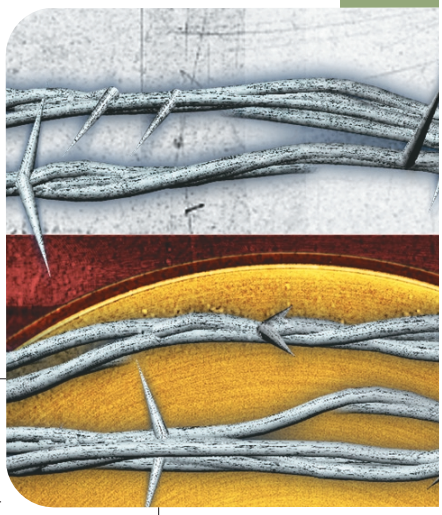
Remembrance of Data Passed: A Study of Disk Sanitization Practices



Many discarded hard drives contain information that is both confidential and recoverable, as the authors' own experiment shows. The availability of this information is little publicized, but awareness of it will surely spread.

SIMSON L. GARFINKEL AND ABHI SHELAT
Massachusetts Institute of Technology

A fundamental goal of information security is to design computer systems that prevent the unauthorized disclosure of confidential information. There are many ways to assure this information privacy. One of the oldest and most common techniques is physical isolation: keeping confidential data on computers that only authorized individuals can access. Most single-user personal computers, for example, contain information that is confidential to that user.

Computer systems used by people with varying authorization levels typically employ authentication, access control lists, and a privileged operating system to maintain information privacy. Much of information security research over the past 30 years has centered on improving authentication techniques and developing methods to assure that computer systems properly implement these access control rules.

Cryptography is another tool that can assure information privacy. Users can encrypt data as it is sent and decrypt it at the intended destination, using, for example, the secure sockets layer (SSL) encryption protocol. They can also encrypt information stored on a computer's disk so that the information is accessible only to those with the appropriate decryption key. Cryptographic file systems[1–3] ask for a password or key on startup, after which they automatically encrypt data as it's written to a disk and decrypt the data as it's read; if the disk is stolen, the data will be inaccessible to the thief. Yet despite the availability of cryptographic file systems, the general public rarely seems to use them.

Absent a cryptographic file system, confidential information is readily accessible when owners improperly retire their disk drives. In August 2002, for example, the United States Veterans Administration Medical Center in Indianapolis retired 139 computers. Some of these systems were donated to schools, while others were sold on the open market, and at least three ended up in a thrift shop where a journalist purchased them. Unfortunately, the VA neglected to *sanitize* the computer's hard drives—that is, it failed to remove the drives' confidential information. Many of the computers were later found to contain sensitive medical information, including the names of veterans with AIDS and mental health problems. The new owners also found 44 credit card numbers that the Indianapolis facility used.[4]

The VA fiasco is just one of many celebrated cases in which an organization entrusted with confidential information neglected to properly sanitize hard disks before disposing of computers. Other cases include:

- In the spring of 2002, the Pennsylvania Department of Labor and Industry sold a collection of computers to local resellers. The computers contained "thousands of files of information about state employees" that the department had failed to remove.[5]
- In August 2001, Dovebid auctioned off more than 100 computers from the San Francisco office of the Viant consulting firm. The hard drives contained confidential client information that Viant had failed to remove.[6]
- A Purdue University student purchased a used Macintosh computer at the school's surplus equipment exchange facility, only to discover that the computer's hard drive contained a FileMaker database containing the names and demographic information for more than 100 applicants to the school's Entomology Department.
- In August 1998, one of the authors purchased 10 used computer systems from a local computer store. The computers, most of which were three to five years old,

Table 1. Tbytes shipped per year on the global hard-disk market.
(Courtesy of IDC research)

YEAR	TBYTES SHIPPED
1992	7,900
1993	16,900
1994	33,000
1995	77,800
1996	155,900
1997	344,700
1998	698,600
1999	1,500,000
2000	3,200,000
2001	5,200,000
2002	8,500,000

contained all of their former owners' data. One computer had been a law firm's file server and contained privileged client–attorney information. Another computer had a database used by a community organization that provided mental health services. Other disks contained numerous personal files.

- In April 1997, a woman in Pahrump, Nevada, purchased a used IBM computer for $159 and discovered that it contained the prescription records of 2,000 patients who filled their prescriptions at Smitty's Supermarket pharmacy in Tempe, Arizona. Included were the patient's names, addresses and Social Security numbers and a list of all the medicines they'd purchased. The records included people with AIDS, alcoholism, and depression.[7]

These anecdotal reports are interesting because of their similarity and their relative scarcity. Clearly, confidential information has been disclosed through computers sold on the secondary market more than a few times. Why, then, have there been so few reports of unintended disclosure? We propose three hypotheses:

- Disclosures of this type are exceedingly rare
- Confidential information is disclosed so often on retired systems that such events are simply not newsworthy
- Used equipment is awash with confidential information, but nobody is looking for it—or else there are people looking, but they are not publicizing that fact

To further investigate the problem, we purchased more than 150 hard drives on the secondary market. Our goal was to determine what information they contained and what means, if any, the former owners had used to clean the drives before they discarded them. Here, we present our findings, along with our taxonomy for describing information recovered or recoverable from salvaged drives.

The hard drive market

Everyone knows that there has been a dramatic increase in disk-drive capacity and a corresponding decrease in mass-storage costs in recent years. Still, few people realize how truly staggering the numbers actually are. According to the market research firm Dataquest, nearly 150 million disk drives will be retired in 2002—up from 130 million in 2001. Although many such drives are destroyed, a significant number are repurposed to the secondary market. (This market is rapidly growing as a supply source for even mainstream businesses, as evidenced by the 15 October cover story in CIO Magazine, "Good Stuff Cheap: How to Use the Secondary Market to Your Enterprise's Advantage."[8])

According to the market research firm IDC, the worldwide disk-drive industry will ship between 210 and 215 million disk drives in 2002; the total storage of those disk drives will be 8.5 million terabytes (8,500 petabytes, or 8.5×10^{18} bytes). While Moore's Law dictates a doubling of integrated circuit transistors every 18 months, hard-disk storage capacity and the total number of bytes shipped are doubling at an even faster rate. Table 1 shows the terabytes shipped in the global hard-disk market over the past decade.

It's impossible to know how long any disk drive will remain in service; IDC estimates the typical drive's lifespan at five years. As Table 2 shows, Dataquest estimates that people will retire seven disk drives for every 10 that ship in the year 2002; this is up from a retirement rate of three for 10 in 1997 (see Figure 1). As the VA Hospital's experience demonstrates, many disk drives that are "retired" by one organization can appear elsewhere. Unless

Table 2. Global hard-disk market. (Courtesy of Dataquest)

YEAR	UNITS SHIPPED (IN THOUSANDS)	COST PER MEGABYTE TO END USER	RETIREMENTS (IN THOUSANDS)	RETIREMENT RATE* (IN PERCENT)
1997	128,331	0.1060	40,151	31.2
1998	143,927	0.0483	59,131	41.0
1999	174,455	0.0236	75,412	43.2
2000	199,590	0.0111	109,852	55.0
2001	195,601	0.0052	130,013	66.4
2002	212,507	0.0025	149,313	70.2

* ratio of drives retired to those shipped each year

retired drives are physically destroyed, poor information security practices can jeopardize information privacy.

The ubiquity of hard disks

Compared with other mass-storage media, hard disks pose special and significant problems in assuring long-term data confidentiality. One reason is that physical and electronic standards for other mass-storage devices have evolved rapidly and incompatibly over the years, while the Integrated Drive Electronics/Advanced Technology Attachment (IDE/ATA) and Small Computer System Interface (SCSI) interfaces have maintained both forward and backward compatibility. People use hard drives that are 10 years old with modern consumer computers by simply plugging them in: the physical, electrical, and logical standards have been remarkably stable.

This unprecedented level of compatibility has sustained both formal and informal secondary markets for used hard drives. This is not true of magnetic tapes, optical disks, flash memory, and other forms of mass storage, where there is considerably more diversity. With current devices, people typically cannot use older media due to format changes (a digital audio tape IV drive, for example, cannot read a DAT I tape, nor can a 3.5-inch disk drive read an 8-inch floppy.)

A second factor contributing to the problem of maintaining data confidentiality is the long-term consistency of file systems. Today's Windows, Macintosh, and Unix operating systems can transparently use the FAT16 and FAT32 file systems popularized by Microsoft in the 1980s and 1990s. (As we discuss in the "Sanitizing through Erasing" section, FAT stands for File Allocation Table and is a linked list of disk clusters that DOS uses to manage space on a random-access device; 16 or 32 refers to the sector numbers' bit length.) Thus, not only are 10-year-old hard drives mechanically and electrically compatible with today's computers, but the data they contain is readily accessible without special-purpose tools. This is not true with old tapes, which are typically written using proprietary backup systems, which might use proprietary compression and/or encryption algorithms as well.

A common way to sanitize a cartridge tape is to use a bulk tape eraser, which costs less than US$40 and can erase an entire tape in just a few seconds. Bulk erasers can erase practically any tape on the market. Once erased, a tape can be reused as if it were new. However, bulk erasers rarely work with hard disks, creating a third factor that complicates data confidentiality. In some cases, commercially available bulk erasers simply do not produce a sufficiently strong magnetic field to affect the disk surface. When they do, they almost always render the disk unusable: in addition to erasing user data, bulk erasers remove low-level track and formatting information. Although it might be possible to restore these formatting codes using vendor-specific commands, such commands are not generally available to users.

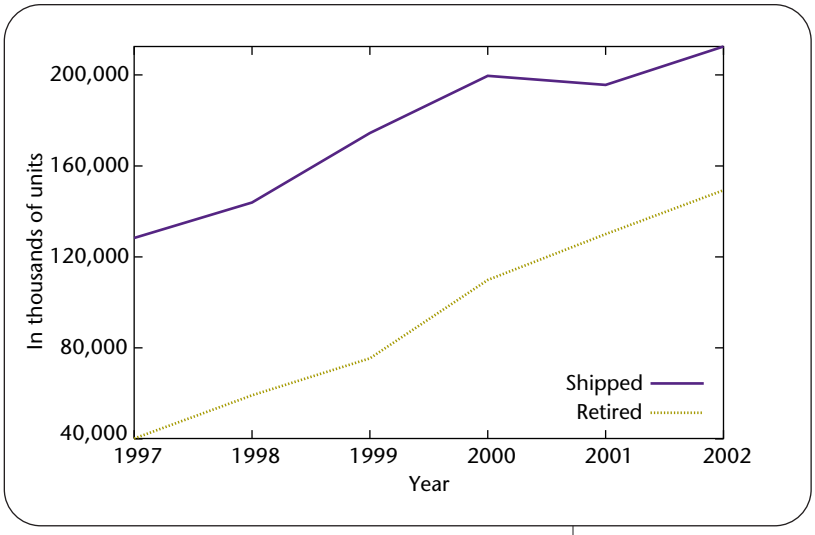


Figure 1. Worldwide hard-disk market in units shipped versus retired each year. (Courtesy of Dataquest)

The sanitization problem

Most techniques that people use to assure information privacy fail when data storage equipment is sold on the secondary market. For example, any protection that the computer's operating system offers is lost when someone removes the hard drive from the computer and installs it in a second system that can read the on-disk formats, but doesn't honor the access control lists. This vulnerability of confidential information left on information systems has been recognized since the 1960s.[9]

Legal protections that assure data confidentiality are similarly void. In *California v. Greenwood*, the US Supreme Court ruled that there is no right to privacy in discarded materials.[10] Likewise, it is unlikely that an individual or corporation could claim that either has a privacy or trade-secret interest in systems that they themselves have sold. Experience has shown that people routinely scavenge electronic components from the waste stream and reuse them without the original owner's knowledge.

Thus, to protect their privacy, individuals and organizations must remove confidential information from disk drives before they repurpose, retire, or dispose of them as intact units—that is, they must sanitize their drives.

The most common techniques for properly sanitizing hard drives include

- Physically destroying the drive, rendering it unusable
- Degaussing the drive to randomize the magnetic domains—most likely rendering the drive unusable in the process
- Overwriting the drive's data so that it cannot be recovered

Sanitizing is complicated by social norms. Clearly, the best way to assure that a drive's information is protected is to physically destroy the drive. But many people feel moral indignation when IT equipment is discarded and destroyed rather than redirected toward schools, commu-

nity organizations, religious groups, or lesser-developed nations where others might benefit from using the equipment—even if the equipment is a few years obsolete.

Sanitizing through erasing

Many people believe that they're actually destroying information when they erase computer files. In most cases, however, `delete` or `erase` commands do not actually remove the file's information from the hard disk. Although the precise notion of "erase" depends on the file system used, in most cases, deleting a file most often merely rewrites the metadata that pointed to the file, but leaves the disk blocks containing the file's contents intact.

Consider the FAT system, which was the dominant file format used in our study. There are four slightly different versions of this file system: FAT12, FAT16, VFAT, and FAT32. A hard disk is always addressed in terms of 512 byte sectors. A FAT file system further groups data sectors into *clusters*, which consist of 2^i sectors where i is a parameter set when the drive is formatted. Each hard-disk cluster has an entry in the FAT that describes its status. The cluster is either

- Part of a file, and points to the next cluster of that file
- The last cluster in a file, and thus holds a special end-of-file (EOF) value
- Free, and thus zero
- Marked defective

Essentially, the FAT is a linked list of clusters that correspond to files. (For a more comprehensive overview of the FAT file system, see Microsoft's specification.[11])

When the operating system erases a FAT file, two things occur. First, the system modifies the filename's first character in the file's directory entry to signal that the file has been deleted and that the directory entry can be recycled. Second, the system moves all of the file's FAT clusters to the hard drive's list of free clusters. The actual file data is never touched. Indeed, there are many programs available that can recover erased files, as we discuss later.

Although our semantic notion of "erasing" implies data removal, the FAT file system (and many other modern file systems) doesn't meet our expectations.

Sanitizing through overwriting

Because physical destruction is relatively complicated and unsatisfying, and because using the operating system to erase files does not effectively sanitize them, many individuals prefer to sanitize hard-drive information by intentionally overwriting that data with other data so that the original data cannot be recovered. Although overwriting is relatively easy to understand and to verify, it can be somewhat complicated in practice.

One way to overwrite a hard disk is to fill every addressable block with ASCII NUL bytes (zeroes). If the disk drive is functioning properly, then each of these blocks reports a block filled with NULs on read-back. We've observed this behavior in practice: for most home and business applications, simply filling an entire disk with ASCII NUL bytes provides sufficient sanitization.

One organization that has addressed the problem of sanitizing storage media is the US Department of Defense, which has created a "Cleaning and Sanitizing Matrix"[12] that gives DoD contractors three government-approved techniques for sanitizing rigid disk drives:

- Degauss with a Type I or Type II degausser
- Destroy by disintegrating, incinerating, pulverizing, shredding, or melting
- Overwrite all addressable locations with a random character, overwrite against with the character's complement, and then verify. (However, as the guidelines state—in all capital letters no less—this method is not approved for sanitizing media that contains top-secret information.)

The DoD's overwriting strategy is curious, both because it does not recommend writing a changing pattern, and because the method is specifically not approved for top-secret information. This omission and restriction is almost certainly intentional. Peter Gutmann, a computer security research at the University of Auckland who has studied this issue, notes: "The…problem with official data destruction standards is that the information in them may be partially inaccurate in an attempt to fool opposing intelligence agencies (which is probably why a great many guidelines on sanitizing media are classified)."[13]

Indeed, some researchers have repeatedly asserted that simple overwriting is insufficient to protect data from a determined attacker. In a highly influential 1996 article, Gutmann argues that it is theoretically possible to retrieve information written to any magnetic recording device because the disk platter's low-level magnetic field patterns are a function of both the written and overwritten data. As Gutmann explains, when a computer attempts to write a one or a zero to disk, the media records it as such, but the actual effect is closer to obtaining 1.05 when one overwrites with a one and 0.95 when a one overwrites a zero. Although normal disk circuitry will read both values as ones, "using specialized circuitry it is possible to work out what previous 'layers' contained."[13] Gutmann claims that "a high-quality digital sampling oscilloscope" or Magnetic Force Microscopy (MFM) can be used to retrieve the overwritten data. We refer to such techniques as exotic because they do not rely on the standard hard-disk interface.

Gutmann presents some 22 different patterns that you can write in sequence to a disk drive to minimize data recovery. In the eight years since the article was published, some sanitation tool developers (such as those on the WIPE project, for example[14]) have taken these "Gutmann patterns" as gospel, and have programmed their tools to

Table 3. A sanitization taxonomy.

LEVEL	WHERE FOUND	DESCRIPTION
Level 0	Regular files	Information contained in the file system. Includes file names, file attributes, and file contents. By definition, no attempts are made to sanitize Level 0 files information. Level 0 also includes information that is written to the disk as part of any sanitization attempt. For example, if a copy of Windows 95 had been installed on a hard drive in an attempt to sanitize the drive, then the files installed into the C:\WINDOWS directory would be considered Level 0 files. No special tools are required to retrieve Level 0 data.
Level 1	Temporary files	Temporary files, including print spooler files, browser cache files, files for "helper" applications, and recycle bin files. Most users either expect the system to automatically delete this data or are not even aware that it exists. Note: Level 0 files are a subset of Level 1 files. Experience has shown that it is useful to distinguish this subset, because many naive users will overlook Level 1 files when they are browsing a computer's hard drive to see if it contains sensitive information. No special tools are required to retrieve Level 1 data, although special training is required to teach the operator where to look.
Level 2	Deleted files	When a file is deleted from a file system, most operating systems do not overwrite the blocks on the hard disk that the file is written on. Instead, they simply remove the file's reference from the containing directory. The file's blocks are then placed on the free list. These files can be recovered using traditional "undelete" tools, such as Norton Utilities.
Level 3	Retained data blocks	Data that can be recovered from a disk, but which does not obviously belong to a named file. Level 3 data includes information in slack space, backing store for virtual memory, and Level 2 data that has been partially overwritten so that an entire file cannot be recovered. A common source of Level 3 data is disks that have been formatted with Windows `Format` command or the Unix `newfs` command. Even though the output of these commands might imply that they overwrite the entire hard drive, in fact they do not, and the vast majority of the formatted disk's information is recoverable with the proper tools. Level 3 data can be recovered using advanced data recovery tools that can "unformat" a disk drive or special-purpose forensics tools.
Level 4	Vendor-hidden data	This level consists of data blocks that can only be accessed using vendor-specific commands. This level includes the drive's controlling program and blocks used for bad-block management.
Level 5	Overwritten data	Many individuals maintain that information can be recovered from a hard drive even after it is overwritten. We reserve Level 5 for such information.

painstakingly use each pattern on every disk that is sanitized. Moreover, other organizations warn that failure to use these patterns or take other precautions, such as physically destroying a disk drive, means that "someone with technical knowledge and access to specialized equipment may be able to recover data from files deleted."[15]

But in fact, given the current generation of high-density disk drives, it's possible that none of these overwrite patterns are necessary—a point that Gutmann himself concedes. Older disk drives left some space between tracks; data written to a track could occasionally be recovered from this inter-track region using special instruments. Today's disk drives have a write head that is significantly larger than the read head: tracks are thus overlapping, and there is no longer any recoverable data "between" the tracks. Moreover, today's drives rely heavily on signal processing for their normal operation. Simply overwriting user data with one or two passes of random data is probably sufficient to render the overwritten information irrecoverable—a point that Gutmann makes in the updated version of the article, which appears on his Web site (www.cryptoapps.com/~peter/usenix01.pdf).

Indeed, there is some consensus among researchers that, for many applications, overwriting a disk with a few random passes will sufficiently sanitize it. An engineer at Maxtor, one of the world's largest disk-drive vendors, recently told us that recovering overwritten data as something akin "to UFO experiences. I believe that it is probably possible…but it is not going to be something that is readily available to anyone outside the National Security Agency."

A sanitization taxonomy

Modern computer hard drives contain an assortment of data, including an operating system, application programs, and user data stored in files. Drives also contain backing store for virtual memory, and operating system meta-information, such as directories, file attributes, and allocation tables. A block-by-block disk-drive examination also reveals remnants of previous files that were deleted but not completely overwritten. These remnants are sometimes called *free space*, and include bytes at the end of partially filled directory blocks (sometimes called *slack space*), startup software that is not strictly part of the operating system (such as boot blocks), and virgin blocks that were initialized at the factory but never written. Finally, drives also contain blocks that are not accessible through the standard IDE/ATA or SCSI interface, including internal drive blocks used for bad-block management and for holding the drive's own embedded software.

To describe data found on recovered disk drives and facilitate discussion of sanitization practices and forensic analysis, we created a *sanitization taxonomy* (see Table 3).

Table 4. A sampling of free and commercially available sanitization tools.

PROGRAM	COST	PLATFORM	COMMENTS
AutoClave http://staff.washington.edu/jdlarios/autoclave	Free	Self-booting PC disk	Writes just zeroes, DoD specs, or the Gutmann patterns. Very convenient and easy to use. Erases the entire disk including all slack and swap space.
CyberScrub www.cyberscrub.com	$39.95	Windows	Erases files, folders, cookies, or an entire drive. Implements Gutmann patterns.
DataScrubber www.datadev.com/ds100.html	$1,695	Windows, Unix	Handles SCSI remapping and swap area. Claims to be developed in collaboration with the US Air Force Information Welfare Center.
DataGone www.powerquest.com	$90	Windows	Erases data from hard disks and removable media. Supports multiple overwriting patterns.
Eraser www.heidi.ie/eraser	Free	Windows	Erases directory metadata. Sanitizes Windows swap file when run from DOS. Sanitizes slack space by creating huge temporary files.
OnTrack DataEraser www.ontrack.com/dataeraser	$30–$500	Self-booting PC disk	Erases partitions, directories, boot records, and so on. Includes DoD specs in professional version only.
SecureClean www.lat.com	$49.95	Windows	Securely erases individual files, temporary files, slack space, and so on.
Unishred Pro www.accessdata.com	$450	Unix and PC hardware	Understands some vendor-specific commands used for bad-block management on SCSI drives. Optionally verifies writes. Implements all relevant DoD standards and allows custom patterns.
Wipe http://wipe.sourceforge.net	Free	Linux	Uses Gutmann's erase patterns. Erases single files and accompanying metadata or entire disks.
WipeDrive www.accessdata.com	$39.95	Bootable PC disk	Securely erases IDE and SCSI drives.
Wiperaser XP www.liveye.com/wiperaser	$24.95	Windows	Erases cookies, history, cache, temporary files, and so on. Graphical user interface.

Sanitization tools

Many existing programs claim to properly sanitize a hard drive, including $1,695 commercial offerings that boast government certifications, more than 50 tools licensed for a single computer system, and free software/open-source products that seem to offer largely the same features. Broadly speaking, two kinds of sanitization programs are available: disk sanitizers and declassifiers, and slack-space sanitizers.

Disk sanitizers and declassifiers aim to erase all user data from a disk before it's disposed of or repurposed in an organization. Because overwriting an operating system's boot disk information typically causes the computer to crash, disk sanitizers rarely operate on the boot disk of a modern operating system. Instead, they're usually run under an unprotected operating system, such as DOS, or as standalone applications run directly from bootable media (floppy disks or CD-ROMs). (It's relatively easy to sanitize a hard disk that is not the boot disk. With Unix, for example, you can sanitize a hard disk with the device `/dev/hda` using the command `dd if=/dev/zero of=/dev/hda`.) Using our taxonomy, disk sanitizers seek to erase all of the drive's Level 1, 2, 3, and 5 information. Sanitizers equipped with knowledge of vendor-specific disk-drive commands can erase Level 4 information as well.

Slack space sanitizers sanitize disk blocks (and portions of disk blocks) that are not part of any file and do not contain valid file system meta-information. For example, if a 512-byte block holds a file's last 100 bytes and nothing else, a slack-space sanitizer reads the block, leaves bytes 1–100 untouched, and zeros bytes 101–512. Slack-space sanitizers also compact directories (removing ignored entries), and overwrite blocks on the free list. Many of these programs also remove temporary files, history files, browser cookies, deleted email, and so on. Using our taxonomy, slack-space sanitizers seek to erase all Level 1 through Level 4 drive information, while leaving Level 0 information intact.

Table 4 offers a few examples of free and commercially available sanitation tools; a complete list is available at www.fortunecity.com/skyscraper/true/882/Comparison_Shredders.htm.

Forensic tools

The flip side of sanitization tools are forensic analysis tools, which are used for recovering hard-disk information. Forensic tools are harder to write than sanitization tools and, not surprisingly, fewer of these tools are available. Many of the packages that do exist are tailored to law enforcement agencies. Table 5 shows a partial list of forensic tools.

Almost all forensic tools let users analyze hard disks or

Table 5. Forensics programs.

PROGRAM	COST	PLATFORM	COMMENTS
DriveSpy www.digitalintel.com	$200–$250	DOS/Windows	Inspects slack space and deleted file metadata.
EnCase www.guidancesoftware.com	$2,495	Windows	Features sophisticated drive imaging and preview modes, error checking, and validation, along with searching, browsing, time line, and registry viewer. Graphical user interface. Includes hash analysis for classifying known files.
Forensic Toolkit www.accessdata.com	$595	Windows	Graphic search and preview of forensic information, including searches for JPEG images and Internet text.
ILook www.ilook-forensics.org	N/A	Windows	Handles dozens of file systems. Explorer interface to deleted files. Generates hashes of files. Filtering functionality. This tool only available to the US government and law enforcement agencies.
Norton Utilities www.symantec.com	$49.95	Windows	Contains tools useful for recovering deleted files and sector-by-sector examination of a computer's hard disk.
The Coroner's Toolkit www.porcupine.org/forensicsl/tct.htm	Free	Unix	A collection of programs used for performing post-mortem forensic analysis of Unix disks after a break-in.
TASK http://atstake.com/research/tools/task	Free	Unix	Operates on disk images created with `dd`. Handles FAT, FAT32, toolkit. Analyzes deleted files and slack space, and includes time-line NTFS, Novel, Unix, and other disk formats. Built on Coroner's Toolkit.

hard-disk images from a variety of different operating systems and provide an Explorer-style interface so you can read the files. Tools are of course limited by the original computer's operating system, as different systems overwrite different amounts of data or metadata when they delete a file or format a disk. Nevertheless, many of these forensic tools can find "undeleted" files (Level 2 data) and display hard-drive information that is no longer associated with a specific file (Level 3 data). Most tools also offer varying search capabilities. Hence, an operator can search an entire disk image for keywords or patterns, and then display the files (deleted or otherwise) containing the search pattern.

Programs tailored to law enforcement also offer to log every keystroke an operator makes during the hard-drive inspection process. This feature supposedly prevents evidence tampering.

O sanitization, where art thou?

Despite the ready availability of sanitization tools and the obvious threat posed by tools that provide forensic analysis, there are persistent reports that some systems containing confidential information are being sold on the secondary market.

We propose several possible explanations for this state of affairs:

- *Lack of knowledge.* The individual (or organization) disposing of the device simply fails to consider the problem (they might, for example, lack training or time).
- *Lack of concern for the problem.* The individual considers the problem, but does not think the device actually contains confidential information.
- *Lack of concern for the data.* The individual is aware of the problem—that the drive might contain confidential information—but doesn't care if the data is revealed.
- *Failure to properly estimate the risk.* The individual is aware of the problem, but doesn't believe that the device's future owner will reveal the information (that is, the individual assumes that the device's new owner will use the drive to store information, and won't rummage around looking for what the previous owner left behind).
- *Despair.* The individual is aware of the problem, but doesn't think it can be solved.
- *Lack of tools.* The individual is aware of the problem, but doesn't have the tools to properly sanitize the device.
- *Lack of training or incompetence.* The individual attempts to sanitize the device, but the attempts are ineffectual.
- *Tool error.* The individual uses a tool, but it doesn't behave as advertised. (Early versions of the Linux `wipe` command, for example, have had numerous bugs which resulted in data not being actually overwritten. Version 0.13, for instance, did not erase half the data in the file due to a bug; see http://packages.debian.org/unstable/utils/wipe.html)
- *Hardware failure.* The computer housing the hard drive might be broken, making it impossible to sanitize the hard drive without removing it and installing it in another computer—a time-consuming process. Alternatively, a computer failure might make it seem that the hard drive has also failed, when in fact it has not.

Among nonexpert users—especially those using the DOS or Windows operating systems—lack of training might be the primary factor in poor sanitization practices.

Among expert users, we posit a different explanation: they are aware that the Windows `format` command does not actually overwrite a disk's contents. Paradoxically, the media's fascination with exotic methods for data recovery might have decreased sanitization among these users by making it seem too onerous. In repeated interviews, users frequently say things like: 'The FBI or the NSA can always get the data back if they want, so why bother cleaning the disk in the first place?" Some individuals fail to employ even rudimentary sanitization practices because of these unsubstantiated fears. This reasoning is flawed, of course, because most users should be concerned with protecting their data from more pedestrian attackers, rather than from US law enforcement and intelligence agencies. Even if these organizations do represent a threat to some users, today's readily available sanitization tools can nevertheless protect their data from other credible threats.

However interesting they might be, informal interviews and occasional media reports are insufficient to gauge current sanitization practices. To do that, we had to acquire numerous disk drives and actually see what data their former owners left behind.

Our experiment

We acquired 158 hard drives on the secondary market between November 2000 and August 2002. We purchased drives from several sources: computer stores specializing in used merchandise, small businesses selling lots of two to five drives, and consolidators selling lots of 10 to 20 drives. We purchased most of the bulk hard drives by winning auctions at the eBay online auction service.

As is frequently the case with secondary-market equipment, the drives varied in manufacturer, size, date of manufacture, and condition. A significant fraction of the drives were physically damaged, contained unreadable sectors, or were completely inoperable.

Because we were interested in each drive's data, rather than its physical deterioration, our goal was to minimize drive handling as much as possible. Upon receipt, we recorded each drive's physical characteristics and source in a database. We then attached the drives to a workstation running the FreeBSD 4.4 operating system, and then copied the drive's contents block-by-block—using the Unix `dd` command from the raw ATA device—into a disk file we called the "image file." Once we completed this imaging operation, we attempted to mount each drive using several file systems: FreeBSD, MS DOS, Windows NT File System, Unix File System, and Novell file systems. If we successfully mounted the drive, we used the Unix `tar` command to transverse the entire file system hierarchy and copy the files into compressed tar files. These files are exactly equal to our taxonomy's Level 0 and Level 1 files.

We then analyzed the data using a variety of tools that we wrote specifically for this project. In particular, we stored the complete path name, length, and an MD5 cryptographic checksum of every Level 0 and Level 1 file in a database. (MD5 is a one-way function that reduces a block of data to a 128-bit electronic "fingerprint" that can be used for verifying file integrity.) We can run queries against this database for reporting on the incidence of these files. In the future, we plan to identify the files' uniqueness by looking for MD5 collisions and by comparing our database against a database of MD5 codes for commercial software that the National Institute of Standards and Technology is assembling.[16]

To ease analysis, we are also creating a "forensic file system," a kind of semantic file system first proposed by Gifford and colleagues.[17] The FFS lets us view and act on forensic information using traditional Unix file system tools such as ls, more, grep, and strings. For example, in the FFS, a directory listing shows both normal and deleted files; it modifies deleted file names to prevent name collisions and to indicate if the file's contents are not recoverable, partially recoverable, or fully recoverable. (The difficulty of forensic analysis depends highly on the operating system used to create the target file system; in particular, it is much easier to undelete files on FAT-formatted disks than on most Unix file systems.)

Initial findings

We acquired a total of 75 Gbytes of data, consisting of 71 Gbytes of uncompressed disk images and 3.7 Gbytes of compressed tar files.

From the beginning, one of the most intriguing aspects of this project was the variation in the disk drives. When we briefed people on our initial project plans, many responded by saying that they were positive that the vast majority of the drives collected would be *X*, and the value of *X* varied depending on speaker. For example, some people were "positive" that all the recovered drives would contain active file systems, while others were sure that all of the drives would be reformatted. Some were certain we'd find data, but that it would be too old to be meaningful, and others were sure that nearly all of the drives would be properly sanitized, "because nobody could be so stupid as to discard a drive containing active data."

File system analysis

The results of even this limited, initial analysis indicate that there are no standard practices in the industry. Of the 129 drives that we successfully imaged, only 12 (9 percent) had been properly sanitized by having their sectors completely overwritten with zero-filled blocks; 83 drives (64 percent) contained mountable FAT16 or FAT32 file systems. (All the drives we collected had ei-

Table 6. Recoverable Level 0 and 1 files by type.

FILE TYPE	NUMBER FOUND	ON DRIVES	MAX FILES PER DRIVE
Microsoft Word (DOC)	675	23	183
Outlook (PST)	20	6	12
Microsoft PowerPoint (PPT)	566	14	196
Microsoft Write (WRI)	99	21	19
Microsoft Works (WKS)	68	1	68
Microsoft Excel (XLS)	274	18	67

ther FAT16 or FAT32 file systems.) Another 46 drives did not have mountable file systems.

Of the 83 drives with mountable file systems, 51 appeared to have been freshly formatted—that is, they either had no files or else the files were created by the DOS `format c:/s` command; another six drives were formatted and had a copy of DOS or Windows 3.1 installed. Of these 51 drives, 19 had recoverable Level 3 data—indicating that the drives had been formatted after they had been used in another application.

Of the 46 drives we could not mount, 30 had more than a thousand sectors of recoverable Level 3 information. Many of these drives had recoverable FAT directory entries as well.

Document file analysis

We performed limited analysis of the mountable file systems to determine the type of documents left on the drives. Table 6 summarizes these results.

Overall, the 28 drives with active file systems contained comparatively few document files—far fewer than we'd expect to find on actively used personal computers. We believe that this is because the drives' previous owners intentionally deleted these files in an attempt to at least partially sanitize the drives before disposing of them.

To test this theory, we wrote a program that lets us scan FAT16 and FAT32 images for deleted files and directories. Using this program, we can scan the disks for data that was presumably deleted by the drive's original owner prior to disposing of the drive. The results are illuminating: with the exception of the cleared disks (all blocks zeroed), practically every disk had significant numbers of deleted directories and files that are recoverable. Even the 28 disks with many undeleted files contained significant numbers of deleted-but-recoverable directories and files as well. A close examination of the deleted files indicates that, in general, users deleted data files, but left application files intact.

Recovered data

Currently, we can use the `tar` files to recover Level 0 and Level 1 files. Some of the information we found in these files included:

- Corporate memoranda pertaining to personnel issues
- A letter to the doctor of a 7-year-old child from the child's father, complaining that the treatment for the child's cancer was unsatisfactory
- Fax templates for a California children's hospital (we expect that additional analysis of this drive will yield medically sensitive information)
- Love letters
- Pornography

Using slightly more sophisticated techniques, we wrote a program that scans for credit card numbers. The program searches for strings of numerals (with possible space and dash delimiters) that pass the mod-10 check-digit test required of all credit card numbers, and that also fall within a credit card number's feasible numerical range. For example, no major credit card number begins with an eight.

In our study, 42 drives had numbers that passed these tests. Determining whether a number is actually a valid credit card number requires an attempted transaction on the credit card network. Rather than do this, we inspected the number's context. Two drives contained consistent financial-style log files. One of these drives (#134) contained 2,868 numbers in a log format. Upon further inspection, it appeared that this hard drive was most likely used in an ATM machine in Illinois, and that no effort was made to remove any of the drive's financial information. The log contained account numbers, dates of access, and account balances. In addition, the hard drive had all of the ATM machine software. Although the drive also contained procedures and software to change the ATM's DES key (which presumably secures transactions between the ATM and the financial network), the actual DES key is apparently stored in a hardware chip in the ATM machine.

Another drive (#21) contained 3,722 credit card numbers (some of them repeated) in a different type of log

Table 7. Disk formatting results.

DISK SIZE	BLOCKS	BLOCKS ALTERED BY WINDOWS 98	BLOCKS ALTERED BY WINDOWS 98
		Fdisk command	Format command
10 GBytes	20,044,160	2.563 (0.01 percent)	21,541 (0.11 percent)

format. The files on this drive appeared to have been erased, and the drive was formatted. Yet another drive (#105) contained 39 credit card numbers in a database file that included the correct type of credit card, and still another (#133) had a credit card number in a cached Web page URL. The URL is a 'GET'-type HTTP form that was submitted to an e-commerce site; it contained all of the address and expiration information necessary to execute an e-commerce transaction. Finally, another drive (#40) had 21 credit card numbers in a file.

We also wrote a program that searches for RFC mail headers. Of the 129 drives analyzed, 66 drives had more than five email messages. We use this threshold because some programs, such as Netscape Navigator, include a few welcome emails upon installation. One drive in our batch contained almost 9,500 email messages, dated from 1999 through 2001. In all, 17 drives had more than 100 email messages and roughly 20 drives had between 20 and 100 email messages. During this analysis, we only investigated the messages' subject headers; contents seemed to vary from typical spam to grievances about retroactive pay.

Understanding DOS format

It's not clear if the 52 formatted drives were formatted to sanitize the data or if they were formatted to determine their condition and value for sale on the secondary market.

In many interviews, users said that they believed DOS and Windows `format` commands would properly remove all hard drive data. This belief seems reasonable, as the DOS and Windows `format` commands specifically warn users that "ALL DATA ON NON-REMOVABLE DISK DRIVE C: WILL BE LOST" when a computer is booted from floppy and the user attempts a `format C:` command. This warning might rightly be seen as a *promise* that using the `format` command will in fact remove all of the disk drive's data.

Many users were surprised when we told them that the `format` command does not erase all of the disk's information. As our taxonomy indicates, most operating system format commands only write a minimal disk file system; they do not rewrite the entire disk. To illustrate this assertion, we took a 10-Gbyte hard disk and filled every block with a known pattern. We then initialized a disk partition using the Windows 98 `FDISK` command and formatted the disk with the `format` command. After each step, we examined the disk to determine the number of blocks that had been written. Table 7 shows the results.

Users might find these numbers discouraging: despite warnings from the operating system to the contrary, the `format` command overwrites barely more than 0.1 percent of the disk's data. Nevertheless, the command takes more than eight minutes to do its job on the 10-Gbyte disk—giving the impression that the computer is actually overwriting the data. In fact, the computer is attempting to *read* all of the drive's data so it can build a bad-block table. The only blocks that are actually written during the `format` process are those that correspond to the boot blocks, the root directory, the file allocation table, and a few test sectors scattered throughout the drive's surface.

Although 158 disk drives might seem like a lot, it's a tiny number compared to the number of disk drives that are sold, repurposed, and discarded each year. As a result, our findings and statistics are necessarily qualitative, not quantitative. Nevertheless, we can draw a few conclusions.

First, people can remove confidential information from disk drives before they discard, repurpose, or sell them on the secondary market. Moreover, freely available tools make disk sanitization easy.

Second, the current definition of "medical records" might not be broad enough to cover the range of medically sensitive information in the home and work environment. For example, we found personal letters containing medically sensitive information on a computer that previously belonged to a software company. Many routine email messages also contain medically sensitive information that should not be disclosed. If an employee sends a message to his boss saying that he'll miss a meeting because he has a specific problem requiring a doctor visit, for example, he has created a record of his medical condition in the corporate email system.

Third, our study indicates that the secondary hard-disk market is almost certainly awash in information that is both sensitive and confidential.

Based on our findings, we make the following recommendations:

- Users must be educated about the proper techniques for sanitizing disk drives.
- Organizations must adopt policies for properly sanitizing drives on computer systems and storage media that are sold, destroyed, or repurposed.
- Operating system vendors should include system tools

that securely delete files, and clear slack space and entire disk drives.
- Future operating systems should be capable of automatically sanitizing deleted files. They should also be equipped with background processes that automatically sanitize disk sectors that the operating system is not currently using.
- Vendors should encourage the use of encrypting file systems to minimize the data sanitization problem.
- Disk-drive vendors should equip their drives with tools for rapidly or even instantaneously removing all disk-drive information. For example, they could equip a disk drive with a cryptographic subsystem that automatically encrypts every disk block when the block is written, and decrypts the block when it is read back. Users could then render the drive's contents unintelligible by securely erasing the key.[18]

With several months of work and relatively little financial expenditure, we were able to retrieve thousands of credit card numbers and extraordinarily personal information on many individuals. We believe that the lack of media reports about this problem is simply because, at this point, few people are looking to repurposed hard drives for confidential material. If sanitization practices are not significantly improved, it's only a matter of time before the confidential information on repurposed hard drives is exploited by individuals and organizations that would do us harm. □

Acknowledgments

Many MIT students and faculty members provided useful comments and insights on this project. We specifically thank professors David Clark and Ron Rivest for their continuing support, suggestions, and comments on previous drafts of this article. Professors Hal Abelson and Charles Leiserson have also been a source of encouragement and moral support. We received helpful comments on previous drafts of this paper from Brian Carrier, Peter Gutmann, Rich Mahn, Eric Thompson, and Wietse Venema.

References

1. Network Associates, *PGP Windows 95, 98 and NT User's Guide, Version 6.0*. 1998; version 6.02 includes the pgpdisk encrypted file system and is available for download at www.pgpi.org/products/pgpdisk.
2. M. Blaze, "A Cryptographic File System for Unix," *1st ACM Conf. Comm. and Computing Security,* ACM Press, New York, 1993, pp. 9–16.
3. Microsoft, "Encrypting File System for Windows 2000," www.microsoft.com/windows2000/techinfo/howitworks/security/encrypt.asp.
4. J. Hasson, "V.A. Toughens Security after PC Disposal Blunders," *Federal Computer Week*, 26 Aug. 2002; www.fcw.com/fcw/articles/2002/0826/news-va-08-26-02.asp.
5. M. Villano, "Hard-Drive Magic: Making Data Disappear Forever," *New York Times*, 2 May 2002.
6. J. Lyman, "Troubled Dot-Coms May Expose Confidential Client Data," *NewsFactor Network*, 8 Aug. 2001; www.newsfactor.com/perl/story/12612.html.
7. J. Markoff, "Patient Files Turn Up in Used Computer," *New York Times*, 4 Apr. 1997.
8. S. Berinato, "Good Stuff Cheap," *CIO*, 15 Oct. 2002, pp. 53–59.
9. National Computer Security Center, "A Guide to Understanding Dataremanence in Automated Information Systems," Library No. 5-236,082, 1991, NCSC-TG-025; www.radium.ncsc.mil/tpep/library/rainbow/NCSC-TG-028.ps
10. *California v. Greenwood*, 486 US 35, 16 May 1988.
11. Microsoft, "Microsoft Extensible Firmware Initiative FAT32 File System Specification," 6 Dec. 2000; www.microsoft.com/hwdev/download/hardware/fatgen103.pdf.
12. US Department of Defense, "Cleaning and Sanitization Matrix," DOS 5220.22-M, Washington, D.C., 1995; www.dss.mil/isec/nispom_0195.htm.
13. P. Gutmann, "Secure Deletion of Data from Magnetic and Solid-State Memory," *Proc. Sixth Usenix Security Symp.*, Usenix Assoc., 1996; www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html.
14. T. Vier, "Wipe 2.1.0," 14 Aug. 2002; http://sourceforge.net/projects/wipe.
15. D. Millar, "Clean Out Old Computers Before Selling/Donating," June 1997; www.upenn.edu/computing/security/advisories/old computers.html.
16. National Institute of Standards and Technology, "National Software Reference Library Reference Data Set"; www.nsrl.nist.gov.
17. D.K. Gifford et al., "Semantic File Systems," *Proc. 13th ACM Symp. on Operating Systems Principles,* ACM Press, 1991, pp. 16–25.
18. G. Di Crescenzo et al., "How to Forget a Secret," *Symposium Theoretical Aspects in Computer Science* (STACS 99), Lecture Notes in Computer Science, Springer-Verlag, Berlin, 1999, pp. 500–509.

Simson L. Garfinkel *is a graduate student in both the Cryptography and Information Security Group and the Advanced Network Architecture Group at MIT's Laboratory for Computer Science. Garfinkel is the author of many books on computer security and policy, including* Database Nation: the Death of Privacy in the 21st Century (O'Reilly, 2000) *and coauthor of* Practical UNIX and Internet Security (O'Reilly, 2003)*. His research interests currently focus on the intersection of security technology and usability. Contact him at simsong@lcs.mit.edu; http://simson.net.*

Abhi Shelat *is a graduate student in the Theory of Computation Group at the Massachusetts Institute of Technology. His research interests include computer security, algorithms, and data compression. He also enjoys taking photos and building furniture. Contact him at abhi@lcs.mit.edu; http://theory.lcs.mit.edu/~abhi.*