Remote Live Forensics for Android Devices

Jonathan Ming
Department of Engineering and Computer Science
Azusa Pacific University
Azusa, CA, USA
jming13@apu.edu

Mengjun Xie
Department of Computer Science
University of Arkansas at Little Rock
Little Rock, AR, USA
mxxie@ualr.edu

***Abstract*—This research focuses on developing a new forensic mechanism to integrate Android devices into existing remote live forensic frameworks. Our mechanism allows incident responders and forensic data analysts to collect detailed usage data from a fleet of mobile devices in a way currently only available on computers. A prototype system named *DroidGRR* was developed to integrate Android devices into the GRR Rapid Response forensic framework. Analysis of our prototype shows that usage data can be successfully collected from a remote Android device through the GRR framework. Our findings indicate that the proposed solution is attainable and can provide new and rich data to incident responders, corporate IT administrators, and forensic analysts regarding the usage of mobile devices.**

***Index Terms*—mobile security, Android, incident response, mobile forensics, data collection, monitoring**

I. Introduction

Mobile devices, as they become more powerful, are becoming increasingly prevalent, especially in the U.S. [1]. A 2014 survey shows that roughly 80% of U.S. employees have a smartphone with Internet access, and 49% have a tablet [2]. As mobile devices gain prevalence, they also have access to more sensitive data than ever before. However, the security measures for protecting those data have not yet adapted to match such a wide adoption of mobile devices. This research aims to push forward these security measures, focusing on the scalable remote live data collection and incident response mechanism for mobile devices.

II. Background

If a smartphone or tablet becomes compromised, incident responders and forensic analysts currently have quite limited options to examine the compromised device remotely. During this research, nine existing mobile device forensic tools [3, 6-14] were examined. We found that only one of them, FireEye Mobile Threat Prevention, provided the ability to remotely collect forensic data from a mobile device without rooting or jailbreaking [3]. The others either required physical access to the device or required insecure modifications to the mobile device's operating system. An ideal mobile forensic solution does not suffer from these problems.

The solution sought by this research aims to achieve the following features: 1) requiring no physical access to the device, 2) requiring no rooting of the device, and 3) being able to access live, relevant data while the phone's internal storage is encrypted. In addition, this solution should leverage and be integrated into existing forensic frameworks, in order to be scalable and easily adopted, and be able to provide aggregate analytics across both mobile and non-mobile devices.

III. Related Work

Two existing forensics projects are particularly relevant to this research. One is GRR Rapid Response [4] and the other is DroidWatch [5]. Some characteristics of these two projects are highlighted as follows.

1) GRR Rapid Response

- "An incident response framework focused on remote live forensics"
- Initially developed at Google, was open-sourced
- Requires one central Linux server; supports numerous Windows, Mac OS X, and Linux clients
  - Largest known open-source deployment has approximately 30,000 clients installed

2) DroidWatch

- A prototype monitoring app designed to collect data from Android devices
- Initially developed by Justin Grover in 2013 as an open source project, no further development since then
- Not integrable with existing forensic frameworks

IV. Design

To facilitate integration, the prototype developed for this research (named *DroidGRR* after the style of DroidWatch) applies the model already employed by GRR Rapid Response. This model follows a request-and-response format (see Fig. 1). First, a framework administrator or data analyst accesses the central server, which then sends requests to a selected client or group of clients. After that, the clients collect the data from their respective host machines and respond to the server with the requested data in an asynchronous manner.

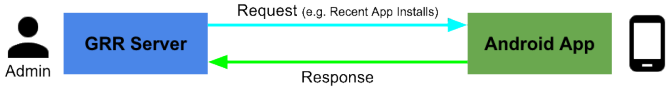


Fig. 1 Request-Response Design Model

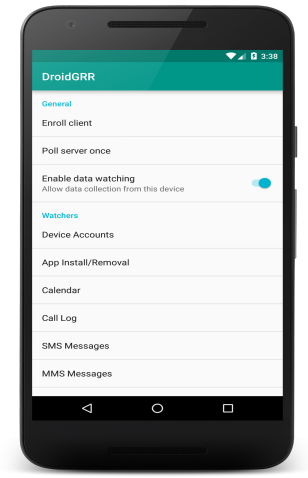
In order to match this model, DroidGRR is required to behave as a standard GRR client. In other words, it has to listen for and respond to requests issued by the GRR server. This design marks the primary difference between DroidGRR

978-1-5090-3065-1/16/$31.00 ©2016 IEEE

and DroidWatch. DroidWatch follows a simple monitoring model designed by Grover, in which interested data are collected continuously and sent to a server automatically [5]. While that approach is straightforward and easy to implement, it is incompatible with GRR's operational model. Therefore, Grover's design model was discarded for this research in order to facilitate integration with GRR.

V. Implementation & Evaluation

During this research, a prototype system of DroidGRR was developed. The screenshot of the DroidGRR application is shown in the left figure. Due to time constraint, the entirety of the model was not implemented. At the time of this writing, DroidGRR has functioning watchers, can communicate with the GRR server, can enroll with the server as an available client, and can poll the server for flow requests.



Because flow request handling and response delivery could not be implemented in the available time, a complete evaluation of the model side by side with a standard GRR client could not be performed. However, the portion of the implementation that was completed did shed light onto the design model and serves well as a proof of concept. DroidGRR, even in its current state, demonstrates the feasibility of live, remote usage data collection from mobile devices and the feasibility of integrating mobile devices into existing live forensic frameworks without compromising the device's security (via rooting).

VI. Challenges

The primary challenge of this research was fashioning the solution to be integrable with existing forensic frameworks. In order to be integrated with GRR Rapid Response, the prototype has to behave and communicate with the server in the same manner GRR's existing PC clients behave and communicate, which was difficult to implement in Android.

A secondary challenge of this research was dealing with the inherent limitations of the Android operating system. In particular, retrieving device logs was a problem in this area. Logcat application activity logs can provide invaluable insight into the inner workings of the device. However, as of Android 4.1 installed apps are denied permission to view Logcat. The only way to circumvent this prohibition on log access is to root the device, which is undesirable for this solution.

VII. Conclusion

DroidGRR's success at collecting data from an Android device and enrolling with a GRR server clearly indicates that mobile device integration into existing forensic frameworks is attainable. Even though DroidGRR's full implementation has not been completed, the developed prototype and functions are sufficient to validate the design described in Section IV. This design will serve as a model for our further development of scalable incident response tools for mobile devices. It can also serve as a model for including mobile devices into existing forensic frameworks.

Moreover, after implementing DroidGRR and being more familiar with GRR Rapid Response framework and Android operating system, it became clear to us that continuing to use and extend the request-response model will not only work, but also enable entirely new categories of data to be added to the GRR framework (e.g., records of dubious MMS messages containing malicious code). These new categories of data would augment GRR's incident response capabilities by enabling responders to evaluate mobile devices as well as traditional computers when considering attack vectors.

Acknowledgment

This work was supported by the Research Experiences for Undergraduates (REU) Program of the National Science Foundation under Award Number 1359323.

References

[1] M. Anderson, "Technology Device Ownership: 2015," Pew Research Center, October 2015.

[2] J. Harter, S. Agrawal, and S. Sorenson, "Most U.S. Workers See Upside to Staying Connected to Work," Gallup, Inc., April 2014. http://www.gallup.com/poll/168794/workers-upside-staying-connected-work.aspx

[3] FireEye, Inc., "FireEye Mobile Threat Prevention Data Sheet." https://www.fireeye.com/content/dam/fireeye-www/global/en/products/pdfs/fireeye-mobile-threat-prevention.pdf

[4] "GRR Project FAQ," GRR Rapid Response Documentation https://github.com/google/grr-doc/blob/master/faq.adoc

[5] J. Grover, "Android forensics: Automated data collection and reporting from a mobile device," Digital Investigation, vol. 10 Supplement, pp. S12-S20, August 2013.

[6] Katana Forensics, "LANTERN Device Acquisition and Analysis." https://katanaforensics.com/products/lantern/

[7] Guidance Software, "Encase Forensic Product Overview," p. 4.

[8] M. Shannon, "F-Response and Android," F-Response. https://www.f-response.com/blog/f-response-and-android

[9] AccessData, "Mobile Phone Examiner Plus." http://accessdata.com/solutions/digital-forensics/mpe

[10] NowSecure, "NowSecure Forensics." https://www.nowsecure.com/forensics/

[11] Open Source Android Forensics, "OSAF Community Site." http://osaf-community.org/home.html

[12] Evidence Talks, "SPEKTOR Forensic Intelligence." http://www.remoteforensics.com/index.php/products/spektor

[13] Lisa Phifer, "How mobile device encryption works to protect sensitive data," TechTarget, March 2013.

[14] Heather Mahalik, "Open Source Mobile Device Forensics," http://www.nist.gov/forensics/upload/6-Mahalik_OSMF.pd