Data remnants analysis of document files in Windows: Microsoft 365 as a case study

By:
Jihun Joun, Sangjin Lee, Jungheum Park

From the proceedings of
The Digital Forensic Research Conference
DFRWS APAC 2023
Oct 17-20, 2023

DFRWS is dedicated to the sharing of knowledge and ideas about digital forensics research. Ever since it organized the first open workshop devoted to digital forensics in 2001, DFRWS continues to bring academics and practitioners together in an informal environment.
As a non-profit, volunteer organization, DFRWS sponsors technical working groups, annual conferences and challenges to help drive the direction of research and development.
https://dfrws.org

Forensic Science International: Digital Investigation 46 (2023) 301612

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

DFRWS 2023 APAC - Proceedings of the Third Annual DFRWS APAC

Data remnants analysis of document files in Windows: Microsoft 365 as a case study

Jihun Joun, Sangjin Lee, Jungheum Park *

School of Cybersecurity, Korea University, 145 Anam-Ro, Seongbuk-Gu, Seoul, South Korea

ARTICLE INFO

Keywords:
Digital forensics
Data remnants
Electronic document
File tracing
Microsoft 365

ABSTRACT

In the era of digitization, electronic evidence has become increasingly important for investigations and legal proceedings. However, traditional digital forensic technologies, such as recovery and carving, face limitations because of difficulties acquiring unallocated areas intact. Furthermore, artifacts and files previously used for tracing can be easily deleted manually or via anti-forensic tools, which hinders traceability. This paper presents a novel framework to overcome these limitations. This method facilitates a more precise and comprehensive tracing of residual files through data remnants analysis, a forensic approach that investigates traces of deleted or overwritten data. By systematically constructing a dataset based on user action, we identify and analyze all data remnants within the system, thereby revealing file traces. The results of a case study on Microsoft 365 demonstrate our proposed framework's superior efficacy and accuracy compared to existing methods. Our approach offers valuable insights into data remnants analysis and contributes to digital forensic investigations conducted on Windows systems.

1. Introduction

As technology progresses, most information is stored in digital form, making digital data crucial as evidence. However, the inherent characteristics of digital data, including the ease of replication and tampering, have led to frequent incidents of intentional or accidental manipulation or deletion of evidence. In several notable cases (Apple Inc. v. Samsung Elecs Co., Ltd, 2012; Learning Care Group, Inc. v. Armetta, 2016; Cat3, LLC. v. Black Lineage, Inc., 2016; BMG Rights Mgmt. (US) LLC. v. Cox Communs., Inc., 2018), the spoliation of evidence has proven a critical point of contention in court cases, substantially impacting the proceedings and outcomes. These cases underscore the importance of data integrity and its influence on the fairness and validity of the final judgment. Criminals have become increasingly adept at manipulating or destroying digital evidence to conceal their tracks. Techniques such as permanent file deletion, removal of unallocated areas, and elimination of artifacts and logs are commonly used to obstruct investigations and hinder legal proceedings, posing substantial challenges in establishing guilt in various criminal activities. Consequently, specialized technologies and methods to investigate data destruction are essential for preventing and solving such crimes.

Conventional digital forensic methodologies, encompassing data recovery (Durrant, 2005; Garrie, 2014) and file carving (Daniel, 2011), have been utilized to detect evidence tampering. However, these techniques primarily focus on the unallocated area, which is progressively arduous to fully acquire because of the variety of encryption, TRIM function, and data wiping tools available (Mitchell et al., 2017).

Well-known artifacts extensively studied by numerous researchers to trace data remnants are easily erased even with commercial tools and can be manually deleted by individuals with expertise in digital forensics. Furthermore, in light of the frequent updates of operating systems (OS) and applications, artifacts' path, content, and structure can vary, leading to potentially incorrect analysis results.

To meet these challenges, it is essential to identify all data remnants in the allocated area, encompassing the system and storage media. Therefore, this paper proposes a systematic framework for identifying as many data remnants as possible in the allocated areas and one that can be easily applied to new versions of files or operating systems. Data remnants refer to residual digital data that remains even after the data has been deleted or modified. This can include metadata such as the filename, path, timestamp, size, as well as the content itself. For instance, even when a file is deleted from the system, its name might still linger in the system registry or link file, serving as a data remnant and a potential source of forensic information.

* Corresponding author.
E-mail addresses: jihunjoun@korea.ac.kr (J. Joun), sangjin@korea.ac.kr (S. Lee), jungheumpark@korea.ac.kr (J. Park).

https://doi.org/10.1016/j.fsidi.2023.301612

Available online 13 October 2023
2666-2817/© 2023 The Author(s). Published by Elsevier Ltd on behalf of DFRWS. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

This research focuses on the data remnants that persist when utilizing document files through Microsoft 365 programs. Microsoft 365 includes a range of programs, such as Office, OneDrive, Teams, Outlook, and OneNote, widely employed for document usage, storage, and sharing. The data remnants left behind while using these programs are identified and analyzed based on our framework.

Using Microsoft 365 as a case study, the subsequent questions are examined.

- RQ1: Can the proposed framework identify both existing and previously unknown data remnants?
- RQ2: What types of previously unknown data remnants are discovered?
- RQ3: Can the discovered data remnants be used to infer the former existence of files?

This paper presents several contributions. First, a methodology for tracking deleted or manipulated files is proposed by identifying data remnants even after anti-forensic actions. Second, we show the usefulness of the methodology by identifying several artifacts that point to the former existence of files but to the best of our knowledge were previously unknown. Third, we provide an automated tool and show that it can effectively be used to identify deleted or manipulated files using the discovered artifacts.

The structure of this paper is as follows: Section 2 introduces related work, Section 3 explains background knowledge, Section 4 proposes a framework for data remnants forensics, Section 5 covers the analysis results of data remnants found on a Windows system, Section 6 describes the developed tool and evaluates the usefulness and applicability of the framework through experiments, and finally, Section 7 concludes this work.

2. Related work

Numerous methods have been researched and proposed to address data remnants and file trace. Quick and Choo analyzed data remnants in a Cloud system, focusing on identifying and examining the user accounts, passwords, and URLs. The authors examined the artifacts and network usage related to targeted cloud storage services, including Dropbox (Quick and Choo, 2013a), Microsoft SkyDrive (OneDrive) (Quick and Choo, 2013b), and Google Drive (Quick and Choo, 2014). Joun et al. introduced a framework for identifying and analyzing data remnants, conducting experiments on macOS (Joun et al., 2023). Our study further refined this latter framework into four systematic stages, and found novel traces that can track the deleted files in Windows, which is the most widely used OS worldwide. Particularly, experiments were conducted with a broader dataset and methods to validate the practicality of this framework when using Microsoft 365.

In addition, SANS created a poster for Windows forensic analysis including Windows artifacts (SANS, 2023). This poster highlights various file and folder opening artifacts that can be found in Open/Save MRU, Recent Files, MS Word Reading Locations, Last Visited MRU, Shortcut (LNK) Files, Office Recent Files, Shell Bags, Jump Lists, Office Trust Records, Office OAlerts, and Internet Explorer files. Furthermore, deleted items and file existence can be identified through artifacts, such as Thumbs.db, Windows Search Database, Internet Explorer files, Search-WordWheelQuery, User Typed Paths, Thumbcache, and Recycle Bin. In our research paper, we utilized these listed artifacts as one of the three methods for identifying file traces during our experiments.

While the artifacts described by SANS appear credible, it is unclear how they were derived. To identify such artifacts, Garfinkel et al. generalized several experimental methods into the approach of differential forensic analysis (Garfinkel et al., 2012). Briefly spoken, differential analysis attempts to identify changes by comparing the state of a filesystem before and after an action has been executed. Kälber et al. refined this method to avoid noise in the file change measurements (Kälber et al., 2013).

Several studies have developed into methods for tracking files within the Windows environment. The file system journaling log file, $LogFile, has been used for analyzing file creation, alteration, and deletion. Oh et al. developed a technique that reconstructs changes in the metadata within the $MFT by using transaction data recorded in the $LogFile (Oh et al., 2021). Additionally, the authors probed the mechanisms of record storage through a deep analysis of the $LogFile structure and suggested a recovery method for records without fixed values (Oh et al., 2022).

Bunting studied a particular method for examining ESI spoliation, in which the author utilized traces of data wiping tool (CCleaner), Restore Points, File History, prefetch, and the history of the command used (Bunting, 2016). This study is distinct in its specialization in spoliation investigation; however, it has limitations as it primarily focused on Windows 7, and the findings were analyzed from only the existing artifacts.

3. Background: data remnants

3.1. Definition of data remnants

Various files exist within a system, including user files, system files, application files, and log files. The contents recorded in these files often remain intact, even after modification or deletion. Data remnants refer to 'residual data' that persists after a file has been deleted. The residual data may encompass the deleted file's name, path, timestamps, size, and content.

3.1.1. Data remnants file (DRF)

Data remnants are not confined to specific files; instead, they can be stored in multiple files, depending on the OS and applications. It is in general very hard to be aware of all files containing these data remnants, so any method to identify them is necessarily incomplete. In this paper, the concept of *DRF* is introduced, which encompasses files, artifacts, logs, and other items that potentially contain data remnants. For instance, if data remnants are present in the registry hive, event logs, file system metadata, or system and application logs, these artifacts and logs may be classified as *DRF*.

To conduct digital forensic investigations concerning evidence spoliation, it is essential to identify information about files and ascertain which files have been manipulated or deleted. However, determining which files contain data remnants poses a challenge. This paper proposes a methodology for systematically examining *DRF* and identifying the files that once existed on the system, thereby providing evidence to support the possibility of spoliation. We categorized and analyzed *DRF* into two distinct categories.

3.1.2. Unstudied data remnants file (UDRF)

UDRF refers to new logs or artifacts that have not been previously discovered. Identifying novel logs or artifacts generated with the introduction of new versions or updates of OSs and applications can prove challenging.

In addition, this type includes the files that have already been discovered but have not had their internal structure studied, or have not been employed in spoliation investigations. In other words, it refers to a file whose potential data remnants of deleted files within the file are not yet known or recognized. Digital forensic experts examine *UDRF* to uncover new evidence and bolster investigations into tampering with evidence.

3.1.3. Studied data remnants file (SDRF)

As research into digital forensic investigation methods continues, most artifacts and logs for each OS have been discovered and examined. Previously studied files are classified and defined as *SDRF*. To ensure whether a *DRF* is studied, we searched for these files across various online resources, including blogs, white papers, technical reports,

educational videos, and papers, using the *DRF* filenames and paths as keywords. In particular, well-studied artifacts and logs listed in the SANS poster (SANS, 2023), such as recently viewed file lists and link files, are being utilized in spoliation investigations.

As numerous researchers have already validated these files, relatively stable results can be obtained during the analysis process. However, the types of *SDRF* have not been exhaustively cataloged, and *DRF* for each OS and application have not been systematically organized.

4. Framework for data remnants forensics

4.1. Overview

This Section introduces a novel method for detecting traces that remain on a system even after a file has been deleted. Using the proposed method, not only can the traces of deleted files be identified, but the file can also be utilized for discovering unknown artifacts, tracking file history, and analyzing user action. Fig. 1 illustrates Data Remnants Forensics divided into four primary stages: Data Remnants File Dataset creation (*DRFD*), Data Remnants File Identification (*DRFI*), Data Remnants File Analysis (*DRFA*), and Data Remnants File Examination (*DRFE*).

We proposed an enhanced framework, building upon a previously developed framework related to data remnants from other researchers (Quick and Choo, 2013a; Quick and Choo, 2013b; Quick and Choo, 2014; Joun et al., 2023).

Before directly searching for *DRF* on the target system, a dataset is created with the same OS and applications as those of the target system. Next, *DRF* is identified through data pre-processing, keyword searching, and classification within the dataset. Subsequently, the file format, structure, purpose, and content of the identified *DRF* are analyzed. Finally, *DRF* is extracted directly from the target system to identify deleted files.

4.2. Data Remnants File Dataset creation (DRFD)

In this phase, we detail the dataset creation method to identify all *DRF*. By providing a systematic procedure, we enable all researchers to reproduce experiments and assess the validity of the results. The dataset is curated in a system environment tailored to the research target and structured based on a typical user's actions.

4.2.1. Scope definition

For the construction of an appropriate dataset, defining an accurate scope is crucial. This task requires a precise understanding of the system environment pertaining to the target under study or investigation. Consequently, careful selection of the OS, applications, and the types and versions of files is essential. For this study, Windows 11 Home v21H2 and Microsoft 365 v2304 (Build 16327) were installed on VMware Workstation Pro v15.5.0.

4.2.2. Listing user actions

User actions that are feasible with the selected applications and file types are listed. Our research findings indicate that actions altering the file content do not leave separate traces. For instance, changes such as modifying the text size or font do not record the filename in the application log. Therefore, the user action lists mainly involve creating, modifying, and accessing the file itself. The user actions listed in this paper for experiments are shown in Table 1. There could be other user actions not included in our list.

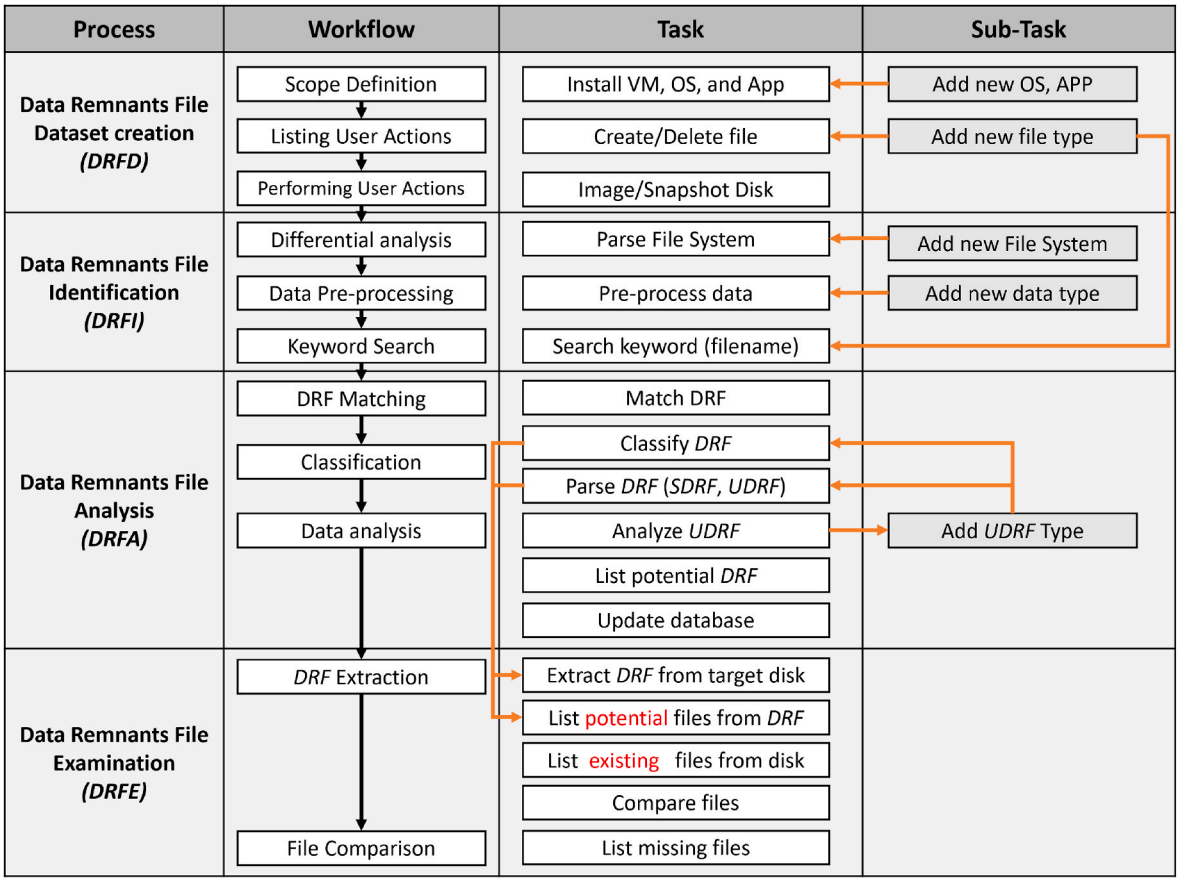

Fig. 1. A comprehensive framework detailing the processes involved in data remnants forensics.

Table 1
An illustrative table detailing various types of user actions considered in this study.

User Actions	Description
Create	Create a new file using the target application
	Create a new file using OneNote
	Create a new file in OneDrive (app)
	Create a new file using 'save as' function
Copy & Share (Syn)	Copy a file between an external drive and an internal drive
	Share a file through Windows Mail
	Synchronize a file through OneDrive (app)
	Send an email (both via web and app)
Download/Upload	Download/upload a file from the OneDrive, Internet, and Teams
Open/Access	Open a file by double-clicking or searching in Explorer
	Open the file properties
	Open a damaged file
	Open a file in OneDrive and Teams (both via web and app)
Modify	Rename a file
	Change the metadata of a file
	Change the content and close without saving
	Save the file as a different file type, such as PDF and HTML
	Change the properties of a file, such as protect file

4.2.3. Performing user actions

The system environment is established to create the dataset, install the OS and applications on a virtual machine, and prepare the necessary accounts. Moreover, having a unique list of filenames and contents to be created facilitates keyword searches.

Once the environment setup is complete, all user actions are performed. Subsequently, all files accessed or created during these user actions are completely deleted. In this case, the term 'completely erased' refers to a file that has been permanently deleted and can no longer be found in the allocated area. For accurate experimental results, files stored not only on local devices but also in conversation contents, emails, and cloud storage services must all be deleted. Depending on the experimental method, file system metadata, artifacts, and logs are also removed. After ensuring the complete removal of all files, the disk is imaged to create the dataset.

4.3. Data Remnants File Identification (DRFI)

DRFI involves identifying *DRF*, which contains data remnants, such as filenames, file paths, timestamps, file sizes, and content, in the created dataset. This stage encompasses the extraction of files from the image, differential analysis, the data pre-processing, and keyword search.

4.3.1. Differential analysis

Extracting all files and reading their contents to search the entire disk image proves to be time-consuming. Therefore, differential analysis (Garfinkel et al., 2012) is utilized to select as many DRF-related files as possible. Our proposed method extracts the files in which the 'modified time' has changed rather than comparing the hash values of files.

However, a full differential analysis often results in numerous false positives and irrelevant results, capturing not only remnants of deleted files but also noise generated by various system operations or processes. While noise can be reduced by repeated differential measurements (Kälber et al., 2013), this process is rather inefficient. Hence, a keyword search technique filters the DRFs that contain potential data remnants.

4.3.2. Data pre-processing

The data pre-processing is essential in establishing an environment suitable for subsequent keyword searches. For the keyword search within the file, all files should be in readable formats. Therefore, data pre-processing procedures such as decompression, deserialization, decoding, and decryption are performed. Most known files can be pre-processed using open-source tools, such as bulk_extractor (Garfinkel, 2013). However, unknown files require manual analysis.

4.3.3. Keyword search

In this phase, the goal is to find data remnants of deleted files from the pre-processed files. Even if the actual file has been completely deleted, data remnants can still be present in some files. Therefore, the search focuses on these files where data remnants may potentially exist. Keyword search is applied to the dataset, not to the target disk involved in the forensic investigation. Keyword search is intended for finding *DRFs* in the dataset, not the deleted files. The list of *DRFs* will be extracted from the target disk to find traces of deleted files.

As such, the filenames and contents used in creating the dataset are searched as keywords. *DRF* matching the keywords are identified and subsequently listed. In addition to filenames and contents, the search can be conducted in various keywords, such as renamed filenames, modified contents, and deleted contents.

4.4. Data Remnants File Analysis (DRFA)

It is essential to verify whether the identified *DRF* has been previously studied or not. As mentioned in Section 3.1.3, this necessitates a literature review from various sources, which can be highly inefficient if repeated for every investigation. Therefore, a *DRF* database is constructed. This database contains elements like *DRF* name, *DRF* filename, *DRF* format, and metadata, as shown in Table 2. Unlike the author's use of metadata, the *DRF* database is populated after manually analyzing the *DRF*. This *DRF* database can be immensely beneficial in establishing a continually useable structure that can be updated during investigations and research. Moreover, the time required for literature review can be significantly reduced by simply matching with the *DRF* stored in the database.

After that identified *DRF* are categorized into two types, *SDRF* and *UDRF*, by reviewing publicly available materials, such as academic papers and technical reports. Most of the file structures for *SDRFs* have already been analyzed, and existing parsers can be used to analyze the data contained within the *SDRF*. However, for *UDRF*, data remnants should be identified and extracted manually by analyzing the file format, structure, purpose, and content. When dealing with an unknown file format, it is best to perform a comprehensive analysis before examining the data remnants. However, due to the complexity and time-consuming nature of this process, it is important to identify only the essential structure of *UDRF*. In the identified *UDRF*, search for filenames using a keyword, then examine the structure used for storing these filenames. As a validation, verify if the same structure for storing filenames exists in identical *UDRF* within other PC or systems. While this method is necessarily imprecise, the results are usually rather good (see Section 5 for details).

4.5. Data Remnants File Examination (DRFE)

This stage is to find deleted files from the target disk or image using identified *DRF*. First, the *DRF* identified and analyzed are extracted from the target device. In this process, the identified files are referred to as 'potential files,' meaning that files may have existed on the target device. Next, 'existing files' are extracted from storage media related to the target, including internal disks, previously connected external storage devices, and cloud systems. If the name of the 'existing file' is not present in the list of 'potential files,' it is considered a file that was once present on the system but is no longer available. By comparing these two file lists, 'missing files' can be identified.

5. Data remnants analysis: Microsoft 365 as a case study

All *DRF* discovered in the *DRFA* phase of the framework are classified and analyzed. As *SDRF* have been known or analyzed in terms of their structure, the presence and type of data remnants within files can easily

Table 2
Summary of analysis results for *DRF* related to Microsoft 365.

Type	DRF name	DRF filename	DRF format	DRF Type	Metadata							
					N	P	C	M	A	O	S	T
MS Office	OfficeFileCache - C4	1380790193167760279.C4	Unknwon	Unstudied	✓	✓						
	TapCache	~null_[timestamp(unix)].json	JSON	Unstudied	✓	✓	✓	✓	✓	✓	✓	✓
	FileActivityStoreV3 - S	[random-base64].S	Unknwon	Unstudied	✓	✓						
	Aggmru - MRU	w-mru4-[language]-[country]-sr.json	JSON	Unstudied	✓	✓	✓	✓		✓	✓	
	Aggmru - REC	w-rec-[language]-[country]-sr.json	JSON	Unstudied	✓	✓	✓	✓	✓	✓		
	MruServiceCache - Documents	Documents_[language]-[country]	JSON	Unstudied	✓	✓	✓	✓	✓	✓		
	MruServiceCache - Places	Places_[language]-[country]	JSON	Unstudied	✓	✓	✓	✓	✓	✓		
	BackstageInAppNavCache	[FileHash-SHA256].json	JSON	Studied	✓	✓	✓	✓	✓	✓	✓	
	Office - Recent File	index.dat	TEXT	Studied	✓	✓						
MS OneDrive	Settings - Dat	[UserCid].dat	LOG	Studied	✓				✓	✓		
	SafeDelete	SafeDelete.db (wal)	SQLite	Unstudied	✓							
MS Teams	Teams - Database	0000[number].ldb (log)	LevelDB	Studied	✓	✓			✓	✓		
	Launcher - Log	Launcher_[date]_[time].log	TEXT	Unstudied	✓	✓			✓	✓		
	MSTeams - Log	MSTeams_[date]_[time].log	TEXT	Unstudied	✓	✓			✓	✓		
	AppSettings	app_settings.json	JSON	Unstudied	✓	✓						
MS Outlook	Mail database	[email adreess].ost	OST	Studied	✓							
MS OneNote	Cache - Bin	[unknown].bin	BIN	Studied	✓							✓

be identified by referring to relevant studies or using digital forensic tools. In contrast, as *UDRF* is an unknown file, its structure and content need to be analyzed to identify data remnants. As this study focuses more on identifying new *DRF* and data remnants rather than analyzing new file formats and structures, only the essential structure and content of the files are analyzed for identifying data remnants. Additionally, we do not claim that the specific actions described are always the direct cause of these data remnants. However, we can confirm that these remnants are related to Microsoft 365 through multiple differential analysis.

This study focuses on identifying *DRF* left by activities related to document files in Microsoft 365 to find deleted document files. Table 2 presents a summary of the analysis results. The analysis was specifically targeted towards remnants that remain even after secure deletion of actual files. The primary emphasis was placed on the *UDRF*. Furthermore, this study solely focuses on the *DRF* related to Microsoft 365 while all other identified *DRF* are listed in Table 3.

The filenames in the format such as [variable], where 'variable' could represent different actual values depending on the context. Furthermore, to unambiguously distinguish files, it is essential to also consider their paths. The metadata column displays the various types of metadata present in artifacts. We categorized these types of metadata based on eight distinct features: Filename (N), Path (P), Created time (C), Modified time (M), Accessed time (A), Other timestamp (O), Size (S), and Text (T). The checkmark (✓) denotes the existence of metadata. For example, the 'OfficeFileCache - C4' contains metadata of the deleted filename and filepath.

5.1. Microsoft Office

MS Office is a component of the integrated software known as Microsoft365, and it connects with various programs, such as OneDrive, SharePoint, and Outlook, generating a range of logs. It is essential to update the analysis results by version, as software updates may alter existing logs or record them in new logs. *DRF* found in this application are stored in %UserProfile%/AppData/Local/Microsoft/Office/[version].

"OfficeFileCache - C4" (1380790193167760279.C4) is a *UDRF* of an

Table 3
Summary of analysis results for *DRF* related to artifacts other than Microsoft 365.

Type	DRF name	DRF filename	DRF format	DRF Type	Metadata							
					N	P	C	M	A	O	S	T
IE, Edge	IE – Web Cache Database	WebCacheV01.dat	EDB	Studied	✓	✓			✓			
	IE – Log	V01.log, V010000[number].log	TEXT	Studied	✓	✓			✓			
Chrome	Download – Log	DownloadMetadata	LOG	Studied	✓							
	Download – Database	[random].ldb (log)	LevelDB	Studied	✓	✓						
	Sessions – Tab	Tabs_[unknown]	SNSS	Studied	✓						✓	
	History	History	SQLite	Studied	✓	✓			✓	✓		
	Favicons	Favicons	SQLite	Studied	✓	✓			✓			
Search	Search – Database	Windows.edb	EDB	Studied	✓	✓	✓	✓	✓		✓	
	Search – Log	edb.jtx, edb[number].jtx, edbtmp.jtx	Unknown	Unstudied	✓							
	Gatherer – Log	SystemIndex.[number].gthr	TEXT	Studied	✓	✓						
EventLog	MS Office Alerts	OAlerts.evtx	EVTX	Studied	✓	✓			✓			
Defender	Protect log	MPLog-[date]-[time].log	TEXT	Studied	✓	✓		✓				
	Download log	MpWppTracing-[date]-[time]~.bin	Unknown	Unstudied	✓	✓						
	Access log	Mpenginedb.db (wal)	SQLite	Unstudied	✓							
Windows Mail	Mail – database	HxStore.hxd	Analyzed	Studied	✓					✓		
File System	File system – MFT	$MFT	Analyzed	Studied	✓	✓	✓	✓	✓		✓	
	File system – Log	$LogFile	Analyzed	Studied	✓					✓		
	File system – Journal	$UsnJrnl:J	Analyzed	Studied	✓					✓		
Memory	Virtual memory file	pagefile.sys	Analyzed	Studied	✓							
OS	Prefetch	[filename]-[hash].pf	Analyzed	Studied	✓	✓						
	MRU – Recent (Registry)	NTUSER.DAT	Analyzed	Studied	✓							
	Link file	[filename].lnk	Analyzed	Studied	✓	✓	✓	✓	✓	✓		
	Jumplist	[random].~Destination-ms	Analyzed	Studied	✓	✓	✓	✓	✓	✓		
	Timeline – Database	ActivitiesCache.db (wal)	Analyzed	Studied	✓	✓					✓	

unknown type, and the principle or rule behind its creation is still not known. This file is stored in ~/OfficeFileCache/0/0/. The filename is always fixed as '13807901931677602 79'. This file contains cached data before being stored on OneDrive or SharePoint and keeps track of the most recently opened files. As depicted in Fig. 2, the file includes the filename as well as file path.

"TapCache" ([unknown]-[unknown]_0_[organization].sharepoint.com_[Language Code]_null_[timestamp(unix)].json) is a *UDRF* and is composed of JSON format which the creation principle or rule remains unknown. This file is stored in ~/[version]/TapCache. The file contains partial information on the filename, file path, timestamps, file size, and file content, as shown in Fig. 3.

"FileActivityStoreV3 - S" ([unknown-base64].S) is a *UDRF* of an unknown file format, and its creation principle or rule remains unknown. This file is stored in ~/UsageMetricStore/FileActivityStoreV3/[application]. The filename is encoded in base64. Analysis results reveal that it records information related to file creation and access when files are opened or created. File information is recorded not only on the local system but also on local cloud storage. As shown in Figs. 4 and 5, the filename and file path are stored in this file.

"Aggmru - MRU" (w-mru[number]-[language]-[country]-sr.json) is *UDRF* in JSON file format, and its creation principle or rule remains unknown. This file is stored in ~/aggmru. According to the analysis result, this file records data related to a file accessed through OneDrive (web). The filename is designated based on the country or language, such as 'w-mru[number]-en-US-sr.json' or 'w-mru[number]-ko-KR-sr.json'. The file contains the filename, file path, timestamps, and file size, as shown in Fig. 6.

"Aggmru - REC" (w-rec-[language]-[country]-sr.json) is a *UDRF* in JSON file format for which the creation principle or rule has not yet been studied. This file is stored in ~/aggmru. This file appears to be a temporary log file, and its name and structure may change depending on the version. Based on the analysis results, this file records information about the file when it is downloaded or accessed through Teams (web or app). The filename is specified according to the country or language, such as 'w-rec-en-US-sr.json' or 'w-rec-ko-KR-sr.json'. As shown in Fig. 6, the filename, file path, and timestamps are stored.

"MruServiceCache" (Documents_[language]-[country]) and (Places_[language]-[country]) are *UDRF* files in JSON format that store the files and folders recently opened by the user. This file is stored in ~/MruServiceCache/[unknown]/[application]. These files are recorded when a document is opened on OneDrive (web or app) or when logged in to MS Office. The files are created based on the application-specific folders (Word, PowerPoint, Excel). Inside the files, as shown in Fig. 7, the filename, file path, and timestamps are stored.

"BackstageInAppNavCache" ([FileHash-SHA256].json) is an *SDRF* in JSON file format that has already been studied (ArsenalRecon, 2019). This file is stored in ~/BackstageInAppNavCache. This file is stored in folders for each local system and account that uses MS Office. When the latest version of the MS Office application is run, it displays recent, pinned, and shared documents, and this information is stored in this file. The file contains the filename, file path, timestamps, and file size.

"Office - Recent File" (index.dat) is an *SDRF* in TEXT file format, and it has already been studied. This file is stored in ~/Recent. This file stores references to files used by Microsoft Office and records a list of recently used files. The experimental result has shown that the file

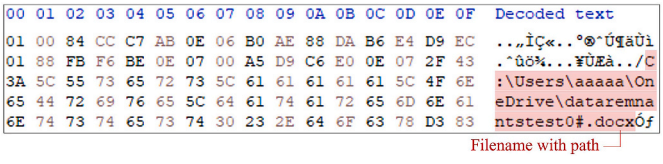

Fig. 2. Example of data remnants, including filename and path, found in the "OfficeFileCache - C4".

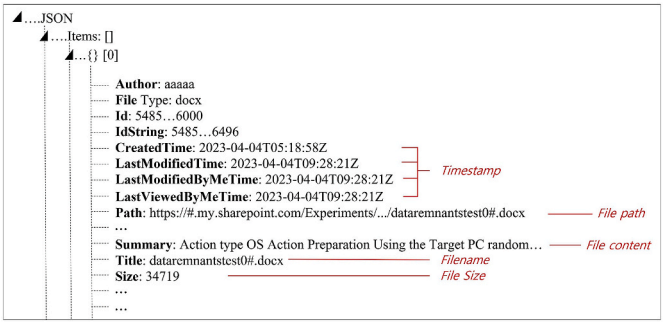

Fig. 3. Example of data remnants, including filename, path, size, content, and various timestamps, found in the "TabCache".

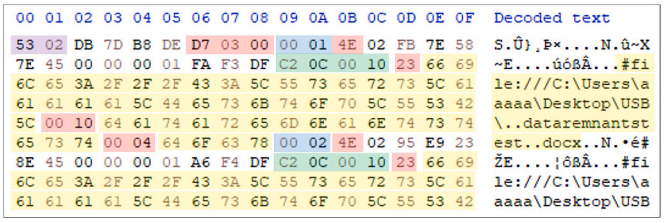

Fig. 4. Example of data remnants, including filename and path, found in the "FileActivityStoreV3 - S".

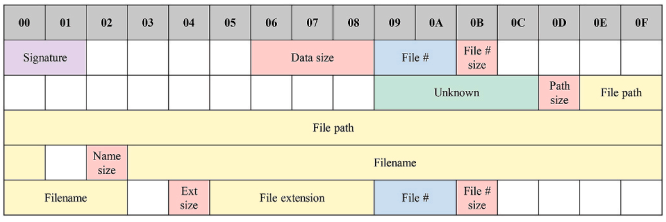

Fig. 5. Internal structure analyzed for identifying data remnants of "FileActivityStoreV3 - S".

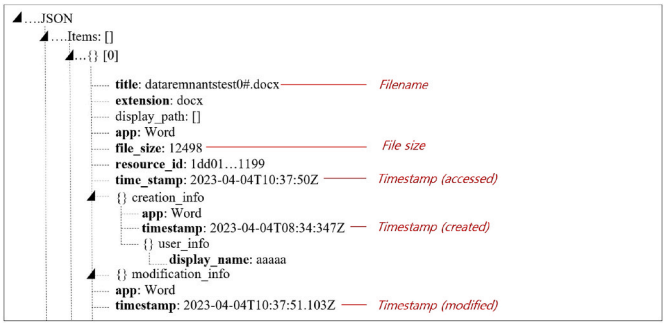

Fig. 6. Example of data remnants, including filename, size, and various timestamps, found in the "Aggmru - MRU and REC".

information is recorded when a file is accessed. The file contains the filename and file path.

5.2. Microsoft OneDrive

OneDrive is a cloud storage service that is a part of the Microsoft 365 suite. OneDrive generates various logs based on user activities, such as creating, downloading, uploading, and accessing files. *DRF* found in this application are stored in %UserProfile%/AppData/Local/Microsoft/OneDrive/settings/Personal or ~/settings/Business.

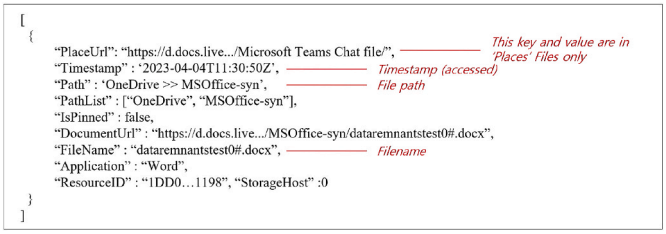

Fig. 7. Example of data remnants, including filename, path, and accessed timestamp, found in the "MruServiceCache".

"OneDrive - Log" ([UserCid].dat) is classified as an *SDRF*. While its format is not officially disclosed by Microsoft, it has been previously analyzed by other researchers (Beercow, 2022; Maloney, 2022; Khatri, 2022). Our analysis shows that it is created whenever a user performs any action related to OneDrive through its web or app interface, including synchronization, downloading, uploading, and accessing files. The file contains the filename and timestamp of each action, as shown in Fig. 8.

"SafeDelete" (SafeDelete.db) is a *UDRF* file in SQLite file format, and it remains unclear how it is generated, or what rules or principles guide its creation. Through experiments, it was found that this file records a list of files in the OneDrive folder. The file contains only the filename, as shown in Fig. 9.

5.3. Microsoft Teams

Microsoft Teams is a collaboration tool that may leave related logs connected to OneDrive. Teams data is stored in the "Teams - Database" ([random].ldb and [random].log). Furkan Paligu analyzed the Microsoft Teams desktop application utilizing IndexedDB storage in Windows (Paligu and Varol, 2022). The author found that information such as accounts, teams, chat messages, and video calls are stored. The downloaded or viewed filename is saved in Unicode (UTF-16 little-endian) encoding. The timestamps for sending and receiving the document are also stored.

In addition, we have found data remnants in two additional *DRF*. These remnants are stored in %UserProfile%/AppData/Local/Packages/MicrosoftTeams/_[random]/LocalCache/Microsoft/MSTeams and ~/MSTeams/Logs. The files, "Launcher - Log" (Launcher_[date]_[timestamp].log) and "MSTeams - Log" (MSTeams_[date]_[timestamp].log) are *UDRF* in the TEXT file format. These specific files have not been extensively studied. These files contain information related to the Teams application, as well as data necessary for its execution. As illustrated in Fig. 10, these log files store data, such as the filename, file path, and timestamps.

"AppSettings" (app_settings.json) is a *UDRF* in JSON file format, which stores Teams application setting information, such as the initial execution time. This file gets recorded when a document is accessed for the last time using the Teams application. The file contains the filename and file path, as shown in Fig. 11.

5.4. Microsoft outlook

Outlook, developed by Microsoft, is a software for email and personal information management. It stores all Outlook email data in the "Mail database" ([email address].ost). This file, formatted as OST, is classified as a *SDRF*. This is located in the path %UserProfile%/AppData/Local/Microsoft/Outlook. Notably, this file records comprehensive information about all sent and received emails, including files attached to these emails. Specifically, this file contains the filename of the attachment files.

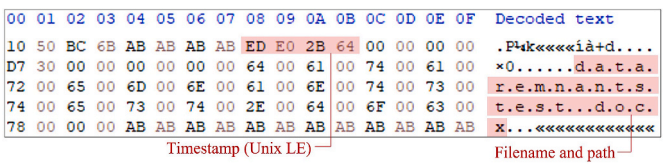

Fig. 8. Example of data remnants, including filename, path, and timestamp, found in the "OneDrive - Log".

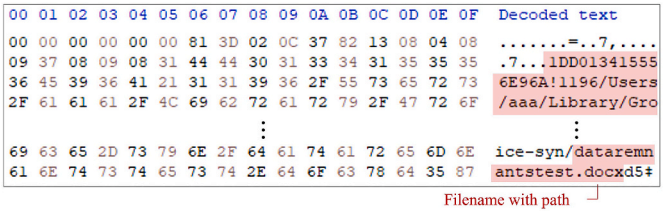

Fig. 9. Example of data remnants, including filename and path, found in the "SafeDelete".

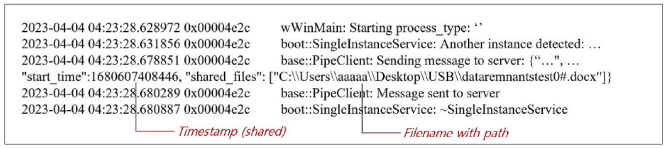

Fig. 10. Example of data remnants, including filename, path, and shared timestamp, found in the "Launcher - Log".

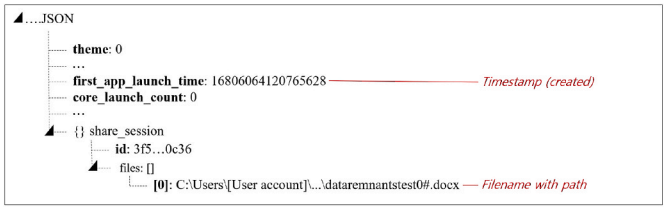

Fig. 11. Example of data remnants, including filename and timestamp, found in the "AppSettings".

5.5. Microsoft OneNote

"Cache - Bin" ([unknown].bin) is a *SDRF* in unknown file format, which stores cache data from the OneNote application. This file is stored in %UserProfile%/AppData/Local/Microsoft/OneDrive/[version]/cache. This file records information about files attached via various functions, such as file printouts, spreadsheets, and tables, across all writable notes in OneNote, including QuickNote, New Section, and individual pages. Most of the filenames and contents attached in OneNote are temporarily recorded in this file.

6. Implementation and experiments

6.1. DRFT: Data Remnants Forensics Tool

DRFT is an implemented tool for data remnants forensics examination based on Python. This tool gathers the deleted file traces from a forensic image as an input source automatically. DRFT performs all stages of the Data Remnants Forensic framework procedure (*DRFI*, *DRFA*, and *DRFE*) except for the dataset creation stage *DRFD*, which is difficult to automate. Fig. 12 shows the overall structure of DRFT, and available on GitHub (Joun, 2023).

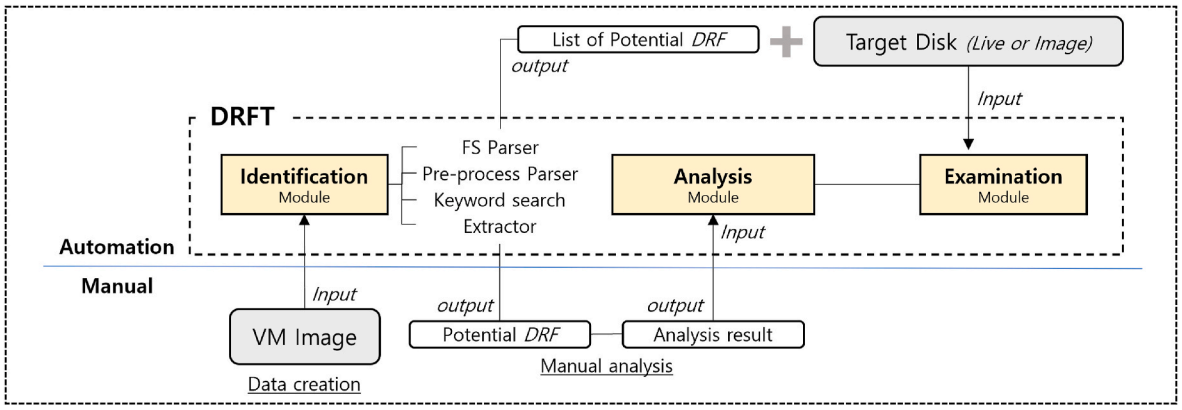

Fig. 12. Design concept of DRFT (Data Remnants Forensics Tool).

6.2. Experimental design: spoliation investigation on a windows system

In this section, a description of the datasets created using the proposed methodology and the corresponding results is provided. The accuracy of the methodology was evaluated by identifying the number of deleted files. The dataset utilized in the experiments can be accessed via a Google Drive link on GitHub. Furthermore, all the data and results used in the experiments have been uploaded to enable validation of the methodology and findings by interested parties. Fig. 13 illustrates the experimental design diagram that outlines the experimental procedure. These experiments were conducted on Windows.

6.3. Dataset creation

Three experimental laptop images were generated, each containing document files that were created and subsequently deleted. The first image is a dataset where sensitive document files were completely erased from a computer environment commonly used by ordinary individuals. The second is a dataset in which all traces or files related to the artifacts listed in the SANS DFIR Windows forensic analysis poster have been deleted (SANS, 2023). The third image was configured to simulate usage by a digital forensic expert with all knowledge of all of *SDRF*. In this experiment, the file system metadata ($LogFile, $UsnJrnl, and $MFT) are excluded, as these could potentially be erased shortly over time. Additionally, the file system metadata, widely known amongst experts, can be easily erased or tampered with, thereby possibly reducing their efficacy in real-world scenarios.

6.3.1. Beginner dataset

The beginner dataset is a forensic disk image that simulates a scenario in which an ordinary user has permanently deleted contentious files. This user did not engage in any anti-forensic activities and solely ensured the permanent deletion of the files.

6.3.2. Intermediate dataset

The intermediate dataset is a forensic disk image that simulates a scenario where an user who knows about digital forensics has attempted to destroy associated evidence. All the traces or files listed in the SANS poster have been deleted. The files deleted are virtual memory (pagefile.sys), Windows Timeline (ActivitiesCache.db), Jump Lists, Prefetch, Registry (MRU), Recent Files, Shortcut (LNK) Files, Office Recent Files, Office OAlerts, IE files (WebCachev*.dat), Windows Search Database (Windows.edb, systemIndex), and web browser data (History, Favicons). Furthermore, “BackstageInAppNavCache” is excluded because it is one of the most well-known artifacts that contains all the accessed Microsoft Office files.

6.3.3. Advanced dataset

The advanced dataset is a forensic disk image that simulates a scenario in which a malefactor with knowledge and expertise in evidence destruction has attempted to erase all traces of their actions on the laptop. All artifacts and logs that fall under the categories of *SDRF* were also removed. Advanced dataset was created to determine whether our proposed methodology can effectively identify traces even in cases where the suspect has attempted to erase all evidence, including those that are difficult for even a digital forensic expert to recognize.

6.4. Evaluation

As there are currently no established frameworks for spoliation investigation, it is challenging to compare the accuracy of our proposed methodology to existing approaches. To address this issue, the method’s ability to identify spoliation trace files in the forensic images was evaluated, and the files that could be identified were categorized.

The number of identified traces in these images are compared using three methods. In Table 4, the entries such as 30/46 and 37/46 represent the results of our experiments. The denominator ‘46’ signifies the total number of files created and subsequently deleted for this experiment.

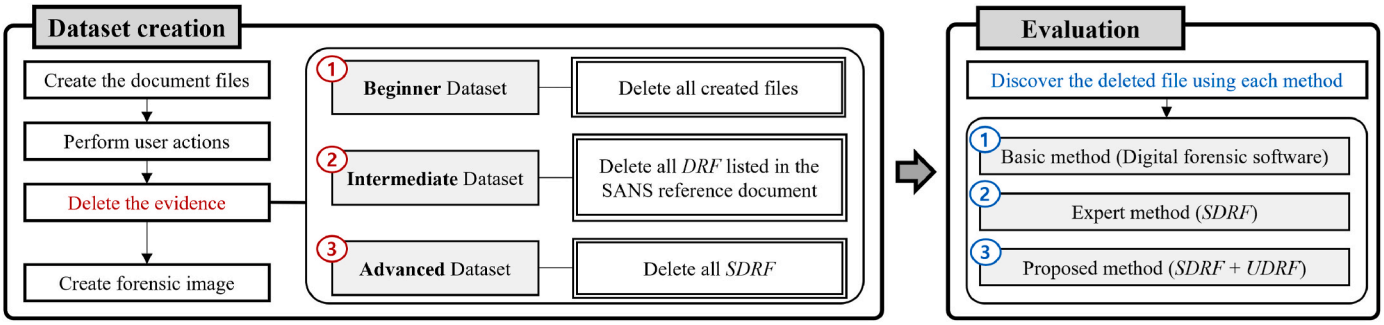

Fig. 13. Diagram illustrating the construction of the three datasets used for experiments and the three evaluation methods applied.

Table 4
Lists of deleted files and the results obtained using each method for each dataset.

Action		Beginner dataset			Intermediate dataset			Advanced dataset		
		Basic (30/46)	Expert (37/46)	Prop (37/46)	Basic (16/46)	Expert (30/46)	Prop (37/46)	Basic (0/46)	Expert (0/46)	Prop (21/46)
Create	Create a file	✓	✓	✓			✓			✓
	Create a file (adding image)	✓	✓	✓			✓			✓
	One Drive (application)	✓	✓	✓	✓	✓	✓			✓
	Save as (source)	✓	✓	✓			✓			✓
	Save as (destination)	✓	✓	✓			✓			✓
	Create shortcut (source)									
	Create shortcut (destination)	✓								
Copy & Paste	External → Internal									
	Internal → Internal									
	Internal → External									
Share	Share file by Mail	✓	✓	✓	✓	✓	✓			
Compress	Compress files (source 1)	✓	✓	✓	✓	✓	✓			
	Compress files (source 2)	✓	✓	✓	✓	✓	✓			
	Compress files (destination)	✓	✓	✓	✓	✓	✓			
Download	One Drive (web)	✓	✓	✓			✓			✓
	Microsoft Teams (application)		✓	✓		✓	✓			✓
	Microsoft Teams (web)	✓	✓	✓		✓	✓			✓
Upload	One Drive (web)		✓	✓		✓	✓			✓
	Microsoft Teams (application)		✓	✓		✓	✓			
	Microsoft Teams (web)		✓	✓		✓	✓			
Open	Double click a file	✓	✓	✓	✓	✓	✓			
	Explorer a file	✓	✓	✓	✓	✓	✓			
	Check properties of a file		✓	✓		✓	✓			
	Open a damaged file	✓	✓	✓	✓	✓	✓			
	One Drive (application)	✓	✓	✓	✓	✓	✓			✓
	One Drive (web - web)	✓	✓	✓			✓			✓
	One Drive (web - app)	✓	✓	✓			✓			✓
	Teams (web) - desktop	✓	✓	✓		✓	✓			✓
	Teams (web) – application	✓	✓	✓		✓	✓			✓
	Teams (application) - desktop	✓	✓	✓		✓	✓			✓
	Teams (application) – application	✓	✓	✓		✓	✓			✓
Modify	Rename a file (source)									
	Rename a file (destination)									
	Change the metadata									
	Modify content	✓	✓	✓		✓	✓			✓
	Modify w/out saving	✓	✓	✓	✓	✓	✓			✓
	Save as PDF	✓	✓	✓	✓	✓	✓			✓
	Save as HTML	✓	✓	✓	✓	✓	✓			✓
	Protect a file									
Email (Outlook)	Send and not delete	✓	✓	✓	✓	✓	✓			
	Send and delete (web)	✓	✓	✓	✓	✓	✓			
	Send and delete (application)	✓	✓	✓	✓	✓	✓			
OneNote	Attach a file in QuickNote		✓	✓		✓	✓			
	Attach a file in new section		✓	✓		✓	✓			
	Attach a file in new page		✓	✓		✓	✓			
	Insert a file as Printout	✓	✓	✓	✓	✓	✓			✓

The numerator, the number preceding '46', represents the count of deleted files that we were able to identify using each respective method. For example, 30 and 37 out of 46 deleted files are identified with different methods.

6.4.1. Basic method

The basic method involves using digital forensic tools to find traces of deleted files. We utilized Magnet AXIOM v6.11.0.34807 and Autopsy v4.19.3, which are capable of analyzing multiple artifacts and identifying traces of files. These tools were employed to identify traces of deleted file through the automatic analysis of artifacts by the tools, rather than through keyword search. In this experiment, AXIOM was configured to analyze all available artifacts, encompassing a total of 252 artifact types. Traces of deleted files were located within nine artifact tabs analyzed by AXIOM: Locally Accessed Files and Folders (V01.log), Email Attachments (OST File), Windows Mail (HxStore.hxd), $LogFile, LNK Files, MRU Opened/Saved Files (NTUSER.DAT), MRU Recent Files & Folders, Windows Search – Document (Windows.edb), and Windows Timeline Activity (ActivitiesCache.db). Autopsy was configured to ingest all files and directories on the target disk. However, traces of deleted files could only be found in the 'Recent Documents tab' (.lnk files). In the beginner dataset, 30 deleted files were found using this method. However, compared to the advanced and proposed methods, a relatively smaller number of files were identified. In the intermediate dataset, most of the well-known artifacts used by the tools were completely deleted, making it possible to identify only 16 deleted files. Furthermore, in the advanced dataset, no traces could be found.

6.4.2. Expert method

The expert method leverages the traces categorized as *SDRF* in Tables 2 and 3. Specifically, this method extracts all *DRFs* identified as *SDRFs* from the target image file and manually finds traces of deleted files within them. Using this method, we were able to identify 37 files in the beginner dataset, which is consistent with the proposed method. In the intermediate dataset, we were able to identify 30 files. These files are *SDRF* which are not listed in the file inventory of the SANS poster. However, in the advanced dataset, which consisted of an image with all *SDRF* removed, not a single deleted file was identified because the expert method relies solely on *SDRF*.

6.4.3. Proposed method

Our proposed methods, which employ both *UDRF* and *SDRF*,

demonstrated superior performance. In all datasets, most of deleted files were still identified, thus validating the effectiveness of our proposed identification methodology. In the beginner dataset, we could not identify nine deleted files. The main reason is that these actions, such as file movement, copying, and name alteration, do not leave any trace except file system metadata. Which is consistent with the performance in the advanced dataset. However, in the intermediate dataset, we achieved the best performance by successfully identifying all 37 deleted files. Finally, in the advanced dataset, we were able to successfully identify 21 deleted files. This validates the strong performance of the proposed method in identifying a substantial number of deleted files, even when all known *SDRF* are completely eliminated.

7. Discussion and conclusion

7.1. Discussion

We now return to the three research questions and discuss their answers as derived from our research.

RQ1: Can the proposed framework identify both existing and previously unknown data remnants?

The framework demonstrated the capability to identify known as well as previously undiscovered *DRF*. Despite the extensive research conducted by numerous scholars in discovering artifacts and logs within the Windows system, the existence of new *DRF* remained uncertain. However, through the framework's analysis, a substantial number of new *DRF* were successfully identified, providing sufficient evidence for identifying deleted files.

RQ2: What types of previously unknown data remnants are discovered?

Although no novel main artifacts such as Registry or Prefetch were identified, the analysis uncovered novel logs and files containing data remnants. Notably, new logs and files associated with Microsoft 365 components, including Office, OneDrive, and Teams, were successfully identified. These files encompassed a diverse range of metadata. The majority of files adhered to known file formats, enabling relatively straightforward analysis of data remnant types and their storage locations through structural examination. However, for files with unknown formats, structural analysis proved challenging. In such cases, an analysis was conducted to determine the file's purpose, the actions recorded by the data remnants, and the minimal structure in which the data remnants were stored.

RQ3: Can the discovered data remnants be used to infer the former existence of files?

Experimental results conducted on three distinct datasets using the proposed method revealed a higher number of traces of deleted files compared to conventional methods. Even in the presence of anti-forensic tools or the deletion of known traces, the analysis method successfully identified a substantial amount of deleted file remnants, thereby enhancing the reliability of the framework being proposed. While finding data remnants does not necessarily imply the existence of such files (in the sense of *sufficient evidence* (Gruber et al., 2023)), the empirical evidence is rather strong.

7.2. Limitations

7.2.1. DRF's volatility

Some identified *DRF* did not contain any data remnants. It is predicted that not all data remnants in a *DRF* are stored permanently. Since most identified *DRF* were temporary log files, these remnants could disappear over time or be overwritten. To assess data volatility, we conducted additional experiments. To overwrite the data remnants in DRF, various user actions (opening, uploading, downloading, and sharing) were performed approximately 20 times each, creating about 100 new files. The following Microsoft 365-related *DRF* lost their data remnants: TapCache - Json, MruServiceCaches, BackstageInAppNavCache – Json, Recent – Index, SafeDelete - db, MSTeams, Cache – Bin. In conclusion, data remnants can disappear depending on the files' characteristics. For example, under the limited experiment conditions, data remnants in LevelDB persist, but the filename is changed. Moreover, the temporary data remnants stored in SQLite's WAL files no longer existed. To verify *DRF* volatility, a systematic process and experiment are necessary.

7.2.2. Automation possibilities

Although a systematic framework has been developed, achieving complete automation is challenging. The *DRFI* phase, involving keyword searches in the input source and *DRF* identification, as well as the *DRFE* phase, focused on extracting and identifying deleted or tampered files from the target PC or image using the analyzed *DRF*, can be automated. However, manual effort is still required for the creation of the dataset in the *DRFD* phase. Furthermore, the analysis of unknown and unexplored *UDRF* necessitates manual file structure and data remnants analysis by investigators or researchers. Therefore, as part of future work, we plan to investigate methods for automating the dataset creation on virtual machines, where users can input the desired OSs and applications for their research.

7.3. Conclusion

The growing importance of digital data as crucial evidence in investigations has led to active research in digital data forensics. However, the easily manipulatable nature of digital data gives rise to ongoing occurrences of evidence tampering crimes. Despite active exploration of digital forensic methods to investigate these crimes, conventional forensic approaches are limited because of rapid technological advancements.

Therefore, a digital forensic framework for analyzing data remnants has been proposed to address these challenges. This framework enables the identification of traces using data remnants, even when files have been completely deleted. A systematic and standardized methodology has been introduced to ensure that all investigators can produce consistent results. This framework can easily be applied to other file types and OS to identify traces.

Experimental results validating the proposed framework demonstrate that it can detect as many traces as possible, even when they are deleted using specialized digital forensic knowledge. Based on these findings, the framework can be applied across diverse fields and prove valuable in real-world investigations. It can also be adopted by organizations globally and will play noteworthy role in digital forensic research and investigative efforts.

Acknowledgements

This work was supported by a Korea University Grant, and also supported by Police-Lab 2.0 Program(www.kipot.or.kr) funded by the Ministry of Science and ICT (MSIT, Korea) & Korean National Police Agency (KNPA, Korea) [Project Name: Research on Data Acquisition and Analysis for Counter Anti-Forensics/Project Number: 210121M07].

References

Apple Inc. v. Samsung Elecs Co., Ltd., 2012, 888 F. Supp. 2d 976, 989 (N.D. Cal. Aug. 21, 2012).

ArsenalRecon, 2019, 2019. Backstage parser. URL. https://github.com/ArsenalRecon/BackstageParser.

Beercow, 2022. OneDriveExplorer. URL. https://github.com/Beercow/OneDriveExplorer.
BMG Rights Mgmt. (US) LLC. v. Cox Communs., Inc., 2018, 881 F.3d 293 (4th Cir. 2018).
Bunting, S., 2016. Forensic analysis of spoliation cases part 2: Windows examinations, eForensics Magazine [Online]. Available: http://www.buntingdigitalforensics.us/publishing.html. (Accessed 13 May 2023).
Cat3, LLC. v. Black Lineage, Inc., 2016, F.Supp, p.488 S.D.N.Y.
Daniel, L., 2011. Digital Forensics for Legal Professionals: Understanding Digital Evidence from the Warrant to the Courtroom. Elsevier.
Durrant, R., 2005. VII. Spoliation of discoverable electronic evidence. Loyola Los Ang. Law Rev. 38 (4), 1803.
Garfinkel, S.L., 2013. Digital media triage with bulk data analysis and bulk extractor. Comput. Secur. 32 (C), 56–72.
Garfinkel, S., Nelson, A.J., Young, J., 2012. A general strategy for differential forensic analysis. Digit. Invest. 9, S50–S59.
Garrie, D.B., 2014. Digital forensic evidence in the courtroom: understanding content and quality, Nw. J. Tech. & Intell. Prop. 12, i.
Gruber, J., Humml, M., Schröder, L., Freiling, F., 2023. Formal verification of necessary and sufficient evidence in forensic event reconstruction. In: Digital Forensics Research Conference Europe (DFRWS EU).
Joun, J., 2023. Data remnants forensic tool. URL. https://github.com/tmpdrft/drft.
Joun, J., Lee, S., Park, J., 2023. Discovering spoliation of evidence through identifying traces on deleted files in macOS. Forensic Sci. Int.: Digit. Invest. 44, 301502.
Kälber, S., Dewald, A., Freiling, F., 2013. Forensic application-fingerprinting based on file system metadata. In: 2013 Seventh International Conference on IT Security Incident Management and IT Forensics. IEEE, pp. 98–112.
Khatri, Y., 2022. Reading OneDrive logs [Online]. Available: https://www.swiftforensics.com/2022/02/reading-onedrive-logs.html. (Accessed 9 July 2023).
Learning Care Group, Inc. v. Armetta, 2016, p.434 (D. Conn. 2016).
Maloney, B., 2022. Recreating OneDrive's folder structure from <UserCid>.dat [Online]. Available: https://www.sans.org/blog/recreating-onedrive-s-folder-structure-from-usercid-dat/. (Accessed 9 July 2023).
Mitchell, I., Anandaraja, T., Hara, S., Hadzhinenov, G., Neilson, D., 2017. Deconstruct and preserve (DaP): a method for the preservation of digital evidence on solid state drives (SSD). In: International Conference on Global Security, Safety, and Sustainability. Springer, pp. 3–11.
Oh, J., Lee, S., Hwang, H., 2021. NTFS Data Tracker: tracking file data history based on $LogFile. Forensic Sci. Int.: Digit. Invest. 39, 301309.
Oh, J., Lee, S., Hwang, H., 2022. Forensic recovery of file system metadata for digital forensic investigation. IEEE Access 10, 111591–111606.
Paligu, F., Varol, C., 2022. Microsoft teams desktop application forensic investigations utilizing indexeddb storage. J. Forensic Sci. 67 (4), 1513–1533.
Quick, D., Choo, K.-K.R., 2013a. Dropbox analysis: data remnants on user machines. Digit. Invest. 10 (1), 3–18.
Quick, D., Choo, K.-K.R., 2013b. Digital droplets: Microsoft skydrive forensic data remnants. Future Generat. Comput. Syst. 29 (6), 1378–1394.
Quick, D., Choo, K.-K.R., 2014. Google Drive: Forensic analysis of data remnants. J. Netw. Comput. Appl. 40, 179–193.
SANS Institute., 2023. Windows forensics analysis [Online]. Available: https://www.sans.org/posters/windows-forensic-analysis. [Accessed 13 May 2023].