KEYLOGG - A TOUCH BASED KEY LOGGING APPLICATION

Prathamesh Bhosale[1], Saurabh Hanchate[2], Ajay Dasarwar[3], Mohak Indurkar[4]

[1]*Student, Computer Engineering, Rajiv Gandhi Institute of Technology, Maharashtra, India*
[2]*Student, Computer Engineering, Rajiv Gandhi Institute of Technology, Maharashtra, India*
[3]*Student, Computer Engineering, Rajiv Gandhi Institute of Technology, Maharashtra, India*
[4]*Student, Computer Engineering, Rajiv Gandhi Institute of Technology, Maharashtra, India*

Abstract

As Open Source Technologies are gaining popularity in the IT industry, consumers and developers look forward to it as a cheap and durable resource. However the concerns on privacy on Open Source are highly overlooked. Here, we intend to highlight the potential security risk in the open source mobile operating system, Android by implementing 'Touchlogger'. This application is capable of recording the touch activities happening on the smart phones keyboard. As the application will be a covert application, we wish to achieve our goal of logging by a third party keyboard. Our application will create a log file of activities done by the user in a certain time span. The activities such as call logs, texts, text messages and browsing history can be logged. The log file will keep the record of all activities and can be accessed by authorized user only. The log activity will be sent to the desired receiver. In user's interest to protect the privacy, one should always install application from Google Play Store only. Upon installation, the user should check the requested permission and be alerted if there is some slowness while typing.

Keywords: *Key Logg, Key Logging, Touchlogger*

---***---

1. INTRODUCTION

Mobile devices are becoming more popular than televisions globally. They are running by different mobile operating systems (OS) such as Google's Android, Apple's iOS, Nokia's Symbian, Blackberry Ltd, Blackberry OS, Bada of Samsung,, Windows Phone of Microsoft, etc. Mobile operating systems can also be extended by installing different kind of mobile applications (apps). It has become more accessible to developers for developing mobile applications and there are plenty of resources and support available for them. As a result, thousands of applications are now available in the market, some of them are free, and others are not. Android's noticeable popularity among consumers and developers alike is tightly related to its openness and powerful development framework. The Android's platform openness has triggered a great rise in privacy concerns and malware. Android Play Store has attracted hackers to spread their malicious apps (malware). The most common Android malwares are spyware and (SMS) Trojans that: collect private information, send SMSs to premium numbers, record voice calls, etc.

There are many third party custom keyboard applications available for android in Google play store. Therefore users can download these third party keyboards from Google play store and replace their default keyboard with these third party keyboards. However, installing these third party keyboard applications might be a serious security problem as the developer of this keyboard may inject his malicious code which may cause many security problems

1.1 Aim And Objectives

In this application we show the potential risks related to downloading and installing third party keyboards and demonstrate a touchlogger application. This application is a system to capture keystrokes given from keyboard and store those captured keystrokes in a log file on the mobile and send that log file to the desired destination when the user's mobile phone will be connected to the internet. The aim of this application is to highlight vulnerabilities in android environment. Our application's objective can be defined depending on the user's motive to use the application.

2. LITERATURE SURVEY

A. The Evolution of Android Keyboards

The Android operating system is actually designed for touch screen devices like smartphone mobiles and tablets. The initial release was missing many features that we consider nowadays necessities, for instance, the on-screen keyboards. The evolution of Android keyboards has gone through many stages: Android 1.5 was supported by both virtual and physical keyboards it is known by its codename, Cupcake. Landscape and portrait orientation modes supported by virtual keyboards it works with the built in and third-party applications. Along with this it provides auto-correct capability, a suggestion algorithm and dictionary of suggestions, and support for custom user dictionaries. Moreover, it supports tactile feedback using screen vibration. Finally, it creates an impactful attention to third-party developers to develop their own customized keyboards. Android 2.0/2.1 known by its codename, Éclair, introduced some improvements over the soft keyboard. For example, Éclair improved performance of the keyboard by adding multitouch so as to detect secondary presses while typing rapidly, because of which typing accuracy was improved. Android 2.3 known by its codename, Gingerbread, improved the keyboard design and functionality. In this design and coloration of the keys can be changed significantly. On the other hand multitouch has improved with “Chording” and allowing users to press multi-key combinations so as to quickly access the secondary symbol keyboard.

The next improved Android 4.0 which is known by Ice Cream Sandwich. In this an attractive implementation of inline spell-check and replacement done along with the correction intelligence. Android 5.0 known by its codename, Lollipop changes include handler 3.2 quick settings, available in one long, fluid swipe down from the top of the screen alongside notifications, a redesigned phone dialer (which also loses its border between keys) and the ability to set up multiple user profiles for a single device.

B. Android Permissions

Android permission system mandates applications to possess permissions in order to make API system calls. The APIs provide access to system and user resources such that contacts, messages and camera. The permissions are granted by the user upon installation. Application developers declare the required permissions in the AndroidManifest.xml file using the uses-permission tag. For example, an application needs to request the READ_CONTACTS permission to read the user’s address book. Once installed, an application’s permission can’t be changed. There are normal permissions and dangerous permissions. The former type has lower-risk and higher-risk. Lower risk gives access to isolated application-level features, minimal risk to other applications, the system as well as the user. System automatically grants this type of permission for installation without user consent. The higher-risk is dangerous permission it gives applications access to private user data or control over the device which negatively impact the user. Because of that, the system displays this injection. The threat level is not only connected to the meaning of a single permission, instead the permissions combinations play an important role in understanding the potential implications. Android keyboard applications can also request to have permissions upon installation, some of these permissions may impose privacy threat.

C. Key-Logging Threat

Keylogger or keystroke recorders are use to monitor user’s keyboard actions. Keylogger mainly classified into hardware and software categories. Hardware touchloggers are electronic devices used to capture the data between a keyboard and I/O port. They have their own built-in memory, the place where the captured data is stored. It can be either plugged into the end of the keyboard cable or installed inside the computer case, or inside the keyboard itself. Hardware keylogger doesn’t use any computer resource because of this they are hard to detect by the anti-viral software or scanners, also it doesn’t use computer hard disk for storing keystroke logs, and it can be placed in different locations. Physical installation is the major disadvantage of this keylogger. Software keylogger collect keystroke data and store them temporary on local storage then send them to the attacker who installed the keylogger. It could also be the case where keystroke data sent directly without temporal local storage. Touchlogger pose security and privacy risks on users. In Android, software touchlogger are the only type that may exist. In that case, the touchlogger must have certain permissions to record, store, and send the keystroke data.

3. PROPOSED SYSTEM

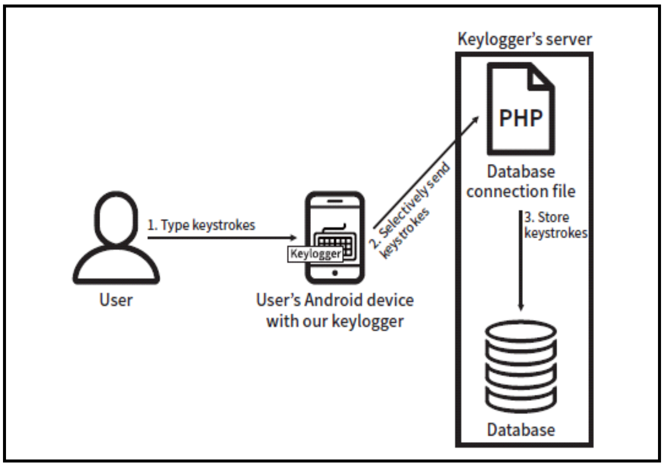

Fig.5.1:- Overview of our logging System

We have proposed a system to capture keystrokes given from keyboard and store those captured keystrokes in a log file on the mobile and send that log file to the desired destination when the user’s mobile phone will be connected to the internet. The keyboard is the primary target for touchlogger to retrieve user input because it is the most common user interface with a mobile. Using keyboards, we are actually taking the input from the user and that all information gets through keyboard will save automatically in one log file which contains all the data typed by the user. There are legal and illegal activities. A legal activity would be capturing keystrokes and storing the captured keystrokes in a log file for auditing purposes. An illegal activity would be stealing credit card information, login details, personal details, etc.

The legal activity mentioned above could be used for multiple purposes such as improving security. In illegal activity mentioned above is generally used for stealing personal information. After installing the Touchlogger application on the mobile phone, it starts running the background in stealth mode and captures every keystrokes given through the keyboard. The Fig 5.1 shows the Android keyboard attack scenario, which involves the attacker and the end user. Here the developer of this malicious keyboard deploys this malicious application on Google Play Store as well as unofficial market. The end user downloads this third party keyboard application and when he starts using this third party keyboard application, the malicious code in this application starts running in the background.

This application requires the following permissions: INTERNET_PERMISSION, READ_EXTERNAL_STORAGE, WRITE_EXTERNAL_STORAGE.

4. METHODOLOGY

The Fig. 5.1 show the overview of our system. In the Android platform, the key logging process can be implemented; our touchlogger application collects user inputs given through the keyboard. When a user types a keystroke, our touchlogger application can obtain the information about the input type for each text field in an application or a web page.

5. ANALYSIS

In Fig.6.1 we show the download distribution. From the figure, we see that thirty five third party keyboard applications deployed in the market and their increasing popularity amongst users.

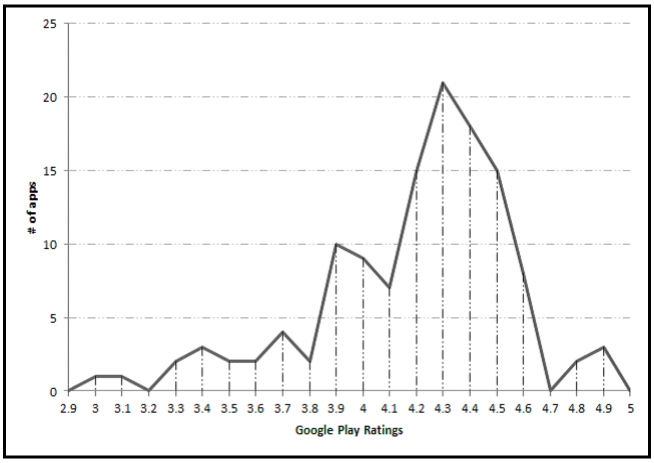

Fig. 6.1- The download frequency for the collected third-party keyboards

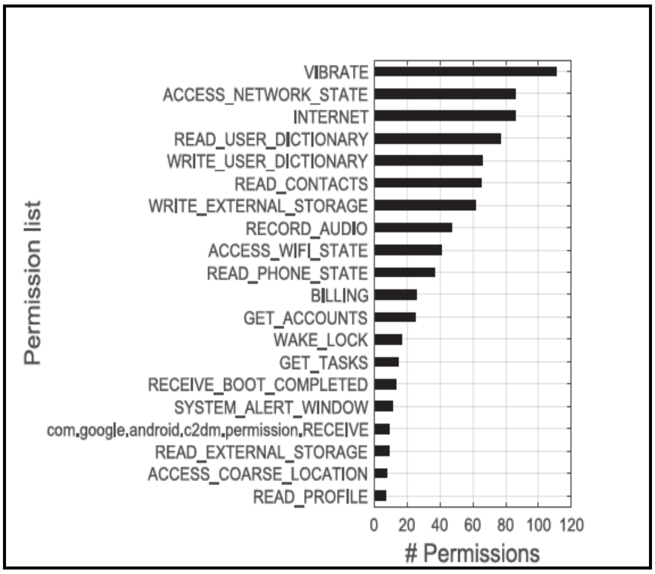

Fig. 6.2- The rating frequency for the collected third-party keyboards.

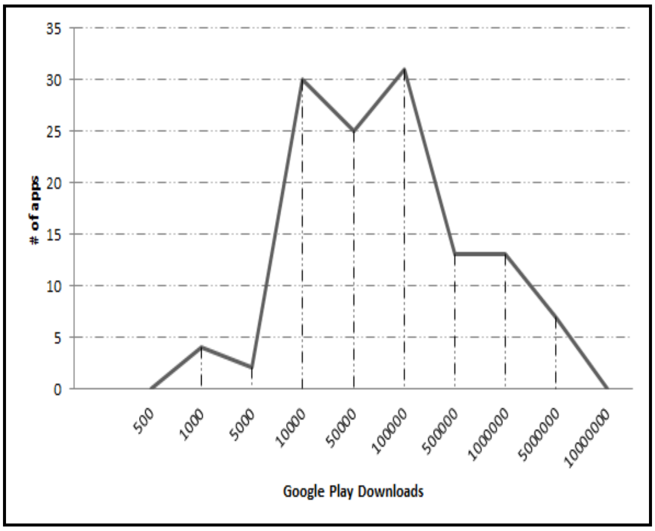

Fig. 6.3- Top 20 most used permissions in the third-party apps on Play Store.

Fig. 6.2 shows the ratings distributions among applications, 62% of the applications have rating range between 4.1 and 4.6 .This figure justifies the user feedback after using third party keyboard applications. The rating of the application depends on users experience and popularity.

Fig. 6.3 shows the list of the top 20 most commonly requested permissions and their distribution. We can see that some potentially dangerous permissions (INTERNET, READ_CONTACTS, WRITE_EXTERNAL_STORAGE, RECORD_AUDIO and READ_PHONE_STATE) were popularly requested although they do not seem to be necessary for the application to function properly**.**

6. PROBLEM STATEMENT AND SCOPE

An average mobile user whose mobile has Internet access is exploited due to the insecure environment which is quite often not realized by the user .Hence whenever the user enters a password, or personal credentials, or other sensitive data, it could be easily detected and captured by an application installed on the device. A users perception of a secure environment is having a trust in the device i.e. typing a password on a keyboard. It is experienced that a monitoring tool often runs in stealth mode, most of the application are inspired with this concept. Touchlogger presents a serious threat to the security of passwords on individual. User cannot identify the presence of touchlogger since it runs in background and also it is not listed in task manager.

7. DETAILS OF HARDWARE AND SOFTWARE

Computer hardware is the physical part of the computer, as distinguished from the software that executes or runs on the hardware. The hardware of the computer is infrequently changed. Computer software is a set of machine-readable instruction that directs a computer machine to perform specific intended operations. Computer is non-tangible.

Hardware Requirements:

- Intel Dual Core CPU or higher
- 2 GB of RAM

- 400 MB of free disk space
- Android Smartphone with 1GB RAM

Software Requirements
- Windows /Mac/Linux Operating System
- Android Studio IDE
- Java Development Kit

Android API of 2.3 or higher.
We can also implement the project on an Android simulator which would have the same functionalities as an actual Smartphone.

The IDE used will be android studio which is an official IDE provided by Google for Android developers.

8. DESIGN DETAILS

A. Input Method Framework Architecture

Android input method framework architecture is composed of following three components: input method manager, client applications and input method.

The InputMethodManager is the mediator that handles communication between the components. Android InputMethod interface contains any method that can generate key events and texts, such as text messages, emails, different languages characters, and send text back to the application while handling the inputs. An Android application that contains an instance of EditText or TextView need not to worry about implementing the InputMethod interface, instead it relies on the standard interaction provided by these two components. Implementing an input method in Android is done through deriving a class from InputMethodService or any of its subclasses. It involves providing two types of interfaces: top-level interface and session interface. The former provides full access to the input method and it is only accessible by the system. Clients should hold the BIND_INPUT_METHOD; otherwise the system won't bind and will consider that method as compromised. The session interface is what client applications use to communicate with the input method.

B. Creating an Input Method

In this section, we show how to create a keyboard in Android, which is an example of input method (IME). Throughout the rest of the paper we use the terms keyboard and IME interchangeably.

C. Declaring Input Method Components

The manifest should contain service declaration, permission request, metadata, intent filter, and an optional "settings" activity. The intent filter must match the action.view.InputMethod, the metadata defines the attributes of IME service, and the settings activity is to allow the user pass new options.

D. Designing the Input Method UI

Android keyboard (IME) main components are: the layout component, manifest entry, xml file, and program component. First of all, the KeyboardView which is a view that renders a virtual keyboard. It handles rendering of keys and detecting key presses and touch movements. Like any other views, KeyboardView should be included in the layout file of the application. The XML description of the keyboard is loaded in the Keyboard class which stores the attributes of the keys. A keyboard consists of rows of keys. There are some UI design considerations for IMEs, like handling different screen sizes and handling different input types (e.g. Text, Numbers, URL, etc.). Developers can handle these issues by modifying the XML files and program section.

E. Sending Text to the Application

The purpose of the IME (e.g. keyboard) is to provide the interface, handle user events and then send the text to applications. Text can be sent to the application in two ways. First by sending the individual key events or editing the existing text. In both cases, an instance of InputConnection is required to deliver the text. This instance can be retrieved by calling InputMethodService.getCurrentInputConnection ().

CONLUSION

This paper simply describes a method that can be used to break the security in android smartphones. Hence we showcase the vulnerabilities in the open source mobile operating system Android. Hence users should be careful while downloading or using a third party application in their android smartphone .we also derive that personal data is also at risk if an application like ours is implemented.

REFERENCES

[1] D. Damopoulos, G. Kambourakis, S. Gritzalis .From touchlogger to touch logger: Take the rough with the smooth. Pg 102-114

[2] Junsung Cho, Geumhwan Cho and Hyoungshick Kim. Keyboard or Touchlogger? A security analysis of third-party keyboards on Android. Pg 1-4

[3] Tuli, Preeti, and Priyanka Sahu. "System Monitoring and Security Using Keylogger." International Journal of Computer Science and Mobile Computing 2, no. 3 (2013): 106-111.

[4] Yuko Hirabe, Yutaka Arakawa and Keiichi Yasumot. Logging All the Touch Operations on Android. 7th international conference on mobile computing and ubiquitous networking 2014. Pg. 93-94.