Watch what you wear: preliminary forensic analysis of smart watches

Ibrahim Baggili, Jeff Oduru, Kyle Anthony, Frank Breitinger, Glenn McGee
University of New Haven : Department of Electrical & Computer Engineering and Computer Science
University of New Haven Cyber Forensics Research and Education Group (UNHcFREG)
West Haven, CT
ibaggili@newhaven.edu

***Abstract*—This work presents preliminary forensic analysis of two popular smart watches, the Samsung Gear 2 Neo and LG G. These wearable computing devices have the form factor of watches and sync with smart phones to display notifications, track footsteps and record voice messages. We posit that as smart watches are adopted by more users, the potential for them becoming a haven for digital evidence will increase thus providing utility for this preliminary work. In our work, we examined the forensic artifacts that are left on a Samsung Galaxy S4 Active phone that was used to sync with the Samsung Gear 2 Neo watch and the LG G watch. We further outline a methodology for physically acquiring data from the watches after gaining root access to them. Our results show that we can recover a swath of digital evidence directly form the watches when compared to the data on the phone that is synced with the watches. Furthermore, to root the LG G watch, the watch has to be reset to its factory settings which is alarming because the process may delete data of forensic relevance. Although this method is forensically intrusive, it may be used for acquiring data from already rooted LG watches. It is our observation that the data at the core of the functionality of at least the two tested smart watches, messages, health and fitness data, e-mails, contacts, events and notifications are accessible directly from the acquired images of the watches, which affirms our claim that the forensic value of evidence from smart watches is worthy of further study and should be investigated both at a high level and with greater specificity and granularity.**

***Keywords*-Smartwatch; Samsung Gear 2 Neo smart watch; LG G smart watch; Wearable devices; Mobile forensics; Smart watch forensics; Android forensics; Tizen forensics**

I. INTRODUCTION

Engineers, scientists, clinicians and consumers have pursued wearable technology for decades. Once a fixture of comic book detective stories and science fiction movies, so called smart watches have inspired innovations to reduce size, increase speed and add resilience. Inventors filed patents for sophisticated wearable technology in watch form in the early 2000s [1]. Analysts are still undecided about the future of specific wearable devices though many see the potential [2]. The present set of products marketed to consumers includes smart watches framed as devices to track and augment daily activities and interact with other devices, applications and lifestyle choices. Collectively, industry activity in the domain was expected to be $2.5 billion in 2014 [3]. Large-scale participants currently shipping smart watches include Samsung, LG, and Motorola. Apples smart watch, the iWatch, which only started recently shipping in April of 2015 is expected to figure heavily in the market [4]. In all these implementations of wearable technology there is significant acquisition, handling and display of data that may be of relevant forensic value.

At the time of writing this paper, most smart watches were designed to compliment smart phones and could interact with several of the applications on them. For example, phone calls one receives on a smart phone connected to a smart watch would show a notification on the watch and in some cases calls may be answered directly from the watch. Text messages received are also relayed to the watch and in some cases can be responded to from the watch. Developers are also working on applications for home automation that will integrate smart watch technology [5]. In these cases, data flows from the smart phone to the smart watch, where the latter acts as a 'smart phone extender'. In many other situations, smart watches will also create, process and transmit data whose primary utility is not tied to any particular smart phone, such as health and fitness data, in which case the smart phone's role and the form of the data and communication of data may be different.

Although there have been several studies conducted on mobile phone forensic analysis, we were unable to identify any articles in peer-reviewed literature on the forensic acquisition and analysis of smart watches. In fact, the only discussion on smart watch forensics we were able to ascertain is a Blog entry[1] and a presentation by J. Kim [6]. In the Blog entry, the author examined the first generation Samsung Gear smart watch and concluded that despite of the watchs ability to display information like contacts, messages and e-mails, it never actually saved any of that data on the watch because an `adb pull`-command could not retrieve the data. In his presentation however, J. Kim was able to physically image an older version of the Galaxy Neo used in our work, but since that presentation, both the partitions as well as the operating system of the Galaxy Neo changed in the second generation. The operating system also changed from Android Wear to Tizen.

In this study we sought to identify evidence of potential forensic value that may be obtained from smart watches and/or as a result of the data accumulated by these devices that is stored elsewhere. To that end, we examined two

[1] http://selil.com/archives/4700 (last accessed May 10th, 2015).

representative smart watches and attempted to identify and breach their security barriers for the purpose of learning how these smart watches and the phones they sync with store, manage and share users data. We present the data we were able to find on the phone that syncs with the watches as well as the techniques used for physically acquiring the contents of both the Samsung Gear 2 Neo as well as the LG G watches. We further present a list of the summary of potential digital artifacts that we were able to extract from the watches, and comment on the intrusiveness of the methods used when acquiring an image from the watches – especially for the LG G watch. We note that our paper is extremely detailed in terms of the procedures followed to inform both scientists and practitioners should the need for the forensic acquisition and analysis of a smart watch arise in future investigations.

II. Background & related work

It is no secret that the use of mobile phones has become prevalent in our society. About 90% of adults in the United States have mobile phones of which 58% are smart phones [7]. Smart watches are a new emerging product category. Smart watches have been subject to little forensic study because of their infancy. However, there have been a number of forensic research projects conducted on other small scale digital and non-traditional devices. They vary from research on the forensics of digital camcorders [8], to video game systems like the Xbox One [9].

Some of the prior work performed on wearable technology includes a case study [10]. Their study focused on smart watches and some of the security threats they pose. They developed an application that can be used dishonestly in academic environments. The researchers called their application ConTest and it was built for the pebble smart watch. Students can utilize it to collaborate on a multiple-choice exam in real time. Their study results indicated that dishonest students could successfully cheat on multiple-choice exams with the same effectiveness as having access to a smart phone.

Since smart phones have a symbiotic relationship with smart watches, and most smart watches cannot operate independently, research on mobile forensics is of major relevance to our work.

Researchers conducted an investigative study which examined the extent that data is stored on phones connected to cloud services and whether users should be concerned about their data being accessible on their smart phones even after being synced to the cloud [11]. Other work analyzed different sources of forensic data on a mobile device [12], while further research identified areas where digital mobile forensic guidelines and standards could be improved [13].

Auxiliary studies however focused their attention to specific mobile operating systems, primarily on the two most popular ones; iOS and Android. There has also been work that examined the logical backup of the iPhone 3GS and its forensic value [14]. In their work, they discovered valuable evidentiary data, which included messages, web browsing history, facebook friends, calendar entries and much more. Alternatively, researchers proposed an operational technique that allows digital forensic practitioners to recover deleted image files by referring to the iOS journaling file system [15]. Furthermore, they successfully recovered a deleted image file using this technique. Other work explored traces of data left behind by the Apple wireless printing feature on its iOS devices, 'AirPrint' [16]. Finally, researchers have also examined the opposite end of the spectrum and discussed different anti-forensics methods for acquiring useful data from iOS devices [17].

The Android OS has also been extensively studied. Published work illustrated a simple and forensically sound manner for examining Android devices [18]. Research by Vidas et al. introduced a general methodology for collecting evidence from Android devices [19]. Research in [20] introduced a concept called Live SD and discussed performing physical image acquisitions from Android devices without installing any forensic software. Some studies have emphasized on specific Android handsets like [21], where they focused on the HTC One smart phone. Several anti-forensic studies have also been conducted on Android devices; this includes work by [22] who proposed a new anti-forensics technique for mobile devices running Android. The technique makes it possible to modify and securely erase digital evidence on an Android device without making any file system changes. Prior work also illustrated that data from Orweb on Android which intends to anonymize browsing data may be recovered after rooting the device [23]. Lastly, past work also analyzed the capability of operating system modifications from an anti-forensics standpoint [24].

The popularity of tablets has also grown in the past several years and they as well have been subject to forensic studies. Work has been published on the forensics of Apple's iPad [25]. Furthermore, prior work also analyzed a logical backup of the Blackberry Playbook tablet which analyzed the generated `.bbb` backup file to create a profile of the device's usage pattern and blackberry phones that were paired with it [26]. Lastly, work in [27] exemplified two techniques for acquiring data from the Amazon Kindle Fire HD. To date, the only information available to digital forensic examiners on smart watch forensics is the aforementioned blog entry and the presentation in [6]. Additionally, of interest to the forensics community is leveraging the work of developers in the hacking and modding community to gain root access to the smart watches, which we had to utilize in our work [28], [29].

As noted previously, there has been no published peer-reviewed work on smart watch forensics. In the following sections we outline the details of what data we were able to recover from the phone that syncs up with the smart watches, as well as techniques for imaging the actual watches. Our goal is to add new knowledge to the domain and make the methods and tools available to the digital forensics community should a practitioner or researcher need access to this information.

III. Examining data on the phone

In this part of the paper, we discuss the data found on the Samsung Galaxy S4 phone that was used for syncing with the watches.

A. Methodology

Our study embodies two overarching parts. The first part of our study focused on data that could possibly be acquired from the phone. Embracing the guidelines for forensically examining artifacts as recommended by National Institute of Standards and Technology (NIST) [30], we carefully conducted the usage scenario, documentation, forensic acquisition process, and analysis of the smart watches and the smart phone synced with them. Presented here are: 1) the tools used in this research, followed by 2) the documented usage activities taken with each watch, then 3) the steps taken to image each watch. We also provide a summary of the results obtained from imaging the watches. In every section, we attempt to be as exhaustive as possible in order for our work to be easily replicated should these methods be used in an investigation.

B. Apparatus

Table I lists the apparatus used in this project accompanied by their functions.

Table I
Apparatus

Tools	Function
Samsung Galaxy S4 Active smart phone	Phone paired with both Watches
Cellebrite UFED 4 PC	Used to acquire a physical image from the phone.
.XRY Logical	Used to acquire logical images from both smart phones for redundancy and comparison reasons
Cellebrite Case with Accessories	Cellebrite kit containing additional required tools like the hardware USB license
Cellebrite Physical Analyzer	Used to analyze the physical image.
LG G watch	One of the watches used in the experiment.
Samsung Gear 2 Neo watch	One of the watches used in the experiment.
Autopsy	Open source forensic image analyzer.

C. Scenario and Documentation

The Samsung Gear 2 Neo watch was paired with the Samsung Galaxy S4 Android Active smart phone via Bluetooth. The LG G smart watch was also paired with the Samsung Galaxy S4 Active also via Bluetooth. The application Samsung Gear Manager was then downloaded on the Galaxy S4 Active smart phone. This application is needed for data such as e-mail, text messages and health data to be synchronized back and forth between the watch and the smart phone. The Android Wear application was also downloaded and installed via the Google Play Store on the S4 Active phone to serve a similar purpose. E-mails, text messages, etc. received on the phone will also be displayed on the watch. The following usage scenario was then followed on both devices:

- Set a calendar event called 'Lunch with Sandra' on the phone.
- Sent an e-mail containing the subject line 'Meeting' and body 'Will take place today at 6' on the phone.
- Tracked footsteps.
- Used Google Now on the Samsung Gear to ask 'How is the weather today?'.
- Used the voice recorder built into the Samsung Gear 2 Neo.
- Checked heart rate with the Samsung Gear 2 Neo.

D. Phone imaging

Physical forensic images were acquired from the Samsung Galaxy S4 smart phone using the Cellebrite UFED 4PC (4.0.0.220) tool. The Samsung Galaxy S4 Active was put in 'download mode' in order for it to connect to Cellebrite. The phone was put into download mode by holding down the power button, home button, and the volume down button simultaneously until the Android logo was displayed on the screen and a message appeared for the user to push the volume up button. SHA256 hashes were calculated by the application before and after imaging to ensure that the data on the device was not altered during the imaging process. The two hash values matched and the forensic image was complete and ready to be analyzed using the accompanying Cellebrite Physical Analyzer (4.0.0277). A logical image was also acquired for comparative purposes using .XRY and the image file was added to Autopsy for analysis.

E. Data examination with Cellebrite Physical Analyzer

After the images were acquired, Cellebrite Physical Analyzer was used to examine the data that was found on the Samsung Galaxy S4 smart phone. When an image file was imported into Cellebrite Physical Analyzer, it parsed the file and created a tab called 'Extraction Summary'. This window displayed the important categories of data parsed from the image file including calendar, call log, contacts, email, etc. A keyword search was also performed on the directory tree for further analysis.

F. Phone examination results

Following are the results after the phone's forensic image was acquired and examined. The results varied between the Samsung Gear 2 Neo and the LG G Smart Watch.

1) Samsung Gear 2 Neo: Preliminary examination of the data found on the forensic image showed several useful forensic artifacts. The primary focus however was to search for possible evidence left behind from syncing with the watch, namely the voice memo, the footsteps tracked by the pedometer and the heart rate data which was all synced back to the Samsung Galaxy S4 smart phone.

In the partition `userdata`, there was the directory `/Root/data/com.samsung.android.app.`

watchmanager that was found to contain information pertaining to the Galaxy Gear Neo 2 watch. This folder contained a file called auto_update.xml which contained a timestamp of the day the Samsung Gear was last updated.

Analysis of the file /Root/data/com.samsung.android.app.watchmanagerstub revealed further information that tied the Samsung Gear 2 Neo to the Samsung Galaxy S4 Active smart phone. Listing 1 shows a file under 'shared preferences' in this directory called hmonlinehelppref.xml that identifies the kind of smart watch that was paired with the phone. The model number of the watch also appears in a separate file in this directory as shown in Listing 2. The same directory contained a sub directory named 'files'. The system photos and icons used in the design of the phone can be found in that directory. While setting up scenarios for the Samsung Gear 2 Neo, a voice message was recorded on the watch which was easily recovered from the phone.

```
<?xml version='1.0' encoding='utf-8' standalone='
    yes'?>
<map>
  <string name=''KindOfGear''>Gear 2 Neo</string
      >
</map>
```

Listing 1. Content of the hmonlinehelppref.xml file.

```
<?xml version='1.0' encoding='utf-8' standalone='
    yes'?>
<map>
  <string name=''key_help_gear_model''>SM-R381</
      string>
</map>
```

Listing 2. Model number found.

The Samsung Gear Manager application is required before the Samsung Gear 2 Neo watch can be used with a smart phone. Because part of the setup process is downloading this application, knowing when it was downloaded is a good indication of when the watch was first used. A timestamp of when the Galaxy Gear Manager application was first downloaded onto the S4 Galaxy Active phone was also found in the /Root/data/com.samsung.android.app.watchmanagerstub directory as shown in Fig. 1. This can be confirmed as the time the application was downloaded because it matched the documented time we kept as a record during the initial setup of the watch.

```
................s E..c...=E..c....
.HTTP/1.1 302 Moved Temporarily.Se
rver: AkamaiGHost.Content-Length: 0
Location: /downapps/com.samsung.an
roid.app.watchmanager/2114100299/c
mmon/en-us/.Cache-Control: max-age
21600.Date: Fri, 10 Oct 2014 20:55
51 GMT.....cXRI..5._s...I.W....67.
18.89.203...P.....................
```

Figure 1. Timestamp found in /Root/data/....

2) LG G Watch: The LG G Watch required the Android Wear application to be installed on the S4 Active. The latest Android Wear, version 1.0.4.1611133 was installed on the S4 Active phone. The primary objective while examining the image file for evidence of the LG G Watch was to look for a file that had to do with the pedometer on the device. No directory names corresponded to pedometer or health in the files analyzed. Unlike the Samsung Gear 2 Neo, the majority of the files that directly related to the watch manager were not preliminarily discovered during the analysis. There was a directory titled com.google.android.wearable.app-2 and an APK file called com.google.androidweable.app-2.apk which appeared to be related to Android Wear but the directory was empty so no further clues were obtained from it.

There were two databases also located in the same directory root/data/com.google.android.wearable.app/databases. The first file, devices.db which contained the list list of devices using Android wear which listed the LG G Watch. The second database file devices.db-journaled.db which did not contain any forensically interesting data when analyzed.

IV. Smart watch forensics

After researching a variety of hacker and modder online communities, we were able to formulate a methodology that examiners may be able to replicate to acquire a forensic image from the watches. In the following sections, we outline the methodology that may be used to physically acquire evidence from the Samsung Gear 2 Neo and the LG G watches. In both cases, putting the devices in debug mode was necessary and the Android Debug Bridge (ADB) or Smart Development Bridge (SDB) had to be utilized. We then present an overall summary of the artifacts we were able to recover from the watches in their respective sections. All the tools that are needed for rooting / imaging the watches, as well as the images that were acquired from the watches can be downloaded from this link **http://www.unhcfreg.com** under Data & Tools.

A. Samsung Gear 2 Neo

The Samsung Gear 2 Neo watch uses the Tizen operating system which is based on the Linux Kernel. The watch has no WiFi connection, and only has Bluetooth capability to connect to the phone it syncs with. Therefore, imaging the watch has to be performed over the USB connection when its connected to a workstation. However, the first step for being able to forensically image the watch was to gain root access.

1) Gaining root access: The first step for gaining root access was to install all the necessary drivers for the workstation to be able to communicate with the watch. In order to gain root access, the ROM of the device must be flashed. A tool, Odin 3.0 was used to flash the ROM along with an .img file that was obtained from the modder community. Before flashing the image onto the watch, the

watch must first be put into download mode. This was executed by continuing to hold down the main button on the watch through the turn off prompt. The watch then started to reboot and during this reboot process the watch button was pushed 3-4 times. This showed a new prompt that allowed us to boot into download mode. On the menu, the button was pressed once to change the selection and then the button was held down to select the download option. After the watch was placed in download mode, Odin 3.0 was launched. Then, the watch was connected to the workstation via USB, and we ensured that it was recognized by the workstation it was connected to. Then on Odin 3.0, "AP" was selected and the .img file that was chosen to be flashed to the device. Start was then clicked and and we waited until the process was complete. The Gear 2 Neo required SDB, not ADB to make connections to the device. This file can be downloaded from the Tizen-SDK but is also included in the our download bundle available on our website **http://www.unhcfreg.com** under Data & Tools. Once the rooting process was complete, the commands `sdb root on` followed by `sdb shell` were executed on the workstation and we were able to see that we had root access to the device.

2) Imaging method: The watch's memory is broken up into a series of partitions (1-15). Each partition's purpose was determined through the rooted shell as shown in Listing 3.

```
/*Start a shell session*/
sdb.exe shell
/* Navigate to this directory */
cd dev/disk/by-partlabel
/* Show files in the directory */
ls -al

drwxr-xr-x 2 root root 340 date .
drwxr-xr-x 8 root root 160 date ..
lrwxrwxrwx 1 root root date boot -> ../../
    mmcblk0p5
lrwxrwxrwx 1 root root date bota0 -> ../../
    mmcblk0p1
lrwxrwxrwx 1 root root date bota1 -> ../../
    mmcblk0p2
lrwxrwxrwx 1 root root date csa -> ../../
    mmcblk0p3
lrwxrwxrwx 1 root root date csc -> ../../
    mmcblk0p12
lrwxrwxrwx 1 root root date fota -> ../../
    mmcblk0p10
lrwxrwxrwx 1 root root date module -> ../../
    mmcblk0p9
lrwxrwxrwx 1 root root date param -> ../../
    mmcblk0p4
lrwxrwxrwx 1 root root date ramdisk1 -> ../../
    mmcblk0p7
lrwxrwxrwx 1 root root date ramdisk2 -> ../../
    mmcblk0p8
lrwxrwxrwx 1 root root date recovery -> ../../
    mmcblk0p6
lrwxrwxrwx 1 root root date rootfs -> ../../
    mmcblk0p15
lrwxrwxrwx 1 root root date system -> ../../
    mmcblk0p11
lrwxrwxrwx 1 root root date system-data -> ../../
    mmcblk0p13
lrwxrwxrwx 1 root root date user -> ../../
    mmcblk0p14 /* User partition */
```

Listing 3. Partitions on Samsung Gear 2 Neo.

Note from the partitions in Listing 3, the one most relevant to an investigator would be the user partition which is `mmcblk0p14`. Once the partitions were determined, another tool was needed to transfer the image of the partition to the host computer. `dd` already came preloaded with the Tizen operating system so there was no need to install `dd` on the watch's system. The tool used in this procedure was `netcat`, which is bundled inside of Toybox. Toybox is a package that contains a series of command utilities which can run on ARM systems. The version used in our setup was `toybox-armv4l`, but version `toybox-armv6l` also worked when we experimented with it. Once downloaded, the Toybox file had to be pushed to the watch. This can be done via Windows Explorer if the user is able to see part of the watch's filesystem on the workstation it is connected to, or by using the `sdb push` command. Once the file was pushed or copied to the watch, the permission of the file had to be modified. In the watch shell, the commands in Listing 5 were employed in the same directory as the `toybox-armv4l` file.

```
/* Modify permissions */
fchmod +x toybox-armv4l
/* image partition with dd and pipe to netcat, -L
     puts netcat in listening mode */
dd if=/dev/mmcblk0p14 | ./toybox-armv4l nc -L
/* Port number being listened to on the watch */
44867
```

Listing 4. Starting `dd` command and `netcat` on the Samsung Gear 2 Neo watch.

```
/* Modify permissions */
fchmod +x toybox-armv4l
/* image partition with dd and pipe to netcat, -L
     puts netcat in listening mode */
dd if=/dev/mmcblk0p14 | ./toybox-armv4l nc -L
/* Port number that is being listened to on the
    watch */
44867
```

Listing 5. Starting `dd` command and `netcat` on the Samsung Gear 2 Neo watch.

Once the previous command was executed on the watch, and the watch was listening for a connection, a port number was displayed to the user as shown in Listing 5 which was port 44867 in our experiment, but this port number is randomly generated. This port number is important because we now had to forward that port number from the debug bridge on the watch to the system by executing the following command

```
sdb.exe forward tcp:44867 tcp:44867
```

Listing 6. SDB port forwarding for Samsung Gear 2 Neo watch.

Once the port was forwarded, the command in Listing 7 was executed on the workstation to receive the image. In our case, since the workstation we used was a Windows workstation, `netcat` for Windows had to be first installed. We were now able to acquire a physical image of the user partition which we were then able to load into widely adopted forensic tools such as the open source tool Autoposy.

```
/* Send request to watch on port number 44867 and
     send it to image file */
nc 127.0.0.1 44867 > gear2neo.img
```

Listing 7. Netcat on workstation for Samsung Gear 2 Neo watch.

3) Results: We were able to find a sleuth of artifacts with possible evidentiary value. We present a summary of our findings in Table II. As can be seen from our summary of findings, we are able to retrieve a significant amount of data. We were also able to recover e-mails that were viewed and received on the Samsung Gear 2 Neo watch watch as well.

B. LG G watch

The LG G watch uses the Android Wear operating system which is based on a modified Linux Kernel. The watch again has no WiFi connection, and only has Bluetooth capability to connect to the phone it syncs with. Therefore, imaging the watch has to be executed over the USB connection when connected to a workstation. However, the first step for being able to image the watch was to gain root access.

1) Gaining root access: It is critical to note that the method we explored will perform a factory reset on the watch and may not be deemed as a forensically sound method. Forensics on a rooted watch would be possible since we would then be able to image the watch.

Before gaining root access, ADB Debug must be enabled on the watch which can be done by going to Settings → About and then clicking on the Build Number multiple times. Once ADB Debug was enabled, the drivers were downloaded for the watch on the workstation and the watch was connected via the USB cable. Using the LG G Watch Restore Tools (contains most of the files used in this setup), the adb devices command was used to ensure that the device is visible and authorized. If the device was unauthorized then we had to unpair the watch from the phone and re-pair the phone to the watch over Bluetooth and then execute the following commands on the workstation:

```
adb kill-server
adb start-server
```

Listing 8. Commands used to start and kill the adb server on the workstation when the watch was seen as an unauthorized device which created a prompt on the phone synced with the LG G watch.

This should create a prompt on the phone that must be accepted. Once the device was listed and authorized, the following commands were executed:

```
/* Boot into bootloader */
adb reboot-bootloader
/* Unlock bootloader */
fastboot oem unlock
```

Listing 9. ADB reboot-bootloader and unlock for the LG G watch.

Yes was then selected manually on the watch; which initiated a factory reset on the device. After the watch booted, the initial setup was performed on it. This included re-enabling ADB Debugging mode and re-pairing the phone with the watch. After the watch was set up, the file `SuperSU-2.4.zip` was downloaded on the workstation and pushed to the device by and executing the commands:

```
/* Push .zip to /sdcard/download */
adb push Wear-SuperSU-2.4.zip /sdcard/download
/* Boot into bootloader */
adb reboot-bootloader
/* Boot into twrp */
fastboot boot openrecovery-twrp-2.8.1.0-dory.img
```

Listing 10. ADB push and twrp on the LG G watch.

The device was then rebooted and a new menu appeared on the watch. On the menu 'install' was clicked to the 'downloads' folder where the `Wear-SuperSU-2.4.zip` was selected. Then on the bottom of the screen, the blue circle was slided across to install the zip file. After the install completed, the system was rebooted. Root access was now granted and can be accessed in the shell using the `su`-command.

2) Imaging method: Once again, like the Samsung Gear 2 Neo watch, `netcat` in Toybox was used to send the data from the watch back to the workstation. After the Toybox ARM binary was downloaded, it was pushed to the device using the following command:

```
/* Push Toybox to /sdcard/download */
adb push toybox-armv4l /sdcard/download
```

Listing 11. Pushing Toybox to the LG G watch.

The file was then moved to the dev directory and its permissions was changed to allow it to have executable privileges as shown:

```
/* Start shell */
adb shell
/* Switch to super user */
su
/* Move Toybox to /dev/ */
mv /sdcard/download/toybox-armv4l /dev/
/* Change ownership to root:root */
chown root:root toybox-armv4l
/* Change permissions */
chmod 755 toybox-armv4l
```

Listing 12. Superuser access /changing permissions and ownership of Toybox on LG G watch.

We then had to explore which partition is the user partition to be imaged. To do that, we had to navigate to `/dev/block/platform/msm_sdcc.1` and executed the command `ls -al by-name` which resulted in Listing 13. This showed the the userdata was located in `/dev/block/mmcblk0p21`.

```
root@dory:/dev/block/platform/msm_sdcc.1 $ ls -al
     by-name
ls -al by-name
lrwxrwxrwx root root date DDR -> /dev/block/
    mmcblk0p18
lrwxrwxrwx root root date aboot -> /dev/block/
    mmcblk0p5
lrwxrwxrwx root root date abootb -> /dev/block/
    mmcblk0p9
lrwxrwxrwx root root date boot -> /dev/block/
    mmcblk0p15
lrwxrwxrwx root root date cache -> /dev/block/
    mmcblk0p20
lrwxrwxrwx root root date grow -> /dev/block/
    mmcblk0p22
lrwxrwxrwx root root date imgdata -> /dev/block/
    mmcblk0p10
```

Table II
SUMMARY OF FINDINGS FOR SAMSUNG GEAR 2 NEO WATCH.

Location	File Name	Purpose
apps.com.samsung.message.data.dbspace	.msg-consumer-server.db	Messages
apps.com.samsung.shealth	.shealth.db	Health/Fitness Data
apps.com.samsung.wemail.data.dbspace	.wemail.db	Email
dbspace	.contacts-svc.db	Contacts/Address book

```
lrwxrwxrwx root root date laf -> /dev/block/
    mmcblk0p14
lrwxrwxrwx root root date metadata -> /dev/block/
    mmcblk0p11
lrwxrwxrwx root root date misc -> /dev/block/
    mmcblk0p12
lrwxrwxrwx root root date persist -> /dev/block/
    mmcblk0p13
lrwxrwxrwx root root date recovery -> /dev/block/
    mmcblk0p16
lrwxrwxrwx root root date rpm -> /dev/block/
    mmcblk0p2
lrwxrwxrwx root root date rpmb -> /dev/block/
    mmcblk0p7
lrwxrwxrwx root root date sbl1 -> /dev/block/
    mmcblk0p1
lrwxrwxrwx root root date sbl1b -> /dev/block/
    mmcblk0p6
lrwxrwxrwx root root date sdi -> /dev/block/
    mmcblk0p4
lrwxrwxrwx root root date ssd -> /dev/block/
    mmcblk0p17
lrwxrwxrwx root root date system -> /dev/block/
    mmcblk0p19
lrwxrwxrwx root root date tz -> /dev/block/
    mmcblk0p3
lrwxrwxrwx root root date tzb -> /dev/block/
    mmcblk0p8
lrwxrwxrwx root root date userdata -> /dev/block/
    mmcblk0p21 /* User partition */
```

Listing 13. LG G watch partitions.

Once the permissions were set, then toybox-armv4l could be executed allowing for dd to be used over netcat as shown here:

```
/* image partition with dd and pipe to netcat, -L
     puts netcat in listening mode */
dd if=/dev/block/mmcblk0p21 | ./toybox-armv4l nc
    -L
/* Port number being listened to on the watch */
44867
```

Listing 14. Using `netcat` with `dd` on the LG G watch.

Once the command was executed on the watch, and the watch was listening, a port number was displayed to the user. In the example shown in Listing 14, the port number was 44867. This port number was important, because we now had to forward that port number from the debug bridge on the watch to the workstation's port. The command shown below was used to achieve that:

```
adb forward tcp:44867 tcp:44867
```

Listing 15. Port forwarding for adb on LG G watch.

Once the ports were forwarded, the command in Listing 16 was executed on the workstation. This command required `netcat` to be installed on the workstation in our case, since the workstation we used was a Windows workstation we had to download `netcat` for Windows.

```
/* Send request to watch on port number 44867 and
     send it to image file */
nc 127.0.0.1 44867 > LG.img
```

Listing 16. Using `netcat` with `dd` on the LG G watch.

We were now able to acquire a physical image of the user partition which we were then easily able to load into widely adopted forensic tools such as the open source tool Autoposy.

3) Results: It is imperative to first note that the method described for gaining root access to the watch performs a factory reset on the watch. Therefore, data that is deleted may be irrecoverable. However, we would like to share the layout of the artifacts in this section for the LG G watch. We would also like to note that although a factory reset was performed on the LG G watch during the rooting process, we were still able to recover the e-mail address of the user.

Overall, we were able to locate artifacts of possible evidentiary value. We present a summary of our findings shown in Table III.

V. FUTURE WORK

Despite significant advances during the past 15 years, smart watch features, uses and design are all at their infancy. Equally, if the excitement among consumers of smart watches and of the data they produce is any indicator, widespread adoption could lead to entirely new applications and vastly expanded data sets. Critical throughout this process will be an understanding of the forensic value of the evidence that is generated by or transmitted through smart watches. It would be extremely relevant to the forensic community to find less intrusive methods for acquiring data from the watches themselves.

VI. CONCLUSIONS

The goal of this work was to provide a preliminary forensic analysis of smart watches. Because of built in features like a heart rate monitor and pedometer, it is easy to see how they can be a haven for valuable forensic data. Our work showed that some useful artifacts can be be recovered from the phone, including set up date, applications on the watch, timestamps of updates performed, and voice memos that were recorded on the watch. At the time of writing this paper, we were unable to determine if data gathered by the applications like actual footsteps taken by the user or the actual rate of a user's heartbeat were stored or encrypted on the phone. We conclude that the method we used to acquire a physical image of the

Table III
SUMMARY OF FINDINGS FOR LG G WATCH.

Location	File Name	Purpose
data.com.android.providers.calendar.databases	calendar.db	Events/Notifications
data.com.android.providers.contacts.databases	contacts2.db	Contacts/Address book
data.com.google.android.apps.fitness.databases	pedometer.db	Health/Fitness Data

Samsung Gear 2 Neo is more forensically sound than the method used to acquire data from the LG G watch since the LG G watch had to be reset to factory settings in order to flash its memory and gain root access. Additional experimentation and reverse engineering is required to gain a more thorough understanding of the forensic value of these devices especially as new smart watches start hitting the market with the soon to be released Apple watch.

REFERENCES

[1] C. Narayanaswami and M. T. Raghunath, "Alarm interface for a smart watch," Nov. 5 2002, uS Patent 6,477,117.

[2] A. Doud, "Now or never: The future of smartwatches," 11 2014. [Online]. Available: http://pocketnow.com/2014/09/03/now-or-never-the-future-of-smartwatches

[3] A. Adams. (2014, July) The size of the smartwatch market & its key players. [Online]. Available: http://www.forbes.com/sites/arieladams/2014/03/07/the-size-of-the-smartwatch-market-its-key-players/

[4] R. Ziegler and S. Scibird. (2014, 11) Why teens will covet the apple watch why teens will covet the apple watch. [Online]. Available: http://www.marketwatch.com/story/why-teens-will-covet-apples-iwatch-2014-09-09

[5] M. Wolf, "Smart watches to have starring role in future of smart home control," 4 2014. [Online]. Available: http://www.forbes.com/sites/michaelwolf/2014/04/08/smart-watches-to-have-starring-role-in-future-of-smart-home-control/

[6] J. Kim, "Digital evidence from android-based smartwatch," Forensic Insight. Forensic Insight, 7 2013.

[7] PewResearchCenter, "Mobile technology fact sheet," 2014. [Online]. Available: http://www.pewinternet.org/fact-sheets/mobile-technology-fact-sheet/

[8] A. Ariffin, K.-K. R. Choo, and J. Slay, "Digital camcorder forensics," in *Proceedings of the Eleventh Australasian Information Security Conference-Volume 138*. Australian Computer Society, Inc., 2013, pp. 39–47.

[9] J. Moore, I. Baggili, A. Marrington, and A. Rodrigues, "Preliminary forensic analysis of the xbox one," *Digital Investigation*, vol. 11, pp. S57–S65, 2014.

[10] A. Migicovsky, Z. Durumeric, J. Ringenberg, and J. A. Halderman, "Outsmarting proctors with smartwatches: A case study on wearable computing security."

[11] G. Grispos, W. B. Glisson, and T. Storer, "Using smartphones as a proxy for forensic evidence contained in cloud storage services," in *System Sciences (HICSS), 2013 46th Hawaii International Conference on*. IEEE, 2013, pp. 4910–4919.

[12] M. Al-Hadadi and A. AlShidhani, "Smartphone forensics analysis: A case study," *International Journal of Computer and Electrical Engineering*, vol. 5, no. 6, 2013.

[13] P. Thomas, P. Owen, and D. McPhee, "An analysis of the digital forensic examination of mobile phones," in *Next Generation Mobile Applications, Services and Technologies (NGMAST), 2010 Fourth International Conference on*. IEEE, 2010, pp. 25–29.

[14] M. Bader and I. Baggili, "iphone 3gs forensics: logical analysis using apple itunes backup utility," *Small scale digital device forensics journal*, vol. 4, no. 1, pp. 1–15, 2010.

[15] A. Ariffin, C. DOorazio, K.-K. R. Choo, and J. Slay, "ios forensics: How can we recover deleted image files with timestamp in a forensically sound manner?" in *Availability, Reliability and Security (ARES), 2013 Eighth International Conference on*. IEEE, 2013, pp. 375–382.

[16] L. Gómez-Miralles and J. Arnedo-Moreno, "Analysis of the forensic traces left by airprint in apple ios devices," in *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*. IEEE, 2013, pp. 703–708.

[17] C. DOrazio, A. Ariffin, and K.-K. R. Choo, "ios anti-forensics: How can we securely conceal, delete and insert data?" in *System Sciences (HICSS), 2014 47th Hawaii International Conference on*. IEEE, 2014, pp. 4838–4847.

[18] J. Lessard and G. Kessler, "Android forensics: Simplifying cell phone examinations." 2010.

[19] T. Vidas, C. Zhang, and N. Christin, "Toward a general collection methodology for android devices," *digital investigation*, vol. 8, pp. S14–S24, 2011.

[20] S.-W. Chen, C.-H. Yang, and C.-T. Liu, "Design and implementation of live sd acquisition tool in android smart phone," in *Genetic and Evolutionary Computing (ICGEC), 2011 Fifth International Conference on*. IEEE, 2011, pp. 157–162.

[21] C. Racioppo and N. Murthy, "Android forensics: A case study of the "htc incredible" phone," in *Proceedings of Student-Faculty Research Day*. CSIS, Pace University. Seidenberg School of CSIS, Pace University, New York, 2012, p. B6.

[22] P. Albano, A. Castiglione, G. Cattaneo, and A. De Santis, "A novel anti-forensics technique for the android os," in *Broadband and Wireless Computing, Communication and Applications (BWCCA), 2011 International Conference on*. IEEE, 2011, pp. 380–385.

[23] N. Al Barghouthy, A. Marrington, and I. Baggili, "The forensic investigation of android private browsing sessions using orweb," in *Computer Science and Information Technology (CSIT), 2013 5th International Conference on*. IEEE, 2013, pp. 33–37.

[24] K.-J. Karlsson and W. B. Glisson, "Android anti-forensics: Modifying cyanogenmod," in *System Sciences (HICSS), 2014 47th Hawaii International Conference on*. IEEE, 2014, pp. 4828–4837.

[25] S. Ali, S. AlHosani, F. AlZarooni, and I. Baggili, "ipad2 logical acquisition: Automated or manual examination?" in *Proceedings of the Conference on Digital Forensics, Security and Law*, 2012, pp. 113–128.

[26] M. Al Marzougy, I. Baggili, and A. Marrington, "Blackberry playbook backup forensic analysis," in *Digital Forensics and Cyber Crime*. Springer, 2013, pp. 239–252.

[27] A. Iqbal, H. Alobaidli, A. Marrington, and I. Baggili, "Amazon kindle fire hd forensics," in *Digital Forensics and Cyber Crime*. Springer, 2014, pp. 39–50.

[28] Ash. (2014) The tizen samsung gear 2 and gear 2 neo get root access. [Online]. Available: http://www.tizenexperts.com/2014/06/hack-tizen-samsung-gear-2-gear-2-neo-get-root-access/

[29] Tomsgt. (2014, 09) Lg g watch root, bootloader unlock, and much more. [Online]. Available: http://forum.xda-developers.com/g-watch/general/videos-lg-g-watch-root-bootloader-unlock-t2815007

[30] K. Kent, S. Chevalier, T. Grance, and H. Dang, "Guide to integrating forensic techniques into incident response," *NIST Special Publication*, pp. 800–86, 2006.