# Analysis of Android Smart Watch Artifacts

Shreyas Parikh<sup>1</sup>, IFS, GFSU, Dhaval Chavda<sup>2</sup>, IFS, GFSU, Shourjo Chakraborty<sup>3</sup>, SysTools Software Pvt. Ltd. Dr. Parag H. Rughani<sup>4</sup>, IFS, GFSU, Dr. M. S. Dahiya<sup>5</sup>, IFS, GFSU

Abstract— Innovations of smart phones have made this era as an era of smart devices. People are using smart phones more than any other device / tool in their day-to-day life. Since, android entered to this market, use of wearable devices became possible technically and economically. These days, wearable devices like wrist band, earring, smart watch, smart shoes, etc. have become easily available and affordable. Android based smart watch is one of such popular devices being used by many people around the world. Though the watch is dependent on android smart phone, it contains lots of useful information about user. The user information stored on the device can be important in tracing any cyber crime / traditional case. This paper discusses what information the Android Smart Watch sores and analysis of this information from forensics point of view.

Index Terms— Android Forensics, Smartwatch Forensics, Wearable Forensics, Android Smartwatch, Android Smartwatch Artifacts, Android Analysis, Artifact Analysis.

## **1** INTRODUCTION

As per emarketer survey there are estimated 1914.6 million people owned a smartphone<sup>[1]</sup>. The major platforms in the current smartphone market are Google's Android with almost eighty percent of the market, Apple's iOS with sixteen percent and Microsoft's Windows Mobile OS with three percent.

This shows Google's Android operating system owns the largest portion of smart phone market. With the aim of integrating all digital platform with smart OS, Google took its first step by introducing Android wear last year in March 2014 to step into wearable market.

With the introduction of the smart watch to the family of wearable computing devices by Android, Smart watches are now seen to be socially acceptable in the modern digital world, and can possibly be used as an alternative interface for information access.

Since, smart watches can be used as an interface for operating and accessing smart phones, the use is expected to increase in near future. This handy gadget will be on the wrists of many people in coming days. As the usage will increase possibilities of number of crimes related to smart watch may also increase.

Looking at the prospectus feature, we can not neglect possibilities of required investigation of Smart watch found from a crime scene. The smart watch can become a very strong evidence in solving a crime. Since it can contain lots of information, it may become one of the important evidence which can reveal useful information related to a victim or suspect.

This work focuses on extraction and investigation of possible sensitive artifacts from smart watch to prove relevance with the case. Main objective of this work is to extensively explore possible locations from where sensitive information as well as information conveyed through communication channels like Bluetooth can be retrieved.

<sup>•</sup> Shreyas Parikh, First Author has completed his M. S. in Digital Forensics and Information Assurance from GFSU, India, E-mail: shreyas.prkh@gmail.com

Dhaval Chavda, Second Author has completed his M. S. in Digital Forensics and Information Assurance from GFSU, India, E-mail: <u>er.dhaval.chavda@gmail.com</u>

<sup>•</sup> Dr. Parag H. Rughani, Corresponding Author Assistant Professor in Digital Forensics and Cyber Security at GFSU, India, E-mail: parag@gfsu.ac.in

## **2 WEARABLE COMPUTING**

## 2.1 Background

The concept of wearable computing emerged during mid of 90's. At that time, carrying an "always- on" computer with head mounted display and control interface became practically possible.

Reduced size of hardware components, availability of low cost sensors and existence of widespread Internet access, allowed wearable computing devices to become more commonplace, readily wearable and socially acceptable. "This energetic expansion of miniature computing devices has led to the concept of The Internet of Things (IoT) which can be described as a collection of interconnected and interactive devices which are able to communicate useful real-time information between one another" (Swan, 2012) <sup>[2].</sup>

The smart watch falls in the category of the Internet of Things, which acts as a peripheral device to a connected smart phone. By establishing connection between smart watch and smart phone, one can easily operate and use smart phone through smart watch.

## 2.2 Android Smart Watch (Android Wear)

Smart Watch is a device that maintains persistent connectivity (wireless connection) to your mobile devices - usually a smart phone and can also receive notification like calls, social updates in terms of notification, instant messages and more.

On 18 th March 2014 Google Official Announced that they are coming for another arena into the world of Wearable Technologies with Android Wear, A New Operating System for Wearable Devices.

The Android Wear operating system uses Bluetooth communication to connect hand-held device(s). Prerequisite of operating system is, it requires hand-held device running on version Android 4.3 or higher and one companion application to synchronize data with device.[3]

Once connection is initiated, the wearable channels information and updates it from paired device, and comfortably displays them on the customer's wrist.

These updates include things like Gmail, Google Calendar, Google Now cards, and phone notifications, such as incoming calls and text messages. Android Wear is a wearable devices/gadget which keeps user's hands free and yet allows user to use his/her smart phone.

## 2.3 Wearable APIs

Basic Structure of Android Wear (Smart Watch and Google Glass) is similar to Android OS structure. As well the Kernel is also same as Android Operating System. Wearable data layer is used to handle lots of complex and complicated data in timely manner by interacting with smart phone.

International Journal of Scientific & Engineering Research, Volume 6, Issue 8, August-2015 ISSN 2229-5518

Basic structure of Android Wear is mainly based on three APIs.

- Data API
- Message API
- Node API

Node API: The Node API informs to smartphone when a node is connected. Node events are delivered to every application on devices.

Message API: The Message API manages API calls between devices.

Data API: The Wearable Data Layer API, which is part of Google Play services, provides a communication channel for the handheld and wearable apps. The API consists of a set of data objects that the system can send and synchronize over the wire and listeners that notify the apps of important events with the data layer.

Saminath explained communication between Android Wear and Android Phone in his work as: "communication between the Android Handheld device and Wearable Device. Wearable API gets reference from connected mobile device. Data Layer API supports syncing data between handheld and wearable devices.

Wearable Listener interface pulling the notifications from paired Android Smartphone's or tablets and display text using Message API. Message API is a one-way communication mechanism that's meant for 'fire-and-forget' tasks". This is just opposed to Wear's Data Layer API. The Google Play service will make this very easy. The communication between two apps over the Bluetooth link that pair two devices" <sup>[5]</sup>

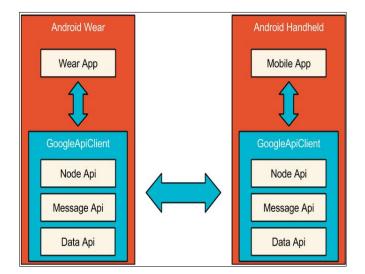


Fig. 2.3.1. Android Wear API Structure [4]

# 3 NEED OF DIGITAL FORENSICS OF SMART WATCH

Number of cyber crimes are increased day by day and criminals use latest tools and technologies for committing sophisticated crimes. As per the Symantec's Internet Security Threat Report published in April, 2015.

"Symantec expects to see further malware development and attacks on the Internet of Things as criminals find new ways to make money from doing so. For example, some attackers have used Darlloz to mine for crypto-currencies similar to bitcoins. Other attackers have leveraged hacked routers to carry out distributed denial-of-service attacks. Experience with PCs and, more recently, with mobile malware suggests that where there is opportunity created by technical exploits and motivation, such as greed, vindictiveness, or revenge, there will be cyberattacks."<sup>[6]</sup>

This not only predicts increase in number of attacks on IoT, it also alerts Cyber Security Experts and Forensic Experts to equip themselves for upcoming threats. Even though, prevention is better than cure, it is not completely possible to mitigate attacks. The incident response and forensic analysis of a crime / attack is equally important and required as securing a network.

This work emphasis mainly on recovering important artificats from Android Smart Watch, which may help Forensic Investigators in collecting information about Victim(s) and / or Criminal(s) related to the case.

# 4 METHODOLOGY

The experiment was carried out on a rooted Sony Smartwatch 3 (model # SW50) running android wear version 5.0.2 OS.

We used android SDK, FTK and Hex Editor to carry out the work.

## 4.1 Acquisition

Creating image of evidence is one of the most important tasks in any digital forensic investigation, because the thumb rule of forensic investigation says that you cannot work on actual evidences. For that we have to create bit by bit image of the target device or evidence in terms of forensics. For checking integrity of the evidence we should make sure that hash is calculated before starting the imaging of evidence or original data. We also need to verify it with the calculated hash of image. Further, to avoid any accidental modifications during the imaging process, the evidence device should be in write protected mode.

Android wear device stores information at different places within /dev partition. With the help of MTD of Linux kernel it stores memory chunks in running OS with names like "mmcblk[#]p[#]"

The command Is -al /proc/partition with su from adb shell will generate following output.

rvxrvxrvx /nncblk0p1		root	2015-04-15 14:26 version-info -> /dev	/bloc
		ok/nlatf	orm/sdhci.1/bu-name # ls -al	
s -al	1004/110	CEA DIACI	orm sulci.i by name # 15 al	
PUXPUXPUX	noot	root	2015-04-15 14:26 abi -> /dev/block/nncblk0p1	
PUXPUXPUX		root	2015-04-15 14:26 abi-sec -> /dev/block/mmcblk0p	<b>,</b>
PUXPUXPUX		root	2015-04-15 14:26 apps_log -> /dev/block/mmcblk0	
PUXPUXPUX		root	2015-04-15 14:26 boot -> /dev/block/mmcblk0p29	pra
PUXPUXPUX		root	2015-04-15 14:26 boot-parm -> /dev/block/mmcblk	0
PUXPUXPUX		root	$2015-04-15$ 14:26 cache $\rightarrow$ /dev/block/mmcblk0p30	
PWXPWXPWX		root	2015-04-15 14:26 cp-boot -> /dev/block/mmcblk0p	
PWXPWXPWX		root	2015-04-15 14:26 cp-image -> /dev/block/mmcblk0	
PUXPUXPUX		root	2015-04-15 14:26 devinfo -> /dev/block/mmcblk0	
PUXPUXPUX				
.ruxruxrux .ruxruxrux		root root	2015-04-15 14:26 dsp-dran -> /dev/block/mmcblk0 2015-04-15 14:26 randump -> /dev/block/mmcblk0p	857
PUXPUXPUX		root	2015-04-15 14:26 randump-dtb -> /dev/block/mmcb	
PUXPUXPUX		root	2015-04-15 14:26 recovery -> /dev/block/mmcblk0	
PWXPWXPWX		root	2015-04-15 14:26 recovery-dtb -> /dev/block/mmcbike	
			2015-04-15 14:26 s1sbl -> /dev/block/mmcblk0p21	откораз
PWXPWXPWX		root		1.11.0П
PWXPWXPWX		root	2015-04-15 14:26 sys-parn-dep -> /dev/block/mnc	
PWXPWXPWX		root	2015-04-15 14:26 sys-parm-ind -> /dev/block/mmc	
PWXPWXPWX		root	2015-04-15 14:26 system -> /dev/block/mmcblk0p3	1
PWXPWXPWX		root	2015-04-15 14:26 ta -> /dev/block/mmcblk0p26	
PWXPWXPWX		root	2015-04-15 14:26 u-boot -> /dev/block/mmcblk0p1	
PUXPUXPUX		root	2015-04-15 14:26 u-boot-env -> /dev/block/mmcbl	
PWXPWXPWX		root	2015-04-15 14:26 ubootlogo -> /dev/block/nmcblk	
PWXPWXPWX		root	2015-04-15 14:26 unts-cal -> /dev/block/nncblk0	
PWXPWXPWX		root	2015-04-15 14:26 userdata -> /dev/block/nncblk0	
PWXPWXPWX	root	root	2015-04-15 14:26 version-info -> /dev/block/mmc	blkØp15

Fig. 4.1.1. Memory Chunks

First line of above output is:

abi -> /dev/block/mmcblk0p1

Which indicates the abi partition data/information is stored in mmcblk0p1 memory block. Similarly different partitions like /boot, /system, /cache, /ram, /apps-log, etc and their relevant memory blocks can be seen from the same output. So for understanding what type of data a block can store one has to carve out that particular memory block.

For example, following command can be used to carve data from memory block /dev/block/mmcblk0p1 which refers to partition /abi.

dd if = /dev/block/mmcblk0p15 of=/sdcard/download/versioninfo.dd

Similarly, other blocks can be carved and separate dd image of each can be created for analysis.

## 4.2 Image Analysis

The paper discusses analysis of the main three images which are crucial as well as complex in any android device investigation. These images are of cache memory, system memory and userdata memory.

#### 4.2.1 Cache

Cache partition contains cached data of a particular application in android that helps device to accelerate performance. In our case, wear device gathers information of various logs inside cache partition like recovery, system and partition logs. For example recovery folder contain recovery information for various partition and other system files. This logs are appended and denoted with 'last\_log.x' where x is appended file number.

dence Tree	× File List	File List									
💁 cache.ing	Name	Size	Туре	Date Modified							
E-[* NONAME [ext4]	version	1	Regular File	4/20/2015 05:4							
	last_install			4/17/2015 05:0							
Inst-found     Inst-found     Inst-cased space     Inst-cased space	last_locale			4/17/2015 05:0							
	last log	16	Regular File	4/20/2015 05:4							
	last_log_1	20	Regular File	4/17/2015 04:4							
	last log.2	6	Regular File	4/16/2015 11:2							
	last_log.3	16	Regular File	4/16/2015 11:2							
	last_log.4	32	Regular File	4/16/2015 11:2							
	last_log.5	5	Regular File	4/16/2015 11:2							
	last log.6	5	Regular File	4/16/2015 11:2							
	log	16	Regular File	4/20/2015 05:4							
	recovery.fstab	2	Regular File	4/20/2015 05:4							
	storage.fstab	1	Regular File	4/20/2015 05:4							
tom Content Sources					4B Free: 170MB Backup Size:						
lence:File System Path File					Wipe_Available_in_GUI_IsPres						
	Primary_Block		/block/mmck	1k0p31							
	Display_Name:										
		Storage_Name: System Backup Path: /system									
	Backup Name: s										
	Backup Display		m								
	Storage Path:										
	Current File S										
	Fatab File Sus										

Fig. 4.2.1.1. Cache Image

## 4.2.2 System Image

System partition contains information about core system such as installed applications, media files, OEM Applications, fonts, framework etc. Command, Is –al can be used to see which memory block contains information about system partition.

Here in our case the partition is mapped with block mmcblk0p31. Following image displays tree structure of ext4 file system partition.

ence Tree X	ce Tree × File List																
😋 system img	Name								1	Si	ze	Тур	e		-	1	Date Modified
- (* NONAME (ext4)		app							4			Directory				-	4/04/2015 10:
	ī.	bin									4	,					4/04/2015 10:
🗈 🛄 app		etc									4						4/04/2015 10:
🗈 🧰 bin	fonts												ctor				1/04/2015 04:
E C forts		framework											ctor				1/04/2015 04:
		b		•													4/04/2015 10:
🗈 🚞 framework	_	ost+1										Directory Directory					H4/04/2010 10
🔅 🧰 lb																	1/04/2015 04:
lost-found		media											ctor				
🕀 🧰 media	sony sony							4 Directory					11/04/2015 04:				
🗈 🧰 priv-app																02/09/2014 19:	
🕀 🧰 sony											Directory					02/09/2014 19:	
🕀 🛅 usr		vende	or								4	Dire	ctor	Y		(	2/09/2014 19:
🖲 🧰 vendor		din									4	Dire	ctor	Y			4/04/2015 10
🛅 xbin	0	build.	prop								2	Reg	ular	File			11/04/2015 04:
- 🛅 [unallocated space]		recovery-from-boot.bak						1	45	Reg	ular	File		(	01/08/2008 12:		
	in the second se																
tom Content Sources									02-2E 00-0B								
tom content sources ×																	lost+found
dence:File System Path File Opt																	10304100md
																	bin-Ìbuil
																	d.prop. «·····
																	etc.gfont
																	sÎfram
									00-00								
	09	0 60	69	62	00	C5	02	00	00-10	00	05	02	6D	65	64	69	lib Åmedi
	0a	61	00	00	00	C7	02	00	00-10	00	80	02	70	72	69	76	aÇpriv
	0b	0 20	61	70	70	EF	02	00	00-00	00	04	02	73	6F	6E	79	-appi ·····sony
	0c	00 00	03	00	00	0C	00	03	02-75	73	72	00	30	03	00	00	usr.0
	0d	0 10	00	06	02	76	65	6E	64-6F	72	00	00	35	03	00	00	····vendor··5···
									6E-AA								····xbin <sup>2</sup> ·····
																	test.txt-from-bo
																	ot.pgüreco
>																	very-from-boot.b
Edt Remove Remove Al Greate Image																	ak
roperties Hex Value I Custom Con	Cur	sor po	is = (	); du	15 =	259	i; log	g sec	= 2076	8							
em.img/NONAME [ext4]/[root]																	

#### Fig. 4.2.2.1. System Image

#### 4.2.3 Data Image

This is where user data is stored. This includes everything from user settings & customizations, apps that you have downloaded and installed, your messages (SMS / MMS) as well as contacts.

rootEtetra ls –al	:/data # ]	ls -al				
-rw	root	root	2	2015-04-16	16:54	layout version
drwx		root		2015-04-16		
druxruxr-x	system	system		2015-04-20	16:11	anr
druxruxx		system		2015-04-21	13:37	app
drwx		root		2015-04-16	16:54	app-asec
drwxrwxx	system	system		2015-04-16	16:54	app-lib
druxruxx	system	system				app-private
lrwxrwxrwx	root	root				<pre>bugreports -&gt; /data/data/com.android.shell/files/}</pre>
drwxrwxx	root	root				dalvik-cache
drwxrwxx		system		2015-04-20		
drwxr-x		log		2015-04-16		
drwxrwx		drm		2015-04-16		
drwxrwx		systen		2015-04-21		
drwxr-xx		root		2015-04-16		
drwxrwx		root				lost+found
drwxrwx				2015-04-16		
drwxrwx				2015-04-16		
druxruxt		nisc		2015-04-16		
drwx		root		2015-04-21		
drwxrwxx		system				resource-cache
drwxxx		system		2015-04-16		
drwxrwxx		systen systen		2015-04-21		tombstones
drwx drwxrwxrwx		systen systen		2015-04-21		
aroxroxrox root@tetra:		system		2012-04-10	10:54	user
rootetetra	vuaca #					

Fig. 4.2.3.1. Data Image

Wiping data partition will restore your phone to factory settings, removing all apps, messages and user settings from the device<sup>[7]</sup>.

This folder has significant importance in investigation as it holds the valuable information. This is the partition that contains most of the data that belongs to user. The following screenshot displays current folder inside partition. Some of the important subdirectories under data folder are: App, Dalvik-cache, Data, Misc, Property and system.

#### 4.3 Important Artifacts

From our experiment we have found few important artificats, analysis of some of them are discussed below.

## 4.3.1 Paired Device Information

Paired device information is located inside /data/misc folder. It contains five subdirectory in which one of it is bt\_config. This file gives information about connected Bluetooth device with device id and mac address, as shown in following figure.

bt_config.aml		4/17/2015 11:3
bt_config.old	3 Regular File	4/17/2015 11:3
<n1 tag="Adapter"></n1>		
	igrationDone" Type="int">1 </th <th></th>	
	s" Type="string">30:a8:db:f4	1:28:8a
	de" Type="int">0	
	eryTimeout" Type="int">0 <th></th>	
		92ec9570f484ed57dcad2e8352e350
		113e14a0d7b21e50d2e49d7c5f7c116
		e9ccfc0ba6a7e92dd549fc4bb762f88
	Type="string">SmartWatch 3 2	
<n9 tag="LE_LOC&lt;/td&gt;&lt;th&gt;AL_KEY_ER" type="binary">8b4<th>12f1fbbff450e8180545255b17c261</th></n9>	12f1fbbff450e8180545255b17c261	
<n2 tag="AutoPairBl&lt;/td&gt;&lt;th&gt;acklist"><th></th></n2>		
<n1 tag="Addres&lt;/td&gt;&lt;th&gt;sBlacklist" type="string">00<th>0:02:C7,00:16:FE,00:19:C1,00:1B:FB,00:1E:3D,00:21:4F,00:</th></n1>	0:02:C7,00:16:FE,00:19:C1,00:1B:FB,00:1E:3D,00:21:4F,00:	
<n2 tag="ExactN&lt;/td&gt;&lt;th&gt;ameBlacklist" type="string"><th>Motorola IHF1000, i. TechBlueBAND, X5 Stereo v1.3, KML_CANK</th></n2>	Motorola IHF1000, i. TechBlueBAND, X5 Stereo v1.3, KML_CANK	
<n3 tag="FixedF&lt;/td&gt;&lt;th&gt;inZerosKeyboardBlacklist" th="" ty<=""><th>/pe="string"&gt;00:0F:F6</th></n3>	/pe="string">00:0F:F6	
<n4 tag="Partia&lt;/td&gt;&lt;th&gt;lNameBlacklist" type="string&lt;/th&gt;&lt;th&gt;g">BMW, Audi, Parrot, Car</n4>		
<n2 tag="Remote"></n2>		
<n1 tag="9c:e6:e7:6&lt;/td&gt;&lt;th&gt;a:54:c8"><th></th></n1>		
<n1 tag="Manufa&lt;/td&gt;&lt;th&gt;cturer" type="int">15</n1> <th></th>		
<n2 int"="" tag="LmpVer&lt;/td&gt;&lt;th&gt;Type=">6</n2> <th></th>		
<n3 tag="LmpSub&lt;/td&gt;&lt;th&gt;Ver" type="int">16653</n3> <th></th>		
<n4 <="" tag="Name" td=""><th>Type="string"&gt;Samsung Galaxy</th><th>/ Grand Duos</th></n4>	Type="string">Samsung Galaxy	/ Grand Duos
<n5 tag="DevCla&lt;/td&gt;&lt;th&gt;ss" type="int">5898764</n5> <th></th>		
<n6 tag="DevTyp&lt;/td&gt;&lt;th&gt;e" type="int">1</n6> <th></th>		
<n7 tag="LinkKe&lt;/td&gt;&lt;th&gt;yType" type="int">5</n7> <th></th>		
<n8 tag="PinLen&lt;/td&gt;&lt;th&gt;gth" type="int">0</n8> <th></th>		
	y" Type="binary">acadbebab59	00ab192b8c789aaa88d530
<n10 tag="Servi&lt;/td&gt;&lt;th&gt;ce" type="string">0000110a-0<th>0000-1000-8000-00805f9b34fb 00001105-0000-1000-8000-0080</th></n10>	0000-1000-8000-00805f9b34fb 00001105-0000-1000-8000-0080	

## 4.3.2 Voice Commands

Smart watch is operable through voice commands. This voice commands are transmitted via google gms service. This commands are also stored on the connected device and synced with respective google account. This gms service related information can be found inside /data/data/ and as a com.google.android.gms subdirectory. Commands that are triggered to perform various task can be identified by looking up for "voice\_action\_title".

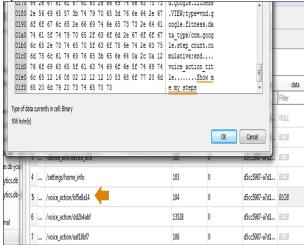


Fig. 4.3.2.1. Voice Commands

## 4.3.3 Bluetooth Packet Analysis

Device uses Bluetooth v4 or v4 LE (Low Energy). It requires 6 digit pass key to connect with host (android wear). But focusing on transmission packets that are transmitted to other device, is in plain text format making it easily readable and this packets can be captured via any network/Bluetooth packet capturing software.

Filter					Expression Cli	ear Apply	Save	
DF 63:54:08	Destination	f4:28:8a	(RCM43341R0	37.4 MHZ	Protocol MLI_EV Sony Br RFCOMM		KLVU	NUMBER
					Sony Br RFCOM			UIH Cha
gE_6a:54:c8 pi_f4:28:8a	(Sar SonyMob1 (BCI SamsungE	f4:28:8a 6a:54:c8	(BCM43341B0 (Samsung Ga	37.4 MHZ laxy Gran	SONY BERFCOM d Duos) RFCOM SONV BERFCOM	433	RCVd Sent	UIH Cha UIH Cha UIH Cha
<ul> <li>Bluetooth</li> <li>Bluetooth</li> <li>Bluetooth</li> <li>Bluetooth</li> <li>Bluetooth</li> <li>Data (990)</li> </ul>	HCI H4 HCI ACL Pack L2CAP Protoc RFCOMM Protoc bytes) adb0a4ad80a0a	ket col ocol		•	tes captured ( 4726f	(8032 bit	s)	
0160 65 61 0170 0a 88 0180 74 61 0190 cb 01 0190 cb 01 0140 65 12 0140 65 12 0140 72 69 0140 72 69 0140 72 69 0140 72 69 0140 73 64 0210 69 63 0220 67 65 0230 5f 73 0240 8b 63 0220 5f 73 0240 8b 73 0240 8b 74 0250 12 8b 0260 73 74 0250 12 8b 0250 12 8b	$\begin{array}{c} 69 & 64 & 12 & 1a \\ 74 & 68 & 65 & 72 \\ 70 & 70 & 20 & 06 & 76 \\ 12 & f6 & 06 & 08 \\ 12 & f6 & 06 & 08 \\ 0a & 08 & 02 & 12 \\ 72 & 65 & 63 & 69 \\ 12 & 04 & 12 & 02 \\ 70 & 74 & 69 & 6f \\ 6e & 12 & 0b \\ 70 & 74 & 69 & 6f \\ 6e & 12 & 0b \\ 70 & 74 & 69 & 6f \\ 6e & 12 & 0b \\ 71 & 76 & 66 & 64 \\ 71 & 66 & 64 \\ 71 & 66 & 64 \\ 72 & 66 & 42 & 17 \\ 71 & 61 & 67 & 64 \\ 71 & 61 & 64 & 64 \\ 12 & 10 & 08 & 02 \\ 70 & 61 & 67 & 64 & 64 \\ 12 & 10 & 08 & 02 \\ 70 & 61 & 67 & 64 & 64 \\ 71 & 61 & 64 & 64 & 64 \\ 71 & 64 & 64 & 64 &$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$ \begin{array}{c} f \ 64 \ 61 \ 75 \\ 55 \ f \ 64 \ 60 \\ 108 \ 0911 \\ 127 \ 77 \ 77 \\ 70 \ 77 \\ 70 \ 77 \\ 77$	9 eather_s u key u 2 tae 2 tae 2 tae exey u 2 taexey u exey u exey u exey u exe u 	<pre>inny_day page_da ,R ppage_da .R ipperatur 41.J idescs reathers reather_pawandwandwadhe recastJ '_forecaR mperatu </pre>		

Fig. 4.3.3.1. Bluetooth Packet Analysis

## 4.3.4 Logcat

The Android logging system provides a mechanism for collecting and viewing system debug output. Logs from various applications and portions of the system are collected in a series of Circular buffers, which then can be viewed and filtered by the logcat command.(Source: Android developer forum)

So we can use Logcat to view different system events like verbose message, debug message, information regarding process, warning and system errors.



#### 4.3.5 Notifications

Application notifications are transmitted to base device via Google gms service. Investigating this directory can revel database with application name and notification. Following screenshot is example of Whatsapp notification received on device with sender number.

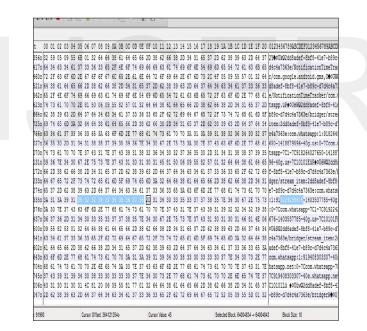


Fig. 4.3.5.1. Notifications (WhatsApp Details)

#### 4.3.6 DropBox Artifacts

The Dropbox Folder located under /data/system/ contains information about synchronize time (start and end time in Unix Timestamp with 13 digit number) with log detail as shown in following figure.

2		anc I	044	1.00	Dute mounicu
🗄 🛅 system		event_data@1429257421309.txt	1	Regular File	4/20/2015 08:0
- 🛅 dropbax		event_data@1429259494051.bxt	1	Regular File	4/20/2015 08:0
🚞 inputmethod	Ĩ	event data@1429262102119.bt	1	Regular File	4/20/2015 08:0
- 🗋 netstats	Ē	event data@1429264262177.bt	1	Regular File	4/20/2015 08:0
- D procstats	Ĩ	event data@1429266079020.bt	1	Regular File	4/20/2015 08:0
- 🗀 recent_tasks - 🫅 registered_services		event data@1429267879135.txt	1	Regular File	4/20/2015 08:0
		event data@1429270164131.bt	1	Regular File	4/20/2015 08:0
sync     sync     sagestats		event_data@1429272439630.txt	1	Regular File	4/20/2015 08:0
H- C users		event data@1429274629908.txt	1	Regular File	4/20/2015 08:0
		event data@1429355929958.txt	1	Regular File	4/20/2015 08:0
:\GFSU intem\shrey		event data@1429358211025.bt	1	Regular File	4/20/2015 08:0
data		event_data@1429360011149.bt	1	Regular File	
- Com amazon m Shoo android		event_data@1429361811309.bt	1	Regular File	
- 🛅 com android backupconfirm		event_data@1429363611429.txt		Regular File	4/20/2015 08:0
- 🛅 com android bluetooth		event log@1429355929952.bt	1	Regular File	4/20/2015 08:0
- 🛅 com.sonymobile.wearable.somcprovider		system app crash@1429258213011.txt	2	Regular File	4/20/2015 08:0
- 🛅 com todoist		SYSTEM_BOOT@1429335744579.txt	1	Regular File	
🗄 🛅 media		SYSTEM BOOT@1429340230901.txt	1	Regular File	
- 🛅 net.dheera.wearcamera		SYSTEM_BOOT@1429507591877.txt	1	Regular File	4/20/2015 08:0
🗄 🛅 system		system server crash@1429339956243.txt	2	Regular File	4/20/2015 08:0
- 🛅 dropbax		SYSTEM TOMBSTONE@1429507591897.bt	1	Regular File	4/20/2015 08/0
- 🗋 fw		3131EW_10WB310WE@1423007331037.00	1	Regular File	4/20/2013 00%
🛅 inputmethod					
- C instal_sessions		start=1429360011092 📥			
- 🔁 job		end=1429361811249			
- Canetstats					
C procetats C recent images					
		1429355929908,0,0,event_log_s	start 🔫		
- inclains - inclains - inclains					
- indixeren selvices					

#### Fig. 4.3.6.1. DropBoxArtifacts

## 4.3.7 Recent Tasks

The Recent task folder stored under /data/system/ contains list of tasks in xml format. It creates separate file for each task. Following figure shows information about 21\_task.xml.

nisc	Name	Size Type	Date Modified	
ropeity	9_task.xml	1 Regula	r File 4/17/2015 11:4	
ystem	21 task.xml	1 Regula		
E:\GFSU intem\raw images\data\data\system	20_task.xml	1 Regula		
- 🔁 dropbox	19 task.xml	1 Regula		
- inputmethod	14 task.xml	1 Regula		
- inetstats	13 task.xml	1 Regula		
- procstats	12 taskxml	1 Regula		
- Carecent_tasks		1 https://	1110 1/11/2010 11:10	
- 🛅 registered_services				
- 🗀 sync - 🛅 usagestats				
- alv	Qual service 15 01 secondary 1	Lation and the defense for all of		
- Cany	xml version="1.0" encoding="</td <td>uti-o stanuaione= yes (&gt;</td> <td></td> <td></td>	uti-o stanuaione= yes (>		
- 🛅 weekly	- <task <="" task_id="21" td=""><td></td><td></td><td></td></task>			
- 🔁 vearly	real_activity= <b>'com.google.</b> a root has reset= <b>'false'</b> auto			
- 🔁 users	first active time='1429270	-		_
- <u>-</u>	never_relinguish_identity='tr			
-	1' next_affiliation='-1' callin			
		y_uu=10010 calliny_pack	aye- connyooyie.an	nonahhaur
	component="com.google.	android anns fitness/con	n annala android wa	arahlo fitnoco
	flags="18010000" />	anurunappsinness/cur	n.yooyie.anuroiu.we	and Die Inches

#### Fig. 4.3.7.1. Recent Tasks

It contains useful information like package name, first active time, last active time, etc.

# **5 FUTURE SCOPE**

International Journal of Scientific & Engineering Research, Volume 6, Issue 8, August-2015 ISSN 2229-5518

The work can be further extended to forensically analyse smart watches from other manufacturers like Apple, Microsoft, Blackberry, etc. The researchers can continue working on the same techniques for retrieving data from other Android Wearable Devices.

## 6 CONCLUSION

The techniques discussed in this paper may assist Forensic Investigators in analysing Android based Smart Watch. The researchers can use this as a first step to find out other techniques for the same family of wearable devices.

## REFERENCES

- [1] http://www.emarketer.com
- [2] http://www.swan-forum.com/swan-2012-conference.html
- [3] http://www.android.com/wear/
- [4] http://swarmnyc.com/whiteboard/building-android-wear-watch-face-with-live-weather-data-3/
- [5] Saminath, "Power of Android Wearable Technology", International Journal of Scientific and Research Publications, Volume 5, Issue 2, February 2015.
- [6] https://www4.symantec.com/mktginfo/whitepaper/ISTR/21347932\_GA-internet-security-threat-report-volume-20-2015-social\_v2.pdf
- [7] http://en.miui.com/thread-12612-1-1.html