Extended Abstracts

Recycling Personal Data: Data Reuse and Use Limitation in Digital Forensics (Extended Abstract)

Bart Custers

Faculty of Law, eLaw, Centre for Law in the Information Society,
Leiden University, Steenschuur 25, 2311 ES Leiden The Netherlands
b.h.m.custers@law.leidenuniv.nl

Keywords: Big Data · Risk profiling · Digital forensics · Personal data · Privacy · Data protection · Data reuse · Data recycling · Use limitation principle

The use of Big Data offers tremendous potential in many fields, including digital forensics. A typical example of this is the use of Big Data for risk profiling. Big Data analysis with data mining tools opens the possibilities of finding previously unknown patterns and relations in the data. Such patterns may be useful to identify individuals, to reveal networks in which people are involved and to build dossiers on suspects or potential suspects. This may be useful in solving crime, but it may also be useful in preventing crime. By using Big Data, characteristics may be discovered indicating specific risks of individuals committing a crime or getting involved in crime, identifying situations that may result to crime and addressing groups that may be 'at risk'. Whereas in most forensics there is a lack of data or samples, in Big Data contexts an overload of data exists. This further enables the use of intelligence before a crime takes place as opposed to the more traditional use of forensics after a crime has taken place.

Of course, bigger is not always better. Not all types of data are useful for all types of purposes. But the suggestion that in forensics only data of offenders is required is a misunderstanding. Typically, in order to detect patterns in deviancies (criminal behavior), data of normal people and normal situations is required. In other words, in order to find the 'exception', the 'standard' has to be known. As a result, risk profiling in digital forensics often also requires data of people who do not have criminal records and of people who have not committed a crime.

Obviously, there are also some pitfalls in the use of risk profiling. Because the results heavily depend on the available data, there may be results that qualify as self-fulfilling prophecies. For instance, when police forces only collect data in particular ethnic minority neighborhoods, the risk profiles resulting from data analyses might show that ethnic minorities are more prone to getting involved in crime, whereas a broader, richer dataset might reveal different results. Obviously, such mistakes may also result in stigmatization and discrimination. At the same time, the use of Big Data in risk profiling may increase the objectiveness of risk profiles. For instance, in cases

J.I. James and F. Breitinger (Eds.): ICDF2C 2015, LNICST 157, pp. 209–211, 2015.
DOI: 10.1007/978-3-319-25512-5

where a particular police officer may have a prejudice, large datasets may prove this perception is wrong. In such cases, it may be helpful to select surveillance areas and people to be searched by law enforcement on the basis of risk profiles rather than by the sole discretion of an individual police officer.

Although there are nowadays many tools that automatically analyze Big Data, such as data mining tools, there is always the risk of data overload or, to be more specific, the risk that human intuition provides insufficient insight in the datasets to choose the right tools for analyses. Data visualization tools may be helpful to counter this.

Another issue is the reliability of data and the resulting risk profiles. Datasets may contain errors or may be incomplete. Risk profiles may contain false positives (e.g., not all Muslims with big black beards taking flight lessons are terrorists) and false negatives (e.g., Muslim terrorists are not the only type of terrorists). However, also in this case, Big Data may yield better results, enabling filtering out data errors, filling gaps in datasets and making more precise predictions.

Apart from the 'garbage-in-garbage-out' argument for datasets, there are also plenty of pitfalls in interpreting datasets and data analysis results. For instance, typical errors may be so-called confirmation bias, i.e., the tendency to search for, interpret or recall information in a way that confirms beliefs or hypotheses. Another mistake may be that statistical relations are interpreted as causal relations or illusory correlations, when people may falsely perceive an association between two events.

Despite these potential pitfalls, there are many benefits of Big Data in forensics. Adding these benefits the question comes to mind why data cannot be reused more often. This may be due to potential unwillingness to share data and to economic, technological, ethical and legal restrictions. In this paper we focus on data protection law. The current European legal framework for personal data protection is based on the idea that there are limits to the collection of personal data (the so-called *collection limitation principle*), that data controllers collect data only for purposes specified in advance (the so-called *purpose specification principle*) and that the data collected are only used for those purposes specified (the so-called *use limitation principle*). Particularly the use limitation principle is relevant in data recycling, since it intends to prevent function creep. The idea behind this principle is obviously to set expectations, especially for data subjects, who have to decide whether or not to provide their personal data in specific contexts and when consenting to the ways in which their personal data may be used.

Data reuse may be widened by renewing models for informed consent. Within the current legal framework, personal data may be anonymized to allow broader use or data subjects may be asked for their consent regarding data reuse. Current practice is to formulate the purposes for data use in very broad ways, encompassing many different types of data use, making consent for data reuse unnecessary. However, limited transparency about the ways in which data are used may be the result. Another approach may be to change the current legislation so that data reuse in some contexts is allowed without bothering data subjects with frequent requests for consent of data reuse. Obviously the question here is when data reuse is close enough to the original consent to assume implicit consent for data reuse and when data reuse should be consented to explicitly. These are just some of the directions in which new solutions can be sought.

This project on modeling the European data economy (EUDECO) is funded by the Research and Innovation Framework Program Horizon 2020 of the European Commission (Grant Agreement No. 645244). The project will assist European science and industry in understanding and exploiting the potentials of data reuse in the context of Big Data and open data, including cloud computing.

Reliability Research about Evidence Acquisition Method of Apple Mac Device (Extended Abstract)

Jisung Choi, Dohyun Kim, and Sangjin Lee

Center for Information Security Technologies (CIST), Korea University,
Anam-Dong, Seongbuk-Gu, Seoul Republic of Korea
{chjs207, exdus84, sangjin}@korea.ac.kr

Abstract. Apple Mac devices, which are increasing in global usage, are more likely to be encountered during a digital investigation and may contain digital evidence. Unlike other devices, Apple devices have unique interfaces and operating systems. For such reasons, digital investigators may have difficulty when they attempt acquire digital evidence from such devices. Further, few reliability research on acquisition method from Apple devices have been conducted. Further verification of the reliability of acquisition methods are needed to supplement existing Apple Mac Device acquisition procedures. This paper describes an acquisition method for volatile and non-volatile data on an Apple Mac device that includes the verification of the reliability of the acquisition method.

Keywords: Apple Mac Device · OS X · Acquisition · Reliability · Digital Forensics

1 Introduction

Apple Mac Device is product which globally used. Mac Device uses unique interface and operating system such as OS X, FireWire, and Thunderbolt. These things can be hard to investigate at Mac Device.

This paper classifies evidence acquisition methods according to volatility of evidence (volatile and non-volatile) to make generally reliable evidence acquisition procedure. And this paper describes an existing research about each evidence types. Volatile data acquisition section describes methods about dumping physical memory at PC of investigation and acquisition data by using operating system command. Non-volatile data acquisition section describes method about storage imaging. And we describe limitations of each method and verify reliability through experiments.

2 Related Work

Evidence acquisition methods of Apple Mac Device have been conducted in most part. But there is no a reliability research about each acquisition methods. To acquire Volatile data, acquisition methods that dump physical memory at outside has been used

J.I. James and F. Breitinger (Eds.): ICDF2C 2015, LNICST 157, pp. 212–214, 2015.
DOI: 10.1007/978-3-319-25512-5

through DMA vulnerability of hardware interface [1]. Pyfw library which use DMA vulnerability of FireWire is opened [2]. And physical memory acquisition research which adopted DMA vulnerability conducted at OS X Lion [3]. Acquisition method which execute on Host PC has been researched by using KVM (Kernel Virtual Memory) [4]. And research which extracts major data from acquired physical memory conducted [5]. Acquisition methods about non-volatile data of Mac Device are conducted Macintosh imaging [6].

3 Acquisition Method and Reliability Verification

Target is Volatile data and Non-volatile data. To acquire volatile data, an investigator must dump physical memory and acquire information through system command. For acquisition physical memory, investigator use FireWire. And they use acquisition software on Host PC. If investigator can't access directly physical memory, they use sleep image or system command. To acquire Non-volatile data, they do storage imaging. These methods are storage separation, using bootable OS, and Target Disk Mode.

Acquisition method with Inception makes kernel panic before 4GiB of physical memory. Thus, we get 3GiB ∼ 3.5GiB physical memory which is below 4GiB of specification. And OSXPmem do normal action on Yosemite (OS X 10.10). And we find Inception minimally affect physical memory during acquisition process of physical memory.

We compare results of MD5 Hash about storage separation, bootable OS, and target disk mode. And we find 3methods can reserve integrity of evidence. But if investigator don't use Write Blocker at target disk mode, evidence can be changed at allocation, catalog, volume header, and alternate volume header. Lastly, we find Bootable OS is the fastest method.

4 Conclusion

This paper describes an existing Mac Device evidence acquisition method which classify by volatile characteristic of evidence. And we experiment about reliability of each method. This paper can be used by investigator as standard guide-line when they does digital forensics about general Mac Device.

References

1. Carrier, B.D., Grand, J.: A hardware-based memory acquisition procedure for digital investigations. Digital Invest. **1**(1), 50–60 (2004)
2. Becher, M., Dornsief, M., Klein, C.N.: FireWire, All your memory are bleong to us. CanSecWest, Vancouver (2005)

3. Mac OS Lion Forensic Memory Acquisition Using IEEE 1394. http://www.frameloss.org/wp-content/uploads/2011/09/Lion-Memory-Acquisition.pdf
4. Singh, A.: Mac OS X Internals. Pearson Education, Boston (2006)
5. Lee, K., Lee, S.: Research on Mac OS X physical memory analysis. J. Korea Inst. Inf. Secur. Crypt. **21**(4), 89–100 (2011)
6. McDonald, K.: To image a Macintosh. Digital Invest. **2**, 175–179 (2005)

Forensic Analysis Using Amcache.hve (Extended Abstract)

Moonho Kim and Sangjin Lee

Center for Information Security Technologies (CIST), Korea University,
Anam-Dong, Seongbuk-Gu, Seoul Republic of Korea
firstkmh81@gmail.com, sangjin@korea.ac.kr

Abstract. Amcache.hve is a registry hive file related to the Program Compatibility Assistant, which stores the execution information of application software. Amcache.hve records the execution path of executable files and the time they are first executed. The utility can also be used to estimate when they were first installed and when they were deleted. Using these features, Amcache.hve can be used to draw up overall timelines of application use when used with the Prefetch and Iconcache.db files. Amcache.hve is an important utility for tracking the activities of anti-forensic programs, portable programs, and external storage devices. This paper illustrates the features of Amcache.hve and how it is used in digital forensics, such as when estimating application deletion times.

Keywords: Digital forensics · Amcache.hve · User behavior

1 Introduction

In digital forensic investigation, tracing application execution history is vital. Examining the execution history of applications enables detection of the use of anti-forensic techniques, and a determination of criminal intent.

In order to trace execution history, one can analyze Prefetch files or Iconcache.db files. Analyzing Prefetch files is limited to the number of Prefetch files available, 128 [1]. Analyzing Iconcache.db files is also limited in that it cannot supply application execution times [2]. On the other hand, analyzing Amcache.hve overcomes the limitations of the Prefetch and Iconcache.db files. It enables analysts to draw up general application execution timelines. It can also trace anti-forensic application use, even when portable applications have been used.

Currently, analysis of Amcache.hve is nonexistent except in Yogesh Katri's personal blog. The blog explains the recording in Amcache.hve of application first execution times, execution paths, SHA-l hash values, product names, and file versions [3]. This paper suggests how it could be used in digital forensics.

J.I. James and F. Breitinger (Eds.): ICDF2C 2015, LNICST 157, pp. 215–216, 2015.
DOI: 10.1007/978-3-319-25512-5

2 Methods for Utilization in Digital Forensics

Amcache.hve is a registry hive file, and “Last Written Time” is saved on each executable file key. This timestamp is created when the executable files are executed for the first time. It is only created upon initial execution and cannot be changed after. Using this feature, the first execution time of the file can be checked.

In the case of package applications, particularly installation files that are installed in the control panel for example, many executable files are tied into a package, allowing the first execution time of each executable to be checked. What the first execution time of each executable file means is described below, using the BCWipe program as an example.

1. bcwipeSetup.exe: install time (“Created Timestamp”: this corresponds to the time the exe file was created by being downloaded from internet or copied from an external drive)
2. BCWipe.exe: the first execution time
3. BCUnInstall.exe: deletion time

This will allow us to identify time information regarding the creation, installation and deletion of the BCWipe program.

In addition, it can trace anti-forensic programs and portable program execution histories, through file paths.

3 Conclusion

Amcache.hve, which is a new artifact in Windows 8, holds varied application information, particularly time information, which is especially important in digital forensic investigations.

Utilizing Amcache.hve in conjunction with the Prefetch and IconCache.db file allows the drawing up of an overall timeline of application execution, confirming initial installation time, how many times and the last time of execution, deletion time and even the number of installations. Moreover, analyzing Amcache.hve can allow the identification of traces of even portable programs, which are commonly employed to hinder forensic investigation. Therefore, Amcache.hve can be very useful in digital forensic investigations.

References

1. MSDN. Misinformation and the The Prefetch Flag. http://blogs.msdn.com/b/ryanmy/archive/2005/05/25/421882.aspx
2. Lee, C.-Y., Lee, S.: Structure and application of IconCache.db files for digital forensics. Digital Invest. **11**(2), 102–110 (2014)
3. Khatri, Y.: Amcache.hve in Windows 8-Goldmine for malware hunters. http://www.swiftforensics.com/2013/12/amcachehve-in-windows-8-goldmine-for.html

Author Index