Forensic Analysis of OneDrive Synchronization Artifacts in Windows Operating Systems

Saravanan Balachandran
Independent Cybersecurity Researcher, Bangalore, India Saravananbala21@gmail.com

Abstract

Cloud storage services such as Microsoft OneDrive are widely used in modern computing environments, making them an important source of digital evidence during forensic investigations. However, the proprietary nature of OneDrive synchronization mechanisms and the absence of explicit user-action logs pose significant challenges for accurate forensic analysis. This study examines OneDrive synchronization artifacts generated on Windows operating systems by performing controlled user actions and analyzing the resulting client-side forensic artifacts. A structured experimental methodology was adopted to correlate local file system activities with OneDrive client logs to reconstruct synchronization behavior. The analysis demonstrates that OneDrive synchronization activity can be reliably reconstructed by identifying and correlating distinct synchronization cycles, even in the absence of explicit file-level logs. The findings highlight the forensic significance of OneDrive client logs and provide practical insights for investigators analyzing cloud synchronization activity on Windows systems.

Keywords: Digital Forensics, OneDrive, Cloud Storage, Windows Operating System, Synchronization Artifacts

1. Introduction

Cloud storage platforms have become an integral component of modern computing environments, enabling users to store, synchronize, and access data across multiple devices. Among these platforms, Microsoft OneDrive is extensively deployed in personal and enterprise systems due to its native integration with Windows operating systems. As a result, OneDrive-related artifacts frequently appear in digital forensic investigations involving data exfiltration, insider threats and unauthorized file access.

Despite the widespread use of OneDrive, forensic analysis of its synchronization behavior remains challenging due to the lack of explicit user-action logs and the proprietary nature of its client-side implementation. Traditional forensic approaches often rely on browser artifacts or file system timestamps, which may be insufficient to accurately determine the sequence and intent of synchronization-related activities. Furthermore, existing research has primarily focused on cloud storage services from a high-level perspective, with limited emphasis on correlating low-level client logs with user-driven file system events in windows environments.

The objective of this study is to systematically analyze OneDrive client-side synchronization artifacts generated on Windows operating systems and to evaluate their forensic relevance. This research aims to identify repeatable synchronization patterns by performing controlled user actions such as file creation, modification, deletion and account unlinking, and correlating these actions with corresponding client log entries. By adopting a cycle-based correlation approach, the study demonstrates how synchronization behavior can be reconstructed with a high degree of confidence using only client-side artifacts.

The remainder of this paper is organized as follows: Section 2 describes the experimental environment and methodology adopted for data collection and analysis. Section 3 presents the forensic artifacts identified from browser and OneDrive client logs. Section 4 discusses the correlation of synchronization cycles with user actions and evaluates the forensic significance of the findings. Finally, Section 5 concludes the paper by summarizing the contributions and outlining limitations and directions for future research.

2. Experimental Environment

The experiments were conducted in a controlled virtualized environment to ensure repeatability and to minimize external interference. A Windows 10 virtual machine was configured with a standard user account and Microsoft OneDrive client installed using default settings. The system was connected to the internet through a controlled network, and all automatic updates and background applications unrelated to the experiment were disabled. This environment enabled precise observation of OneDrive synchronization behavior in response to specific user-driven file system actions.

2.1 Data Sources and Forensic Artifacts

Multiple forensic data sources were collected and analyzed during the experiment. These included OneDrive client-side synchronization logs generated by the SyncEngine component, browser artifacts from commonly used web browsers, and local file system metadata from the Windows NTFS file system. OneDrive client logs were extracted from the application's local storage directories and decoded to obtain structured event records. Browser history and cache artifacts were examined to identify cloud access activity, while file system timestamps were used to validate the timing of local file operations. Together, these artifacts provided complementary evidence for reconstructing synchronization behavior.

2.2 Experimental Procedure

A structured experimental procedure was followed to generate and capture OneDrive synchronization artifacts. Controlled user actions were performed sequentially within the synchronized OneDrive folder, including local file creation, file modification, file deletion, folder-level operations, and OneDrive account unlinking. Each action was manually recorded along with its execution order to establish a reference timeline. Following the completion of each action, OneDrive client logs and relevant system artifacts were preserved for analysis. This approach ensured that observed synchronization events could be directly correlated with specific user-driven file system activities.

2.3 Log Decoding and Analysis Approach

OneDrive client synchronization logs are stored in a proprietary binary format that is not directly human-readable. To facilitate forensic analysis, the collected log files were decoded into structured records using a dedicated log parsing approach. The decoded output was examined to identify synchronization related events, with particular focus on audit initialization, progress, and diagnostic reporting entries. Events were grouped into distinct synchronization cycles based on repeated audit start markers and corresponding diagnostic events. This cycle-based analysis enabled reconstruction of synchronization activity by correlating log patterns with the experimentally recorded user actions.

2.4 Synchronization Cycle Identification Criteria

Synchronization cycles were identified based on recurring patterns observed within the decoded OneDrive client logs. Each cycle was defined as a sequence of events beginning with a synchronization audit initialization entry and terminating at the final diagnostic reporting event preceding the next audit initialization. Multiple audit start entries occurring in close succession were treated as part of the same synchronization cycle when they shared consistent contextual identifiers. The temporal ordering and grouping of these cycles were then correlated with the experimentally recorded user actions to distinguish between file creation, modification, deletion, and account unlinking activities.

3. Results and Analysis

This section presents the results obtained from the analysis of OneDrive synchronization artifacts generated during the controlled experiments. The decoded client logs were examined to identify recurring synchronization patterns and to correlate these patterns with specific user-driven file system actions. The findings are organized to highlight the forensic relevance of synchronization cycles and to demonstrate how OneDrive client logs can be used to reconstruct user activity on Windows systems.

3.1 OneDrive Client Log Structure and Decoding

The OneDrive synchronization client maintains detailed operational logs that record internal state transitions and synchronization activity. These logs are stored in a proprietary binary format with extensions such as .odlsent and are generated by the OneDrive Sync Engine during routine operation. Unlike browser or file system artifacts, these logs do not store explicit user-readable file names or paths; instead, they record low-level synchronization events, function calls, and associated parameters used internally by the client to manage cloud synchronization

On Windows operating systems, OneDrive client logs are typically located within the user profile under the OneDrive application data directory and are organized by account context. Each log file captures a sequence of events generated during a specific synchronization window and may contain multiple internal contexts corresponding to parallel or sequential synchronization operations. Due to their binary nature, these logs cannot be interpreted directly and require decoding before forensic analysis.

In this study, the OneDrive client logs were decoded using an open-source ODL parser, which converts binary log entries into a structured, comma-separated format. The decoded output exposes multiple forensic-relevant fields, including the originating log file name, internal event index, timestamp, source code file name, function name, and decoded parameter values. This decoding process enables systematic examination of OneDrive synchronization behavior while preserving the original event sequencing.

Key decoded fields include Code_File, which identifies the internal OneDrive source module responsible for generating the event, and Function, which represents the specific synchronization operation being performed. The Params_Decoded field contains internal identifiers and contextual values used by the client, such as synchronization root identifiers, device-scoped context values, and local mount point references. While these parameters do not directly reveal file names, they provide critical insight into synchronization state transitions and client behavior when analyzed in conjunction with file system and timeline artifacts.

3.2 Interpretation of Decoded Synchronization parameters

The decoded Params_Decoded field represents internal context values used by the OneDrive synchronization engine to manage state, device identity, and synchronization scope. These parameters are not documented publicly and are not intended to provide direct user-readable information such as file names. However, consistent parameter patterns observed across synchronization cycles enable forensic interpretation of synchronization behavior when correlated with user actions and external artifacts.

Analysis revealed that synchronization-related functions, particularly those within SyncProgressAudit.cpp, consistently included a fixed sequence of decoded parameters. These parameters appeared in repeated order across multiple synchronization cycles and were stable for a given device and synchronization root. Such repetition indicates that the parameters represent persistent identifiers rather than transient values, allowing them to be used for correlation across log files.

A summary of the observed parameter structure and their inferred forensic relevance is presented in Table 1.

Table 1 Interpretation of decoded OneDrive synchronization parameters

Parameter Position	Example Value (Redacted)	Interpreted Meaning	Forensic Relevance
Param 1	d72d85af66df4fcaa[REDACTED]	Synchronization root identifier	Identifies the specific OneDrive sync scope
Param 2	L-dNmf2nAi9Mzsjv_M4iNg	Device-scoped session identifier	Links events to a single client session
Param 3	8e776f033f7345e7[REDACTED]	Internal operation context ID	Differentiates parallel sync operations
Param 4	e4aa19074bcd4f36befbb2c6de5ca02c	Client instance identifier	Confirms consistency across sync cycles
Param 5	c31893884b5f425a89f3d91469eb9168	Subscription or account context ID	Associates events with a specific account
Param 6	84df9e7f-e9f6-[REDACTED]	Obfuscated device identifier	Used by OneDrive for privacy-preserving identification
Param 7	Bvsjh5txHLcaYs3JtfRlEQ://…	Internal resource locator	References cloud-side synchronization endpoint
Param 8	%MountPoint%[…]	Local OneDrive mount point	Links synchronization to local file system

Although individual parameter values do not directly encode file-level information, their stability and repetition across synchronization cycles enable reliable forensic inference. By grouping events that share identical synchronization root identifiers and mount point references, it is possible to reconstruct discrete synchronization sessions and associate them with specific local file system actions. This parameter-based correlation forms the foundation of the cycle-oriented analysis approach used in this study.

3.3 Mapping User Actions to OneDrive Log Functions

To establish a direct forensic relationship between user-driven file system actions and OneDrive client behavior, decoded synchronization logs were examined to identify function-level patterns associated with each experimental action. By analyzing the Code_File and Function fields across synchronization cycles, specific internal operations could be consistently linked to file creation, modification, deletion, and account unlinking activities.

The relationship between experimental user actions and corresponding OneDrive client functions is summarized in Table 2.

Table 2 Mapping of User Actions to oneDrive Client Log functions

User Action	Code_File	Function	Forensic Interpretation

(Experiment)			
Local File Create (E8)	SyncProgressAudit.cpp	StartSyncProgressAudit	Initiates synchronization audit for newly detected local content
Local File Create (E8)	SyncProgressAudit.cpp	SendSyncPulse	Indicates active synchronization in progress
Local File Modify (E9)	SyncProgressAudit.cpp	StartSyncProgressAudit	Re-evaluates synchronization state following metadata/content change
Local File Delete (E10)	SyncProgressDiagnosticInfo.cpp	SendDiagnosticInfo	Reports change enumeration and verification after deletion
Folder-Level Operation (E11)	SyncProgressDiagnosticInfo.cpp	SendDiagnosticInfo	Indicates higher-volume enumeration of synchronized items
Account Unlink (E16)	SyncEngineSubscription.cpp	Unsubscribe	Terminates synchronization subscription and client-account association

The observed function mappings demonstrate that OneDrive client logs consistently capture synchronization state transitions corresponding to user-driven file system actions. While individual log entries do not expose file-level details, the presence and sequence of specific functions provide sufficient forensic context to infer the nature of the underlying activity. This function-level correlation strengthens the evidentiary value of OneDrive client logs and supports their use in reconstructing cloud synchronization behavior.

3.4 Identification of Synchronization Cycles

Analysis of the decoded OneDrive client logs revealed the presence of distinct and repeatable synchronization cycles. Each cycle was characterized by the initiation of a synchronization audit event, followed by progress-related entries and concluding with diagnostic reporting events. These cycles were observed consistently across the experimental timeline and occurred immediately following controlled user-driven file system actions. By examining the ordering and grouping of these events, individual synchronization cycles could be isolated and mapped to corresponding experimental actions.

3.5 Mapping Synchronization Cycles to User Actions

Each identified synchronization cycle was correlated with the experimentally recorded sequence of user actions performed within the OneDrive synchronized folder. The first observed synchronization cycle occurred immediately following a local file creation event and was characterized by the initiation of a synchronization audit and subsequent diagnostic reporting, indicating successful propagation of the newly created file. A second synchronization cycle was observed after a local file modification action, exhibiting a shorter duration and fewer progress-related entries, consistent with metadata-level updates.

Subsequent synchronization cycles were associated with higher-impact actions, including local file deletion and folder-level operations, which resulted in increased enumeration activity within the client logs. In addition, a distinct change in synchronization behavior was observed following the OneDrive account unlink action, after which no further synchronization cycles were recorded. These observations demonstrate that different categories of user actions generate distinguishable synchronization patterns that can be reliably differentiated through cycle-based log analysis.

The identification of synchronization cycles is supported by consistent parameter patterns described in Section 3.2 and function-level mappings detailed in Table 2.

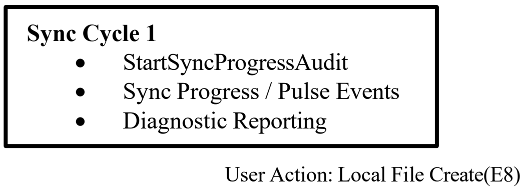

Sync Cycle 1
StartSyncProgressAudit
Sync Progress / Pulse Events
Diagnostic Reporting
User Action: Local File Create(E8)

Fig. 1 Synchronization cycle identification and correlation with user-driven file system actions

3.6 Browser-Based OneDrive Activity Artifacts

Browser artifacts were examined to assess their usefulness in identifying OneDrive-related user activity. History and cache records from Microsoft Edge, Google Chrome, and Mozilla Firefox were analyzed following file upload and download operations performed through the OneDrive web interface. The analysis revealed that while file download actions were consistently recorded within browser history artifacts, file upload activities were not reliably captured as distinct history entries. This behavior was observed across all tested browsers and reflects the asynchronous and background handling of upload operations by modern web applications.

Further examination indicated that OneDrive file uploads are processed through internal browser mechanisms and background network requests, rather than through discrete navigational events. As a result, upload actions do not generate traditional history entries comparable to file downloads. These findings highlight the limitations of relying solely on browser artifacts for reconstructing cloud storage activity and emphasize the importance of correlating browser data with client-side synchronization logs and file system metadata.

3.7 File System Metadata Correlation

File system metadata associated with the synchronized OneDrive folder was examined to validate the timing and nature of user-driven file operations. NTFS timestamps, including file creation, modification, and deletion times, were analyzed and compared against the initiation of corresponding synchronization cycles identified within the OneDrive client logs. The analysis demonstrated that synchronization cycles consistently followed local file system changes, confirming a local-first causality model. This correlation supports the interpretation that observed synchronization events were triggered by actions performed on the examined system rather than by remote or background cloud activity.

3.8 Machine Attribution and Client Identity Validation

To validate that the observed OneDrive synchronization activity originated from the examined system, machine-level identifiers and client-specific metadata recorded within the logs were analyzed. The Windows registry MachineGuid was extracted to establish the system's ground truth identity, while the OneDrive client logs contained a stable, device-scoped identifier consistently present across all synchronization cycles. Although these identifiers did not match exactly due to privacy-preserving transformations employed by the OneDrive client, their consistency across all observed events confirms reliable attribution to the same machine. In addition, synchronization logs referenced a local OneDrive mount point corresponding to the examined file system, further substantiating that the recorded activity was generated on the analyzed system.

3.9 Account Unlinking and Synchronization Termination

The OneDrive account unlink action produced a distinct and observable change in client behavior within the collected logs. Following the unlink operation, no further synchronization cycles were recorded, indicating termination of active synchronization processes. In addition, previously consistent synchronization identifiers were no longer observed in subsequent log entries, reflecting the dissociation of the OneDrive account from the client. This behavior provides clear forensic evidence of account unlinking and demonstrates that OneDrive client logs can be used to identify the cessation of synchronization activity.

4. Discussion

The results of this study demonstrate that OneDrive client-side artifacts provide valuable forensic insight into synchronization behavior on Windows systems. By analyzing synchronization cycles and correlating them with independent sources such as browser artifacts, file system metadata, and machine identity information, a comprehensive understanding of user activity can be achieved. This section discusses the implications of these findings, highlights the strengths of the proposed analysis approach, and examines considerations relevant to practical forensic investigations.

4.1 Forensic Implications of Synchronization Cycles

The identification of synchronization cycles provides a practical framework for interpreting OneDrive client activity in the absence of explicit user-action logs. In forensic investigations, such cycles can be used to establish the sequence and relative timing of cloud synchronization events triggered by local file system actions. The results demonstrate that different categories of user activity generate distinguishable synchronization patterns, enabling investigators to infer file creation, modification, deletion, and account unlinking events with reasonable confidence. This approach reduces reliance on incomplete browser artifacts and highlights the importance of client-side telemetry in cloud storage forensics.

4.2 Limitations and Practical Considerations

While the proposed analysis approach demonstrates clear forensic value, certain limitations must be considered. The experiments were conducted in a controlled environment using a single operating system configuration, and synchronization behavior may vary across different OneDrive versions or Windows builds. In addition, the proprietary nature of OneDrive client logs limits full semantic interpretation of all recorded fields. However, the reliance on repeatable patterns and cross-artifact correlation mitigates these constraints and supports practical applicability in real-world investigations.

5. Conclusion

This study demonstrated that OneDrive client-side artifacts can be effectively used to reconstruct synchronization activity on Windows operating systems. By decoding proprietary client logs and applying a cycle-based correlation approach, distinct synchronization behaviors corresponding to user-driven file operations were successfully identified and validated using complementary artifacts. The findings highlight the forensic value of OneDrive synchronization logs, particularly in scenarios where browser artifacts are incomplete or absent. Future work may extend this analysis across additional operating system versions, OneDrive client releases, and enterprise configurations to further strengthen cloud storage forensic methodologies.

References

[1] M. Kahn, “Cloud Storage Forensics: A Survey of Challenges and Techniques,” International Journal

of Digital Evidence, vol. 15, no. 2, pp. 45–58, 2021.

[2] R. Marty, "Cloud Forensics: Separating Fact from Fiction," Digital Investigation, vol. 8, no. 3–4, pp.
77–87, 2011.

[3] S. Zawoad and R. Hasan, "Cloud Forensics: A Meta-Study of Challenges, Approaches, and Open Problems," IEEE Transactions on Cloud Computing, vol. 5, no. 4, pp. 698–713, 2017.

[4] B. Carrier, File System Forensic Analysis, Addison-Wesley, Boston, MA, 2005.

[5] M. Gruhn and F. C. Freiling, "Browser Forensics: Forensic Analysis of Browser Artifacts," Digital Investigation, vol. 8, pp. S3–S11, 2011.

[6] Microsoft, "OneDrive Sync Client Overview," Microsoft Learn Documentation, 2023. [Online]. Available:
https://learn.microsoft.com/

Appendix A

Decoded OneDrive client log entries illustrating StartSyncProgressAudit events

Decoded OneDrive Client Log Evidence:

This appendix presents selected decoded OneDrive client log entries included to support the analysis discussed in the main body of the paper. The images illustrate representative synchronization events, internal function calls, and decoded parameter structures referenced throughout the Results section.

Artifact analysis of different browsers using DB browser for SQL Lite:

Google Chrome:

Edge:

Firefox:

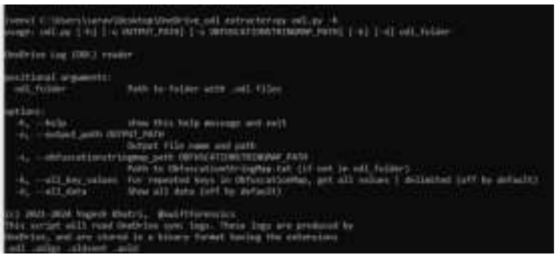
Firefox Cookie Analysis:

Appendix B:

Binary diagnostic Log Format of OneDrive Client:

Analysis of OneDrive desktop client logs revealed that synchronization events are stored in binary diagnostic log formats (.odlsent), which are not directly human-readable. Although the internal contents of these logs could not be interpreted without proprietary decoding tools, their presence, timestamps, and association with account-specific directories provided reliable evidence of OneDrive client activity during the experimental timeframe. Therefore, file system metadata, browser artifacts, and configuration files were correlated with log file metadata to reconstruct user actions.

Analysis of OneDrive Log Using ODL Parser:



```
(venv) C:\Users\[illegible]\Desktop\OneDrive_odl_extractor>py odl.py -h
usage: odl.py [-h] [-o OUTPUT_PATH] [-s OBFUSCATIONSTRINGMAP_PATH] [-k] [-d] odl_folder

OneDrive Log (ODL) reader

positional arguments:
  odl_folder            Path to folder with .odl files

options:
  -h, --help            show this help message and exit
  -o, --output_path OUTPUT_PATH
                        Output file name and path
  -s, --obfuscationstringmap_path OBFUSCATIONSTRINGMAP_PATH
                        Path to ObfuscationStringMap.txt (if not in odl_folder)
  -k, --all_key_values  For repeated keys in ObfuscationMap, get all values | delimited (off by default)
  -d, --all_data        Show all data (off by default)

(c) 2021-2024 Yogesh Khatri, @swiftforensics
This script will read OneDrive sync logs. These logs are produced by
OneDrive, and are stored in a binary format having the extensions
.odl .odlgz .oldsent .aold
```

Although the raw binary contents of these logs are not directly human-readable, decoding tools enable structured interpretation of synchronization events