Hal Berghel and David Hoelzer

Disk Wiping By Any Other Name

What does a disk wiper wipe when a disk wiper does wipe disks?

With the advance of modern computer forensics tools, disk wiping (aka data wiping or disk/data erasing) has become increasingly important in the protection of proprietary, confidential, private, and personal information for the law-abiding computer user. Because of Windows' ubiquity in the desktop and notebook markets, a cornucopia of disk wiping utilities are now available. We wondered just how effectively these utilities are cleansing disks.

Our interest in disk wiping began with the observation that the Windows built-in utility, cipher, wipes disks by filling a file, EFSTMPWP, with enough data to consume all available non-allocated space. Most noteworthy was our observation that EFSTMPWP could on occasion take up so much space that the OS no longer had room to breath and would hang up. Windows wouldn't reload until EFSTMPWP was deleted by booting to a non-resident OS.

Our appetite whetted, we sought to observe the behavior of some of these utilities and compare the results. Though we document our results here, our greatest interest lies in understanding why these utilities produced the observed results. Toward that end, we'll begin with a brief overview of the Windows New Technology File System (NTFS) and then link the NTFS data structures to the disk residue.

NTFS VERSION 5

NTFS version 5 is the most common Windows file system in use today. It has some clever features that add efficiency, especially when it comes to file searching. The most important disk structure on an NTFS drive is the MFT (Master File Table), which is stored in the root directory of the volume as the file, "$MFT." The MFT is used to keep track of how the disk blocks are allocated and what is stored in them. The MFT records are themselves rather flexible, having resident and non-resident "attributes" based on the type of information being stored. For instance, a standard file would have attributes like "$STANDARD_INFORMATION,"

PAUL WILEY

Utilities and observed results.

Tool	Options	Residue
Cipher.exe	/w Setup: 1. Three-pass wipe, zero, 0xff, random Claims: n/a	1. Directory structures intact in $INDEX_ROOT 2. $MFT metadata timestamps intact 3. Alternate data stream information intact 4. Small files stored in $MFT intact
CyberScrub 3.5 TEST 1	Setup: 1. Quick wipe random pass 2. Wipe free space = ON 3. Wipe slack of existing files = ON 4. Scramble file and folder properties =OFF Claims: "This method leaves the entire disk surface filled with unclassified (random) information and no trace of the original data."	1. Directory structures intact in $INDEX_ROOT 2. $MFT metadata timestamps intact 3. Alternate data stream information intact 4. Small files stored in $MFT intact
CyberScrub 3.5 TEST 2	Setup: Same as TEST 1 except for 1. Scramble file and folder properties = ON	1. Content of small files stored in $MFT intact
PGP Shred	Setup: 1. Select files and folders in Windows Explorer 2. Right click and choose PGP Shred	1. File names changed to "0x00" 2. Alternate data stream names unmolested 3. Small files stored in $MFT intact
PGP Wipe	Setup: 1. Following PGP Shred (as above): a. Free space wipe b. Three-pass wipe c. Directory structure wipe	1. File names changed to "0x00" 2. Alternate data stream names 3. Small files stored in $MFT intact
Wincleaner Destroy-it! Pro v. 8.2.5	Setup: 1. Destruction Method set to "Adequately Secure" 2. Select Target Folders 3. Destroy slack files = ON 4. N.b. file name destruction not available for NTFS drives	1. Directory structures perturbed, but metadata remained left intact 2. File timestamps available 3. Folder timestamps available 4. Non-resident file content unchanged!
Evidence Eliminator	1. Drive = J:\ 2. Drive scanning enabled 3. High-Performance enabled a. Delete directory structure entries = ON b. Secure under-writing = OFF c. File slack erase = ON d. Attribute Scrambling: i. Scramble and randomize dates and times for files and folders = ON; ii. Randomize range (future) = 6 months iii. Randomize range (past) = 6 months e. Free space/hidden area wipe = ON f. Recycle bin wipe = ON	No discernable data residue

containing security identifiers, file access and modification times, owner identifiers, and so forth. A standard file will also have at least one "$FILE-NAME" attribute that is used to describe the filename (a second "$FILENAME" is not uncommon, used to represent the old DOS 8.3 convention). The actual file content is either stored in the MFT entry itself (if it's small enough) or referenced by the "$DATA" structure within the file's MFT entry. Storing a file within the MFT is indicated by the "resident" attribute; else $DATA points to the blocks where data is stored and the attribute is marked as non-resident.

Directories share $STANDARD_INFORMATION and $FILENAME attributes, but include additional structures like $INDEX_ROOT and $INDEX_ALLOCATION that make up the B-tree structure used to keep track of directory entries. The use of B-trees adds complexity to the identification of a deleted or moved file's parent directory since the entire B-tree is re-sorted every time a file is deleted. This usually overwrites the old reference to the deleted or moved file. In essence, $INDEX_ROOT and $INDEX_ALLOCATION are the structures that make disk organization at the file name layer possible.

Disk Wiping 101

While there are many variations on this theme, the preferred approach to disk wiping at this writing seems to involve the creation of a new file containing a

It is clear that most disk wipers leave behind a lot of telltale information that may have proprietary or security implications.

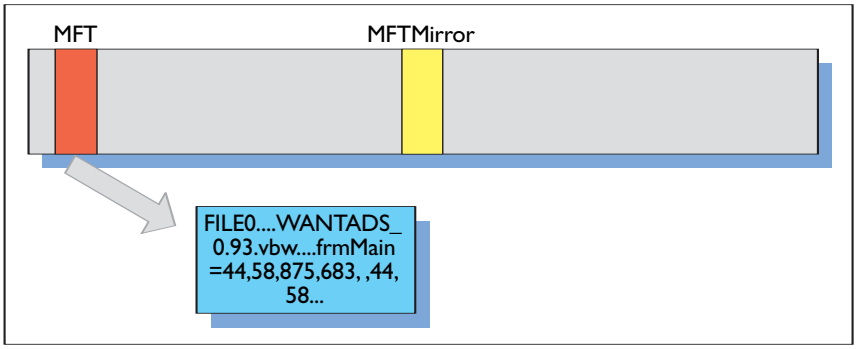

Figure 1. MFT residue.

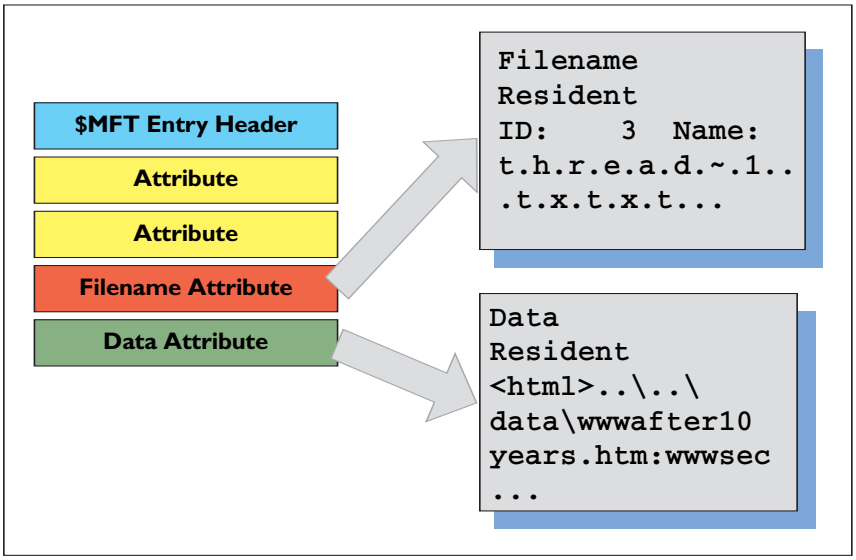

Figure 2. Filename and data residue.

wiping pattern (for example, all zeros, all ones, random zeroes and ones). This "single-file" approach takes advantage of the host file system for efficiency, because the "pattern" is created and applied to all available free disk space (including, most importantly, deleted files). This makes it unnecessary to deal with blocks, clusters, and sectors individually. We ignore partially allocated disk space (such as RAM slack and file slack) for the present purposes. The general theme is this: Windows doesn't delete files, it simply marks the physical space that the files occupied as unallocated and available for reuse. If a disk wiping utility obliterates all of the unallocated space it will, among other things, obliterate the space formerly occupied by deleted files. It's just that simple.

After the free space has been wiped, some utilities make an effort to scour through the $INDEX_ROOT and $INDEX_ALLOCATION of the directories to be sure that everything has been cleared out. For all but one of the products we tested, this is the extent of the wiping that is accomplished. Even after this second step, considerable data residue usually remains.

Recall the earlier discussion of the MFT structures. The directories themselves are simply constructs to allow for the user's organization of the items on the disk. The MFT entries connect the user level to the data level. While some of the wiping tools did seem to make changes to the contents of the deleted MFT entries, we found that most of these tools consistently failed to remove all information.

Figure 1 shows that we can clearly derive file names, both 8.3 and long (Unicode) file names. The tools, with one exception, make no effort to overwrite the old MFT entries. This is a problem for two reasons: first, the file and directory names are commonly indicative of content. One might infer from such information the nature of the business, the level of confidentiality, names, and so forth. Worse yet, if the files are small enough (if the resident attribute was set) $DATA will contain all of the original data (see Figure 2).

What went wrong? Remember that the disk wiping utilities typically wipe unallocated space. In order to be confident that this approach works completely, it is

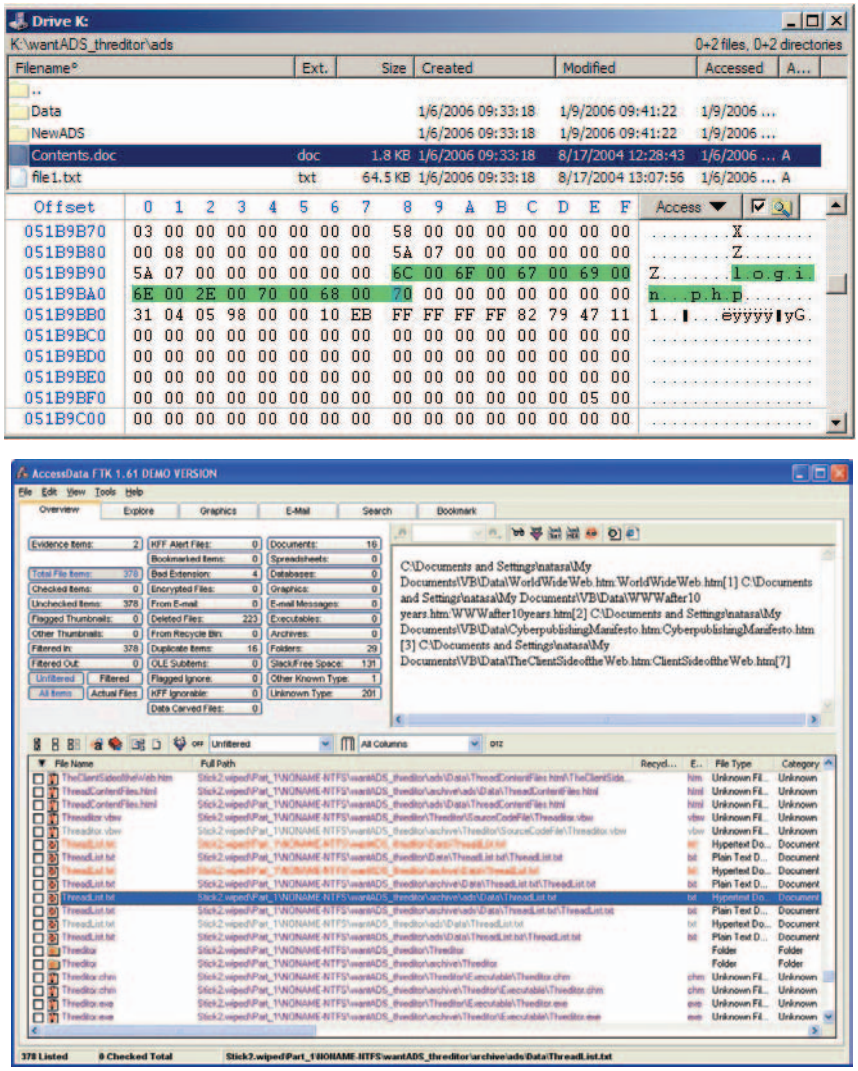

Figure 3a. Winhex 12.75 integrated with X-Ways Forensics. Note the presence of MFT metadata and persistent filenames.

Figure 3b. Access Data's Forensics Tool Kit 1.61. Note directory structure and file name residue.

incumbent on the user to determine exactly where the data is (allocated space, unallocated space, slack space, MFT resident). It is unreasonable to expect users to have that level of awareness. In this case the data is stored *within the $MFT and the $MFT_MIRROR*, which are allocated space. While there may be slack space associated with them, the area where MFT entries exist is clearly *not* slack space. For these reasons, most disk wiping utilities miss them.

The Experiment

To determine how some of the popular utilities handle disk sanitization, we copied a subdirectory from an NTFS disk to several NTFS-formatted, SanDisk 256MB memory cards. The subdirectory structure included files with alternate data streams (ADS), word processing documents, programs, and graphics. The test sequence involved:

1. Using Windows to erase all files and directories
2. Using a utility to wipe the memory card
3. Examination of the remnants, including:
   a. Directory entries
   b. Timestamps
   c. File contents

For post-erasure disk analysis, we used WinHex, Access Data's FTK, and a piece of custom C code (see URL Pearls). Figures 3a and 3b show two different perspectives on analyzing disk residue and illustrate the underlying digital forensics with two different forensics tools. A summary of our findings is shown in the table, which lists utilities and our observed results.

Conclusion

What we found is that only one product in our test environment, Evidence Eliminator, eliminated enough of the data to fall within our comfort zone. It is clear that most disk wipers leave behind a lot of telltale information that may have proprietary or security implications. Caveat emptor is appropriate here: disk wiping utilities (with the single excep-

URL PEARLS

For those bold enough to search your disks for residue yourself, a copy of the C code we created for this project is available online at www.cyber-defense.org/MFT_Extractor.php.

Cipher is a built-in core utility for Microsoft's Encrypted File System. As such, it's capability goes far beyond disk cleaning. Functional descriptions may be found by entering cipher /? from the command prompt. The /w option is the disk wiping parameter. For additional information on third-party vendors, see CyberScrub (www.cyberscrub.com),

Evidence Eliminator (www.evidence-eliminator.com), WinCleaner (www.wincleaner.com). PGP Shred and PGP Wipe are utilities within PGP Desktop Professional (www.pgp.com).

tion), especially including the built-ins, may leave enough metadata residue for an observer to tell a lot about you and your organization. And if the files are small enough, the entire files are left untouched.

We emphasize that these results must be taken in context. First and foremost, we limited our concern to data residue that could potentially be recovered with software. The reason for this is that hardware recovery is expensive enough to make casual snooping impractical. The use of sophisticated magnetic sensors and electron microscopes to recover erased data places most of the risk in the realm of governments and government agencies that may be more likely to use digital surveillance and real-time capture (such as Carnivore and Magic Lantern). We note that some of the disk wiping utilities we used did have features that purport to mitigate hardware recovery.

Second, we did not test all storage environments. For that reason, we provided the setup configuration settings so others may duplicate the experiments in their own environments. We predict that an NTFS file system on any medium will behave in a similar, but not necessarily identical, fashion. We have experienced one disk wipe on an 80GB external USB drive that produced more residue than we found on the memory cards, even with Evidence Eliminator. So a word of caution is appropriate.

Third, we didn't make any effort to clean the registry hive. "Messing with the registry is really dangerous," says Microsoft, and one is wise to take their word for it. Telltale residue is left behind in the registry without question—how valuable this information is to an onlooker is open to conjecture. Some vendors, such as Evidence Eliminator, encourage the use of registry cleaning tools such as Microsoft's own REGCLEAN, but our fear of turning our workstations into boat anchors disabused us of any temptation to run it. For those who are tempted, beware that REGCLEAN has been reported to cause as many problems as it fixes, and REGCLEAN only cleans HKEY_CLASSES_ROOT, which is not usually the most trouble-prone part of the registry. HKLM, for example, is unaffected by REGCLEAN. If that doesn't scare you away, consider that Microsoft no longer supports REGCLEAN.

Finally, there is another category of product that we didn't test: the so-called disk sanitizers or disk purgers. These are products that are marketed to people who intend to repurpose or recycle their computers. In the absence of empirical test results, our advice would be to favor those that claim compliance with appropriate government standards and receive high marks in trade reviews. C

HAL BERGHEL is associate dean of the Howard R. Hughes College of Engineering at the University of Nevada-Las Vegas, the director of the Center for Cybermedia Research (ccr.i2.nscee.edu), and co-director of the Identity Theft and Financial Fraud Research and Operations Center (www.itffroc.org).

DAVID HOELZER is a director of the Internet Forensics Lab at CCR and a faculty member of the SANS Institute. He is the owner of CyberDefense (www.cyber-defense.org), a computer security and forensics consultancy.