THEFT ANALYSIS DETECTION FOR ANDROID SMARTPHONE

Dushyant Shinde[1], Abhishek Dhumal [2], Vishnu Ingole [3], Navnath Polekar [4], Prachi Sorte[5]

[1] *Student, Information Technology, RMD Sinhgad School of Engineering, Maharashtra, India*
[2] *Student, Information Technology, RMD Sinhgad School of Engineering, Maharashtra, India*
[3] *Student, Information Technology, RMD Sinhgad School of Engineering, Maharashtra, India*
[4] *Student, Information Technology, RMD Sinhgad School of Engineering, Maharashtra, India*
[5]*Teacher, Information Technology, RMD Sinhgad School of Engineering, Maharashtra, India*

ABSTRACT

Smartphones have spreading widely throughout group of people due to the availability of office applications, Internet, games, vehicle guidance using location-based services apart from conventional services like voice calls, SMSs and multimedia services. Android devices have gained huge market share due to the open architecture of Android, and the popularity of its application programming interface (APIs) in the developer community. The security issue of Android has caught great concerns among mobile users and researchers. The proposed system implemented on android phones equipped with a front-face camera. The spy camera apps are completely translucent to phone users and work without causing any abnormal experiences. If Mobile gets stolen, the phone camera is launched automatically without the attacker's notice, and it captures the photo of attacker. When Wi-Fi access is enabled, the captured photo of attacker along with its location information is transmitted to the user via Email/MMS services. The proposed system introduces possible system based on spy camera. The system should appear normal to user experience. It runs stealthily and silently so that they do not cause a user alert. It has a translucent view make no sound or vibration, and check phone resource utilization before launching themselves.

Keyword: - *Mobile phones, Camera, Android.*

1. INTRODUCTION

When talking about privacy protection, most mobile phone users pay attention to the safety of SMS, email, contact list, calling history, location information and private files. They may be surprised that the phone camera could become a traitor, for example, attackers could stealthily take pictures and record videos by using the phone camera. Nowadays, spy camera apps have become quite popular on Android application markets. As for Google Play, there are nearly one hundred spy camera apps, which allow phone users to take pictures or record videos of other people without their permission. However, believe it or not, phone users themselves could also become victims.

Attackers can implement spy camera in malicious apps such that the phone camera is launched automatically without the device owner's notice and the captured photos and videos that contain user's daily activities and conversations are sent out to the remote user. Even worse, according to a survey on Android malware analysis, camera permission ranks 12th of the most commonly requested permissions among benign apps, while it is out of the top 20 in malwares. The popularity of camera usage in benign apps and relatively scarce usage in malware lower user's alertness to camera-based multimedia attacks. The security issue of Android has caught great concerns among mobile users and researchers. There are more vulnerability related to camera like any application can take the access over the camera without its knowledge to user.

The system must focuses on security issues related to mobile phone cameras. It should discover several new theft attacks that are based on the use of phone cameras. It must focuses on tracing location of attacker also. System can gives the notification to User incase of any application can access the camera if its default system camera application then it does not give the notification.

2. RELATED WORK

The security issue of Android has caught great concerns among mobile users and researchers. The system implements the vulnerabilities related of phone cameras. Specifically, it discover and present several camera-based attacks including the basic camera attack and advanced passcode inference attacks. The system presents the basic attack, and two passcode inference attacks: the application-oriented attack and screen unlocking attack. System runs these attacks along with several popular antivirus softwares to test their stealthiest, and conduct experiments on video-based passcode inference attacks. The results show the existence to the conveniently and effectiveness of these attacks [1].

A number of recent works have studied the issue of obtaining private information on smartphones using multimedia devices such as microphones and cameras. For example, Soundcomber is lightweight and stealthy. It uses targeted profiles to locally analyze portions of speech likely to contain information such as credit card numbers. The system present Sound comber, its permissions, which can extract a small amount of targeted private information from the audio sensor of the phone. Soundcomber performs efficient, behaving, local extraction, thereby greatly reducing the communication cost for delivering stolen data [2].

In order to protect Android users, applications access to phone resources is restricted with permissions. An application must obtain permissions in order to use sensitive resources like the camera, microphone, or call log. For example, an system require the read contact permission in order to read entries in a users phonebook. Obtaining permissions is a two-step process. First, an application developer declares that his or her application requires certain permissions in a file that is packaged with the application. Second, the user must approve the permissions requested before installation. Each application has its own set of permissions that reflects its functionality and requirements. Users can weigh the permissions against their trust of the application and personal privacy concerns. Android smart phone user’s privacy relates to primarily focused on location tracking and sharing. Although location sharing is an important aspect of smart phone privacy. Android privacy researchers have built several tools to help users avoid privacy violations. Most research has focused on identifying malicious behavior. Creation of a sensor-access widget, which visually notifies the user when a sensor likes the camera is active. This system evaluates android users to pay attention, to understand, and act on permission information during installation [3].

Android security enforceable mechanisms, threats to the existing security enforceable and related issues, malware growth and surreptitious action techniques employed by the malware authors, in addition to the existing detection methods. Academia and industry researchers have proposed solutions and frameworks to analyze, and detect the Android malware threats. Goal of the system can be either app security assessment, analysis or malware detection. An app security assessment solution determines the vulnerabilities, which if exploited by an adversary, harms the user and device security. Analysis solutions check for the malicious behavior within unknown apps, whereas detection solutions aim to prevent the on-device installation. Methodology to achieve the above goals can be either static or dynamic analysis based approaches to detect malware [4].

The proposed system approach operates on image captured by inexpensive commodity cameras, such as those found in modern Smart phones. The low resolution of these cameras makes visual analysis difficult, even for humans, and severely limits the possibility of directly identifying on-screen text. Low pixel resolution of the phone image is one of the key problems it encountered. It can be caused by a variety of factors, including camera aperture, wide angle lenses, motion blur, and large distance. All of these make the phone’s appearance so blurry that no reliable features can be extracted, and so phone stabilization and alignment methods fail in certain cases. It can be believe this could be addressed by using more revealing capture techniques, as in, which addresses the allied problem of capturing clear images from reflections. These are possible defenses against the attacks proposed in this work [5].

3. ATTACKS BASED ON SPY CAMERA

The proposed system first introduces possible attacks based on spy camera. The attacks should appear normal to user experience. The main challenge is to make the attacks run stealthily and silently so that they do not cause a user alert. Specifically, the attacks are supposed to have a translucent view, make no sound or vibration, and check phone resource utilization before launching themselves. The general architecture should include the following six parts.

Step 1: To prevent the user from suspecting, the application should consider the current CPU, memory usage and battery status. Users tend to be concerned about the unsmooth experience, and check if any app or service is running in the background. Similar concern happens with energy consumption, especially when the phone’s battery is low and is not being charged. A camera attack could drain the battery faster than the user’s expectation, and cause user suspicion about possible attacks. Hence, before launching the attack, malicious camera apps want to ensure that system resources are plentiful. For Android phones, memory usage could be obtained through the getMemoryInfo() function of ActivityManager, while current battery level and charging status can be obtained by registering a BroadcastReceiver with ACTION BATTERY CHANGED [1].

Step 2: The difficult task is to hide the camera preview. At the beginning, the layout containing the SurfaceView is inflated into a view via LayoutInflater.inflate. Then the app can change the parameters of that view by setting the attributes of WindowManager, LayoutParams. Two important attributes must be set: TYPE SYSTEM OVERLAY, which makes the preview window always stay on top of other apps; the other one is FLAG NOT FOCUSABLE, which disables the input focus of a spy camera app such that input values would be passed to the first focusable window underneath. This would turn the camera preview into a floating and not focusable layer. Then the app changes the size of preview (SurfaceView) to the minimum pixel (1 pixel), which human eyes cannot notice. This cannot be set directly through setPreviewSize(). Instead, the app needs to get the layout parameter of SurfaceView by using SurfaceView, getLayoutParams(). Notice that the type of SurfaceView, getLayoutParams () is ViewGroup. LayoutParams instead of the a aforementioned Window Manager, LayoutParams. Finally, the app can add the hidden preview dynamically to the window by the addView function [1].

Step 3: After setting up the layout, the attack could be launched as follows: initialize the SurfaceHolder, choose which camera (front or back) is used, and open the camera to take pictures or record videos. The photo/video data are supposed to be stored in disguises, including using confusing file names and seldom visited directories. The app releases the camera after the above actions [1].

Step 4: The last step of the attack is to transmit the collected data to the outside. Since cellular network usage and MMS may cause extra fees, the best choice is to wait until free Wi-Fi access is available. For example, it could use the javax mail to send the data as an email attachment. Most email systems limit the maximum size of attachments, so the length of the captured video should have an upper bound specific to the email service [1].

4. SYSTEM ARCHITECTURE

The security issue of Android has caught great concerns among mobile users and researchers. There are more vulnerability related to camera like any application can take the access over the camera without its knowledge to user. System can capture the picture and store on SD card in such format that cannot be recognized by the User and Email to the owner.

In case of mobile stolen, User can track the location of mobile by using its IMEI number or name it can view the latitude and longitude along with its area name. System can gives the notifications to User incase of any application can access the camera if its default system camera application then it does not give the notification.

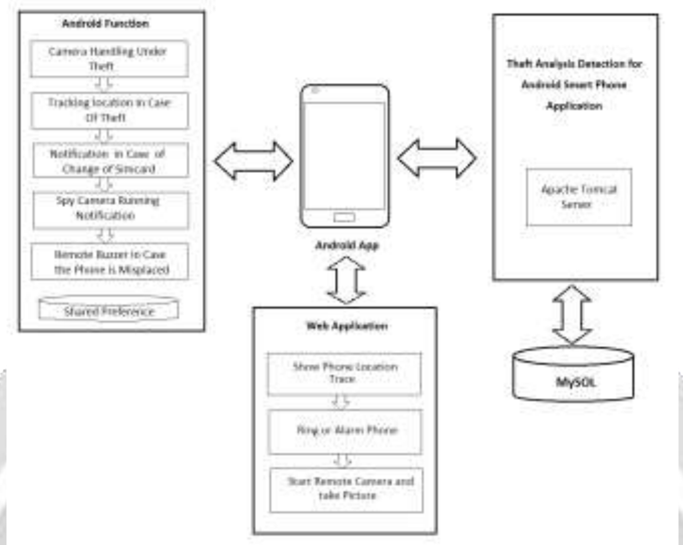

Android Function
Camera Handling Under Theft
Tracking location in Case Of Theft
Notification in Case of Change of Simcard
Spy Camera Running Notification
Remote Buzzer in Case the Phone is Misplaced
Shared Preference
Android App
Theft Analysis Detection for Android Smart Phone Application
Apache Tomcat Server
MySQL
Web Application
Show Phone Location Trace
Ring or Alarm Phone
Start Remote Camera and take Picture

Fig -1: Architecture of proposed system

These are some feature of proposed system architecture:

1) Camera Handling under Theft**:** Take Attacker image through spycam.

2) Tracking Location In Case of Theft: Trace the current location. GPS based distance algorithm and Haversine algorithm *is* used for tracing location.

3) Notification in Case of Change of Simcard: If simcard is changed or removed then owner gets the notification

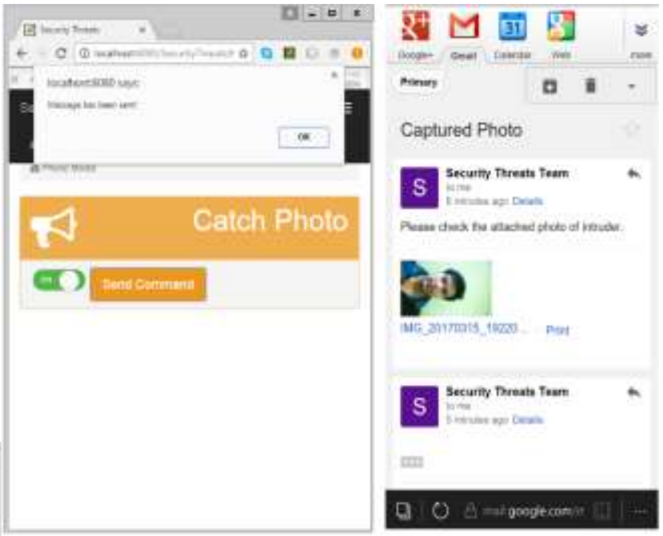
4) Remote Buzzer in Case the Phone is misplaced**:** User can play a remote buzzer or ring an alarm in case if he/she wants to find out a misplaced handset in the house or outdoor

4.1 Web Application Module

This module provides the interfacing of the system to user. In this module user can login. In web application, only authenticated user can have access to the system. User should be able to control and detect the camera using Android Mobile phone through the server. User should notify in case of any one change the SIM card. User can play the alarm for finding mobile in case of misplaced in the house.

4.1 Android Application Module

This application is provided for end user registration. End user can get access to system by userid and password after registration. This application connected to server through the IP address. So user has full control on his mobile through web application. User can get location of mobile. User can get photo of theft, can get sim change notification to server through this application.



localhost:8080 says:

Message has been sent!

Captured Photo

Security Threats Team
to me
5 minutes ago Details

Please check the attached photo of intruder.

IMG_20170315_19220...

Security Threats Team
to me
5 minutes ago Details

Fig -2: Proposed System

5. CONCLUSION

The proposed system focus on security issues related to mobile phone cameras. Specifically, proposed system discover several new attacks that are based on the use of phone cameras. The application of proposed system implements the attacks on real phones, and demonstrates the feasibility and effectiveness of the attacks. Furthermore, It is a lightweight defense scheme that can effectively detect these attacks. This system can perform faster in case of the mobile phone is stolen then the android mobile application can take the picture of user and send it to the owner. User can track the theft by log in into web server there user can view the latitude and longitude of android mobile along with the area name.

6. REFERENCES

[1]. Longfei Wu, Xiaojiang Du, Li Wang, Xinwen Fu, Ralph O. Mbouna, and Seong G. Kong "Analyzing Mobile Phone Vulnerabilities Caused by Camera", OCT 2014.

[2]. Kehuan Zhang, Xiaoyong Zhou, Mehool Intwala, Apu Kapadia, XiaoFeng Wang "Soundcomber: A Stealthy and Context-Aware Sound Trojan for Smartphones", JULY 2013

[3]. Adrienne Porter Felt, Elizabeth Ha Serge Egelman, ArielHaney, ErikaChin, David Wagner, " Android Permissions: User Attention, Comprehension, and Behavior", March-April 2012

[4]. Parvez Faruki, Ammar Bharmal, VijayLaxmi, Vijay Ganmoor, ManojSingh Gaur, Mauro Contiand Muttu krishnan Rajarajan "Android Security: A Survey of Issues, Malware Penetration and Defenses", JANUARY 2015

[5]. Rahul Raguram, Andrew M. White, Dibenyendu Goswami, Fabian Monrose and Jan-Michael Frahm "iSpy: Automatic Reconstruction of Typed Input from Compromising Reflections", OCT 2011.