information

IMPACT FACTOR 2.4

CITESCORE 6.9

Article

Analyzing Tor Browser Artifacts for Enhanced Web Forensics, Anonymity, Cybersecurity, and Privacy in Windows-Based Systems

Muhammad Shanawar Javed, Syed Muhammad Sajjad, Danish Mehmood, Khawaja Mansoor, Zafar Iqbal, Muhammad Kazim and Zia Muhammad

Special Issue

Cybersecurity, Cybercrimes, and Smart Emerging Technologies

Edited by

Dr. Mohamed Hammad, Dr. Ahmed A. Abd El-Latif, Dr. Abdelhamied Ashraf Ateya and Prof. Dr. Mohammed ElAffendi

MDPI

https://doi.org/10.3390/info15080495

information

Article

Analyzing Tor Browser Artifacts for Enhanced Web Forensics, Anonymity, Cybersecurity, and Privacy in Windows-Based Systems

Muhammad Shanawar Javed [1], Syed Muhammad Sajjad [1], Danish Mehmood [2], Khawaja Mansoor [1], Zafar Iqbal [1], Muhammad Kazim [3] and Zia Muhammad [3,4,*]

[1] Department of Cyber Security, Air University, Islamabad 44000, Pakistan; 211831@students.au.edu.pk (M.S.J.); muhammad.sajjad@mail.au.edu.pk (S.M.S.); mansoor.hassan@au.edu.pk (K.M.); zafar.iqbal@mail.au.edu.pk (Z.I.)
[2] Department of Computing, Shaheed Zulfiqar Ali Bhutto Institute Of Science and Technology, Islamabad 44000, Pakistan; dr.danish@szabist-isb.edu.pk
[3] Department of Computer Science, North Dakota State University, Fargo, ND 58102, USA; muhammad.kazim@ndsu.edu
[4] Department of Computer Science and Technology, University of Jamestown, Jamestown, ND 58405, USA
* Correspondence: zia.muhammad@ndsu.edu

Abstract: The Tor browser is widely used for anonymity, providing layered encryption for enhanced privacy. Besides its positive uses, it is also popular among cybercriminals for illegal activities such as trafficking, smuggling, betting, and illicit trade. There is a need for Tor Browser forensics to identify its use in unlawful activities and explore its consequences. This research analyzes artifacts generated by Tor on Windows-based systems. The methodology integrates forensic techniques into incident responses per NIST SP (800-86), exploring areas such as registry, storage, network, and memory using tools like bulk-extractor, autopsy, and regshot. We propose an automated PowerShell script that detects Tor usage and retrieves artifacts with minimal user interaction. Finally, this research performs timeline analysis and artifact correlation for a contextual understanding of event sequences in memory and network domains, ultimately contributing to improved incident response and accountability.

Keywords: Tor Browser; web forensics; digital forensics; anonymity; privacy; Windows-based systems; operating system forensics; cybersecurity; network forensics; internet forensics

Citation: Javed, M.S.; Sajjad, S.M.; Mehmood, D.; Mansoor, K.; Iqbal, Z.; Kazim, M.; Muhammad, Z. Analyzing Tor Browser Artifacts for Enhanced Web Forensics, Anonymity, Cybersecurity, and Privacy in Windows-Based Systems. *Information* **2024**, *15*, 495. https://doi.org/10.3390/info15080495

Academic Editor: Aneta Poniszewska-Maranda

Received: 8 July 2024
Revised: 8 August 2024
Accepted: 13 August 2024
Published: 19 August 2024

1. Introduction

The Tor project stands at the forefront of endeavors to safeguard user privacy and anonymity in the domain of Internet usage [1]. By providing a mechanism for complete user anonymity on the Internet, Tor emerges as a powerful tool that, unfortunately, also attracts cybercriminals seeking to exploit its high level of anonymity. The inherent anonymity of Tor makes it an appealing choice for those with malicious intent, creating a potential pathway for the stealthy transmission of data across the tunneled network [2]. The malicious use of Tor extends beyond covert data transfers; evil actors could employ this browser to orchestrate numerous criminal activities. The intentional deletion of the Tor Browser post-use further complicates matters, as it enables agents to remain virtually untraceable by Law Enforcement Agencies (LEAs) or vigilant System Administrators [3]. Consequently, the increasing adoption of Tor by individuals with malicious motives underscores the pressing need for System Administrators and LEAs to grasp the paramount importance of forensic methodologies [4].

Understanding the significance of forensic methods becomes crucial in environments where Tor usage is prohibited [5]. System Administrators and LEAs must equip themselves with the necessary tools and knowledge to effectively counteract the illicit use of Tor. Forensic methods, tailored to extract precise results, become indispensable in transforming

digital artifacts associated with Tor usage into legal evidence [6]. This transformation is essential for navigating legal landscapes where Tor usage is a potential violation, demanding rigorous adherence to legal procedures and protocols [7]. In the intricate interplay between the advocacy for privacy and the imperatives of law enforcement, the roles assumed by System Administrators and Law Enforcement Agencies (LEAs) become pivotal [8]. Balancing the respect for privacy rights with enforcing the law requires a careful understanding of forensic techniques designed for examining Tor-related activities [9,10]. The challenge is to use these techniques effectively to find evidence that can stand up to tough legal examinations, ensuring a fair legal process.

Forensic methods applied in the context of Tor usage extend beyond conventional digital investigations. They look into the details of registry forensics, network forensics, memory forensics, and storage forensics, each playing a crucial role in understanding Tor-related activities [11,12]. Network forensics examines how data move within the Tor network, trying to trace digital footprints left by those with malicious intentions [13].

As System Administrators and LEAs struggle with the challenges posed by the misuse of Tor, it becomes imperative to foster a collaborative environment [14]. Collaborative efforts between these entities can increase the exchange of knowledge and insights, empowering both parties to stay ahead of evolving cyber threats. Furthermore, cultivating a shared understanding of the legal frameworks surrounding Tor usage is crucial to ensure that forensic evidence stands the test of legal inspection. In brief, the evolving landscape of Internet anonymity, represented by tools like Tor, necessitates a proactive approach from System Administrators and LEAs. The dual challenge of protecting individual privacy rights while shortening potential cyber threats underscores the importance of tough forensic methodologies [3]. As technology continues to advance, staying one step ahead of cybercriminals becomes contingent on continuous learning, adaptation, and collaboration between those responsible for maintaining system integrity and those entrusted with upholding the law [15]. The synergy between forensic methods, legal frameworks, and collaborative efforts is the linchpin in establishing a resilient defense against the misuse of Tor and similar anonymizing technologies [16,17].

Analyzing Tor Browser artifacts for enhanced web forensics is crucial in today's digital landscape where anonymity, cybersecurity, and privacy are increasingly important concerns [18,19]. As the use of the Tor Browser and other anonymous networks grows, understanding how to detect and analyze their artifacts becomes vital for law enforcement agencies, security researchers, and organizations seeking to protect sensitive information. By analyzing Tor Browser artifacts, investigators can identify patterns and connections that may be indicative of malicious activities, such as criminal activity, espionage, or cyberattacks [20]. Furthermore, analyzing Tor Browser artifacts can help uncover the identities of individuals using these browsers to engage in illegal or unethical activities, thus promoting accountability and safety online [10]. Additionally, analyzing Tor Browser artifacts can also provide valuable insights into the development of more effective countermeasures against anonymity networks, ultimately enhancing overall cybersecurity and privacy [21].

The proposed research aims to contribute significantly to the field of digital forensics by undertaking a comprehensive investigation of Tor Browser activity. Unlike previous studies, which often focused on limited subsets of forensic techniques, this research sought to leverage all four essential methods: memory forensics, storage forensics, network forensics, and registry forensics. By integrating these techniques, this study endeavored to achieve a more thorough understanding of Tor Browser usage patterns and associated forensic artifacts. This research aimed to answer the following questions:

1. What specific artifacts are generated by the Tor Browser on Windows-based operating systems?
2. How can these artifacts be effectively identified and analyzed using forensic methodologies?

3. Can an automated PowerShell script be developed to detect and retrieve Tor artifacts with minimal user interaction, thereby enhancing the efficiency of forensic investigations?
4. How does the correlation of these artifacts contribute to understanding the sequence of events in cases involving the use of the Tor Browser?

One of the primary goals of this research was to expand the scope of Tor artifacts in Windows registries, aiming to collect a broader range of artifacts from target PCs. Reverse engineering techniques in storage forensics will be employed to recover deleted artifacts, providing deeper insights into user activity. Network forensics will play a crucial role, followed by a simulated dark web usage scenario for analyzing PCAP and network behavior. Through ethical research practices and careful analysis, this study sought to advance the efficiency and accuracy of digital forensic investigations, ultimately providing valuable insights for cybersecurity professionals and law enforcement agencies. By addressing the limitations of previous methodologies and embracing a holistic approach, this research aims to contribute meaningfully to the advancement of digital forensic techniques for investigating Tor Browser activity and an automated approach to gather and discover Tor Browser traces in the computer, saving the investigator's time. The key contributions of this study are the following:

1. This research aims to broaden the scope of Tor artifacts in Windows registries, collecting a wider range of artifacts from target PCs. It employs reverse engineering techniques in storage forensics to recover deleted artifacts, providing deeper insights into user activity.
2. This study seeks to enhance the efficiency and accuracy of digital forensic investigations. It addresses the limitations of previous methodologies and contributes to the advancement of digital forensic techniques for investigating Tor Browser activity.
3. This research proposes an automated PowerShell script that detects Tor usage and conducts a forensic analysis to retrieve Tor artifacts with minimal user interaction. This approach saves the investigator's time and improves the overall efficiency of the investigation process.
4. This study performed timeline analysis and artifact correlation for a contextual understanding of how events unfold across both memory and network domains. This contributes to improved incident response and enhanced online security, providing valuable insights for cybersecurity professionals and law enforcement agencies.

Section 2 of this paper defines Tor's background, windows architecture, and related work. Section 3 provides a methodology that includes the experimental setup, tools, and their rationale. Section 4 contains the method's deployment and implementation. Section 5 provides an automated script for Tor artifact retrieval. Section 6 provides the results and discussion, followed by the conclusion and future work in Section 7.

2. Background

Tor 13.5.2 is free and open-source software [22]. It is an implementation in Firefox used to carry out onion routing; it encrypts all data during communication, and this network is maintained by servers called volunteer nodes [23]. Onion routing typically operates in a decentralized manner, meaning that there is no single point of control. The distributed nature of the network enhances its robustness and resistance to attacks [3]. The encryption of data using the Tor Browser is accomplished in stages [24,25], like an onion. The directory server dedicates a path for routing. Figure 1 describes the main components of the Tor Browser, which are the following:

- Tor Browser: This is the main application, represented as the central node in the diagram.
- Modified Firefox ESR (Extended Support Release):
    1. The Tor Browser is built on a customized version of Firefox ESR.
    2. It is specifically modified for enhanced privacy and anonymity.
- Tor Network:

1. This is the core of Tor's anonymity features.
2. It encrypts and routes traffic through multiple nodes: a. Entry Node (Guard): The first node in the Tor circuit. b. Middle Node: An intermediate node in the circuit. c. Exit Node: The final node that connects to the internet.

- Privacy Extensions:
  1. NoScript:
     (a) Blocks JavaScript by default to prevent potential security vulnerabilities.
     (b) Users can selectively enable scripts for trusted sites.
  2. HTTPS Everywhere:
     (a) Enforces secure connections by automatically upgrading HTTP connections to HTTPS when possible.
  3. Tor Button:
     (a) Manages Tor connection settings.
     (b) Allows users to access Tor network settings and change their circuit.

The following diagram shows how these components interact:

- The Tor Browser integrates the modified Firefox ESR, connects to the Tor Network, and includes the privacy extensions.
- The Tor Network is represented as a series of nodes (Entry, Middle, Exit) that encrypt and route the user's traffic.
- Each privacy extension (NoScript, HTTPS Everywhere, Tor Button) is connected to the main Extensions node.

This visual representation helps to understand how the Tor Browser works to protect user privacy by routing traffic through multiple encrypted nodes and employing various privacy-enhancing extensions and modifications to the base Firefox browser.

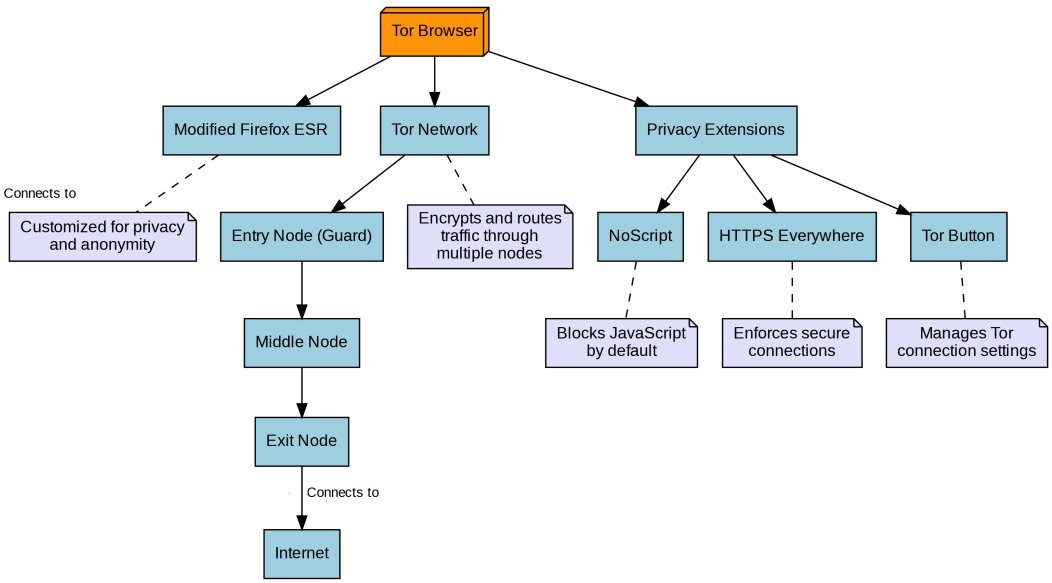

Figure 1. Tor components.

At each hop, one layer is decrypted from the original payload, and every node knows its decryption and sends the remaining encrypted payload to the other hop, where another layer is decrypted; hence, at the exit node, all layers are decrypted, and the decrypted payload is sent to the responder [26]. The same procedure is followed when the responder sends back the payload. This concept of constructing a virtual circuit for each network communication consists of three successive random relays. These random relays are assigned by the directory servers [27]. The Tor Browser is compatible with onion websites,

the websites that have the ".onion" extension in their address. These sites are accessible only through the Tor Browser, and they provide their users a more private and anonymous online experience. The ".onion" domain is not part of the traditional Domain Name System (DNS) used on the clarinet (the regular internet), and these sites can only be accessed by users who are connected to the Tor network [14]. The working principles of Tor are illustrated in Figure 2. The figure shows that onion routing consists of the following components:

1. **Client:** The application for sending is normally indicated to the Tor Browser (in our case).
2. **Directory Server:** In onion routing, a server that stores info about other nodes in the network is called a directory server [28]. It is responsible for providing the client with a safe passage to the server through onion routers. It decides the route that should be given to the client. All the active tor relays are listed in the directory server.
3. **Onion Encryption:** In onion routing, data are encrypted in layers as they travel through different routers. Each layer corresponds to a specific router in the circuit. At each router, one layer is decrypted, which exposes the next destination [29]. This layered encryption ensures the anonymity of both the sender and the final destination throughout the process.
4. **Entry Node or Guard Node:** The first relay in the chain. Also called an entry node or entry guard [30]. This node knows the sender's identity but does not know the ultimate destination or full path.
5. **Relay Node or Onion Router (OR):** It is a node that is neither an entrance nor an exit node; instead, it acts as a client for traffic routing [23]. All traffic going through this node remains encrypted.
6. **Exit Node or Egress Point:** It decrypts the third and last onion encryption and sends the data to the server.

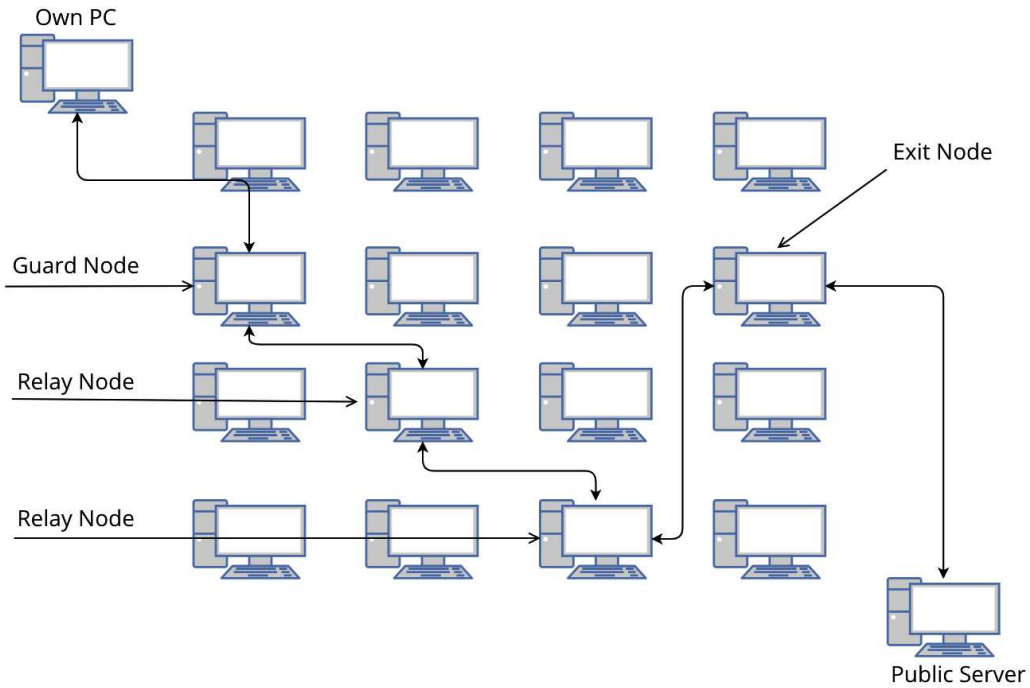

Figure 2. Working principles of Tor Browser.

2.1. Difference between Windows Versions

The differences between Windows 10 and Windows 11 are significant. From a hardware perspective, Windows 11 requires a 64-bit processor with two cores, 4 GB RAM, and 64 GB storage, whereas Windows 10 requires a 64-bit processor with a 1 GHz clock speed,

2 GB RAM, and 20 GB HDD. This implies that Windows 11 may have more resources available for analysis and may utilize different file systems and encryption methods.

The System32 Config folder in Windows 11 has a larger size range of 1.8 GB to 2.4 GB and contains approximately 400k to 500k registry entries, whereas in Windows 10, it ranges from 1.5 GB to 2.2 GB with around 300k to 400k registry entries. The new user interface in Windows 11, featuring a redesigned Start Menu and Taskbar, may store different information and artifacts compared to Windows 10. For instance, the pinned and recommended apps in the Start Menu may reveal user preferences and activities.

Windows 11 has enhanced security features, including Windows Hello and TPM 2.0, which provide more robust authentication and encryption mechanisms compared to Windows 10. Additionally, Windows 11 supports virtualization-based security (VBS), enabling security capabilities such as memory integrity (hypervisor-protected code integrity or HVCI) [31,32]. This means that Windows 11 may have stronger protection against malware and tampering, potentially affecting the integrity and reliability of forensic data.

The Task Manager in Windows 11 has been updated with more details and options compared to Windows 10. This may provide more information and control over system processes and resources, which can aid in the forensic analysis of system performance and activity. Furthermore, the redesigned File Explorer in Windows 11 features a new layout and additional features compared to the traditional File Explorer in Windows 10. This may affect the way files and folders are organized and accessed on the system, potentially influencing the forensic analysis of the file system and metadata.

2.2. Related Work

In this section, we will discuss Tor forensics, which was proposed in the past literature. So far, in the area of Tor forensics, there are research gaps because very minimal research has been conducted in this domain. There has been a lot of research about privacy and security in the Tor network but very limited in the field of dark web forensics or Tor forensics. A few research types are considered here in the context of forensic analysis of the dark web over Tor. In recent studies, various research efforts have been undertaken to explore the forensic implications of browsing activities conducted through the Tor network.

A study conducted in 2021 [33] focused on digital forensic methodology for Windows 10 and Android 10 devices, employing tools such as Regshot, RegScanner, and FTK Imager. However, the research was limited in scope as it did not encompass network forensics and solely concentrated on artifacts about Windows 10, neglecting the latest OS, Windows 11. Furthermore, the investigation failed to address artifacts remaining post-browser deletion. In 2017, ref. [7] utilized three forensic techniques to collect data from Windows 10 systems, employing tools such as Regripper, XWays, and Volatility. While the research claimed minimal traces left by the Tor Browser after deletion, it lacked clarity in defining the tools used and neglected a focus on network forensic artifacts. (See Table 1).

In 2019, ref. [8] employed memory, registry, and hard-disk forensic techniques, using tools such as FTK Imager, Magnet AXIOM, and Bulk Extractor. Despite yielding findings such as email addresses from memory and public keys of Tor relays, this research overlooked network forensic artifacts and utilized non-open-source tools, without incorporating automation for result gathering. In 2021, ref. [14] analyzed prefetch files and conducted RAM forensics using tools like Smsniff and Linux Reader, identifying Tor presence from prefetch and email addresses, yet lacked clarity in tool definitions and neglected the examination of network and registry artifacts. These studies collectively underscore the need for comprehensive forensic methodologies, encompassing network, memory, storage, and registry forensics, to effectively extract artifacts from Tor-enabled systems, thus enhancing investigative capabilities and ensuring thorough analysis of digital evidence.

Table 1. Literature review.

No	Research and Year	OS	Tools	Methodology	Contribution/ Findings	Limitations/ Assumptions
1.	[6], 2021	Windows 10	Regshot, RegScanner, FTK Imager	Used digital forensics methodology for Windows 10 and Android 10 devices.	Browsing history, registries, email address, downloaded timestamps.	Network forensics is not conducted in research; also, research has been based on Windows 10, while the latest OS is Windows 11. Moreover, the research has not focused on artifacts after the browser is deleted.
2.	[7], 2017	Windows 10	Regshot, Regripper, XWays, Volatility	Used three forensic techniques to collect data from Windows 10 device.	Path of file, processID, ports used, registries.	Claimed that the Tor Browser leaves no or very few traces after deletion. Also, the tools used here are not defined properly for forensic techniques except registry forensic tools.
3.	[8], 2019	Windows 8.1	FTK Imager, Magnet AXIOM, Bulk extractor, EnCase, Regshot, Volatility, Hex workshop	Used memory, registry and hard-disk forensic techniques.	Registries, emails from memory, public keys of Tor relays.	No artifacts from network forensics are collected. Also, the tool used for storage forensics is not open-source or free. No automation for gathering results is defined.
4.	[11], 2021	Windows 10	Smsniff, Linux Reader	Analyzed prefetch file and used RAM forensics to collect artifacts.	Tor presence from prefetch, email addresses, and public keys of the user.	Tools used were not defined properly. Also, the network and registry artifacts were not examined at all.
5.	[17], 2014	Windows 7	Wireshark, FTK, Network Miner	Used digital forensics techniques related to network, registry, and RAM forensics	Registries, browsing history, analysis of PCAPs.	This research was conducted in 2014 and is considered old at this point. Also, no storage artifacts were analyzed.
6.	[18], 2022	Windows 10	FTK Imager, Magnet RAM Capture, Belkasoft RAM Live Capturer, Bulk Extractor, Wireshark	Used registry, memory, hard-disk and network forensics on target system.	Browsing activities, HTTP header, tor exe, tor button, registries.	Could not reveal any browsing data with the network analysis. Did not cover memory artifacts after the Tor Browser is deleted.

Table 1. *Cont.*

No	Research and Year	OS	Tools	Methodology	Contribution/ Findings	Limitations/ Assumptions
7.	[20], 2022	Windows 10	FTK Imager, Autopsy, Bulk Extractor, Hex workshop	Gathered artifacts from memory, registry, and storage.	Browsing history, tor icon, public keys of tor relay (if browser is open), three registries.	Only three registry keys were found, and no automated methodology was presented for time-efficient findings.
8.	[21], 2021	Windows 10	Autopsy, Dumpit, Bulk Extractor Viewer	Analyzed browsing sessions with autopsy and collected artifacts from memory forensic techniques.	Email addresses, artifacts about Tor's presence.	No browsing history was recovered; also, the scope was limited to just memory and storage forensics. Network data were not examined at all.
9.	[22], 2021	Not defined	FTK, Belkasoft RAM capturer, Volatility	Gathered artifacts from RAM using memory forensics.	Network ports, browsing history.	Evidence acquisition process was not clearly defined. Also, it did not cover artifacts after the deletion of the browser.
10.	[33], 2019	Windows 10	FTK, HxD, SQLite,	Database and memory forensics.	Browsing history, cookies.	Used just memory forensics and examined a database file from prefetch. Also, a few artifacts were found. No automation for ease of examination.
11.	Proposed Research	Windows 11	Regshot, FTK Imager, Autopsy, Bulk Extractor, Volatility, NetworkMiner, RawCap	Used all the four forensic techniques (memory, network, storage, registry) for extracting artifacts from the target system.	URLs, email addresses, credentials, public keys, ports used, registries, prefetch files, location of Tor, location of Tor installer, usernames, Tor's HTTP header.	-

In 2014, ref. [13] focused on digital forensic techniques related to network, registry, and RAM forensics on Windows 7 systems, utilizing tools such as Wireshark, FTK, and Network Miner. Despite its comprehensive approach, the research's age renders its findings less applicable in the current context, and it did not encompass storage artifact analysis. Conversely, a more recent study in 2022 [3] employed a multifaceted approach, utilizing tools such as FTK Imager, Magnet RAM Capture, and Wireshark to examine registry, memory, hard disk, and network forensics on Windows 10 systems. While the study uncovered browsing activities and HTTP headers, it failed to reveal browsing data through network analysis and omitted memory artifact analysis post-Tor Browser deletion.

Furthermore, another study in 2022 [27] gathered artifacts from memory, registry, and storage using tools such as FTK Imager and Autopsy, yet only identified three registry keys, lacking an automated methodology for efficient findings. Similarly, in 2021, ref. [14] analyzed browsing sessions using Autopsy and collected artifacts from memory forensic techniques, revealing email addresses but failing to recover browsing history and neglecting network data examination.

Additionally, a study in 2021 [29] focused on memory forensics using tools like FTK and Volatility, uncovering network ports and browsing history but lacked clarity in the evidence acquisition process and did not cover artifacts post-browser deletion. Moreover, in 2019, ref. [6] utilized database and memory forensic techniques on Windows 10 systems, examining browsing history and cookies. However, the research solely relied on memory forensics, resulting in limited artifact findings, and lacked automation for ease of examination.

In 2021, ref. [34] gathered artifacts using memory forensics and prefetch view, yet did not conduct network analysis and overlooked registry examination. Finally, a study conducted in 2021 [35] employed memory forensics and analyzed paging files on Windows 10 systems, identifying visited URLs and browsing history, albeit with a limited scope, the absence of network forensics, and undefined deleted artifacts.

In light of these studies, it is evident that while considerable efforts have been made to analyze Tor-enabled systems, there remains a need for more time-effective forensic methodologies that encompass all facets of digital evidence extraction, including network, memory, storage, and registry forensics. Such methodologies would not only enhance investigative speed but also ensure thorough analysis and interpretation of digital artifacts in the context of Tor browsing activities.

Despite recent research, there are several issues in the forensics investigation of the Tor Browser. The previous studies performed Tor forensics but did not cover all forensic techniques (network, storage, memory, and registry forensics). The previous studies used the most recent version of the Tor Browser and Windows OS of that time, but those versions are now considered older [2,8,14]. Also, the previous studies provided no automation for the gathering of artifacts from any forensic technique in a time-effective way. Moreover, some of the previous studies did not cover artifacts that persist after the browser is deleted [3,14,26].

In most research, the computer systems vary in their RAM and storage capacities, and it is essential to reconstruct experimentation, considering that the results of one experiment may not be directly transferable to another. This discrepancy can become particularly pronounced over time as technology evolves, rendering what is novel today as outdated in the future.

When considering the motivation for focusing on Tor Browser forensics in Windows 11, despite the extensive literature available for Windows 10, several factors can justify the need for updated research and forensic methodologies as users migrate to Windows 11, forensic investigators will increasingly encounter this operating system in their work. It is important to understand the distinctions of Tor Browser forensics on the latest platform to stay ahead of potential challenges. Tor Browser usage is banned in many organizations and companies, not because of its anonymity and privacy; instead, it presents potential risks for organizations as Tor is a desired tool for cybercriminals. Existing research on investigating Tor Browser usage within organizations has limitations. Studies often miss key forensic techniques like network, storage, memory, and registry forensics. They also use Tor Browser versions that are outdated now, and lack automation for efficient artifact gathering. Additionally, they overlook artifacts that remain after browser deletion. Addressing these gaps requires a focused approach that covers all forensic techniques, uses up-to-date tools, and considers artifact persistence. This is crucial for organizations to effectively detect and respond to security incidents related to Tor usage, strengthening their overall security measures.

2.3. Limitations of the Literature with Respect to Windows 10

Despite recent research, there are a number of issues in the forensic investigation of the Tor Browser, including the following:

- The previous studies performed Tor forensics but did not cover all forensic techniques (network, storage, memory and registry forensics).

- The previous studies used the most recent version of the Tor Browser and Windows OS of that time, but those versions are now considered older [4,8,21].
- The previous studies provided no automation for the gathering of artifacts from any forensic technique in a time-effective way.
- Previous studies did not cover artifacts that persist after the browser is deleted [11,16,19].

In most research, the computer systems vary in their RAM and storage capacities, and it is essential to reconstruct experimentation, considering that the results of one experiment may not be directly transferable to another. This discrepancy can become particularly pronounced over time as technology evolves, rendering what is novel today as outdated in the future.

2.4. Limitations in the Literature with Respect to Windows 11

Despite the extensive literature available for Windows 10, several factors can justify the need for updated research and forensic methodologies as users migrate to Windows 11; forensic investigators will increasingly encounter this operating system in their work. It is important to understand the distinctions of Tor Browser forensics on the latest platform to stay ahead of potential challenges, including the following:

- Windows 11 introduces a new file system called "WinFS", which may have implications for digital forensics. WinFS was designed to provide a unified storage system for both local and networked data. This potentially impacts the way data are stored and accessed, requiring forensic analysts to adapt new methodologies for storage forensics. Also, Windows 11 includes enhanced security features, such as secure core and hardware-based virtualization-based security (VBS). These features impact the ability to access and analyze data related to the Tor Browser, potentially making it more challenging to acquire certain forensic artifacts from RAM and storage.
- Before the introduction of the "WinFS" file system in Windows 11, the primary file system used in Windows operating systems was the New Technology File System (NTFS). The NTFS has been the default file system for Windows since the release of Windows NT 3.1 in 1993. WinFS has stronger encryption than NTFS, which makes the deleted data extraction more complex than Windows 10.
- The Windows 11 System32\Config folder is of 1.8 GB to 2.4 GB and contains about 400k to 500k registries, whereas the Windows 10 System32\Config folder is of 1.5 GB to 2.2 GB and contains about 300k to 400k registries. This makes the registry forensic job more thorough.
- Windows 11 needs more RAM for usage; hence, the memory dump will be much bigger for this. Hence, some more high-end hardware in the system must be used for the analysis of memory artifacts.
- The forensic community must adapt to the latest technologies to remain effective. As operating systems evolve, so too must the tools and techniques used by forensic investigators evolve. By staying current with the latest platform, investigators ensure they are prepared to handle cases involving the most recent technologies, which is essential for maintaining the integrity and reliability of forensic investigations.

3. Proposed Methodology

The proposed methodology uses two distinct forensic acquisition approaches: (1) live acquisition, primarily for network and memory forensics, and (2) static acquisition, specifically for registry and storage forensics. To simulate a real-world scenario of exploring the dark web on a Windows 11 system using the Tor Browser, we must precisely acquire the system under legal forensic standards. After that, we will use data forensic tools to inspect and excavate any traces of the Tor Browser.

The proposed methodology integrates forensic techniques into incident response according to NIST SP (800-86) and explores fundamental areas including registry, storage, network, and memory. The NIST SP 800-86 is a guide published by the National Institute of Standards and Technology (NIST) that provides practical guidance on integrating forensic

techniques into incident response [36]. This guide is intended to assist organizations in investigating computer security incidents and troubleshooting some information technology (IT) operational problems. It presents forensics from an IT view, not a law enforcement view [37].

The guide covers various data sources, including files, operating systems, network traffic, and applications. It describes a systematic approach to incident response, emphasizing the need for organizations to establish incident response capabilities that align with their specific needs. It also outlines a four-step process for applying digital forensic techniques in a consistent manner: collection, examination, analysis, and reporting. Our methodology is based on this standard and follows these guidelines and processes, integrating forensic techniques into the incident response process. By following the NIST SP 800-86, the methodology can provide a more comprehensive and effective response to security incidents. It can help identify the cause of an incident, limit the damage, and prevent similar incidents in the future

3.1. Experimental Setup

For the experimentation, we set up a fresh Windows 11 local machine, or we could also use a virtual machine with the installation of the Windows operating system (Windows 11 Pro version 22H2) and storage of data to work in a clean environment to investigate the pieces of evidence from the network, memory, storage, and artifacts from registry hives.

The following tools were used in this research to aid in the investigation. The operating system used was Windows 11 Pro version 22H2, while additional tools included Tor Browser version 12.0 for anonymous browsing, Autopsy 4.21.0 for digital forensics, TestDisk 7.2 for data recovery, Volatility Framework for memory analysis, RawCap 0.2.10 for network traffic capture, NetworkMiner 2.7.3 for network protocol analysis, Regshot 64-bit 1.9.0 for registry analysis, SQLite 64-bit 3.12.2 for database analysis, Bulk Extractor 2.0.0 for bulk data extraction, and FTK Imager 4.7.1 for disk imaging and analysis.

3.2. ToolsUsed

The following tools were used to aid in the investigation:

1. **FTK Imager:** FTK Imager was chosen for its efficiency in capturing memory dumps, a crucial aspect of forensic analysis. Its user-friendly interface and strong memory-capturing capabilities make it an excellent tool for this task.
2. **Regshot and Regedit:** Regshot is used to compare registry snapshots before and after using the Tor Browser, with Regedit being employed to view registry artifacts. Regshot is efficient in generating and comparing registry snapshots, which makes it an ideal choice for identifying changes after using the Tor Browser. Regedit complements this by allowing the detailed inspection of registry artifacts.
3. **RawCap:** RawCap is employed to capture network traffic. RawCap is a lightweight command-line tool designed specifically for capturing raw network traffic. It captures packets at a low level without parsing them. It preserves the entire content of each packet, including headers and payload, without any alteration, preserving its raw format. It is lightweight and is fully compatible with Windows-based systems.
4. **NetworkMiner:** NetworkMiner is an open-source tool used to extract various data types from PCAP files. NetworkMiner's versatility in extracting diverse data types from PCAP files, coupled with its user-friendly interface, makes it valuable for forensic analysis in Windows environments. Its open-source nature allows users to inspect and modify the tool according to their requirements. Additionally, being free makes it accessible to a broad audience.
5. **Volatility:** Volatility is a popular open-source memory forensic framework that is widely used by digital forensic investigators and incident responders to analyze memory dumps, which are images of a computer's memory at a specific point in time. Volatility's primary function is to extract and analyze the running processes,

modules, and other system information from a memory dump. It does this using various plugins, which are essentially small programs that perform specific tasks.

6. **Bulk Extractor:** Bulk Extractor was selected to efficiently extract various artifacts from memory dumps. It was explicitly designed for artifact-focused extraction from memory dumps. Bulk Extractor allows for targeted keyword searches, enabling the focused extraction of specific artifacts. Bulk Extractor's open-source nature allows for customization and collaboration, enhancing its effectiveness in forensic investigations.
7. **TestDisk and Autopsy:** TestDisk and Autopsy were chosen for storage analysis and recovering some data artifacts. It excels in recovering lost partitions, while Autopsy provides a comprehensive platform for digital forensics. Both of them are compatible with Windows, and their open-source nature contributes to their selection.

3.3. Systematic Sequence of Activities

The methodology for analyzing the system after simulating Tor Browser usage involves a systematic sequence of activities. Figure 3 provides a graphical representation of the proposed methodology that is followed during forensic analysis. It visualizes a variety of acquisition techniques and steps used to acquire artifacts.

This strategic approach ensures a detailed and layered examination of the Windows 11 operating system in the context of dark web exploration using the Tor Browser. The combination of live and static acquisition techniques enhances the ability to capture a complete view of the system's forensic landscape, providing valuable data for in-depth analysis and insights.

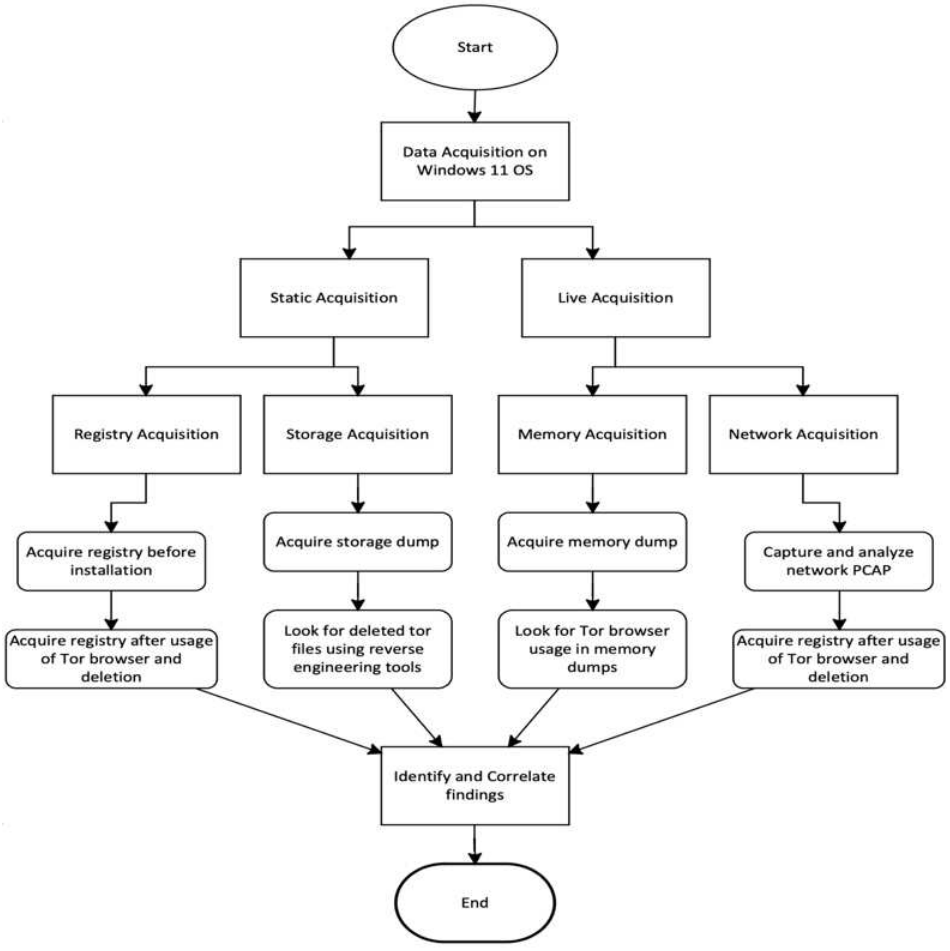

Figure 3. Flow diagram for the proposed methodology used to analyze Tor Browser artifacts.

3.4. Data Collection Procedure

The data acquisition process involves three key steps, each focusing on a specific area to provide a comprehensive understanding of Tor Browser usage. The flow diagram in Figure 3 outlines a systematic approach for conducting a forensic analysis of the Tor Browser. It begins by capturing an initial snapshot of the system registry to create a baseline. Next, the Tor Browser is installed and executed, followed by capturing network packets to monitor communication. After establishing a connection with the Tor network and simulating user activities, the Tor Browser is closed and uninstalled. A memory dump and a second registry snapshot are then taken to compare with the initial baseline. Storage is also analyzed, and forensic techniques are applied across various domains, including registry, network, memory, and storage forensics. The process concludes by identifying and extracting evidence of Tor Browser usage and correlating findings to provide a comprehensive understanding, which is then documented.

The first step involves taking a snapshot of the registry before and after Tor Browser usage, enabling a comparative analysis of registry changes. This live acquisition provides valuable insights into the system's registry modifications caused by the Tor Browser.

The second step involves acquiring memory dumps and network PCAPs during the live phase, offering real-time insights into system activities and network interactions related to Tor Browser usage. This step provides a detailed understanding of the system's behavior during Tor Browser usage.

The third step involves examining stored images for registry and storage forensics, specifically after Tor Browser usage and deletion. This static acquisition provides a comprehensive approach to identifying residual traces left behind by the Tor Browser, even after deletion. By analyzing these stored images, investigators can gather evidence of Tor Browser usage and reconstruct the user's activities. The steps are outlined below:

1. Capture an initial snapshot of the system registry to establish a baseline.
2. Install the Tor Browser on the system.
3. Execute the Tor Browser to initiate its functionalities.
4. Start capturing network packets to monitor communication.
5. Establish a connection with the Tor network to simulate dark web usage.
6. Interact with the Tor Browser to simulate user activities.
7. Close the Tor Browser after usage.
8. Delete both the Tor Browser installer and the application.
9. Capture a memory dump of the system post-Tor Browser usage.
10. Capture a second snapshot of the system registry for comparison.
11. Conduct a comparative analysis of the two system registry snapshots.
12. Capture a storage dump of the system for detailed analysis.
13. Employ forensic techniques across domains: registry, network, memory, and storage forensics.
14. Identify and extract evidence related to Tor Browser usage, including browsing history, registry entries, and application locations.
15. Establish correlations between different forensic artifacts for a holistic understanding.
16. Compile and document the results obtained from the forensic analysis.

By following this methodology, forensic investigators can uncover and correlate evidence related to Tor Browser usage, answering the first research question and contributing to a comprehensive forensic report. In the initial steps, we initiate a live acquisition process, capturing snapshots of the system's registry both before and after the Tor Browser is employed and subsequently uninstalled. This strategic comparison allows us to discriminate any alterations or artifacts introduced during the browser's usage, offering valuable insights into the system's registry entries. The later stages involve the analysis of memory dumps and network packet capture (PCAP) files during the live phase of data acquisition.

This dynamic approach enables us to examine real-time activities and interactions associated with the presence of the Tor Browser in a Windows 11 environment. By investi-

gating through memory and network forensics, we aim to find sensitive details that might not be easy to acquire through static analyses. After this, we perform static acquisition for registry and storage forensics. We carefully examine the stored images of the system's storage, particularly after the Tor Browser has been used and then deleted. This detailed examination is crucial for identifying any residual artifacts, traces, or alterations in the storage structure resulting from the browser's activities.

4. Forensic Analysis of Tor Browser Activity on Windows

This section describes the process of accessing the dark web using the Tor Browser on Windows, intending to identify artifacts remaining in the system's memory, storage, and registry. The analysis also captures a PCAP file during a simulated scenario.

To begin, a Windows 11 virtual machine (VM) was created, and a registry snapshot was taken. The Tor Browser was installed and executed, allowing dark web users' behavior to be monitored and captured in the form of a PCAP file by surfing multiple websites with the Tor Browser. Before simulation, a packet sniffing tool (Rawcap) was activated to capture data.

Once the simulation was complete, a memory dump and disk image were obtained. After deleting the Tor Browser, another registry snapshot was taken and compared with the previous one to identify any changes.

In this analysis, two types of forensic acquisitions were performed: live acquisition for network and memory forensics, and static acquisition for registry and storage forensics. The scenario involves simulating surfing the dark web on Windows 11 using the Tor Browser.

The data acquisition procedure involves capturing memory images after using and deleting the Tor Browser and analyzing memory dumps for artifacts related to Tor Browser usage. Registry snapshots were taken before and after the installation, use, and deletion of the Tor Browser, and compared to identify any changes. Storage images were analyzed using reverse engineering tools to look for evidence of Tor Browser installation, use, and uninstallation.

The captured PCAP file was investigated using Network Miner to identify any browsing-related artifacts in the network packets. Finally, all acquired data (registry snapshots, memory images, PCAP files, and disk images) were dumped into an external drive for examination during forensic analysis.

4.1. Forensic Techniques

This section outlines the forensic techniques used to gather evidence of Tor Browser activity and identify related artifacts. The investigation employed a combination of memory forensics, network forensics, and registry forensics to collect and analyze evidence.

4.1.1. Memory Forensics

Memory imaging was conducted using FTK Imager to capture snapshots of the system's memory at three critical stages: when the Tor Browser is open, closed, and uninstalled. The acquired memory dumps were then analyzed using various techniques to identify relevant digital traces.

The results of this analysis are presented in Figure 4, which illustrates the identified onion URLs retrieved from the memory dumps. To extract this specific information, a keyword search was conducted using HxD, a hexadecimal editor. This process involved searching for particular text strings or phrases that are indicative of Tor Browser usage. Through carefully analyzing the memory dumps, these searches allowed for the identification of relevant data, which were then compiled and presented in the figure.

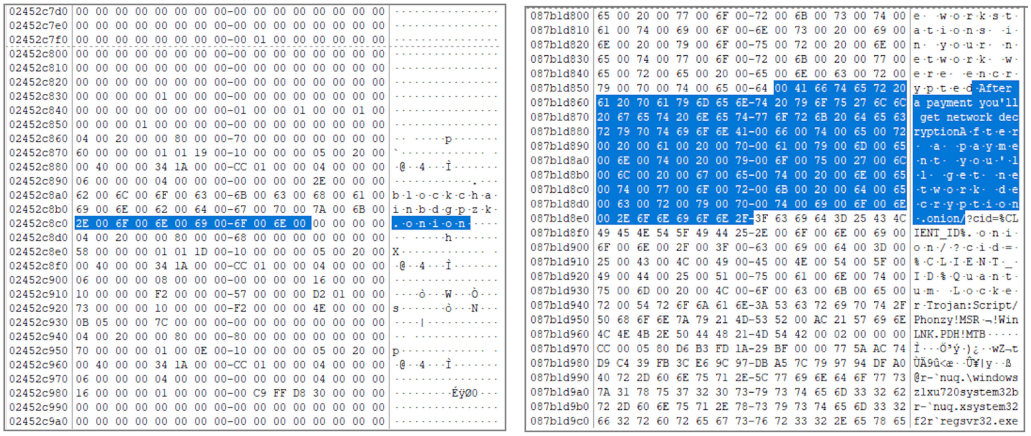

Figure 4. Results and onion URLs found from memory dumps. Dumps of onion links are highlighted in blue color.

Additionally, a Bulk Extractor was used to gather evidence from the memory dump. This tool extracts various artifacts, including the visited sites, Tor launcher, Tor installer, running processes at specific times, and other relevant data. The extracted artifacts are organized into folders within a specified directory. The resulting files, such as the 'domain.txt' file, can be easily viewed to identify visited websites.

A summary of the extracted artifacts is presented in Table 2. The comprehensive approach employed in memory forensics enables a thorough examination of memory data, allowing for the identification of crucial evidence related to Tor Browser usage and its associated activities. By analyzing the memory dumps at different stages (open, closed, and uninstalled), a detailed picture of Tor Browser activity can be reconstructed, including information on visited onion sites, the presence of Tor.exe, and Tor Browser launcher timestamps.

Table 2. Results obtained after memory forensics.

Tor Browser Stage	Tool Used	Visited Sites	tor.Exe	Tor Launcher	Timestamps	Registry Values
Open	FTK Imager + Bulk Extractor	Yes	Yes	Yes	Yes	Yes
Closed	FTK Imager + Bulk Extractor	Yes	Yes	Yes	Yes	No
Uninstalled	FTK Imager + Bulk Extractor	Yes	Yes	Yes	Yes	No

4.1.2. Registry Forensics

Registry forensics is a crucial step in identifying Tor Browser artifacts, even after the browser has been deleted. As shown in Figure 5, several registry entries remain present after Tor's removal, providing evidence of its usage. These artifacts can serve as key evidence in forensic investigations, as they often contain information related to the installation paths, user preferences, and recent activity associated with the Tor Browser.

Table 3 summarizes the artifacts discovered through registry analysis, with the first three entries obtained before running the Tor Browser and the remaining entries detected after its use. Notably, these registry entries remain persistent even after the Tor Browser is uninstalled, indicating that they are not temporary or volatile. This persistence suggests that even after attempts to remove traces of the Tor browser, valuable forensic data remain accessible, offering a window into past user activities. The analysis revealed a total of 42 registry entries, with 8 significant ones highlighted in Table 3.

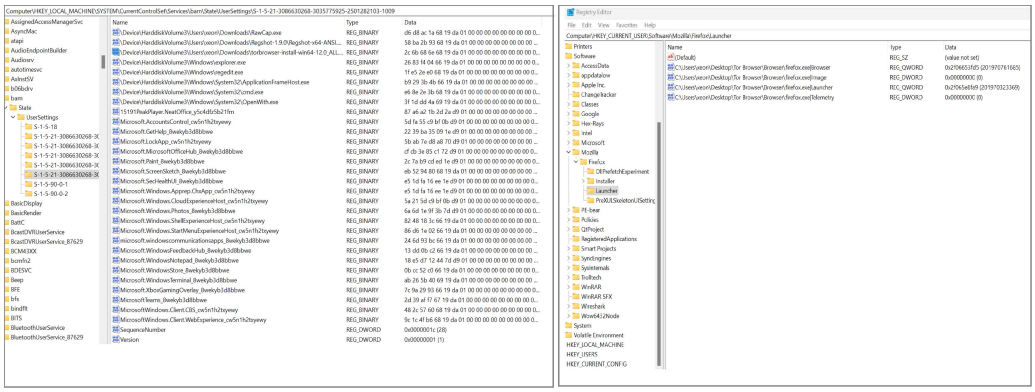

Figure 5. Tor Browser installer and artifacts left behind in Windows registry.

Table 3. Artifacts discovered through registry analysis.

No	Description	Locations of Registries Belong to Tor Browser
1	Tor Browser Installer	Computer\HKEY_LOCAL_MACHINE\SYSTEM\Current ControlSet\Services\bam\State\UserSettings\S-1-5-21-3086630268-3035775925-2501282103-1009
2	Location of Tor Installer	Computer\HKEY_USERS\S-1-5-21-3086630268-3035775925-2501282103-1009\Software\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Compatibility Assistant\Store
3	Name of application	Computer\HKEY_USERS\S-1-5-21-3086630268-3035775925-2501282103-1009\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\MuiCache
4	Firefox location	Computer\HKEY_LOCAL_MACHINE\SYSTEM\Current ControlSet\Services\bam\State\UserSettings\S-1-5-21-3086630268-3035775925-2501282103-1009
5	Application user-friendly name	Computer\HKEY_USERS\S-1-5-21-3086630268-3035775925-2501282103-1009\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\MuiCache
6	Company name of application	Computer\HKEY_USERS\S-1-5-21-3086630268-3035775925-2501282103-1009\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\MuiCache
7	Some traces of Tor Browser	Computer\HKEY_USERS\S-1-5-21-3086630268-3035775925-2501282103-1009\Software\Mozilla\Firefox\Launcher
8	Firefox installer location	Computer\HKEY_LOCAL_MACHINE\SYSTEM\Current ControlSet\Services\bam\State\UserSettings\S-1-5-21-3086630268-3035775925-2501282103-1009

These registry entries provide valuable information about Tor's activity, including the installation and uninstallation processes, as well as any configuration settings or preferences made during its usage. By examining these entries, forensic investigators can reconstruct the timeline of Tor Browser activity and identify potential evidence of user behavior.

4.1.3. Storage Forensics

The process of obtaining a disk image is facilitated by FTK Imager. In cases where Windows is installed within a VirtualBox environment, a disk image in the raw image format (.img) is created using VirtualBox, which is compatible with Autopsy and TestDisk for forensic analysis. After deleting the Tor Browser, this study attempted to recover any artifacts related to its presence in storage.

The locations of tor.exe, firefox.exe, and the Tor installation file can be easily identified. Autopsy also reveals these programs in recently run programs. All downloaded files are available in the deleted files folder. The Tor installer location was found using Autopsy, as shown in Figure 6. Autopsy, a digital forensic tool, allows for the recovery of deleted files and helped pinpoint the exact location where the Tor Browser was installed.

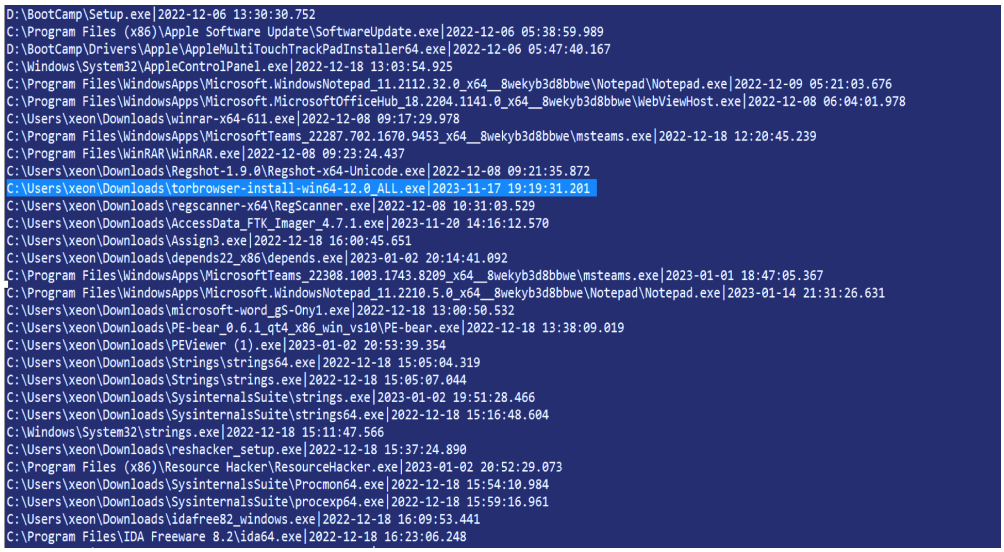

Figure 6. Tor Browser installed location found in the data recovery process. The installer location is highlighted in blue color.

Tor has this file in the profile folder "C:\Users\USERNAME\Desktop\TorBrowser\Data\Browser\profile.default". The tool SQLite viewer was used to view the default file even after the Tor Browser was deleted. SQLite Viewer provides access to database files that contain valuable records of Tor Browser activity, which remain intact despite the browser's removal. Some data about Tor usage could also be found in the prefetch file shown in Figure 7. Prefetch files, which are used by the operating system to speed up application launch times, also retained traces of the Tor Browser, offering another layer of evidence.

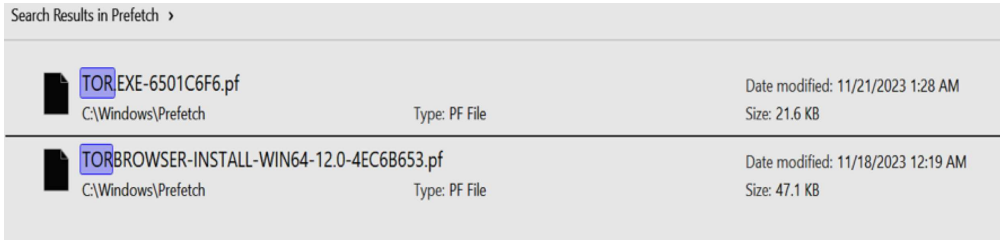

Figure 7. Tor artifacts in prefetch.

The artifacts obtained through storage forensics includes information such as the location of the Tor executable file, Firefox executable file, Tor process ID, Firefox process ID, and registry entries and other relevant files associated with the browser's operation. These artifacts provide a comprehensive view of the data remnants left behind by Tor, contributing significantly to the forensic analysis. These artifacts were obtained using TestDisk and Autopsy, both powerful and open-source tools.

4.1.4. Network Forensics

Due to the inherent characteristics of the Tor Browser, conventional traffic analysis techniques are limited in their effectiveness. This is because an eavesdropper can only detect network traffic originating from a Tor user, without being able to observe the complete circuit used to reach the intended destination. The purpose of this case study was not to launch an offensive against the Tor network or uncover the user's browsing activities, but rather to assess the feasibility of detecting Tor browsing activities in a censored environment where the use of anonymous browsers like Tor is prohibited.

To achieve this goal, we employed Rawcap, a packet sniffer that can handle the covert and stealthy features of the Tor Browser. The acquired PCAP files were then analyzed using Network Miner to extract relevant information.

The examination of the PCAP files revealed several interesting artifacts, including the host IP address (illustrated in Figure 8), which is crucial in identifying the user's location and potentially compromised devices. The image provides valuable insight into the network traffic patterns and helps investigators understand the extent of Tor usage.

Furthermore, we discovered user credentials in the PCAP file, as depicted in Figure 9. These credentials are essential in identifying the user's identity and understanding their online activities. The image highlights the importance of protecting user credentials and emphasizes the need for robust security measures.

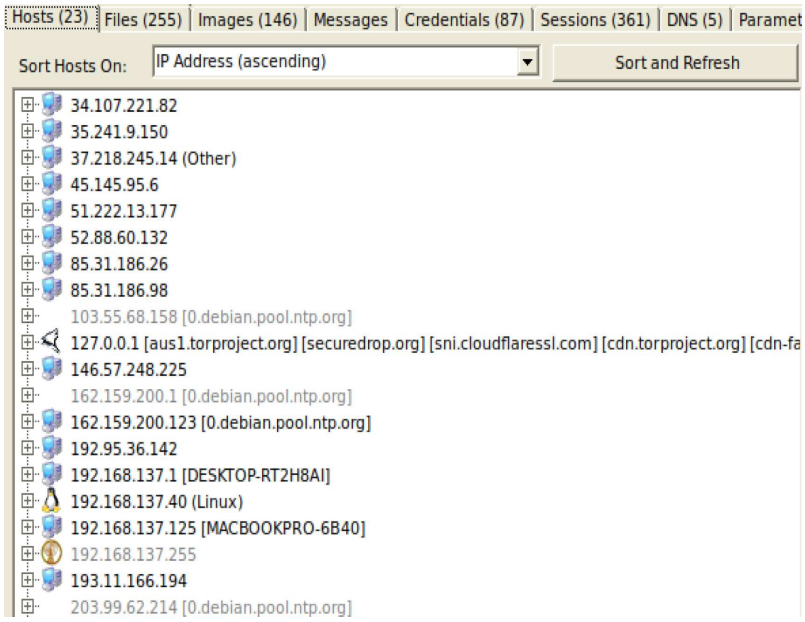

Figure 8. Hosts from PCAP in Network Miner.

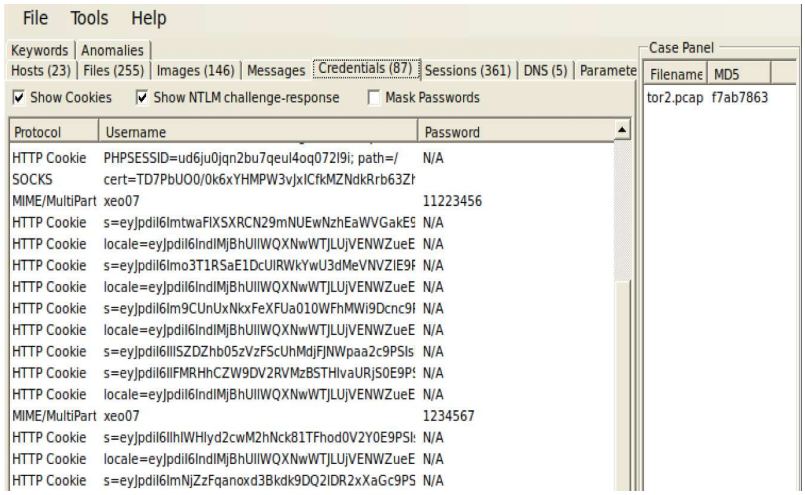

Figure 9. Credentials found using Network Miner.

Additionally, some parameters showed SOCKS proxy connections related to Tor Browser connections with dark web websites. Table 4 presents the results obtained after network forensics, which include information about all visited onion websites, the user's host IP address, user credentials in plain text, images from visited websites, and usage timestamps. These comprehensive data provide valuable insights into Tor Browser usage patterns and helps investigators to better understand online activities.

Table 4. Results obtained after network forensics.

No.	Artefact	Yes/No
1.	All onion websites	Yes
2.	Credentials	Yes
3.	Website's images (downloaded and visited images)	Yes
4.	Timestamps	Yes

4.2. Timeline Analysis for Artifact Correlation

In memory forensics, which involves the analysis of a system's volatile memory, and network forensics, which scrutinizes network traffic and communication patterns, timestamps play a crucial role. These timestamps mark specific points in time when actions or events occurred, creating a temporal framework for investigation.

By aligning the timestamps from memory and network artifacts on a unified timeline, forensic analysts can draw connections and reveal patterns of activity. For instance, identifying a process or code injection in memory at a particular timestamp may coincide with network traffic indicating communication with a suspicious external entity. This correlation enhances the investigative process by providing a contextual understanding of how events unfold across both memory and network domains.

Figure 10 illustrates the importance of timeline analysis by displaying running processes on the x-axis and their up-time on the y-axis (in minutes). The figure highlights the significance of visualizing process creation times and up-times to identify patterns or clusters that may be indicative of malicious activity.

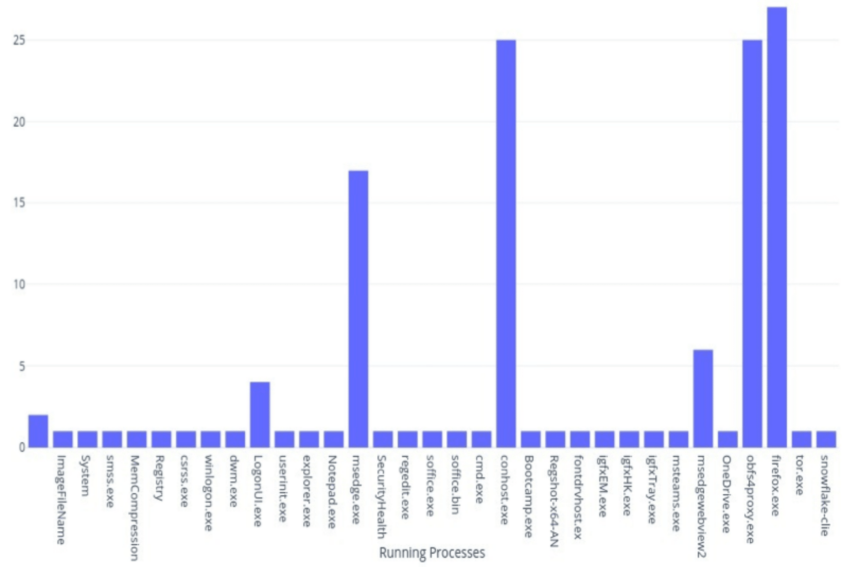

Figure 10. An overview of different running processes.

The memory dump was analyzed, and artifacts about Tor or Firefox usage were extracted from the memory image. The corresponding timestamps were also obtained for correlation. Figure 11 provides a detailed view of the timeline, where the x-axis represents the time of process creation, and the y-axis indexes each process occurrence. Each point on the timeline represents the creation of a process, with 'tor.exe' and 'firefox.exe' differentiated by markers. This visualization enables analysts to quickly identify patterns or clusters in process creation times, which can be useful for further analysis.

Timestamps are also acquired from network PCAP files, which show URL access times. Figure 12 illustrates the onion URLs on the y-axis and timestamps on the x-axis. The figure

reveals that on 20 November 2023, at 13:10:00, the Tor Browser was installed on the system, followed by its launch before 13:20:00 and usage up to 14:10:00.

The timestamps from PCAP also show activity during this period, specifically between 13:39:00 and 13:43:00, during which some onion URLs were visited. By analyzing these timelines together, investigators can gain a comprehensive understanding of Tor Browser usage patterns and identify potential indicators of malicious activity.

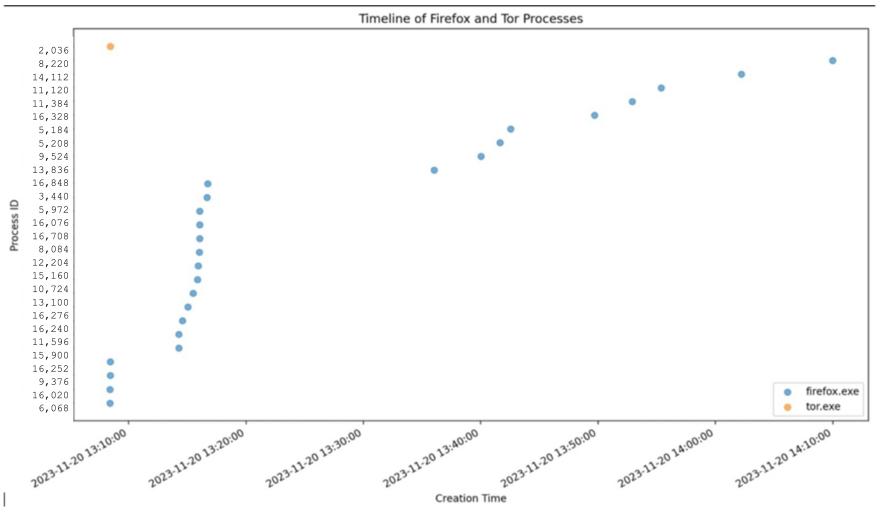

Figure 11. Tor usage timeline detected during analysis.

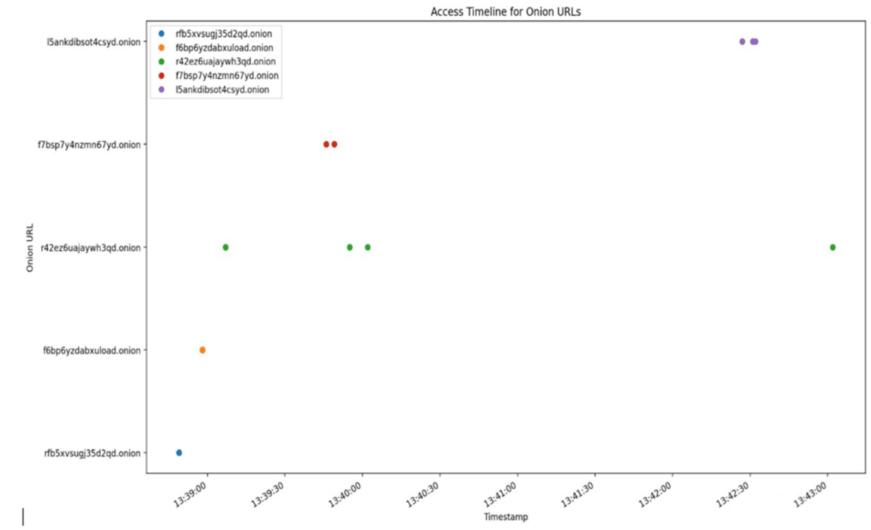

Figure 12. Timestamps of visited onion sites retrieved during the analysis.

5. Automated Script for Tor Artifact Retrieval

This section describes the development of a PowerShell script to retrieve Tor-related artifacts from the system registry of a suspect device. The script determines whether Tor usage is detected on a machine or not. The script flow diagram, illustrated in Figure 13, outlines the key activities of the automated script, demonstrating the sequence of actions performed to retrieve Tor artifacts from system registries and indicate whether they are present. The script performs the following key activities:

1. **Registry Query for Firefox Launcher:** The script queries the Windows Registry to check for the existence of a specific location related to the Firefox Launcher.
2. **Defined Registry Keys and Locations:** Several registry keys and their expected locations are defined, primarily focusing on paths associated with the Tor Browser.

3. **Registry Key Existence Checks:** A function (**Check-RegistryKey**) is created to check if specified registry keys exist at designated locations. The script then employs this function to verify the presence of predefined registry keys.
4. **Additional Registry Locations Checked:** The script investigates additional registry locations associated with Mozilla, Firefox, and the Tor Project.
5. **Elevated Permissions Check:** It verifies whether the script is running with elevated (administrator) permissions and attempts to relaunch with elevated privileges if necessary.
6. **Detection of Tor Usage:** The script notifies the user if it detects any artifacts related to Tor, providing information on the number of Tor-related registry entries found.
7. **Query and Analysis of Registry Values:** The script retrieves and analyzes all registry values in a specified location, specifically searching for values containing the strings 'Tor' or 'tor'.
8. **Script Completion Message:** A completion message is displayed at the end of the script execution.

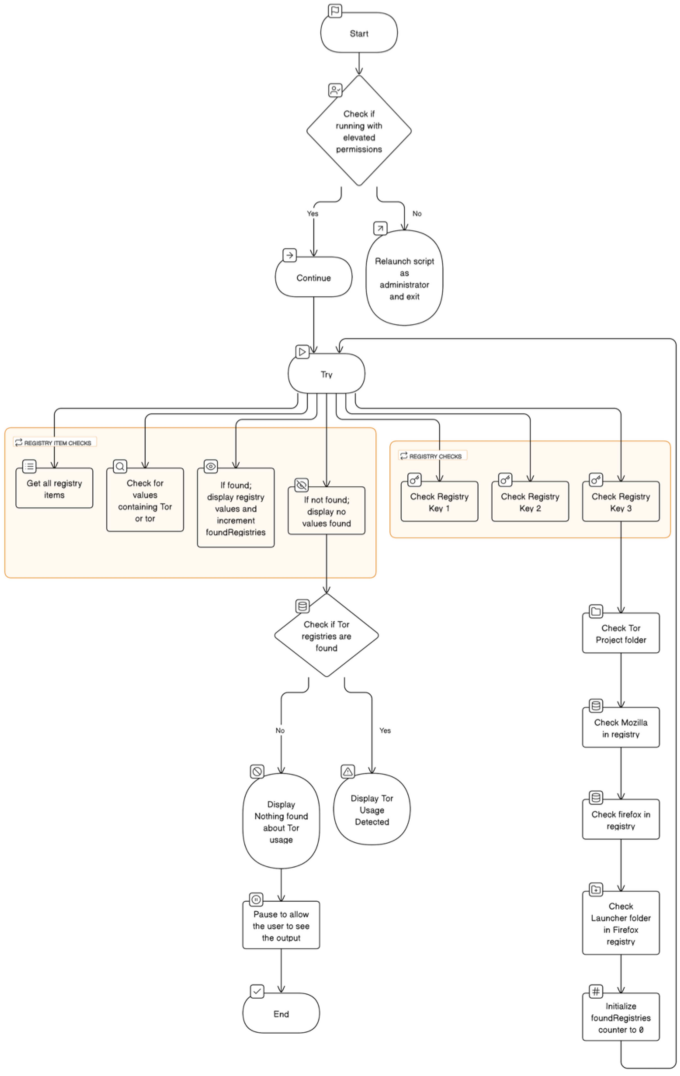

Figure 13. Flow diagram for proposed automated script, which can be used to retrieve artifacts from Tor Browser in Windows environment.

The script was designed to conduct a forensic analysis on a Windows system by examining the Windows Registry for artifacts related to the Tor Browser, Mozilla Firefox, and the Tor Project. When activated, the script follows a series of steps: first, it checks if it is running with elevated permissions to ensure it has sufficient privileges to access the registry.

Then, it searches for registry items related to the Tor Browser, including values, folders, and other artifacts. The script collects and stores the identified artifacts in a centralized location for further analysis. Finally, it analyzes the collected artifacts to determine whether Tor usage is detected on the machine.

Algorithm 1 for the PowerShell script designed to retrieve Tor artifacts from a Windows system registry involves several key steps followed to efficiently and reliably identify evidence of Tor Browser usage. Initially, the script imports necessary libraries such as PSReadline for enhanced command-line functionality and Microsoft.PowerShell.Security for managing security features, ensuring that the environment is equipped with the required tools for registry access and manipulation. The script then verifies that it is being run with elevated permissions (i.e., as an administrator) because accessing certain registry keys requires administrative rights. If the script does not have these permissions, it outputs an error message and terminates.

Algorithm 1 PowerShell Script for Tor Artifact Retrieval

Require: Windows system with PowerShell, elevated permissions (Administrator), access to system registry
Ensure: Artifact collection and analysis for Tor Browser usage

```
procedure RETRIEVETORARTIFACTS
    Import Libraries and Modules
    Import-Module -Name 'PSReadline'
    Import-Module -Name 'Microsoft.PowerShell.Security'
    Check for Elevated Permissions
    $isElevated = (New-Object Security.Principal.WindowsPrincipal
([Security.Principal.  WindowsIdentity]::GetCurrent())).
IsInRole([Security.Principal.  WindowsBuiltInRole]::Administrator)
    if !isElevated then
        Write-Error 'Script needs to be run as an administrator.'
        return
    end if
    Define Registry Paths for Tor Artifacts
    $registryPaths = @('HKCU:\Software\Tor Browser',
'HKCU:\Software\Mozilla\Firefox\TorBrowserData')
    Initialize Collection
    $artifacts = @()
    Search for Tor Artifacts in Registry
    for each $path in $registryPaths do
        if Test-Path $path then
            $artifacts += Get-ItemProperty -Path $path
        end if
    end for
    Store Artifacts in Centralized Location
    $outputPath = 'C:\Forensics\TorArtifacts.txt'
    $artifacts | Out-File -FilePath $outputPath
    Analyze Artifacts for Tor Usage
    if $artifacts.Count -gt 0 then
        Write-Output 'Tor usage detected on the machine.'
    else
        Write-Output 'No Tor usage detected on the machine.'
    end if
    Notify Forensic Investigator
    Send-MailMessage -To 'investigator@example.com' -From
'forensictool@example.com' -Subject 'Tor Artifact Analysis Results'
-Body 'Please check the attached artifact file.'  -Attachments
$outputPath
end procedure
```

5.1. Pseudocode of PS Script

1. **Print "Executing script. . . "**
2. **Define registry locations and keys to be checked:**
   - Registry location for Tor installer and usage data.
   - Keys related to Tor and Firefox executables.
3. **Initialize a counter for found registry entries.**
4. **Define a function to check if a registry key exists:**
   - If key exists, print confirmation and increment the counter.
   - If key does not exist, print a message indicating its absence.
5. **Check for the presence of defined registry keys using the function.**
6. **Run a command to query a specific registry location.**
7. **Check if certain registry paths or entries exist:**
   - If found, print confirmation and increment the counter.
   - If not found, print a message indicating its absence.
8. **Check if the script has elevated (administrator) permissions:** If not, restart the script with elevated permissions.
9. **Check if any Tor-related registry entries were found:**
   - If found, print "Tor Usage Detected" and the number of findings.
   - If none found, print "Nothing found about Tor usage".
10. **Try to obtain all registry items from the specified location:**
    - If successful, check for registry values containing "Tor".
    - If found, print and increment the counter.
    - If an error occurs, print an error message.
11. **Pause to allow the user to see output, then continue.**
12. **Print "Script completed".**

Next, the algorithm defines specific registry paths that are known to contain artifacts related to Tor Browser and potentially related applications such as Mozilla Firefox. These paths are stored in a list and will be used for searching. The script initializes an empty collection to store any discovered artifacts.

The script then iterates through the predefined registry paths, checking if each path exists in the system registry. If a path is found, the script retrieves the properties (artifacts) associated with that path and adds them to the collection. Once all specified paths have been checked, the collected artifacts are saved to a centralized location, specifically to a file in the *C: Forensics directory*, facilitating easy access for further analysis.

After storing the artifacts, the script evaluates the collected data to determine whether any artifacts indicative of Tor usage are present. If any artifacts are found, the script concludes that Tor usage is detected and outputs a corresponding message. Conversely, if no artifacts are found, the script indicates that no Tor usage has been detected on the machine. Finally, the script sends a notification email to a forensic investigator, attaching the file containing the artifacts.

The script provides a notification indicating whether Tor usage is detected on the machine or not. This output is essential for forensic investigators to quickly identify potential evidence of Tor browser usage on a Windows system.

5.2. Script Usage

The intended use of this script is for digital forensics or system analysis to identify traces of Tor Browser usage on a Windows system. Researchers can leverage this script to automate the process of examining relevant registry entries, providing insights into potential user activity involving Tor and Firefox. It serves as a tool for security analysts and researchers who are interested in investigating evidence of privacy-focused browser

usage on Windows systems. Figure 14 shows the results obtained through the retrieval of registries from the script.

The developed PowerShell script represents a noticeable and practical outcome of this research, offering a valuable tool for forensic analysts and researchers engaging in the dynamic domain of digital investigations. The considerations that highlight the script's significance in the context of this research on Tor browser forensics are as follows.

Automation for Enhanced Efficiency: The developed script automates the process of querying and analyzing registry entries related to the Tor Browser and Firefox. This automation streamlines investigative workflows, allowing forensic analysts to expedite the identification of pertinent artifacts.

Standardized and Reproducible Analysis: By providing a standardized and reproducible method for registry analysis, the script contributes to the establishment of a consistent framework for forensic examinations. This consistency is crucial for reliability and comparability in digital investigations.

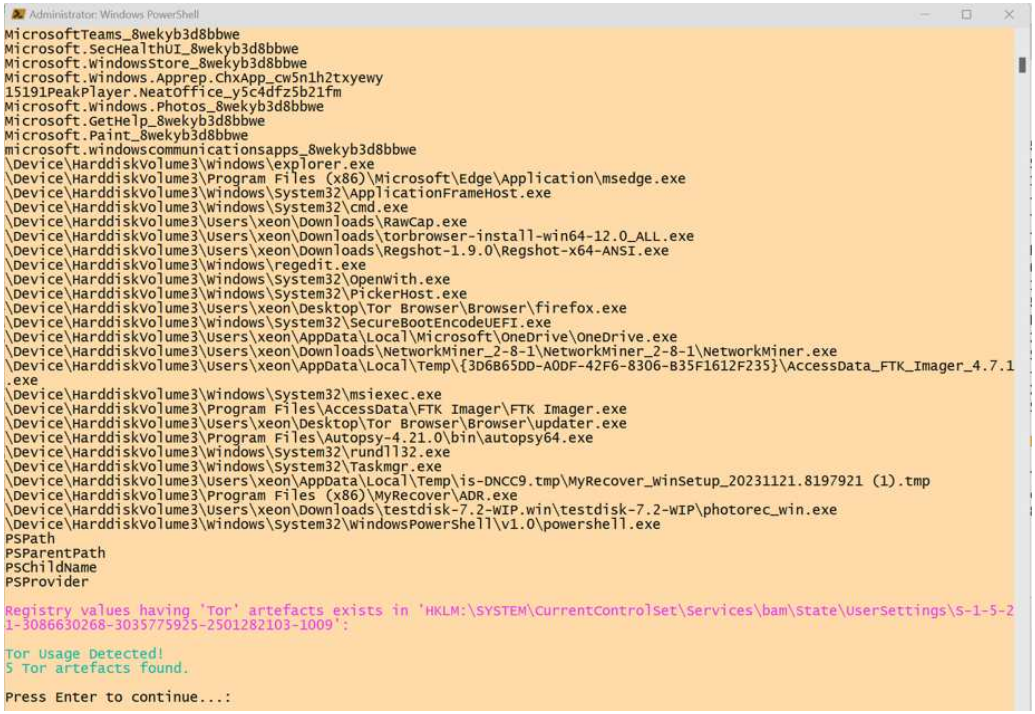

Figure 14. Results obtained after executing the developed script.

Integration into Comprehensive Methodology: The script seamlessly integrates into the broader methodology outlined in this research. Its inclusion adds a layer of technical sophistication to the investigative toolkit, reinforcing the depth and comprehensiveness of the proposed approach to Tor Browser Forensics.

Complementing Manual Analysis: While manual analysis remains fundamental, the developed script serves as a complementary tool, augmenting the capabilities of forensic analysts. The combination of manual inspection and automated registry analysis ensures a more thorough exploration of potential artifacts.

Adaptability to Windows 11: As this research lays the groundwork for an in-depth exploration of Tor Browser Forensics in the Windows 11 operating system, the developed script is designed to be adaptable to the evolving technological landscape. This adaptability positions it as a valuable resource for future investigations on Windows 11.

Contribution to Ethical Framework: The script aligns with the ethical considerations outlined in this research, ensuring that digital forensic practices adhere to ethical standards. Its transparent and controlled approach to registry analysis reflects its commitment to responsible and principled investigative methodologies.

6. Results and Discussion

This section provides results and some discussion on the obtained artifacts from the network, storage, memory, and registry analysis. The main results were obtained during the comprehensive analysis process, and evidently, artifacts about Tor Browser usage across all stages were successfully gathered. The stage at which each artifact was obtained is well documented, as well as the specific tools employed for their collection. In particular, the artifacts acquired after the browser's closure or uninstallation stage offer compelling substantiation of Tor Browser usage and enable the exposure of the Tor Browser's presence in the system. These findings hold significant weight in the realm of information security, providing intricate technical insights into user behavior and demonstrating the extensive forensic capabilities employed in uncovering such evidence.

Conducting memory forensics on a system where the Tor Browser was utilized involves analyzing the contents of the system's RAM (Random Access Memory) to identify and extract pertinent artifacts linked to Tor usage. The following outlines the significant findings observed during the memory forensic process related to the Tor Browser. We presented the identified activities, along with the corresponding analysis stages and the tools utilized in the process. The following are some identified artifacts from memory forensics: (1) visited onion sites, (2) HTTP header information, and (3) Tor launcher.

In this research, a thorough examination of the Tor Browser across three distinct stages was executed, focusing on registry forensics. Before the Tor Browser installation, we conducted an in-depth examination of the system registry. Remarkably, relevant registry entries, even if the browser had been deleted, were successfully identified, providing valuable artifacts indicative of the presence of the Tor Browser. We provide a comprehensive overview of all artifacts discovered in the registry analysis. It is important to note that divergences in the results can be attributed to the inherent variances between the utilized tools, highlighting the significance of tool selection in registry analysis.

Moreover, the disk image of the system was investigated, and the investigation employed VirtualBox to generate the disk image in raw format *.img*. Following Tor Browser usage, a systematic effort was made to recover artifacts, resulting in the identification of essential files and registry keys associated with Tor. The detailed findings highlight the successful retrieval of *tor.exe*, *firefox.exe*, Tor installation files, and downloaded files. Autopsy's efficacy in uncovering recently run programs, coupled with storage forensics, facilitated a thorough exploration of Tor-related activities on the system.

Finally, through network analysis, the system's connection to the Tor Browser was successfully detected, and the presence of numerous onion websites that are exclusively accessible through the dark web was identified. Furthermore, some images from the visited websites as well as certain credentials were uncovered.

This study showcases the efficacy of digital forensics in determining the intricate details of Tor Browser usage and the invaluable insights it yields for information security practitioners.

7. Conclusions and Future Work

This research advanced digital forensics by employing all four fundamental techniques for Tor Browser analysis: memory forensics, storage forensics, network forensics, and registry forensics. This comprehensive approach provides a more holistic understanding of Tor Browser activity compared to previous studies that used only a subset of these techniques. This study collected and analyzed 41 registry entries from the target PC, surpassing the previous maximum of 26 registries, thereby enhancing the scope of registry forensics.

This study achieved the precise time correlation of processes, which contributed significantly to the accurate reconstruction of events. This study utilized advanced reverse engineering methods to recover deleted artifacts, deepening the analysis of storage forensics. This study conducted real-time simulations of dark web usage scenarios, allowing for the thorough inspection of packet captures to identify potential malicious activities.

For future work, we plan to extend this study's findings by validating and applying the research methodologies to other Windows-based operating systems, including the Microsoft Windows Server, to assess their effectiveness in diverse environments. It is also relevant to Explore novel forensic techniques with a focus on automating forensic methods to provide time-effective and cost-effective solutions. Finally, there is a need to adapt and refine forensic methodologies to accommodate advancements and emerging technologies in the digital age.

Author Contributions: Conceptualization, M.S.J.; Validation, D.M. and K.M.; Formal analysis, D.M. and K.M.; Resources, Z.I.; Writing—original draft, M.S.J.; Writing—review & editing, Z.I., M.K. and Z.M.; Visualization, M.K.; Supervision, S.M.S.; Project administration, S.M.S.; Funding acquisition, Z.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare that they have no conflict of interest with respect to the author or publication of this article.

References

1. Schriner, J. *Monitoring the Dark Web and Securing Onion Services*; City University of New York: New York, NY, USA, 2017.
2. Kumar, A.; Sondarva, K.; Gohil, B.N.; Patel, S.J.; Shah, R.; Rajvansh, S.; Sanghvi, H. Forensics Analysis of TOR Browser. In Proceedings of the International Conference on Information Security, Privacy and Digital Forensics, Goa, India, 2–3 December 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 331–341.
3. Angeli, V.M.; Atamli, A.; Karafili, E. Forensic analysis of Tor in Windows environment: A case study. In Proceedings of the 17th International Conference on Availability, Reliability and Security, Vienna, Austria, 23–26 August 2022; pp. 1–10.
4. Teng, S.Y.; Wen, C.Y. A forensic examination of anonymous browsing activities. *Forensic Sci. J.* **2018**, *17*, 1–8.
5. Mehta, S.D.; Upadhyay, D. A review on classification of tor-nontor traffic and forensic analysis of tor browser. *Int. J. Eng. Res. Technol. (IJERT)* **2020**, *9*, 776–778.
6. Huang, M.J.C.; Wan, Y.L.; Chiang, C.P.; Wang, S.J. Tor browser forensics in exploring invisible evidence. In Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan, 7–10 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 3909–3914.
7. Warren, A. *Tor browser Artifacts in Windows 10*; SANS Information Security Reading Room: Rockville, MD, USA, 2017.
8. Jadoon, A.K.; Iqbal, W.; Amjad, M.F.; Afzal, H.; Bangash, Y.A. Forensic analysis of Tor browser: A case study for privacy and anonymity on the web. *Forensic Sci. Int.* **2019**, *299*, 59–73. [CrossRef] [PubMed]
9. Muir, M.; Leimich, P.; Buchanan, W.J. A forensic audit of the tor browser bundle. *Digit. Investig.* **2019**, *29*, 118–128. [CrossRef]
10. Fiaz, F.; Sajjad, S.M.; Iqbal, Z.; Yousaf, M.; Muhammad, Z. MetaSSI: A Framework for Personal Data Protection, Enhanced Cybersecurity and Privacy in Metaverse Virtual Reality Platforms. *Future Internet* **2024**, *16*, 176. [CrossRef]
11. Nelson, R.; Shukla, A.; Smith, C. Web browser forensics in google chrome, mozilla firefox, and the tor browser bundle. In *Digital Forensic Education: An Experiential Learning Approach*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 219–241.
12. Asif, S.; Ambreen, M.; Muhammad, Z.; ur Rahman, H. Cloud computing in healthcare-investigation of threats, vulnerabilities, future challenges and counter measure. *LC Int. J. STEM* **2022**, *3*, 63–74. ISSN: 2708-7123
13. Darcie, W.; Boggs, R.; Sammons, J.; Fenger, T. Online anonymity: Forensic analysis of the tor browser bundle. *Forensic Sci. Int.* **2014**. Available online: https://www.marshall.edu/forensics/files/WinklerDarcie_ResearchPaper_8-6-141.pdf (accessed on 7 July 2024).
14. Gunapriya, S.; Vatsavayi, V.K.; Varma, K.S. Forensic Investigation of Tor Bundled Browser. In Proceedings of the International Conference on Intelligent and Smart Computing in Data Analytics: ISCDA 2020, Guntur, India, 13 March 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 281–289.
15. Alfosail, M.; Norris, P. Tor forensics: Proposed workflow for client memory artefacts. *Comput. Secur.* **2021**, *106*, 102311. [CrossRef]
16. Leng, T.; Yu, A. A framework of darknet forensics. In Proceedings of the 3rd International Conference on Advanced Information Science and System, Sanya, China, 26–28 November 2021; pp. 1–6.
17. Rehman, F.; Muhammad, Z.; Asif, S.; Rahman, H. The next generation of cloud security through hypervisor-based virtual machine introspection. In Proceedings of the 2023 3rd International Conference on Artificial Intelligence (ICAI), Islamabad, Pakistan, 22–23 February 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 116–121.
18. Flanagan, J. Using Tor in Cybersecurity Investigations. Master's Thesis, Utica College, Utica, NY, USA, 2015.

19. Irfan, M.; Ali, S.T.; Ijlal, H.S.; Muhammad, Z.; Raza, S. Exploring The Synergistic Effects of Blockchain Integration with IOT and AI for Enhanced Transparency and Security in Global Supply Chains. *Int. J. Contemp. Issues Soc. Sci* **2024**, *3*, 1326–1338.
20. Akintaro, M.; Pare, T.; Dissanayaka, A.M. Darknet and black market activities against the cybersecurity: A survey. In Proceedings of the Midwest Instruction and Computing Symposium (MICS), North Dakota State University, Fargo, ND, USA, 5–6 April 2019.
21. Syverson, P. Practical vulnerabilities of the tor anonymity network. *Adv. Cyber Secur. Technol. Oper. Exp.* **2013**, *60*, 60–73.
22. Reed, M.G.; Syverson, P.F.; Goldschlag, D.M. Anonymous connections and onion routing. *IEEE J. Sel. Areas Commun.* **1998**, *16*, 482–494. [CrossRef]
23. Dingledine, R.; Mathewson, N.; Syverson, P.F. Tor: The second-generation onion router. In Proceedings of the USENIX Security Symposium, San Diego, CA, USA, 9–13 August 2004; Volume 4, pp. 303–320.
24. Aggarwal, G.; Bursztein, E.; Jackson, C.; Boneh, D. An analysis of private browsing modes in modern browsers. In Proceedings of the 19th USENIX Security Symposium (USENIX Security 10), Washington, DC, USA, 11–13 August 2010.
25. Iesar, H.; Iqbal, W.; Abbas, Y.; Umair, M.Y.; Wakeel, A.; Illahi, F.; Saleem, B.; Muhammad, Z. Revolutionizing Data Center Networks: Dynamic Load Balancing via Floodlight in SDN Environment. In Proceedings of the 2024 5th International Conference on Advancements in Computational Sciences (ICACS), Lahore, Pakistan, 19–20 February 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1–8.
26. Ghafarian, A.; Seno, S.A.H. Analysis of privacy of private browsing mode through memory forensics. *Int. J. Comput. Appl.* **2015**, *132*, 27–34. [CrossRef]
27. Kauser, S.; Malik, T.S.; Hasan, M.H.; Akhir, E.A.P.; Kazmi, S.M.H. Windows 10's Browser Forensic Analysis for Tracing P2P Networks' Anonymous Attacks. *Comput. Mater. Contin.* **2022**, *72*, 1251–1273. [CrossRef]
28. Hejazi, S.M.; Talhi, C.; Debbabi, M. Extraction of forensically sensitive information from windows physical memory. *Digit. Investig.* **2009**, *6*, S121–S131. [CrossRef]
29. Chetry, A.; Sharma, U. Dark web Activity on Tor—Investigation challenges and retrieval of memory artifacts. In Proceedings of the International Conference on Innovative Computing and Communications: Proceedings of ICICC, Delhi, India, 21–23 February 2020; Springer: Berlin/Heidelberg, Germany, 2021; Volume 1, pp. 953–964.
30. Goldschlag, D.M.; Reed, M.G.; Syverson, P.F. Hiding routing information. In Proceedings of the International Workshop on Information Hiding, Cambridge, UK, 30 May–1 June 1996; Springer: Berlin/Heidelberg, Germany, 1996; pp. 137–150.
31. Rehman, A.U.; Nadeem, A.; Malik, M.Z. Fair feature subset selection using multiobjective genetic algorithm. In Proceedings of the Genetic and Evolutionary Computation Conference Companion, Boston, MA, USA, 9–13 July 2022; pp. 360–363.
32. Fatima, M.; Abbas, H.; Yaqoob, T.; Shafqat, N.; Ahmad, Z.; Zeeshan, R.; Muhammad, Z.; Rana, T.; Mussiraliyeva, S. A survey on common criteria (CC) evaluating schemes for security assessment of IT products. *PeerJ Comput. Sci.* **2021**, *7*, e701. [CrossRef] [PubMed]
33. Arshad, M.R.; Hussain, M.; Tahir, H.; Qadir, S.; Memon, F.I.A.; Javed, Y. Forensic analysis of tor browser on windows 10 and android 10 operating systems. *IEEE Access* **2021**, *9*, 141273–141294. [CrossRef]
34. Sajan, P.P.; Balan, C.; Priya, M.J.D.; Sreedeep, A.L. Tor browser forensics. *Turk. J. Comput. Math. Educ. (TURCOMAT)* **2021**, *12*, 5599–5608.
35. Pizzolante, R.; Castiglione, A.; Carpentieri, B.; Contaldo, R.; D'Angelo, G.; Palmieri, F. A machine learning-based memory forensics methodology for TOR browser artifacts. *Concurr. Comput. Pract. Exp.* **2021**, *33*, e5935. [CrossRef]
36. Kent, K.; Chevalier, S.; Grance, T.; Dang, H. *Sp 800-86. Guide to Integrating Forensic Techniques into Incident Response*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2006.
37. Hariyadi, D.; Kusuma, M.; Sholeh, A.; Fazlurrahman. Digital Forensics Investigation on Xiaomi Smart Router Using SNI ISO/IEC 27037: 2014 and NIST SP 800-86 Framework. In Proceedings of the International Conference on Science and Engineering (ICSE-UIN-SUKA 2021), Yogyakarta, Indonesia, 27 October 2021; Atlantis Press: Amsterdam, The Netherlands, 2021; pp. 143–147.