STUDENT SESSION

DIGITAL FORENSICS ARTIFACTS OF THE MICROSOFT PHOTOS APPLICATION IN WINDOWS 10

Luka Jovanović,(student)*,
Saša Adamović,(mentor)

[1]Singidunum University,
Belgrade, Serbia

Abstract:
Modern operating systems come equipped with software designed to improve user experience and offer a user-friendly environment while maintaining seamless integration and constant operation times, even when tackling demanding tasks. Applications generate extensive operation logs to aid the diagnostics and maintenance processes. Modern software often relies on the mechanism for cashing and storing additional data that can later be used to access requested resources faster, improving performance. These behaviors often lead to the creation of software artifacts in the form of logs, temporary files, and databases. These artifacts become crucial sources of information in forensic investigations. This paper analyses a set of artifacts created by the Microsoft Photos application that is the default photo viewer and editor in the Windows 10 operating system. Conducted research indicates that a large amount of potentially useful information is created during normal software operation. These fragments can be located and analyzed in files located in system directories. Further exploration reveals a set of valid digital forensic assets. These consist of user action logs, facial recognition identification results, optical character recognition strings acquired from images, metadata, information on devices used to capture the photo, and other information sources.

Keywords:
Digital forensics, Database forensics, Software artifact forensics, Microsoft Photos, Windows 10.

INTRODUCTION

The field of Digital Forensics (DF) has evolved from various forms of abstruse tradecrafts into a staple of modern investigation, examination, and data analysis [1]. Numerous DF tools and approaches are utilized to aid in countless cases daily, and many investigators rely on results acquired using these tools, often without even realizing it [2]. Coupled with these tools, a set of best practices has slowly emerged for tackling specific situations. However, each investigation can pose new challenges and investigators often need to make decisions concerning which actions are best to take on a case-by-case basis [3]. Having a thorough understanding of systems and, in turn, potential sources of digital evidence is a crucial skill through every investigation. Nevertheless, the systems that may find

Correspondence:
Luka Jovanović

e-mail:
luka.jovanovic.191@singimail.rs

DOI: 10.15308/Sinteza-2022-427-434

themselves in front of an investigator keep changing and evolving, as do methods of avoiding detection and obfuscating the evidence. The search for novel sources of digital evidence becomes an ever more crucial component of DF research.

Investigations relying on DF often center around gathering digital evidence from scattered digital forensic artifacts. An equivalent to real word forensic evidence, these artifacts are often left behind unintentionally or unknowingly and often invisible to users. An experienced investigator can gather and analyze artifacts generated by user actions or background processes, to reconstruct incidents and provide valid evidence aiding the investigation. Artifacts can take many forms, filesystem fragments, log files, shell bags, SQL databases (DB) are just some common, noteworthy examples [4]. However, with the ever-increasing size of storage media, the issue of finding artifacts becomes overshadowed by the number of artifacts presented. This in turn manifests as a problem of insufficient time for analysis. Knowing where to look for the right kind of artifact that can serve as evidence in a case is becoming an ever more pivotal ability.

Microsoft's popular operating system (OS) Microsoft Windows, often simply called Windows comprises several graphical OS groups. The most popular of which, at the time of writing, is a version of Windows NT named Windows 10. It remains a dominant competitor in the home and business markets, and as such is probably the most common type of OS seen in DF investigations. A default installation of this OS provides a user with several useful software packs included. These assist the user in common computational tasks, such as word processing, image manipulation, music reproduction, web browsing, and many other functionalities. Many users rely on these default applications through their everyday use. As such, data generated by these applications can present a promising resource when searching for artifacts to guide DF investigations.

A notable example available on almost all machines running Windows 10 is the Microsoft Photos application. Due to the practical integration, as well as user familiarity with the application, few users go out of their way to replace this default software with an alternative. However, the increased random-access memory (RAM) and processor usage, in comparison to other contemporary applications suggest additional functionality present in the application unknown to the ordinary user. A deeper look into the internal structures and files of this application suggests many such background processes being carried out on images. These include location classification, optical character recognition (OCR), and even facial recognition. Additionally, the application maintains logs of user actions, image metadata, and information about the devices used to capture images accessed by the application. In this research paper, the focus is on artifacts generated by this application, as well as their practical applications in digital investigations.

In summary, the main contributions of the conducted research are:

- Presenting the locations, formats, and types of forensic artifacts created by the default Photos application of the Windows 10 operating system
- Demonstrating how user interactions affect artifact generation
- Proposing methods for effective artifact acquisition and analysis
- Exhibiting the forensic significance and user privacy implications of the created artifacts

The remainder of this paper is structured according to the following. Section 2 examines research papers that cover similar topics and gives a summary of the inspiration behind this paper. Section 3 gives an in-depth view of the functions behind the Photos application system. Section 4 covers tools utilized in this study, as well as an extensive examination of the artifacts that can be found inside the application, their significance in forensic investigations. This is followed by a practical example. Finally, a conclusion and future work is given in Section 5.

2. PRELIMINARIES AND RELATED WORKS

When discussing digital investigations, it is impossible not to mention the tools and techniques used when attaining evidence. Many such tools exist, often not constructed to be used in investigations such as disk image creation tools and search tools. Specially designed tools do exist, they are usually custom-built, proprietary, and difficult or expensive to acquire. A well-known open-source alternative used when conducting this research is Autopsy. A DF platform that is equipped with tools found in commercial forensic tools and available free of charge. An additional advantage of this tool is that it allows customization and inclusion of custom community-made modules that extend on the base functionality of the software. With all this in mind, Autopsy offers an excellent set of tools regarding artifact search, as well as DB analysis. It has been used by many researchers in similar DF studies on various applications [5] [6].

In addition to the standard tooling provided in the Autopsy software package, a supplementary open-source software specializing in SQLite. A DB Browser for SQLite is used to better demonstrate database structures and relations in a visually more appealing manner.

Modern OSs offer sleek user interfaces and smooth operating times. However in this process, many artifacts are generated, some of which are valuable when conducting DF investigations. In the case of this research, the Windows 10 OS offers a wealth of artifacts, as a side product of normal software functionality [4]. These artifacts result from operations designed to benefit users and programmers alike. Depending on the nature of the investigation the appropriate approach may vary, data recovery may require additional steps. Sometimes simply proving possession of certain information can lead investigations in the right direction. Filesystems, registries, event logs, and executable files all need to be considered as sources when locating evidence [7]. Mechanisms for prefetching files allow for a smoother user experience by improving response times. This process also generates files that can indicate that a specific program was executed. These files prove an asset when considering anti-forensic tools possibly brought into play by the user [8] [9]. In addition, the Windows Registry, a binary hierarchal DB can provide details on user configurations. It also records user-specific information to better the user experience, essentially providing a record of user-system interaction activities [6].

As open-source relation, DB SQLite provides a relation DB management system (DBMS) often used by developers due to ease of use, accessibility, simple integration, and comparatively small footprint. In contrast to other such DBMS, SQLite does not constitute a client-server model, instead, the system is embedded into the end software. It is strongly favored by web browsers, phone application developers across all platforms, but is also found in many desktop applications, and is amongst the most popular DBMS [10]. Microsoft's version of SQLite Is quartet by Microsoft, and Microsoft frameworks provide SQLite support to assist developers. By architectural design, the entire DB can be contained inside a single file, and atomic commits allow for increased DB integrity. However, this leads to the creation of rollback journals that essentially record transactions before their execution [11]. This paper will not focus on these behaviors, since despite these settings being the default, they can be altered or disabled, making rollback journals an unreliable source. SQLite is considered a good choice for caching and storing data because of its smaller sizes and administration-less capabilities. It is precisely these data caching behaviors that lead to the generation of a plethora of forensic artifacts [12] [13].

With all this in mind, data stored inside an SQLite DB is easily acquirable. The only requirements are a DB browser and the DB itself. Just with these an investigator can query any data values currently stored in the DB. By default, SQLite will not delete records, instead simply marking them as deleted, so the possibility of recovering additional information on deleted files persists [14].

With costs of development and distribution decreasing, as well as the increasing popularity of social media, user-installed applications are becoming a convenient focus when searching for new forensic artifact sources. Third-party software often focuses on cross-functionality, accessibility, and ease of use hence often generating many artifacts. Due to user convenience and familiarity with these applications, communications over such platforms have become exceedingly common. With the rapidly changing landscape and alleged increased message security through encryption or obfuscation, in addition to the often-international nature of data traffic, gathering evidence is becoming somewhat challenging. This has led to illicit activities often taking place over these communication channels as opposed to more established, better regulated, and monitored traditional methods for communication. Investigators often need to rely on the evidence available on the user side rather than information provided by service providers outside jurisdictions. Due to these factors combined, various forensic artifacts of popular communication applications have been analyzed and well documented. However, relying on the user-installed application as a source of evidence can prove unreliable, as these applications tend to drastically vary on a case-by-case basis. A more pragmatic approach involves focusing on applications most likely to be available. This makes default OS applications particularly attractive in terms of research, and some such software has already been undergoing analysis [15]. To the best of the authors' knowledge no such research has yet been conducted on the specific application, Microsoft's Photos, covered in this work.

3. THE MICROSOFT PHOTOS APPLICATION SYSTEM

Introduced in 2012 along with Windows 8, as a successor to the Windows Photo Viewer, the Microsoft Photos application comes preinstalled on machines running the Windows 10 operating system. It is set as the default image viewing tool. It is also available on the windows store for installation under a freemium license. It allows for image viewing, organization, raster editing and is, at user discretion, capable of uploading images directly onto social media platforms, such as Twitter, Facebook, Instagram, and others.

Additional capabilities include photo editing, with functions such as cropping and rotating, color correction, regulating image noise, adjusting red-eye artifacts as well as touching up spots and blemishes. This process created a proxy image during editing, and the changes are only stored upon saving the new image. This allows users to compare edited images to the original, and revert changes. These actions also result in the creation of forensics artifacts in the application SQLite DB, that can be leveraged in digital investigations concerning fraud.

In the default configuration, the Photos application organizes digital images sorting images according to dates. Images can additionally be arranged into albums, which can be either user-created or automatically generated. To accomplish this the application runs background processes to attain metadata and classify images accordingly. Residual artifacts of these results are stored inside an SQLite DB that will be covered in more detail in the following section.

4. ARTIFACTS IN THE PHOTOS APPLICATION

This study covers easily accessible software data, generated by the everyday functioning of the Microsoft Photos application. No alterations to the application or the OS have been made unless explicitly stated otherwise.

4.1. LOCATIONS AND FILES OF INTEREST

Locations of files and directories of notes in the conducted research are given in Table 1: Notable file directories. The directories shown assume a default installation, on the C:\ drive, values for %VERSION% varies according to the installed version, and the variable %USERNAME% depends on the account username.

4.2. THE MEDIADB.V1 SQLITE DB

The primary focus of this research, and proposed source of forensic data, the MediaDb.v1 file is an SQLite DB file. While access to the DB file requires an administrative password, the database itself is unencrypted, making it easily accessible for investigators.

The DB itself has an extensive structure consisting at the time of writing of a total of 128 tables. The database contains extensive data on user actions, access logs, directories containing images, image formats, hash tables, image analysis results, location data, OCR, facial recognition features, and classifications. This research will only cover in detail data considered most interesting and relevant to forensic investigations, while the remaining data will only be presented as a summary.

The material covered includes:

- Data concerning user actions, application execution, image alterations, printing, user searches, and image sharing
- Image geographic location data
- Information concerning devices used for image capture
- OCR data extracted from images, and the generated data's when usability concerning personal documents
- Facial recognition data

Name	Path
Microsoft.Photos	C:\Program Files\WindowsApps\Microsoft.Windows.Photos_%VERSION%_x64__8wekyb3d8bbwe\
Settings	C:\Users\N3cr0\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\Settings\
MedaiDB.v1	C:\Users\%USERNAME%\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\LocalState\

Table 1: Notable file directories

Additionally, it is worth noting that, by default, the data contained in the DB remains after images have been removed, and as such persists after file deletion or encryption. As this DB is often overlooked by file shredding applications, and users obfuscating their actions, it can prove a valuable source of evidence in investigations where data was destroyed. Using the data generated by the Photos application and permanently stored in this database could play a valuable role as a sort of pseudo recovery. It could become a source of valid evidence where none would be available.

In this research, a practical example of a database has been extracted from a researcher's computer using well-known forensic practices and methods. An image of the machine was created using tools available in Windows 10. Following this, the image was loaded into Autopsy, and the database was extracted from the appropriate source. SQLite browser is then used to explore and document findings. This procedure is repeated several times, to document how user behavior and setting influence artifact generation on a system.

The database primarily contains information on accessed file locations, folders used to store images, such as the photos and downloads folder as well as additional user-created folders. The purpose of this is to help the application quickly locate images when presenting them to the user. These tables are updated and indexed every time a user relies on the search functionality of the application. New folders are added, and image locations updated. However, as previously stated, deleted images are not removed from the table.

These simple indexing directories can help investigators locate images, and possibly narrow down the search when looking for illicit content. Image access times are also contained in these tables, so finding recently accessed files is trivialized.

4.2.1. User action data

Extensive data on user actions is maintained inside the DB. This includes data on software launches, entry points, as well as timestamps of when the events have taken place. Data is kept on searches performed by the user, including search box text used in the search, numbers of results recovered, and timestamps of these actions are saved. Additional actions logged are user views and slide shows, data in these tables cover entry points, and timestamps. Print and share events are also tracked similarly. Share events including the data on the targeted platform of distribution, as well as a column storing data on whether the shared attempt was successful. The table labeled sqlite_sequence contains the total numbers of specific actions users have taken, in terms of views, prints, slideshows, etc.

Data in these tables is significant, as it provides definitive logs tying user access to images. In cases where the distribution of illicit material is deemed relevant, print, and share events could help investigators determine when and if reproduction and distribution have taken place. Search results and camera device data can be used to track down image origin points even when images are no longer directly available.

4.2.2. Image geographic location data

Image metadata is, when available, also processed, and the country, region, and district of where the photo was taken are all recorded. Additionally, regardless of metadata availability, the images are processed and classified, and appropriate tags are applied during processing. At the time of writing the Tag Variant table, responsible for storing all the possible classifications of images, consists of 500 entities. These include airplane, celebration, child, document, drawing, car, vehicle, night, newborn, nature, landmark. While the full list includes far too many entries to list here, the reader is encouraged to explore the DB at their own accord for further details.

It is abundantly clear how these automatically applied classification tags can help narrow down investigator search areas when looking for sources of evidence. As previously stated, when images are deleted or encrypted, the information contained in these tables can offer clues as to the original contents. The classifications the application performs are quite extensive and can prove an asset to any investigation.

4.2.3. Image capture device data

Tables CameraModel and CameraManufacturer contain data on the make and model of devices used to capture images accessed by the application. This extends to mobile phones, and device manufacturers and exact models can be recovered from these tables. Useful when tracking the origin of images.

4.2.4. OCR data

During the image search and indexing process, OCR is performed on images, they are assigned a text tag and resulting strings of text are stored inside the DB. This makes images searchable by the text they contain, also leading to the formation of very useful forensic artifacts. Entries of interpreted text taken from images are created in the DB and associated with the image. This in turn means that despite possible images of documents being deleted, removed, or encrypted, their contents remain stored in the DB.

The performance of the OCR algorithm varies. The main limitation is poor performance when processing foreign languages and writing systems. However, it is reasonably good when processing printed English text and performed consistently well when handling printed digits. In the conducted, presented in following sections, tests most information present on publicly available samples of user ID cards and driver licenses were fully recoverable from the DB. This kind of approach can prove fruitful in cases concerning identity theft, fraud, or any cases dealing with access to sensitive documents of any kind. This is a major cause for concern when considering privacy and security. Recently more services are requiring registration using images of valid government-issued documents. It is worth considering that data on the images processed by the default image viewing application will persist after the files are removed from the machine.

4.2.5. Facial Recognition data

Unlike the other findings presented in this paper, facial recognition is not enabled by default and needs to be user enabled. However, when facial recognition is enabled, a plethora of information is generated and stored. Once enabled, like the other processes mentioned here, facial recognition is carried out in the background. Face features are extracted. Faces are also classified according to expression and position. In addition to this, faces are associated with persons in the DB, and the probability of each face belonging to each person is calculated. Should a person be confirmed by a user when classifying photos, their name and, when available email, are stored in the DB. The application keeps track of the number of occurrences of each person available in the dataset.

Using data available on faces and persons can prove an asset in investigations. This data can be used to prove association or possession of images containing certain faces. However, with this feature not enabled by default, facial recognition data is not as reliable of a source as the other sources listed in this work.

4.3. A PRACTICAL EXAMPLE

For a practical demonstration, a total of 5 publicly available examples of personal documents, namely driver's licenses were taken from public sources and loaded into the photos application. After a few minutes of software use, browsing images, indexing and processing were completed in the background, the database is again loaded into appropriate software, and analysis is repeated.

The results attained from OCR varied based on image composition and quality. They were nevertheless sufficient to demonstrate that it is possible to fully recover personal information from data generated by the photos application. Names, dates of birth, identification numbers, expiration dates, and other data visible on the documents were later fully recovered from DB tables. Additionally, facial recognition ID can be matched if additional photos of persons are available. Examples of the attained results can be seen in Figure 1: Personal document samples and OCR artifacts recovered from them. The documents used were acquired from official government websites [16] [17] respectively. These are sample documents meant to mirror legitimate documents while not compromising individual private information. Below the images that were used in this example are the OCR items recovered from the database. As shown, ID numbers, names, dates of birth, date issues, and expiration dates were all recoverable from the OCR artifacts.

The procedure was also repeated on an example of a local document, a government-issued ID card. Results attained from OCP were less impressive, due to language barriers. However, digit data was fully recovered as well as facial recognition data.

Additional tests were performed where Face IDs were able to be matched between images on the ID cards, and images of the individual taken at a different time. These gave consistent promising results. However, due to the sensitive nature of the data presented in these documents results are not practically shown.

Figure 1- Personal document samples, and OCR artifacts recovered from them

5. CONCLUSION

As anti-forensic and encryption tools become exceedingly common, investigators need to continuously strive to improve on and look for new sources of evidence often overlooked by end-users. In cases where large amounts of files are available and need to be processed quickly, the information on user actions taken on files provided by these sources can point investigators in the right direction, conserving resources and precious time. In cases where images were changed or removed, evidence of these actions can be gathered from temporary and DB files created by the internal mechanisms of the Photos applications. In investigations dealing with removed or encrypted images, additional data and information are contained in them. This data bay takes the form of OCR results, and facial recognition IDs can be recovered from a DB that maintains this information in the default configuration even after the files have been encrypted or removed. The conducted research presented in this paper takes a DF approach to examine the Microsoft Photos application. To the best of the authors' knowledge, it is the first academic work addressing forensic artifacts generated by the Microsoft Photos application on a Windows 10 OS. Proposals are made on how and where these artifacts can be located, and their forensic significance is additionally elaborated on. Furthermore, the approach utilizes open-source tools, demonstrating that this kind of investigation can be carried out without overly specialized and often inaccessible, expensive tools and reasonable computational resources.

In future works, we aim to extend methodologies applied in this paper, looking at other available applications that may provide valid evidence often overlooked by tools attempting to obfuscate it. Also, we may introduce tools that streamline the process of data acquisition, further reducing investigation requirements in terms of time and resources.

6. REFERENCES

[1] S. L. Garfinkel, "Digital forensics research: The next 10 years," *digital investigation*, vol. 7, pp. 64-73, 2010.

[2] M. Dan, C. Anna, S. Ramos, G. Alain, K. Matthew and T. Jeremy, "Is the open way a better way? Digital forensics using open source tools," in *40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, 2007.

[3] C. Arumugam and S. Shunmuganathan, "Digital Forensics: Essential Competencies of Cyber-Forensics Practitioners," in *Advances in Machine Learning and Computational Intelligence*, Springer, 2021, pp. 843--851.

[4] C. Altheide and H. Carvey, Digital forensics with open source tools, Elsevier, 2011.

[5] M. N. Yusoff, A. Dehghantanha and R. Mahmod, "Forensic investigation of social media and instant messaging services in Firefox OS: Facebook, Twitter, Google+, Telegram, OpenWapp, and Line as case studies," in *Contemporary digital forensic investigations of cloud and mobile applications*, Elsevier, 2017, pp. 41--62.

[6] H. Kwon, S. Lee and D. Jeong, "User profiling via application usage pattern on digital devices for digital forensics," *Expert Systems with Applications,* vol. 168, p. 114488, 2021.

[7] P. Lewulis, "Digital forensic standards and digital evidence in Polish criminal proceedings. An updated definition of digital evidence in forensic science," *International Journal of Electronic Security and Digital Forensics*, vol. 13, pp. 403-417, 2021.

[8] M. Ölvecký and D. Gabriška., "Wiping techniques and anti-forensics methods," in *2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)*, 2018.

[9] J. P. A. Yaacoub, H. N. Noura, O. Salman and A. Chehab, "Digital Forensics vs. Anti-Digital Forensics: Techniques, Limitations and Recommendations," *arXiv preprint arXiv:2103.17028*, 2021.

[10] K. Kolonko, Performance comparison of the most popular relational and non-relational database management systems, 2018.

[11] D. Pawlaszczyk and C. Hummert, "Making the Invisible Visible--Techniques for Recovering Deleted SQLite Data Records," *International Journal of Cyber Forensics and Advanced Threat Investigations*, vol. 1, pp. 27-41, 2021.

[12] H. Yaoyi, H. Huiqi, Z. Xuan and Z. Aoying, "SQLite-CC based on non-volatile memory cache," *Journal of East China Normal University (Natural Science)*, vol. 2021, p. 124, 2021.

[13] L. Andrade and M. Antonio, "Digital Forensic Artifacts Of SQlite-Based Windows 10 Applications" 2021.

[14] S. Nemetz, S. Schmitt and F. Freiling, "A standardized corpus for SQLite database forensics," *Digital Investigation*, vol. 24, pp. 121-130, 2018.

[15] P. Domingues, M. Frade, L. M. Andrade and J. Silva, "Digital forensic artifacts of the Your Phone application in Windows 10," Digital Investigation, vol. 30, pp. 32-42, 2019.

[16] "New York Department of Motor Vehicles," [Online]. Available: https://dmv.ny.gov/id-card/sample-photo-documents. [Accessed 07 January 2022].

[17] "Republic of Serbia Ministry of Interior," [Online]. Available: http://www.mup.gov.rs/wps/portal/sr/gradjani/dokumenta/Licna+karta. [Accessed 07 January 2022].