**“Identification of Text Data Based on Key Strokes”**

***A***

***Project Report***

*submitted in partial fulfillment of the*

*requirements for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE & ENGINEERING**

**by**

|  |  |
| --- | --- |
| **Name** | **Roll No.** |
| **Rahul Roy** | **R164216048** |
| **Tarun** | **R164216066** |
| **Tarun Varshney** | **R164216067** |
| **Vaastav Jain** | **R164216069** |

***under the guidance of***

**Mr. Amitava Choudhury**

**Assistant Professor**

**School of Computer Science, UPES, Dehradun**

**C:\Users\aktomar\Desktop\logo.png**

**School of Computer Science**

**Department of Systemics**

**University of Petroleum & Energy Studies**

**Bidholi, Via Prem Nagar, Dehradun, UK**

**October – 2018**

**C:\Users\aktomar\Desktop\logo.png**

**CANDIDATE’S DECLARATION**

We hereby certify that the project work entitled **“ Identification of Text Data Based On Key Strokes ”** in partial fulfillment of the requirements for the award of the Degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING with specialization in INTERNET OF THINGS AND SMART CITIES and submitted to the School of Computer Science, Department of Systemics, University of Petroleum & Energy Studies, Dehradun, is an authentic record of my/ our work carried out during a period from **August**, **2018**  to **October**, **2018** under the supervision of **Mr.Amitava Choudhary, Assistant Professor, School of Computer Science**.

The matter presented in this project has not been submitted by me/ us for the award of any other degree of this or any other University.

**Tarun Rahul Roy**

**(Enroll No. R164216066) (Enroll No. R164216048)**

**Tarun Varshney Vaastav Jain**

**(Enroll No. R164216067) (Enroll No. R164216069)**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_2018 Mr **Amitava Choudhury**

**Assistant Professor**

**School of Computer Science, UPES, Dehradun**

**Dr. Neelu Jyoti Ahuja**

HOD - Systemics

School of Computer Science

University of Petroleum & Energy Studies

Dehradun – 248 001 (Uttarakhand)

**ACKNOWLEDGEMENT**

We wish to express our deep gratitude to our guide **Mr. Amitava Choudhary**, for all advice, encouragement and constant support he has given us through out our project work. This work would not have been possible without his support and valuable suggestions.

We sincerely thank to our respected HoD of the Department, **Dr. Neelu Jyoti Ahuja**, for her great support in doing our project in **Cyber Security** at **SoCS**.

We are also grateful to **Dr Maneesh Pratik** SoCS, UPES for giving us the necessary facilities to carry out our project work successfully.

We would like to thank all our **friends** for their help and constructive criticism during our project work. Finally we have no words to express our sincere gratitude to our **parents** who have shown us this world and for every support they have given us.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Rahul Roy** | **Tarun** | **Tarun Varshney** | **Vaastav Jain** |
| **Roll No.** | **R164216048** | **R164216066** | **R16216067** | **R164216069** |

**ABSTRACT**

Access to information is paramount for any business or organization, and millions have been invested in information technology systems and infrastructures. The pervasiveness of the Internet has led to an increase in employees who use workplace system for purposes not related to work. As technology advances, making it easy and inexpensive to install and operate surveillance systems, some companies are implementing workplace-monitoring programs. Key logger is one such software techniques used to generally track online activities by employees, measure productivity and collect proof of hours worked. Employers can be alerted when an employee or when the worker is not meeting productivity goals performs certain actions. All activities logged can be then presented in easy-to-read graphical reports. The keylogger need to stores all the keystrokes offline i.e. on the victim's device without the user’s awareness. A function inside the key logger, identifies and triggers when a key is pressed. Also the key logger must be running in hidden mode as it cannot be seen by the user. Lastly, all the text data should be stored inside the log or backup for future access if required. From this project, we hope to build this software program, which can be useful for various organizations.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Contents** | **Page No.** |
| **1.** | **Introduction** | **1** |
|  | * 1. History   2. Requirement Analysis   3. Main Objective   4. Sub Objectives   5. Pert Chart Legend | 1  1  1  2  2 |
| **2.** | **System Analysis** | **3** |
|  | * 1. Existing System   2. Motivation   3. Proposed System   4. Modules   2.4.1. kbhit()  2.4.2. GetAsyncKeyState()  2.4.3. GetKeyState | 3  3  4  4  5  5  5 |
| **3.** | **Design** | **6** |
| **4.** | **Implementation** | **8** |
|  | 4.1. Functionality of keyboard  4.2. Algorithm | 8  8 |
| **5.** | **Output Screens** | **9** |
| **6.** | **Limitations and Future Enhancements** | **10** |
| **7.** | **Conclusion** | **11** |
|  | **References** | **12** |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Figures** | **Page No.** |
| **1** | **Chapter 1** |  |
|  | Fig.1.1 Pert Chart | 2 |
| **2.** | **Chapter 3** |  |
|  | Fig. 3.1 Bitwise & Operation | 6 |
| **3.** | **Chapter 4** |  |
|  | Fig. 4.1 Block Diagram | 8 |
| **4.** | **Chapter 5** |  |
|  | Fig.5.1 Compile Output | 9 |
|  | Fig. 5.2 Result Output | 9 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Table** | **Page No.** |
| 1. | **Chapter 3** |  |
|  | Table 3.1 ASCII code Table | **7** |

1. **INTRODUCTION**
2. **Keystroke Logger**

A keystroke logger (also known as keylogger) is software that tracks or logs the keys when struck on the keyboard. It does this task in a hidden manner, so that user don’t know that his/her actions are being monitored.. 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.

* 1. **History**

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.

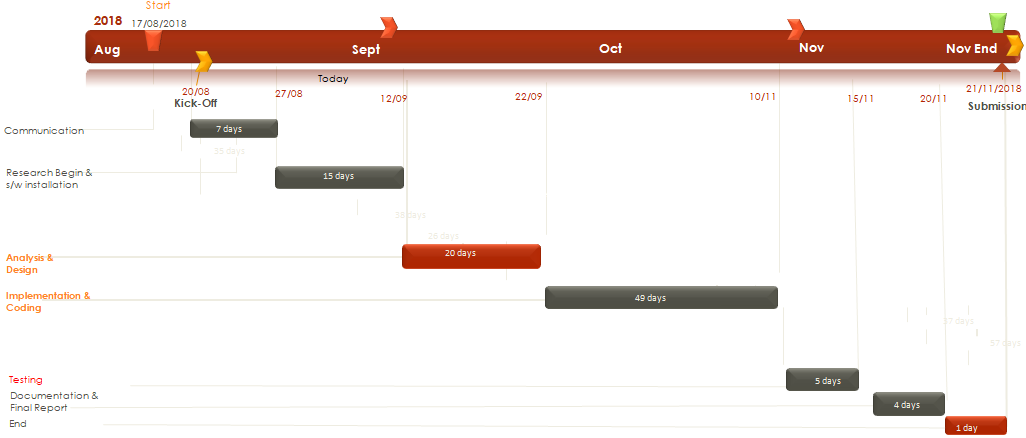
* 1. **Requirement Analysis**

The proposed work run with minimum version of system specification, but we will test on

* 1GB RAM, 128 GB HDD
* Windows/Linux Operating System
* Keyboard with English US layout
* i3 Processor
  1. **Main Objective**

To implement an algorithm to identify the text based on data on key strokes.

* 1. **Sub Objective**
* Program should be hidden (not easy to detect).
* Reusability of log data if required.
* Data saved in log may be encrypted.
  1. **Pert Chart**

****

**Fig1.1 Gantt Chart**

1. **SYSTEM ANALYSIS**

**2.1. Existing System**

You will find many commercial, non-commercial, & open-source keyloggers out there. There are also many traps & fakes on the internet. Some free keyloggers will even steal your data. I would suggest to be on the safe side, & either use an open-source (free) or a commercial (not free) keylogger.

Some very simple open-source keyloggers (for Windows) are-

* MinhasKamal/StupidKeylogger
* Siegfried5/Keylogger
* TheFox/keylogger
* anujagarwal464/Keylogger

Well known commercial keyloggers-

* Computer Monitoring Software
* Spytech Spy Software

**2.2. Motivation**

Security [1] [2] [3] is essential in an operating system, and there are many software vendors that provide antivirus software and other protection software so that the computer is free from threats or unwanted actions such as phishing, malware, keylogger and others. Keylogger [4] is a tool that most progressed because this application is very rarely recognized as a malicious program by antivirus [5], the function of keylogger will record all activities related to the emphasis on the keyboard keys or activity on the operating system [4][5].

Why we chose keylogger over other surveillance methods - a keylogger is inexpensive, tricky to detect, works fast and more importantly customizable. That means, we know that a keylogger captures every key one has typed. Now we can customize, how we want the data to reach us - via mail, or uploaded to a server, or any other way. Hence, the way the data reaches to us is decided by itself. There would be no security in the way of data and receiver except the device's firewall. Once the keystrokes are recorded and bypassed through device firewall, there is no other barrier in the way.

**2.3. Proposed System**

The keylogger need to stores all the keystrokes offline i.e. on the victim's device without the user’s awareness. A function inside the key logger, identifies and triggers when a key is pressed. Also the key logger must be running in hidden mode as it cannot be seen by the user. Lastly, all the text data should be stored inside the log or backup for future access if required. Keylogger will comprise of three main components:

**Log**: A log is the automatically produced and time-stamped documentation of events relevant to a particular system.

**GetKeyState()**: It returns a specific value when the ASCII/hex value of the character keypressed.

**Hidden Mode**: The whole software isn’t visible or detectable by the user.

**2.4. Modules**

There are various ways by keys can be intercepted and three of them are:

1. kbhit()

2. GetAsyncKeyState()

3. GetKeyState()

**2.4.1. kbhit()**

Prototype: int kbhit(void);  
Header File: conio.h  
Explanation: This function is not defined as part of the ANSI C/C++ standard. It is generally used by Borland's family of compilers. It returns a non-zero integer if a key is in the keyboard buffer. It will not wait for a key to be pressed.

**2.4.2. GetAsyncKeyState()**

Prototype: short GetAsyncKeyState(int vKey);

Header File: Windows.h  
Explanation: Determines whether a key is up or down at the time the function is called, and whether the key was pressed after a previous call to GetAsyncKeyState. If the function succeeds, the return value specifies whether the key was pressed since the last call to GetAsyncKeyState, and whether the key is currently up or down. If the most significant bit is set, the key is down, and if the least significant bit is set, the key was pressed after the previous call to GetAsyncKeyState.

**2.4.3. GetKeyState()**

Prototype: short GetKeyState(int vKey);

Header File: Windows.h

Explanation: Retrieves the status of the specified virtual key. The status specifies

whether the key is up, down, or toggled. The return value specifies the status of

the specified virtual key, as follows:

* If the high-order bit is 1, the key is down; otherwise, it is up.
* If the low-order bit is 1, the key is toggled. A key, such as the CAPS LOCK key, is toggled if it is turned on. The key is off and untoggled if the low-order bit is 0. A toggle key's indicator light (if any) on the keyboard will be on when the key is toggled, and off when the key is untoggled.

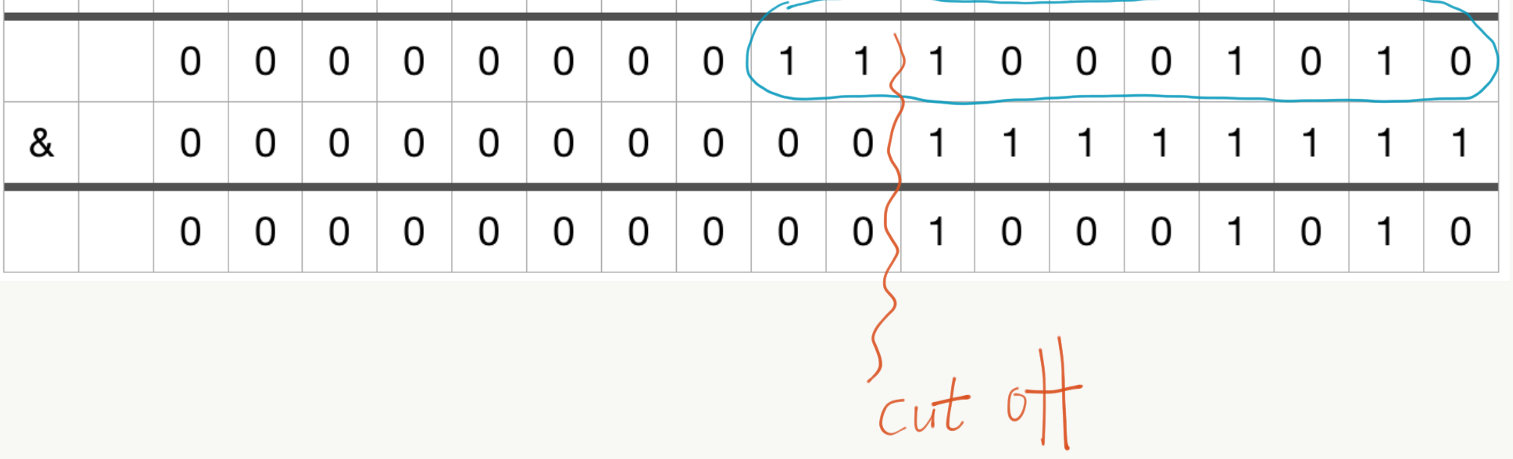
1. **DESIGN**

First we will be using while loop for true condition so that it neve\r stops detecting key strokes. Then we need to make sure that it doesn’t consume 100% CPU usage, to prevent that sleep function is required with 10/20ms as the parameter. Then we need to define key state for **Capslock**, **Numlock**, **left-right Shift** button by using GetKeyState() and Bitwise AND operator. It is necessary to **Bitwise AND(&)** between **GetKeyState()** and **0xff**.

0xff is the hexadecimal number **FF** which has a integer value of **255**. And the binary representation of FF is **00000000000000000000000011111111** (under the 32-bit integer).

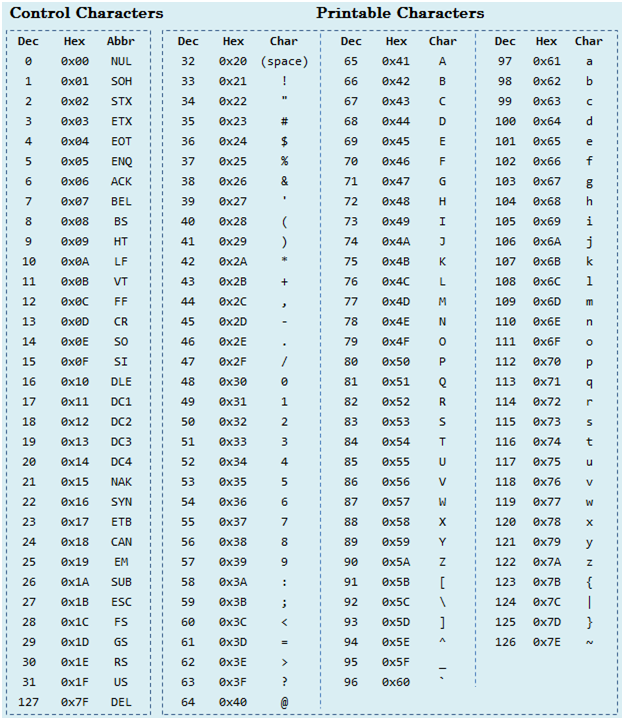
The & operator performs a bitwise AND operation. a & b will give you an integer with a bit pattern that has a 0 in all positions where b has a 0, while in all positions where b has a 1.

So “& 0xff” effectively masks the variable so it leaves only the value in the last 8 bits, and ignores all the rest of the bits.

****

**Fig 3.1 Bitwise & Operation**

Next step is to define “for loop” for each character set with their hex values limits like specific loop for num, printable char, special char etc.

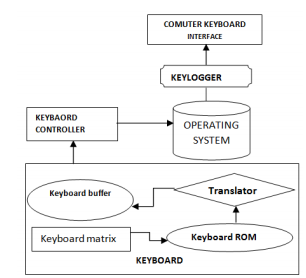
****

**Table 3.1 ASCII Code (Source:** [www.ascii-code.org](https://www.ascii-code.org/)**)**

Then all intercepted values should be stored in log file (with fixed directory). The log file should be open only in **append mode** only.

1. **IMPLEMENTAION**

**4.1. Functionality of keyboard**

****

**Fig 4.1 Block Diagram**

* 1. **Algorithm**

1. Create an empty log file for storing keylogs.

2. Intercept keys pressed by user using GetAsynckeyState()/GetKeyState function.

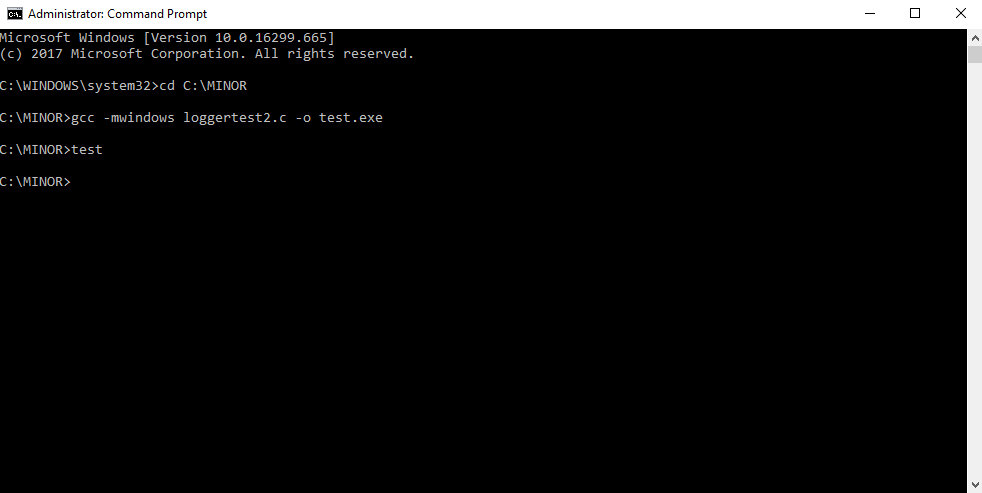
3. Store these intercepted values in file.

4. Hide the running window dialog to make it undetectable.

5. Use while loop to make it running in all conditions.

6. Add sleep() function to reduce the CPU usage.

1. **OUTPUT SCREENS**

****

**Fig 5.1 Compile Output**

****

**Fig 5.2 Result Output**

1. **LIMITATION & FUTURE ENHANCEMENT**

While keyloggers have many acceptable uses they also have many malicious uses [6][7].

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 numbers

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.

1. **CONCLUSION**

Software that can not only monitor every keystroke and action performed at a PC but also be used as legally binding evidence of wrong-doing has been unveiled. Worries about cyber-crime and sabotage have prompted many employers to consider monitoring employees. They have joined forces to create a system which can monitor computer activity, store it and retrieve disputed files within minutes… "People need to recognize that

you are using a PC as a representative of a company and that employers have a legal requirement to store data.

Website monitoring service can check HTTP pages, HTTPS, SNMP, FTP, SMTP, POP3, IMAP, DNS, SSH, TELNET, SSL, TCP, ping and a range of other ports with great variety of check intervals from every 4 hours to every one minute.

Typically, most network monitoring services test your server anywhere between once-per hour to once-perminute.

Features:

• Protect intellectual property and business secrets

• Prevent and stop sabotage and data theft

• Prevent Internet/email abuse

• Reduce workplace slackers

• Improve efficiency and productivity

**References**

[1] R. Rahim and A. Ikhwan, "Study of Three-Pass Protocol on Data Security," International Journalof Science and Research (IJSR), vol. 5, no. 11, pp. 102-104, 2016.

[2] R. Rahim, "128 Bit Hash of Variable Length in Short Message Service Security," InternationalJournal of Security and Its Applications, vol. 11, no. 1, pp. 45-58, 2017.

[3] A. P. U. Siahaan and R. Rahim, "Dynamic Key Matrix of Hill Cipher Using Genetic Algorithm,"International Journal of Security and its Applications, vol. 10, no. 8, pp. 173-180, 2016.

[4] R. Venkatesh and R. K. Sekhar, "User Activity Monitoring Using Keylogger," Asia Journal ofInformation Technology, vol. 15, no. 23, pp. 4758-4762, 2015.

[5] D. Y. Pawade, A. Lahigude and D. Reja, "Review Report On Security Breaches Using KeyloggerAnd Clickjacking," International Journal of Advance Foundation and Research in Computer(IJAFRC), vol. 2, no. Special Issue (NCRTIT 2015), pp. 55-59, 2015.

[6] N. Pathak, A. Pawar, and B. Patil, "A Survey on Keylogger: A malicious Attack," InternationalJournal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 4,no. 4, pp. 1465-1469, 2015.

[7] P. Tuli and P. Sahu, "System Monitoring and Security Using Keylogger," International Journalof Computer Science and Mobile Computing, vol. 2, no. 3, pp. 106-111, 2013.