2014

The impact of custom ROM backups on android external storage erasure

Haydon Hope
Edith Cowan University

Peter Hannay
Edith Cowan University

DOI: 10.4225/75/57b3e5dbfb880

12th Australian Digital Forensics Conference. Held on the 1-3 December, 2014 at Edith Cowan University, Joondalup Campus, Perth, Western Australia.

THE IMPACT OF CUSTOM ROM BACKUPS ON ANDROID EXTERNAL STORAGE ERASURE

Haydon Hope[1], Peter Hannay[2]
School of Computer and Security Science[1, 2] Security Research Institute[2]
Edith Cowan University[1,2] Perth, Australia
hphope@our.ecu.edu.au, p.hannay@ecu.edu.au

Abstract

The Android operating system is the current market leader on mobile devices such as smartphones and tablet computers. The core operating system is open source and has a number of developers creating variants of this operating system. These variants, often referred to as custom ROMs are available for a wide number of mobile devices. Custom ROMs provide a number of features, such as enhanced control over the operating system, variation in user interfaces and so on. The process of installing custom ROMs is often accomplished through the use of a ROM manager application. Such applications often provide mechanisms to back up the contents of the mobile device prior to upgrade. This mechanism is utilised in the case of a failed update to restore the device to its previous functional state. Backups produced in this manner are often stored in on an external media such as a micro-SD card.In the conducted research we evaluated devices inbuilt data erasure mechanisms within the context of erasure of backups produced by ROM managers. It was found that simply using the devices Format External / SD function is not an effective means of completely erasing these backups. Once recovered, these backups offer a quick source of information that a potential attacker could carve to retrieve user files such as media transferred to the external or from applications. Although the same files could be recovered from an image of the external storage itself, the carving process is more efficient than traditional carving methods.

Keywords
Android, Custom ROMS, Data Erasure

INTRODUCTION

Technically capable users of Android devices (including smart phones and tablets) are known to install custom Firmware (Agomuoh, 2013) on such devices to alter the system, gain extra control over the device and/or personalize their experience using the device (Singh, 2012). Through various mechanisms Android devices are interfaced with in order to allow users to easily reinstall and replace the firmware with other firmware via a ROM Manager. These mechanisms include the use of official Google tools such as fastboot or third party utilities such as Heimdall, the particular mechanism used is heavily dependent on the device being altered (Dobell, 2014; Levi, 2012).

Prior to the installation of a new Custom Read Only Memory (ROM) (a modified version of the standard Android Operating System) it is recommended that a system backup using a ROM Manager be taken and saved to the external storage of the device so if there is an issue installing the new Firmware, the backup can be loaded, restoring the device to its original state. In this restoration both the operating system and user data are restored.

The turnover rate of smart phones is quite high with over one billion smart phones being sold each year (IDC, 2014). A portion of these will be on-sold or removed from circulation as users decide to rid themselves of obsolete or unwanted devices. When selling or disposing of a device capable of storing data, it is critical that any sensitive data on the device be erased (Jones et al., 2010). This erasure can be achieved in a variety of ways. Smart Phones running the Android system usually have a function that allows the user to clear all the information on the device and return the device to its original factory state (including clearing any external memory / storage such as an inserted SD card), however, as revealed in (Sansurooah, Hope, Almutairi, Alnazawi, & Jiang, 2013) not all programs that claim to erase all information on a flash device will completely erase it.

Considering this, if a user activates the reformat external storage function and the restore to default function, then sells or destroys the device, how well is the information deleted and what data can be retrieved? Furthermore, if the user has installed a Custom ROM and used a backup manager from this ROM to create a backup onto the external storage, what impact will these backups have on the formatting and resetting process?

Significance of External Storage

The Android operating system divides its persistent data across internal (System) storage and external storage. External storage may be a removable object such as an SD card or a chip mounted within the phone. In the case of an internal NAND chip the phone makes use of emulation to make the NAND chip appear as an SD card.

Primarily, the external storage is used for the storing of public data that can be read, altered and saved by any application or service running on the device. The external storage can also be used as mass storage by connecting to a computer via a USB cable and enabling the appropriate function. The host computer is then able to mount the device as external storage and transfer files to/from the device through standard interfaces.

This external storage can be formatted into exFAT or any other FAT file system. FAT file systems (such as exFAT, FAT16 and FAT32) are more widely used file system than the Android YAFS / YAFS2 file systems used for internal purposes on Android devices. This wider use could potentially mean that the tools for recovering data from a storage device in this format are more numerous and stable, thus leading to an easier acquisition if any remnant data is found.

MATERIALS

Selected Device

For this experiment, a HTC Magic (HTC, 2009) is the device under examination. The HTC Magic was released in 2009 and the external storage is provided via a mini-USB, which allows users to upgrade their devices storage size easily. For forensic viability, the best way to acquire images of the device external storage would be to remove the micro-SD card from the device, insert the micro SD card into a micro-SD to USB reader, and then connect the reader to a write-blocker. This method was not used however as recent devices use internal, non-removable chips for their external storage (although the Android system may still recognize it as an SD card). The method used should be applicable to all Android devices; therefore images of the SD card were taken via physical acquisition with the device in USB Mass Storage mode.

Device Under Examination

Manufacturer: HTC.

Model: Magic.

Operating System: Android 2.2.1

Accepted External Storage Media: Micro-SD card.

SD Card: 1.86 GB.

Examination Platform

Operating System: Windows 7 32 bit.

CPU: 2.9 GHz Dual Core.

RAM: 4 GB.

Tools Used

Autopsy for Windows 3.0.9

A program for browsing image files, including .img, .dd or .raw. Autopsy also allows for the collection of deleted files that can still be recovered entirely (SleuthKit, 2013).

Forensic Tool Kit (FTK) Imager Version 3.1.3

Free software for creating images of the SD card at various stages of the experiment and calculating hash sums to confirm integrity of the images after each use (AccessData, 2013).

Sans SIFT workstation image Version 2.14

A distribution of the LINUX Ubuntu operating system to be run in a virtual environment, pre-loaded with some of the tools used in the experiment, including Scalpel file carver (SANS, 2011).

VMware Player Version 5.02

The program to be used for running the virtual workstations on the host system. The player has been set to allow the workstations access to one processor core and 1gb of RAM.

DATA SETS

Transfer Data

The external storage of an Android device can be used like a regular USB mass storage device to transfer files between the android device and a computer. For the purpose of testing a collection of files of different file type and size was transferred to the phone. This data set will show if the type of file being erased or the file size has any impact on the erasure efficiency of the Factor Default reset function.

Five unique files of each given format were used in order to create an adequate amount of data to be erased.

The file types used in this data set were:

.JPEG, .BMP, .PNG, .ZIP, .Doc, .PDF, .PPT, .SQL, .MP3, .MP4, .TXT Total: 55 files, 65Mb (full details of this data set including file names and sizes can be found in Appendix A).

Saved Data

Locally generated data was also added to the SD card using the device itself. Five photos were taken using the devices single camera and five audio clips were recorded using the "Sound Recorder" application (a default app on this device). This saved data resulted in five .jpeg files and five .3gpps files being added to the device (full details available in Appendix A).

METHODOLOGY

The methodology used in this analysis was similar to those carried about by Sansurooah et al in 2013 as these two experiments share similar goals (to test the erasure efficiency of a particular tool on a form of storage media), although they are still different.

Preparation

1. Connect device to computer.
2. Enable USB mass storage on the device.
3. Acquire an image of the SD card

Population

1. Disconnect Device from Computer.
2. Take 5 photos with the camera, check where they are saved.
3. Record 5 pieces of audio, check where they are saved.
4. Reconnect Device to computer.
5. Enable USB mass storage on device.
6. Transfer the transfer data set to the device.

Pre-Backup Image

1. Create image of the USB device with FTK.
2. Disable USB mass storage and disconnect device.

Creating The Backup

1. Using ClockworkMods Custom Rom Manager, create a backup of the device.
2. Reconnect to computer and activate USB mass storage.

3. Create image of the USB device with FTK.
4. Disable USB storage and disconnect.

Erasure

1. On the device, select settings.
2. Select SD card and phone storage.
3. Select Un-Mount SD card.
4. Select Format SD card.
5. Confirm Formatting.
6. Remount SD once formatting is complete.

Recovery

1. Reconnect to computer
2. Enable USB mass storage
3. Create Image of formatted SD
4. Use Autopsy to browse the image of the formatted SD card and extract any files that can be found.
5. Use the Scalpel file carving program on the image of the formatted SD card and on any images that can be extracted via Autopsy to find any files.

PRELIMINARY INVESTIGATION

Custom ROM

The custom ROM used on the device under examination is CynogenMod 6.1.0-RC1-D5. The CynogenMod was "designed to increase performance and reliability over Android-based ROMs released by vendors and carriers such as Google, T-Mobile, HTC, etc." (Cyanogen, 2012). CyanogenMod was selected due to its popularity as the most common custom Android ROM.

Cynogenmod also makes the following changes to the base Android Operating System and the unique functions added by the device creator or the functions added by carrier:

- Allows users to remove apps and functions added by the device creator or the service provider to free-up space and remove unneeded functions.
- Get access to the latest version of Android supplied by Google as some service providers can take months to release the latest version modified to suit their needs.
- Allow users to add extra features, and push custom apps to the device.

Layout of the SD Card

Before carrying out the Erasure and Recovery processes, it is important to fully understand the layout of the SD card. Browsing the SD card via USB transfer between the device and the system reveals the following files and folders:

- Android/data: this folder is used to hold any files that are to be used by any application. This folder also contains com.cooliris.media. CoolIris is an application used by the Android system to display photos in a gallery.
- .andoid_secure is also found on external storage. This folder is normally used to store application data when the application is moved to the external storage / SD card via the devices "Move to SD Card" function in "Application Management".
- DCIM/Camera is where all photographs taken using the devices camera are stored. The file names of these photos can be used to show when they were taken. For example, the filename IMG_20140402_082713 shows that the photo was taken at 08:27:13 am on the second of April, 2014.

All the data from the transferred data set and all the recordings made with the devices sound recorder app are stored at the root folder of the SD card.

The ClockwordMod backups are located in /clockworkmod/backup/2014-04-05-13.44.00. This shows the date and time the backup was taken. However, upon formatting the device, this folder is deleted and the images are found in the /$OrphanFiles folder.

RECOVERY RESULT

Using Autopsy Forensic Browser to inspect the image of the formatted USB, none of the test data was found in a deleted state or otherwise where it had been before formatting. However inspection of the /$OrphanFiles folder did reveal some deleted files and folders that could be extracted.

Several of these deleted files were .IMG files, images of the Android devices system created by the ClockworkMod ROM Manager Recovery application. The .IMG files are shown in Table 1 below.

Table -1 - A listing of backup files created by the ClockworkMod ROM Manager Recovery application

Name	**Size**
BOOT.IMG	2 560 kb / 2.5mb
CACHE.IMG	33 349 kb / 32.5mb
DATA.IMG	7 282 kb /7.11mb
RECOVERY.IMG	5 120 kb / 5mb
SYSTEM.IMG	88 588 kb / 86.4mb

Each of the above .IMG files is a backup of the corresponding partition in the Android operating system, therefore these partitions are /boot, /cache, /data, /recovery and /system. Each of these partitions is used for different functions in the operating system.

As indicated by the name, /boot is used by the system to boot, containing all the files necessary for the device to start. This partition usually includes the kernel and the ramdisk. If this partition was missing, the device will not start.

The /cache partition is used by Android to store commonly accessed data and components. Wiping the cache will not affect the device or any personal data on the device. Wiping the cache can help to improve the devices operating speed.

The /data directory stores the user's information, including contacts, messages, settings and applications. Wiping this partition deletes all of this information, practically carrying out a Factory Reset. The device used in this examination was free of user information such as contacts or emails and so none would be retrievable. However, as contacts and such are stored in SQLite databases and .sql files could not be retrieved (as shown below); it is possible even if this information was available on the device they could have been completely erased during the formatting of the device.

The /recovery partition can be considered "an alternative boot partition" (Raja, 2011). The recovery partition allows the device to be started up into recovery, this recovery console allows for the original operating system to be reinstalled.

Finally, the /system partition. This partition contains the operating system (apart from the kernel and ramdisk, found in /boot). The /system partition usually includes the applications that come pre-installed on the device, including the custom ROMS unique applications such as SuperUser. Wiping this partition alone will remove the operating system from the device but still allow the device to be started in Recovery Mode for the installing of a new ROM. With each of these partitions being described, a quick preliminary investigation was undertaken to find if any of the files were still available on the images that were recovered from the formatted card.

This preliminary investigation involved simple string search in Autopsy for ".jpeg", the file extension for some of the picture files from the test data set. This search revealed the string "dalek.jpeg" in the hex values of SYSTEM.IMG and CACHE.IMG. Dalek.jpeg being the file name of an image in the transferred data set, the file itself could still be in the SYSTEM and CACHE files.

To investigate the possibility of files being on the various .IMG files, the .IMG files were extracted and saved. As these images were made by the ClockworkMod ROM Manager Recovery program, these files would be in the Android file system and would not be recognized by Windows. Therefore, mounting, then browsing the images in Windows Explorer would not be possible. In this instance it was determined that file-carving as an approach would provide means to extract these files. File-carving involves matching headers and footers within the data being analysed and extracting these in an attempt to form complete files.

To achieve the file carving, the .IMG files were copied and the working copies then dragged into a virtual machine of SANS Sift Forensic Workstation. Once the files were loaded into the virtual environment, Scalpel

was used on each image to extract all possible files. Resulting extracted data was saved to a folder unique to the image being examined (such as SYSTEM.IMG outputting to system_scalp). Once all the .IMG's were carved, the output folders were copied back to the host operating system to be examined. The image of the formatted SD card that the .IMG files were extracted from was also carved and saved to SD_scalp.

CARVING RESULTS

The resulting folders from the file carving process are detailed in Table 2 bellow.

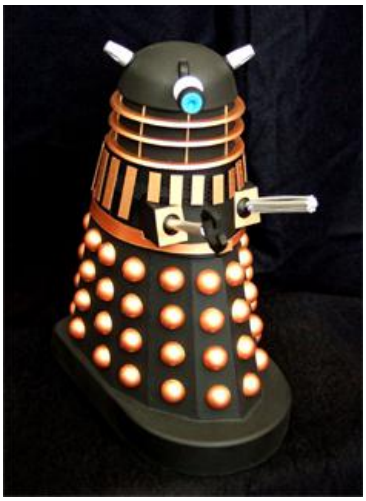
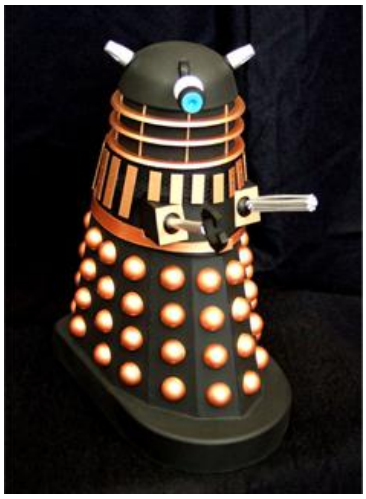
*Table **Error! No text of specified style in document.**2 - A table showing items recovered through the carving process*

Input	Input Size	Output Folder	Contents of Folder
BOOT.IMG	2.5mb	boot_scalp	30.7mb
CACHE.IMG	32.5mb	cache_scalp	787mb
DATA.IMG	7.11mb	data_scalp	110mb
RECOVERY.IMG	5mb	recovery_scalp	40.7mb
SYSTEM.IMG	86.4mb	system_scalp	3450.88mb
Sd_formatted.dd	1904.64mb	Sd_scalp	19968mb

This difference in size between the original files and the resulting Scalpel carve output shows that there is potentially high numbers of false positives and multiples of the same files. The result of browsing the output folders shows that some of the files from the test data sets were retrieved. This is shown below.

Some of the files that were retrieved were complete files, where all the detail of the original file is visible. Others were partial, not showing all the detail of the original but still being identifiable. The following table shows the difference between a "completely" retrieved file and a "partially" retrieved file.

Table 3 - An example of a recovered file showing original, complete recovery and partial recovery states

	Original	Complete	Partial
File			
Name	Dalek(1).jpeg	00000006.jpeg	00000053.jpeg
Carved From		SYSTEM.IMG	SYSTEM.IMG

Although the dimensions and appearance of the partial have changed, it is still identifiable as Dalek(1).jpeg from the test data set. Both these images were carved from SYSTEM.IMG, this shows Scalpel carving false positives. Some files were recoverable from more than one of the backup images. The above example of dalek(1).jpeg was also found complete in CACHE.IMG and in the sd_formatted.dd image.

An interesting finding from examination of all the output files is the distribution of retrievable file types across the different images made by the backup. The following figure shows this distribution.

Table 4 - A listing of the file types recovered in partial and complete states

Image File Carved	Complete File Types Recovered	Partial File Types Recovered
BOOT.IMG	PDF, GIF, MP3	GIF
CACHE.IMG	PDF, JPEG, ZIP, DOC, PPT	JPEG
DATA.IMG		
RECOVERY.IMG	PDF	
SYSTEM.IMG	PDF, BMP, JPEG, ZIP, DOC, PPT	BMP, JPEG, PNG
SD_formatted.dd	PDF, MP3, JPEG, JPEG, GIF, BMP, DOC, PPT	MP3, JPEG

From the above distribution table it can be seen that nothing was retrievable from the DATA.IMG file, this could be because the /data partition is only for users phone data (such as contacts) and not media such as files from the data set. Interestingly, none of the text files (.txt and .sql) could be retrieved via Scalpel and neither could any of the mp4 video files or 3ggp files (which is the formatted used to store sound recordings captured by the Android device). However, some picture files used by the system, such as backgrounds, could be retrieved. Furthermore, many .RPM files were recovered from all the images.

RPM files are "Red Hat Package Manager" files. These files are used in Linux for management of packages (including apps and other files), therefore, these extracted RPM files are probably used by the Android operating system.

The following figures show how many of the data set files could be retrieved from the backup images.

Table 5 - A table showing the number of files recovered of each type

File Type	Complete Retrieved	Partial Retrieved
JPEG (Transferred)	5/5	5/5
JPEG (Saved)	2/5	4/5
GIF	2/5	5/5
BMP	4/5	3/5
PDF	3/5	0/5
DOC	5/5	0/5
ZIP	2/5	0/5
MP3	1/5	0/5
PPT	4/5	0/5
PNG	0/5	5/5

Table 5 above demonstrates that 28 complete and 22 partial files from the test set were recoverable from the extracted, previously deleted, backups. Considering the complete data set was 65 files, 43% were totally retrievable, therefore the Android devices "Reformat SD Card" function can be said to be 57% efficient at erasing a backup created with a Custom ROM backup application such as ClockworkMod Recovery.

An interesting result from this experiment and examination of retrievable files is that only one sound file from the dataset (5 mp3s from USB transfer and 5 .3gpp files from saved recording made with the Sound Recorder application on the device) could be retrieved. Also, no video files could be retrieved in a playable state

CONCLUSION

Browsing the .dd file of the formatted SD card in Autopsy revealed that none of the files from the test set could be retrieved from where they had been prior to formatting. However, if a backup is taken using a Custom ROMS backup function, the backups can still be retrieved.

The image of the SD card and the backups can contain information which can be retrieved via a simple carving process. This process is notably longer when carving the entire SD card but takes mere seconds to carve each of the five image files extracted via Autopsy. Therefore, carving the backups is a quicker and just as way to extract data from an Android devices external storage.

It is simple for media such as transferred files to be recovered from the device. However, the recovered .IMG backups are too damaged to be used as a backup or to be browsed via Android. Therefore the users other information (such as contacts, applications and so on) cannot be recovered.

Considering that the MP4 video files were some of the largest files in the data set and that txts and .3ggps were the smallest, this indicates that file size may not impact on the formatting processes ability to delete the backups completely

RECOMMENDATION

From this experiment, it was revealed that formatting an Android devices external storage is not enough to securely erase all the information on the device.

The experiment also revealed that backups taken with a Custom ROMS recovery / backup app (in this case CyanogenMod with ClockworkMod Recovery) can easily be extracted after formatting and can be used as a quick way to carve out system information, rather than carving an entire image of the formatted storage.

The best recommendation for a user seeking to securely erase their Android device if it has a Custom ROM and has used a backup application would be to wipe their external storage with a program such as CCleaner erasure function (as recommended by (Sansurooah et al., 2013)), rather than relying on the Android devices "Format External / SD" command.

REFERENCES

AccessData. (2013). Forensic Toolkit Imager. Retrieved March, 2014, from http://accessdata.com/support/adownloads - FTKImager

Agomuoh, F. (2013). CynogenMod Download Stats: Custom ROM Installed On Over 10 Million Devices. Retrieved April, 2014, from http://www.ibtimes.com/cyanogenmod-download-stats-custom-rom-installed-over-10-million-devices-1519062

Cyanogen. (2012). Android Community Operating System. Retrieved March, 2014, from http://www.cyanogenmod.org/

Dobell, B. (2014). Heimdall. Retrieved May, 2014, from http://glassechidna.com.au/heimdall/

HTC. (2009). HTC Magic – Full Phone Specifications. Retrieved March, 2014, from http://www.htc.com/www/support/htc-magic/

IDC. (2014). Smartphone Vendor Market Share, Q2 2014. Retrieved June, 2014, from http://www.idc.com/prodserv/smartphone-market-share.jsp

Jones, A., Valli, C., Dardick, G. S., Sutherland, I., Dabibi, G., & Davies, G. (2010). The 2009 analysis of information remaining on disks offered for sale on the second hand market.

Levi, J. (2012). Android Power User: How to Flash an Image with Fastboot. Retrieved June, 2014, from http://pocketnow.com/2012/10/24/flashing-with-fastboot

Raja, H. (2011). Android Partitions Explained: boot, system, recovery, data, cache & misc. Retrieved March, 2014, from http://www.addictivetips.com/mobile/android-partitions-explained-boot-system-recovery-data-cache-misc/

SANS. (2011). SANS Sift Virtual Workstation. Retrieved March, 2014, from http://digital-forensics.sans.org/community/downloads

Sansurooah, K., Hope, H., Almutairi, H., Alnazawi, F., & Jiang, Y. (2013). *An Investigation Into The Efficiency Of Forensic Data Erasure Tools For Removable Usb Flash Memory Storage Devices*.

Singh, A. (2012). Android: Stock ROM vs. Custom ROM. Retrieved March, 2014, from http://www.ahemahem.com/android-stock-rom-custom-rom/295/

SleuthKit. (2013). Autopsy Forensic Browser. Retrieved March, 2014, from http://www.sleuthkit.org/autopsy/