Asian Journal of Computer Science and Technology (AJCST)
Vol.2.No.2 2014 pp 45-51
available at: www.goniv.com
Paper Received :05-03-2014
Paper Published:28-03-2014
Paper Reviewed by: 1. John Arhter 2. Hendry Goyal
Editor : Prof. P.Muthukumar

AN ANTI-FORENSICS APPROACH FOR ANDROID OS

C Ram Prabhu[#1], P Savaridassan[*2]

[#]Information Security and Cyber Forensics Department,
SRM University
Kattankulathur, TamilNadu, India.
ram_prabhu@srmuniv.edu.in savaridassan.p@ktr.srmuniv.edu.in

ABSTRACT

Android devices such as smartphones and tablets have emerged as an essential part of people's personal life. Smartphones, in meticulous, with rising storage capability and computing abilities, provide a neat mobile computer, and so, can cover details about call histories and friend's contact lists, but also web browser history, passwords, media files, and credit card records. All these protected files can fix the life or decease of an individual. The assured elimination of sensitive data can increase customer confidentiality in various situations and, also in some extreme environments. At current time we are using only zero pass erase method for securely removing data from the smart phones. Here we proposed three new techniques for secure removing of data from the NAND flash drives and all the storage component of the smartphones. We used three new technologies (three pass, seven pass and 35 pass) for removing data securely and improving the privacy of the users who are using Android devices.

Keywords— Android operating system, Digital Forensics, Mobile Forensics, Anti-Forensics, Mobile Anti-Forensics Counter-Forensics, Android Forensics, Android anti-forensics, santization, NAND flash memories.

1. INTRODUCTION

According to gartner.com, there were 875 million Android devices in use as of January 28, 2014. Number of Smartphones around the world top 1 billion projected to double by 2015. Mobile devices such as mobile phones and smartphones have emerged as an important part of people personal life. Smartphones, in particular, with computing and increasing storage capacity, gives user access a smart mobile computer, which gives access, to users information about call histories and contact lists, but also browser history, mails, account passwords, media files, and credit card numbers.

As a consequence, smartphones have outsold PCs (including pads) for the first time in 2011 [1]. With their extensive working, mobile devices can able to provide the finest source of civil and criminal confirmation. Mobile forensics has developed as an key research area, providing the means of extracting critical evidence to detectives. However, the ever-changing technologies and the lack of standardization means that mobile forensics present some challenges. These will be discussed in more detail in the following section.

Because the field of mobile forensics is relatively new, there is even less research regarding anti-forensics, the process of negotiating digital proof. Here, we will focus on testing existing anti-forensics tools for smartphones and their effectiveness against commercial forensic applications and also introduce some new technology for secure removal of the information from the device and flash memory. For the understanding of the android phones first we understand the architectures of the android OS. At its core, Android OS was built on the Linux kernel. When operating on a hard drive, the Linux system device goes to the first physical memory. In addition, Linux only understands character and block devices, such as keyboards and flash drives, respectively. Using Linux on flash, though, a Flash Transition layer provides the system device functionality. A Memory Technology Device (MTD) is needed to provide an interface between the Linux OS and the physical flash device

because flash memory devices are not seen as character or block devices. To this extent, two platforms have been used, Android and Apple IOS, and a series of freely available anti-forensic approaches. According to [2], Google and Apple have been in the top of the smartphones market share in 2014 by operating system. We are introducing new approach in this paper for the secure erase of the private information from the smart phones. This paper is organized in the following manner; the next section describes background of anti-forensic techniques of androids. Afterwards, we describe our suggested anti-forensic method and techniques for android mobile phones.

2. BACKGROUND

There are currently some research dealing with the problem of secure modification in flash memories (to the best of the authors' knowledge), few studies have recently proposed techniques for the secure erasing of NAND flash memory drives that are commonly used in smart phones. NANDs are different from hard drives and it is uncertain whether techniques and commands developed for hard drives [3] will be effective on NANDs. In [4] and [5], all the data is encrypted and each file has its own encryption key which is stored in the header of the file. When wanting to delete a single file, simply delete or overwrite the header. Encrypting the file system or modifying it in order to apply anti-forensics techniques creates a great deal of suspicion during a forensic analysis. In [6], exploiting a security feature of Android, it is possible that digital evidence is hidden in a private folder which is inaccessible to third-party applications. The private folder is a private directory, created when an application is installed, in which it is possible to save any file type (e.g., text files, multimedia files). According to [6], when a given Android application is uninstalled, the entire set of the related information, including data files and directories, is logically deleted from the file system.

3. METHODOLOGY.

Criminals could use smart phones for a number of activities such as committing fraud over e-mail, harassment through text messages, trafficking of child pornography, communications related to narcotics, etc. The data stored on smart phones could be extremely useful to analysts through the course of an investigation. Indeed, mobile devices are already showing themselves to have a large volume of

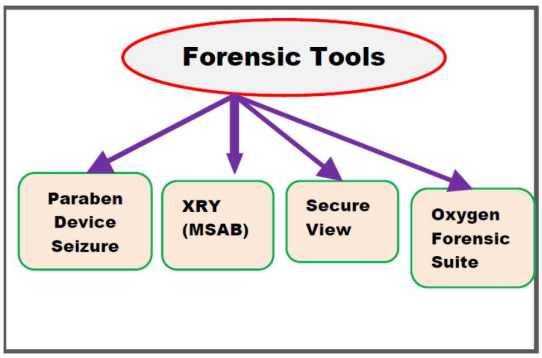

Fig. 1: Forensic Tools

probative information that is linked to an individual with just basic call history, contact, and text message data; smart phones contain even more useful information, such as e-mail, browser history, and chat logs. Mobile devices probably have more probative information that can be linked to an individual per byte examined than most computers and this data is harder to acquire in a forensically proper fashion.

There are three primary methods of acquiring data using forensic tools:

- Manual Acquisition
- Physical Acquisition
- Logical Acquisition

In Manual acquisition, the user interface can be utilized to investigate the content of the memory. The device is used as normal and pictures are taken from the screen. This method has the advantage that the operating system makes the transformation of raw data into human interpretable format. The disadvantage of this method is that only data visible to the operating system can be recovered and that all data are only available in form of pictures.

In contrast, Physical acquisition implies a bit-by-bit copy of an entire physical storage. This acquisition method has the advantage of allowing deleted files to be examined. Physical extraction acquires information from the device by direct access to the flash memories. Generally, this is harder to achieve because the device vendors needs to secure against arbitrary reading of memory so that a device may be locked to a certain operator. Logical acquisition implies a bit-by-bit copy of logical storage objects that reside on a logical store. This method of acquisition has the advantage that system data structures are easier for a tool to extract and organize. Logical extraction acquires information from the device using the interface for synchronizing the contents of the phone with the analyzing device (e.g., PC). This method usually does not produce any deleted information, due to it normally being removed from the file system

of the phone. Paraben Device Seizure [12] is an advanced forensic acquisition and analysis tool for examining cell phones, PDAs, and GPS devices. The tool is able to acquire both logical and physical data. It contains a report generation tool that allows for the convenient presentation of data. The tool is able to generate reports using the report wizard in csv, html, text or xls formats. The tool is designed to be able to recover the deleted data and retrieve physical data from some devices and has a fairly simple user interface.

XRY [13] is developed by Micro Systemation (MSAB), and based on our evaluation [34] is one of the best-dedicated mobile device forensic tools. 'XRY Complete' is a package containing both software and hardware to allow both logical and physical analysis of mobile devices. The unified logical/physical extraction wizard in XRY and the resulting reports help to show the examiner the full contents of the device in a clean and professional manner. The tool is able to connect to the cell phone via IR, Bluetooth or cable interfaces.

SecureView [43] tool provides various smart features for cell phone forensics such as: svSmart (ability for streamlining and presetting conditions to attain evidence in the field quickly); svPin (ability for unlocking CDMA cell phone passwords); svLoader (ability to download, analyze, verify and validate other sources and aid in creating csv files and upload, back up files from RIM and iPhone); and svReviewer (ability to share data without multiple licenses). The tool contains a SIM card reader to extract data from the SIM cards of GSM phones. The tool acquires data via cable, Bluetooth or IR interfaces. Reports are then generated in a print ready format. It provides a friendly interface and strong phone support for over 2000 phones. The tool also provides the option of generating a detail report of data and output in a pdf format. In addition, the report can be merged with the import report from other forensic tools by using Svprobe.

B. Anti-Forensic Tools

Anti-Forensic is a quite young and immature discipline even more if we consider the Mobile Environment (ME); regarding ME, a number of difficulties and issues during forensics analysis are still to overcome (Jansen et al., 2008), hence the possible shapes of AF techniques are continuously and rapidly evolving (Geiger, 2005; Peron; Berghel, 2007). Currently, there is no unique and standard definition of AF, while several definitions exist and focus on different and specific aspects. Among those, some focuses on breaking forensic tools or avoiding the detection of evidence [17] while some others relate AF to system intrusions [18]. However, in accordance with [19], in this paper we consider AF to be any attempts to compromise the availability or usefulness of evidence in the forensic process. The availability of the evidence can be compromised by preventing its creation, hiding its existence and by manipulating the evidence as well; the usefulness can be compromised by deleting the evidence or by tampering its integrity. By the comprehension and the study of the AF techniques, a number of useful conclusions and guidelines can be drawn, in order to improve and harden the currently used forensic tools and techniques. Types of anti-forensic are shown in figure 2 and explain bellow.

Destroying evidences involves the destruction of evidence, in order to make it unusable during the investigative process. Although the destruction of evidence is often fatal, it is worth noticing that the tools, or the operations, used to destroy the evidence can produce evidence themselves in terms of traces of their usage. Actions taken to subvert the analyst, rather than a specific forensic analysis application, to decrease, or even nullify, the evidence visibility during the forensics analysis. The strength of this technique is strictly connected to the limitations of the people or, if any, of the used forensics tools. As for the previous item, the presence of any hiding tools can generate evidence.

It is the neutralization of the evidentiary sources; this technique does not concern the destruction but the prevention of evidence creation. It is the creation of a fake version of the evidence which is properly made to carry wrong or deviated information in order to divert the forensic process.

C. Android Anti-Forensic

From the previous research we have many useful anti-forensic techniques which are explain below.

1. Instantiating Anti-Forensic

In this anti-forensic technique we depict some possible instances and focusing on an MD equipped with Android and on the related main ideas that are behind the work being presented.

a. Exploiting android features

Android relies on the strong Linux processes and users management policies, which have been improved and hardened with the introduction of the Sandbox execution approach. By default, every application, identified as a different user, runs in an isolated safe area. Furthermore, the protection of application's files is ensured by the file permissions management. In such scenario, if a given application desires to

defend some files or directories, the protection is ensured and enforced at OS level.

b. Private folder

Due to the standard Android security features, for a given application it is possible to create, in the desired storage volume, a directory that is inaccessible for any other applications. Such directory, which can be defined as private, can be used to store any kind of information (e.g., text files, multimedia), it is created at install time and removed, including the entire content, when the owning application is uninstalled. Moreover, the private folder can be used to transfer any kind of compromising data (e.g., paedopornographic materials) without using removable volumes which can be easily investigated. Finally, it is worth noticing that the data stored by the private folder are not required to be encrypted; in such way, some issues related to cryptography (e.g., keys management) are avoided in favor of a kind of steganography which is ensured by the OS of the device.

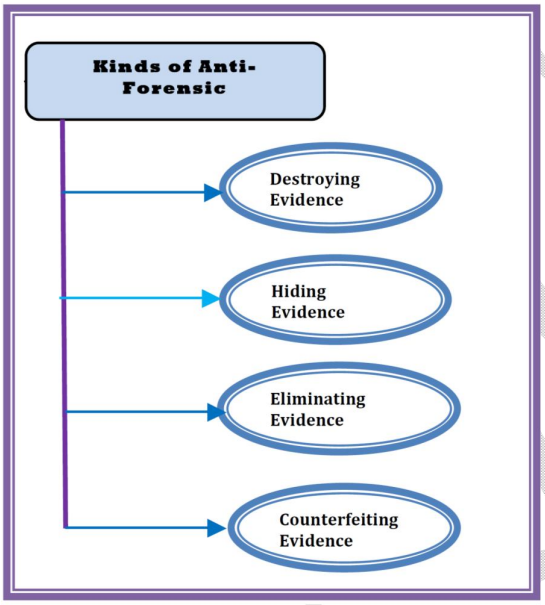

Fig.2 Types of Anti-Forensic

3. The Evidence Import Process (EIP)

The private folder mechanism is fundamental in order to protect the evidence; however it can be used as the pillar in order to reconstruct the evidence previously exported. In fact, the occulted evidence could be as precious as compromising; hence, the capability to restore the previous state of the device could be a valuable knowledge asset. The EIP is able to restore the previous state of device to reverse the EEP.

4. The evidence destruction process (EDP)

c. Anti-Forensic by AFDroid Application

All above techniques are implemented by a common android application which is called AFDroid that can be installed and executed onto the device.

At install time, AFDroid creates the private folder while, at execution time, it allows the execution of two distinct processes:

- The Evidence Export Process: it refers to the application of the AF techniques;
- The Evidence Import Process: it aims at reversing the Export Process.

2. The Evidence Export Process(EEP)

In this process the export file that is produced contains the evidence gathered by the target Android databases and it is stored by the private directory and erased by following manner.

- Android destroying evidence
- Android hiding evidence
- Android eliminating evidence
- Android counterfeiting evidence

When a given common Android application is uninstalled, the entire set of the related information, including data files, directories is logically deleted from File System. This also applies to any private folders and to the related content. Regarding the objective of the Evidence Destruction, it is interesting to investigate the possibility that a deleted data can be identified and, possibly, retrieved through some forensic techniques and tools currently available both in the literature and as COTS.

4. PROPOSED TECHNOLOGY

In this we proposed the techniques which are applicable for the santization or removal of the data from the private folder in the NAND flash memory. For this first we must know about how to encode the data in the flash memory. So the general concept behind a writing scheme is to flip each magnetic domain on the disk back and forth as much as possible (this is the basic idea behind degaussing) without writing the same pattern twice in a row, If data was encoded directly then, we could simply choose the desired overwrite pattern of ones and zeroes and write it repeatedly. But, disks generally use some form of run-length limited (RLL) encoding, therefore the adjacent ones won't be written. This

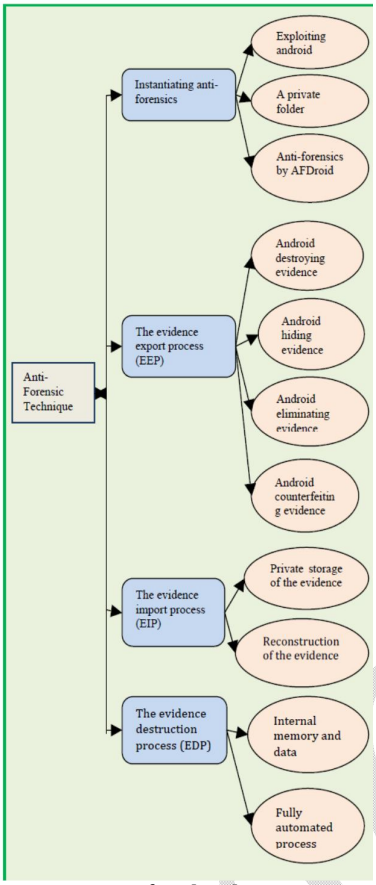

Fig.3 Types of android Anti-Forensic

Finally, the AFDroid application allows also a secure Evidence Destruction Process which is quick and straightforward. encoding is used to ensure that transitions aren't placed too closely together or too far apart, which would mean the drive would lose track of where it was in the data.

To erase magnetic media, we need to overwrite it many times with alternating patterns in order to expose it to a magnetic field oscillating fast enough that it does the desired flipping of the magnetic domains in a reasonable amount of time. The RLL encoding used in most current drives isdescribed by pairs of run-length limits (*d*, *k*), where *d* is the minimum number of 0 symbols which must occur between each 1 symbol in the encoded data, and *k* is the maximum. The parameters (*d*, *k*) are chosen to place adjacent 1's far enough apart to avoid problems with inter symbol intrusion. The different pass erase techniques used in creating the application was explained below

A. Zero Pass Erase Techniques:

1. *For total free space(either internal or SD card) start a for loop.*
2. *For each value in for loop create a file(of some defined size) "hemo" on directory(temoToDelete) which we have create in first step.*
3. *Write random byte on this file.*
4. *Write zero on this file.*
5. *Delete the files when value in for loop is equal to free space/10.*

B. Seven Pass Erase Techniques:

1. *Start a for loop of seven.*
2. *when value in for loop is equal to one*

a. *for total free space(either internal or SD card) start a for loop*
b. *for each value in for loop create a file(of some defined size)"hemo i" on directory(temoToDelete) which we have create in first step.*
c. *write random byte on this file d. write zero on this file*
e. *delete the files when value in for loop is equal to free space/10.*

3. *When value in for loop is equal to zero*

a. *for total free space(either internal or SD card) start a for loop.*
b. *for each value in for loop create a file(of some defined size) "hemo i" on directory(temoToDelete) which we have create in first step.*
c. *Write random byte on this file. d. Write one on this file.*
e. *Delete the files when value in for loop is equal to free space/10.*

4. *When value in for loop is greater than one*

a. *For total free space(either internal or SD card) start a for loop.*
b. *For each value in for loop create a file (of some defined size) "hemo i" on directory(temoToDelete) which we have create in first step.*
c. *Write random byte on this file.*
d. *Delete the files when value in for loop is equal to free space/10.*

C. 35-Pass Erase Techniques

a. *zero Start a for loop of 35*

b. *For each value in for loop create a file(of some defined size) "hemo i" directory(temoToDelete) which we have create in first step.*
c. *Write random byte on this file.*
d. *Delete the files when value in for loop is equal to free space/10.*

The procedure consists of calls to routines to maintain graphical entities, like lines, buttons, text, etc., and it is executed in one of three modes: Update, Show, and Erase. The top section shows the basic idea, that parameters of graphical entities are simultaneously written out to a sequential file (to be read on next pass), and read in (from the prior pass). The new and old values can be compared and used to update the graphical entities. The middle section shows how IF statements are handled.

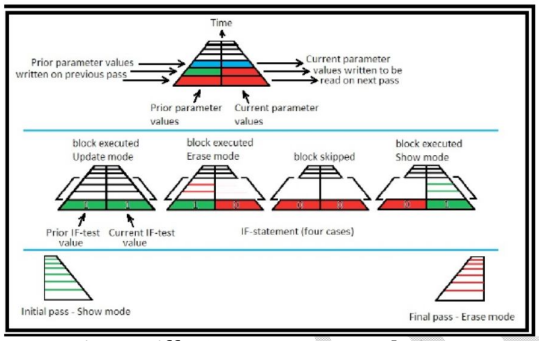

Fig.4: Different pass Erase techniques

Note that Show mode disables reading, and Erase mode disables writing.

5. CONCLUSIONS

In this paper, we explain the classification of the Anti- Forensics techniques and application of it .In past, the removal of data from the private folder in the NAND flash memory was done by only zero pass technique. This was not effective. We have proposed some new instances of such techniques to the mobile environment and, in particular, to Android mobile devices such as 3-pass,7-pass and 35-pass erasable techniques. The techniques we proposed were fully automated and supported by a common Android application, called AFDroid, that has been specifically designed and implemented and were much more effective to remove data from NAND flash memory. Finally, in order to test the effectiveness and strength of the implemented Anti-Forensic techniques, we planned and performed some experiments proving that AFDroid is able to hold on versus both the cursory examination of the device and some tools cannot able to acquire the data from the internal memory and external memory of Android devices.

REFERENCES

[1] Gartner Inc. and/or its Affiliates, "Gartner Analysts, "http://www.gartner.com/ newsroom/id/ 2645115(accessed January, 2014).

[2] R. Harris, "Arriving at an anti-forensics consensus: Examining how to define and control the anti-forensics problem," in *The 6th Annual Digital Forensic Research Workshop (DFRWS 2006)*, Aug. 2006. [Online]. Available: http://dfrws.org/2006/proceedings/6-Harris.pdf.

[3] A. Castiglione, G. Cattaneo, G. De Maio, and A. De Santis, "Automatic, Selective and Secure Deletion of Digital Evidence," *In: Proceedings of the Sixth International Conference on Broadband and Wireless Computing, Communication and Applications, BWCCA 2011*, IEEE Computer Society, Barcelona, Spain, October 26-28, 2011.

[4] D. W. Byunghee Lee, Kyungho Son and S. Kim, "Secure data deletion for usb flash memory," *Journal of Information Science and Engineering*, pp. 1710–1714, 2011.

[5] J. Lee, J. Heo, Y. Cho, J. Hong, and S. Y. Shin, "Secure deletion for nand flash file system," in *ACM Symposium on Applied Computing*, 2008, pp. 1710–1714.

[6] A. Distefano, G. Me, and F. Pace, "Android anti-forensics through a local paradigm," *Digital Investigation*, vol. 7, pp. S83–S94,Aug.2010.Available: http://dx.doi.org/10.1016/j.diin.2010.05.011

[7] C. Manning, "YAFFS: the NAND-specific flash file system – Introductory Article," http://www.yaffs.net/yaffs- nand-specific-flash-file-systemintroductory- article, 2011 (accessed May, 2011).

[8] M. Y. C. Wei, L. M. Grupp, F. E. Spada, and S. Swanson, "Reliably erasing data from flash-based solid state drives," in *FAST*, G. R. Ganger and J. Wilkes, Eds. USENIX, 2011, pp.

[9] J. Lessard and G. C. Kessler, "Android forensics: Simplifying cell phone examinations," *Small scale digital device forensics journal*, Aug. 2010.

[10] P. O'Brien, "Android @ modaco.com," http://android.modaco.com/, 2011 (accessed May, 2011).

[11] XDAdevelopers, "Android forum & windows phonediscussion,"http://forum.xdadevelopers.com /index.php, 2011 (accessed May, 2011).

[12] NIST, "NIST 800-88, Guidelines for Media Santization, "http://csrc.nist. gov/publications/ nistpubs/800-88/NISTSP800-88rev1.pdf, 2011(accessed May, 2011).

[13] MIUI Team, "MIUI Android ROM," http://www.miui.com, 2011 (accessed June, 2011).

[14] A. De Santis, A. Castiglione, G. Cattaneo, G. De Maio, and M. Ianulardo, "Automated Construction of a False Digital Alibi," in *ARES 2011*, A. M. Tjoa, G. Quirchmayr, I. You, and L. Xu, Eds., vol. 6908. Lecture Notes in Computer Science, Springer, 2011, pp. 359–373.

[15] P. Albano, A. Castiglione, G. Cattaneo, G. De Maio, and A. De Santis, "On the Construction of a False Digital Alibi on the Android OS," *Submitted*, July 2011.

[16] L. M. Grupp, A. M. Caulfield, J. Coburn, S. Swanson, E. Yaakobi, P. H. Siegel, and J. K.Wolf. Characterizing flash memory: Anomalies, observations and applications. In MICRO'09: Proceedings of ..., New York, NY, USA, 2009. ACM, IEEE.

[17] P. Gutmann. Secure deletion of data from magnetic and solid-state memory. In SSYM'96: Proceedings of the 6th conference on USENIX Security Symposium, Focusing on Applications of Cryptography, pages 8–8, Berkeley, CA, USA, 1996. USENIX Association.

[18] P. Gutmann. Data remanence in semiconductor devices. In SSYM'01: Proceedings of the 10th conference on USENIX Security Symposium, pages 4–4, Berkeley, CA, USA, 2001. USENIX Association.

[19] J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten. Lest we remember: cold-boot attacks on encryption keys. Commun. ACM, 52(5):91–98, 2009.

[20] J. Lee, J. Heo, Y. Cho, J. Hong, and S. Y. Shin. Secure deletion for nand flash file system. In SAC '08: Proceedings of the 2008 ACM symposium on Applied computing, pages 1710–1714, New York, NY, USA, 2008. ACM.

[21] LSoft Technologies Inc. Active@ KillDisk. http: //www.killdisk.com/.

[22] U. S. National Institute of Standards and Technology. Advanced Encryption Standard (AES) (FIPS PUB 197), November 2001.