2021

Forensic Analysis of Tor Browser on Windows 10 and Android 10 Operating Systems

Muhammad Raheel Arshad

Mehdi Hussain

Hasan Tahir

Sana Qadir

Faraz Iqbal Ahmed Memon

See next page for additional authors

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 17, 2021, accepted October 4, 2021, date of publication October 13, 2021, date of current version October 22, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3119724

Forensic Analysis of Tor Browser on Windows 10 and Android 10 Operating Systems

MUHAMMAD RAHEEL ARSHAD[1], MEHDI HUSSAIN[1], (Member, IEEE), HASAN TAHIR[1], (Senior Member, IEEE), SANA QADIR[1], FARAZ IQBAL AHMED MEMON[1], AND YOUSRA JAVED[2]

[1]School of Electrical Engineering and Computer Science, National University of Sciences & Technology (NUST), Islamabad 44000, Pakistan
[2]School of Information Technology, Illinois State University, Normal, IL 61761, USA

Corresponding author: Mehdi Hussain (mehdi.hussain@seecs.edu.pk)

ABSTRACT Smartphones and Internet have become prevalent in our society with various applications in businesses, education, healthcare, gaming, and research. One of the major issues with the Internet today is its lack of security since an eavesdropper can potentially intercept the communication. This has contributed towards an increased number of cyber-crime incidents, resulting in an increase in users' consciousness about the security and privacy of their *communication*. One example is the shift towards using *private browsers* such as Tor. Tor is a well-recognized and widely used privacy browser based on *The Onion Router network* that provisions anonymity over the insecure Internet. This functionality of Tor has been a major hurdle in cybercrime investigations due to the complex nature of its anonymity. This paper investigates artifacts from the Tor privacy browser on the latest Windows 10 and Android 10 devices to determine potential areas where evidence can be found. We examine the registry, storage, and memory of Windows 10 devices and the memory, storage, logs, and Zram of Android 10 devices for three possible scenarios i.e. before, during, and after use of the Tor browser. Our results do not support the claims made by the Tor Project regarding user privacy and anonymity. We find that it is possible to retrieve significant details about a user's browsing activities while the Tor browser is in use as well as after it is closed (on both operating systems). This paper also provides an investigative methodology for the acquisition and analysis of Tor browser artifacts from different areas of the targeted operating systems. Therefore, it can serve as a base to expand research in the forensic analysis of other privacy browsers and improve the efficiency of cybercrime investigations efficiency.

INDEX TERMS Tor, browser forensic, windows 10, windows forensic, android forensic, privacy, android 10, anonymous browser.

I. INTRODUCTION

The prevalence of workstations, laptops, and smartphones is increasing on a daily basis. These devices have now become a lifeline of our society. Since its introduction back in 1994, the Simon Personal Communicator (SPC) created by IBM emerged as the first smartphone and. Later then in 2007, Apple Inc. become became the first modern smartphone manufacturer with their iPhone running a proprietary mobile operating system iOS. These devices offered consumers the ability to browse the web just as they would do on a desktop computer. Android was the next mobile operating system to be officially introduced in 2008.[1] It immediately became a popular platform in the smartphone market due to its open-source license and the availability of a wide range of applications. In the case of computers and laptops, Microsoft Windows became the first choice for the user because it came pre-loaded with necessary software(s), a feature-rich user-friendly graphical user interface, and provided a much wider driver and peripheral compatibility.[2] As of the first quarter of 2020-21, Android shares 71.81% of the worldwide

The associate editor coordinating the review of this manuscript and approving it for publication was Aniello Castiglione.

[1]Android versions: A living history from 1.0 to 12. Computerworld, JR Raphael, https://www.computerworld.com/article/3235946/android-versions-aliving- history-from-1-0-to-today.html [2021 March 12].

[2]Why windows os is popular than linux and mac for Desktop and Laptop? ourtechroom.com, DiwasPoudelhttps://ourtechroom.com/tech/why-windows-ospopular- than-linux-for-desktop-laptop[2020 September 20].

smartphone market and Windows shares 75.55% of the worldwide PC market.[3] Laptops and smartphones have been purchased in an almost equal ratio in 2019, 2020, and 2021.[4] On one hand, this widespread adoption of Android smartphones and laptops creates an opportunity for businesses and industries to expand their productivity and resources. But, on the other hand, it has created problems for law enforcement agencies and other Internet users because it has provided more mobility and agility to cybercriminals, enabling them to launch sophisticated cyber-attacks. One such problem is the anonymity that enables individuals to engage in illicit activities without revealing themselves and/or their actions to others because they are constantly able to cover their tracks [1]. They are also able to maintain anonymity over the public network owing to the use of VPNs and other privacy protection software.

Tor privacy browser is one such privacy protection software that is widely used for anonymity by both ordinary users and cyber-criminals. For the common user, the aim is to provide privacy protection on the insecure Internet while cyber-criminals use it to cover their tracks[5] while carrying out illegal activities. Tor browser works by directing encrypted traffic via an overlay of layered networks [2]. The digital investigation of a Tor network is a complex and tedious task. However, things can be simplified by investigating a seized suspect device (mobile or PC) to look for traces of illicit online activities.

The evolution of operating systems and application development technologies has posed considerable challenges in conducting digital investigations which serves as the motivation for our work. Although several studies have been conducted on forensic analysis of the Tor privacy browser, they have focused on the older versions of Android and Windows platforms with limited browsing activities. These studies are also limited in their examination of storage, registry, and ADB logs. No single study analyzes the Tor privacy browser on both Windows and Android systems. In this study, we undertake the forensic analysis of the current version of the Tor privacy browser on the latest builds of two different operating systems. As per our knowledge, the targeted builds of the Tor privacy browser, Windows OS, and Android OS have not been explored yet.

In this research, we design and simulate a dark web cybercrime scenario and then acquire and analyze evidence of the Tor browser from both operating systems and try to identify the suspect's online activities. More specifically, we aim to better understand the following questions:

- What methods are used for the collection of evidence?
- What kind of challenges can be faced?
- What kind of evidence can be extracted?

The remainder of this paper is organized as follows. Section 2 contains the work related to the forensic investigation of the Tor browser. Section 3 outlines the methodology for this study and section 4 explains the evidence acquisition for Windows 10 and Android 10 OS. Section 5 provides the findings from both devices while section 6 provides a comparison with existing research. Section 7 highlights the recommendations for Tor project developers and Section 8 presents the conclusion and directions for future work.

II. RELATED WORK

In this section, we will discuss the related work. In [3], a study was conducted to examine Orweb (now called Tor browser for mobile) browsing sessions on Samsung Galaxy S2 running Android. The device was examined in both rooted and unrooted states for the Tor privacy browser. It was concluded that browsing sessions were recovered only on rooted devices. Meanwhile, the selected version of Android was too old 2.3.3 as compared to the latest Android 10. In [4], a similar study was conducted on Samsung Galaxy S2 running Android 4.1.1. It proposed that there is no need to root the device as evidence can also be obtained by flashing the custom recovery on the device and then acquiring an image of the device's flash memory. Although, this method proves to be very useful from a forensic point of view but again this custom flashing recovery method is different on the latest devices.

In [5], the researchers performed a thorough analysis of Orweb and Orfox (another version of Orweb with bookmark feature – currently both versions are combined into a single version) on Samsung Galaxy S5 running Android version 5.0 and extracted the artifacts. However, no details about the employed tools and techniques were provided. Furthermore, the browsing history was not fully extracted in this research. Moreover, this research was also conducted on an old version of Android and Tor privacy browser that is not compatible with the recent version. In [6] researchers examined 6 different privacy browsers i.e. Epic Privacy Browser, Secure Browser, Comodo Dragon, SRWare Iron, Dooble, and Maxthon along with Tor privacy browser on Windows OS. Evidence was captured using filesystem analysis, registry analysis, network packet captures, memory analysis, and unallocated space analysis. Techniques can be mapped to Android OS but the actual methodology would be different. Similarly, in [7], the authors developed a tool named AndroKit to conduct web browser forensic on rooted Android devices. The tool targets the four famous web browsers available on Android i.e. Chrome, Opera, Mozilla Firefox, and Dolphin. A comparative analysis of AndroKit with standard forensics toolkits was also presented. The tool can recover cookies, bookmarks, web history, visited URLs, stored sessions, and URL credentials from these browsers. This work also employed older versions of Android, Android emulators,

[3]Mobile and Desktop Operating System Market Share Worldwidel StatCounter Global Stats, StatCounter Global Stats https://gs.statcounter.com/osmarket- share [April 2021].

[4]Tablets, laptops & PCs sales forecast 2023 | Statista, Statista,. https://www.statista.com/statistics/272595/global-shipments-forecast-for-tablets-laptops-anddesktop- pcs/ [2021]

[5]Browse Privately. Explore Freely,https://www.torproject.org/ [2021].

and Web Browsers. AndroKit can be used to perform Tor browser forensic as it is based on the Mozilla Firefox web browser.

In [8], the researchers performed a forensic analysis of Tor browser version 5.0 on 64-bit Windows 10. They analyzed the registry settings before and after installation, other filesystem artifacts, and memory of the system to conclude that the Tor browser leaves minimal on-disk evidence. Further, in [9], the authors performed a forensic analysis of Tor privacy browser 7.02 (32-bit) on Windows 8.1 OS in which they analyzed Tor browser artifacts from registry, memory, and storage. However, they only covered normal surface-web based user browsing activities on Tor privacy browser to uncover artifacts related to Tor. They considered only ''Browser open'' and ''Browser closed'' scenarios for memory and storage analysis aspects. Rebecca N and *et al.* [10], recovered forensic artifacts from normal and private browsing modes of two famous browsers i.e. Google Chrome and Mozilla Firefox. The private browsing results were compared with the famous anonymous browser TOR v7.0.5 on Windows 7 (64-bit) using AccessData FTK as a primary tool. Their research predominantly uncovered artifacts from the storage of experimental VMs with the conclusion that the Tor browser reveals limited user browsing artifacts when compared to private browsing modes of Chrome and Firefox. Satrya and Kurniawan [11] proposed a novel Android internal memory forensic acquisition tool called fridump to aid in acquiring Android internal memory more effectively as compared to preceding proposed methodologies, tools, and techniques. They used GDrive as a case study to uncover artifacts from the victim and investigator's Android smartphones i.e. Samsung A7 and Oppo A37F. However, there are some limitations in the tool since it works only with running processes that need to be monitored. Similarly, other works [12]–[15] proposed a framework to recover artifacts of Tor privacy browser from memory, but their investigation covers Windows 10 build 10586 only on memory to reveal user-related information. They have not explored any other areas of the operating system (i.e. registry, file system) for artifacts relevant to the Tor browser.

In the aforementioned related work, most of the techniques only consider the basic Tor browsing activity (i.e. open, close, normal website browsing) for investigation purposes. In addition, the older versions of the Tor browser and operating system builds were employed for experiments that are not useful on the latest versions of applications. For example, due to the significant evolution of applications and platforms that may update its internal structures and the results in these previous studies may not be repetitive and not fresh anymore for further forensic investigations. Therefore, there exists a dire need to explore the latest Tor browser version(s) on the latest OS builds that can help us to perform an evidence profiling of the Tor Browser application. In addition, this can aid investigators in conducting effective forensic investigations for Tor Browser.

To the best of our knowledge, the current version of the Tor browser has not been explored, and no recent study has simultaneously forensically analyzed the Tor browser on two different OS platforms. We forensically analyze the latest version of Tor privacy browser artifacts on the latest builds of Windows and Android OS after simulating a dark-web-based cyber-crime scenario.

This study aims to identify potential areas in Windows and Android devices where a forensic investigator can look for evidence related to the Tor privacy browser. Our findings will help the forensic practitioners to *identify and analyze the artifacts of illicit activity* conducted on seized Windows and Android-based devices which may contribute as digital evidence in the court of law.

III. PROPOSED METHODOLOGY

The objective of this research was to collect evidentiary artifacts related to the usage of the Tor privacy browser on a Windows 10 host machine and an Android 10 smartphone. We simulate dark-web browsing scenarios and analyze the registry, memory, and storage on Windows 10 while on Android 10, we analyze storage, zram (swap partition), and memory for potential artifacts. We then perform a cross-platform comparison of the results. This research methodology is primarily based on earlier work by A. Jadoon et.al. [9] and R.Nelson et.al. [10] with NIST SP 800-63 guidelines.

A. WINDOW 10

Three different areas of the Windows 10operating system are explored i.e. Registry, Memory, and Storage. Acquisition and analysis are aimed at collecting potential artifacts generated during the installation, execution (with or without any browsing), and uninstallation of the Tor privacy browser. We didn't cover uninstallation activity in storage analysis on Windows 10 because Tor uninstallation simply involves the deletion of an application folder.

B. ANDROID 10

For Android 10 devices, we explore four different areas for artifacts i.e. Storage, Zram, memory (RAM), and ADB (Android Device Bridge) Logs.

ADB is a command-line tool that allows us to communicate with the device [16] and fetch Android device logs using two important tools. The first is *logcat* [17] that outputs logs of system messages and the second is *Dumpsys* [18] that outputs information about system services.

In Linux-based OS such as Android, Zram is a compressed block device in RAM which 1) can be used as a swap space because it does not have an exclusive swap 2) helps to increase the memory available on Android by compressing the excessive storage resources to a dedicated space in RAM which can be later retrieved by operating system 3) mounts as a block device in Android and its acquisition can be easily achieved using a simple copy-paste operation via the ADB shell or a forensic tool.

The acquisition of RAM on Android requires installation of a specialized kernel module e.g. LiME, or execution of

specialized binary e.g. Frida-server which may compromise device storage evidence. Even then, the acquired RAM is for a specific process when it is executing. Our aim is the acquisition of artifacts generated during the installation, execution (with or without any browsing activity), and un-installation of the Tor privacy browser. The exception is that we cover only execution activity (with and without any browsing) in memory acquisition and analysis due to the reason outlined in section IV(II).

In addition to the browsing activities, our experimentation also considers three different states for Android devices i.e. Unrooted Android device (without admin privileges) and Rooted Android device (with admin privileges) [19], and NANDroid Backup (with Custom Recovery software installed making a perfect mirror image of the device) [20].

C. EXPERIMENTAL SETUP

1) WINDOWS 10

To work in a clean environment, a fresh Windows 10 virtual machine was created to analyze the registry, memory, and storage artifacts. The tools used include:

- VMware® Workstation 14 Pro (Version 14.0.0 Build 6661328)
- Window 10 Pro 64-bit (Version 20H2 Build 19042.746)
- Tor Browser for Windows 64-bit (Version 10.0.7)
- Regshot 64-bit (Version 1.9.0)
- Regscanner 64-bit (Version 2.60)
- AccessData FTK Imager (Version 4.5.0.3)

2) ANDROID 10

Clean Android 10 devices were utilized to analyze the storage, Zram, and memory artifacts. The devices and tools used are:

- Xiaomi Mi A3 with Android 10 (Build 10 QKQI.190910.002 V11.0.15.0.QFQMIXM)
- Samsung A30S with Android 10 (Build QP1A.190711.020.A307FNXXU2BTL2)
- Nokia 5.1 with Android 10 (Version 4.160)
- Tor Browser (Version 68.7.0 Build 2015690707)
- Android SDK Platform Tools (Version 29.0.6)
- TWRP (Version 3.4.0)
- Magisk (Version 21.4)
- Belkasoft Evidence Center 64-bit (Version 9.9800 Build 4928)
- MOBILedit Forensic Express 64-bit (Version 7.0.3.16830)
- Python3
- FRIDA Tools (Version 9.1.0) [21]
- Frida Server for android-arm64 (Version 14.2.11) [22]
- Fridump– A novel open-source Android memory dumping tool (Version 0.1)

3) OTHER ANALYSIS TOOLS

- HxD Hex Editor 64-bit (Version 2.2.0.0)
- DCode (Version 4.02a Build 9306)
- GrepWin 64-bit (Version 1.9.2)
- WinDiff
- WinMerge 32-bit (Version 2.16.6.0)
- DB Browser for SQLite 64-bit (Version 3.12.1)

D. INVESTIGATIVE SCENARIO

To perform forensic analysis of the Tor privacy browser on both OS, we simulate the following cyber-crime scenario:

"*A suspect was arrested by the Law Enforcement Agency (LEA) based on information from intelligence agencies. The allegation against the suspect is* ***"... the breach of confidential information related to government/corporate employees and its communication to foreign intelligence agencies via dark web".*** *The suspect was caught with his laptop and an Android-based smartphone by LEAs.*

Upon preliminary physical inspection, the laptop was running the latest Windows 10 OS (with the latest update installed) and the smartphone was running the Android 10 operating system. The device(s) were seized and wrapped in a faraday bag(s) attached to specialized power sources. Chain-of-custody form(s) was signed, and evidence was handed over to the forensic investigation lab for further investigation.

LEAs required the following information from the lab about suspect activities:

- *Any evidence of the use of Tor browser for exfiltration of confidential information*
- *Any email/website visited/used where we can find evidence of his activity*
- *Any clues related to his activity*
- *Any credentials used for suspicious communication*
- *Any other files of interest"*

We simulate every possible activity (browsing or non-browsing) that a suspect may have performed using the Tor privacy browser as per the simulated scenario mentioned above. This includes visiting various kinds of scenario-specific websites including dark-web (.onion) websites. Websites included for suspicious browsing activities are as follows:

1) DARK-WEB (.ONION) WEBSITES

- Hidden Wiki – a very famous wiki on the dark web
- Three different darknet search engines i.e. Ahmia, DuckDuckGo, and Excavator
- Secmail - a tor based secure email service
- Galaxy3 – a tor based social networking platform
- StealthPay – an anonymous money transfer platform
- Keybase – a secure communication website
- Anonymous text sharing websites i.e. ZeroBin and StrongHold Paste
- Anonymous file sharing website i.e. SecureDrop

2) NORMAL WEBSITES

- Gmail & Google Drive with same Gmail account
- Outlook & Skype Web with same Hotmail account
- MEGA free cloud storage

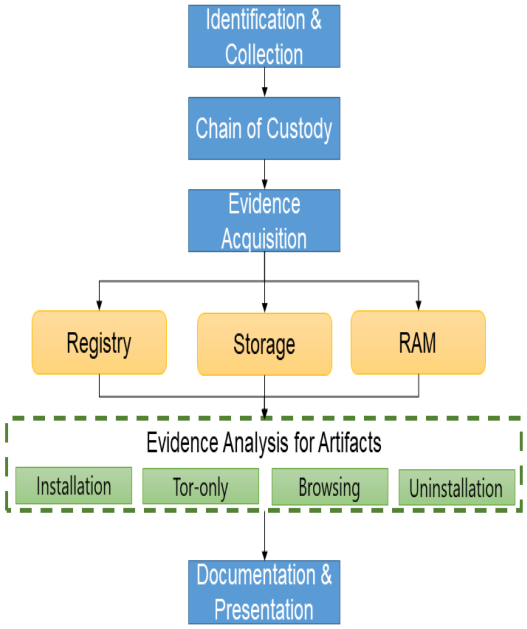


FIGURE 1. Investigation methodology on Windows 10.

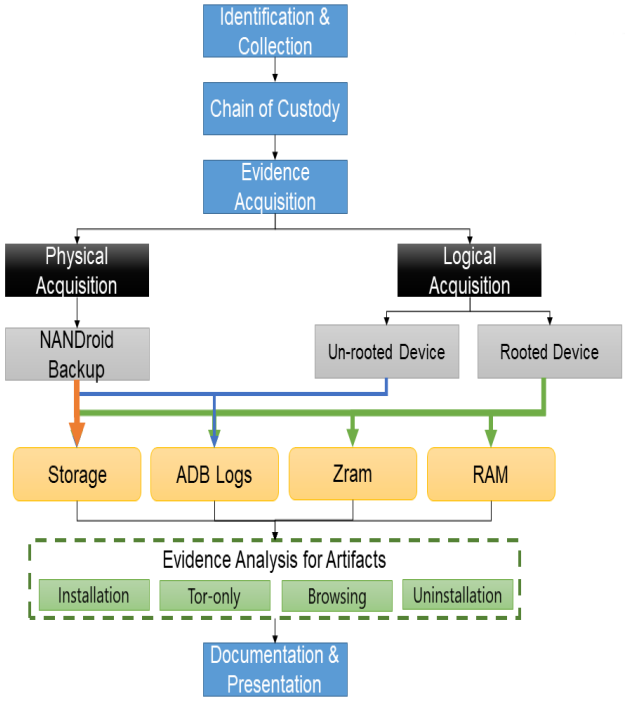


FIGURE 2. Investigation methodology on Android 10.

The details of all the browsing activities on Windows 10 and Android 10 devices are provided in Table 1 and Table 2 respectively.

After the acquisition, the Windows 10 virtual machine was returned to a clean state snapshot and the free space on the Android 10 file system was shredded. The flowcharts of our proposed digital investigation methodology (based on NIST Special Publication 800-86 [23]) and adopted for our targeted platforms are shown in Figures 1 & 2.

E. TARGETED TOR BROWSER ACTIVITIES FOR DIGITAL INVESTIGATION

We covered four different activities of Tor privacy browser to acquire evidence(s) linked to the application lifecycle on Windows 10 and Android 10 operating systems and these are described below:

I. **Installation –** the Tor browser is installed but not executed.
II. **Simple Execution –**the Tor browser is executed. Browser is connected to the Tor network, but no browsing activity is performed during this time.
III. **Browse**– the browsing activities mentioned in Table 1 and 2 of section III(D) are performed in this activity
IV. **Un-installation -** the Tor browser is uninstalled.

IV. EVIDENCE ACQUISITION

A. WINDOWS 10

Three different types of acquisitions were performed on Windows 10:

- Registry
- Storage
- Memory

1) BRIEF EVIDENCE ACQUISITION METHODOLOGY WITH TOOLS USED

i. Registry snapshots are acquired using the Regshot tool before and after the below-mentioned activities:
   a. Installation
   b. Execution (with or without browsing)
   c. Post-Execution
   d. Uninstallation
ii. FTK imager and VMware snapshot virtual memory VMEM files are used for acquiring storage and memory images that were acquired during *Simple Execution* and *Browsing* activity. In the *Browsing* activity, we consider two more states of the browser for evidence acquisition:
   a. ***Browser Open–*** Image acquired when browser remained open on last opened tab.
   b. ***Browser Closed–***Image acquired when the browser is closed.

B. ANDROID 10

In the case of the Android device, we performed different acquisitions according to the state of the device that was encountered during our digital investigation scenario.

1) ANDROID DEVICE STATE(S) WITHA PARTICULAR TYPE OF ACQUISITION

- **Un-rooted Android device**
  - Storage (including ADB Logs)
- **NANDroid Backup**
  - Storage

TABLE 1. Browsing activities performed on Windows 10 virtual machine.

Sr.	Cat.	Title	Website/URL visited	Credentials	Activities Performed
1	Wiki	Hidden Wiki	zqktlwiuavvvqqt4ybvgvi7tyo4hjl5xgfuvpdf6otjiycgwqbym2qad.onion/wiki/index.php/Main_Page	-	1. Browsing 2. Whistleblowing link clicked
2	Search Engines	Ahmia	msydqstlz2kzerdg.onion	-	1. Browsing 2. Search query **"sell official data"** 3. Clicked first result & redirected to **5j7saze5byfqccf3.onion/data/bullseye/main/** 4. Download **components-mips64el.yml.gz** from URL
3		DuckDuck Go	3g2upl4pq6kufc4m.onion	-	1. Browsing 2. Search query **"sell official data"**
4	Cloud Storage/ Sharing	Google Drive	drive.google.com	Torforensics @gmail.com	1. Browsing only after login to mail.google.com using Google credentials* 2. Upload Text file "~res-x64-1.txt"
5		MEGA	mega.nz/login mega.nz/fm	torforensics @gmail.com	1. Login 2. Right-click PDF file and get sharing link **https://mega.nz/file/zz40xB6S#isXGprskZbLP4KnLNuNHcbI279s6FnLcsj8Vydm_sio** **3.** Copy link to clipboard
6		ZeroBin	zerobinqmdqd236y.onion	-	1. Browsing 2. **Clipboard: Mega Sharing Link** pasted 3. Link copied **http://zerobinqmdqd236y.onion/?be163e348777b667#H7xg5DfMboatOgot8q439QNYTogRfXLAP74fmqzeXjI=**
7	Money	StealthPay	https://www.stealthpay.com/requestmoney	-	1. Browsing only
8	Secure Comm	Keybase	fncuwbiisyh6ak3i.onion	-	1. Browsing 2. Tried downloading software from /download 3. Visited .../docs/the_app/install_windows
9	Emails	SecMail	secmail63sex4dfw6h2nsrbmfz2z6alwxe4e3adtkpd4pcvkhht4jdad.onion	adamjames555@secmail.pro	1. Browsing 2. Login only 3. Open the first email in Inbox 4. Reply email to Gmail as shown below: To: **"torforensics@gmail.com"** Subject: **"Re: Impt Data"** Body: **"https://mega.nz/file/zz40xB6S#isXGprskZbLP4KnLNuNHcbI279s6FnLcsj8Vydm_sio"**
10		Gmail	mail.google.com	torforensics @gmail.com	1. Browsing 2. Login only 3. Check email
11		Outlook	outlook.live.com	torforensics @outlook.com	1. Browsing 2. Login 3. Send email to Gmail as shown below: Subject: **"Imp Stuff"** Body: **"Send money at my wallet"**
12		Skype	Web.skype.com Secure.skype.com www.skype.com		1. Browsing 2. Login 3. web.skype.com opened but the **"browser not supported"** message received
13	Social Media	Galaxy3	galaxy3bhpzxecbywoa2j4tg43muepnhfalars4cce3fcx46qlc6t3id.onion	adamjames555@tutanota.com	1. Browsing 2. Login 3. Write Wire Blog Post with the content shown below: **http://zerobinqmdqd236y.onion/?be163e348777b667#H7xg5DfMboatOgot8q439QNYTogRfXLAP74fmqzeXjI=**

TABLE 2. Browsing activities performed on Android 10 device.

Sr.	Cat.	Title	Website/URL	Credentials	Activities
1	Wiki	Hidden Wiki	zqktlwiuavvvqqt4ybvgvi7tyo4hjl5xgfuvpdf6otjiycgwqbym2qad.onion/wiki/index.php/Main_Page	-	1. Browsing 2. Whistleblowing link clicked
2	Search Engines	Ahmia	msydqstlz2kzerdg.onion	-	1. Browsing 2. Search query **"sell official data"** 3. Clicked first result & redirected to **5j7saze5byfqccf3.onion/data/experimental/main/** 4. Download **components-arm64.yml.xz** from URL
3		DuckDuck Go	3g2upl4pq6kufc4m.onion	-	1. Browsing 2. Search query **"sell official data"**
4		Excavator	2fd6cemt4gmccflhm6imvdfvli3nf7zn6rfrwpsy7uhxrgbypvwf5fad.onion	-	1. Browsing 2. Search query **"sell official data"**
5	Cloud Storage/ Sharing	Google Drive	drive.google.com	torforensics@gmail.com	1. Browsing only after login to mail.google.com using Google credentials*
6		MEGA	mega.nz/login mega.nz/fm	torforensics@gmail.com	1. Browsing 2. Login 3. Upload IMG-20210122-WA0005.jpg 4. Right-click and get sharing a link 5. Copy link to clipboard
7		ZeroBin	zerobinqmdqd236y.onion	-	1. Browsing 2. Mega.nz file-sharing link pasted 3. Get Paste link containing "**/?a3e1481092fb04b9"**
8		StrongHold Paste	nzxj65x32vh2fkhk.onion	-	1. Browsing 2. Set the Paste title to **"Pix"** with content **"https://goo.gl/xZgh1qu"** 3. Password-protect it 4. Get paste link containing **"/pocsxm1d5/2uo2vh"**
9		SecureDrop	arujlhu2zjjhc3bw.onion arujlhu2zjjhc3bw.onion/lookup	-	1. Browsing 2. Click **"Get started"** and get codename **"unloving cornflake ecosphere decipher trifocals scotch reiterate"** 3. Click **"Submit documents"** 4. Upload **IMG-20210122-WA0005.jpg**
10	Money	StealthPay	https://www.stealthpay.com/requestmoney	-	1. Browsing only
11	Secure Comm	Keybase	fncuwbiisyh6ak3i.onion	-	1. Browsing 2. click **"Send secure message"** it will redirect to **"play.google.com"** for **key base** android apk installation.
12	Emails	SecMail	secmail63sex4dfw6h2nsrbmfz2z6alwxe4e3adtkpd4pcvkhht4jdad.onion	adamjames555@secmail.pro	1. Browsing 2. Login 3. Check Emails received from **Gmail** and **Outlook** email addresses 4. Reply was sent to Gmail as shown below: Subject: **"Re: Impt Data"** Body: **"Find here: https://goo.gl/xZgh1qu"**
13		Gmail	mail.google.com	torforensics@gmail.com	1. Browsing 2. Login 3. Email composed and sent to **adamjames555@secmail.pro** as shown below: Subject: **"Impt Data"** Body: **"Please share link to receive data"**
14		Outlook	outlook.live.com	torforensics@outlook.com	1. Browsing 2. Login 3. Email composed and sent to **adamjames555@secmail.pro** as shown below: Subject: **"Imp Data"** Body: **"Please share link to receive data"**
15		Skype	Web.skype.com Secure.skype.com www.skype.com		1. Browsing 2. Login 3. Visited **Account overview** 4. **web.skype.com** opened but **"browser not supported"** message received on the webpage
16	Social Media	Galaxy3	galaxy3bhpzxecbywoa2j4tg43muepnhfalars4cce3fcx46qlc6t3id.onion	adamjames555@tutanota.com	1. Browsing 2. Login

TABLE 2. *(Continued.)* **Browsing activities performed on Android 10 device.**

					3. **/Settings** link visited 4. Blogs link **"/blog/owner/aj555"** visited
17	Torrents	The Pirate Bay	https://thepiratebay.cx/en1/	-	1. Browsing 2. search **"privacy"** with **Application** check box clicked on the webpage 3. Privacy Sheild URL opened from results 4. Get Magnet Link **"magnet:?xt=urn:btih:2A3B…"**

- **Rooted Android device**
  - Storage
  - Zram
  - Memory

2) BRIEF EVIDENCE ACQUISITION METHODOLOGY WITH TOOLS USED

a) First, we tried to acquire as much evidence as possible from an unrooted Android device after installation, browsing, and uninstallation. Since we do not have a lot of access, we are only able to acquire ADB logs and other basic non-browsing evidence(s) from emulated storage using ADB platform tools and MOBILedit Forensic Express.

b) Next, we unlocked the bootloader of our targeted Android device [24] using ABD platform tools in Fastboot mode to install a custom recovery software i.e. TWRP [25] to acquire NANDroid backup of the device's filesystem. NANDroid backup is a physical backup of the Android device. It is occasionally performed by investigators to access the underlying restricted filesystem areas most specifically /data/data/ directory. We stored the NANDroid backup on SD Card for further analysis. Using TWRP, we can only be able to acquire storage evidence for the ''Browser Closed'' state because NANDroid backup requires rebooting the device into recovery mode.

c) Finally, we rooted our device using Magisk [26] to gain unrestricted access to the underlying filesystem. In this way, we were able to acquire storage and Zram evidence for all the targeted activities mentioned in section III(E) using MOBILedit Forensic Express. However, we were only able to acquire memory evidence using the most efficient Android memory forensic tool developed by Satrya and Kurniawan [11] for Simple Execution and Browsing activity because Fridump tool only let us acquire memory evidence while the process is running.

Warning: Acquisition methodologies mentioned at Sr. No. 2 & 3 were only emulated here for experimentation. Use of these methodologies in real case scenarios without any authorization & precautions will be dangerous and can destroy your seized evidence. These evidence acquisition techniques are only recommended if the device already has an unlocked bootloader or is rooted which may vary.

To cover all the activities, we mentioned in section III(E) on both OS, we performed the evidence acquisition as per the matrix given in Table 3.

After completion of each phase, the system is reverted to a clean state and/or restarted to ensure that no artifacts from the previous acquisition phase remain on the system. Acquired images are dumped to the external storage and then to the forensic workstation to ensure host integrity.

V. EVIDENCE ANALYSIS AND RESULTS

A. WINDOWS 10

Forensics analysis on Windows 10 was done in three phases. In the first phase, registry snapshots were analyzed for all our targeted activities while memory and storage images were analyzed in the next two phases.

TABLE 3. **Evidence acquisition matrix along with targeted activities.**

. Targeted Application Activities		Windows 10			Android 10								
		Registry	Storage	RAM	Zram			Storage			RAM		
					UR[1]	NB[2]	RT[3]	UR	NB	RT	UR	NB	RT
Installation		Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	No	No	No
No-browsing Execution		Yes	Yes	Yes	No	No	Yes	Yes	No	Yes	No	No	Yes
Browsing Execution	**Browser Open**	Yes	Yes	Yes	No	No	Yes	Yes	No	Yes	No	No	Yes
	Browser Closed	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	No	No	No
Uninstallation		Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	No	No	No

1. UR – Unrooted Android Device 2. NB – NANDroid Backup 3. RT – Rooted Android Device

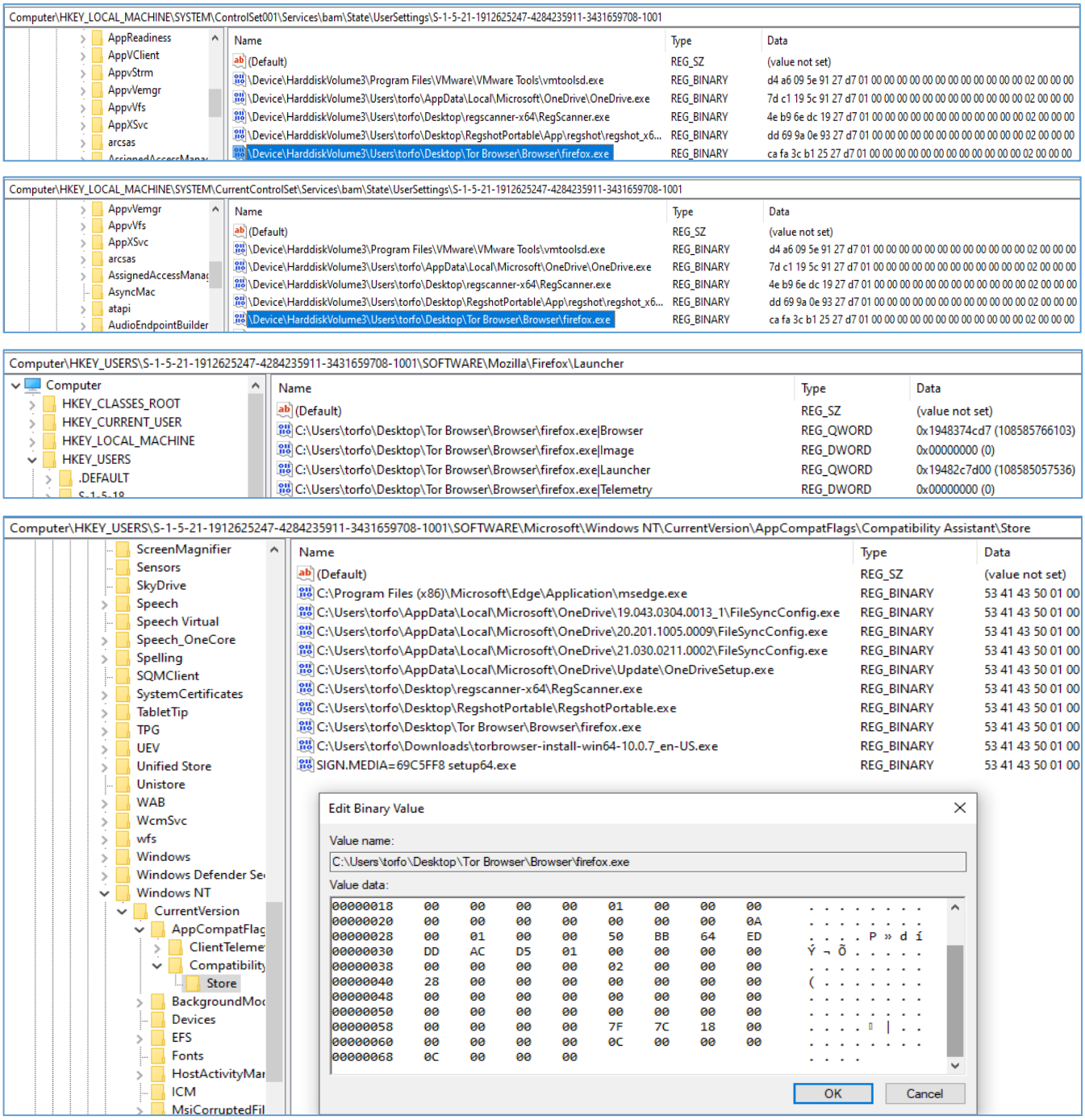


FIGURE 3. Registry remnants of Tor privacy browser on Windows 10 after uninstallation.

1) REGISTRY ANALYSIS

In Windows forensic investigations, the Registry is considered as the heart of the Windows operating system and an important forensic resource that provides significant information about who, what, where, and when something that took place on a system which can directly link the suspect to the actions being taken i.e., users, the time when they last used the system or the application. Registry files normally store data (values) under unique values called "Keys" which requires investigators to acquire sufficient knowledge about Registry keys and the data which are stored under those Keys for conducting effective forensic analysis.

We used Regshot, RegScanner, Notepad++, and WinMerge tools to analyze our registry snapshots. Our analysis reveals that the Tor browser adds eight (08) registry keys after installation and three (03) other registry keys relevant

TABLE 4. Registry artifacts retrieved from Windows 10.

Registry Artifact(s)	Artifacts of Interest
Pre-Installation (Registry Keys relevant to Tor Browser Installer)	
HKU\S-1-5-21-1912625247-4284235911-3431659708-1001\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Compatibility Assistant\Store\C:\Users\torfo\Downloads\torbrowser-install-win64-10.0.7_en-US.exe	Tells us about either the installation activity ever took place on the system.
HKLM\SYSTEM\ControlSet001\Services\bam\State\UserSettings\S-1-5-21-1912625247-4284235911-3431659708-1001\\Device\HarddiskVolume3\Users\torfo\Downloads\torbrowser-install-win64-10.0.7_en-US.exe	
HKLM\SYSTEM\CurrentControlSet\Services\bam\State\UserSettings\S-1-5-21-1912625247-4284235911-3431659708-1001\\Device\HarddiskVolume3\Users\torfo\Downloads\torbrowser-install-win64-10.0.7_en-US.exe	
Post-Installation	
HKU\S-1-5-21-1912625247-4284235911-3431659708-1001\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Compatibility Assistant\Store\C:\Users\torfo\Desktop\Tor Browser\Browser\firefox.exe	Tells us about the number of executions of the Tor browser since installation
HKU\S-1-5-21-1912625247-4284235911-3431659708-1001\SOFTWARE\Microsoft\Internet Explorer\LowRegistry\Audio\PolicyConfig\PropertyStore\97a7a40b_0\: "{2}.\\?\hdaudio#func_01&ven_15ad&dev_1975&subsys_15ad1975&rev_1001#{6994ad04-93ef-11d0-a3cc-00a0c9223196}\elineouttopo/00010001\Device\HarddiskVolume3\Users\torfo\Desktop\Tor Browser\Browser\firefox.exe%b{00000000-0000-0000-0000-000000000000}"	
HKU\S-1-5-21-1912625247-4284235911-3431659708-1001\SOFTWARE\Mozilla\Firefox\Launcher\C:\Users\torfo\Desktop\Tor Browser\Browser\firefox.exe|Image	
HKU\S-1-5-21-1912625247-4284235911-3431659708-1001\SOFTWARE\Mozilla\Firefox\Launcher\C:\Users\torfo\Desktop\Tor Browser\Browser\firefox.exe|Telemetry	
HKU\S-1-5-21-1912625247-4284235911-3431659708-1001\SOFTWARE\Mozilla\Firefox\Launcher\C:\Users\torfo\Desktop\Tor Browser\Browser\firefox.exe|Launcher	Values changes at every opening and closing of Tor browser
HKU\S-1-5-21-1912625247-4284235911-3431659708-1001\SOFTWARE\Mozilla\Firefox\Launcher\C:\Users\torfo\Desktop\Tor Browser\Browser\firefox.exe|Browser	
HKLM\SYSTEM\ControlSet001\Services\bam\State\UserSettings\S-1-5-21-1912625247-4284235911-3431659708-1001\\Device\HarddiskVolume3\Users\torfo\Desktop\Tor Browser\Browser\firefox.exe	
HKLM\SYSTEM\CurrentControlSet\Services\bam\State\UserSettings\S-1-5-21-1912625247-4284235911-3431659708-1001\\Device\HarddiskVolume3\Users\torfo\Desktop\Tor Browser\Browser\firefox.exe	
Other Interesting Registry keys to check for recent programs	
HKCR\Local Settings\Software\Microsoft\Windows\Shell\MuiCache	
HKCR\Local Settings\Software\Microsoft\Windows\Shell\BagMRU	
HKU\S-1-5-21-1912625247-4284235911-3431659708-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\CIDSizeMRU	
HKU\S-1-5-21-1912625247-4284235911-3431659708-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedPidlMRU	
HKU\S-1-5-21-1912625247-4284235911-3431659708-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePidlMRU	
Uninstallation	
All registry artifacts created at the time of installation found	

to the Tor Brower installer file during installation. All these registry keys have varying values which are dependent on the opening and closing of the Tor browser which will be very helpful in cases where an investigator is interested to know that whether the user just installed the Tor browser or used it as well after installation, but unfortunately, they do not provide any information related to the user browsing activities. In addition to these keys, some keys will be helpful for investigators to check recent programs executed on the system.

All these keys persist in the registry after uninstallation as shown in Fig 3 and may help the investigator in building a hypothesis about the case. For further details regarding registry artifacts, refer to Table 4.

2) MEMORY ANALYSIS

In Memory analysis, we first extract **"Tor browser only artifacts"** and then in the second phase we look for **"Browsing artifacts"**.

a: TOR ONLY ARTIFACTS

In this phase, we only extract artifacts that are related to the Tor application. We extract artifacts left on the memory of the system after installation, first time and subsequent

TABLE 5. Tor only artifacts from memory on Windows 10.

Sr.No.	Type of Artifact(s)
1	Application paths
2	Loaded EXE (firefox.exe and tor.exe) & DLL files
3	SQlite Files; Tables names and application performed DB operations Application used Built-in Windows Functions
4	Application used Resources traces
5	Router's information including IP Addresses, nicknames, last available timestamps, Public keys used by Tor Router
6	User-agent info (Mozilla/5.0)
7	Application related Blocklists and Extensions data (included timestamps) Registry keys and values
	In case, an application was uninstalled immediately after recent browsing, you may find:
8	Opened/Redirected URL Traces
9	Website components traces (.js, .cssetc)
10	Downloaded filenames & URLs
11	Login email address traces
12	Timestamps
13	Sessions IDs or other session related information
14	Traces of any clipboard operation performed in the context of browser

executions, and after uninstallation of the Tor browser. HxD and Belkasoft Evidence Center are used for forensics analysis of acquired memory images. A list of all recovered artifacts during this phase of memory analysis is given in Table 5.

Artifacts related to router information can also be helpful for law enforcement agencies in case of backtracking a Tor user for any illegal activity. This can be done by collecting artifacts from the relays with the aid of respective LEAs and ISPs. However, it was beyond our scope of digital investigation.

b: BROWSING ARTIFACTS

In this phase, we only looked for user browsing artifacts in the memory. As explained in the Data Acquisition section, two VMware snapshots were taken for "Browser Open" and "Browser Closed" scenarios. Memory images (.vmem files) of these two VMware snapshots were analyzed for browsing artifacts using HxD and Belkasoft Evidence Center.

We performed most of our analysis using string searches and found remnants of visited websites/URLs, search queries, credentials (emails, usernames, and passwords), emails sent/received, uploaded & downloaded files, and other artifacts. All emails in the Inbox of Gmail, Outlook, and Secmail accounts including unread emails are present in memory. The artifacts we found in the "Browser Open" memory image were almost identical to the "Browser Closed" memory image which implies that the Tor browser does not instantly clear the user browsing history from memory while closing the application. Screenshots of some of these artifacts are shown in Fig. 4. The summary of all the user browsing artifacts found in memory is listed in Table 6.

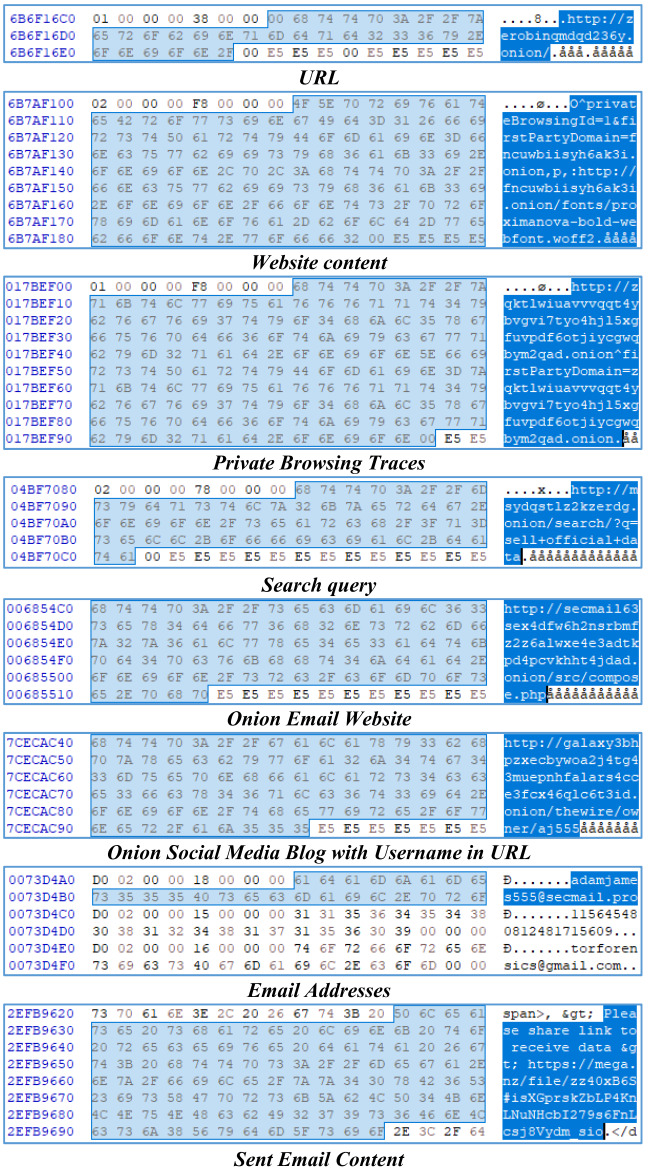


FIGURE 4. User browsing artifacts from memory on Windows 10.

3) STORAGE ANALYSIS

In this phase, we analyzed forensic images of the Tor Browser application. Three image files were analyzed which include one for "Post-Installation", second for "Browser Open" and third for "Browser Closed" scenario. Application-related configuration and database files were analyzed in this phase to look for timestamps, bookmarks, and traces of user browsing activity, but no browsing evidence was found on the filesystem. Uninstallation activity was not covered purposely because it just involves deleting the main application folder from the filesystem (https://tb-manual.torproject.org/uninstalling/). Only file carving and deleted data recovery can be performed which we have omitted from the scope of this research

TABLE 6. Browsing related memory artifacts from Windows 10.

Sr.	Websites	Activities	Artifact(s) found while Browser Open	Artifact(s) found while Browser Closed
1	Hidden Wiki	1. Browsing 2. Whistleblowing link clicked	▪ Website/URL traces ▪ Visited/Redirected URLs traces ▪ Website components (js,css) traces ▪ SOCKS socket traces	▪ Website/URL traces ▪ Visited/Redirected URLs traces ▪ Website components (js,css) traces
2	Ahmia	1. Browsing 2. Search query **"sell official data"** 3. Clicked first result & redirected to **5j7saze5byfqccf3.onion/data/bullseye/main/** 4. Download **components-mips64el.yml.gz** from URL	▪ Website/URL traces ▪ Visited/Redirected URLs ▪ Website components (js,css) ▪ Search query traces ▪ Downloaded file & URL traces ▪ Download timestamps	▪ Website/URL traces ▪ Visited/Redirected URLs traces ▪ Search query traces ▪ Downloaded file & URL traces ▪ Download timestamps
3	DuckDuckGo	1. Browsing 2. Search query **"sell official data"**	▪ Website/URL traces ▪ Website components (js,css) traces ▪ Search query traces ▪ SOCKS socket traces	▪ Website/URL traces ▪ Search query traces
4	Google Drive	1. Browsing only after login to mail.google.com using Google credentials* 2. Upload Text file "~res-x64-1.txt"	▪ Website/URL traces ▪ Website components (js,css) traces ▪ Uploaded file traces ▪ Timestamps ▪ Login Email & Password traces	▪ Website/URL traces ▪ Website components (js,css) traces
5	MEGA	1. Login 2. Right-click PDF file and get sharing link **https://mega.nz/file/zz40xB6S#isXGprskZbLP4KnLNuNHcbI279s6FnLcsj8Vydm_sio** **3.** Copy link to clipboard	▪ Website/URL traces ▪ Website components (js,css) traces ▪ Clipboard Operation traces ▪ Timestamps ▪ Local Megasync client socket 127.0.0.1:6341 ▪ SOCKS Username/Password Traces ▪ SOCKS Socket Traces ▪ Login Email Traces	▪ Website/URL traces ▪ Clipboard Operation traces
6	ZeroBin	1. Browsing 2. **Clipboard: Mega Sharing Link** pasted 3. Link copied **http://zerobinqmdqd236y.onion/?be163e348777b667#H7xg5DfMboatOgot8q439QNYTogRfXLAP74fmqzeXjI=**	▪ Website/URL traces ▪ Website components (js,css) traces ▪ Clipboard Operation traces ▪ Generated Filesharing/Paste URL information traces ▪ Timestamps ▪ SOCKS Username/Password Traces ▪ Paste Token ID traces	▪ Website/URL traces ▪ Generated Filesharing/Paste URL information traces
7	StealthPay	1. Browsing only	▪ Only domain name found	Nothing found
8	Keybase	1. Browsing 2. Tried downloading software from /download 3. Visited .../docs/the_app/install_windows	▪ Website/URL traces ▪ Website components (js,css) traces ▪ Visited/Redirected URLs traces ▪ Download URL traces ▪ Timestamps ▪ SOCKS Username/Password Traces ▪ SOCKS Socket Traces ▪ Response header traces	Nothing found
9	SecMail	1. Browsing 2. Login only 3. Open first email in Inbox 4. Reply email to Gmail as shown below: To: **"torforensics@gmail.com"** Subject: **"Re: Impt Data"** Body: **"https://mega.nz/file/zz40xB6S#isXGprskZbLP4KnLNuNHcbI279s6FnLcsj8Vydm_sio"**	▪ Website/URL traces ▪ Website components (js,css) traces ▪ Inbox & Sent Emails traces ▪ Timestamps ▪ SOCKS Username/Password Traces ▪ Login Email traces	▪ Website/URL traces ▪ Login Email traces
10	Gmail	1. Browsing 2. Login only 3. Check email	▪ Website/URL traces ▪ Website components (js,css) traces ▪ Only Inbox Emails traces ▪ Timestamps	▪ Website/URL traces ▪ Website components (js,css) traces

TABLE 6. *(Continued.)* **Browsing related memory artifacts from Windows 10.**

			▪ Cookies ▪ Response header traces ▪ Login timestamps ▪ Login Email & Password traces	
11	Outlook	1. Browsing 2. Login 3. Send email to Gmail as shown below: Subject: **"Imp Stuff"** Body: **"Send money at my wallet"**	▪ Website/URL traces ▪ Website components (js,css) traces ▪ Inbox & Sent Emails traces ▪ Timestamps ▪ Session IDs ▪ Login Email & Password traces	▪ Website/URL traces ▪ Login Email & Password traces ▪ Timestamps ▪ Session IDs
12	Skype	1. Browsing 2. Login 3. web.skype.com opened but the **"browser not supported"** message received	▪ Website/URL traces ▪ Website components (js,css) traces ▪ Timestamps ▪ SOCKS Socket Traces ▪ Skype Local Socket ▪ Login timestamps ▪ Login Email & Password traces	▪ Website/URL traces ▪ Login Email traces ▪ Timestamps
13	Galaxy3	1. Browsing 2. Login 3. Write Wire Blog Post with the content shown below: **http://zerobinqmdqd236y.onion/?be163e348777b667#H7xg5DfMboatOgot8q439QNYTogRfXLAP74fmqzeXjI=**	▪ Website/URL traces ▪ Website components (js,css) traces ▪ Timestamps ▪ Visited/Redirected URLs traces ▪ Login Email & Password traces ▪ Username ▪ Content of the wire blog	▪ Website/URL traces ▪ Visited/Redirected URLs traces ▪ Username ▪ Login Password traces

TABLE 7. **Summary of browsing artifacts from Window 10.**

Browsing Artifacts	Evidence Locations		
	Filesystem	*RAM*	*Registry*
URLs	No	Yes	No
Website Content	No	Yes	No
Search Queries	No	Yes	No
Bookmarks	Yes	Yes	No
Cookies	No	No	No
Email Addresses	No	Yes	No
Email Content	No	Yes	No
Usernames	No	Yes	No
Passwords	No	Yes	No
Downloaded Files	Yes	Yes	No
Browsing Timestamps	No	Yes	No
Usage/Session Timestamps	Yes	No	No

a: POST-INSTALLATION

In this stage, the artifacts produced after the Tor browser was installed on the Windows 10 were analyzed. Moreover, the browser was not executed at all. Only application-related configuration files along with installation timestamps were found at this stage.

```
[ 05-23 12:51:55.876  1108: 2488 I/am_create_activity ]
[0,157806928,688,org.torproject.torbrowser/.App,android.intent.action.MAIN,NULL,NULL,278921216]
```

FIGURE 5. **Application activity traces in events.log file.**

b: BROWSING–BROWSER OPEN

Artifacts that are present in the hard disk when the browser is open were searched in this part of the analysis. The artifacts we found had all the downloaded data and bookmarks information and timestamps. No user browsing-related information was found in this stage. However, all registry artifacts were present.

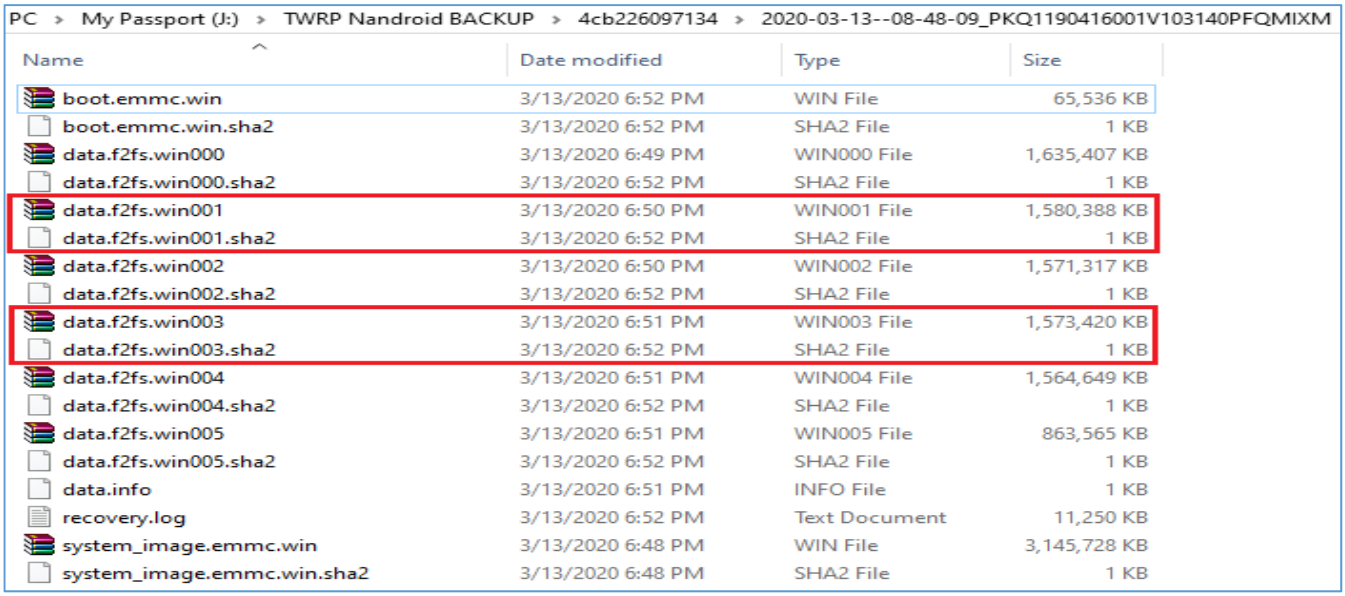
c: BROWSING–BROWSER CLOSED

In this stage of analysis, all those artifacts were searched which are present on the filesystem after the browser was closed. All steps performed in the previous part of the storage analysis were also repeated in this stage. Artifacts similar to those found in the browser open stage were present in this stage. However, user browsing information was still not available. A summary of all the browsing artifacts retrieved from the Tor privacy browser on Windows 10 is provided in Table 7.

B. ANDROID 10

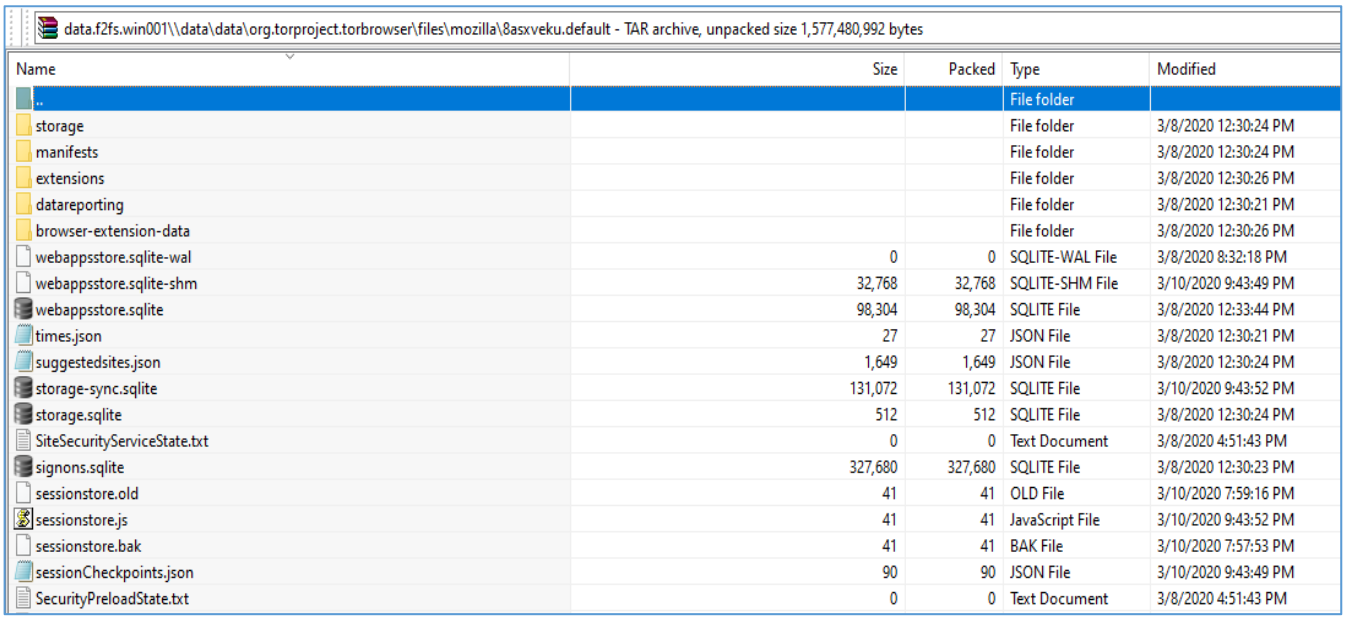
On Android, forensic analysis is done in three phases.

In our first phase, filesystem and ADB logs were analyzed for artifacts on an un-rooted Android device which is a normal state of an android device we usually use in our daily life, while in the second phase, we performed NANDroid backup of our device for storage artifacts and in the third phase,



Name	Date modified	Type	Size
boot.emmc.win	3/13/2020 6:52 PM	WIN File	65,536 KB
boot.emmc.win.sha2	3/13/2020 6:52 PM	SHA2 File	1 KB
data.f2fs.win000	3/13/2020 6:49 PM	WIN000 File	1,635,407 KB
data.f2fs.win000.sha2	3/13/2020 6:52 PM	SHA2 File	1 KB
data.f2fs.win001	3/13/2020 6:50 PM	WIN001 File	1,580,388 KB
data.f2fs.win001.sha2	3/13/2020 6:52 PM	SHA2 File	1 KB
data.f2fs.win002	3/13/2020 6:50 PM	WIN002 File	1,571,317 KB
data.f2fs.win002.sha2	3/13/2020 6:52 PM	SHA2 File	1 KB
data.f2fs.win003	3/13/2020 6:51 PM	WIN003 File	1,573,420 KB
data.f2fs.win003.sha2	3/13/2020 6:52 PM	SHA2 File	1 KB
data.f2fs.win004	3/13/2020 6:51 PM	WIN004 File	1,564,649 KB
data.f2fs.win004.sha2	3/13/2020 6:52 PM	SHA2 File	1 KB
data.f2fs.win005	3/13/2020 6:51 PM	WIN005 File	863,565 KB
data.f2fs.win005.sha2	3/13/2020 6:52 PM	SHA2 File	1 KB
data.info	3/13/2020 6:51 PM	INFO File	1 KB
recovery.log	3/13/2020 6:52 PM	Text Document	11,250 KB
system_image.emmc.win	3/13/2020 6:48 PM	WIN File	3,145,728 KB
system_image.emmc.win.sha2	3/13/2020 6:48 PM	SHA2 File	1 KB

FIGURE 6. Important archives in NANDroid Backup from forensic point of view (Highlighted).



Name	Size	Packed	Type	Modified
..			File folder	
storage			File folder	3/8/2020 12:30:24 PM
manifests			File folder	3/8/2020 12:30:24 PM
extensions			File folder	3/8/2020 12:30:26 PM
datareporting			File folder	3/8/2020 12:30:21 PM
browser-extension-data			File folder	3/8/2020 12:30:26 PM
webappsstore.sqlite-wal	0	0	SQLITE-WAL File	3/8/2020 8:32:18 PM
webappsstore.sqlite-shm	32,768	32,768	SQLITE-SHM File	3/10/2020 9:43:49 PM
webappsstore.sqlite	98,304	98,304	SQLITE File	3/8/2020 12:33:44 PM
times.json	27	27	JSON File	3/8/2020 12:30:21 PM
suggestedsites.json	1,649	1,649	JSON File	3/8/2020 12:30:24 PM
storage-sync.sqlite	131,072	131,072	SQLITE File	3/10/2020 9:43:52 PM
storage.sqlite	512	512	SQLITE File	3/8/2020 12:30:24 PM
SiteSecurityServiceState.txt	0	0	Text Document	3/8/2020 4:51:43 PM
signons.sqlite	327,680	327,680	SQLITE File	3/8/2020 12:30:23 PM
sessionstore.old	41	41	OLD File	3/10/2020 7:59:16 PM
sessionstore.js	41	41	JavaScript File	3/10/2020 9:43:52 PM
sessionstore.bak	41	41	BAK File	3/10/2020 7:57:53 PM
sessionCheckpoints.json	90	90	JSON File	3/10/2020 9:43:49 PM
SecurityPreloadState.txt	0	0	Text Document	3/8/2020 4:51:43 PM

FIGURE 7. Tor application files inside NANDroid Backup archive.

the device was rooted and its storage including ADB logs, Zram and memory images were analyzed for artifacts.

1) UN-ROOTED DEVICE - STORAGE ANALYSIS

On an un-rooted device, analysis of storage (including ADB logs) does not yield any significant evidence of user browsing activities except downloaded files and application related files Analysis of ADB logs (Dumpsys and Logcat service logs) only show underlying activities of Tor application on the device including timestamps as shown in Fig 5.

2) NANDROID BACKUP - STORAGE ANALYSIS

We dumped the **org.torproject.torbrowser** directory from ***/data/data*** folder in user data archive available in the NANDroid backup we performed as shown in Fig 6 & 7. Analysis of the files using HxD, Notepad, and DB Browser for SQLite yields only Bookmarks, timestamps, and Tor circuit information from the NANDroid backup. No user browsing information was retrieved from the NANDroid backup except downloaded files. ADB Logs were not available in NANDroid backup.

TABLE 8. Tor only artifacts from Zram on Android 10.

.No.	Type of Artifact(s)
1	Application related paths
2	Application related loaded configuration files
3	Application used functions and resources
4	Application related Blocklists and Extensions data (included timestamps)
5	SQlite/DB Files; Tables names and application performed DB operations
6	Tor control port
7	Router's information including IP Addresses, nicknames, last available timestamps, Public keys used by Tor Router
8	Circuit related information
9	User-agent info (Mozilla/5.0)
10	Bookmark data
In case, application was uninstalled immediately after recent browsing, you may find a few:	
1	URLs and domain names
2	Website components traces (.js, .cssetc)
3	Downloaded files along with their local path
4	Uploaded files remnants
5	Login email address traces

3) ROOTED DEVICE - STORAGE ANALYSIS (INCLUDING ADB LOGS)

As in NANDroid backup, rooting the device allows us to access the Tor application root directory **/data/data/org.torproject.torbrowser/** on filesystem using Root Browser application and MOBILEdit Forensic Express. Analysis of the files using HxD, Notepad, and DB Browser for SQLite tools yield only bookmarks, timestamps, and Tor circuit information. No user browsing information was retrieved from the rooted android device except downloaded files. Analysis of ADB logs only shows underlying activities of the Tor application on the device including timestamps.

4) ROOTED DEVICE - ZRAM ANALYSIS

As per our existing knowledge and research, this area of Android device is explored for the first time to retrieve browsing and other application-related evidence because private browsers do not offer 100% privacy in terms of user browsing history as they leave many artifacts in RAM/memory. As Zram is a part of our device's physical RAM, its analysis revealed potential evidence of illicit browsing activities from the forensic point of view.

a: TOR ONLY ARTIFACTS

During this stage, we analyzed the artifacts left on Zram during the Tor browser's installation and execution without any browsing and uninstallation. Summary of all the artifacts retrieved during these activities are listed in Table 8 below:

b: BROWSING ARTIFACTS

i. Browser Open

Our analysis uncovers most of the websites/URLs and domain names we visited in our sample investigative scenario This includes few webpage components, redirected/visited URLs information; Downloaded files information including

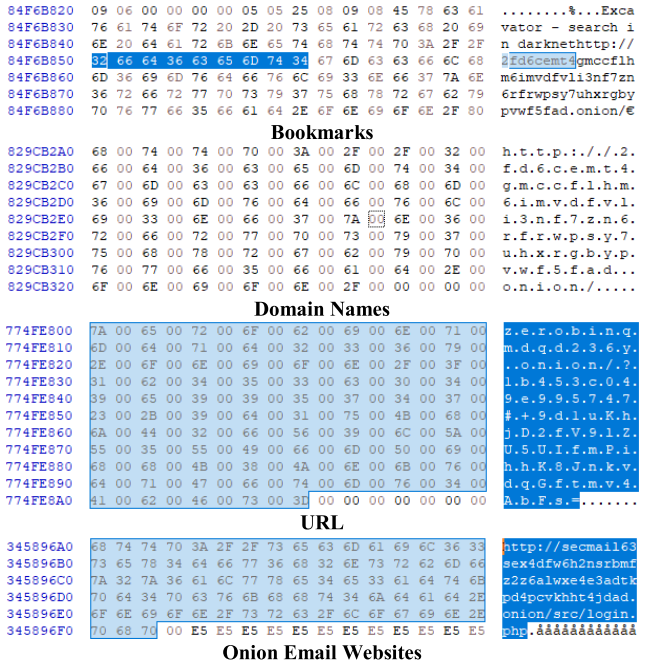


FIGURE 8. User browsing traces in Zram of Android 10 during browser open.

filename, URLs, and local paths; most of the search queries we performed, and clipboard content from Tor; traces of few email addresses, and usernames used for login and communication. No passwords and email content were found, but session information, timestamps of few visited websites, and bookmarked websites information were found. In application-related traces, we found application-related file paths, loaded application files, functions, resources, SQLite DB Tables and operations, Tor control port, routers info, circuit Info, public keys, router's nicknames, User-agent. Some of these artifacts are shown in Fig 8.

ii. Browser Closed

Analysis in this case only reveals traces of very few visited websites/URLs and domain names including few webpage components and redirected/visited URLs information; Downloaded files information contains only local path and filenames; No search queries and clipboard content was found. Very few traces of email addresses used for login and communication were found, but no password and email content were found. In application-related traces, a small number of file paths and only some loaded application files were found. Summary of all the Tor browsing artifacts retrieved from Android 10 Zram is listed in Table 9.

5) ROOTED DEVICE - MEMORY ANALYSIS

In this analysis, we only covered two types of activities because of our memory acquisition tool's limitation as mentioned in section IV(II). We analyzed the Tor-only artifacts and user browsing artifacts during the "Browser Open" scenario.

TABLE 9. Browsing related artifacts from Zram on Android 10.

Sr.	Websites	Activities	Artifact(s) found while Browser Open	Artifact(s) found while Browser Closed
1	Hidden Wiki	1. Browsing 2. Whistleblowing link clicked	No artifact found	No artifact found
2	Ahmia	1. Browsing 2. Search query **"sell official data"** 3. Clicked first result & redirected to **5j7saze5byfqccf3.onion/data/experimental/main/** 4. Download **components-arm64.yml.xz** from URL	▪ Website/URL traces ▪ Visited/Redirected URLs ▪ Website components (js,css) ▪ Search query traces ▪ Downloaded file & URL traces	Downloaded filename and local path on a storage
3	DuckDuckGo	1. Browsing 2. Search query **"sell official data"**	No artifact found	No artifact found
4	Excavator	1. Browsing 2. Search query **"sell official data"**	▪ Website/URL traces ▪ Search query traces	No artifact found
5	Google Drive	1. Browsing only after login to mail.google.com using Google credentials*	No artifact found	No artifact found
6	MEGA	1. Browsing 2. Login 3. Upload IMG-20210122-WA0005.jpg 4. Right click and get sharing link 5. Copy link to clipboard	▪ Website/URL traces ▪ Website components (js,css) ▪ Uploaded file traces ▪ Clipboard operation traces	▪ Website/URL traces ▪ Uploaded file traces
7	ZeroBin	1. Browsing 2. Mega.nz file sharing link pasted 3. Get Paste link containing **"/?a3e1481092fb04b9"**	▪ Only domain name	▪ Only domain name
8	StrongHold Paste	1. Browsing 2. Set the Paste title to **"Pix"** with content **"https://goo.gl/xZgh1qu"** 3. Password-protect it 4. Get paste link containing **"/pocsxm1d5/2uo2vh"**	▪ Only domain name	Only domain name
9	SecureDrop	1. Browsing 2. Click **"Get started"** and get codename **"unloving cornflake ecosphere decipher trifocals scotch reiterate"** 3. Click **"Submit documents"** 4. Upload **IMG-20210122-WA0005.jpg**	▪ Only domain name	No artifact found
10	StealthPay	1. Browsing only	▪ Website/URL traces ▪ Website components (js,css)	▪ Only domain name
11	Keybase	1. Browsing 2. click **"Send secure message"** it will redirect to **"play.google.com"** for **keybase** android apk installation.	▪ Visited/Redirected URLs	No artifact found
12	SecMail	1. Browsing 2. Login 3. Check Emails received from **Gmail** and **Outlook** email addresses 4. Reply was sent to Gmail as shown below: Subject: **"Re: Impt Data"** Body: **"Find here: https://goo.gl/xZgh1qu"**	▪ Website/URL traces ▪ Website components (js,css)	No artifact found
13	Gmail	1. Browsing 2. Login 3. Email composed and sent to **adamjames555@secmail.pro** as shown below: Subject: **"Impt Data"** Body: **"Please share link to receive data"**	▪ Website/URL traces ▪ Website components (js,css) ▪ Login Email Address traces ▪ Timestamps ▪ Sessions IDs ▪ Cookies ▪ Response Headers	Only Login email address traces
14	Outlook	1. Browsing 2. Login 3. Email composed and sent to **adamjames555@secmail.pro** as shown below: Subject: **"Imp Data"** Body: **"Please share link to receive data"**	▪ Website/URL traces ▪ Website components (js,css) ▪ Login Email Address traces ▪ Sessions IDs	Only Login email address traces
15	Skype	1. Browsing 2. Login 3. Visited **Account overview** 4. **web.skype.com** opened but **"browser not supported"** message received on webpage	▪ Website/URL traces ▪ Login Email Address traces ▪ Sessions IDs	Only secure.skype.com domain name
16	Galaxy3	1. Browsing 2. Login	▪ Website/URL traces ▪ Website components (js,css)	▪ Login Email Address traces

TABLE 9. *(Continued.)* **Browsing related artifacts from Zram on Android 10.**

		3. **/Settings** link visited 4. Blogs link **"/blog/owner/aj555"** visited	▪ Login Email Address traces ▪ Username	
17	PirateBay	1. Browsing 2. search **"privacy"** with **Application** check box clicked on webpage 3. Privacy Sheild URL opened from results 4. Get Magnet Link **"magnet:?xt=urn:btih:2A3B…"**	▪ Website/URL traces ▪ Website components (js,css) ▪ Downloaded magnet filename & URL traces	▪ Website/URL traces ▪ Website components (js,css) ▪ Downloaded magnet filename & URL traces

TABLE 10. **Tor only artifacts from Memory on Android 10.**

S.No.	Type of Artifact(s)
1	Application related paths
2	Application related loaded configuration files
3	Application used functions and resources
4	Application related Blocklists and Extensions data (included timestamps)
5	SQlite/DB Files; Tables names and application performed DB operations
6	Tor control port
7	Router's information including IP Addresses, nicknames, last available timestamps, Public keys used by Tor Router
8	Circuit related information
9	User-agent info (Mozilla/5.0)
10	Bookmark data
	In case, application was uninstalled immediately after recent browsing, you may find few:
1	URLs and domain names
2	Website components traces (.j.e. css etc)
3	Downloaded files along with their local path
4	Uploaded files remnants
5	Login email address traces

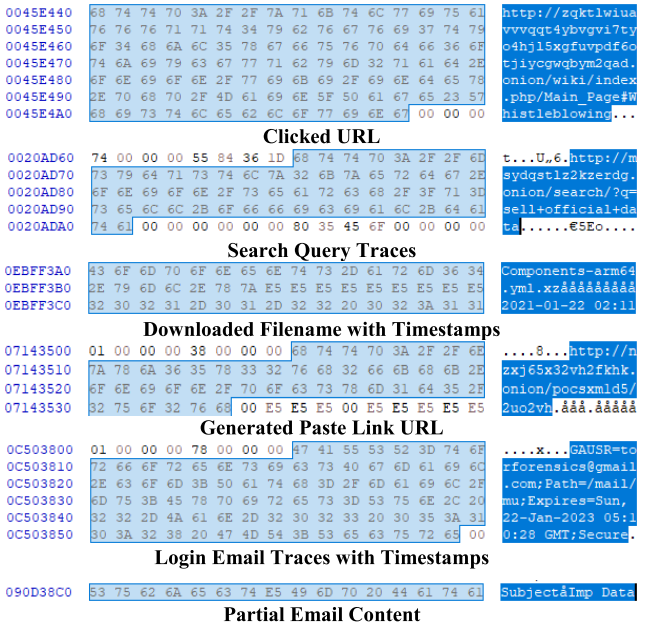


FIGURE 9. **User browsing traces in memory of Android 10 during browser open.**

a: TOR ONLY ARTIFACTS

Unlike Zram, we only analyzed the artifacts left on the memory Tor browser that was opened either with or without any browsing activity performed. Summary of all the artifacts retrieved during these activities are listed in Table 10 as shown below:

b: BROWSING ARTIFACTS

i. Browser Open

Analysis reveals significant information about user browsing activities including visited websites/URLs including webpage components and redirected/visited URLs information; Downloaded files information including filename, URL, timestamps, and local paths; Uploaded file information; all search queries performed & clipboard content from Tor; Traces of most email addresses &usernames used for login and communication, and few passwords were also found but no email content was found. In addition, session information and timestamps of few visited websites were also found. We also found bookmarked websites. In application-related traces, we found file paths, loaded application files, functions, resources, SQLite DB Tables and operations, tor control port, routers info, circuit Info, public keys, router's nicknames, User-agent info. Some of these artifacts we discovered are shown in Fig 9.

ii. Browser Closed

Analysis in this state is not possible due to our tool's limitation so it did not reveal anything. The summary of all the Tor browsing artifacts we retrieved from Android 10 RAM is listed in Table 11.

All the browsing artifacts gathered from Android 10's experimental setup are listed in Table 12.

VI. COMPARISON WITH EXISTING RESEARCH

A vast amount of research has been conducted on the security and privacy of the Tor network, but limited research has been performed in the field of Tor forensics especially on the latest Windows and Android OS builds.

We only found three studies focused on forensics analysis of the Tor browser performed on different Windows OS version(s):

1) On Windows 10 version 1709 by Warren [8] – this study examined the registry, storage, and memory after normal

TABLE 11. User browsing artifacts from memory on Android 10.

Sr.	Websites	Activities	Artifact(s) found while Browser Open
1	Hidden Wiki	1. Browsing 2. Whistleblowing link clicked	▪ Website/URL traces ▪ Visited/Redirected URLs ▪ Website components (js,css) ▪ SOCKS socket traces ▪ Response Headers ▪ Bookmarked information
2	Ahmia	1. Browsing 2. Search query **"sell official data"** 3. Clicked first result & redirected to **5j7saze5byfqccf3.onion/data/experimental/main/** 4. Download **components-arm64.yml.xz** from URL	▪ Website/URL traces ▪ Visited/Redirected URLs ▪ Website components (js,css) ▪ Search query traces ▪ SOCKS socket traces ▪ Downloaded filename & URL traces ▪ Download timestamps
3	DuckDuckGo	1. Browsing 2. Search query **"sell official data"**	▪ Website/URL traces ▪ Visited/Redirected URLs ▪ Website components (js,css) ▪ Search query traces ▪ SOCKS socket traces
4	Excavator	1. Browsing 2. Search query **"sell official data"**	▪ Website/URL traces ▪ Visited/Redirected URLs ▪ Website components (js,css) ▪ Search query traces ▪ SOCKS socket traces ▪ Bookmarked information
5	Google Drive	1. Browsing only after login to mail.google.com using Google credentials*	▪ Website/URL traces ▪ Website components (js,css) ▪ Login Email address traces ▪ Response Headers
6	MEGA	1. Browsing 2. Login 3. Upload IMG-20210122-WA0005.jpg 4. Right-click and get sharing link 5. Copy link to clipboard	▪ Website/URL traces ▪ Website components (js,css) ▪ Uploaded file information ▪ Clipboard operation traces ▪ Local upload temp folder ▪ SOCKS socket traces ▪ Login Email address traces
7	ZeroBin	1. Browsing 2. Mega.nz file-sharing link pasted 3. Get Paste link containing "**/?a3e1481092fb04b9"**	▪ Website/URL traces ▪ Website components (js,css) ▪ Clipboard operation traces ▪ Generated Filesharing/Paste URL information traces ▪ SOCKS socket traces
8	StrongHold Paste	1. Browsing 2. Set the Paste title to **"Pix"** with content **"https://goo.gl/xZgh1qu"** 3. Password-protect it 4. Get paste link containing **"/pocsxm1d5/2uo2vh"**	▪ Website/URL traces ▪ Website components (js,css) ▪ Clipboard operation traces ▪ Generated Filesharing/Paste URL information traces ▪ SOCKS socket traces ▪ Response Headers ▪ Timestamps
9	SecureDrop	1. Browsing 2. Click **"Get started"** and get codename **"unloving cornflake ecosphere decipher trifocals scotch reiterate"** 3. Click **"Submit documents"** 4. Upload **IMG-20210122-WA0005.jpg**	▪ Website/URL traces ▪ Website components (js,css) ▪ Generated Random Username traces ▪ SOCKS socket traces
10	StealthPay	1. Browsing only	▪ Website/URL traces ▪ Website components (js,css) ▪ SOCKS socket traces ▪ Response Headers ▪ PHP Session IDs
11	Keybase	1. Browsing 2. click **"Send secure message"** it will redirect to **"play.google.com"** for **keybase** android apk installation.	▪ Website/URL traces ▪ Visited/Redirected URLs ▪ Website components (js,css) ▪ SOCKS socket traces
12	SecMail	1. Browsing 2. Login 3. Check Emails received from **Gmail** and **Outlook** email addresses 4. Reply was sent to Gmail as shown below:	▪ Website/URL traces ▪ Website components (js,css) ▪ SOCKS socket traces ▪ Only partial received email traces

TABLE 11. *(Continued.)* **User browsing artifacts from memory on Android 10.**

		Subject: **"Re: Impt Data"** Body: **"Find here: https://goo.gl/xZgh1qu"**	
13	Gmail	1. Browsing 2. Login 3. Email composed and sent to **adamjames555@secmail.pro** as shown below: Subject: **"Impt Data"** Body: **"Please share link to receive data"**	▪ Website/URL traces ▪ Website components (js,css) ▪ SOCKS socket traces ▪ Login email address traces ▪ Login Timestamps ▪ Cookies ▪ Response Headers ▪ Only To: & Subject: header of emails found
14	Outlook	1. Browsing 2. Login 3. Email composed and sent to **adamjames555@secmail.pro** as shown below: Subject: **"Imp Data"** Body: **"Please share link to receive data"**	▪ Website/URL traces ▪ Website components (js,css) ▪ Login email address & Password traces ▪ Session Information ▪ Cookies ▪ Response Headers ▪ Only Subject: header & body of emails found
15	Skype	1. Browsing 2. Login 3. Visited **Account overview** 4. **web.skype.com** opened but the **"browser not supported"** message received on the webpage	▪ Website/URL traces ▪ Website components (js,css) ▪ Login email address traces ▪ Base64 encoded Session Token ▪ X-CSRF Token ▪ Cookies ▪ Timestamps ▪ Response Headers
16	Galaxy3	1. Browsing 2. Login 3. **/Settings** link visited 4. Blogs link **"/blog/owner/aj555"** visited	▪ Website/URL traces ▪ Website components (js,css) ▪ Login email address & Password traces ▪ Usernames ▪ Timestamps ▪ Session Token ▪ SOCKS Username, Password
17	PirateBay	1. Browsing 2. search "**privacy"** with **the Application** check box clicked on the webpage 3. Privacy Sheild URL opened from results 4. Get Magnet Link **"magnet:?xt=urn:btih:2A3B…"**	▪ Website/URL traces ▪ Website components (js,css) ▪ Search Results traces ▪ Downloaded magnet filename & URL traces

websites e.g. google.com were visited. They discovered mostly application-related artifacts and were only able to retrieve bookmarks (browsing artifacts) from storage. They did not include any significant effort for discovering browsing artifacts from registry and memory.

2) On Windows 8.1 by Jadoon *et.al.* [9] - this research examined the registry, storage, and memory and included a lot of effort into the exploration of user-browsing artifacts but lacked the exploration of Tor application-related artifacts.
3) On Windows 10 version 1703 by Muir *et.al.* [12] – this study also examined the registry, storage, and memory for Tor browser artifacts and was able to uncover most of the application-related and browsing artifacts for normal websites. However, Tor-based websites and its related artifacts were missing. Also, this study was limited to Windows and did not cover Tor for Android.

In contrast to the above-mentioned research work, we have performed a forensic analysis of the latest Tor browser version on the latest Windows build i.e. version 20H2 (October 2020 build), and in various directions (i.e. registry, storage, memory). We also include normal and Tor-based

TABLE 12. **Summary of all user browsing artifacts from Android 10 device.**

Browsing Artifacts	Evidence Locations			
	Storage	RAM	Zram	ADB Logs
URLs	No	Yes	Yes	No
Website Content	No	Yes	Few	No
Search Queries	No	No	Few	No
Bookmarks	Yes	Yes	Yes	No
Cookies	No	No	No	No
Email Addresses	No	Yes	Rare	No
Email Content	No	Yes	No	No
Usernames	No	No	No	No
Passwords	No	No	No	No
Downloaded Files	Yes	Yes*	No	No
Browsing Timestamps	No	Yes	Few	No
Usage/ Session Timestamps	Yes	No*	No*	Yes

TABLE 13. Detailed comparison of existing work and adopted methodology of Tor browser forensics.

Related work	OS Platform(s)	Tor Browser version	Evidence Venues Explored	Installation	No-Browsing Execution	Execution &Browsing		Uninstallation
						Browser Open	Browser Closed	
Nedaa Al Barghouthy et.al.	Android 2.3.4	V2.28	Storage	X	X	✓	X	X
Nedaa Al Barghouthy et.al.	Android 4.1.1	V2.28	Storage	X	X	✓	X	X
C. Meda et.al	Android 5.0	V15.0.1-RC-3	Storage	✓	✓	✓	X	✓
R.Nelson et.al	Windows 7	V7.0.5	Storage	✓	✓	✓	✓	X
			Registry	✓	X	✓	X	X
A. Jadoon et.al	Windows 8.1	V7.0.2	Storage	X	X	✓	✓	X
			Registry	✓	X	X	X	✓
			Memory	X	✓	✓	✓	X
M.Muir et.al	Windows 10 Version 1703	V7.5.2	Registry	X	✓	✓	✓	X
			Storage	✓	✓	✓	✓	✓
			Memory	✓	✓	✓	✓	✓
A. Warren	Windows 10 October 2017 build	V5.0	Registry	✓	X	X	X	X
			Storage	✓	✓	✓	X	X
			Memory	✓	✓	✓	X	X
Proposedwork	Windows 10 Version 20H2 October 2020	V10.0.7	Storage	✓	✓	✓	✓	X
			Registry	✓	✓	✓	✓	✓
			Memory	✓	✓	✓	✓	✓
	Android 10 June 2020 update	V68.7.0	Storage	✓	✓	✓	✓	✓
			ADB Logs	✓	✓	✓	✓	✓
			Zram	✓	✓	✓	✓	✓
			Memory	✓	X	✓	X	X

websites and retrieve both browsing and application-related artifacts.

Similarly, for Android OS, previous research works have only examined storage and file systems for Tor browser artifacts and generally on rooted Android devices. The only exception is Al Barghouthy and Marrington [4] in which the NANDroid backup is also examined. In contrast, our research work explores four distinct areas of Android 10 OS (i.e. storage, ADB Logs, Zram, and memory) and three different device states (i.e. Un-rooted, Rooted, and NANDroid backup) for Tor browser application-related and browsing artifacts.

A detailed comparison of proposed and existing work can be seen in Table 13. We have made an effort to cover every possible scenario an investigator may face during the forensic analysis of Tor on both platform(s) with tools that are either open-source (due to limited budget) or recognized as an industry-standard. This can help forensic investigators and developers reproduce our results

VII. RECOMMENDATIONS FOR TOR PROJECT DEVELOPERS

Tor developers have implemented numerous **decoy** settings to provide fail-safe anonymity and privacy. However, several browser-related settings and timestamps are stored in plain-text files which can forensically reveal usage patterns of the Tor browser. In this regard, the inclusion of a mechanism for the storage of browser-based settings in encrypted files is recommended. These files should only be decrypted by the browser binary while it is executing. Secondly, as we have shown a significant amount of user browsing information can be retrieved from Zram (in Android only) and RAM (in Windows and Android). This can have a significant impact on a user's privacy and this issue should be addressed in upcoming releases. A memory encryption scheme that can encrypt and decrypt ''Tor only'' and ''User browsing'' artifacts from RAM is recommended.

VIII. CONCLUSION AND FUTURE WORK

This **paper** investigated artifacts from the **Tor** privacy browser on the latest Windows 10 and Android 10 devices to determine potential **areas** where evidence can be found. Our analysis suggests that the Tor browser leaves limited information about a user's browsing activity in the storage of both (Windows and Android) platforms. However, there is still ample evidence concerning the usage of the Tor browser in storage (including in ADB logs and registry). This work was explored **Android swap file (Zram) (which has not been analyzed before)** for evidence related to the Tor browser. A deeper analysis revealed that the knowledge and likelihood of extraction from Zram is approximately 60 percent.

This percentage can be considered good for an anonymous browser especially if there is not enough time and resources to explore the RAM.

Our results also show that the Tor browser leaves more artifacts in the RAM of Windows 10 OS than on the Android 10 platform. However, just like previous research, the probability of user attribution based on these artifacts is very little.

As part of our future work, we intend to carry out detailed network forensic analysis of the Tor circuit on Android 10 and Windows 10 platforms as limited research has been performed in this area. We also plan to perform a detailed forensic analysis of the Tor browser on iOS devices. Lastly, we would like to develop a specialized cross-platform module(s) for MobilEdit and other forensic tools for the acquisition and analysis of evidence from the Tor privacy browser.

REFERENCES

[1] Obstacles to Cybercrime Investigations. (2019). *Cybercrime Module 5 Key Issues*. UNODC.Org. [Online]. Available: https://www.unodc.org/e4j/en/cybercrime/module-5/key-issues/obstacles-to-cybercrime-investigations.html

[2] J. Porup, *What is the Tor Browser? And How the Dark Web Browser Works*. New Delhi, India: CSO, 2019.

[3] N. A. Barghouthy, A. Marrington, and I. Baggili, "The forensic investigation of Android private browsing sessions using orweb," in *Proc. 5th Int. Conf. Comput. Sci. Inf. Technol.*, Mar. 2013, pp. 33–37.

[4] N. Al Barghouthy and A. Marrington, "A comparison of forensic acquisition techniques for Android devices: A case study investigation of orweb browsing sessions," in *Proc. 6th Int. Conf. New Technol., Mobility Secur. (NTMS)*, Mar. 2014, pp. 1–4.

[5] C. M. Meda and Epifani, *Study and Analysis of Orweb (and Orfox) Anonymizer(S) on Android Devices*. Oxford, U.K.: Dfrws Eu 2016.

[6] S. Teng and C. Wen, "A forensic examination of anonymous browsing activities," *Forensic Sci. J.*, vol. 17, no. 1, pp. 1–8, 2018.

[7] M. Asim, M. F. Amjad, W. Iqbal, H. Afzal, H. Abbas, and Y. Zhang, "AndroKit: A toolkit for forensics analysis of web browsers on Android platform," *Future Gener. Comput. Syst.*, vol. 94, pp. 781–794, May 2019.

[8] A. Warren, "Tor browser artifacts in windows 10," Tech. Rep., 2017.

[9] A. K. Jadoon, W. Iqbal, M. F. Amjad, H. Afzal, and Y. A. Bangash, "Forensic analysis of tor browser: A case study for privacy and anonymity on the web," *Forensic Sci. Int.*, vol. 299, pp. 59–73, Jun. 2019.

[10] R. Nelson, A. Shukla, and C. Smith, "Web browser forensics in Google Chrome, Mozilla Firefox, and the tor browser bundle," in *Digital Forensic Education* (Studies in Big Data), vol. 61, X. Zhang and K. K. Choo, Eds. Cham, Switzerland: Springer, 2019, doi: 10.1007/978-3-030-23547-5_12.

[11] G. B. Satrya and F. Kurniawan, "A novel Android memory forensics for discovering remnant data," *Int. J. Adv. Sci., Eng. Inf. Technol.*, vol. 10, no. 3, p. 1008, Jun. 2020.

[12] M. Muir, P. Leimich, and W. J. Buchanan, "A forensic audit of the tor browser bundle," *Digit. Invest.*, vol. 29, pp. 118–128, Jun. 2019.

[13] G. Horsman, B. Findlay, J. Edwick, A. Asquith, K. Swannell, D. Fisher, A. Grieves, J. Guthrie, D. Stobbs, and P. McKain, "A forensic examination of web browser privacy-modes," *Forensic Sci. Int., Rep.*, vol. 1, Nov. 2019, Art. no. 100036.

[14] K. Satvat, M. Forshaw, F. Hao, and E. Toreini, "On the privacy of private browsing—A forensic approach," in *Data Privacy Management and Autonomous Spontaneous Security*. Berlin, Germany: Springer, 2013, pp. 380–389.

[15] M. Alfosail and P. Norris, "Tor forensics: Proposed workflow for client memory artefacts," *Comput. Secur.*, vol. 106, Jul. 2021, Art. no. 102311.

[16] *Android Debug Bridge (ADB) | Android Developers*, Android Developers, Palo Alto, CA, USA, 2021.

[17] *Logcat Command-Line Tool*, Android Developers, Palo Alto, CA, USA, 2021.

[18] *Dumpsys*, Android Developers, Palo Alto, CA, USA, 2021.

[19] H. Bilal. (2019). *What is the Difference Between a Rooted and Unrooted Android? Quora*. [Online]. Available: Quora.com

[20] D. Stieben, *What is a Nandroid Backup and How Exactly Does it Work*. Washington, DC, USA: MUO, 2014.

[21] O. A. V. Ravnås. (2021). *Frida-Tools*. PyPI. [Online]. Available: https://pypi.org/project/frida-tools/

[22] O. A. Ravnås. (2021). *Frida A World-Class Dynamic Instrumentation Framework*. [Online]. Available: https://frida.re/

[23] K. Kent, S. Chevalier, T. Grance, and H. Dang, *Guide to Integrating Forensic Techniques into Incident Response*. Gaithersburg, MD, USA: NIST Technical Series Publications, 2006.

[24] M. Sahu. (2021). *How to Unlock Bootloader on Xiaomi Mi A3—Mi Official*. [Online]. Available: https://myphoneupdate.com/unlock-bootloader-xiaomi-mi-a3/

[25] M Sahu. (2020). *How to Install TWRP Recovery on Xiaomi Mi A3?* [Online]. Available: https://myphoneupdate.com/twrp-recovery-xiaomi-mi-a3/

[26] M. Bhardwaj. (2020). *How to Install Magisk and Root Android Devices With it*. [Online] Available: https://www.thecustomdroid.com/install-magisk-root-android-devices/

MUHAMMAD RAHEEL ARSHAD received the B.S. degree in computer engineering from COMSATS University Islamabad, in 2011. He is currently pursuing the M.S. degree in information security with the School of Electrical Engineering & Computer Science (SEECS), NUST, Islamabad.

From 2011 to 2014, he worked as a Software Engineer with the System Department of Internet Service Provider Company based, Islamabad. Since 2014, he is working as a Consultant (Cyber Security & Information Assurance) in a private firm. He holds accredited certifications in Cyber Security i.e., CEH, CHFI, ECSA from EC-Council, and CCDA from GlobalACE-CyberSecurity Malaysia. He also received Common Criteria (CC) training from CyberSecurity Malaysia, in 2018. He is also working as a Researcher in the field of digital forensics and investigations with the School of Electrical Engineering & Computer Science (SEECS), NUST, Islamabad, since 2018. His research interests include penetration testing, digital forensics, web application security, and privacy applications.

MEHDI HUSSAIN (Member, IEEE) received the B.S. and M.S. degrees in computer science, in 2006 and 2011, respectively, and the Ph.D. degree in information security (information hiding) from the University of Malaya, Malaysia, in 2017. From 2006 to 2014, he worked at various positions in renowned U.S.-based software houses in Pakistan. He has experience in various video/image-based projects (i.e., video conferencing, H.264 encoding, and license plate recognition). He is currently an Assistant Professor with the School of Electrical Engineering & Computer Science (SEECS), National University of Science and Technology (NUST), Pakistan. His research interests include information hiding, digital forensic, multimedia security, and video compression.

HASAN TAHIR (Senior Member, IEEE) received the B.E. degree in software engineering from Bahria University, Islamabad, Pakistan, the M.S. degree in software engineering from the College of EME, NUST, and the Ph.D. degree in information security from the University of Essex, U.K. He is currently an Assistant Professor with the School of Electrical Engineering & Computer Science (SEECS), NUST. He specializes in computer security and the IoT. He actively researches applications of cryptography in one-to-one and group settings. He teaches courses related to applied cryptography, cyber security, and information security management, cloud computing security, software engineering, software requirements analysis, and design. His research interest includes the use of physically unclonable functions for securing a group of devices. He has served as a committee member in many renowned IEEE conferences. He was a recipient of the University of Essex Doctoral Scholarship Award.

SANA QADIR received the bachelor's degree (Hons.) in IT from the University of Southern Queensland, Australia, and the M.Sc. and Ph.D. degrees from International Islamic University Malaysia, in 2010 and 2016, respectively. She is currently working as an Assistant Professor with the School of Electrical Engineering & Computer Science (SEECS), NUST. She also has more than three years' experience working as a Research and Development Software Engineer. Her research interests include network security and applied cryptography.

FARAZ IQBAL AHMED MEMON received the B.E. degree in electrical and the M.E. degree in telecommunications and computer networks from the Mehran University of Engineering and Technology, Sindh, Pakistan. He is currently pursuing the Ph.D. degree in information security with NUST, Islamabad, Pakistan.

Keeping an experience of more than 20 years in the domain of digital forensics and cyber security. Holding Core Certifications, such as CHFI, CEH, Access Data Certified Investigator (ADCI), CCNP, CS-CISecS, CS-CFWS, CCNA, CNSS-4011 (National Security Agency-USA), MCSA (more than MSG), MCSA, MCP, SCP, MS (virtualization), CloudU Certified, JNCIA-ER, JNCIS-ER, JNCIA-EX, JNCIA-JUNOS, JNCIS-SEC (Security), Brocade (BCEFP), and Palo Alto Networks (ACE). Over the years worked with international LEA's Platform, such as Australian Federal Police and UNODC. He worked as a Forensics Investigator and a Cyber Security Specialist in different LEA's Platform's at a national level.

YOUSRA JAVED received the B.S. and M.S. degrees in information and communication systems engineering from the National University of Sciences and Technology, Pakistan, in 2011, and the Ph.D. degree in computing and information systems from the University of North Carolina, Charlotte, NC, USA, in 2017. From 2012 to 2014, she worked as a Research Assistant with the Laboratory of Information Integration Security and Privacy, Charlotte. After receiving her Ph.D., she is currently working as an Assistant Professor with the School of Information Technology, Illinois State University, USA. She has authored more than 20 articles. Her research interests include usable security, privacy, and human–computer interaction.