PROJECT TOPIC: Keylogger Defender

Specialization: CSE

Project Group Members:

1. Amrendra Singh (2115000141.)
2. Hitansh Mangla (2115000473)
3. Nikunj Maheshwari (2115000672)

Project Mentor: Dr. Arvind Prasad, Assistant Professor

Objective: The purpose is to develop a cybersecurity tool that detects keylogger infections by simulating keylogging activity using .NET, receiving logs via a Python Flask server, and analysing them through a Python GUI. The system monitors keystrokes and external IP communication, generating alerts upon detecting suspicious activity, thus providing real-time defense and understanding of malware behaviour.

Tools required:

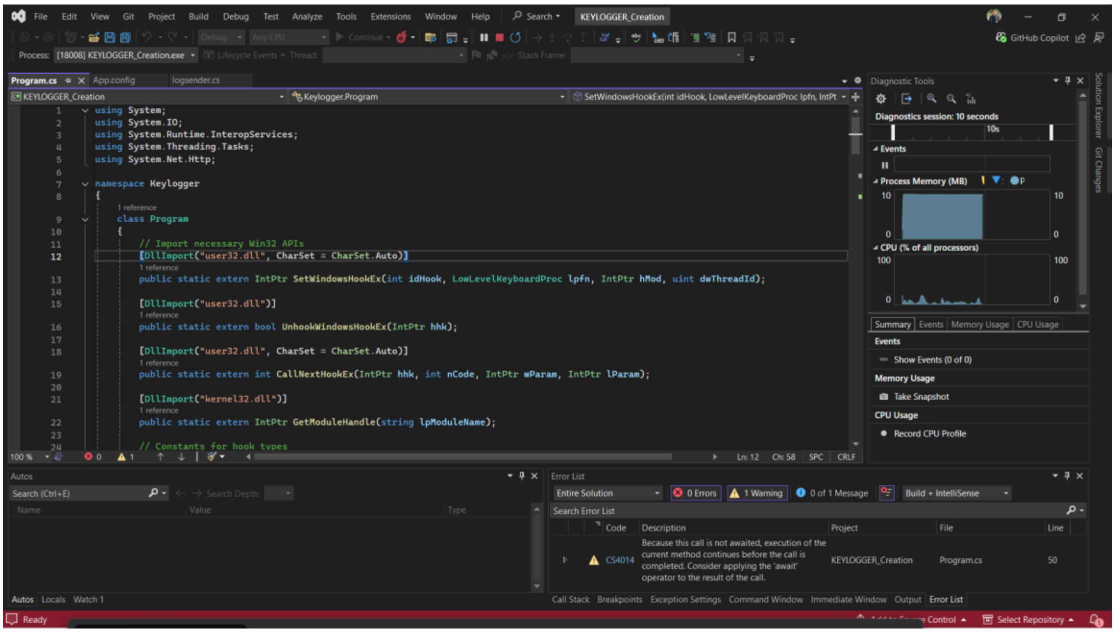
- **Hardware Requirements:**
- Windows Based System
- Internet Connection for communication

- **Software Requirements:**
- Visual Studio (.NET Framework - C#)
- Flask (Python Framework)
- Tkinter (GUI)
-

Abstract: The “Keylogger Defender Tool” is a cybersecurity-focused project aimed at simulating keylogger activity and developing an intelligent detection mechanism. The system consists of three core modules – a keylogger developed in C# which captures keystrokes and transmits them to a server; a Flask-based server that receives and stores logs; and a Python GUI that displays incoming logs and performs detection of malicious behavior based on external communication. This project demonstrates a complete attack-defense cycle, highlighting how malware behaves and how it can be traced and blocked using real-time monitoring and alerts.

Outcome:

- **Successfully built a functioning keylogger in .NET**



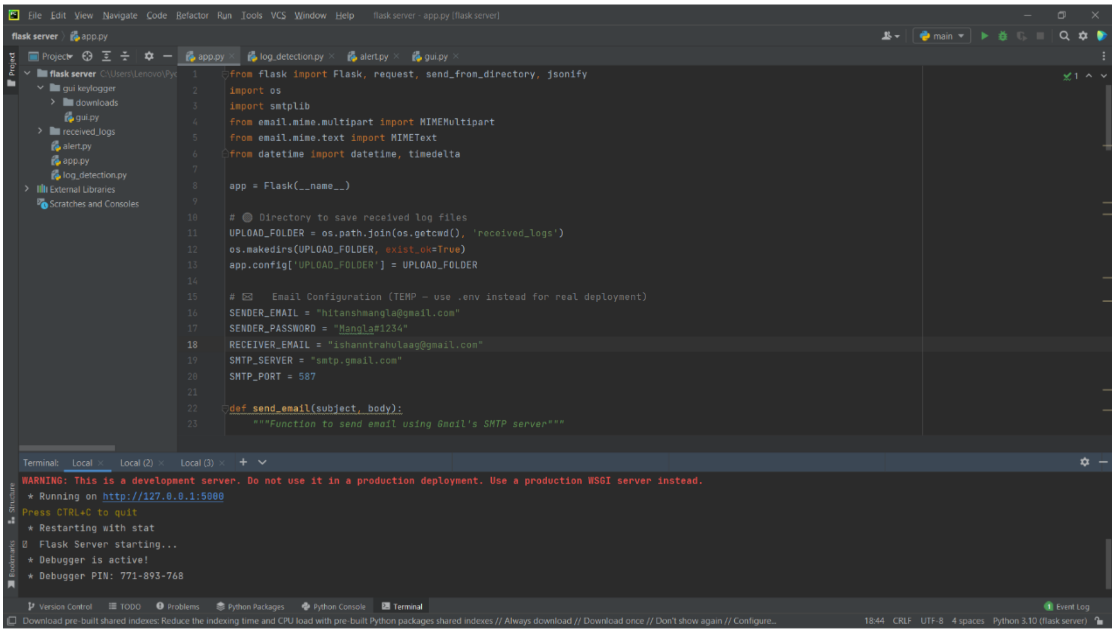
```
using System;
using System.IO;
using System.Runtime.InteropServices;
using System.Threading.Tasks;
using System.Net.Http;

namespace Keylogger
{
    class Program
    {
        // Import necessary Win32 APIs
        [DllImport("user32.dll", CharSet = CharSet.Auto)]
        public static extern IntPtr SetWindowsHookEx(int idHook, LowLevelKeyboardProc lpfn, IntPtr hMod, uint dwThreadId);

        [DllImport("user32.dll")]
        public static extern bool UnhookWindowsHookEx(IntPtr hhk);

        [DllImport("user32.dll", CharSet = CharSet.Auto)]
        public static extern int CallNextHookEx(IntPtr hhk, int nCode, IntPtr wParam, IntPtr lParam);

        [DllImport("kernel32.dll")]
        public static extern IntPtr GetModuleHandle(string lpModuleName);

        // Constants for hook types
```

Error List: Entire Solution | 0 Errors | 1 Warning | 0 of 1 Message | Build + IntelliSense

Code	Description	Project	File	Line
CS4014	Because this call is not awaited, execution of the current method continues before the call is completed. Consider applying the 'await' operator to the result of the call.	KEYLOGGER_Creation	Program.cs	50

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
KEYLOGGER_Creation
Process: [18008] KEYLOGGER_Creation.exe
Program.cs
App.config
logsender.cs
Keylogger.Program
SetWindowsHookEx(int idHook, LowLevelKeyboardProc lpfn, IntPt
Diagnostic Tools
Diagnostics session: 10 seconds
Events
Process Memory (MB)
CPU (% of all processors)
Summary Events Memory Usage CPU Usage
Show Events (0 of 0)
Take Snapshot
Record CPU Profile
GitHub Copilot
Ln: 12 Ch: 58 SPC CRLF
Autos Locals Watch 1
Call Stack Breakpoints Exception Settings Command Window Immediate Window Output Error List
Ready
Select Repository

- **Flask Server Console receiving files**



```
from flask import Flask, request, send_from_directory, jsonify
import os
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from datetime import datetime, timedelta

app = Flask(__name__)

# ⚫ Directory to save received log files
UPLOAD_FOLDER = os.path.join(os.getcwd(), 'received_logs')
os.makedirs(UPLOAD_FOLDER, exist_ok=True)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

# 🖂  Email Configuration (TEMP — use .env instead for real deployment)
SENDER_EMAIL = "hitanshmangla@gmail.com"
SENDER_PASSWORD = "Mangla#1234"
RECEIVER_EMAIL = "ishanntrahulaag@gmail.com"
SMTP_SERVER = "smtp.gmail.com"
SMTP_PORT = 587

def send_email(subject, body):
    """Function to send email using Gmail's SMTP server"""
```

```
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
Ø  Flask Server starting...
 * Debugger is active!
 * Debugger PIN: 771-893-768
```

File Edit View Navigate Code Refactor Run Tools VCS Window Help
flask server - app.py [flask server]
flask server
app.py
main
Project
flask server C:\Users\Lenovo\Pyc
gui keylogger
downloads
gui.py
received_logs
alert.py
app.py
log_detection.py
External Libraries
Scratches and Consoles
log_detection.py
alert.py
gui.py
Terminal: Local Local (2) Local (3)
Version Control TODO Problems Python Packages Python Console Terminal
Event Log
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download once // Don't show again // Configure...
18:44 CRLF UTF-8 4 spaces Python 3.10 (flask server)