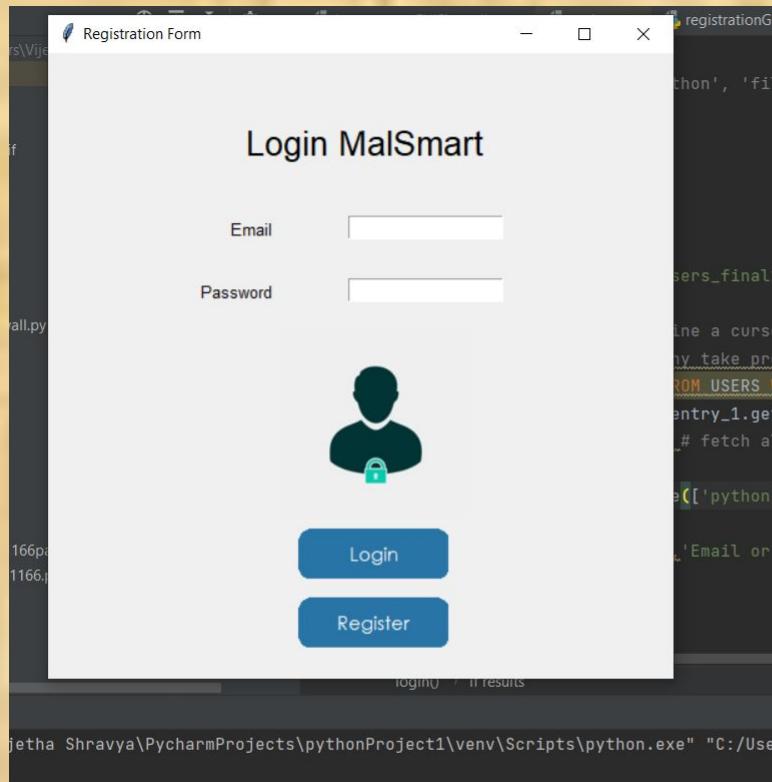
# Screenshot 3:



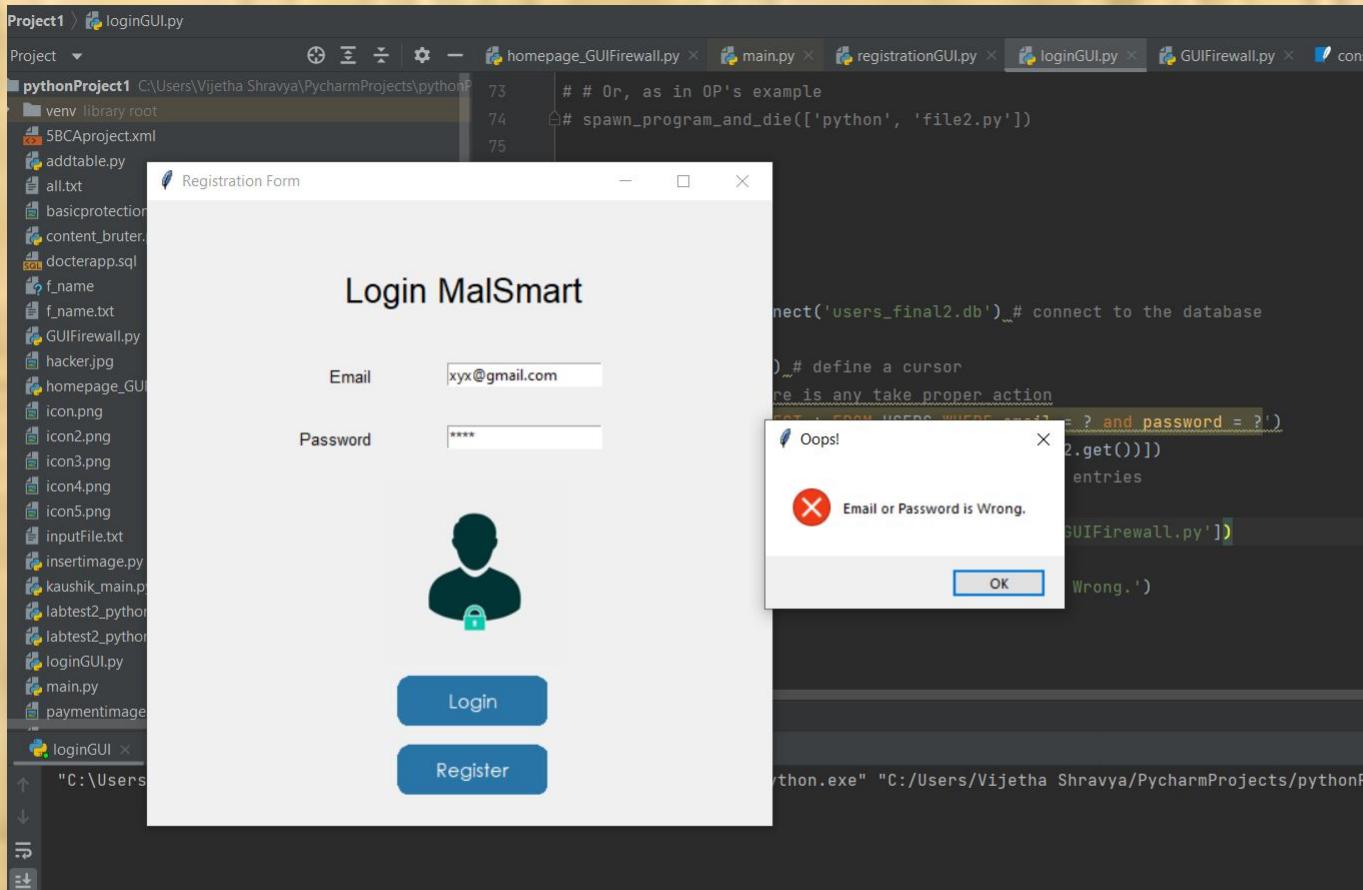
# Screenshot 4:



# Screenshot :



# Screenshot :



# Screenshot 5:

The screenshot shows a PyCharm IDE interface with a Python project named 'pythonProject1'. The 'Database' tool window is open, displaying a query in the SQL tab:

```
SELECT * FROM USERS;
```

The results are shown in the SQL View tab, listing five users from a table named 'main.USERS':

USER_ID	fname	email	password	gender	age
1	vijetha	vij@onmail.com	1234Vv!	Female	19
2	vij	a_email text.com	1234ab!	Female	19
3	vij	vij2@gmail.com	123!Vv	Female	19
4	Vijetha S	abcd@gmail.com	123\$vij	Female	19
5	Lakshmi	lal@gmail.com	1234L	Female	19

The Services tool window on the left shows several database connections, including 'users\_2', 'users\_3', 'users\_final', and 'users\_final2'. The bottom status bar indicates Python 3.9 (venv) (2), 21:05, 26°C, ENG, 24-09-2021, and a notification count of 3.

## Screenshot 6:

MalSmart Dashboard

File Help

### You have basic protection

The dashboard features a 2x4 grid of blue rectangular buttons, each containing a white text label representing a protection module:

Firewall	Web	Mobile	VPN
Update	Online scan	Sandbox	Scan

# Screenshot :

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface with the following details:

- Explorer View:** Shows the project structure under "UNTITLED (WORKSPACE)". Files listed include: pythonProject1, \_pycache\_, .idea, venv, 5BCAproject.xml, addtable.py, all.txt, basicprotectiongif.gif, content\_bruter.py, docterapp.sql, f\_name, f\_name.txt, GUIFirewall.py, hacker.jpg, homepage\_GUILfire... (partially visible), icon.png, icon2.png, icon3.png, icon4.png, icon5.png, inputFile.txt, insertimage.py, kaushik\_main.py, labtest2\_python\_194..., labtest2\_python194..., loginGUI.py, main.py, paymentimage.png, PG3.ipynb, prg1labtest.py, prg2\_2.py.
- Editor View:** The main editor window displays the content of content\_bruter.py. The code implements a directory brute-force attack using a queue-based approach. It checks for file extensions and iterates over a list of attempts to build a wordlist and check URLs.
- Status Bar:** At the bottom, the status bar shows the following information:
  - File Path: n 3.9.6 64-bit (venv: venv)
  - File Name: content\_bruter.py
  - Line/Column: Ln 6, Col 13
  - Text Encoding: Spaces: 4
  - File Type: UTF-8
  - Line Endings: CRLF
  - Language: Python
  - Date/Time: 08:55
  - Date: 25-09-2021

# Screenshot :

The screenshot shows a Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** tkinter\_custom\_button.py - Untitled (Workspace) - Visual Studio Code.
- Explorer:** Shows the project structure under UNTITLED (WORKSPACE). Files listed include: icon4.png, icon5.png, inputFile.txt, insertimage.py, kaushik\_main.py, labtest2\_python\_194..., labtest2\_python194..., loginGUI.py, main.py, paymentimage.png, PG3.ipynb, prg1labtest.py, prg2\_2.py, prg2labtest.py, register imageGUI.jp..., registerbuttonIMAG..., registrationGUI.py, securityimage.jpeg, settingsimage.png, soutput.xml, stackprg.py, tickimage.png, tkinter\_custom\_but..., tokyoprg.ipynb, users\_2.db, users\_3.db, users\_final.db, users\_final2.db, users.db, xmlhandling194116..., registrationGUI.py, loginGUI.py, content\_bruter.py, all.txt, homepage\_G...
- Code Editor:** Displays Python code for a custom Tkinter button.

```
pythonProject1 > tkinter_custom_button.py > TkinterCustomButton
1     import tkinter
2     import sys
3
4
5     class TkinterCustomButton(tkinter.Frame):
6         """ tkinter custom button with border, rounded corners and hover effect
7
8             Arguments: master= where to place button
9             bg_color= background color, None is standard,
10            fg_color= foreground color, blue is standard,
11            hover_color= foreground color, lightblue is standard,
12            border_color= foreground color, None is standard,
13            border_width= border thickness, 0 is standard,
14            command= callback function, None is standard,
15            width= width of button, 110 is standard,
16            height= width of button, 35 is standard,
17            corner_radius= corner radius, 10 is standard,
18            text_font= (<Name>, <Size>),
19            text_color= text color, white is standard,
20            text= text of button,
21            hover= hover effect, True is standard,
22            image= PIL.PhotoImage, standard is None"""
23
24     def __init__(self,
25                  bg_color=None,
26                  fg_color="#2874A6",
27                  hover_color="#5499C7",
28                  border_color=None,
29                  border_width=0,
30                  command=None,
31                  width=120,
32                  height=40,
33                  corner_radius=10,
34                  text_font=None,
35                  text_color="white",
36                  text="CustomButton",
37                  hover=True,
38                  image=None)
```

The screenshot shows a Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** content\_bruter.py - Project 1 - Visual Studio Code
- Left Sidebar:** Includes icons for Welcome, content\_bruter.py, and all.txt.
- Code Editor:** Displays a Python script named content\_bruter.py. The code iterates over a list of attempts, constructs URLs with formatted parameters, sends requests, and prints the response code. It handles URL errors and 404 responses.
- Bottom Navigation:** PROBLEMS, OUTPUT, TERMINAL (selected), SQL CONSOLE, DEBUG CONSOLE.
- Terminal Output:** Shows a series of [200] status codes for various URLs on the testphp.vulnweb.com domain.
- Bottom Status Bar:** Python 3.9.7 64-bit, 0 errors, 0 warnings, Connect, AchintyaVatsraj, Live Share, Line 5, Column 1, Spaces: 4, UTF-8, CRLF, MagicPython, Go Live, Prettier.

The screenshot shows the PyCharm IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help.
- Project Tab:** Project 1 - main.py
- Code Editor:** Displays Python code for a system scan. The code includes methods for listing files on drives, listing available disk drives, performing a full scan, and updating hash versions. It uses the glob module for file operations and the re module for version number extraction.
- Toolbars and Status Bar:** Includes tabs for main.py, version.txt, and a progress bar indicating "Indexing...". The status bar at the bottom shows "Indexing Python SDK 'Python 3.9'" and "Event Log".
- Sidemenu and Bottom Bar:** Includes sections for Project, Favorites, Structure, TODO, Problems, Terminal, Python Packages, Python Console, and Event Log.

```
1399 class systemscan:
1400     def list_all_file_scan(self,drive):
1401         a = 1
1402         for file_name in glob.iglob('%s/Program Files/**/*' % drive, recursive=True):
1403             # print(file_name)
1404             # for file_name in glob.iglob("%s/Users/**/*" % drive, recursive=True):
1405             # for file_name in glob.iglob('C:/Users/kaush/PycharmProjects/antimalware/test/**/*.exe', recursive=True):
1406                 Static_Analysis.file_hash(Static_Analysis,file_name)
1407
1408     def list_disk_available(self):
1409         dl = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
1410         drives = ['%s:' % d for d in dl if os.path.exists('%s:' % d)]
1411         return drives
1412
1413     def full_scan(self):
1414         drives = systemscan.list_disk_available(systemscan)
1415         for i in drives:
1416             systemscan.list_all_file_scan(systemscan,i)
1417
1418 # /////////////////////////////////
1419 print("Test")
1420
1421 class Static_Analysis:
1422     def update_hash(self,latest_version):
1423         version_list = []
1424         for file_name in glob.iglob('./hashes/*.txt', recursive=True):
1425             a = ((re.findall("[0-9]+",str(file_name))))
1426             version_list.append(int(a[0]))
1427         version = (max(version_list))
1428         if version < latest_version:
1429             systemscan > list_all_file_scan() > for file_name in glob.iglob('%s...'
```

The screenshot shows a PyCharm IDE interface with a terminal window displaying the results of a security audit script. The terminal output lists various files and their status:

```
C:/Program Files/7-Zip/bwapp/ba_insecure_login_1.php Good file
C:/Program Files/7-Zip/bwapp/ba_insecure_login_2.php Good file
C:/Program Files/7-Zip/bwapp/ba_insecure_login_3.php Good file
C:/Program Files/7-Zip/bwapp/ba_logout.php Good file
C:/Program Files/7-Zip/bwapp/ba_logout_1.php Good file
C:/Program Files/7-Zip/bwapp/ba_pwd_attacks.php Good file
C:/Program Files/7-Zip/bwapp/ba_pwd_attacks_1.php Good file
C:/Program Files/7-Zip/bwapp/ba_pwd_attacks_2.php Good file
C:/Program Files/7-Zip/bwapp/ba_pwd_attacks_3.php Good file
C:/Program Files/7-Zip/bwapp/ba_pwd_attacks_4.php Good file
C:/Program Files/7-Zip/bwapp/ba_weak_pwd.php Good file
C:/Program Files/7-Zip/bwapp/bugs.txt Good file
C:/Program Files/7-Zip/bwapp/bugs_low_sev.txt Good file
C:/Program Files/7-Zip/bwapp/captcha.php Malicious file found
C:/Program Files/7-Zip/bwapp/captcha_box.php Good file
C:/Program Files/7-Zip/bwapp/clickjacking.php Good file
C:/Program Files/7-Zip/bwapp/commandi.php Good file
C:/Program Files/7-Zip/bwapp/commandi_blind.php Good file
C:/Program Files/7-Zip/bwapp/config.inc Good file
C:/Program Files/7-Zip/bwapp/config.inc.php Good file
C:/Program Files/7-Zip/bwapp/connect.php Good file
C:/Program Files/7-Zip/bwapp/connect_i.php Good file
C:/Program Files/7-Zip/bwapp/credits.php Good file
C:/Program Files/7-Zip/bwapp/csrf_1.php Good file
C:/Program Files/7-Zip/bwapp/csrf_2.php Good file
C:/Program Files/7-Zip/bwapp/csrf_3.php Good file
C:/Program Files/7-Zip/bwapp/cs_validation.php Good file
C:/Program Files/7-Zip/bwapp/directory_traversal_1.php Good file
C:/Program Files/7-Zip/bwapp/directory_traversal_2.php Good file
C:/Program Files/7-Zip/bwapp/documents/bWAPP_intro.pdf Good file
```

The PyCharm interface includes a Project tool window on the left, a Run tool bar at the bottom, and various status indicators at the bottom right.

# Screenshot :



hash1.txt



hash2.txt



hash3.txt



hash4.txt



hash5.txt



hash6.txt



hash7.txt



hash8.txt

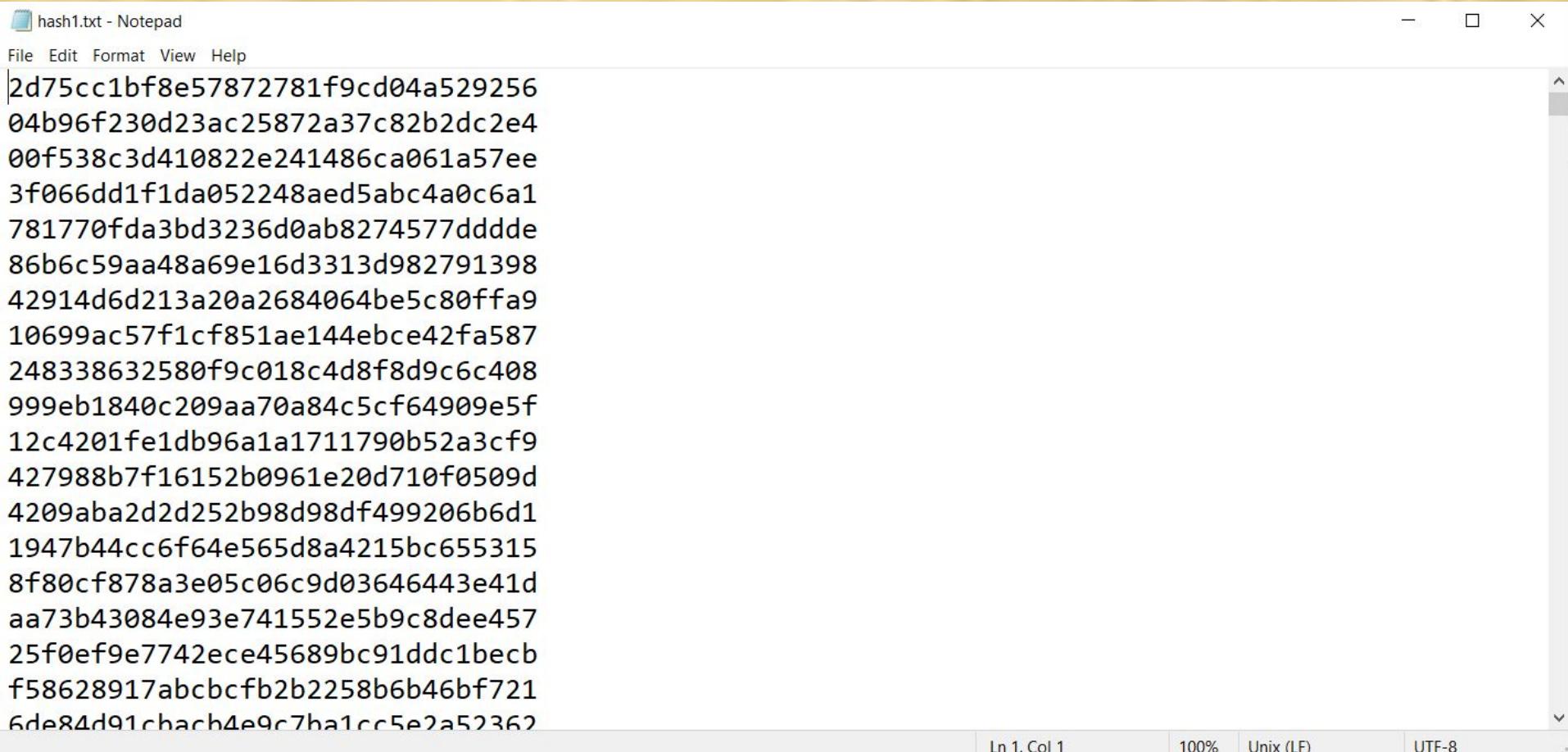


hash9.txt



hash10.txt

# Screenshot :



A screenshot of a Windows Notepad window titled "hash1.txt - Notepad". The window contains a single column of 40 SHA-256 hash strings, each consisting of 64 hex digits. The hashes are separated by new lines. The Notepad interface includes a menu bar with File, Edit, Format, View, and Help, and a status bar at the bottom showing "Ln 1, Col 1", "100%", "Unix (LF)", and "UTF-8".

```
2d75cc1bf8e57872781f9cd04a529256
04b96f230d23ac25872a37c82b2dc2e4
00f538c3d410822e241486ca061a57ee
3f066dd1f1da052248aed5abc4a0c6a1
781770fda3bd3236d0ab8274577dddde
86b6c59aa48a69e16d3313d982791398
42914d6d213a20a2684064be5c80ffa9
10699ac57f1cf851ae144ebce42fa587
248338632580f9c018c4d8f8d9c6c408
999eb1840c209aa70a84c5cf64909e5f
12c4201fe1db96a1a1711790b52a3cf9
427988b7f16152b0961e20d710f0509d
4209aba2d2d252b98d98df499206b6d1
1947b44cc6f64e565d8a4215bc655315
8f80cf878a3e05c06c9d03646443e41d
aa73b43084e93e741552e5b9c8dee457
25f0ef9e7742ece45689bc91ddc1becb
f58628917abcbcfcfb2b2258b6b46bf721
6de84d91chach4e9c7ha1cc5e2a52362
```

# Interesting Fact about Analysis

We have 10L+ Hashes to do comparison that we will regularly improve/update to get better results and accuracy in Analysis.



# Screenshot of Database Creation:

```
### Database Code

con = sqlite3.connect('users_final2.db')

cur = con.cursor()

cur.execute('''CREATE TABLE IF NOT EXISTS USERS(
    USER_ID INTEGER PRIMARY KEY AUTOINCREMENT,
    fname text NOT NULL,
    email text UNIQUE NOT NULL,
    password text NOT NULL,
    gender text NOT NULL,
    age INTEGER NOT NULL
)
''')

con.commit() # add the table if not already existing.

###
```

## Screenshot of Imports for the Project:

```
from tkinter_custom_button import *
from tkinter import *
import sqlite3
import tkinter as tk
import awesometkinter as atk
from idlelib.tooltip import Hovertip
from tkinter import messagebox
from tkinter import messagebox as ms
import re
import subprocess
import sys

from PIL import ImageTk
from PIL import Image
```

## Screenshots:

```
def login():
    con = sqlite3.connect('users_final2.db') # connect to the database

    cur = con.cursor() # define a cursor
    #Find user If there is any take proper action
    find_user = ('SELECT * FROM USERS WHERE email = ? and password = ?')
    cur.execute(find_user,[ (entry_1.get()),(entry_2.get())])
    results = cur.fetchall() # fetch all the found entries
    if results:
        spawn_program_and_die(['python','homepage_GUIFirewall.py'])
    else:
        ms.showerror('Oops!', 'Email or Password is Wrong.' )
```

# Screenshots for Content/Folder BruteForcing:

The screenshot shows a Windows Notepad window titled "all.txt - Notepad". The content of the file is a list of files and folders found in a directory, likely the result of a folder brute-force attack. The list includes:

- common
- CVS
- root
- Entries
- lang
- home.php
- setup.php
- install.txt
- default
- init.php
- Repository
- en.php
- .project
- admin
- FCKeditor
- list.php
- templates\_c
- AUTHORS
- help\_nhn

At the bottom of the Notepad window, there are status bars showing "Ln 43135, Col 13", "100%", "Windows (CRLF)", and "UTF-8".

# Chrome extension screenshots

Screenshot 1:

The screenshot displays a list of installed Chrome extensions and apps. Each item is shown in a card-like format with its icon, name, description, ID, and status controls.

- IP address 1.1.0**  
TO find ip address of website  
ID: lkkmdmaffifgblohedgokhigobpltnf  
Inspect views [background page](#)  
Details Remove C
- Malware 1.0**  
Antivirus  
ID: dbebhgplncnnhmckonkhnmaibmdcoine  
Details Remove C
- Malware Ads blocker 0.6**  
TO Block customized Websites  
ID: anjiocieidbdmefiacloloomkhhmljo  
Inspect views [background page](#)  
Details Remove C
- Vault 1.0**  
A modern, secure and simplified password manager.  
ID: mgklegloiehdbhmdahpepjahddpleoi  
Details Remove C
- Google Docs Offline 1.31.0**  
Edit, create, and view your documents, spreadsheets, and presentations – all without internet access.  
ID: ghbmnjooekpmoeecnnilnnbdlolhkhi  
Inspect views [background page \(Inactive\)](#)  
Details Remove C

Chrome Apps

## Screenshot 2:

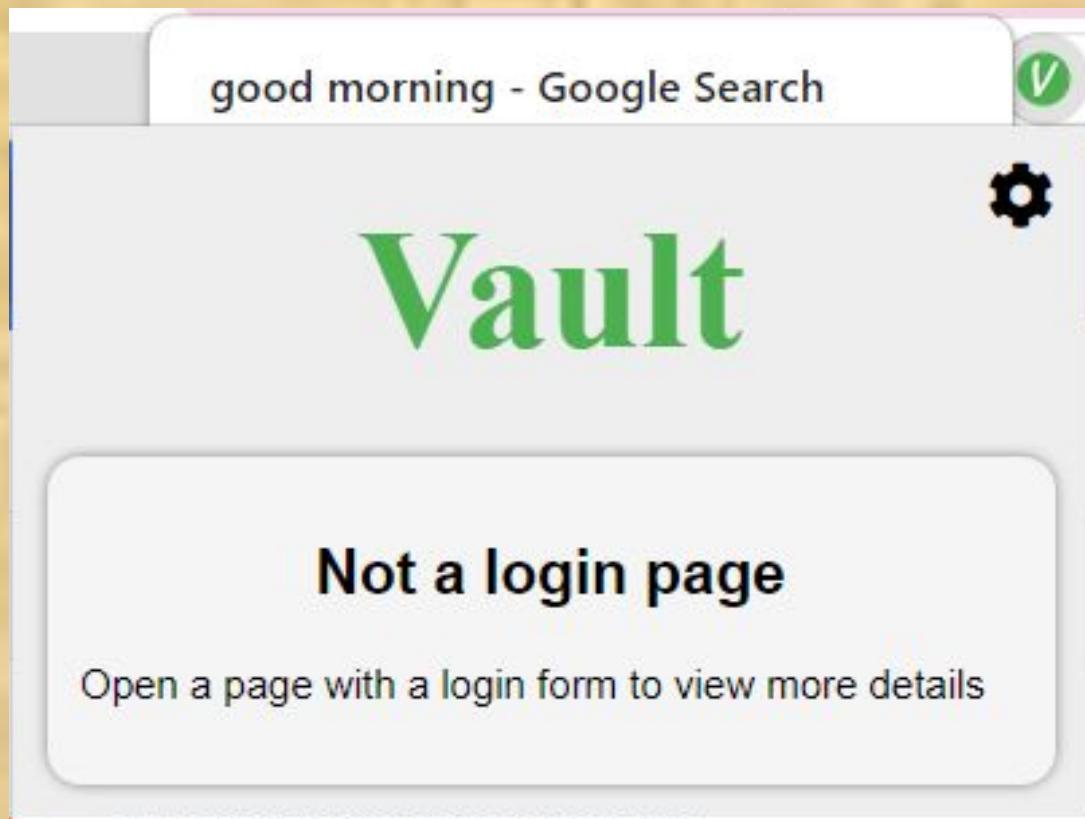


## Screenshot 3:

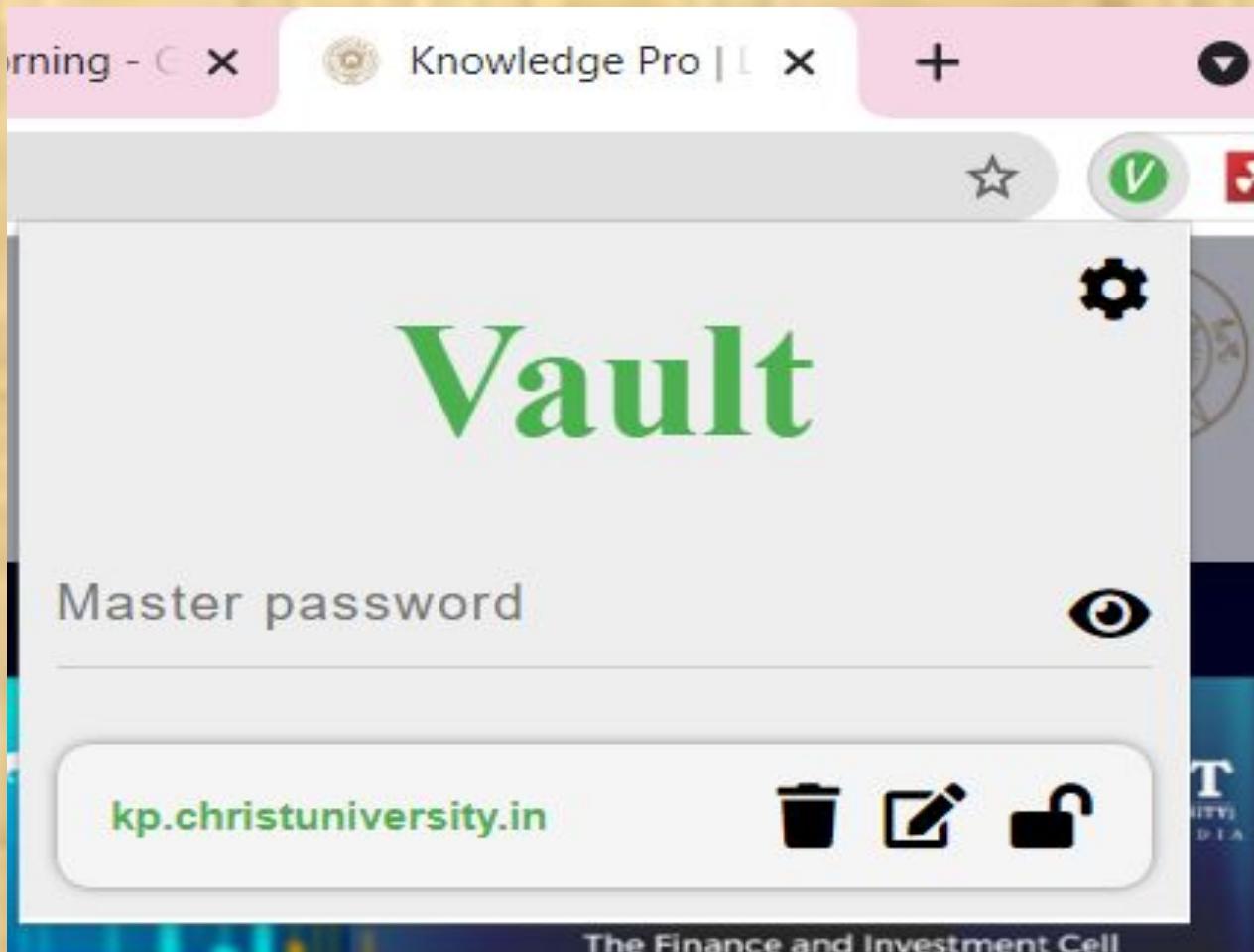
google.com/search?q=good+morning&source=hp&ei=X\_ZNYdn5JMSc4-EPitOyoAg&iflsig=ALs-wAMAAAAAYU4Eb2PRnaBw\_h9IRfYuFg2t1tTSyyY4&oq=good+&gs\_lcp=Cgdnd3Mtd2l6... ☆

142.250.71.4

## Screenshot 4:



## Screenshot 5:



## Screenshot 7:

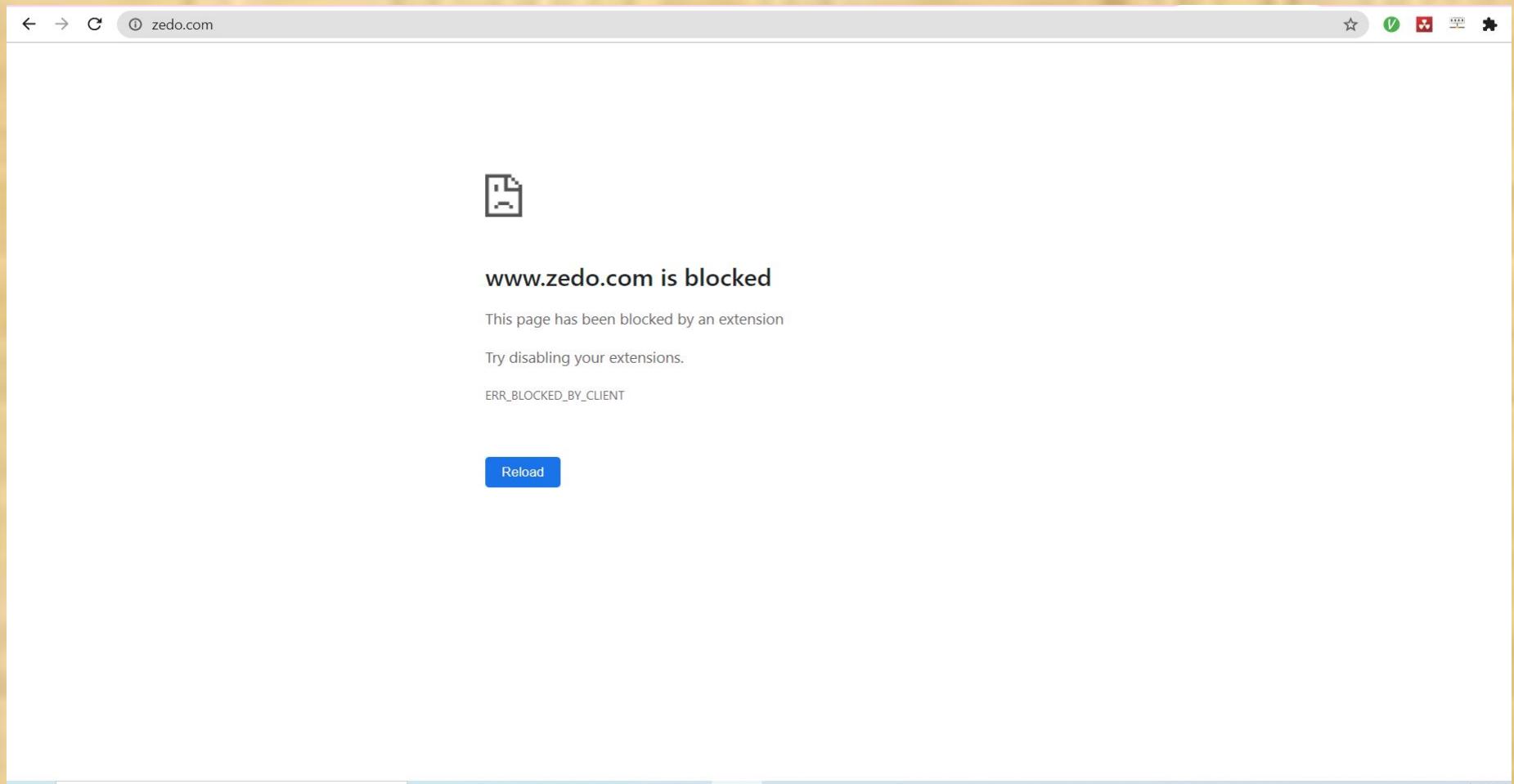
The screenshot shows the homepage of zedo.com. At the top, there is a navigation bar with links for Products, News, About Us, Blog, Contact Us, and a prominent Login button. The main header features the ZEDO logo, which consists of a blue circular icon with three dots and the word "ZEDO" in green. Below the header, a large teal banner with white text reads "Advertising Technology Partner for Publishers" and "World's Largest Independent Global Ad Server, Ad Technology Innovator and Publisher Monetization Partner". The background of the page has a subtle geometric pattern.

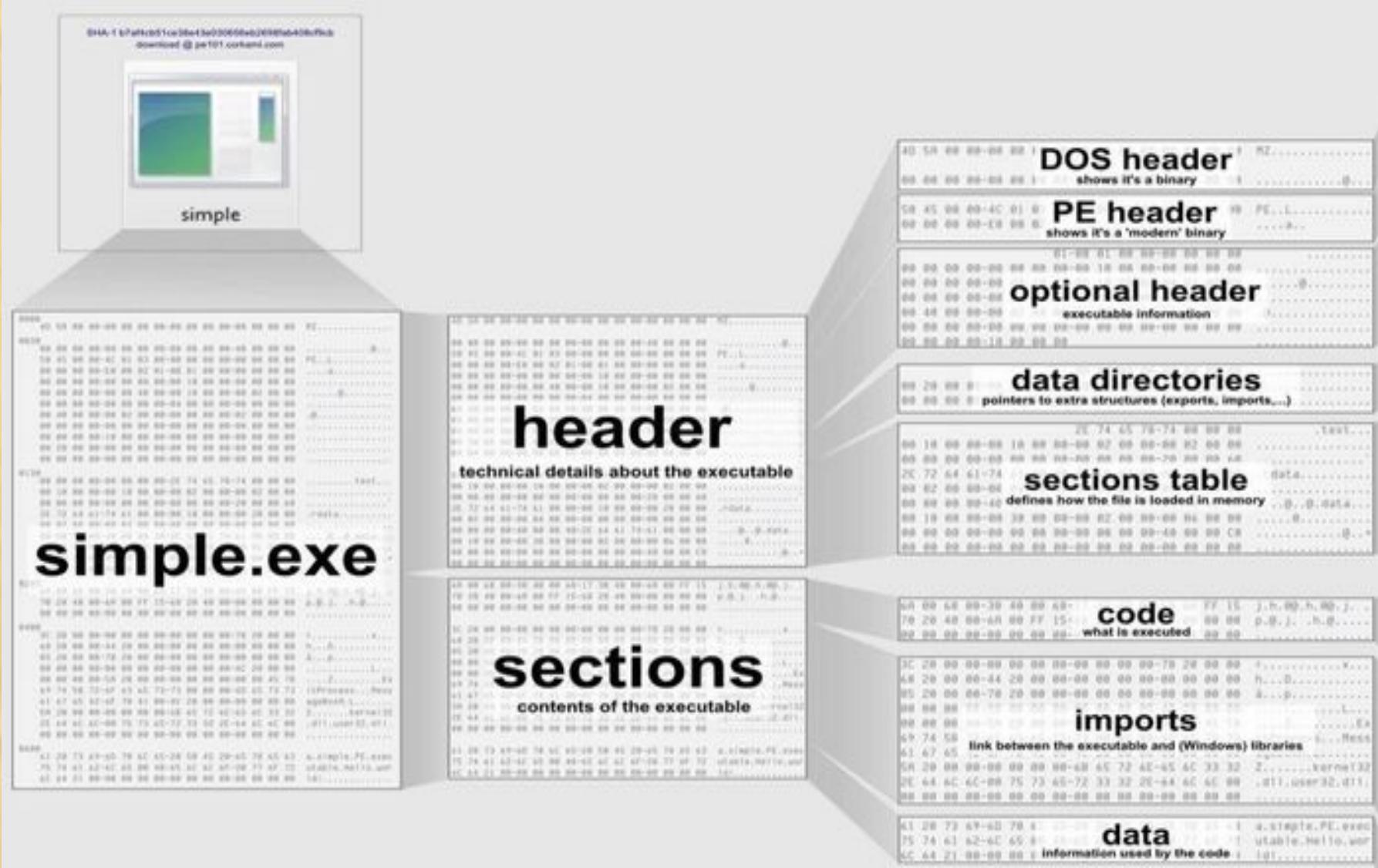
ADVERTISING THAT WORKS !



23.35.2.191

## Screenshot 6:





```
-----DOS_HEADER-----  
[ IMAGE_DOS_HEADER ]  
0x0      0x0    e_magic:          0x5A4D  
0x2      0x2    e_cblp:  
0x4      0x4    e_cp:  
0x6      0x6    e_crlc:  
0x8      0x8    e_cparhdr:  
0xA      0xA    e_minalloc:  
0xC      0xC    e_maxalloc:  
0xE      0xE    e_ss:  
0x10     0x10   e_sp:  
0x12     0x12   e_csum:  
0x14     0x14   e_ip:  
0x16     0x16   e_cs:  
0x18     0x18   e_lfarlc:  
0x1A     0x1A   e_ovno:  
0x1C     0x1C   e_res:  
0x24     0x24   e_oemid:  
0x26     0x26   e_oeminfo:  
0x28     0x28   e_res2:  
0x3C     0x3C   e_lfanew:  
0x78  
-----NT_HEADERS-----  
[ IMAGE_NT_HEADERS ]
```

```
In [15]: └─▶ from sklearn.model_selection import train_test_split
```

```
In [16]: └─▶ # Features  
      x  
      # Labels  
      y = data.label
```

```
In [17]: └─▶ X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.33, random_state=42)
```

```
In [18]: └─▶ from sklearn.naive_bayes import MultinomialNB  
      clf = MultinomialNB()  
      clf.fit(X_train,y_train)  
      clf.score(X_test,y_test)
```

Out[18]: 0.9722674661631376

```
In [19]: └─▶ print("Accuracy of Model",clf.score(X_test,y_test)*100,"%")
```

Accuracy of Model 97.22674661631376 %

```
In [20]: └─▶ sample_url = ["buyfakebillsonlinee.blogspot.com"]  
      vect = cv.transform(sample_url).toarray()
```

Accuracy of Model 97.22674661631376 %  
google.com/ not malicious  
Accuracy of Model 97.22674661631376 %  
radsport-voggel.de/wp-admin/includes/log.exe malicious  
Accuracy of Model 97.22674661631376 %

## Screenshot 7:

Malsmart

File

URL IP File

IP report tab!

IP:

Country:

Owner:

Number of detected URLs:

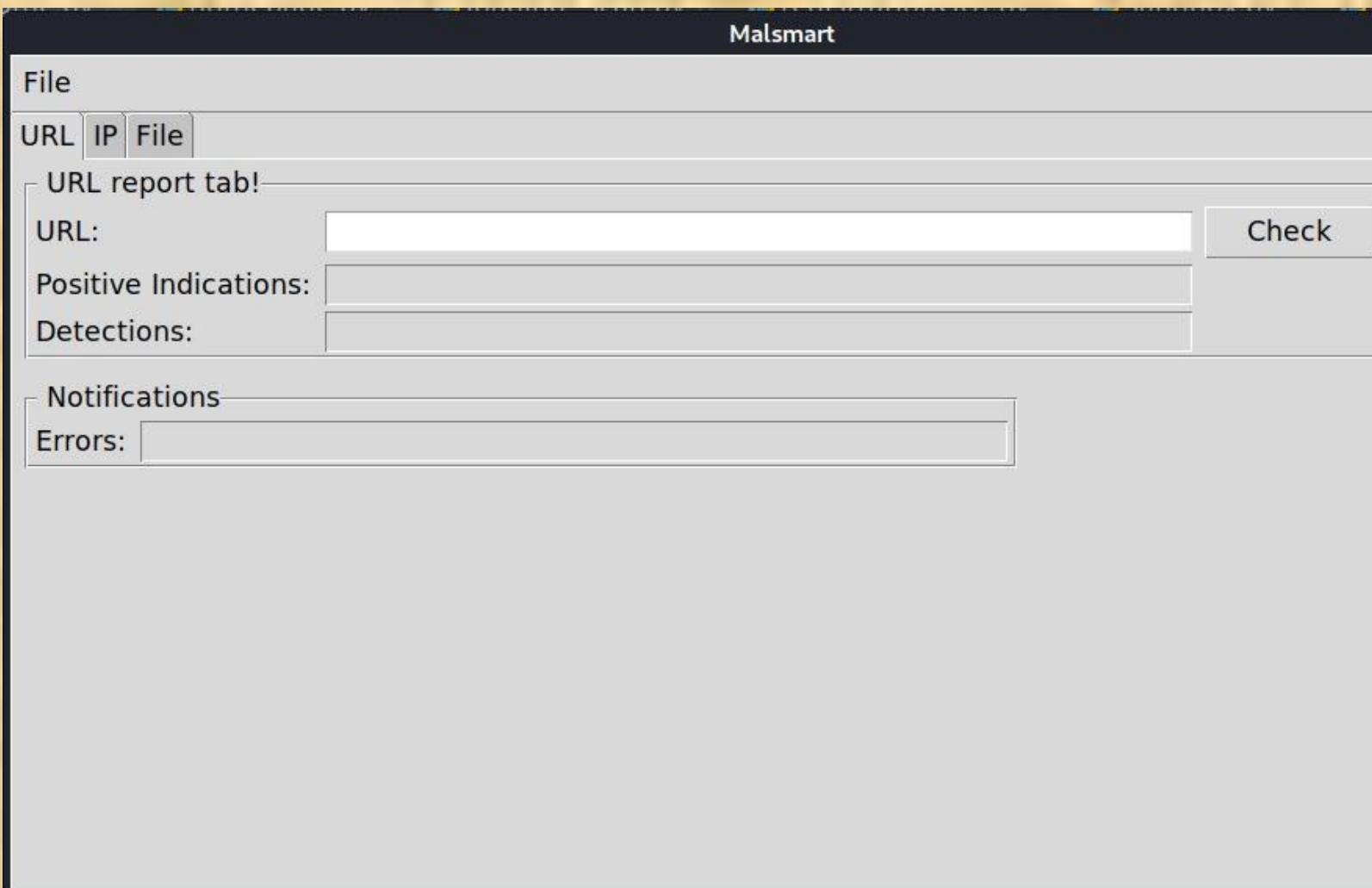
Number of detected malicious files:

Check

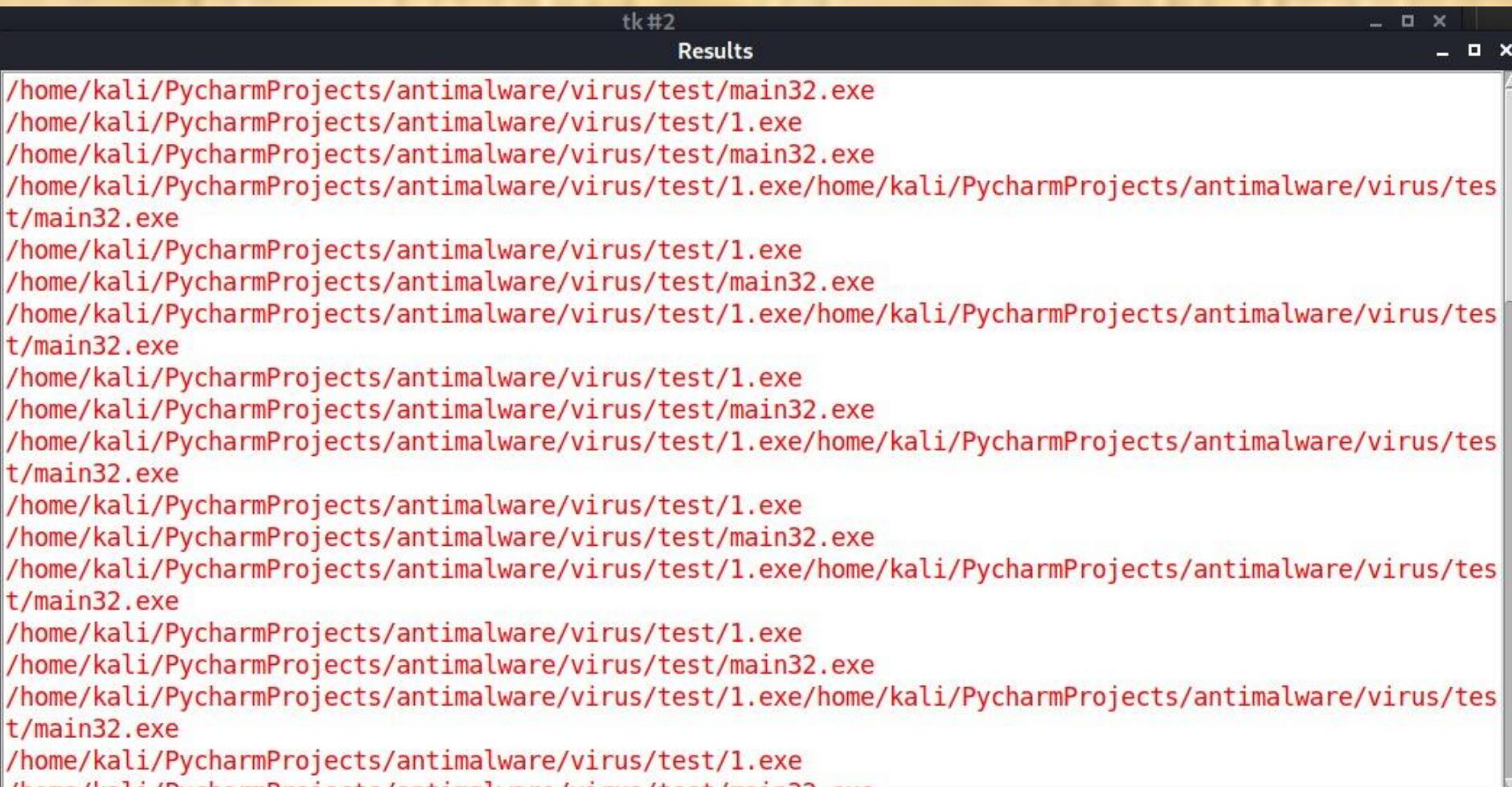
Notifications

Errors:

## Screenshot 8:



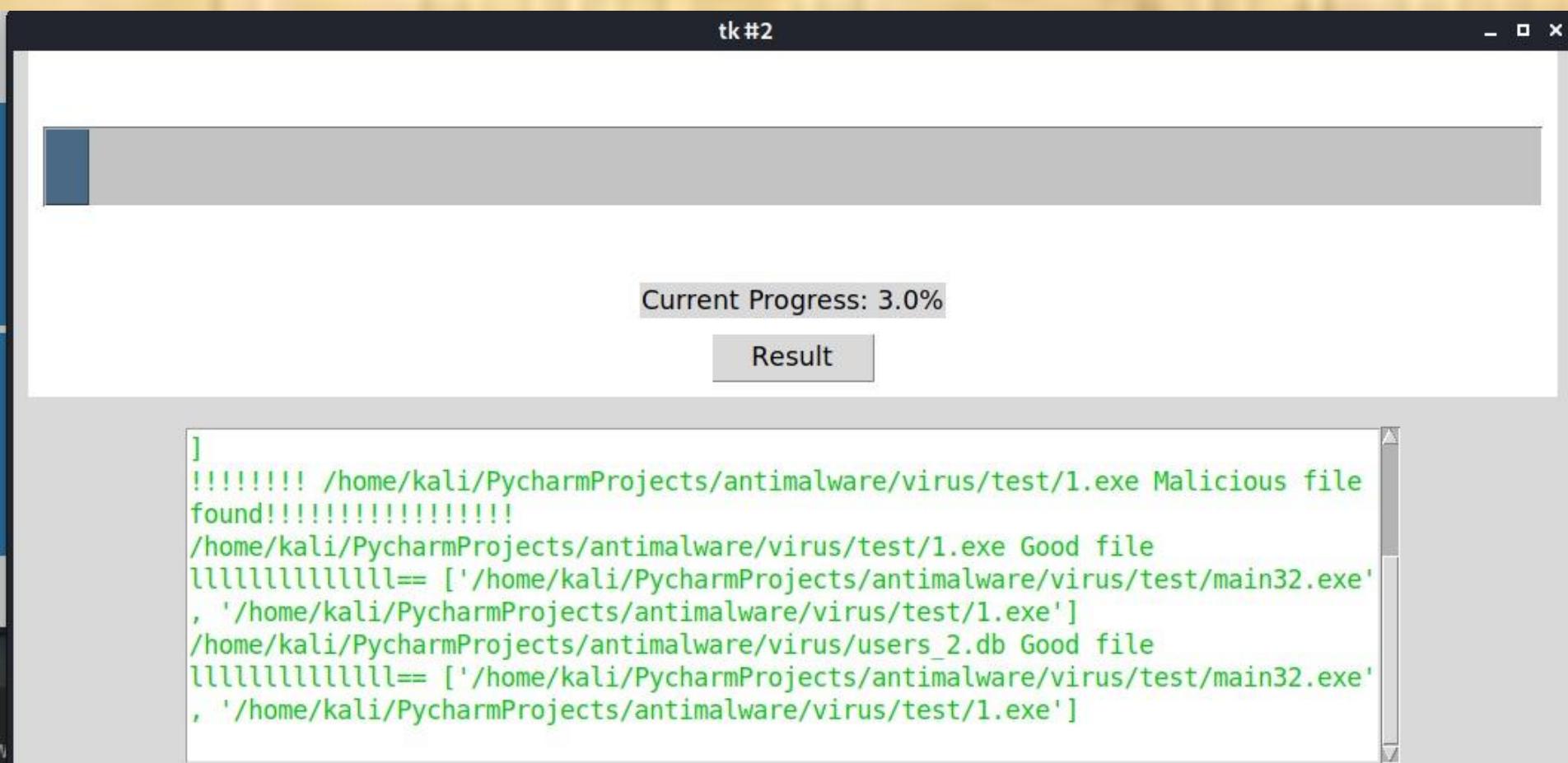
## Screenshot 9:



```
tk #2
Results

/home/kali/PycharmProjects/antimalware/virus/test/main32.exe
/home/kali/PycharmProjects/antimalware/virus/test/1.exe
/home/kali/PycharmProjects/antimalware/virus/test/main32.exe
/home/kali/PycharmProjects/antimalware/virus/test/1.exe/home/kali/PycharmProjects/antimalware/virus/test/main32.exe
/home/kali/PycharmProjects/antimalware/virus/test/1.exe
```

## Screenshot 10:



## Screenshot 11:

FIREWALL

S.no	PORT	ALLOW/DENY	IN/OUT BOUND	FROM
[ 1]	2222	ALLOW	IN	Anywhere
[ 2]	2222	ALLOW	OUT	Anywhere
[ 3]	5555	ALLOW	IN	Anywhere
[ 4]	1111	ALLOW	IN	Anywhere
[ 5]	2222	(v6)	ALLOW	IN
[ 6]	2222	(v6)	ALLOW	OUT
[ 7]	443	(v6)	DENY	OUT

Add entry

PORT :

BOUND:

Allow/Deny:

FROM:

Add Record

DELETE ENTRY

S.NO:

Delete Record

Refresh

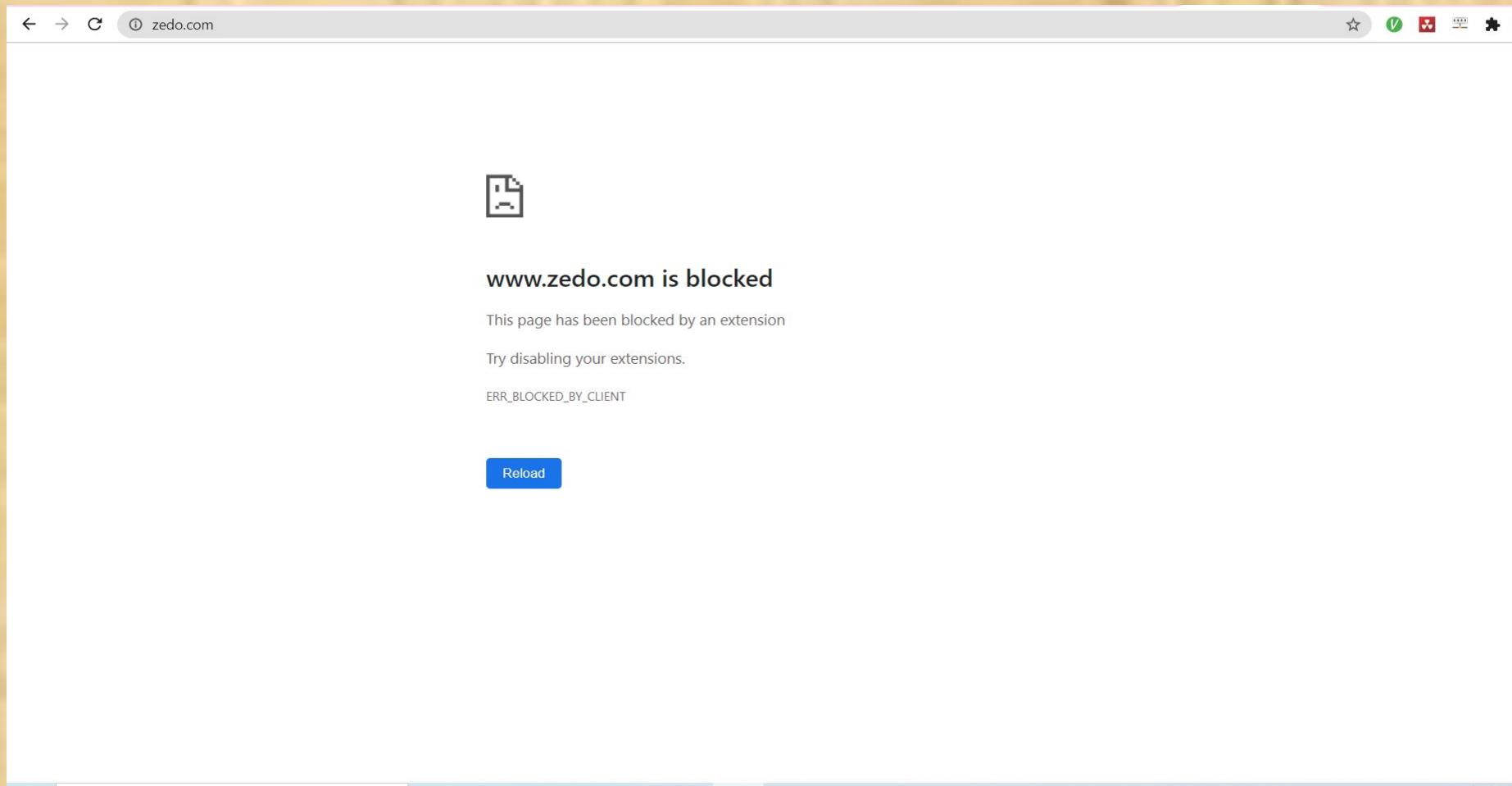
## Screenshot 12:

```
class TkinterCustomButton(tkinter.Frame):
    """ tkinter custom button with border, rounded corners and hover effect

    Arguments: master= where to place button
               bg_color= background color, None is standard,
               fg_color= foreground color, blue is standard,
               hover_color= foreground color, lightblue is standard,
               border_color= foreground color, None is standard,
               border_width= border thickness, 0 is standard,
               command= callback function, None is standard,
               width= width of button, 110 is standard,
               height= width of button, 35 is standard,
               corner_radius= corner radius, 10 is standard,
               text_font= (<Name>, <Size>),
               text_color= text color, white is standard,
               text= text of button,
               hover= hover effect, True is standard,
               image= PIL.PhotoImage, standard is None"""

```

## Screenshot 13:



## Github LInk

<https://github.com/AchintyaVatsraj/MalSmart>

ANY  
QUESTION



