2006

Assessing Trace Evidence Left By Secure Deletion Programs

Paul Burke

Philip Craiger
University of Central Florida, craigerj@erau.edu

Chapter 15

ASSESSING TRACE EVIDENCE LEFT BY SECURE DELETION PROGRAMS

Paul Burke and Philip Craiger

Abstract Secure deletion programs purport to permanently erase files from digital media. These programs are used by businesses and individuals to remove sensitive information from media, and by criminals to remove evidence of the tools or fruits of illegal activities. This paper focuses on the trace evidence left by secure deletion programs. In particular, five Windows-based secure deletion programs are tested to determine if they leave identifiable signatures after deleting a file. The results show that the majority of the programs leave identifiable signatures. Moreover, some of the programs do not completely erase file metadata, which enables forensic investigators to extract the name, size, creation date and deletion date of the "deleted" files.

Keywords: Secure deletion, trace evidence, Windows XP, FAT12 file system

1. Introduction

Demand for secure deletion programs has grown with the enormous amounts of sensitive information stored on digital media. Several programs that offer to remove computer users' digital tracks are being marketed. These programs supposedly delete all traces of files and offer "immunity" from forensic analysis.

This paper evaluates five Windows-based programs based on their ability to delete files from a FAT12 file system. The research goal is to determine what, if any, trace evidence remains after using these programs. Trace evidence is a term used in traditional forensics to refer to evidence left in small, yet measurable, amounts. Our hypothesis is that different programs use different software routines for deleting files. Consequently, the programs may be identified by their signatures left on the file system. From a law enforcement perspective, knowing that a

suspect used a software program to delete a file at a certain time can be used to establish intent. Therefore, while it may not be possible to easily recover the deleted files, it is useful to extract whatever proof there may be pertaining to the deletion programs that were employed and the files they were used to delete.

The tests are performed on five popular programs:

- Evidence Eliminator 5.0 (Robin Hood Software)
- CyberScrub Privacy Suite 4.0 (CyberScrub)
- East-Tec Eraser 2005 (EAST Technologies)
- UltraSentry 2.0 (IDM Computer Solutions)
- Eraser 5.3 (Sami Tolvanen)

As this paper was nearing completion, we became aware of a study by Geiger and Cranor [2] that examined six secure deletion programs (including two that are evaluated here). Geiger and Cranor investigated the ability of the programs to remove personal data from a Windows-based computer. As such, our work is complementary as it focuses specifically on the residual binary structures, an issue not explored in detail in Geiger and Cranor's work.

2. Background

File systems can be thought of as logical shelves that organize and structure the placement of raw data on digital media. In the grand scheme of information storage they are merely another abstraction layer above the raw data transfers to and from digital media. A file system not only records the location of a file on the media, but also metadata – descriptive information about the file that is not actually part of the file, e.g., filename, timestamps associated with its creation and modification, physical size, and starting location of the file. File systems break up the physical media into logical fixed-size sections called clusters. Each file is allocated a set of clusters in which it is stored. There are several dozen commonly used file systems, each of which structures this information in a different manner.

The family of file systems used with consumer-level Windows versions up until the introduction of Windows XP is called FAT (File Allocation Table) [4]. Several versions of the FAT file system exist, each primarily differing in the number of bits used to address the file system. An area called the root directory stores information pertaining to the metadata of individual files (see Figure 1). Locations of currently utilized clusters are

```
00025f0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0002600: e559 4649 4c45 2020 5458 5420 1814 2056  .YFILE  TXT .. V
0002610: e532 e532 0000 d154 e532 0200 9334 0000  .2.2...T.2...4..
0002620: 0000 0000 0000 0000 0000 0000 0000 0000  ................
```

Figure 1. Hex view of the directory entry of a deleted file.

tracked via two identical file allocation tables. A file allocation table is a singly linked list of pointers that identify the specific clusters comprising a file. The actual file contents are stored in the data area elsewhere on the file system.

3. Traditional File Deletion

When a file is deleted in Windows, two changes occur at the file system level. First, both file allocation tables are updated to reflect that the space occupied by the file is now available. Second, the file's directory entry is marked as deleted by overwriting the first character of the filename with `0xE5` as shown in Figure 1. All remaining file metadata in the root directory, including the rest of the filename and its former initial position on the disk, remain unmodified. The contents of the file itself are literally untouched by this process. This is why it is fairly simple to undelete a file – provided that the space that the file occupied on the media has not been overwritten.

The fact that deleting a file does not delete its contents is the basis for the demand of secure deletion software: these programs are designed to permanently remove files. Secure deletion programs go beyond traditional file deletion to prevent file recovery.

4. Secure File Deletion

Most modern media records data in the form of magnetic traces. Thus, a device such as a hard disk can be read at a sufficiently low level to extract the polarity a specific location on the disk had in the past. This is possible because the transition from one polarity to another is not complete – slight magnetic traces remain after a write operation. Thus, specialized equipment can be used to recover data even after it is overwritten.

Several data wiping strategies have been proposed to combat the use of equipment that can read magnetic traces. The general consensus is to perform multiple overwrites using a combination of random characters and zeroes. This data overwriting method has two obvious benefits. First, at some point it becomes infeasible to determine which traces belonged to which write operation. Second, including random

data patterns makes it extremely difficult to determine which traces are the remains of the original data.

Three popular secure deletion strategies related to this general concept are highlighted in the U.S. Department of Defense's National Industrial Security Program Operating Manual (NISPOM) (DoD 5220.22-M) [1] and in a paper by Guttman [3]. Both documents are approximately ten years old, but the techniques they describe are implemented in nearly all modern secure deletion programs.

The 5220.22-M document provides a cleaning/sanitization matrix for different types of media. Within this matrix are two strategies that are often used on magnetic media:

1. Level C: Overwrite all addressable locations with a single character. This is often referred to as "DoD Short." It is usually implemented as a full zeroing of the data area.

2. Level D: Overwrite all addressable locations with a character, its complement, then a random character, and verify that this has occurred. This is often referred to as "DoD 3-pass" (or some variant).

Some secure deletion programs reference a 7-pass DoD secure delete that complies with 5220.22-M, but such a strategy is not detailed in the document. In fact, the February 2001 revision of NISPOM omits all recommendations for low-level sanitization strategies.

Guttman's paper [3] discusses Magnetic Force Microscopy (MFM) that can be used to read deviations in the magnetic force on a hard disk platter and determine data from past writes. Based on the binary encoding that a hard disk drive uses on the platter, Guttman proposes several permutations of binary patterns (35 in all, eight of which involve random data) for the purpose of reliably erasing data from various hard disks. A 35-pass technique implementing all of Guttman's suggestions is commonly used by secure deletion programs. However, as Guttman notes in a follow-up to his original paper [3], it is only necessary to use a subset of his suggestions for the disk drives in use today. In fact, due to the encoding schemes used in drives manufactured within the past ten years, multiple overwrites with random data is the most effective secure deletion strategy.

5. Complete Deletion Protocol

Based on our research, we posit that a secure deletion program must perform four operations on a file in order to assure its complete removal from a FAT file system.

1. Identify the physical locations of the file on the media and overwrite those areas to the end of the occupied clusters, performing either a single pass or multiple passes with all `0`s, all `1`s, a random data pattern, or a combination thereof.

2. Locate the file's root directory entry and remove all metadata associated with the file by overwriting with random data in multiple passes. Data in the entry should be carefully maintained to appear correlated with a regular file delete, e.g., date modified should not be after the current date.

3. Locate and remove the file allocation table entries for the file using multiple passes, freeing up the space used by the file at the logical level.

4. Purge all the information regarding the file that remains in memory. Depending on the operating system in use, this may require the computer to be rebooted.

6. Testing Procedure

A series of tests were performed to determine what trace evidence remained on a file system after a secure deletion operation was performed on a file. All testing occurred on a computer running Windows XP (Service Pack 2). The following protocol was used to develop the test media:

1. Overwrite a 3.5" floppy disk with a pattern of binary zeroes using the GNU `dd` program in Linux.

2. Format the floppy disk with a full format in Windows XP to create a FAT12 file system on the floppy disk.

3. Copy a single text file (`myfile.txt`) of size 13459 bytes to the floppy disk.

4. Eject and image the floppy on a separate computer.

Five identical tests were performed with each program to determine what, if anything, varied after the secure deletion process. By comparing each individual test it was possible to determine which elements of the deletion process were constant and which were random. The following protocol was used for each run:

1. Insert the floppy disk into the disk drive and start the secure deletion program.

2. Select the file to be securely deleted on the floppy disk and run the secure deletion operation on it.

3. Image the floppy using the GNU `dd` program in Linux.

4. Wipe the floppy with a pattern of binary zeroes to remove any remaining trace evidence, and load the original image onto the floppy using the GNU `dd` program in Linux.

5. Reboot the computer and repeat the first step for the next test.

The five acquired images were then compared to analyze the variations in the results produced by each secure deletion program.

A separate Windows XP install was created for each program to eliminate potential conflicts between programs. KNOPPIX 3.8.1 was used to image the partition for each install. All systems involved in data acquisition were isolated from the Internet to prevent potential contamination. Analyses were performed at the binary level using the console hex editors `xxd` and `hte`; `cmp` (part of the GNU `diff` suite) was used to determine any changes between tests. Additionally, Sleuth Kit and Forensic Toolkit (FTK) were used to confirm the initial findings at a higher level.

The default preferences for all programs were retained. No modifications were made to the program preferences after the initial installation process. Due to the diversity of options and functions available in the programs, it was not feasible to try to match up similar erasing techniques. The assumption is that most users do not modify the default preferences, particularly as one must have some technical knowledge to understand the differences between deletion strategies.

Table 1. Default secure deletion protocols.

Secure Deletion Program	**Default Protocol**
Evidence Eliminator	Zeroes file (1 pass)
UltraSentry	DoD Level C
CyberScrub	DoD 5220.22-M (7 passes)
Eraser	Guttman (35 passes)

As the analysis was not performed at the raw magnetic level, no statements can be made here about the effectiveness of each program in that regard. All the programs deleted the file with at least one pass. Table 1 summarizes the default protocols used by the secure deletion programs.

7. Results

As expected, all the programs completely deleted the file contents. However, several programs did leave digital signatures within the file's root directory entry. The findings are presented below.

7.1 Evidence Eliminator

Robin Hood Software's Evidence Eliminator 5.0 offers numerous anti-forensic features, including several different methods for erasing trace evidence left during regular Windows use. In addition, it provides a single-file secure deletion operation, the function tested in this study. The test was conducted under "Quick Mode" because it does not require a restart to complete the erasing process and is the option most likely to be chosen by a novice user.

```
00025f0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0002600: e559 4649 4c45 2020 5458 5420 1814 2056  .YFILE  TXT .. V
0002610: e532 e532 0000 4e5b e532 0200 0080 0000  .2.2..N[.2......
0002620: e545 0045 002d 002d 002d 000f 00aa 2d00  .E.E.-.-.-....-.
0002630: 2d00 2d00 2d00 2e00 7400 0000 6d00 7000  -.-.-...t...m.p.
0002640: e545 2d2d 2d2d 7e31 544d 5020 0014 2056  .E----~1TMP .. V
0002650: e532 e532 0000 4f5b e532 0200 0100 0000  .2.2..O[.2......
0002660: 0000 0000 0000 0000 0000 0000 0000 0000  ................
```

Figure 2. Residual evidence left by Evidence Eliminator.

Evidence Eliminator left a conspicuous signature: it created an additional temporary file on the file system, something no other program appears to have done (see Figure 2). This file is consistently named `EE-------.tmp` (short filename: `?E----~1.TMP`, first character deleted) as reported by Sleuth Kit's `fls`, which lists filenames in a file system image. Windows is apparently left in control of the attributes for both files so the date modified attribute of the files indicates the approximate time the file was deleted. The temporary file has the same creation date as the deleted file.

The file itself is marked as deleted in the traditional manner, using `0xE5` as an indicator of its status. Evidence Eliminator leaves the rest of the filename and its attributes untouched. This is oddly inconsistent with the program's default preferences, which state "For extra security, rename and zero sizes when wiping files." The only piece of information in the directory entry that distinguishes an Evidence Eliminator deletion from a standard Recycle Bin deletion is the file size, which was modified in the test case to a full 32 KB (from its original 13.1 KB). The file's creation date is preserved and, as noted above, the written

Table 2. Evidence Eliminator summary.

Filename	`?yfile.txt`
Creation Date	Unmodified
Modification Date	Date of file deletion
File Attributes	Unmodified
Logical Size	Increased to 32768 Bytes
Data Area	Zeroed out
Notes	Added temporary file named `EE-------.tmp` (short: `?E----~1.TMP`) with the same creation date and similar written times

attribute indicates the time when the file was deleted. The written date of the temporary file listed above and the file itself are usually only seconds apart, most likely depending on the speed of the delete operation. The data area of the file was zeroed out. The test results for Evidence Eliminator are summarized in Table 2.

7.2 UltraSentry

IDM Computer Solutions' UltraSentry 2.0 offers the standard array of features for purging trace evidence. In addition to offering the usual set of deletion strategies, UltraSentry permits a user to customize file deletion by specifying the number and types of passes on a file.

```
00025f0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0002600: e550 5050 5050 2020 5050 5020 1814 2056  .PPPPP  PPP .. V
0002610: e532 e832 0000 755a e832 0200 9334 0000  .2.2..uZ.2...4..
0002620: 0000 0000 0000 0000 0000 0000 0000 0000  ................
```

Figure 3. Residual evidence left by UltraSentry.

UltraSentry makes an attempt at hiding the filename by overwriting each character in the filename with an upper case P (`0x50`), in addition to marking it as deleted (first character: `0xE5`) (see Figure 3).

The number of characters used in the new filename appears to remain consistent with the original. The file written time is modified to reflect the time that the file was deleted; the creation time and file size are unchanged. Like Evidence Eliminator, UltraSentry zeroes out the data area where the file existed. The test results for UltraSentry are summarized in Table 3.

Table 3. UltraSentry summary.

Filename	`?PPPPP.PPP`
Creation Date	Unmodified
Modification Date	Date of file deletion
File Attributes	Unmodified
Logical Size	Unmodified
Data Area	Zeroed out
Notes	Number of letters used to overwrite the filename is the same as the length of the original filename

7.3 CyberScrub and East-Tec Eraser

CyberScrub's Privacy Suite 4.0 and EAST Technologies' East-Tec Eraser 2005 are grouped together because they are functionally similar. In fact, the two programs are almost identical down to the online registration system. Some of the file sizes for the programs and their associated libraries differ, but the two programs are the same in terms of their user interfaces and end results.

```
00025f0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0002600: e537 3037 3732 3539 3346 3222 00c6 0180  .70772593F2"....
0002610: af2a af2a 0000 0280 af2a 0000 0000 0000  .*.*.....*......
0002620: e538 3736 3331 3732 5749 5022 10ab e172  .8763172WIP"...r
0002630: e732 e732 0000 5973 e732 0000 0000 0000  .2.2..Ys.2......
0002640: e538 3132 3746 4639 3346 3222 00c7 af5e  .8127FF93F2"...^
0002650: 9a2e 9a2e 0000 b05e 9a2e 0000 0000 0000  .......^........
0002660: e538 3439 3337 3444 3346 3222 00b5 c572  .849374D3F2"...r
0002670: eb2c eb2c 0000 c672 eb2c 0000 0000 0000  .,.,...r.,......
```

Figure 4. Directory entries resulting from a CyberScrub deletion.

The CyberScrub and East-Tec Eraser programs go far as to overwrite all unused entries in the directory entry area with what appears to be random data. Each deleted file's root directory entry is overwritten with a deleted file with a random filename with the same three-letter extension `.WIP` (see Figure 4).

The date and timestamp of each file appears to be randomly selected from dates prior to the date of deletion, and the size of the file is set to zero. Unlike the other secure deletion programs, CyberScrub and East-Tec Eraser set the deleted file's Hidden attribute. Similar to the other programs, the file's data area is filled with random data up to the end of the file's last cluster. The test results for CyberScrub and East-Tec Eraser are summarized in Table 4.

Table 4. CyberScrub/East-Tec Eraser summary.

Filename	Name and extension of the deleted file are overwritten with random characters
Creation Date	Randomized, but never after current date
Modification Date	Within 2 seconds of creation date
File Attributes	Added hidden attribute
Logical Size	Increased to 32768 Bytes
Data Area	Randomized to the end of the file's last cluster
Notes	Directory entries are filled with deleted entries, each having the same three-character extension; one entry has the `.wip` extension

7.4 Eraser

Sami Tolvanen's Eraser 5.3 is provided under the GNU General Public License and is available free-of-charge to the public. The version tested was the final version released by Tolvanen before maintainership was transferred to Garrett Trant of Heidi Computers Limited. Several versions of the program have since been released, the most recent being Eraser 5.7. Despite its age, the Eraser 5.3 program is compatible with the latest version of Windows.

```
00025f0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0002600: e541 315a 544f 2020 554c 4520 0000 0000  .A1ZTO  ULE ....
0002610: 2100 2100 0000 0000 2100 0000 0000 0000  !.!.....!.......
0002620: 0000 0000 0000 0000 0000 0000 0000 0000  ................
```

Figure 5. Residual evidence left by Eraser.

Eraser deleted the test file, randomizing its filename but keeping the number of characters used for the filename the same as that of the original file (see Figure 5). The file extension is similarly randomized. The date attributes of the file, instead of being zeroed out or set to a random date, are consistently set to midnight, January 1, 1980 (corresponding to the start of the FAT file system epoch). The file size is set to zero, and the data area is filled with random bits up to the end of the cluster in which the file existed. The test results for Eraser are summarized in Table 5.

8. Conclusions

Secure deletion programs claim to permanently remove files from digital media. However, tests of five popular Windows-based programs demonstrate that each program leaves unique signatures, which could

Table 5. Eraser summary.

Filename	Name and extension of the deleted file are overwritten with random characters
Creation Date	January 1, 1980
Modification Date	January 1, 1980
File Attributes	Unmodified
Logical Size	Null
Data Area	Randomized to the end of the file's last cluster
Notes	Date fields are set to the start of the FAT file system epoch rather than being zeroed out

assist examiners in determining whether a secure deletion was performed and in identifying the program used to perform the deletion.

Our tests did not investigate the numerous options provided by the secure deletion programs. In fact, the testing used the default preferences, which are not necessarily optimized for effectiveness. Several of the program options, if properly utilized, may alter or remove the trace evidence found during testing. We propose to extend this research to other file systems (e.g., FAT32, NTFS, ext2), other operating systems (Windows, Linux, UNIX, Mac OS X), as well as the full cadre of secure deletion utilities that exist for these operating systems.

References

[1] Defense Security Service, *National Industrial Security Program Operating Manual (NISPOM)*, DoD 5220.22-M, U.S. Department of Defense (www.dss.mil/isec/nispom_0195.pdf), 1995.

[2] M. Geiger and L. Cranor, Counter-Forensic Privacy Tools: A Forensic Evaluation, Technical Report CMU-ISRI-05-119, Institute for Software Research International, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania (reports-archive.adm.cs.cmu.edu/anon/isri2005/CMU-ISRI-05-119.pdf), 2005.

[3] P. Guttman, Secure deletion of data from magnetic and solid-state memory, *Proceedings of the Sixth USENIX Security Symposium* (www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html), 1996.

[4] Microsoft Corporation, FAT32 File System Specification (www.microsoft.com/whdc/system/platform/firmware/fatgen.mspx), 2000.