*Journal of The Korea Institute of Information Security & Cryptology* ISSN 1598-3986(Print)
VOL.25, NO.3, Jun. 2015 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2015.25.3.573

# 앰캐시(Amcache.hve) 파일을 활용한 응용 프로그램 삭제시간 추정방법*

김 문 호,† 이 상 진‡
고려대학교 정보보호대학원

# Method of estimating the deleted time of applications using Amcache.hve*

Moon-Ho Kim,† Sang-jin Lee‡
Center for Information Security Technologies, Korea University

요 약

앰캐시(Amcache.hve) 파일은 프로그램 호환성 관리자(Program Compatibility Assistant)와 관련된 레지스트리 하이브 파일로 응용 프로그램의 실행정보를 저장한다. 이 파일을 통해서 응용 프로그램의 실행경로, 최초 실행시간을 확인할 수 있을 뿐 아니라, 삭제시간까지 추정할 수 있다. 응용 프로그램의 최초 설치시간 및 삭제시간까지 확인할 수 있기 때문에 프리패치(Prefetch) 파일, 아이콘캐시(Iconcache.db) 파일 분석과 병행하면 응용 프로그램의 전체적인 타임라인을 구성할 수 있다. 또한, 앰캐시 파일은 안티포렌식 프로그램, 포터블 프로그램 및 외장저장장치 흔적을 기록하고 있어 디지털 포렌식 관점에서 중요한 아티팩트이다.

본 논문에서는 앰캐시 파일의 특성과 응용 프로그램 삭제시간 추정 등 디지털 포렌식 기술로서의 활용방안을 제시한다.

ABSTRACT

Amcache.hve file is a registry hive file regarding Program Compatibility Assistant, which stores the executed information of applications. With Amcache.hve file, We can know execution path, first executed time as well as deleted time. Since it checks both the first install time and deleted time, Amcache.hve file can be used to draw up the overall timeline of applications when used with the Prefetch files and Iconcache.db files. Amcache.hve file is also an important artifact to record the traces of anti-forensic programs, portable programs and external storage devices. This paper illustrates the features of Amcache.hve file and methods for utilization in digital forensics such as estimation of deleted time of applications.

**Keywords:** Digital forensics, Amcache.hve, User behavior

## I. 서 론

디지털 포렌식 수사 시 응용 프로그램의 실행 흔적을 확인하는 것은 매우 중요하다. 응응 프로그램의 실행 흔적을 확인함으로써 안티포렌식 기술 사용 여부 및 범행의도 등을 파악할 수 있기 때문이다.

실행 흔적 정보를 확인하기 위한 방법으로는 프리패치(Prefetch) 파일, 아이콘캐시(Iconcache.db) 파일을 분석하는 방법 등이 있다.

프리패치 파일은 윈도우 운영체제에서 응용 프로그램 실행 시 메모리에 로드되는 코드와 데이터를 파일로 생성한 것으로, 이를 분석하면 응용 프로그램의 명칭, 실행 횟수, 마지막 실행시간 등의 정보를 확인할 수 있

접수일(2014년 12월 15일), 수정일(2015년 3월 5일), 게재확정일(2015년 4월 6일)
* 이 논문은 2014년도 정부(미래창조과학부)의 재원으로 한국연구재단-공공복지안전사업의 지원을 받아 수행된 연구임(2012M3A2A1051106)
† 주저자, firstkmh81@gmail.com
‡ 교신저자, sangjin@korea.ac.kr(Corresponding author)

다[9]. 하지만 프리패치 파일은 최대 128개로 제한되어 있으며, 초과 시에는 제일 오래된 프리패치 파일부터 삭제되어 디지털 포렌식 관점에서 일정한 한계가 있다.

아이콘캐시 파일은 응용 프로그램의 아이콘 이미지를 BMP 파일 형태로 기록하며, 응용 프로그램의 설치, 복사, 열람 행위에 대한 정보를 확인할 수 있다[10]. 하지만 응용 프로그램의 실행시간에 대한 정보를 확인할 수 없어 이 또한 디지털 포렌식 관점에서 일정한 한계가 있다.

한편, 윈도우 8 운영체제가 출시되면서 새롭게 추가된 앰캐시(Amcache.hve) 파일은 응용 프로그램 실행 이후에 남는 로그와 같은 정보이며, 이를 통해서 해당 응용 프로그램의 최초 실행시간, 삭제시간 등 시간정보를 확인할 수 있다.

앰캐시 파일을 분석하면 프리패치 파일에서 확인할 수 없는 응용 프로그램의 최초 설치시간과 오래된 응용 프로그램의 사용흔적을 확인할 수 있고, 아이콘캐시 파일에서 확인할 수 없는 시간정보를 확인할 수 있으며, 삭제시간까지 추정할 수 있기 때문에 응용 프로그램의 전체적인 타임라인을 구성할 수 있다.

또한, 안티포렌식 응용 프로그램 사용 흔적을 확인할 수 있고, 포터블 응용 프로그램과 CCleaner 사용 시에도 관련 흔적을 저장하고 있다.

본 논문에서는 앰캐시 파일의 특성을 알아보고, 응용 프로그램의 삭제시간 추정방법과 디지털 포렌식 측면의 활용방안을 제안한다.

## II. 관련 연구

현재까지 앰캐시 파일에 대한 분석은 요게시카트리(Yogesh Katri)의 블로그 내용 외에는 전무한 실정이다.

이 블로그에서는 앰캐시 파일에 기록되어 있는 응용 프로그램의 최초 실행시간, 실행경로, SHA-1 해쉬값 등에 대한 설명만 있을 뿐 앰캐시 파일의 특성에 대한 설명과 이를 어떻게 활용할 지에 대한 구체적인 방법론은 없다[2].

추가적으로 앰캐시 파일의 특성과 이를 이용한 디지털 포렌식 기술로서의 활용방안에 대한 연구가 필요하다.

## III. 앰캐시(Amcache.hve) 파일

윈도우 7 운영체제에서 RecentFileCache.bcf 파일은 응용 프로그램 호환성 기능을 지원하는데, 윈도우 8 운영체제가 출시되면서 앰캐시 파일로 대체되었다.

Table 1. Names and storage paths of Amcache.hve in versions of Windows

| File name | OS Version | File path |
|---|---|---|
| RecentFileCache.bcf | Windows 7 | %SystemDrive%\Windows\AppCompat\Programs\RecentFileCache.bcf |
| Amcache.hve | Windows 8 | %SystemDrive%\Windows\AppCompat\Programs\Amcache.hve |

앰캐시 파일과 RecentFileCache.bcf 파일이 저장되는 경로는 Table 1.과 같다.

본 장에서는 RecentFileCache.bcf 파일을 간단히 살펴본 다음 앰캐시 파일의 구조와 특성에 대하여 살펴본다.

### 3.1 RecentFileCache.bcf 파일

윈도우 운영체제가 새롭게 출시될 때 문제가 되는 것은 소프트웨어와 드라이버 호환성 문제로 인한 오류이다. 이러한 문제점을 해결하기 위하여 Application Experience Lookup 서비스(AELookupSvc, 이하 AE라 함)는 윈도우 서버 2003 서비스팩 1에 최초로 도입되었다.

AE는 응용 프로그램 호환성관리자(Application Compatibility Administrator)의 일부분이다. 이 서비스는 응용 프로그램이 시작될 때 응용 프로그램 호환성 조회 요청을 처리하고 호환성 문제를 보고하며, 프로그램에 호환성 소프트웨어 업데이트를 자동으로 적용한다.

AE는 RecentFileCache.bcf 파일에 응용 프로그램의 실행경로를 기록한다. Fig.1.과 같이 단순히 실행파일의 전체경로가 유니코드 형식으로 저장되어 있다. 이 파일은 프로세스 생성 시 프로그램 경로를 임시로 저장하기 위해 사용하는 것으로 최근 실행한 프로그램에 관한 정보만 저장되어 있다.

이러한 정보는 윈도우 작업스케줄에서 ProgramDataUpdater의 설정에 의하여 초기화된다[4]. 기본 설정시간(매일 오후 11:30)에 초기화되나, 매주・매월 1회 등의 기간으로 설정 가능하다. 하지만, 대다수의 PC가 기본 설정으로 되어 있기 때문에 1일 1회 초기화되어 포렌식 관점에서 일정한 한계가 있다.

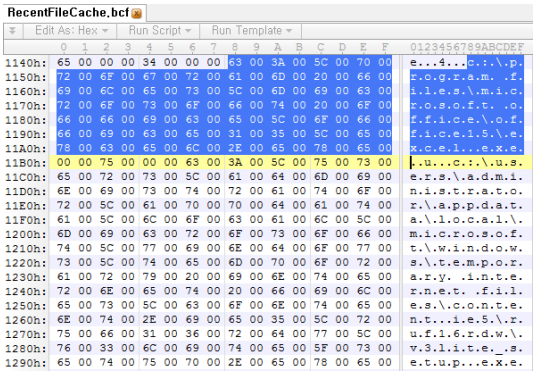


Fig. 1. File structure of the RecentFileCache.bcf

### 3.2 앰캐시 파일의 특성

앰캐시 파일은 하이브 파일로, 파일 구조는 Fig.2.와 같이 Root 키 아래에 File 키, Generic 키, Orphan 키, Programs 키로 구성된다.

Generic 키와 Orphan 키는 GUID 또는 파일 ID 관련 정보를 저장하고 있고, Programs 키는 설치된 응용 프로그램명, 버전 등의 정보를 저장하고 있으며[3],

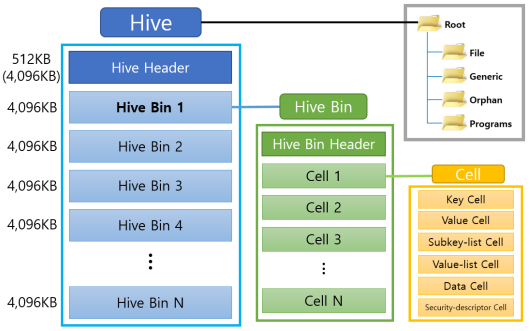


Fig. 2. Subkeys of Amcache.hve

File 키는 응용 프로그램 관련 최초 실행시간 등 많은 정보를 저장하고 있다.

앰캐시 파일의 특성은 프로그램 호환성 관리자(Program Compatibility Assistant, 이하 PCA라 함)와 연관이 있다.

PCA는 호환성 문제가 있음에도 불구하고 윈도우 비스타 이상의 운영체제에서 레거시 응용 프로그램이 실행되도록 활성화한다. PCA는 버전 검사 중에 불일치로 인해 야기되는 응용 프로그램 설치 실패를 탐지하고 문제의 응용 프로그램에 대하여 적절한 호환성 설정을 적용해 실패로부터 복구를 시도한다. PCA는 알려진 호환성 문제를 가진 프로그램에 대한 데이터베이스를 관리하여 프로그램 시작 시에 잠재적인 문제를 사용자에게 통지해준다[8].

즉, PCA는 운영체제 상에서 응용 프로그램이 정상적으로 동작하는지 확인하고 만약 응용 프로그램이 정상적으로 동작하지 않으면 호환성 데이터베이스에서 문제를 해결하도록 한다.

윈도우 8 운영체제에서 PCA 기능은 윈도우 7 운영체제와 다르다. 첫째, 응용 프로그램 실행 정보를 앰캐시 파일에 접근하여 저장한다. 둘째, third party program 실행흔적을 설치여부와 상관없이 사용자 계정 레지스트리 하이브 파일(HKCU\Software\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Compatibility Assistant\Store)에 기록한다. 그 외에 응용 프로그램 호환성 데이터베이스에 접근하는 것은 윈도우 7 운영체제와 동일하다[7].

위에서 언급한 PCA 동작과정을 살펴보자. 윈도우 8 운영체제에서 Eraser.exe를 실행한 후 Process monitor를 이용하여 관찰한 결과, Fig.3.과 같이 PCA가 레지스트리 Store 키에 관련 흔적을 저장하고, 호환성 데이터베이스와 앰캐시 파일에 접근하는 것

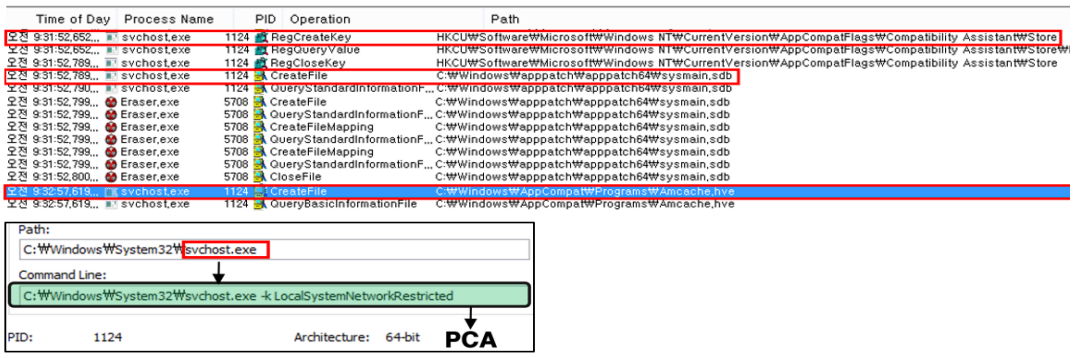


Fig. 3. Amcache.hve on Process monitor after executing Eraser.exe

Table 2. Value Names and the descriptions in File key

| Value | Description | Data Type |
|---|---|---|
| 0 | Product Name | UNICODE string |
| 1 | Company Name | UNICODE string |
| 2 | File version number only | UNICODE string |
| 3 | Language code (1033 for en-US) | DWORD |
| 4 | SwitchBackContext | QWORD |
| 5 | File Version | UNICODE string |
| 6 | File Size (in bytes) | DWORD |
| 7 | PE Header field - SizeOfImage | DWORD |
| 8 | Hash of PE Header (unknown algorithm) | UNICODE string |
| 9 | PE Header field - Checksum | DWORD |
| a | File Version | QWORD |
| b | File Version | QWORD |
| c | File Description | UNICODE string |
| d | Unknown, maybe Major &Minor OS version | DWORD |
| f | Linker (Compile time) Timestamp | DWORD - Unix time |
| 10 | Unknown | DWORD |
| 11 | Last Modified Timestamp | FILETIME |
| 12 | Created Timestamp | FILETIME |
| 15 | Full path to file | UNICODE string |
| 16 | Unknown | DWORD |
| 17 | Last Modified Timestamp 2 | FILETIME |
| 100 | Program ID | UNICODE string |
| 101 | SHA1 hash of file | UNICODE string |

을 확인할 수 있다.

앰캐시 파일의 File 키에 기록되는 정보는 Table 2.와 같다[2].

앰캐시 파일의 크기는 최초 256KB로 설정되어 있고, 응용 프로그램이 실행되면 관련 정보가 앰캐시 파일에 누적되어 기록된다. 응용 프로그램 실행 시 해당 프로그램의 파일 참조 키[1](File Reference Key)가 생성되면서 관련 정보가 앰캐시 파일에 기록된다. 만약 파일 참조 키가 생성되지 않는다면 관련 정보가 앰캐시에 기록되지 않는다. 다시 말해서, 응용 프로그램 관련 정보는 최초 실행 시에만 앰캐시 파일에 기록되고 두 번째 실행 이후부터는 기록되지 않는다.

이를 확인하기 위해서 제어판에 설치되는 실행파일(bcwipeSetup.exe[2])과 그렇지 않은 실행파일(Dictionary.exe[3])로 구분하여 실험을 진행하였다.

먼저, C:\Users\KMH\Desktop\A 폴더 내에 원본 파일인 bcwipeSetup.exe(v.6.05.1)을 저장하고, 이 원본 파일의 이름을 변경한 bcwipeSetup(renamed).exe과 해시값을 변경(프로그램의 동작에는 영향을 미치지 않는 범위 내에서 데이터 변경)한 bcwipeSetup(modified).exe을 저장한다. 이후에 C:\Users\KMH\Desktop\B 폴더 내에 상기 기술한 대로 3개의 파일을 동일하게 저장한다. 그리고 각각의 파일을 ①→⑥ 순으로 실행하여 앰캐시 파일에 관련 정보가 남는지 확인한다. 또한, 원본파일을 삭제하고 동일한 파일을 폴더 A와 B에 각각 저장한 후 ⑦→⑧ 순으로 실행한다. 실험 결과는 Table 3.과 같이 ①→⑥ 순으로 관련정보가 누적되어 기록되었으나, ⑦, ⑧과 관

1) 파일 참조 키는 특정한 실행파일을 가리키며 NTFS file Id와 순서번호로 이루어져 있다[2].

2) http://software.naver.com/software/version.nhn?softwareId=MFS_104266&categoryId=B0400000

3) http://moaimoai.tistory.com/41

Table 3. The test results for bcwipeSetup.exe

| Folder | File | Hash value | File reference address | Time information |
|---|---|---|---|---|
| Folder "A" | ① Original file | Same | Created | Written |
| | ② Renamed file | Same | Created | Written |
| | ③ Modified file | Different | Created | Written |
| Folder "B" | ④ Original file | Same | Created | Written |
| | ⑤ Renamed file | Same | Created | Written |
| | ⑥ Modified file | Different | Created | Written |
| Folder "A" | ⑦ Original file (deleted and re-execution) | Same | Not Created | Not Written |
| Folder "B" | ⑧ Original file (deleted and re-execution) | Same | Not Created | Not Written |

련된 정보는 기록되지 않았다.

다음으로, C:\Users\KMH\Desktop\C 폴더 내에 원본 파일인 Dictionary.exe을 저장하고, 이 원본 파일의 이름을 변경한 Dictionary(renamed).exe과 해시값을 변경한 Dictionary(modified).exe을 저장한다. 이후에 C:\Users\KMH\Desktop\D 폴더 내에 상기 기술한 대로 3개의 파일을 동일하게 저장한다. 그리고 각각의 파일을 ⑨→⑭ 순으로 실행하여 앰캐시 파일에 관련 정보가 남는지 확인한다. 또한, 원본파일을 재실행하고(⑮), 삭제한 후 다시 동일한 파일을 폴더 C에 저장하여 재실행한다. 실험 결과는 Table 4.와 같이 ⑨→⑭ 순으로 관련정보가 누적되어 기록되었으나, ⑮, ⑯과 관련된 정보는 기록되지 않았다.

위의 실험결과를 분석해 보면, 동일한 프로그램이라고 하더라도 설치 및 실행경로, 프로그램명이 다를 경우 파일 참조 키가 새로 생성되면서 앰캐시 파일에 관련 정보가 누적되어 기록되는 것을 확인할 수 있다. 하지만, 동일한 폴더에서 동일한 프로그램을 삭제(제어판에서 프로그램 제거 및 해당 폴더에서 Shift+Del 키를 이용하여 파일 삭제)한 후 재설치하거나 동일한 실행파일을 두 번 실행할 경우에는 파일 참조 키가 새로 생성되지 않아 앰캐시 파일에 관련 정보가 기록되지 않는 것을 확인할 수 있다.

Table 4. The test results for Dictionary.exe

| Folder | File | Hash value | File reference address | Time information |
|---|---|---|---|---|
| Folder "C" | ⑨ Original file | Same | Created | Written |
| | ⑩ Renamed file | Same | Created | Written |
| | ⑪ Modified file | Different | Created | Written |
| Folder "D" | ⑫ Original file | Same | Created | Written |
| | ⑬ Renamed file | Same | Created | Written |
| | ⑭ Modified file | Different | Created | Written |
| Folder "C" | ⑮ Original file (2nd execution) | Same | Not Created | Not Written |
| | ⑯ Original file (deleted and re-execution) | Same | Not Created | Not Written |

이렇게 앰캐시 파일에 누적된 정보를 삭제하기 위해서는 앰캐시 파일 자체를 삭제할 수밖에 없다. 하지만 앰캐시 파일을 삭제하는 것은 매우 어렵다. 이 파일은 항상 시스템에서 열려진 상태이기 때문에 일반적인 방법으로는 삭제할 수 없다. 그렇기 때문에 안티포렌식에 대응할 수 있다는 이점도 있다.

본 연구에서 중점적으로 다루고자 하는 부분은 앰캐시 파일에 기록되는 시간정보이다. 특히, 제어판에 설치되는 패키지 응용 프로그램의 경우 앰캐시 파일에 기록되는 시간정보는 아래와 같다.

첫째, 응용 프로그램의 최초 설치・실행・삭제시간을 확인할 수 있다. 앰캐시 파일은 레지스트리 하이브 파일이며, 실행파일의 각 키에 "마지막 수정 시간(Last Written Time)[4]"이 저장된다. 이를 통해서 각 실행파일의 최초 실행시간을 확인할 수 있다. 이러한 특성을 활용하면 실행 뿐 아니라, 설치・삭제행위와 관련된 실행파일을 실행할 경우에도 최초 설치・삭제 시간까지 확인할 수 있다.

둘째, 응용 프로그램의 생성시간을 확인할 수 있다. 해당 실행파일의 "Created Timestamp[5]"에서 인터넷에서 다운로드한 시간 또는 외장저장장치에서 복사를 한 시간을 추정할 수 있다.

Fig.4.는 앰캐시 파일에서 시간정보를 확인하기 위한 순서도를 나타내며 분석과정은 다음과 같다.

① 첫 번째, 응용 프로그램에 대한 파일 참조 키 생성 여부를 확인한다.

② 두 번째, 응용 프로그램의 경로가 "C:\Users" 또는 외부 저장장치인지 확인한다.

③ 세 번째, 응용 프로그램의 크기가 앰캐시 파일에 기록되어 있는지 여부를 확인한다.

④ 네 번째, 응용 프로그램이 제어판에 설치되는 지

4) Last Written Time : 실행파일이 최초 실행될 때 키(key)에 생성되는 시간정보이며, 최초 실행 시에만 생성되고 이후에는 수정되지 않는 특징이 있다.

5) Created Timestamp : 앰캐시 파일에 기록되는 응용 프로그램의 크기가 '0byte'가 아닌 경우에만 기록되는 시간정보이다. 이 시간정보는 Table 1.의 value 12에서 확인할 수 있으며, 응용 프로그램의 생성시간을 추정할 수 있다. 즉, 이 시간정보는 인터넷을 통한 다운로드 또는 외장 저장장치를 통한 복사로 파일이 생성된 시간이라고 추정할 수 있다.

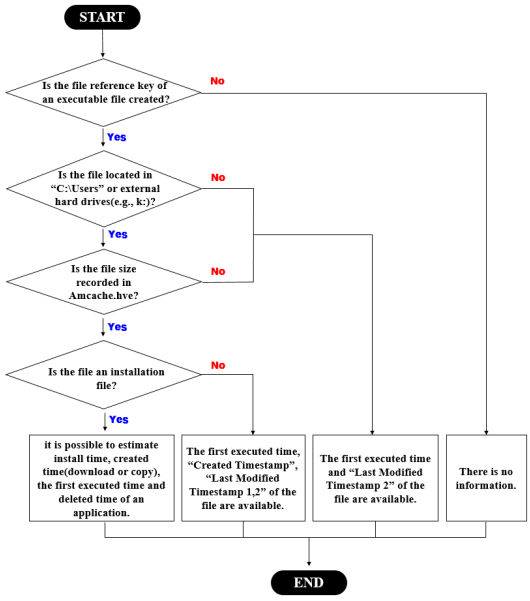


Fig. 4. Flowchart on timestamps of an executable file available in Amcache.hve

확인한다.

위와 같은 분석을 통해서 응용 프로그램에 대한 시간 정보를 확인할 수 있다. ①~④의 분류 상 모두 "YES" 이면 응용 프로그램의 최초 설치시간, 생성시간, 최초 실행시간, 삭제시간을 추정하는 것이 가능하다.

네 번째 과정에서 "NO"이면 실행파일의 4가지 시간 정보를 모두 확인할 수 있고, 그 외에는 2가지 시간정보(최초 실행시간, 마지막 변경시간)를 확인할 수 있거나 시간정보 자체를 전혀 확인할 수 없다.

## IV. 앰캐시 파일 활용방안

앞서 살펴본 대로 앰캐시 파일의 특성 상 응용 프로그램의 최초 실행시간을 확인할 수 있다. 특히, 제어판에 설치되는 응용 프로그램의 경우에는 다수의 실행파일이 패키지화되어 있기 때문에 각각의 실행파일에 대한 최초 실행시간을 확인할 수 있다. BCWipe 프로그램을 예로 들면 아래와 같다.

① bcwipeSetup.exe : 최초 설치시간 및 생성시간
② BCWipe.exe : 최초 실행시간
③ BCUnInstall.exe : 최초 삭제시간

즉, 앰캐시 파일을 통해서 응용 프로그램의 생성, 설치, 삭제시간까지 확인할 수 있다.

또한, 전체경로에 기록되어 있는 안티포렌식 프로그램, 포터블 프로그램 실행 흔적 등을 확인할 수 있다.

본 절에서는 앰캐시 파일의 활용방안을 제시하고자 한다.

### 4.1 응용 프로그램 생성, 설치, 삭제시간 확인

3절에서 언급하였듯이, 응용 프로그램 실행 시 PCA가 앰캐시에 실행 흔적을 기록한다. 이를 통하여 응용 프로그램의 생성시간, 최초 설치·실행·삭제시간을 확인할 수 있다.

먼저, bcwipeSetup.exe 파일을 인터넷에서 다운

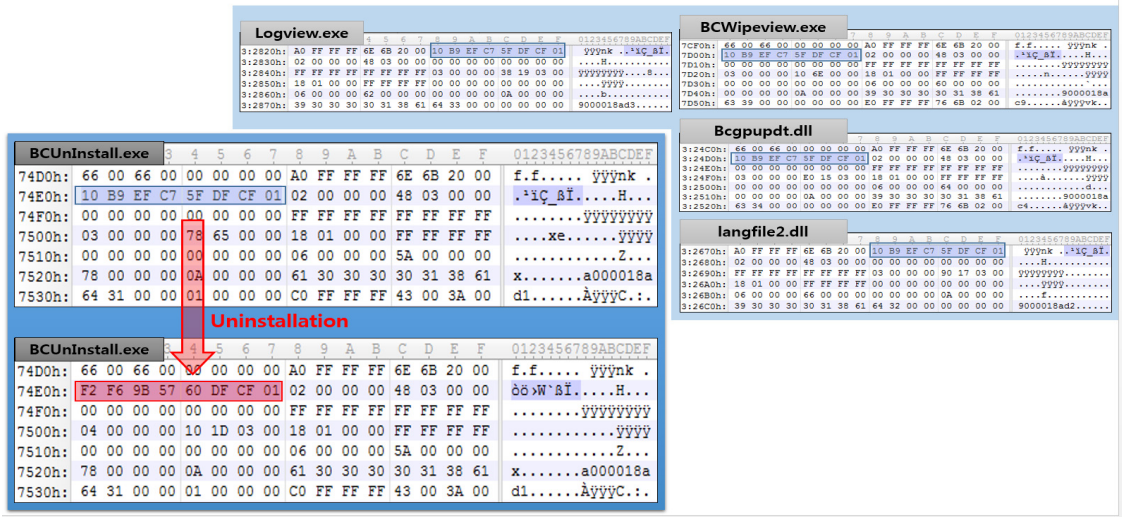


Fig. 5. Deleted time

로드 받아 설치하면 앰캐시 파일에 관련 정보가 기록된다. 다운로드 시간은 이 실행파일의 “Created Timestamp”에서 확인할 수 있는데, Fig.6.과 같이 2014년 10월 4일 08:09:35 이다. 만약 외장 저장장치에서 파일을 복사하였다면 복사한 시간도 이와 같은 방법으로 추정할 수 있다.

설치시간은 “마지막 수정 시간(Last Written Time)”에서 Fig.7.과 같이 2014년 10월 4일 08:10:02 임을 확인할 수 있다.

다음, BCWipe.exe 파일을 실행하면 앰캐시 파일에 관련 정보가 기록된다. 최초 실행시간은 실행파일의 “마지막 수정 시간(Last Written Time)”에서 Fig.8.과 같이 2014년 10월 4일 08:15:08 임을 확인할 수 있다.

마지막으로 제어판이나 삭제프로그램(revouninstaller)을 이용하여 삭제하면 삭제 기능과 관련된 BCUnInstall.exe 실행파일이 실행되면서 앰캐시 파일에 관련 정보가 기록된다. 이 실행파일의 “마지막 수정 시간(Last Written Time)”에서 삭제시간을 확인할 수 있다. Fig.5.와 같이 삭제 전에는 다른 실행파일과 동일한 시간정보를 가지고 있으나(2014년 10월 4일 08:14:28), 삭제 후에는 자신만의 시간정보를 갖게 되는데, 이를 통해서 2014년 10월 4일 08:18:29 이 삭제된 시간임을 알 수 있다.

또한, BCWipe 프로그램을 삭제하고 재설치하여도 실행 흔적이 남는다. 즉, 삭제프로그램(revouninstaller)으로 BCWipe 프로그램을 삭제하고 eraser로 사용자 컴퓨터에 저장되어 있는 설치파일을 완전 삭제한 후에 재설치할 경우에도 관련 행위에 대한 시간이 앰캐시 파일에 누적되어 기록되는 것을 확인할 수 있다.

동일한 프로그램을 상기 언급한 대로 삭제하고 인터넷에서 재다운로드하여 설치하면 2번째 다운로드 시간은 Fig.9.와 같이 2014년 10월 6일 09:00:13 임을 확인할 수 있고, 2번째 설치시간은 Fig.10.과 같이 2014년 10월 6일 09:00:59 임을 확인할 수 있다.

또한, 재설치 이후, 최초 실행시간 및 삭제시간은 Fig.11.~12.와 같이 각각 2014년 10월 6일 09:07:44, 09:13:28 임을 확인할 수 있다.

앰캐시 파일 분석을 통한 응용 프로그램의 타임라인 구성을 위한 방법을 묘사하면 Fig.13.과 같다. BCWipe 프로그램의 Volume GUID는 ‘bf2461ca-816f-11e3-be65-806e6f6e6963’이고 이러한 Volume GUID 아래 bcwipeSetup.exe, BCWipe.exe, BCUninstall.exe의 파일 참조 키는 각각 ‘b0000189fb’, ‘9000018acd’, ‘a000018ad1’이며, 각 실행파일의 value를 확인하여 BCWipe 프로그램의 생성시간, 설치시간, 최초 실행시간, 삭제시간을 추정할 수 있다. 또한 Prefetch 파일을 통해서 최종 실행시간을 확인할 수 있으므로, 이를 종합하면 BCWipe 프로그램의 전체적인 실행 타임라인을 완성

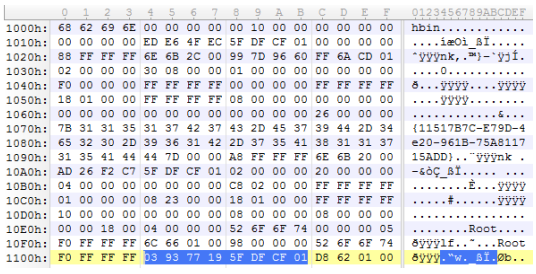

Fig. 6. Download time

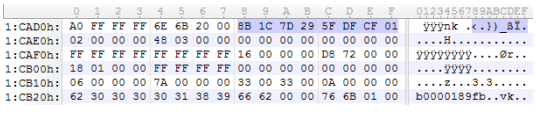

Fig. 7. Install time

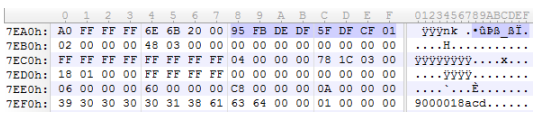

Fig. 8. First executed time

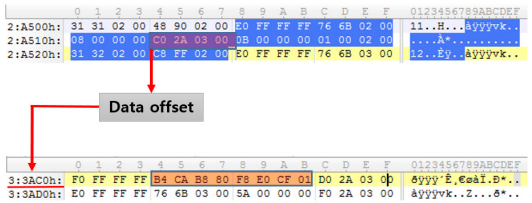


Fig. 9. Download time after re-installation

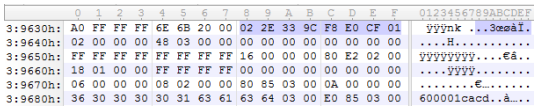

Fig. 10. Install time after re-installation

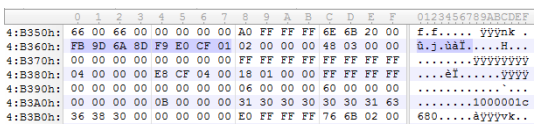

Fig. 11. The first executed time after re-installation

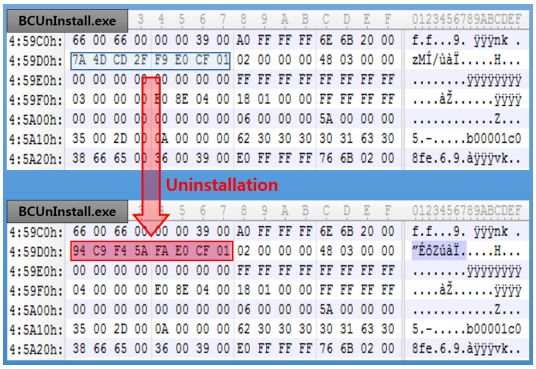


Fig. 12. Deleted time after re-installation

할 수 있는 것이다.

### 4.2 안티포렌식 프로그램 사용 흔적 확인

데이터 삭제방법에는 Shift+Del 키를 이용한 일반적인 파일 삭제뿐만 아니라 인터넷에서 무료로 다운로드할 수 있는 파일 완전삭제 도구를 이용하는 방법이 있다.

Shift+Del 키를 이용한 일반적인 파일 삭제는 FTK Imager와 같은 포렌식 도구에 의하여 복구가 가능하지만, Eraser를 이용하여 삭제할 경우에는 흔적이 남지 않는다[6].

Eraser와 같은 완전삭제 프로그램을 이용하여 삭제한 경우 삭제된 데이터의 흔적은 쉽게 찾을 수 없다. 하지만, 안티포렌식 행위를 하였다는 정황증거를 확인할 수 있는데, Eraser와 같이 널리 사용되고 있는 완전삭제 프로그램인 BCWipe를 사용하여 실험한 결과 Fig.14.와 같이 앰캐시 파일에는 BCWipe의 흔적이 남아있다.

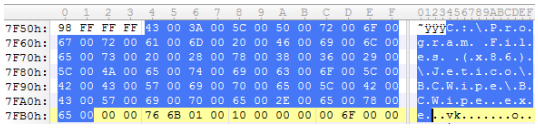

Fig. 14. Stored name of application and full path

### 4.3 포터블 프로그램 사용 흔적 확인

안티포렌식 도구의 공통적인 단점 중 하나가 사용 후 시스템에 흔적을 남기는 것이다. 그렇기 때문에 최근의 안티포렌식 기법은 흔적을 최소화하기 위하여 포터블 프로그램을 이용한다.

포터블 프로그램은 디스크에 설치되지 않고 실행되는데, 가상의 파일시스템과 레지스트리를 사용하고 종료 시에는 해당 흔적을 삭제한다. 이러한 특징 때문에 디지털 포렌식 수사의 방해요소로 작용하고 있다.

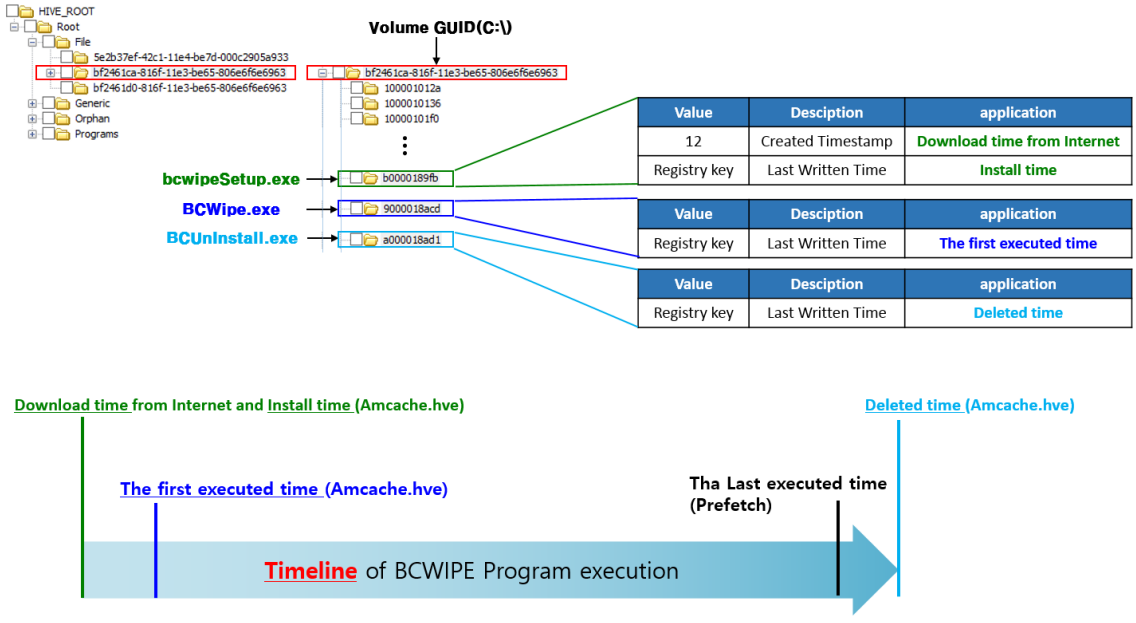


Fig. 13. The overall timeline of BCWIPE Program using Amcache.hve file

하지만, 앰캐시 파일을 분석하면 포터블 프로그램 실행흔적을 확인할 수 있다. Fig.15.와 같이 E:에서 Eraser Portable 프로그램이 실행되었음을 확인할 수 있다.

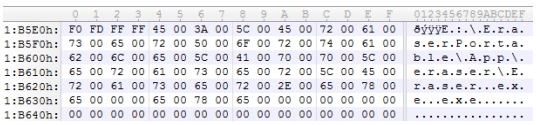


Fig. 15. Eraser Portable

## 4.4 CCleaner 사용 후에도 응용 프로그램 흔적 확인

응용 프로그램 실행흔적은 프리패치 파일에도 존재한다. 사용흔적 삭제도구로 일반적으로 널리 사용되고 있는 CCleaner를 이용하여 프리패치 파일을 대상으로 실험을 한 결과, Fig.16.과 같이 상당수의 응용 프로그램이 삭제된 사실을 확인할 수 있다.

앰캐시 파일은 Fig.17.과 같이 하이브리스트(hivelist)에 없는 레지스트리 하이브 파일 구조를 가진 파일이다. 그렇기 때문에 레지스트리 정리를 하더라도 앰캐시 파일에 저장되어 있는 정보는 그대로 존재한다.

CCleaner를 실행한 결과 Fig.18.과 같이 실행 전·후의 차이가 없음을 알 수 있다.

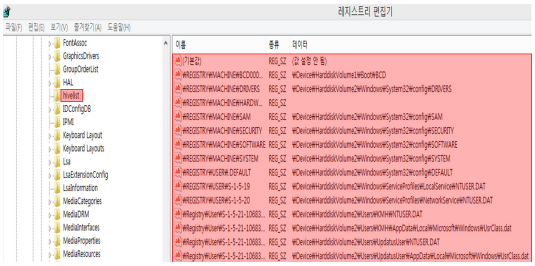


Fig. 17. The hivelist

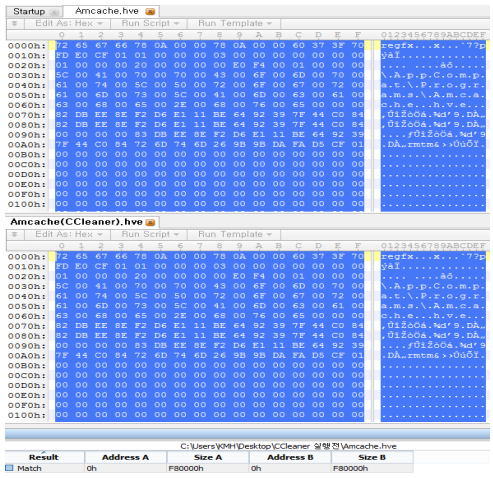


Fig. 18. Comparision of Amcache.hve files before and after execution of the CCleaner

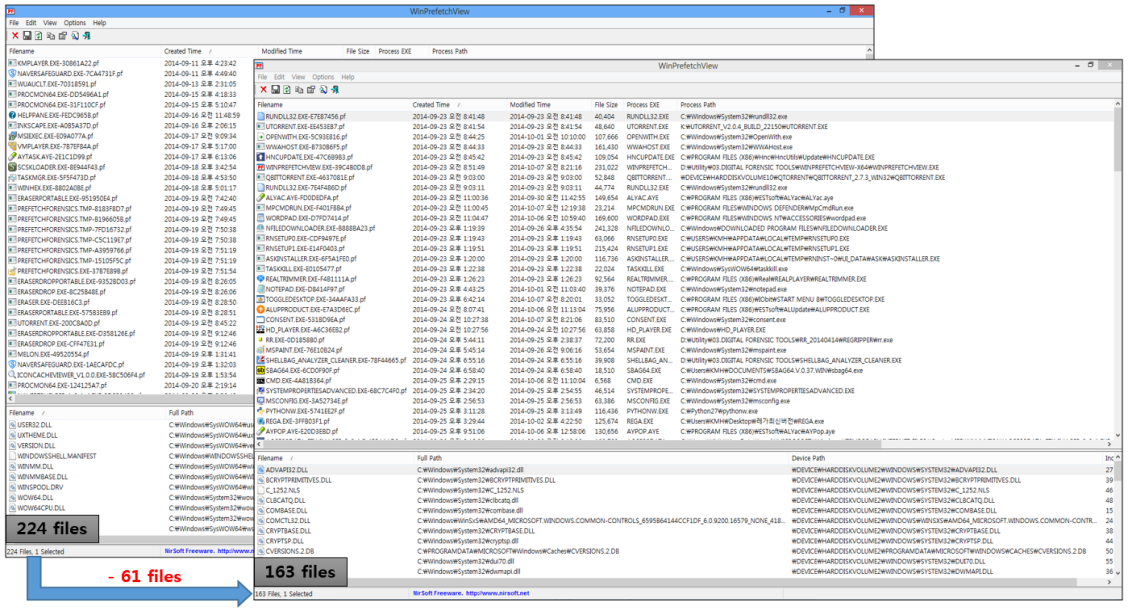


Fig. 16. Prefetch files after execution of CCleaner

### 4.5 마운트된 장치 흔적 추적

응용 프로그램 실행 시 Fig. 19.와 같이 마운트된 장치의 GUID(Globally Unique Identification Number)를 통하여 외장저장장치 흔적을 확인할 수 있다. 이는 시스템 하이브의 MountedDevices 레지스트리키에 있는 GUID와 동일하다는 것을 알 수 있다[2].

또한, 이러한 GUID에 "마지막 수정 시간(Last Written Time)"이 저장되는데, 이 시간정보는 마운트된 장치에 저장되어 있는 응용 프로그램이 마지막으로 실행되었을 때의 시간으로 갱신되어 저장된다.


Fig. 19. Volume GUIDs referenced in Amcache.hve file

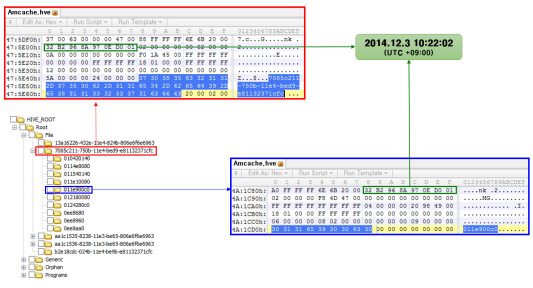


Fig. 20. The last executed time stored in volume GUIDs

## V. 결 론

윈도우 8 운영체제에서 추가된 아티팩트인 앰캐시 파일은 응용 프로그램과 관련하여 다양한 정보를 저장하고 있다. 특히, 디지털 포렌식 수사 시 중요한 정보인 시간정보를 저장하고 있다.

프리패치 파일 및 아이콘캐시도 응응프로그램과 관련하여 많은 정보를 저장하고 있지만, 프리패치는 프리패치 파일 개수가 128개로 제한되어 오래된 흔적을 파악할 수 없고, 아이콘캐시는 열람, 실행, 복사, 저장된 응용 프로그램의 경로정보를 저장하고 있지만, 시간정보를 저장하지 않는다.

앰캐시 파일은 응용 프로그램의 생성시간 및 최초 설치·실행·삭제시간을 저장하고 있기 때문에 상기 언급한 프리패치와 아이콘캐시의 일정한 한계점을 극복할 수 있다.

그렇기 때문에 프리패치와 아이콘캐시 분석과 함께 앰캐시 파일 분석을 병행한다면 컴퓨터 사용자가 해당 응용 프로그램을 언제 최초로 설치하여 몇 번 실행하였고, 언제 마지막으로 실행하였는지 확인 가능하며, 삭제 및 설치 횟수까지 파악할 수 있어서 개별 응용 프로그램의 전체적인 타임라인을 파악하는 것이 가능하다.

또한, 앰캐시 파일 분석을 통하여 안티포렌식 프로그램, 포터블 프로그램, 레지스트리 정리 프로그램 실행흔적과 외장저장장치 흔적까지도 식별할 수 있다.

현재까지 앰캐시 파일 분석을 위한 디지털 포렌식 도구는 없다. 향후에 본 논문에서 제시한 앰캐시 파일의 활용방안을 구현하기 위한 분석도구를 개발할 예정이다.

## References

[1] Microsoft, "Inside the Registry," http://technet.microsoft.com/en-us/library/cc750583.aspx ; December 4, 2013[accessed March 2014].

[2] Yogesh Khatri, "Amcache.hve in Windows 8 - Goldmine for malware hunters," http://www.swiftforensics.com/2013/12/amcachehve-in-windows-8-goldmine-for.html ; December 4, 2013[accessed March 2014].

[3] Yogesh Khatri. "Amcache.hve - Part 2," http://www.swiftforensics.com/2013/12/amcachehve-part-2.html ; December 26, 2013[accessed March 2014].

[4] Corey Harrell, "Revealing the RecentFile Cache.bcf File," http://journeyintoir.blog

spot.in/2013/12/revealing-recentfilecachebcf-file.html ; December 2, 2013{accessed March 2014).
〔5〕 M. Russinovich and B. Cogswell, Microsoft, process monitor(2013) Available from http://technet.microsoft.com/en-ie/sysinternals/bb896645.aspx
〔6〕 Jain, Anu, and Gurpal Singh Chhabra, "Anti-forensics techniques: An analytical review," Contemporary Computing (IC3), 2014 Seventh International Conference on. IEEE, pp. 412 - 418, Aug. 2014.
〔7〕 Corey Harrell, "Revealing Program Compatibility Assistant HKCU AppCompatFlags Registry Keys," http://journeyintoir.blogspot.kr/2013/12/revealing-program-compatibility.html ; December 17, 2013{accessed March 2014).
〔8〕 Mark E. Russinovich, David A. Solomon and Alex Ionescu, Windows® Internals, 5th Ed., Microsoft press, pp. 332-333, 2009.
〔9〕 MSDN Blogs. "Misinformation and the The Prefetch Flag," http://blogs.msdn.com/b/ryanmy/archive/2005/05/25/421882.aspx ; December 17, 2013{accessed March 2014).
〔10〕 Chan-Youn Lee and Sangjin Lee, "Structure and application of IconCache.db files for digital forensics," Digital Investigation, vol. 11, no. 2, pp. 102-110, June. 2014.

## 〈저자소개〉

김 문 호 (Moon-Ho Kim) 학생회원
2004년 3월: 육군사관학교 일본어과 졸업
2014년 3월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 디지털 포렌식, 정보보호

이 상 진 (Sang-jin Lee) 종신회원
1987년 2월: 고려대학교 수학과 학사
1989년 2월: 고려대학교 수학과 석사
1994년 8월: 고려대학교 수학과 박사
1989년 10월~1999년 2월: ETRI 선임 연구원
1999년 3월~2001년 8월: 고려대학교 자연과학대학 조교수
2001년 9월~현재: 고려대학교 정보보호대학원 교수
2008년 3월~현재: 고려대학교 디지털포렌식연구센터 센터장
〈관심분야〉 디지털 포렌식, 심층 암호, 해쉬 함수