Security Vulnerability Assessment on Threads Application Through Digital Forensics Analysis

Wadduwage Shanika Perera, Ahsan Islam and Cihan Varol
Sam Houston State University, Huntsville, Texas, USA

scp030@shsu.edu
axi034@shsu.edu
cvarol@shsu.edu

Abstract: The rapid emergence of new social media applications has introduced fresh vectors for cybercrime, highlighting the need for timely security vulnerability assessments. This paper presents a comprehensive security vulnerability assessment of Threads, a newly emerging social networking application, by examining its behaviour and data handling through a digital forensic analysis. The study followed a structured experiment which involved installing the the Universal Windows Platform (UWP) applications for Instagram and Threads on a Windows 11 device, conducting typical user activities between two test accounts, acquiring forensic disk images and memory dumps, capturing network traffic, followed by a digital forensic analysis of the discovered artifacts. The primary motivation behind this analysis is to uncover potential security vulnerabilities of the application through a forensic examination of data remnants left by the application. Data acquisition and analysis were carried out using tools such as FTK Imager, Autopsy, Belkasoft Evidence Center, Volatility 3 and Wireshark. The study revealed a range of security and privacy concerns related to the application's data storage, memory usage, and network utilization. For instance, user-generated content and application metadata were found in application files without adequate encryption and sensitive user credentials were discovered, in plaintext. Additionally, insecure handling of backend communications and permissive CORS configurations were observed, introducing risks such as session hijacking and Cross-Site Scripting (XSS) vulnerabilities. Findings of this research underscore the need for improved security mechanisms in modern social media applications. This study provides valuable insights for developers, cybersecurity professionals, and digital forensic investigators to strengthen the security posture of current social networking applications.

Keywords: Social network application security and privacy, Security vulnerability assessment, Social media forensics, Digital Forensics, Threads application

1. Introduction

The increase in the use of social media as a tool of communication has rapidly changed the way people interact and exchange information. A prime example of a new development is Threads, a Meta-owned text-only social networking application (Meta, 2023). Threads application is directly linked with Instagram application, which allows sharing username and profile picture. Users can share text or any multimedia files including images and videos, while messaging is supported only through Instagram. Koetsier (2023) states that Threads became the fastest-growing social media platform in 2023, reaching 100 million users in just 5 days, but its growth has since significantly slowed. Despite the decline in its user base, Threads continues to rank among the most widely used social media platforms. Unlike Instagram, which prioritizes images and videos, Threads is designed for text-based conversations, like Twitter/X, which makes it appealing for discussions, updates, and real-time interactions. Kominers and Wu (2023) state that Threads is moving forward supporting ActivityPub, the decentralized social networking protocol used by Mastodon, allowing cross-platform interactions unlike closed networks like Twitter/x. Thus, Threads remains a strong competitor in microblogging space.

While these platforms enhance connectivity, they are highly vulnerable to cybercrimes due to their diverse and anonymous nature, allowing criminals to hide behind fake identities and exploit personal information. Phishers, scammers and child predators can create fake profiles and exploit personal details including daily routines and locations that are shared in social media platforms, leading to criminal activities such as fraud, stalking, and the spreading of illegal or false content. As Threads continues to play a larger role in both professional and personal communication, the risk from hackers, fraudsters and misinformation agents in form of data leakage, identity theft and fake news make security measures increasingly crucial. Attackers target host programs through compromising on application weaknesses, weak encryption methods, bad session handling, and insecure communication channels. For instance, ephemeral messaging highlighted by Threads is a major selling point that looks to offer secure and private messaging but may put users in harm's way if it is developed poorly. Brown, Onik and Baggili (2024) state that Threads's cross-communication with Instagram expands the attack vector, calling for detailed security analyses to protect users' right to privacy and data authenticity. Threads has not yet been thoroughly studied from a digital forensic standpoint, particularly considering its tight integration with Instagram and its availability as a Universal Windows Platform (UWP) app.

Menahil et al (2021) highlights the need for forensic analysis of emerging social media applications primarily due to the importance of maintaining an up-to-date understanding of the types of artifacts that can be recovered from new and evolving applications. Such knowledge ensure that digital evidence can be accurately identified, interpreted, and presented in a court of law. Beyond its value in supporting digital investigations, we argue that forensic analysis is also a valuable approach for identifying security flaws in new social media applications. Unlike traditional security assessments that often rely on static code reviews or surface-level penetration testing, forensic investigations examine the application's actual behaviour during real-world usage. By analysing disk artifacts, memory states, and network interactions, forensic techniques can expose hidden or unintended data persistence, insecure storage practices, and improper data handling that may not be visible through conventional testing. Given the sensitive data processed by modern social media apps, forensic methods offer a ground-truth approach to evaluating security.

This study assesses the Threads application's vulnerabilities through forensic analysis of disk images, memory dumps, and network traffic in a Windows 11 environment. Although Threads is primarily designed for mobile use, in this study the UWP (Universal Windows Platform) app versions of Threads and Instagram were used on a Windows PC to facilitate easy artifact retrieval while maintaining a similar behaviour of apps to their mobile counterparts. By correlating findings across disk, memory, and network layers, this research aimed to identify and highlight critical security and privacy concerns. The key objectives of our study are:

- Retrieve and examine disk artifacts, including allocated and unallocated space, and windows registry.
- Investigate memory dumps for session credentials, user activities, and unencrypted sensitive data.
- Analyze network traffic for insecure communication and protocol vulnerabilities.

The contributions of this work are threefold: (1) it presents, to the best of our knowledge, the first forensic analysis of the Threads application using desktop UWP apps, (2) it uncovers security flaws of Threads application in storage, memory utilization and network communications, and (3) it provides actionable insights for developers to improve application security and user privacy, while offering digital forensics practitioners a up-to-date understanding of the recoverable artifacts that can be used as evidence in a court of law.

The remainder of this paper is organized as follows. Section 2 reviews related work on security analysis of social media applications using various approaches. Section 3 outlines the methodology including the scenario setup, data acquisition techniques, artifact retrieval processes, forensic analysis, and the tools used for disk, memory and network traffic analysis. Section 4 discusses the key findings and security implications observed during digital forensic analysis. We conclude the paper in Section 5 with suggestions for future work.

2. Related Work

Digital forensic analysis has now become one of the principles in evaluating security risks of Social Media Applications. Several papers have investigated the processes of artifact acquisition, the encryption readability, and the privacy issues of platforms such as Instagram, Twitter/X and YouTube. These studies offer valuable information about methodological approaches and recommendations for the investigation of social media environments. However, limited research has been conducted on Threads, leaving a gap in understanding its forensic artifacts, data security, and privacy implications.

Brown, Onik and Baggili (2024) used a network traffic based digital forensic analysis approach to evaluate program interface communication (API) architectures of Threads and Bluesky application. Their analysis revealed a stronger security posture in Threads compared to Bluesky, likely due to Meta's infrastructure. Robust encryption using TLS 1.2 and HTTPS was used, with secure authentication mechanisms, including client-side encrypted passwords and CSRF protection. Minor security vulnerabilities such as missing security flags on some cookies, were identified. The mobile analysis showed minimal user data storage, making forensic data extraction challenging. Attempts to probe the platform using automated security tools were largely blocked, demonstrating active security measures. The absence of significant quantities of material on Mobile Device Artifact (MDA) also acted as a confining factor for the acquisition of proof in device-only examinations. This research does not provide a comprehensive analysis of the privacy of Threads related data retained in the volatile or non-volatile storages of the devices.

While limited research exists on Threads specifically, several studies have examined vulnerabilities in other popular social media networks. Kumar and Karabiyik (2021) analyzed the forensic implications of disappearing messages on Instagram, highlighting the challenges investigators face in tracking digital evidence related to criminal activities such as cyberbullying and illegal transactions. The study identified the presence of vanished

messages in the application database, inconsistencies in message visibility, and the storage mechanisms of user-uploaded media. Additionally, it examined user searches and interactions within the platform. This work demonstrates that digital forensics can be effectively utilized to identify and analyse artifacts, including disappearing messages, in social media applications, aiding forensic investigations. In another study, Alisabeth and Pramadi (2020) stressed databases, cache, and log files as the most important forensic artifacts created by Instagram on Android devices. They demonstrated the ability to reconstruct exchanged messages, including photos, videos, and voice recordings, as well as locate sent media in local storage. Additionally, it found that video uploads, photos, and Instagram stories are retained on the device. However, the study faced limitations in recovering media messages received through direct messages, as they were not stored locally but only accessible via URL links in the database, which expire over time. Furthermore, they were not able to find data related to comments and likes retained in local storage, posing challenges for forensic investigations. Both studies highlight the persistence of forensic artifacts despite Instagram's focus on ephemeral content.

Expanding beyond Instagram, Menahil et al. (2021) conducted a comprehensive forensic investigation of five widely used social networking apps: Instagram, LINE, Whisper, WeChat, and Wickr to determine the recoverability of user data from internal device storage. Their research focused on identifying, extracting, and analysing artifacts using Magnet AXIOM, Autopsy, and XRY. The study evaluated each tool's performance with NIST standards. The findings demonstrate that substantial forensic evidence can persist even after data deletion or app uninstallation, underscoring the importance of social media app analysis in criminal investigations. In a related study, Johnson et al (2022) thoroughly examined how alternative social media applications such as MeWe, CloutHub, Minds Mobile and Safechat store user-related artifacts. The study identified stored user account details, including usernames, user IDs, and authentication tokens, as well as traces of social media interactions, such as posts, comments, and private messages. Cached data was analysed, demonstrating how applications store media files, chat records, and other user activity artifacts. Their study uncovered vulnerabilities where unencrypted media links were stored and could be accessed without authentication, posing security risks. However, this study is limited by the selective discussion of artifacts, as only the most significant findings were analysed while redundant data was excluded. These studies collectively show that significant forensic traces can remain even after deletion or app uninstallation.

Focusing on Twitter, Badillah et al (2023) conducted a forensic investigation of defamatory content on Twitter using static forensic methods. Static forensic method is beneficial in uncovering digital evidence that is inactive or deleted after the incident. The researchers successfully extracted and analysed digital evidence, including emails, usernames, deleted comments and posts, and modifications to the perpetrator's account, resulting in a comprehensive understanding of the context and nature of defamatory material. However, the investigation faced limitations in retrieving certain deleted data, such as chat records, due to the limitations of the capabilities of the forensic tools used. Umrani, Javed, and Iftikhar (2022) analysed encrypted Twitter traffic on Android to identify patterns linked to user activities. They classified user behaviours by examining fixed traffic patterns and extracting artifacts, also revealing hidden platform flexibilities through firewall analysis. They were able to identify user-related data despite encryption, but tracing traffic origins has become more difficult due to anonymizing tools like VPNs and legal privacy constraints. Together, these studies show both the potential and limitations of forensic tools in extracting relevant evidence, particularly in contexts where encryption or deletion is involved.

Research into broader web application vulnerabilities has also contributed to the understanding of social media risks. Meiser, Laperdrix, and Stock (2021) analysed cross-origin communication in 5,000 web applications, focusing on third-party integrations like ads, analytics, and social media. They used a trust graph to map trust relationships through mechanisms such as domain relaxation, postMessages, and CORS. Their study showed how vulnerabilities, especially XSS, on trusted sites could be exploited to compromise others, highlighting the expanded attack surface and risks of implicit trust in cross-origin interactions. In a more targeted app analysis, Hu and Karabiyik (2024) performed a forensic analysis of TikTok on Android and iOS, focusing on the TikTok Shop feature. They used Magnet AXIOM and Belkasoft X to extract artifacts related to user activity, media interactions, and shopping transactions, including credit card data from Android and address fragments from iOS. They mapped up artifacts in TikTok-related folder structures in Android and database paths, into user behavior. However, they faced challenges in identifying legacy files and distinguishing video playback states on Android, highlighting the need for continuous forensic adaptation to evolving social media platforms.

In another significant research, Alyahya and Kausar (2017) conducted a forensic analysis of Snapchat on Android using Magnet AXIOM and Autopsy, simulating various user activities. They found that, despite

Snapchat's ephemeral design, messages, snap metadata, and contact information could be recovered from device memory, especially from the tcspahn.db database. While AXIOM offered structured metadata views, it lacked clear video previews and deletion indicators. Autopsy provided better visuals but weaker artifact categorization. The study emphasized limitations in both tools and the need for manual checks to improve forensic accuracy.

Our study adds to the existing literature, by conducting a detailed digital forensic analysis on Threads platform, focusing on disk space analysis, data carving, memory artifact analysis and network traffic analysis. The research provides practical recommendations to improve encryption, secure session management, and address these threats, advancing social media forensics and enhancing security and privacy on social media platforms. The proposed techniques are generic thus can be applied to any similar application.

3. Methodology

In this study, we implemented an experiment with three main steps, scenario creation and execution, data acquisition, and artifact retrieval, to simulate a realistic forensic investigation workflow. This is shown in Figure 1. The methodology was designed to comprehensively capture and analyse forensic artifacts from multiple system layers, ensuring a holistic approach. The division of our analysis into disk image, memory dump, and network traffic analysis enabled the identification of persistent, volatile, and real-time artifacts respectively. This tri-layered approach was chosen to maximize evidence coverage and reflect standard best practices in digital forensics investigations, thereby enhancing the validity and reliability of our findings.

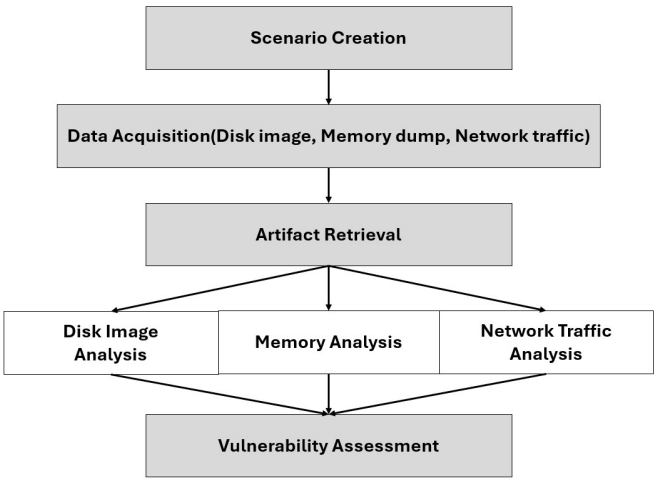

Figure 1: Workflow for digital forensic investigation of Threads application

3.1 Scenario Creation and Execution

For our analysis we used two different Windows machines with the configurations shown in Table 1, as test environments. Threads (version: 359.1) and Instagram (version: 358.0.0.51.97) applications were downloaded (on 27th of November 2024) from Microsoft Store app and installed on these two Windows machines. The applications were installed as Universal Windows Platform (UWP) apps which were designed to run seamlessly across various Microsoft Windows devices.

Table 1: Machine configurations

Specification	Configuration	
	Machine 1	**Machine 2**
Operating system	Windows 11 Pro (64-bit)	Windows 11 Pro (64-bit)
Processor	Intel Core i7-12700K (3.60 GHz)	AMD Ryzen 5 5600G (3.90 GHz)
RAM	32 GB	16 GB
GPU	NVIDIA GeForce RTX 3080 with 10 GB VRAM	Analyse captured network traffic
Storage	1 TB SSD	1 TB SSD

During the scenario creation and execution, 12 different activities and interactions were performed within the Threads and Instagram applications using two user accounts as shown in Table 2. Two different new Threads accounts were created with usernames “Bob” (user ID: “bobmarleysam") and “Trudy” (user ID: “tru_dy1234”). The two test environments were not controlled to replicate a real-world incident scenario. However, the machines were rebooted before executing the case scenario.

Table 2: Case scenario

Activity/ Interaction	User	
	From/ By	To/ Of
Create account	Bob, Trudy	-
Login to account	Bob, Trudy	-
Post new thread with text-only, image, video	Bob, Trudy	-
Add location tag when posting a thread	Bob, Trudy	-
Save existing thread	Bob, Trudy	-
Like/React to existing thread	Bob	Trudy
	Trudy	Bob
Comment on existing thread with text-only and image	Bob	Trudy
	Trudy	Bob
Direct Message to other account with text-only, image,video	Trudy	Bob
	Bob	Trudy
Search within app	Bob, Trudy	-
Delete post	Bob, Trudy	-
Delete comment	Bob, Trudy	-
Delete message	Bob, Trudy	-

3.2 Data Acquisition

Data acquisition was done on 27th of November 2024. During the execution of the case scenario, the live network traffic was captured and recorded using the Wireshark software tool. After the execution of the case scenario, the volatile memory (RAM) of the machines was captured and memory dumps were acquired using the FTK Image software tool. Volatile data were also acquired from live memory using Windows commands executed in Command Prompt. Finally disk images of the Solid-State Drives (SSD) were acquired using the FTK Imager software tool, which captured the allocated, unallocated, and system-only spaces. Cryptographic hash values of the images were computed to assure the validity. The versions of the tools are mentioned in Table 3.

3.3 Artifact Retrieval

Once the case scenario execution and data acquisition phases were complete, we retrieved artifacts related to the user activities and user interactions within Threads and Instagram applications from the captured network traffic, disk image, memory dump and live memory using software tools Mentioned in Table 3.

3.3.1 Disk image

In disk image analysis, it was investigated how Threads application related artifacts are saved into storage and whether there are possibilities for exploitation. Autopsy and Belkasoft Evidence Center were used to search and locate cache data, temp files, database files. The analysis focused on disk spaces where Threads application saves the working data including users’ interaction records, multimedia files and settings. User’s messages, meta-information and multimedia files were obtained from database files. In addition, Foremost tool was used to carve files in fragments in the unallocated disk space which were deleted beforehand. From the Windows registry entries, Application specific configuration, information about user preferences, session identifiers and application state information were obtained. Temporary files and thumbnail files related to images and videos were explored.

3.3.2 Memory dump

Volatility 3 tool was used to retrieve artifacts from volatile memory dumps. Both live memory analysis and offline memory dump analysis were conducted to gather artifacts. Key memory regions and plugins such as pslist, handles, and dlllist were used to identify active processes and associated resources. Application-specific data were acquired, including internal app states, configuration settings, and session identifiers. User-related information such as login credentials, session tokens, and interaction data was retrieved using string extraction and memory carving techniques. Additionally, memory regions were searched for geolocation data and metadata associated with user activity. Artifacts linked to multimedia content, including cached images and videos, were also captured. Network-related data such as open connections, session details, and in-memory logs of server requests and responses were obtained using relevant Volatility modules, enabling a structured collection of all transient data present during app execution.

3.3.3 Network traffic

An analysis of overall network traffic was explored while using Threads application to identify probable threats such as transferring of plain texts, open or unmarked APIs, and leaking of data. Packet capture was carried out using Wireshark. Network traffic was generated by login into the system, creating and updating statuses and uploading posts, media content and messages. The traffic logs were assessed with the help of an application called Network Miner which deals with network artifacts retrieval and analysis. Using Network Miner, we obtained artifacts such as time of the session, file transfer, DNS requests and communication models. The investigation also involved decrypting headers of HTTPS requests and responses to identify security weaknesses.

3.3.4 Tools used and tools setup

For data acquisition, artifact retrieval, and analysis, widely used tools were employed, as listed in Table 3. GUI-based tools were chosen for their free availability, user-friendliness, and minimal setup requirements. Volatility 3, a command-line tool, was used for memory analysis due to its extensive set of built-in plugins for examining various memory data types.

Table 3: Tools used

Software tool	Version	Usage
FTK Imager	v4.7.3.81	Acquire disk image and memory dump
Wireshark	v4.4.2	Capture and analyze network traffic
Network Miner	v2.9	Analyse captured network traffic
Autopsy	4.21.0	Analyse disk image
Belkasoft	X- Trail	Analyse disk image
Volatility 3	v2.8.0	Analyse memory dump
Foremost	1.5.7	Data carving

4. Results and Discussion

In this section we present the results of our analysis in three sections: network traffic analysis, disk image analysis and volatile memory analysis.

4.1 Disk Image Analysis

Disk image evaluation provided insights into storage practices within applications, revealing crucial artifacts and metadata related to HDD storage. The investigation uncovered that user screenshots of application homepages were stored randomly in application data directories, likely for caching or debugging purposes. This raises concerns about privacy if such files are saved without user consent. Metadata analysis revealed timestamps and modification dates, aiding in reconstructing application usage timelines. Additionally, user interactions, comments, and responses were logged in cache files and linked to unique user IDs. Records of shared posts, in-app searches, and liked posts were found in the cache files. However, no direct messages with images attached were recovered. These findings highlight the importance of forensic frameworks in assessing data retention policies and underscore the need for transparency and stronger privacy measures in social media data management. The results are detailed in Table 4.

Table 4: Results of disk analysis

Discovered artifact	Finding(s)
Randomly captured homepage screenshots	Insights into caching or debugging practices
Last Modified dates & timestamps associated with stored files	Timeline reconstruction
Post responses and comments linked to User IDs	Evidence of user interaction and association of activities with unique IDs
Shared posts, in-app searches and liked posts stored in app data	Details on user behivior and preferences
Downloaded Images	Downloaded shared image

4.2 Volatile memory analysis

During the memory analysis, we were able to uncover traces of sensitive user data, application data, insecure network communications and potential application security vulnerabilities.

4.2.1 Offline volatile memory analysis

The artifacts retrieved from the memory dump using Volatility 3 and the findings are summarized in Table 5. Although UWP apps enhance security by running within a sandboxed environment, memory analysis revealed sensitive data stored in plain text, such as user and application credentials and Hash-Based Message Authentication Code (HMAC) which is used to verify integrity and authentication of transmitted data. This suggests that login credentials could be compromised if an attacker gains access to the machine. The revealed login credentials can be used in user account hijacking attempts and Man-in-the-middle (MITM) attacks to bypass authentication and steal further sensitive data during communication.

Table 5: Results of offline memory analysis

Tool (plugin)	Discovered artifact(s)	Finding(s)
volatility3.plugins.windows.pstree	process names, process IDs, parent process IDs, location of .exe files, execution timestamp	Process data
volatility3.plugins.windows.pslist	process IDs, parent process IDs, process names, process created time, process exit time in UTC, memory offsets of each process	Process data
volatility3.plugins.windows.strings	Passwords in plain-text, account ID, account name, user e-mail ID, user ID, app ID, session ID, device ID, HMAC (Hash-Based Message Authentication Code)	Sensitive user credentials
	posted text, post creation timestamps, locations in posts, hints of media upload, photo/image information with dates, auto-generated captions of uploaded images and videos	Threads posted by user in plain-text
	viewed user account IDs and names	Profiles viewed by user
	comments in plain text	Comments/replied posed by user
	Search queries in plain text, search results	In-app search by user
	sent messges in plain text, received messages in plain text, URLs in messages in plain-text	Messages sent through Instagram by user
	Searched locations, location tags in craeted posts, locations of viewd posts, timestamps of location tags	Locations related to user
	Deleted posts in plain-text, delted messages in plain-text, deleted comments on existing posts, deletion timestamps	Deletions by user
	Hints of app installation, installer file location, application .exe location,	App Installation by user
	Names of images uplaoded, image location on system, image usage timestamp	Images used by user
	inline/Eval in a backend request	Cross-origin communication between Threads and Instagram

Tool (plugin)	Discovered artifact(s)	Finding(s)
	session cookie user id, csrftoken, network protocol used	Cookies related to user
	executed method names and status codes	Code base of Threads
volatility3.plugins.windows.handles (File handles and Registry handles)	User ID of user, user IDs of other interacted users, search within app in plain-text, email ID, accountKeyString, seed, userKeyBaseString	Sensitive network transmission and encryption data
	Local machine data, current user	Local machine and currrent user data

The information revealed related to user activities such as texts, locations and hyperlinks in the posted threads, comments and direct messages were stored in plain text in volatile memory during the app execution. This highlights inadequate data protection and absence of encryption. Memory analysis also revealed multimedia content such as images and videos used during Threads and Instagram processes, with original names replaced. If an attacker gains access to the memory, they can easily extract this information without any additional effort. If a device is compromised or a data breach occurs, unencrypted personal information such as login credentials, location history, and posted content can be exposed, potentially leading to identity theft, location tracking, or other malicious activities. Plain-text storage of user data violates privacy expectations and exposes sensitive information that attackers can exploit for targeted phishing, stalking, or social engineering attacks.

On the other hand, the same sensitive data including the search queries and the results which are stored in plaintext can be used for digital forensics purposes to support crime investigations. Artifacts such as messages in plaintext can be used as legal evidence. Our analysis suggests that investigators can reveal users' intentions by analyzing search data, deleted content, and app installation data. Timestamps provide investigators with crucial information related to the timing and sequence of activities. This extracted content can reveal user habits and interests as well. The metadata of multimedia content can offer contextual evidence that can corroborate or refute other findings during investigations.

The backend requests that were extracted during the memory analysis uncovered some parameters such as "cross-origin-resource-policy:cross-origin" which suggests that resources are explicitly shared with Instagram indicating internal communication between Threads and Instagram services. This is because Threads application uses Instagram application for messaging functionality. According to Amazon Web Services (2024), CORS (Cross-Origin Resource Sharing) is a mechanism used for smooth integration between applications which defines a way for client web applications from one domain to interact with resources in a different domain. Such backend requests to Instagram's services that serve the purpose of fetching message data revealed that it allows "Unsafe-inline/Eval". This indicates it permits the use of inline styles or JavaScript and use of JavaScript functions like "eval ()", "setTimeout()" with string arguments which can expose the app to Cross-Site Scripting (XSS) attacks if input validation and sanitization are not handled properly. Through such attacks, attackers can inject malicious code/scripts into the web page's content. This can be considered a major vulnerability in Threads application. File handles revealed some sensitive information such as "accountKeyString", "seed" and "userKeyString". These can be used to extract encryption keys and used to decrypt the sensitive data stored in backlogs or transmitted via HTTPS. The keys might allow session hijacking and data exfiltration.

The revealed cookie data showed the use of Cross-site request forgery (CSRF) tokens in the data transmission communications. Microsoft (2024) states that CSRF tokens/AntiForgery tokens help protect against CSRF attacks. They were tied to the user's session, ensuring that requests to the server come from the authenticated user, preventing unauthorized actions by third-party sites. The tokens were also sent only over secure connections, adding an extra layer of protection. This is a good security measure.

4.2.2 Live volatile memory analysis

The artifacts retrieved from the live memory using Windows commands and the evidence derived are summarized in Table 6.

Table 6: Results of live volatile memory analysis

Tool (command)	Discovered artifact(s)	Findings
tasklist	process IDs, process names, window names, running user	Process data
netstat	local address and ports, foreign address and ports, connection status and time spent in status, process IDs	Network connections related to Threads and Instagram

The live memory analysis revealed additional important information such as the windows names and the running usernames. It also helped to find the geological location using the local IP address. Also, the remote ports that the relevant processes were connected to were using HTTPS protocol which suggests the applications ensure data privacy, integrity, and security over the network, protecting sensitive information.

4.3 Network Traffic Analysis

Network traffic analysis revealed that Threads utilizes the HTTPS protocol for encrypted message exchange and data transmission, minimizing the exposure of user-generated content. The app clients-initiated communications using TLS 1.3 for secure data transfer. However, the decrypted HTTPS headers revealed the application names and the Windows version and architecture (Windows NT 10.0; Win64; x64). This information can support profiling of attacker tools or malware behaviour. Captured packets contained valuable information, including Media Access Control (MAC) addresses, IP addresses, and network request timestamps, which helped track user activity timelines. The analysis of source and destination IP addresses provided insights into the application's communication patterns and the servers within the Threads network. These findings highlight HTTPS as a critical safeguard for secure data transmission while emphasizing the forensic significance of metadata in reconstructing timelines, characterizing network behaviour, and supporting investigative objectives when content remains unrecoverable.

5. Conclusion and Future Work

In this research study, we examined and analyzed the vulnerabilities in user interactions within the Threads application and messaging through Instagram application particularly in cases where device memory and storage are compromised. The findings indicate that while secure communication protocols such as HTTPS and encryption can prevent some attacks like man-in-the-middle (MITM) and phishing, flaws in their implementation can still open the door to potential exploitation. Moreover, cross-origin communication within the app has revealed certain vulnerabilities that need to be addressed to secure user interactions further.

The analysis of volatile memory was conducted using the Volatility tool, which presented challenges due to complex dependencies with specific python packages and pre-compiled windows symbols. While these issues hindered the setup and execution of some plugins, the insights gained from memory analysis were instrumental in understanding the vulnerabilities present in the application. Additionally, the lack of tool-specific technical knowledge and frequent updates were obstacles that slowed down the process but also highlighted areas for improvement in memory analysis tools. Despite these challenges, the study emphasizes the importance of securing communication in applications like Threads, where sensitive user data can be exposed if proper measures are not taken.

As future work, a focus on expanding the scope of analysis further to explore multi-media files used in app interactions, which can contain valuable forensic data, and to dive deeper into encrypted communication protocols that may still harbor vulnerabilities are needed. Additionally, the work can be extended to the investigation to include memory handles analysis related to the Threads application usage. Memory handles analysis can reveal additional clues about application security vulnerabilities, through possible memory leaks, dangling pointers, access to sensitive data, race conditions and improper memory access. Furthermore, an in-depth analysis of cross-reference communication would provide a more comprehensive understanding of the system's behavior during the app’s operation, aiding in the detection of anomalies. To enhance the research’s applicability and widen its scope, future work should also consider analyzing different operating systems such as Android OS and iOS. Finally, we think there is a need to improve the automation of memory analysis scripts to make the process more scalable and user-friendly.

In conclusion, this research has laid the foundation for securing user data in social media applications and has pointed to several areas that need further exploration. As user application vulnerability assessment and digital forensics continue to evolve, addressing these challenges will be crucial to ensuring the safety and privacy of users in increasingly complex and dynamic environments.

Ethics declaration: Ethical clearance was not required for this research.

References

Alisabeth, C. and Pramadi, Y.R., (2020) 'Forensic analysis of Instagram on Android', IOP Conference Series: Materials Science and Engineering, 1007, 012116. DOI: 10.1088/1757-899X/1007/1/012116

Alyahya, T. and Kausar, F. (2017) 'Snapchat analysis to discover digital forensic artifacts on Android smartphone', Procedia Computer Science, 109, pp. 1035–1040. doi:10.1016/j.procs.2017.05.421.

Amazon Web Services (2024) *What is CORS? - Cross-Origin Resource Sharing Explained*. [online] Available at: https://aws.amazon.com/what-is/cross-origin-resource-sharing/ (Accessed: 8 December 2024).

Brown, J., Onik, A.R. and Baggili, I. (2024) 'Blue Skies from (X?s) Pain: A Digital Forensic Analysis of Threads and Bluesky', In *Proceedings of the 19th International Conference on Availability, Reliability and Security (ARES 2024)*. Vienna, Austria: Association for Computing Machinery, pp. 1–12.

Hu, X. and Karabiyik, U. (2024) 'Shopping while watching: An updated forensic analysis of TikTok on Android and IOS', 2024 International Symposium on Networks, Computers and Communications (ISNCC), pp. 1–8. doi:10.1109/isncc62547.2024.10759027.

Johnson, H., Volk, K., Serafin, R., Grajeda, C. and Baggili, I. (2022) 'Alt-tech social forensics: Forensic analysis of alternative social networking applications', *Forensic Science International: Digital Investigation*, 42(Supplement), p. 301406

Koetsier, J. (2023) *100 million sign-ups in 5 days. 8 reasons why threads is blowing up*, *Forbes*. [online] Available at: https://www.forbes.com/sites/johnkoetsier/2023/07/07/70-million-sign-ups-8-reasons-why-threads-is-blowing-up/ (Accessed: 08 December 2024).

Kominers, Scott and Wu, Liang., (2023). *Threads Foreshadows a Big — and Surprising — Shift in Social Media*. [online] Available at: https://hbr.org/2023/07/threads-foreshadows-a-big-and-surprising-shift-in-social-media

Kumar, S.T. and Karabiyik, U. (2021) 'Instagram Forensic Analysis Revisited: Does anything really vanish?', In *The 2021 International Symposium on Networks, Computers and Communications (ISNCC 2021)*. Dubai, UAE: IEEE, pp. 1–6.

Meiser, G., Laperdrix, P. and Stock, B., (2021). 'Careful Who You Trust: Studying the Pitfalls of Cross-Origin Communication'. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security* (ASIA CCS '21), Virtual Event, Hong Kong, 110–122. Association for Computing Machinery. DOI: 10.1145/3433210.3437510.

Menahil, A. et al. (2021) 'Forensic Analysis of Social Networking Applications on an Android Smartphone'. *Wireless Communications and Mobile Computing*, pp.1–36. doi:10.1155/2021/5567592.

Meta (2023) *Introducing Threads: A New Way to Share With Text.* [online] Available at: https://about.fb.com/news/2023/07/introducing-threads-new-app-text-sharing/ (Accessed: 08 December 2024).

Microsoft (2024) *Prevent Cross-Site Request Forgery (XSRF/CSRF) attacks in ASP.NET Core*. [online] Available at: https://learn.microsoft.com/en-us/aspnet/core/security/anti-request-forgery?view=aspnetcore-9.0 (Accessed 8 December 2024).

Reski Badillah, R. *et al.* (2023) 'Digital Forensic Evidence Analysis in revealing defamation on social media (Twitter) using the static forensics method', *Ceddi Journal of Information System and Technology (JST)*, 2(2), pp. 22–33. doi:10.56134/jst.v2i2.45.

Umrani, A., Javed, Y. and Iftikhar, M. (2022) 'Network forensic analysis of Twitter application on Android OS', 2022 International Conference on Frontiers of Information Technology (FIT), pp. 249–254. doi:10.1109/fit57066.2022.00053.