

Article

Digital Forensics Analysis of Ubuntu Touch on PinePhone

Yansi Keim ^{*,†} , Yung Han Yoon ^{*,†}  and Umit Karabiyik ^{*} 

Department of Computer and Information Technology, Purdue University, West Lafayette, IN 47907, USA

* Correspondence: ykeim@purdue.edu (Y.K.); yoon127@purdue.edu (Y.H.Y.); umit@purdue.edu (U.K.)

† These authors contributed equally to this work.

Abstract: New smartphones made by small companies enter the technology market everyday. These new devices introduce new challenges for mobile forensic investigators as these devices end up becoming pertinent evidence during an investigation. One such device is the PinePhone from Pine Microsystems (Pine64). These new devices are sometimes also shipped with OSes that are developed by open source communities and are otherwise never seen by investigators. Ubuntu Touch is one of these OSes and is currently being developed for deployment on the PinePhone. There is little research behind both the device and OS on what methodology an investigator should follow to reliably and accurately extract data. This results in potentially flawed methodologies being used before any testing can occur and contributes to the backlog of devices that need to be processed. Therefore, in this paper, the first forensic analysis of the PinePhone device with Ubuntu Touch OS is performed using Autopsy, an open source tool, to establish a framework that can be used to examine and analyze devices running the Ubuntu Touch OS. The findings include analysis of artifacts that could impact user privacy and data security, organization structure of file storage, app storage, OS, etc. Moreover, locations within the device that stores call logs, SMS messages, images, and videos are reported. Interesting findings include forensic artifacts, which could be useful to investigators in understanding user activity and attribution. This research will provide a roadmap to the digital forensic investigators to efficiently and effectively conduct their investigations where they have Ubuntu Touch OS and/or PinePhone as the evidence source.



Citation: Keim, Y.; Yoon, Y.H.; Karabiyik, U. Digital Forensics Analysis of Ubuntu Touch on PinePhone. *Electronics* **2021**, *10*, 343. <https://doi.org/10.3390/electronics10030343>

Academic Editor: Khaled Elleithy
Received: 27 December 2020
Accepted: 27 January 2021
Published: 1 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: Ubuntu Touch OS; digital forensics; mobile forensics; security; privacy; operating system

1. Introduction

Mobile device forensics is a continuously changing field that faces major challenges. Constant updates in device software make forensic investigations difficult as each new OS update may also cause a large shift in how data are stored and processed. These changes can result in forensics tools being unable to acquire key pieces of evidence for investigation properly [1,2].

Another challenge facing mobile forensics is the increasing number of phone manufacturers who currently represent a small part of the market but have the potential to grow in the near future. Devices such as the PinePhone experience limited support from mainline digital forensic tools as they are not as widely adopted. There is a substantial lack of research that goes into exploring these lesser-known devices. Investigators are left in a conundrum when unpopular devices appear in a case as there are few frameworks or formal documentation on how these phones should be investigated. Moreover, phones with open source operating systems may end up becoming increasingly prevalent, especially in underdeveloped and developing countries [3,4]. Digital forensics experts must resort to conducting their investigations in parallel with experimenting and using trial and error to determine the best method of acquiring and analyzing data that could be stored differently than what investigators are used to. During an active investigation, this workflow is not ideal as it may introduce errors into the investigation, and an untested methodology being used live for the first time in cases is not a standard that should be

promoted unless necessary. It is also possible that such untested practices may cause the evidence to be partially or entirely destroyed.

The PinePhone supports all major Linux distributions [5], and for the specific scope of this research, we proceeded with Ubuntu Touch, an open-source Linux OS. Ubuntu Touch is widely used on the top of PinePhone [6], which makes it a favorable candidate among hackers and thus for us to perform Ubuntu Touch forensics. Ubuntu Touch is also one of the operating systems that do not hold much market share. As a result, Ubuntu Touch is not well supported by forensic tools, which could be problematic if a device involved in a criminal investigation is discovered to be using it. Given these issues, in this paper, we investigate both the PinePhone and Ubuntu Touch OS currently being developed for it to gain an understanding of how forensic analysis on these novel platforms can be done. The lack of existing forensic support for Ubuntu Touch serves as a driving motivation to conduct this research. In the absence of a technical forensic methodology, the authors learn and inform as they proceed in its investigation. Since data extraction differs from Android to Linux to iOS, this particular variant of Linux has no literature on how to target data extraction. Our research provides a roadmap for investigators to investigate unusual phones. This OS analysis tangentially focuses on finding encrypted communication and data storage, especially passwords. This will enlighten the examiners on how and where Ubuntu Touch uses encryption.

Given the challenges and the gap in the literature, our contributions in this paper are as follows:

- analysis and increased understanding of Ubuntu Touch OS;
- creation of a list of important artifacts and locations for relevant forensic data for the PinePhone running Ubuntu Touch OS;
- increased understanding of how current forensic tools can be applied to less well-known OSes and devices;
- analysis of various third-party applications currently available on the Open Store;
- aid in preparedness for investigators who may encounter similar devices in future cases.

The rest of this paper is organized into sections to focus on multiple aspects of this research. Section 2 summarizes the current literature. Section 3 discusses our research methodology as well as the framework devised during and after the analysis of Ubuntu Touch OS. Section 4 presents the final results of this research. Section 5 discusses the challenges that occurred during our research and investigation. Section 6 enlists the advantages, disadvantages, and similarities identified in Ubuntu Touch OS with respect to its closest OS, Android. Finally, Section 7 concludes the paper.

2. Related Work

Ubuntu Touch is an open source OS with no systemic or organizational backing. As stated on its community website, it is built and maintained by an “international community of passionate volunteers” [7]. The goal of Ubuntu Touch is to create a new OS for mobile devices. As there is no current research that has forensically examined Ubuntu Touch OS, we will need to create a novel but reasonable and evidence-centric methodology. We shall start by working backward from the ultimate end goal and the desired results of this study. The ultimate goal of this research is to create a framework to help investigators understand where to look for commonly sought-after evidence such as call logs, messages, emails, contacts, videos, and images. To create a framework for investigating this new OS a priori, we also rely on previous research conducted in underlying or similar technologies, with the assumption that they are related and can act as a useful guide. This method of creating a framework does assume that there are few radical differences between Ubuntu Touch and the systems we will draw on to create this framework.

First, the Ubuntu Touch OS draws elements from the original desktop version of Ubuntu OS. As stated by the developers, Ubuntu Touch is “an extract of parts of Ubuntu, adapted to run in a mobile touch-screen environment but also capable of functioning as

a desktop” [8]. From this, we can gather that there will be a substantial portion of the Ubuntu file system and organization or, more generally, a Linux file structure. Ubuntu is a well-known Linux system that is compliant with the Filesystem Hierarchy Standard (FHS) [9], which is defined by the Linux Foundations’ Linux Standard Base workgroup [10]. As such, we can look to the file system standards defined by the LSB workgroup to create a shortlist of directories and paths that we should look at first to determine their forensic investigative worth. The FHS standard defines a required directory structure. In Table 1, we evaluate the forensics implications of each subdirectory and provide a rationale for why we will or will not include them as part of our framework for investigating Ubuntu Touch.

Table 1. High-level directory structure of Linux file systems and its forensic relevance.

Directory	Description	Forensic Rationale
/bin	Essential command binaries	May not contain user data, contains basic tools like bash, gcc, etc.
/boot	Static files of the boot loader	Out of scope
/dev	Device files	Not needed for mobile device
/etc	Host-specific system configuration	Application specific investigations may have configurations
/lib	Essential shared libraries and kernel modules	Assumed clean; Attacks are out of scope
/media	Mount point for removable media	Not needed for mobile device
/mnt	Mount point for mounting a filesystem temporarily	Not needed for mobile device
/opt	Add-on application software packages	Potentially useful if investigating apps
run	Data relevant to running processes	More in depth, potentially out of scope
/sbin	Essential system binaries	Highly technical, malware investigation relevant. May not have user data relevant to an investigation
/srv	Data for services provided by system	Out of scope
/tmp	Temporary files	Cached and temp files could contain user data
/usr	Secondary hierarchy	Contains user data
/var	Variable data	Application logs, potential user data present
/home	Stores user home directories	Large amount of user data

We will, therefore, include the directories that contain relevant user data as part of the investigation framework for Ubuntu Touch. Ubuntu, or more generally Linux-based operating systems, forensics is an area that has been widely studied. Tools such as Autopsy and Scalpel are noted as being able to conduct Ubuntu forensics as they are able to read the most commonly used EXT file systems [11]. Fairbanks et al. [12] provide a helpful guide to understand and dive deeper into the EXT4 file system, which is also Ubuntu Touch’s file system as investigated and reported further in this paper. Fairbanks et al. have further studied data recovery in the EXT4 file system in [13]. For acquisition and analysis, the authors prefer Autopsy, which is a trusted and most updated open-source forensic tool for investigators [14]. Previous work conducted in this area has identified certain directories and specific files such as */etc/shadow*, */user/lib*, */etc/shadow* and */etc/passwd* as files of importance due to their contents being especially relevant to an investigator [11]. We will use this framework of known Linux file system directories and files to guide our own analysis of Ubuntu Touch to help us more efficiently look through the acquired image to determine where relevant forensic artifacts may be stored.

Second, Ubuntu Touch is not alone in being an OS adapted from one hardware platform to another. Other OSes like Windows have undergone similar adaptations when moving from a pure desktop or laptop environment to a tablet or phone operating environment. Research in that area can, therefore, be used here as guidance.

It is noted in Windows phone forensics, and mobile forensics in general, that data acquisition involves the installation of a small program on the device to extract data [15]. However, not only is Ubuntu Touch still under development, but the underlying hardware is as well. As such, it is unlikely that we will be able to find any forensics software that is capable of installing such an application that would enable us to extract data from

the phone in a conventional manner. Other forensic works on Windows Phone 7 have shown that it is still possible to extract useful data from new platforms by using already developed and available tools [15]. Hence, we may try to accomplish our task using a similar methodology. There are a number of tools available for imaging Ubuntu OS or, more generally, Linux machines. The list of tools includes, but not limited to, dd, FTK Imager, EnCase, and Magnet AXIOM.

As the Ubuntu Touch is a smartphone OS that is Linux based, it may be prudent to attempt to use the Android Debug Bridge (ADB) tool as well to extract data. However, it must be noted that Ubuntu Touch is not the same as Android in terms of data extraction. In the Android operating system, use of ADB is a very popular method of pulling data from the mobile device. However, there is not enough literature on Ubuntu Touch OS for data extraction procedures, nor is Ubuntu Touch an Android-based OS. While both OSes may be based on Linux, they are significantly different, and any attempted use of ADB will likely fail. We now have a list of forensic tools that we can use to attempt the acquisition of this novel OS installed on a novel hardware platform.

Third, the Ubuntu Touch OS has been modified to allow for mobile phone capabilities. Ubuntu Touch is derived from the desktop Ubuntu OS, and it has a similar development path and history as Windows for mobile devices, being a desktop OS adapted for use on mobile devices. Finally, Ubuntu Touch, as an OS for mobile devices, will have functionality similar to Android and will share some forensic similarity in terms of how and where data related to smartphone functionality are stored. Research in Android forensics has already uncovered the basic layout of the file structure. The Android architecture as well as the general partitioning scheme used for Android devices is laid out in [16]. One of the more important partitions that we need to look at, if it is indeed present in Ubuntu Touch, is the */userdata* partition as it may contain user-installed applications and application data [16]. Interestingly, the */system* partition is where the OS files are stored, which is relevant to our study as we would like to investigate how Ubuntu Touch OS exists on disk [16]. If this partition exists, we will also need to take a closer look at it. Android devices, such as the numerous lines of Samsung smartphones, also pose additional data acquisition challenges in the form of OEM locking [16]. However, as we are conducting this study using the hardware platform of the PinePhone, which is supposed to be an open source platform, we do not expect these kinds of locks and restrictions to exist and cause any hindrances.

Other research has been conducted in novel mobile operating systems, such as the analysis of palm webOS [17]. Authors cite the need for research in these novel systems as relevant information pertinent to investigators may be present, but not recoverable by current tools. Research into novel operating systems will also grow as data security and consumer data privacy interest grow. New research prototyping novel operating systems designed to secure user data are already being published to create secure mobile platforms, sometimes called “paranoid”, as part of their core system design [18]. These new paranoid OSes will pose further challenges to investigators and will also need further study in the future. Understandably, there is a risk to this type of research focused specifically on unpopular mobile operating systems. They may be discontinued, such as in the case of Firefox OS, which has had previous research published [19]. However, despite Firefox OS being discontinued, the research conducted is still important. Another aspect of this research is to curb the cybercrimes spreading through the use of mobile phones [20,21]. With the advent of mobile OS with a focus on anonymity, it sometimes gives a free hand to attackers to perform malicious activity and get away with it. Therefore, this research is still valid as it could be relevant in the future if the old paradigms are reworked and get a second breath of life.

3. Methodology

In this section we introduce the methodology used to conduct this research. First, we go over the methodology used to setup the mobile device after it was received from the

manufacturer. We discuss the applications that were populated with data, and finally we explain the methodology used to image the device and perform the analysis.

There are two methods currently available to flash an Ubuntu Touch OS onto smartphones: (1) installing from an image file, and (2) using the UBports GUI application to install OS over USB connection from a computer to the device. The GUI installer does not officially or even unofficially support the Braveheart line of PinePhone. As such, while it may have decent support, the lack of official recognition from either Ubuntu Touch or Pine's developers means that there could be substantial issues with the GUI installer. As such, the manual installation method is preferred. The manual installation method used was provided by PinePhone as a set of procedures to follow when installing any image onto an SD card. The instructions relevant to installing Ubuntu Touch were followed, and the steps are listed below.

Installation procedure:

1. Downloaded the Ubuntu Touch OS image for PinePhone from: <https://ci.ubports.com/job/rootfs/job/rootfs-pinephone/>.
2. Connected SD card using an adapter to the computer station.
3. Flashed *img.gz* file onto SD card using Balena Etcher [22].
4. Inserted SD card into PinePhone and booted up the device.
5. Followed the on-screen instructions to set up the operating system:
 - (a) Selected system language: English (United States).
 - (b) Connected to the local WiFi network.
 - (c) Selected Time Zone: Indianapolis.
 - (d) Set preferred name: Test.
 - (e) No password was set.
 - (f) The OS setup is completed, and it is now functional.

After the installation of Ubuntu Touch was complete, various third-party and native applications were installed. A test account, focyber86@gmail.com, was created and used on these apps for sign-up. Each application was then populated using basic user interactions such as sending a message, sending pictures, or browsing the web. The applications were chosen in an attempt to cover as many applications seen in real life, which could provide useful forensic information to investigators. These are applications such as social media, web browsers, communication platforms, and cloud storage. A full list of applications and how they were populated are shown in Table 2.

After the data population was finished, we made the image of the device. During testing, we discovered that removing the SD card which had been flashed with Ubuntu Touch and then booting the phone would result in the phone entering the stock firmware that it had shipped with, Postmarket OS. After the removal of the SD card, there was no evidence that any part of the installed OS remained. We reasoned that the Ubuntu Touch OS and all corresponding user data would thus be stored primarily on the SD card and that for acquisition purposes, imaging the SD card would be appropriate. This idea was validated by checking the acquired image to see if the OS and user files were present, which would indicate that we have collected data of forensic relevance. The procedures for imaging the PinePhone are given below.

1. Shutdown the PinePhone
2. Removed the back cover
3. Removed the SD card
4. Connected the SD card to the forensic station
5. Created a physical image using the raw format by utilizing FTK Imager on the forensic station
6. Load acquired image into analysis tool (Autopsy)

Table 2. Applications that were installed and attempted at populating with user data.

Native Application Name (Version)	Developer	Data Population/App Status (if Not Working)
Calendar	UBPorts	Created 2 events at different days and times with messages and locations
Phone (Calls)	UBPorts	Made outgoing calls, received incoming calls, tested missed calls
Phone (Contacts)	UBPorts	Added 3 new contacts with phone and name details
Phone (Messages)	UBPorts	Sent and received basic text messages; images could not be sent due to error
Morph Browser	UBPorts	Browsed the web; downloaded various images; browsed news articles
Gallery	UBPorts	Attempted to view images downloaded via email and web browser; Not working as loading an image from File Manager results in hang
Notes (v0.11.0)	UBPorts	Added a note with a tag for organization
Bluetooth	UBPorts	Not working turning on Bluetooth generates no list of devices to pair with, the switch then flips itself back off
3rd-Party Application Name (Version)	Developer	Data Population/App Status (if Not Working)
Skype (v1.3)	Ruben Carneiro	Had conversations with other account, sent and received multimedia; initiated and received Skype call
Google Mail (v0.4)	Jose M Reyes	Received various emails with text, mp3, mp4, and image content
Pesbuk (1.3) Facebook app	Kugi Eusebio	Logged into account; browsed Facebook feed; looked at messages; looked at Facebook Messenger chat
WebTelegram (v0.1)	Jan Sprinz	Sent and received texts; images and multimedia could not be sent; old images and conversations could be seen
Axolotl Beta (v0.7.7.2)	Aaron Kimmig	Linked account
LinkedIn (v1)	Andy Bleaden	Signed into account; added connections; sent and received messages
Google Maps UT (v0.1)	Steven Barson	Not working
WhatsApp (v1.0)	Robin	Not working
Instagram (v0.3)	Ivo Xavier	Not working
Twitter (v1.2)	Gergely Barna	Sent direct messages; followed a few people; added friends
Youtube (v0.7)	Alan Pope	Watched a few videos; searched for a few videos
uReadIt (v4.8)	Jan Sprinz	Browsed reddit; following links to other sites resulted in hanging; following reddit hosted text posts was okay
Music	UBPorts	Opened app; attempted to load downloaded music, but hangs instead
Keeweb (v1.4.1)	Joan CiberSheep	Added a few fake passwords to websites
Onion Browser (v0.3)	Aaron Kimmig	Browsed the web through the TOR browser, no .onion sites were visited
Google Drive Basic (0.1)	Zsolt Mester-Darok	Not working; crashes
Drop Box (v1)	Andy Bleaden	Can log in, but no uploading possible
Log Viewer (v2.3)	Jan Sprinz	Not tested, used for troubleshooting only

The hardware used for this research included a SanDisk 64GB MicroSD card, and the Pinephone Braveheart edition. The Pinephone was running Ubuntu Touch build #270. Software used included balenaEtcher v1.5.76, FTK Imager v4.2.0.13, Autopsy v4.14 and v4.15, and DB Browser for SQLite v3.11.2.

4. Results and Discussion

As this study intends to analyze Ubuntu Touch on the PinePhone hardware platform as well as the behaviors and forensic potential of user applications, our results will be reported in three parts. The first part will go over the general phone functions, including the file system structure and its hierarchy. The second goes through native app features like contacts, calendars, and phone functionality (such as calls and text). The third part will identify third-party apps installed through *OpenStore* of the OS and list any findings related to each specific application. Both native and external apps report forensic artifacts of interest related to hidden directories, encrypted folders, archived, deleted, and carved files/folders.

4.1. Acquisition and Analysis

The acquired image of Ubuntu Touch (Build #270) from the PinePhone (Braveheart Edition) was taken using FTK Imager (version 4.2.0.13). The analyses were done with Autopsy (version 4.14.0 and version 4.15.0). Note that Ubuntu Touch is under constant development; hence, a new build is released every day. All ingest modules, including Hash Lookup, EXIF Parser, Encryption Detection, and Correlation Engine, among others, were run for better analysis. Note that running all ingest modules is a time-consuming process. Figure 1 shows the imaging specifications as acquired. Important attributes include the number of images, videos, audio; allocated space size; slack space. The imaging procedure took 56 min, and the file size (DD) was 59.4 GB.

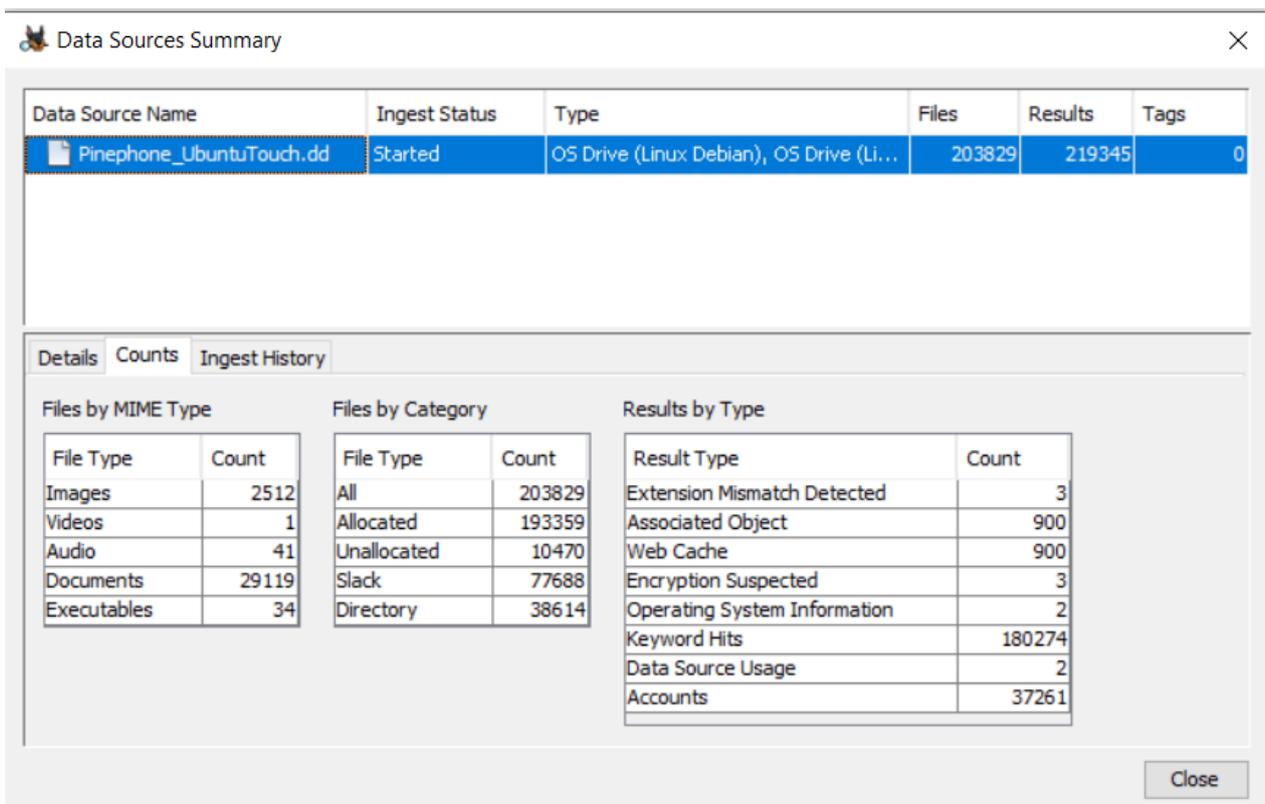


Figure 1. Case File Data Type Summary in Autopsy.

Here we note two things for digital forensic investigators. As the entire image of the phone exists on the micro SD card, acquisition of the device image is relatively easy compared to the acquisition of mobile devices where rooting or bypassing of OEM/bootloader locking restrictions may be required. The methodology that should be used is more akin to the imaging of USB/SD card removable media, including all relevant acquisition hardware such as SD card write blockers. The second thing of note is that the SIM card accepted by the PinePhone is a Micro SIM rather than a Nano-SIM. Investigators who wish to experiment with the PinePhone themselves, or want to replace the SIM for whatever reason, should be aware of the Micro SIM requirement and use an appropriate SIM card adapter if needed.

4.2. Core Phone Function Findings

Upon analyzing the acquired phone, we observed that Ubuntu Touch is organized similarly as an Ubuntu desktop machine. Figure 2 shows the Filesystem Hierarchy Standard (FHS) in most Ubuntu (from Linux Distributions) filesystems. Exploring each of the directories, we found that most of these directories, like a normal Ubuntu machine, do not contain third-party application data. There were no notable differences between the

Ubuntu Touch's file structure and its contents compared to a desktop installation of Ubuntu. Furthermore, upon initial analysis, we have verified that Ubuntu Touch runs using an EXT4 file system, as shown in Figure 3.

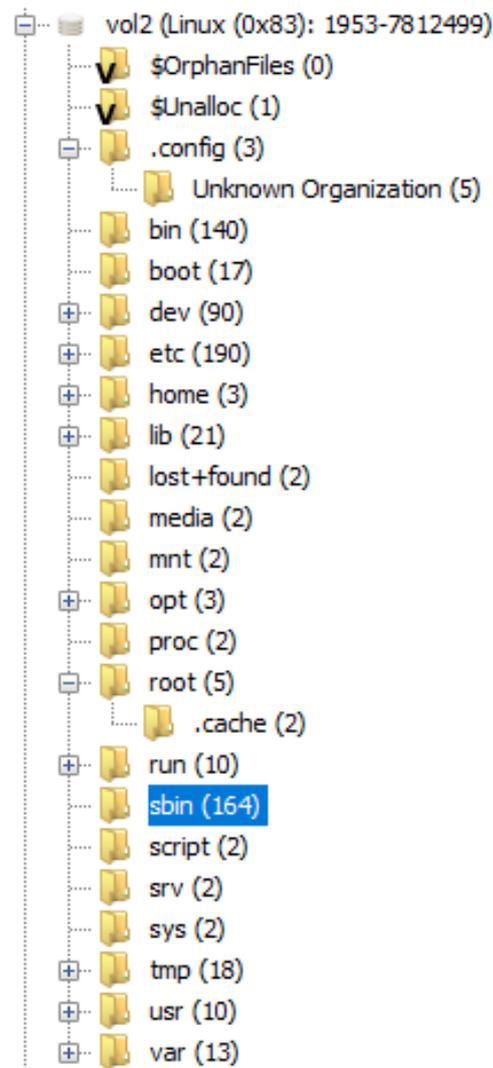
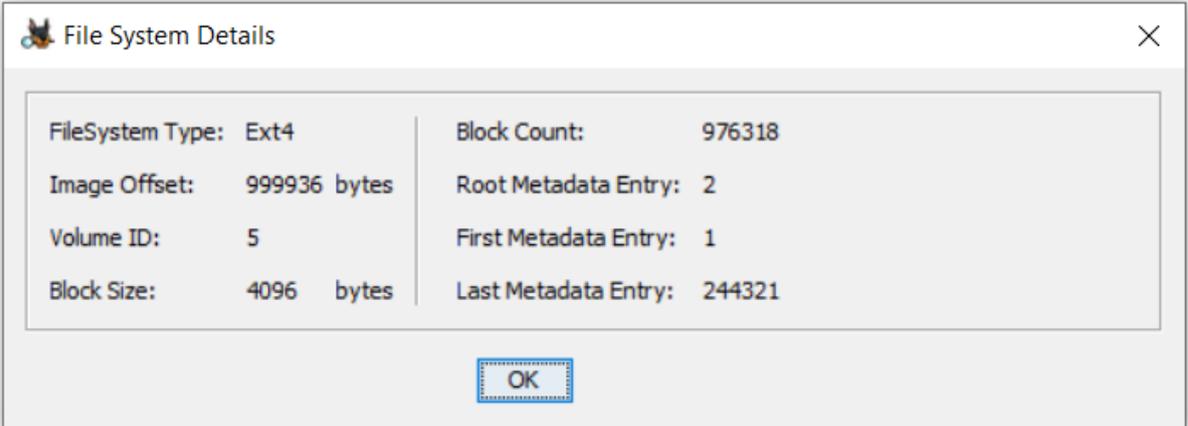


Figure 2. Root file system hierarchy.

Looking deeper, we analyzed the home directory, where we found a user called *phablet*. This appears to be the default user that all Ubuntu Touch systems come pre-installed with. The layout of the directory was the same as a traditional desktop installation of Ubuntu, refer to Figure 4. It is under the home directory that most of the phone's user data are stored. For example, during the data population, we would save images from email attachments or directly from the internet via a web browser. Searching for these downloaded images would lead us to this home directory. Under the home directory, we can see the standard structure of an Ubuntu desktop installed graphically as common Music (*/home/phablet/Music*), Documents (*/home/phablet/Documents*), and Downloads (*/home/phablet/Downloads*) folders are present among other common folders.

The International Mobile Equipment Identification (IMEI) number of the device was found by using the phone directly. Then, the image was searched through using Autopsy's ingest feature to see where the IMEI could be found within the image. However, the IMEI number could not be found using the keyword search ingest.

Name	ID	Starting Sector	Length in Sectors	Description
vol1 (Unallocated: 0-1952)	1	0	1953	Unallocated
vol2 (Linux (0x83): 1953-7812499)	2	1953	7810547	Linux (0x83)
vol3 (Unallocated: 7812500-124735487)	3	7812500	116922988	Unallocated



The screenshot shows a dialog box titled "File System Details" with a close button (X) in the top right corner. The dialog contains a table of file system metadata:

FileSystem Type:	Ext4	Block Count:	976318
Image Offset:	999936 bytes	Root Metadata Entry:	2
Volume ID:	5	First Metadata Entry:	1
Block Size:	4096 bytes	Last Metadata Entry:	244321

At the bottom of the dialog is an "OK" button.

Figure 3. The file system type for Ubuntu Touch OS as reported by Autopsy.

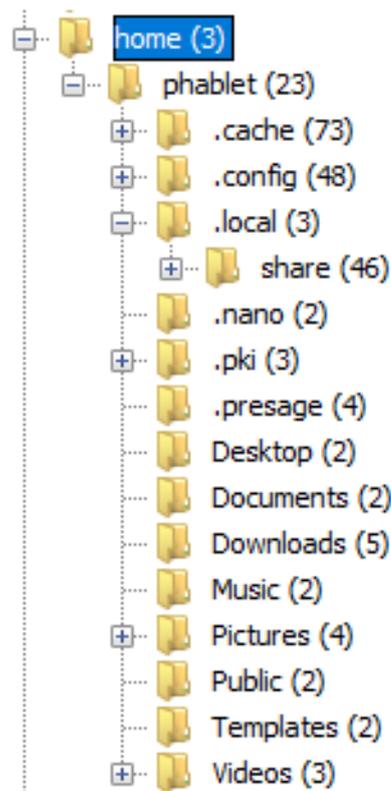


Figure 4. Main directory for analysis is the *phablet* users home directory.

4.3. Findings from Native Applications

Native apps here refer to those apps which come pre-installed with the Ubuntu Touch operating system. This section intends to cover some of the important native apps used during the testing. Table 3 provides a comprehensive summary of the directories and files of interest found during the investigation. Note that one interesting property of Ubuntu Touch is the folder `/home/phablet/.local/share/applications`. This folder contains files with the `.desktop` extension. It appears that this location is a list of icons that exist on the phone's

main application drawer. This information may be useful to investigators as a quick way to see what applications have been installed on the device.

Table 3. Application directories in the file system along with interesting sub-files.

S.No.	Native App Name	Directory in File Structure	Interesting Sub-Files
1.	Calendar	/home/phablet/.local/share/evolution/calendar/system/ calendar.ics	None
2.	Phone (Calls)	/home/phablet/.local/share/history-service/history.sqlite	Table:voice event under history.sqlite
3.	Phone (Contacts)	/home/phablet/.local/share/evolution/addressbook/system/contacts.db	Table: folder id, folder_id_phone_list under contacts.db
4.	Phone (Messages)	/home/phablet/.local/share/history-service/history.sqlite	None
5.	Morph Browser	home/phablet/.local/share/morph-browser/history.sqlite	bookmarks.sqlite, downloads.sqlite
6.	Gallery	home/phablet/Downloads	Image/Video Tab, gallery.sqlite
7.	Notes	/home/phablet/.local/share/com.ubuntu.reminders/@local11	note-ba9def24- 0882-4140-b5c5-abc733c38fda.info
S.No.	Third Party App Name	Directory in File Structure	Interesting Sub-Files
8.	Skype	home/phablet/.local/share/skype.rubencarneiro/	000003.log
9.	Google Mail	home/phablet/.local/share/googlemail.josele13	/databases/ https_mail.google.com_0/1.sqlite
10.	Pesbuk	home/phablet/.local/share/pesbuk.kugiigi	pesbuk.kugiigi /QTWebEngine/Default
11.	WebTelegram	home/phablet/.local/share/webtelegram.neothethird	00003.log
12.	Axolotl	/home/phablet/.local/share/textsecure.nanuc/	/storage/identity/http_password, db/db.sql
13.	LinkedIn	/home/phablet/.local/share/linkedin.andybleaden/	Local Storage/https_www.linkedin.com_0.localstorage
14.	Twitter	/home/phablet/.local/share/twitter.toshi/	000003.log
15.	YouTube	/home/phablet/.local/share/com.popey.youtube/	None
16.	uReadIt	/home/phablet/.local/share/ureadit.neothethird/	Databases/
17.	Keepweb	/home/phablet/.local/share/keepweb.cibersheep/	None
18.	Onion Browser	/home/phablet/.local/share/onion.nanuc.org/	None
19.	Drop Box	/home/phablet/.local/share/dropbox.andybleaden/	Local Storage/https_www.dropbox.com_0.localstorage

4.3.1. Calendar

The native calendar app of the phone was found pretty intuitively on its forensic examination. The file *calendar.ics* in the specified directory contains all events in the sequential order of their creation. Authors populated calendar events on the phone, two examples of which are shown in Figure 5. First an event by the title *Test event name* with its DTStart (Start Date), DTend (End Date), location, summary, description creation time, and last modified time. Second, an alarm set for 15 min by the name *test event name*. Recovering calendar events during digital forensics investigations could help the analyst know the suspect's planned activities and associated date/time information.

```

BEGIN:VCALENDAR
CALSCALE:GREGORIAN
PROPID: -//Ximian//NONSGML Evolution Calendar//EN
VERSION:2.0
X-EVOLUTION-DATA-REVISION:2020-04-08T06:58:15.187026Z(1)
BEGIN:VEVENT
UID:20200408T065733Z-29391-32011-2681-1@ubuntu-phablet
DTSTAMP:20200408T065733Z
DTSTART;TZID=/freeassociation.sourceforge.net/US/Eastern:20200415T030000
DTEND;TZID=/freeassociation.sourceforge.net/US/Eastern:20200415T040000
LOCATION:This is the event location
SUMMARY:Test event name
DESCRIPTION:Event description is here. 15 min reminder
CREATED:20200408T065733Z
LAST-MODIFIED:20200408T065733Z
BEGIN:VALARM
X-EVOLUTION-ALARM-UID:20200408T065733Z-29391-32011-2681-2@ubuntu-phablet
ACTION:DISPLAY
DESCRIPTION:Test event name
TRIGGER;VALUE=DURATION;RELATED=START:-PT15M
END:VALARM
END:VEVENT
    
```

Figure 5. Highlighted text depicts a test alarm event from *calendar.ics* file viewed in Notepad++.

4.3.2. Phone (Calls)

Phone calls are one of the most interesting events for an analyst. The table *voice_events* in the file *history.sqlite*, when viewed in the DB Browser for SQLite, provides a tabular representation of all incoming and outgoing calls. Figure 6 represents the findings. The first entry in the table snapshot is a test event. However, the second and third entries signify the outgoing call to the given phone number. The fourth entry shows the incoming call to the phone number used in the phone.

Filter	accountId	threadId	eventId	senderId	timestamp	newEvent	duration	missed	muteParticipant
1	ofono/ofono/quectelqmi_0	+13126429864	+13126429864:Sat Apr 11 12:30:39 2020	+13126429864	2020-04-11T16:30:39.999Z	1	-381706216	1	+13126429864
2	ofono/ofono/quectelqmi_0	3464012318	3464012318:Sat Apr 11 12:54:51 2020	self	2020-04-11T16:54:51.179Z	0	14	0	3464012318
3	ofono/ofono/quectelqmi_0	3464012318	3464012318:Sat Apr 11 12:56:55 2020	self	2020-04-11T16:56:55.508Z	0	65	0	3464012318
4	ofono/ofono/quectelqmi_0	3464012318	3464012318:Sat Apr 11 13:17:28 2020	+13464012318	2020-04-11T17:17:28.431Z	0	61	0	+13464012318

Figure 6. Voice Call Events in *history.sqlite* database I (as reported by Autopsy).

4.3.3. Phone (Contacts)

The *Contacts.db* file contains the contact list of the phone users. Upon viewing this database file in DB Browser for SQLite, it was found that all contacts are assigned a UID, which is part of the table *folder_id* (contains first name, last name). Using this UID, the table redirects to another table, *folder_id_phone_list*, to fetch the phone number of the user, which is not present in the *folder_id* table. Figures 7 and 8 show the three contacts added during our data population with the same UIDs.

Filter	uid	value
1	pas-id-5E8D8A0200000000	1111-111-111
2	pas-id-5E8D8A6100000001	(222) 222-2222
3	pas-id-5E91F6D100000000	(346) 401-2318

Figure 7. Voice call events in *history.sqlite* database II (as reported by Autopsy).

uid	Rev	file_as	file_as_localizec	nickname	full_name	given_name	an_name_locali	family_name	ily_name_locali	is_list	_show_ad
1 pas-id-5E8D8A0200000000	2020-04-08T08:23:31Z(2)	cena, john	BLOB	NULL	john cena	john	BLOB	cena	BLOB	0	0
2 pas-id-5E8D8A6100000001	2020-04-08T08:25:06Z(6)	hofstadter, penny	BLOB	NULL	penny hofstadter	penny	BLOB	hofstadter	BLOB	0	0
3 pas-id-5E91F6D100000000	2020-04-11T16:56:49Z(2)	yansi	BLOB	NULL	yansi	yansi	BLOB		000-00	0	0

Figure 8. Phone contacts stored in *Contacts.db* database file as reported by Autopsy.

4.3.4. Phone (Messages)

After phone calls, text messages are probably the second most searched location in any suspected phone. The table *text_events* under the file *history.sqlite* provides details on all incoming and outgoing messages. Figure 9 presents the findings. Note that the messaging feature was not fully functional during our data population. We were not able to send multimedia messages using MMS function because the application continued to crash when we tried.

accountId	threadId	eventId	senderId	timestamp	newEvent	message
1 ofono/ofono/...	1000000000	2020-04-11T1...	1000000000	2020-04-11T1...	0	vvm.mobile.att.net:5400?f=0&v=1010&m=7652..
2 ofono/ofono/...	1000000000	2020-04-11T1...	1000000000	2020-04-11T1...	0	vvm.mobile.att.net:5400?f=0&v=1010&m=7652..
3 ofono/ofono/...	broadcast:7a...	/quectelqmi_...	self	2020-04-11T1...	0	Test message from pinephone
4 ofono/ofono/...	broadcast:7a...	2020-04-11T1...	+13464012318	2020-04-11T1...	0	Got it. Thanks!

Figure 9. Phone text events stored in *history* database file shown in Autopsy.

4.3.5. Morph Browser

The default internet browser for Ubuntu Touch is Morph Browser. Artifacts of interest in this app include searches, downloads, bookmarks, history, etc. The browser directory maintains a different database for all these artifacts. Table 3 lists all such locations. Figure 10 represents the database view of *history.sqlite*, which includes the visited URL, timestamp (lastVisit column), and other related information. Access to such a wealth of data from the browser would help the investigation significantly.

url	domain	title	icon	visits	lastVisit
1 https://m.facebook.com/usnew...	facebook.com	U.S. News and World Report - Home Facebook	https://static...	3	1586338210
2 https://mobile.twitter.com/usne...	twitter.com	https://mobile.twitter.com/usnews	https://abs.t...	1	1586338209
3 https://duckduckgo.com/?q=Co...	duckduckgo.c...	Covid at DuckDuckGo	https://duckd...	1	1586338189
4 https://duckduckgo.com/?q=&ia...	duckduckgo.c...	DuckDuckGo — Privacy, simplified.	https://duckd...	1	1586338176
5 https://duckduckgo.com/?q=20...	duckduckgo.c...	200 pixels image at DuckDuckGo	https://duckd...	1	1586338165
6 https://duckduckgo.com/?q=20...	duckduckgo.c...	200 pixels image at DuckDuckGo	https://duckd...	1	1586338152
7 https://www.adorama.com/alc/...	adorama.com	Budget Cameras for Student Filmmakers - 42 West	https://www...	1	1586338133
8 https://duckduckgo.com/?q=20...	duckduckgo.c...	200 pixels image at DuckDuckGo	https://duckd...	1	1586338125
9 https://duckduckgo.com/?q=20...	duckduckgo.c...	200 pixels image at DuckDuckGo	https://duckd...	1	1586335058
10 https://duckduckgo.com/?q=20...	duckduckgo.c...	200 pixels image at DuckDuckGo	https://duckd...	1	1586245960
11 https://duckduckgo.com/?q=20...	duckduckgo.c...	200 pixels image at DuckDuckGo	https://duckd...	1	1586216501
12 https://m.facebook.com/	facebook.com	Facebook - Log In or Sign Up	https://static...	1	1586216489
13 https://www.whatsapp.com/	whatsapp.com	WhatsApp	https://static...	2	1586214950
14 https://play.google.com/store/a...	google.com	WhatsApp Messenger - Apps on Google Play	https://www...	1	1586214924
15 https://contacts.google.com/?hl...	google.com	Google Contacts	https://ssl.gs...	3	1586214891
16 https://mydevices.google.com/e...	google.com	My Devices	https://www...	1	1585880263
17 https://play.google.com/store/a...	google.com	Instagram - Apps on Google Play	https://www...	1	1585880009
18 https://accounts.google.com/si...	google.com	Google Play	https://www...	1	1585879988
19 https://play.google.com/store/a...	google.com	Instagram - Apps on Google Play	https://www...	1	1585879968
20 https://accounts.google.com/si...	google.com	Google Play	https://www...	1	1585879941
21 https://play.google.com/store/a...	google.com	Instagram - Apps on Google Play	https://www...	1	1585879925
22 https://www.instagram.com/ac...	instagram.com	Login • Instagram	https://www...	1	1585879913
23 https://duckduckgo.com/?q=ins...	duckduckgo.c...	instaaram loain at DuckDuckGo	httos://duckd...	1	1585879908

Figure 10. Morph Browser reported search history in *history.sqlite* database file shown in Autopsy.

4.3.6. Gallery (The Application Was Not Functional)

The gallery is deemed to be an intuitive app on any phone. Yet, Pinephone (Build #270) does not support the gallery function in its graphical form. When attempting to load an image into the gallery from elsewhere on the phone, the image is selected using the embedded file explorer. However, after selecting an image, the gallery does not load the image, and it remains empty. Repeated attempts, including restarting the device, did not change the outcome. However, Autopsy provides a very convenient way to look at the contents of the gallery populated through Downloaded Photos. The *Images/Video* tab sorts the images and/or videos in grouped folders. These folders are prioritized based on the density of hash hits and the number of images in the folder. We downloaded three images (see Figure 11) for this investigation, and the directory is listed in Table 3. The file *gallery.sqlite* has interesting subtables such as *MediaTable*, *AlbumTable*, and *PhotoEditTable*; however, they were not populated at the time of investigation due to the gallery's limited functionality.

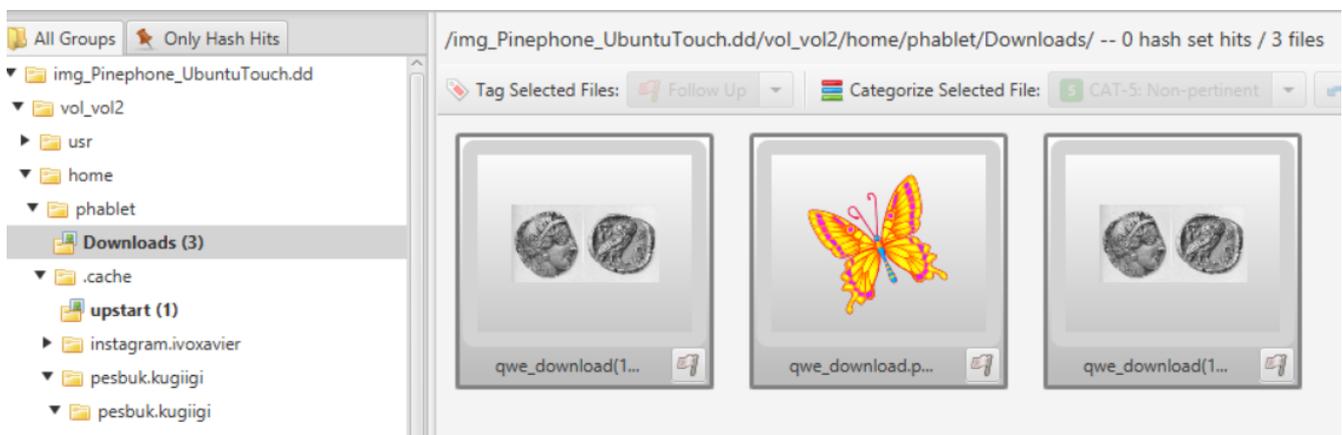


Figure 11. Autopsy-generated result of Gallery depicting downloaded images (images from all sources).

4.3.7. Bluetooth (The Application Was Not Functional)

Attempting to turn on Bluetooth by toggling the button slider would result in an approximately ten (10) second delay as it attempts to detect nearby Bluetooth devices. After the delay and no devices were discovered, the Bluetooth button slider would flip back into the off position without user interaction. Bluetooth information can still be read from the *syslog* file */var/log/syslog*, like any other Ubuntu desktop system. We ran the following command: `cat syslog|grep bluetooth > parsed_syslog.txt` to read the exported file and filter only references to Bluetooth. Looking through the new file, we can gain some information on Bluetooth, although it was not functional. No error information was found during the time frame we conducted testing, but we were able to find the current Bluetooth daemon version is v5.41 in Figure 12.

```
Apr 11 13:29:10 ubuntu-phablet bluetoothd[1784]: Bluetooth daemon 5.41
Apr 11 13:29:10 ubuntu-phablet bluetoothd[1784]: Starting SDP server
Apr 11 13:29:10 ubuntu-phablet bluetoothd[1784]: Bluetooth management interface 1.14 initialized
Apr 11 13:29:13 ubuntu-phablet NetworkManager[2129]: <info> [1586626153.5848] Loaded device plugin:
NMBBluezManager (/usr/lib/aarch64-linux-gnu/NetworkManager/libnm-device-plugin-bluetooth.so)
```

Figure 12. Content of created Log file for the Bluetooth App in Autopsy.

4.3.8. Notes

Another interesting app for the analyst could be Notes. PinePhone supports this feature but requires periodic updates from the OpenStore. We observed frequent crashes for the app, but the OS updates helped to restore its stable version. Figure 13 depicts the content of a note created in the phone and stored in the file *note-ba9def24-0882-4140-b5c5-abc733c38fda.info*.

Type	Value	Source(s)
Keyword Preview	tagline= title=This is	Keyword Search
Keyword	notes	Keyword Search
Keyword Search Type	0	Keyword Search
Source File Path	/img_Pinephone_UbuntuTouch.dd/vol_vol2/home/phablet/.local/share/com.ubuntu.reminders/@local/note-ba9def24-0882-4140-b5c5-abc733c38fda.info	
Artifact ID	-9223372036854552645	

Figure 13. Note App Directory and its content in Autopsy.

4.3.9. Music (The Application Was Not Functional)

During our data population, all attempts to load music kept crashing the app on startup. Examination of the music applications folder under the directory */home/phablet/.local/share/com.ubuntu.music/* shows only one folder called *Databases*. Within this folder, numerous SQLite databases can be found, as shown in Figure 14. These databases are unpopulated as we were unable to populate the app with music; however, the databases, in aggregate, contain tables named *playlist*, *queue*, *recent*, and *track*.

Name	MIME Type	S	C
[current folder]			
[parent folder]			
2be3974e34f63282a99a37e9e2077ee4.sqlite	application/octet-stream		
d332dbaaf4b3a1a7909b1d623eb1d02b.ini.lock	application/octet-stream		
d332dbaaf4b3a1a7909b1d623eb1d02b.sqlite-journal	application/octet-stream		
b01631344d4b1a36d6dec3179aff8784.sqlite	application/x-sqlite3		
d332dbaaf4b3a1a7909b1d623eb1d02b.sqlite	application/x-sqlite3		
d332dbaaf4b3a1a7909b1d623eb1d02b.ini.n17336	text/plain		
2be3974e34f63282a99a37e9e2077ee4.ini	text/x-ini		
b01631344d4b1a36d6dec3179aff8784.ini	text/x-ini		
d332dbaaf4b3a1a7909b1d623eb1d02b.ini	text/x-ini		

Figure 14. Autopsy results for Database files for the Music application.

4.4. Findings from Third-Party Applications

In this section, we will list and explain what artifacts and other notable information can be found in each of the applications that we tested. We will consider information to be notable if it gives insight into user activities, user communication, or could potentially help with attribution. A summary of our findings for the installed third-party applications is shown in Table 3. We will go more in-depth to explain each application’s available artifacts in later sections.

A common file found in the directory of most apps was a file titled *Cookies*. This is a database file that seems to store a multitude of different website cookies. As most of these applications are simple web apps that allow a user to access the respective service, such as Google Drive, these cookies are likely being used and collected by the web application to function. They are not abnormal; however, it is interesting to note which websites the web application has stored cookies as it may indicate which website the web app is visiting or pulling data from.

4.4.1. Axolotl Beta

Axolotl Beta is a popular Ubuntu Touch alternative to the Android- and iOS-based Signal app. Although our investigation revealed no user data, a few things of forensic

worth were still discovered. Note that the directory name does not seem to be correct; however, after looking under the `/opt` directory the app details were found in the file located at `/opt/click.ubuntu.com/textsecure.nanuc/0.7.7.2/click/status`, which verifies that we are looking at the right location. This information can be useful in checking the version of the app installed on the device, and potentially getting into contact with the app maintainer if further information is needed for an investigation.

First, an empty database file called `db.sql` was discovered under the `db` directory with the table setup to seemingly store user data in Figure 15. Second, a file named `http_password` was found under the `.storage/identity` directory. The content of the file (see Figure 16) seems to be encrypted. Going by the name of the file itself, it would appear that this is a password of some kind; however, its true purpose can only be speculated.

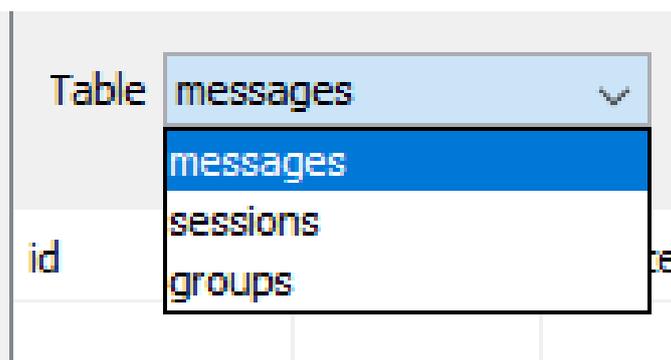


Figure 15. Axolotl Beta App reported empty database file in Autopsy.

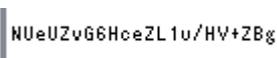


Figure 16. Encrypted content that is possibly a password shown in Autopsy.

4.4.2. Skype

Skype conversations can be viewed in clear plaintext using Autopsy at `/home/user/phablet/.local/share/skype.rubencarneiro/IndexedDB/https_web.skype.com_0.indexeddb.leveldb/000003.log`. Figure 17 represents the chat between the two test users. However, our concern is that it is merely a log file. The file is also missing the videos and images exchanged during the chat. Directory path and the filename of this forensically interesting log file are given in Table 3 for the Skype app.

```
Thank you. It has definitely
garnered a lot of attention in the past few years. Both of the shows
have won a lot of Emmy Awards too."
composetime"
2020-04-06T22:26:34.709Z{
cuid"
8775021594881247940"
conversationId"
8:live:yansi.keim"
createdTimeN
creator"
8:live:.cid.fe8a9e5928fb86b8"
composeTimeNP
content"
```

Figure 17. Results of Skype chat content.

4.4.3. LinkedIn

LinkedIn is an essential app that may yield information like personal identifiable information (name, email, phone number), search history, billing information (only on premium accounts), education history, and professional connections. For the purpose of data population, the authors communicated with multiple accounts and sent files. Files downloaded from the application were stored under the *Downloads* directory (see Figure 18). Besides, downloads also seem to be recorded in an SQLite database (see Figure 19) located in the *LinkedIn* folder itself. It is not known if the downloaded files that were deleted would continue to exist within this database; however, if they do, it would be a valuable artifact location to be aware of.

△ Name	S	C	O	Modified Time	Change Time	Access Time	Created Time
Say it in 6.docx			1	2020-04-06 20:19:41 EDT	2020-04-06 20:19:41 EDT	2020-04-06 20:19:41 EDT	2020-04-06 20:19:41 EDT
Say it in 6.docx.download			1	2020-04-06 20:19:41 EDT	2020-04-06 20:19:41 EDT	2020-04-06 20:19:41 EDT	2020-04-06 20:19:41 EDT

Figure 18. LinkedIn web app's Location of downloaded files in Autopsy.

△ downloadId	url	path	mimetype
15862187811371		/home/phablet/.local/share/linkedin.andyleaden/Downloads/Say it in 6.docx	application/vnd

Figure 19. LinkedIn web app's "Downloads" database file in Autopsy.

The *Local Storage* directory contains SQLite databases that contain some forensically valuable information. However, one table called *ItemTable* within the *https_www.linkedin.com_0.localstorage* file contains binary data that can be read directly using a database browser. Figure 20 shows the table exported out of the image and loaded using DB Browser for SQLite. Selecting the BLOB information reveals that the Binary Large Object (BLOB) contains legible text (see Figure 21). The legible text can be transcribed as "yung han yoon", "sheldon cooper", and "connect". It is not known what the numbers interwoven with these strings "158621838720" represent as converting it under the assumption that it is a UNIX epoch timestamp leads to a nonsensical answer; however, given the population steps taken, we suspect that the presence of these three strings is the log of the conversation using LinkedIn's chat features. The conversation occurred between the account on the PinePhone, Sheldon Cooper, and the researcher's personal account.

The *Service Worker/Cache Storage* directory contains potentially useful information as well. The organization of this storage is unknown; however, we discovered certain web elements of the LinkedIn site are stored here under randomly named directories. Of the various cached web page elements found, only one had any forensic potential. We observed that one of the accounts we had followed was listed in the cache, as well as the fact that we are currently following them, as shown in Figure 22.

Table: ItemTable	
key	value
Filter	Filter
1 feedLastUpdatedTime	BLOB
2 isUserVerified	BLOB
3 notificationPermission	BLOB
4 K}zK~v~svKIAIGIBIDD:REGISTRATION_FLOW_PROFILE_VISIBILITY	BLOB
5 badges	BLOB
6 voyager-web:new-tab-beacon	BLOB
7 voyager-web:rt-last-active-tab	BLOB
8 voyager-web:launchpad_show_done_message	BLOB
9 C_C_M	BLOB
10 PT_C_M	BLOB
11 voyager-web:profile-actions-platform	BLOB

Figure 20. Table entry reported by DB Browser for SQLite containing previous actions taken on the LinkedIn web app.

```
[{"id": "y.u.n.g.-.h.a.n.-.y.o.o.n.-.6.6.9.1.4.3.1.2.7.s.h.e.l.d.o.n.-.c.o.o.p.e.r.-.2.0.9.9.a.b.l.a.6.c.o.n.n.e.c.t.", "username": "sheldoncooper"}]
```

Figure 21. Usernames of LinkedIn user and contacts visible as shown in DB Browser for SQLite.

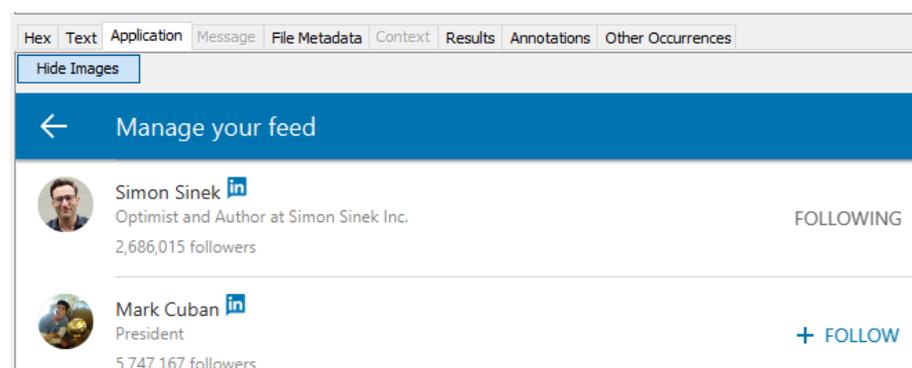


Figure 22. Cached web page elements depicting user following status shown in Autopsy.

4.4.4. Twitter

Similar to the LinkedIn app, the downloaded files can be found under the *Downloads* directory, while an entry for the downloaded file can be found under the *downloads.sqlite* database. The log file at *IndexedDB/https_mobile.twitter.com_0.indexeddb.leveldb/000003.log* contains publicly accessible links to the profile picture of the user of the application, as shown in Figures 23 and 24. Other than this image, no other information of note was found. Other information such as followers and followed people or even a Twitter username was not found. %clearpage

```
Sheldon Cooper"
profile_image_url_https`https://pbs.twimg.com/profile_images/1247309888818208768/sIUqxiRR_normal.jpg"
result_context "
```

Figure 23. Log file containing URL to the user’s Twitter profile as shown in Autopsy.

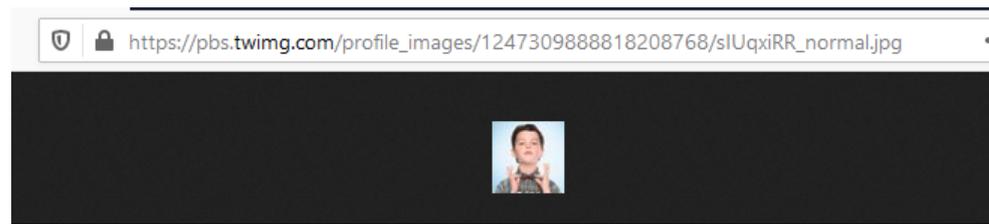


Figure 24. Navigating to the URL shows you a thumbnail of the user's profile image in web browser.

4.4.5. Youtube

No artifacts of user data with forensic value were found. None of the searches or evidence of the videos that were watched could be found, nor the Google account used to login to the app.

4.4.6. UReadIt

During the data population, we did not associate any account with this application as Reddit can be browsed anonymously. However, two SQLite databases with random names under the directory *Databases* were found. They contained tables that appear to be prepared to store user account data, as shown in Figure 25. No other information was found, such as the posts and Reddit threads that were opened.

Name	type	Schema
Tables (2)		
ActiveRedditUser		CREATE TABLE ActiveRedditUser(username TEXT UNIQUE)
RedditUsers		CREATE TABLE RedditUsers(username TEXT UNIQUE, passwd TEXT, subscribed TEXT)

Figure 25. Reddit Database Tables which could store user account data shown in DB Browser for SQLite.

4.4.7. Keepweb

During our analysis, we could not find any artifacts containing user data with forensic value. Given the nature of this application in particular, and the fact that it stores user passwords as its primary purpose, this should be considered as positive from the security and privacy point of view. This means the saved passwords are not vulnerable to being stolen or cracked.

4.4.8. Onion Browser

Onion Browser is one of the most important applications that might yield significant forensic artifacts due to its popularity among criminals. The file found at `/opt/click.ubuntu.com/onion.nanuc.org/0.3/share/tor/torrc` contains an interesting string content that seems to specify a data directory as `/home/phablet/.local/share/onion.nanuc.org/.tor` (see Figure 26).

```
SOCKSPort 9050
DataDirectory /home/phablet/.local/share/onion.nanuc.org/.tor
```

Figure 26. String located in installation folder for browser pointing to potential user data storage location shown in Autopsy.

When we investigated that location, we found many cached files. The cached files are shown in Figure 27 that mainly contain string and text data as well as public RSA keys. We cannot conclude for certain what services these public RSA keys belong to; however, it is highly likely that these are the public keys for TOR nodes available to the public to form TOR circuits. That being the case, these keys are likely not very forensically interesting.

Name	S	C	Modified Time	Change
state.tmp			2020-04-07 04:16:30 EDT	2020-04
state			2020-04-07 04:16:30 EDT	2020-04
lock			2020-04-07 03:55:44 EDT	2020-04
dialog.xlb.dpkg-new			2020-04-02 22:05:19 EDT	2020-04
cached-microdescs.new			2020-04-07 03:55:50 EDT	2020-04
cached-microdescs			2020-04-07 03:55:46 EDT	2020-04
cached-microdesc-consensus.tmp			2020-04-07 03:55:48 EDT	2020-04
cached-microdesc-consensus			2020-04-07 03:55:48 EDT	2020-04
cached-certs			2020-04-07 03:56:46 EDT	2020-04

Figure 27. Cache files found at the “DataDirectory” location using Autopsy.

The files that we downloaded during the data population process were not listed within the *downloads.sqlite* database file, nor any websites visited in the *history.sqlite* file. This information is likely not readily available or could be stored in different places. It is important to note that the image of a Greek drachma that was downloaded via the TOR browser has been saved directly to the *Downloads* directory under the user’s home directory. This means that this TOR browser on Ubuntu Touch does not, by default, save images to a separate, non-standard directory.

4.4.9. Dropbox

Dropbox is one of the most commonly used cloud storage applications. Therefore, it is essential to understand the structure of the data being stored, if any, by this app. Dropbox’s *downloads.sqlite* database file stored under the main directory was found to be empty. The *Local Storage* directory with multiple SQLite databases that contain BLOB files is shown in Figure 28. Examining the BLOBs within the SQLite browser reveals mostly random numbers, non-intelligible data, or data that did not seem important. The actual files that were uploaded to the cloud were not available.

Furthermore, the *Cookies* database that we found contains interesting artifacts. Cookies can be found from LinkedIn, Google, Facebook, Yahoo, and even Youtube. We are not certain why these cookies from websites unrelated to Dropbox can be found in this file. This behavior is seen across almost all web apps. There is no evidence of any other files that were available from Dropbox on the local image.

https_accounts.google.com_0.localstorage	application/x-sqlite3
https_cdn.krxd.net_0.localstorage	application/x-sqlite3
https_marketing.dropbox.com_0.localstorage	application/x-sqlite3
https_www.dropbox.com_0.localstorage	application/x-sqlite3
https_www.google.com_0.localstorage	application/x-sqlite3

Figure 28. Multiple SQLite databases for Dropbox reported in Autopsy.

4.4.10. Pesbuk

The authors attempted to install Facebook on the Ubuntu Touch OS. However, we found Pesbuk, which is a Facebook alternative available on the OpenStore. The app’s usability provides a Facebook-like interface and functionality to its users. On the other hand, its investigation did not yield significant results except for *Cookies* values (see Figure 29).

These values store a cookie’s creation date, path, expiration date, last access date and time, if already expired (1 for yes, 0 for no), and the data encryption method. These cookies may be useful for examining when a user either installed the app, or when it may have been used.

	creation_utc	host_key	name	value	expires_utc	last_access_utc
	Filter	Filter	Filter	Filter	Filter	Filter
1	13230690276786063	.facebook.com	datr	5L6LXmZnSH4QriPigLuD-hA7	13293762276954632	13230690520188815
2	13230690276955136	.facebook.com	sb	5L6LXkfzZDAyOWWVHdRiH0P2	13293762367629873	13230690550337503
3	13230690277342200	.facebook.com	m_pixel_ratio	1	0	13230690535254702

Figure 29. Some Cookies values found in Pesbuk App as found by DB Browser for SQLite.

4.4.11. WebTelegram

A common observation regarding apps for Ubuntu Touch is that it supports the web versions of their original apps; Telegram is one of them. While there is no desktop version available on OpenStore yet, the web version still gives the same functionality in terms of usability and investigation. All images go under the location *webtelegram.netothethird/IndexeDB /https_web.telegram.org_0.indexeddb.blob/1/00*. Figure 30 represents one of those pictures. Again, all these pictures can be viewed in Image/Video tab more conveniently. Some parts of the chats can be viewed in the file *webtelegram.netothethird/IndexeDB /https_web.telegram.org_0.indexeddb.leveldb/00003.log* as well.

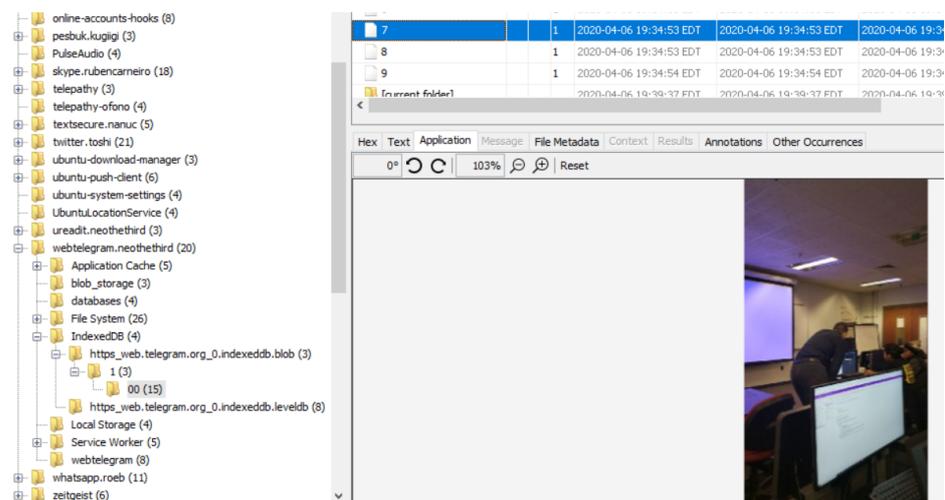


Figure 30. Directory of Telegram Pictures fetched from Autopsy.

4.4.12. Google Mail

Google Mail is one of the most frequent mailing apps and is found to be of primary interest to analysts. Our investigation for this app was quite similar to regular apps and rather easy. However, its directory is more sorted and has further directly named sub-folders such as *Databases*, *Downloads*, and *Application Cache*. Moreover, chat, conversations, attachments, and email content are found in a database file titled as *1* stored under */home/phablet/local/share/googlemail.josele13/databases /https_mail.google.com_0/1*. Figure 31 shows the contents of table *cached_conversation_headers* stored under the database file named *1*. Images downloaded as part of the exchanged emails are stored under the *Downloads* directory of the phone and can be viewed through *Image/Videos* Tab of Autopsy. These findings from Google Mail application would be indispensable for the forensic investigators when suspects communicate and share images using this app.

rel	subject	snippetHtml	senderListHtml	num
	Filter	Filter	Filter	Filter
1	Do more with your phone. Add a way to p...	A wider range of apps, games, and more is a tap away.	Google Pl...	1
2	focyber86, see Ciara, Lewis Hamilton and ...	Follow Ciara, Lewis Hamilton and others you know to se	Instagra...	1
3	Manage your storage and save data	See app updates, control your storage, and more. Googl	Google Play	1
4	Earn points and get rewards on Google Play	Google Play Points Check out Google Play Points Check o	Google Pl...	1
5	Learn how Google Play protects you.	Your security is our biggest priority. Google Play We	Google Pl...	1
6	Sheldon, welcome to Google Play	Find all the apps and games you need. Google Play Welc	Google Pl...	1
7	Sheldon: your job alert for Cyber Security ...	See results for your search query 'Cyber Security S	LinkedIn ...	1
8	Sheldon, your pin is 422117. Please confir...	LinkedIn Thanks for signing up. Please confirm your ema	LinkedIn Mes...	1
9	Subject line: Sending image	Body of Image sending	Somebody In ...	1
10	Subject Line: Sending mp3	Body of MP3 email	Somebody In ...	1
11	Subject line: Sending MP4	Body of MP4 email message	Somebody In ...	1

Figure 31. Gmail conversation of test account fetched from Autopsy and viewed on DB Browser.

5. Research Challenges

This research is the first of its kind, so naturally we faced a series of challenges during the study. Interestingly, GitLab [23], the official project management by UBPorts, brings researchers, developers, and enthusiasts together to report their issues and find the existing ones. We initiated more than five issues on the website and received a great response in return, which eventually helped in bettering our research.

Instagram and Google Drive were among some of the applications that we attempted to install and populate, but we were unsuccessful. Usually, the applications are installed and then run. However, in our case, the application would either crash back to the home page of Ubuntu Touch, or the application screen would be completely blank during the application start-up. This is highly likely because the application may not have been updated to work with the build of Ubuntu Touch for PinePhone that we utilized. However, there is a good chance that they will store data in ways similar to what we have previously seen in other applications. Another challenge we faced is that the IMEI number of the phone can be found by going to the "About" page in the settings app. However, using a keyword search of the known IMEI number over the image did not reveal where it may be stored.

6. Discussion

Going over the investigation, we found some similarities that are generally reported in an Android-based OS investigation. Here, we discuss some advantages and disadvantages posed by Ubuntu Touch and PinePhone to digital forensics.

- The previous literature on the analysis of apps (based on Android) helped authors target specific data points, directories, and formats, thus saving crucial examination time. However, the analysis of apps on Ubuntu Touch still gave us expected yet unique data points, directories, and formats.
- The rooting process may be bypassed in the case of the PinePhone, either because it comes rooted, or because the acquisition process via SDcard does not require root access. Whereas other Android devices, especially newer models, require the device to be rooted before accessing specific application data, the Pinephone does not require such procedures.
- As the data are stored on the SD card itself, it is also possible to do a dead-box acquisition of the device by creating a bit stream image of the device [24]. Whereas other Android devices usually require the device to be run so that ADB, or other methods, can be used to pull data, the fact that data are stored on the SD card means data can be acquired without requiring the device to be turned on. The common criticism of possible data corruption or unexpected data manipulation that accompanies mobile forensic acquisitions can be avoided here as the device does not need to be turned on for data to be acquired.

- Some disadvantages are clearly present as well. With the Ubuntu OS still being in development, and the fact that full disk encryption is not even available, there is a possibility that future versions of the PinePhone may prove to be much more secure and forensically challenging.

Overall, the advantages presented by the PinePhone are that more traditional forensics procedures can be used, and rooting the device is no longer a necessity.

7. Conclusions and Future Work

This research is focused on understanding and investigating Ubuntu Touch OS deployed on the PinePhone from a new phone manufacturer. PinePhone is available only at \$149.99 in the market as of December 2020. The motivation behind this research is the phone's lower market share and a new and yet robust OS, which also provides customization options and control to its users. This combination might lure an attacker into spending less on technology yet still conduct their malicious actions and cheaply dispose of the physical device.

In our attempt to investigate, not much information can be gained yet from third-party applications. This may be due to the web application's use of mainly online storage, and the fact that the OS, Ubuntu Touch, as well as our hardware platform, Pinephone, are still very much in development. However, we have confirmed that the Ubuntu Touch has a file system very similar to that of Ubuntu.

A quick analysis of the system log files reveals that Bluetooth and likely many other "desktop" oriented services likely function similarly to that of Ubuntu's desktop distribution, meaning a lot of what is known in Ubuntu, or more generally Linux forensics, can most likely be applied here. We have identified data that are highly sought after by investigators, such as contacts and file locations, and where they are stored within the file system. What is currently available to investigators from third-party apps has been noted.

Future research can still be done on certain native apps that are semi-functional in Build #270. These apps include Gallery, Videos, Bluetooth, Camera, Downloads, and Location Services. Since it is time-sensitive research, it is believed that once developers have a stable version of these apps, researchers would have a wider scope to investigate.

With an already wider scope to investigate in Ubuntu Touch, it is recommended to put a "Relevance Rating" with each app's findings. These ratings would help an analyst to sift through the important artifacts, which could be beneficial in a time-sensitive investigation. Categories could be low relevance, intermediate relevance, or high relevance. For example, outgoing call events are of high relevance in any investigation and comparatively more important than logs from a video game.

Author Contributions: The authors of this paper have contributed to this work in the following ways. Conceptualization, Y.K., Y.H.Y. Data curation, Y.K., Y.H.Y. Formal analysis, Y.K., Y.H.Y. Investigation, Y.K., Y.H.Y. Methodology, Y.K., Y.H.Y. Project administration, Y.K., Y.H.Y. Resources, Y.K., Y.H.Y. Supervision, U.K. Validation, Y.K., Y.H.Y. Visualization, Y.K., Y.H.Y. Writing—original draft, Y.K., Y.H.Y. Writing—Review and editing, Y.K., Y.H.Y., U.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors would like to thank Pine64 and its community for supporting our research, as well as providing us with a Braveheart PinePhone to conduct our research.

Conflicts of Interest: The authors declare no conflict of interest. The Pine64 manufacturers who provided the hardware for this study had no role in the design; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

ADB	Android Debug Bridge
BLOB	Binary Large Object
DB	Database
EXT4	Fourth Extended File System
FHS	Filesystem Hierarchy Standard
FTK	Forensic Toolkit
GUI	Graphical User Interface
IMEI	International Mobile Equipment Identity
iOS	iPhone Operating System
LSB	Linux Standard Base
MMS	Multimedia Messaging Service
OEM	Original Equipment Manufacturer
OS	Operating System
RSA	Rivest, Shamir, and Adleman (encryption technique)
SD	Secure Digital
SIM	Subscriber Identity Module
SQL	Structured Query Language
SQLite	Structured Query Language Lite
TOR	The Onion Router
UID	User Identification
UNIX	UNiplexed Information Computing System
URL	Uniform Resource Locator

References

1. Pandey, A.K.; Tripathi, A.K.; Kapil, G.; Singh, V.; Khan, M.W.; Agrawal, A.; Kumar, R.; Khan, R.A. Current Challenges of Digital Forensics in Cyber Security. In *Critical Concepts, Standards, and Techniques in Cyber Forensics*; IGI Global: Hershey, PA, USA, 2020; pp. 31–46.
2. Shimmi, S.S.; Dorai, G.; Karabiyik, U.; Aggarwal, S. Analysis of iOS SQLite schema evolution for updating forensic data extraction tools. In Proceedings of the 2020 8th International Symposium on Digital Forensics and Security (ISDFS), Beirut, Lebanon, 1–2 June 2020; pp. 1–7.
3. Andriotis, P.; Tryfonas, T. Impact of User Data Privacy Management Controls on Mobile Device Investigations. In *Advances in Digital Forensics XII*; IFIP Advances in Information and Communication Technology; Springer International Publishing: Cham, Switzerland, 2016; Volume AICT-484, pp. 89–105.
4. Tzvetanov, K.; Karabiyik, U. A first look at forensic analysis of sailfishos. *Comput. Secur.* **2020**, *99*, 102054. [CrossRef]
5. PinePhone. 2020. Available online: <https://wiki.pine64.org/index.php/PinePhone> (accessed on 25 January 2020).
6. Ubuntu on PinePhone. 2020. Available online: <https://www.omgubuntu.co.uk/2020/01/ubuntu-touch-on-the-pinephone-is-coming-along-nicely> (accessed on 30 January 2020).
7. Ubuntu Touch. 2020. Available online: <https://ubports.com/> (accessed on 12 February 2020).
8. About Ubuntu Touch. 2020. Available online: <https://ubuntu-touch.io/> (accessed on 11 February 2020).
9. The Directory Tree. Available online: <https://help.ubuntu.com/lts/installation-guide/armhf/apcs02.html> (accessed on 13 February 2020).
10. Filesystem Hierarchy Standard. Available online: https://refspecs.linuxfoundation.org/FHS_3.0/fhs-3.0.pdf (accessed on 9 March 2020).
11. Patil, D.N.; Meshram, B.B. Digital forensic analysis of ubuntu file system. *Int. J. Cyber Secur. Digit. Forensics* **2016**, *4*, 175–186. [CrossRef]
12. Fairbanks, K.D.; Lee, C.P.; Owen, H.L., III. Forensic implications of ext4. In CSIIRW '10, Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research, Oak Ridge, TN, USA, 21–23 April 2010; pp. 1–4.
13. Fairbanks, K.D. An analysis of Ext4 for digital forensics. *Digit. Investig.* **2012**, *9*, S118–S130. [CrossRef]
14. Autopsy for Forensics. 2020. Available online: <https://www.autopsy.com/> (accessed on 1 April 2020).
15. Schaefer, T.; Höfken, H.; Schuba, M. Windows phone 7 from a digital forensics' perspective. In Proceedings of the International Conference on Digital Forensics and Cyber Crime, Dublin, Ireland, 26–28 October 2011; pp. 62–76.
16. Hazra, S.; Mateti, P. Challenges in android forensics. In Proceedings of the International Symposium on Security in Computing and Communication, Manipal, India, 13–16 September 2017; pp. 286–299.
17. Casey, E.; Cheval, A.; Lee, J.Y.; Oxley, D.; Song, Y.J. Forensic acquisition and analysis of palm webOS on mobile devices. *Digit. Investig.* **2011**, *8*, 37–47. [CrossRef]

18. Coelho, N.M.; Peixoto, M.; Cruz-Cunha, M.M. Prototype of a paranoid mobile operating system distribution. In Proceedings of the 2019 7th International Symposium on Digital Forensics and Security (ISDFS), Barcelos, Portugal, 10–12 June 2019; pp. 1–6.
19. Yusoff, M.N.; Mahmud, R.; Dehghantanha, A.; Abdullah, M.T. Advances of mobile forensic procedures in Firefox OS. *Int. J. Cyber-S Secur. Digit. Forensics* **2014**, *3*, 183–199. [[CrossRef](#)]
20. Lin, I.L.; Yen, Y.S.; Chang, A. A study on digital forensics standard operation procedure for wireless cybercrime. In Proceedings of the 2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Seoul, Korea, 30 June–2 July 2011; pp. 543–548.
21. Riadi, I.; Umar, R.; Firdonsyah, A. Identification Of Digital Evidence On Android’s Blackberry Messenger Using NIST Mobile Forensic Method. *Int. J. Comput. Sci. Inf. Secur. (IJCSIS)* **2017**, *15*, 3–8.
22. Download balenaEtcher. 2020. Available online: <https://www.balena.io/etcher/> (accessed on 30 January 2020).
23. GitLab by UBPorts. 2020. Available online: <https://gitlab.com/ubports/community-ports/pinephone> (accessed on 4 April 2020).
24. Kent, K.; Chevalier, S.; Grance, T.; Dang, H. Guide to Integrating Forensic Techniques into Incident Response. 2006. Available online: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-86.pdf> (accessed on 15 August 2006).