Digital Investigation of Tor browser on Windows 10 and Android 10

By

Muhammad Raheel Arshad

Fall 2018-MS(IS) - 00000278141

Supervisor

Dr. Mehdi Hussain

Department Of Computing

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Information Security (MSIS)

In

School of Electrical Engineering and Computer Science

National University of Sciences and Technology (NUST),

Islamabad, Pakistan

(February 2022)

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis entitled "Digital Investigation of Tor browser on Windows 10 and Android 10" written by MUHAMMAD RAHEEL ARSHAD, (Registration No 00000278141), of SEECS has been vetted by the undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/M Phil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: ______ Hussain ______

Name of Advisor: Dr. Mehdi Hussain

Date: 04-Feb-2022

Signature (HOD): ______________________

Date: ______________________

Signature (Dean/Principal): ______________________

Date: ______________________

Approval

It is certified that the contents and form of the thesis entitled "Digital Investigation of Tor browser on Windows 10 and Android 10" submitted by MUHAMMAD RAHEEL ARSHAD have been found satisfactory for the requirement of the degree

Advisor : Dr. Mehdi Hussain

Signature: ____________________

Date: 04-Feb-2022

Committee Member 1:Dr. Dr Hasan Tahir

Signature: ____________________

Date: 04-Feb-2022

Committee Member 2:Dr. Qaiser Riaz

Signature: ____________________

Date: 04-Feb-2022

Committee Member 3:Dr. Sana Qadir

Signature: ____________________

Date: 04-Feb-2022

Dedication

I dedicate this work to my **parents** and **spouse** for all their trust in my decisions and giving me the freedom to achieve this milestone.

Certificate of Originality

I hereby declare that this submission titled "Digital Investigation of Tor browser on Windows 10 and Android 10" is my own work. To the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics, which has been acknowledged. I also verified the originality of contents through plagiarism software.

Student Name: MUHAMMAD RAHEEL ARSHAD

Student Signature: ______________

Acknowledgment

Thanks to the Almighty ALLAH for letting me pursue and fulfill this research work. I could not have achieved anything without HIS utmost support and countless blessings.

I am thankful to my beloved parents for their support throughout my educational career. I dedicate this work to my lovely parents, spouse, honorable teachers, my dear siblings, and friends. They have always supported and encouraged me to do the best in all matters of life.

I am truly grateful to my supervisor Dr. Mehdi Hussain for his guidance, supervision, and encouragement to complete this work. He has been a great source of inspiration for me during my research, thank you so very much. I am obliged to all my respectable teachers for providing me withtheir valuable time and considerations. I believe that this work would not have been possible without their guidance and expert suggestions.

I am also thankful to my committee members Dr. Hasan Tahir, Dr. Qaiser Riaz, Dr. Sana Qadir and ex-committee member Dr. Yousra Javed for their contribution and timely suggestions toward the successful completion of this thesis. Thanks to all my friends and colleagues at SEECS for their help and suggestions.

Despite all the assistance provided by the supervisor, committee members, and others, I take the responsibility for any errors and omissions which may unwittingly remain.

Muhammad Raheel Arshad

Contents

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Internet, Computers and Mobile phones especially smartphones roll out as a lifeline of our society since last decade with plenty of applications in business, education, gaming, and research. However, one of the major issues faced using Internet is its lack of privacy and security since it is still possible for an eavesdroppers/attackers to intercept communication between users. As a result, the number of cyber crime incidents i.e. exploiting confidentiality has increased over time. Therefore, users have become more anxious about the security of their communication. In this regard, some users have preferred to use private browsers for safeguarding their communication privacy. Tor privacy browser is one of the most famous and extensively used privacy browser that is based on *The Onion Router (Tor)* network to sustain anonymity over the Internet. However, Tor browser at all times remains a major obstacle in the network centric cybercrime investigations due to sophisticated level of anonymity provided over specialized overlay network. In this study, we have investigated the Tor privacy browser artifacts on Windows 10 and Android 10 devices and identify the potential areas in an operating system where evidence can be found that will help the investigators in e-discovery. In this research, we investigated the artifacts left by theTor privacy browser on the Registry, Storage, and Memory of Windows 10 device; and similarly we investigated the Memory, Storage, ADB logs and Zram for Android 10 device to find out how it left the evidence on these areas in operating before, during, and after usage. Analysis of our results confirmed against claims of user's privacy and anonymity made by the Tor Project. Because our investigation on both operating systems uncover significant number of evidence about user browsing activities while Tor browser was left open, in use and even after closing the browser. This study proposed an investigative methodology to acquire and analyze the Tor browser artifacts from different areas of targeted operating system which will serve as a foundation for expanding this research to conduct forensic analysis of additional privacy browsers and enhances the investigator's competency to achieve easier application's forensic investigation process.

Keywords

Tor, browser forensic, Windows 10, Windows forensic, Android forensic, privacy, Android 10, anonymous browser, privacy browsers, The Onion Router

Chapter 1

Introduction

In this chapter the topics we cover in subsequent sub-sections are as follows:

- Section 1.1 important terms and concept
- Section 1.2 highlights motivation of this research
- Section 1.3 explain problem statement
- Section 1.4 define goals and objectives
- Section 1.5 provides scope of this research
- Section 1.6 describes the challenges faced in this research
- Section 1.7 describes the formation of this thesis document

The prevalence of workstations, laptops, and smartphones is increasing day by day and these devices have now become a lifeline of our society. Ever since the introduction of smartphones in earlier days of 1994, IBM developed the Simon Personal Communicator (SPC) that surfaced as the first smartphone in history, and then later on in the year 2007, Apple Inc. become became the first modern-day smartphone manufacturer with their iPhone brand that is powered by a proprietary mobile operating system known as iOS [33]. Android was the next mobile operating system officially launched in 2008 [28]. It immediately became the most popular mobile operating system in the market due to its open-source licensing and wide range of mobile applications availability. These earlier mobile devices provide consumers with the ability to browse the Internet just as they were accustomed to using it on their PCs; that serves as a game-changer in the technological era and leads to instant adaptation of these modern devices a.k.a Smartphones.

Well similarly, in the case of computers and laptops, Microsoft Windows always remain the first choice for the users because it contains pre-loaded necessary software(s), a feature-rich user-friendly graphical user interface (GUI), and provides a wide range of peripheral compatibility using built-in drivers [29].

According to the statistics, Android shares 71.81% of the worldwide smartphone market and Windows shares 75.55% of the worldwide PC market in the first quarter of 2020-21 [30] Laptops and smartphones have been purchased in an almost equal ratio in 2019, 2020, and 2021 [31]

This widespread adoption of Android smartphones and Windows-based PCs/Laptops delivers an opportunity for businesses and industries to expand their productivity and resources. While, on the other hand, it has produced several problems for law enforcement agencies and other Internet users because these devices i.e., mostly Smartphone and Laptops has offered much more flexibility and agility to the cybercriminals; enabling them to launch sophisticated cyber-attacks that include exfiltrating confidential data, performing unauthorized modifications to Intellectual property (IP), perform digital frauds, and service disruptions.

One such problem that worried Law Enforcement Agencies the most is the **anonymity** that enables individuals (either malicious or benign) to engage in prohibited activities online without revealing themselves and/or their actions to others [2]; because anonymity allows them to constantly cover their tracks even over the public network owing to the use of VPNs and other privacy protection software(s).

Tor privacy browser is one of such privacy protection software(s) that is broadly used for achieving anonymity by Internet users both normal and cyber-criminals [33]. Ordinary Internet users take advantage of Tor to achieve privacy protection on the insecure Internet while cyber-criminals use it to cover their tracks while carrying out their illegal activities. The mechanism of Tor browser working is directing the encrypted Internet traffic via an overlay of layered private networks by establishing a circuit [3].

The digital investigation of a Tor network circuit is a very complex and tiresome mission because it involves not simply one node, it involves multiple nodes that are mostly outside the jurisdiction and geographical boundaries of a locality, city, or country. Although these matters can be simplified by investigating a captured suspect device (either smartphone or PC) to find the traces of illicit online activities using Tor privacy browser on a device. [1]

1.1 Important Terms and Concepts

In this section, we will briefly describe a few terms and terminologies used throughout this study.

Android — A Linux based mobile operating system is widely used today in every field. The current stable version today is Android 11 also called Android Q.

Windows — Famous operating system is widely used these days in homes and enterprises. Also quite famous for being under attack by cybercriminals. The current stable version in use today is Windows 10.

Digital Forensic — Area of forensic science that deals with identifying, acquiring, processing, analyzing, and presenting evidence from digital devices i.e. computers, mobile phones, etc. in a court of law.

Digital Investigation — A methodology to respond to suspicions about digital states and events of an electronic device relating to an incident. Digital investigation doesn't involve all the processes involved in digital forensics. And, we have

	performed a digital investigation in this research sake of simplicity.
Rooting	A process that involves escalating privileges on the Android device by exploiting an inherent vulnerability/loophole in the Android operating system to gain admin privileges; normal Android smartphones come pre-shipped with standard user-level privileges. Rooting has both upsides and downsides but the rooting process is commonly adopted by the forensic community to access underlying filesystem that can help acquire digital evidence.
Stock ROM	In simple terms, stock ROM contains the system image of an Android OS and associated apps that come installed on the phone from the manufacturer while a “custom ROM” comes from a third party that can be installed by the user.
Stock Recovery	A software that allows you to fix problems or reset your Android mobile device when there’s some error on the device.
Custom Recovery	A software that’s developed by a third party and provides more advanced features in comparison to Android stock recovery.
Chain of Custody	A document used to maintain a history of the control, transfer, and disposal of evidence to demonstrate that the same evidence was presented in a court of law that was recovered at the incident site.
Brick	To corrupt or cause malfunction in a mobile device
Hard Brick	Hard-bricked devices cannot boot or power on and may require hardware-level manipulation to bring the device back to normal which is rare.
Soft Brick	Soft-bricked device is one in which the device is powered on but stuck in a bootloop or at some other process.

NANDroid Backup	A backup is performed using custom recovery software or other specialized software that helps the investigators to perform a byte-level copy of the data on the device.
Logical Acquisition	A kind of evidence/data acquisition technique that involves extracting the storage objects such as files and directories from the filesystem of the Android using APIs provided by the device manufacturer.
Physical Acquisition	A kind of evidence/data acquisition technique that seize all the data (bit-by-bit copy) of a storage device including deleted data.
Privacy Browers	Web Browsers used to visit websites without creating a search history, protect your personal information, and prevent websites from keeping track of your browsing habits.
TOR	Stands for *"The Onion Router"*; a project developed by a non-profit organization to protect user privacy by providing anonymity over the internet. Free and open-source software that enables anonymous communication by directing user's Internet traffic through free and volunteered overlay networks comprising thousand of relays to hide user's location and internet usage from authorities performing network surveillance and analysis.
Bootloader	A software component of Android that loads up the operating system on the device.
Zram	Zram is a Linux kernel module for creating an on-the-fly compressed block device in RAM that can be used for swap or platform-independent RAM disk. The two most common uses for Zram are the storage of temporary files and as a swap [34]
LEAs	Stand for *"Law Enforcement Agencies"*; body or institution or organization responsible to conduct civil and criminal investigations. LEAs play an important role while acting as first responders in case of any incident or crime. In this research, we are dealing with a digital crime and LEAs perform the first responder steps.

ISPs	ISPs *(Internet Service Providers)* play an important role while tracing an online activity of a user suspected of performing illicit activities.
Digital Forensic Process	The digital forensic process is an established scientific procedure used in digital forensic examinations involving computers, mobile, and other digital devices. Digital Forensic Process involves Evidence Seizure, Acquisition, Analysis, Documentation, and Presentation.
Unlocking Bootloader	Unlocking the Bootloader allows us to install customized firmware and access full privileges on Android device to perform modifications to the phone that involves replacing pre-installed software or operating system
Flashing	A slang term used in the tech world that simply means installing a new firmware or operating system on a smartphone. In simple terms, flashing your phone usually means *Installing custom recovery* or installing *arbitrary files* that allow you to root your phone.
Onion Websites	Websites accessible over .onion domain that can only be accessed using Tor browser. Dark-web websites hosted over Tor network use .onion top-level domain (TLD) that is not registered in the Internet DNS root
Dark Web	A kind of World Wide Web that is accessible on networks made between trusted nodes and requires specialized software, tools, or equipment to open i.e. Tor and I2P are two commonly used examples to access the dark web [35]
Surface Web	A kind of World Wide Web that is accessed by the user in their daily life, offered to the general public using standard search engines and standard web browsers that do not need any special configuration [35]
Deep Web	A kind of World Wide Web that is not listed or searchable by conventional search engines; websites or services hosted of the deep web requires authentication, or are accessible only through the specific IP address [35]

Digital Evidence	Any information found on a computer hard drive, or mobile phone storage that is stored or communicated in a binary form and can be presented in a court of law is considered as Digital evidence. It is usually associated with electronic crime, or e-crime and may be used to prosecute all types of crimes, not just e-crime [36]
Evidence Acquisition	A process of generating a bit-by-bit duplicate of data stored on a storage device in a forensically sound manner using tools that do not affect the integrity and authenticity of the evidence
Evidence Analysis	A process that involves thorough analysis and assessment of electronically stored information (ESI), to identifying evidence that may aid or challenge questions in civil or criminal investigations
NIST	Stands for “National Institute of Standards and Technology”, a body that provides standards to assist enterprises to secure information that is sensitive but not classified

1.2 Motivation

During this research, we came across different studies that perform the forensic analysis of Tor browser application independently on Android or on Windows operating system but no one attempts to perform this study simultaneously for both operating systems. Some of these prior studies only attempts to perform limited user browsing activities to gather evidence. Previous studies tried to cover forensic analysis of Tor privacy browser on older versions of these operating systems that make applying the existing forensic techniques on the current version of Windows, Android, and Tor browser as well useless. Because these anonymity/privacy web browsing applications and others alike and all software applications have evolved over time with continuous bug fixes and improvements that possibly make it harder for investigators to extract evidence from them anymore.

As per our research, no single study has been found as yet that contains the digital investigation or say forensic analysis of Tor privacy browser on Windows and Android in combined research which also entails their latest versions. Also, previous studies failed to report on purely anonymous web browsing activities encompassing the dark web that can be simulated as a real-life cyber crime incident that can test the capabilities and skills of forensic investigators in this regard.

So, this all serves as our motivation for this study because we feel that there exists a strong necessity to address the following:

a) *To forensically analyze newer version of the Tor browser application on the latest Windows and Android build*

b) *To propose a model or framework that will help the investigators in conducting a digital investigation of anonymity web browsers more effectively*

1.3 Problem Statement

As per our research, problem statement address in subsequent sections:

a) Investigating the Tor privacy browser on suspect's system (running Windows 10) and smartphone device (running Android 10) and analyzing user's activities over the dark web to reveal the application and browsing artifacts under a simulated cybercrime scenario.

b) Identify the potential artifacts that can help the LEAs prosecute a suspect or establish a considerable degree of user attribution from the evidence retrieved.

1.4 Goals and Objectives

As per our findings from previous studies, we have established a few goals and objectives that we are going to achieve in this research are mentioned as under:

1.4.1 Goals

a) Extract every possible artifact of Tor privacy browser

b) Establish hypothesis about user's malicious activities

c) Help investigators in profiling the latest Tor browser artifacts

1.4.2 Objectives

a) Extract forensic evidence of the latest Tor Browsing application from Windows and Android

b) Extract evidence of Tor application usage and user browsing from Android 10 operating system.

c) *Extract evidence from different access level of the Android device i.e. Unrooted, Rooted Android Device and NANDroid Backup*

d) *Extract evidence of Tor application usage and user browsing from Windows 10 operating system*

e) *Extract every possible Tor application artifact that can help the investigators develop a hypothesis about online malicious activities of a user*

1.5 Scope

The scope of proposed research is currently limited to the latest Windows and Android build. On Windows 10, we gather and analyze evidence of Tor browser from Registry, Storage/Filesystem, and Memory (RAM) while on Android 10, we have gathered the Tor browser evidences from its ADB Logs, Storage/Filesystem, Zram (*first time explored for gathering evidence*) and Memory (RAM).

We currently didn't target Linux, MacOS, and iOS operating systems to keep our scope limited due to a shortage of time.

Secondly, we cover only commonly used components of subject operating systems to keep this research concise.

Third, we have not covered the data carving and deleted data recovery because it's a time and storage-intensive process.

In this research, we designed and simulated a dark web-based 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 from these evidence.

1.6 Challenges and Limitations

In this research, following challenges and limitations are faced

a) *Difficulty to preserve the data on Android device while unlocking the bootloader*

b) *Difficulty and variation in OEM unlocking the bootloader based on vendors*

c) *Custom recovery software e.g. TWRP may soft-brick the device if the installation process was not handled carefully*

d) *Rooting the device may also soft-brick the device if the flashing process was not handled carefully*

e) *The larege amount of storage is required on forensic workstations to acquire evidence of every possible activity of Tor browser on each of the subject operating systems we are targeting i.e. Windows 10 and Android 10*

f) *Acquiring the Zram (Android only) and memory (RAM) image at the right time after user browsing activity performed*

g) *To be careful in selecting the correct encoding scheme in Hex Editor application while searching for string/pattern of Tor browser artifacts*

h) *Acquiring professional forensic tools because they have expensive subscriptions*

i) *Android memory forensic tool i.e. Fridump doesn't allow us to acquire evidences from memory while Tor browser or any other application is just closed after execution*

Finally, in a nutshell, this study intends to find answers for the below-mentioned questions [1]:

- What are the possible methods/techniques by using which an investigator can find evidence of Tor privacy browser usage and browsing from the latest Windows and Android devices?

- What type of challenges can be faced by a forensic investigator?

- What type of evidence can be extracted?

1.7 Formation of Thesis Document

This thesis document is divided into six sections, with the first section comprising of detailed introduction, and Section 2 contains the work related to Tor browser forensics highlighting different studies on Windows and Android operating system and with other privacy-preserving web browsing application, Section outline the digital investigative methodology adopted for this study, Section 4 details all the experimentation performed across Windows 10 and Android 10 operating system, Section 5 cover findings and analysis that results from the examination of artifacts from both devices, Section 6 highlight comparison with existing research, Conclusion and Future Work.

Chapter 2

Related Work

In this chapter we have tried to briefly discuss different studies targeting Forensics of Tor Browser on Android operating and different Windows operating system flavours. To conclude this chapter, relevant research information has been collected from Google Scholar, IEEE Xplore, ACM Digital Library, ScienceDirect, Researchgate, SANS Digital Forensic Workshop and many other academic databases using various keywords and search patterns.

In work by N. Barghouthy et.al. [4], the study was conducted to examine Orweb (now called Tor browser for mobile) browsing sessions being performed on Samsung Galaxy S2 running Android 2.3.3. The device was examined in both rooted and unrooted states and it was concluded that browsing sessions were only on a rooted device, but old versions of Android and Tor privacy browser were examined in the study.

In [5], a similar study was conducted again by N. Barghouthy et.al. as in [3] but now Samsung Galaxy S2 was running Android 4.1.1 and it was proposed that there is no such 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. However, this method proves to be very useful from a forensic point of view but again this custom flashing recovery method is very different on the latest devices as old versions of Android and Tor privacy browser were examined in the study.

In study by C. Meda et.al. [6], 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 extract the artifacts but the authors didn't mention the tools and techniques used. Browsing history was not fully extracted in this research. Again this research was conducted on old version of Android and Tor privacy browser.

In [7] researchers S. Teng et.al examined six (06) different privacy browsers such as Epic Privacy Browser, Secure Browser, Comodo Dragon, SRWare Iron, Dooble, and Maxthon along with Tor privacy browser on Windows OS. Evidence were 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 study by M. Asim et.al. [8], 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. Again, as in previous researches, older version of Android, Android emulators, and web Browsers were targeted in this research. AndroKit was used for Tor browser forensic as it is based on Mozilla firefox web browser.

In another study by A. Warren [9], the researcher performed a forensic analysis of Tor Browser version 5.0 was performed 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 Tor browser leaves minimal on-disk evidence.

Furthermore, in work by A. Jadoon et.al. [10], the authors performed a forensic analysis of Tor privacy browser 7.02 (32-bit) on Windows 8.1 OS in which they analyze Tor browser artifacts from registry, memory, and storage but they cover 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.

In work by Rebecca N and et al. [11] forensic artifacts were recovered from normal and private browsing modes of two famous browsers i.e. Google Chrome and Mozilla Firefox and their private browsing results were compared with famous anonymous browser TOR v7.0.5 on Windows 7 (64-bit) using AccessData FTK as a primary tool. Their research predominantly uncovers artifacts from the storage of experimental VMs with the conclusion that Tor browser reveals less user browsing artifacts when compared to private browsing modes of Chrome and Firefox.

In work by Gandeva B Satrya et al. [12] a novel android internal memory forensic acquisition tool called fridump was proposed 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 victim and investigator’s android smartphones ie. Samsung A7 and Oppo A37F but still there are some limitations in the tool that it works only with running processes which need to be addressed.

Similarly, other works by Muir et.al. [13], Horsman et.al [14], Satvat [15], Alfosail et.al. [16] proposed a framework to recover artifacts of Tor privacy browser from memory, but their work only cover Windows 10 build 10586 memory to reveal user-related information. They have not attempted to recover other usage or browsing related artifacts from Windows storage, logs and registry.

In comparison to all the previous research works performed on Tor privacy browser, our work mainly targets the *latest version of Windows and Android OS with latest version of Tor privacy browser*. This study aims to identify every potential area in Windows and Android devices where a forensic investigator can look for evidence related to Tor privacy browser. This study 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 a court of law.

A vast literature review about the forensic analysis of Tor privacy browser is accomplished to identify the objectives of this research. Also, gap analysis is carried out with previous research to further explain the objectives from a forensic investigator's point of view. [1]

Chapter 3

Digital Investigation Methodology

The main objective of this research was to collect potential evidentiary artifacts linked to the usage of Tor privacy browser on both Windows 10 and Android 10 operating system devices.

To accomplish this research, we mimicked a cybercrime scenario that involves:

a. Browsing distinct cyber-crime relevant websites on Tor privacy browser installed in each of the operating systems
b. Acquiring and analyzing the evidence(s) for traces of user browsing and application usage remnants

3.1 Simulated Investigative Scenario

To accomplish an appropriate, detailed, and result-oriented digital investigation of the Tor privacy browser, we decided to simulate a realistic cyber-crime scenario whose particulars are mentioned below:

"A suspect was arrested by Law Enforcement Agencies (LEAs) based on information received from Intelligence Agencies.

*Allegations against the suspect was **"He's breaching confidential information/data about high-profile Government and corporate employees to foreign intelligence agencies using covert communication channels".***

At the time of the suspect's apprehension, LEAs discovered the following devices in his custody:

- *Laptop*
- *Smartphone*

***First Responder staff** of **Digital Incident Response and Investigation team** conducted the preliminary physical inspection on seized devices and concluded the following information:*

- *Laptop device was running the latest Windows 10 operating system*
- *The smartphone device was running the Android 10 operating system*

Seized devices were attached to a specialized power source(s) to keep them running and then wrapped in an anti-static faraday bag(s) to prevent any electromagnetic tampering to digital evidence stored in them. Chain of custody form(s) was filled and *signed by First Responders, and devices were handed over to the **Forensic Investigation lab** for further investigation.*

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

- *Usage and existence of Tor (Privacy browser) used for exfiltrating confidential information*
- *Any email/website the suspect has visited from where evidence of his activity can be found*
- *Any credentials used for suspicious communication*
- *Any other clues related to the suspect's activity*
- *Any other files of interest"*

In the above-simulated scenario, we try to cover every possible activity related to the Tor privacy browser from a suspect point of view (either it involves browsing or non-browsing execution) [1].

Furthermore, to adhere to the scenario-specific activities, we also covered installation and un-installation activities to discover the presence of Tor as being a Forensic Investigator, we must make sure that sufficient evidence information should be provided to the LEAs about usage and presence of Tor.

Several kinds of scenario-specific websites were visited in this simulation of realistic cybercrime incidents that include both **dark-web** (.onion) and **surface web** (normal) websites whose details are as follows:

3.1.1 Dark-web websites (.onion) [1]

- "Hidden Wiki" is quite a famous website in the dark web realm.
- "Darknet search engines" to search for .onion websites
  - Ahmia
  - DuckDuckGo
  - Excavator
- "Secmail" which is a Tor based secure email service
- "Galaxy3" which is a Tor based social networking platform
- "StealthPay" which is an anonymous money transfer platform
- "Keybase" which is secure communication applications provider website
- "Anonymous text sharing" websites
  - ZeroBin
  - StrongHold Paste

- "SecureDrop" which is an anonymous file-sharing website

3.1.2 Surface web (Normal) websites [1]

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

The details of all the browsing activities that we have covered in our sample investigative scenario along with respective credentials used are listed in Table 3.1 for Windows 10 operating system and in Table 3.2 for Android 10 operating system.

Website Cat.	Sr.	Browsing Information		Browsing Activities Performed
Wiki	1	**URL Title**	Hidden Wiki	1. Browsed 2. **Whistleblowing** hyperlink clicked
		Website/ URL visited	zqktlwiuavvvqqt4ybvgvi7tyo4hjl5xgfuvpdf6otjiycgwqbym2qad.onion/wiki/index.php/Main_Page	
		Credentials Used	Not applicable	
Search Engines	2	**URL Title**	Ahmia	1. Browsed 2. Search query **"sell official data"** performed 3. Clicked the first result & get redirected to **5j7saze5byfqccf3.onion/data/bullseye/main/** URL 4. Download **components-mips64el.yml.gz** file from the above URL
		Website/ URL visited	msydqstlz2kzerdg.onion	
		Credentials Used	Not applicable	
	3	**URL Title**	Duck-DuckGo	1. Browsed 2. Search query **"sell official data"** performed
		Website/ URL visited	3g2upl4pq6kufc4m.onion	
		Credentials Used	Not applicable	
Cloud Storage/ Sharing	4	**URL Title**	Google Drive	1. Browsed after login to ***Gmail*** using Google credentials at ***Sr. 10*** 2. Uploaded text file **"~res-x64-1.txt"**
		Website/ URL visited	drive.google.com	
		Credentials Used	torforensics@gmail.com	
	5	**URL Title**	MEGA	1. Browsed and then Login 2. Right click already uploaded **PDF file** and get Sharing link **https://mega.nz/file/zz40xB6S#isXGprskZbLP4KnLNuNHcbI279s6FnLcsj8Vydm_sio** 3. Copied the link to clipboard
		Website/ URL visited	mega.nz/login mega.nz/fm	
		Credentials Used	torforensics@gmail.com	
	6	**URL Title**	ZeroBin	1. Browsed 2. Pasted the **Mega Sharing Link** from Clipboard copied
		Website/ URL visited	zerobinqmdqd236y.onion	

		Credentials Used	Not applicable	in ***Sr. 5*** 3. Generate the Paste link 4. Link to the Paste **http://zerobinqmdqd236y.onion/?be163e348777b667#H7xg5DfMboatOgot8q439QNYTogRfXLAP74fmqzeXjI=** copied to clipboard
Money Transfer	7	**URL Title**	Stealth-Pay	Browsed only
		Website/ URL visited	https://www.stealthpay.com/requestmoney	
		Credentials Used	Not applicable	
Secure Commu-nications	8	**URL Title**	Keybase	1. Browsed 2. Tried downloading software from **/download** 3. Just visited the URL **.../docs/the_app/install_windows**
		Website/ URL visited	fncuwbiisyh6ak3i.onion	
		Credentials Used	Not applicable	
	9	**URL Title**	SecMail	1. Browsed 2. Login 3. Open first email in the Inbox which is from **torforensics@gmail.com** for reading 4. Replied the email with contents as shown below: *Email To:* ***"torforensics@gmail.com"*** *Email Subject: "**Re: Impt Data**"* *Email Body:* ***"https://mega.nz/file/zz40xB6S#isXGprskZbLP4KnLNuNHcbI279s6FnLcsj8Vydm_sio"***
		Website/ URL visited	secmail63sex4dfw6h2nsrbmfz2z6alwxe4e3adtkpd4pcvkhht4jdad.onion	
		Credentials Used	adamjames555@secmail.pro	
Emails	10	**URL Title**	Gmail	1. Browsed 2. Login 3. Checked the email
		Website/ URL visited	mail.google.com	
		Credentials Used	torforensics@gmail.com	
	11	**URL Title**	Outlook	1. Browsed 2. Login 3. Composed and sent the email with contents as shown below:
		Website/ URL visited	outlook.live.com	
		Credentials Used	torforensics@outlook.com	

				Email To: ***torforensics@gmail.com*** *Email Subject: "**Imp Stuff**"* *Email Body: "**Send money at my wallet**"*
Voice/ Video Chat	12	**URL Title**	Skype	1. Browsed 2. Login 3. URL **web.skype.com** was opened but received **"browser not supported"** message
		Website/ URL visited	Web.skype.com Secure.skype.com www.skype.com	
		Credentials Used	torforensics@outlook.com	
Social	13	**URL Title**	Galaxy3	1. Browsed 2. Login 3. Composed the **Wire Blog** post with content as shown below: ***http://zerobinqmdqd236y.onion/?be163e348777b667#H7xg5DfMboatOgot8q439QNYTogRfXLAP74fmqzeXjI=***
		Website/ URL visited	galaxy3bhpzxecbywoa2j4tg43muepnhfalars4cce3fcx46qlc6t3id.onion	
		Credentials Used	adamjames555@tutanota.com	

Table 3.1: Browsing activities performed on Windows 10 virtual machine

Website Cat.	**Sr.**	**Browsing Information**		**Browsing Activities Performed**
Wiki	1	**URL Title**	Hidden Wiki	1. Browsed 2. **Whistleblowing** hyperlink clicked
		Website/ URL visited	zqktlwiuavvvqqt4ybvgvi7tyo4hjl5xgfuvpdf6otjiycgwqbym2qad.onion/wiki/index.php/Main_Page	
		Credentials Used	Not applicable	
Search Engines	2	**URL Title**	Ahmia	1. Browsed 2. Search query **"sell official data"** performed 3. Clicked the first result & get redirected to **5j7saze5byfqccf3.onion/data/experimental/main/** URL 4. Download **components-arm64.yml.xz** file from above URL
		Website/ URL visited	msydqstlz2kzerdg.onion	
		Credentials Used	Not applicable	
	3	**URL Title**	DuckDuckGo	1. Browsed 2. Search query **"sell official data"** performed
		Website/ URL visited	3g2upl4pq6kufc4m.onion	
		Credentials Used	Not applicable	
	4	**URL Title**	Excavator	1. Browsed 2. Search query **"sell official data"** performed
		Website/ URL visited	2fd6cemt4gmccflhm6imvdfvli3nf7zn6rfrwpsy7uhxrgbypvwf5fad.onion	
		Credentials Used	Not applicable	
Cloud Storage/ Sharing	5	**URL Title**	Google Drive	1. Browsed only after login to ***Gmail*** using Google credentials at ***Sr. 13***
		Website/ URL visited	drive.google.com	
		Credentials Used	torforensic@gmail.com	
	6	**URL Title**	MEGA	1. Browsed and then Login

		Website/ URL visited	mega.nz/login mega.nz/fm	2. Uploaded the file **IMG-20210122-WA0005.jpg** from device 3. Retrieved the sharing link of uploaded file in ***Pt. 2*** 4. Copied the link to clipboard
		Credentials Used	torforensic@gmail.com	
	7	**URL Title**	ZeroBin	1. Browsed 2. Pasted the **Mega.nz** file sharing link copied to clipboard at ***Sr. 7*** 3. Generated the Paste link containing content **"/?a3e1481092fb04b9"**
		Website/ URL visited	zerobinqmdqd236y.onion	
		Credentials Used	Not applicable	
	8	**URL Title**	StrongHold Paste	1. Browsed 2. Composed the Paste with content as shown below: *Paste title: "**Pix**"* *Paste data:* ***https://goo.gl/xZgh1qu*** 3. Password-protected the Paste 4. Generated the Paste link containing content **"/pocsxm1d5/2uo2vh"**
		Website/ URL visited	nzxj65x32vh2fkhk.onion	
		Credentials Used	Not applicable	
	9	**URL Title**	SecureDrop	1. Browsed 2. Clicked **"Get started"** hyperlink and received codename **"unloving cornflake ecosphere decipher trifocals scotch reiterate"** on next page 3. Clicked **"Submit documents"** on page 4. Uploaded **IMG-20210122-WA0005.jpg** to webserver
		Website/ URL visited	arujlhu2zjjhc3bw.onion arujlhu2zjjhc3bw.onion/lookup	
		Credentials Used	Not applicable	
Money Transfer	10	**URL Title**	Stealth-Pay	Browsed only
		Website/ URL visited	https://www.stealthpay.com/requestmoney	

		Credentials Used	Not applicable	
Secure Commu-nications	11	**URL Title**	Keybase	1. Browsed 2. Clicked **"Send secure message" hyperlink and get** redirected to **"play.google.com"** for Keybase Android APK installation page.
		Website/ URL visited	fncuwbiisyh6ak3i.onion	
		Credentials Used	Not applicable	
Emails	12	**URL Title**	SecMail	1. Browsed 2. Login 3. Checked emails received from Gmail and Outlook email addresses at ***Sr. 13 & 14*** 4. Email from Gmail account was replied with content as shown below: *Email To:* ***torforensics@gmail.com*** *Email Subject:* ***"Re: Impt Data"*** *Email Body:* ***"Find here: https://goo.gl/xZgh1qu"***
		Website/ URL visited	secmail63sex4dfw6h2nsrbmfz2z6alwxe4e3adtkpd4pcvkhht4jdad.onion	
		Credentials Used	adamjames555@secmail.pro	
	13	**URL Title**	Gmail	1. Browsed 2. Login 3. Email was composed and sent with content as shown below: *Email To:* ***adamjames555@secmail.pro*** *Email Subject:* ***"Impt Data"*** *Email Body:* ***"Please share link to receive data"***
		Website/ URL visited	mail.google.com	
		Credentials Used	torforensic@gmail.com	
	14	**URL Title**	Outlook	1. Browsed 2. Login 3. Email was composed and sent
		Website/ URL visited	outlook.live.com	

		Credentials Used	torforensic@outlook.com	with content as shown below: *Email To:* ***adamjames555@secmail.pro*** *Email Subject:* ***"Imp Data"*** *Email Body:* ***"Please share link to receive data"***
Voice/ Video Chat	15	**URL Title**	Skype	1. Browsed 2. Login 3. Visited Account overview page 4. URL **web.skype.com** was opened but received **"browser not supported"** message
		Website/ URL visited	Web.skype.com Secure.skype.com www.skype.com	
		Credentials Used	torforensic@outlook.com	
Social Media	16	**URL Title**	Galaxy3	1. Browsed 2. Login 3. **/Settings** link visited 4. Blogs link **"/blog/owner/aj555"** was visited
		Website/ URL visited	galaxy3bhpzxecbywoa2j4tg43muepnhfalars4cce3fcx46qlc6t3id.onion	
		Credentials Used	adamjames555@tutanota.com	
Torrents	17	**URL Title**	The Pirate Bay	1. Browsed 2. Search query **"privacy"** was performed with Application check box marked on webpage 3. From the result, **Privacy Shield** URL was opened 4. Torrent magnet link was copied to clipboard with content as shown below: **"magnet:?xt=urn:btih:2A3B…"**
		Website/ URL visited	https://thepiratebay.cx/en1/	
		Credentials Used	Not applicable	

Table 3.2: Browsing activities performed on Android 10 device

3.2 Targeted application usage and browsing activities

In this research, we are targeting some of the common application usage activities because any application whether it is a web application or a desktop program or a mobile application has some common lifecycle activities even if it is in a development stage or a user-experience stage.

In this research, we explore four (04) of our application's usage lifecycle activities that a suspect/user will certainly follow as per our simulated scenario. On each of our targeted operating system(s), we acquire evidence for each of these activities that we pick out for *Tor privacy browser* and then we analyze for artifacts. These activities are defined below [1]:

1. **Application Installation**
   - Tor browser has just been installed but not executed for once.
2. **Application Execution (No browsing)**
   - Tor browser is executed. Tor circuit is bootstrapped, and the browser is connected to the Tor network, but no browsing activity is performed.
3. **Application Execution & Closure (with Browsing)**
   - Browsing activities mentioned in Table 1 and 2 were performed and evidence were acquired in these two stages:
     - **i.** The browser remained open and evidence was acquired
     - **ii.** The browser was closed and immediately evidence was acquired
4. **Application Uninstallation**
   - Tor browser was simply uninstalled from the system/device.

3.3 Targeted Operating System Components

3.3.1 *Windows 10*

Three different components of Windows 10 operating system were explored in this research i.e.

- Registry
- Memory
- Storage

For ***storage*** acquisition and analysis on this system, we do not cover uninstallation activity deliberately because this application activity simply involves deleting an application folder which is not worthwhile [37].

3.3.2 Android 10

For Android operating system, our experimentation also studies three different access level/states of an Android device i.e

- Un-rooted Android device (without administrative privileges)
- Rooted Android device (with administrative privileges) [20]
- NANDroid Backup (with Custom Recovery software installed making a perfect mirror image of the device) [21].

Considering the above access level/states, four different components were explored for each state in search for artifacts i.e.

- Storage
- Zram
- Memory (RAM)
- ADB (Android Device Bridge) Logs. ADB which is a command-line tool allows us to communicate with the device [17] and fetch Android device logs using two important tools:
    1) *logcat* [18] that output logs of system messages
    2) *Dumpsys* [19] that output information about system services

For ***memory acquisition and analysis*** on Android 10 operating system, we only cover application execution activity (either with or without browsing) due to the reason which we will discuss in ***section 4.2.2.2(3).***

Based on all the above-mentioned activities and components being targeted in this research, we devised the organized OS-specific digital investigation methodology for Tor privacy browser. The methodology we opt for Windows 10 and Android 10 operating systems in this research is based on NIST Special Publication 800-86 document [24] *"Guide to Integrating Forensic Techniques into Incident Response"* which can be extended to include other versions of Windows and Android operating system(s). The flowcharts of our methodologies are shown below in Figures 3.1 & 3.2 [1]

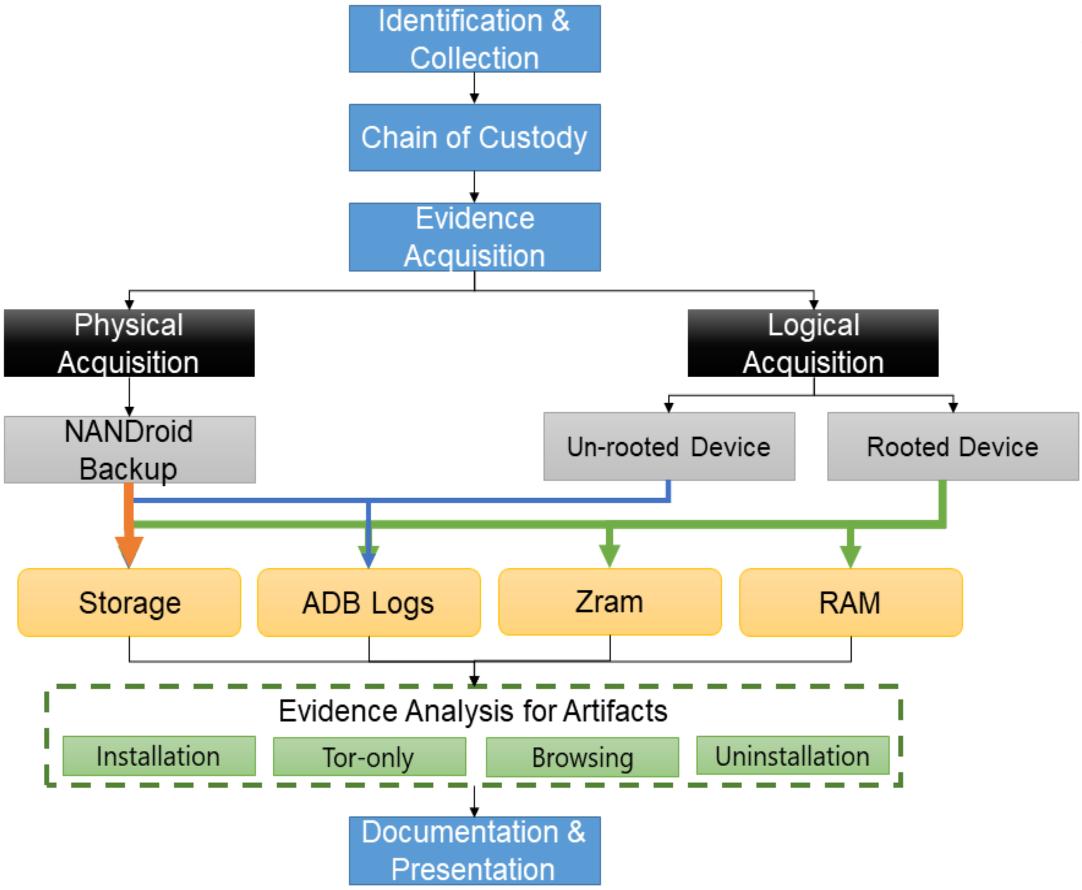


Figure 3.1: Investigation Methodology on Windows 10

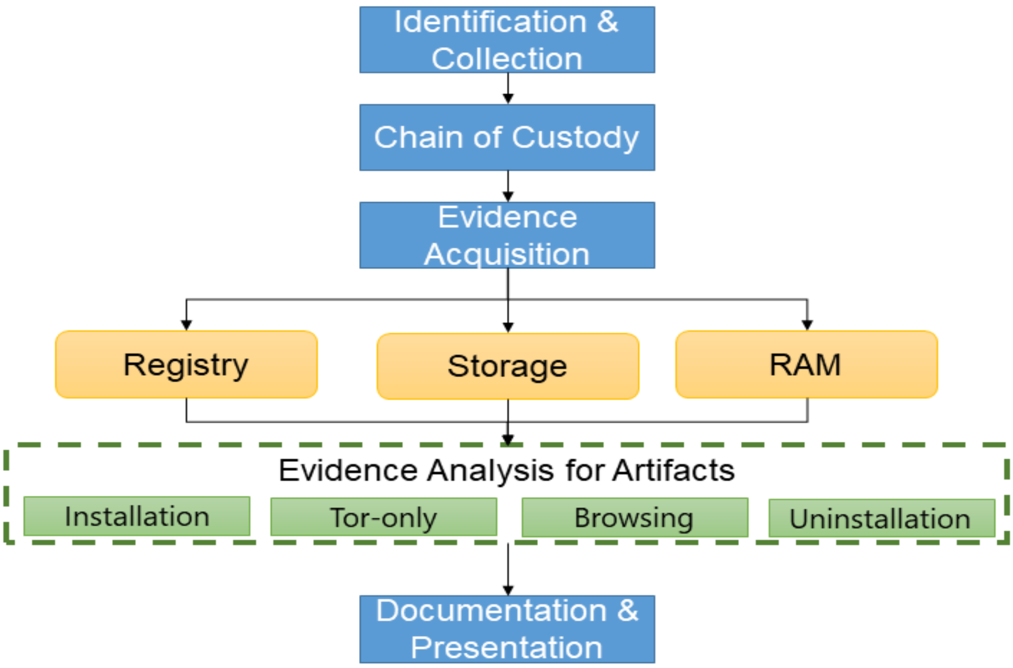


Figure 3.2: Investigation Methodology on Android 10

Chapter 4

Experimentation

Experimentations that were performed in this research are designed in such a way that they can serve as a proof-of-concept (PoC) and are based upon the following two major points:

1) Simulated dark web-based cyber-crime scenario
2) Digital investigation methodology designed for our target operating systems

The investigative methodologies that we devised in Section 3 are going to be implemented in two experimental test environments; one for each operating system that we are targeting i.e Windows 10 and Android 10 operating system.

After completion of our evidence acquisition activities, operating system installation(s)/ their storage were reverted to a fresh/clean state for two purposes:

1) To achieve residual data sanitization
2) To perform the next session of simulating our browsing scenario and following our proposed digital investigative methodology that we mentioned above

Multiple sessions of experimentation were performed to achieve multiple iterative results:

1) 8 x experimental sessions for Windows 10 machine
    a. Involving browsing and evidence acquisition
2) 21 x sessions for Android 10 device(s)
    a. 5 x sessions on un-rooted device
    b. 16 x sessions on rooted device
    c. 2 x sessions for NANDroid backup
    d. Point a & b involve browsing and evidence acquisition and point c involves evidence acquisition after rebooting the device only moments after recent user browsing.

4.1 Experimental Setup

4.1.1 Windows 10

To work in a clean and neat environment for experimentation, the virtual environment is used, and a fresh Windows 10 virtual machine is created for this purpose to acquire and analyze the evidence(s) from the following system components that we already discussed in section 3 of this research:

1) Registry
2) Memory (RAM)
3) Storage (Filesystem i.e. NTFS)

A list of software tools that are used during this part of the digital investigation is listed as:

- **Operating system**
    - Window 10 Pro 64-bit (Version 20H2 Build 19042.746)
- **Software Tools**
    - VMware® Workstation 14 Pro (Version 14.0.0 Build 6661328)
    - 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)

After creating a fresh Windows 10 virtual machine on VMWare workstation, a first snapshot was captured to return the system for two purposes:

1) Return the system to a completely clean state before performing the next session of experimentation from scratch
2) Any other issue/error encountered during system execution

4.1.2 Android 10

Three clean and factory reset Android 10 devices were utilized for this experimentation to acquire and analyze the evidence(s) from the following system components that we already discussed in section 3 of this research:

1) Storage (Filesystem – user data partiti–n - F2FS) [38]
2) ADB Logs (Dumpsys and Logcat logs)
3) Zram partition (acts as virtual memory in Android devices)
4) Memory (RAM) artifacts.

But not all components are being explored during the evidence acquisition phase due to three common reasons in the case of Android devices:

1) Device access/level state
2) Privilege level
3) Technical limitation

These reasons were briefly discussed in the subsequent section.

A list of software tools that are used during this part of the digital investigation is listed as:

- **Android Smartphone devices with Operating system Build**
  - 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)
- **Software Tools**
  - 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) [22]
  - Frida Server for android-arm64 (Version 14.2.11) [23]
  - Fridump – A novel open source Android memory dumping tool (Version 0.1) [39,40]

4.1.3 Evidence Analysis Tools

In addition to above-mentioned software tools used for both concerned operating systems, some other useful software tools that are being utilized for analysis of the evidence on the Forensic Workstation.

These software tools used for analysis are listed below:

- 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)

4.2 Evidence Acquisition

4.2.1 Windows 10

Three different types of evidence acquisitions were performed on Windows 10 operating system[1]:

- Registry
- Storage
- Memory

4.2.1.1 Brief Evidence Acquisition Methodology With Tools Used

Registry snapshots are acquired using Regshot tool before & after below-mentioned activities[1]:

a. Installation
b. Execution (with or without browsing)
c. Post-Execution
d. Uninstallation

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.

4.2.2 Android 10

In the case of Android device, we performed different acquisitions according to the state of the device we encounter during our digital investigation scenario:

4.2.2.1 Android device state(s) with a particular type of acquisition

- **Un-rooted Android device**
  - Storage
  - ADB Logs
- **NANDroid Backup**
  - Storage
- **Rooted Android device**
  - Storage
  - ADB Logs
  - Zram
  - Memory

4.2.2.2 Brief evidence acquisition methodology with tools used

1. First, we try to acquire as much as possible evidence from an unrooted android device after installation, browsing, and uninstallation **because** we do not have a lot of access, so 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.

2. Next, we unlocked the bootloader of our targeted Android device [25] using ABD platform tools in Fastboot mode to install a custom recovery software. i.e. TWRP [26] to acquire NANDroid backup of 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 "Browser Closed" state because NANDroid backup requires rebooting the device into recovery mode.
3. Finally, we root our device using Magisk [27] 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 3.5 using MOBILedit Forensic Express but we were only able to acquire memory evidence using the most efficient Android memory forensic tool developed by Satrya, G. B et.al [12] for Simple Execution and Browsing activity because Fridump tool only let us acquire memory evidence while the process is running.

Warning: *Acquisition methodologies that are mentioned at Sr. No. 2 and 3 were only performed for experimentation in this research. The use of these methodologies in a real case scenario without any "written authorization, expert supervision and pre-cautions" will be dangerous and will destroy the seized evidence. These evidence acquisition techniques are only recommended if the device already has an unlocked bootloader or is rooted which may vary according to the case scenario. All these acquisition methodologies approach for device "Xiaomi Mi A3" are explained in the section "Gaining Access on Android Device" because the methodology performed to gain access on other two devices are similar.*

4.2.2.3 Gaining Access on Android Device

a. Unlocking the Bootloader [25]

For the sake of experimentation, we first unlocked the bootloader of our targeted Android device. The pre-requisite for this process requires the following tools and settings on the targeted device and investigator workstation:

Settings required on Android Device

- *Developer options* should be enabled
- *USB Debugging* should be enabled
- *OEM Unlocking* should be turned on (This may vary from device to device)

Tools setup required on investigator workstation

- Universal Android USB drivers should be installed
- Platform Tools should be installed, and *PATH* environment variable should be set to the path of investigators liking
- Command Prompt should be open

Procedure

After setting up the device and necessary tools on the investigator workstation, the bootloader was unlocked using the following method:

i. First, Android device should be connected to the workstation using USB cable (Type B or Type C)
ii. Make sure that the device is recognizable and detected by the operating system and necessary drivers are installed
iii. Using command prompt, boot the device into a fastboot mode using the following command(s):

    - `adb devices`
    - `adb reboot bootloader`

iv. After the appearance of the bunny logo on device, make sure that device is successfully booted into the fastboot mode by executing the following command:

    - `fastboot devices`

v. Now as the device is successfully booted into fastboot mode, now unlock the bootloader using the following command:

    - `fastboot flashing unlock`

vi. After the last command, your device will restart and boot automatically into fastboot mode. Now execute the final fastboot mode command to completely unlock the bootloader on your device:

    - `fastboot flashing unlock_critical` [41]

vii. After the execution the last command, your data will be completely erased and your device will restart as fresh, and “Unlocked” starts appearing written on the boot screen of your Android device as shown in Figure 4.1 [42]

Figure 4.1: Android 10 powered Mi A3 with unlocked bootloader

viii. Please make sure that you have performed the complete backup before performing the step (v) and (vi)

b. Installing TWRP

This step is only required if you want to take the NANDroid Backup of your device or want to root your device using a custom recovery software instead of Android's stock recovery software; otherwise, this step is optional in investigations because the device can still be rooted using other methods.

We chose TWRP because of its wide usage and support for different Android devices. Other than TWRP, there exists plenty of other custom recovery software available online.

Pre-requisites/Settings required on Android device

- *Developer options* should be enabled
- *USB Debugging* should be enabled
- *Bootloader* should be unlocked

Tools setup required on investigator workstation

- Universal Android USB drivers should be installed
- Platform Tools should be installed, and *PATH* environment variable should be set to path of investigators liking
- Command Prompt should be open

Files required

- TWRP Recovery Image
- TWRP Zip Installer file

Procedure

After setting up the pre-requisites and necessary tools, TWRP custom recovery was installed using the following method:

i. First, Android device should be connected to the workstation using USB cable (Type B or Type C)
ii. Make sure that device is recognizable and detected by the operating system and necessary drivers are installed
iii. Move the TWRP recovery image file to the location where platform binaries exist e.g. C:\platform-tools\ and TWRP Zip installer file to device's internal storage using USB device
iv. Now using command prompt, boot the device into a fastboot mode using following command(s):

    - `adb devices`
    - `adb reboot bootloader`

v. After the appearance of bunny logo on device, make sure that device is successfully booted into the fastboot mode by executing the following command:

    - `fastboot devices`

vi. After performing the above step, find out the active slot of your Android device. Present day Android OS on devices comes with two partitions (A/B) [43] so it is highly recommended to find them before executing the flashing activity because it may soft-brick your device so be careful. Find and change the active slot by executing the following command in the command prompt:

    - `fastboot getvar current-slot`
    - `fastbo– --set-active=a` (if active slot is B)

vii. Then flash the TWRP custom recovery image onto your device using the following command:

    - `fastboot flash boot twrp-recovery-image.img`

viii. After this reboot your device using the following command in fastboot while holding the Volume Up button of your device:

    - `fastboot reboot`

ix. After the above step, TWRP Flash Screen will appear [44], and you will be prompted for PIN/Password you have previously set on your device. Enter the password and you will be presented with TWRP menu as show in Figure 4.2:

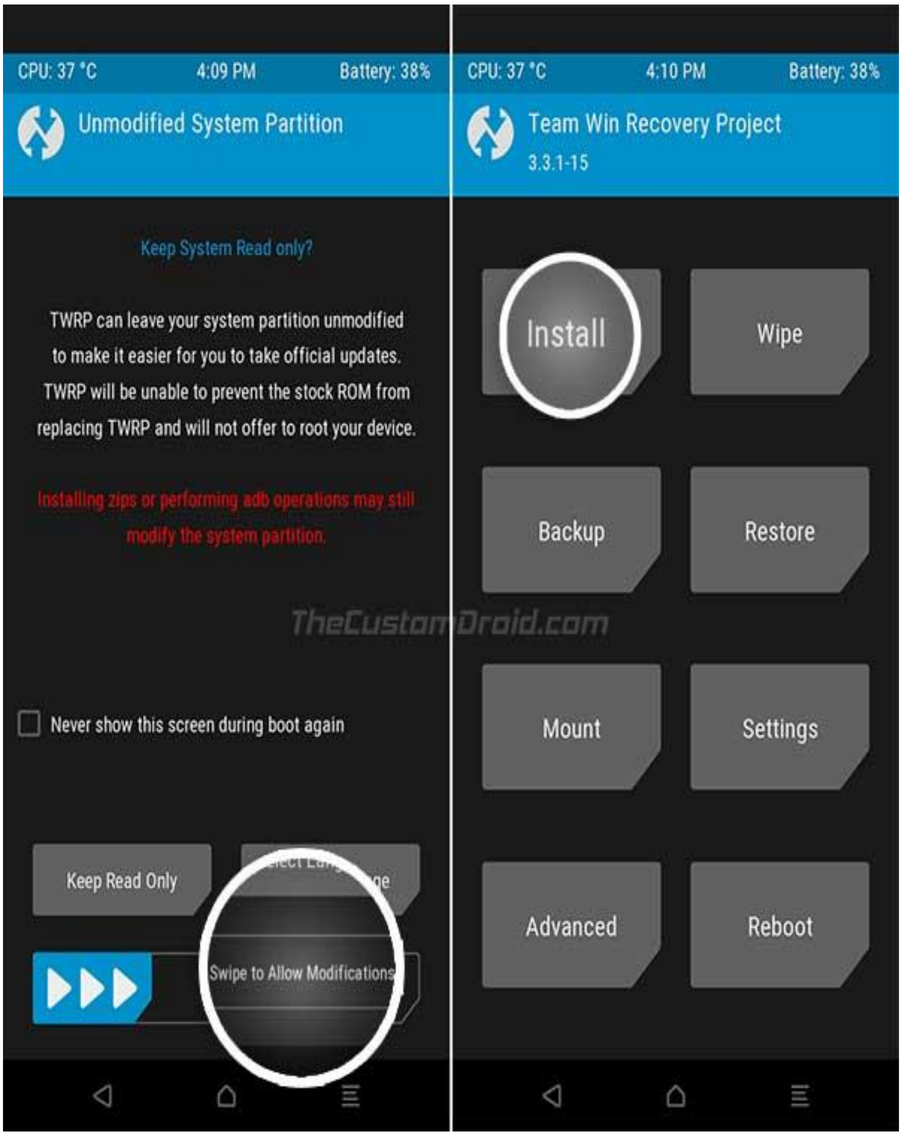


Figure 4.2: TWRP Flash Screen after installation

x. Click *"Install"* and select the TWRP Installer zip file you already copied to your device's internal storage and swipe the *"Swipe to confirm Flash"* to start flashing the TWRP installer file as shown in Figure 4.3 [44].

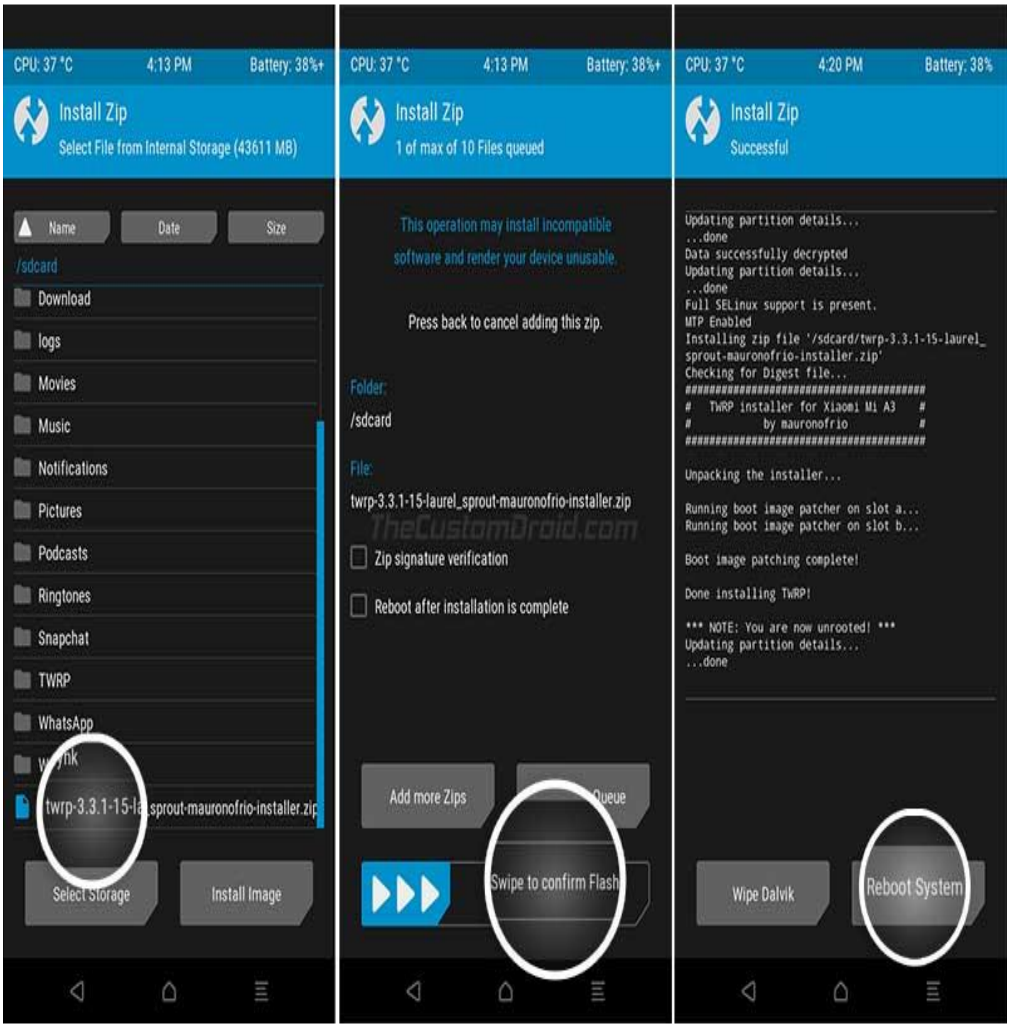


Figure 4.3: Post TWRP flashing process

xi. Click *"Reboot"* in TWRP menu and then select the Active Slot that was previously active e.g. Slot A.

c. Taking NANDroid Backup

Procedure

i. Press and hold the Volume Down and Power button simultaneously to enter the TWRP mode or click "Recovery" in TWRP menu to boot the device again into TWRP mode if you are already at above-mentioned step (xi) of Section "b"

ii. On the main TWRP menu, select "Backup" option and then select below mentioned options on next screen and swipe to confirm to start taking your NANDroid Backup [45]:

   a. Boot
   b. System
   c. Data

iii. NANDroid backup will be stored on your phone's internal storage. Copy/Move that backup file to the investigator's workstation using USB cable and ensure the safe custody of the acquired evidence and maintain chain of custody form as per digital investigation methodology devised in this research.

d. Rooting the Android device

This is an important step to acquire unlimited access to the device's underlying storage to the investigator and is required for accessing the RAM, performing physical acquisition of your device.

Three different methods can be used to root the device [27]:

a) Using TWRP or other custom recovery software
b) Using patched boot image of stock OS using rooting software/app
c) Using specialized software(s) to automatically root the device using PC

We have used methods (a) and (b) to root the Android device in this research.

Pre-requisites/Settings required on Android device

- *Developer options* should be enabled
- *USB Debugging* should be enabled
- *Bootloader* should be unlocked

Tools setup required on investigator workstation

- Universal Android USB drivers should be installed
- Platform Tools should be installed, and *PATH* environment variable should be set to path of investigators liking
- Command Prompt should be open

Apps/Software required on Android device

- TWRP Custom recovery (required for rooting method (a))
- Magisk Manager APK from Official Magisk project website (required for rooting method (a) and (b))
- Automated rooting tools available on Internet for rooting method (c)

Files required

- Boot image file from Stock ROM installed on the Android device
- Magisk Zip Installer file

Procedure

<u>Method (a)</u>

i. Copy the Magisk Installer zip file to phone's internal storage using USB cable

ii. Download the Magisk Manager APK from Magisk website to the device and install it
iii. Power off the device
iv. Press and hold the **VOLUME UP** and **POWER** button to boot the device into TWRP recovery mode
v. After the above step, TWRP Flash Screen will appear, and you will be prompted for PIN/Password. Enter the password and you will be presented with TWRP menu.
vi. Click *"Install"* and select the Magisk Installer zip file on your device's internal storage" and swipe the *"Swipe to confirm Flash"* to start flashing the Magisk
vii. Select *"Reboot"* once flashing process completes.
viii. Open Magisk manager app once it boots and verify the "Magisk in installed" on main screen to ensure that the device is rooted successfully.

Method (b)

i. Download the Stock Fastboot ROM of the Android version installed on the device
ii. Extract the **boot.img** file from ROM archive and copy it to your device's internal storage using USB cable
iii. Download and install the Magisk Manager APK from Magisk website on device
iv. Open Magisk Manager app on your phone and click the **"INSTALL"** button in front of **"Magisk is not installed"** written in **RED**.
v. Tap **"Select and Patch a File"** from the menu
vi. Select **boot.img** file copied to your device's internal storage and Magisk will start patching the boot image and when it completes; then save the patched **boot.img** file on your device's internal storage.
vii. Copy the patched **boot.img** file to your PC using USB cable or using **"adb pull"** command from ADB platform tools using command prompt:
viii. Power off the device
ix. Press and hold the **VOLUME DOWN** and **POWER** button to boot the device into Fastboot mode
x. Flash the patched **boot.img** file on your device using following command:

➢ `fastboot flash boot magisk_patched.img`

xi. After flashing process complete, reboot the device using following command:

➢ `fastboot continue`

xii. After reboot, open Magisk Manager app and verify the **"Magisk in installed"** on main screen to ensure that device is rooted successfully.

Method (c)

i. This method is automatic and can be accomplished using any professional forensic software(s) e.g. Belkasoft Evidence Center etc. and other rooting software(s) e.g. KingoRoot etc
ii. Just connect the device to forensic workstation using USB cable and open the software and accomplish rooting using interactive interface of the software
iii. This method is not 100% guaranteed depending upon the availability and validity of the exploit available to root the targeted Android device.

e. Android Memory Acquisition Setup

This is the most important setup which is required for acquiring memory (RAM) of our Android device using specialized software i.e. **fridump** developed by G. Satrya and F. Kurniawan [12]

Pre-requisites/Settings required on Android device

- *Developer options* should be enabled
- *USB Debugging* should be enabled
- *Bootloader* should be unlocked
- Device should be *rooted*

Tools setup on investigator workstation

- Universal Android USB drivers should be installed
- Platform Tools should be installed, and *PATH* environment variable should be set to path of investigators liking
- Command Prompt should be open
- *Python runtime environment* should be installed

Files required on Android device

- Frida-server binary (this depends upon your android device architecture eg. Arm64 etc.)

Files required on the workstation

- Fridump python script (fridump.py)
- Frida and Frida-tools should be installed (either in Linux based VM preferably Ubuntu or Windows-based Cygwin runtime environment)

Procedure

i. Connect your device to workstation using USB cable
ii. Launch the command prompt and write the following commands to upload and execute frida-server binary on your android device.

- ➢ `adb devices`
- ➢ `adb root`
- ➢ `adb pu39etriida-server /data/local/tmp/`
- ➢ `adb shell "chmod 755 /data/local/t39etriida-server"`
- ➢ `adb shell "/data/local/t39etriida-server &"`
- ➢ On Linux based VM, run the following command in terminal to verify that Frida-server is running properly and enumerate processes running on your Android devic39etri`ida-ps -U`

iii. After above, copy the PID of Tor browser's application to acquire memory (RAM) used by the selected process i.e. **org.torproject.torbrowser**

iv. Open python runtime environment on Windows or simply launch the *Fridump* python tool in Linux and use the following command to acquire the Tor browser's memory:

- ➢ `python fridump.py -U -v -s org.torproject.torbrowser`

v. *Fridump* tool will prompt you for Memory Dump file name. Enter the descriptive filename including date and time (highly recommended). RAM Dump will be stored in the current working directory from where fridump script will be executed.

vi. Copy the RAM dump file to a secure location and maintain the chain of custody form.

In the research, we have performed the evidence acquisition as per matrix given in Table 4.1 to cover all the activities that we have mentioned in section 3.2 for Windows 10 and Android 10 (even in every Android device state i.e. Unrooted, NANDroid and Rooted)

Acquired evidence images are copied to the external storage or padlock and then transferred to the forensic workstation to ensure the experimental host/device integrity and to conduct further analysis.

Targeted Application and Browsing Activities		**Windows 10**			**Android 10**											
		Registry	**Storage**	**RAM**	**Zram**			**Storage**			**RAM**			**ADB Logs**		
					UR[1]	**NB[2]**	**RT[3]**	**UR**	**NB**	**RT**	**UR**	**NB**	**RT**	**UR**	**NB**	**RT**
Application Installation		Yes	Yes	Yes	No	No	Yes	No	No	No	No	No	No	Yes	No	Yes
Application Execution (No browsing)		Yes	Yes	Yes	No	No	Yes	No	No	Yes	No	No	Yes	Yes	No	Yes
Application Execution and Closure (with Browsing)	**Browser Open**	Yes	Yes	Yes	No	No	No	No	Yes	Yes	No	No	No	Yes	No	Yes
	Browser Closed	Yes	Yes	Yes	No	No	No	No	No	Yes	No	No	No	Yes	No	Yes
Application Uninstallation		Yes	Yes	Yes	No	No	Yes	No	No	No	No	No	No	Yes	No	Yes

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

Table 4.1: Evidence Acquisition matrix for both platforms

Chapter 5

Evidence Analysis And Results

5.1 Windows 10

Forensics analysis of the evidence on Windows 10 is done in three phases:

Phase–1 - Registry analysis

Phase 2 – Storage Analysis

Phase 3 – Memory Analysis

Snapshots and evidence acquired in Section 3 were analyzed for all the targeted activities we have defined relevant to Tor privacy browser usage.

Phase–1 - Registry Analysis

Every application installed on Windows operating perform significant changes in the operating system filesystem hierarchy and Windows registry. Following tools are being used in this phase for analysis of registry snapshots:

a. Regshot
b. RegScanner
c. Notepad++
d. WinMerge

Our analysis of Windows 10 registry reveals that Tor browser adds the following number of registry keys:

1. Eight (08) registry keys – After installation
2. Three (03) other registry keys are relevant to Tor Brower installer file – during installation.

These registry keys are very helpful because all these have some erratic HEX values which change on the opening and closing of Tor browser. This helps the investigator build up the hypothesis that whether the user just installed the Tor privacy browser or used it as well after installation. In addition to this Tor privacy browser-related keys, there are a few other registry keys that will be helpful for investigators to check recent programs executed on the system. All these registry keys persist in the registry after uninstallation as shown in Figure 5.1 [1]

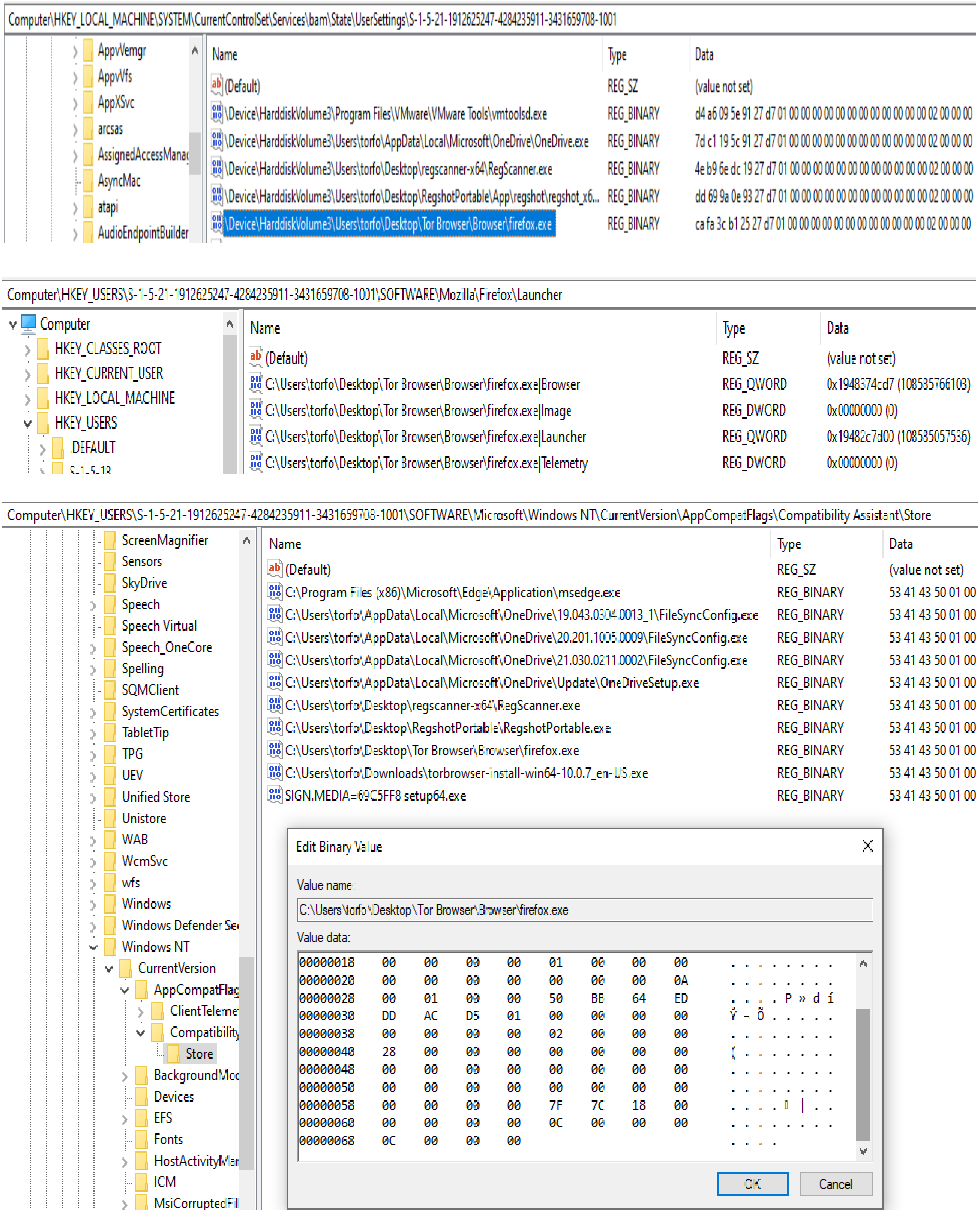


Figure 5.1: Registry Remnants of Tor Privacy Browser on Windows 10 After Uninstallation

Unfortunately, registry keys discovered do not provide any information related to the user browsing activities so user browsing habits cannot be explored from the registry analysis.

For further details, refer to Table 5.1 [1].

<table>
<tr><th>Registry Artifact(s)</th><th>Artifacts of Interest</th></tr>
<tr><td colspan="2">Pre-Installation (Registry Keys relevant to Tor Browser Installer)</td></tr>
<tr><td>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</td><td rowspan="3">Tells us about either the installation activity ever took place on the system.</td></tr>
<tr><td>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</td></tr>
<tr><td>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</td></tr>
<tr><td colspan="2">Post-Installation</td></tr>
<tr><td>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</td><td>Tells us about the number of executions of Tor browser since installation</td></tr>
<tr><td>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{00000000-0000-0000-0000-0000000000”0}"</td><td></td></tr>
<tr><td>HKU\S-1-5-21-1912625247-4284235911-3431659708-1001\SOFTWARE\Mozilla\Firefox\Launcher\C:\Users\torfo\Desktop\Tor Browser\Browser\firefox.exe|Image</td><td></td></tr>
<tr><td>HKU\S-1-5-21-1912625247-4284235911-3431659708-1001\SOFTWARE\Mozilla\Firefox\Launcher\C:\Users\torfo\Desktop\Tor Browser\Browser\firefox.exe|Telemetry</td><td></td></tr>
<tr><td>HKU\S-1-5-21-1912625247-4284235911-3431659708-1001\SOFTWARE\Mozilla\Firefox\Launcher\C:\Users</td><td>Values changes at every opening and closing of Tor browser</td></tr>
</table>

\torfo\Desktop\Tor Browser\Browser\firefox.exe|Launcher	
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	
Application Uninstallation	
All registry artifact created at the time of installation found	

Table 5.1: Registry Artifacts Retrieved from Windows 10

Phase–2 - Storage Analysis

In this phase, we analyzed forensic images we acquired of Tor Browser application root folder using FTK Imager. Three image files were analyzed [1]:

a. Post-Installation
b. Browser Open
c. Browser Closed

Application-related configuration and database files were analyzed in this phase to look for timestamps, bookmarks, and traces of user browsing activity, but no user browsing evidence was found on the filesystem. Uninstallation activity was not covered purposely for Windows 10 because it just involves deleting the main application root 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 because it requires plenty of time to recover deleted data which may hinder the timelines of digital investigations.

a. Post-installation

Artifacts produced on filesystem after when the Tor browser was installed but not executed on the Windows 10 were analyzed. In this stage, we only found application-related configuration files along with installation timestamps were found.

b. Browsing – Browser Open

Artifacts that are present in the hard disk once the browser was open are searched in this part of the analysis. Artifacts we found had only the downloaded data, bookmarks information, and application usage timestamps. Information related to the user browsing was still not found in this stage. However, all the registry artifacts we discovered in Phase 1 were present.

c. Browsing – Browser closed

In this stage of analysis, all those artifacts were searched which are present on the filesystem after browser was closed. All steps performed in previous part of storage analysis were also repeated in this stage. Similar artifacts we found in section ‘b’ were present in this stage. User browsing information is still not available. Similarly, all the registry artifacts we discovered in Phase 1 were present.

Phase–3 - Memory Analysis

In-memory analysis, we have divided the analysis into two parts [1]:

a. Tor only artifacts

For this part, we only extract artifacts that are related to the Tor application and its execution. We have extracted artifacts left on the memory of the Windows system:

1. After installation of Tor Browser
2. First time execution of Tor Browser
3. Subsequent executions of Tor Browser
4. After uninstallation of Tor Browser

HxD and Belkasoft Evidence Center are used for forensics analysis of all the acquired memory images in this part. After a comprehensive analysis of Tor only artifacts, we compiled a list of all retrieved artifacts during this part that are shown in Table 5.2 [1].

Sr.No.	Type of Artifact(s)
1	Application paths
2	Loaded EXE (firefox.exe and tor.exe) & DLL files

3	SQlite files along with Tables name and remnants of DB operations performed by Tor browser application
4	Built-in Windows Functions used by Tor browser application
5	Remnants of Resources used by Tor browser application
6	Tor router's information including 1) IP Addresses 2) Nicknames 3) Last available timestamps 4) Public keys used by Tor Router
7	User-agent information (Mozilla/5.0)
8	Blocklists and Extensions data (included timestamps) used by Tor browser application
9	Registry keys and values

Table 5.2: "Tor only" artifacts from Memory (RAM) on Windows 10

Artifacts at Sr. 6 are very helpful for law enforcement agencies in backtracking any user performing illegal activities using Tor Browser. This works by collecting artifacts from the Tor network relays with the help of concerned LEAs and ISPs but for the sake of this research, it is beyond our current scope of digital investigation; because in that case geographical laws and regulations come into play which may contribute to extended delays in completing investigations [1].

b. Browsing Artifacts

For this part of memory analysis, we only look for user browsing artifacts present in the memory. As explained earlier in Data Acquisition section, two VMware snapshots were acquired; one for each *"Browser Open"* and *"Browser Closed"* scenario. Memory images in VMware snapshots (.vmem files) are then analyzed using HxD and Belkasoft Evidence Center tools for browsing artifacts [1].

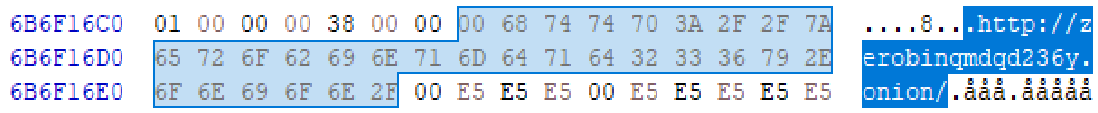
The technique used for analysis in this part is called "string searching" or "pattern matching". Using this technique, we found the following remnants in the memory of the under-study Windows system:

a. Websites/URLs visited by the user
b. Search queries
c. Credentials used (emails, usernames, and passwords)
d. Emails sent/ received using Webmail on Tor Browser
e. Uploaded & downloaded files using Tor Browser
f. Other artifacts

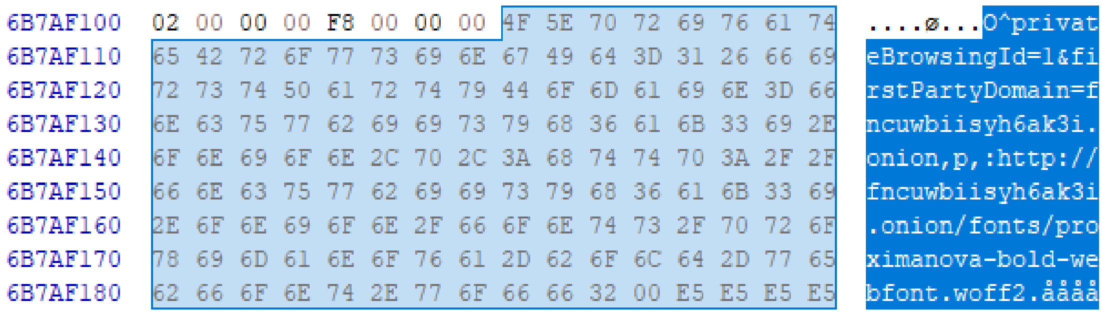
Most interesting part of our analysis is that we uncovered remnants of all the emails (including unread emails) present in Inbox of our Gmail, Outlook, and Secmail accounts used for this research.

The artifacts we found in **"Browser Open"** memory image were almost identical to the **"Browser Closed"** memory image which clearly implies that Tor browser does not instantly clear the user browsing history from memory while closing the Tor browser application.

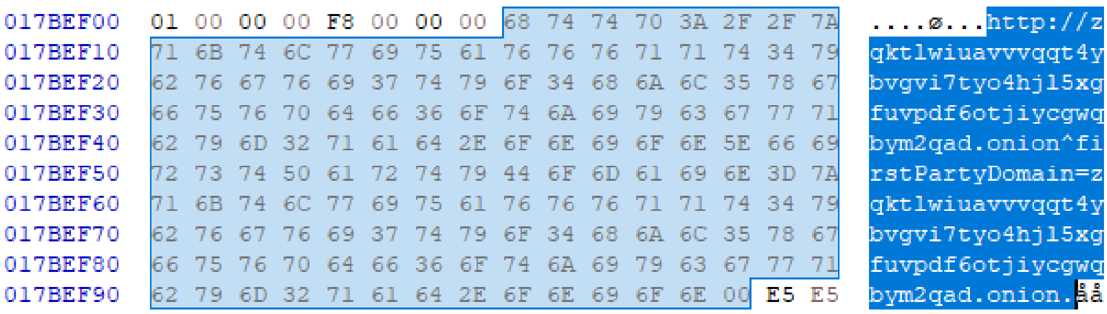
Screenshots of some of these artifacts are shown in Figure 5.2 [1].



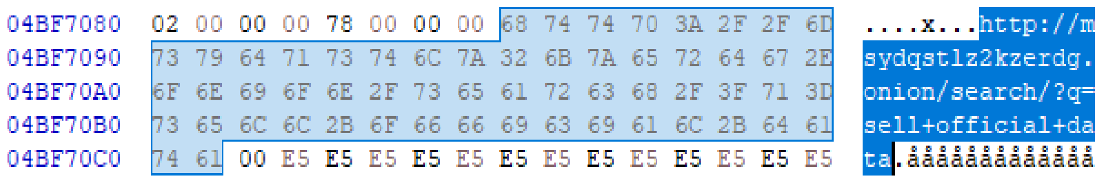
URL



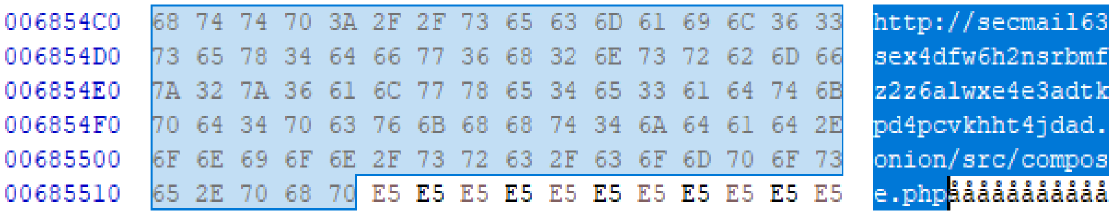
Website content



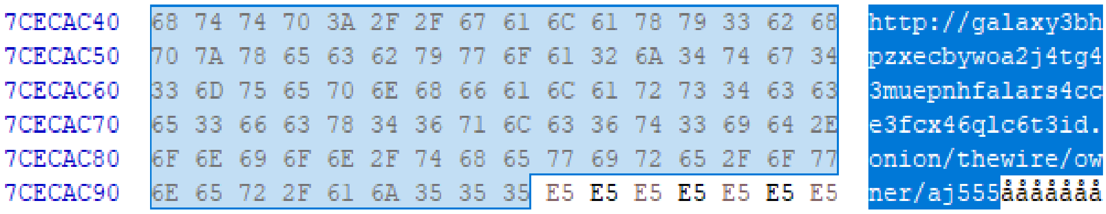
Private Browsing Traces



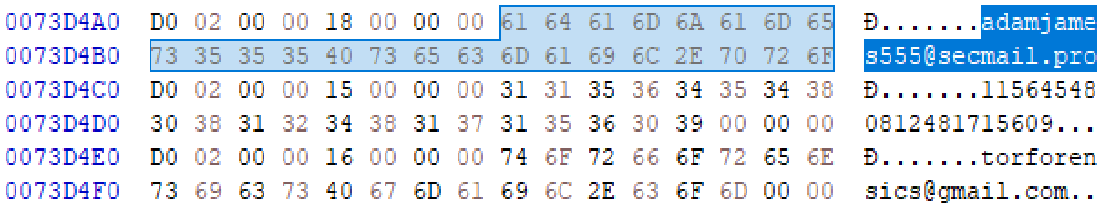
Search query



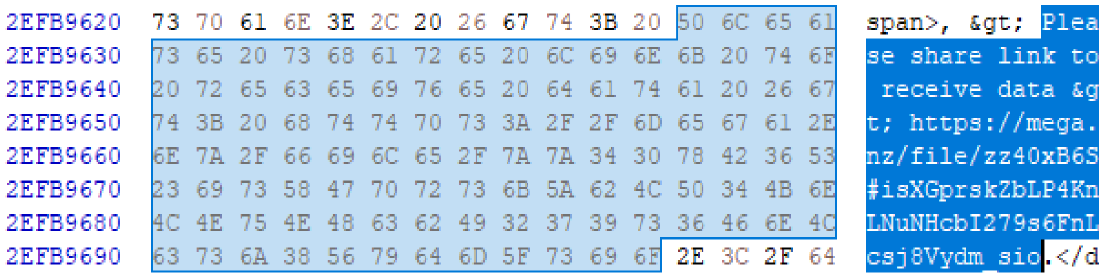
Onion Email Website



Onion Social Media Blog with Username in URL



Email Addresses



Sent Email Content

Figure 5.2: User Browsing Artifacts of Tor Browser from Memory On Windows 10

Summary of user browsing artifacts that we found in our Windows system memory are listed here in Table 5.3

<table>
<tr><th rowspan="2">Website Cat.</th><th rowspan="2">Sr.</th><th rowspan="2" colspan="2">Browsing Information</th><th rowspan="2">Browsing Activities Performed</th><th colspan="2">Browsing Artifacts found when</th></tr>
<tr><th>Browser Open</th><th>Browser Closed</th></tr>
<tr><td rowspan="3">Wiki</td><td rowspan="3">1</td><td>URL Title</td><td>Hidden Wiki</td><td rowspan="3">1. Browsed
2. Whistleblowing hyperlink clicked</td><td rowspan="3">▪ Website/URL traces
▪ Visited/Redirected URLs traces
▪ Website components (js,css) traces
▪ SOCKS socket traces</td><td rowspan="3">▪ Website/URL traces
▪ Visited/Redirected URLs traces
▪ Website components (js,css) traces</td></tr>
<tr><td>Website/ URL visited</td><td>zqktlwiuavvvqqt4ybvgvi7tyo4hjl5xgfuvpdf6otjiycgwqbym2qad.onion/wiki/index.php/Main_Page</td></tr>
<tr><td>Credentials Used</td><td>Not applicable</td></tr>
<tr><td rowspan="3">Search Engines</td><td rowspan="3">2</td><td>URL Title</td><td>Ahmia</td><td rowspan="3">1. Browsed
2. Search query "sell official data" performed
3. Clicked the first result & get redirected to 5j7saze5byfqccf3.onion/data/bullseye/main/ URL
4. Download components-mips64el.yml.gz file from the above URL</td><td rowspan="3">▪ Website/URL traces
▪ Visited/Redirected URLs
▪ Website components (js,css)
▪ Search query traces
▪ Downloaded file & URL traces
▪ Download timestamps</td><td rowspan="3">▪ Website/URL traces
▪ Visited/Redirected URLs traces
▪ Search query traces
▪ Downloaded file & URL traces
▪ Download timestamps</td></tr>
<tr><td>Website/ URL visited</td><td>msydqstlz2kzerdg.onion</td></tr>
<tr><td>Credentials Used</td><td>Not applicable</td></tr>
</table>

	3	**URL Title**	Duck-DuckGo	1. Browsed 2. Search query **"sell official data"** performed	▪ Website/URL traces ▪ Website components (js,css) traces ▪ Search query traces ▪ SOCKS socket traces	▪ Website/URL traces ▪ Search query traces
		Website/ URL visited	3g2upl4pq6kufc4m.onion			
		Credentials Used	Not applicable			
Cloud Storage/Sharing	4	**URL Title**	Google Drive	1. Browsed after login to ***Gmail*** using Google credentials at ***Sr. 10*** 2. Uploaded 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
		Website/ URL visited	drive.google.com			
		Credentials Used	torforensics@gmail.com			
	5	**URL Title**	MEGA	1. Browsed and then Login 2. Right click already uploaded **PDF file** and get Sharing link **https://mega.nz/file/zz40xB6S#isXGprskZbLP4KnLNuNHcbI279s6FnLcsj8Vydm_sio** 3. Copied the link to clipboard	▪ Website/URL traces ▪ Website components (js,css) traces ▪ Clipboard Operation traces ▪ Timestamps	▪ Website/URL traces ▪ Clipboard Operation traces
		Website/ URL visited	mega.nz/login mega.nz/fm			
		Credentials Used	torforensics@gmail.com			

					▪ Local Megasync client socket 127.0.0.1:6341 ▪ SOCKS Username/Password Traces ▪ SOCKS Socket Traces ▪ Login Email Traces	
	6	**URL Title**	ZeroBin	1. Browsed 2. Pasted the **Mega Sharing Link** from Clipboard copied in ***Sr. 5*** 3. Generate the Paste link 4. Link to the Paste **http://zerobinqmdqd236y.onion/?be163e348777b667#H7xg5DfMboatOgot8q439QNYTogRfXLAP74fmqzeXjI=** copied to clipboard	▪ Website/URL traces ▪ Website components (js,css) traces ▪ Clipboard Operation traces ▪ Generated Filesharing/Paste URL information traces ▪ Timestamps ▪ SOCKS Username/Password Traces	▪ Website/URL traces ▪ Generated Filesharing/Paste URL information traces
		Website/ URL visited	zerobinqmdqd236y.onion			
		Credentials Used	Not applicable			

					▪ Paste Token ID traces	
Money Transfer	7	**URL Title**	Stealth-Pay	Browsed only	▪ Only domain name found	▪ Nothing found
		Website/ URL visited	https://www.stealthpay.com/requestmoney			
		Credentials Used	Not applicable			
Secure Commu-nications	8	**URL Title**	Keybase	1. Browsed 2. Tried downloading software from **/download** 3. Just visited the URL **.../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
		Website/ URL visited	fncuwbiisyh6ak3i.onion			
		Credentials Used	Not applicable			

	9	**URL Title**	SecMail	1. Browsed 2. Login 3. Open first email in the Inbox which is from **torforensics@gmail.com** for reading 4. Replied the email with contents as shown below: *Email To:* ***"torforensics@gmail.com"*** *Email Subject: **"Re: Impt Data"*** *Email 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
		Website/ URL visited	secmail63sex4dfw6h2nsrbmfz2z6alwxe4e3adtkpd4pcvkhht4jdad.onion			
		Credentials Used	adamjames555@secmail.pro			
Emails	10	**URL Title**	Gmail	1. Browsed 2. Login 3. Checked the email	▪ Website/URL traces ▪ Website components (js,css) traces ▪ Only Inbox Emails traces ▪ Timestamps ▪ Cookies ▪ Response header traces	▪ Website/URL traces ▪ Website components (js,css) traces
		Website/ URL visited	mail.google.com			
		Credentials Used	torforensics@gmail.com			

					▪ Login timestamps ▪ Login Email & Password traces	
	11	**URL Title**	Outlook	1. Browsed 2. Login 3. Composed and sent the email with contents as shown below: *Email To:* ***torforensics@gmail.com*** *Email Subject: **"Imp Stuff"*** *Email 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
		Website/ URL visited	outlook.live.com			
		Credentials Used	torforensics@outlook.com			
Voice/ Video Chat	12	**URL Title**	Skype	1. Browsed 2. Login 3. URL **web.skype.com** was opened but received **"browser not supported"** message	▪ Website/URL traces ▪ Website components (js,css) traces ▪ Timestamps ▪ SOCKS Socket Traces ▪ Skype Local Socket	▪ Website/URL traces ▪ Login Email traces ▪ Timestamps
		Website/ URL visited	Web.skype.com Secure.skype.com www.skype.com			
		Credentials Used	torforensics@outlook.com			

					▪ Login timestamps ▪ Login Email & Password traces	
Social	13	**URL Title**	Galaxy3	1. Browsed 2. Login 3. Composed the **Wire Blog** post with content as 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
		Website/ URL visited	galaxy3bhpzxecbywoa2j4tg43muepnhfalars4cce3fcx46qlc6t3id.onion			
		Credentials Used	adamjames555@tutanota.com			

Table 5.3: "User Browsing" artifacts retrieved from Memory (RAM) on Window 10

In case, the application was uninstalled immediately after recent browsing, the investigator may find traces of artifacts shown in Table 5.4 below [1]:

In case, the application was uninstalled immediately after recent browsing, you may find:	
1	Opened/Redirected URL Traces
2	Website components traces (.js, .css etc)
3	Downloaded filenames & URLs
4	Login email address traces
5	Timestamps
6	Sessions IDs or other session related information
7	Traces of any clipboard operation performed in the context of browser

Table 5.4: Remnants of User Browsing Artifacts of Tor on Memory (RAM) Of Window 10 after consequent uninstallation

Summary of all the browsing artifacts retrieved from Tor privacy browser on Windows 10 are listed in Table 5.5 [1]:

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

Table 5.5: Summary of "User Browsing" Artifacts retreived from Window 10

5.2 Android 10

On Android, forensic analysis is performed a little bit different as compared to what we have performed in case of Windows 10; because in case of Android 10, we must perform analysis on evidence acquired in three different access level/states of Android device that are:

Phase 1 – ADB Logs Analysis of "Un-rooted" and "Rooted" Android device

Phase 2 – Storage Analysis of "Un-rooted", "Rooted" Android device and "NANDroid backup" acquired

Phase–3 - Zram analysis of "Rooted" Android device

Phase 4 – Memory Analysis of "Rooted" Android device

In our first phase, ADB logs of Un-rooted and rooted Android device for artifacts that were acquired during execution and browsing activities of Tor browser.

While in second phase, we performed storage analysis on evidence acquired from un-rooted, rooted and NANDroid backup of our device for storage artifacts

Finally, in third phase and fourth phase, Zram and memory images were analyzed for artifacts on rooted Android device.

Snapshots and evidence acquired in Section 3 were analyzed for all the targeted activities we have defined relevant to Tor privacy browser usage [1].

Phase 1 – ADB Logs Analysis

Un-rooted Device

On an un-rooted device, ADB logs (Dumpsys and Logcat service logs) analysis doesn't yield any significant evidence of user browsing activities. However, they only reveal very less information which includes underlying activities of Tor browser application on device including timestamps as show in Figure 5.3.

```
[ 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 5.3: Application Activity Traces in Events.Log File

Rooted Device

Similarly, on rooted Android device, analysis of ADB logs still show only underlying activities of Tor browser application on device including timestamps. No user browsing information cannot be retrieved.

Phase 2 – Storage Analysis

Un-rooted Device

On an un-rooted device, analysis of Tor browser evidence acquired from storage does not yield any significant evidence of user browsing activities except downloaded files and application-related files.

Application-related files only reveal about few timestamps relevant to Installation of Tor on Android device.

Rooted Device

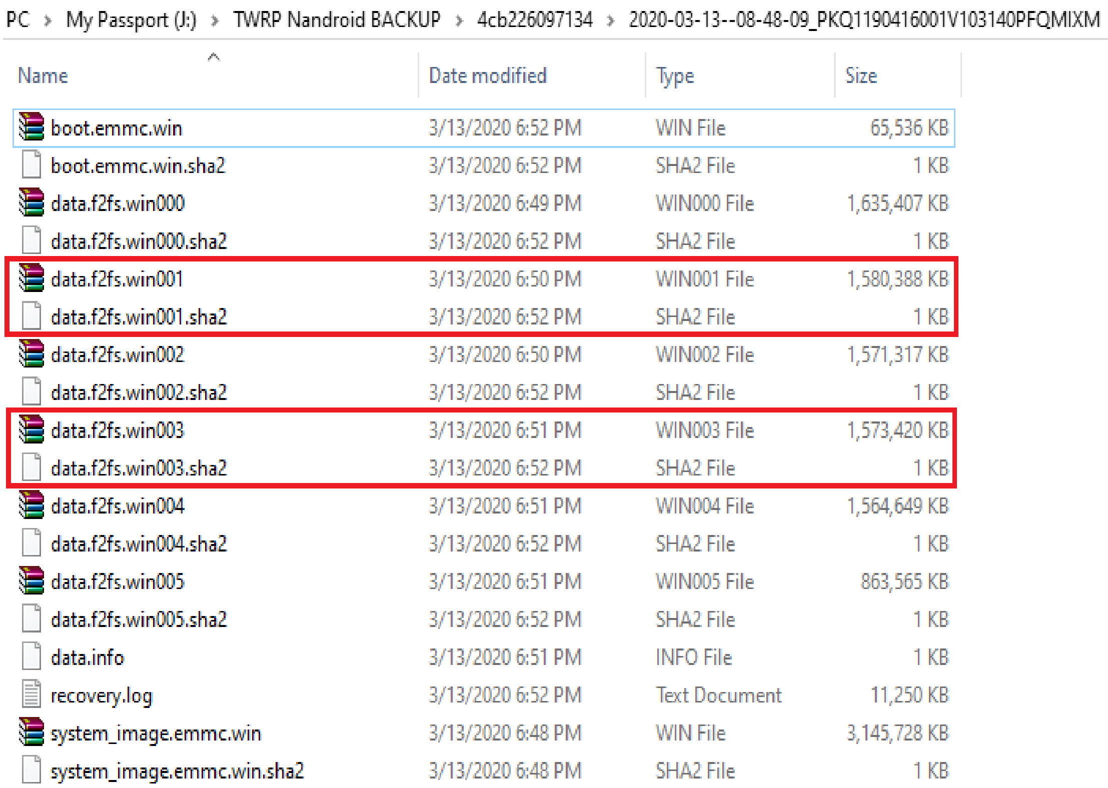
Rooting the device allows us to access the Tor application root directory **/data/data/org.torproject.torbrowser**/ on filesystem using Root Browser application and MOBILEdit Forensic Express. Analysis of the files using HxD, Notepad and DB Browser for SQLite tools yield only following information:

- Bookmarks
- Timestamps
- Tor circuit information

No user browsing information was retrieved from the Tor browser storage evidence acquired from the rooted android device except downloaded files.

NANDroid Backup

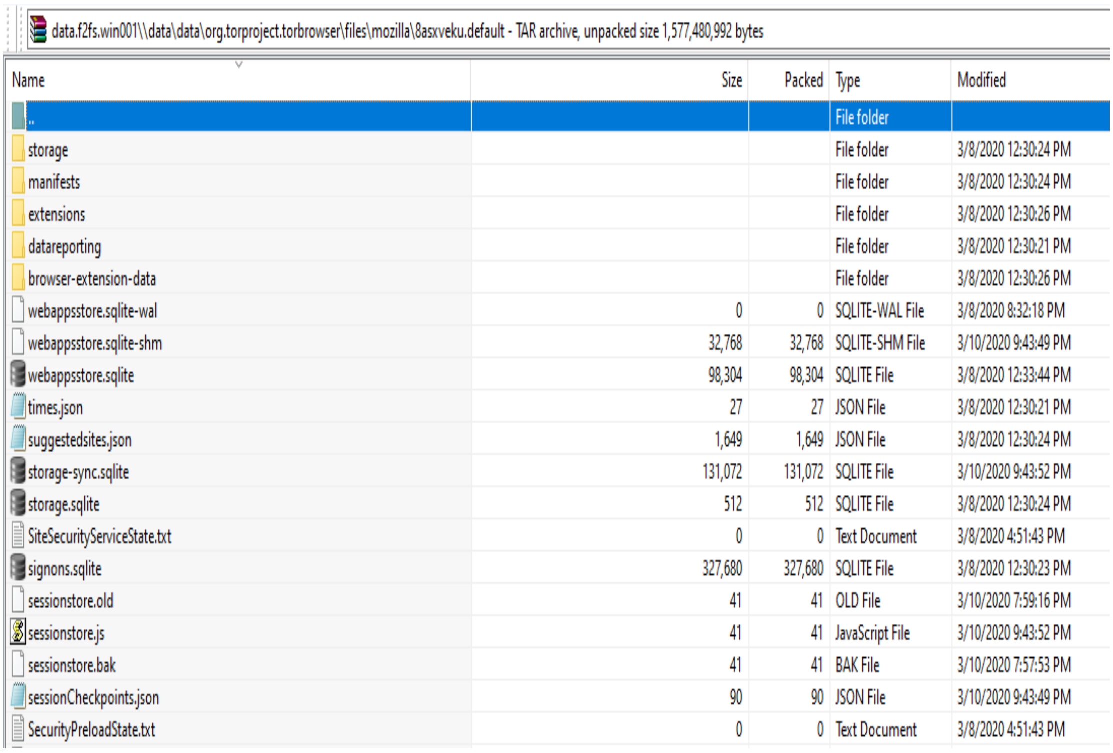
After extracting the **org.torproject.torbrowser** directory from ***/data/data*** folder in **userdata** archive available in the NANDroid backup (acquired as per Section 4) as shown in Figure 5.4 and 5.5 below [1].



PC › My Passport (J:) › TWRP Nandroid BACKUP › 4cb226097134 › 2020-03-13--08-48-09_PKQ1190416001V103140PFQMIXM

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 5.4: Important archives in NANDroid Backup from forensic point of view (Highlighted)



data.f2fs.win001\\data\data\org.torproject.torbrowser\files\mozilla\8asxveku.default - TAR archive, unpacked size 1,577,480,992 bytes

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 5.5: Tor Application files inside NANDroid Backup archive

Analysis of the files using HxD, Notepad and DB Browser for SQLite application yields only following information:

- Bookmarks
- Timestamps
- Tor circuit information

No user browsing information was retrieved from the NANDroid backup except downloaded files. ADB Logs were not available in NANDroid backup.

Phase 3 – Zram Analysis (Rooted Android Device Only)

As per our existing information and research, this area of Android device was first time explored for retrieving browsing and other application related digital evidence.

a. Tor only artifacts

During this stage, we analyzed the artifacts left on Zram during Tor browser's installation, execution without any browsing and uninstallation.

Summary of all the artifacts retrieved during these activities are listed in Table 5.6 below [1]:

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 (.js, .css etc)
3	Downloaded files along with their local path
4	Uploaded files remnants
5	Login email address traces

Table 5.6: "Tor only" artifacts from Zram on Android 10

b. Browsing Artifacts

i. Browser Open

Our analysis uncovers most of the websites/URLs and domain names that we have visited in our sample investigative scenario.

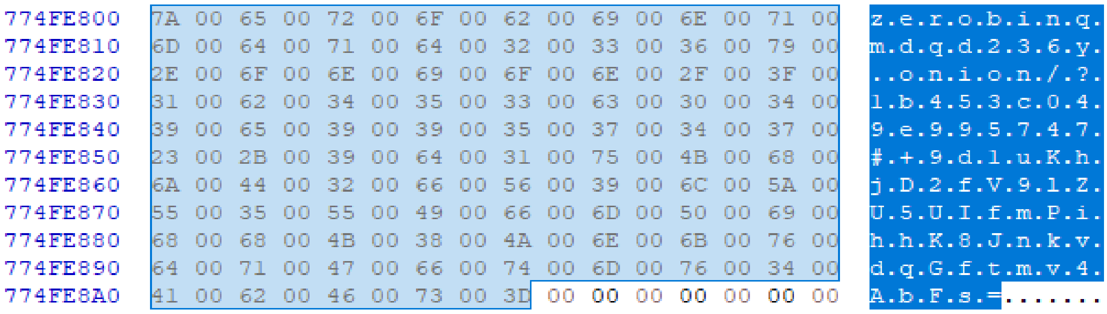
This includes few webpage components, redirected/visited URLs information; Downloaded files information including filename, URLs, and local paths; most of the search queries we performed & clipboard content from Tor; traces of few email addresses & usernames used for login and communication, but no passwords and email content was found; session information, timestamps of few visited websites were also found. We also bookmarked websites. 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 info were found in this case. Some of these artifacts are shown in Figure 5.6 [1]

```
84F6B820  09 06 00 00 00 00 05 05 25 08 09 08 45 78 63 61  ........%...Exca
84F6B830  76 61 74 6F 72 20 2D 20 73 65 61 72 63 68 20 69  vator - search i
84F6B840  6E 20 64 61 72 6B 6E 65 74 68 74 74 70 3A 2F 2F  n darknethttp://
84F6B850  32 66 64 36 63 65 6D 74 34 67 6D 63 63 66 6C 68  2fd6cemt4gmccflh
84F6B860  6D 36 69 6D 76 64 66 76 6C 69 33 6E 66 37 7A 6E  m6imvdfvli3nf7zn
84F6B870  36 72 66 72 77 70 73 79 37 75 68 78 72 67 62 79  6rfrwpsy7uhxrgby
84F6B880  70 76 77 66 35 66 61 64 2E 6F 6E 69 6F 6E 2F 80  pvwf5fad.onion/€
```

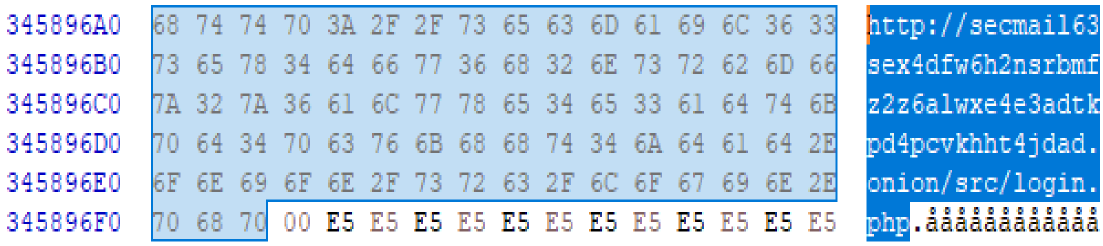
Bookmarks

```
829CB2A0  68 00 74 00 74 00 70 00 3A 00 2F 00 2F 00 32 00  h.t.t.p.:././.2.
829CB2B0  66 00 64 00 36 00 63 00 65 00 6D 00 74 00 34 00  f.d.6.c.e.m.t.4.
829CB2C0  67 00 6D 00 63 00 63 00 66 00 6C 00 68 00 6D 00  g.m.c.c.f.l.h.m.
829CB2D0  36 00 69 00 6D 00 76 00 64 00 66 00 76 00 6C 00  6.i.m.v.d.f.v.l.
829CB2E0  69 00 33 00 6E 00 66 00 37 00 7A 00 6E 00 36 00  i.3.n.f.7.z.n.6.
829CB2F0  72 00 66 00 72 00 77 00 70 00 73 00 79 00 37 00  r.f.r.w.p.s.y.7.
829CB300  75 00 68 00 78 00 72 00 67 00 62 00 79 00 70 00  u.h.x.r.g.b.y.p.
829CB310  76 00 77 00 66 00 35 00 66 00 61 00 64 00 2E 00  v.w.f.5.f.a.d...
829CB320  6F 00 6E 00 69 00 6F 00 6E 00 2F 00 00 00 00 00  o.n.i.o.n./.....
```

Domain Names



URL



Onion Email Websites

Figure 5.6: User browsing traces in Zram of Android 10 during Browser Open

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 found. In application related traces, a small number of application related file paths and only some loaded application files were found in this case.

Summary of all the Tor browsing artifacts we retrieved from Android 10 Zram are listed in Table 5.7 [1]

Website Cat.	**Sr.**	**Browsing Information**		**Browsing Activities Performed**	**Browsing Artifacts found when**	
					Browser Open	**Browser Closed**
Wiki	1	**URL Title**	Hidden Wiki	1. Browsed 2. **Whistleblowing** hyperlink clicked	▪ No artifact found	▪ No artifact found
		Website/URL visited	zqktlwiuavvvqqt4ybvgvi7tyo4hjl5xgfuvpdf6otjiycgwqbym2qad.onion/wiki/index.php/Main_Page			
		Credentials Used	Not applicable			
Search Engines	2	**URL Title**	Ahmia	1. Browsed 2. Search query **"sell official data"** performed 3. Clicked the first result & get redirected to **5j7saze5byfqccf3.onion/data/experimental/main/** URL 4. Download **components-arm64.yml.xz** file from above URL	▪ Website/URL traces ▪ Visited/Redirected URLs ▪ Website components (js,css) ▪ Search query traces ▪ Downloaded file & URL traces	▪ Downloaded filename and local path on storage
		Website/URL visited	msydqstlz2kzerdg.onion			
		Credentials Used	Not applicable			
	3	**URL Title**	DuckDuckGo	1. Browsed 2. Search query **"sell official data"** performed	▪ No artifact found	▪ No artifact found
		Website/URL visited	3g2upl4pq6kufc4m.onion			

		Credentials Used	Not applicable			
	4	**URL Title**	Excavator	1. Browsed 2. Search query **"sell official data"** performed	▪ Website/URL traces ▪ Search query traces	▪ No artifact found
		Website/URL visited	2fd6cemt4gmccflhm6imvdfvli3nf7zn6rfrwpsy7uhxrgbypvwf5fad.onion			
		Credentials Used	Not applicable			
Cloud Storage/ Sharing	5	**URL Title**	Google Drive	1. Browsed only after login to ***Gmail*** using Google credentials at ***Sr. 13***	▪ No artifact found	▪ No artifact found
		Website/URL visited	drive.google.com			
		Credentials Used	torforensic@gmail.com			
	6	**URL Title**	MEGA	1. Browsed and then Login 2. Uploaded the file **IMG-20210122-WA0005.jpg** from device 3. Retrieved the sharing link of uploaded file in ***Pt. 2*** 4. Copied the link to clipboard	▪ Website/URL traces ▪ Website components (js,css) ▪ Uploaded file traces ▪ Clipboard operation traces	▪ Website/URL traces ▪ Uploaded file traces
		Website/URL visited	mega.nz/login mega.nz/fm			
		Credentials Used	torforensic@gmail.com			

	7	**URL Title**	ZeroBin	1. Browsed 2. Pasted the **Mega.nz** file sharing link copied to clipboard at ***Sr. 7*** 3. Generated the Paste link containing content **"/?a3e1481092fb04b9"**	▪ Only domain name	▪ Only domain name
		Website/URL visited	zerobinqmdqd236y.onion			
		Credentials Used	Not applicable			
	8	**URL Title**	StrongHold Paste	1. Browsed 2. Composed the Paste with content as shown below: *Paste title: **"Pix"*** *Paste data:* ***https://goo.gl/xZgh1qu*** 3. Password-protected the Paste 4. Generated the Paste link containing content **"/pocsxm1d5/2uo2vh"**	▪ Only domain name	▪ Only domain name
		Website/URL visited	nzxj65x32vh2fkhk.onion			
		Credentials Used	Not applicable			
	9	**URL Title**	SecureDrop	1. Browsed 2. Clicked **"Get started"** hyperlink and received codename **"unloving cornflake ecosphere decipher trifocals scotch reiterate"** on next page 3. Clicked **"Submit documents"** on page 4. Uploaded **IMG-20210122-WA0005.jpg** to webserver	▪ Only domain name	▪ No artifact found
		Website/URL visited	arujlhu2zjjhc3bw.onion arujlhu2zjjhc3bw.onion/lookup			
		Credentials Used	Not applicable			

Money Transfer	10	**URL Title**	Stealth-Pay	Browsed only	▪ Website/URL traces ▪ Website components (js,css)	▪ Only domain name
		Website/URL visited	https://www.stealthpay.com/requestmoney			
		Credentials Used	Not applicable			
Secure Commun-ications	11	**URL Title**	Keybase	1. Browsed 2. Clicked **"Send secure message" hyperlink and get** redirected to **"play.google.com"** for Keybase Android APK installation page.	▪ Visited/Redirected URLs	▪ No artifact found
		Website/URL visited	fncuwbiisyh6ak3i.onion			
		Credentials Used	Not applicable			
Emails	12	**URL Title**	SecMail	1. Browsed 2. Login 3. Checked emails received from Gmail and Outlook email addresses at ***Sr. 13 & 14*** 4. Email from Gmail account was replied with content as shown below: *Email To:* ***torforensics@gmail.com*** *Email Subject:* ***"Re: Impt Data"*** *Email Body:* ***"Find here: https://goo.gl/xZgh1qu"***	▪ Website/URL traces ▪ Website components (js,css)	▪ No artifact found
		Website/URL visited	secmail63sex4dfw6h2nsrbmfz2z6alwxe4e3adtkpd4pcvkhht4jdad.onion			
		Credentials Used	adamjames555@secmail.pro			

	13	**URL Title**	Gmail	1. Browsed 2. Login 3. Email was composed and sent with content as shown below: *Email To:* ***adamjames555@secmail.pro*** *Email Subject:* ***"Impt Data"*** *Email 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
		Website/URL visited	mail.google.com			
		Credentials Used	torforensic@gmail.com			
	14	**URL Title**	Outlook	1. Browsed 2. Login 3. Email was composed and sent with content as shown below: *Email To:* ***adamjames555@secmail.pro*** *Email Subject:* ***"Imp Data"*** *Email 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
		Website/URL visited	outlook.live.com			
		Credentials Used	torforensic@outlook.com			
Voice/ Video Chat	15	**URL Title**	Skype	1. Browsed 2. Login 3. Visited Account overview page	▪ Website/URL traces	▪ Only secure.skype.com domain name
		Website/URL visited	Web.skype.com Secure.skype.com			

			www.skype.com	4. URL **web.skype.com** was opened but received **"browser not supported"** message	▪ Login Email Address traces ▪ Sessions IDs	
		Credentials Used	torforensic@outlook.com			
Social Media	16	**URL Title**	Galaxy3	1. Browsed 2. Login 3. **/Settings** link visited 4. Blogs link **"/blog/owner/aj555"** was visited	▪ Website/URL traces ▪ Website components (js,css) ▪ Login Email Address traces ▪ Username	▪ Login Email Address traces
		Website/URL visited	galaxy3bhpzxecbywoa2j4tg43muepnhfalars4cce3fcx46qlc6t3id.onion			
		Credentials Used	adamjames555@tutanota.com			
Torrents	17	**URL Title**	The Pirate Bay	1. Browsed 2. Search query **"privacy"** was performed with Application check box marked on webpage 3. From the result, **Privacy Shield** URL was opened 4. Torrent magnet link was copied to clipboard with content as shown below: **"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
		Website/URL visited	https://thepiratebay.cx/en1/			
		Credentials Used	Not applicable			

Table 5.7: "Browsing" related artifacts from Zram on Android 10

Phase 4 – Memory Analysis (Rooted Android Device Only)

In this analysis, we only cover two types of activities in memory analysis because of our memory acquisition tool's limitation as mentioned in section 4.2. We analyzed the "Tor only" and "User Browsing" artifacts during "Browser Open" scenario [1].

a. Tor only artifacts

Unlike Zram, we only analyzed the artifacts left on memory Tor browser was opened either with or without any browsing activity performed.

Summary of all the artifacts retrieved during these activities are listed in Table 5.8 as shown below [1]:

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 (.js, .css etc)
3	Downloaded files along with their local path
4	Uploaded files remnants
5	Login email address traces

Table 5.8: "Tor only" artifacts from Memory (RAM) on Android 10

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; session information and timestamps of few visited websites were also found. We also found bookmarked websites. 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 info were found in this case.

Some of these artifacts we discovered are shown in Figure 5.7 [1]

```
0045E440  68 74 74 70 3A 2F 2F 7A 71 6B 74 6C 77 69 75 61  http://zqktlwiua
0045E450  76 76 76 71 71 74 34 79 62 76 67 76 69 37 74 79  vvvqqt4ybvgvi7ty
0045E460  6F 34 68 6A 6C 35 78 67 66 75 76 70 64 66 36 6F  o4hjl5xgfuvpdf6o
0045E470  74 6A 69 79 63 67 77 71 62 79 6D 32 71 61 64 2E  tjiycgwqbym2qad.
0045E480  6F 6E 69 6F 6E 2F 77 69 6B 69 2F 69 6E 64 65 78  onion/wiki/index
0045E490  2E 70 68 70 2F 4D 61 69 6E 5F 50 61 67 65 23 57  .php/Main_Page#W
0045E4A0  68 69 73 74 6C 65 62 6C 6F 77 69 6E 67 00 00 00  histleblowing...
```

Clicked URL

```
0020AD60  74 00 00 00 55 84 36 1D 68 74 74 70 3A 2F 2F 6D  t...U„6.http://m
0020AD70  73 79 64 71 73 74 6C 7A 32 6B 7A 65 72 64 67 2E  sydqstlz2kzerdg.
0020AD80  6F 6E 69 6F 6E 2F 73 65 61 72 63 68 2F 3F 71 3D  onion/search/?q=
0020AD90  73 65 6C 6C 2B 6F 66 66 69 63 69 61 6C 2B 64 61  sell+official+da
0020ADA0  74 61 00 00 00 00 00 00 80 35 45 6F 00 00 00 00  ta......€5Eo....
```

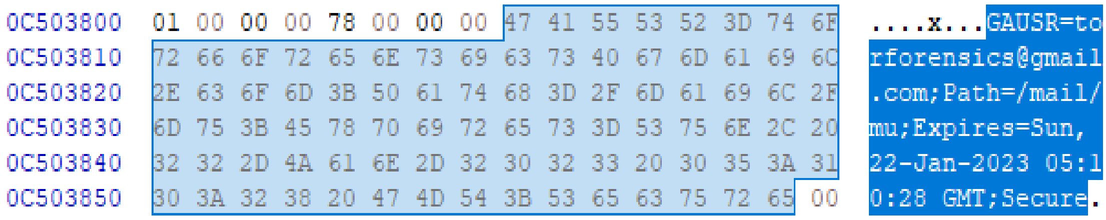
Search Query Traces

```
0EBFF3A0  43 6F 6D 70 6F 6E 65 6E 74 73 2D 61 72 6D 36 34  Components-arm64
0EBFF3B0  2E 79 6D 6C 2E 78 7A E5 E5 E5 E5 E5 E5 E5 E5 E5  .yml.xzååååååååå
0EBFF3C0  32 30 32 31 2D 30 31 2D 32 32 20 30 32 3A 31 31  2021-01-22 02:11
```

Downloaded Filename with Timestamps

```
07143500  01 00 00 00 38 00 00 00 68 74 74 70 3A 2F 2F 6E  ....8...http://n
07143510  7A 78 6A 36 35 78 33 32 76 68 32 66 6B 68 6B 2E  zxj65x32vh2fkhk.
07143520  6F 6E 69 6F 6E 2F 70 6F 63 73 78 6D 6C 64 35 2F  onion/pocsxmld5/
07143530  32 75 6F 32 76 68 00 E5 E5 E5 00 E5 E5 E5 E5 E5  2uo2vh.ååå.ååååå
```

Generated Paste Link URL



Login Email Traces with Timestamps

Partial Email Content

Figure 5.7: User browsing traces in memory of Android 10 during Browser Open

ii. Browser Closed

Analysis in this state is not possible due to our tool's limitation so it reveals nothing [1].

Summary of all the Tor browsing artifacts that we retrieved from Android 10 RAM are listed in Table 5.9 [1]

Website Cat.	**Sr.**	**Browsing Information**		**Browsing Activities Performed**	**Browsing Artifacts found when**	
					Browser Open	**Browser Closed**
Wiki	1	**URL Title**	Hidden Wiki	1. Browsed 2. **Whistleblowing** hyperlink clicked	▪ Website/URL traces ▪ Visited/Redirected URLs ▪ Website components (js,css) ▪ SOCKS socket traces ▪ Response Headers ▪ Bookmarked information	No evidence of Tor browsing can be acquired from memory due to memory acquisition tool limitations
		Website/URL visited	zqktlwiuavvvqqt4ybvgvi7tyo4hjl5xgfuvpdf6otjiycgwqbym2qad.onion/wiki/index.php/Main_Page			
		Credentials Used	Not applicable			
Search Engines	2	**URL Title**	Ahmia	1. Browsed 2. Search query **"sell official data"** performed 3. Clicked the first result & get redirected to **5j7saze5byfqccf3.onion/data/experimental/main/** URL 4. Download **components-arm64.yml.xz** file from above URL	▪ Website/URL traces ▪ Visited/Redirected URLs ▪ Website components (js,css)	
		Website/URL visited	msydqstlz2kzerdg.onion			
		Credentials Used	Not applicable			

					▪ Search query traces ▪ SOCKS socket traces ▪ Downloaded filename & URL trace ▪ Download timestamps	
	3	**URL Title**	DuckDuckGo	1. Browsed 2. Search query **"sell official data"** performed	▪ Website/URL traces ▪ Visited/Redirected URLs ▪ Website components (js,css) ▪ Search query traces ▪ SOCKS socket traces	
		Website/URL visited	3g2upl4pq6kufc4m.onion			
		Credentials Used	Not applicable			
	4	**URL Title**	Excavator	1. Browsed 2. Search query **"sell official data"** performed	▪ Website/URL traces	
		Website/URL visited	2fd6cemt4gmccflhm6imvdfvli3nf7zn6rfrwpsy7uhxrgbypvwf5fad.onion			

		Credentials Used	Not applicable		▪ Visited/Redirected URLs ▪ Website components (js,css) ▪ Search query traces ▪ SOCKS socket traces ▪ Bookmarked information	
Cloud Storage/ Sharing	5	**URL Title**	Google Drive	1. Browsed only after login to ***Gmail*** using Google credentials at ***Sr. 13***	▪ Website/URL traces ▪ Website components (js,css) ▪ Login Email address traces ▪ Response Headers	
		Website/URL visited	drive.google.com			
		Credentials Used	torforensic@gmail.com			
	6	**URL Title**	MEGA	1. Browsed and then Login 2. Uploaded the file **IMG-20210122-WA0005.jpg** from device 3. Retrieved the sharing link of uploaded file in ***Pt. 2*** 4. Copied the link to clipboard	▪ Website/URL traces ▪ Website components (js,css) ▪ Uploaded file information	
		Website/URL visited	mega.nz/login mega.nz/fm			
		Credentials Used	torforensic@gmail.com			

					▪ Clipboard operation traces ▪ Local upload temp folder ▪ SOCKS socket traces ▪ Login Email address traces	
	7	**URL Title**	ZeroBin	1. Browsed 2. Pasted the **Mega.nz** file sharing link copied to clipboard at ***Sr. 7*** 3. Generated the Paste link containing content **"/?a3e1481092fb04b9"**	▪ Website/URL traces ▪ Website components (js,css) ▪ Clipboard operation traces ▪ Generated Filesharing/Paste URL information traces ▪ SOCKS socket traces	
		Website/URL visited	zerobinqmdqd236y.onion			
		Credentials Used	Not applicable			
	8	**URL Title**	StrongHold Paste	1. Browsed 2. Composed the Paste with content as shown below: *Paste title: "**Pix**"*	▪ Website/URL traces	
		Website/URL visited	nzxj65x32vh2fkhk.onion			

		Credentials Used	Not applicable	*Paste data:* ***https://goo.gl/xZgh1qu*** 3. Password-protected the Paste 4. Generated the Paste link containing content **"/pocsxm1d5/2uo2vh"**	▪ Website components (js,css) ▪ Clipboard operation traces ▪ Generated Filesharing/Paste URL information traces ▪ SOCKS socket traces ▪ Response Headers ▪ Timestamps	
	9	**URL Title**	SecureDrop	1. Browsed 2. Clicked **"Get started"** hyperlink and received codename **"unloving cornflake ecosphere decipher trifocals scotch reiterate"** on next page 3. Clicked **"Submit documents"** on page 4. Uploaded **IMG-20210122-WA0005.jpg** to webserver	▪ Website/URL traces ▪ Website components (js,css) ▪ Generated Random Username traces ▪ SOCKS socket traces	
		Website/URL visited	arujlhu2zjjhc3bw.onion arujlhu2zjjhc3bw.onion/lookup			
		Credentials Used	Not applicable			
Money Transfer	10	**URL Title**	Stealth-Pay	Browsed only	▪ Website/URL traces	
		Website/URL visited	https://www.stealthpay.com/requestmoney			

		Credentials Used	Not applicable		▪ Website components (js,css) ▪ SOCKS socket traces ▪ Response Headers ▪ PHP Session IDs	
Secure Commun-ications	11	**URL Title**	Keybase	1. Browsed 2. Clicked **"Send secure message" hyperlink and get** redirected to **"play.google.com"** for Keybase Android APK installation page.	▪ Website/URL traces ▪ Visited/Redirected URLs ▪ Website components (js,css) ▪ SOCKS socket traces	
		Website/URL visited	fncuwbiisyh6ak3i.onion			
		Credentials Used	Not applicable			
Emails	12	**URL Title**	SecMail	1. Browsed 2. Login 3. Checked emails received from Gmail and Outlook email addresses at ***Sr. 13 & 14*** 4. Email from Gmail account was replied with content as shown below: *Email To:* ***torforensics@gmail.com***	▪ Website/URL traces ▪ Website components (js,css) ▪ SOCKS socket traces ▪ Only partial received email traces	
		Website/URL visited	secmail63sex4dfw6h2nsrbmfz2z6alwxe4e3adtkpd4pcvkhht4jdad.onion			
		Credentials Used	adamjames555@secmail.pro			

				Email Subject: ***"Re: Impt Data"*** *Email Body:* ***"Find here: https://goo.gl/xZgh1qu"***		
	13	**URL Title**	Gmail	1. Browsed 2. Login 3. Email was composed and sent with content as shown below: *Email To:* ***adamjames555@secmail.pro*** *Email Subject:* ***"Impt Data"*** *Email 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	
		Website/URL visited	mail.google.com			
		Credentials Used	torforensic@gmail.com			
	14	**URL Title**	Outlook	1. Browsed 2. Login 3. Email was composed and sent with content as shown below: *Email To:* ***adamjames555@secmail.pro*** *Email Subject:* ***"Imp Data"***	▪ Website/URL traces ▪ Website components (js,css)	
		Website/URL visited	outlook.live.com			
		Credentials Used	torforensic@outlook.com			

				*Email Body: "**Please share link to receive data**"*	▪ Login email address & Password traces ▪ Session Information ▪ Cookies ▪ Response Headers ▪ Only Subject: header & body of emails found	
Voice/ Video Chat	15	**URL Title**	Skype	1. Browsed 2. Login 3. Visited Account overview page 4. URL **web.skype.com** was opened but received **"browser not supported"** message	▪ Website/URL traces ▪ Website components (js,css) ▪ Login email address traces ▪ Base64 encoded Session Token ▪ X-CSRF Token ▪ Cookies ▪ Timestamps ▪ Response Headers	
		Website/URL visited	Web.skype.com Secure.skype.com www.skype.com			
		Credentials Used	torforensic@outlook.com			

Social Media	16	**URL Title**	Galaxy3	1. Browsed 2. Login 3. **/Settings** link visited 4. Blogs link **"/blog/owner/aj555"** was visited	▪ Website/URL traces ▪ Website components (js,css) ▪ Login email address & Password traces ▪ Usernames ▪ Timestamps ▪ Session Token ▪ SOCKS Username, Password	
		Website/URL visited	galaxy3bhpzxecbywoa2j4tg43muepnhfalars4cce3fcx46qlc6t3id.onion			
		Credentials Used	adamjames555@tutanota.com			
Torrents	17	**URL Title**	The Pirate Bay	1. Browsed 2. Search query **"privacy"** was performed with Application check box marked on webpage 3. From the result, **Privacy Shield** URL was opened 4. Torrent magnet link was copied to clipboard with content as shown below: **"magnet:?xt=urn:btih:2A3B…"**	▪ Website/URL traces ▪ Website components (js,css) ▪ Search Results traces ▪ Downloaded magnet filename & URL traces	
		Website/URL visited	https://thepiratebay.cx/en1/			
		Credentials Used	Not applicable			

Table 5.9: "User Browsing" artifacts from Memory (RAM) On Android 10

So finally, all the user browsing artifacts that we gathered from our experimental Android 10 setup are listed below in Table 5.10 [1]:

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 5.10: Summary of all User Browsing artifacts from Android 10 device

Chapter 6

Discussions

6.1 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 [9] – this study examined the registry, storage, and memory after normal 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. [10] - 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. [13] – 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 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 [5] 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.

We have made an effort to cover every possible scenario that an investigator may face during the forensic analysis of Tor on both platform(s) [1] 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.

A detailed comparison of proposed and existing work can be seen in Table 6.1 [1].

Related Work	***OS Platform(s)***	***Tor Browser version***	***Evidence venues explored***	***Installation***	***No-browsing Execution***	***Execution with Browsing***		***Uninstallation***
						Browser Open	***Browser Closed***	
Nedaa Al Barghouthy et.al.	*Android 2.3.4*	*V2.28*	*Storage*	✗	✗	✓	✗	✗
Nedaa Al Barghouthy et.al.	*Android 4.1.1*	*V2.28*	*Storage*	✗	✗	✓	✗	✗
C.Meda et.al.	*Android 5.0*	*V15.0.1-RC-3*	*Storage*	✓	✓	✓	✗	✓
R. Nelson et.al.	*Windows 7*	*V7.0.5*	*Storage*	✓	✓	✓	✓	✗
			Registry	✓	✗	✓	✗	✗
A. Jadoon et.al.	*Windows 8.1*	*V7.0.2*	*Storage*	✗	✗	✓	✓	✗
			Registry	✓	✗	✗	✗	✓
			Memory	✗	✓	✓	✓	✗
		V7.5.2	*Storage*	✗	✓	✓	✓	✗

M.Muir et.al.	*Windows 10 version 1703*		*Registry*	✓	✓	✓	✓	✓
			Memory	✓	✓	✓	✓	✓
A. Warren	*Windows 10 October 2017 build*	*V5.0*	*Storage*	✓	✓	✓	✓	✗
			Registry	✓	✓	✓	✓	✓
			Memory	✓	✓	✓	✓	✓
Proposed Work	*Windows 10 version 20H2 October 2020*	*V10.0.7*	*Storage*	✓	✓	✓	✓	✓
			Registry	✓	✓	✓	✓	✓
			Memory	✓	✓	✓	✓	✓
	Android 10 *June 2020 update*	*V68.7.0*	*Storage*	✓	✓	✓	✓	✓
			ADB Logs	✓	✓	✓	✓	✓
			Registry	✓	✓	✓	✓	✓
			Memory	✗	✓	✓	✓	✗

Table 6.1: Detailed Comparison Of Existing Work And Adopted Methodology Of Tor Browser Forensics

6.2 Recommendations for Tor Project Developers

From this study, we infer that Tor developers have employed several decoy settings in Tor Browser to provide fail-safe anonymity and privacy to its users, but these settings do nothing to extend the default privacy provided in the browser. However, several browser-related configuration settings and timestamps are stored in plaintext files on the filesystem which can forensically reveal usage patterns of the Tor browser.

In this regard, we recommend that respected developers should include the mechanism to store browser-based settings in encrypted files that can only be decrypted by the browser executable or application while it is executing; and cannot be extracted using any other text editor.

We have demonstrated in this research that 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 highly recommended [1].

Chapter 7

Conclusions

After completing this research, we have concluded that the Tor privacy browser leaves only limited amount of *user browsing* information on the filesystem/storage in each of the operating system i.e. Windows 10 and Android 10 but enough information about evidence of *Tor browser usage* on device/operating system can be retrieved from the storage; this application usage evidence can also be retrieved from the *Registry on Windows 10* and *ADB Logs on Android 10* in addition to the filesystem/storage of both.

Zram is very interesting component of the Android operating system that acts as a swap filesystem; first time explored in this study as per our knowledge and information. We have also explored *memory (RAM)* for extracting Tor usage and browsing artifacts on both Windows 10 and Android 10 operating system in this study.

Based on our analysis results, we have determined that the possibility of extracting evidence from Zram i.e. both usage and browsing artifacts is approximately 60 percent which is considerably good for an anonymous browser if investigators have time and resource constraints to explore for more evidence in RAM [1]. On the other hand, we have retrieved more user browsing artifacts from memory than from Zram on Android operating system. However, as a comparative study, we observed that *Tor browser reveals more artifacts on memory (RAM) of Windows than memory on Android* either it is usage or browsing artifacts.

Similar to previous studies conducted on respective problem, we have also concluded that the possibility of determining user attribution using the retrieved Tor browser artifacts is complicated.

7.1 Future work

After completing this research, we have decided to conduct a significant amount of future work that will help the forensic community.

i. We will carry out detailed network forensic analysis of Tor circuit established while using Tor privacy browser. We have plans to conduct this network forensic on latest Android, iOS, Windows, MacOS and different Linux distributions in enterprise usage, because limited research has been performed in this area. There is a need to extract potential evidence from network circuit information to trace nodes where the Tor browser relays information so that anonymous/illicit browsing information can be retrieved from the nodes which are in the geographical and legal boundary of the area with the help of ISPs and LEAs.

ii. We also desire to perform detailed forensic analysis of Tor browser on MacOS and iOS devices as this area is rarely covered by researchers and will help forensic investigators in getting comfortable with forensically analyzing these platforms.

iii. We also like to extend this research to develop a specialized module(s) for MobilEdit and other forensic tools; to carry out the detailed evidence acquisition and analysis of Tor privacy browser on Windows, Linux, Android, and iOS platforms that will help reduce forensic investigator's burden.

References

[1] M. Arshad, M. Hussain, H. Tahir, S. Qadir, F. Memon and Y. Javed, "Forensic Analysis of Tor Browser on Windows 10 and Android 10 Operating Systems", IEEE Access, pp. 1-1, 2021. Available: 10.1109/access.2021.3119724 [Accessed 30 November 2021].

[2] Obstacles to Cybercrime Investigations, "Cybercrime Module 5 Key Issues", UNODC.org, https://www.unodc.org/e4j/en/cybercrime/module-5/key-issues/obstacles-to-cybercrime-investigations.html 2019. [Online].

[3] J. Porup, "What is the Tor Browser? And how the dark web browser works", CSO Online, 2019. [Online].

[4] N. Barghouthy, A. Marrington and I. Baggili, "The forensic investigation of android private browsing sessions using orweb", 2013 5th International Conference on Computer Science and Information Technology, 2013.

[5] N. Al Barghouthy and A. Marrington, "A Comparison of Forensic Acquisition Techniques for Android Devices: A Case Study Investigation of Orweb Browsing Sessions", 2014 6th International Conference on New Technologies, Mobility and Security (NTMS), 2014.

[6] C. Meda And M. Epifani, "Study And Analysis Of Orweb (And Orfox) Anonymizer(S) On Android Devices" Dfrws Eu 2016

[7] S. Teng and C. Wen, "A Forensic Examination of Anonymous Browsing Activities", 2018.

[8] M. Asim, M. Amjad, W. Iqbal, H. Afzal, H. Abbas and Y. Zhang, "AndroKit: A toolkit for forensics analysis of web browsers on android platform", Future Generation Computer Systems, vol. 94, pp. 781-794, 2019.

[9] A. Warren, "Tor Browser Artifacts in Windows 10", 2017.

[10] A. Jadoon, W. Iqbal, M. Amjad, H. Afzal and Y. Bangash, "Forensic Analysis of Tor Browser: A Case Study for Privacy and Anonymity on the Web", Forensic Science International, vol. 299, pp. 59-73, 2019.

[11] R. Nelson, A. Shukla and C. Smith, "Web Browser Forensics in Google Chrome, Mozilla Firefox, and the Tor Browser Bundle", Studies in Big Data, pp. 219-241, 2019.

[12] G. Satrya and F. Kurniawan, "A Novel Android Memory Forensics for Discovering Remnant Data", International Journal on Advanced Science, Engineering and Information Technology, vol. 10, no. 3, p. 1008, 2020.

[13] Muir, Matt, Petra Leimich, and William J. Buchanan. "A forensic audit of the tor browser Bundle." Digital Investigation 29 (2019): 118-128.

[14] Horsman, Graeme, Ben Findlay, Josh Edwick, Alisha Asquith, Katherine Swannell, Dean Fisher, Alexander Grieves, Jack Guthrie, Dylan Stobbs, and Peter McKain. "A forensic examination of web browser privacy-modes." Forensic Science International: Reports 1 (2019): 100036.

[15] Satvat, Kiavash, Matthew Forshaw, Feng Hao, and Ehsan Toreini. "On the privacy of private browsing–a forensic approach." In Data Privacy Management and Autonomous Spontaneous Security, pp. 380-389. Springer, Berlin, Heidelberg, 2013.

[16] Alfosail, Malak, and Peter Norris. "Tor Forensics: Proposed Workflow for Client Memory Artefacts." Computers & Security (2021): 102311.

[17] "Android Debug Bridge (adb) | Android Developers", Android Developers, 2021. [Online].

[18] "Logcat command-line tool", Android Developers, 2021. [Online].

[19] "dumpsys", Android Developers, 2021. [Online].

[20] H. Bilal, "What is the difference between a rooted and unrooted Android? - Quora", Quora.com, 2019. [Online].

[21] D. STIEBEN, "What Is A Nandroid Backup and How Exactly Does It Work?", MUO, 2014. [Online].

[22] O. André Vadla Ravnås, "frida-tools", PyPI, 2021. [Online]. Available: https://pypi.org/project/frida-tools/.

[23] "Frida • A world-class dynamic instrumentation framework", Frida • A world-class dynamic instrumentation framework, 2021. [Online].

[24] K. Kent, S. Chevalier, T. Grance and H. Dang, "Guide to Integrating Forensic Techniques into Incident Response", NIST Technical Series Publications, 2006. [Online].

[25] M. Sahu, "How To Unlock Bootloader On Xiaomi Mi A3 – Mi Official", MyPhoneUpdate, 2020.

[26] M. Sahu, "How To Install TWRP Recovery On Xiaomi Mi A3", MyPhoneUpdate, 2019.

[27] D. Bhardwaj, "How to Install Magisk on Android? (Comprehensive Guide)", The Custom Droid, 2020.

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

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

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

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

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

[33] Apple Reinvents the Phone with iPhone, Apple Newsroom, 2021

[34] zram: Compressed RAM-based block devices — The Linux Kernel documentation", Kernel.org, 2021

[35] Cybersecurity Spotlight - The Surface Web, Dark Web, and Deep Web", CIS, 2021

[36] Digital Evidence and Forensics", National Institute of Justice, 2021

[37] UNINSTALLING | Tor Project | Tor Browser Manual, Tb-manual.torproject.org, 2021

[38] What is F2FS - Let's Know in details, Xiaomi Community, 2021

[39] Introduction to Fridump, PenTest Corner, 2021

[40] GitHub - Nightbringer21/fridump: A universal memory dumper using Frida, GitHub, 2021

[41] How to Unlock the Bootloader of Xiaomi Mi A3, GetDroidTips, 2021

[42] A. Jude, How To Unlock Xiaomi MI A3 Bootloader (Android One), AdimorahBlog, 2021

[43] Implementing A/B Updates | Android Open Source Project, Android Open Source Project, 2021

[44] Install TWRP Recovery on Xiaomi Mi A3 (laurel_sprout) - The Step-by-Step Guide, The Custom Droid, 2021

[45] TWRP 101: How to Make a NANDroid Backup & Restore Your Entire Phone", Gadget Hacks, 2021